FMI, Info, Anul II, 2019-2020
Programare logică

Seminar 6-7
Rezoluţie. Rezoluţie SLD

Teorie pentru S6-7.1:

Rezoluţia în calculul propoziţional

- În calculul propoziţional un literal este o variabilă sau negaţia unei variabile.

$$literal := \ p \mid \neg p \qquad \text{unde } p \text{ este variabilă propoziţională}$$

- O formulă este în formă normală conjunctivă (FNC) dacă este o conjuncţie de disjuncţii de literali.
- Pentru orice formulă α din există o FNC α^{fc} astfel încât $\alpha \mathrel{\text{⫤}\!\models} \alpha^{fc}$.
- O clauză este o disjuncţie de literali.
- Observăm că o FNC este o conjuncţie de clauze.
- Clauza vidă $\square$ nu este satisfiabilă.
- Mulţimea de clauze vidă $\{\}$ este satisfiabilă.
- Dacă φ este o formulă în calculul propoziţional, atunci

$$\varphi^{fc} = \bigwedge_{i=1}^{k} \bigvee_{j=1}^{n_i} L_{ij} \text{ unde } L_{ij} \text{ sunt literali}$$

- Ştim că:

$$\varphi \text{ este satisfiabilă dacă şi numai dacă}$$

$$\varphi^{fc} \text{ este satisfiabilă dacă şi numai dacă}$$

$$\{\{L_{11}, \ldots, L_{1n_1}\}, \ldots, \{L_{k1}, \ldots, L_{kn_k}\}\} \text{ este satisfiabilă}$$

- Regula Rezoluţiei păstrează satisfiabilitatea:

$$Rez \ \frac{C_1 \cup \{p\}, C_2 \cup \{\neg p\}}{C_1 \cup C_2}$$

unde C_1, C_2 clauze, iar p este variabila propoziţională astfel încât
$\{p, \neg p\} \cap C_1 = \emptyset$ şi $\{p, \neg p\} \cap C_2 = \emptyset$.

- Algoritmul Davis-Putnam:
Intrare: o mulţime $\mathcal{C}$ de clauze

Se repetă următorii paşi:

  - se elimină clauzele triviale
  - se alege o variabilă p
  - se adaugă la mulţimea de clauze toţi rezolvenţii obţinuţi prin aplicarea Rez pe variabila p
  - se şterg toate clauzele care conţin p sau $\neg p$

Ieşire: dacă la un pas s-a obţinut $\square$, mulţimea $\mathcal{C}$ nu este satisfiabilă;
altfel $\mathcal{C}$ este satisfiabil

(S6-7.1) Folosind algoritmul Davis-Putnam, cercetaţi dacă următoarea mulţime de clauze din calculul propoziţional este satisfiabilă:

$$\mathcal{C} = \{\{v_0\}, \{\neg v_0, v_1\}, \{\neg v_1, v_2, v_3\}, \{\neg v_3, v_4\}, \{\neg v_4\}, \{\neg v_2\}\}$$

Demonstraţie:

Pasul 1.
Alegem variabila v_0 şi selectăm $\mathcal{C}_0^{v_0} := \{\{v_0\}\}$, $\mathcal{C}_0^{\neg v_0} = \{\{\neg v_0, v_1\}\}$.
Mulţimea rezolvenţilor posibili este $\mathcal{R}_0 := \{\{v_1\}\}$.
Se elimină clauzele în care apare v_0, adăugăm rezolvenţii şi obţinem:

$$\mathcal{C}_1 := \{\{\neg v_1, v_2, v_3\}, \{\neg v_3, v_4\}, \{\neg v_4\}, \{\neg v_2\}, \{v_1\}\}$$

Pasul 2.
Alegem variabila v_1 şi selectăm $\mathcal{C}_1^{v_1} := \{\{v_1\}\}$ şi $\mathcal{C}_1^{\neg v_1} := \{\{\neg v_1, v_2, v_3\}\}$.
Mulţimea rezolvenţilor posibili este $\mathcal{R}_1 := \{\{v_2, v_3\}\}$.

