Limbajul logicii de ordinul I

Semantică - interpretare și denotare

$\mathcal{I} = \langle \Delta, I \rangle$

$I[P] \subseteq \underbrace{\Delta \times \ldots \times \Delta}_{n \text{ ori}}$ dacă P predicat de aritate n

Exemplu

$\Delta = \{ d_1, d_2, d_3$, ana, maria, costel, vasile, george, ion ...

$I[\text{Dog}] = \{ d_1, d_2, d_3 \}$

$I[\text{OlderThen}] = \{ (\text{ana}, \text{maria}), (\text{costel}, \text{maria}) \}$

$I[f] : \underbrace{\Delta \times \ldots \times \Delta}_{n \text{ ori}} \rightarrow \Delta$ f este funcție n-ară

$I[\text{bestFriend}] : \Delta \rightarrow \Delta$

$I[\text{bestFriend}](\text{ana}) = \text{maria}$

$I[\text{primulCopilAl}] : \Delta \times \Delta \rightarrow \Delta$

$I[\text{primulCopilAl}](\text{maria}, \text{vasile}) = \text{george}$

$I[\text{mamaTreiFrati}](\text{george}, \text{ion}, \text{ana}) = \text{maria}$

$I[a] = d_2$

Denotarea

Se aplică termenilor.

In interpretarea $\mathcal{I} = \langle \Delta, I \rangle$ și asignarea de variabile μ avem

$\| x \|_{\mathcal{I},\mu} = \mu[x]$

$\| f(t_1, \ldots, t_n) \|_{\mathcal{I},\mu} = I[f](\|t_1\|_{\mathcal{I},\mu}, \ldots, \|t_n\|_{\mathcal{I},\mu})$

$\| \text{bestFriend}(x) \|_{\mathcal{I},\mu} = I[\text{bestFriend}](\mu[x])$

dacă $\mu[x] = \text{ion}$ avem $I[\text{bestFriend}](\text{ion}$

Un termen denotat este întotdeauna un element din Δ.

<u>Componenta pragmatică</u> - specificarea modului în care sunt folosite expresiile cu sens dpv sintactic și semantic

Pentru a putea raționa, trebuie să dăm conceptelor precum "Dog" "DemocraticCountry" o interpretare dorită.

<u>Consecința logică</u>

Deși interpretarea semantică depinde de interpretarea simbolurilor nonlogice, există legături între propozițiile din FOL ce nu depind de înțelesul acestor simboluri

<u>Exemplu</u>. α, β propoziții în FOL

definim $\gamma = \neg(\beta \wedge \neg\alpha)$

Fie $\mathfrak{I}$ o interpretare în care α este adevărat. Rezultă că γ este adevărat și această valoare de adevăr nu depinde de cum înțelegem simbolurile nonlogice din α sau β.

Spunem că γ este <u>consecința logică</u> a lui α.

Fie S o mulțime de propoziții și α o propoziție. Spunem că α este o <u>consecință logică</u> a lui S (sau că S <u>implică logic</u> α) și scriem $S \models \alpha$ dnd pentru orice $\mathfrak{I}$ cu $\mathfrak{I} \models S$ avem $\mathfrak{I} \models \alpha$ ($\mathfrak{I}$ satisface α).

Sau echivalent, nu există nicio interpretare $\mathfrak{I}$ cu $\mathfrak{I} \models S \cup \{\neg\alpha\}$

O propoziție este <u>validă logic</u> $\models \alpha$ dacă este consecința logică a mulțimii vide.

Adică α validă dnd $\forall\, \mathfrak{I}$ avem $\mathfrak{I} \models \alpha$.

Dacă $S = \{\alpha_1, \ldots, \alpha_n\}$ finită atunci implicația $S \models \alpha$ se reduce la validitate

$S \models \alpha$ dnd propoziția $[(\alpha_1 \wedge \ldots \wedge \alpha_n) \supset \alpha]$ este validă

Implicația logică este cheia unui sistem bazat pe cunoștințe.

De exemplu, dacă Fido este un câine atunci sistemul ar trebui să ajungă la concluzia că Fido este un mamifer, carnivor

Dacă S implică logic o propoziție α, atunci în orice interpretare în care S este adevărat avem α adevărat.

Alte propoziții ce nu sunt o implicație a lui S, pot fi sau nu adevărate, dar un sistem bazat pe cunoștințe conclude sigur că propozițiile implicate sunt adevărate.

Dacă într-o interpretare "Dog (fido)" este adevărat, atunci sistemul sistemul poate deduce că "$\neg\neg$ Dog (fido)", "Dog(fido) $\vee$ Happy(john)" sunt adevărate. Dar aceste cunoștințe sunt deja implicite prin ceea ce cunoaștem.

