

SISTEME EXPERT – LABORATOR 2

Aflarea maximului

```
(numar 1)
(numar 2)
(numar 3)
-----|-----
(maxim 3)
-----||-----
```

- Prima varianta

```
CLIPS> (assert (maxim -999))
CLIPS> (defrule maxim
  (numar ?x)
  ?f<-(maxim ?y&:(> ?x ?y))
  =>
  (retract ?f)
  (assert (maxim ?x))
); de la defrule
CLIPS>
```

- A doua varianta

```
CLIPS> (defrule maxim
  (numar ?x)
  (not (numar ?y&:(> ?y ?x)))
  =>
  (assert (maxim ?x))
); de la defrule
CLIPS>
```

Intersectia a doua multimi

```
(mul1 3 7 10 20)
(mul2 3 10 15 17)
-----|-----
(intersectie 3 10)
-----||-----
```

```
CLIPS> (defrule intersectie
  (mul1 $? ?x $?)
  (mul2 $? ?x $?)
  ?f<-(intersectie $?i)
  (not (intersectie $? ?x $?))
  =>
  (assert (intersectie $?i ?x))
  (retract ?f)
); de la defrule
CLIPS>
```

Reuniunea a doua multimi

```
(mul1 3 7 10 20)
(mul2 3 10 15 17)
```

```
-----|-----  
(reuniune 3 7 10 15 17 20)  
-----|-----
```

```
CLIPS> (defrule concateneaza_multimi  
        (mul1 $?x)  
        (mul2 $?y)  
        =>  
        (assert (lista $?x $?y))  
        ); de la defrule  
CLIPS>  
CLIPS> (defrule elimina_duplicate  
        ?f<-(lista $?i1 ?x $?i2 ?x $?i3)  
        =>  
        (retract ?f)  
        (assert (lista $?i1 ?x $?i2 $?i3))  
        ); de la defrule  
CLIPS>
```

Sortari

- Prima varianta

```
CLIPS> (defrule sort  
        ?f1<-(lista $?i1 ?x $?i2)  
        (not (lista $? ?y&:(< ?y ?x) $?))  
        ?f<-(sortat $?l)  
        =>  
        (retract ?f)  
        (retract ?f1)  
        (assert (lista $?i1 $?i2))  
        (assert (sortat $?l ?x))  
        ); de la defrule  
CLIPS>
```

- A doua varianta

```
CLIPS> (defrule sort  
        ?f<-(lista $?i1 ?x ?y&:(< ?y ?x) $?i2)  
        =>  
        (retract ?f)  
        (assert $?i1 ?y ?x $?i2)  
        ); de la defrule  
CLIPS>
```

Permutari

```
CLIPS> (defrule permutare  
        (lista $?i1 ?x ?y $?i2)  
        =>  
        (assert (lista $?i1 ?y ?x $?i2))  
        ); de la defrule
```

Calcularea sirului lui Fibonacci

```

CLIPS>(assert (fib 0 0))
CLIPS> (assert (fib 1 1))
CLIPS> (assert (continui 1))
CLIPS> (assert (calc_fibo 15))

CLIPS> (defrule calc_fib
      ?f<-(continui 1)
      (fib ?n ?v1)
      (fib ?m&:(= ?n (- ?m 1)) ?v2)
      =>
      (assert (fib (+ ?m 1) (+ ?v1 ?v2)))
      (retract ?f)
      (assert (continui 1))
    ); de la defrule

CLIPS>
CLIPS> (defrule oprire
      (declare (salience 5))
      (calc_fibo ?n)
      (fib ?n ?v)
      ?f<-(continui 1)
      =>
      (retract ?f)
      (assert (continui 0))
    ); de la defrule

```

CMMDC-ul a doua numere

Teorie:

$cmmdc(a,b)=cmmdc(a-b,b)$ daca $a \geq b$
 $cmmdc(a,b)=cmmdc(b,a)$

```

CLIPS> (defrule oprire
      (calc_cmmdc ?a 0)
      =>
      (assert (cmmdc ?a))
    ); de la defrule

CLIPS>(defrule scadere
      (calc_cmmdc ?a ?b&:(>= ?a ?b))
      =>
      (assert(calc_cmmdc ?b (- ?a ?b)))
    ); de la defrule

CLIPS>(defrule inversare
      (calc_cmmdc ?a ?b&:(> ?b ?a))
      =>
      (assert (calc_cmmdc ?b ?a))
    ); de la defrule

```

Sortarea crescatoare a elementelor dintr-o lista

```

CLIPS> (defrule sortare_crescat
      ?f<-(lista $?i1 ?x $?i2 ?y&:(> ?x ?y) $?i3)
      =>
      (assert (lista $?i1 ?y $?i2 ?x $?i3))
      (retract ?f)
    ); de la defrule

```