Se elimină clauzele în care apare v_1, adăugăm rezolvenţii şi obţinem: $\mathcal{C}_2 := \{\{\neg v_3, v_4\}, \{\neg v_4\}, \{\neg v_2\}, \{v_2, v_3\}\}$.
Pasul 3.
Alegem variabila v_2 şi selectăm $\mathcal{C}_2^{v_2} := \{\{v_2, v_3\}\}$, $\mathcal{C}_2^{\neg v_2} := \{\{\neg v_2\}\}$.
Mulţimea rezolvenţilor posibili este $\mathcal{R}_2 := \{\{v_3\}\}$.
Se elimină clauzele în care apare v_2, adăugăm rezolvenţii şi obţinem: $\mathcal{C}_3 := \{\{\neg v_3, v_4\}, \{\neg v_4\}, \{v_3\}\}$.
Pasul 4.
Alegem variabila v_3 şi selectăm $\mathcal{C}_3^{v_3} := \{\{v_3\}\}$, $\mathcal{C}_3^{\neg v_3} := \{\{\neg v_3, v_4\}\}$.
Mulţimea rezolvenţilor posibili este $\mathcal{R}_3 := \{\{v_4\}\}$.
Se elimină clauzele în care apare v_3, adăugăm rezolvenţii şi obţinem: $\mathcal{C}_4 := \{\{\neg v_4\}, \{v_4\}\}$.
Pasul 5.
Alegem variabila v_4 şi selectăm $\mathcal{C}_4^{v_4} := \{\{v_4\}\}$, $\mathcal{C}_4^{\neg v_4} := \{\{\neg v_4\}\}$.
Mulţimea rezolvenţilor posibili este $\mathcal{R}_4 := \{\square\}$.
Se elimină clauzele în care apare v_4, adăugăm rezolvenţii şi obţinem: $\mathcal{C}_5 := \{\square\}$.
Deoarece $\mathcal{C}_5 = \{\square\}$, obţinem că mulţimea de clauze $\mathcal{C}$ nu este satisfiabilă. $\square$

Teorie pentru S6-7.2:

Rezoluţia pentru clauze închise în logica de ordinul I

- În logica de ordinul I un literal este o formulă atomică sau negaţia unei formule atomice.

$$\mathit{literal} := \ P(t_1, \ldots, t_n) \mid \neg P(t_1, \ldots, t_n)$$

  unde $P \in \mathbf{R}, ari(P) = n$, şi $t_1, \ldots, t_n$ sunt termeni.

- Pentru un literal L vom nota cu L^c literalul complement.

  De exemplu, dacă $L = \neg P(x)$ atunci $L^c = P(x)$ şi invers.

- O formulă φ este formă clauzală dacă

$$\varphi := \forall x_1 \ldots \forall x_n \psi \text{ unde } \psi \text{ este FNC}$$

- Pentru orice formulă φ din logica de ordinul I există o formă clauzală φ^{fc} astfel încât

$$\varphi \text{ este satisfiabilă dacă şi numai dacă } \varphi^{fc} \text{ este satisfiabilă}$$

- Pentru o formulă φ, forma clauzală φ^{fc} se poate calcula astfel:

  (i) se determină forma rectificată

  (ii) se cuantifică universal variabilele libere

  (iii) se determină forma prenex

  (iv) se determină forma Skolem

  în acest moment am obţinut o formă Skolem $\forall x_1 \ldots \forall x_n \psi$

  (v) se determină o FNC ψ' astfel încât $\psi \mathrel{\text{⫤}\!\!\vDash} \psi'$

(vi) φ^{fc} este $\forall x_1 \ldots \forall x_n \psi'$

- Fie C o clauză. Spunem că C' este o instanţă a lui C dacă există o substituţie $\theta : V \to Trm_{\mathcal{L}}$ astfel încât $C' = \theta(C)$.

  Spunem că C' este o instanţă închisă a lui C dacă există o substituţie $\theta : V \to T_{\mathcal{L}}$ astfel încât $C' = \theta(C)$ (C' se obţine din C înlocuind variabilele cu termeni din universul Herbrand)

- Fie $\mathcal{C}$ o mulţime de clauze. Definim

$$\mathcal{H}(\mathcal{C}) := \{\theta(C) \mid C \in \mathcal{C}, \theta : V \to T_{\mathcal{L}}\}$$

  O mulţime de clauze $\mathcal{C}$ este nesatisfiabilă dacă şi numai dacă există o submulţime finită a lui $\mathcal{H}(\mathcal{C})$ care este nesatisfiabilă.

  $\mathcal{H}(\mathcal{C})$ este mulţimea instanţelor închise ale clauzelor din $\mathcal{C}$.

