

Ejemplo de uso de factores de Certeza

Juan Luis Castro

Objetivo

Implementar en CLISP un sistema que haga un diagnóstico sobre la avería de coche que no arranca de acuerdo a las siguientes reglas:

- R1:** SI el motor obtiene gasolina Y el motor gira ENTONCES problemas con las bujías con certeza 0,7
- R2:** SI NO gira el motor ENTONCES problema con el starter con certeza 0,8
- R3:** SI NO encienden las luces ENTONCES problemas con la batería con certeza 0,9
- R4:** SI hay gasolina en el deposito ENTONCES el motor obtiene gasolina con certeza 0,9
- R5:** SI hace intentos de arrancar ENTONCES problema con el starter con certeza -0,6
- R6:** SI hace intentos de arrancar ENTONCES problema con la batería 0,5

Representación

; (FactorCerteza ?h si|no ?f) representa que ?h se ha deducido con factor de certeza ?f

;?h podrá_ser:

- problema_starter
- problema_bujias
- problema_batería
- motor_llega_gasolina

; (Evidencia ?e si|no) representa el hecho de si evidencia ?e se da

; ?e podrá ser:

- hace_intentos_arrancar
- hay_gasolina_en_deposito
- encienden_las_luces
- gira_motor

Reglas

;;; convertimos cada evidencia en una afirmación sobre su factor de certeza

```
(defrule certeza_evidencias
```

```
(Evidencia ?e ?r)
```

```
=>
```

```
(assert (FactorCerteza ?e ?r 1)) )
```

;; También podríamos considerar evidencias con una cierta

;;incertidumbre: al preguntar por la evidencia, pedir y recoger

;;directamente el grado de certeza

Traducir reglas (ponemos un ejemplo)

;R1: SI el motor obtiene gasolina Y el motor gira ENTONCES problemas ;
con las bujías con certeza 0,7

```
(defrule R1  
  (FactorCerteza motor_llega_gasolina si ?f1)  
  (FactorCerteza gira_motor si ?f2)  
  (test (and (> ?f1 0) (> ?f2 0)))  
  =>  
  (assert (FactorCerteza problema_bujias si (encadenado (* ?f1 ?f2) 0,7))))
```

Función encadenado

```
(deffunction encadenado (?fc_antecedente ?fc_regla)
  (if (> ?fc_antecedente 0)
    then
      (bind ?rv (* ?fc_antecedente ?fc_regla))
    else
      (bind ?rv 0) )
  ?rv)
```

Combinar distintas deducciones

;;;;;; Combinar misma deduccion por distintos caminos

```
(defrule combinar
```

```
(declare (salience 1))
```

```
?f <- (FactorCerteza ?h ?r ?fc1)
```

```
?g <- (FactorCerteza ?h ?r ?fc2)
```

```
(test (neq ?fc1 ?fc2))
```

```
=>
```

```
(retract ?f ?g)
```

```
(assert (FactorCerteza ?h ?r (combinacion ?fc1 ?fc2))) )
```

Función Combinación

```
(deffunction combinacion (?fc1 ?fc2)
  (if (and (> ?fc1 0) (> ?fc2 0) )
    then
      (bind ?rv (- (+ ?fc1 ?fc2) (* ?fc1 ?fc2) ) )
    else
      (if (and (< ?fc1 0) (< ?fc2 0) )
        then
          (bind ?rv (+ (+ ?fc1 ?fc2) (* ?fc1 ?fc2) ) )
        else
          (bind ?rv (/ (+ ?fc1 ?fc2) (- 1 (min (abs ?fc1) (abs ?fc2)))) )
      )
  )
  ?rv)
```


Certeza de las hipótesis

- Aunque en este ejemplo no se da, puede ocurrir que tengamos deducciones de hipótesis en positivo y negativo que hay que combinar para compararlas

```
(defrule combinar_signo
```

```
(declare (salience 2))
```

```
(FactorCerteza ?h si ?fc1)
```

```
(FactorCerteza ?h no ?fc2)
```

```
=>
```

```
(assert (Certeza ?h (- ?fc1 ?fc2))) )
```

Ejercicio: Acabar el ejemplo

Hay que:

- preguntar por las posibles evidencias,
- añadir el resto de las reglas, y
- tras razonar quedarse con la hipótesis con mayor certeza
- Añadir o modificar las reglas para que el sistema explique el por qué de las afirmaciones