Rezoluce v predikátové logice I. řádu

(pokračování)

Formule

- **literál** l
  - **pozitivní literál** = atomická formule $p(t_1, \cdots, t_n)$
  - **negativní literál** = negace atomické formule $\neg p(t_1, \cdots, t_n)$

Formule

- **literál** l
  - **pozitivní literál** = atomická formule $p(t_1, \cdots, t_n)$
  - **negativní literál** = negace atomické formule $\neg p(t_1, \cdots, t_n)$
- **klauzule** C = konečná množina literálů reprezentující jejich disjunkci
  - příklad: $p(X) \vee q(a, f) \vee \neg p(Y)$ notace: $\{p(X), q(a, f), \neg p(Y)\}$
  - **klauzule je pravdivá** $\Longleftrightarrow$ je pravdivý alespoň jeden z jejích literálů
  - **prázdná klauzule** se značí $\square$ a je vždy nepravdivá (neexistuje v ní pravdivý literál)

Formule

- **literál** l
  - **pozitivní literál** = atomická formule $p(t_1, \cdots, t_n)$
  - **negativní literál** = negace atomické formule $\neg p(t_1, \cdots, t_n)$
- **klauzule** C = konečná množina literálů reprezentující jejich disjunkci
  - příklad: $p(X) \vee q(a, f) \vee \neg p(Y)$ notace: $\{p(X), q(a, f), \neg p(Y)\}$
  - **klauzule je pravdivá** $\Longleftrightarrow$ je pravdivý alespoň jeden z jejích literálů
  - **prázdná klauzule** se značí $\square$ a je vždy nepravdivá (neexistuje v ní pravdivý literál)
- **formule** F = množina klauzulí reprezentující jejich konjunkci
  - formule je v tzv. konjuktivní normální formě (konjunkce disjunkcí)
  - příklad: $(p \vee q) \wedge (\neg p) \wedge (p \vee \neg q \vee r)$ notace: $\{\{p, q\}, \{\neg p\}, \{p, \neg q, r\}\}$

Formule

- **literál** l
  - **pozitivní literál** = atomická formule $p(t_1, \cdots, t_n)$
  - **negativní literál** = negace atomické formule $\neg p(t_1, \cdots, t_n)$
- **klauzule** C = konečná množina literálů reprezentující jejich disjunkci
  - příklad: $p(X) \vee q(a, f) \vee \neg p(Y)$ notace: $\{p(X), q(a, f), \neg p(Y)\}$
  - **klauzule je pravdivá** $\Longleftrightarrow$ je pravdivý alespoň jeden z jejích literálů
  - **prázdná klauzule** se značí $\square$ a je vždy nepravdivá (neexistuje v ní pravdivý literál)
- **formule** F = množina klauzulí reprezentující jejich konjunkci
  - formule je v tzv. konjuktivní normální formě (konjunkce disjunkcí)
  - příklad: $(p \vee q) \wedge (\neg p) \wedge (p \vee \neg q \vee r)$ notace: $\{\{p, q\}, \{\neg p\}, \{p, \neg q, r\}\}$
  - **formule je pravdivá** $\Longleftrightarrow$ všechny klauzule jsou pravdivé
  - prázdná formule je vždy pravdivá (neexistuje klauzule, která by byla nepravdivá)

Formule

- **literál** l
  - **pozitivní literál** = atomická formule $p(t_1, \cdots, t_n)$
  - **negativní literál** = negace atomické formule $\neg p(t_1, \cdots, t_n)$
- **klauzule** C = konečná množina literálů reprezentující jejich disjunkci
  - příklad: $p(X) \vee q(a, f) \vee \neg p(Y)$ notace: $\{p(X), q(a, f), \neg p(Y)\}$
  - **klauzule je pravdivá** $\Longleftrightarrow$ je pravdivý alespoň jeden z jejích literálů
  - **prázdná klauzule** se značí $\square$ a je vždy nepravdivá (neexistuje v ní pravdivý literál)
- **formule** F = množina klauzulí reprezentující jejich konjunkci
  - formule je v tzv. konjuktivní normální formě (konjunkce disjunkcí)
  - příklad: $(p \vee q) \wedge (\neg p) \wedge (p \vee \neg q \vee r)$ notace: $\{\{p, q\}, \{\neg p\}, \{p, \neg q, r\}\}$
  - **formule je pravdivá** $\Longleftrightarrow$ všechny klauzule jsou pravdivé
  - prázdná formule je vždy pravdivá (neexistuje klauzule, která by byla nepravdivá)
- **množinová notace:** literál je prvek klauzule, klauzule je prvek formule, . . .

Rezoluční princip ve výrokové logice

- **Rezoluční princip** = pravidlo, které umožňuje odvodit z klauzulí $C_1 \cup \{l\}$ a $\{\neg l\} \cup C_2$ klauzuli $C_1 \cup C_2$

$$\frac{C_1 \cup \{l\} \qquad\qquad \{\neg l\} \cup C_2}{C_1 \cup C_2}$$

- $C_1 \cup C_2$ se nazývá **rezolventou** původních klauzulí

Rezoluční princip ve výrokové logice

- **Rezoluční princip** = pravidlo, které umožňuje odvodit z klauzulí $C_1 \cup \{l\}$ a $\{\neg l\} \cup C_2$ klauzuli $C_1 \cup C_2$

$$\frac{C_1 \cup \{l\} \qquad\qquad \{\neg l\} \cup C_2}{C_1 \cup C_2}$$

- $C_1 \cup C_2$ se nazývá **rezolventou** původních klauzulí
- příklad:

$$\frac{\{p, r\} \qquad\qquad \{\neg r, s\}}{\{p, s\}} \qquad\qquad \frac{(p \vee r) \wedge (\neg r \vee s)}{p \vee s}$$

Rezoluční princip ve výrokové logice

- **Rezoluční princip** = pravidlo, které umožňuje odvodit
  z klauzulí $C_1 \cup \{l\}$ a $\{\neg l\} \cup C_2$ klauzuli $C_1 \cup C_2$

$$\frac{C_1 \cup \{l\} \qquad\qquad \{\neg l\} \cup C_2}{C_1 \cup C_2}$$

- $C_1 \cup C_2$ se nazývá **rezolventou** původních klauzulí
- příklad:

$$\frac{\{p, r\} \qquad\qquad \{\neg r, s\}}{\{p, s\}} \qquad\qquad \frac{(p \vee r) \wedge (\neg r \vee s)}{p \vee s}$$

obě klauzule $(p \vee r)$ a $(\neg r \vee s)$ musí být pravdivé

protože r nestačí k pravdivosti obou klauzulí,

musí být pravdivé p (pokud je pravdivé $\neg r$) nebo s (pokud je pravdivé r),

tedy platí klauzule $p \vee s$

Rezoluční důkaz

- **rezoluční důkaz klauzule C z formule F** je konečná posloupnost $C_1, \ldots, C_n = C$ klauzulí taková, že C_i je buď klauzule z F nebo rezolventa C_j, C_k pro $k, j < i$.

Rezoluční důkaz

- **rezoluční důkaz klauzule C z formule F** je konečná posloupnost $C_1, \ldots, C_n = C$ klauzulí taková, že C_i je buď klauzule z F nebo rezolventa C_j, C_k pro $k, j < i$.
- příklad: rezoluční důkaz $\{p\}$ z formule $F = \{\{p, r\}, \{q, \neg r\}, \{\neg q\}\}$

Rezoluční důkaz

- **rezoluční důkaz klauzule C z formule F** je konečná posloupnost $C_1, \ldots, C_n = C$ klauzulí taková, že C_i je buď klauzule z F nebo rezolventa C_j, C_k pro $k, j < i$.
- příklad: rezoluční důkaz $\{p\}$ z formule $F = \{\{p, r\}, \{q, \neg r\}, \{\neg q\}\}$

  $C_1 = \{p, r\}$ klauzule z F

Rezoluční důkaz

- **rezoluční důkaz klauzule C z formule F** je konečná posloupnost $C_1, \ldots, C_n = C$ klauzulí taková, že C_i je buď klauzule z F nebo rezolventa C_j, C_k pro $k, j < i$.
- příklad: rezoluční důkaz $\{p\}$ z formule $F = \{\{p, r\}, \{q, \neg r\}, \{\neg q\}\}$

  $C_1 = \{p, r\}$ klauzule z F

  $C_2 = \{q, \neg r\}$ klauzule z F

Rezoluční důkaz

- **rezoluční důkaz klauzule C z formule F** je konečná posloupnost $C_1, \ldots, C_n = C$ klauzulí taková, že C_i je buď klauzule z F nebo rezolventa C_j, C_k pro $k, j < i$.
- příklad: rezoluční důkaz $\{p\}$ z formule $F = \{\{p, r\}, \{q, \neg r\}, \{\neg q\}\}$

  $C_1 = \{p, r\}$ klauzule z F

  $C_2 = \{q, \neg r\}$ klauzule z F

  $C_3 = \{p, q\}$ rezolventa C_1 a C_2

Rezoluční důkaz

- **rezoluční důkaz klauzule C z formule F** je konečná posloupnost $C_1, \ldots, C_n = C$ klauzulí taková, že C_i je buď klauzule z F nebo rezolventa C_j, C_k pro $k, j < i$.
- příklad: rezoluční důkaz $\{p\}$ z formule $F = \{\{p, r\}, \{q, \neg r\}, \{\neg q\}\}$

