Factores de Certeza

Los factores de certeza se basan en el juicio que tiene un experto sobre las ocurrencias de ciertas situaciones o relaciones cuyo conocimiento se desea incluir en una base de conocimientos.

Estas medidas de confianza o factores de certeza son evaluaciones o apreciaciones personales de los expertos que añaden al enunciado de su conocimiento.

La certeza (también conocida como certeza epistémica o certeza objetiva) es la propiedad epistémica de las creencias de las que una persona no tiene motivos racionales para dudar. La epistemología o teoría del conocimiento es la rama de la filosofía que se ocupa del conocimiento.

Una forma estándar de definir la certeza epistémica es que una creencia es cierta si y solo si la persona que sostiene esa creencia no puede estar equivocada al sostener esa creencia. Otras definiciones comunes de certeza involucran la naturaleza indudable de tales creencias o definen la certeza como una propiedad de esas creencias con la mayor justificación posible.

Es importante destacar que la certeza epistémica no es lo mismo que la certeza psicológica (también conocida como certeza subjetiva), que describe el grado más alto en el que una persona puede estar convencida de que algo es cierto. Si bien una persona puede estar completamente convencida de que una creencia en particular es verdadera, e incluso puede ser psicológicamente incapaz de admitir su falsedad, esto no implica que la creencia en sí misma esté más allá de toda duda racional o incapaz de ser falsa. Si bien la palabra "certeza" a veces se usa para referirse a la certeza subjetiva de una persona sobre la verdad de una creencia, los filósofos están interesados principalmente en la cuestión de si una creencia puede alcanzar la certeza objetiva. La cuestión filosófica de si uno puede estar verdaderamente seguro de algo ha sido ampliamente debatida durante siglos.

Un alto grado de certeza, comporta una conciencia sobre ciertos hechos que se admiten sin sombra de duda, con alta confianza en que lo sabido sobre tales hechos sería verdadero y válido. Basada en la evidencia supone un conocimiento comunicable y reconocible por cualquier otro entendimiento racional. Un bajo grado de certeza es una situación en la que el conocimiento estaría más o menos cerca de la ignorancia. En una situación así, no es posible afirmar algo con seguridad.

Ejemplo: Si se da A entonces se dará C *casi con toda seguridad*.

Los factores de certeza se expresan mediante un número o "factor de certeza".

Los factores de certeza no se rigen por probabilidad. No se obtienen de poblaciones muestrales, sino de experiencia.

En probabilidad la suma de la probabilidad de que se dé un hecho y su contrario es 1.

Un experto puede sentir que algo es cierto de forma importante, pero puede no saber cuanto de importante es lo contrario.

En muchas situaciones del "mundo real" existe incertidumbre, por ejemplo:

- Un sistema de diagnóstico médico no cuenta con toda la información del paciente

- Un robot tiene sensores con limitaciones y ruidosos
- Un agente financiero tiene información limitada de las empresas y no puede conocer todos los factores que las afectan

Un sistema inteligente debe poder tomar decisiones aunque no tenga toda la información o conocimiento necesarios, e incluso cuando existan errores en la información que recibe o en su conocimiento

Los sistemas inteligentes deben ser capaces de representar y razonar con incertidumbre. Existen varias causas de incertidumbre, las cuales tienen que ver con:

- la información: Incompleta, Poco confiable, Ruido, distorsión
- el conocimiento: Impreciso, Contradictorio y
- la representación: No adecuada, Falta de poder descriptivo

Se han desarrollado diversas técnicas para manejo de incertidumbre en sistemas inteligentes, entre las cuales cabe mencionar los sistemas expertos. Un sistema experto (SE) es un sistema informático que emula el razonamiento actuando tal y como lo haría un experto en cualquier área de conocimiento.

Los sistemas expertos son una de las aplicaciones de la inteligencia artificial que pretende simular el razonamiento humano, de la misma manera que lo haría un experto en un área de especialización.

Los primeros sistemas expertos se crearon en la década de 1970 y después proliferaron en los años 80.

Un sistema experto se divide en dos subsistemas: el motor de inferencia y la base de conocimiento. La base de conocimiento representa hechos y reglas. El motor de inferencia aplica las reglas a los hechos conocidos para deducir nuevos hechos. Los motores de inferencia también pueden incluir habilidades de explicación y depuración.

