

Oferta especial en Sortilegios Weasley

Fuente: Ejercicios de Lógica Computacional

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

Sortilegios Weasley, también conocida como Weasley & Weasley, es una tienda de artículos de bromas ubicada en el Callejón Diagon propiedad de los gemelos Fred y George Weasley. Entre otros productos en la tienda disponen de *Sombreros acéfalos* (A), *Bombones desmayo* (B), *Caramelos longuilinguos* (C), *Fantasías patentadas* (F), *Libro mordedor* (L) y *Orejas extensibles* (O).

Los hermanos Weasley han sacado una caja sorpresa a la venta de la que han hecho varios carteles en los que se puede leer:

- Si la caja contiene sombreros acéfalos y bombones desmayo entonces también tiene caramelos longuilinguos.
- ¡Que no te engañen! En la caja nada de fantasías patentadas, lo que sí tiene es un libro mordedor y sombreros acéfalos.
- ¿Y si tuviera sombreros acéfalos y orejas extensibles? Entonces, tendría también fantasías patentadas.
- ¿Qué tal un libro mordedor? En tal caso deberían incluirse unos bombones desmayo como compensación.
- No somos partidarios de incluir sombreros acéfalos y libro mordedor, pero en tal caso también incluiríamos unas orejas extensibles.

Hermione, que es muy avispada, concluyó que los hermanos no habían dicho la verdad en todos los carteles, sin necesidad de recurrir a la maldición *Cruciatus*. ¿Estaba en lo cierto?

Solución

Los enunciados de los carteles pueden formalizarse en el conjunto

$$\{A \wedge B \rightarrow C, \neg F \wedge L \wedge A, A \wedge O \rightarrow F, L \rightarrow B, A \wedge L \rightarrow O\}$$

Demostremos que el conjunto es inconsistente utilizando distintas estrategias de resolución. Nótese que corresponde a un conjunto de cláusulas de Horn, por lo que todas las estrategias expuestas son refutacionalmente completas.

Expresemos el conjunto como un conjunto de cláusulas:

```
import LogicUS.PL.SyntaxSemantics exposing (..)
import LogicUS.PL.Clauses exposing (..)
import LogicUS.PL.Resolution exposing (..)
```

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

Definimos las fórmulas:

```
f1 : FormulaPL
f1 = fplRead "a & b -> c"

f2 : FormulaPL
f2 = fplRead "¬f & l & a"

f3 : FormulaPL
f3 = fplRead "a & o -> f"

f4 : FormulaPL
f4 = fplRead "l -> b"

f5 : FormulaPL
f5 = fplRead "a & l -> o"
```

Entonces, el conjunto de cláusulas correspondería a:

```
cs : ClausePLSet
cs = splToClauses [f1,f2,f3, f4, f5]
```

$$\{\{\neg a, \neg b, c\}, \{\neg f\}, \{l\}, \{a\}, \{\neg a, f, \neg o\}, \{b, \neg l\}, \{\neg a, \neg l, o\}\}$$

Resolución regular

```
cs_reg : (Bool, List (List ClausePL))
cs_reg = csplRegularResolution (csplSymbols cs) cs
```

True

De forma que el proceso correspondería a :

$$\begin{aligned} &\{\{\neg a, \neg b, c\}, \{\neg f\}, \{l\}, \{a\}, \{\neg a, f, \neg o\}, \{b, \neg l\}, \{\neg a, \neg l, o\}\} \\ &\quad \{\{\neg f\}, \{l\}, \{b, \neg l\}, \{\neg b, c\}, \{f, \neg o\}, \{\neg l, o\}\} \\ &\quad \quad \{\{\neg f\}, \{l\}, \{f, \neg o\}, \{\neg l, o\}, \{c, \neg l\}\} \\ &\quad \quad \quad \{\{\neg f\}, \{l\}, \{f, \neg o\}, \{\neg l, o\}\} \\ &\quad \quad \quad \quad \{\{\neg l, o\}, \{\neg o\}\} \\ &\quad \quad \quad \quad \quad \{\{\neg o\}, \{o\}\} \\ &\quad \quad \quad \quad \quad \quad \{\square\} \end{aligned}$$