  $C_1 = \{p, r\}$ klauzule z F

  $C_2 = \{q, \neg r\}$ klauzule z F

  $C_3 = \{p, q\}$ rezolventa C_1 a C_2

  $C_4 = \{\neg q\}$ klauzule z F

Rezoluční důkaz

- **rezoluční důkaz klauzule C z formule F** je konečná posloupnost $C_1, \ldots, C_n = C$ klauzulí taková, že C_i je buď klauzule z F nebo rezolventa C_j, C_k pro $k, j < i$.
- příklad: rezoluční důkaz $\{p\}$ z formule $F = \{\{p, r\}, \{q, \neg r\}, \{\neg q\}\}$

  $C_1 = \{p, r\}$ klauzule z F

  $C_2 = \{q, \neg r\}$ klauzule z F

  $C_3 = \{p, q\}$ rezolventa C_1 a C_2

  $C_4 = \{\neg q\}$ klauzule z F

  $C_5 = \{p\} = C$ rezolventa C_3 a C_4

Substituce

- **co s proměnnými?** **vhodná substituce a unifikace**
  - $f(X, a, g(Y)) < 1, f(h(c), a, Z) < 1, \quad X = h(c), Z = g(Y) \Longrightarrow f(h(c), a, g(Y)) < 1$

Substituce

- **co s proměnnými?** **vhodná substituce a unifikace**
  - $f(X, a, g(Y)) < 1, f(h(c), a, Z) < 1, \quad X = h(c), Z = g(Y) \Longrightarrow f(h(c), a, g(Y)) < 1$
- **substituce** je libovolná funkce θ zobrazující výrazy do výrazů tak, že platí
  - $\theta(E) = E$ pro libovolnou konstantu E
  - $\theta(f(E_1, \cdots, E_n)) = f(\theta(E_1), \cdots, \theta(E_n))$ pro libovolný funkční symbol f
  - $\theta(p(E_1, \cdots, E_n)) = p(\theta(E_1), \cdots, \theta(E_n))$ pro libovolný predik. symbol p
- **substituce** je tedy homomorfismus výrazů, který **zachová vše kromě proměnných** – ty lze nahradit čímkoliv
- substituce zapisujeme zpravidla ve tvaru seznamu $[X_1/\xi_1, \cdots, X_n/\xi_n]$ kde X_i jsou proměnné a ξ_i substituované termy
  - příklad: $p(X)[X/f(a)] \equiv p(f(a))$

Substituce

- co s proměnnými? vhodná substituce a unifikace
  - $f(X, a, g(Y)) < 1, f(h(c), a, Z) < 1, \quad X = h(c), Z = g(Y) \Longrightarrow f(h(c), a, g(Y)) < 1$
- **substituce** je libovolná funkce θ zobrazující výrazy do výrazů tak, že platí
  - $\theta(E) = E$ pro libovolnou konstantu E
  - $\theta(f(E_1, \cdots, E_n)) = f(\theta(E_1), \cdots, \theta(E_n))$ pro libovolný funkční symbol f
  - $\theta(p(E_1, \cdots, E_n)) = p(\theta(E_1), \cdots, \theta(E_n))$ pro libovolný predik. symbol p
- **substituce** je tedy homomorfismus výrazů, který **zachová vše kromě proměnných** – ty lze nahradit čímkoliv
- substituce zapisujeme zpravidla ve tvaru seznamu $[X_1/\xi_1, \cdots, X_n/\xi_n]$ kde X_i jsou proměnné a ξ_i substituované termy
  - příklad: $p(X)[X/f(a)] \equiv p(f(a))$
- **přejmenování proměnných**: speciální náhrada proměnných proměnnými
  - příklad: $p(X)[X/Y] \equiv p(Y)$

Unifikace

- Ztotožnění dvou literálů p, q pomocí vhodné substituce σ takové, že $p\sigma = q\sigma$ nazýváme **unifikací** a příslušnou substituci **unifikátorem**.
- **Unifikátorem** množiny S literálů nazýváme substituce θ takovou, že množina

$$S\theta = \{t\theta | t \in S\}$$

má jediný prvek.

Unifikace

- Ztotožnění dvou literálů p, q pomocí vhodné substituce σ takové, že $p\sigma = q\sigma$ nazýváme **unifikací** a příslušnou substituci **unifikátorem**.
- **Unifikátorem** množiny S literálů nazýváme substituce θ takovou, že množina

  $$S\theta = \{t\theta | t \in S\}$$

  má jediný prvek.
  - příklad: $S = \{$ datum(D1, M1, 2003), datum(1, M2, Y2) $\}$
    unifikátor $\theta =$ [D1/1, M1/2, M2/2, Y2/2003] $\quad S\theta = \{$ datum(1, 2, 2003) $\}$

Unifikace

- Ztotožnění dvou literálů p, q pomocí vhodné substituce σ takové, že $p\sigma = q\sigma$ nazýváme **unifikací** a příslušnou substituci **unifikátorem**.
- **Unifikátorem** množiny S literálů nazýváme substituce θ takovou, že množina

$$S\theta = \{t\theta | t \in S\}$$

  má jediný prvek.

  - příklad: $S = \{$ datum(D1, M1, 2003), datum(1, M2, Y2) $\}$
    unifikátor θ = [D1/1, M1/2, M2/2, Y2/2003] $S\theta = \{$ datum(1, 2, 2003) $\}$
- Unifikátor σ množiny S nazýváme **nejobecnějším unifikátorem** (**mgu – *most general unifier***), jestliže pro libovolný unifikátor θ existuje substituce λ taková, že $\theta = \sigma\lambda$.

Unifikace

- Ztotožnění dvou literálů p, q pomocí vhodné substituce σ takové, že $p\sigma = q\sigma$ nazýváme **unifikací** a příslušnou substituci **unifikátorem**.
- **Unifikátorem** množiny S literálů nazýváme substituce θ takovou, že množina

$$S\theta = \{t\theta | t \in S\}$$

  má jediný prvek.

  - příklad: S = { datum(D1, M1, 2003), datum(1, M2, Y2) }
    unifikátor θ = [D1/1, M1/2, M2/2, Y2/2003] $S\theta$ = { datum(1, 2, 2003) }

- Unifikátor σ množiny S nazýváme **nejobecnějším unifikátorem** (**mgu** – ***most general unifier***), jestliže pro libovolný unifikátor θ existuje substituce λ taková, že $\theta = \sigma\lambda$.

  - příklad (pokrač.): nejobecnější unifikátor σ = [D1/1, Y2/2003, M1/M2],

Unifikace

- Ztotožnění dvou literálů p, q pomocí vhodné substituce σ takové, že $p\sigma = q\sigma$ nazýváme **unifikací** a příslušnou substituci **unifikátorem**.
- **Unifikátorem** množiny S literálů nazýváme substituce θ takovou, že množina

$$S\theta = \{t\theta | t \in S\}$$

  má jediný prvek.

  - příklad: $S = \{$ datum(D1, M1, 2003), datum(1, M2, Y2) $\}$
    unifikátor θ = [D1/1, M1/2, M2/2, Y2/2003] $S\theta = \{$ datum(1, 2, 2003) $\}$

- Unifikátor σ množiny S nazýváme **nejobecnějším unifikátorem** (**mgu – *most general unifier***), jestliže pro libovolný unifikátor θ existuje substituce λ taková, že $\theta = \sigma\lambda$.
  - příklad (pokrač.): nejobecnější unifikátor σ = [D1/1, Y2/2003, M1/M2], λ=[M2/2]

Rezoluční princip v PL1

- základ:
  - rezoluční princip ve výrokové logice $\dfrac{C_1 \cup \{l\} \qquad \{\neg l\} \cup C_2}{C_1 \cup C_2}$
  - substituce, unifikátor, nejobecnější unifikátor

Rezoluční princip v PL1

- základ:
  - rezoluční princip ve výrokové logice $\dfrac{C_1 \cup \{l\} \qquad \{\neg l\} \cup C_2}{C_1 \cup C_2}$
  - substituce, unifikátor, nejobecnější unifikátor
- **rezoluční princip v PL1** je pravidlo, které
  - připraví příležitost pro uplatnění vlastního rezolučního pravidla nalezením vhodného unifikátoru
  - provede rezoluci a získá rezolventu

Rezoluční princip v PL1

- základ:
  - rezoluční princip ve výrokové logice $\dfrac{C_1 \cup \{l\} \qquad \{\neg l\} \cup C_2}{C_1 \cup C_2}$
  - substituce, unifikátor, nejobecnější unifikátor
- **rezoluční princip v PL1** je pravidlo, které
  - připraví příležitost pro uplatnění vlastního rezolučního pravidla nalezením vhodného unifikátoru
  - provede rezoluci a získá rezolventu

$$\frac{C_1 \cup \{A\} \qquad \{\neg B\} \cup C_2}{C_1\rho\sigma \cup C_2\sigma}$$

  - kde ρ **je přejmenováním proměnných** takové,
    že klauzule $(C_1 \cup A)\rho$ a $\{B\} \cup C_2$ nemají společné proměnné
  - σ **je nejobecnější unifikátor** klauzulí $A\rho$ a B

Příklad: rezoluce v PL1

- příklad: $C_1 = \{p(X,Y), \quad q(Y)\} \qquad C_2 = \{\neg q(a), \quad s(X,W)\}$

Příklad: rezoluce v PL1

- příklad: $C_1 = \{p(X,Y), \quad q(Y)\} \qquad C_2 = \{\neg q(a), \quad s(X,W)\}$
- přejmenování proměnných: $\rho = [X/Z]$

  $C_1 = \{p(Z,Y), \quad q(Y)\} \qquad C_2 = \{\neg q(a), \quad s(X,W)\}$

Příklad: rezoluce v PL1

- příklad: $C_1 = \{p(X,Y), \quad q(Y)\} \qquad C_2 = \{\neg q(a), \quad s(X,W)\}$
- přejmenování proměnných: $\rho = [X/Z]$