En los sistemas basados en reglas se tiene en general una estructura similar a la siguiente:

Si: se observa cierta evidencia E,

Entonces: se concluye cierta hipótesis H con probabilidad (certeza, ...) P

De aquí surgen varias interrogantes:

- ¿Cómo obtener estas medidas?
- ¿Cómo combinar estas medidas?
- ¿Cómo interpretar estas medidas?

La teoría de los factores de certeza se introdujo por primera vez en el sistema MYCIN (Buchanan y Shortliffe , 1979)

Mycin

Mycin es un sistema experto desarrollado a principios de los años 70 en la Universidad de Stanford. Su principal función consistía en el diagnóstico de enfermedades infecciosas de la sangre. Además, Mycin era capaz de "razonar" el proceso seguido para llegar a estos diagnósticos, y de recetar medicaciones personalizadas a cada paciente (según su estatura, peso, etc.).

El funcionamiento de Mycin se basaba principalmente en un sencillo motor de inferencia, que manejaba una base de conocimiento de aproximadamente unas 500 reglas. El programa capturaba las entradas a partir de una serie de preguntas (como por ejemplo, ¿Tiene el paciente molestias en el pecho?, o ¿Ha sido operado el paciente anteriormente?), que usualmente respondía el médico del paciente.

Tras este proceso, Mycin mostraba la salida por pantalla, que consistía en una serie de posibles enfermedades (ordenadas por su probabilidad asociada), la explicación del por qué de cada uno de estos diagnósticos, y una serie de recomendaciones sobre el tratamiento a seguir por el paciente.

Factores de Certeza

Para calcular la probabilidad de cada uno de los resultados, los autores desarrollaron una técnica empírica basada en factores de certeza. Estos factores de certeza se calculaban de tal manera que en función de unas evidencias se asigna a la hipótesis un factor de certeza.

MYCIN define un Factor de Certeza (CF) que se asocia a cada regla y cada evidencia, y se definen un conjunto de reglas para combinar estos factores.

Los CFs están en el rango de $[-1, +1]$, donde -1 denota que la evidencia es totalmente en contra de la hipótesis y $+1$ que es totalmente a favor

La forma típica de usar factores de certeza (por ejemplo, en MYCIN) es asociándolos a reglas de producción.

Ejemplo:

Si

la coloración del organismo es gram positivo y

la morfología del organismo es coco y

la forma de crecimiento del organismo es a base de cadenas

Entonces

hay una evidencia (0.7) de que la identidad del organismo sea un estreptococo.

donde 0.7 representa el factor de certeza asociado a la regla. El factor de certeza es un valor en el intervalo $[-1, 1]$, donde 1 indica completa confianza, y -1 completa no creencia.

En MYCIN era posible que dos o más reglas pudieran sacar conclusiones sobre un parámetro con diferentes pesos de evidencia.

Por ejemplo, una regla puede concluir que el organismo en cuestión es E. Coli con una certeza de 0,8 mientras que otra concluye que es E. Coli con una certeza de 0,5 o incluso -0,8. En el caso de que la certeza sea menor que cero, la evidencia en realidad está en contra de la hipótesis.

En un sistema que utiliza CF, las reglas deben estar estructuradas para que cualquier regla dada permita aumentar la creencia en una conclusión dada o en su detrimiento aumente la incredulidad (i.e. reduzca la creencia) correspondiente.

Para calcular el factor de certeza, MYCIN combinó estos pesos utilizando la fórmula que aparece a

$$CF(x,y) = \begin{cases} X + Y - XY & \text{if } X, Y > 0 \\ X + Y + XY & \text{if } X, Y < 0 \\ \frac{X+Y}{1-\min(|X|,|Y|)} & \text{otherwise} \end{cases}$$

Donde X e Y son los factores de certeza. Esta fórmula se puede aplicar más de una vez si más de dos reglas extraen conclusiones sobre el mismo parámetro. Es conmutativa, por lo que no importa en qué orden se combinaron los pesos.

