INFORME 2. SISTEMAS BASADOS EN REGLAS Y TEORIA DE LA INCERTIDUMBRE

Alumno: Mohammed Amrou Labied Nasser
Subgrupo: 1.1
Tutora: María del Carmen Garrido Carrera
Año: 2024/25

INDICE

1. ENCADENAMIENTO HACIA ATRÁS CON FACTORES DE CERTEZA

La implementación de este algoritmo tendrá dos partes como sucede en el encadenamiento hacia atrás general, donde se especifica un meta objetivo y se trata de determinar si la meta se verifica teniendo en cuenta el contenido de la base de hechos.

Pero, a la hora de introducir incertidumbre necesitaremos acumular toda la información que se va obteniendo cuando se aplican distintas reglas, esto se traduce en el código en un análisis de casos en el que antes de añadir la meta a la base de hecho se realizan distintas operaciones (depende de cada caso) de los factores de certeza obteniendo la acumulación todos ellos. A continuación, mostraremos la función verificar donde se comprueba si la meta está en la base de hechos.

```
FUNCION verificar(meta)
    SI meta está en BH RETORNAR “verdadero”
    SINO
        CC ← equiparar(meta, BC)
       MIENTRAS NoVacio(CC)
         hacer
         regla = seleccionar(CC)
         CC++;
         nuevasMetas ← seleccionarMeta(regla)
         verificado ← verdadero
      MIENTRAS noVacio(nuevasMetas) Y verificado
            nuevaMeta = seleccionar(nuevasMetas)
            nuevasMetas++
            verificado = verificar(nuevaMeta)
        SI verificado
            // CASO 1
            factAnterior ← fc (del primer antecedente en BH)
            operador ← identificar(operador)

            PARA CADA antecedente DESPUÉS DEL PRIMERO EN nuevasMetas
                fact ← FC (antecedente en BH, Factor de certeza actual)
                SI operador = "y"
                    factAnterior ← min(factAnterior, fact)
                SINO SI operador ES "o"
                    factAnterior ← max(factAnterior, fact)
                FIN SI
            FIN PARA
            regla = setFCAccion(factAnterior)
            // CASO 3
            SI getFCAccion(regla) >= 0
               regla = setFCAccion(fcRegla * getFCAccion(regla))
            SINO
               regla = setFCAccion(fcRegla * max(0, getFCAccion(regla))
        FIN SI
      FIN MIENTRAS

      SI MasDeUna(CC)
        fc ← seleccionar(CC, FCAccion)
        eliminarPrimero(CC)
```

```
        MIENTRAS noVacio(CC)
           // CASO 2
            fcAct ← seleccionar(CC, FCAccion)
            SI fc >= 0 Y fcAct >= 0
               fc = fc + fcAct * (1 – fcA)
            SINO fc <= 0 Y fcAct <= 0
               fc = fc + fcAct * (1 + fcA)
            SINO
               fc = (fc + fcAct) / (1 – min(|fcA|, |fcAct|))
            FIN SI
        FIN MIENTRAS
            anadirMeta(meta, fc)
            RETORNAR verdadero
      SINO
            fc = seleccionar(CC, FCAccion)
            anadirMeta(meta, fc)
            RETORNAR verdadero
      FIN SI
      RETORNAR falso
FIN FUNCION
```

Tras ver la función verificar, vamos a observar la función principal que simplemente carga la base de hechos inicial y utiliza la anterior función descrita para comprobar si la meta está en la base de hechos.

```
FUNCION encadenamientoHaciaAtras(meta)
    BH = leerBH(BaseH)
    SI verifica(meta) entonces RETORNAR “verdadero”
    SINO RETORNAR “falso”
```

2. PRUEBAS

En este apartado vamos a realizar distintas pruebas proporcionadas para comprobar el correcto funcionamiento de nuestro sistema, además de solucionar estos problemas con nuestro SBR-FC.

