```
(deffacts reglas-incio
(H)(K)
)
(defrule let-A
(A)
=>
(assert (E))
(printout t " Se activo E" crlf))

(defrule let-B
(B)
 =>
(assert (D))
(printout t "Se activo D" crlf))

(defrule let-H
(H)
=>
(assert (A))
(printout t " Se activo A" crlf))

(defrule let-EG
(E)(G)
=>
(assert (C))
(printout t " Se activo C" crlf))

(defrule let-EK
(E)(K)
=>
(assert (B))
(printout t "  Se activo B" crlf))

(defrule let-DEK
(D)(E)(K)
=>
(assert (C))
(printout t "Se activo C" crlf))

(defrule let-GKF
(G)(K)(F)
=>
(assert (A))
(printout t " Se activo A" crlf) )
```