Sean dos reglas R_1 y R_2 que alcanzan la misma conclusión h, a partir de dos evidencias e_1 y e_2 diferentes:

R_1 : Si e_1 entonces $h, CF(h, e_1)$

R_2 : Si e_2 entonces $h, CF(h, e_2)$

El factor de certeza de h se calculará como:

- $CF(h, e_1) + CF(h, e_2)(1 - CF(h, e_1))$, si $CF(h, e_1) > 0$ y $CF(h, e_2) > 0$
- $CF(h, e_1) + CF(h, e_2)(1 + CF(h, e_1))$, si $CF(h, e_1) < 0$ y $CF(h, e_2) < 0$
- $[CF(h, e_1) + CF(h, e_2)]/(1 - min(|CF(h, e_1)|, |CF(h, e_2)|)$,en cualquier otro caso

Ejemplo:

R_1 : Si viernes entonces trafico, 0.8

R_2 : Si llueve entonces tráfico, 0.7

Hoy es viernes y llueve ¿cuál es la certeza de tráfico?

(se refuerza la certeza con ambas positivas)

In [1]:

```
0.8 + 0.7*(1-0.8)
```

Out[1]:

```
0.94
```

Otro ejemplo:

R_1 : Si viernes entonces trafico, 0.8

R_2 : Si fin de mes entonces tráfico -0.4

Hoy es viernes y fin de mes ¿cuál es la certeza de tráfico? (disminuye la fuerza de la mayor)

In [5]:

```
(0.8 + (-0.4))/ (1 - min(0.8, 0.4))
```

Out[5]:

```
0.6666666666666667
```

Entre las reglas de combinación se encontraban:

1. Propagación (f_{prop}) o reglas en serie:

$$CF(h, e') = CF(h, e) \times max\{0, CF(e, e')\}$$

2. AND (conjunción), OR (disjunción) de evidencias (f_{and}, f_{or}):

$$CF(e_1 \ and \ e_2, e') = min\{CF(e_1, e'), CF(e_2, e')\}$$

$$CF(e_1 \ or \ e_2, e') = max\{CF(e_1, e'), CF(e_2, e')\}$$

Ejemplo: sean

- h_1: inversor_de_alto_riesgo
- h_2: inversor_de_bajo_riesgo
- e_1: joven
- e_2: ingresos_altos
- e_3: casado

Reglas:

- Si joven Entonces inversor_de_alto_riesgo, 0.6
- Si ingresos_altos Entonces inversor_de_alto_riesgo, 0.1
- Si casado Entonces inversor_de_alto_riesgo, -0.7

In [6]:

```
def reglas(evid):
    if evid == 'joven':
        return ('inversor_de_alto_riesgo', 0.6)
    elif evid == 'ingresos_altos':
        return ('inversor_de_alto_riesgo', 0.1)
    elif evid == 'casado':
        return ('inversor_de_alto_riesgo', -0.7)
```

Si se dan e_1 y e_2, se pueden usar las dos primeras reglas y ambas concluirán h_1. El valor de certeza de h_1 será: $CF(h, e_1 \wedge e_2) = 0.6 + 0.1(1 - 0.6) = 0.64$ (la 2^a regla aporta poco)

In [7]:

```
(h1, cf1) = reglas('joven')
(h1, cf1)
```

Out[7]:

```
('inversor_de_alto_riesgo', 0.6)
```

In [8]:

```
(h2, cf2) = reglas('ingresos_altos')
(h2, cf2)
```

Out[8]:

```
('inversor_de_alto_riesgo', 0.1)
```

In [14]:

```
# definimos una función para calcular el factor de certeza combinado de
# dos reglas que comparten la misma hipótesis

def factor_certeza(h, x, y):
    if x>0 and y>0:
        return (h, x+y-x*y)
    elif x<0 and y<0:
        return (h, x+y+x*y)
    else:
        return (h, (x+y)/(1-min(abs(x),abs(y))))
```

In [15]:

```
# como h1=h2 entonces
(h,cf) = factor_certeza(h1,cf1,cf2)
(h,cf)
```

Out[15]:

```
('inversor_de_alto_riesgo', 0.6399999999999999)
```

Lo cual confirma el cálculo manual.

