Curs 12

2015-2016 Programare Logică

Cuprins

1. Programare logică ecuațională (*)

2. Programare logică clasică (*)

Programare logică ecuațională (*)

Amintiri

- (S, Σ) signatură multisortată și Γ mulțime de ecuații condiționate
- G o mulțime de ecuații de forma $(\forall X)t \doteq_s t'$, $t, t' \in T_\Sigma(X)$.
- Problema programării logice ecuaționale:

$$\Gamma \models (\exists X)G.$$

Soluție

- (S, Σ) signatură multisortată și Γ mulțime de ecuații condiționate
- G o mulțime de ecuații de forma $(\forall X)t \doteq_s t'$, $t, t' \in T_\Sigma(X)$.

Definiție

Soluție

- (S, Σ) signatură multisortată și Γ mulțime de ecuații condiționate
- G o mulțime de ecuații de forma $(\forall X)t \doteq_s t'$, $t, t' \in T_\Sigma(X)$.

Definiție

Un morfism $f : T_\Sigma(X) \to \mathcal{A}$ este soluție pentru $(\exists X)G$ dacă

$$f(G) \subseteq \equiv_{\Gamma,\mathcal{A}}$$

$$\equiv_{\Gamma,\mathcal{A}} := \bigcap \{Ker(h) \mid h : \mathcal{A} \to \mathcal{B},\ \mathcal{B} \models \Gamma\}.$$

Soluție

- (S, Σ) signatură multisortată și Γ mulțime de ecuații condiționate
- G o mulțime de ecuații de forma $(\forall X)t \stackrel{.}{=}_s t'$, $t, t' \in T_\Sigma(X)$.

Definiție

Un morfism $f : T_\Sigma(X) \to \mathcal{A}$ este soluție pentru $(\exists X)G$ dacă

$$f(G) \subseteq \equiv_{\Gamma,\mathcal{A}}$$

$$\equiv_{\Gamma,\mathcal{A}} := \bigcap\{Ker(h) \mid h : \mathcal{A} \to \mathcal{B},\ \mathcal{B} \models \Gamma\}.$$

- Δ este o mulțime de egalități adevărate din $T_\Sigma(X)$.
- Compunerea a două soluții este tot o soluție.

Context extins

- (S, Σ) signatură și X mulțime de variabile
- Un termen $c \in T_\Sigma(X \cup \{z\})_s$ se numește context dacă $nr_z(c) = 1$.
- Pentru un context $c \in T_\Sigma(X \cup \{z\})$, notăm

$$c[z \leftarrow t_0] := \{z \leftarrow t_0\}(c).$$

Context extins

- (S, Σ) signatură și X mulțime de variabile
- Un termen $c \in T_\Sigma(X \cup \{z\})_s$ se numește context dacă $nr_z(c) = 1$.
- Pentru un context $c \in T_\Sigma(X \cup \{z\})$, notăm
$$c[z \leftarrow t_0] := \{z \leftarrow t_0\}(c).$$
- Un context extins este o ecuație de forma
$$c \doteq_s t \text{ sau } t \doteq_s c$$
unde $c \in T_\Sigma(X \cup \{z\})_s$ și $t \in T_\Sigma(X)_s$.
- Notăm un context extins cu C.
- Observăm că $(c \doteq_s t)[z \leftarrow t_0]$ înseamnă $c[z \leftarrow t_0] \doteq_s t$.
- Notăm $C[z \leftarrow t_0]$ cu $C[t_0]$.

Reguli de deducție

Regula Morfismului $\boxed{\dfrac{G}{\theta(G)}}$ G mulțime de ecuații, $\theta : T_\Sigma(X) \to T_\Sigma(Y)$

Reguli de deducție

Regula Morfismului $\boxed{\dfrac{G}{\theta(G)}}$ G mulțime de ecuații, $\theta : T_\Sigma(X) \to T_\Sigma(Y)$

Regula Reflexiei extinse $\boxed{\dfrac{G \cup \{l \doteq_s r\}}{\theta(G)}}$ G mulțime de ecuații, $\theta : T_\Sigma(X) \to T_\Sigma(Y)$ a.î. $\theta_s(l) = \theta_s(r)$

Reguli de deducție

Regula Morfismului $\boxed{\dfrac{G}{\theta(G)}}$ G mulțime de ecuații, $\theta : T_\Sigma(X) \to T_\Sigma(Y)$

Regula Reflexiei extinse $\boxed{\dfrac{G \cup \{l \doteq_s r\}}{\theta(G)}}$ G mulțime de ecuații, $\theta : T_\Sigma(X) \to T_\Sigma(Y)$ a.î. $\theta_s(l) = \theta_s(r)$

Regula Reflexiei $\boxed{\dfrac{G \cup \{l \doteq_s r\}}{\theta(G)}}$ G mulțime de ecuații, $\theta : T_\Sigma(X) \to T_\Sigma(Y)$ a.î. $\theta = cgu(l, r)$

Reguli de deducție

Regula Morfismului $\boxed{\dfrac{G}{\theta(G)}}$ G mulțime de ecuații, $\theta : T_\Sigma(X) \to T_\Sigma(Y)$

Regula Reflexiei extinse $\boxed{\dfrac{G \cup \{l \doteq_s r\}}{\theta(G)}}$ G mulțime de ecuații, $\theta : T_\Sigma(X) \to T_\Sigma(Y)$ a.î. $\theta_s(l) = \theta_s(r)$

Regula Reflexiei $\boxed{\dfrac{G \cup \{l \doteq_s r\}}{\theta(G)}}$ G mulțime de ecuații, $\theta : T_\Sigma(X) \to T_\Sigma(Y)$ a.î. $\theta = cgu(l, r)$

Regula Pararescrierii $\boxed{\dfrac{G \cup \{C[\theta_s(l)]\}}{G \cup \theta(H) \cup \{C[\theta_s(r)]\}}}$ G mulțime de ecuații, $(\forall Y) l \doteq_s r \text{ if } H \in \Gamma$, $\theta : T_\Sigma(Y) \to T_\Sigma(X)$, C context extins

Reguli de deducție

Regula Paramodulației extinse

$$\boxed{\frac{G \cup \{C[a]\}}{\theta(G \cup H \cup \{C[r]\})}}$$

G mulțime de ecuații,
$(\forall Y) l \doteq_s r$ if $H \in \Gamma$,
$X \cap Y = \emptyset$,
$\theta : T_\Sigma(X \cup Y) \to T_\Sigma(Z)$ a.î.
$\theta_s(l) = \theta_s(a)$, $a \in T_\Sigma(X)_s$
C context extins