Ce ne dorim este ca din "Dog (fido)" sistemul să deducă "Mammal (fido)".

Fie $\mathcal{I} = \langle D, I \rangle$ o interpretare în care

$$D = \{d\}$$
$$I[\text{Dog}] = \{d\}$$
$$I[P] = \{\ \} \quad \text{pentru orice } P \neq \text{Dog}$$
$$\text{pentru orice funcție } f.\ I[f](d, \dots, d) = d$$

În această interpretare "Dog(d)" este adevărat și "Mammal(~~fido~~ d)" este fals

Deci nu există o legătură logică între propozițiile "Dog (d)" și "Mammal(d)"

Pentru a crea această legătură trebuie să includem în setul de propoziții S o afirmație ce leagă cele două simboluri nonlogice

$$\forall x.\ \text{Dog}(x) \supset \text{Mammal}(x)$$

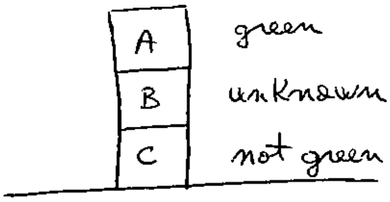
Astfel, Mammal(x) devine consecința logică a lui Dog(x).

In felul acesta, excludem interpretările ce cee în care mulțimea câinilor nu este inclusă în mulțimea mamiferelor.

Dacă continuăm să adăugăm reguli în S, excludem interpretări nedorite și în final consecința logică este un adevăr în interpretarea dorită.

Raționamentul bazat pe consecința logică permite să fie trase doar concluzii sigure, garantate logic.

Exercițiu



Există un bloc verde așezat imediat deasupra unuia care nu este verde?

Formalizare în FOL

a, b, c numele blocurilor

G - predicat unar pentru "verde"

O - predicat binar pentru "on"

Faptele din S sunt $\{ O(a,b), O(b,c), G(a), \neg G(c) \}$

α este $\exists x \exists y . G(x) \wedge \neg G(y) \wedge O(x,y)$

Vrem să demonstrăm $S \models \alpha$

Fie $\mathcal{I}$ o interpretare ce satisface S

1. dacă $\mathcal{I} \models G(b)$
   deoarece $\neg G(c)$ și $O(b,c) \in S$ $\Big\} \Rightarrow \mathcal{I} \models G(b) \wedge \neg G(c) \wedge O(b,$

$\Longrightarrow \mathcal{I} \models \exists x \exists y . G(x) \wedge \neg G(y) \wedge O(x,y)$

2. dacă $\mathcal{I} \models \neg G(b)$
   deoarece $G(a), O(a,b) \in S$ $\Big\} \Rightarrow \mathcal{I} \models G(a) \wedge \neg G(b) \wedge O(a,b) =$

$\Longrightarrow \mathcal{I} \models \exists x \exists y . G(x) \wedge \neg G(y) \wedge O(x,y)$

În ambele cazuri $\mathcal{I} \models \alpha$ deci α este o consecință logică a lui S.

În FOL problema determinării dacă o propoziție este consecința logică a altor propoziții este în general nerezolvabilă. Nu există o procedură automată care să spună dacă în toate cazurile o propoziție este consecința logică a altor propoziții.

Exprimarea cunoștințelor - crearea unei baze de cunoștințe

Primul pas în crearea unei baze de cunoștințe constă în stabilirea tipurilor de obiecte necesare, a proprietăților acestora și a relațiilor dintre ele. Acest proces s.n. ingineria cunoașterii.

Vocabularul

Se începe prin identificarea entităților esențiale din lumea agentului.

- simboluri de constante:
  - persoane maryJones
  - instituții guvern
  - locuri tomsHouse

- descrierea tipurilor de obiecte de obicei prin predicate unare:

  Person(x)
  Country(x)
  Restaurant(x)

- descrierea atributelor obiectelor Rich, Beautiful, Nice

- descrierea relațiilor DaughterOf(ana, mary)
  MarriedTo(ana, ion)

- funcțiile importante ale domeniului bestFriendOf(john)
  firstChildOf(ana, john)

Fapte de bază

Sunt reprezentate de propoziții atomice sau negații de acestora (adică $P(t_1,\dots,t_n)$ și $t_1 = t_2$)

Man(john); Rich(mary); WorksFor(john, george)

$\neg$ HappilyMarried(john)

bestFriendOf(john) = george

Fapte complexe

se folosesc conectorii logici.

