

BAZE DE CUNOȘTINȚE

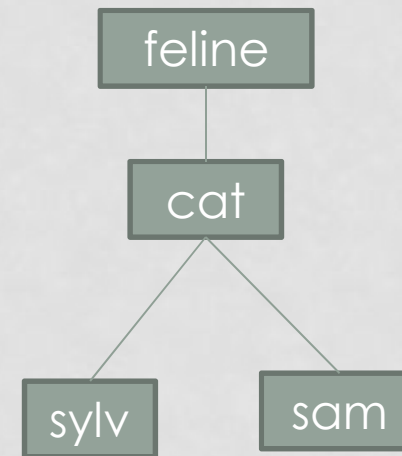
REPREZENTĂRI STRUCTURATE
BAZE DE CUNOȘTINȚE CU CADRE

M C
I O
H L
A H
E O
L N
A

BAZE DE CUNOȘTINȚE CU CADRE

STUDIU DE CAZ 1

frame feline;
 default legs **are** 4.
frame cat **is a** feline;
 default habitat **is** house
 default meal **is** kit_e_kat.
instance sylvester **is a** cat.
instance sammy **is a** cat.



frame(feline,[], [**attr**(legs,4)]).
frame(cat,[feline], [**attr**(habitat, house), **attr**(meal, kit_e_kat)]).
frame(sylvester, [cat], []).
frame(sammy, [cat], []).

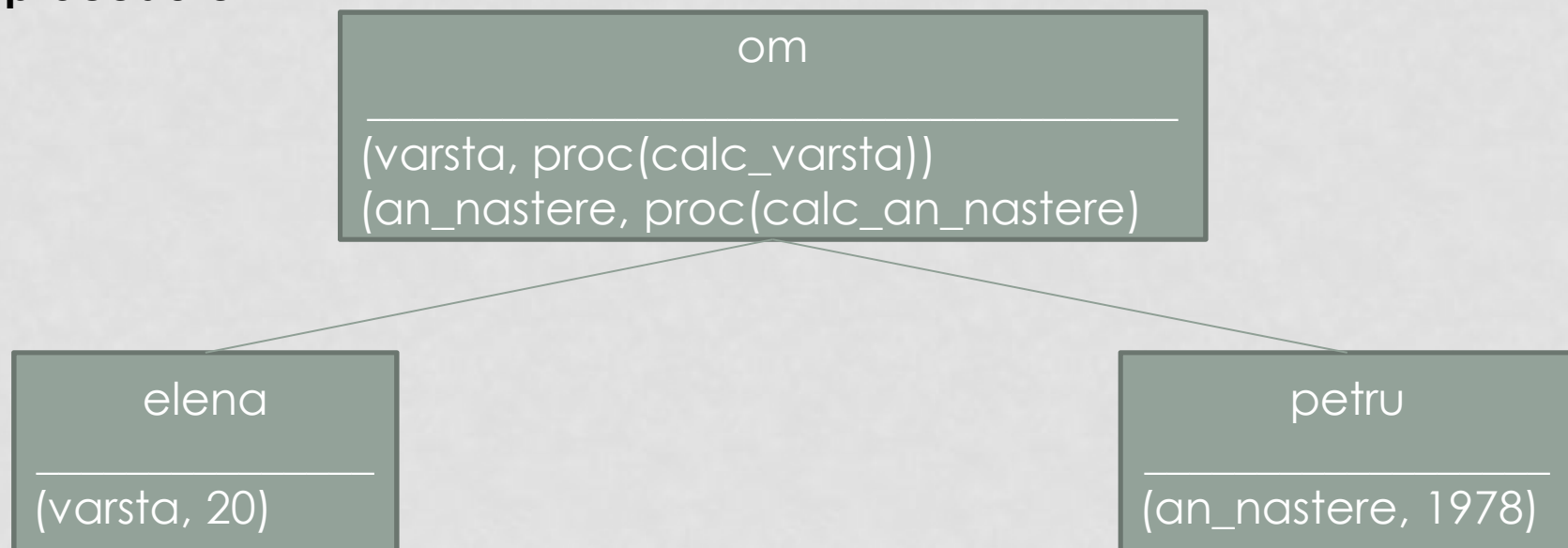
BAZE DE CUNOȘTINȚE

STUDIU DE CAZ 2

Fiecare om are o varstă și un an de naștere. Elena și Petru sunt oameni. Elena are 20 de ani și Petru e născut în 1978.

