# Algoritmia y Complejidad

Salvador Carrillo Fuentes

Grado Ingeniería Informática
Universidad de Málaga

*MineConsistency* $\in$ NPC

# Tabla de contenidos

- Tablero en el que cada casilla contiene una mina o está vacía.
- Si se elige una casilla que contiene una mina, se pierde.
- Si se elige una casilla que está vacía, se proporciona información acerca del número de minas adyacentes.
- Por ejemplo, si una casilla tiene el número 3, significa que de las 8 casillas adyacentes (si no es una esquina o borde) hay 3 con minas.
- Objetivo: marcar todos las casillas vacías.

- El tablero y las casillas se representan mediante un grafo no dirigido $G$.
- Algunos nodos de $G$ están etiquetados con números, $G$ es un grafo parcialmente etiquetado.
- Objetivo: determinar si es posible una asignación de minas en los nodos sin etiquetar que sea **consistente**.
- Una asignación es consistente si cualquier nodo $v$ etiquetado con el número $m$ tiene exactamente $m$ nodos adyacentes que contienen minas.

## Definición

Un **grafo parcialmente etiquetado** es una tupla $(V, E, L, m)$ donde $(V, E)$ es un grafo, $L \subseteq V$ y $m : L \to \mathbb{N}$ es una función de etiquetado.

