Curs 7

2021-2022

Fundamentele Limbajelor de Programare

Cuprins

Rezoluție SLD - cazul logicii propoziționale

Metodă de decizie

Avem o metodă de decizie *(decision procedure)* pentru a verifica $\mathcal{S} \vdash q$

Metoda constă în:

- calcularea celui mai mic punct fix X al funcției $f_{\mathcal{S}}$
- dacă $q \in X$ atunci returnăm **true**, altfel returnăm **false**

Metodă de decizie

Avem o metodă de decizie *(decision procedure)* pentru a verifica $\mathcal{S} \vdash q$

Metoda constă în:

- calcularea celui mai mic punct fix X al funcției $f_{\mathcal{S}}$
- dacă $q \in X$ atunci returnăm **true**, altfel returnăm **false**

Această metodă se termină.
Exercițiu. De ce?

Metodă de decizie

Avem o metodă de decizie *(decision procedure)* pentru a verifica $\mathcal{S} \vdash q$

Metoda constă în:

- calcularea celui mai mic punct fix X al funcției $f_{\mathcal{S}}$
- dacă $q \in X$ atunci returnăm **true**, altfel returnăm **false**

Această metodă se termină.
Exercițiu. De ce?

Program Prolog = baza de cunoștințe

- Un program Prolog reprezintă o bază de cunoștințe (knowledge base) KB. Cel mai mic punct fix al funcției f_{KB} definește totalitatea cunoștintelor care pot fi deduse din KB.

Metodă de decizie

Avem o metodă de decizie *(decision procedure)* pentru a verifica $\mathcal{S} \vdash q$

Metoda constă în:

- calcularea celui mai mic punct fix X al funcției $f_{\mathcal{S}}$
- dacă $q \in X$ atunci returnăm **true**, altfel returnăm **false**

Această metodă se termină.
Exercițiu. De ce?

Program Prolog = baza de cunoștințe

- Un program Prolog reprezintă o bază de cunoștințe (knowledge base) KB. Cel mai mic punct fix al funcției f_{KB} definește totalitatea cunoștintelor care pot fi deduse din KB.
- Pentru o bază de cunoștințe formată numai din clauze propoziționale definite, cel mai mic punct fix poate fi calculat în timp liniar.

Clauze definite

- Singurele formule admise sunt de forma:
  - q
  - $p_1 \wedge \ldots \wedge p_n \rightarrow q$, unde toate p_i, q sunt variabile propozitionale.
- O clauză definită $p_1 \wedge \ldots \wedge p_n \rightarrow q$ poate fi gândită ca formula

$$\neg p_1 \vee \ldots \vee \neg p_n \vee q$$

Clauze definite

- Singurele formule admise sunt de forma:
  - q
  - $p_1 \wedge \ldots \wedge p_n \rightarrow q$, unde toate p_i, q sunt variabile propozitionale.
- O clauză definită $p_1 \wedge \ldots \wedge p_n \rightarrow q$ poate fi gândită ca formula

$$\neg p_1 \vee \ldots \vee \neg p_n \vee q$$

Echivalent, putem reprezenta clauza definită de mai sus și prin $\{\neg p_1, \ldots, \neg p_n, q\}$

Calculul celui mai mic punct fix

KB:

$\{oslo\}$

$\{\neg oslo, windy\}$

$\{\neg oslo, norway\}$

$\{\neg norway, cold\}$

$\{\neg cold, \neg windy, winter\}$

LFP:

Calculul celui mai mic punct fix

$\{\neg oslo, windy\}$

$\{\neg oslo, norway\}$

$\{\neg norway, cold\}$

$\{\neg cold, \neg windy, winter\}$

LFP:

$\{oslo\}$

Calculul celui mai mic punct fix

$\{\neg oslo, windy\}$

$\{\neg oslo, norway\}$

$\{\neg norway, cold\}$

$\{\neg cold, \neg windy, winter\}$

LFP:

$\{oslo\}$

Calculul celui mai mic punct fix

$\{\neg norway, cold\}$

$\{\neg cold, \neg windy, winter\}$

LFP:

$\{oslo\}$

$\{windy\}$

$\{norway\}$

Calculul celui mai mic punct fix

$\{\neg norway, cold\}$

$\{\neg cold, \neg windy, winter\}$

LFP:

$\{oslo\}$

$\{windy\}$

$\{norway\}$

Calculul celui mai mic punct fix

$\{\neg cold, \neg windy, winter\}$

LFP:

$\{oslo\}$

$\{windy\}$

$\{norway\}$

$\{cold\}$

Calculul celui mai mic punct fix

$\{\neg cold, \neg windy, winter\}$

LFP:

$\{oslo\}$

$\{windy\}$

$\{norway\}$

$\{cold\}$

Calculul celui mai mic punct fix

$\{\neg windy, winter\}$

LFP:

$\{oslo\}$

$\{windy\}$

$\{norway\}$

$\{cold\}$

Calculul celui mai mic punct fix

$\{\neg windy, winter\}$

LFP:

$\{oslo\}$

$\{windy\}$

$\{norway\}$

$\{cold\}$

Calculul celui mai mic punct fix

LFP:

$\{oslo\}$

$\{windy\}$

$\{norway\}$

$\{cold\}$

$\{\mathbf{winter}\}$

Propagarea unității

- În procedeul anterior am folosit o metodă asemănătoare rezoluției în care una din clauze are un singur literal.
- Clauzele formate dintr-un singur literal se numesc clauze unitate (*unit clause*), iar metoda anterioară se numește propagarea unității (*unit propagation*).
- Printr-o reprezentare adecvată a datelor, propagarea unității poate fi implementată în timp liniar în raport cu dimensiunea bazei de cunoștințe inițiale.
- Clauzele Horn propoziționale sunt clauze care au cel mult un literal pozitiv. Clauzele propoziționale definite sunt clauze Horn care au exact un literal pozitiv. Folosind metoda de propagare a unității problema satsfiabilității pentru clauze Horn propoziționale HORNSAT poate fi rezolvată în timp liniar.

Forward chaining / Backward chaining

- Metoda anterioară este centrată pe *lărgirea bazei de cunoștințe*.
- Pentru a afla răspunsul la o întrebare (`-? winter`) adăugăm pas cu pas cunoștințe noi, verificând de fiecare dată dacă am răspuns la întrebare.
- Acest procedeu se numește forward chaining.

Forward chaining / Backward chaining

- Metoda anterioară este centrată pe *lărgirea bazei de cunoștințe*.
- Pentru a afla răspunsul la o întrebare (`-? winter`) adăugăm pas cu pas cunoștințe noi, verificând de fiecare dată dacă am răspuns la întrebare.
- Acest procedeu se numește forward chaining.

Nu acesta este algoritmul folosit de Prolog!

Forward chaining / Backward chaining

- Metoda anterioară este centrată pe *lărgirea bazei de cunoștințe*.
- Pentru a afla răspunsul la o întrebare (`-? winter`) adăugăm pas cu pas cunoștințe noi, verificând de fiecare dată dacă am răspuns la întrebare.
- Acest procedeu se numește forward chaining.

