

## Reprezentarea orientată obiect

Reprezentările FOL, regulile în sisteme de producție sunt "plate", adică fiecare unitate de reprezentare este independentă de celelalte. Cunoștințele despre un obiect pot fi împăstrate în toate baze de cunoștințe pentru propoziții/reguli cu care nu au nicio legătură. În baze de cunoștințe mari este esențial ca propozițiile/regulile să fie organizate într-un fel.

Într-un sistem procedural cu reprezentări orientate obiect, pentru diverse tipuri de obiecte considerăm tipurile de raționamente relevante și criem proceduri pentru a le implementa.

În 1975 Marvin Minsky a propus ideea utilizării grupurilor de proceduri orientate obiect pentru a recunoaște și testa structuri noi. El a folosit termenul de "cadru" pentru structurile de date ce reprezintă aceste structuri. Cadrele pot fi utilizate în raționamente, iar pentru acest lucru este nevoie de un limbaj de reprezentare pentru a exprima structura cadrelor.

În reprezentarea bazată pe cadre, faptele sunt concentrate în jurul obiectelor. Prin obiect putem înțelege un obiect fizic concret sau un concept mai abstract, cum ar fi o clasă de obiecte sau o situație.

### Cadre individuale și generice

Cadrele individuale reprezintă obiecte, cadrele generice reprezintă categorii sau clase de obiecte.

Cadrele individuale sunt similare elementelor din memorie de lucru a sistemelor de producție.

Cadrul este o listă denumită de slot-uri în care se pun filler-e.

(Frame-name

< slot-name1 filler1 >

< slot-name2 filler2 >

...)

Numele cadrelor individuale încep cu literă mică, numele cadrelor generice încep cu majusculă.

Filler-urile sunt valori (numere, stringuri) sau nume de alte cedre individuale.

Numele slot-urilor încep cu majusculă și sunt prefixate de ":".

(tripLeg10

<: INSTANCE-OF TripLeg >

<: Destination toronto > ...)

(toronto

<: INSTANCE-OF CanadianCity >

<: Province ontario >

<: Population 4.5M > ...)

(CanadianCity

<: IS-A City >

<: Province CanadianProvince >

<: Country canada >)

Cedre individuale au un slot special numit :INSTANCE-OF al cărui filler este numele unui cedru generic ce reprezintă categoria obiectului reprezentat. Adică cedrul individual este o instanță a cedrului generic.

Cedrele generice au un slot special numit :IS-A al cărui filler este numele unui cedru generic mai general. Spunem despre cedrul generic că este o specializare a cedrului generic mai general.

Slot-urile cedrelor generice pot avea asociate proceduri

IF-ADDED și IF-NEEDED

regula  $\text{Parent}(x,y) \Leftarrow \text{Mother}(x,y)$  poate fi interpretată procedural astfel:

1. if-needed - când avem de rezolvat un scop ce unifică cu  $\text{Parent}(x,y)$ , rezolvăm  $\text{Mother}(x,y)$  (adică întoarcere înapoi). Procedurul, conexiunea dintre mamă și părinte se face când trebuie să demonstrăm ceva despre părinte.
2. if-added - când un fapt ce unifică cu  $\text{Mother}(x,y)$  este adăugat în KB, adăugăm și  $\text{Parent}(x,y)$  în KB (întoarcere înainte). Conexiunea dintre  $\text{Mother}(x,y)$  și  $\text{Parent}(x,y)$  se face când aflăm ceva nou despre relația  $\text{Mother}(x,y)$ .

(Trip

&lt;: TotalCost [IF-NEEDED ComputeCost] &gt; ...)

(Lecture

&lt;: DayOfWeek WeekDay &gt;

&lt;: Date [IF-ADDED ComputeDayOfWeek] &gt; ...)

Într-un cadru un slot poate avea atât un filler cât și o procedură asociată.

### Mostenire

Într-un sistem de cadre raționamentale implică crearea de instanțe de cadrele generice, completarea/deducerea valorilor slot-urilor.

Cadrele generice pot fi folosite pentru a completa valorile ce nu sunt menționate explicit la crearea instanței și pentru a declanșa acțiuni adiționale când filler-ele sunt cunoscute.

Dacă ne interesează o valoare dintr-o instanță, putem să o determinăm prin moștenire (cadrele "copii" moștenesc proprietăți de la "părinți")

De exemplu, dacă ne interesează slot-ul :Country din cadrul toronto îl putem afla INSTANCE-OF ce indică spre cadrul generic CanadianCity, unde găsim valoarea canade.

