Curs 8

Cuprins

Forma normală conjunctivă și forma clauzală

Literali. FNC

- În calculul propozițional un literal este o variabilă sau negația unei variabile.

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

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

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

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

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

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

  O formulă este în formă normală conjunctivă (FNC) dacă este o conjuncție de disjuncții de literali.

FNC în calculul propozițional

- Pentru orice formulă α există o FNC α^{fc} astfel încât $\alpha \mathrel{\text{⫤⊨}} \alpha^{fc}$.
- Pentru o formulă din calculul propozițional determinăm FNC corespunzătoare prin următoarele transformări:

1. înlocuirea implicațiilor și echivalențelor

$$\begin{aligned} \varphi \to \psi &\quad \mathrel{\text{⫤⊨}} \quad \neg\varphi \vee \psi \\ \varphi \leftrightarrow \psi &\quad \mathrel{\text{⫤⊨}} \quad (\neg\varphi \vee \psi) \wedge (\neg\psi \vee \varphi) \end{aligned}$$

2. regulile De Morgan

$$\begin{aligned} \neg(\varphi \vee \psi) &\quad \mathrel{\text{⫤⊨}} \quad \neg\varphi \wedge \neg\psi \\ \neg(\varphi \wedge \psi) &\quad \mathrel{\text{⫤⊨}} \quad \neg\varphi \vee \neg\psi \end{aligned}$$

3. principiului dublei negații

$$\neg\neg\psi \quad \mathrel{\text{⫤⊨}} \quad \psi$$

4. distributivitatea

$$\begin{aligned} \varphi \vee (\psi \wedge \chi) &\quad \mathrel{\text{⫤⊨}} \quad (\varphi \vee \psi) \wedge (\varphi \vee \chi) \\ (\psi \wedge \chi) \vee \varphi &\quad \mathrel{\text{⫤⊨}} \quad (\psi \vee \varphi) \wedge (\chi \vee \varphi) \end{aligned}$$

FNC în calculul propozițional

Exemplu

Determinați FNC pentru formula

$(\neg p \to \neg q) \to (p \to q)$

$\boxminus \neg(\neg p \to \neg q) \vee (p \to q)$

$\boxminus \neg(p \vee \neg q) \vee (\neg p \vee q)$

$\boxminus (\neg p \wedge q) \vee (\neg p \vee q)$

$\boxminus (\neg p \vee \neg p \vee q) \wedge (q \vee \neg p \vee q)$

$\boxminus (\neg p \vee q) \wedge (q \vee \neg p)$

FNCP și forma clauzală în logica de ordinul I

- O formulă este formă normală conjunctivă prenex (FNCP) dacă
  - este în formă prenex $Q_1 x_1 \ldots Q_n x_n \psi$ ($Q_i \in \{\forall, \exists\}$ oricare i)
  - ψ este FNC
- O formulă este formă clauzală dacă este enunț universal și FNCP:

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

Exemplu

- $\forall y \exists z \, ((P(f(y)) \vee Q(z)) \wedge (\neg Q(z) \vee \neg P(g(z)) \vee Q(y)))$

  este FNCP
- $\forall y \forall z \, ((P(f(y)) \vee Q(z)) \wedge (\neg Q(z) \vee \neg P(g(z)) \vee Q(y)))$

  este formă clauzală.

Forma clauzală în logica de ordinul I

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

  φ este satisfiabilă dacă și numai dacă φ^{fc} este satisfiabilă

- Pentru o formulă φ, forma clauzală φ^{fc} se poate calcula astfel:
  1. se determină forma rectificată
  2. se cuantifică universal variabilele libere
  3. se determină forma prenex
  4. se determină forma Skolem

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

  5. se determină o FNC ψ' astfel încât $\psi \vDash\!\dashv \psi'$
  6. φ^{fc} este $\forall x_1 \ldots \forall x_n \psi'$

Literali, clauze, mulțimi de clauze

Clauze

- O clauză este o disjuncție de literali.
- Dacă $L_1, \ldots, L_n$ sunt literali atunci clauza $L_1 \vee \ldots \vee L_n$ o vom scrie ca mulțimea $\{L_1, \ldots, L_n\}$

  clauză = mulțime de literali

- Clauza $C = \{L_1, \ldots, L_n\}$ este satisfiabilă dacă $L_1 \vee \ldots \vee L_n$ este satisfiabilă.
- O clauză C este trivială dacă conține un literal și complementul lui.
- Când $n = 0$ obținem clauza vidă, care se notează $\square$
- Prin definiție, clauza $\square$ nu este satisfiabilă.

