Curs 2

Din cursul trecut

Programare logică – cazul logicii propoziționale

- O clauză definită este o formulă care poate avea una din formele:
  1. q (clauză unitate) (un fapt în Prolog `q.`)
  2. $p_1 \wedge \ldots \wedge p_k \to q$ (o regulă în Prolog `q :- p1,...,pn`)

  unde $q, p_1, \ldots, p_n$ sunt atomi.

- Un "program logic" este o listă $F_1, \ldots, F_n$ de clauze definite.

- O țintă (*goal*) este o listă $g_1, \ldots, g_m$ de atomi.

- Sarcina sistemului este să stabilească:

$$F_1, \ldots, F_n \models g_1 \wedge \ldots \wedge g_m.$$

- Scop: Vrem să găsim metode sintactice pentru a rezolva problema de mai sus!

Cuprins

Cazul logicii propoziționale

Sistem de deducție

Sistem de deducție pentru clauze definite propoziționale

Pentru o mulțime S de clauze definite propoziționale, avem

Sistem de deducție

Sistem de deducție pentru clauze definite propoziționale

Pentru o mulțime S de clauze definite propoziționale, avem

- Axiome: orice clauză din S

Sistem de deducție

Sistem de deducție pentru clauze definite propoziționale

Pentru o mulțime S de clauze definite propoziționale, avem

- Axiome: orice clauză din S
- Reguli de deducție:

$$\frac{P \quad P \to Q}{Q}\ (MP) \qquad\qquad \frac{P \quad Q}{P \wedge Q}\ (andI)$$

  - Aceste reguli ne permit să deducem formula de sub linie din formulele de deasupra liniei.

Sistemul de deducție

$$\frac{P \quad P \to Q}{Q}\ (MP) \qquad\qquad \frac{P \quad Q}{P \wedge Q}\ (andI)$$

Exemplu

$$\begin{array}{rcl}
\texttt{oslo} & \to & \texttt{windy} \\
\texttt{oslo} & \to & \texttt{norway} \\
\texttt{norway} & \to & \texttt{cold} \\
\texttt{cold} \wedge \texttt{windy} & \to & \texttt{winterIsComing} \\
 & & \texttt{oslo}
\end{array}$$

Sistemul de deducție

$$\frac{P \quad P \to Q}{Q}\ (MP) \qquad\qquad \frac{P \quad Q}{P \land Q}\ (andI)$$

Exemplu

$$\begin{array}{rcl}
\texttt{oslo} & \to & \texttt{windy} \\
\texttt{oslo} & \to & \texttt{norway} \\
\texttt{norway} & \to & \texttt{cold} \\
\texttt{cold} \land \texttt{windy} & \to & \texttt{winterIsComing} \\
 & & \texttt{oslo}
\end{array}$$

$$\dfrac{\dfrac{\dfrac{oslo \qquad oslo \to norway}{norway} \qquad norway \to cold}{cold} \qquad \dfrac{oslo \qquad oslo \to windy}{windy}}{cold \land windy}$$

Sistemul de deducție

$$\frac{P \quad P \to Q}{Q}\ (MP) \qquad\qquad \frac{P \quad Q}{P \land Q}\ (andI)$$

Exemplu

$$\begin{array}{rcl}
\texttt{oslo} & \to & \texttt{windy} \\
\texttt{oslo} & \to & \texttt{norway} \\
\texttt{norway} & \to & \texttt{cold} \\
\texttt{cold} \land \texttt{windy} & \to & \texttt{winterIsComing} \\
 & & \texttt{oslo}
\end{array}$$

$$\dfrac{\dfrac{\dfrac{oslo \qquad oslo \to norway}{norway} \qquad norway \to cold}{cold} \qquad \dfrac{oslo \qquad oslo \to windy}{windy}}{cold \land windy}$$

$$\frac{cold \land windy \qquad cold \land windy \to winterIsComing}{winterIsComing}$$

Sistemul de deducție

Spunem că putem deduce o formulă Q din S în sistemul de deducție,

$$S \vdash Q$$

dacă există o secvență de formule $Q_1, \ldots, Q_n$ astfel încât $Q_n = Q$ și fiecare Q_i:

- fie aparține lui S
- fie se poate deduce din $Q_1, \ldots, Q_{i-1}$ folosind regulile de deducție

Sistemul de deducție

Putem folosi sistemul de deducție pentru a deduce alți atomi:

Sistemul de deducție

Putem folosi sistemul de deducție pentru a deduce alți atomi:

- Clauzele unitate din S (atomii $p_i \in S$) sunt considerate adevărate.
  - Sunt deduși ca axiome.