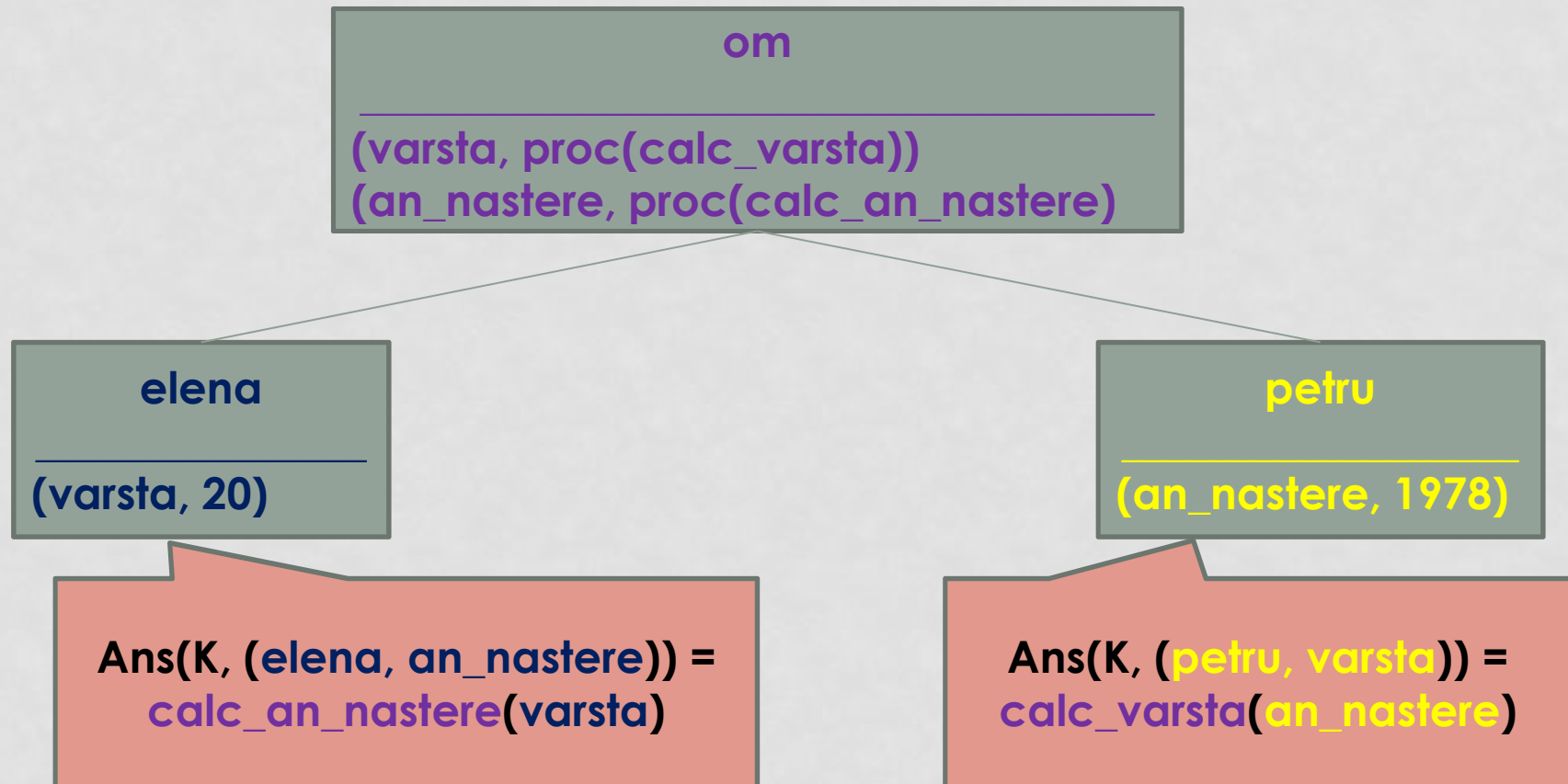
```
procedure calc_an_nastere(varsta)
    return an_curent – varsta
endprocedure
```

```
procedure calc_varsta(an_nastere)
    return an_curent – an_nastere
endprocedure
```