Forma clauzală

- Observăm că o FNC este o conjuncție de clauze.
- Dacă $C_1, \ldots, C_k$ sunt clauze atunci $C_1 \wedge \ldots \wedge C_k$ o vom scrie ca mulțimea $\{C_1, \ldots, C_k\}$

FNC = mulțime de clauze

- O mulțime de clauze $\mathcal{C} = \{C_1, \ldots, C_k\}$ este satisfiabilă dacă $C_1 \wedge \ldots \wedge C_k$ este satisfiabilă
- Când $k = 0$ obținem mulțimea de clauze vidă, pe care o notăm $\{\}$
- Prin definiție, mulțimea de clauze vidă $\{\}$ este satisfiabilă.

$\{\}$ este satisfiabilă, dar $\{\square\}$ nu este satisfiabilă

Forma clauzală

- Dacă φ este o formulă în calculul propozițional, atunci
$$\varphi^{fc} = \bigwedge_{i=1}^{k} \bigvee_{j=1}^{n_i} L_{ij} \text{ unde } L_{ij} \text{ sunt literali}$$

- Dacă φ o formulă în logica de ordinul I, atunci
$$\varphi^{fc} = \forall x_1 \ldots \forall x_n \left(\bigwedge_{i=1}^{k} \bigvee_{j=1}^{n_i} L_{ij} \right) \text{ unde } L_{ij} \text{ sunt literali}$$

φ este satisfiabilă dacă și numai dacă

φ^{fc} este satisfiabilă dacă și numai dacă

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

Forma clauzală

Exemplu

- În calculul propozițional:

  pentru a verifica satisfiabilitatea lui $\varphi := (\neg p \to \neg q) \to (p \to q)$

  determinăm $\varphi^{fc} := (\neg p \vee q) \wedge (q \vee \neg p)$

  și analizăm mulțimea de clauze $\{\{\neg p, q\}, \{q, \neg p\}\}$.

- În logica de ordinul I:

  pentru a verifica satisfiabilitatea formulei
  $\varphi := \forall y \forall z ((P(f(y)) \vee Q(z)) \wedge (Q(z) \to (\neg P(g(z)) \vee Q(y))))$

  determinăm
  $\varphi^{fc} := \forall y \forall z\, ((P(f(y)) \vee Q(z)) \;\wedge\; (\neg Q(z) \vee \neg P(g(z)) \vee Q(y)))$

  și analizăm mulțimea de clauze
  $\{\{P(f(y)), Q(z)\}, \{\neg Q(z), \neg P(g(z)), Q(y)\}\}$

Deducție și satisfiabilitate

Fie $\varphi_1, \ldots, \varphi_n, \varphi$ formule în logica propozițională
(enunțuri în calculul cu predicate).

$\{\varphi_1, \ldots, \varphi_n\} \models \varphi$ este echivalent cu

$\models \varphi_1 \wedge \ldots \wedge \varphi_n \rightarrow \varphi$ este echivalent cu

$\models \neg\varphi_1 \vee \ldots \neg\varphi_n \vee \varphi$ este echivalent cu

$\varphi_1 \wedge \ldots \wedge \varphi_n \wedge \neg\varphi$ este satisfiabilă

Pentru a cerceta satisfiabilitatea este suficient să studiem forme clauzale

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

atât în logica propozițională, cât și în calculul cu predicate.

Rezoluție

Rezoluția este o metodă de verificare a satisfiabilității formelor clauzale.

- Rezoluția în calculul propozițional (recap.)
- Rezoluția în logica de ordinul I
  - cazul clauzelor fără variabile
  - cazul general

Rezoluția în calculul propozițional (recap.)

Regula rezoluției

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

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

Propoziție

Regula *Rez* păstrează satisfiabilitatea. Sunt echivalente:

- mulțimea de clauze $\{C_1 \cup \{p\}, C_2 \cup \{\neg p\}\}$ este satisfiabilă,
- clauza $C_1 \cup C_2$ este satisfiabilă.

Exemplu

$$\frac{\{p, \neg q\}, \{\neg p, q\}}{\{q, \neg q\}}$$

Este mulțimea de clauze $\{\{p, \neg q\}, \{\neg p, q\}\}$ satisfiabilă?

Derivare prin rezoluție

Fie $\mathcal{C}$ o mulțime de clauze. O derivare prin rezoluție din $\mathcal{C}$ este o secvență finită de clauze astfel încât fiecare clauză este din $\mathcal{C}$ sau rezultă din clauzele anterioare prin rezoluție (este rezolvent).

Exemplu

Fie $\mathcal{C} = \{\{\neg q, \neg p\}, \{q\}, \{p\}\}$ o mulțime de clauze. O derivare prin rezoluție pentru $\square$ din $\mathcal{C}$ este

$$
\begin{array}{lll}
C_1 = & \{\neg q, \neg p\} & \\
C_2 = & \{q\} & \\
C_3 = & \{\neg p\} & (Rez, C_1, C_2) \\
C_4 = & \{p\} & \\
C_5 = & \square & (Rez, C_3, C_4)
\end{array}
$$

Teorema de completitudine

$\models \varphi$ dacă și numai dacă există o derivare prin rezoluție a lui $\square$ din $(\neg\varphi)^{fc}$.

Procedura Davis-Putnam DPP (informal)

Intrare: o mulțime $\mathcal{C}$ de clauze

Se repetă următorii pași:

- se elimină clauzele triviale
- se alege o variabilă p
- se adaugă la mulțimea de clauze toți rezolvenții obținuti prin aplicarea *Rez* pe variabila p
- se șterg toate clauzele care conțin p sau $\neg p$

Ieșire: dacă la un pas s-a obținut $\square$, mulțimea $\mathcal{C}$ nu este satisfiabilă; altfel $\mathcal{C}$ este satisfiabilă.

Procedura Davis-Putnam DPP

Exemplu

Este $\mathcal{C}_0 = \{\{p, \neg r\}, \{q, p\}, \{q, \neg p, r\}, \{q, \neg r\}\}$ *satisfiabilă?*

Alegem variabila r și selectăm $\mathcal{C}_0^r := \{\{q, \neg p, r\}\}$,
$\mathcal{C}_0^{\neg r} := \{\{p, \neg r\}, \{q, \neg r\}\}$.
Mulțimea rezolvenților posibili este $\mathcal{R}_0 := \{\{q, \neg p, p\}, \{q, \neg p\}\}$;
Se observă că $p, \neg p \in \{q, \neg p, p\}$ deci $\mathcal{R}_0 := \{\{q, \neg p\}\}$

Se elimină clauzele în care apare r și se adaugă noii rezolvenți
$\mathcal{C}_1 := \{\{q, p\}, \{q, \neg p\}\}$

Alegem variabila q și selectăm $\mathcal{C}_1^q := \{\{q, p\}, \{q, \neg p\}\}$, $\mathcal{C}_1^{\neg q} := \emptyset$.
Mulțimea rezolvenților posibili este vidă $\mathcal{R}_1 := \emptyset$.

Se elimină clauzele în care apare q și se adaugă noii rezolvenți
$\mathcal{C}_2 := \{\}$ mulțimea de clauze vidă

Deoarece $\{\}$ este satisfiabilă, rezultă că $\mathcal{C}_0$ este satisfiabilă.

Atenție! La fiecare pas se alege pentru prelucrare o singură variabilă.

