

Sistemas Basados en Reglas

Sergio García Prado

11 de diciembre de 2016

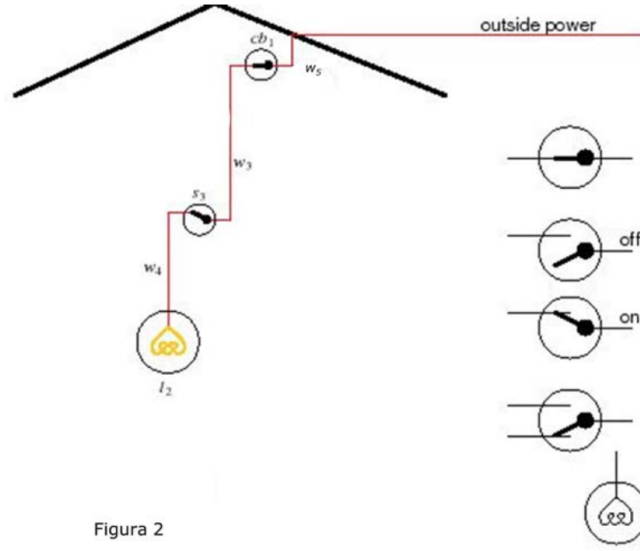


Figura 2

I. DESARROLLAR UNA BASE DE CONOCIMIENTO PARA LA VERSIÓN REDUCIDA DEL ASISTENTE AL DIAGNÓSTICO QUE MUESTRA LA FIGURA. UTILIZAR UN LENGUAJE DE TRIPLETES O-A-V, PERMITIENDO EL USO DE VARIABLES EN LAS REGLAS PARA LOS OBJETOS Y LOS VALORES

I. Ontología General

La base de conocimiento necesaria para representar el problema requiere de un conjunto tanto de objetos como de atributos de los mismos. Mientras que el conjunto de objetos se ha referenciado de forma particular, la declaración de atributos (DA) se ha dejado especificada de manera general para poder ser aprovechada para su reutilización en otros problemas utilizando subíndices:

$$O = \{l_2, s_3, cb_1, w_3, w_4, w_5, outside\}$$

$$DA = \{ \begin{array}{l} l_i.live^s : boolean, l_i.type^s, l_i.lit^s : boolean, l_i.ok^s : boolean, \\ s_j.live^s : boolean, s_j.type^s, s_j.state^s, s_j.connected^m, s_j.ok^s : boolean, \\ cb_k.live^s : boolean, cb_k.type^s, cb_k.ok^s : boolean, cb_k.connected^s, \\ w_l.live^s : boolean, w_l.type^s, w_l.ok^s : boolean, w_l.connected^s, \\ outside.live^s : boolean, outside.ok^s : boolean, outside.connected^s \end{array} \}$$

$$DD = O \cup DA$$

A continuación se muestran el conjunto de reglas necesarias para llevar a cabo la producción de conocimiento y por lo tanto, la modificación de la memoria de trabajo (MT), que en un estado inicial será inicializada a partir del conjunto de hechos indicados en la Ontología Específica, tal y como se describirá próximamente.

1. **if** *equals*(?x, *connected*, ?y) **and** *equals*(?x, *live*, t) **and** *equals*(?x, *ok*, t)
then *add*(?y, *live*, t)
2. **if** *equals*(?x, *type*, *lighth*) **and** *equals*(?x, *live*, t) **and** *equals*(?x, *ok*, t)
then *add*(?x, *lit*, t)
3. **if** *equals*(?x, *type*, *switch*) **and** *equals*(?x, *live*, t) **and** *equals*(?x, *state*, ?y)
and *equals*(?x, *connected*^{?y}, ?z)
then *add*(?z, *live*, t)

II. Ontología Específica

$$\begin{aligned} l_2.type &= lighth, l_2.ok = true, \\ s_3.type &= switch, s_j.state = up, s_j.connected^{up} = w_4, s_j.ok = true, \\ cb_1.type &= circuitbreaker, cb_1.ok = true, cb_1.connected = w_3, \\ w_3.type &= wire, w_l.ok = true, w_l.connected = s_3, \\ w_4.type &= wire, w_l.ok = true, w_l.connected = l_2, \\ w_5.type &= wire, w_l.ok = true, w_l.connected = cb_1, \\ outside.live &= true, outside.ok = true, outside.connected = w_5 \end{aligned}$$

II. OBTENER LA RED RETE QUE GENERA EL SIGUIENTE CONJUNTO DE REGLAS

```
R1:  if iguales(bombilla, ?x, t) and iguales(ok, ?x, t) and iguales(tension, ?x, t) then
añadir(luce, ?x, t) fi
R2:  if iguales(bombilla, ?x, t) and iguales(ok, ?x, f) then añadir(luce, ?x, f) fi
R3:  if iguales(bombilla, ?x, t) and iguales(ok, ?x, t) and iguales(tension, ?x, f) then
añadir(luce, ?x, f) fi
```

La estrategia que se pretende lograr mediante la utilización de una red generada por el algoritmo RETE es generar una máquina de estados que reduzca el número de comparaciones realizadas para disparar una regla debido a la repetición de partes del antecedente de las mismas. Por lo tanto, el diagrama de la red RETE que surge de la base de conocimiento descrita en el enunciado es la siguiente:

