

Sistemas Inteligentes 3º curso

Practica 2 – Diseño de un Sistema Basado en Reglas
con razonamiento hacia atrás.



Fecha de entrega domingo 15 de diciembre.

Alumno: Jorge Serrano Rueda

Grupo: 2.3

Profesor: Víctor Gomollón Martos

Correo: Jorge.serranor@um.es

Contenido

Modificaciones realizadas al algoritmo	3
Pruebas.....	4
Prueba 2	4
Formalización.....	5
Base de conocimiento	5
Base de hechos	5
Red de inferencia	6
Prueba 3	6
Formalización.....	6
Base de conocimiento	6
Base de hechos	7
Red de inferencia.....	7
Prueba A	8
Enunciado	8
Formalización.....	8
Base de conocimiento	9
Base de hechos	9
Red de inferencia.....	10
Ejecuciones.....	11
Prueba 1	11
Prueba 2	11
Prueba 3	12
Prueba A	12
Conclusión.....	13

Modificaciones realizadas al algoritmo

En esta sección vamos a presentar las modificaciones que han sido necesarias para adaptar al algoritmo proporcionado para que funcione con incertidumbre.

```
FUNCIÓN VERIFICAR
    Verificado = FALSO;

    SI Contendida(Meta, BH) ENTONCES
        DEVOLVER VERDADERO;

    CC = Equiparar(Consecuentes(BC), Meta);

    MIENTRAS NoVacio(CC) Y NO Verificado HACER
        R = Resolver(CC);
        Eliminar(R, CC);
        NuevasMetas = ExtraerAntecedentes(R);
        Verificado = VERDADERO;

        MIENTRAS NoVacio(NuevasMetas) Y Verificado HACER
            NMet = SeleccionarMeta(NuevasMetas);
            Eliminar(NMet, NuevasMetas);
            Verificado = VERIFICAR(NMet, BH);

    SI Verificado ENTONCES
        Añadir(Meta, BH);

    DEVOLVER Verificado;
```

1-Algoritmo original sin FC

La primera modificación necesaria es sustituir la variable verificado por una variable que almacene el factor de certeza de nuestra meta.

Si la meta está contenida en la base de hechos se devuelve su factor de certeza, si no lo está se equipara y vamos buscando nuevas metas mientras haya reglas en el conjunto conflicto. Sabemos que si entra en el primer bucle while al menos hay un antecedente por lo que no hace falta comprobación, extraemos la nueva meta y llamamos a la función verificar para esta nueva meta. La primera llamada de la nueva meta se hace fuera del bucle while porque no sabemos ante qué tipo de regla (sencilla, conjunción o disyunción) nos encontramos, si la regla actual tiene más de un antecesor esta vez entrara al bucle y se ejecutara el caso uno entre los factores de certeza de ambos antecesores.

Una vez que no queden nuevas metas para la regla actual se ejecuta el caso 3 sobre el factor de certeza resultado, finalmente en cada iteración de la regla actual se comprueba si es necesario aplicar el caso dos, es decir que había dos o más reglas que tenían como consecuente la meta actual. Una vez hecha la comprobación y si ha sido necesario aplicado el caso dos, cuando ya no queden más reglas en el conjunto conflicto se almacena la meta con su factor de certeza en la base de hechos y la función devuelve como resultado el factor de certeza calculado.