- Putem deduce că un nou atom r este adevărat dacă
  - am dedus că $p_1, \ldots, p_n$ sunt adevărați, și
  - $p_1 \wedge \ldots \wedge p_n \rightarrow r$ este în S.

  O astfel de derivare folosește de $n - 1$ ori **andI** și o data **MP**.

Sistemul de deducție

Putem folosi sistemul de deducție pentru a deduce alți atomi:

- Clauzele unitate din S (atomii $p_i \in S$) sunt considerate adevărate.
  - Sunt deduși ca axiome.

- Putem deduce că un nou atom r este adevărat dacă
  - am dedus că $p_1, \ldots, p_n$ sunt adevărați, și
  - $p_1 \wedge \ldots \wedge p_n \to r$ este în S.

  O astfel de derivare folosește de $n-1$ ori **andI** și o data **MP**.

Deci putem construi mulțimi din ce în ce mai mari de atomi care sunt consecințe logice din S, și pentru care există derivări din S.

Completitudinea sistemului de deducție

- Se poate demonstra că aceste reguli sunt corecte, folosind tabelele de adevăr.
  - Dacă formulele de deasupra liniei sunt adevărate, atunci și formula de sub linie este adevărată.

Completetitudinea sistemului de deducție

- Se poate demonstra că aceste reguli sunt corecte, folosind tabelele de adevăr.
  - Dacă formulele de deasupra liniei sunt adevărate, atunci și formula de sub linie este adevărată.

- Mai mult, sistemul de deducție este și complet. Adică

  dacă $S \models Q$, atunci $S \vdash Q$.

  - Dacă Q este o consecință logică, atunci există o derivare a sa folosind sistemul de deducție, unde Q este o conjuncție de atomi.

Completetitudinea sistemului de deducție

- Se poate demonstra că aceste reguli sunt corecte, folosind tabelele de adevăr.
  - Dacă formulele de deasupra liniei sunt adevărate, atunci și formula de sub linie este adevărată.
- Mai mult, sistemul de deducție este și complet. Adică

  $$\text{dacă } S \models Q, \text{ atunci } S \vdash Q.$$

  - Dacă Q este o consecință logică, atunci există o derivare a sa folosind sistemul de deducție, unde Q este o conjuncție de atomi.
- În continuare vom demonstra că această afirmație este adevărată.

Mulțimi de mulțimi

- Fiind dată o mulțime X, putem considera mulțimea tuturor submulțimilor (mulțimea părților) lui X, notată $\mathcal{P}(X)$.

  Exemplu. $\mathcal{P}(\{1,2,3\})$ este mulțimea cu 8 mulțimi:
  $$\{\{\},\{1\},\{2\},\{3\},\{1,2\},\{1,3\},\{2,3\},\{1,2,3\}\}$$

  - Dacă X are n elemente, atunci $\mathcal{P}(X)$ are 2^n elemente.
  - Putem considera și mulțimea părților unei mulțimi infinite.

Mulțimi de mulțimi

- Fiind dată orice mulțime de mulțimi Y (nu neapărat mulțimea părților unei mulțimi), definim intersecția lui Y, notată $\bigcap Y$, ca fiind mulțimea tuturor elementelor ce apar în **toate** mulțimile lui Y.

$$x \in \bigcap Y \quad \text{ddacă} \quad \forall Z (Z \in Y \to x \in Z)$$

Exemplu. $\bigcap\{\{1,2,3\},\{1,3\},\{2,3\}\} =$

Mulțimi de mulțimi

- Fiind dată orice mulțime de mulțimi Y (nu neapărat mulțimea părților unei mulțimi), definim intersecția lui Y, notată $\bigcap Y$, ca fiind mulțimea tuturor elementelor ce apar în **toate** mulțimile lui Y.

$$x \in \bigcap Y \quad \text{ddacă} \quad \forall Z(Z \in Y \to x \in Z)$$

Exemplu. $\bigcap\{\{1,2,3\},\{1,3\},\{2,3\}\} = \{3\}$

Funcții monotone

Fie X, Y mulțimi de mulțimi și $f : X \to Y$ o funcție.