BAZE DE CUNOȘTINȚE

STUDIU DE CAZ 2

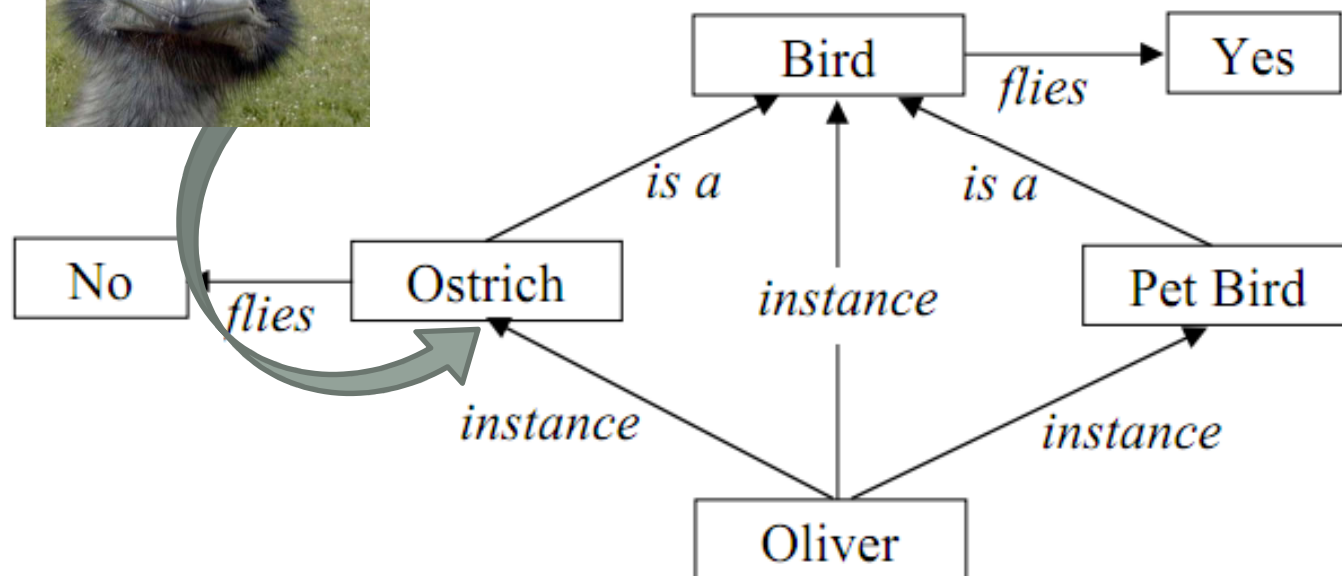


BAZE DE CUNOȘTINȚE CU CADRE

STUDIU DE CAZ 3

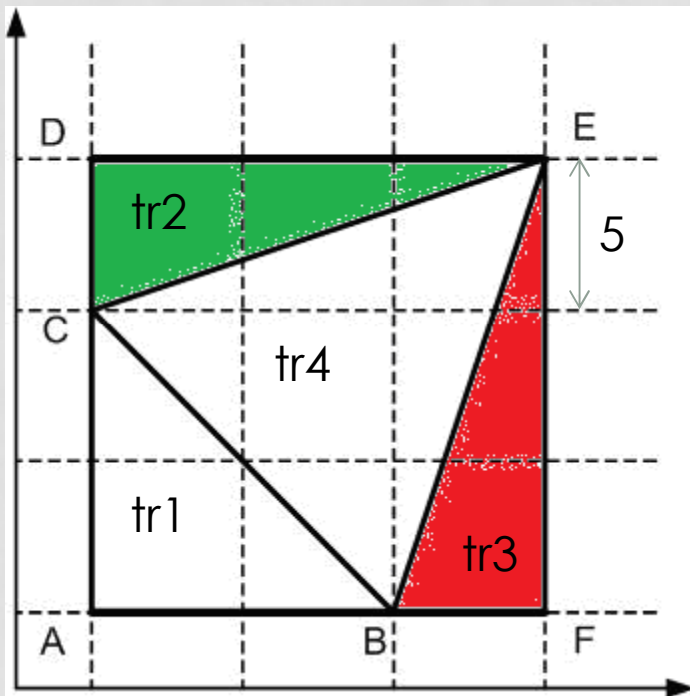


Question: "Can Oliver fly?"