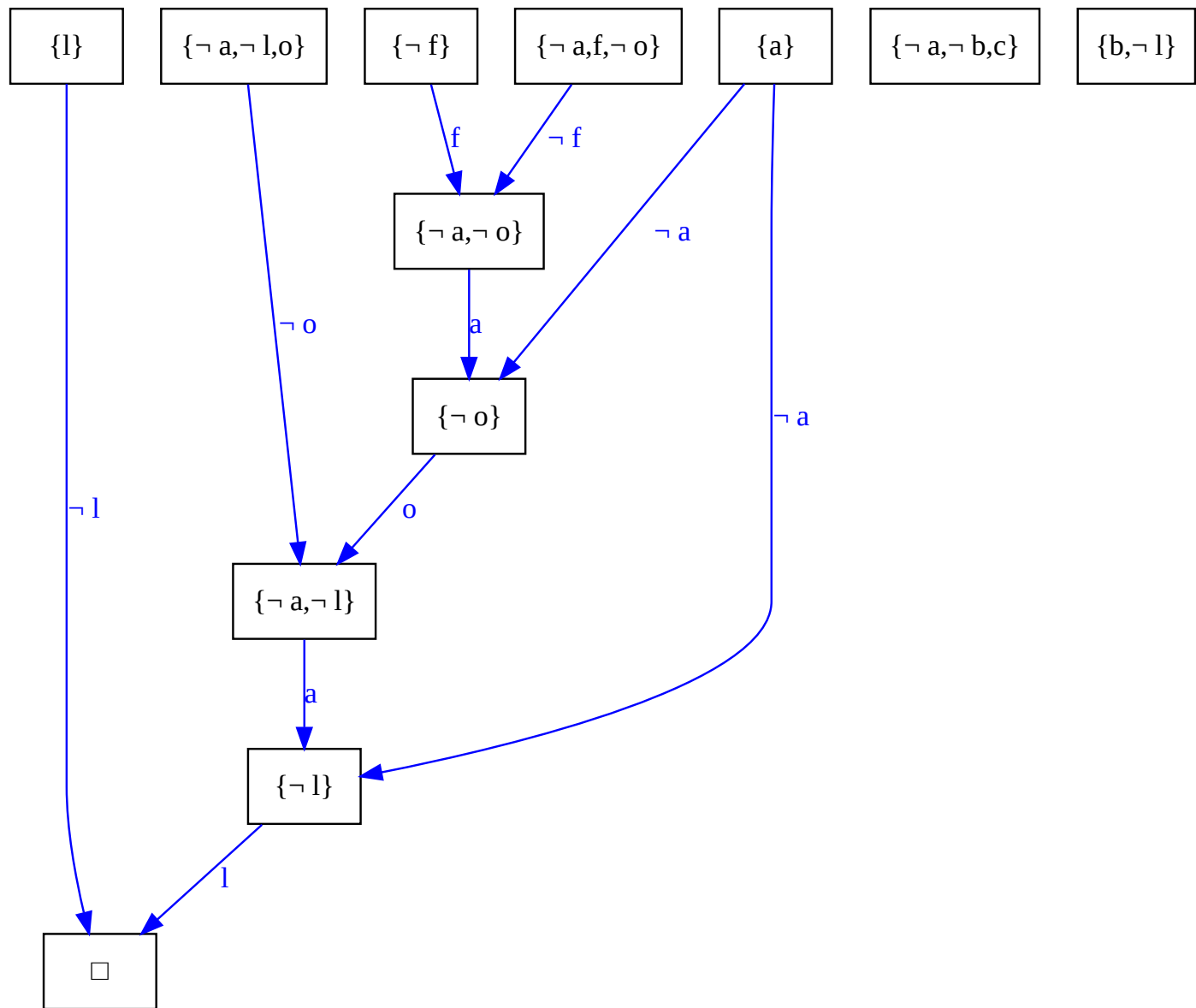
En efecto, se llega a la cláusula vacía luego el conjunto es insatisfactible.

