

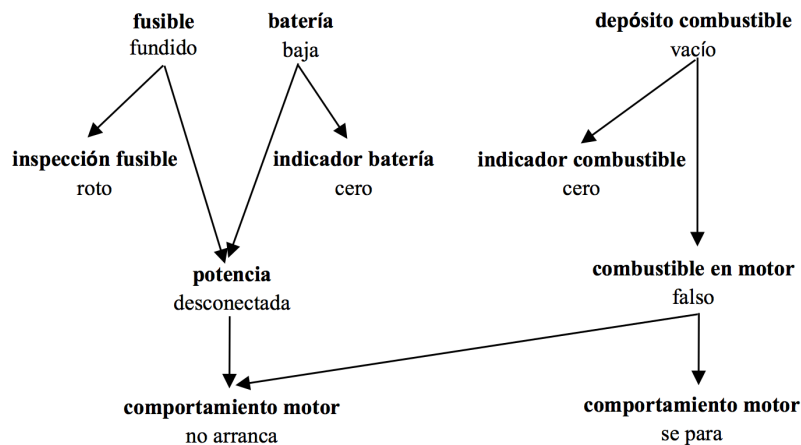
# Clips 1

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* : {*noArranca, sePara, correcto*}  
     *potencia.estado* : {*desconectada, conectada*}  
     *combustibleEnMotor.estado* : {*f, t*}  
     *fusible.observacion* : {*fundido, correcto*}, *fusible.estado* : {*fundido, correcto*}  
     *bateria.observacion* : *number*, *bateria.estado* : {*baja, normal*}  
     *depositoCombustible.observacion* : *number*, *depositoCombustible.estado* : {*vacio, noVacio*}  
     *solucion.causa* : {*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**



**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*, *lighth*) 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*, ?y)  
 and *equals*(?x, *connected*<sup>?y</sup>, ?z)  
 then *add*(?z, *live*, t) fi

## II. Ontología Específica

*l*<sub>1</sub>.*type* = *lighth*, *l*<sub>1</sub>.*ok* = *true*,  
*l*<sub>2</sub>.*type* = *lighth*, *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>.*connected*<sup>up</sup> = *w*<sub>1</sub>, *s*<sub>1</sub>.*connected*<sup>down</sup> = *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>.*connected*<sup>up</sup> = *w*<sub>1</sub>, *s*<sub>2</sub>.*connected*<sup>down</sup> = *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>.*connected*<sup>up</sup> = *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>