Este sería el pseudocódigo para un SBR-FC:

```
FUNCIÓN VERIFICAR
FCResultado = 0

SI Contenida(Meta, BaseHechos) ENTONCES
    DEVOLVER Meta.factorCerteza

CC = Equiparar(BaseConocimientos, Meta)
nCC = 0 // Contador para el número de reglas que tienen como consecuente la meta

MIENTRAS NoVacio(CC) HACER
    nCC++ // Contar las reglas

    R = Resolver(CC)
    Eliminar(CC)
    NuevasMetas = ExtraerAntecedentes(R)
    FCVerificado = VERIFICAR(SeleccionarMeta(NuevasMetas), BaseHechos, BaseConocimientos)
    Eliminar(NuevasMetas)

    // Si entra en el bucle es porque la regla tiene almenos una conjunción o disyunción
    MIENTRAS NoVacio(NuevasMetas) HACER
        Nmet = SeleccionarMeta(NuevasMetas)
        Eliminar(NuevasMetas)

        // Aplicar el caso 1 para cada nueva meta y el FC acumulado
        FCVerificado = aplicarCaso1(R, Nmet, FCVerificado, VERIFICAR(Nmet, BaseHechos, BaseConocimientos))

    FIN MIENTRAS

FCVerificado = aplicarCaso3(R, FCVerificado) // Siempre se aplica el caso 3

    // Si la meta es consecuente de más de una regla, aplicar el caso 2
    SI nCC > 1 ENTONCES
        FCResultado = aplicarCaso2(FCVerificado, FCResultado )
    SINO
        // solo se ha explorado una regla o no quedan más en el CC
        FCResultado = FCVerificado
    FIN SI

FIN MIENTRAS

AñadirBH(Meta, FCresultado, BaseHechos)

DEVOLVER FCresultado
```

2-Algoritmo-SBR-FC

Pruebas

En esta sección se proporcionan la información para completar tanto la prueba dos como la tres además de la prueba ideada por el alumno.

Prueba 2

En este caso, se presenta una eliminatoria final de la Liga ACB entre el Real Madrid y el Estudiantes, que están empatados a dos partidos. El quinto partido se jugará en la cancha del Estudiantes. La tarea consiste en evaluar la probabilidad de que gane el Real Madrid o el Estudiantes, considerando varios factores que impactan el rendimiento de ambos equipos.

Formalización

$\Sigma = \{\text{localEST}, \text{visitanteRM}, \text{arbMod}, \text{publicoMayEST}, \text{publicoEqui}, \text{les2pivEST}, \text{les2pivRM}, \text{GanaEST}, \text{GanaRM}\}$ donde

localEST = “ El estudiantes es el equipo local” .

visitanteRM = “ El Real Madrid es el equipo visitante” .

arbMod = “Los árbitros son moderados” .

publicoMayEST = “El público es mayoritariamente del estudiantes” .

publicoEqui = “El público esta equilibrado en dos equipos”.

les2pivEST = “El estudiantes tiene 2 pivots lesionados”.

les2pivRM = “El real Madrid tiene 2 pivots lesionados”.

GanaEST = “el estudiantes gana”.

GanaRM = “El Real Madrid gana”.

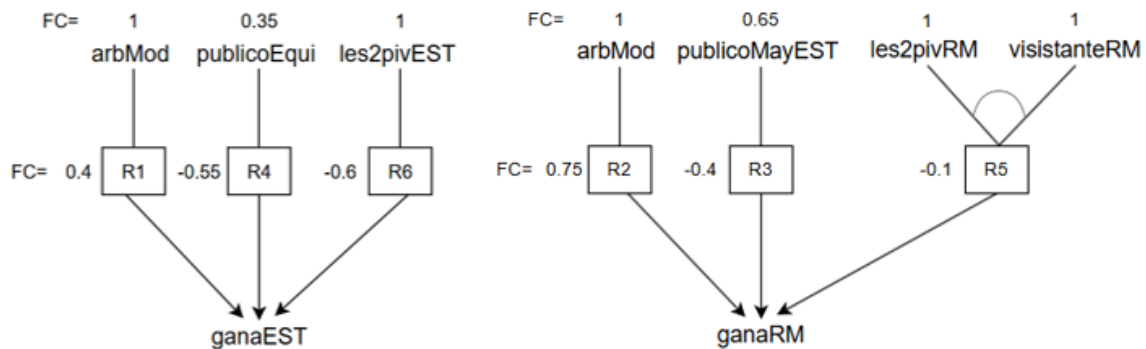
Base de conocimiento

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

Base de hechos

localEST,	FC=1
visitanteRM,	FC=1
arbMod,	FC=1
publicoMayEST,	FC=0.65
publicoEqui,	FC=0.35
les2pivEST,	FC=1
les2pivRM,	FC=1

Red de inferencia



Prueba 3

Para esta prueba se examina un conductor, sospechoso de ser responsable de un accidente de tráfico debemos averiguar si ha sido el responsable, teniendo en cuenta diversos factores que influyen en esta probabilidad: la experiencia del conductor, el estado de cansancio, si viaja solo o acompañado, la edad y si ha consumido alcohol.