2.1 PRUEBA 2 Y 3 CON FORMALIZACIÓN Y ESTRUCTURA

Comenzaremos con la formalización de la prueba tres que consiste en identificar todas las variables para poder construir el conjunto de reglas y hechos.

1. Formalizamos:
Sea la siguiente signatura: ∑ = {c2y3, c3+, co2y3, co3+, condC, condE, noVs, cAcc, condJ, bA} donde;

c2y3 = “Conductor con antigüedad entre 2-3 años”
c3+ = “Conductor con antigüedad mayor a 3 años”
co2y3 = “Conducir entre 2-3 horas”
co3+ = “Conducir más de 3 horas”
condC = “Conductor está cansado”
condE = “Conductor es experimentado”
noVs = “No viaja solo”
cAcc = “Causante del accidente “
condJ = “Conductor joven”
bA = “ha bebido alcohol”

2. Construimos el conjunto de reglas y hechos:

REGLAS

R1: Si c2y3 entonces condE, FC=0.5
R2: Si c3+ entonces condE, FC=0.9
R3: Si co2y3 entonces condC, FC=0.5
R4: Si co3+ entonces condC, FC=1.0
R5: Si condE y noVs entonces cAcc, FC = -0.5
R6: Si condC entonces cAcc, FC = 0.5
R7: Si condJ o bA entonces cAcc, FC= 0.7

HECHOS

c3+, FC=1.0
co2y3, FC=1.0
condJ, FC=0.4
noVs, FC=-1.0
co3+, FC=-1.0
bA, FC=0.0
c2y3, FC=1.0

Tras formalizar la prueba tres, vamos a especificar la base de hechos y de conocimiento que deberemos de pasar como parámetros para la prueba dos y tres.

PRUEBA 2

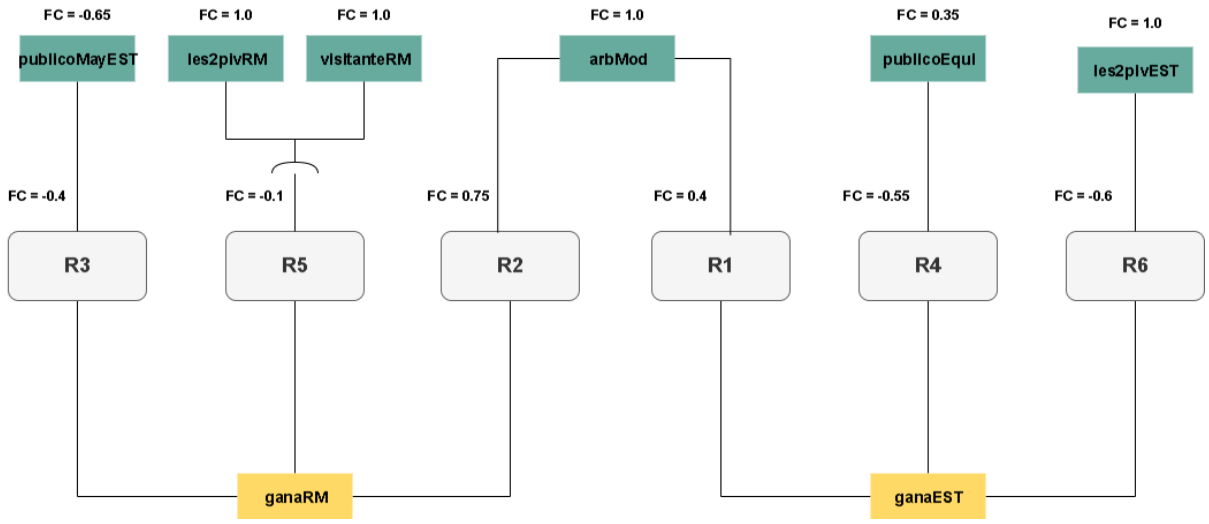
// Base de conocimiento	// Base de hechos
6 R1: Si arbMod Entonces ganaEST, FC=0.4 R2: Si arbMod Entonces ganaRM, FC=0.75 R3: Si publicoMayEST Entonces ganaRM, FC= 0.4 R4: Si publicoEqui Entonces ganaEST, FC= 0.55 R5: Si les2pivRM y visitanteRM Entonces ganaRM, FC= 0.1 R6: Si les2pivEST Entonces ganaEST, FC= 0.6	7 localEST, FC=1 visitanteRM, FC=1 arbMod, FC=1 publicoMayEST, FC=0.65 publicoEqui, FC=0.35 les2pivEST, FC=1 les2pivRM, FC=1 Objetivo ganaEST