Nu acesta este algoritmul folosit de Prolog!

- Metoda folosită de Prolog se numește backward chaining. Această metodă este centrată pe *găsirea răspunsului la întrebare*.

Backward chaining

- În *backward chaining* pornim de la întrebare (`-? winter`) și analizăm baza de cunoștinte, căutând o regulă care are drept concluzie scopul (`winter :- cold, windy`).
- În continuare vom încerca să satisfacem scopurile noi (`cold` și `windy`) prin același procedeu.
- Această metodă este realizată printr-o implementare particulară a rezoluției - rezoluția SLD.

Rezoluția SLD (cazul propozițional)

Fie S o mulțime de clauze definite.

$$\text{SLD}\ \boxed{\frac{\neg p_1 \vee \cdots \vee \neg q \vee \cdots \vee \neg p_n}{\neg p_1 \vee \cdots \vee \neg q_1 \vee \cdots \vee \neg q_m \vee \cdots \vee \neg p_n}}$$

unde $q \vee \neg q_1 \vee \cdots \vee \neg q_m$ este o clauză definită din S.

Rezoluția SLD

Fie S o mulțime de clauze definite și q o întrebare.

O derivare din S prin rezoluție SLD este o secvență

$$G_0 := \neg q, \quad G_1, \quad \ldots, \quad G_k, \ldots$$

în care G_{i+1} se obține din G_i prin regula SLD.

Dacă există un k cu $G_k = \square$ (clauza vidă), atunci derivarea se numește SLD-respingere.

Rezoluția SLD

Teoremă (Completitudinea SLD-rezoluției)

Sunt echivalente:

- *există o SLD-respingere a lui q din S,*
- $S \vdash q$,
- $S \models q$.

Rezoluția SLD

Baza de cunoștințe KB:

```
oslo .
windy :- oslo.
norway :- oslo.
cold :- norway.
winter :- cold, windy.
```

Întrebarea:

```
-? winter.
```

Rezoluția SLD

Baza de cunoștințe KB:

```
oslo .
windy :- oslo.
norway :- oslo.
cold :- norway.
winter :- cold, windy.
```

Întrebarea:

```
-? winter.
```

- Formă clauzală:

  $$\text{KB} = \{\{oslo\}, \{\neg oslo, windy\}, \{\neg oslo, norway\}, \{\neg norway, cold\}, \{\neg cold, \neg windy, winter\}\}$$

- $\text{KB} \vdash winter$ dacă și numai dacă $KB \cup \{\neg winter\}$ este satisfiabilă.

Rezoluția SLD

Baza de cunoștințe KB:

```
oslo .
windy :- oslo.
norway :- oslo.
cold :- norway.
winter :- cold, windy.
```

Întrebarea:

```
-? winter.
```

- Formă clauzală:

  $$\text{KB} = \{\{oslo\}, \{\neg oslo, windy\}, \{\neg oslo, norway\}, \{\neg norway, cold\}, \{\neg cold, \neg windy, winter\}\}$$

- $\text{KB} \vdash winter$ dacă și numai dacă $KB \cup \{\neg winter\}$ este satisfiabilă.

- Satisfiabilitatea este verificată prin rezoluție

  SLD = Linear resolution with Selected literal for Definite clauses

Clause Horn propoziționale - rezoluția SLD

Exemplu

Demonstrăm KB $\vdash$ *winter* prin rezoluție SLD:

$\{\neg winter\}$

Clause Horn propoziționale - rezoluția SLD

Exemplu

Demonstrăm KB $\vdash$ *winter* prin rezoluție SLD:

$\{\neg winter\}$ $\{\neg cold, \neg windy, winter\}$

$\{\neg cold, \neg windy\}$

Clause Horn propoziționale - rezoluția SLD

Exemplu

Demonstrăm KB $\vdash$ *winter* prin rezoluție SLD:

$\{\neg winter\}$ $\{\neg cold, \neg windy, winter\}$

$\{\neg cold, \neg windy\}$ $\{\neg norway, cold\}$

$\{\neg norway, \neg windy\}$

Clause Horn propoziționale - rezoluția SLD

Exemplu

Demonstrăm KB $\vdash$ *winter* prin rezoluție SLD:

$\{\neg winter\}$ $\{\neg cold, \neg windy, winter\}$

$\{\neg cold, \neg windy\}$ $\{\neg norway, cold\}$

$\{\neg norway, \neg windy\}$ $\{\neg oslo, norway\}$

$\{\neg oslo, \neg windy\}$

Clause Horn propoziționale - rezoluția SLD

Exemplu

Demonstrăm KB $\vdash$ *winter* prin rezoluție SLD:

$\{\neg winter\}$ $\quad$ $\{\neg cold, \neg windy, winter\}$

$\{\neg cold, \neg windy\}$ $\quad$ $\{\neg norway, cold\}$

$\{\neg norway, \neg windy\}$ $\quad$ $\{\neg oslo, norway\}$

$\{\neg oslo, \neg windy\}$ $\quad$ $\{oslo\}$

$\{\neg windy\}$

Clause Horn propoziționale - rezoluția SLD

Exemplu

Demonstrăm KB $\vdash$ *winter* prin rezoluție SLD:

$\{\neg winter\}$ $\{\neg cold, \neg windy, winter\}$

$\{\neg cold, \neg windy\}$ $\{\neg norway, cold\}$

$\{\neg norway, \neg windy\}$ $\{\neg oslo, norway\}$

$\{\neg oslo, \neg windy\}$ $\{oslo\}$

$\{\neg windy\}$ $\{\neg oslo, windy\}$

$\{\neg oslo\}$

Clause Horn propoziționale - rezoluția SLD

Exemplu

Demonstrăm KB $\vdash$ *winter* prin rezoluție SLD:

$\{\neg winter\}$	$\{\neg cold, \neg windy, winter\}$
$\{\neg cold, \neg windy\}$	$\{\neg norway, cold\}$
$\{\neg norway, \neg windy\}$	$\{\neg oslo, norway\}$
$\{\neg oslo, \neg windy\}$	$\{oslo\}$
$\{\neg windy\}$	$\{\neg oslo, windy\}$
$\{\neg oslo\}$	$\{oslo\}$

$\square$

Clause Horn propoziționale - rezoluția SLD

Exemplu

Demonstrăm KB $\vdash$ *winter* prin rezoluție SLD:

$\{\neg winter\}$	$\{\neg cold, \neg windy, winter\}$
$\{\neg cold, \neg windy\}$	$\{\neg norway, cold\}$
$\{\neg norway, \neg windy\}$	$\{\neg oslo, norway\}$
$\{\neg oslo, \neg windy\}$	$\{oslo\}$
$\{\neg windy\}$	$\{\neg oslo, windy\}$
$\{\neg oslo\}$	$\{oslo\}$

$\square$

În continuare vom studia aceste mecanisme în logica de ordinul I.