Reguli de deducție

Regula Paramodulației extinse

$$\boxed{\frac{G \cup \{C[a]\}}{\theta(G \cup H \cup \{C[r]\})}}$$

G mulțime de ecuații,
$(\forall Y) l \doteq_s r$ if $H \in \Gamma$,
$X \cap Y = \emptyset$,
$\theta : T_\Sigma(X \cup Y) \to T_\Sigma(Z)$ a.î.
$\theta_s(l) = \theta_s(a)$, $a \in T_\Sigma(X)_s$
C context extins

Regula Paramodulației

$$\boxed{\frac{G \cup \{C[a]\}}{\theta(G \cup H \cup \{C[r]\})}}$$

G mulțime de ecuații,
$(\forall Y) l \doteq_s r$ if $H \in \Gamma$,
$X \cap Y = \emptyset$,
$\theta : T_\Sigma(X \cup Y) \to T_\Sigma(Z)$ a.î.
$\theta = cgu(l, a)$, $a \in T_\Sigma(X)_s$
C context extins

Legături între regulile de deducție

Paramodulație

Pararescriere

Morfism

Reflexie extinsă

Paramodulație extinsă

Eliminarea egalităților

Reflexie

Legături între regulile de deducție

Paramodulație

Pararescriere

Morfism

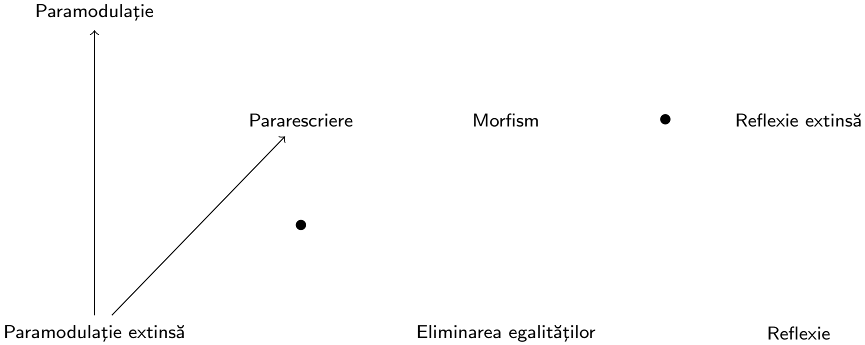
Reflexie extinsă

Paramodulație extinsă

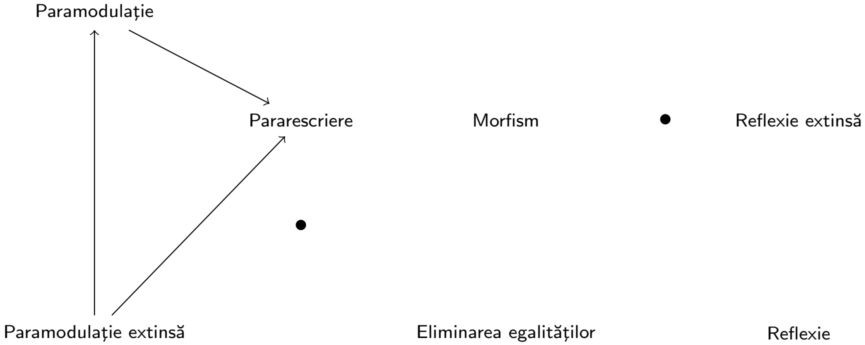
Eliminarea egalităților

Reflexie

Legături între regulile de deducție



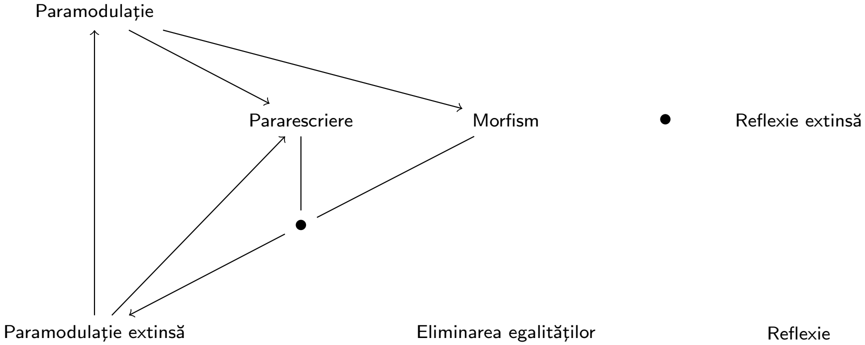
Legături între regulile de deducție



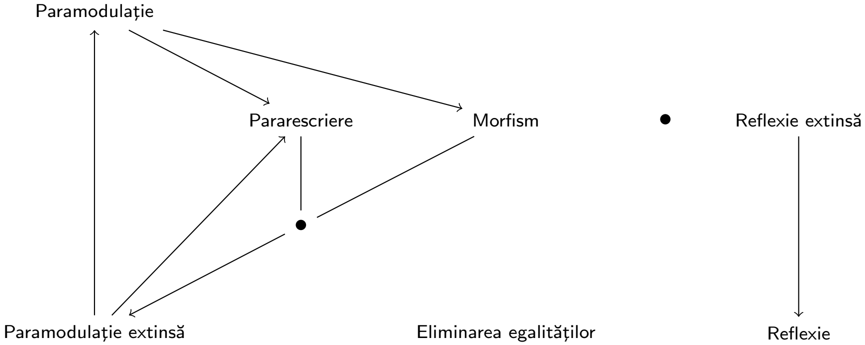
Legături între regulile de deducție

$$
\begin{tikzcd}
\text{Paramodulație} \arrow[dr] \arrow[drr] & & & & \\
& \text{Pararescriere} & \text{Morfism} & \bullet & \text{Reflexie extinsă} \\
& \bullet & & & \\
\text{Paramodulație extinsă} \arrow[uuu] \arrow[uur] & & \text{Eliminarea egalităților} & & \text{Reflexie}
\end{tikzcd}
$$

