

## Cerinta Proiect 3

- Algoritmul de unificare -

# Algoritmul de unificare

Scriti un modul Maude in care implementati algoritmul de unificare cand semnatura este fixata tot intr-un modul Maude.

Puteti continua programul de mai jos in Maude:

```
fmod SIGNATURA is
  sorts Variabila SimbOp .
  ops  x y z  u v  w : -> Variabila .
  ops  f g h q  p : -> SimbOp .
endfm
```

```

*** Definirea problemei de unificare ca multime de ecuatii
fmod PB-UNIF is
  protecting SIGNATURA .
  sorts Termen TermenPr Ecuatie ListaEc ProblemaUn ListaArg .
  subsorts Variabila TermenPr < Termen .
  subsort Ecuatie < ListaEc .
  subsort Termen < ListaArg .
  op nilA : -> ListaArg .
  op _,_ : ListaArg ListaArg -> ListaArg [assoc id: nilA] .
  op _[_] : SimbOp ListaArg -> TermenPr .
  op so : TermenPr -> SimbOp .
  op arg : TermenPr -> ListaArg .
  var s : SimbOp . var L : ListaArg .
  eq so(s [ L ]) = s .
  eq arg( s [ L ]) = L .
  op _~_ : Termen Termen -> Ecuatie [prec 50] .
  op nilE : -> ListaEc .
  op _,_ : ListaEc ListaEc -> ListaEc [assoc comm id: nilE prec 60] .
  op {_} : ListaEc -> ProblemaUn .
endfm

```

```

*** Algoritmul de unificare
fmod ALG-UNIF is
    protecting PB-UNIF .
    var X : Variabila . var TP : TermenPr . var T : Termen .
    *** Orienteaza - regula ajutatoare
    eq TP ~ X = X ~ TP .
    *** Scoate
    eq T ~ T = nilE .
    *** DE CONTINUAT
endfm

red so( f [x , y] ) .
red arg( f [x , y] ) .
red { f [ x , y ] ~ x ; f [ nilA ] ~ f [nilA] ; y ~ z } .

```