$$\forall y\,[(Nice(y) \wedge Man(y)) \Rightarrow Loves(y, mary)]$$

$$\forall y\,[(Women(y) \wedge y \neq jane) \supset Loves(y, john)]$$

Putem exprima fapte generale

$$\forall x\,\forall y\,[Loves(x,y) \supset \neg Blackmails(x,y)]$$

sau cunoștinte incomplete

$$Loves(jane, john) \vee Loves(jane, jim)$$

$$\exists x\,[Adult(x) \wedge Blackmails(x, john)]$$

Relații între predicate

- dacă john este Man atunci răspunsul la întrebarea Women(john) ar trebui să fie nu.
- dacă MarriedTo(ane, john) este adevărat atunci și MarriedTo(john, ane) ar trebui să fie adevărat

Dar nimic din baza de cunoștinte nu generează astfel de inferențe de aceea este necesar să furnizăm fapte ce descriu terminologia folosită.

Disjunctivitate: presupunerea unui predicat implică negarea celuilalt

$$\forall x\,[Man(x) \supset \neg Woman(x)]$$

Subtipuri: $\forall x\,[Surgeon(x) \supset Doctor(x)]$

Exhaustivitate: două sau mai multe subtipuri explică un supertip

$$\forall x\,[Adult(x) \supset (Man(x) \vee Woman(x))]$$

Simetrie: $\forall x,y\,[MarriedTo(x,y) \supset MarriedTo(y,x)]$

Inversiune: $\forall x,y\,[ChildOf(x,y) \supset ParentOf(y,x)]$

Restricții de tip: în definirea sensului unor predicate, argumentele trebuie să fie de un anumit fel

$$\forall x,y\,[MarriedTo(x,y) \Rightarrow Person(x) \wedge Person(y)]$$

Definire completă: predicate ce sunt complet definite de o combinație logică a altor predicate

$$\forall x\,[RichMan(x) \equiv Rich(x) \wedge Man(x)]$$

<u>Implicațiile logice</u> - derivarea concluziilor implicite plecând de la cunoștințele reprezentate explicit

<u>Exemplu</u> Să presupunem că baza de cunoștințe este formată din

$$KB \begin{cases} 1.\ Rich(john) \\ 2.\ Man(john) \\ 3.\ \forall y[Rich(y) \wedge Man(y) \supset Loves(y, jane)] \\ 4.\ john = ceoOf(insuranceCompany) \\ 5.\ Company(insuranceCompany) \end{cases}$$

Întrebare: Există o companie al cărei CEO o iubește pe Jane?

În FOL $\exists x[Company(x) \wedge Loves(ceoOf(x), jane)]$?

Fie $\mathcal{I}$ o interpretare ce satisface KB. Deci $\mathcal{I}$ satisface 1,2,3

$$\left.\begin{array}{l} \Rightarrow \mathcal{I} \models Loves(john, jane) \\ \text{din 4. } \mathcal{I} \models john = ceoOf(insuranceCompany) \end{array}\right\} \Rightarrow$$

$$\left.\begin{array}{l} \mathcal{I} \models Loves(ceoOf(insuranceCompany), jane) \\ \text{din 5. } \mathcal{I} \models Company(insuranceCompany) \end{array}\right\} \Rightarrow$$

$$\mathcal{I} \models Company(insuranceCompany) \wedge Loves(ceoOf(insuranceCompany), jane)$$

$$\Rightarrow \mathcal{I} \models \exists x[Company(x) \wedge Loves(ceoOf(x), jane)]$$

$KB \models \alpha$ dnd $\forall \mathcal{I}$ cu $\mathcal{I} \models KB$ atunci $\mathcal{I} \models \alpha$

$KB \models (\alpha \supset \beta)$ dnd $\forall \mathcal{I}$ cu $\mathcal{I} \models KB$ atunci $\mathcal{I} \models \neg\alpha \vee \beta$

1. dacă $\mathcal{I} \models \neg\alpha$ atunci $\mathcal{I} \models \neg\alpha \vee \beta$

2. dacă $\mathcal{I} \models \alpha$ atunci $\mathcal{I} \models (\alpha \supset \beta)$ dnd $\mathcal{I} \models \beta$

deci $KB \models (\alpha \supset \beta)$ dnd $KB \cup \{\alpha\} \models \beta$

<u>Exemplu</u> Avem o bază de cunoștințe alcătuită din

$$KB \left[\begin{array}{l} 1.\ \exists x[Adult(x) \wedge Blackmails(x, john)] \\ 2.\ \forall x\,[Adult(x) \supset (Man(x) \vee Woman(x))] \\ 3.\ Loves(john, jane) \\ 4.\ \forall y\,[(Woman(y) \wedge y \neq jane) \supset Loves(y, john)] \\ 5.\ \forall x \forall y\,[Loves(x,y) \supset \neg Blackmails(x,y)] \end{array}\right.$$

Întrebare: Dacă niciun bărbat nu îl șantajează pe John, atunci este el șantajat de cineva pe care iubește?