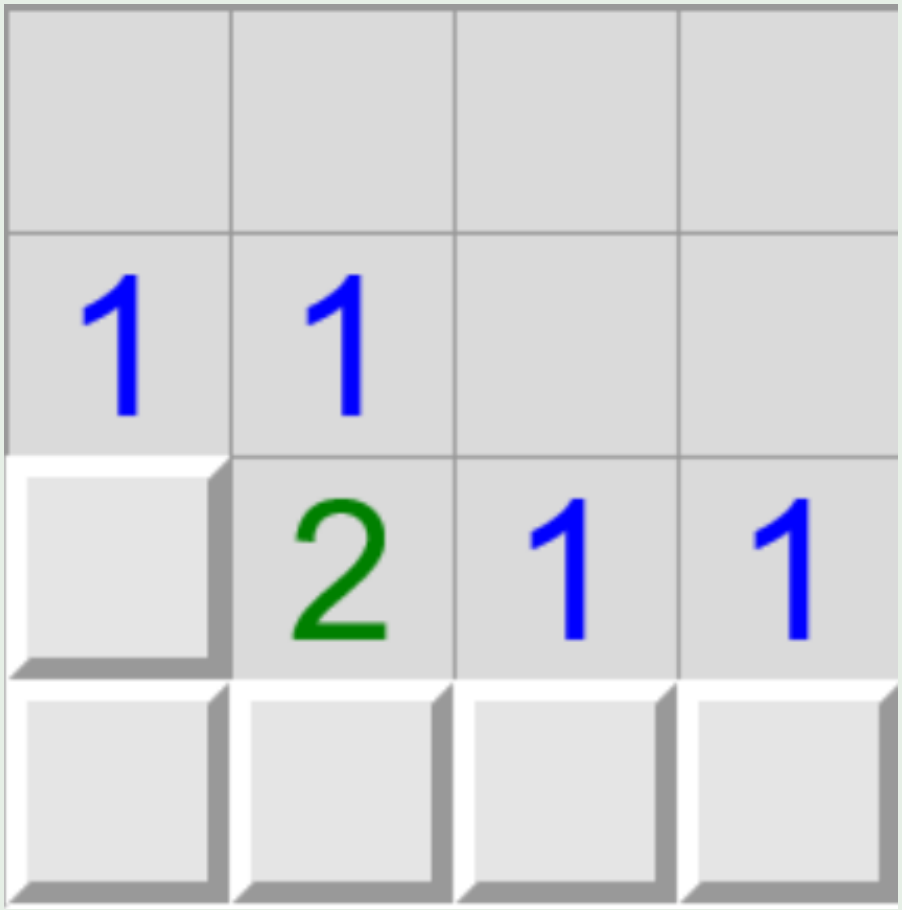
## Definición

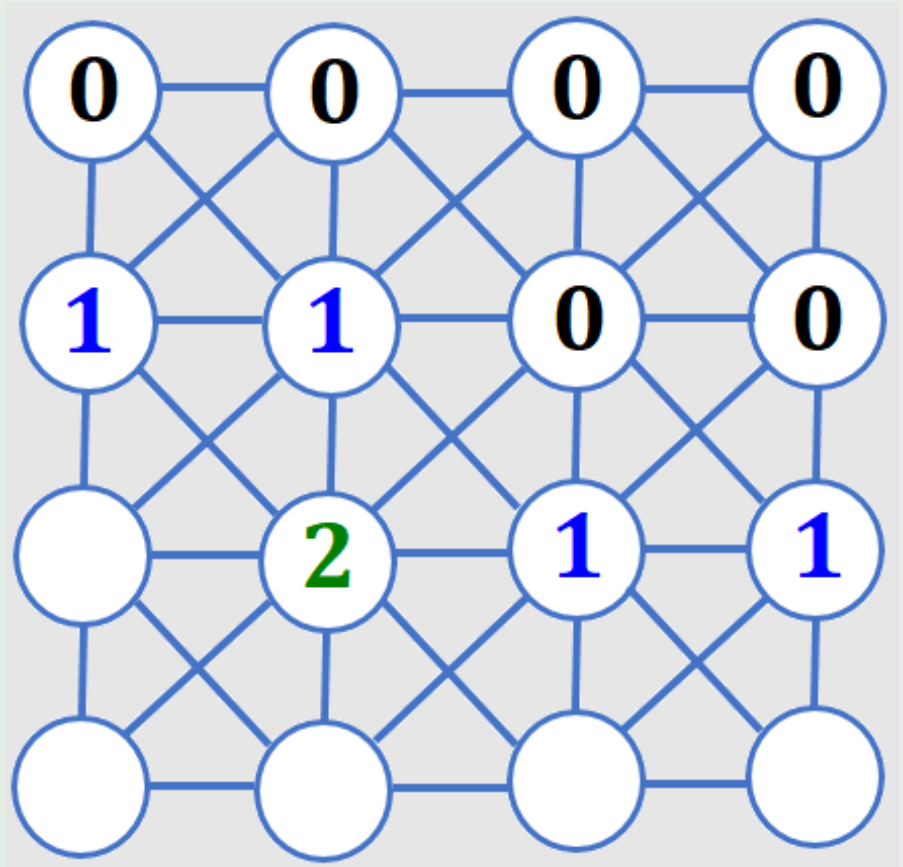
Una **asignación consistente de minas** en un grafo parcialmente etiquetado es un subconjunto de $V \setminus L$ tal que para cada $v \in L$, $m(v) = |N(v) \cap M|$, donde $N(v)$ son los nodos vecinos de $v$ y $M$ los nodos que contienen minas.

## Definición

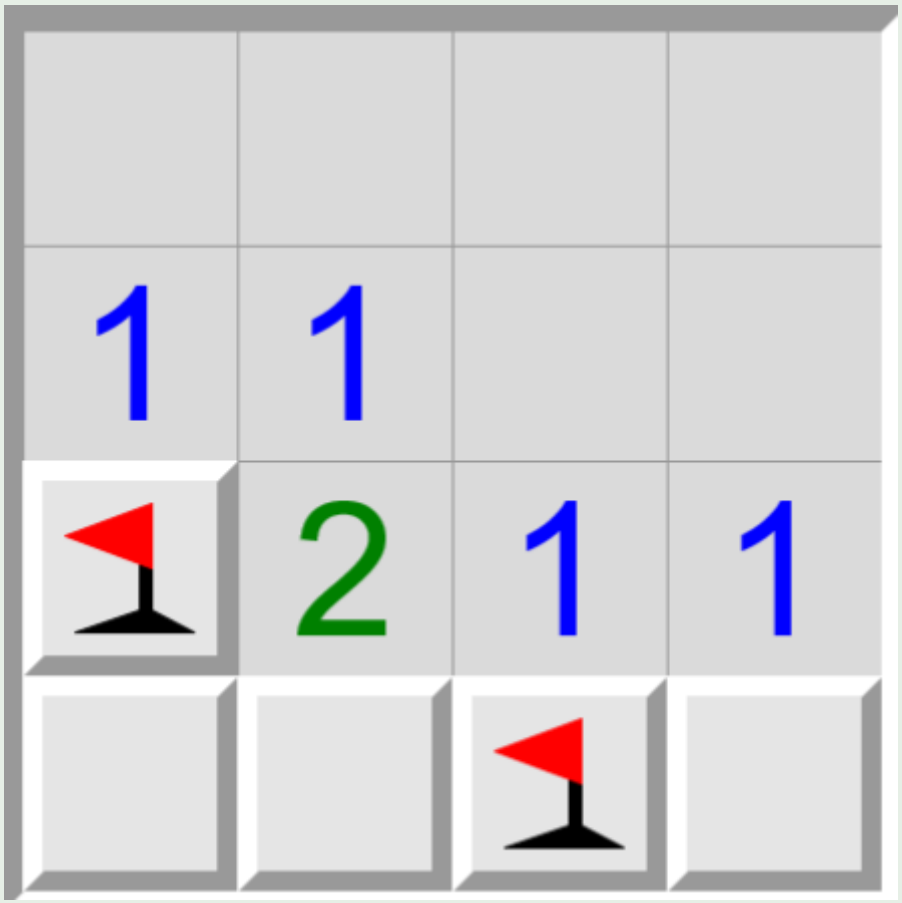
Sea $MineConsistency = \{\langle G \rangle \mid G$ es un grafo parcialmente etiquetado que admite una asignación consistente de minas$\}$