Rezoluția în logica de ordinul I

Clauze închise

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

  Spunem că C' este o instanță închisă a lui C dacă există o substituție $\theta : V \to T_{\mathcal{L}}$ such that $C' = \theta(C)$ (C' se obține din C înlocuind variabilele cu termeni din universul Herbrand)

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

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

  $\mathcal{H}(\mathcal{C})$ este mulțimea instanțelor închise ale clauzelor din $\mathcal{C}$.

Teoremă

O mulțime de clauze $\mathcal{C}$ este satisfiabilă dacă și numai dacă $\mathcal{H}(\mathcal{C})$ este satisfiabilă. O mulțime de clauze $\mathcal{C}$ este nesatisfiabilă dacă și numai dacă există o submulțime finită a lui $\mathcal{H}(\mathcal{C})$ care este nesatisfiabilă.

Clauze închise

Exemplu

Cercetați satisfiabilitatea mulțimii de clauze

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

Dacă c este constantă atunci
$\{\{\neg P(c), Q(c)\}, \{P(c)\}, \{\neg Q(c)\}\} \subseteq \mathcal{H}(\mathcal{C})$.
Considerăm toate valorile de adevăr pentru $P(c)$ și $Q(c)$:

$P(c)$	$Q(c)$	$(\neg P(c) \vee Q(c)) \wedge P(c) \wedge \neg Q(c)$
0	0	0
0	1	0
1	0	0
1	1	0

$\{\{\neg P(c), Q(c)\}, \{P(c)\}, \{\neg Q(c)\}\}$ este nesatisfiabilă, deci
$\mathcal{C}$ este nesatisfiabilă.

Putem gândi formulele atomice închise ca variabile propoziționale.

Rezoluția pe clauze închise

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

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

Propoziție

Regula *Rez* păstrează satisfiabilitatea. Sunt echivalente:

- mulțimea de clauze $\{C_1 \cup \{L\}, C_2 \cup \{\neg L\}\}$ este satisfiabilă,
- clauza $C_1 \cup C_2$ este satisfiabilă.

Teoremă

Fie φ o formulă arbitrară în logica de ordinul I. Atunci $\models \varphi$ dacă și numai dacă există o derivare pentru $\square$ din $\mathcal{H}(\mathcal{C})$ folosind *Rez*, unde $\mathcal{C}$ este mulțimea de clauze asociată lui $(\neg\varphi)^{fc}$.

Rezoluția pe clauze închise

Exemplu

Fie f, g simboluri de funcții unare, P, Q simboluri de predicate unare. Cercetați satisfiabilitatea formulei:

$$\varphi = \forall x((\neg P(x) \vee Q(f(x))) \wedge P(g(x)) \wedge \neg Q(x))$$

Determinăm forma clauzală:

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

Pentru c o constantă obținem următoarea derivare:

$$\begin{array}{lll}
C_1 = & \{\neg P(g(c)), Q(f(g(c)))\} & \\
C_2 = & \{P(g(c))\} & \\
C_3 = & \{Q(f(g(c)))\} & Rez, C_1, C_2 \\
C_4 = & \{\neg Q(f(g(c)))\} & \\
C_5 = & \square & Rez, C_3, C_4
\end{array}$$

Rezoluția pe clauze arbitrare

Observații:

- Unificarea literalilor revine la unificarea argumentelor

  Dacă $\sigma : V \to Trm_{\mathcal{L}}$ o substituție, atunci sunt echivalente
  - $\sigma(P(t_1, \ldots, t_n)) = \sigma(P(t'_1, \ldots, t'_n))$
  - $\sigma(\neg P(t_1, \ldots, t_n)) = \sigma(\neg P(t'_1, \ldots, t'_n))$
  - $\sigma(t_1) = \sigma(t'_1), \ldots, \sigma(t_n) = \sigma(t'_n)$

- Redenumirea variabilelor în clauze păstrează validitatea

Exemplu

Deoarece $\forall x(\varphi \land \psi) \mathrel{\text{⫤}\!\vDash} (\forall x\varphi) \land (\forall x\psi)$ obținem