  $C_1 = \{p(Z,Y), \quad q(Y)\} \qquad C_2 = \{\neg q(a), \quad s(X,W)\}$
- nejobecnější unifikátor: $\sigma = [Y/a]$

  $C_1 = \{p(Z,a), \quad q(a)\} \qquad C_2 = \{\neg q(a), \quad s(X,W)\}$

Příklad: rezoluce v PL1

- příklad: $C_1 = \{p(X,Y), \quad q(Y)\} \qquad C_2 = \{\neg q(a), \quad s(X,W)\}$
- přejmenování proměnných: $\rho = [X/Z]$

  $C_1 = \{p(Z,Y), \quad q(Y)\} \qquad C_2 = \{\neg q(a), \quad s(X,W)\}$
- nejobecnější unifikátor: $\sigma = [Y/a]$

  $C_1 = \{p(Z,a), \quad q(a)\} \qquad C_2 = \{\neg q(a), \quad s(X,W)\}$
- rezoluční princip: $C = \{p(Z,a), \quad s(X,W)\}$

Příklad: rezoluce v PL1

- příklad: $C_1 = \{p(X,Y), \quad q(Y)\} \qquad C_2 = \{\neg q(a), \quad s(X,W)\}$
- přejmenování proměnných: $\rho = [X/Z]$

  $C_1 = \{p(Z,Y), \quad q(Y)\} \qquad C_2 = \{\neg q(a), \quad s(X,W)\}$
- nejobecnější unifikátor: $\sigma = [Y/a]$

  $C_1 = \{p(Z,a), \quad q(a)\} \qquad C_2 = \{\neg q(a), \quad s(X,W)\}$
- rezoluční princip: $C = \{p(Z,a), \quad s(X,W)\}$

- vyzkoušejte si:

  $C_1 = \{q(X), \quad \neg r(Y), \quad p(X,Y), \quad p(f(Z),f(Z))\}$

  $C_2 = \{n(Y), \quad \neg r(W), \quad \neg p(f(W),f(W)\}$

Rezoluce v PL1

Obecný rezoluční princip v PL1

$$\frac{C_1 \cup \{A_1, \cdots, A_m\} \qquad\qquad \{\neg B_1, \cdots, \neg B_n\} \cup C_2}{C_1 \rho\sigma \cup C_2\sigma}$$

- kde ρ je přejmenováním proměnných takové, že množiny klauzulí $\{A_1\rho, \cdots, A_m\rho, C_1\rho\}$ a $\{B_1, \cdots, B_n, C_2\}$ nemají společné proměnné
- σ je nejobecnější unifikátor množiny $\{A_1\rho, \cdots, A_m\rho, B_1, \cdots, B_n\}$

Rezoluce v PL1

- **Obecný rezoluční princip v PL1**

$$\frac{C_1 \cup \{A_1, \cdots, A_m\} \qquad \{\neg B_1, \cdots, \neg B_n\} \cup C_2}{C_1\rho\sigma \cup C_2\sigma}$$

  - kde ρ je přejmenováním proměnných takové, že množiny klauzulí $\{A_1\rho, \cdots, A_m\rho, C_1\rho\}$ a $\{B_1, \cdots, B_n, C_2\}$ nemají společné proměnné
  - σ je nejobecnější unifikátor množiny $\{A_1\rho, \cdots, A_m\rho, B_1, \cdots, B_n\}$
  - příklad: $A_1 = a(X)$ vs. $\{\neg B_1, \neg B_2\} = \{\neg a(b), \neg a(Z)\}$

    v jednom kroku potřebuji vyrezolvovat zároveň B_1 i B_2

Rezoluce v PL1

- **Obecný rezoluční princip v PL1**

$$\frac{C_1 \cup \{A_1, \cdots, A_m\} \qquad \{\neg B_1, \cdots, \neg B_n\} \cup C_2}{C_1\rho\sigma \cup C_2\sigma}$$

  - kde ρ je přejmenováním proměnných takové, že množiny klauzulí $\{A_1\rho, \cdots, A_m\rho, C_1\rho\}$ a $\{B_1, \cdots, B_n, C_2\}$ nemají společné proměnné
  - σ je nejobecnější unifikátor množiny $\{A_1\rho, \cdots, A_m\rho, B_1, \cdots, B_n\}$
  - příklad: $A_1 = a(X)$ vs. $\{\neg B_1, \neg B_2\} = \{\neg a(b), \neg a(Z)\}$

    v jednom kroku potřebuji vyrezolvovat zároveň B_1 i B_2

- Rezoluce v PL1
  - **korektní**: jestliže existuje rezoluční vyvrácení F, pak F je nesplnitelná
  - **úplná**: jestliže F je nesplnitelná, pak existuje rezoluční vyvrácení F

Zefektivnění rezoluce

- rezoluce je intuitivně efektivnější než axiomatické systémy
  - axiomatické systémy: který z axiomů a pravidel použít?
  - rezoluce: pouze jedno pravidlo

Zefektivnění rezoluce

- rezoluce je intuitivně efektivnější než axiomatické systémy
  - axiomatické systémy: který z axiomů a pravidel použít?
  - rezoluce: pouze jedno pravidlo
- stále ale příliš mnoho možností, jak hledat důkaz v prohledávacím prostoru
- problém SAT= $\{S|S$ je splnitelná $\}$ NP úplný,

  nicméně: menší prohledávací prostor vede k rychlejšímu nalezení řešení
- strategie pro zefektivnění prohledávání $\Rightarrow$ varianty rezoluční metody

Zefektivnění rezoluce

- rezoluce je intuitivně efektivnější než axiomatické systémy
  - axiomatické systémy: který z axiomů a pravidel použít?
  - rezoluce: pouze jedno pravidlo
- stále ale příliš mnoho možností, jak hledat důkaz v prohledávacím prostoru
- problém SAT= $\{S | S$ je splnitelná $\}$ NP úplný,

  nicméně: menší prohledávací prostor vede k rychlejšímu nalezení řešení
- strategie pro zefektivnění prohledávání $\Rightarrow$ varianty rezoluční metody
- vylepšení prohledávání
  - zastavit prohledávání cest, které nejsou slibné
  - specifikace pořadí, jak procházíme alternativními cestami

Varianty rezoluční metody

- **Věta:** Každé omezení rezoluce je korektní.
  - stále víme, že to, co jsme dokázali, platí

Variantty rezoluční metody

- **Věta:** Každé omezení rezoluce je korektní.
  - stále víme, že to, co jsme dokázali, platí
- T-**rezoluce**: klauzule učastnící se rezoluce nejsou tautologie úplná
  - tautologie nepomůže ukázat, že formule je nesplnitelná

Varianty rezoluční metody

- **Věta:** Každé omezení rezoluce je korektní.
  - stále víme, že to, co jsme dokázali, platí
- T-**rezoluce**: klauzule učastnící se rezoluce nejsou tautologie úplná
  - tautologie nepomůže ukázat, že formule je nesplnitelná
- **sémantická rezoluce**: úplná
  zvolíme libovolnou interpretaci a pro rezoluci používáme jen takové klauzule, z nichž alespoň jedna je v této interpretaci nepravdivá
  - pokud jsou obě klauzule pravdivé, těžko odvodíme nesplnitelnost formule

Varianty rezoluční metody

- Věta: Každé omezení rezoluce je korektní.
  - stále víme, že to, co jsme dokázali, platí
- T-**rezoluce**: klauzule účastnící se rezoluce nejsou tautologie úplná
  - tautologie nepomůže ukázat, že formule je nesplnitelná
- **sémantická rezoluce**: úplná
  zvolíme libovolnou interpretaci a pro rezoluci používáme jen takové klauzule, z nichž alespoň jedna je v této interpretaci nepravdivá
  - pokud jsou obě klauzule pravdivé, těžko odvodíme nesplnitelnost formule
- **vstupní (*input*) rezoluce:** neúplná
  alespoň jedna z klauzulí, použitá při rezoluci, je z výchozí **vstupní množiny** S

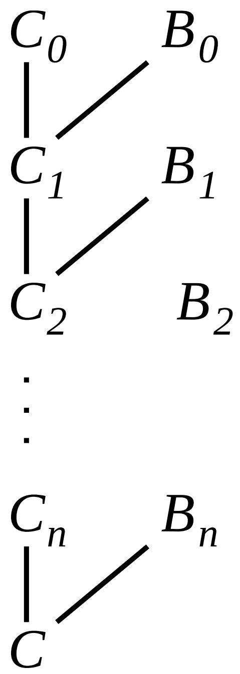
Varianty rezoluční metody

- **Věta:** Každé omezení rezoluce je korektní.
  - stále víme, že to, co jsme dokázali, platí
- ***T*-rezoluce**: klauzule účastnící se rezoluce nejsou tautologie úplná
  - tautologie nepomůže ukázat, že formule je nesplnitelná
- **sémantická rezoluce**: úplná
  zvolíme libovolnou interpretaci a pro rezoluci používáme jen takové klauzule, z nichž alespoň jedna je v této interpretaci nepravdivá
  - pokud jsou obě klauzule pravdivé, těžko odvodíme nesplnitelnost formule
- **vstupní (*input*) rezoluce:** neúplná
  alespoň jedna z klauzulí, použitá při rezoluci, je z výchozí **vstupní množiny** S
  - $\{\{p,q\}, \quad \{\neg p,q\}, \quad \{p,\neg q\}, \quad \{\neg p,\neg q\}\}$
    existuje rezoluční vyvrácení
    neexistuje rezoluční vyvrácení pomocí vstupní rezoluce