Spunem că f este monotonă dacă pentru orice mulțimi $X_1, X_2 \in X$,

$$\text{dacă } X_1 \subseteq X_2, \text{ atunci } f(X_1) \subseteq f(X_2).$$

Funcții monotone

Fie X, Y mulțimi de mulțimi și $f : X \to Y$ o funcție.

Spunem că f este monotonă dacă pentru orice mulțimi $X_1, X_2 \in X$,

$$\text{dacă } X_1 \subseteq X_2, \text{ atunci } f(X_1) \subseteq f(X_2).$$

Exemplu

Fie următoarele funcții $f_i : \mathcal{P}(\{1,2,3\}) \to \mathcal{P}(\{1,2,3\})$

- $f_1(Y) = Y \cup \{1\}$ este monotonă.

Funcții monotone

Fie X, Y mulțimi de mulțimi și $f : X \to Y$ o funcție.

Spunem că f este monotonă dacă pentru orice mulțimi $X_1, X_2 \in X$,

$$\text{dacă } X_1 \subseteq X_2, \text{ atunci } f(X_1) \subseteq f(X_2).$$

Exemplu

Fie următoarele funcții $f_i : \mathcal{P}(\{1,2,3\}) \to \mathcal{P}(\{1,2,3\})$

- $f_1(Y) = Y \cup \{1\}$ este monotonă.
- $f_2(Y) = \begin{cases} \{1\} & \text{dacă } 1 \in Y \\ \{\} & \text{altfel} \end{cases}$ este monotonă.

Funcții monotone

Fie X, Y mulțimi de mulțimi și $f : X \to Y$ o funcție.

Spunem că f este monotonă dacă pentru orice mulțimi $X_1, X_2 \in X$,

$$\text{dacă } X_1 \subseteq X_2, \text{ atunci } f(X_1) \subseteq f(X_2).$$

Exemplu

Fie următoarele funcții $f_i : \mathcal{P}(\{1,2,3\}) \to \mathcal{P}(\{1,2,3\})$

- $f_1(Y) = Y \cup \{1\}$ este monotonă.
- $f_2(Y) = \begin{cases} \{1\} & \text{dacă } 1 \in Y \\ \{\} & \text{altfel} \end{cases}$ este monotonă.
- $f_2(Y) = \begin{cases} \{\} & \text{dacă } 1 \in Y \\ \{1\} & \text{altfel} \end{cases}$ nu este monotonă.

Puncte fixe

- Un punct fix al unei funcții $f : \mathcal{P}(X) \to \mathcal{P}(X)$ este o mulțime $Y \subseteq X$ astfel încât $f(Y) = Y$.

Puncte fixe

- Un punct fix al unei funcții $f : \mathcal{P}(X) \to \mathcal{P}(X)$ este o mulțime $Y \subseteq X$ astfel încât $f(Y) = Y$.
- Un cel mai mic punct fix (lfp) Y al lui f este un punct fix conținut în toate celelalte puncte fixe ale lui f, adică
  - Y este punct fix, și
  - dacă Z este tot un punct fix, atunci $Y \subseteq Z$.

Puncte fixe

- Un punct fix al unei funcții $f : \mathcal{P}(X) \to \mathcal{P}(X)$ este o mulțime $Y \subseteq X$ astfel încât $f(Y) = Y$.
- Un cel mai mic punct fix (lfp) Y al lui f este un punct fix conținut în toate celelalte puncte fixe ale lui f, adică
  - Y este punct fix, și
  - dacă Z este tot un punct fix, atunci $Y \subseteq Z$.

Teoremă

Dacă $f : \mathcal{P}(X) \to \mathcal{P}(X)$ este monotonă, atunci f are un cel mai mic punct fix.

Clauze definite și funcții monotone

Fie A mulțimea atomilor $p_1, p_2, \ldots$ care apar în S.

Clauze definite și funcții monotone

Fie A mulțimea atomilor $p_1, p_2, \ldots$ care apar în S.

Fie $Baza = \{p_i \mid p_i \in S\}$ mulțimea atomilor care apar în clauzele unitate din S.

Clauze definite și funcții monotone

Fie A mulțimea atomilor $p_1, p_2, \ldots$ care apar în S.

Fie $Baza = \{p_i \mid p_i \in S\}$ mulțimea atomilor care apar în clauzele unitate din S.