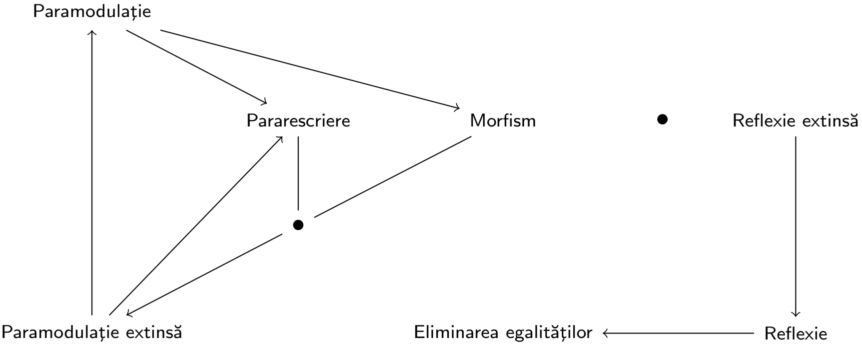
Legături între regulile de deducție



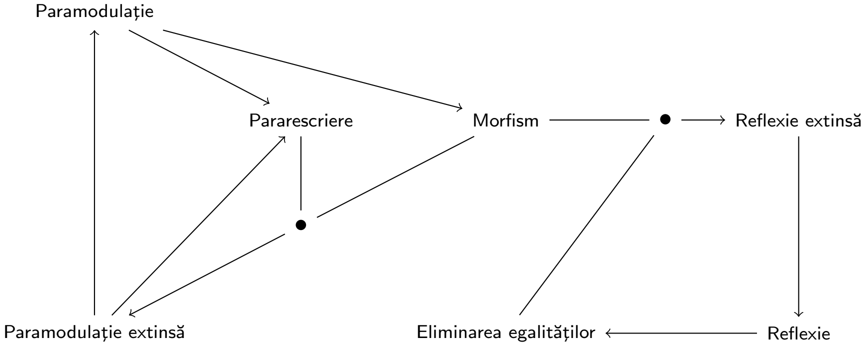
Legături între regulile de deducție



Legături între regulile de deducție



Legături între regulile de deducție



Regula narrowing

Regula Narrowing $\boxed{\dfrac{G \cup \{C[a]\}}{\theta(G \cup H \cup \{C[r]\})}}$

G mulțime de ecuații,
$(\forall Y) l \doteq_s r$ if $H \in \Gamma$,
l nu este variabilă,
$X \cap Y = \emptyset$,
$a \in T_\Sigma(X)_s$, $a \notin X$,
$\theta : T_\Sigma(X \cup Y) \to T_\Sigma(Z)$ a.î.
$\theta = cgu(l, a)$,
C context extins

- Caz particular de Paramodulație.

Exemplu

Exemplu

- $S = \{nat, nlist, list\}$
- $\Sigma = \{0 :\to nat, s : nat \to nat, nil :\to list,$
  $\_,\_ : list\ list \to list, \_,\_ : nlist\ nlist \to nlist,$
  $head : nlist \to nat, cdr : nlist \to list, \# : list \to nat\}$

Exemplu

Exemplu

- $S = \{nat, nlist, list\}$
- $\Sigma = \{0 :\to nat, s : nat \to nat, nil :\to list,$
  $\_,\_ : list\ list \to list, \_,\_ : nlist\ nlist \to nlist,$
  $head : nlist \to nat, cdr : nlist \to list, \# : list \to nat\}$
- $\Gamma = \{(\forall\{E, L\})head(E, L) \doteq E,$
  $(\forall\{E, L\})cdr(E, L) \doteq L,$
  $(\forall\emptyset)\#(nil) \doteq 0,$
  $(\forall\{E, L\})\#(E, L) \doteq s(\#(L))\}$

Exemplu

Exemplu

- $S = \{nat, nlist, list\}$
- $\Sigma = \{0 :\to nat, s : nat \to nat, nil :\to list,$
  $\_,\_ : list\ list \to list, \_,\_ : nlist\ nlist \to nlist,$
  $head : nlist \to nat, cdr : nlist \to list, \# : list \to nat\}$
- $\Gamma = \{(\forall\{E, L\}) head(E, L) \doteq E,$
  $(\forall\{E, L\}) cdr(E, L) \doteq L,$
  $(\forall\emptyset)\#(nil) \doteq 0,$
  $(\forall\{E, L\})\#(E, L) \doteq s(\#(L))\}$

Căutăm o soluție pentru problema:

$$(\exists L)\{\#(L) \doteq s(s(0)), head(L) \doteq 0\}.$$

Exemplu

Exemplu

- $\{\#(L) \doteq s(s(0)), head(L) \doteq 0\}$

Regula Paramodulației

$$\boxed{\frac{G \cup \{C[a]\}}{\theta(G \cup H \cup \{C[r]\})}}$$

G mulțime de ecuații,
$(\forall Y) l \doteq_s r$ if $H \in \Gamma$,
$X \cap Y = \emptyset$,
$\theta : T_\Sigma(X \cup Y) \to T_\Sigma(Z)$ a.î.
$\theta = cgu(l, a)$, $a \in T_\Sigma(X)_s$
C context extins

- $(\forall\{E1, L1\})\#(E1, L1) \doteq s(\#(L1)) \in \Gamma$
- $C : \bullet = s(s(0))$
- $a : \#(L)$
- θ cgu pt $\#(L)$ și $\#(E1, L1)$: $\theta(L) = E1, L1$

Exemplu

Exemplu

- $\{\#(L) \doteq s(s(0)), head(L) \doteq 0\}$

Regula Paramodulației $\boxed{\dfrac{G \cup \{C[a]\}}{\theta(G \cup H \cup \{C[r]\})}}$

G mulțime de ecuații,
$(\forall Y) l \doteq_s r$ if $H \in \Gamma$,
$X \cap Y = \emptyset$,
$\theta : T_\Sigma(X \cup Y) \to T_\Sigma(Z)$ a.î.
$\theta = cgu(l, a)$, $a \in T_\Sigma(X)_s$
C context extins

  - $(\forall\{E1, L1\})\#(E1, L1) \doteq s(\#(L1)) \in \Gamma$
  - $C : \bullet = s(s(0))$
  - $a : \#(L)$
  - θ cgu pt $\#(L)$ și $\#(E1, L1)$: $\theta(L) = E1, L1$

- $\{s(\#(L1)) \doteq s(s(0)), head(E1, L1) \doteq 0\}$ cu morfismul $h_1 : T_\Sigma(\{L\}) \to T_\Sigma(\{E1, L1\})$, $h(L) = E1, L1$

Exemplu

Exemplu

- $\{s(\#(L1)) \doteq s(s(0)), head(E1, L1) \doteq 0\}$

Regula Pararescrierii $\boxed{\dfrac{G \cup \{C[\theta_s(l)]\}}{G \cup \theta(H) \cup \{C[\theta_s(r)]\}}}$

G mulțime de ecuații,
$(\forall Y)l \doteq_s r$ if $H \in \Gamma$,
$\theta : T_\Sigma(Y) \to T_\Sigma(X)$
C context extins