BAZE DE CUNOȘTINȚE

STUDIU DE CAZ 4



triunghi

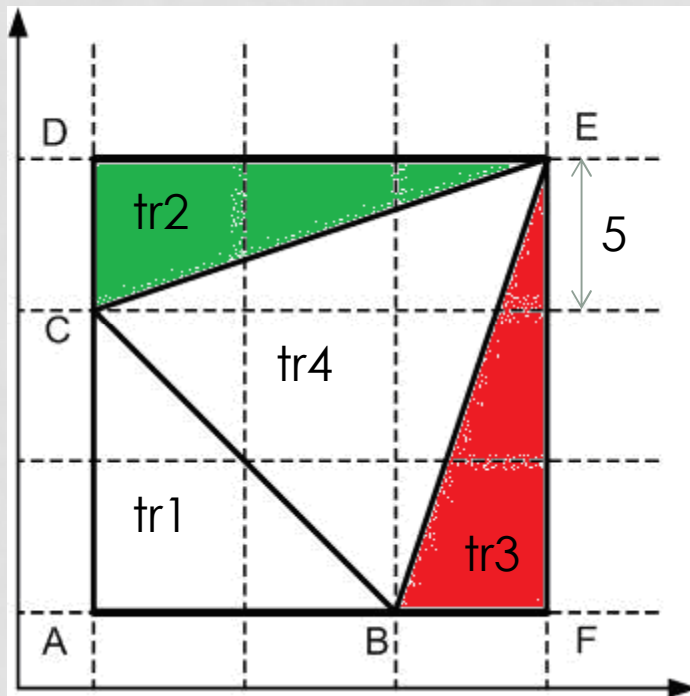
```
(arie, proc(calc_arie_tr)  
(perimetru, proc(calc_perim))  
(l1, demon)  
(l2, demon)  
(l3, demon)
```

```
procedure calc_arie_tr(perimetru, l1, l2, l3)  
    p ← perimetru/2  
    return sqrt(p(p-l1)(p-l2)(p-l3))  
endprocedure
```

```
procedure calc_perim(l1, l2, l3)  
    return (l1+l2+l3)/2  
endprocedure
```