Rezoluce a logické programování

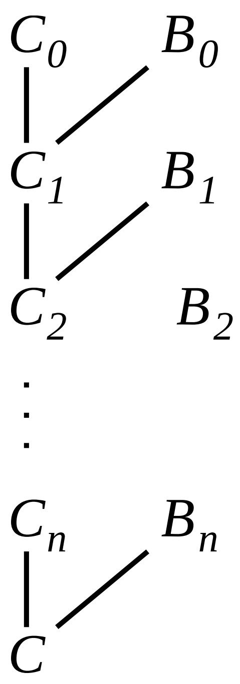
Lineární rezoluce

- varianta rezoluční metody
  - snaha o generování lineární posloupnosti místo stromu
  - v každém kroku kromě prvního můžeme použít bezprostředně předcházející rezolventu a k tomu buď některou z klauzulí vstupní množiny S nebo některou z předcházejících rezolvent



Lineární rezoluce

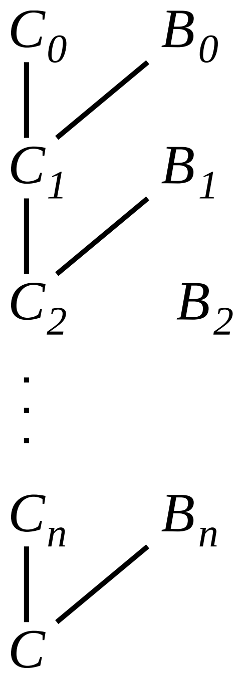
- varianta rezoluční metody
  - snaha o generování lineární posloupnosti místo stromu
  - v každém kroku kromě prvního můžeme použít bezprostředně předcházející rezolventu a k tomu buď některou z klauzulí vstupní množiny S nebo některou z předcházejících rezolvent



- **lineární rezoluční důkaz C z S** je posloupnost dvojic $\langle C_0, B_0\rangle, \ldots \langle C_n, B_n\rangle$ taková, že $C = C_{n+1}$ a
  - C_0 a každá B_i jsou prvky S nebo některé $C_j, j < i$
  - každá $C_{i+1}, i \leq n$ je rezolventa C_i a B_i

Lineární rezoluce

- varianta rezoluční metody
  - snaha o generování lineární posloupnosti místo stromu
  - v každém kroku kromě prvního můžeme použít bezprostředně předcházející rezolventu a k tomu buď některou z klauzulí vstupní množiny S nebo některou z předcházejících rezolvent
- **lineární rezoluční důkaz C z S** je posloupnost dvojic $\langle C_0, B_0 \rangle, \ldots \langle C_n, B_n \rangle$ taková, že $C = C_{n+1}$ a
  - C_0 a každá B_i jsou prvky S nebo některé $C_j, j < i$
  - každá $C_{i+1}, i \leq n$ je rezolventa C_i a B_i
- **lineární vyvrácení S** = lineární rezoluční důkaz $\square$ z S



Lineární rezoluce II.

- příklad: $S = \{A_1, A_2, A_3, A_4\}$

  $A_1 = \{p, q\}$

  $A_2 = \{p, \neg q\}$

  $A_3 = \{\neg p, q\}$

  $A_4 = \{\neg p, \neg q\}$

Lineární rezoluce II.

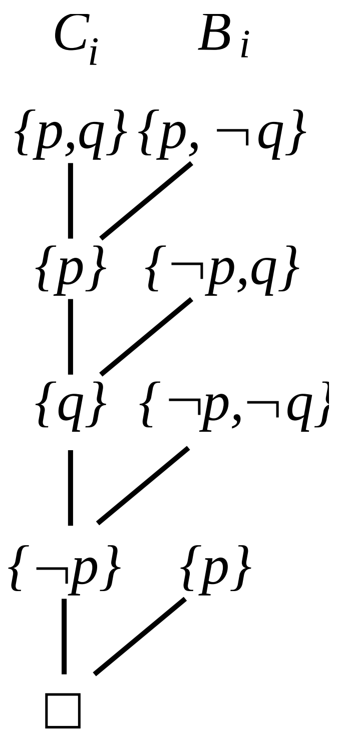
- příklad: $S = \{A_1, A_2, A_3, A_4\}$

  $A_1 = \{p, q\}$

  $A_2 = \{p, \neg q\}$

  $A_3 = \{\neg p, q\}$

  $A_4 = \{\neg p, \neg q\}$



Lineární rezoluce II.

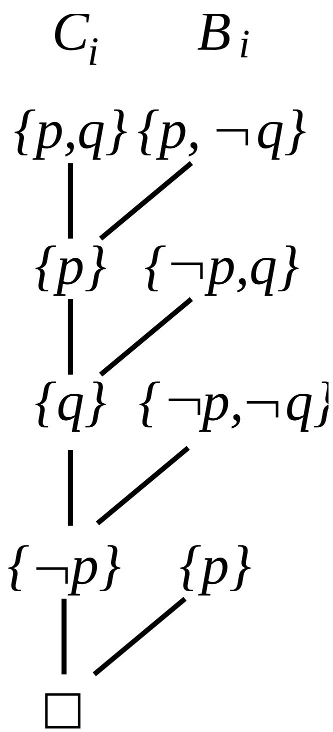
- příklad: $S = \{A_1, A_2, A_3, A_4\}$

  $A_1 = \{p, q\}$

  $A_2 = \{p, \neg q\}$

  $A_3 = \{\neg p, q\}$

  $A_4 = \{\neg p, \neg q\}$



- S: **vstupní množina** klauzulí
- C_i: **střední klauzule**
- B_i: **boční klauzule**

Prologovská notace

- Klauzule v matematické logice
  - $\{H_1, \cdots, H_m, \neg T_1, \cdots, \neg T_n\}$ $\quad H_1 \vee \cdots \vee H_m \vee \neg T_1 \vee \cdots \vee \neg T_n$

Prologovská notace

- Klauzule v matematické logice
  - $\{H_1, \cdots, H_m, \neg T_1, \cdots, \neg T_n\}$ $\quad$ $H_1 \vee \cdots \vee H_m \vee \neg T_1 \vee \cdots \vee \neg T_n$
- **Hornova klauzule**: nejvýše jeden pozitivní literál
  - $\{H, \neg T_1, \ldots, \neg T_n\}$ $\quad$ $\{H\}$ $\quad$ $\{\neg T_1, \ldots, \neg T_n\}$
  - $H \vee \neg T_1 \vee \cdots \vee \neg T_n$ $\quad$ H $\quad$ $\neg T_1 \vee \cdots \vee \neg T_n$

Prologovská notace

- Klauzule v matematické logice
  - $\{H_1, \cdots, H_m, \neg T_1, \cdots, \neg T_n\}$ $\qquad H_1 \vee \cdots \vee H_m \vee \neg T_1 \vee \cdots \vee \neg T_n$
- **Hornova klauzule**: nejvýše jeden pozitivní literál
  - $\{H, \neg T_1, \ldots, \neg T_n\}$ $\qquad \{H\}$ $\qquad \{\neg T_1, \ldots, \neg T_n\}$
  - $H \vee \neg T_1 \vee \cdots \vee \neg T_n$ $\qquad H$ $\qquad \neg T_1 \vee \cdots \vee \neg T_n$
- **Pravidlo**: jeden pozitivní a alespoň jeden negativní literál
  - Prolog: $H :- T_1, \cdots, T_n.$

Prologovská notace

- Klauzule v matematické logice
  - $\{H_1, \cdots, H_m, \neg T_1, \cdots, \neg T_n\}$ $\quad H_1 \vee \cdots \vee H_m \vee \neg T_1 \vee \cdots \vee \neg T_n$
- **Hornova klauzule**: nejvýše jeden pozitivní literál
  - $\{H, \neg T_1, \ldots, \neg T_n\}$ $\quad \{H\}$ $\quad \{\neg T_1, \ldots, \neg T_n\}$
  - $H \vee \neg T_1 \vee \cdots \vee \neg T_n$ $\quad H$ $\quad \neg T_1 \vee \cdots \vee \neg T_n$
- **Pravidlo**: jeden pozitivní a alespoň jeden negativní literál
  - Prolog: $H :- T_1, \cdots, T_n.$ $\quad$ Matematická logika: $H \Leftarrow T_1 \wedge \cdots \wedge T_n$

Prologovská notace

- Klauzule v matematické logice
  - $\{H_1, \cdots, H_m, \neg T_1, \cdots, \neg T_n\}$ $\quad H_1 \vee \cdots \vee H_m \vee \neg T_1 \vee \cdots \vee \neg T_n$
- **Hornova klauzule**: nejvýše jeden pozitivní literál
  - $\{H, \neg T_1, \ldots, \neg T_n\}$ $\quad \{H\}$ $\quad \{\neg T_1, \ldots, \neg T_n\}$
  - $H \vee \neg T_1 \vee \cdots \vee \neg T_n$ $\quad H$ $\quad \neg T_1 \vee \cdots \vee \neg T_n$
- **Pravidlo**: jeden pozitivní a alespoň jeden negativní literál
  - Prolog: $H :- T_1, \cdots, T_n.$ $\quad$ Matematická logika: $H \Leftarrow T_1 \wedge \cdots \wedge T_n$
  - $H \Leftarrow T$