- Rezoluţia pe clauze închise păstrează satisfiabilitaea

$$Rez \ \frac{C_1 \cup \{L\}, C_2 \cup \{\neg L\}}{C_1 \cup C_2}$$

  unde C_1, C_2 clauze închise, iar L este o formulă atomică închisă astfel încât $\{L, \neg L\} \cap C_1 = \emptyset$ şi $\{L, \neg L\} \cap C_2 = \emptyset$

(S6-7.2) Considerăm următoarea mulţime de clauze în logica de ordinul I:

$$\mathcal{C} = \{\, \{\neg P(f(a)), Q(y)\}, \, \{P(y)\}, \, \{\neg Q(b)\} \,\}$$

Arătaţi că $\mathcal{C}$ nu este satisfiabilă prin următoarele metode:

1) Găsiţi o submulţime finită nesatisfiabilă lui $\mathcal{H}(\mathcal{C})$.

2) Găsiţi o derivare pentru $\square$ folosind rezoluţia pe clauze închise.

Demonstraţie:

1) $\mathcal{H}(\mathcal{C}) = \{\, \{\neg Q(b)\}, \, \{\neg P(f(a)), Q(a)\}, \{\neg P(f(a)), Q(b)\}, \, \{P(a)\}, \, \{P(b)\}, \, \{P(f(a))\}, \cdots \}$

   Submulţimea nesatisfiabilă este:

   $\{\{\neg P(f(a)), Q(b)\}, \, \{P(f(a))\}, \, \{\neg Q(b)\}\}$

   Explicaţie: Se face tabelul de adevăr pentru formula: $(\neg P(f(a)) \vee Q(b)) \wedge P(f(a)) \wedge \neg Q(b)$ şi se observă că este nesatisfiabilă (adică o contradicţie).

2) Derivare pentru $\square$:

$C_1 = \{\neg P(f(a)), Q(b)\}$
$C_2 = \{P(f(a))\}$
$C_3 = \{\neg Q(b)\}$
$C_4 = \{Q(b)\}$ din Rez, C_1, C_2
$C_5 = \square$ din Rez, C_3, C_4

$\square$

Teorie pentru S6-7.3, S6-7.4:

Regula rezoluţiei pentru clauze arbitrare

- Regula rezoluţiei păstrează satisfiabilitatea:

$$Rez \ \frac{C_1, C_2}{(\sigma C_1 \setminus \sigma Lit_1) \cup (\sigma C_2 \setminus \sigma Lit_2)}$$

  dacă următoarele condiţii sunt satisfăcute:

  (i) C_1, C_2 clauze care nu au variabile comune,

  (ii) $Lit_1 \subseteq C_1$ şi $Lit_2 \subseteq C_2$ sunt mulţimi de literali,

  (iii) σ este un cgu pentru Lit_1 şi Lit_2^c, adică
  σ unifică toţi literalii din Lit_1 şi Lit_2^c.

- O clauză C se numeşte rezolvent pentru C_1 şi C_2 dacă există o redenumire de variabile $\theta : V \to V$ astfel încât C_1 şi θC_2 nu au variabile comune şi C se obţine din C_1 şi θC_2 prin Rez.

- Fie $\mathcal{C}$ o mulţime de clauze. O derivare prin rezoluţie din mulţimea $\mathcal{C}$ pentru o clauză C este o secvenţă $C_1, \ldots, C_n$ astfel încât $C_n = C$ şi, pentru fiecare $i \in \{1, \ldots, n\}$, $C_i \in \mathcal{C}$ sau C_i este un rezolvent pentru două cauze C_j, C_k cu $j, k < i$.

- O mulţime de clauze $\mathcal{C}$ este nesatisfiabilă dacă şi numai dacă există o derivare a clauzei vide $\square$ din $\mathcal{C}$ prin Rez.

(S6-7.3) Găsiţi doi rezolvenţi pentru următoarele clauze:

$$\begin{aligned} C_1 &= \{P(x), P(g(y)), Q(x)\} \\ C_2 &= \{\neg P(x), R(f(x), a)\} \end{aligned}$$

unde P, Q, R sunt simboluri de relaţii, a este o constantă, x, y sunt variabile.

