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.

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

- 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 $\alpha \mathrel{\text{⫤}\!\vDash} \alpha^{fc}$.
- 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:

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

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.

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

  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

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

- 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 $\forall x_1 \dots \forall x_n \psi$
  - (v) se determină o FNC ψ' astfel încât $\psi \mathrel{\text{⊨⊣}} \psi'$
  - (vi) φ^{fc} este $\forall x_1 \dots \forall x_n \psi'$

- 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.

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.

(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.

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
  $\vDash \varphi_1 \wedge \ldots \wedge \varphi_n \to \varphi$ 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:

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

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

(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.

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

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

  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).
```

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).
```