Logica de ordinul I - recapitulare

Limbaje de ordinul I

Un limbaj $\mathcal{L}$ de ordinul I este format din:

- o mulțime numărabilă de variabile $V = \{x_n \mid n \in \mathbb{N}\}$
- conectorii $\neg, \rightarrow, \wedge, \vee$
- paranteze
- cuantificatorul universal $\forall$ și cuantificatorul existențial $\exists$
- o mulțime $\mathbf{R}$ de simboluri de relații
- o mulțime $\mathbf{F}$ de simboluri de funcții
- o mulțime $\mathbf{C}$ de simboluri de constante
- o funcție aritate $ar : \mathbf{F} \cup \mathbf{R} \rightarrow \mathbb{N}^*$

Logica de ordinul I

- $\mathcal{L}$ este unic determinat de $\tau = (\mathbf{R}, \mathbf{F}, \mathbf{C}, \mathit{ari})$
- τ se numește signatura (vocabularul, alfabetul) lui $\mathcal{L}$

Logica de ordinul I

- $\mathcal{L}$ este unic determinat de $\tau = (\mathbf{R}, \mathbf{F}, \mathbf{C}, ari)$
- τ se numește signatura (vocabularul, alfabetul) lui $\mathcal{L}$

Exemplu

Un limbaj $\mathcal{L}$ de ordinul I în care:

- $\mathbf{R} = \{P, R\}$
- $\mathbf{F} = \{f\}$
- $\mathbf{C} = \{c\}$
- $ari(P) = 1$, $ari(R) = 2$, $ari(f) = 2$

Sintaxa Prolog

Atenție!

- În sintaxa Prolog
  - termenii compuși sunt predicate: `father(eddard, jon_snow)`
  - operatorii sunt funcții: `+, *, mod`
- Sintaxa Prolog nu face diferență între simboluri de funcții și simboluri de predicate!
- Dar este important când ne uităm la teoria corespunzătoare programului în logică să facem această distincție.

Logica de ordinul I

Termenii lui $\mathcal{L}$ sunt definiți inductiv astfel:

- orice variabilă este un termen;
- orice simbol de constantă este un termen;
- dacă $f \in \mathbf{F}$, $ar(f) = n$ și $t_1, \ldots, t_n$ sunt termeni, atunci $f(t_1, \ldots, t_n)$ este termen.

Notăm cu $Trm_{\mathcal{L}}$ mulțimea termenilor lui $\mathcal{L}$.

Logica de ordinul I

Termenii lui $\mathcal{L}$ sunt definiți inductiv astfel:

- orice variabilă este un termen;
- orice simbol de constantă este un termen;
- dacă $f \in \mathbf{F}$, $ar(f) = n$ și $t_1, \ldots, t_n$ sunt termeni, atunci $f(t_1, \ldots, t_n)$ este termen.

Notăm cu $Trm_{\mathcal{L}}$ mulțimea termenilor lui $\mathcal{L}$.

Exemplu

$$c, \quad x_1, \quad f(x_1, c), \quad f(f(x_2, x_2), c)$$

Logica de ordinul I

Formulele atomice ale lui $\mathcal{L}$ sunt definite astfel:

- dacă $R \in \mathbf{R}$, $ar(R) = n$ și $t_1, \ldots, t_n$ sunt termeni, atunci $R(t_1, \ldots, t_n)$ este formulă atomică.

Logica de ordinul I

Formulele atomice ale lui $\mathcal{L}$ sunt definite astfel:

- dacă $R \in \mathbf{R}$, $ar(R) = n$ și $t_1, \ldots, t_n$ sunt termeni, atunci $R(t_1, \ldots, t_n)$ este formulă atomică.

Exemplu

$$P(f(x_1, c)), \quad R(c, x_3)$$

Logica de ordinul I

Formulele lui $\mathcal{L}$ sunt definite astfel:

- orice formulă atomică este o formulă
- dacă φ este o formulă, atunci $\neg\varphi$ este o formulă
- dacă φ și ψ sunt formule, atunci $\varphi \vee \psi$, $\varphi \wedge \psi$, $\varphi \to \psi$ sunt formule
- dacă φ este o formulă și x este o variabilă, atunci $\forall x\, \varphi$, $\exists x\, \varphi$ sunt formule

Logica de ordinul I

Formulele lui $\mathcal{L}$ sunt definite astfel:

- orice formulă atomică este o formulă
- dacă φ este o formulă, atunci $\neg\varphi$ este o formulă
- dacă φ și ψ sunt formule, atunci $\varphi \vee \psi$, $\varphi \wedge \psi$, $\varphi \to \psi$ sunt formule
- dacă φ este o formulă și x este o variabilă, atunci $\forall x\, \varphi$, $\exists x\, \varphi$ sunt formule

Exemplu

$$P(f(x_1, c)), \quad P(x_1) \vee P(c), \quad \forall x_1\, P(x_1), \quad \forall x_2\, R(x_2, x_1)$$

Logica de ordinul I

Exemplu

Fie limbajul $\mathcal{L}_1$ cu $\mathbf{R} = \{<\}$, $\mathbf{F} = \{s, +\}$, $\mathbf{C} = \{0\}$ și $ari(s) = 1$, $ari(+) = ari(<) = 2$.

Logica de ordinul I

Exemplu

Fie limbajul $\mathcal{L}_1$ cu $\mathbf{R} = \{<\}$, $\mathbf{F} = \{s, +\}$, $\mathbf{C} = \{0\}$ și $ari(s) = 1$, $ari(+) = ari(<) = 2$.

Exemple de termeni:

Logica de ordinul I

Exemplu

Fie limbajul $\mathcal{L}_1$ cu $\mathbf{R} = \{<\}$, $\mathbf{F} = \{s, +\}$, $\mathbf{C} = \{0\}$ și $ari(s) = 1$, $ari(+) = ari(<) = 2$.

Exemple de termeni:

0, x, $s(0)$, $s(s(0))$, $s(x)$, $s(s(x))$, ...,

Logica de ordinul I

Exemplu

Fie limbajul $\mathcal{L}_1$ cu $\mathbf{R} = \{<\}$, $\mathbf{F} = \{s, +\}$, $\mathbf{C} = \{0\}$ și $ari(s) = 1$, $ari(+) = ari(<) = 2$.

Exemple de termeni:

$0, x, s(0), s(s(0)), s(x), s(s(x)), \ldots,$
$+(0,0), +(s(s(0)), +(0, s(0))), +(x, s(0)), +(x, s(x)), \ldots,$

Logica de ordinul I

Exemplu

Fie limbajul $\mathcal{L}_1$ cu $\mathbf{R} = \{<\}$, $\mathbf{F} = \{s, +\}$, $\mathbf{C} = \{0\}$ și
$ari(s) = 1$, $ari(+) = ari(<) = 2$.