PRUEBA 3

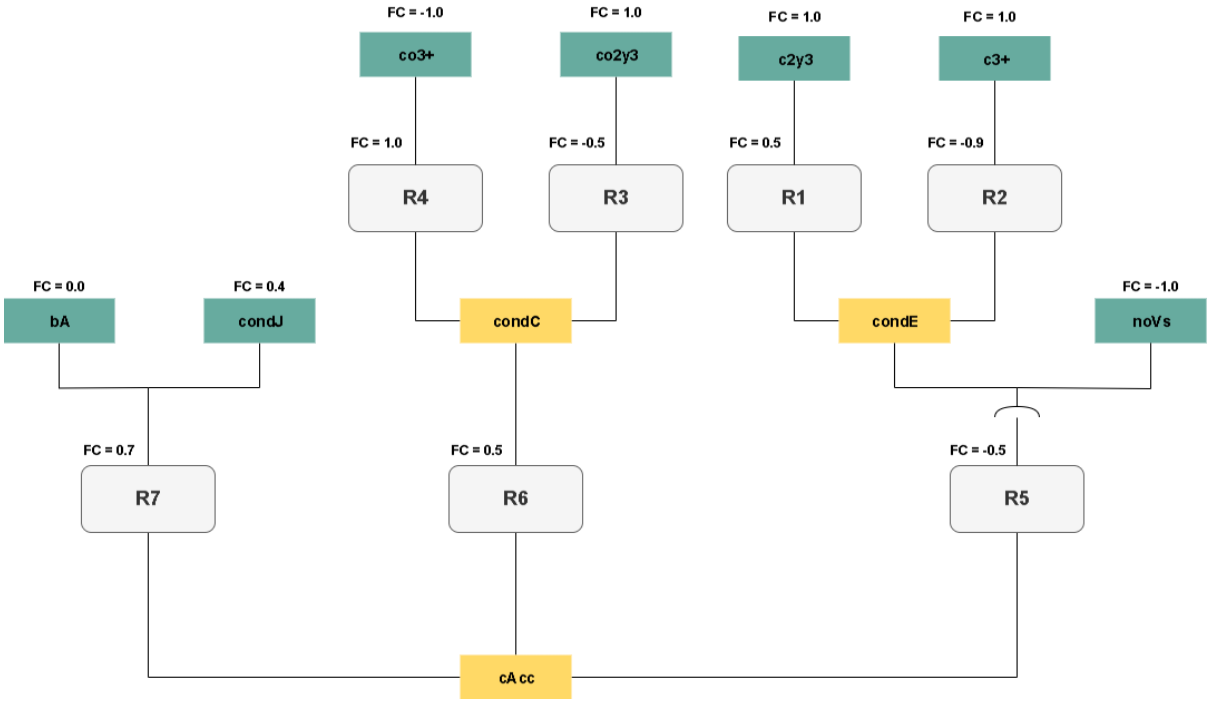
// Base de conocimiento	// Base de hechos
7 R1: Si c2y3 Entonces condE, FC=0.5 R2: Si c3+ Entonces condE, FC=0.9 R3: Si co2y3 Entonces condC, FC=0.5 R4: Si co3+ Entonces condC, FC=1.0 R5: Si condE y noVs Entonces cAcc, FC = -0.5 R6: Si condC Entonces cAcc, FC = 0.5 R7: Si condJ o bA Entonces cAcc, FC= 0.7	7 c3+, FC=1.0 co2y3, FC=1.0 condJ, FC=0.4 noVs, FC=-1.0 co3+, FC=-1.0 bA, FC=0.0 c2y3, FC=1.0 Objetivo cAcc

A continuación, vamos a mostrar las redes de inferencia de la prueba 2 y la prueba 3. Los grafos Y se representan como el carácter C hacia abajo.