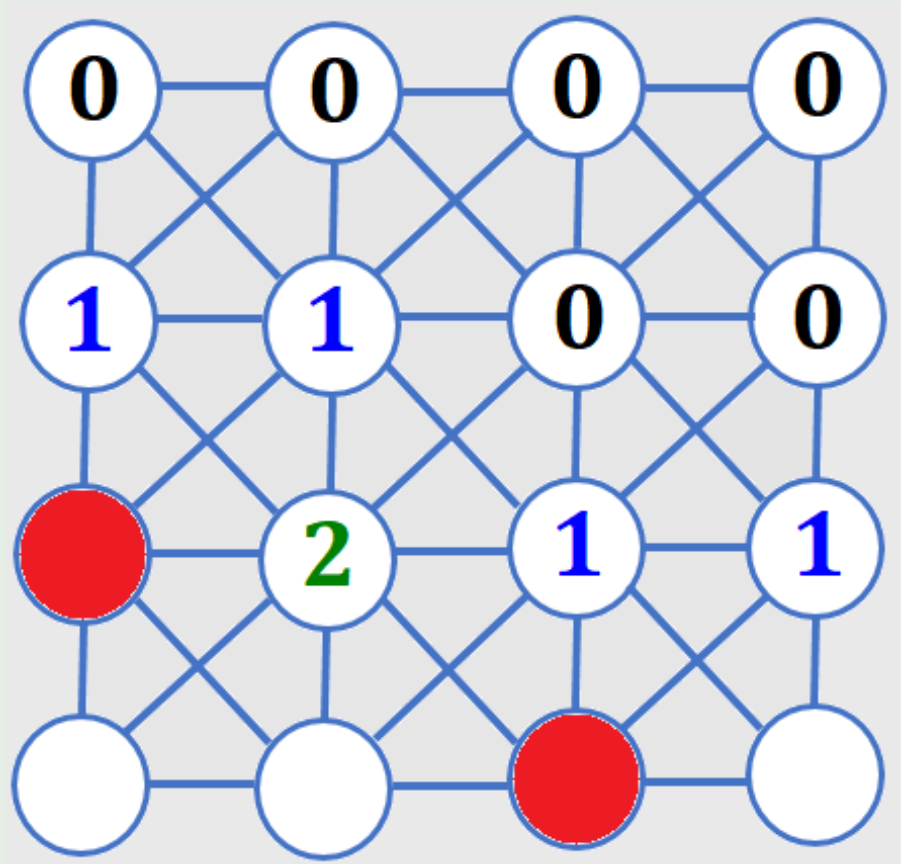
## Ejemplo de instancias consistentes



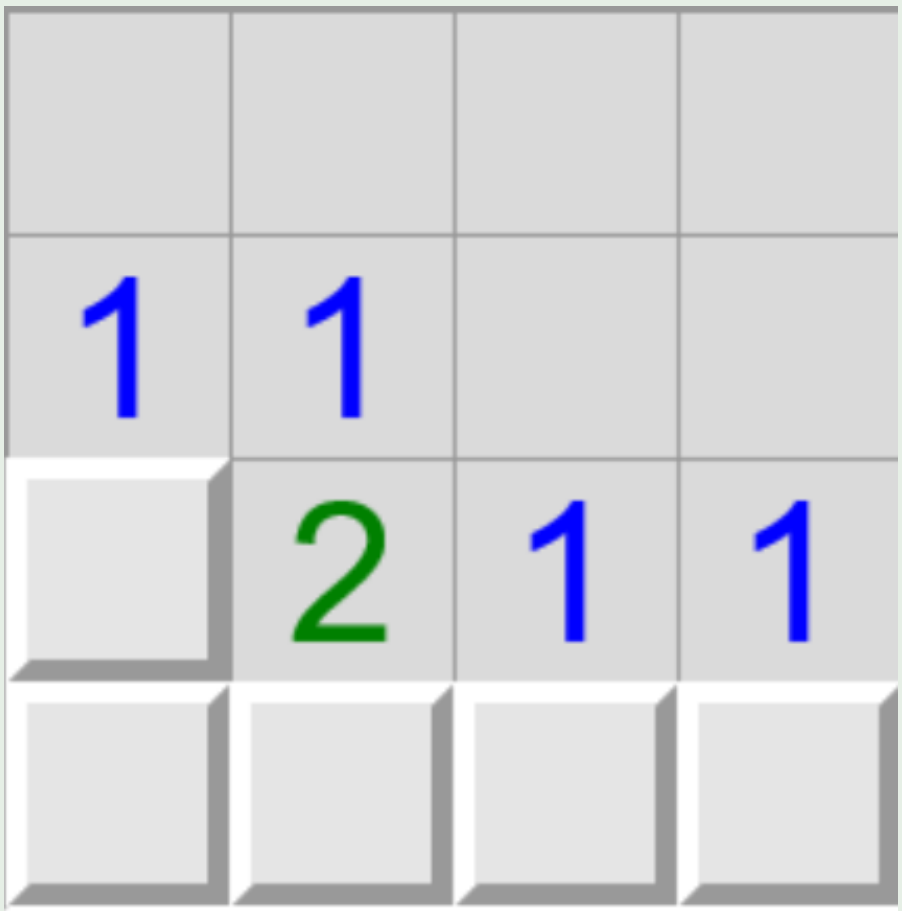