$\forall x((P_1(x) \lor P_2(x)) \land (Q_1(x) \lor Q_2(x)))$

$\mathrel{\text{⫤}\!\vDash} (\forall x(P_1(x) \lor P_2(x))) \land (\forall x(Q_1(x) \lor Q_2(x)))$
$\mathrel{\text{⫤}\!\vDash} (\forall x(P_1(x) \lor P_2(x))) \land (\forall y(Q_1(y) \lor Q_2(y)))$

$\mathrel{\text{⫤}\!\vDash} \forall x \forall y(P_1(x) \lor P_2(x)) \land (Q_1(y) \lor Q_2(y))$

Rezoluția pe clauze arbitrare

Regula rezoluției pentru clauze arbitrare

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

dacă următoarele condiții sunt satisfăcute:

1. C_1, C_2 clauze care nu au variabile comune,
2. $Lit_1 \subseteq C_1$ și $Lit_2 \subseteq C_2$ sunt mulțimi de literali,
3. σ este un cgu pentru Lit_1 și Lit_2^c, adică
   σ unifică toți literalii din Lit_1 și Lit_2^c.

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

Rezoluția în logica de ordinul I

Exemplu

Găsiți un rezolvent pentru clauzele:

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

- redenumim variabilele pentru a satisface condițiile din *Rez*
  $\theta C_2 = \{\neg P(f(f(a)), g(z)),\ Q(f(a), g(z))\}$ unde $\theta = \{y \leftarrow z\}$
- determinăm Lit_1 și Lit_2
  $$Lit_1 = \{ P(f(x), g(y))\} \text{ și } Lit_2 = \{\neg P(f(f(a)), g(z))\}$$
- găsim un cgu σ care este unificator pentru
  $$Lit_1 = \{ P(f(x), g(y))\} \text{ și } Lit_2^c = \{P(f(f(a)), g(z))\}$$
  $$\sigma = \{x \leftarrow f(a), y \leftarrow z\}$$
- Rezolventul este $C = (\sigma C_1 \setminus \sigma Lit_1) \cup (\sigma(\theta C_2) \setminus \sigma Lit_2)$
  $$C = \{Q(f(a), z), Q(f(a), g(z))\}$$

Rezoluția în logica de ordinul I

- Fie $\mathcal{C}$ o mulțime de clauze. O derivare prin rezoluție din mulțimea $\mathcal{C}$ pentru o clauză C este o secvență $C_1, \ldots, C_n$ astfel încât $C_n = C$ și, 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 $j, k < i$.

Teoremă

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

Rezoluția este corectă și completă în calculul cu predicate,
dar nu este procedură de decizie.

Exemplu

Găsiți o derivare a $\square$ din $\mathcal{C} = \{C_1, C_2, C_3, C_4\}$ unde:

$$C_1 = \{\neg P(x,y),\ P(y,x)\}$$
$$C_2 = \{\neg P(x,y),\ \neg P(y,z),\ P(x,z)\}$$
$$C_3 = \{P(x, f(x))\}$$
$$C_4 = \{\neg P(x,x)\}$$

$C_3' = \{P(x_1, f(x_1))\}$	redenumire în C_3
$C_5 = \{P(f(x), x)\}$	Rez, $\sigma = \{x_1 \leftarrow x, y \leftarrow f(x)\}$, C_1, C_3'
$C_3'' = \{P(x_2, f(x_2))\}$	redenumire în C_3
$C_6 = \{\neg P(f(x), z),\ P(x,z)\}$	Rez, $\sigma = \{x_2 \leftarrow x, y \leftarrow f(x)\}$, C_2, C_3''
$C_5' = \{P(f(x_3), x_3)\}$	redenumire în C_5
$C_7 = \{P(x,x)\}$	Rez, $\sigma = \{x_3 \leftarrow x, z \leftarrow x\}$, C_6, C_5'
$C_4' = \{\neg P(x_4, x_4)\}$	redenumire în C_4
$C_5 = \square$	Rez, $\sigma = \{x_4 \leftarrow x\}$, C_7, C_4'

Exemplu: "Did curiosity kill the cat?"