PRUEBA 2



PRUEBA 3



2.2 PRUEBA DISEÑADA CON ENUNCIADO, FORMALIZACIÓN Y ESTRUCTURA

Para validar nuestro propio sistema SBR-FC, vamos a diseñar un problema donde indicaremos su enunciado, para formalizar y extraer las bases de conocimiento y hechos. El problema es el siguiente.

Sea un sistema inteligente en el que disponemos de una base de conocimiento compuesta por 5 reglas:

R1. Si se cumple h2 o h3 se cumplirá h1 con una certeza del 0.35
R2. Si se cumple h4 y h8 se cumplirá h1 con una certeza del 0.55
R3. Si se cumple tanto h5 como h6 se cumplirá h3 con una certeza del 0.25
R4. Si se cumple h7 es muy posible que no se cumpla h3 con una certeza del –0.5
R5. Si se cumple h5 o h8 es muy posible que se cumpla h2 con una certeza del 0.75

En un momento determinado se tiene la siguiente información sobre los hechos. Se está cumpliendo h8 con grado 0.6, h4 con grado –0.4, h5 con grado -0.3, h6 con grado 0.45 y h7 con grado –0.15
Con esta información, ¿Se está cumpliendo h1?

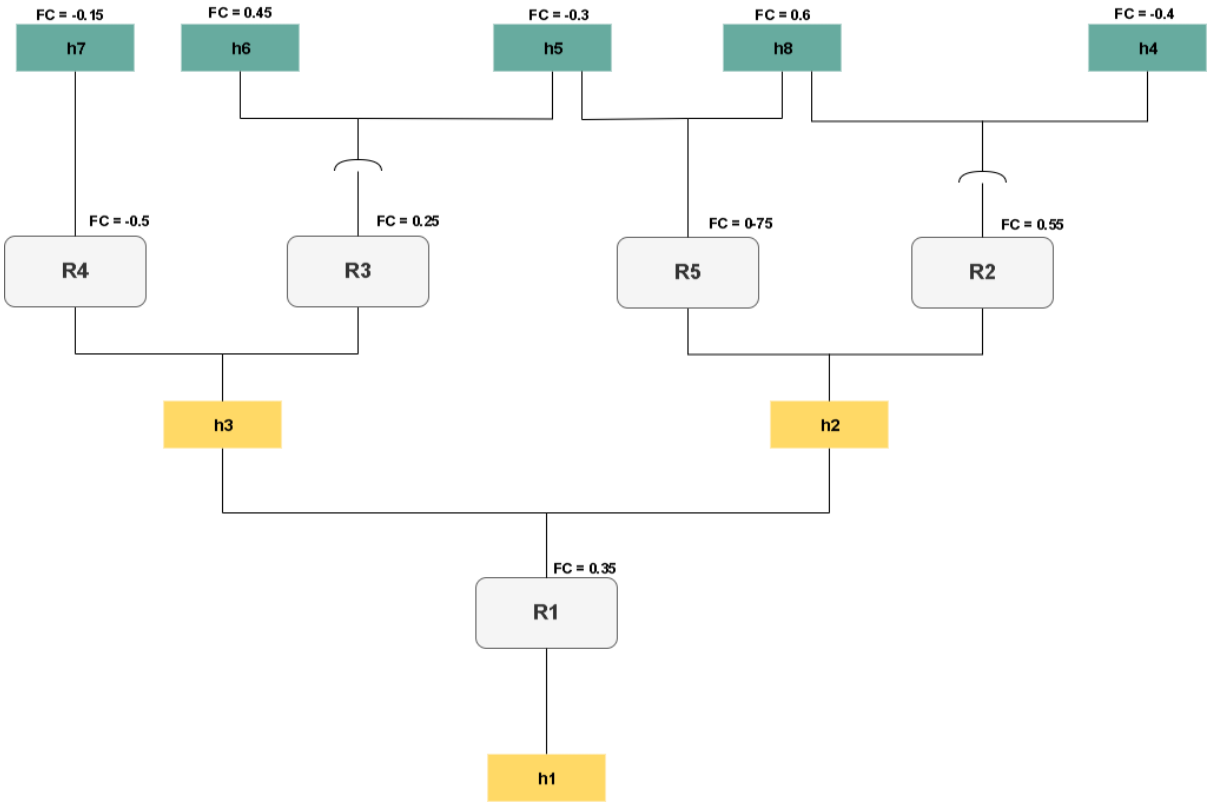
1. Formalizamos:
Sea la siguiente signatura ∑ = {h1, h2, h3, h4, h5, h6, h7, h8} donde
h1 =" se cumple h1"; h2 =" se cumple h2"; h3 =" se cumple h3";
h4 =" se cumple h4"; h5 =" se cumple h5"; h6 =" se cumple h6";
h7 =" se cumple h7; h8 =" se cumple h8"

