ΠΑΡΑΡΤΗΜΑ Ι

ΚΩΔΙΚΑ΢ ΠΡΟΓΡΑΜΜΑΤΟΣ ΗΙΡΟΦΙΛΟΣ Ι

LUTS DIAGNOSIS

;+---------------------------------------------+
;+---------------------------------------------+
;+---------------------------------------------+

defunction ask-question (?question $?allowed-values)

  (printout t ?question)

  (bind ?answer (read))

  (if (lexemep ?answer)
    then (bind ?answer (lowcase ?answer)))

  (while (not (member ?answer ?allowed-values)) do
    (printout t ?question)

    (bind ?answer (read))

    (if (lexemep ?answer)
      then (bind ?answer (lowcase ?answer))))

  ?answer)

;***********************************************************
;*****Questions Prostate Symptoms ****************************
;***********************************************************

;*****Questions Prostate Symptoms ****************************

(derule Questions_Prostate "ask-question"

  (initial-fact)

  =>

  (printout t "Question About Prostate ") t)
(bind ?pass_ure (ask-question " When you want to pass urine, is there a delay before you start(yes/no)? " yes no )

(assert (pass_ure ?pass_ure))

(bind ?flow (ask-question " When you pass urine, do you find your flow stops and starts(yes/no)? " yes no )

(assert (flow ?flow))

(bind ?trickles (ask-question " When you think you have finished passing urine, do you find a bit more trickles out, and sometimes stains your underwear (yes/no)?" yes no )

(assert (trickles ?trickles))

(bind ?thinner (ask-question " Has your stream become weaker or thinner in the past year (yes/no)?" yes no )

(assert (thinner ?thinner))

(bind ?empty (ask-question " Do you feel your bladder is not quite empty after you have been to pass urine (yes/no)?" yes no )

(assert (empty ?empty))

(bind ?get_up (ask-question " do you usually get up many times in the night to pass urine(yes/no)?" yes no )

(assert (get_up ?get_up))

(bind ?daytime (ask-question " do you go to pass urine in the more than 4 time (yes/no)?" yes no )

(assert (daytime ?daytime))

(bind ?straight (ask-question " When you want to pass urine, do you feel you have to go straight away (yes/no)?" yes no )


(assert (straight ?straight))

(bind ?mean (ask-question " Do you ever pass urine when you don't mean to (yes/no)?" yes no) )

(assert (mean ?mean)))

(defrule PASS_Urine

(pass_urine yes)

=>

(assert(A yes)))

(defrule FLOW

(flow yes)

=>

(assert(A yes)))

(defrule TRICKLES

(trickles yes)

=>

(assert(A yes)))

(defrule THINNER

(thinner yes)

=>

(assert(A yes)))

(defrule EMPTY

(empty yes)

=>

(assert(A yes)))

(defrule GET_UP

(get_up yes)
(assert(A yes)))
(defrule DAYTIME
(daytime yes)
=>
(assert(A yes)))
(defrule STRAIGHT
(straight yes)
=>
(assert(A yes)))
(defrule MEAN
(mean yes)
=>
(assert(A yes)))
(defrule a
(declare(salience 40))
(or(pass_urine yes)
(flow yes)
(trickles yes)
(thinner yes)
(empty yes)
(get_up yes)
(daytime yes)
(straight yes)
(mean yes))
(assert (A yes)))

CLIPS> (defrule b

(or(pass_urine no)
(flow no)
(trickles no)
(thinner no)
(empty no)
(get_up no)
(daytime no)
(straight no)
(mean no))

=>

(assert(B yes )))

(defrule print_a

(A yes)
(end yes)

=>

(printout t "*** you have severe symptoms and should consult your own doctor **
You may need an examination, and possibly a blood test. Your doctor may consider referring you for an operation to remove the prostate gland, or may consider putting you on a course of tablets * crlf))

;(halt))

(defrule print_b

(B yes)
(end yes)
(printout t "*** you have no difficulty passing urine, and nothing to worry about ***
 t))

(halt)

PROSTATE DISEASES DIAGNOSIS

PROSTATE FUNCTION

(deffunction readdatafacts (?a)
  (while (bind ?i (str-index "," ?a))
    (bind ?a (str-cat (sub-string 1 (- ?i 1) ?a) " " (sub-string (+ ?i 1) (str-length ?a) ?a))))
  )
  (bind ?a (explode$ ?a))
  (bind ?i (assert (luts
    (age (0 0.1) (100 0.1))
    (haematuria (1 0.5) (2 0.5))
    (haemo_spermia(1 0.5) (2 0.5))
    (painful_ejaculation (1 0.5) (2 0.5))
    (fever (1 0.5) (2 0.5))
    (chills (1 0.5) (2 0.5))
    (bone_pain (1 0.5) (2 0.5))
    (perineal_pain (1 0.5) (2 0.5))
    (big (1 0.5) (2 0.5))
  )
  )

=>

(printout t "*** you have no difficulty passing urine, and nothing to worry about ***
 t))

(halt)
(painful_DRE (1 0.5) (2 0.5))
(stony_hard_DRE (1 0.5) (2 0.5))
(pyuria (1 0.5) (2 0.5))
(haematuria (1 0.5) (2 0.5))
(PsA (0 0.5) (100 0.5))
)
)
)
/bind ?t (nth$ 16 ?a))
;if (numberp ?t) then
;if (= ?t 1) then
; (bind ?i (modify ?i (PCA yes))))
;if (= ?t 2) then
; (bind ?i (modify ?i (AP yes))))
;if (= ?t 3) then
; (bind ?i (modify ?i (BHP yes))))
)