BAZE DE CUNOȘTINȚE

STUDIU DE CAZ 4



triunghi drept

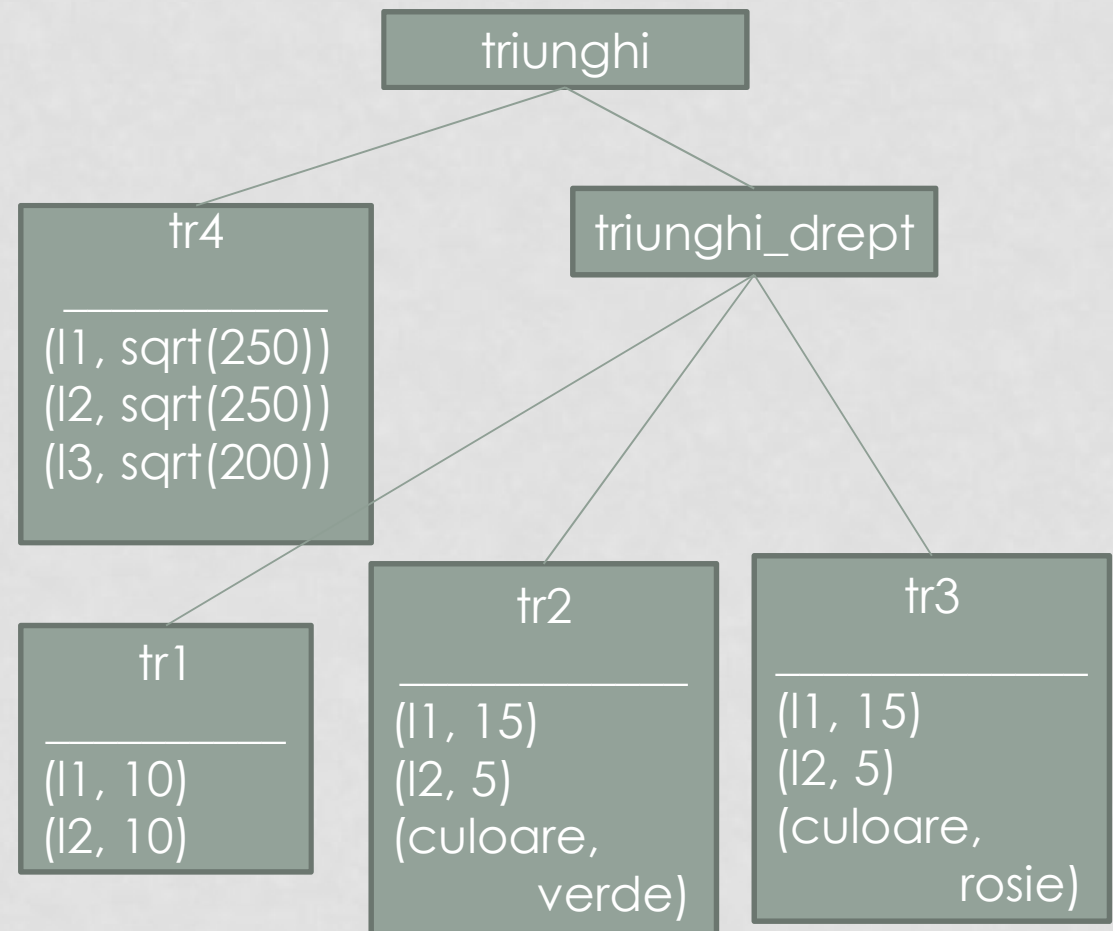
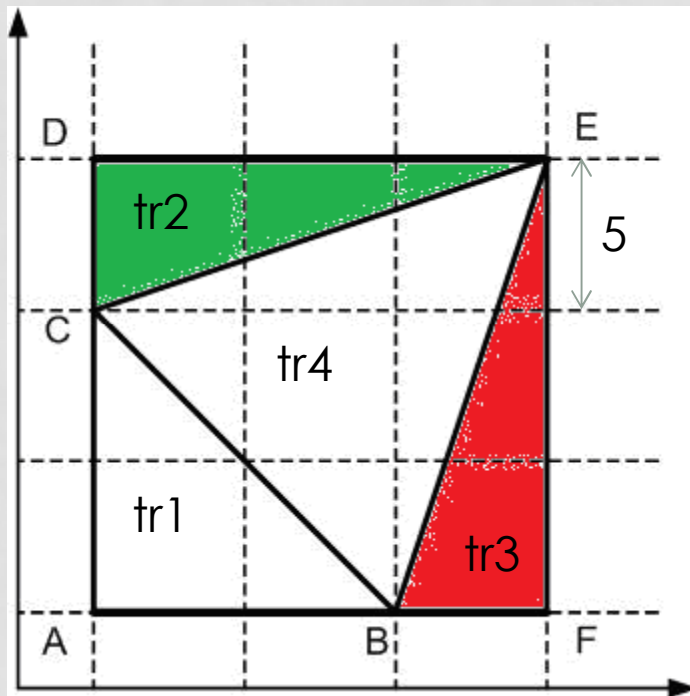
```
(arie, proc(calc_arie_tr_drept)
(l1, demon)
(l2, demon)
(l3, proc(calc_ip))
(culoare, demon)
```

```
procedure calc_arie_tr_drept(l1,l2)
    return (l1×l2)/2
endprocedure
```

```
procedure calc_ip(l1,l2)
    return sqrt(l1×l1+l2×l2)
endprocedure
```

BAZE DE CUNOȘTINȚE

STUDIU DE CAZ 4



triunghi

(arie, proc(calc_arie_tr)
(perimetru, proc(calc_perim))
(l1, demon)
(l2, demon)
(l3, demon)

triunghi drept

(arie, proc(calc_arie_tr_drept)
(l1, demon)
(l2, demon)
(l3, proc(calc_ip))
(culoare, demon)

tr4

(l1, sqrt(250))
(l2, sqrt(250))
(l3, sqrt(200))

tr1

(l1, 10)
(l2, 10)

tr2

(l1, 15)
(l2, 5)
(culoare, verde)

tr3

(l1, 15)
(l2, 5)
(culoare, rosie)

Ans(K, (tr3, perimetru)) =
calc_perim(l1, l2, l3) =
calc_perim(l1, l2,
calc_ip(l1, l2))

triunghi

(arie, proc(calc_arie_tr)
(perimetru, proc(calc_perim))
(l1, demon)
(l2, demon)
(l3, demon)

triunghi drept

(arie, proc(calc_arie_tr_drept)
(l1, demon)
(l2, demon)
(l3, proc(calc_ip))
(culoare, demon)

tr4

(l1, sqrt(250))
(l2, sqrt(250))
(l3, sqrt(200))

Ans(K, (tr1, culoare)) =
demon,

tr1

(l1, 10)
(l2, 10)

tr2

(l1, 15)
(l2, 5)
(culoare, verde)

tr3

(l1, 15)
(l2, 5)
(culoare, rosie)

Ans(K, (tr3, arie)) =
calc_arie_tr_drept(l1, l2), =
calc_arie_tr_drept(l1, l2),

A SYSTEM TO AID OBJECT-ORIENTED ANALYSIS AND PROJECT - SAAPOO

Să considerăm următoarele propoziții:

"A car is a vehicle with four steels, body and engine."