2. Construimos el conjunto de reglas y hechos;

REGLAS
R1: Si h2 o h3 entonces h1, FC=0.35
R2: Si h4 y h8 entonces h1, FC=0.55
R3: Si h5 y h6 entonces h2, FC=0.25
R4: Si h7 entonces h3, FC=-0.5
R5: Si h5 o h8 entonces h2, FC =0.75
HECHOS
h8, FC=0.6
h4, FC=-0.4
h5, FC=-0.3
h6, FC=0.45
h7, FC=-0.15
Tras formalizar la prueba de diseño, vamos a especificar la base de hechos y de conocimiento que deberemos de pasar como parámetros al ejecutable.

// Base de conocimiento	// Base de hechos
5	5
R1: Si h2 o h3 Entonces h1, FC=0.35	h8, FC=0.6
R2: Si h4 y h8 Entonces h3, FC=0.55	h4, FC=-0.4
R3: Si h5 y h6 Entonces h3, FC=0.25	h5, FC=-0.3
R4: Si h7 Entonces h3, FC=-0.5	h6, FC=0.45
R5: Si h5 o h8 Entonces h2, FC=0.75	h7, FC=-0.15
	Objetivo
	h1

Vamos a ver la red de inferencia de nuestro problema diseñado.



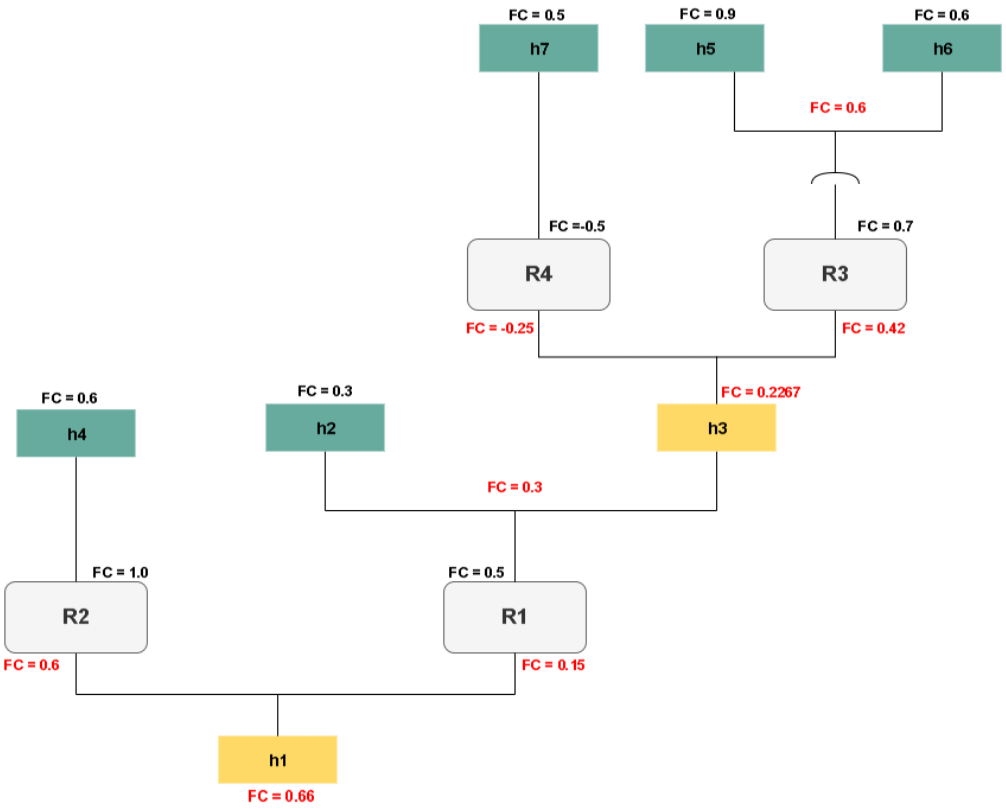
3. EJECUCIONES DE LAS DISTINTAS PRUEBAS

A continuación, vamos a mostrar el proceso seguido por nuestro algoritmo para el cálculo de los factores de incertidumbre de cada problema y responderemos a las preguntas que se nos plantean.

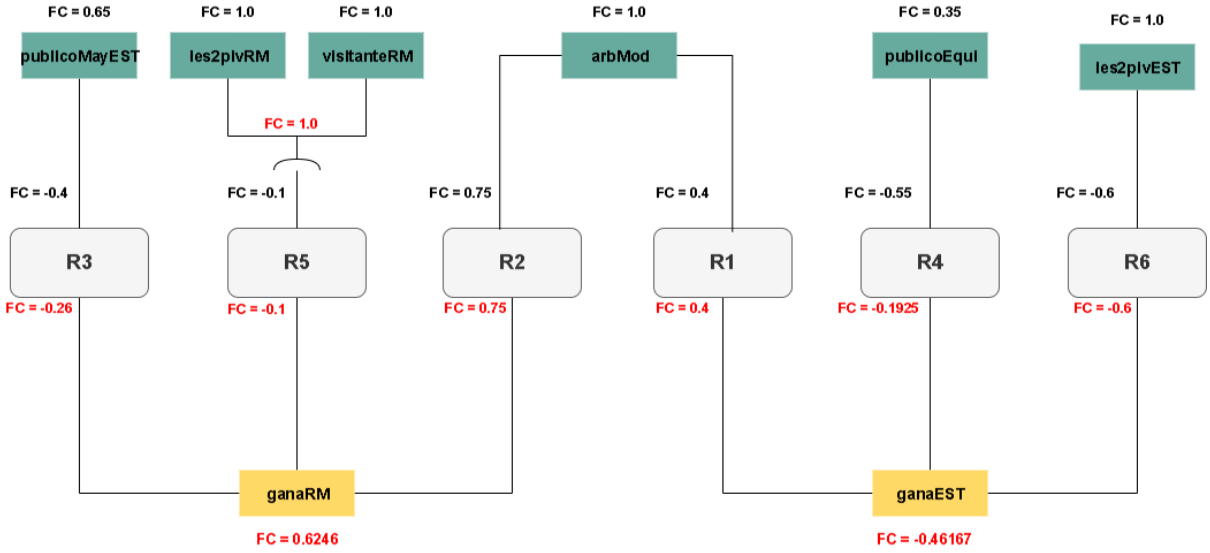
3.1. REDES DE INFERENCIA Y SUS FACTORES DE CERTEZA

Vamos a simular el proceso de cálculo en cada nivel de regla, y observaremos el resultado que se obtienes tras encadenar varios factores según su tipo. Para mayor exactitud, se puede consultar el informe generado por el ejecutable donde muestra que caso se aplica.