Formalización

$\Sigma = \{ 2_3\text{años}, \text{mas3años}, \text{conduce2_3horas}, \text{conducemas3horas}, \text{experimentado}, \text{cansado}, \text{acompañado}, \text{joven}, \text{ebrio}, \text{responsable} \}$ donde

2_3años = "El conductor tiene entre 2 y 3 años de experiencia".

mas3años = "El conductor tiene más de 3 años de experiencia".

conduce2_3horas = "Ha conducido entre 2 y 3 horas".

conducemas3horas = "Ha conducido más de 3 horas".

experimentado = "Es un conductor experimentado".

cansado = "El conductor está cansado".

acompañado = "El conductor no viajaba solo".

joven = "El conductor es joven".

ebrio = "El conductor estaba bajo los efectos del alcohol".

responsable = "El conductor es responsable del accidente".

Base de conocimiento

R1: Si 2_3años Entonces experimentado,	FC=0.5
R2: Si mas3años Entonces experimentado,	FC=0.9
R3: Si conduce2_3horas Entonces cansado,	FC=0.5
R4: Si conducemas3horas Entonces cansado,	FC=1

R5: Si experimentado y acompañado Entonces responsable, FC=-0.5

R6: Si cansado Entonces responsable, FC=0.5

R7: Si joven o ebrio Entonces responsable, FC=0.7

Una aclaración importante es que cuando un conductor tiene más de tres años de experiencia podríamos pensar que el hecho 2_3años tiene factor de certeza de uno pero es justo lo contrario como sabemos que tiene más de tres años de experiencia estamos totalmente seguros de que no tiene entre dos y tres años, lo mismo pasa con el conduce entre dos y tres horas.

