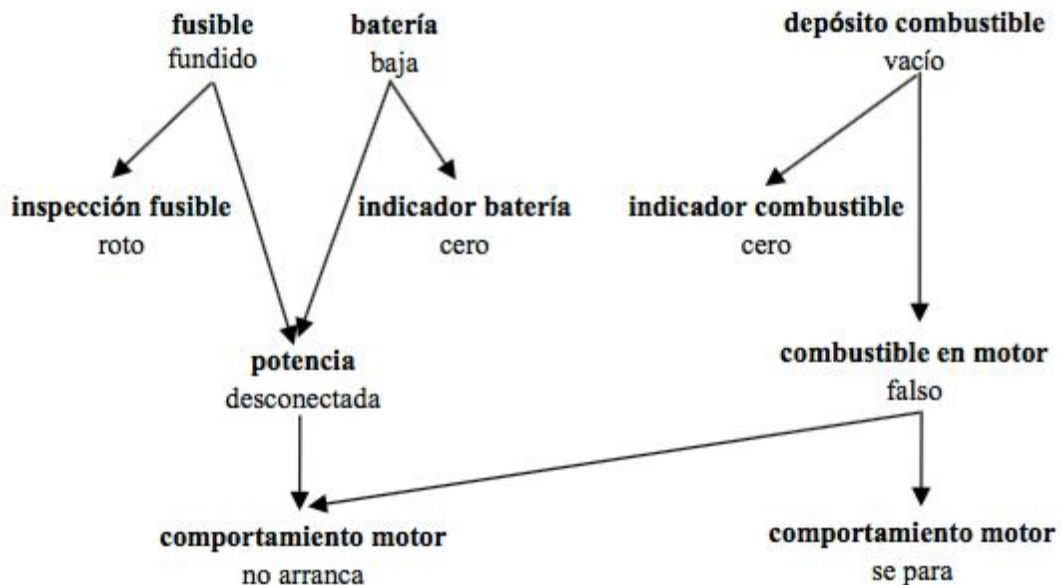


# INGENIERÍA DEL CONOCIMIENTO

## PRÁCTICA CLIPS I

Diciembre de 2015 - Jairo Velasco Martín

### Problema 3.



### Declaración de dominio:

$O = \{ \text{fusible, batería, deposito_combustible, inspeccion_fusible, indicador_bateria, indicador_combustible, potencia, combustible_motor, comportamiento_motor} \}$

$DA = \{ \text{comportamiento\_motor.sintoma}^s : \{ \text{se\_para, no\_arranca} \},$   
 $\text{combustible\_motor.causa}^s : \text{boolean},$   
 $\text{potencia.causa}^s : \{ \text{conectada, desconectada} \},$   
 $\text{inspeccion\_fusible.ok}^s : \{ \text{funcional, roto} \},$   
 $\text{indicador\_bateria.ok}^s : \{ \text{cero, no\_cero} \},$   
 $\text{indicador\_combustible.ok}^s : \{ \text{cero, no\_cero} \},$   
 $\text{deposito\_combustible.estado}^s : \{ \text{vacío, no\_vacío} \},$   
 $\text{bateria.estado}^s : \{ \text{baja, no\_baja} \},$   
 $\text{fusible.estado}^s : \{ \text{fundido, no\_fundido} \} \}$

$DD = O \cup DA$

*Nota: Aunque para cada atributo pueden reducirse sus posibles valores a booleanos, es más claro el diagnóstico de la avería al incluir palabras como roto, vacío o fundido en lugar de especificar simplemente false.*

### Reglas:

```
if    iguales(comportamiento_motor, sintoma, se_para)
then  añadir(combustible_motor, causa, falso)
fi
```

```
if    iguales(comportamiento_motor, sintoma, no_arranca)
then  añadir(combustible_motor, causa, falso)
      añadir(potencia, causa, desconectada)
fi
```

```
if    iguales(combustible_motor, causa, falso) and
      iguales(indicador_combustible, ok, cero)
then  añadir(deposito_combustible, estado, vacio)
fi
```

```
if    iguales(potencia, causa, desconectada) and
      iguales(indicador_bateria, ok, cero)
then  añadir(bateria, estado, baja)
fi
```

```
if    iguales(potencia, causa, desconectada) and
      iguales(inspeccion_fusible, ok, roto)
then  añadir(fusible, estado, fundido)
fi
```