Si además se tiene e_3, entonces se puede usar la tercera regla que también concluye h_1. Entonces su factor de certeza será:

$$CF(h, e_1 \wedge e_2 \wedge e_3) = \frac{[CF(h, e_1 \wedge e_2) + CF(h, e_3)]}{(1 - min(|CF(h, e_1 \wedge e_2)|, |CF(h, e_3)|)}$$
$$= \frac{[0.64 + (-0.7)]}{[1 - 0.64]} = -0.16$$

Si la persona es casada, se disminuye la certeza del perfil de alto riesgo

In [16]:

```
(h3, cf3) = reglas('casado')
(h3, cf3)
```

Out[16]:

```
('inversor_de_alto_riesgo', -0.7)
```

In [17]:

```
# de donde
(ht,cft) = factor_certeza(h1,cf,cf3)
(ht,cft)
```

Out[17]:

```
('inversor_de_alto_riesgo', -0.16666666666666677)
```

Un factor de certeza se puede asociar no sólo a una regla, sino que también se puede asociar a una condición de una regla. Sea

R_1: Si e_1, $CF(e_1)$ Entonces $h, CF_{R1}(h, e_1)$

donde:

> $CF_{R1}(h, e_1)$ es el factor de certeza de la regla R_1 y

> $CF(e_1)$ es el factor de certeza de e_1

¿cuál es la certeza de h con la evidencia e_1?

$$CF(h, e_1) = CF(e_1) * CF_R(h, e_1)$$

Si una regla tiene varias condiciones enlazadas con una conjunción (&) y cada una tiene asociado un factor de certeza:

R_1: Si e_1, $CF(e_1) \& \cdots \& e_n$, $CF(e_n)$ Entonces $h, CF_R(h, e_1 \wedge \cdots \wedge e_n)$

donde

> $CF_R(h, e_1 \wedge \cdots \wedge e_n)$ es el factor de certeza de la regla R_1

> $CF(e_1)$ es el factor de certeza de e_1 y

> $CF(e_n)$ es el factor de certeza de e_n

¿cuál es la certeza de h con las n evidencias presentes?

$$CF(h, e_1 \wedge \cdots \wedge e_n) = CF_R(h, e_1 \wedge \cdots \wedge e_n) * min[CF(e_1), \cdots, CF(e_n)]$$

Si una regla tiene varias condiciones enlazadas con una disyunción (|) y cada una tiene asociado un factor de certeza:

R_1: Si $e_1, CF(e_1) | \cdots | e_n, CF(e_n)$ Entonces $h, CF_R(h, e_1 \vee \cdots \vee e_n)$

> $CF_R(h, e_1 \vee \cdots \vee e_n)$ es el factor de certeza de la regla R_1

> $CF(e_1)$ es el factor de certeza de e_1 y

> $CF(e_n)$ es el factor de certeza de e_n

¿cuál es la certeza de h con las n evidencias presentes?

$$CF(h, e_1 \vee \cdots \vee e_n) = CF_R(h, e_1 \vee \cdots \vee e_n) * max[CF(e_1), \cdots, CF(e_n)]$$

Ejemplo:

R_1: Si $A, 0.7$ & $B, 0.2$ Entonces $C, -0.8$

R_2: Si $D, 0.3$ | $F, 0.6$ Entonces $C, -0.4$

Si se dan las evidencias A, B, D y F ¿Cuál es el factor de certeza de C?

$$CF(C, A \wedge B) = 0.2 * (-0.8) = -0.16$$

$$CF(C, D \vee F) = 0.6 * (-0.4) = -0.24$$

$$CF(C, (A \wedge B) \wedge (D \vee F)) = -0.16 - 0.24 * (1 - 0.16) = -0.33$$

In [2]:

```
import factores_certeza as fc

# R1 : Si (A,0.7) y (B,0.2) Entonces (C,-0.8)
def regla1(tups):
    if tups[0] == ('A',0.7) and tups[1] == ('B',0.2):
        return ('C',-0.8)
    else:
        return None
```

In [3]:

```
# R2: Si (D,0.3) o (F,0.6) Entonces (C,-0.4)
def regla2(tups):
    if tups[0] == ('D',0.3) and tups[1] == ('F',0.6):
        return ('C',-0.4)
    else:
        return None
```

In [4]:

```
def reglas(r,tups):
    if r == 1:
        return regla1(tups)
    if r == 2:
        return regla2(tups)
    return None
```

In [5]:

```
# Si se dan las evidencias A, B, D y F
evidencias = [('A',0.7), ('B',0.2), ('D',0.3), ('F',0.6)]
```

In [6]:

```
# ¿Cuál es el factor de certeza de C?
# si CF(C, A^B) = 0.2*(-0.8) = -0.16
def factor_certeza_and(r, h,levid):
    r1 = reglas(r,levid)
    if tuple(r1) and h == r1[0]: # ¿h=='C'?
        cfs = [cf for (_,cf) in levid]
        return (h,r1[1]*min(cfs)) # r1[1] es CFR(h,e1 and e2)
    else:
        return None

cfC1 = factor_certeza_and(1,'C',evidencias[:2])
cfC1
```

