

Laborator1

- Vom lucra în SICStus Prolog.
- Bibliografie:
 - F. Hristea, M.F. Balcan, Aspecte ale căutării și reprezentării cunoștințelor în inteligența artificială
 - I. Bratko, Prolog programming for artificial intelligence
 - A.M. Florea, Programare în Prolog pentru inteligența artificială
 - Fernando C.N. Pereira, Prolog and natural language analysis
 - Patrick Blackburn, Learn Prolog now
 - http://sicstus.sics.se/sicstus/docs/4.0.2/html/sicstus/lib_002dlists.html
 - http://sicstus.sics.se/sicstus/docs/3.7.1/html/sicstus_19.html

Nota finală pentru acest curs este formată din:

- 1/3 nota laborator
- 2/3 nota examen

Nota de la laborator este formată astfel:

- Test în ultimul laborator (valorează 75% din nota de laborator)
- Activitatea din timpul laboratorului (valorează 25% din nota de laborator)
 - Activitatea din timpul laboratorului va fi cuantificată astfel:
 - Se va face prezentă la fiecare laborator
 - Cei prezenți vor trimite la încheierea laboratorului (**nu mai târziu**) fișierul .pl cu tot ce au lucrat în timpul laboratorului
 - Dacă rămân exerciții nerezolvate în timpul laboratorului, acestea rămân ca temă pentru acasă și se poate trimite fișierul .pl în cursul săptămânii (înainte de laboratorul următor)
 - Lista de prezente va fi folosită doar pentru a verifica fișierele .pl primite de la voi.

❖ Pentru inceput...

Prolog este un limbaj de programare simbolic, potrivit pentru probleme ce implica obiecte si relatii intre obiecte.

```
femeie(alina).  
femeie(maria,ana).
```

Un program Prolog este o **baza de cunostinte** care poate fi interogata. Mai exact, un program prolog este format din *fapte* si *reguli*.

- *Faptele* sunt afirmatii despre universul problemei, afirmatii ce sunt intotdeauna adevarate:

```
zi_din_saptamana(joi): - "Joi este o zi a saptamanii".
```

- *Regulile* implementeaza conditiile ce trebuie indeplinite pentru a rezolva o problema (**a satisface un scop**). O regula are forma generala $R:-C_1, C_2, \dots, C_N$ si are semnificatia "R este adevarat daca C_1, C_2, \dots, C_N ":

```
copil(X, Y):-parinte(Y, X).
```

adica "*daca Y este parintele lui X atunci X este copilul lui Y*", **sau** "*daca X este parintele lui Y, atunci Y este copilul lui X*". Aveti deci control complet asupra ordinei argumentelor, insa aceasta ordine trebuie decisa de la inceput si trebuie pastrata in tot programul scris de voi in Prolog.

❖ Exemplu de program Prolog

% Clauze/Fapte:

```
parinte(ion,maria).
parinte(ana,maria).
parinte(ana,dan).
parinte(maria,elena).
parinte(maria,radu).
parinte(elena,nicu).
parinte(andrei,gabriela).
parinte(radu,gigi).
parinte(radu,dragos).
parinte(ion,andrei).
```

% Reguli:

```
bunic(ion):-parinte(ion , andrei),
             parinte(andrei, gabriela).
% Acestea sunt comentarii:
% “Daca ( :- ) Ion este parintele lui Andrei si ( , ) Andrei
% este parintele Gabrielei, atunci Ion este bunic”.
```

Observatii!

- fiecare instructiune Prolog se incheie cu punct.
- virgula este echivalentul conjunctiei (si).
- disjunctia (sau) este reprezentata prin “ ; ”
- “ -> “ este echivalent cu “then”
- pot fi folosite paranteze rotunde pentru a grupa blocuri de instructiuni
- variabile in Prolog incep cu litera mare, sau cu “_” (exemplu: _variabila). Variabilele precedate de caracterul underscore se numesc **variabile anonime**, deoarece pe noi nu ne intereseaza valoarea cu care sunt instantiate.
- domeniul de valabilitate al unei variabile este clauza/regula in care apare variabila respectiva.
- constantele alfanumerice (atomi) incep cu litera mica.
- raspunsurile se obtin in ordinea in care sunt puse informatiile in baza de cunostinte.