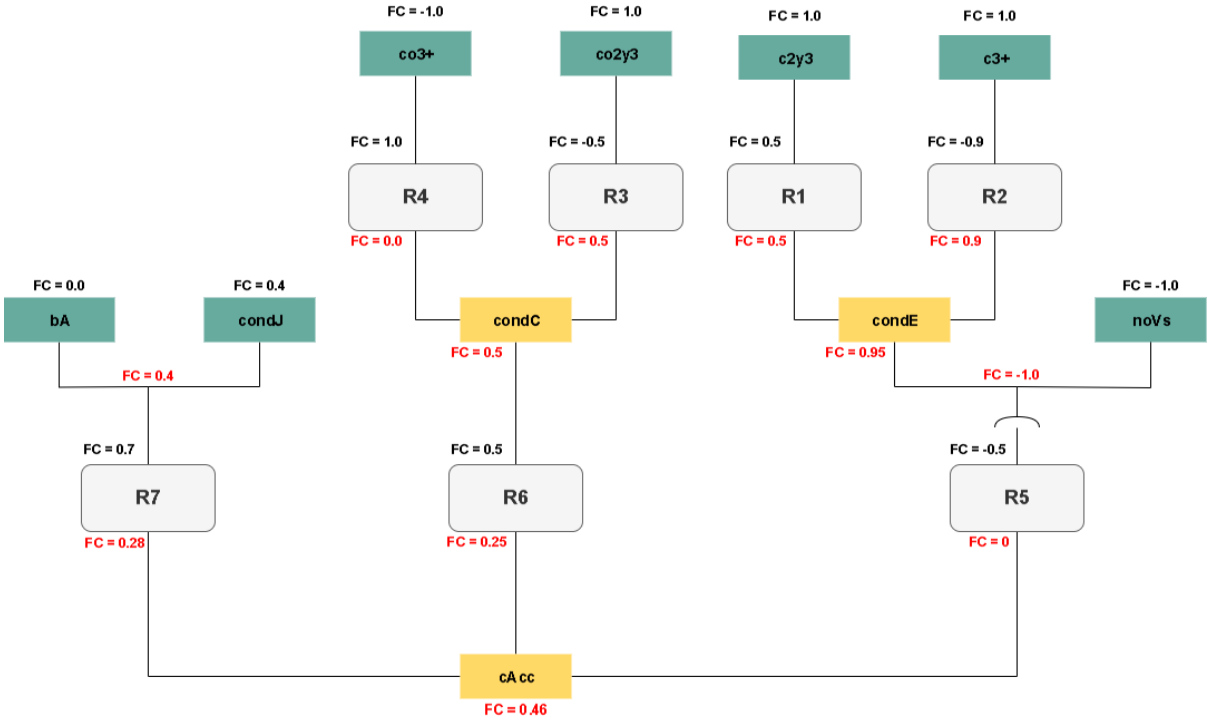
PRUEBA 1 CON LOS FACTORES INFERIDOS



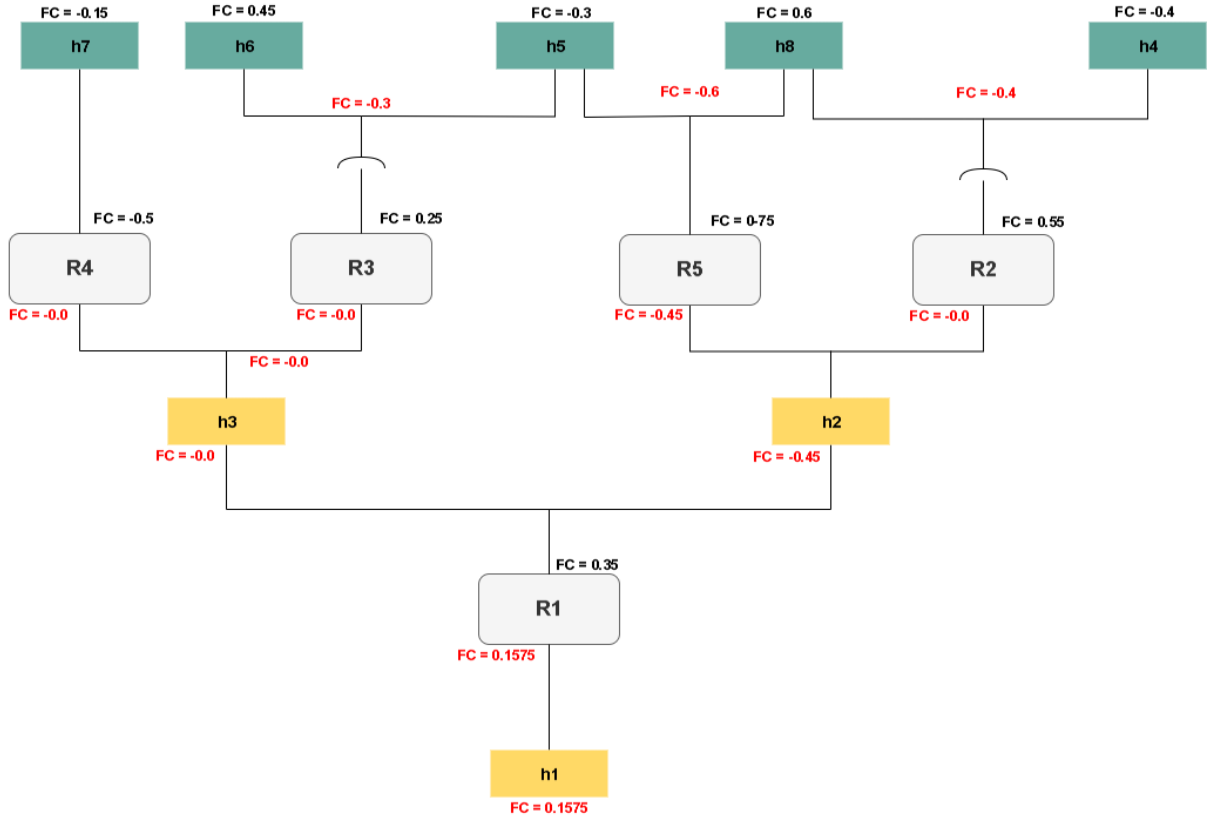
PRUEBA 2 CON LOS FACTORES INFERIDOS



PRUEBA 3 CON LOS FACTORES INFERIDOS



PRUEBA DISEÑADA CON LOS FACTORES INFERIDOS



3.2. CONCLUSION DE LA PRUEBA Y RESULTADO

Para la prueba 1 podemos concluir que h1 se cumple con un factor de certeza de 0.66.
Para la prueba 2 tenemos que comprobar dos objetivos y ver entre los dos equipos que factor de certeza es mayor, este será quien gane el partido y, por tanto, la liga. Para ello hemos calculado

el factor de certeza de ganaEST que es un –0.46 y el factor de certeza de ganaRM es de un 0.62. En conclusión, el equipo ganador es el Real Madrid.

Para la prueba 3 tenemos como objetivo si el conductor es causante de un accidente en base a algunas evidencias y el resultado es de 0.46. Con este factor de certeza podemos concluir que es responsable del accidente con un grado del 73% (este porcentaje se calcula como 0.5 + 0.46/2) por lo que podemos decir que es un buen dato para que sea culpable.

Y, por último, para la prueba diseñada h1 se cumple con un factor de certeza del 0.1575.