"A taxi is a car that takes passengers."

"Passengers and taxi drivers are people."

Iată reprezentările relațiilor existente între obiecte (generate de SAAPOO):

```
class(car) class(vehicle) class(steel) class(body)  
class(engine) class(taxi) class(passenger)  
class(driver) class(taxiDriver) class(people)
```

```
subclass(car,vehicle) subclass(taxi,car)  
subclass(passenger,people)  
subclass(taxiDriver,people) subclass(taxiDriver,driver)
```

A SYSTEM TO AID OBJECT-ORIENTED ANALYSIS AND PROJECT - SAAPOO

Să considerăm următoarele propoziții:

"A car is a vehicle with four steels, body and engine."

"A taxi is a car that takes passengers."

"Passengers and taxi drivers are people."

Iată reprezentările proprietăților de obiecte (generate de SAAPOO):

attribute(car,fourSteels) attribute(car,body)

attribute(car,engine) attribute(taxi,taxiDriver)

attribute(taxi,takes,passengers)

A SYSTEM TO AID OBJECT-ORIENTED ANALYSIS AND PROJECT - SAAPOO

Cum se genereaza reprezentările cu frame-uri?

- Un nume propriu este un obiect. De exemplu, substantivul „John” devine obiectul (john).
- Un substantiv singular este o clasă de obiecte. De exemplu, expresia substantivală "pisica" devine clasa (pisica).
- Dacă două expresii substantivale sunt legate de verbul "a fi", atunci între ele există o relație de generalizare. De exemplu, "Rațele sunt păsări" devine: clasa (rațe), clasa (pasări), subclasa (rațe, pasări).

A SYSTEM TO AID OBJECT-ORIENTED ANALYSIS AND PROJECT - SAAPOO

Cum se genereaza reprezentările cu frame-uri?

- Două expresii substantivale legate printr-o prepoziție reprezintă o relație de generalizare. De exemplu, expresia substantivală "masă din lemn" are următoarea reprezentare: `clasa (masă)`, `clasa (lemn)`, `clasa (masaDinLemn)`, `subclasa (masaDinLemn, masă)`, `atribut (masaDinLemn, material, lemn)`.
- O expresie substantivală cu un adjectiv determină o reprezentare clasă-subclasă. De exemplu, expresia substantivală „Măr Roșu” devine `clasa (măr)`, `clasa (mărRoșu)`, `subclasa (mărRoșu, măr)`, `atribut (mărRoșu, culoare, roșie)`.

A SYSTEM TO AID OBJECT-ORIENTED ANALYSIS AND PROJECT - SAAPOO

Cum se genereaza reprezentările cu frame-uri?

- O frază substantivală urmată de un verb intransitiv definește o proprietate. Această proprietate ar fi, eventual, puse în aplicare în continuare ca un demon în ierarhia cadru. De exemplu, notare de "pisici alerga" este: clasa (pisici), atribut (pisici, acțiune, aleargă), etc
- O frază substantivală urmată de un verb tranzitiv și o altă expresie substantivală corespunde unui atribut cu valori limitate. De exemplu "Tom mănâncă șobolani" este: obiect (Tom), clasa (șobolani), atribut (Tom, mănâncă, șobolani).

SRPC CU FRAME-URI

Limbajul de intrare

Pentru a defini funcția de intrare *Upd* e nevoie să definim limbajul de intrare L_i . În cazul SRPC-urilor care folosesc baze de cunoștințe cu cadre, actualizările sunt de forma următoare:

- (**del_slot**, frame_name, attribute_name)
- (**del_frame**, frame_name)
- (**add_slot**, frame_name, attribute_name)
- (**add_frame**, frame)
- (**modify_val_attr**, frame_name, attribute_name, value)

Vă mulțumesc!