Base de hechos

2_3años, FC=-1

mas3años, FC=1

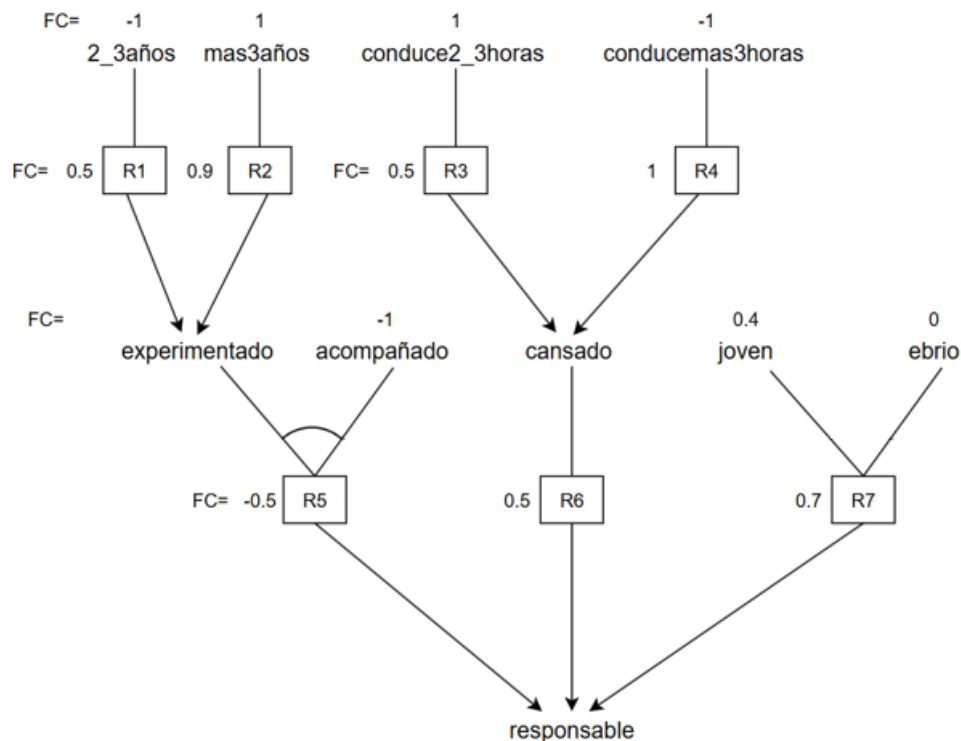
conduce2_3horas, FC=1

conducemas3horas FC=-1

acompañado, FC=-1

joven, FC=0.4

Red de inferencia



Prueba A

Para esta prueba tendremos que ayudar a un estudiante que necesitaba saber si llegará temprano a su examen en la universidad teniendo en cuenta diversos factores.

Enunciado

Tenemos a un estudiante preocupado porque mañana tiene un examen en la universidad al que debe presentarse, para ello primero debe prepararse para coger el bus a tiempo y posteriormente rezar para que no haya tráfico y el bus vaya en hora.

Sabemos que al estudiante le afectan los siguientes factores:

- Si se levanta temprano, hay evidencia a favor(0.8) de que saldrá temprano de casa.
- Si se ducha antes de irse hay evidencia en contra(-0.2) de que no saldrá temprano de casa.
- En caso de que desayune o se lave los dientes hay evidencia en contra de que no saldrá temprano de casa(-0.3).
- Si el autobús llega puntual a la parada anterior hay evidencia a favor(0.7) de que llegará puntual a la parada del estudiante.
- Si sale temprano de casa hacia la parada hay evidencia a favor(0.9) de que cogerá el bus.
- Si hay hueco en el autobús hay evidencia a favor (0.2) de que lo cogerá.
- Si el bus va puntual hay evidencia en contra(-0.2) de que no lo cogerá.
- Si coge el bus y el bus va en hora hay evidencia a favor de que llega a tiempo al examen(0.9)
- Si hay atasco en el camino hay evidencia en contra de que llegara tarde (-0.5)

Sabemos que el estudiante se levanta temprano con una certeza de 0.8, se va duchar, se lavara los dientes 0.5 y que desayunará, además el bus normalmente va en hora en la parada anterior asumiremos que mañana irá en hora con una certeza de 0.7, sabemos que el bus tiene hueco con certeza de 0.8, además hay atasco con certeza 0.4.

¿Llegará a tiempo el estudiante a su examen? ¿Tendrá que presentarse a la siguiente convocatoria por su insensatez?

Formalización

$\Sigma = \{ \text{seLevantaTiempo}, \text{seDucha}, \text{lavadientes}, \text{desayuna}, \text{busPuntualParadaAnterior}, \text{saleTempranoCasa}, \text{hayEspacioBus}, \text{busPuntual}, \text{cogeBus}, \text{hayAtasco} \}$ donde

seLevantaTiempo: El estudiante se levanta a tiempo (a las 8:00).

seDucha: El estudiante se ducha antes de ir a clase.

Lavadientes: El estudiante se lava los dientes.

Desayuna: El estudiante desayuna.

busPuntualParadaAnterior: El autobús llega tarde a la parada anterior.

saleTempranoCasa: El estudiante sale temprano de casa hacia la parada.

hayEspacioBus: Queda espacio en el autobús para que el estudiante se monte.

busPuntual: El autobús va en hora cuando llega a la parada del estudiante.

cogeBus: El estudiante coge el autobús y este va en horario.

hayAtasco: Hay tráfico o atasco en el camino hacia el destino.