Demonstraţie: Se redenumeşte $C_2' = \{\neg P(z), R(f(z), a)\}$

Rezolvent 1: $Lit_1 = \{P(x)\}$, $Lit_2 = \{\neg P(z)\}$, substituţie $\sigma = \{z \leftarrow x\}$,
rezolvent $C = \{P(g(y)), Q(x), R(f(x), a)\}$

Rezolvent 2: $Lit_1 = \{P(x), P(g(y))\}$, $Lit_2 = \{\neg P(z)\}$, substituţie $\sigma = \{z \leftarrow g(y), x \leftarrow g(y)\}$,
rezolvent $C = \{Q(g(y)), R(f(g(y)), a)\}$

$\square$

(S6-7.4) Găsiţi o derivare prin rezoluţie a $\square$ pentru următoarea mulţime de clauze:

$C_1 = \{\, \neg P(x),\ R(x, f(x))\, \}$
$C_2 = \{\, \neg R(a, x), Q(x)\, \}$
$C_3 = \{P(a)\}$
$C_4 = \{\neg Q(f(x))\}$

unde P, Q, R sunt simboluri de relaţii, f e simbol de funcţie, a este o constantă, x, y sunt variabile.
Demonstraţie:

$C_5 = \{R(a, f(a))\}$ din $Rez, C_1, C_3, \sigma = \{x \leftarrow a\}$
$C_4' = \{\neg Q(f(z))\}$ redenumire
$C_6 = \{\neg R(a, f(z))\}$ din $Rez, C_4', C_2, \sigma = \{y \leftarrow f(z)\}$
$C_7 = \square$ din $Rez, C_6, C_5, \sigma = \{z \leftarrow a\}$ $\square$

Teorie pentru S6-7.5, S6-7.6:

Deducţie şi satisfiabilitate

- Dacă $\varphi_1, \ldots, \varphi_n, \varphi$ sunt formule (în logica propoziţională sau calculul cu predicate) atunci:
  $\{\varphi_1, \ldots, \varphi_n\} \vDash \varphi$ este echivalent cu
  $\vDash \varphi_1 \wedge \ldots \wedge \varphi_n \rightarrow \varphi$ este echivalent cu
  $\vDash \neg\varphi_1 \vee \ldots \neg\varphi_n \vee \varphi$ este echivalent cu
  $\varphi_1 \wedge \ldots \wedge \varphi_n \wedge \neg\varphi$ este satisfiabilă.
- În particular, $\vDash \varphi$ dacăşi numai dacă există o derivare pentru $\square$ din forma clauzală a lui $\neg\varphi$.
- Pentru a cerceta satisfiabilitatea este suficient să studiem forme clauzale.

(S6-7.5) Folosind rezoluţia, arătaţi că următoarea formulă este validă în logica de ordinul I:

$$\varphi := (\forall x\, (P(x) \rightarrow Q(x))) \rightarrow (\exists x\, P(x) \rightarrow \exists x\, Q(x))$$

Indicaţie: se găseşte o derivare pentru $\square$ din forma clauzală a lui $\neg\varphi$.

Demonstraţie:

O formă prenex a $\neg\varphi$ este:

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

Atunci, $(\neg\varphi)^{fc}$ este identică cu forma Skolem a acesteia:

$$\forall x((\neg P(x) \vee Q(x)) \wedge P(c) \wedge \neg Q(x))$$

Deci forma clauzală este:

$$\mathcal{C} = \{\, \{\neg P(x), Q(x)\},\ \{P(c)\},\ \{\neg Q(x)\}$$

Derivare prin rezoluţie pentru $\square$:
$C_1 = \{\neg P(x), Q(x)\}$
$C_2 = \{P(c)\}$
$C_3 = \{\neg Q(x)\}$
$C_4 = \{Q(c)\}$ din $Rez, C_1, C_2, \sigma = \{x \leftarrow c\}$
$C_5 = \square$ din $Rez, C_3, C_4, \sigma = \{x \leftarrow c\}$

$\square$

(S6-7.6) Avem următorul raţionament:

”*Există elevi cărora le plac toate lecturile. Nici unui elev nu îi plac lucrurile plictisitoare. În consecinţă, nici o lectură nu este plictisitoare.*”

Definim predicatele
$E(x) =$ ”x este elev”
$L(x) =$ ”x este lectură”
$P(x) =$ ”x este plictisitor”
$R(x, y) =$ ”x place y”

1) Folosind predicatele E, L, P, R, exprimaţi fiecare afirmaţie în logica de ordinul I.

2) Demonstraţi prin rezoluţie că raţionamentul este corect.