Definim funcția $f_S : \mathcal{P}(A) \to \mathcal{P}(A)$ prin

$$\boxed{\begin{aligned} f_S(Y) = Y \cup Baza & \\ & \cup \{a \in A \mid (s_1 \wedge \ldots \wedge s_n \to a) \text{ este în } S, \\ & \qquad s_1 \in Y, \ldots, s_n \in Y\} \end{aligned}}$$

Clauze definite și funcții monotone

Fie A mulțimea atomilor $p_1, p_2, \ldots$ care apar în S.

Fie $Baza = \{p_i \mid p_i \in S\}$ mulțimea atomilor care apar în clauzele unitate din S.

Definim funcția $f_S : \mathcal{P}(A) \to \mathcal{P}(A)$ prin

$$
\boxed{
\begin{aligned}
f_S(Y) = Y \cup Baza \\
& \cup \{a \in A \mid (s_1 \land \ldots \land s_n \to a) \text{ este în } S, \\
& \qquad s_1 \in Y, \ldots, s_n \in Y\}
\end{aligned}
}
$$

Exercițiu. Arătați că funcția f_S este monotonă.

Clauze definite și funcții monotone

- Analizați ce se întamplă când considerâm succesiv

$$\{\}, \quad f_S(\{\}), \quad f_S(f_S(\{\})), \quad f_S(f_S(f_S(\{\}))), \ldots$$

La fiecare aplicare a lui f_S, rezultatul fie se mărește, fie rămâne neschimbat.

Clauze definite și funcții monotone

- Analizați ce se întamplă când considerâm succesiv

$$\{\}, \quad f_S(\{\}), \quad f_S(f_S(\{\})), \quad f_S(f_S(f_S(\{\}))), \ldots$$

  La fiecare aplicare a lui f_S, rezultatul fie se mărește, fie rămâne neschimbat.

- Să presupunem că în S avem k atomi. Atunci după $k+1$ aplicări ale lui f_S, trebuie să existe un punct în șirul de mulțimi obținute de unde o nouă aplicare a lui f_S nu mai schimbă rezultatul (punct fix):

$$f_S(X) = X$$

Clauze definite și funcții monotone

- Analizați ce se întamplă când considerâm succesiv

$$\{\}, \quad f_S(\{\}), \quad f_S(f_S(\{\})), \quad f_S(f_S(f_S(\{\}))), \ldots$$

  La fiecare aplicare a lui f_S, rezultatul fie se mărește, fie rămâne neschimbat.

- Să presupunem că în S avem k atomi. Atunci după $k+1$ aplicări ale lui f_S, trebuie să existe un punct în șirul de mulțimi obținute de unde o nouă aplicare a lui f_S nu mai schimbă rezultatul (punct fix):

$$f_S(X) = X$$

- Dacă aplicăm f_S succesiv ca mai devreme până găsim un X cu proprietatea $f_S(X) = X$, atunci găsim cel mai mic punct fix al lui f_S.

Cel mai mic punct fix

$$
\begin{aligned}
f_S(Y) = Y \cup \mathit{Baza} \\
&\cup \{a \in A \mid (s_1 \wedge \ldots \wedge s_n \to a) \text{ este în } S, \\
&\qquad s_1 \in Y, \ldots, s_n \in Y\}
\end{aligned}
$$

Exemplu

Pentru

$$
\begin{array}{rcl}
\mathit{cold} & \to & \mathit{wet} \\
\mathit{wet} \wedge \mathit{wet} & \to & \mathit{scotland}
\end{array}
$$

obținem $f_S(\{\}) = \{\}$, deci $\{\}$ este cel mai mic punct fix.

De aici deducem că niciun atom nu este consecință logică a formulelor de mai sus.

Exemplu

Exemplu

În sintaxa Prolog

```
cold.
wet :- cold.
dry :- dry.
scotland :- wet, cold.
```

$$f_S(Y) = Y \cup Baza$$
$$\cup\{a \in A \mid (s_1 \wedge \ldots \wedge s_n \rightarrow a) \text{ este în } S,$$
$$s_1 \in Y, \ldots, s_n \in Y\}$$

$$\begin{array}{rcl} f_S(\{\}) & = & \{\ cold\ \} \\ f_S(\{\ cold\ \}) & = & \{\ cold, wet\ \} \\ f_S(\{\ cold, wet\ \}) & = & \{\ cold, wet, scotland\ \} \\ f_S(\{\ cold, wet, scotland\ \}) & = & \{\ cold, wet, scotland\ \} \end{array}$$

Deci cel mai mic punct fix este $\{\ cold, wet, scotland\ \}$.