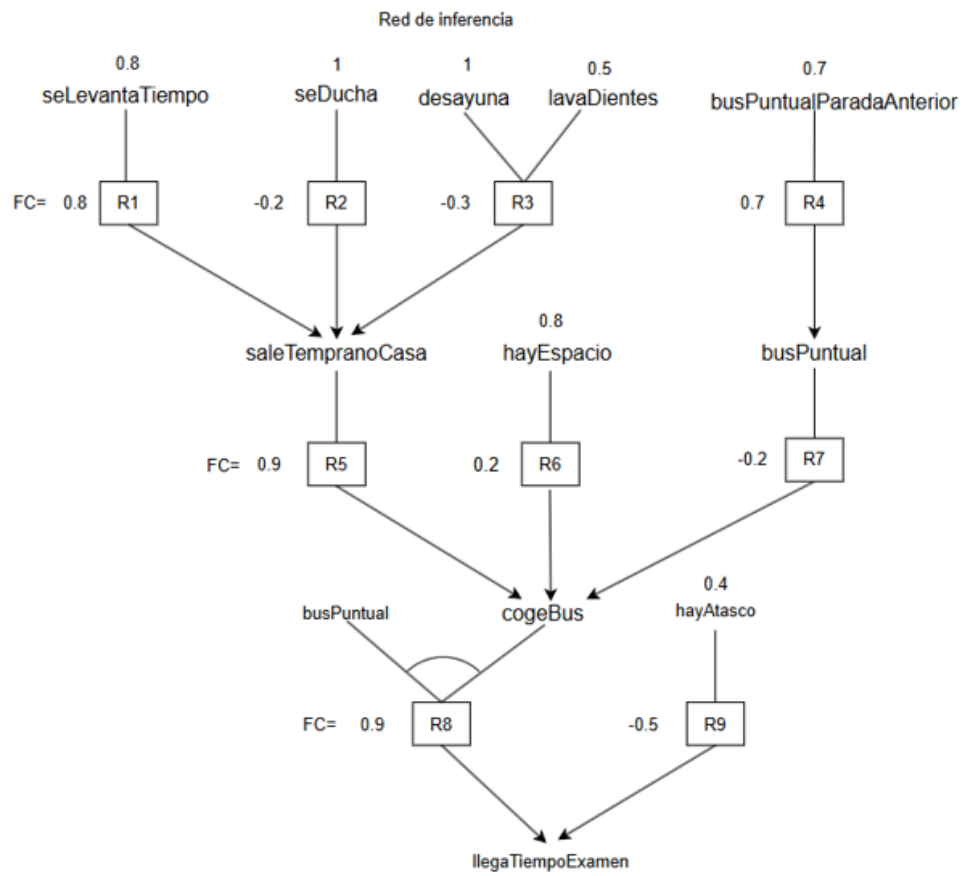
Base de conocimiento

R1: Si seLevantaTiempo Entonces saleTempranoCasa,	FC=0.8
R2: Si seDucha Entonces saleTempranoCasa,	FC=-0.2
R3: Si desayuna o lavaDientes Entonces saleTempranoCasa,	FC=-0.3
R4: Si busPuntualParadaAnterior Entonces busPuntual,	FC=0.7
R5: Si saleTempranoCasa Entonces cogeBus,	FC=0.9
R6: Si hayEspacioBus Entonces cogeBus,	FC=0.2
R7: Si busPuntual Entonces cogeBus,	FC=-0.2
R8: Si cogeBus y busPuntual Entonces llegaTiempoExamen,	FC=0.9
R9: Si hayAtasco Entonces llegaTiempoExamen,	FC=-0.5

Base de hechos

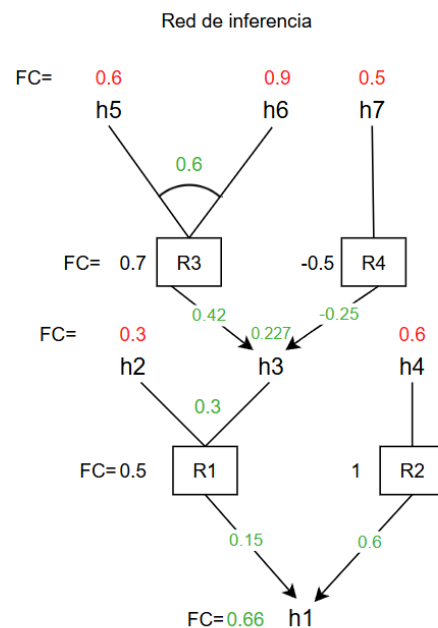
seLevantaTiempo,	FC=0.8
seDucha,	FC=1
desayuna,	FC= 1
lavadientes,	FC=0.5
busPuntualParadaAnterior,	FC=0.7
hayEspacioBus,	FC=0.8
hayAtasco,	FC=0.4