Exemple de termeni:

0, x, $s(0)$, $s(s(0))$, $s(x)$, $s(s(x))$, ...,
$+(0,0)$, $+(s(s(0)),+(0,s(0)))$, $+(x,s(0))$, $+(x,s(x))$, ...,

Exemple de formule atomice:

$<(0,0)$, $<(x,0)$, $<(s(s(x)),s(0))$, ...

Logica de ordinul I

Exemplu

Fie limbajul $\mathcal{L}_1$ cu $\mathbf{R} = \{<\}$, $\mathbf{F} = \{s, +\}$, $\mathbf{C} = \{0\}$ și $ari(s) = 1$, $ari(+) = ari(<) = 2$.

Exemple de termeni:

$0, x, s(0), s(s(0)), s(x), s(s(x)), \ldots,$
$+(0,0), +(s(s(0)), +(0, s(0))), +(x, s(0)), +(x, s(x)), \ldots,$

Exemple de formule atomice:

$< (0,0), < (x,0), < (s(s(x)), s(0)), \ldots$

Exemple de formule:

$\forall x \forall y < (x, +(x, y))$
$\forall x < (x, s(x))$

Semantica

Pentru a stabili dacă o formulă este adevărată, avem nevoie de o interpretare într-o structură!

Structură

Definiție

O structură este de forma $\mathcal{A} = (A, \mathbf{F}^{\mathcal{A}}, \mathbf{R}^{\mathcal{A}}, \mathbf{C}^{\mathcal{A}})$, unde

- A este o mulțime nevidă
- $\mathbf{F}^{\mathcal{A}} = \{f^{\mathcal{A}} \mid f \in \mathbf{F}\}$ este o mulțime de operații pe A; dacă f are aritatea n, atunci $f^{\mathcal{A}} : A^n \to A$.
- $\mathbf{R}^{\mathcal{A}} = \{R^{\mathcal{A}} \mid R \in \mathbf{R}\}$ este o mulțime de relații pe A; dacă R are aritatea n, atunci $R^{\mathcal{A}} \subseteq A^n$.
- $\mathbf{C}^{\mathcal{A}} = \{c^{\mathcal{A}} \in A \mid c \in \mathbf{C}\}$.

- A se numește universul structurii $\mathcal{A}$.
- $f^{\mathcal{A}}$ (respectiv $R^{\mathcal{A}}$, $c^{\mathcal{A}}$) se numește interpretarea lui f (respectiv R, c) in $\mathcal{A}$.

Structură

Exemplu

$\mathcal{L}_1 : \mathbf{R} = \{<\}$, $\mathbf{F} = \{s, +\}$, $\mathbf{C} = \{0\}$ cu $ari(s) = 1$, $ari(+) = ari(<) = 2$.

$\mathcal{N} = (\mathbb{N}, s^{\mathcal{N}}, +^{\mathcal{N}}, <^{\mathcal{N}}, 0^{\mathcal{N}})$ unde

- $s^{\mathcal{N}} : \mathbb{N} \to \mathbb{N}, \quad s^{\mathcal{N}}(n) := n + 1,$
- $+^{\mathcal{N}} : \mathbb{N} \times \mathbb{N} \to \mathbb{N}, \quad +^{\mathcal{N}}(n, m) := n + m,$
- $<^{\mathcal{N}} \subseteq \mathbb{N} \times \mathbb{N}, \quad <^{\mathcal{N}} = \{(n, m) \mid n < m\},$
- $0^{\mathcal{N}} := 0$

Interpretare

Fie $\mathcal{L}$ un limbaj de ordinul I și $\mathcal{A}$ o ($\mathcal{L}$-)structură.

Definiție

O interpretare a variabilelor lui $\mathcal{L}$ în $\mathcal{A}$ este o funcție

$$I : V \to A.$$

Interpretare

Fie $\mathcal{L}$ un limbaj de ordinul I și $\mathcal{A}$ o ($\mathcal{L}$-)structură.

Definiție

O interpretare a variabilelor lui $\mathcal{L}$ în $\mathcal{A}$ este o funcție

$$I : V \to A.$$

Definiție

Inductiv, definim interpretarea termenului t în $\mathcal{A}$ sub I ($t_I^{\mathcal{A}}$) prin:

- dacă $t = x_i \in V$, atunci $t_I^{\mathcal{A}} := I(x_i)$
- dacă $t = c \in \mathbf{C}$, atunci $t_I^{\mathcal{A}} := c^{\mathcal{A}}$
- dacă $t = f(t_1, \ldots, t_n)$, atunci $t_I^{\mathcal{A}} := f^{\mathcal{A}}((t_1)_I^{\mathcal{A}}, \ldots, (t_n)_I^{\mathcal{A}})$

Interpretare

Definim inductiv faptul că o formulă este adevărată în $\mathcal{A}$ sub interpretarea I astfel:

Interpretare

Definim inductiv faptul că o formulă este adevărată în $\mathcal{A}$ sub interpretarea I astfel:

- $\mathcal{A}, I \models P(t_1, \ldots, t_n)$ dacă $P^{\mathcal{A}}((t_1)_I^{\mathcal{A}}, \ldots, (t_n)_I^{\mathcal{A}})$

Interpretare

Definim inductiv faptul că o formulă este adevărată în $\mathcal{A}$ sub interpretarea I astfel:

- $\mathcal{A}, I \models P(t_1, \ldots, t_n)$ dacă $P^{\mathcal{A}}((t_1)_I^{\mathcal{A}}, \ldots, (t_n)_I^{\mathcal{A}})$
- $\mathcal{A}, I \models \neg\varphi$ dacă $\mathcal{A}, I \not\models \varphi$

Interpretare

Definim inductiv faptul că o formulă este adevărată în $\mathcal{A}$ sub interpretarea I astfel:

- $\mathcal{A}, I \models P(t_1, \ldots, t_n)$ dacă $P^{\mathcal{A}}((t_1)_I^{\mathcal{A}}, \ldots, (t_n)_I^{\mathcal{A}})$
- $\mathcal{A}, I \models \neg\varphi$ dacă $\mathcal{A}, I \not\models \varphi$
- $\mathcal{A}, I \models \varphi \vee \psi$ dacă $\mathcal{A}, I \models \varphi$ sau $\mathcal{A}, I \models \psi$

Interpretare

Definim inductiv faptul că o formulă este adevărată în $\mathcal{A}$ sub interpretarea I astfel:

- $\mathcal{A}, I \models P(t_1, \ldots, t_n)$ dacă $P^{\mathcal{A}}((t_1)_I^{\mathcal{A}}, \ldots, (t_n)_I^{\mathcal{A}})$
- $\mathcal{A}, I \models \neg\varphi$ dacă $\mathcal{A}, I \not\models \varphi$
- $\mathcal{A}, I \models \varphi \vee \psi$ dacă $\mathcal{A}, I \models \varphi$ sau $\mathcal{A}, I \models \psi$
- $\mathcal{A}, I \models \varphi \wedge \psi$ dacă $\mathcal{A}, I \models \varphi$ și $\mathcal{A}, I \models \psi$