- $\forall\{E, L\})head(E, L) \doteq E \in \Gamma$
- $C : \bullet = 0$
- $\theta : T_\Sigma(\{E, L\}) \to T_\Sigma(\{E1, L1\})$, $\theta(E) = E1$ și $\theta(L) = L1$

Exemplu

Exemplu

- $\{s(\#(L1)) \doteq s(s(0)), head(E1, L1) \doteq 0\}$

Regula Pararescrierii $\boxed{\dfrac{G \cup \{C[\theta_s(l)]\}}{G \cup \theta(H) \cup \{C[\theta_s(r)]\}}}$

G mulțime de ecuații,
$(\forall Y) l \doteq_s r$ if $H \in \Gamma$,
$\theta : T_\Sigma(Y) \to T_\Sigma(X)$
C context extins

  - $\forall\{E, L\}) head(E, L) \doteq E \in \Gamma$
  - $C : \bullet = 0$
  - $\theta : T_\Sigma(\{E, L\}) \to T_\Sigma(\{E1, L1\})$, $\theta(E) = E1$ și $\theta(L) = L1$

- $\{s(\#(L1)) \doteq s(s(0)), E1 \doteq 0\}$ cu morfismul
  $h_2 : T_\Sigma(\{E1, L1\}) \to T_\Sigma(\{E1, L1\})$ identitatea

Exemplu

Exemplu

- $\{s(\#(L1)) \doteq s(s(0)), E1 \doteq 0\}$

Regula Reflexiei $\boxed{\dfrac{G \cup \{l \doteq_s r\}}{\theta(G)}}$ G mulțime de ecuații, $\theta : T_\Sigma(X) \to T_\Sigma(Y)$ a.î. $\theta = cgu(l, r)$

  - $\theta : T_\Sigma(\{E1, L1\}) \to T_\Sigma(\{L1\})$, $\theta(E1) = 0$ este cgu pt $E1$ și 0

Exemplu

Exemplu

- $\{s(\#(L1)) \doteq s(s(0)), E1 \doteq 0\}$

Regula Reflexiei $\boxed{\dfrac{G \cup \{l \doteq_s r\}}{\theta(G)}}$ G mulțime de ecuații, $\theta : T_\Sigma(X) \to T_\Sigma(Y)$ a.î. $\theta = cgu(l, r)$

  - $\theta : T_\Sigma(\{E1, L1\}) \to T_\Sigma(\{L1\})$, $\theta(E1) = 0$ este cgu pt $E1$ și 0

- $\{s(\#(L1)) \doteq s(s(0))\}$ cu morfismul $h_3 : T_\Sigma(\{E1, L1\}) \to T_\Sigma(\{L1\})$, $h_3(E1) = 0$

Exemplu

Exemplu

- $\{s(\#(L1)) \doteq s(s(0))\}$

Regula Paramodulației

$$\boxed{\dfrac{G \cup \{C[a]\}}{\theta(G \cup H \cup \{C[r]\})}}$$

G mulțime de ecuații,
$(\forall Y) l \doteq_s r$ if $H \in \Gamma$,
$X \cap Y = \emptyset$,
$\theta : T_\Sigma(X \cup Y) \to T_\Sigma(Z)$ a.î.
$\theta = cgu(l, a)$, $a \in T_\Sigma(X)_s$
C context extins

- $(\forall\{E, L\})\#(E, L) \doteq s(\#(L)) \in \Gamma$
- $C : s(\bullet) = s(s(0))$
- $a : \#(L1)$
- $\theta(L1) = E, L$ este cgu pt $\#(E, L)$ si $\#(L1)$

Exemplu

Exemplu

- $\{s(\#(L1)) \doteq s(s(0))\}$

Regula Paramodulației $\boxed{\dfrac{G \cup \{C[a]\}}{\theta(G \cup H \cup \{C[r]\})}}$

G mulțime de ecuații,
$(\forall Y) l \doteq_s r$ if $H \in \Gamma$,
$X \cap Y = \emptyset$,
$\theta : T_\Sigma(X \cup Y) \to T_\Sigma(Z)$ a.î.
$\theta = cgu(l, a)$, $a \in T_\Sigma(X)_s$
C context extins

  - $(\forall\{E, L\})\#(E, L) \doteq s(\#(L)) \in \Gamma$
  - $C : s(\bullet) = s(s(0))$
  - $a : \#(L1)$
  - $\theta(L1) = E, L$ este cgu pt $\#(E, L)$ si $\#(L1)$

- $\{s(s(\#(L))) \doteq s(s(0))\}$ cu morfismul $h_4 : T_\Sigma(\{L1\}) \to T_\Sigma(\{E, L\})$, $h_4(L1) = E, L$

Exemplu

Exemplu

- $\{s(s(\#(L))) \doteq s(s(0))\}$

Regula Paramodulației

$$\boxed{\frac{G \cup \{C[a]\}}{\theta(G \cup H \cup \{C[r]\})}}$$

G mulțime de ecuații,
$(\forall Y) l \doteq_s r$ if $H \in \Gamma$,
$X \cap Y = \emptyset$,
$\theta : T_\Sigma(X \cup Y) \to T_\Sigma(Z)$ a.î.
$\theta = cgu(l, a)$, $a \in T_\Sigma(X)_s$
C context extins

- $(\forall \emptyset) \#(nil) \doteq 0 \in \Gamma$
- $C : s(s(\bullet)) = s(s(0))$
- $a : \#(L)$
- $\theta(L) = nil$ este cgu pt $\#(nil)$ și $\#(L)$

Exemplu

Exemplu

- $\{s(s(\#(L))) \doteq s(s(0))\}$

Regula Paramodulației

$$\boxed{\dfrac{G \cup \{C[a]\}}{\theta(G \cup H \cup \{C[r]\})}}$$

G mulțime de ecuații,
$(\forall Y)l \doteq_s r$ if $H \in \Gamma$,
$X \cap Y = \emptyset$,
$\theta : T_\Sigma(X \cup Y) \to T_\Sigma(Z)$ a.î.
$\theta = cgu(l, a)$, $a \in T_\Sigma(X)_s$
C context extins

  - $(\forall\emptyset)\#(nil) \doteq 0 \in \Gamma$
  - $C : s(s(\bullet)) = s(s(0))$
  - $a : \#(L)$
  - $\theta(L) = nil$ este cgu pt $\#(nil)$ și $\#(L)$