Red de inferencia



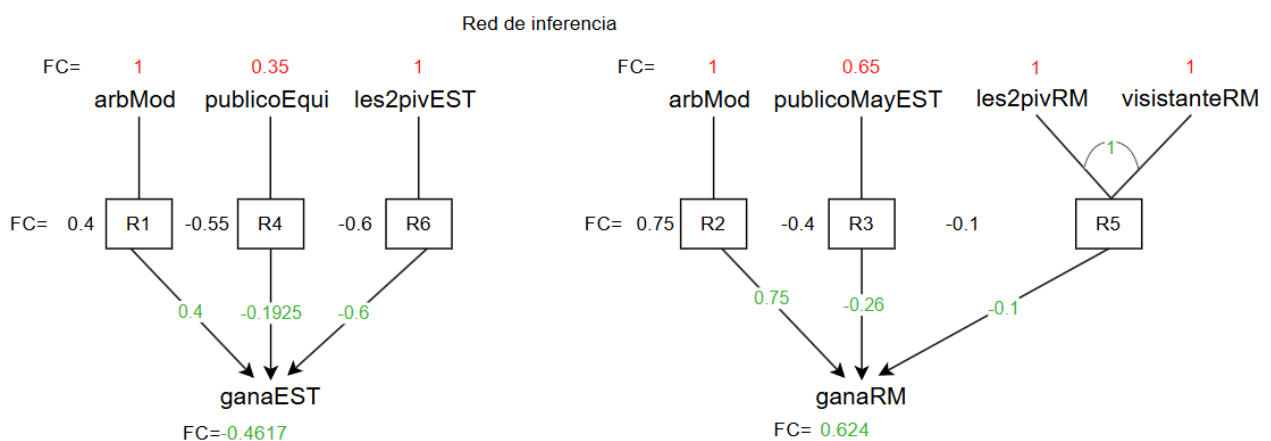
Ejecuciones

Prueba 1



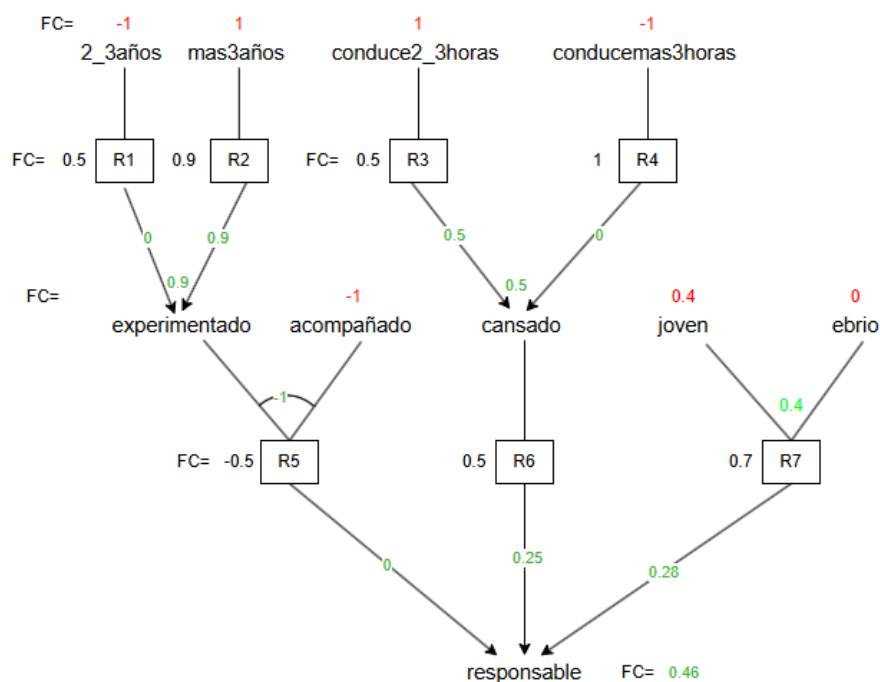
El resultado indica que es probable que el hecho $h1$ suceda o haya sucedido, aunque no es del todo fiable.