### Hechos iniciales:

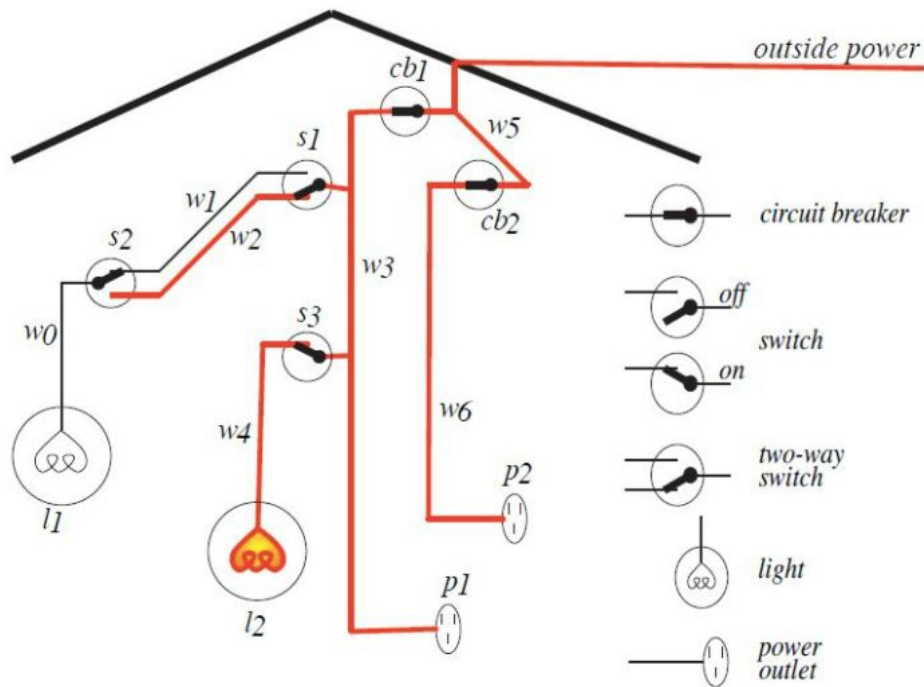
#### Ejemplo 1

```
comportamiento_motor.sintoma = {no_arranca}
indicador_bateria.ok = {cero}
```

#### Ejemplo 2

```
comportamiento_motor.sintoma = {se_para}
indicador_combustible.ok = {cero}
```

#### Problema 4.



#### Declaración de dominio:

$O = \{\text{outside}, w5, cb1, cb2, w6, p2, w3, s3, w4, l2, p1, s1, w1, w2, s2, w0, l1\}$

DA = {outside.live<sup>s</sup> : boolean,  
w6.conectado<sup>s</sup> ,  
w6.live<sup>s</sup> : boolean,  
w5.conectado<sup>s</sup> ,  
w5.live<sup>s</sup> : boolean,  
w4.conectado<sup>s</sup> ,  
w4.live<sup>s</sup> : boolean,  
w3.conectado<sup>s</sup> ,  
w3.live<sup>s</sup> : boolean,  
w2.conectado<sup>s</sup> ,  
w2.live<sup>s</sup> : boolean,  
w1.conectado<sup>s</sup> ,  
w1.live<sup>s</sup> : boolean,  
w0.conectado<sup>s</sup> ,  
w0.live<sup>s</sup> : boolean,  
cb1.ok<sup>s</sup> : boolean,  
cb2.ok<sup>s</sup> : boolean,  
s1.estado<sup>s</sup> : {up, down} ,  
s1.ok<sup>s</sup> : boolean,

```

s2.estados : {up, down} ,
s2.oks : boolean,
s3.estados : {up, down} ,
s3.oks : boolean,
l1.lives : boolean,
l1.lights : boolean,
l1.lits : boolean,
l1.oks : boolean,
l1.conectados
l2.lives : boolean,
l2.lights : boolean,
l2.lits : boolean,
l2.oks : boolean,
l2.conectados
p1.conectados ,
p1.lives : boolean,
p1.oks : boolean,
p2.conectados ,
p2.lives : boolean,
p2.oks : boolean,
}

```

DD = 0 ∪ DA

### Reglas generales:

Saber si luce una bombilla.

```

if    iguales (?objeto, light, t) and
      iguales (?objeto, live, t) and
      iguales(?objeto, ok, t)
then
      añadir(?objeto, lit, t)
fi

```

Saber si un componente tiene corriente.

```

if    iguales(?obj1, conectado, ?obj2) and
      iguales(?obj2, live, t)
then
      añadir(?obj1, live, t)
fi

```

### Instancia específica:

```
H = { outside.live : t,
      w5.conectado = outside,
      cb1.ok : t,
      s3.estado : {up} ,
      s3.ok : t,
      l2.light : t,
      l2.ok : t,
      l2.conectado = w4,
      cb2.ok : t,
      p2.conectado : w6,
      p1.conectado : w3,
      p1.ok : t,
      p2.ok : t,
      s1.conectado : w3,
      s1.ok : t,
      s1.estado : {down},
      s2.ok : t,
      s2.estado : {up},
      w0.conectado : s2,
      l1.conectado : w0,
      l1.light : t,
      l1.ok : t,
    }

    if iguales(cb1, ok, t)
    then añadir(w3, conectado, w5)
    fi

    if iguales(cb2, ok, t)
    then añadir(w6, conectado, w5)
    fi

    if iguales(s3, ok, t) and
      iguales(s3, estado, up)
    then
      añadir(w4, conectado, w3)
    fi

    if iguales(s1, ok, t) and
      iguales(s1, estado, down)
    then
      añadir(w2, conectado, w3)
    fi
```

```
if    iguales(s1, ok, t) and
      iguales(s1, estado, up)
then
      añadir(w1, conectado, w3)
fi

if    iguales(s2, ok, t) and
      iguales(s2, estado, down)
then
      añadir(w0, conectado, w2)
fi

if    iguales(s2, ok, t) and
      iguales(s2, estado, up)
then
      añadir(w0, conectado, w1)
fi
```