Exemplu

Cercetați dacă din ipotezele (i1)-(i6) se deduce (c), unde

(i1) Jack owns a dog.

(i2) Anyone who owns a dog is a lover of animals.

(i3) Lovers of animals do not kill animals.

(i4) Either Jack killed Tuna or curiosity killed Tuna.

(i5) Tuna is a cat.

(i6) All cats are animals.

(c) Curiosity killed Tuna.

Vom formaliza ipotezele și concluzia în logica de ordinul I si vom face demonstrația folosind rezoluția.

Exemplu: "Did curiosity kill the cat?"

Exemplu

Formalizăm ipotezele și determinăm forma clauzală:

(i1) Jack owns a dog.

$\exists x\,(D(x) \wedge O(jack, x))$ $\mathbf{R} = \{D, O\}, \mathbf{C} = \{jack\}$

$D(dog) \wedge O(jack, dog)$ Skolemizare

$\mathbf{R} = \{D, O\}, \mathbf{C} = \{jack, dog\}$

Exemplu: "Did curiosity kill the cat?"

Exemplu

(i2) Anyone who owns a dog is a lover_of_animals.

$\forall x\,(\exists y\,(D(y) \wedge O(x,y))) \rightarrow L(x))$ $\quad$ $\mathbf{R} = \{D, O, L\}, \mathbf{C} = \{jack, dog\}$

$\forall x\,((\neg(\exists y\,(D(y) \wedge O(x,y))) \vee L(x))$

$\forall x\,((\forall y\,(\neg D(y) \vee \neg O(x,y))) \vee L(x))$

$\forall x\,\forall y\,(\neg D(y) \vee \neg O(x,y)) \vee L(x))$

$\neg D(y) \vee \neg O(x,y)) \vee L(x)$

Exemplu: "Did curiosity kill the cat?"

Exemplu

(i3) Lovers_of_animals do not kill animals.

$\forall x\,(L(x) \rightarrow (\forall y\, A(y) \rightarrow \neg K(x,y)))$
$\mathbf{R} = \{D, O, L, A, K\}, \mathbf{C} = \{jack, dog\}$

$\forall x\,(\neg L(x) \vee (\forall y\, \neg A(y) \vee \neg K(x,y)))$

$\forall x\, \forall y(\neg L(x) \vee \neg A(y) \vee \neg K(x,y))$

$\neg L(x) \vee \neg A(y) \vee \neg K(x,y)$

Exemplu: "Did curiosity kill the cat?"

Exemplu

(i4) Either Jack killed Tuna or curiosity killed Tuna.

$K(jack, tuna) \wedge K(curiosity, tuna)$

$$\mathbf{R} = \{D, O, L, A, K\}, \mathbf{C} = \{jack, dog, tuna, curiosity\}$$

(i5) Tuna is a cat.

$C(tuna)$

$$\mathbf{R} = \{D, O, L, A, K, C\}, \mathbf{C} = \{jack, dog, tuna, curiosity\}$$

(i6) All cats are animals.

$\forall x\, C(x) \rightarrow A(x)$
$\neg C(x) \vee A(x)$

Exemplu: "Did curiosity kill the cat?"

Exemplu

(i1) $D(dog) \wedge O(jack, dog)$

(i2) $\neg D(y) \vee \neg O(x, y) \vee L(x)$

(i3) $\neg L(x) \vee \neg A(y) \vee \neg K(x, y)$

(i4) $K(jack, tuna) \vee K(curiosity, tuna)$

(i5) $C(tuna)$

(i6) $\neg C(x) \vee A(x)$

(c) $K(curiosity, tuna)$

Exemplu: "Did curiosity kill the cat?"

