

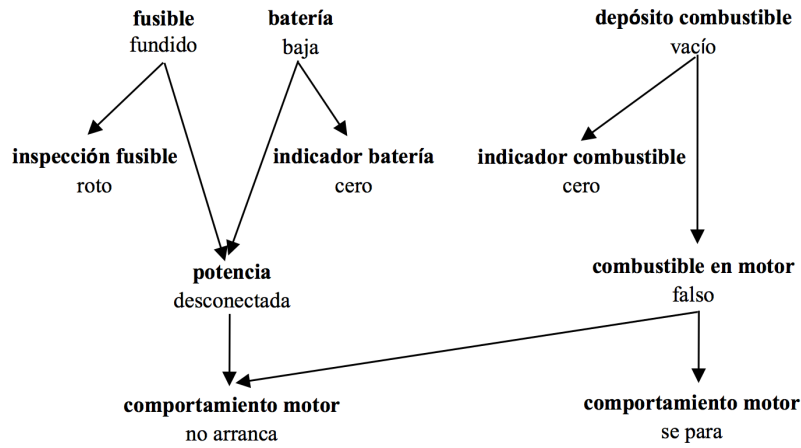
Clips 1

Fernández Angulo, Óscar

García Prado, Sergio

13 de diciembre de 2016

III. LA FIGURA MUESTRA UN FRAGMENTO DE UNA RED CAUSAL QUE MODELA CONOCIMIENTO DEL DOMINIO PARA LA TAREA DE DIAGNOSIS EN EL DOMINIO DE LOS AUTOMÓVILES. LA RED ASOCIA POSIBLES CAUSAS DE FALLO (FUSIBLE FUNDIDO, BATERÍA BAJA O DEPÓSITO DE COMBUSTIBLE VACÍO) CON ESTADOS INTERMEDIOS (POTENCIA, COMBUSTIBLE EN MOTOR Y SÍNTOMAS COMPORTAMIENTO MOTOR, INSPECCIÓN FUSIBLE, INDICADOR BATERÍA...).



La base de conocimiento necesaria para representar el problema requiere de un conjunto tanto de objetos como de atributos de los mismos. Esto se describe a continuación a partir de la Definición del Dominio (DD), el cual se compone del conjunto de Objetos (O) y la Definición de Atributos (DA):

$$O = \{\text{coche}, \text{motor}, \text{potencia}, \text{batería}, \text{fusible}, \text{combustible}, \text{deposito}\}$$

$$DA = \{$$

$$\begin{aligned}
 & \text{coche.sePara}^s : \text{boolean} \\
 & \text{motor.arranca}^s : \text{boolean}, \text{motor.combustible}^s : \text{boolean}, \\
 & \text{potencia.conectada}^s : \text{boolean} \\
 & \text{batería.indicador}^s : \text{number}, \text{batería.estado}^s, \\
 & \text{fusible.inspeccion}^s, \text{fusible.estado}^s, \\
 & \text{combustible.indicador}^s : \text{number}, \\
 & \text{deposito.estado}^s
 \end{aligned}$$

}

A continuación se describe el conjunto de reglas de la base de conocimiento:

R1: *if equals(motor, arranca, false)*
then add(potencia, conectada, false) fi

En el caso de que el motor no arranque, significa que no está conectada la potencia.

R2: *if equals(motor, arranca, false)*
then add(motor, combustible, false) fi

Si el motor no arranca otra de las posibles causas es que no tenga combustible.

R3: *if equals(coche, sePara, true)*
then add(motor, combustible, false) fi

La causa de que el coche se haya parado viene dada por la falta de combustible en el motor.

R4: *if equals(potencia, conectada, false) and equals(fusible, inspeccion, roto)*
then add(fusible, estado, fundido) fi

Si la potencia no está conecta y la inspección del fusible determina que este está roto entonces se puede determinar que el estado del fusible es fundido.

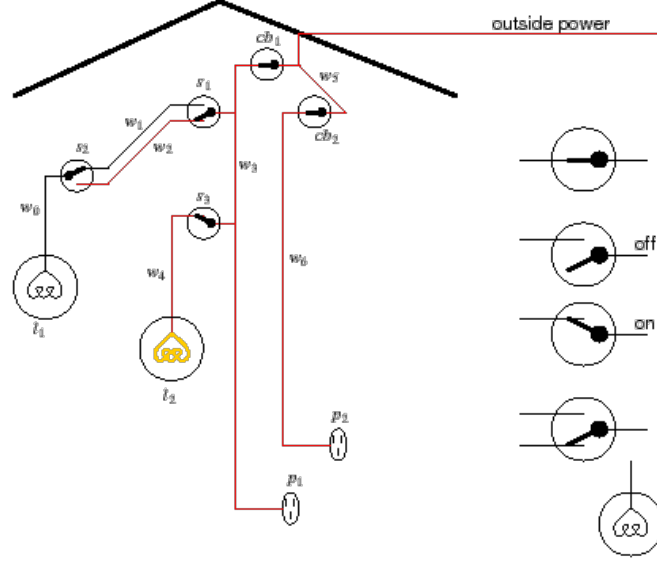
R5: *if equals(potencia, conectada, false) and equals(bateria, indicador, 0)*
then add(bateria, estado, baja) fi

Si la potencia no está conectada y la batería indica el valor cero entonces el estado de la batería es baja.