- $\{s(s(0)) \doteq s(s(0))\}$ cu morfismul $h_5 : T_\Sigma(\{E, L\}) \to T_\Sigma(\{E\})$, $h_5(L) = nil$

Exemplu

Exemplu

- Un morfism $f : T_\Sigma(X) \to \mathcal{A}$ este soluție pentru $(\exists X)G$ dacă

$$f(G) \subseteq \equiv_{\Gamma,\mathcal{A}}$$

- Soluția cautată este: $h_1; h_2; h_3; h_4; h_5 : T_\Sigma(\{L\}) \to T_\Sigma(\{E\})$

Exemplu

Exemplu

- Un morfism $f : T_\Sigma(X) \to \mathcal{A}$ este soluție pentru $(\exists X)G$ dacă

$$f(G) \subseteq \equiv_{\Gamma,\mathcal{A}}$$

- Soluția cautată este: $h_1; h_2; h_3; h_4; h_5 : T_\Sigma(\{L\}) \to T_\Sigma(\{E\})$
  - $h_1 : T_\Sigma(\{L\}) \to T_\Sigma(\{E1, L1\})$, $h(L) = E1, L1$
  - $h_2 : T_\Sigma(\{E1, L1\}) \to T_\Sigma(\{E1, L1\})$
  - $h_3 : T_\Sigma(\{E1, L1\}) \to T_\Sigma(\{L1\})$, $h_3(E1) = 0$
  - $h_4 : T_\Sigma(\{L1\}) \to T_\Sigma(\{E, L\})$, $h_4(L1) = E, L$
  - $h_5 : T_\Sigma(\{E, L\}) \to T_\Sigma(\{E\})$, $h_5(L) = nil$

$$(h_1; h_2; h_3; h_4; h_5)(\#(L)) = (h_1; h_2; h_3; h_4; h_5)(s(s(0)))$$
$$(h_1; h_2; h_3; h_4; h_5)(head(L)) = (h_1; h_2; h_3; h_4; h_5)(0)$$

Concluzii

Teoremă

În cadrul ecuațional, rezoluția se poate obține din narrowing și eliminarea egalităților adevărate.

Rezoluție = Narrowing = Paramodulație

În ce logică suntem?

În ce logică suntem?

Logica de ordinul I (**FOL**)

În ce logică suntem?

Logica de ordinul I (**FOL**)

- Var mulț. variabilelor,
- $\mathcal{F}$ mulț. simb. de funcții,
- $\mathcal{P}$ mulț. simbolurilor de relații,
- $\dot{=}, \neg, \rightarrow, \vee, \wedge, \forall, \exists$.

În ce logică suntem?

Logica de ordinul I (**FOL**)

- *Var* mulț. variabilelor,
- $\mathcal{F}$ mulț. simb. de funcții,
- $\mathcal{P}$ mulț. simbolurilor de relații,
- $\dot{=}, \neg, \rightarrow, \vee, \wedge, \forall, \exists$.

- Termen: $x \in Var$, $f(t_1, \ldots, t_n)$

În ce logică suntem?

Logica de ordinul I (**FOL**)

- Var mulț. variabilelor,
- $\mathcal{F}$ mulț. simb. de funcții,
- $\mathcal{P}$ mulț. simbolurilor de relații,
- $\doteq, \neg, \rightarrow, \vee, \wedge, \forall, \exists$.

- Termen: $x \in Var$, $f(t_1, \ldots, t_n)$
- Formulă atomică: $P(t_1, \ldots, t_n)$, $t_1 \doteq t_2$

În ce logică suntem?

Logica de ordinul I (**FOL**)

- Var mulț. variabilelor,
- $\mathcal{F}$ mulț. simb. de funcții,
- $\mathcal{P}$ mulț. simbolurilor de relații,
- $\doteq, \neg, \rightarrow, \vee, \wedge, \forall, \exists$.

- Termen: $x \in Var$, $f(t_1, \ldots, t_n)$
- Formulă atomică: $P(t_1, \ldots, t_n)$, $t_1 \doteq t_2$
- Formulă: formulă atomică, $\neg\varphi$, $\varphi \rightarrow \psi$, $\varphi \vee \psi$, $\varphi \wedge \psi$, $(\forall x)\varphi$, $(\exists x)\varphi$

În ce logică suntem?

Logica de ordinul I (**FOL**)

- Var mulț. variabilelor,
- $\mathcal{F}$ mulț. simb. de funcții,
- $\mathcal{P}$ mulț. simbolurilor de relații,
- $\doteq, \neg, \to, \vee, \wedge, \forall, \exists$.

- Termen: $x \in Var$, $f(t_1, \ldots, t_n)$
- Formulă atomică: $P(t_1, \ldots, t_n)$, $t_1 \doteq t_2$
- Formulă: formulă atomică, $\neg\varphi$, $\varphi \to \psi$, $\varphi \vee \psi$, $\varphi \wedge \psi$, $(\forall x)\varphi$, $(\exists x)\varphi$
  - Clauză Horn: $(\forall x_1 \ldots x_k)((Q_1 \wedge \ldots \wedge Q_n) \to Q)$,
    - $Q_1, \ldots, Q_n, Q$ sunt formule atomice
    - Q if $\{Q_1, \ldots, Q_n\}$

Subsisteme ale **FOL**

Subsisteme ale **FOL**

- **HCL**:
  - formulele sunt clauzele Horn
  - fundamentul teoretic al limbajului Prolog

Subsisteme ale **FOL**

- **HCL**:
  - formulele sunt clauzele Horn
  - fundamentul teoretic al limbajului Prolog
- **EQL**:
  - formulele sunt ecuații cuantificate universal
  - $\mathcal{P} = \emptyset$

Subsisteme ale **FOL**

- **HCL**:
  - formulele sunt clauzele Horn
  - fundamentul teoretic al limbajului Prolog
- **EQL**:
  - formulele sunt ecuații cuantificate universal
  - $\mathcal{P} = \emptyset$
- **CEQL**:
  - **HCL** pentru $\mathcal{P} = \emptyset$

Subsisteme ale **FOL**

- **HCL**:
  - formulele sunt clauzele Horn
  - fundamentul teoretic al limbajului Prolog
- **EQL**:
  - formulele sunt ecuații cuantificate universal
  - $\mathcal{P} = \emptyset$
- **CEQL**:
  - **HCL** pentru $\mathcal{P} = \emptyset$

La curs vom folosi **CEQL** (logica ecuațională condiționată) în varianta multisortată!