(bind ?t (nth$ 1 ?a))
(if (numberp ?t) then
 (bind ?i (modify ?i (age (create-fuzzy-value fz_age (?t 0) (?t 1) (?t 0))))))
)
(bind ?t (nth$ 2 ?a))

(if (numberp ?t) then

  (bind ?i (modify ?i (sex (create-fuzzy-value fz_sex (?t 0) (?t ;1) (?t 0)))))
)

(bind ?t (nth$ 3 ?a))

(if (numberp ?t) then

  (bind ?i (modify ?i (haematuria (create-fuzzy-value fz_haematuria (?t 0) (?t 1) (?t 0)))))
)

(bind ?t (nth$ 5 ?a))

(if (numberp ?t) then

  (bind ?i (modify ?i (fever (create-fuzzy-value fz_fever (?t 0) (?t 1) (?t 0)))))
)

(bind ?t (nth$ 6 ?a))

(if (numberp ?t) then

  (bind ?i (modify ?i (chills (create-fuzzy-value fZ_chills (?t 0) (?t 1) (?t 0)))))
)
(bind ?t (nth$ 7 ?a))

(if (numberp ?t) then

   (bind ?i (modify ?i (bone_pain (create-fuzzy-value fz_bone_pain (?t 0) (?t 1) (?t 0))))))
)

(bind ?t (nth$ 8 ?a))

(if (numberp ?t) then

   (bind ?i (modify ?i (perineal_pain (create-fuzzy-value fz_perineal_pain (?t 0) (?t 1) (?t 0))))))
)

; (bind ?t (nth$ 9 ?a))

; (if (numberp ?t) then

;   (bind ?i (modify ?i (consistency_DRE (create-fuzzy-value fz_consistency_DRE (?t 0) (?t 1) (?t 0))))))

; )

;

(bind ?t (nth$ 9 ?a))

(if (numberp ?t) then

   (bind ?i (modify ?i (big (create-fuzzy-value fz_big (?t 0) (?t 1) (?t 0))))))
)

(bind ?t (nth$ 10 ?a))

(if (numberp ?t) then

   (bind ?i (modify ?i (painful_DRE (create-fuzzy-value fz_painful_DRE (?t 0) (?t 1) (?t 0))))))
(bind ?t (nth$ 11 ?a))

(if (numberp ?t) then
    (bind ?i (modify ?i (stony_hard_DRE (create-fuzzy-value fz_stony_hard_DRE (?t 0) (?t 1) (?t 0))))))

(bind ?t (nth$ 12 ?a))

(if (numberp ?t) then
    (bind ?i (modify ?i (fz_pyuria (create-fuzzy-value fz_pyuria (?t 0) (?t 1) (?t 0))))))

(bind ?t (nth$ 13 ?a))

(if (numberp ?t) then
    (bind ?i (modify ?i (haematuria1 (create-fuzzy-value fz_haematuria1 (?t 0) (?t 1) (?t 0))))))

; (bind ?t (nth$ 14 ?a))
; (if (numberp ?t) then
;    (bind ?i (modify ?i (DISU (create-fuzzy-value fz_DISU (?t 0) (?t 1) (?t 0))))))
; )

; (bind ?t (nth$ 15 ?a))
; (if (numberp ?t) then
; (bind ?i (modify ?i (ATFOU (create-fuzzy-value ;fz_ATFOU (?t 0) (?t 1) (?t 0)))) )
;

; (bind ?t (nth$ 16 ?a))
; (if (numberp ?t) then
; (bind ?i (modify ?i (BDU (create-fuzzy-value fz_BDU (?t 0) ;(?t 1) (?t 0))))))
;

; (bind ?t (nth$ 17 ?a))
; (if (numberp ?t) then
; (bind ?i (modify ?i (FOIBE (create-fuzzy-value fz_FOIBE ;(?t 0) (?t 1) (?t 0))))))
)

; (bind ?t (nth$ 18 ?a))
; (if (numberp ?t) then
; (bind ?i (modify ?i (UR (create-fuzzy-value fz_UR (?t 0) (?t ;1) (?t 0))))))
;

; (bind ?t (nth$ 19 ?a))
; (if (numberp ?t) then
; (bind ?i (modify ?i (PDTU (create-fuzzy-value fz_PDTU ;(?t 0) (?t 1) (?t 0))))))
;

; (bind ?t (nth$ 20 ?a))
; (if (numberp ?t) then
; (bind ?i (modify ?i (DTRBOU (create-fuzzy-value ;fz_DTRBOU (?t 0) (?t 1) (?t 0))))))
(bind ?t (nth$ 14 ?a))

(if (numberp ?t) then
    (bind ?i (modify ?i (PsA (create-fuzzy-value fz-PsA (?t 0) (?t 1) (?t 0)))))
)

(assert (PCA yes) CF 0.01)
(assert (AP yes) CF 0.01)
(assert (BPH yes) CF 0.01)

)

**PROSTATE TEMPLATE**

(deftemplate fz_haematuria

  1 2
  ( (yes (1 1) (1.01 0))
    (no (1.99 0) (2 1))
  )

)

(deftemplate fz_haematuria1

  1 2
  ( (yes (1 1) (1.01 0))
    (no (1.99 0) (2 1))
  )

)
(deftemplate fz_haemo_spermia

1 2

( (yes (1 1) (1.01 0))

(no (1.99 0) (2 1))

)

)

(deftemplate fz_painful_ejaculation

1 2

( (yes (1 1) (1.01 0))

(no (1.99 0) (2 1))

)

)

(deftemplate fz_fever

1 2

( (yes (1 1) (1.01 0))

(no (1.99 0) (2 1))

)
(deftemplate fZ_chills

1 2

( (yes (1 1) (1.01 0))
 (no (1.99 0) (2 1))
 )