Out[6]:

```
('C', -0.16000000000000003)
```

In [7]:

```
# ¿Cuál es el factor de certeza de C?
# si CF(C, DvF) = 0.6*(-0.4) = -0.24
def factor_certeza_or(r,h,levid):
    r2 = reglas(r, levid)
    if tuple(r2) and h == r2[0]: # ¿h=='C'?
        cfs = [cf for (_,cf) in levid]
        return (h,r2[1]*max(cfs)) # r1[1] es CFR(h,e1 and e2)
    else:
        return None

cfC2 = factor_certeza_or(2,'C',evidencias[2:4])
cfC2
```

Out[7]:

```
('C', -0.24)
```

In [9]:

```
# ¿Cuál es el factor de certeza de C?
# si CF(C, (A^B)^(DvF)) = -0.16-0.24*(1-0.16) = -0.33
cfC3 = fc.factor_certeza2('C',cfC1[1], cfC2[1])
cfC3
```

Out[9]:

```
('C', -0.36160000000000003)
```

NOTA: se pueden comprobar los resultados obtenidos utilizando Excel

Ejemplo. Sean las reglas siguientes:

$R1 : e_1 \rightarrow h, CF_{R1} = 0.75$

$R2 : e_2 \rightarrow h, CF_{R2} = 0.6$

$R3 : e_3 \rightarrow h, CF_{R3} = 0.8$

Donde e_1, e_2, e_3 y h significan lo mismo que en caso anterior.

Sea otra regla adicional:

$R4 : h \rightarrow g, CF_{R4} = -0.7$ donde g: invertir_en_bonos

Supongamos que los factores de certeza de e_1, e_2, e_3 son $0.9, 0.5, 0.9$ respectivamente, de donde:

$R1 : (e1, 0.9)- > (h, CFR1 = 0.75)$

$R2 : (e2, 0.5)- > (h, CFR2 = 0.6)$

$R3 : (e3, 0.9)- > (h, CFR3 = -0.8)$

$R4 : (h, 1)- >$("invertir_en_bonos", $-0.7)$

Supongamos que observamos e_1, e_2, e_3

$$CF(h, e_1) = CF(e_1) * CF_{R1}(h, e_1) = 0.9 * 0.75 = 0.67$$
$$CF(h, e_2) = CF(e_2) * CF_{R2}(h, e_2) = 0.5 * 0.6 = 0.3$$
$$CF(h, e_3) = CF(e_3) * CF_{R3}(h, e_3) = 0.9 * (-0.8) = -0.72$$

In [11]:

```
hipotesis = ['inversor_de_alto_riesgo', 'inversor_de_bajo_riesgo',
             'invertir_en_bonos']
evidencias = [('joven',0.9), ('ingresos_altos',0.5), ('casado',0.9),
              ('inversor_de_alto_riesgo',1)]

def reglas(r,tups):
    if r==1 and tups == ('joven',0.9):
        return ('inversor_de_alto_riesgo', 0.75)
    elif r == 2 and tups == ('ingresos_altos',0.5):
        return ('inversor_de_alto_riesgo', 0.6)
    elif r == 3 and tups == ('casado',0.9):
        return ('inversor_de_alto_riesgo', -0.8)
    elif r == 4 and tups[0] == 'inversor_de_alto_riesgo':
        return ('invertir_en_bonos', -0.7)
    else:
        return None

# Un factor de certeza se puede asociar no sólo a una regla,
# sino que también se puede asociar a una condición de una regla
def factor_certeza_cond(r,h, levid):
    # print('factor_certeza_cond(r,h, levid)\n\t',r,h, levid)
    res = reglas(r,levid)
    # print('reglas(r,levid): ', r,levid, reglas(r,levid))
    if tuple(res) and h == res[0]:
        cfs = levid[1]
        return (h,res[1]*cfs)
    else:
        return None
```