Prologovská notace

- Klauzule v matematické logici
  - $\{H_1, \cdots, H_m, \neg T_1, \cdots, \neg T_n\}$ $\qquad H_1 \vee \cdots \vee H_m \vee \neg T_1 \vee \cdots \vee \neg T_n$
- **Hornova klauzule**: nejvýše jeden pozitivní literál
  - $\{H, \neg T_1, \ldots, \neg T_n\}$ $\qquad \{H\}$ $\qquad \{\neg T_1, \ldots, \neg T_n\}$
  - $H \vee \neg T_1 \vee \cdots \vee \neg T_n$ $\qquad H$ $\qquad \neg T_1 \vee \cdots \vee \neg T_n$
- **Pravidlo**: jeden pozitivní a alespoň jeden negativní literál
  - Prolog: $H :- T_1, \cdots, T_n.$ $\qquad$ Matematická logika: $H \Leftarrow T_1 \wedge \cdots \wedge T_n$
  - $H \Leftarrow T$ $\qquad H \vee \neg T$

Prologovská notace

- Klauzule v matematické logice
  - $\{H_1, \cdots, H_m, \neg T_1, \cdots, \neg T_n\}$ $\quad H_1 \vee \cdots \vee H_m \vee \neg T_1 \vee \cdots \vee \neg T_n$
- **Hornova klauzule**: nejvýše jeden pozitivní literál
  - $\{H, \neg T_1, \ldots, \neg T_n\}$ $\quad \{H\}$ $\quad \{\neg T_1, \ldots, \neg T_n\}$
  - $H \vee \neg T_1 \vee \cdots \vee \neg T_n$ $\quad H$ $\quad \neg T_1 \vee \cdots \vee \neg T_n$
- **Pravidlo**: jeden pozitivní a alespoň jeden negativní literál
  - Prolog: $H :- T_1, \cdots, T_n.$ $\quad$ Matematická logika: $H \Leftarrow T_1 \wedge \cdots \wedge T_n$
  - $H \Leftarrow T$ $\quad H \vee \neg T$ $\quad H \vee \neg T_1 \vee \cdots \vee \neg T_n$

Prologovská notace

- Klauzule v matematické logice
  - $\{H_1, \cdots, H_m, \neg T_1, \cdots, \neg T_n\}$ $\quad H_1 \vee \cdots \vee H_m \vee \neg T_1 \vee \cdots \vee \neg T_n$
- **Hornova klauzule**: nejvýše jeden pozitivní literál
  - $\{H, \neg T_1, \ldots, \neg T_n\}$ $\quad \{H\}$ $\quad \{\neg T_1, \ldots, \neg T_n\}$
  - $H \vee \neg T_1 \vee \cdots \vee \neg T_n$ $\quad H$ $\quad \neg T_1 \vee \cdots \vee \neg T_n$
- **Pravidlo**: jeden pozitivní a alespoň jeden negativní literál
  - Prolog: $H :- T_1, \cdots, T_n.$ Matematická logika: $H \Leftarrow T_1 \wedge \cdots \wedge T_n$
  - $H \Leftarrow T$ $\quad H \vee \neg T$ $\quad H \vee \neg T_1 \vee \cdots \vee \neg T_n$ Klauzule: $\{H, \neg T_1, \ldots, \neg T_n\}$

Prologovská notace

- Klauzule v matematické logice
  - $\{H_1, \cdots, H_m, \neg T_1, \cdots, \neg T_n\}$ $\quad$ $H_1 \vee \cdots \vee H_m \vee \neg T_1 \vee \cdots \vee \neg T_n$
- **Hornova klauzule**: nejvýše jeden pozitivní literál
  - $\{H, \neg T_1, \ldots, \neg T_n\}$ $\quad$ $\{H\}$ $\quad$ $\{\neg T_1, \ldots, \neg T_n\}$
  - $H \vee \neg T_1 \vee \cdots \vee \neg T_n$ $\quad$ H $\quad$ $\neg T_1 \vee \cdots \vee \neg T_n$
- **Pravidlo**: jeden pozitivní a alespoň jeden negativní literál
  - Prolog: $H :- T_1, \cdots, T_n.$ $\quad$ Matematická logika: $H \Leftarrow T_1 \wedge \cdots \wedge T_n$
  - $H \Leftarrow T$ $\quad$ $H \vee \neg T$ $\quad$ $H \vee \neg T_1 \vee \cdots \vee \neg T_n$ $\quad$ Klauzule: $\{H, \neg T_1, \ldots, \neg T_n\}$
- **Fakt**: pouze jeden pozitivní literál
  - Prolog: $H.$ $\quad$ Matematická logika: H $\quad$ Klauzule: $\{H\}$

Prologovská notace

- Klauzule v matematické logici
  - $\{H_1, \cdots, H_m, \neg T_1, \cdots, \neg T_n\}$ $\quad$ $H_1 \vee \cdots \vee H_m \vee \neg T_1 \vee \cdots \vee \neg T_n$
- **Hornova klauzule**: nejvýše jeden pozitivní literál
  - $\{H, \neg T_1, \ldots, \neg T_n\}$ $\quad$ $\{H\}$ $\quad$ $\{\neg T_1, \ldots, \neg T_n\}$
  - $H \vee \neg T_1 \vee \cdots \vee \neg T_n$ $\quad$ H $\quad$ $\neg T_1 \vee \cdots \vee \neg T_n$
- **Pravidlo**: jeden pozitivní a alespoň jeden negativní literál
  - Prolog: $H : - T_1, \cdots, T_n.$ $\quad$ Matematická logika: $H \Leftarrow T_1 \wedge \cdots \wedge T_n$
  - $H \Leftarrow T$ $\quad$ $H \vee \neg T$ $\quad$ $H \vee \neg T_1 \vee \cdots \vee \neg T_n$ $\quad$ Klauzule: $\{H, \neg T_1, \ldots, \neg T_n\}$
- **Fakt**: pouze jeden pozitivní literál
  - Prolog: $H.$ $\quad$ Matematická logika: H $\quad$ Klauzule: $\{H\}$
- **Cílová klauzule**: žádný pozitivní literál
  - Prolog: $: - T_1, \ldots T_n.$ $\quad$ Matematická logika: $\neg T_1 \vee \cdots \vee \neg T_n$ $\quad$ Klauzule: $\{\neg T_1, \cdots \neg T_n\}$

Logický program

- **Programová klauzule**: právě jeden pozitivní literál (fakt nebo pravidlo)
- **Logický program**: konečná množina programových klauzulí
- Příklad:
  - logický program jako množina klauzulí:

    $P = \{P_1, P_2, P_3\}$

    $P_1 = \{p\}, \quad P_2 = \{p, \neg q\}, \quad P_3 = \{q\}$

Logický program

- **Programová klauzule**: právě jeden pozitivní literál (fakt nebo pravidlo)
- **Logický program**: konečná množina programových klauzulí
- Příklad:
  - logický program jako množina klauzulí:

    $P = \{P_1, P_2, P_3\}$

    $P_1 = \{p\}, \quad P_2 = \{p, \neg q\}, \quad P_3 = \{q\}$
  - logický program v prologovské notaci:

    $p.$

    $p :- q.$

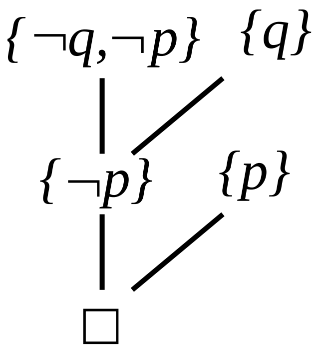
    $q.$
  - cílová klauzule: $G = \{\neg q, \neg p\}$ $\quad :- q, p.$

Lineární rezoluce pro Hornovy klauzule

- Začneme s cílovou klauzulí: $C_0 = G$
- Boční klauzule vybíráme z programových klauzulí P
- $G = \{\neg q, \neg p\}$ $\quad P = \{P_1, P_2, P_3\}: \quad P_1 = \{p\}, \quad P_2 = \{p, \neg q\}, \quad P_3 = \{q\}$
- $:- q, p.$ $\quad p.$ $\quad p :- q,$ $\quad q.$

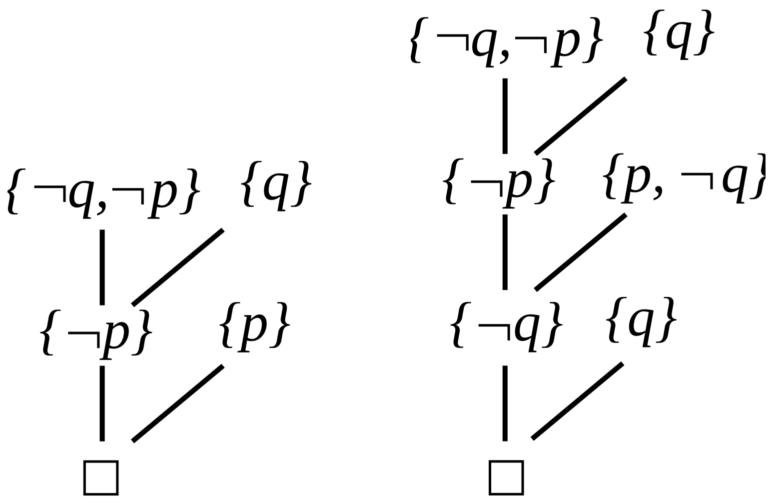
Lineární rezoluce pro Hornovy klauzule

- Začneme s cílovou klauzulí: $C_0 = G$
- Boční klauzule vybíráme z programových klauzulí P
- $G = \{\neg q, \neg p\}$ $\quad P = \{P_1, P_2, P_3\}: \quad P_1 = \{p\}, \quad P_2 = \{p, \neg q\}, \quad P_3 = \{q\}$
- $:- q, p. \quad p. \quad p :- q, \quad q.$