)

(deftemplate fz_bone_pain

1 2

( (yes (1 1) (1.01 0))
 (no (1.99 0) (2 1))
 )

)

(deftemplate fz_perineal_pain

1 2

( (yes (1 1) (1.01 0))
 (no (1.99 0) (2 1))
 )

)

(deftemplate fz_big
1 2
( (yes (1 1) (1.01 0))
  (no (1.99 0) (2 1))
)

(deftemplate fz_stony_hard_DRE

1 2
( (yes (1 1) (1.01 0))
  (no (1.99 0) (2 1))
)

(deftemplate fz_painful_DRE

1 2
( (yes (1 1) (1.01 0))
  (no (1.99 0) (2 1))
)

(deftemplate fz_pyuria

1 2
( (yes (1 1) (1.01 0))
(deftemplate PSA

(slot value (type SYMBOL) (allowed-symbols no yes))

)

(deftemplate BHP

0 1

(high (.1 0) (.2 .2))

(low (.1 .1) (.2 0))

)

)

(deftemplate AP

1 2

(yes (1 1) (1.01 0))
(no (1.99 0) (2 1))
)

(deftemplate CP

(slot value (type SYMBOL) (allowed-symbols no yes))
)

(deftemplate PcA

(slot value (type SYMBOL) (allowed-symbols no yes))
)

;(defglobal ?*score* = 0)

;)

(deftemplate fz_age

  0 100 years

  ( (young (0 1) (30 1) (40 0))

  (old (35 0) (45 1))

  )
(deftemplate fz_yesno ; yes/no
  1 2
  ( (yes (1 1) (1.01 0))
   (no (1.99 0) (2 1))
  )
)

(deftemplate fz-PsA
  0 100 mg/dl
  ( (normal (0 0) (12 1) (15 1) (19 0))
   (medium (1.99 0) (2 1))
   (high (20 0) (26 1))
   (very_high (50 0) (65 1))
  )
)

(deftemplate luts

  (slot age (type FUZZY-VALUE fz_age))
  (slot haematuria (type FUZZY-VALUE fz_haematuria))
  (slot haemo_spermia (type FUZZY-VALUE fz_haemo_spermia))
  (slot painful_ejaculation (type FUZZY-VALUE fz_painful_ejaculation))
)
(slot fever (type FUZZY-VALUE fz_fever))

(slot chills (type FUZZY-VALUE fz_chills))

(slot bone_pain (type FUZZY-VALUE fz_bone_pain))

(slot perineal_pain (type FUZZY-VALUE fz_perineal_pain))

(slot big (type FUZZY-VALUE fz_big))

(slot painful_DRE (type FUZZY-VALUE fz_painful_DRE))

(slot stony_hard_DRE (type FUZZY-VALUE fz_stony_hard_DRE))

(slot pyuria (type FUZZY-VALUE fz_pyuria))

(slot haematuria1 (type FUZZY-VALUE fz_haematuria1))

(slot PsA (type FUZZY-VALUE fz-PsA))

)

PROSTATE RULES

;*********************************************************
; Protective rules
;*********************************************************

(defrule age;
 (declare (CF 0.2))
 (luts (age young))
 =>
 (assert (PCA yes))
)

(defrule lut_haemo
(declare (CF 0.1))

(luts (haemo_spermia yes))

=>

(assert (AP yes))

)

(defrule luts_spermia

(declare (CF 0.3))

(luts (haematuria yes) (haemo_spermia yes))

=>

(assert (PCA yes))

)

(defrule femal_egender

(declare (CF 0))

(luts (sex female))

=>

(assert (PCA yes))

)

(defrule luts_haem_o

(declare (CF 0.2))

(luts (haematuria yes))

=>

(assert (BHP yes))
(defrule lut_pain_der
  (declare (CF 0.9))
  (luts (painful_DRE yes))
  =>
  (assert (PCA yes))
)

(defrule luts_haem_pain
  (declare (CF 0.9))
  (luts (haematuria yes) (painful_DRE yes))
  =>
  (assert (PCA yes))
)

; Bad prognosis rules

(defrule luts_sp_pain
  (declare (CF 0.35))
  (luts (haemo_spermia yes)(painful_DRE yes))
(assert (PCA yes))
)

(defrule luts_ha_sp_pain
  (declare (CF 0.5))
  (luts (haematuria yes) (haemo_spermia yes) (painful_DRE yes))
  =>
  (assert (PCA yes))
)

(defrule luts_fever
  (declare (CF 0.5))
  (luts (fever yes))
  =>
  (assert (PCA yes))
)

(defrule luts_fe_ha
  (declare (CF 0.4))
  (luts (haematuria yes) (fever yes))
  =>
  (assert (PCA yes))
)
(defrule luts_hesp_fever
  (declare (CF 0.7))
  (luts (haemo_spermia yes) (fever yes))
  =>
  (assert (PCA yes))
)

(defrule luts_h_f_hesp
  (declare (CF 0.7))
  (luts (haematuria yes) (fever yes) (haemo_spermia yes))
  =>
  (assert (PCA yes))
)

(defrule luts_pain_der_fe
  (declare (CF 0.7))
  (luts (painful_DRE yes) (fever yes))
  =>
  (assert (PCA yes))
)

(defrule lut_f_pa_ha
  (declare (CF 0.9))
  (luts (haematuria yes) (painful_DRE yes) (fever yes))
  =>
(assert (PCA yes))
)

(defrule luts_hsp_pain_fe
    (declare (CF 0.9))
    (luts (haemo_spermia yes) (painful_DRE yes) (fever yes))
    =>
    (assert (PCA yes))
)