Programe logice și cel mai mic punct fix

Teoremă

Fie X este cel mai mic punct fix al funcției f_S. Atunci

$$q \in X \quad \textit{ddacă} \quad S \models q.$$

Intuiție: Cel mai mic punct fix al funcției f_S este mulțimea tuturor atomilor care sunt consecințe logice ale programului.

Funcția $f_S : \mathcal{P}(A) \to \mathcal{P}(A)$ este definită prin

$$\boxed{\begin{aligned} f_S(Y) = {} & Y \cup Baza \\ & \cup \{a \in A \mid (s_1 \wedge \ldots \wedge s_n \to a) \text{ este în } S, s_1 \in Y, \ldots, s_n \in Y\} \end{aligned}}$$

unde A este mulțimea atomilor din S și $Baza = \{p_i \mid p_i \in S\}$ este mulțimea atomilor care apar în clauzele unitate din S.

Programe logice și cel mai mic punct fix

Demonstrație

$(\Rightarrow)$ $q \in X \Rightarrow S \models q$.

- Funcția f_S conservă atomii adevărați.
- Deci, dacă fiecare clauză unitate din S este adevărată, după fiecare aplicare a funcției f_S obținem o mulțime adevărată de atomi.

Programe logice și cel mai mic punct fix

Demonstrație

$(\Rightarrow)$ $q \in X \Rightarrow S \models q$.

- Funcția f_S conservă atomii adevărați.
- Deci, dacă fiecare clauză unitate din S este adevărată, după fiecare aplicare a funcției f_S obținem o mulțime adevărată de atomi.

$(\Leftarrow)$ $q \in X \Leftarrow S \models q$.

- Fie $S \models q$. Presupunem prin absurd că $q \notin X$.
- Căutăm o interpretare I care face fiecare formulă din S adevărată, dar q falsă.

Programe logice și cel mai mic punct fix

Demonstrație (cont.)

- Fie interpretarea

$$I(p) = \begin{cases} \textbf{true}, & \text{dacă } p \in X \\ \textbf{false}, & \text{altfel} \end{cases}$$

Programe logice și cel mai mic punct fix

Demonstrație (cont.)

- Fie interpretarea

$$I(p) = \begin{cases} \textbf{true}, & \text{dacă } p \in X \\ \textbf{false}, & \text{altfel} \end{cases}$$

- Evident, această interpretare face q falsă.

Programe logice și cel mai mic punct fix

Demonstrație (cont.)

- Fie interpretarea
$$I(p) = \begin{cases} \textbf{true}, & \text{dacă } p \in X \\ \textbf{false}, & \text{altfel} \end{cases}$$
- Evident, această interpretare face q falsă.
- Arătăm că $I \models P$, pentru orice formulă $P \in S$.

Programe logice și cel mai mic punct fix

Demonstrație (cont.)

- Fie interpretarea

$$I(p) = \begin{cases} \textbf{true}, & \text{dacă } p \in X \\ \textbf{false}, & \text{altfel} \end{cases}$$

- Evident, această interpretare face q falsă.
- Arătăm că $I \models P$, pentru orice formulă $P \in S$.
- Fie $P \in S$. Avem două cazuri:

Demonstrație (cont.)

- Fie interpretarea

$$I(p) = \begin{cases} \textbf{true}, & \text{dacă } p \in X \\ \textbf{false}, & \text{altfel} \end{cases}$$

- Evident, această interpretare face q falsă.
- Arătăm că $I \models P$, pentru orice formulă $P \in S$.
- Fie $P \in S$. Avem două cazuri:

1. P este o clauză unitate. Atunci $P \in X$, deci $I \models P$.

Programe logice și cel mai mic punct fix

Demonstrație (cont.)

- Fie interpretarea
$$I(p) = \begin{cases} \textbf{true}, & \text{dacă } p \in X \\ \textbf{false}, & \text{altfel} \end{cases}$$
- Evident, această interpretare face q falsă.
- Arătăm că $I \models P$, pentru orice formulă $P \in S$.
- Fie $P \in S$. Avem două cazuri:

1. P este o clauză unitate. Atunci $P \in X$, deci $I \models P$.
2. P este de forma $p_1 \wedge \ldots \wedge p_n \to r$. Atunci avem două cazuri:
   - există un p_i, $i = 1, \ldots, n$, care nu este în X. Deci $I \models P$.
   - toți p_i, $i = 1, \ldots, n$, sunt în X. Atunci $r \in f_S(X) = X$, deci $I \models r$. În concluzie $I \models P$.