Programare logică clasică (*)

Signatura multisortata de ordinul I

O signatură multisortată de ordinul I este un triplet (S, Σ, Π) unde:

- (S, Σ) este o signatură multisortată,
- $\Pi = \{\Pi_w\}_{w \in S^+}$ familie de simboluri de predicate.

Signatura multisortata de ordinul I

O signatură multisortată de ordinul I este un triplet (S, Σ, Π) unde:

- (S, Σ) este o signatură multisortată,
- $\Pi = \{\Pi_w\}_{w \in S^+}$ familie de simboluri de predicate.

Exemplu

- $NATPRED = (S, \Sigma, \Pi)$
- $S := \{nat\}$, $\Sigma := \{0 :\to nat, succ : nat \to nat\}$,
- $\Pi_{nat} := \{pos\}$, $\Pi_{nat\ nat} := \{<\}$

Signatura multisortata de ordinul I

Unei signaturi multisortate de ordinul I (S, Σ, Π) îi asociem o signatură multisortată $(S^b, \Sigma^b \cup \Pi^b)$ astfel:

- $S^b := S \cup \{b\}$, unde $b \notin S$,
- $\Sigma^b := \Sigma \cup \{true :\to b\}$,
- $\Pi^b_{w,b} := \Pi_w$ (predicatele devin operații cu rezultat boolean)

Signatura multisortata de ordinul I

Unei signaturi multisortate de ordinul I (S, Σ, Π) îi asociem o signatură multisortată $(S^b, \Sigma^b \cup \Pi^b)$ astfel:

- $S^b := S \cup \{b\}$, unde $b \notin S$,
- $\Sigma^b := \Sigma \cup \{true :\to b\}$,
- $\Pi^b_{w,b} := \Pi_w$ (predicatele devin operații cu rezultat boolean)

Exemplu

- $NATPRED = (S, \Sigma, \Pi)$
  - $S := \{nat\}$,
  - $\Sigma := \{0 :\to nat, succ : nat \to nat\}$,
  - $\Pi_{nat} := \{pos\}$, $\Pi_{nat\ nat} := \{<\}$
- $NATPRED^b = (S^b, \Sigma^b \cup \Pi^b)$
  - $S^b := \{nat, b\}$,
  - $\Sigma^b := \{0 :\to nat, succ : nat \to nat, true :\to b\}$,
  - $\Pi_{nat,b} := \{pos : nat \to b\}$, $\Pi_{nat\ nat,b} := \{<: nat\ nat \to b\}$

Modele și morfisme

- Un (S, Σ, Π)-model este $\mathcal{A} = (A_S, A_\Sigma, A_\Pi)$ unde:
  - (A_S, A_Σ) este o (S, Σ)-algebră,
  - $A_\Pi = \{A_\pi \subseteq A_w \mid \pi \in \Pi_w\}$.

  Notație: $A_w := A_{s_1} \times \cdots \times A_{s_n}$ dacă $w = s_1 \cdots s_n$.

Modele și morfisme

- Un (S, Σ, Π)-model este $\mathcal{A} = (A_S, A_\Sigma, A_\Pi)$ unde:
  - (A_S, A_Σ) este o (S, Σ)-algebră,
  - $A_\Pi = \{A_\pi \subseteq A_w \mid \pi \in \Pi_w\}$.

  Notație: $A_w := A_{s_1} \times \cdots \times A_{s_n}$ dacă $w = s_1 \cdots s_n$.

- Fie $\mathcal{A}$ și $\mathcal{B}$ două (S, Σ, Π)-modele. O funcție $f : A \to B$ este (S, Σ, Π)-morfism dacă:
  - f este morfism de (S, Σ)-algebre,
  - $(a_1, \ldots, a_n) \in A_\pi \Rightarrow (f_{s_1}(a_1), \ldots, f_{s_n}(a_n)) \in B_\pi$, oricare $\pi \in \Pi_{s_1 \cdots s_n}$.

Modele și morfisme

Exemplu

- *NATPRED*-modelul $\mathcal{A}$:
  - $A_{nat} := \mathbb{N}$, $A_0 := 0$, $A_{succ}(n) := n + 1$
  - $A_{pos} := \{n \mid n > 0\}$, $A_< := \{(k, n) \mid k \leq n\}$
- *NATPRED*-modelul $\mathcal{B}$:
  - $B_{nat} := \{2^n | n \in \mathbb{N}\}$, $B_0 := 1$, $B_{succ}(2^n) := 2^{n+1}$
  - $B_{pos} := \{2^n \mid n > 0\}$, $B_< := \{(2^k, 2^n) \mid k \leq n\}$
- $f : \mathcal{A} \to \mathcal{B}$, $f(n) := 2^n$ or. $n \in \mathbb{N}$, morfism de *NATPRED*-modele

Universul Herbrand

(S, Σ, Π) signatură multisortată de ordinul I.

- Mulțimea termenilor fără variabile T_Σ se mai numește și Universul Herbrand.
- Definim (S, Σ, Π)-modelul $\mathcal{T}_{\Sigma,\Pi} := (T_\Sigma, T_\Pi)$, unde
  - T_Σ este (S, Σ)-algebra termenilor fără variabile,
  - $T_\pi := \emptyset$, oricare $\pi \in \Pi$.

Universul Herbrand

(S, Σ, Π) signatură multisortată de ordinul I.

- Mulțimea termenilor fără variabile T_Σ se mai numește și Universul Herbrand.
- Definim (S, Σ, Π)-modelul $T_{\Sigma,\Pi} := (T_\Sigma, T_\Pi)$, unde
  - T_Σ este (S, Σ)-algebra termenilor fără variabile,
  - $T_\pi := \emptyset$, oricare $\pi \in \Pi$.
- $T_{\Sigma,\Pi}$ este (S, Σ, Π)-model inițial.

Clauze

(S, Σ, Π) signatură multisortată de ordinul I.

Definim următoarele formule de ordinul I peste (Σ, Π):

Clauze

(S, Σ, Π) signatură multisortată de ordinul I.

Definim următoarele formule de ordinul I peste (Σ, Π):

- formulă atomică: $\pi(t_1, \cdots, t_n)$, cu $t_i \in T_\Sigma(X)$ or. i

Clauze

(S, Σ, Π) signatură multisortată de ordinul I.

Definim următoarele formule de ordinul I peste (Σ, Π):

- formulă atomică: $\pi(t_1, \cdots, t_n)$, cu $t_i \in T_\Sigma(X)$ or. i
- literal: formulă atomică sau negația unei formule atomice
  - literal pozitiv: $\pi(t_1, \cdots, t_n)$
  - literal negativ: $\neg\pi(t_1, \cdots, t_n)$

Clauze

(S, Σ, Π) signatură multisortată de ordinul I.