Interpretare

Definim inductiv faptul că o formulă este adevărată în $\mathcal{A}$ sub interpretarea I astfel:

- $\mathcal{A}, I \models P(t_1, \ldots, t_n)$ dacă $P^{\mathcal{A}}((t_1)_I^{\mathcal{A}}, \ldots, (t_n)_I^{\mathcal{A}})$
- $\mathcal{A}, I \models \neg\varphi$ dacă $\mathcal{A}, I \not\models \varphi$
- $\mathcal{A}, I \models \varphi \vee \psi$ dacă $\mathcal{A}, I \models \varphi$ sau $\mathcal{A}, I \models \psi$
- $\mathcal{A}, I \models \varphi \wedge \psi$ dacă $\mathcal{A}, I \models \varphi$ și $\mathcal{A}, I \models \psi$
- $\mathcal{A}, I \models \varphi \to \psi$ dacă $\mathcal{A}, I \not\models \varphi$ sau $\mathcal{A}, I \models \psi$

Interpretare

Definim inductiv faptul că o formulă este adevărată în $\mathcal{A}$ sub interpretarea I astfel:

- $\mathcal{A}, I \models P(t_1, \ldots, t_n)$ dacă $P^{\mathcal{A}}((t_1)_I^{\mathcal{A}}, \ldots, (t_n)_I^{\mathcal{A}})$
- $\mathcal{A}, I \models \neg\varphi$ dacă $\mathcal{A}, I \not\models \varphi$
- $\mathcal{A}, I \models \varphi \vee \psi$ dacă $\mathcal{A}, I \models \varphi$ sau $\mathcal{A}, I \models \psi$
- $\mathcal{A}, I \models \varphi \wedge \psi$ dacă $\mathcal{A}, I \models \varphi$ și $\mathcal{A}, I \models \psi$
- $\mathcal{A}, I \models \varphi \to \psi$ dacă $\mathcal{A}, I \not\models \varphi$ sau $\mathcal{A}, I \models \psi$
- $\mathcal{A}, I \models \forall x\, \varphi$ dacă pentru orice $a \in A$ avem $\mathcal{A}, I_{x_i \leftarrow a} \models \varphi$
- $\mathcal{A}, I \models \exists x\, \varphi$ dacă există $a \in A$ astfel încât $\mathcal{A}, I_{x_i \leftarrow a} \models \varphi$

unde pentru orice $a \in A$, $I_{x \leftarrow a}(y) = \begin{cases} I(y) & \text{dacă } y \neq x \\ a & \text{dacă } y = x \end{cases}$

Interpretare

- O formulă φ este adevărată într-o structură $\mathcal{A}$, notat $\mathcal{A} \models \varphi$, dacă este adevărată în $\mathcal{A}$ sub orice interpretare.

  Spunem că $\mathcal{A}$ este model al lui φ.

- O formulă φ este adevărată în logica de ordinul I, notat $\models \varphi$, dacă este adevărată în orice structură.

Model

Exemplu

Fie limbajul $\mathcal{L}$ cu $\mathbf{F} = \{s\}$, $\mathbf{R} = \{P\}$, $\mathbf{C} = \{0\}$ cu $ari(s) = ari(P) = 1$.

Model

Exemplu

Fie limbajul $\mathcal{L}$ cu $\mathbf{F} = \{s\}$, $\mathbf{R} = \{P\}$, $\mathbf{C} = \{0\}$ cu $ari(s) = ari(P) = 1$.

Fie structura $\mathcal{N} = (\mathbb{N}, s^{\mathcal{N}}, P^{\mathcal{N}}, 0^{\mathcal{N}})$ unde $0^{\mathcal{N}} := 1$ și

- $s^{\mathcal{N}} : \mathbb{N} \to \mathbb{N}$, $s^{\mathcal{N}}(n) := n^2$
- $P^{\mathcal{N}} \subset \mathbb{N}$, $P^{\mathcal{N}} = \{n \mid n \text{ este impar } \}$

Model

Exemplu

Fie limbajul $\mathcal{L}$ cu $\mathbf{F} = \{s\}$, $\mathbf{R} = \{P\}$, $\mathbf{C} = \{0\}$ cu $ari(s) = ari(P) = 1$.

Fie structura $\mathcal{N} = (\mathbb{N}, s^{\mathcal{N}}, P^{\mathcal{N}}, 0^{\mathcal{N}})$ unde $0^{\mathcal{N}} := 1$ și

- $s^{\mathcal{N}} : \mathbb{N} \to \mathbb{N}$, $s^{\mathcal{N}}(n) := n^2$
- $P^{\mathcal{N}} \subset \mathbb{N}$, $P^{\mathcal{N}} = \{n \mid n \text{ este impar } \}$

Demonstrați că $\mathcal{N} \models \forall x\,(P(x) \to P(s(x)))$.

Model

Exemplu

Fie limbajul $\mathcal{L}$ cu $\mathbf{F} = \{s\}$, $\mathbf{R} = \{P\}$, $\mathbf{C} = \{0\}$ cu $ari(s) = ari(P) = 1$.

Fie structura $\mathcal{N} = (\mathbb{N}, s^{\mathcal{N}}, P^{\mathcal{N}}, 0^{\mathcal{N}})$ unde $0^{\mathcal{N}} := 1$ și

- $s^{\mathcal{N}} : \mathbb{N} \to \mathbb{N}$, $s^{\mathcal{N}}(n) := n^2$
- $P^{\mathcal{N}} \subset \mathbb{N}$, $P^{\mathcal{N}} = \{n \mid n \text{ este impar } \}$

Demonstrați că $\mathcal{N} \models \forall x\,(P(x) \to P(s(x)))$.

Fie $I : V \to \mathbb{N}$ o interpretare. Observăm că
$\mathcal{N}, I \models P(x)$ dacă $P^{\mathcal{N}}(I(x))$, adică

Model

Exemplu

Fie limbajul $\mathcal{L}$ cu $\mathbf{F} = \{s\}$, $\mathbf{R} = \{P\}$, $\mathbf{C} = \{0\}$ cu $ari(s) = ari(P) = 1$.

Fie structura $\mathcal{N} = (\mathbb{N}, s^{\mathcal{N}}, P^{\mathcal{N}}, 0^{\mathcal{N}})$ unde $0^{\mathcal{N}} := 1$ și

- $s^{\mathcal{N}} : \mathbb{N} \to \mathbb{N}$, $s^{\mathcal{N}}(n) := n^2$
- $P^{\mathcal{N}} \subset \mathbb{N}$, $P^{\mathcal{N}} = \{n \mid n \text{ este impar } \}$

Demonstrați că $\mathcal{N} \models \forall x\,(P(x) \to P(s(x)))$.

Fie $I : V \to \mathbb{N}$ o interpretare. Observăm că
$\mathcal{N}, I \models P(x)$ dacă $P^{\mathcal{N}}(I(x))$, adică $\mathcal{N}, I \models P(x)$ dacă $I(x)$ este impar.