(defrule luts_hae_sp_paired_f
    (declare (CF 0.9))
    (luts (haematuria yes) (haemo_spermia yes) (painful_DRE yes) (fever yes))
    =>
    (assert (PCA yes))
)

(defrule luts_fever_ch
    (declare (CF 0.1))
    (luts (fever yes) (chills yes))
    =>
    (assert (AP yes))
)
(defrule luts_fe_sc_hae
  (declare (CF 0.1))
  (luts (haematuria yes) (fever yes) (chills yes))
  =>
  (assert (AP yes))
)

(defrule luts_bone
  (declare (CF 0.7))
  (luts(bone_pain yes))
  =>
  (assert (PCA yes))
)

(defrule luts_hea_bone
  (declare (CF 0.2))
  (luts(haematuria yes)(bone_pain yes))
  =>
  (assert (PCA yes))
)

(defrule hluts_sp_bone
  (declare (CF 0.5))
  (luts (haemo_spermia yes) (bone_pain yes))
(defrule luts_sp_hea_bone
  (declare (CF 0.8))
  (luts (haematuria yes) (haemo_spermia yes) (bone_pain yes))
  =>
  (assert (PCA yes))
)

(defrule luts_pain_bone
  (declare (CF 0.8))
  (luts (painful_DRE yes) (bone_pain yes))
  =>
  (assert (PCA yes))
)

(defrule luts_hae_pain_bone_der
  (declare (CF 0.8))
  (luts (haematuria yes) (painful_DRE yes) (bone_pain yes))
  =>
  (assert (PCA yes))
)
(defrule luts_hesp_pain_bone
  (declare (CF 0.8))
  (luts (haemo_spermia yes) (painful_DRE yes) (bone_pain yes)) =>
  (assert (PCA yes)))

(defrule luts_hae_sp_pain
  (luts (haematuria yes) (haemo_spermia yes) (painful_DRE yes) (bone_pain yes)) =>
  (assert (PCA yes)))

(defrule luts_fev_bone
  (declare (CF 0.9))
  (luts (fever yes) (bone_pain yes)) =>
  (assert (PCA yes)))

(defrule luts_hae_bone_fever
  (declare (CF 0.5))
  (luts (haematuria yes) (bone_pain yes) (fever yes)) =>
  (assert (PCA yes)))
(defrule luts_hsp_fever_bone

(declare (CF 0.7))

(luts (haemo_spermia yes)(fever yes) (bone_pain yes))

=>

(assert (PCA yes))

)

(defrule patientgoingtodie

(declare (salience -10))

?f <- (PCA yes)

?g <- (AP yes)

?e <- (BHP yes)

=>

(bind ?*out* (- 2 (- (get-cf ?f) (get-cf ?g) (get-cf ?e))))

)
WEKA RULES

;******************************************************************************
; Bad prognosis rules
;******************************************************************************

(defglobal ?*out* = 0)

(defrule R_1
  (declare (CF 0.9))
  (fever yes)
  =>
  (assert (PROSTATITIS yes))
)

(defrule R_2
  (declare (CF 0.2))
  (and (fever no)(GDRENLARGED yes))
  =>
  (assert (BPH yes))
)

(defrule R_3
  (declare (CF 0.3))
  (and(fever no)(GDRENLARGED no)(PSA normal))
)
(defrule R_4
  (declare (CF 0.1))
  (and (fever no) (GDRENLARGED no) (PSA high))
  =>
  (assert (PCA yes))
)

(defrule R_5
  (declare (CF 0.2))
  (and (fever no) (GDRENLARGED no) (PSA medium) (age middle))
  =>
  (assert (BPH yes))
)

(defrule R_6
  (declare (CF 0.2))
  (and (fever no) (GDRENLARGED no) (PSA medium) (age old))
  =>
  (assert (BPH yes))
)
(deftemplate PROSTATITIS
  0 1
  ( (yes (1 1) (1.01 0))
   (no (1.99 0) (2 1)))
)

(deftemplate BHP
  0 1
  ( (yes (1 1) (1.01 0))
   (no (1.99 0) (2 1)))
)

(deftemplate PCA
  0 1
  ( (yes (1 1) (1.01 0))
   (no (1.99 0) (2 1)))
)
(deftemplate GDRENLARGED

0 1
( (yes (1 1) (1.01 0))
  (no (1.99 0) (2 1))
)

)

(deftemplate fever

0 1
( (yes (1 1) (1.01 0))
  (no (1.99 0) (2 1))
)

)

(deftemplate age

0 100 years
( (middle (30 1) (85 0));<<=79
(deftemplate fz_yesno ; yes/no
0 1
( (yes (1 1) (1.01 0))
 (no (1.99 0) (2 1)))
)
)

(deftemplate PSA
0 100 mg/dl
( (normal (0 1) (3 1) (5 0));<=4
 (medium (4 0) (7 1) (12 0));>4 and <10
 (high (7 0) (10 1) (15 1));>=10
)
)

(deffunction readdatafacts (?a)

 (while (bind ?i (str-index "," ?a))

 (bind ?a (str-cat (sub-string 1 (- ?i 1) ?a) " " (sub-string (+ ?i 1) (str-length ?a) ?a)))))
)

(bind ?a (explode$ ?a))

(bind ?i (assert(BHP yes))
  (assert(GDRENLAGED yes))
  (assert(fever yes))
  (assert(age middle))
  (assert(PSA normal))
)

(bind ?t (nth$ 6 ?a))
(if (numberp ?t) then
  (bind ?i (modify ?i (fever yes)))
)

(bind ?t (nth$ 10 ?a))
(if (numberp ?t) then
  (bind ?i (modify ?i (GDRENLAGED yes)))
)
(bind ?t (nth$ 1 ?a))