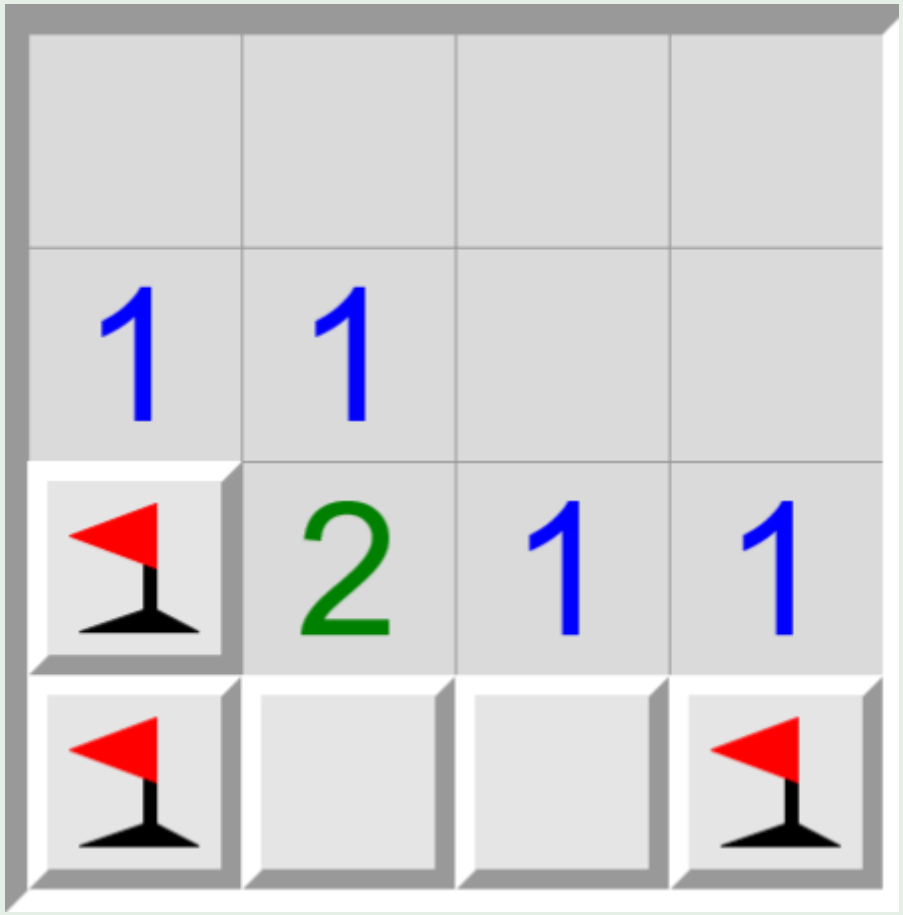
## Ejemplo de instancias consistentes (solución)