Model

Exemplu

Fie limbajul $\mathcal{L}$ cu $\mathbf{F} = \{s\}$, $\mathbf{R} = \{P\}$, $\mathbf{C} = \{0\}$ cu $ari(s) = ari(P) = 1$.

Fie structura $\mathcal{N} = (\mathbb{N}, s^{\mathcal{N}}, P^{\mathcal{N}}, 0^{\mathcal{N}})$ unde $0^{\mathcal{N}} := 1$ și

- $s^{\mathcal{N}} : \mathbb{N} \to \mathbb{N}$, $s^{\mathcal{N}}(n) := n^2$
- $P^{\mathcal{N}} \subset \mathbb{N}$, $P^{\mathcal{N}} = \{n \mid n \text{ este impar } \}$

Demonstrați că $\mathcal{N} \models \forall x\,(P(x) \to P(s(x)))$.

Fie $I : V \to \mathbb{N}$ o interpretare. Observăm că
$\mathcal{N}, I \models P(x)$ dacă $P^{\mathcal{N}}(I(x))$, adică $\mathcal{N}, I \models P(x)$ dacă $I(x)$ este impar.

$\mathcal{N}, I \models \forall x\,(P(x) \to P(s(x)))$ dacă

Model

Exemplu

Fie limbajul $\mathcal{L}$ cu $\mathbf{F} = \{s\}$, $\mathbf{R} = \{P\}$, $\mathbf{C} = \{0\}$ cu $ari(s) = ari(P) = 1$.

Fie structura $\mathcal{N} = (\mathbb{N}, s^{\mathcal{N}}, P^{\mathcal{N}}, 0^{\mathcal{N}})$ unde $0^{\mathcal{N}} := 1$ și

- $s^{\mathcal{N}} : \mathbb{N} \to \mathbb{N}$, $s^{\mathcal{N}}(n) := n^2$
- $P^{\mathcal{N}} \subset \mathbb{N}$, $P^{\mathcal{N}} = \{n \mid n \text{ este impar } \}$

Demonstrați că $\mathcal{N} \models \forall x\,(P(x) \to P(s(x)))$.

Fie $I : V \to \mathbb{N}$ o interpretare. Observăm că
$\mathcal{N}, I \models P(x)$ dacă $P^{\mathcal{N}}(I(x))$, adică $\mathcal{N}, I \models P(x)$ dacă $I(x)$ este impar.

$\mathcal{N}, I \models \forall x\,(P(x) \to P(s(x)))$ dacă
$\mathcal{N}, I_{x \leftarrow n} \models P(x) \to P(s(x))$ oricare $n \in N$

Model

Exemplu

Fie limbajul $\mathcal{L}$ cu $\mathbf{F} = \{s\}$, $\mathbf{R} = \{P\}$, $\mathbf{C} = \{0\}$ cu $ari(s) = ari(P) = 1$.

Fie structura $\mathcal{N} = (\mathbb{N}, s^{\mathcal{N}}, P^{\mathcal{N}}, 0^{\mathcal{N}})$ unde $0^{\mathcal{N}} := 1$ și

- $s^{\mathcal{N}} : \mathbb{N} \to \mathbb{N}$, $s^{\mathcal{N}}(n) := n^2$
- $P^{\mathcal{N}} \subset \mathbb{N}$, $P^{\mathcal{N}} = \{n \mid n \text{ este impar } \}$

Demonstrați că $\mathcal{N} \models \forall x\,(P(x) \to P(s(x)))$.

Fie $I : V \to \mathbb{N}$ o interpretare. Observăm că
$\mathcal{N}, I \models P(x)$ dacă $P^{\mathcal{N}}(I(x))$, adică $\mathcal{N}, I \models P(x)$ dacă $I(x)$ este impar.

$\mathcal{N}, I \models \forall x\,(P(x) \to P(s(x)))$ dacă
$\mathcal{N}, I_{x \leftarrow n} \models P(x) \to P(s(x))$ oricare $n \in N$
$\mathcal{N}, I_{x \leftarrow n} \not\models P(x)$ sau $\mathcal{N}, I_{x \leftarrow n} \models P(s(x))$ oricare $n \in N$

Model

Exemplu

Fie limbajul $\mathcal{L}$ cu $\mathbf{F} = \{s\}$, $\mathbf{R} = \{P\}$, $\mathbf{C} = \{0\}$ cu $ari(s) = ari(P) = 1$.

Fie structura $\mathcal{N} = (\mathbb{N}, s^{\mathcal{N}}, P^{\mathcal{N}}, 0^{\mathcal{N}})$ unde $0^{\mathcal{N}} := 1$ și

- $s^{\mathcal{N}} : \mathbb{N} \to \mathbb{N}$, $s^{\mathcal{N}}(n) := n^2$
- $P^{\mathcal{N}} \subset \mathbb{N}$, $P^{\mathcal{N}} = \{n \mid n \text{ este impar } \}$

Demonstrați că $\mathcal{N} \models \forall x\,(P(x) \to P(s(x)))$.

Fie $I : V \to \mathbb{N}$ o interpretare. Observăm că
$\mathcal{N}, I \models P(x)$ dacă $P^{\mathcal{N}}(I(x))$, adică $\mathcal{N}, I \models P(x)$ dacă $I(x)$ este impar.

$\mathcal{N}, I \models \forall x\,(P(x) \to P(s(x)))$ dacă
$\mathcal{N}, I_{x \leftarrow n} \models P(x) \to P(s(x))$ oricare $n \in N$
$\mathcal{N}, I_{x \leftarrow n} \not\models P(x)$ sau $\mathcal{N}, I_{x \leftarrow n} \models P(s(x))$ oricare $n \in N$
$I_{x \leftarrow n}(x)$ nu este impar sau $I_{x \leftarrow n}(s(x))$ este impar oricare $n \in \mathbb{N}$

Model

Exemplu

Fie limbajul $\mathcal{L}$ cu $\mathbf{F} = \{s\}$, $\mathbf{R} = \{P\}$, $\mathbf{C} = \{0\}$ cu $ari(s) = ari(P) = 1$.

Fie structura $\mathcal{N} = (\mathbb{N}, s^{\mathcal{N}}, P^{\mathcal{N}}, 0^{\mathcal{N}})$ unde $0^{\mathcal{N}} := 1$ și

- $s^{\mathcal{N}} : \mathbb{N} \to \mathbb{N}$, $s^{\mathcal{N}}(n) := n^2$
- $P^{\mathcal{N}} \subset \mathbb{N}$, $P^{\mathcal{N}} = \{n \mid n \text{ este impar } \}$

Demonstrați că $\mathcal{N} \models \forall x\,(P(x) \to P(s(x)))$.

Fie $I : V \to \mathbb{N}$ o interpretare. Observăm că
$\mathcal{N}, I \models P(x)$ dacă $P^{\mathcal{N}}(I(x))$, adică $\mathcal{N}, I \models P(x)$ dacă $I(x)$ este impar.