Instanța toronto moștenește proprietatea :Country din CanadianCity.

Dacă în toronto nu am avea un filler pentru slot-ul :Province, prin moștenire am ști că trebuie să conținem o instanță a CanadianProvince.

Dacă adăugăm cadrul generic

(City

&lt;: Population NonNegativeNumber &gt; ...)

City

CanadianCity

toronto

- dacă slot-ul :Population nu are valoare, am ști că trebuie să conținem o instanță a NonNegativeNumber

## Menținerea procedurilor eteretice

```
(ExpensiveTrip  
  <:IS-A Trip> ...)
```

```
(ExoticExpensiveTrip  
  <:IS-A ExpensiveTrip> ...)
```

Dacă creăm o instanță a Trip și vrem să găsim un filler pentru slot-ul :TotalCost în această instanță, folosim procedura eteretă IF-NEEDED. Această procedură poate fi folosită prin menținere dacă am creea o instanță a codului ExoticExpensiveTrip.

Dacă creăm o instanță a codului Lecture cu data specificată explicit

```
(lecture1  
  <:INSTANCE-OF Lecture>  
  <:Date 16NOV> ...)
```

atunci procedura eteretă IF-ADDED se execută imediat pentru a calcula ziua din săptămână pentru care se completează slot-ul :DayOfWeek. Dacă ulterior modificăm slot-ul :Date, :DayOfWeek va fi și el modificat de procedură.

Într-un sistem de cunoaștere folosim o valoare menținută doar dacă nu putem găsi un filler altfel.

Un filler dintr-un cod generic poate fi reprodus explicit în instanțe și în coduri specializate.

```
(Elephant  
  <:IS-A Mammal>  
  <:EarSize large>  
  <:Color grey> ...)
```

```
(reje  
  <:INSTANCE-OF Elephant>  
  <:EarSize small> ...)
```

```
(RoyalElephant  
  <:IS-A Elephant>  
  <:Color white> ...)
```

```
(Clyde  
  <:INSTANCE-OF RoyalElephant> ...)
```

Într-un sistem de cadre este permisă existența multiple.

Un cadru individual poate fi instanță mai multor cadre generice.

Un cadru generic poate fi specializarea mai multor cadre generice.

(AfricanElephant  
  <IS - A Elephant>  
  <IS - A AfricanAnimal> ...)

### Raționamente folosind cadre

Într-un sistem de cadre, raționamentul începe când sistemul recunoaște un obiect ce fiind o instanță a unui cadru generic și aplică procedurile declanșate de această instanță. Aceste proceduri produc date noi și schimbări ale bazei de cunoștințe. Sistemul se oprește când nu mai există proceduri aplicabile.

Avem un ciclu în 3 pași:

1. Cineva (un utilizator sau un sistem extern) declară că există un obiect, adică o instanță a unui cadru generic;
2. Orice filler care nu este dat explicit dar poate fi dedus de această instanță <sup>noastră</sup> este dedus.
3. Pentru fiecare slot care are filler, dacă există o procedură IF-ADDED ce poate fi executată, atunci ea este executată. Prin execuție pot fi completate noi slot-uri, sau pot fi instanțiate cadre noi; mergi la pasul 1.

Dacă un filler este cerut de un utilizator, de sistemul extern sau de o procedură activată, atunci:

1. dacă filler-ul are valoare, ea este returnată
2. altfel, dacă există o procedură IF-NEEDED ce poate fi executată, atunci ea este executată, calculând filler-ul dar putând completa și alte slot-uri sau instanțiate cadre noi.

Dacă pașii 1 și 2 nu produc un rezultat, atunci valoarea slot-ului se consideră necunoscută.

Obs. Existența valorilor se face în momentul creării instanței; dar procedurile IF-NEEDED sunt invocate doar la cerere.

Cedulele generice sunt create pentru tipuri de obiecte/situații majore din aplicatie. Constrangerea dintre slot-uri sunt exprimate prin proceduri atesate IF-ADDED & IF-NEEDED. Programatorul decide dacă raționamentul va fi ghidat de date sau de scopuri.

Valorile implicite sunt completate atunci când sunt disponibile.

Valorile implicite au un rol important în recunoașterea obiectelor/situațiilor — oamenii sunt inclinați să generalizeze din ceea ce au văzut înainte, și presupun că obiectele/situațiile sunt tipice decât nu observă caracteristici excepționale într-un caz individual.