In [13]:

```
h = 'inversor_de_alto_riesgo'
cfhe1 = factor_certeza_cond(1,h,evidencias[0])
print('CF(h,e1)=',cfhe1)
cfhe2 = factor_certeza_cond(2,h,evidencias[1])
print('CF(h,e2)=',cfhe2)
cfhe3 = factor_certeza_cond(3,h,evidencias[2])
print('CF(h,e3)=',cfhe3)
```

```
CF(h,e1)= ('inversor_de_alto_riesgo', 0.675)
CF(h,e2)= ('inversor_de_alto_riesgo', 0.3)
CF(h,e3)= ('inversor_de_alto_riesgo', -0.7200000000000001)
```

entonces

$$CF(h, e_1 \wedge e_2) = CF(h, e_1) + CF(h, e_2) * (1 - CF(h, e_1))$$

$$= 0.67 + 0.3 * (1 - 0.67) = 0.77$$

$$CF(h, e_1 \wedge e_2 \wedge e_3) = \frac{[CF(h,e_1 \wedge e_2)+CF(h,e_3)]}{(1-min(|CF(h,e_1 \wedge e_2)|,|CF(h,e_3)|)} = 0.77 + (-0.72))/(1 - 0.72) = 0.05/0.28 = 0.17$$

In [14]:

```
def factor_certeza2(h, cf_h_e1, cf_h_e2):
    if cf_h_e1>0 and cf_h_e2>0:
        return (h, cf_h_e1 + cf_h_e2 - cf_h_e1 * cf_h_e2)
    elif cf_h_e1<0 and cf_h_e2<0:
        return (h, cf_h_e1 + cf_h_e2 + cf_h_e1 * cf_h_e2)
    else:
        return (h, (cf_h_e1 + cf_h_e2) / (1 - min(abs(cf_h_e1), abs(cf_h_e2))))
```

In [16]:

```
cfhe1e2 = fc.factor_certeza2(h,cfhe1[1], cfhe2[1])
print('CF(h,e1&e2)=', cfhe1e2)
cfhe1e2e3 = fc.factor_certeza2(h,cfhe1e2[1], cfhe3[1])
print('CF(h,e1&e2&e3)=', cfhe1e2e3)
```

```
CF(h,e1&e2)= ('inversor_de_alto_riesgo', 0.7725000000000001)
CF(h,e1&e2&e3)= ('inversor_de_alto_riesgo', 0.18750000000000003)
```

En esta situación, habiendo calculado h a partir de las evidencias e1,e2,e3, podemos aplicar la regla 4:

> R4: (h,1) -> ("invertir_en_bonos",-0.7)

$$CF(g,h) = CF(h) * CF_{R4}(g,h) = 0.17 * (-0.7) = 0.12$$

(Conclusión: No debería invertir)

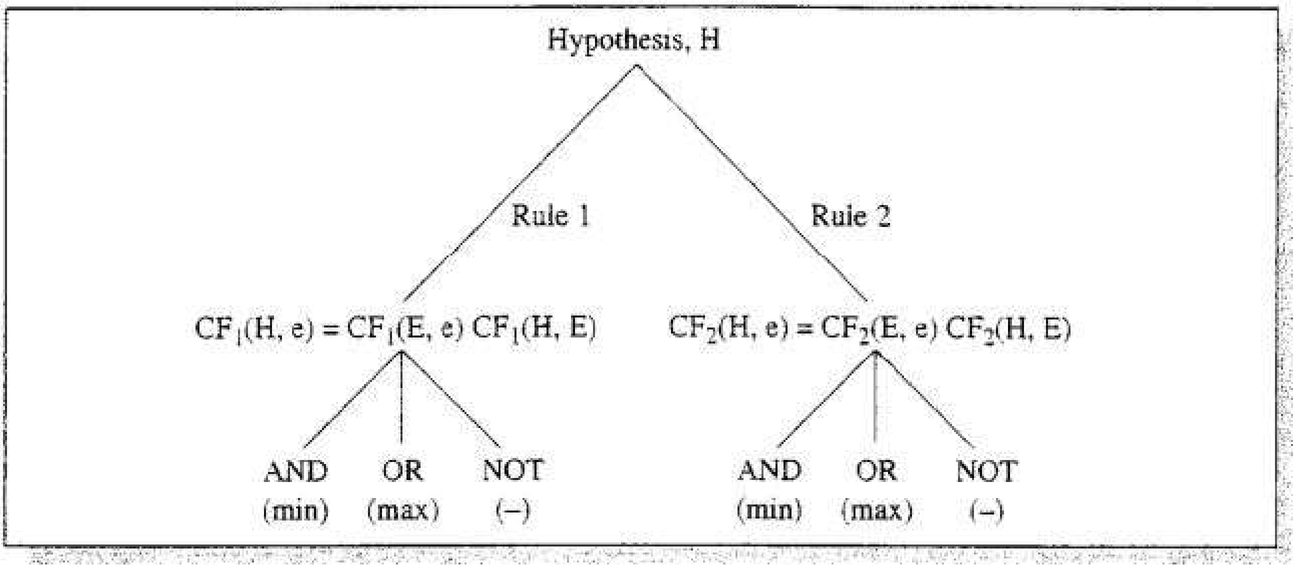
In [17]:

```
# por lo resultados previos tenemos que:
#   CF(h,e1&e2&e3)= ('inversor_de_alto_riesgo', 0.18750000000000003)
#   CFR4(g,h) = ("invertir_en_bonos",-0.7)
invertir = factor_certeza_cond(4,"invertir_en_bonos", cfhe1e2e3)
print('invertir?:', invertir)
```

```
invertir?: ('invertir_en_bonos', -0.13125)
```