(if (numberp ?t) then
    (bind ?i (modify ?i (age middle)))
)

(bind ?t (nth$ 15 ?a))

(if (numberp ?t) then
    (bind ?i (modify ?i (PSA normal)))
)

(assert (PCA yes) CF 0.01)

(assert (PROSTATITIS yes) CF 0.01)

(assert (BPH yes) CF 0.01)

)
ONLINE_REFRESH TIME = 55;
ONLINE_TIMEOUT COUNT = 1100;
ONLINE_CODE = OFF;
ONLINE_TRACE BUFFER = (OFF, PAR(0));
COMMENTS = ON;
FTL BUFFER = (OFF, PAR(1));
PASSWORD = OFF;
PUBLIC_IO = ON;
FAST_CMBF = OFF;
FAST_COA = ON;
BTYPE = DOUBLE;
C_TYPE = ANSI;
} /* SHELLOPTIONS */

MODEL {

VARIABLE_SECTION {

LVAR {

NAME = AGE;

BASEVAR = Units;

LVRANGE = MIN(37.0), MAX(90.0), MINDEF(0), MAXDEF(65535),

DEFAULT_OUTPUT(63.5);

RESOLUTION = XGRID(5.3), YGRID(0.1), SHOWGRID (ON), SNAPTOGRID(OFF);

COLOR = RED (255), GREEN (0), BLUE (0);

INPUT = CMBF;

} /* SHELLOPTIONS */

MODEL {

VARIABLE_SECTION {

LVAR {

NAME = AGE;

BASEVAR = Units;

LVRANGE = MIN(37.0), MAX(90.0), MINDEF(0), MAXDEF(65535),

DEFAULT_OUTPUT(63.5);

RESOLUTION = XGRID(5.3), YGRID(0.1), SHOWGRID (ON), SNAPTOGRID(OFF);

COLOR = RED (255), GREEN (0), BLUE (0);

INPUT = CMBF;

}}
POS  = -273, -246;

TERM {
    TERMNAME = low;
    POINTS   = (37.0, 1.0),
               (50.25, 1.0),
               (63.5, 0.0),
               (90.0, 0.0);
    SHAPE   = LINEAR;
    COLOR   = RED (255), GREEN (0), BLUE (0);
}

TERM {
    TERMNAME = medium;
    POINTS   = (37.0, 0.0),
               (50.25, 0.0),
               (63.5, 1.0),
               (76.75, 0.0),
               (90.0, 0.0);
    SHAPE   = LINEAR;
    COLOR   = RED (0), GREEN (128), BLUE (0);
}

TERM {
    TERMNAME = high;
    POINTS   = (37.0, 0.0),
               (63.5, 0.0),
               (76.75, 1.0),
               (90.0, 0.0);
    SHAPE   = LINEAR;
    COLOR   = RED (0), GREEN (0), BLUE (128);
}
SHAPE = LINEAR;
COLOR = RED (0), GREEN (0), BLUE (255);
}
} /* LVAR */

LVAR {
NAME = BONE PAIN;
BASEVAR = Units;
LV RANGE = MIN(0.0), MAX(1.0),
          MINDEF(0), MAXDEF(65535),
          DEFAULT_OUTPUT(0.5);
RESOLUTION = XGRID(0.1), YGRID(0.1),
             SHOWGRID (ON), SNAPTOGRID(OFF);
COLOR = RED (128), GREEN (255), BLUE (0);
INPUT = CMBF;
POS = -279, -68;
TERM {
   TERMNAME = false;
   POINTS = (0.0, 1.0),
            (0.33334, 1.0),
            (0.66666, 0.0),
            (1.0, 0.0);
   SHAPE = LINEAR;
   COLOR = RED (255), GREEN (0), BLUE (0);
}
}
TERM {
    TERMNAME = true;
    POINTS   = (0.0, 0.0),
              (0.33334, 0.0),
              (0.66666, 1.0),
              (1.0, 1.0);
    SHAPE    = LINEAR;
    COLOR    = RED (0), GREEN (128), BLUE (0);
}
}
} /* LVAR */
LVAR {
    NAME       = CHILLS;
    BASEVAR    = Units;
    LVRANGE    = MIN(0.0), MAX(1.0),
                 MINDEF(0), MAXDEF(65535),
                 DEFAULT_OUTPUT(0.5);
    RESOLUTION = XGRID(0.1), YGRID(0.1),
                 SHOWGRID (ON), SNAPTOGRID(OFF);
    COLOR      = RED (255), GREEN (0), BLUE (255);
    INPUT      = CMBF;
    POS        = -276, -97;
TERM {
    TERMNAME = false;
    POINTS   = (0.0, 1.0),
              (0.33334, 1.0),

(0.66666, 0.0),
(1.0, 0.0);
SHAPE    = LINEAR;
COLOR    = RED (255), GREEN (0), BLUE (0);
}
TERM {
    TERMNAME = true;
    POINTS   = (0.0, 0.0),
               (0.33334, 0.0),
               (0.66666, 1.0),
               (1.0, 1.0);
    SHAPE    = LINEAR;
    COLOR    = RED (0), GREEN (128), BLUE (0);
}
} /* LVAR */
LVAR {
    NAME       = DRE: ENLARGED;
    BASEVAR    = Units;
    LVRANGE    = MIN(0.0), MAX(1.0),
                MINDEF(0), MAXDEF(65535),
                DEFAULT_OUTPUT(0.5);
    RESOLUTION = XGRID(0.1), YGRID(0.1),
                 SHOWGRID (ON), SNAPTOGRID(OFF);
    COLOR      = RED (128), GREEN (128), BLUE (128);
    INPUT      = CMBF;
POS = -282, -6;