Resolución lineal SCF

```
cs_lin : (Bool, ResolutionTableau)
cs_lin = csplSCFLinearResolution cs
```

True

De forma que el proceso correspondería a :



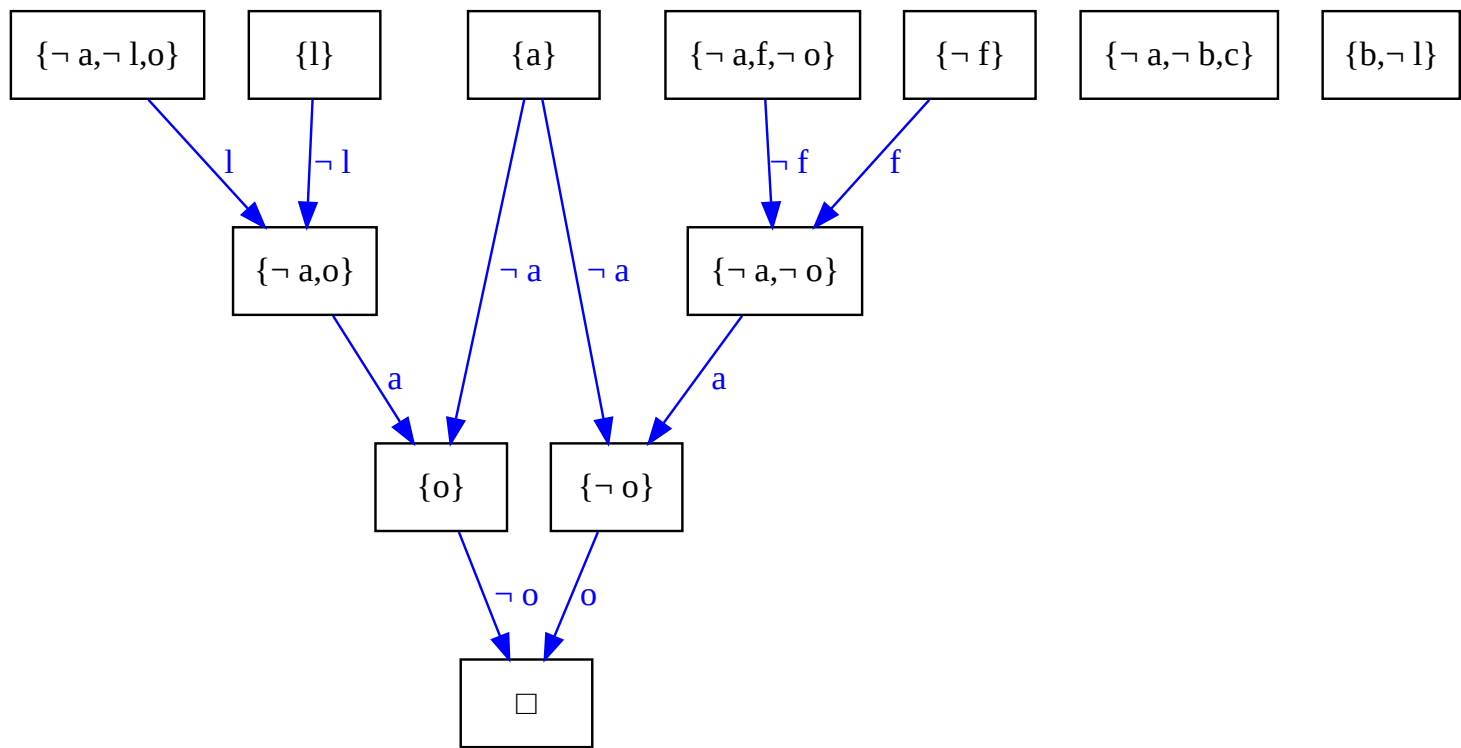
En efecto, se llega a la cláusula vacía luego el conjunto es insatisfactible. Nótese que también es por entradas, unitaria y negativa. Aunque existan otras, ejcutemos la unitaria por ejemplo:

Resolución unitaria SCF

```
cs_unit : (Bool, ResolutionTableau)
cs_unit = csplSCFUnitaryResolution cs
```

True

De forma que el proceso correspondería a :



Salen otros procesos de resolución, pero en este caso no es ni lineal ni por entradas.

Veamos finalmente una resolución positiva:

Resolución positiva SCF

```
cs_pos : (Bool, ResolutionTableau)
cs_pos = csplSCFPositiveResolution cs
```

True

De forma que el proceso correspondería a :