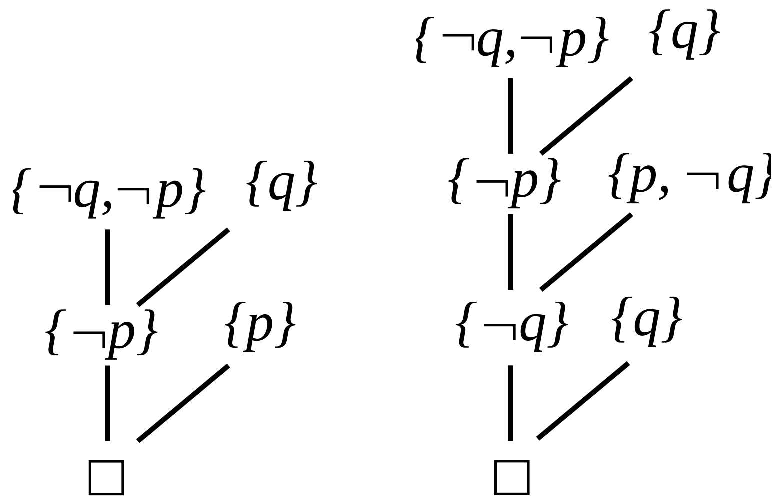
Lineární rezoluce pro Hornovy klauzule

- Začneme s cílovou klauzulí: $C_0 = G$
- Boční klauzule vybíráme z programových klauzulí P
- $G = \{\neg q, \neg p\}$ $\quad P = \{P_1, P_2, P_3\}: \quad P_1 = \{p\}, \quad P_2 = \{p, \neg q\}, \quad P_3 = \{q\}$
- $:-q, p.$ $\quad p. \quad p :- q, \quad q.$



Lineární rezoluce pro Hornovy klauzule

- Začneme s cílovou klauzulí: $C_0 = G$
- Boční klauzule vybíráme z programových klauzulí P
- $G = \{\neg q, \neg p\}$ $\quad P = \{P_1, P_2, P_3\}:$ $\quad P_1 = \{p\},$ $\quad P_2 = \{p, \neg q\},$ $\quad P_3 = \{q\}$
- $: -q, p.$ $\quad p.$ $\quad p : -q,$ $\quad q.$



- **Střední klauzule jsou cílové klauzule**

Lineární vstupní rezoluce

- **Vstupní rezoluce na $P \cup \{G\}$**
  - (opakování:) alespoň jedna z klauzulí použitá při rezoluci je z výchozí vstupní množiny
  - začneme s cílovou klauzulí: $C_0 = G$
  - boční klauzule jsou vždy z P (tj. jsou to programové klauzule)

Lineární vstupní rezoluce

- **Vstupní rezoluce na $P \cup \{G\}$**
  - (opakování:) alespoň jedna z klauzulí použitá při rezoluci je z výchozí vstupní množiny
  - začneme s cílovou klauzulí: $C_0 = G$
  - boční klauzule jsou vždy z P (tj. jsou to programové klauzule)
- (Opakování:) **Lineární rezoluční důkaz C z P** je posloupnost dvojic $\langle C_0, B_0 \rangle, \ldots \langle C_n, B_n \rangle$ taková, že $C = C_{n+1}$ a
  - C_0 a každá B_i jsou prvky P **nebo některé $C_j, j < i$**
  - každá $C_{i+1}, i \leq n$ je rezolventa C_i a B_i

Lineární vstupní rezoluce

- **Vstupní rezoluce na** $P \cup \{G\}$
  - (opakování:) alespoň jedna z klauzulí použitá při rezoluci je z výchozí vstupní množiny
  - začneme s cílovou klauzulí: $C_0 = G$
  - boční klauzule jsou vždy z P (tj. jsou to programové klauzule)
- (Opakování:) **Lineární rezoluční důkaz** C **z** P je posloupnost dvojic $\langle C_0, B_0 \rangle, \ldots \langle C_n, B_n \rangle$ taková, že $C = C_{n+1}$ a
  - C_0 a každá B_i jsou prvky P **nebo některé** $C_j, j < i$
  - každá $C_{i+1}, i \leq n$ je rezolventa C_i a B_i
- **Lineární vstupní (*Linear Input*) rezoluce (LI-rezoluce)** C z $P \cup \{G\}$ posloupnost dvojic $\langle C_0, B_0 \rangle, \ldots \langle C_n, B_n \rangle$ taková, že $C = C_{n+1}$ a
  - $C_0 = G$ **a každá** B_i **jsou prvky** P — lineární rezoluce + vstupní rezoluce
  - každá $C_{i+1}, i \leq n$ je rezolventa C_i a B_i

Cíle a fakta při lineární rezoluci

- **Věta:** Je-li S nesplnitelná množina Hornových klauzulí, pak S obsahuje alespoň **jeden cíl a jeden fakt**.
  - pokud nepoužiji cíl, mám pouze fakta (1 pozit.literál) a pravidla (1 pozit.literál a alespoň jeden negat. literál), při rezoluci mi stále zůstává alespoň jeden pozit. literál

Cíle a fakta při lineární rezoluci

- **Věta:** Je-li S nesplnitelná množina Hornových klauzulí, pak S obsahuje alespoň **jeden cíl a jeden fakt**.
  - pokud nepoužiji cíl, mám pouze fakta (1 pozit.literál) a pravidla (1 pozit.literál a alespoň jeden negat. literál), při rezoluci mi stále zůstává alespoň jeden pozit. literál
  - pokud nepoužiji fakt, mám pouze cíle (negat.literály) a pravidla (1 pozit.literál a alespoň jeden negat. literál), v rezolventě mi stále zůstávají negativní literály

Cíle a fakta při lineární rezoluci

- **Věta:** Je-li S nesplnitelná množina Hornových klauzulí, pak S obsahuje alespoň **jeden cíl a jeden fakt**.
  - pokud nepoužiji cíl, mám pouze fakta (1 pozit.literál) a pravidla (1 pozit.literál a alespoň jeden negat. literál), při rezoluci mi stále zůstává alespoň jeden pozit. literál
  - pokud nepoužiji fakt, mám pouze cíle (negat.literály) a pravidla (1 pozit.literál a alespoň jeden negat. literál), v rezolventě mi stále zůstávají negativní literály

- **Věta:** Existuje-li rezoluční důkaz prázdné množiny z množiny S Hornových klauzulí, pak tento rezoluční strom má v listech **jedinou cílovou klauzuli**.
  - pokud začnu důkaz pravidlem a faktem, pak dostanu zase pravidlo
  - pokud začnu důkaz dvěma pravidly, pak dostanu zase pravidlo
  - na dvou faktech rezolvovat nelze

Cíle a fakta při lineární rezoluci

- **Věta:** Je-li S nesplnitelná množina Hornových klauzulí, pak S obsahuje alespoň **jeden cíl a jeden fakt**.
  - pokud nepoužiji cíl, mám pouze fakta (1 pozit.literál) a pravidla (1 pozit.literál a alespoň jeden negat. literál), při rezoluci mi stále zůstává alespoň jeden pozit. literál
  - pokud nepoužiji fakt, mám pouze cíle (negat.literály) a pravidla (1 pozit.literál a alespoň jeden negat. literál), v rezolventě mi stále zůstávají negativní literály

- **Věta:** Existuje-li rezoluční důkaz prázdné množiny z množiny S Hornových klauzulí, pak tento rezoluční strom má v listech **jedinou cílovou klauzuli**.
  - pokud začnu důkaz pravidlem a faktem, pak dostanu zase pravidlo
  - pokud začnu důkaz dvěma pravidly, pak dostanu zase pravidlo
  - na dvou faktech rezolvovat nelze

  $\Rightarrow$ dokud nepoužiji cíl pracuji stále s množinou faktů a pravidel

Cíle a fakta při lineární rezoluci

- **Věta:** Je-li S nesplnitelná množina Hornových klauzulí, pak S obsahuje alespoň **jeden cíl a jeden fakt**.
  - pokud nepoužiji cíl, mám pouze fakta (1 pozit.literál) a pravidla (1 pozit.literál a alespoň jeden negat. literál), při rezoluci mi stále zůstává alespoň jeden pozit. literál
  - pokud nepoužiji fakt, mám pouze cíle (negat.literály) a pravidla (1 pozit.literál a alespoň jeden negat. literál), v rezolventě mi stále zůstávají negativní literály

- **Věta:** Existuje-li rezoluční důkaz prázdné množiny z množiny S Hornových klauzulí, pak tento rezoluční strom má v listech **jedinou cílovou klauzuli**.
  - pokud začnu důkaz pravidlem a faktem, pak dostanu zase pravidlo
  - pokud začnu důkaz dvěma pravidly, pak dostanu zase pravidlo
  - na dvou faktech rezolvovat nelze

  $\Rightarrow$ dokud nepoužiji cíl pracuji stále s množinou faktů a pravidel
  - pokud použiji v důkazu cílovou klauzuli,
    fakta mi ubírají negat.literály, pravidla mi je přidávají,
    v rezolventě mám stále samé negativní literály, tj. nelze rezolvovat s dalším cílem

Korektnost a úplnost

- **Věta:** Množina S Hornových klauzulí je nesplnitelná, právě když existuje rezoluční vyvrácení S pomocí **vstupní rezoluce**.
- **Korektnost** platí stejně jako pro ostatní omezení rezoluce
- **Úplnost LI-rezoluce pro Hornovy klauzule**:

  Nechť P je množina programových klauzulí a G cílová klauzule.

  Je-li množina $P \cup \{G\}$ Hornových klauzulí nesplnitelná,

  pak existuje rezoluční vyvrácení $P \cup \{G\}$ pomocí LI-rezoluce.

  - vstupní rezoluce pro (obecnou) formuli sama o sobě není úplná

    $\Longrightarrow$ LI-rezoluce aplikovaná na (obecnou) formuli nezaručuje,

    že nalezeneme důkaz, i když formule platí!

