Problema del robo

> Fuente: Ejercios de Lógica Computacional
> (https://www.cs.us.es/~fsancho/?p=logica-informatica-2020-21)

Alberto, Berta y Carlos son los tres sospechosos de un robo. Se les interroga por separado y éstas son sus declaraciones:

- *Alberto*: Berta es culpable y Carlos es inocente
- *Berta*: Si Alberto es culpable, Carlos también.
- *Carlos*: Yo soy inocente, pero al menos uno de los otros dos es culpable

¿Son consistentes las declaraciones?¿quién es inocente y quién culpable si todos dijeron la verdad?¿y si únicamente lo hicieron las personas inocentes?

Solución

Parte I

Sean a, b, c variables proposicionales que indican si Alberto, Berta y Carlos son culpables (V) o si son inocentes (F). De forma que los testimonios podrían expresarse como:

$$\{b \wedge \neg c,\ a \rightarrow c,\ \neg c \wedge (a \vee b)\}$$

Ahora, determinar la consistencia del conjunto es equivalente a determinar la satisfactibilidad de la conjunción de las fórmulas. Para lo que utilizaremos la FND. Haciendo uso de LogicUS:

```
import LogicUS.PL.SintaxSemantics exposing (..)
import LogicUS.PL.NormalForms exposing (..)

fplRead : String -> FormulaPL
fplRead = fplReadExtraction << fplReadFromString
```

Definimos las fórmulas:

```
f1 : FormulaPL
f1 = fplRead "b & ¬ c"

f2 : FormulaPL
f2 = fplRead "a -> c"

f3 : FormulaPL
f3 = fplRead "¬c & (a | b)"
```

$$(b \wedge \neg c), (a \rightarrow c), (\neg c \wedge (a \vee b))$$

Entonces ahora para determinar la satisfactibilidad podemos hacerlo a través de la función `fplSatisfiabilityDNF` y la conjunción de las fórmulas del conjunto:

```
g : FormulaPL
g = splConjunction [f1, f2, f3]
```

que corresponde a:

$$(((b \wedge \neg c) \wedge (a \rightarrow c)) \wedge (\neg c \wedge (a \vee b)))$$

De forma que veamos si g es satisfactible. Vamos a hacerlo pasos a paso, en vez de utilizar directamente `fplSatisfiabilityDNF` :

1. Hallamos una FND de g:

```
g1 : FormulaPL
g1 = fplToDNF g
```

$$(((((b \wedge \neg c) \wedge \neg a) \wedge (\neg c \wedge a)) \vee (((b \wedge \neg c) \wedge \neg a) \wedge (\neg c \wedge b))) \vee ((((b \wedge \neg c) \wedge c) \wedge (\neg c \wedge a)) \vee (((b \wedge \neg c) \wedge c) \wedge (\neg c \wedge b))))$$

O como conjunto de conjuntos de literales:

```
g1_ls : Maybe (List SetPL)
g1_ls = dnfAsLiteralSets g1
```

$$\{\neg a,\, b,\, \neg c,\, a\}, \{\neg a,\, b,\, \neg c\}, \{b,\, c,\, \neg c,\, a\}, \{b,\, c,\, \neg c\}$$

De forma que la fórmula, y por ende, el conjunto es consistente ya que existe al menos un conjunto (de hecho sólo existe 1) que no posee literales complementarios. Obsérvese también que dado que se ha obtenido un único conjunto sin literales complementarios en el que además aparecen todos los símbolos proposicionales entonces éste proporciona el único modelo para la fórmula. De forma que es fácil deducir que, si todos dijeron la verdad, la culpable fue Berta.

Parte II

Ahora, si solo dijeron la verdad los inocentes entonces el problema podríamos plantearlo como:

$$\{(b \wedge \neg c) \leftrightarrow \neg a,\, (a \rightarrow c) \leftrightarrow \neg b,\, (\neg c \wedge (a \vee b)) \leftrightarrow \neg c\}$$

Ahora, determinar la consistencia del conjunto es equivalente a determinar la satisfactibilidad de la conjunción de las fórmulas. Para lo que utilizaremos la FND. Haciendo uso de LogicUS:

```
f1 : FormulaPL
f1 = fplRead "(b & ¬c) <-> ¬a"

f2 : FormulaPL
f2 = fplRead "(a -> c) <-> ¬b"

f3 : FormulaPL
f3 = fplRead "(¬c & (a | b)) <-> ¬c"
```

$$((b \wedge \neg c) \leftrightarrow \neg a), ((a \rightarrow c) \leftrightarrow \neg b), ((\neg c \wedge (a \vee b)) \leftrightarrow \neg c)$$

De forma que comprobamos que el conjunto es consistente:

```
g : FormulaPL
g = splConjunction [f1,f2,f3]

gIsSAT : Bool
gIsSAT = fplSatisfiabilityDNF g
```

True

Y sus modelos corresponden a

```
gModels : List Interpretation
gModels =  fplModelsDNF g
```

$$\{a : T, b : F, c : T\}$$

De manera que bajo el segundo supuesto, los culpables se habrían aliado para delatar a Berta.