Facts

(deftemplate parents (slot child) (slot father) (slot mother))
(deftemplate is-a (slot type) (slot name))

(assert (parents (child John) (father Top)))
(assert (parents (child John) (mother Susan)))
(assert (parents (child John) (father Tom) (mother Susan)))

(assert (is-a (name Tom) (type father)))
(assert (is-a (name Susan) (type mother)))
(assert (is-a (name John) (type son)))
(assert (is-a (name Tom) (type male)))
(assert (is-a (name Susan) (type female)))
(assert (is-a (name John) (type male)))

Writing Rules

(defrule fireA
  (or (paper)
      (wood)
      (cloth))
  =>
  (printout t "Type A fire" crlf))

(defrule fireB
  (or (oil)
      (gas)
      (greases))
  =>
  (printout t "Type B fire" crlf))

(defrule fireC
  (energised_electrical_equipment)
  =>
  (printout t "Type C fire" crlf))

(assert (paper))
Writing rules 2

(deftemplate candidate (slot name) (multislot subjects) (multislot grades))

(defrule univ_decision
  (candidate (name ?n) (subjects $?s&: (>= (length$ $?s 4))) (grades $?g)
            (grades $?g &: (> (/ (sum $?g) (length$ $?s)) 50)))
 =>
   (printout t "student: " ?n " accepted" crlf))
)

(deffunction sum (?g)
  (bind ?result 0)
  (foreach ?e ?g
    (bind ?result (+ ?result ?e)))
)

; testing
(assert (candidate (name john) (subjects a b c d) (grades 51 49 65 0)))
(assert (candidate (name Jim) (subjects a f g r) (grades 51 49 65 45)))