

Testarea unei relatii daca este totala

```
(defrule members
(or (R ? ?x) (R ?x ?))
=?
(assert (member ?x)))
```

```
(defrule transitivity
(R ?x ?y)
(R ?y ?z)
=>
(assert (R ?x ?z)))
```

```
(defrule not_total
(declare (salience -10))
(member ?x)
(member ?y&~?x)
(not (or (R ?x ?y) (R ?y ?x)))
=>
(printout t "Relatia nu este totala" crlf)
(halt))
```

```
(defrule total
(declare (salience -20))
=>
(printout t "Relatia este totala" crlf)
(halt))
```

```
(def facts ff
(R a b) (R c d) (R e f) (R d e) (R b c))
```

Determinare VIP

```
(defmodule MAIN (export deftemplate ?ALL))
```

```
(deftemplate knows (slot who) (slot whom))
```

```
(defrule start => (focus TRANSITIVE))
```

```
(defmodule TRANSITIVE
(import transitive-relation
(knows (who ?x) (whom ?y))
(knows (who ?y (whom ?z&~?x)))
=>
(assert (knows (who ?x) (whom ?z))))
```

```
(defrule non-reflexive-relation
?k <- (knows (who ?x) (whom ?x))
=>
```

```
(retract ?k))
```

```
(defrule most-important  
(declare (salience -10))  
=>  
(focus VIP))
```

```
(defmodule VIP  
(import MAIN deftemplate ?ALL))
```

```
(deftemplate importance (slot of) (slot is))
```

```
(defrule ini-importance  
(knows (whom ?y))  
(not (importance (of ?y)))  
=>  
(assert (importance (of ?y) (is 0))))
```

```
defrule update-importance  
?k <- (knows (whom ?x))  
?i <- (importance (of ?x) (is ?m))  
=>  
(modify ?i (is (+ 1 ?m))  
=>  
(modify ?i (is (+ 1 ?m)))  
(retract ?k))
```

```
(defrule most-important  
(declare (salience -10))  
(importance (of ?x) (is ?n))  
(not (importance (is ?m&:(> ?m ?n))))  
=>  
(printout t ?n “ “ ?x crlf))
```

```
(def facts knows  
(knows (who A) (whom B))  
(knows (who B) (whom D))  
(knows (who C) (whom E))  
(knows (who D) (whom F))  
(knows (who E) (whom G))  
(knows (who F) (whom G))  
(knows (who G) (whom F))  
(knows (who A) (whom A)))
```