Korektnost a úplnost

- **Věta:** Množina S Hornových klauzulí je nesplnitelná, právě když existuje rezoluční vyvrácení S pomocí **vstupní rezoluce**.
- **Korektnost** platí stejně jako pro ostatní omezení rezoluce
- **Úplnost LI-rezoluce pro Hornovy klauzule**:

  Nechť P je množina programových klauzulí a G cílová klauzule.

  Je-li množina $P \cup \{G\}$ Hornových klauzulí nesplnitelná,

  pak existuje rezoluční vyvrácení $P \cup \{G\}$ pomocí LI-rezoluce.

  - vstupní rezoluce pro (obecnou) formuli sama o sobě není úplná

    $\Longrightarrow$ LI-rezoluce aplikovaná na (obecnou) formuli nezaručuje,

    že nalezeneme důkaz, i když formule platí!
- **Význam LI-rezoluce pro Hornovy klauzule:**
  - $P = \{P_1, \ldots, P_n\}$, $G = \{G_1, \ldots, G_m\}$
  - LI-rezolucí ukážeme nesplnitelnost $P_1 \wedge \cdots \wedge P_n \wedge (\neg G_1 \vee \cdots \vee \neg G_m)$

Korektnost a úplnost

- **Věta:** Množina S Hornových klauzulí je nesplnitelná, právě když existuje rezoluční vyvrácení S pomocí **vstupní rezoluce**.
- **Korektnost** platí stejně jako pro ostatní omezení rezoluce
- **Úplnost LI-rezoluce pro Hornovy klauzule**:
  Nechť P je množina programových klauzulí a G cílová klauzule.
  Je-li množina $P \cup \{G\}$ Hornových klauzulí nesplnitelná,
  pak existuje rezoluční vyvrácení $P \cup \{G\}$ pomocí LI-rezoluce.
  - vstupní rezoluce pro (obecnou) formuli sama o sobě není úplná

    $\Longrightarrow$ LI-rezoluce aplikovaná na (obecnou) formuli nezaručuje,
    že nalezeneme důkaz, i když formule platí!
- **Význam LI-rezoluce pro Hornovy klauzule:**
  - $P = \{P_1, \ldots, P_n\}$, $G = \{G_1, \ldots, G_m\}$
  - LI-rezolucí ukážeme nesplnitelnost $P_1 \wedge \cdots \wedge P_n \wedge (\neg G_1 \vee \cdots \vee \neg G_m)$
  - pokud tedy předpokládáme, že program $\{P_1, \ldots, P_n\}$ platí,
    tak musí být nepravdivá $(\neg G_1 \vee \cdots \vee \neg G_m)$, tj. musí platit $G_1 \wedge \cdots \wedge G_m$

Uspořádané klauzule (*definite clauses*)

- Klauzule = množina literálů
- **Uspořádáná klauzule (*definite clause*)** = **posloupnost literálů**
  - nelze volně měnit pořadí literálů
- **Rezoluční princip pro uspořádané klauzule**:

$$\frac{\{\neg A_0, \ldots, \neg A_n\} \qquad \{B, \neg B_0, \ldots, \neg B_m\}}{\{\neg A_0, \ldots, \neg A_{i-1}, \neg B_0\rho, \ldots, \neg B_m\rho, \neg A_{i+1}, \ldots, \neg A_n\}\sigma}$$

  - **uspořádaná rezolventa**: $\{\neg A_0, \ldots, \neg A_{i-1}, \neg B_0\rho, \ldots, \neg B_m\rho, \neg A_{i+1}, \ldots, \neg A_n\}\sigma$
  - ρ je přejmenování proměnných takové, že klauzule $\{A_0, \ldots, A_n\}$ a $\{B, B_0, \ldots, B_m\}\rho$ nemají společné proměnné
  - σ je nejobecnější unifikátor pro A_i a $B\rho$

Uspořádané klauzule (*definite clauses*)

- Klauzule = množina literálů
- **Uspořádáná klauzule (*definite clause*)** = **posloupnost literálů**
  - nelze volně měnit pořadí literálů
- **Rezoluční princip pro uspořádané klauzule**:

$$\frac{\{\neg A_0, \ldots, \neg A_n\} \qquad \{B, \neg B_0, \ldots, \neg B_m\}}{\{\neg A_0, \ldots, \neg A_{i-1}, \neg B_0\rho, \ldots, \neg B_m\rho, \neg A_{i+1}, \ldots, \neg A_n\}\sigma}$$

  - **uspořádaná rezolventa**: $\{\neg A_0, \ldots, \neg A_{i-1}, \neg B_0\rho, \ldots, \neg B_m\rho, \neg A_{i+1}, \ldots, \neg A_n\}\sigma$
  - ρ je přejmenování proměnných takové, že klauzule $\{A_0, \ldots, A_n\}$ a $\{B, B_0, \ldots, B_m\}\rho$ nemají společné proměnné
  - σ je nejobecnější unifikátor pro A_i a $B\rho$
  - **rezoluce je realizována na literálech $\neg A_i\sigma$ a $B\rho\sigma$**
  - je dodržováno pořadí literálů, tj.
    $\{\neg B_0\rho, \ldots, \neg B_m\rho\}\sigma$ jde do uspořádané rezolventy přesně na pozici $\neg A_i\sigma$

Uspořádané klauzule II.

- Uspořádané klauzule

$$\frac{\{\neg A_0, \ldots, \neg A_n\} \qquad \{B, \neg B_0, \ldots, \neg B_m\}}{\{\neg A_0, \ldots, \neg A_{i-1}, \neg B_0\rho, \ldots, \neg B_m\rho, \neg A_{i+1}, \ldots, \neg A_n\}\sigma}$$

  Hornovy klauzule

$$\frac{:-A_0, \ldots, A_n. \qquad B :- B_0, \ldots, B_m.}{:-(A_0, \ldots, A_{i-1}, B_0\rho, \ldots, B_m\rho, A_{i+1}, \ldots, A_n)\sigma.}$$

Uspořádané klauzule II.

- Uspořádáné klauzule

$$\frac{\{\neg A_0, \ldots, \neg A_n\} \qquad \{B, \neg B_0, \ldots, \neg B_m\}}{\{\neg A_0, \ldots, \neg A_{i-1}, \neg B_0\rho, \ldots, \neg B_m\rho, \neg A_{i+1}, \ldots, \neg A_n\}\sigma}$$

  Hornovy klauzule

$$\frac{:-A_0, \ldots, A_n. \qquad B :- B_0, \ldots, B_m.}{:-(A_0, \ldots, A_{i-1}, B_0\rho, \ldots, B_m\rho, A_{i+1}, \ldots, A_n)\sigma.}$$

- Příklad:

$$\frac{\{\neg s(X), \neg t(1), \neg u(X)\} \qquad \{t(Z), \neg q(Z,X), \neg r(3)\}}{\{\neg s(X), \neg q(1,A), \neg r(3), \neg u(X)\}}$$

$$\frac{:-s(X), t(1), u(X). \qquad t(Z) :- q(Z,X), r(3).}{:-s(X), q(1,A), r(3), u(X).}$$

$$\rho = [X/A] \qquad \sigma = [Z/1]$$

LD-rezoluce

- **LD-rezoluční vyvrácení** množiny uspořádaných klauzulí $P \cup \{G\}$ je posloupnost $\langle G_0, C_0 \rangle, \ldots, \langle G_n, C_n \rangle$ taková, že
  - G_i, C_i jsou uspořádané klauzule
  - $G = G_0$
  - $G_{n+1} = \square$
  - G_i je uspořádaná cílová klauzule
  - C_i je přejmenování klauzule z P
    - C_i neobsahuje proměnné, které jsou v $G_j, j \leq i$ nebo v $C_k, k \leq i$
  - $G_{i+1}, 0 \leq i \leq n$ je uspořádaná rezolventa G_i a C_i

LD-rezoluce

- **LD-rezoluční vyvrácení** množiny uspořádaných klauzulí $P \cup \{G\}$ je posloupnost $\langle G_0, C_0 \rangle, \ldots, \langle G_n, C_n \rangle$ taková, že
  - G_i, C_i jsou uspořádané klauzule
  - $G = G_0$
  - $G_{n+1} = \square$
  - G_i je uspořádaná cílová klauzule
  - C_i je přejmenování klauzule z P
    - C_i neobsahuje proměnné, které jsou v $G_j, j \leq i$ nebo v $C_k, k \leq i$
  - $G_{i+1}, 0 \leq i \leq n$ je uspořádaná rezolventa G_i a C_i
- LD-rezoluce: korektní a úplná

SLD-rezoluce

- **Lineární rezoluce se selekčním pravidlem = SLD-rezoluce (*Selected Linear resolution for Definite clauses*)**
  - rezoluce
  - **Selekční** pravidlo
  - **Lineární** rezoluce
  - ***Definite*** (uspořádané) klauzule
  - vstupní rezoluce

SLD-rezoluce

- **Lineární rezoluce se selekčním pravidlem = SLD-rezoluce (*Selected Linear resolution for Definite clauses*)**
  - rezoluce
  - **Selekční** pravidlo
  - **Lineární** rezoluce
  - ***Definite*** (uspořádané) klauzule
  - vstupní rezoluce
- **Selekční pravidlo** R je funkce, která každé neprázdné klauzuli C přiřazuje nějaký z jejích literálů $R(C) \in C$
  - při rezoluci vybírám z klauzule literál určený selekčním pravidlem