Exemplu

(i1) $D(dog)$

(i1) $O(jack, dog)$

(i2) $\neg D(y) \vee \neg O(x, y) \vee L(x)$

(i3) $\neg L(x) \vee \neg A(y) \vee \neg K(x, y)$

(i4) $K(jack, tuna) \vee K(curiosity, tuna)$

(i5) $C(tuna)$

(i6) $\neg C(x) \vee A(x)$

($\neg$c) $\neg K(curiosity, tuna)$

Exemplu: "Did curiosity kill the cat?"

Exemplu

$D(dog)$

$O(jack, dog)$

$\neg D(y) \vee \neg O(x, y) \vee L(x)$

$\neg L(x) \vee \neg A(y) \vee \neg K(x, y)$

$K(jack, tuna) \vee K(curiosity, tuna)$

$C(tuna)$

$\neg C(x) \vee A(x)$

$\neg K(curiosity, tuna)$

Exemplu: "Did curiosity kill the cat?"

Exemplu

$D(dog)$

$O(jack, dog)$

$\neg D(y) \vee \neg O(x, y) \vee L(x)$

$\neg L(x) \vee \neg A(y) \vee \neg K(x, y)$

$K(jack, tuna)$

$C(tuna)$

$\neg C(x) \vee A(x)$

Exemplu: "Did curiosity kill the cat?"

Exemplu

$D(dog)$

$O(jack, dog)$

$\neg D(y) \lor \neg O(x, y) \lor L(x)$

$\neg L(x) \lor \neg A(y) \lor \neg K(x, y)$

$K(jack, tuna)$

$C(tuna)$

$\neg C(x) \lor A(x)$ $\{x \leftarrow tuna\}$

Exemplu: "Did curiosity kill the cat?"

Exemplu

$D(dog)$

$O(jack, dog)$

$\neg D(y) \vee \neg O(x, y) \vee L(x)$

$\neg L(x) \vee \neg A(y) \vee \neg K(x, y)$

$K(jack, tuna)$

$A(tuna)$

Exemplu: "Did curiosity kill the cat?"

Exemplu

$D(dog)$

$O(jack, dog)$

$\neg D(y) \vee \neg O(x, y) \vee L(x)$

$\neg L(x) \vee \neg A(y) \vee \neg K(x, y)$ $\{y \leftarrow tuna\}$

$K(jack, tuna)$

$A(tuna)$

Exemplu: "Did curiosity kill the cat?"

Exemplu

$D(dog)$

$O(jack, dog)$

$\neg D(y) \vee \neg O(x, y) \vee L(x)$

$\neg L(x) \vee \neg K(x, tuna)$

$K(jack, tuna)$

Exemplu: "Did curiosity kill the cat?"

Exemplu

$D(dog)$

$O(jack, dog)$

$\neg D(y) \vee \neg O(x, y) \vee L(x)$

$\neg L(x) \vee \neg K(x, tuna)\ \{x \leftarrow jack\}$

$K(jack, tuna)$

Exemplu: "Did curiosity kill the cat?"

Exemplu

$D(dog)$

$O(jack, dog)$

$\neg D(y) \vee \neg O(x, y) \vee L(x)$

$\neg L(jack)$

Exemplu: "Did curiosity kill the cat?"

Exemplu

$D(dog)$

$O(jack, dog)$

$\neg D(y) \vee \neg O(x, y) \vee L(x)\ \{x \leftarrow jack\}$

$\neg L(jack)$

Exemplu: "Did curiosity kill the cat?"

Exemplu

$D(dog)$

$O(jack, dog)$

$\neg D(y) \vee \neg O(jack, y)$

Exemplu: "Did curiosity kill the cat?"

Exemplu

$D(dog)$

$O(jack, dog)$

$\neg D(y) \vee \neg O(jack, y)$ $\{y \leftarrow dog\}$

"Curiosity killed the cat!"

Exemplu

$O(jack, dog)$

$\neg O(jack, dog)$

$\square$

În concluzie, am arătat că

$$\{(\text{i1}), (\text{i2}), (\text{i3}), (\text{i4}), (\text{i5}), (\text{i6})\} \vDash (\text{c})$$

Pe săptămâna viitoare!