NOTA: nuevamente la diferencia en los resultados se debe a la mayor precisión utilizada en este caso, lo cual se puede comprobar utilizando Excel para repetir los cálculos

CF of Two Rules with the Same Hypothesis Based on Uncertain Evidence



El modelo basado en factores de certeza:

- se ajusta bien al paradigma basado en reglas, no requiere cambios esenciales en la representación
- facil de implementar
- tuvo algún éxito práctico
- las reglas de propagación no parecen razonables a simple vista
- interpretación, significado de CF no es claro
- no hay teoria de la decision asociada a CF
- las reglas de propagacion no se pueden justificar

MYCIN provocó un debate sobre el uso de los factores de certeza (su marco de incertidumbre ad hoc). Los desarrolladores realizaron estudios que mostraron que el rendimiento de MYCIN se vio mínimamente afectado por las perturbaciones en las métricas de incertidumbre asociadas con las reglas individuales, lo que sugiere que el poder del sistema estaba más relacionado con su esquema de razonamiento y representación del conocimiento que con los detalles de su modelo de incertidumbre numérica.

Algunos observadores consideraron que se podían haber utilizado estadística bayesiana clásica. Los desarrolladores de MYCIN argumentaron que esto requeriría suposiciones poco realistas de independencia probabilística o requeriría que los expertos proporcionen estimaciones para un número inviablemente grande de probabilidades condicionales.

Las investigaciones realizadas por la Escuela de Medicina de Stanford, desvelaron que Mycin tuvo una tasa de aciertos de aproximadamente el 65, lo cual mejoraba las estadísticas de la mayoría de los médicos no especializados en el diagnóstico de infecciones bacterianas (dominio en el que Mycin estaba especializado), que ejercían la profesión en aquellos años. Los médicos que trabajaban exclusivamente en este campo conseguían una tasa del 80%.

Estudios posteriores demostraron más tarde que el modelo basado en el factor de certeza podría interpretarse en un sentido probabilístico y destacaron los problemas con los supuestos implícitos de dicho modelo. Sin embargo, la estructura modular del sistema resultó muy exitosa, lo que llevó al desarrollo de modelos gráficos como las redes bayesianas.

MYCIN nunca se usó en la práctica, ya que algunos observadores plantearon cuestiones éticas y legales relacionadas con el uso de computadoras en medicina, en cuanto a la responsabilidad de los médicos en caso de que el sistema diera un diagnóstico erróneo: si Mycin se equivocaba en algún diagnóstico, ¿quién asumía la culpa, el programador o el médico?

Otro de los motivos se achaca a la excesiva dificultad que suponía el mantenimiento del programa. Era este uno de los principales problemas de Mycin, y en general, de los sistemas expertos de la época, en los cuales se dedicaban muchos esfuerzos y recursos a extraer el conocimiento necesario de los expertos en dominio para construir el motor de inferencia.

Bibliografía

- Métodos de Inteligencia Artificial (https://ccc.inaoep.mx/~esucar/Clases-MetIA/MetIA-09.pdf)
- Mycin (https://es.wikipedia.org/wiki/Mycin)
- Mycin (https://en.wikipedia.org/wiki/Mycin)
- Sistema experto (https://es.wikipedia.org/wiki/Sistema_experto)

In [ ]:

```

```