TERM {
    TERMNAME = false;
    POINTS = (0.0, 1.0),
             (0.33334, 1.0),
             (0.66666, 0.0),
             (1.0, 0.0);
    SHAPE = LINEAR;
    COLOR = RED (255), GREEN (0), BLUE (0);
}

TERM {
    TERMNAME = true;
    POINTS = (0.0, 0.0),
             (0.33334, 0.0),
             (0.66666, 1.0),
             (1.0, 1.0);
    SHAPE = LINEAR;
    COLOR = RED (0), GREEN (128), BLUE (0);
}

} /* LVAR */

LVAR {
    NAME = DRE: PAINFUL;
    BASEVAR = Units;
    LVRANGE = MIN(0.0), MAX(1.0),
              MINDEF(0), MAXDEF(65535),
}
DEFAULT_OUTPUT(0.5);

RESOLUTION = XGRID(0.1), YGRID(0.1),

SHOWGRID (ON), SNAPTOGRID(OFF);

COLOR = RED (0), GREEN (255), BLUE (255);

INPUT = CMBF;

POS = -283, 21;

TERM {
  TERMNAME = false;
  POINTS = (0.0, 1.0),
          (0.33334, 1.0),
          (0.66666, 0.0),
          (1.0, 0.0);
  SHAPE = LINEAR;
  COLOR = RED (255), GREEN (0), BLUE (0);
}

TERM {
  TERMNAME = true;
  POINTS = (0.0, 0.0),
          (0.33334, 0.0),
          (0.66666, 1.0),
          (1.0, 1.0);
  SHAPE = LINEAR;
  COLOR = RED (0), GREEN (128), BLUE (0);
}

} /* LVAR */
LVAR {
    NAME       = DRE: STONY/HARD;
    BASEVAR    = Units;
    LVRANGE    = MIN(0.0), MAX(1.0),
                  MINDEF(0), MAXDEF(65535),
                  DEFAULT_OUTPUT(0.5);
    RESOLUTION = XGRID(0.1), YGRID(0.1),
                  SHOWGRID (ON), SNAPTOGRID(OFF);
    COLOR      = RED (255), GREEN (255), BLUE (0);
    INPUT      = CMBF;
    POS        = -279, 45;

    TERM {
        TERMNAME = false;
        POINTS   = (0.0, 1.0),
                  (0.33334, 1.0),
                  (0.66666, 0.0),
                  (1.0, 0.0);
        SHAPE    = LINEAR;
        COLOR    = RED (255), GREEN (0), BLUE (0);
    }

    TERM {
        TERMNAME = true;
        POINTS   = (0.0, 0.0),
                  (0.33334, 0.0),
                  (0.66666, 1.0),
(1.0, 1.0);
SHAPE = LINEAR;
COLOR = RED (0), GREEN (128), BLUE (0);
}
} /* LVAR */
LVAR {
    NAME = FEVER;
    BASEVAR = Units;
    LV RANGE = MIN(0.0), MAX(1.0),
               MINDEF(0), MAXDEF(65535),
               DEFAULT_OUTPUT(0.5);
    RESOLUTION = XGRID(0.1), YGRID(0.1),
                 SHOWGRID (ON), SNAPTOGRID(OFF);
    COLOR = RED (255), GREEN (0), BLUE (128);
    INPUT = CMBF;
    POS = -280, -125;
    TERM {
        TERMNAME = false;
        POINTS = (0.0, 1.0),
                 (0.33334, 1.0),
                 (0.66666, 0.0),
                 (1.0, 0.0);
        SHAPE = LINEAR;
        COLOR = RED (255), GREEN (0), BLUE (0);
    }
}
TERM {
    TERMNAME = true;
    POINTS   = (0.0, 0.0),
              (0.33334, 0.0),
              (0.66666, 1.0),
              (1.0, 1.0);
    SHAPE    = LINEAR;
    COLOR    = RED (0), GREEN (128), BLUE (0);
}
} /* LVAR */
LVAR {
    NAME       = GHAEMATURIA;
    BASEVAR    = Units;
    LV RANGE   = MIN(0.0), MAX(1.0),
                 MINDEF(0), MAXDEF(65535),
                 DEFAULT_OUTPUT(0.5);
    RESOLUTION = XGRID(0.1), YGRID(0.1),
                 SHOWGRID (ON), SNAPTOGRID(OFF);
    COLOR      = RED (128), GREEN (0), BLUE (0);
    INPUT      = CMBF;
    POS        = -278, -212;
}
TERM {
    TERMNAME = false;
    POINTS   = (0.0, 1.0),
              (0.33334, 1.0),
              (1.0, 1.0);
}
(0.66666, 0.0),
(1.0, 0.0);
SHAPE = LINEAR;
COLOR = RED (255), GREEN (0), BLUE (0);
}

TERM {
   TERMNAME = true;
   POINTS = (0.0, 0.0),
   (0.33334, 0.0),
   (0.66666, 1.0),
   (1.0, 1.0);
SHAPE = LINEAR;
COLOR = RED (0), GREEN (128), BLUE (0);
}
}

LVAR {
   NAME = HEMOSPERMIA;
   BASEVAR = Units;
   LVLRANGE = MIN(0.0), MAX(1.0),
               MINDEF(0), MAXDEF(65535),
               DEFAULT_OUTPUT(0.5);
   RESOLUTION = XGRID(0.1), YGRID(0.1),
               SHOWGRID (ON), SNAPTOGRID(OFF);
   COLOR = RED (0), GREEN (128), BLUE (128);
   INPUT = CMBF;
POS = -279, -184;