Demonstraţie:

$\varphi_1 := \exists x(E(x) \wedge \forall y(L(y) \to R(x, y)))$
$\varphi_2 := \forall x(E(x) \to \forall y(P(y) \to \neg R(x, y)))$
$\psi := \forall x(L(x) \to \neg P(x))$
Calculăm formele clauzale pentru φ_1, φ_2 şi $\neg\psi$:

pt φ_1: $\mathcal{C}_1 = \{\, \{E(a)\}, \, \{\neg L(y), R(a,y)\} \,\}$
pt φ_2: $\mathcal{C}_2 = \{\, \{\neg E(x), \neg P(y), \neg R(x,y)\} \,\}$
pt $\neg\psi$: $\mathcal{C} = \{\, \{L(b)\}, \, \{P(b)\} \,\}$
unde a, b sunt constantele care apar din Skolemizare.
$\{\varphi_1, \varphi_2\} \vDash \psi$ ddacă există o derivare pentru $\square$ din $\mathcal{C}_1 \cup \mathcal{C}_1 \cup \mathcal{C}$.

$C_1 = \{\neg L(y), R(a,y)\}$
$C_2 = \{L(b)\}$
$C_3 = \{\neg E(x), \neg P(y), \neg R(x,y)\}$
$C_4 = \{P(b)\}$
$C_5 = \{E(a)\}$
$C_6 = \{R(a,b)\} \quad Rez, C_1, C_2, \theta = \{y \leftarrow b\}$
$C_7 = \{\neg E(x), \neg R(x,b)\} \quad Rez, C_3, C_4, \theta = \{y \leftarrow b\}$
$C_8 = \{\neg R(a,b)\} \quad Rez, C_5, C_7, \theta = \{x \leftarrow a\}$
$C_9 = \square$ $\square$

Teorie pentru S6-7.7:

- O *clauză definită* este o formulă de forma:
  - $P(t_1, \ldots, t_n)$ (formulă atomică), unde P este un simbol de predicat, iar $t_1, \ldots, t_n$ termeni
  - $P_1 \wedge \ldots \wedge P_n \to Q$, unde toate P_i, Q sunt formule atomice.
- O regulă din Prolog $\mathtt{Q :- P_1, \ldots, P_n}$ este o clauză $P_1 \wedge \ldots \wedge P_n \to Q$, iar un fapt din Prolog $\mathtt{P(t_1, \ldots, t_n)}$ este o formulă atomică $P(t_1, \ldots, t_n)$.
- O clauză definită $P_1 \wedge \ldots \wedge P_n \to Q$ poate fi gândită ca formula $Q \vee \neg P_1 \vee \ldots \vee \neg P_n$.
- Pentru o mulţime de clauze definite T, *regula rezolutiei SLD* este

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

  unde $Q \vee \neg Q_1 \vee \cdots \vee \neg Q_m$ este o clauză definită din T (în care toate variabilele au fost redenumite) şi θ este c.g.u pentru P_i şi Q.
- Fie T o mulţime de clauze definite şi $P_1 \wedge \ldots \wedge P_m$ o ţintă, unde P_i sunt formule atomice. O derivare din T prin rezoluţie SLD este o secvenţă $G_0 := \neg P_1 \vee \ldots \vee \neg P_m$, G_1, …, G_k, … în care G_{i+1} se obţine din G_i prin regula SLD. Dacă există un k cu $G_k = \square$ (clauza vidă), atunci derivarea se numeşte *SLD-respingere*.

Teorema 1 (Completitudinea SLD-rezoluţiei)**.** *Sunt echivalente:*

(i) există o SLD-respingere a lui $P_1 \wedge \ldots \wedge P_m$ din T,

(ii) $T \vDash P_1 \wedge \cdots \wedge P_m$.

(S6-7.7) Găsiţi o SLD-respingere pentru următoarele programe Prolog şi ţinte:

(a)
```
1. r :- p,q.      5. t.      ?- w.
2. s :- p,q.      6. q.
3. v :- t,u.      7. u.
4. w :- v,s.      8. p.
```

(b)
```
1. q(X,Y) :- q(Y,X), q(Y,f(f(Y))).      ?- q(f(Z),a).
2. q(a,f(f(X))).
```

(c)
```
1. p(X) :- q(X,f(Y)), r(a).      4. r(X) :- q(X,Y).      ?- p(X), q(Y,Z).
2. p(X) :- r(X).                 5. r(f(b)).
3. q(X,Y) :- p(Y).
```

Demonstraţie:

(a)

$G_0 = \neg w$	
$G_1 = \neg v \vee \neg s$	(4)
$G_2 = \neg t \vee \neg u \vee \neg s$	(3)
$G_3 = \neg u \vee \neg s$	(5)
$G_4 = \neg s$	(7)
$G_5 = \neg p \vee \neg q$	(2)
$G_6 = \neg q$	(8)
$G_7 = \square$	(6)