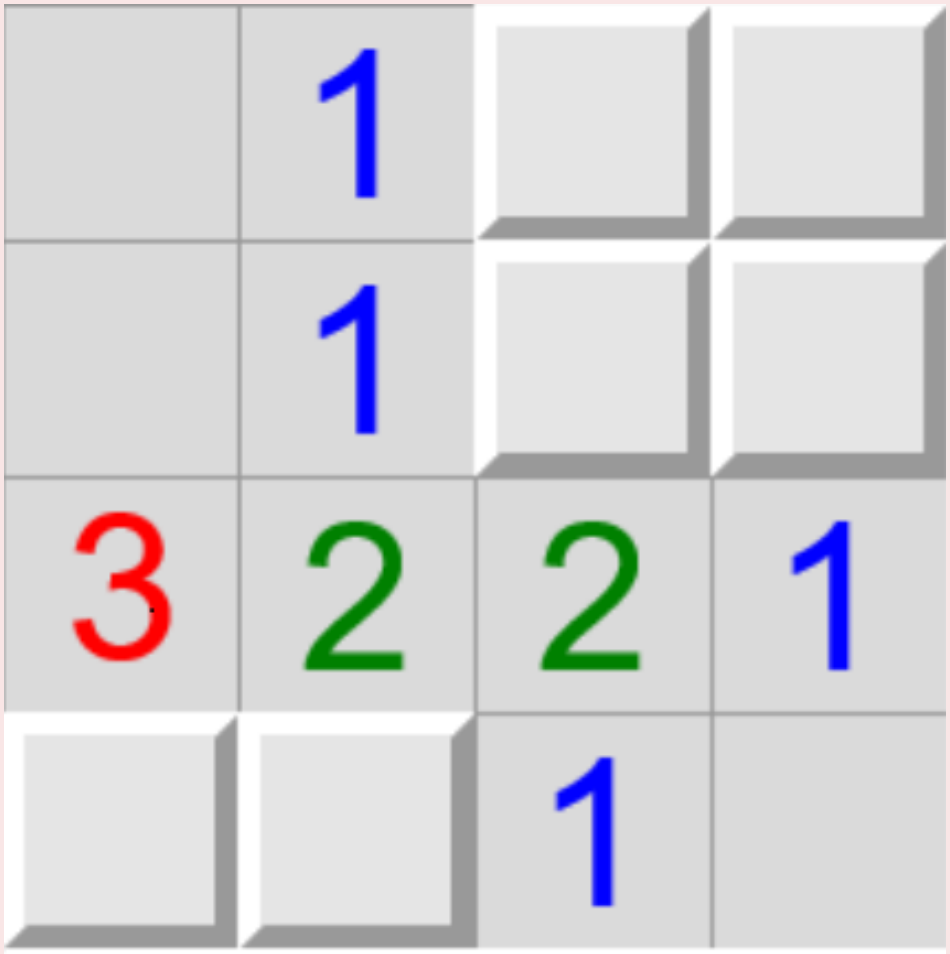



## Ejemplo de instancia consistente (solución alternativa)



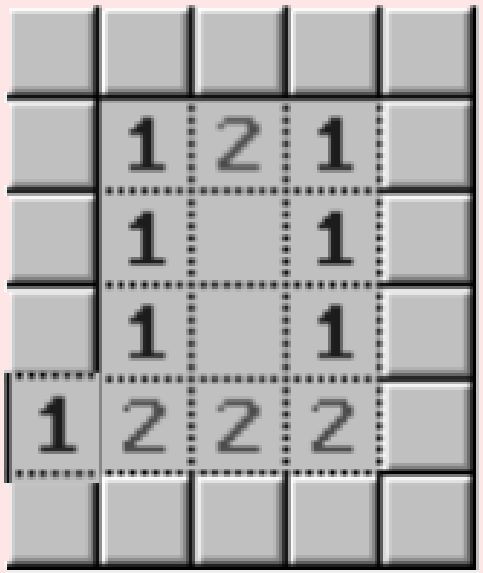



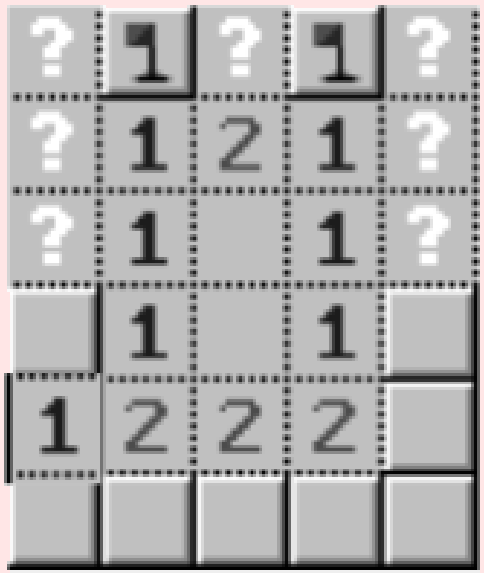
## Tablero inconsistente

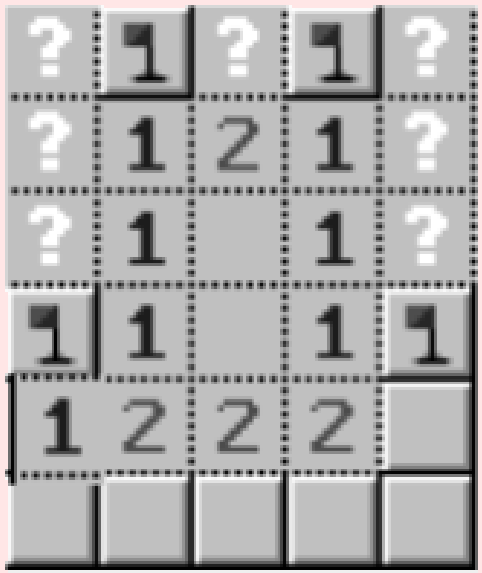



La casilla con el **3** no puede tener **3** minas adyacentes porque solo tiene dos casillas adyacentes sin descubrir.

## Tablero inconsistente







**Teorema**

$\boldsymbol{Mineconsistency}$ está en NP.

Demostración: Sea $\boldsymbol{M}$ una máquina de Turing no determinista que se ejecuta en tiempo polinomial para $\boldsymbol{MineConsistency}$.

$\boldsymbol{M}$ = "Para la entrada $\langle \boldsymbol{G} \rangle$, donde $\boldsymbol{G}$ es un grafo parcialmente etiquetado:

1. De forma no determinista, realiza una asignación de minas en los nodos sin etiquetar de $\boldsymbol{G}$.
2. Comprueba si la asignación es consistente.
3. Si lo es, *aceptar*; en otro caso, *rechazar*."

Idea de la demostración alternativa: El certificado es la asignación de minas consistente.

Demostración alterntiva:
Sea $V$ una verificador en tiempo polinomial para $MineConsistency$.

$V =$ "Para la entrada $\langle G, c \rangle$, donde $G$ es un grafo parcialmente etiquetado:

1. Comprueba que $c$ es una asignación consistente de minas para $G$.
2. Comprueba que los nodos de $c$ están en $G$ sin etiquetar.
3. Si se dan ambas comprobaciones, *aceptar*; en otro caso, *rechazar*."