□

Sistemul de deducție

Corolar

Sistemul de deducție pentru clauze definite propoziționale este complet pentru a arăta clauze unitate:

dacă $S \models q$, atunci $S \vdash q$.

Demonstrație

- Presupunem $S \models q$.
- Atunci $q \in X$, unde X este cel mai mic punct fix al funcției f_S.
- Fiecare aplicare a funcției f_S produce o mulțime demonstrabilă de atomi.
- Cum cel mai mic punct fix este atins după un număr finit de aplicări ale lui f_S, orice $a \in X$ are o derivare.

$\square$

Metodă de decizie

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

Metoda constă în:

- calcularea celui mai mic punct fix X al funcției f_S
- dacă $q \in X$ atunci returnăm **true**, altfel returnăm **false**

Metodă de decizie

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

Metoda constă în:

- calcularea celui mai mic punct fix X al funcției f_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 $S \vdash q$:

Metoda constă în:

- calcularea celui mai mic punct fix X al funcției f_S
- dacă $q \in X$ atunci returnăm **true**, altfel returnăm **false**

Această metodă se termină.
Exercițiu. De ce?

Nu acesta este algoritmul folosit de Prolog!

Cazul calculului cu predicate

Logica de ordinul I

- Sloganul programării logice:

  Un program este o teorie într-o logica formală,
  iar execuția sa este o deducție în teorie.

- Programarea logică țintește să folosească logica de ordinul I (calculul cu predicate) ca limbaj de reprezentare.

- În această reprezentare, programele sunt teorii logice – mulțimi de formule din calculul cu predicate.

- Reamintim că problema constă în căutarea unei derivări a unei întrebări (formule) din program (teorie).

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 $\mathbf{P}$ de simboluri de relații
- o mulțime $\mathbf{F}$ de simboluri de funcții
- o mulțime $\mathbf{C}$ de simboluri de constante
- o funcție aritate $ar : \mathbf{F} \cup \mathbf{P} \rightarrow \mathbb{N}^*$

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 $\mathcal{L}$

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!

- 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 acestă distincție.

Logica de ordinul I

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

- 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 asftel:

- 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 asftel:

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

Logica de ordinul I

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

- dacă $R \in \mathbf{R}$, $ar(R) = n$ si $t_1, \ldots, t_n$ 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 asftel:

- orice formulă atomică este o formulă
- dacă A este o formulă, atunci $\neg A$ este o formulă
- dacă A și B sunt formule, atunci $A \vee B$, $A \wedge B$, $A \rightarrow B$ sunt formule
- dacă A este o formulă și x_i este o variabilă, atunci $(\forall x_i)A$, $(\exists x_i)A$ sunt formule

Logica de ordinul I

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

- orice formulă atomică este o formulă
- dacă A este o formulă, atunci $\neg A$ este o formulă
- dacă A și B sunt formule, atunci $A \vee B$, $A \wedge B$, $A \rightarrow B$ sunt formule
- dacă A este o formulă și x_i este o variabilă, atunci $(\forall x_i)A$, $(\exists x_i)A$ sunt formule

Exemplu

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

Semantica

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

Modelarea unei lumi

Presupunem că putem descrie o lume prin:

- o mulțime de obiecte
- funcții
- relații

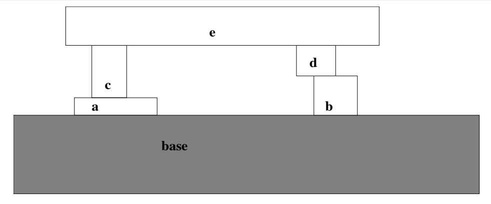
unde

- funcțiile duc obiecte în obiecte
- relațiile cu n argumente descriu proprietățile a n obiecte

Modelarea unei lumi

Exemplu

Să considerăm o lume în care avem cutii:



- Putem descrie lumea folosind obiecte
$$O = \{base, a, b, c, d, e\}.$$
- Putem descrie ce obiect se află deasupra altui obiect folosind un predicat binar *on*:
$$on = \{(e, c), (c, a), (e, d), (d, b), (a, base), (b, base)\}$$

Structură

Definiție

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

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

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

Structură

Exemplu

Lumea în care avem cutii.