(b)

$G_0 = \neg q(f(Z), a)$	
$G_1 = \neg q(a, f(Z)) \vee \neg q(a, f(f(a)))$	(1 cu $\theta(X) = f(Z)$ şi $\theta(Y) = a$)
$G_2 = \neg q(a, f(Z))$	(2 cu $\theta(X) = a$)
$G_3 = \square$	(2 cu $\theta(Z) = f(X)$)

(c)

$G_0 = \neg p(X) \vee \neg q(Y, Z)$	
$G_1 = \neg r(X_1) \vee \neg q(Y, Z)$	(2 cu $\theta(X) = X_1$)
$G_2 = \neg q(Y, Z)$	(5 cu $\theta(X_1) = f(b)$)
$G_3 = \neg p(Z_1)$	(3 cu $\theta(X) = Y_1$ şi $\theta(Y) = Z_1$)
$G_4 = \neg r(X)$	(2 cu $\theta(Z_1) = X$)
$G_5 = \square$	(5 cu $\theta(X) = f(b)$)

$\square$

Teorie pentru S6-7.8:

Fie T o mulţime de clauze definite şi o ţintă $G_0 = \neg P_1 \vee \ldots \vee \neg P_m$. Un *arbore SLD* este definit astfel:

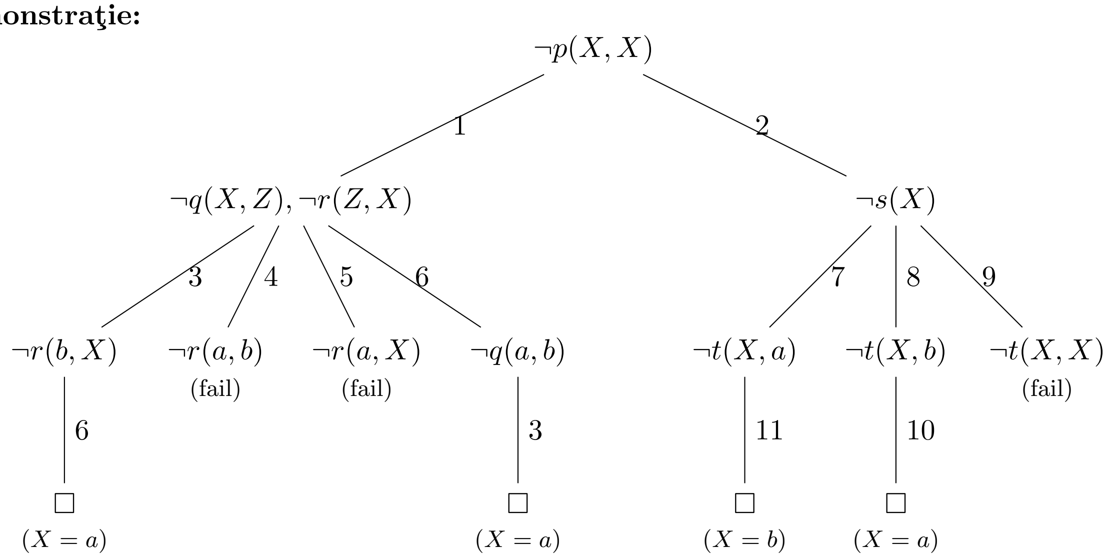
- Fiecare nod al arborelui este o ţintă (posibil vidă)
- Rădăcina este G_0
- Dacă arborele are un nod G_i, iar G_{i+1} se obţine din G_i folosind regula SLD folosind o clauză $C_i \in T$, atunci nodul G_i are copilul G_{i+1}. Muchia dintre G_i şi G_{i+1} este etichetată cu C_i.

Dacă un arbore SLD cu rădăcina G_0 are o frunză $\square$ (clauza vidă), atunci există o SLD-respingere a lui G_0 din T.

(S6-7.8) Desenaţi arborele SLD pentru programul Prolog de mai jos şi ţinta `?- p(X,X).`

```
1.  p(X,Y) :- q(X,Z), r(Z,Y).
2.  p(X,X) :- s(X).
3.  q(X,b).
4.  q(b,a).
5.  q(X,a) :- r(a,X).
6.  r(b,a).
7.  s(X) :- t(X,a).
8.  s(X) :- t(X,b).
9.  s(X) :- t(X,X).
10.  t(a,b).
11.  t(b,a).
```

Demonstraţie:



$\square$