$\mathcal{N}, I \models \forall x\,(P(x) \to P(s(x)))$ dacă
$\mathcal{N}, I_{x \leftarrow n} \models P(x) \to P(s(x))$ oricare $n \in N$
$\mathcal{N}, I_{x \leftarrow n} \not\models P(x)$ sau $\mathcal{N}, I_{x \leftarrow n} \models P(s(x))$ oricare $n \in N$
$I_{x \leftarrow n}(x)$ nu este impar sau $I_{x \leftarrow n}(s(x))$ este impar oricare $n \in \mathbb{N}$
n este par sau n^2 este impar oricare $n \in \mathbb{N}$

Model

Exemplu

Fie limbajul $\mathcal{L}$ cu $\mathbf{F} = \{s\}$, $\mathbf{R} = \{P\}$, $\mathbf{C} = \{0\}$ cu $ari(s) = ari(P) = 1$.

Fie structura $\mathcal{N} = (\mathbb{N}, s^{\mathcal{N}}, P^{\mathcal{N}}, 0^{\mathcal{N}})$ unde $0^{\mathcal{N}} := 1$ și

- $s^{\mathcal{N}} : \mathbb{N} \to \mathbb{N}$, $s^{\mathcal{N}}(n) := n^2$
- $P^{\mathcal{N}} \subset \mathbb{N}$, $P^{\mathcal{N}} = \{n \mid n \text{ este impar } \}$

Demonstrați că $\mathcal{N} \models \forall x\,(P(x) \to P(s(x)))$.

Fie $I : V \to \mathbb{N}$ o interpretare. Observăm că
$\mathcal{N}, I \models P(x)$ dacă $P^{\mathcal{N}}(I(x))$, adică $\mathcal{N}, I \models P(x)$ dacă $I(x)$ este impar.

$\mathcal{N}, I \models \forall x\,(P(x) \to P(s(x)))$ dacă
$\mathcal{N}, I_{x \leftarrow n} \models P(x) \to P(s(x))$ oricare $n \in N$
$\mathcal{N}, I_{x \leftarrow n} \not\models P(x)$ sau $\mathcal{N}, I_{x \leftarrow n} \models P(s(x))$ oricare $n \in N$
$I_{x \leftarrow n}(x)$ nu este impar sau $I_{x \leftarrow n}(s(x))$ este impar oricare $n \in \mathbb{N}$
n este par sau n^2 este impar oricare $n \in \mathbb{N}$

ceea ce este întodeauna adevărat.

Consecință logică

Definiție

O formulă φ este o consecință logică a formulelor $\varphi_1, \ldots, \varphi_n$, notat

$$\varphi_1, \ldots, \varphi_n \models \varphi,$$

dacă pentru orice structură $\mathcal{A}$

dacă $\mathcal{A} \models \varphi_1$ și ... și $\mathcal{A} \models \varphi_n$, atunci $\mathcal{A} \models \varphi$

Consecință logică

Definiție

O formulă φ este o consecință logică a formulelor $\varphi_1, \ldots, \varphi_n$, notat

$$\varphi_1, \ldots, \varphi_n \models \varphi,$$

dacă pentru orice structură $\mathcal{A}$

$$\text{dacă } \mathcal{A} \models \varphi_1 \text{ și } \ldots \text{ și } \mathcal{A} \models \varphi_n, \text{ atunci } \mathcal{A} \models \varphi$$

Problemă semidecidabilă!

Nu există algoritm care să decidă mereu dacă o formula este sau nu consecință logică a altei formule în logica de ordinul I!

Logica de ordinul I - sintaxa

Limbaj de ordinul I $\mathcal{L}$

- unic determinat de $\tau = (\mathbf{R}, \mathbf{F}, \mathbf{C}, ari)$

Termenii lui $\mathcal{L}$, notați $Trm_{\mathcal{L}}$, sunt definiți inductiv astfel:

- orice variabilă este un termen;
- orice simbol de constantă este un termen;
- dacă $f \in \mathbf{F}$, $ar(f) = n$ și $t_1, \ldots, t_n$ sunt termeni, atunci $f(t_1, \ldots, t_n)$ este termen.

Formulele atomice ale lui $\mathcal{L}$ sunt definite astfel:

- dacă $R \in \mathbf{R}$, $ar(R) = n$ și $t_1, \ldots, t_n$ sunt termeni, atunci $R(t_1, \ldots, t_n)$ este formulă atomică.

Formulele lui $\mathcal{L}$ sunt definite astfel:

- orice formulă atomică este o formulă
- dacă φ este o formulă, atunci $\neg\varphi$ este o formulă
- dacă φ și ψ sunt formule, atunci $\varphi \vee \psi$, $\varphi \wedge \psi$, $\varphi \to \psi$ sunt formule
- dacă φ este o formulă și x este o variabilă, atunci $\forall x\, \varphi$, $\exists x\, \varphi$ sunt formule

Logica de ordinul I - semantică (opțional)

O structură este de forma $\mathcal{A} = (A, \mathbf{F}^{\mathcal{A}}, \mathbf{R}^{\mathcal{A}}, \mathbf{C}^{\mathcal{A}})$, unde

- A este o mulțime nevidă
- $\mathbf{F}^{\mathcal{A}} = \{f^{\mathcal{A}} \mid f \in \mathbf{F}\}$ este o mulțime de operații pe A; dacă f are aritatea n, atunci $f^{\mathcal{A}} : A^n \to A$.
- $\mathbf{R}^{\mathcal{A}} = \{R^{\mathcal{A}} \mid R \in \mathbf{R}\}$ este o mulțime de relații pe A; dacă R are aritatea n, atunci $R^{\mathcal{A}} \subseteq A^n$.
- $\mathbf{C}^{\mathcal{A}} = \{c^{\mathcal{A}} \in A \mid c \in \mathbf{C}\}$.

O interpretare a variabilelor lui $\mathcal{L}$ în $\mathcal{A}$ ($\mathcal{A}$-interpretare) este o funcție $I : V \to A$.

Inductiv, definim interpretarea termenului t în $\mathcal{A}$ sub I notat $t_I^{\mathcal{A}}$.

Inductiv, definim când o formulă este adevărată în $\mathcal{A}$ în interpretarea I notat $\mathcal{A}, I \models \varphi$. În acest caz spunem că $(\mathcal{A}, I)$ este model pentru φ.

O formulă φ este adevărată într-o structură $\mathcal{A}$, notat $\mathcal{A} \models \varphi$, dacă este adevărată în $\mathcal{A}$ sub orice interpretare. Spunem că $\mathcal{A}$ este model al lui φ.

O formulă φ este adevărată în logica de ordinul I, notat $\models \varphi$, dacă este adevărată în orice structură. O formulă φ este validă dacă $\models \varphi$.

O formulă φ este satisfiabilă dacă există o structură $\mathcal{A}$ și o $\mathcal{A}$-interpretare I astfel încât $\mathcal{A}, I \models \varphi$.

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

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

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

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

Logica Horn

Literali

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

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

Literali

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

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

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

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

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

Clauze

- O clauză este o disjuncție de literali.

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

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.

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.

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 $\square$

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.

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.

Rezoluția este o metodă de verificare a satisfiabilității unei mulțimi de clauze.

Clauze în logica de ordinul I

$$\{\neg Q_1, \ldots, \neg Q_n, P_1, \ldots, P_k\}$$

unde $n, k \geq 0$ și $Q_1, \ldots, Q_n, P_1, \ldots, P_k$ sunt formule atomice.

- formula corespunzătoare este

$$\forall x_1 \ldots \forall x_m (\neg Q_1 \vee \ldots \vee \neg Q_n \vee P_1 \vee \ldots \vee P_k)$$

  unde $x_1, \ldots, x_m$ sunt toate variabilele care apar în clauză

- echivalent, putem scrie

$$\forall x_1 \ldots \forall x_m (Q_1 \wedge \ldots \wedge Q_n \rightarrow P_1 \vee \ldots \vee P_k)$$

- cuantificarea universală a clauzelor este implicită

$$Q_1 \wedge \ldots \wedge Q_n \rightarrow P_1 \vee \ldots \vee P_k$$

Clauze definite. Programe logice. Clauze Horn

- clauză:

  $\{\neg Q_1, \ldots, \neg Q_n, P_1, \ldots, P_k\}$ sau $Q_1 \wedge \ldots \wedge Q_n \rightarrow P_1 \vee \ldots \vee P_k$

  unde $n, k \geq 0$ și $Q_1, \ldots, Q_n, P_1, \ldots, P_k$ sunt formule atomice.

Clauze definite. Programe logice. Clauze Horn

- clauză:

  $\{\neg Q_1, \ldots, \neg Q_n, P_1, \ldots, P_k\}$ sau $Q_1 \wedge \ldots \wedge Q_n \rightarrow P_1 \vee \ldots \vee P_k$

  unde $n, k \geq 0$ și $Q_1, \ldots, Q_n, P_1, \ldots, P_k$ sunt formule atomice.

- clauză program definită: $k = 1$
  - cazul $n > 0$: $Q_1 \wedge \ldots \wedge Q_n \rightarrow P$
  - cazul $n = 0$: $\top \rightarrow P$ (clauză unitate, fapt)

  Program logic definit = mulțime finită de clauze definite

Clauze definite. Programe logice. Clauze Horn

- clauză:

  $\{\neg Q_1, \ldots, \neg Q_n, P_1, \ldots, P_k\}$ sau $Q_1 \wedge \ldots \wedge Q_n \rightarrow P_1 \vee \ldots \vee P_k$

  unde $n, k \geq 0$ și $Q_1, \ldots, Q_n, P_1, \ldots, P_k$ sunt formule atomice.

- clauză program definită: $k = 1$
  - cazul $n > 0$: $Q_1 \wedge \ldots \wedge Q_n \rightarrow P$
  - cazul $n = 0$: $\top \rightarrow P$ (clauză unitate, fapt)

  Program logic definit = mulțime finită de clauze definite

- clauză scop definită (țintă, întrebare): k=0
  - $Q_1 \wedge \ldots \wedge Q_n \rightarrow \bot$

- clauza vidă $\square$: $n = k = 0$

Clauze definite. Programe logice. Clauze Horn

- clauză:

  $\{\neg Q_1, \ldots, \neg Q_n, P_1, \ldots, P_k\}$ sau $Q_1 \wedge \ldots \wedge Q_n \rightarrow P_1 \vee \ldots \vee P_k$

  unde $n, k \geq 0$ și $Q_1, \ldots, Q_n, P_1, \ldots, P_k$ sunt formule atomice.

- clauză program definită: $k = 1$
  - cazul $n > 0$: $Q_1 \wedge \ldots \wedge Q_n \rightarrow P$
  - cazul $n = 0$: $\top \rightarrow P$ (clauză unitate, fapt)

  Program logic definit = mulțime finită de clauze definite

- clauză scop definită (țintă, întrebare): k=0
  - $Q_1 \wedge \ldots \wedge Q_n \rightarrow \bot$

- clauza vidă $\square$: $n = k = 0$

  Clauza Horn = clauză program definită sau clauză scop ($k \leq 1$)

Clauze Horn țintă

- clauză scop definită (țintă, întrebare): $Q_1 \wedge \ldots \wedge Q_n \rightarrow \bot$
  - fie $x_1, \ldots, x_m$ toate variabilele care apar în $Q_1, \ldots, Q_n$
    $\forall x_1 \ldots \forall x_m(\neg Q_1 \vee \ldots \vee \neg Q_n) \boxminus \neg\exists x_1 \ldots \exists x_m(Q_1 \wedge \ldots \wedge Q_n)$
  - clauza țintă o vom scrie $Q_1, \ldots, Q_n$

Negația unei "întrebări" în PROLOG este clauză Horn țintă.

Programare logica

- Logica clauzelor definite/Logica Horn: un fragment al logicii de ordinul I în care singurele formule admise sunt clauze Horn
  - formule atomice: $P(t_1, \ldots, t_n)$
  - $Q_1 \wedge \ldots \wedge Q_n \rightarrow P$
    unde toate Q_i, P sunt formule atomice, $\top$ sau $\bot$

Programare logica

- Logica clauzelor definite/Logica Horn: un fragment al logicii de ordinul I în care singurele formule admise sunt clauze Horn
  - formule atomice: $P(t_1, \ldots, t_n)$
  - $Q_1 \wedge \ldots \wedge Q_n \rightarrow P$
    unde toate Q_i, P sunt formule atomice, $\top$ sau $\bot$

- Problema programării logice: reprezentăm cunoștințele ca o mulțime de clauze definite KB și suntem interesați să aflăm răspunsul la o întrebare de forma $Q_1 \wedge \ldots \wedge Q_n$, unde toate Q_i sunt formule atomice

$$KB \models Q_1 \wedge \ldots \wedge Q_n$$

  - Variabilele din KB sunt cuantificate universal.
  - Variabilele din $Q_1, \ldots, Q_n$ sunt cuantificate existențial.

Limbajul PROLOG are la bază logica clauzelor Horn.

Logica clauzelor definite

Exemplu

Fie următoarele clauze definite:

$father(jon, ken).$
$father(ken, liz).$
$father(X, Y) \rightarrow ancestor(X, Y)$
$daugther(X, Y) \rightarrow ancestor(Y, X)$
$ancestor(X, Y) \wedge ancestor(Y, Z) \rightarrow ancestor(X, Z)$

Putem întreba:

- $ancestor(jon, liz)$
- dacă există Q astfel încât $ancestor(Q, ken)$
  (adică $\exists Q\, ancestor(Q, ken)$)

Pe săptămâna viitoare!