R6: *if equals(motor, combustible, false) and equals(combustible, indicador, 0)*
then add(deposito, estado, vacio) fi

Si no llega combustible al motor y además el indicador de combustible indica el valor cero entonces el resultado es que el depósito está vacío.

IV. CONSIDERAR EL ASISTENTE AL DIAGNOSTICO PROPUESTO POR POOL Y MACKWORTH



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:

$$O = \{l_1, l_2, p_1, p_2, s_1, s_2, s_3, cb_1, cb_2, w_1, w_2, w_3, w_4, w_5, w_6, outside\}$$

$$DA = \{$$

$$\begin{aligned}
 & l_i.live^s : boolean, l_i.type^s, l_i.lit^s : boolean, l_i.ok^s : boolean, \\
 & p_i.live^s : boolean, p_i.type^s, p_i.electrize^s : boolean, p_i.ok^s : boolean, \\
 & s_j.live^s : boolean, s_j.type^s, s_j.state^s : up, down, s_j.up^s, s_j.down^s, s_j.ok^s : boolean, \\
 & cb_k.live^s : boolean, cb_k.type^s, cb_k.ok^s : boolean, cb_k.connected^s, \\
 & w_l.live^s : boolean, w_l.type^s, w_l.ok^s : boolean, w_l.connected^m, \\
 & outside.live^s : boolean, outside.ok^s : boolean, outside.connected^s
 \end{aligned}$$

$$\}$$

$$DD = O \cup DA$$

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.

- R1:** if *equals*(?x, *connected*, ?y) and *equals*(?x, *live*, t) and *equals*(?x, *ok*, t)
 then *add*(?y, *live*, t) fi
- R2:** if *equals*(?x, *type*, *ligh*) and *equals*(?x, *live*, t) and *equals*(?x, *ok*, t)
 then *add*(?x, *lit*, t) fi
- R3:** if *equals*(?x, *type*, *power_outlet*) and *equals*(?x, *live*, t) and *equals*(?x, *ok*, t)
 then *add*(?x, *electrize*, t) fi
- R4:** if *equals*(?x, *type*, *switch*) and *equals*(?x, *live*, t) and *equals*(?x, *state*, *up*)
 and *equals*(?x, *up*, ?y)
 then *add*(?y, *live*, t) fi
- R5:** if *equals*(?x, *type*, *switch*) and *equals*(?x, *live*, t) and *equals*(?x, *state*, *down*)
 and *equals*(?x, *down*, ?y)
 then *add*(?y, *live*, t) fi

II. Ontología Específica

*l*₁.*type* = *ligh*, *l*₁.*ok* = *true*,
*l*₂.*type* = *ligh*, *l*₂.*ok* = *true*,
*p*₁.*type* = *power_outlet*, *p*₁.*ok* = *true*,
*p*₂.*type* = *power_outlet*, *p*₂.*ok* = *true*,
*s*₁.*type* = *switch*, *s*₁.*state* = *down*, *s*₁.*up* = *w*₁, *s*₁.*down* = *w*₂, *s*₁.*ok* = *true*,
*s*₂.*type* = *switch*, *s*₂.*state* = *up*, *s*₂.*up* = *w*₁, *s*₂.*down* = *w*₂, *s*₂.*ok* = *true*,
*s*₃.*type* = *switch*, *s*₃.*state* = *up*, *s*₃.*up* = *w*₄, *s*₃.*ok* = *true*,
*cb*₁.*type* = *circuit_breaker*, *cb*₁.*connected* = *w*₃, *cb*₁.*ok* = *true*,
*cb*₂.*type* = *circuit_breaker*, *cb*₂.*connected* = *w*₆, *cb*₂.*ok* = *true*,
*w*₀.*type* = *wire*, *w*₁.*ok* = *true*, *w*₀.*connected* = *l*₁,
*w*₁.*type* = *wire*, *w*₁.*ok* = *true*,
*w*₂.*type* = *wire*, *w*₂.*ok* = *true*,
*w*₃.*type* = *wire*, *w*₃.*ok* = *true*, *w*₃.*connected* = {*s*₁, *s*₃, *p*₁},
*w*₄.*type* = *wire*, *w*₄.*ok* = *true*, *w*₄.*connected* = *l*₂,
*w*₅.*type* = *wire*, *w*₅.*ok* = *true*, *w*₅.*connected* = {*cb*₁, *cb*₂},
*w*₆.*type* = *wire*, *w*₆.*ok* = *true*, *w*₆.*connected* = *p*₂,
outside.*live* = *true*, *outside*.*ok* = *true*, *outside*.*connected* = *w*₅