Definim următoarele formule de ordinul I peste (Σ, Π):

- formulă atomică: $\pi(t_1, \cdots, t_n)$, cu $t_i \in T_\Sigma(X)$ or. i
- literal: formulă atomică sau negația unei formule atomice
  - literal pozitiv: $\pi(t_1, \cdots, t_n)$
  - literal negativ: $\neg\pi(t_1, \cdots, t_n)$
- clauză: $L_1 \vee \cdots \vee L_m$, unde L_i literal or. i
  (sau echivalent $\{L_1, \ldots, L_m\}$)

Clauze

(S, Σ, Π) signatură multisortată de ordinul I.

Definim următoarele formule de ordinul I peste (Σ, Π):

- formulă atomică: $\pi(t_1, \cdots, t_n)$, cu $t_i \in T_\Sigma(X)$ or. i
- literal: formulă atomică sau negația unei formule atomice
  - literal pozitiv: $\pi(t_1, \cdots, t_n)$
  - literal negativ: $\neg\pi(t_1, \cdots, t_n)$
- clauză: $L_1 \vee \cdots \vee L_m$, unde L_i literal or. i
  (sau echivalent $\{L_1, \ldots, L_m\}$)
- clauza vidă: $\Box$ (disjuncție indexată de $\emptyset$)

Satisfiabilitate

Fie $\mathcal{A}$ un (S, Σ, Π)-model, X o mulț. de variabile și $t_1, \ldots, t_n \in T_\Sigma(X)$.

Satisfiabilitate

Fie $\mathcal{A}$ un (S, Σ, Π)-model, X o mulț. de variabile și $t_1, \ldots, t_n \in T_\Sigma(X)$.

- $\mathcal{A} \models \pi(t_1, \ldots, t_n)$ dacă $(h(t_1), \cdots, h(t_n)) \in A_\pi$, oricare $h : T_\Sigma(X) \to \mathcal{A}$ morfism.

Satisfiabilitate

Fie $\mathcal{A}$ un (S, Σ, Π)-model, X o mulț. de variabile și $t_1, \ldots, t_n \in T_\Sigma(X)$.

- $\mathcal{A} \models \pi(t_1, \ldots, t_n)$ dacă $(h(t_1), \cdots, h(t_n)) \in A_\pi$, oricare $h : T_\Sigma(X) \to \mathcal{A}$ morfism.
- $\mathcal{A} \models \neg\pi(t_1, \ldots, t_n)$ dacă $\mathcal{A} \not\models \pi(t_1, \ldots, t_n)$

Satisfiabilitate

Fie $\mathcal{A}$ un (S, Σ, Π)-model, X o mulț. de variabile și $t_1, \ldots, t_n \in T_\Sigma(X)$.

- $\mathcal{A} \models \pi(t_1, \ldots, t_n)$ dacă $(h(t_1), \cdots, h(t_n)) \in A_\pi$, oricare $h : T_\Sigma(X) \to \mathcal{A}$ morfism.
- $\mathcal{A} \models \neg\pi(t_1, \ldots, t_n)$ dacă $\mathcal{A} \not\models \pi(t_1, \ldots, t_n)$
- $\mathcal{A} \models L_1 \vee \cdots \vee L_m$ dacă $\mathcal{A} \models L_i$ pentru $i \in \{1, \ldots, m\}$

Satisfiabilitate

Fie $\mathcal{A}$ un (S, Σ, Π)-model, X o mulț. de variabile și $t_1, \ldots, t_n \in T_\Sigma(X)$.

- $\mathcal{A} \models \pi(t_1, \ldots, t_n)$ dacă $(h(t_1), \cdots, h(t_n)) \in A_\pi$, oricare $h : T_\Sigma(X) \to \mathcal{A}$ morfism.
- $\mathcal{A} \models \neg\pi(t_1, \ldots, t_n)$ dacă $\mathcal{A} \not\models \pi(t_1, \ldots, t_n)$
- $\mathcal{A} \models L_1 \vee \cdots \vee L_m$ dacă $\mathcal{A} \models L_i$ pentru $i \in \{1, \ldots, m\}$
- Dacă Γ este o mulțime de clauze, atunci

  $$\mathcal{A} \models \Gamma \Leftrightarrow \mathcal{A} \models \gamma, \text{ or. } \gamma \in \Gamma.$$

  (spunem că $\mathcal{A}$ este model pentru Γ)
- O mulțime de clauze Γ se numește satisfiabilă dacă are un model.
- Clauza vidă $\square$ nu este satisfiabilă.

Consecința semantică și satisfiabilitatea

Γ mulțime de clauze și $P_1, \ldots, P_n$ formule atomice cu variabile din X.

Teoremă

Sunt echivalente:

- $\Gamma \models \exists X(P_1 \wedge \cdots \wedge P_n)$,
- $\Gamma \cup \{\neg(\exists X(P_1 \wedge \cdots \wedge P_n))\}$ *nu e satisfiabilă,*
- $\Gamma \cup \{(\forall X(\neg P_1 \vee \cdots \vee \neg P_n))\}$ *nu e satisfiabilă.*

Rezoluția SLD

$$\Gamma \models \exists X(P_1 \wedge \cdots \wedge P_n) \quad \Leftrightarrow \quad \Gamma \cup \{(\forall X(\neg P_1 \vee \cdots \vee \neg P_n))\} \text{ nu e satisfiabilă.}$$

- Rezoluția este o metodă prin care verificăm dacă o mulțime de clauze nu este satisfiabilă.
- Limbajul PROLOG are la bază rezoluția SLD pentru clauze Horn.

SLD= Selected, Linear, Definite

Clauze Horn

- O clauză Horn are cel mult un literal pozitiv.

Clauze Horn

- O clauză Horn are cel mult un literal pozitiv.
- Clauzele Horn au trei forme:

rule (clauză definită)	$\neg P_1 \vee \cdots \vee \neg P_n \vee P$
fact	P
query (clauză scop)	$\neg P_1 \vee \cdots \vee \neg P_n$

unde $P_1, \ldots, P_n, P$ sunt formule atomice.

Clauze Horn

- O clauză Horn are cel mult un literal pozitiv.

- Clauzele Horn au trei forme:

rule (clauză definită)	$\neg P_1 \vee \cdots \vee \neg P_n \vee P$
fact	P
query (clauză scop)	$\neg P_1 \vee \cdots \vee \neg P_n$

  unde P_1, …, P_n, P sunt formule atomice.