Prueba 2



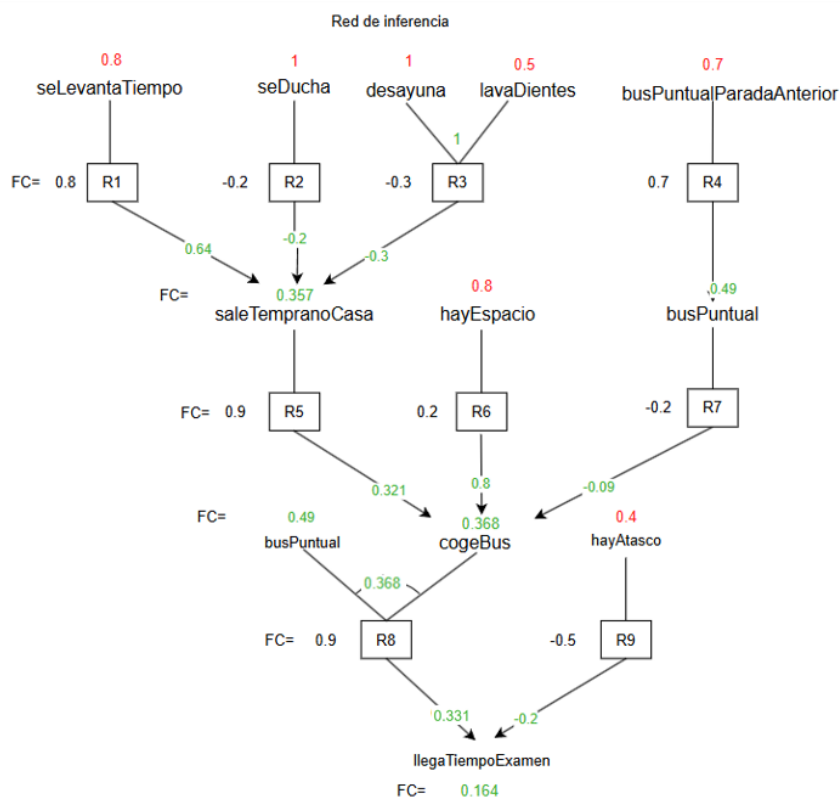
Analizando los resultados para $ganaEST$ es probable que no gane y para $ganaRM$ es un más probable que si lo hagan.

Prueba 3



Podemos concluir que es probable que el conductor fuese el responsable del accidente.

Prueba A



Finalmente, para el caso A lamentándolo mucho el estudiante no podrá llegar a tiempo a su examen y tendrá que presentarse a la siguiente convocatoria.

Conclusión

Este trabajo ha sido interesante. En mi caso, he hecho tantas veces los cálculos que he interiorizado el cálculo de los tres casos, y diría que estoy casi al nivel de velocidad del algoritmo. La práctica ha sido útil como introducción a los sistemas basados en reglas. Programar un poco en C++ no viene mal, sobre todo este año en el que no tenemos muchas asignaturas.

Sin embargo, esta práctica se me ha hecho muy tediosa, principalmente por el sentimiento constante de que siempre se puede hacer algo más o mejor. Cuando estaba programando la salida de la ejecución, no sé cuántas veces la habré cambiado. Finalmente, estoy contento con el resultado obtenido y, aún más, feliz de haber terminado el trabajo.

Espero que mi arduo esfuerzo haya valido la pena.