

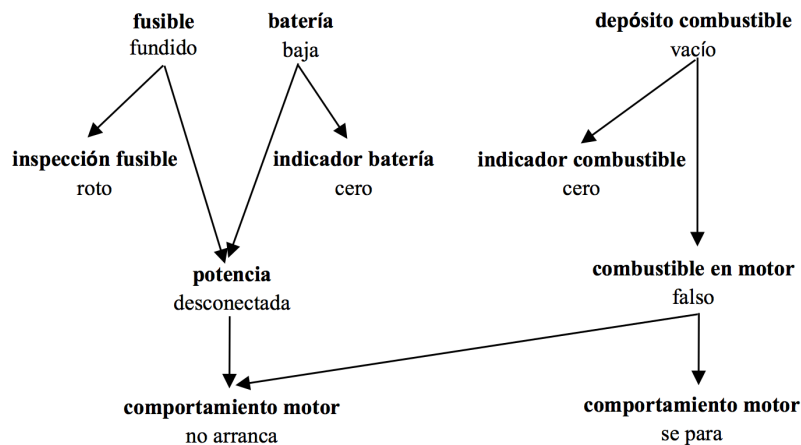
# Clips 1

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...) SE PUEDE OBSERVAR QUE LA RED REFLEJA LA DIRECCIÓN CAUSAL: LA CAUSA “DEPÓSITO DE COMBUSTIBLE VACÍO” TIENE COMO EFECTO “COMBUSTIBLE EN MOTOR FALSO” QUE A SU VEZ ES CAUSA DE “COMPORTAMIENTO MOTOR SE PARA”.



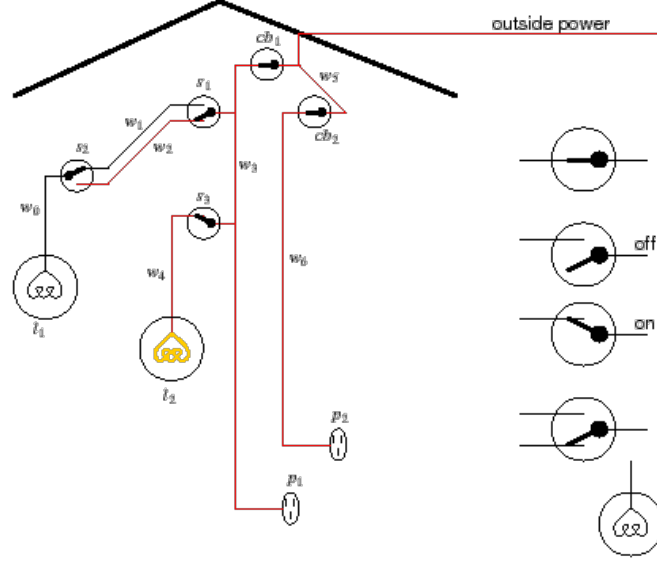
La base de conocimiento necesaria para representar el problema requiere de un conjunto tanto de objetos como de atributos de los mismos. Esto se describe a continuación a partir de la Definición del Dominio (DD) y el conjunto de reglas:

$$O = \{comportamientoMotor, potencia, combustibleEnMotor, fusible, bateria, depositoCombustible, solucion\}$$

$DA = \{$   
    *comportamientoMotor.observacion*<sup>s</sup> : {noArranca, sePara, correcto}  
    *potencia.estado*<sup>s</sup> : {desconectada, conectada}  
    *combustibleEnMotor.estado*<sup>s</sup> : {f, t}  
    *fusible.observacion*<sup>s</sup> : {fundido, correcto}, *fusible.estado*<sup>s</sup> : {fundido, correcto}  
    *bateria.observacion*<sup>s</sup> : number, *bateria.estado*<sup>s</sup> : {baja, normal}  
    *depositoCombustible.observacion*<sup>s</sup> : number, *depositoCombustible.estado*<sup>s</sup> : {vacio, noVacio}  
    *solucion.causa*<sup>s</sup> : {fusibleFundido, bateriaBaja, sinCombustible}  
 $\}$