în FOL: $\forall x\,[Man(x) \supset \neg Blackmails(x, john)] \supset$

$$\exists y\,[Loves(john, y) \wedge Blackmails(y, john)]\ ?$$

Fie $\mathfrak{I} \models KB$ și $\mathfrak{I} \models \forall x\,[Man(x) \supset \neg Blackmails(x, john)]$ (6)

Trebuie să arătăm că $\mathfrak{I} \models \exists y\,[Loves(john, y) \wedge Blackmails(y, john)]$

Din 1, 2, 6 $\Rightarrow$ $\mathfrak{I} \models \exists x\,[Woman(x) \wedge Blackmails(x, john)]$

Din 4, 5 $\Rightarrow$ $\mathfrak{I} \models \forall y\,[(Woman(y) \wedge y \neq jane) \supset \neg Blackmails(y, john)]$

$$\left.\begin{array}{l} \Rightarrow \mathfrak{I} \models Blackmails(jane, john) \\ 3.\ \ \mathfrak{I} \models Loves(john, jane) \end{array}\right\} \Rightarrow$$

$\Rightarrow$ $\mathfrak{I} \models Loves(john, jane) \wedge Blackmails(jane, john)$

$\Rightarrow$ $\mathfrak{I} \models \exists y\,[Loves(john, y) \wedge Blackmails(y, john)]$

Indivizi abstracți

În FOL se reprezintă fapte dintr-un domeniu. În multe situații obținem o mai mare flexibilitate în reprezentare dacă mapăm obiecte unor predicate și funcții.

Reificare înseamnă transformarea unei propoziții într-un obiect (prin crearea unor "indivizi abstracți")

De exemplu, putem exprima faptul că John cumpără o bicicletă

Purchases (john, bike)   sau

Purchases (john, bike, oct11)   sau

Purchases (john, bike, oct11, \$200) ...

Problema este că aritatea predicatului "Purchase" depinde de nivelul de detalii pe care dorim să-l exprimăm.

Vom considera "cumpărarea" ca fiind un individ abstract, numit de exemplu p17. Putem descrie acum "cumpărarea" la orice nivel de detalii folosind predicate și funcții.

$$Purchase(p17) \wedge agent(p17) = john \wedge obiect(p17) = bike \wedge$$
$$\wedge\ amount(p17) = \$200 \wedge time(p17) = \text{"16:23:01"}$$

În loc de time(p17) = "16:23:01" putem folosi

$$time(p17) = t19 \wedge hour(t19) = 16 \wedge Minute(t19) = 23 \wedge \ldots$$

Marele ~~avantaj~~ avantaj acum este că aritatea predicatelor și funcțiilor este stabilită dinainte.

Indivizii abstracți sunt utili și în reprezentarea unor cantități

ageInYears (suzy) = 14

ageInMonths (suzy) = 172

Relația dintre ageInYears și ageInMonths este aceeași cu cea dintre durationInYears și durationInMonths.

Pentru a capta aceste regularități, poate fi util să introducem un individ abstract care să reprezinte durata în timp (independent de o unitate de timp): age(suzy)

$$years(age(suzy)) = 14$$

$$months(x) = 12 * years(x)$$

Alte tipuri de fapte

1. Fapte statistice și probabiliste:

   Jumătate dintre companii sunt în vestul țării.

   Cei mai mulți dintre angajați muncesc din greu.

2. Fapte implicite și prototipice – sunt de obicei adevărate (dacă nu se specifică contrariul)

   Bicicleta are două roți.

   Păsările zboară.

   In general companiile nu permit angajaților ce lucrează împreună să se căsătorească

3. Fapte intenționale:

   John crede că Ana încearcă să-l sentajeze

   Jane nu vrea ca Jim să știe că ea îl iubește.

Aceste fapte pot fi reprezentate cu extensii ale FOL sau cu alte limbaje de reprezentare.
Alegerea limbajului de reprezentare depinde de tipurile de fapte din aplicație.

KB:

- Tony, Mike and John belong to the Alpine Club.
- Every member of the Alpine Club who is not a skier is a mountain climber.
- Mountain climbers do not like rain and anyone who does not like snow is not a skier.
- Mike dislikes whatever John likes and likes whatever Tony dislikes.
- Tony likes rain and snow.

a) Represent the above sentences in FOL, using a consistent vocabulary (which you must define)

b) Prove that the sentences logically entail that there is a member of the Alpine Club who is a mountain climber but not a skier