- daca interogarea contine variabile, Prologul gaseste obiectele particulare (instante) pentru care raspunsul este adevarat, unul cate unul, in ordinea in care ele apar in baza de cunostinte.

❖ **Exercitii:**

1. Intr-un fisier cu extensia .pl salvati codul dat ca exemplu mai sus. Deschideti SICStus Prolog si din **File** alegeti optiunea **Consult** si selectati fisierul vostru.

Daca nu exista erori in fisier, o sa primiti mesajul *%Consulted...* .

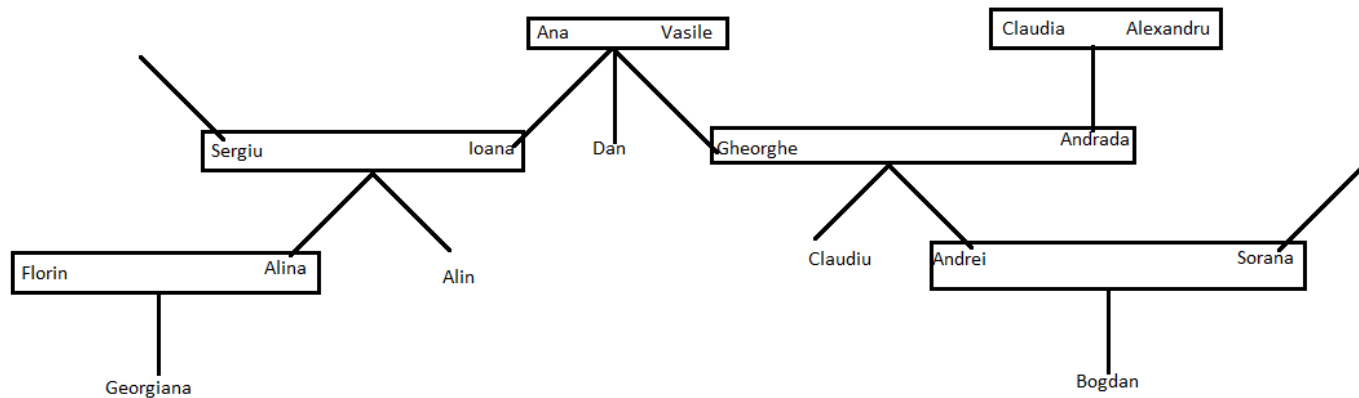
Daca exista erori in codul vostru, veti primi mesajul "Syntax error" si vi se va arata predicatul in care aveti eroarea respectiva.

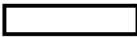
Daca modificati fisierul .pl, selectati fin **File** optiunea **Reconsult** si nu veti mai fi nevoiti astfel sa cautati fisierul vostru. Interogati predicatul bunic asa cum se cere la subpunctele de mai jos. Pentru a interoga, folositi linia de comanda a SICStus Prolog.

- a. bunic(ion)?
- b. bunic(Cine)? Interogati din nou si folositi " ; " dupa ce primiti un raspuns. Mai primiti alte raspunsuri?
- c. bunic(_Cine)?
- d. bunic.

Ce observati? Care este diferenta dintre raspunsuri pentru subpunctele a-d?

2. Scrieti relatiile de frate si sora. Interogati. Folositi " ; " pentru a obtine mai multe solutii.
3. Adaugati in fisierul vostru relatiile de rudenie prezente in figura de mai jos. Scrieti reguli care sa determine grade de rudenie precum frate, sora, verisor, verisoara, unchi, matusa, cumnat, cumnata, bunic, bunica, strabunic, strabunica. Interogati.



 Casatoriti
 Copil