- R1:** if equals(*comportamientoMotor*, *observacion*, noArranca)  
    then add(*potencia*, *estado*, desconectada) fi
- R2:** if equals(*comportamientoMotor*, *observacion*, noArranca)  
    then add(*combustibleEnMotor*, *estado*, f) fi
- R3:** if equals(*comportamientoMotor*, *observacion*, sePara)  
    then add(*combustibleEnMotor*, *estado*, f) fi
- R4:** if equals(*potencia*, *estado*, desconectada) and equals(*fusible*, *observacion*, fundido)  
    then add(*fusible*, *estado*, fundido) fi
- R5:** if equals(*potencia*, *estado*, desconectada) and equals(*bateria*, *observacion*, cero)  
    then add(*bateria*, *estado*, baja) fi
- R6:** if equals(*combustibleEnMotor*, *estado*, f) and equals(*depositoCombustible*, *observacion*, cero)  
    then add(*depositoCombustible*, *estado*, vacio) fi
- R7:** if equals(*fusible*, *estado*, fundido)  
    then add(*solucion*, *causa*, fusibleFundido) fi
- R8:** if equals(*bateria*, *estado*, baja)  
    then add(*solucion*, *causa*, bateriaBaja) fi
- R9:** if equals(*depositoCombustible*, *estado*, vacio)  
    then add(*solucion*, *causa*, sinCombustible) fi

#### 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 Especifica, 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*<sub>1</sub>.*type* = *ligh*, *l*<sub>1</sub>.*ok* = *true*,  
*l*<sub>2</sub>.*type* = *ligh*, *l*<sub>2</sub>.*ok* = *true*,  
*p*<sub>1</sub>.*type* = *power\_outlet*, *p*<sub>1</sub>.*ok* = *true*,  
*p*<sub>2</sub>.*type* = *power\_outlet*, *p*<sub>2</sub>.*ok* = *true*,  
*s*<sub>1</sub>.*type* = *switch*, *s*<sub>1</sub>.*state* = *down*, *s*<sub>1</sub>.*up* = *w*<sub>1</sub>, *s*<sub>1</sub>.*down* = *w*<sub>2</sub>, *s*<sub>1</sub>.*ok* = *true*,  
*s*<sub>2</sub>.*type* = *switch*, *s*<sub>2</sub>.*state* = *up*, *s*<sub>2</sub>.*up* = *w*<sub>1</sub>, *s*<sub>2</sub>.*down* = *w*<sub>2</sub>, *s*<sub>2</sub>.*ok* = *true*,  
*s*<sub>3</sub>.*type* = *switch*, *s*<sub>3</sub>.*state* = *up*, *s*<sub>3</sub>.*up* = *w*<sub>4</sub>, *s*<sub>3</sub>.*ok* = *true*,  
*cb*<sub>1</sub>.*type* = *circuit\_breaker*, *cb*<sub>1</sub>.*connected* = *w*<sub>3</sub>, *cb*<sub>1</sub>.*ok* = *true*,  
*cb*<sub>2</sub>.*type* = *circuit\_breaker*, *cb*<sub>2</sub>.*connected* = *w*<sub>6</sub>, *cb*<sub>2</sub>.*ok* = *true*,  
*w*<sub>0</sub>.*type* = *wire*, *w*<sub>1</sub>.*ok* = *true*, *w*<sub>0</sub>.*connected* = *l*<sub>1</sub>,  
*w*<sub>1</sub>.*type* = *wire*, *w*<sub>1</sub>.*ok* = *true*,  
*w*<sub>2</sub>.*type* = *wire*, *w*<sub>2</sub>.*ok* = *true*,  
*w*<sub>3</sub>.*type* = *wire*, *w*<sub>3</sub>.*ok* = *true*, *w*<sub>3</sub>.*connected* = {*s*<sub>1</sub>, *s*<sub>3</sub>, *p*<sub>1</sub>},  
*w*<sub>4</sub>.*type* = *wire*, *w*<sub>4</sub>.*ok* = *true*, *w*<sub>4</sub>.*connected* = *l*<sub>2</sub>,  
*w*<sub>5</sub>.*type* = *wire*, *w*<sub>5</sub>.*ok* = *true*, *w*<sub>5</sub>.*connected* = {*cb*<sub>1</sub>, *cb*<sub>2</sub>},  
*w*<sub>6</sub>.*type* = *wire*, *w*<sub>6</sub>.*ok* = *true*, *w*<sub>6</sub>.*connected* = *p*<sub>2</sub>,  
*outside*.*live* = *true*, *outside*.*ok* = *true*, *outside*.*connected* = *w*<sub>5</sub>