

Félix y el TX-150

Fuente: Ejercicios de Lógica Computacional

(<https://www.cs.us.es/~fsancho/?p=logica-informatica-2020-21>)

Félix tiene en su garaje un taller que utiliza para la reparación de máquinas. Un buen amigo suyo va a hacerle una visita y obtiene la siguiente información:

- Todos los ordenadores son máquinas.
- El TX-150 es un ordenador
- Para cualquier máquina, Félix sabe sólo cómo estropearla o sólo cómo arreglarla.
- Toda máquina puede ser arreglada por alguien (incluso por otras máquinas), pero una máquina no puede arreglarse por sí misma.
- Las cosas sólo desesperan a quienes no son capaces de arreglarlas.
- Félix está desesperado con el TX-150.

Entonces, ¿puede estropear Félix el TX-150?

Solución

Formalicemos el problema en el ámbito de la Lógica de Primer Orden. Para ello consideramos:

- Los símbolos de predicado: $O(x)$: " x es un ordenador", $M(x)$: " x es una máquina", $A(x, y)$: " x puede arreglar y ", $E(x, y)$: " x puede estropear y " y $D(x, y)$: " x desespera a y "
- Y los símbolos de constante a y b para representar al TX-150 y a Félix, respectivamente.

Antes de nada, vamos a importar el módulo de Sintaxis y Semántica en Primer Orden y algunos otros paquetes que nos harán falta para poder trabajar con las fórmulas.

Entonces las fórmulas asociadas a cada uno de los asertos corresponden a:

```
import LogicUS.FOL.SyntaxSemantics exposing (..)
import LogicUS.FOL.Clauses exposing (..)
import LogicUS.FOL.Resolution exposing (..)
import LogicUS.FOL.NormalForms exposing (..)
import Set exposing (Set)
import Dict exposing (Dict)

ffolRead : String -> FormulaFOL
ffolRead s = ffolReadExtraction <| ffolReadFromString s

psi1 : FormulaFOL
psi1 = ffolRead "!A[x](O(x) -> M(x))"

psi2 : FormulaFOL
psi2 = ffolRead "O(*a)"

psi3 : FormulaFOL
psi3 = ffolRead "!A[x](M(x) -> (A(*b, x) <-> ¬E(*b, x)))"

psi4 : FormulaFOL
psi4 = ffolRead "!A[x](M(x) -> (!E[y] A(y, x) & ¬A(x, x)))"

psi5 : FormulaFOL
psi5 = ffolRead "!A[x]!A[y](D(x, y) <-> ¬A(y, x))"

psi6 : FormulaFOL
psi6 = ffolRead "D(*a, *b)"
```

$$\begin{aligned} &\forall x (O(x) \rightarrow M(x)) \\ &O(a) \\ &\forall x (M(x) \rightarrow (A(b, x) \leftrightarrow \neg E(b, x))) \\ &\forall x (M(x) \rightarrow (\exists y A(y, x) \wedge \neg A(x, x))) \\ &\forall x \forall y (D(x, y) \leftrightarrow \neg A(y, x)) \\ &D(a, b) \end{aligned}$$

Ahora veámos si de la información anterior podemos deducir que Félix ha de poder estropear el TX-150. Esto es si $\Psi \models E(b, a)$.

```
psi7 : FormulaFOL
psi7 = ffolRead "E(*b, *a)"
```

$$E(b, a)$$

Para poder utilizar el método de resolución no restringida, primero hemos de pasar las fórmulas a forma Clausal. De forma que el conjunto de cláusulas asociadas a $\Psi \cup \{egE(*b, *a)\}$ corresponde a

```
cs : ClauseFOLSet
cs = sfolToClauses [psi1, psi2, psi3, psi4, psi5, psi6, Neg psi7]
```

$$M(x), \neg O(x), O(a), \neg A(b, x), \neg E(b, x), \neg M(x), A(b, x), E(b, x), \neg M(x), A(\xi_1(x), x), \neg M(x), \neg A(x, x), \neg M(x), \neg A(y, x), \neg D(x, y), A(y, x),$$

Una vez obtenida el conjunto de cláusulas podemos aplicar la Resolución no restringida en Primer Orden como sigue. Comprobamos que es insatisfactible:

```
res : ( Bool, ResolutionTableau )
res = csfolSCFResolution cs
```

True

Mostrando el proceso haciendo uso de la orden `(resolutionTableauToDOT << Tuple.second) res` y eliminando las cláusulas irrelevantes:

```
"digraph G {\n rankdir=TB\n graph TD\n node [shape=box, color=white, fontcolor=black]\n edge [dir=none, color=blue, fontcolor=blue]\n\n 1 -> 5\n 2 -> 12\n 3 -> 27\n 4 -> 5\n [label=<{x/a}>] 5 -> 26\n 10 -> 12\n [label=<{x/a, y/b}>] 12 -> 25\n 25\n 24 -> 25\n [label=<{x/a}>] 25 -> 26\n 26 -> 27\n\n 1 [label=<{O(a)}>] 2 [label=<{D(a,b)}>] 3 [label=<{-E(b,a)}>] 4 [label=<{M(x), -O(x)}>] 5 [label=<{M(a)}>] 10 [label=<{-A(y,x), -D(x,y)}>] 12 [label=<{-A(b,a)}>] 24 [label=<{-A(b,x), E(b,x), -M(x)}>] 25 [label=<{E(b,a), -M(a)}>] 26 [label=<{E(b,a)}>] 27 [label=<□>] 28 [label=<{-A(b,x), -E(b,x), -M(x)}>] 29 [label=<{A(s1(x), x), -M(x)}>] 30 [label=<{-A(x,x), -M(x)}>] 31 [label=<{A(y,x), D(x,y)}>]\n\n {rank=same; 1,2;3,4;10,24;28,29;30,31;}\n}"
```