- Limbajul $\mathcal{L}$
  - $\mathbf{R} = \{on\}$
  - $\mathbf{F} = \emptyset$
  - $\mathbf{C} = \emptyset$
  - $ari(on) = 2$

- O structură $\mathcal{S}$:
  - $S = \{base, a, b, c, d, e\}$
  - $\mathbf{F}^{\mathcal{S}} = \emptyset.$
  - $\mathbf{C}^{\mathcal{S}} = \emptyset.$
  - $\mathbf{R}^{\mathcal{S}} = \{on^{\mathcal{S}}\}$, unde
    $on^{\mathcal{S}} = \{(e, c), (c, a), (e, d), (d, b), (a, base), (b, base)\} \subseteq S^2.$

Interpretare

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

Definiție

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

$$I : V \to S.$$

Interpretare

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

Definiție

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

$$I : V \to S.$$

Definiție

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

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

Interpretare

Putem defini inductiv când o formulă este adevărată în $\mathcal{S}$ sub interpretarea I:

- $\mathcal{S}, I \models P(t_1, \ldots, t_n)$ dacă $P^{\mathcal{S}}(t_1^{\mathcal{S}}, \ldots, t_n^{\mathcal{S}})$
- $\mathcal{S}, I \models \neg B$ dacă $\mathcal{S}, I \not\models B$
- $\mathcal{S}, I \models B \vee C$ dacă $\mathcal{S}, I \models B$ sau $\mathcal{S}, I \models C$
- $\mathcal{S}, I \models B \wedge C$ dacă $\mathcal{S}, I \models B$ si $\mathcal{S}, I \models C$
- $\mathcal{S}, I \models B \rightarrow C$ dacă $\mathcal{S}, I \not\models B$ sau $\mathcal{S}, I \models C$
- $\mathcal{S}, I \models (\forall x)B$ dacă pentru orice interpretare $I_{x \leftarrow a}$ avem $\mathcal{S}, I_{x_i \leftarrow a} \models B$
- $\mathcal{S}, I \models (\exists x)B$ dacă există o interpretare $I_{x \leftarrow a}$ astfel încât $\mathcal{S}, I_{x_i \leftarrow a} \models B$

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

Interpretare

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

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

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

Consecință logică

Definiție

O formulă G este o consecință logică a formulelor $F_1, \ldots, F_n$, notat

$$F_1, \ldots, F_n \models G,$$

dacă pentru orice structură $\mathcal{S}$

$$\text{dacă } S \models F_1 \text{ și } \ldots \text{ și } S \models F_n, \text{ atunci } S \models G$$

Problemă semidecidabilă!

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

Logica clauzelor definite

Alegem un fragment al logicii de ordinul I astfel:

- Renunțăm la cuantificatori (dar păstrăm variabilele!)
- Renunțăm la $\neg, \vee$ (dar păstrăm $\wedge, \rightarrow$!)
- Singurele formule admise sunt de forma:
  - formule atomice: $P(t_1, \ldots, t_n)$
  - $A_1 \wedge \ldots \wedge A_n \rightarrow B$, unde toate A_i, B sunt formule atomice.

Astfel de formule se numesc clauze definite (sau clauze Horn).

Acest fragment al logicii de ordinul I se numește logica clauzelor definite (sau logica clauzelor Horn).

Programare logica

- Presupunem că putem reprezenta cunoștințele ca o mulțime de clauze definite T și suntem interesați să aflăm răspunsul la o întrebare de forma $A_1 \wedge \ldots \wedge A_n$, unde toate A_i sunt formule atomice.
- Adică vrem să aflăm dacă

$$T \models A_1 \wedge \ldots \wedge A_n$$

- Variabilele din partea stângă sunt considerate ca fiind cuantificate universal!
- Variabilele din partea dreaptă sunt considerate ca fiind cuantificate existențial!

Logica clauzelor definite

Exemplu

Fie următoarele clauze definite:

$$father(jon, ken).$$
$$father(ken, liz).$$
$$father(X, Y) \to ancestor(X, Y)$$
$$dauther(X, Y) \to ancestor(Y, X)$$
$$ancestor(X, Y) \wedge ancestor(Y, Z) \to ancestor(X, Z)$$

Putem întreba:

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

Pe săptămâna viitoare!