TERM {
    TERMNAME = false;
    POINTS = (0.0, 1.0),
             (0.33334, 1.0),
             (0.66666, 0.0),
             (1.0, 0.0);
    SHAPE = LINEAR;
    COLOR = RED (255), GREEN (0), BLUE (0);
}

TERM {
    TERMNAME = true;
    POINTS = (0.0, 0.0),
             (0.33334, 0.0),
             (0.66666, 1.0),
             (1.0, 1.0);
    SHAPE = LINEAR;
    COLOR = RED (0), GREEN (128), BLUE (0);
}

} /* LVAR */

LVAR {
    NAME = PAIN EJAC;
    BASEVAR = Units;
    LVRANGE = MIN(0.0), MAX(1.0),
              MINDEF(0), MAXDEF(65535),
DEFAULT_OUTPUT(0.5);
RESOLUTION = XGRID(0.1), YGRID(0.1),
SHOWGRID (ON), SNAPTOGRID(OFF);
COLOR = RED (0), GREEN (0), BLUE (128);
INPUT = CMBF;
POS = -276, -154;
TERM {
TERMNAME = false;
POINTS = (0.0, 1.0),
(0.33334, 1.0),
(0.66666, 0.0),
(1.0, 0.0);
SHAPE = LINEAR;
COLOR = RED (255), GREEN (0), BLUE (0);
}
TERM {
TERMNAME = true;
POINTS = (0.0, 0.0),
(0.33334, 0.0),
(0.66666, 1.0),
(1.0, 1.0);
SHAPE = LINEAR;
COLOR = RED (0), GREEN (128), BLUE (0);
}
} /* LVAR */
LVAR {
    NAME       = PERINEAL PAIN;
    BASEVAR    = Units;
    LVRANGE    = MIN(0.0), MAX(1.0),
                  MINDEF(0), MAXDEF(65535),
                  DEFAULT_OUTPUT(0.5);
    RESOLUTION = XGRID(0.1), YGRID(0.1),
                  SHOWGRID (ON), SNAPTOGRID(OFF);
    COLOR      = RED (0), GREEN (0), BLUE (0);
    INPUT      = CMBF;
    POS        = -280, -35;
}

TERM {
    TERMNAME = false;
    POINTS   = (0.0, 1.0),
              (0.33334, 1.0),
              (0.66666, 0.0),
              (1.0, 0.0);
    SHAPE    = LINEAR;
    COLOR    = RED (255), GREEN (0), BLUE (0);
}

TERM {
    TERMNAME = true;
    POINTS   = (0.0, 0.0),
              (0.33334, 0.0),
              (0.66666, 1.0),
SHAPE  = LINEAR;
COLOR  = RED (0), GREEN (128), BLUE (0);

*/ LVAR */

LVAR {
NAME       = PSA;
BASEVAR    = Units;
LVRANGE    = MIN(1.0), MAX(2.0),
               MINDEF(0), MAXDEF(65535),
               DEFAULT_OUTPUT(1.5);
RESOLUTION = XGRID(0.1), YGRID(0.1),
               SHOWGRID(ON), SNAPTOGRID(OFF);
COLOR      = RED (0), GREEN (255), BLUE (128);
INPUT      = CMBF;
POS        = -286, 126;

TERM {
TERMNAME  = low;
POINTS     = (1.0, 1.0),
            (1.25, 1.0),
            (1.5, 0.0),
            (2.0, 0.0);
SHAPE     = LINEAR;
COLOR     = RED (255), GREEN (0), BLUE (0);
}
TERM {
    TERMNAME = medium;
    POINTS   = (1.0, 0.0),
               (1.25, 0.0),
               (1.5, 1.0),
               (1.75, 0.0),
               (2.0, 0.0);
    SHAPE    = LINEAR;
    COLOR    = RED (0), GREEN (128), BLUE (0);
}

TERM {
    TERMNAME = high;
    POINTS   = (1.0, 0.0),
               (1.5, 0.0),
               (1.75, 1.0),
               (2.0, 1.0);
    SHAPE    = LINEAR;
    COLOR    = RED (0), GREEN (0), BLUE (255);
}
}
} /* LVAR */

LVAR {
    NAME     = Ur. An. : HAEMAT;
    BASEVAR  = Units;
    LVRANGE  = MIN(0.0), MAX(1.0),
               MINDEF(0), MAXDEF(65535),

DEFAULT_OUTPUT(0.5);

RESOLUTION = XGRID(0.1), YGRID(0.1),

SHOWGRID (ON), SNAPTOGRID(OFF);

COLOR      = RED (0), GREEN (128), BLUE (255);

INPUT      = CMBF;

POS        = -286, 101;

TERM {
    TERMNAME = false;
    POINTS   = (0.0, 1.0),
                (0.33334, 1.0),
                (0.66666, 0.0),
                (1.0, 0.0);
    SHAPE    = LINEAR;
    COLOR    = RED (255), GREEN (0), BLUE (0);
}

TERM {
    TERMNAME = true;
    POINTS   = (0.0, 0.0),
                (0.33334, 0.0),
                (0.66666, 1.0),
                (1.0, 1.0);
    SHAPE    = LINEAR;
    COLOR    = RED (0), GREEN (128), BLUE (0);
}
}