- Clauză Horn: $(\forall X)P\ \mathit{if}\ \{P_1, \ldots, P_n\}$

Clauze Horn

- O clauză Horn are cel mult un literal pozitiv.
- Clauzele Horn au trei forme:

rule (clauză definită)	$\neg P_1 \vee \cdots \vee \neg P_n \vee P$
fact	P
query (clauză scop)	$\neg P_1 \vee \cdots \vee \neg P_n$

  unde P_1, ..., P_n, P sunt formule atomice.

- Clauză Horn: $(\forall X) P \text{ if } \{P_1, \ldots, P_n\}$
- Notația Prolog este:
  - rule: $P :- P_1, \cdots, P_n$
  - fact: P
  - query: $:- P_1, \cdots, P_n$

  În această notație toate variabilele sunt cuantificate universal.

Clauze Horn ecuaționale

(S, Σ, Π) signatură multisortată de ordinul I și $(S^b, \Sigma^b \cup \Pi^b)$ signatura multisortată asociată.

- Unei clauze Horn
$$(\forall X)P \; if \{P_1, \ldots, P_n\}$$
îi asociem ecuația condiționată
$$(\forall X)P \doteq_b true \; if \{P_1 \doteq_b true, \ldots, P_n \doteq_b true\}.$$

- Unui (S, Σ, Π)-model $\overline{\mathcal{A}}$ îi asociem o $(S^b, \Sigma^b \cup \Pi^b)$-algebră $\mathcal{A}$
$$(a_1, \ldots, a_m) \in \overline{A}_\pi \Leftrightarrow A_\pi(a_1, \ldots, a_m) = A_{true}$$

- Sunt echivalente:
  - $\overline{\mathcal{A}} \models_{\Sigma,\Pi} (\forall X)P \; if \{P_1, \ldots, P_n\}$
  - $\mathcal{A} \models_{\Sigma^b,\Pi^b} (\forall X)P \doteq_b true \; if \{P_1 \doteq_b true, \ldots, P_n \doteq_b true\}$

Rezoluția SLD

Γ mulțime de clauze definite.

$$\text{SLD}\quad \boxed{\dfrac{\neg P_1 \vee \cdots \vee \neg P_i \vee \cdots \vee \neg P_n}{(\neg P_1 \vee \cdots \vee \neg Q_1 \vee \cdots \vee \neg Q_m \vee \cdots \vee \neg P_n)\theta}}$$

unde

- $Q \vee \neg Q_1 \vee \cdots \vee \neg Q_m$ este o clauză definită din Γ (în care toate variabilele au fost redenumite) și
- θ este c.g.u pentru P_i și Q.

Rezoluția SLD

Γ mulțime de clauze definite.

$$\text{SLD} \quad \boxed{\frac{\neg P_1 \lor \cdots \lor \neg P_i \lor \cdots \lor \neg P_n}{(\neg P_1 \lor \cdots \lor \neg Q_1 \lor \cdots \lor \neg Q_m \lor \cdots \lor \neg P_n)\theta}}$$

unde

- $Q \lor \neg Q_1 \lor \cdots \lor \neg Q_m$ este o clauză definită din Γ (în care toate variabilele au fost redenumite) și
- θ este c.g.u pentru P_i și Q.

Notația Prolog pentru rezoluția SLD:

- G_i este $:- \ P_1, \cdots, P_i, \cdots, P_n$
- C_i este $Q :- \ Q_1, \cdots, Q_m$
- $\theta_i(Q) = \theta_i(P_i)$
- G_{i+1} este $:- \ (P_1, \cdots, Q_1, \cdots, Q_m, \cdots, P_n)\theta_i$
  unde $P\theta := \theta(P)$, $\theta(p(t)) := p(\theta(t))$ cu $p \in \Pi$

Rezoluția SLD

Γ mulțime de clauze definite și G clauza scop.

- O derivare din Γ prin rezoluție SLD este o secvența

$$G_0 := G,\ G_1,\ \ldots,\ G_k,\ \ldots$$

  în care G_{i+1} se obține din G_i prin regula SLD.
- Dacă există un k cu $G_k = \square$ atunci derivarea se numește SLD-respingere.

Rezoluția SLD

Γ mulțime de clauze definite și G clauza scop.

- O derivare din Γ prin rezoluție SLD este o secvența

$$G_0 := G,\ G_1,\ \ldots,\ G_k,\ \ldots$$

  în care G_{i+1} se obține din G_i prin regula SLD.
- Dacă există un k cu $G_k = \square$ atunci derivarea se numește SLD-respingere.

Teoremă (Completitudinea SLD-rezoluției)

Fie G clauza $:- P_1, \cdots, P_n$. Sunt echivalente:

- *există o SLD-respingere a lui G din Γ,*
- $\Gamma \cup \{(\forall X(\neg P_1 \vee \cdots \vee \neg P_n))\}$ *nu e satisfiabilă,*
- $\Gamma \models \exists X(P_1 \wedge \cdots \wedge P_n)$.

Rezoluția SLD

Exemplu

- Fie Γ următoarea mulțime de clauze Horn (bază de date):
  1. $stramos(X, Y) :- parinte(X, Y)$
  2. $stramos(X, Y) :- parinte(X, Z), stramos(Z, Y)$
  3. $parinte(dan, bogdan)$
  4. $parinte(bogdan, ana)$

- Găsiți o respingere din Γ pentru

$$:- stramos(Y, bogdan), stramos(bogdan, Z)$$

  - există Y și Z astfel încât
  - Y este *stramos* al lui *bogdan* și
  - *bogdan* este *stramos* al lui Z

Rezoluția SLD

Exemplu (cont.)

Vom nota $p := parinte$, $q := stramos$ (predicate binare)
$b := bogdan,\ d := dan,\ a := ana$ (constante)

$G_0 : q(Y, b),\ q(b, Z)$

---

(1) $q(X', Y') : -p(X', Y'),\ \theta_1 := \{X' \leftarrow Y, Y' \leftarrow b\}$
$G_1 : p(Y, b),\ q(b, Z)$

---

(3) $p(d, b),\ \theta_2 := \{\ Y \leftarrow d\ \}$
$G_2 : q(b, Z)$

---

(1) $q(X'', Y'') : -p(X'', Y''),\ \theta_3 := \{X'' \leftarrow b, Y'' \leftarrow Z\}$
$G_3 : p(b, Z)$

---

(4) $p(b, a),\ \theta_4 := \{\ Z \leftarrow a\ \}$
$G_4 : \square$

Pe săptămâna viitoare!