

## LABORATOR 3

Se da  $S$ , o multime finita de clauze propozitionale, scrise in CNF sub forma  $[[w,s,n(p)], [a,n(w),r,t], [q]]$ . Cu  $n(p)$  s-a notat negatia lui  $p$ .

Datele se vor citi din fisier si rezultatele se vor afisa intr-un fisier.

1. Sa se implementeze procedura de rezolutie. Pentru datele de intrare  $S$ , procedura va afisa SATISFACUT, respectiv NU POATE FI SATISFACUT dupa cum este cazul.
2. Sa se implementeze procedura SAT Davis-Putnam. Pentru datele de intrare  $S$ , procedura va afisa DA, respectiv NU dupa cum  $S$  poate sau nu poate fi satisfacuta. In cazul DA, procedura va afisa si valorile de adevar asiguate literalilor (adica solutia  $\{w/\text{true}; s/\text{false}; p/\text{false} \dots\}$ ).

Ambele proceduri vor fi implementate in variantele prezentate la curs (din Ronald Brachman, Hector Levesque. Knowledge representation and reasoning, Morgan Kaufmann 2004)

Fiecare subiect va primi o nota. Pentru nota 10, implementarea se va face in PROLOG. Daca folositi alt limbaj de programare, nota maxima va fi 8.

-----

### Citire/scriere in Prolog

In fisierul de citire datele se separa cu .

```
see(user). %deschide mediul curent de citire - user reprezinta tastatura
```

```
see('c:\\prolog\\a.txt').
```

```
seen. %inchide mediul curent de citire
```

```
read(X).
```

```
read(end_of_file). %atom special ce detecteaza sfarsitul de fisier
```

```
tell('c:\\prolog\\a.txt'). %deschide mediul curent de scriere
```

```
told. %inchide mediul curent de scriere
```

```
write(parinte(ion,maria)).
```

### Predicate dinamice in Prolog

Se declara cu :-dynamic p/1. (numepredicat/aritate)

predicate pentru adaugare dinamica: asserta, assertz, assert (adaugare la inceput, la sfarsit)

predicate pentru stergere dinamica: retract, retractall.

:-dynamic fib/2.

fib(1,1).

fib(2,1).

fib(N,F):-N>2, N1 is N-1, fib(N1,F1), N2 is N-2, fib(N2,F2), F is F1+F2, asserta(fib(N,F):-!).