} /* LVAR */
LVAR {
    NAME       = Ur. An. : PYURIA;
    BASEVAR    = Units;
    LV RANGE   = MIN(0.0), MAX(1.0),
                  MINDEF(0), MAXDEF(65535),
                  DEFAULT_OUTPUT(0.5);
    RESOLUTION = XGRID(0.1), YGRID(0.1),
                  SHOWGRID (ON), SNAPTOGRID(OFF);
    COLOR      = RED (0), GREEN (255), BLUE (0);
    INPUT      = CMBF;
    POS        = -283, 74;
}
TERM {
    TERMNAME = false;
    POINTS   = (0.0, 1.0),
               (0.33334, 1.0),
               (0.66666, 0.0),
               (1.0, 0.0);
    SHAPE    = LINEAR;
    COLOR    = RED (255), GREEN (0), BLUE (0);
}
TERM {
    TERMNAME = true;
    POINTS   = (0.0, 0.0),
               (0.33334, 0.0),
               (0.66666, 1.0),
(1.0, 1.0);
SHAPE = LINEAR;
COLOR = RED (0), GREEN (128), BLUE (0);
}
} /* LVAR */

LVAR {
NAME = DIAGNOSIS;
BASEVAR = Units;
LVRANGE = MIN(1.0), MAX(3.0),
          MINDEF(0), MAXDEF(65535),
          DEFAULT_OUTPUT(2.0);
RESOLUTION = XGRID(0.2), YGRID(0.1),
             SHOWGRID (ON), SNAPTOGRID(OFF);
COLOR = RED (255), GREEN (0), BLUE (0);
OUTPUT = COM;
POS = 137, -108;
TERM {
TERMNAME = negative;
POINTS = (1.0, 0.0),
(1.5, 1.0),
(2.0, 0.0),
(3.0, 0.0);
SHAPE = LINEAR;
COLOR = RED (255), GREEN (0), BLUE (0);
}
TERM {
    TERMNAME = zero;
    POINTS   = (1.0, 0.0),
               (1.5, 0.0),
               (2.0, 1.0),
               (2.5, 0.0),
               (3.0, 0.0);
    SHAPE    = LINEAR;
    COLOR    = RED (0), GREEN (128), BLUE (0);
}

TERM {
    TERMNAME = positive;
    POINTS   = (1.0, 0.0),
               (2.0, 0.0),
               (2.5, 1.0),
               (3.0, 0.0);
    SHAPE    = LINEAR;
    COLOR    = RED (0), GREEN (0), BLUE (255);
}

} /* LVAR */

} /* VARIABLE_SECTION */

OBJECT_SECTION {
    RULEBLOCK {
        NAME     = RB1;
    }
}
INPUT = AGE, BONE PAIN, CHILLS, DRE: ENLARGED, DRE: PAINFUL, DRE: STONY/HARD, FEVER, GHAEMATURIA, PSA, Ur. An.: HAEMAT, Ur. An.: PYURIA;

OUTPUT = DIAGNOSIS;

AGGREGATION = (MIN_MAX, PAR (0.0));

RESULT_AGGR = MAX;

POS = -78, -172;

RULES {
  IF AGE = low
    AND DRE: ENLARGED = false
    AND FEVER = false
    AND PSA = medium
  THEN DIAGNOSIS = negative WITH 0.438;

  IF AGE = low
    AND DRE: ENLARGED = false
    AND FEVER = false
    AND GHAEMATURIA = false
    AND PSA = medium
  THEN DIAGNOSIS = positive WITH 0.148;

  IF AGE = medium
    AND DRE: ENLARGED = false
    AND FEVER = false
    AND PSA = medium
  THEN DIAGNOSIS = negative WITH 0.547;

  IF AGE = medium
    AND DRE: ENLARGED = false
    AND PSA = medium
  THEN DIAGNOSIS = positive WITH 0.148;

  IF AGE = medium
    AND DRE: ENLARGED = false
    AND PSA = medium
  THEN DIAGNOSIS = negative WITH 0.547;
AND FEVER = false
AND GHAEMATURIA = false
AND PSA = medium
THEN DIAGNOSIS = positive WITH 0.164;

IF AGE = high
AND DRE: ENLARGED = false
AND FEVER = false
AND GHAEMATURIA = false
AND PSA = medium
THEN DIAGNOSIS = negative WITH 0.172;

IF FEVER = true
THEN DIAGNOSIS = zero WITH 0.813;

IF DRE: ENLARGED = true
AND FEVER = false
THEN DIAGNOSIS = negative WITH 0.023;

IF DRE: ENLARGED = false
AND FEVER = false
AND PSA = medium
THEN DIAGNOSIS = positive WITH 0.234;

IF DRE: ENLARGED = false
AND FEVER = false
AND PSA = low
THEN DIAGNOSIS = positive WITH 0.578;

IF DRE: ENLARGED = false
AND FEVER = false
AND PSA = high

THEN DIAGNOSIS = positive WITH 0.656;

IF DRE: ENLARGED = false
AND FEVER = false
AND GHAEMATURIA = false
AND PSA = medium

THEN DIAGNOSIS = positive WITH 0.781;

IF DRE: ENLARGED = false
AND FEVER = false
AND GHAEMATURIA = false
AND PSA = low

THEN DIAGNOSIS = positive WITH 0.258;

} /* RULES */

} /* RULEBLOCK */

} /* OBJECT_SECTION */

} /* MODEL */

} /* PROJECT */

/* fuzzyTECH 5.54d Online Edition */

ΠΧ-Όττ

ΤΜπή"ΞΒ†TM = 20090202134722UT;

} /* ONLINE */

NEUROFUZZY {

LEARNRULE =RandomMethod;

STEPWIDTHDOS = 0.100000;
STEPWIDTHTERM = 1.000000;
MAXDEVIATION = (50.000000, 1.000000, 0.750000);
AVGDEVIATION = 0.100000;
MAXSTEPS = 100;
NEURONS = 1;
DATASEQUENCE = RANDOM;
UPDATEDBGWIN = OFF;
} /* NEUROFUZZY */