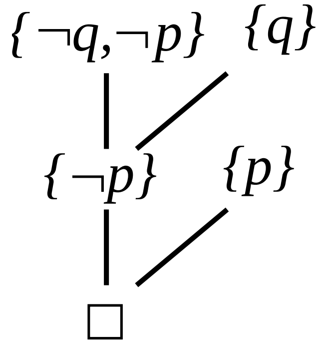
SLD-rezoluce

- **Lineární rezoluce se selekčním pravidlem = SLD-rezoluce (*Selected Linear resolution for Definite clauses*)**
  - rezoluce
  - **Selekční** pravidlo
  - **Lineární** rezoluce
  - ***Definite*** (uspořádané) klauzule
  - vstupní rezoluce
- **Selekční pravidlo** R je funkce, která každé neprázdné klauzuli C přiřazuje nějaký z jejích literálů $R(C) \in C$
  - při rezoluci vybírám z klauzule literál určený selekčním pravidlem
- Pokud se R neuvádí, pak se předpokládá výběr **nejlevějšího literálu**
  - nejlevější literál vybírá i Prolog

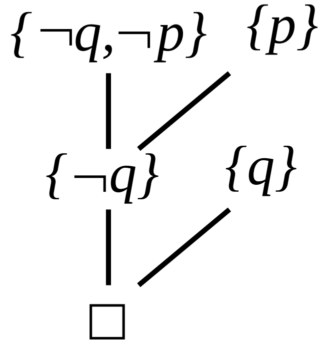
Lineární rezoluce se selekčním pravidlem

- $P = \{\{p\}, \{p, \neg q\}, \{q\}\}, \quad G = \{\neg q, \neg p\}$

výběr nejlevějšího literálu

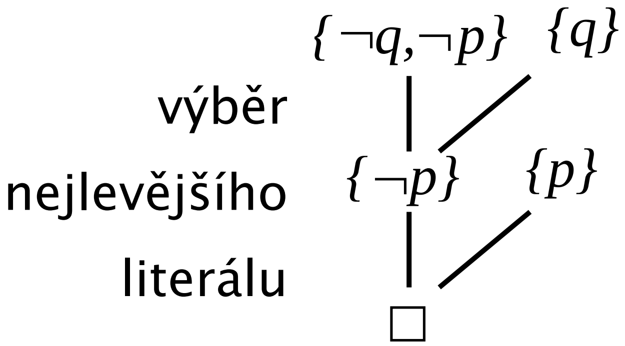


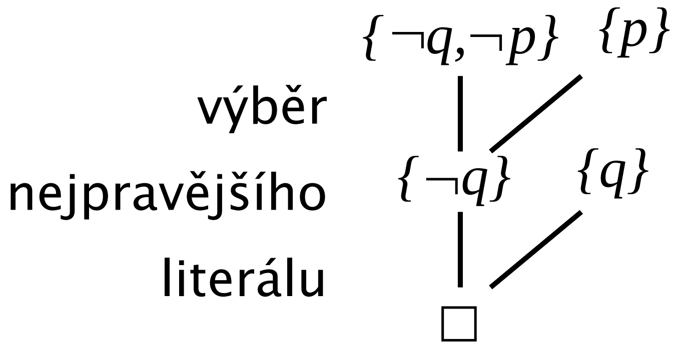
výběr nejpravějšího literálu



Lineární rezoluce se selekčním pravidlem

- $P = \{\{p\}, \{p, \neg q\}, \{q\}\}, \quad G = \{\neg q, \neg p\}$

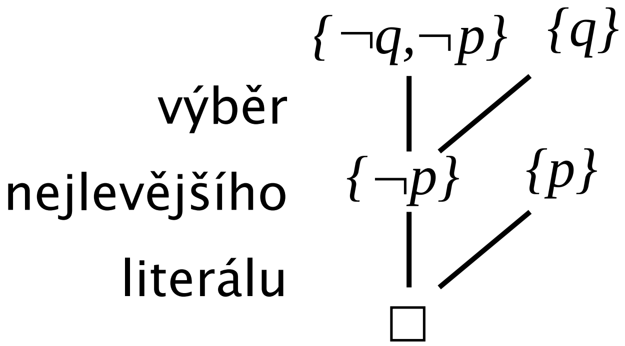


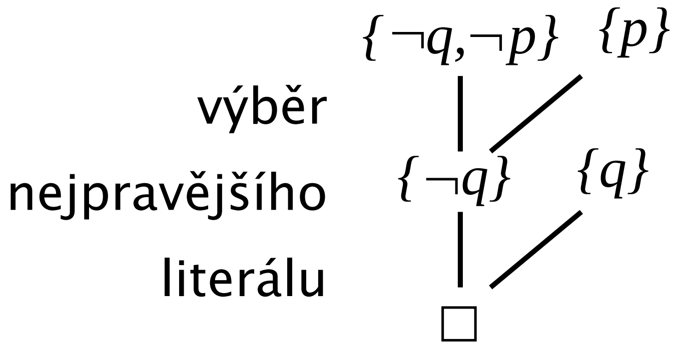


- **SLD-rezoluční vyvrácení** $P \cup \{G\}$ pomocí selekčního pravidla R je LD-rezoluční vyvrácení $\langle G_0, C_0\rangle, \ldots, \langle G_n, C_n\rangle$ takové, že $G = G_0, G_{n+1} = \square$ a $R(G_i)$ je literál rezolvovaný v kroku i

Lineární rezoluce se selekčním pravidlem

- $P = \{\{p\}, \{p, \neg q\}, \{q\}\}, \quad G = \{\neg q, \neg p\}$



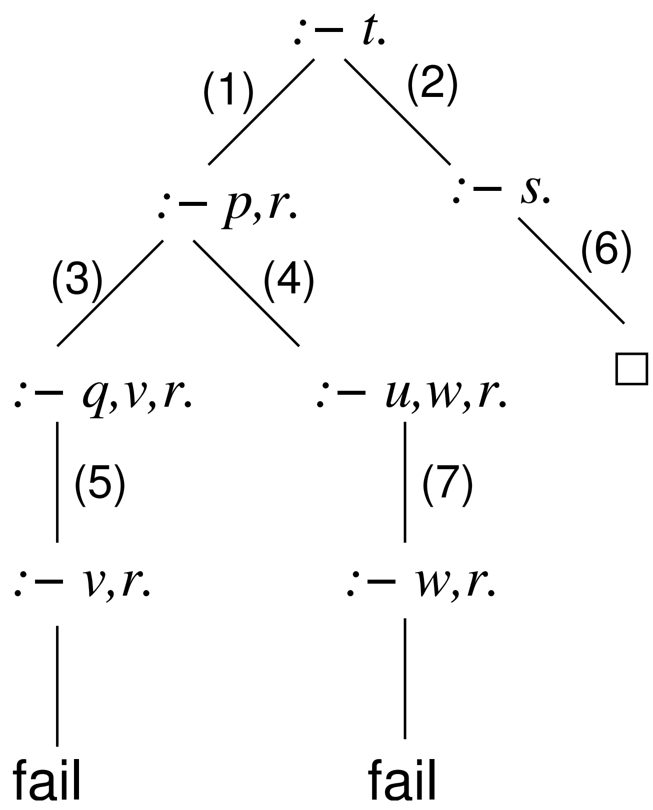


- **SLD-rezoluční vyvrácení** $P \cup \{G\}$ pomocí selekčního pravidla R je LD-rezoluční vyvrácení $\langle G_0, C_0\rangle, \ldots, \langle G_n, C_n\rangle$ takové, že $G = G_0, G_{n+1} = \square$ a $R(G_i)$ je literál rezolvovaný v kroku i
- SLD-rezoluce – korektní, úplná
- Efektivita SLD-rezoluce je závislá na
  - selekčním pravidle R
  - způsobu výběru příslušné programové klauzule pro tvorbu rezolventy
    - v Prologu se vybírá vždy klauzule, která je v programu první

Příklad: SLD-strom

$t :- p, r.$	(1)
$t :- s.$	(2)
$p :- q, v.$	(3)
$p :- u, w.$	(4)
$q.$	(5)
$s.$	(6)
$u.$	(7)

$:- t.$



Strom výpočtu (SLD-strom)

- **SLD-strom** je strom tvořený všemi možnými výpočetními posloupnostmi logického programu P vzhledem k cíli G

Strom výpočtu (SLD-strom)

- **SLD-strom** je strom tvořený všemi možnými výpočetními posloupnostmi logického programu P vzhledem k cíli G
- kořenem stromu je cílová klauzule G
- v uzlech stromu jsou rezolventy (rodiče uzlu a programové klauzule)
  - číslo vybrané programové klauzule pro rezoluci je v příkladu uvedeno jako ohodnocení hrany
- listy jsou dvojího druhu:
  - označené prázdnou klauzulí – jedná se o **úspěšné uzly** (*succes nodes*)
  - označené neprázdnou klauzulí – jedná se o **neúspěšné uzly** (*failure nodes*)

Strom výpočtu (SLD-strom)

- **SLD-strom** je strom tvořený všemi možnými výpočetními posloupnostmi logického programu P vzhledem k cíli G
- kořenem stromu je cílová klauzule G
- v uzlech stromu jsou rezolventy (rodiče uzlu a programové klauzule)
  - číslo vybrané programové klauzule pro rezoluci je v příkladu uvedeno jako ohodnocení hrany
- listy jsou dvojího druhu:
  - označené prázdnou klauzulí – jedná se o **úspěšné uzly** (*succes nodes*)
  - označené neprázdnou klauzulí – jedná se o **neúspěšné uzly** (*failure nodes*)
- úplnost SLD-rezoluce zaručuje **existenci** cesty od kořene k úspěšnému uzlu pro každý možný výsledek příslušející cíli G