- Caso particular de $\boldsymbol{SAT}$ en el que la fórmula está en una forma especial.
- Un literal es una variable Booleana o su negación: $\boldsymbol{x}, \boldsymbol{\bar{x}}$.
- Una cláusula es la conexión de varios literales mediante $\vee$'s:

$$(\boldsymbol{x_1} \vee \boldsymbol{\bar{x}_2} \vee \boldsymbol{\bar{x}_3} \vee \boldsymbol{x_4})$$

- Una fórmula está en forma normal conjuntiva ($\boldsymbol{fnc}$) si está formada por varias cláusulas conectadas mediante $\wedge$'s:

$$(\boldsymbol{x_1} \vee \boldsymbol{\bar{x}_2} \vee \boldsymbol{\bar{x}_3} \vee \boldsymbol{x_4}) \wedge (\boldsymbol{x_3} \vee \boldsymbol{\bar{x}_5} \vee \boldsymbol{x_6}) \wedge (\boldsymbol{x_3} \vee \boldsymbol{\bar{x}_6})$$

- Tenemos una fórmula $\boldsymbol{3fnc}$ si todas las cláusulas tiene tres literales:

$$(\boldsymbol{x_1} \vee \boldsymbol{\bar{x}_2} \vee \boldsymbol{\bar{x}_3}) \wedge (\boldsymbol{x_3} \vee \boldsymbol{\bar{x}_5} \vee \boldsymbol{x_6}) \wedge (\boldsymbol{x_3} \vee \boldsymbol{\bar{x}_6} \vee \boldsymbol{x_4})$$

**Definición**

Sea $\boldsymbol{3SAT} = \{\langle \phi \rangle \mid \phi$ es una fórmula $\boldsymbol{3fnc}$ satisfacible$\}$

- Tomamos como resultado previo que $\boldsymbol{3SAT}$ es NP-completo.

# *MineConsistency* es NP-completo

## Función necesaria

- Reducimos en tiempo polinomial $\boldsymbol{3SAT}$ a $\boldsymbol{MineConsistency}$.

- Buscamos una función computable en tiempo polinómico $\boldsymbol{f : \Sigma^* \to \Sigma^*}$, tal que, para cada $\boldsymbol{w}$,

$$\boldsymbol{w \in 3SAT \Leftrightarrow f(w) \in MineConsistency}$$
$$\boldsymbol{\phi \in 3SAT \Leftrightarrow G \in MineConsistency}$$

- El objetivo es encontrar estructuras en $\boldsymbol{MineConsistency}$ que simulen a las variables y las cláusulas de la fórmula $\boldsymbol{3fnc}$.
- Sean $\boldsymbol{c_1, ..., c_s}$ las cláusulas de $\boldsymbol{\phi}$ y $\boldsymbol{x_1, ..., x_n}$ sus variables.
- Una variable $\boldsymbol{x_i}$ aparece en $\boldsymbol{\phi}$ si $\boldsymbol{x_i}$ o $\boldsymbol{\bar{x}_i}$ aparece en alguna cláusula de $\boldsymbol{\phi}$.

**Teorema**

$3SAT$ es reducible en tiempo polinomial a $MineConsistency$.

Demostración: El algoritmo $R$ computa la reducción.

$R =$ "Para la entrada $\langle \phi \rangle$, donde $\phi$ es una fórmula $3fnc$:

**1.** Para cada variable $x_i$ que aparece en $\phi$:

**1.1** Crear tres nodos $v_i, x_i^t, x_i^f$.

**1.2** Añadir las aristas $(v_i, x_i^t), (v_i, x_i^f)$.

**1.3** Establecer $m(v_i) = 1$ y $m(x_i^t)$, $m(x_i^f)$ sin etiquetas.

**2.** Para cada cláusula $c_j$ de $\phi$:

**2.1** Crear tres nodos $c_j, c_j^1, c_j^2$.

**2.2** Añadir aristas desde $c_j$ hasta los nodos correspondientes a los tres literales de $c_j$.

**2.3** Establecer $m(c_j) = 3$ y $m(c_j^1)$, $m(c_j^2)$ sin etiquetas."

- Tras la ejecución de $\boldsymbol{R}$, obtenemos un grafo parcialemente etiquetado $\boldsymbol{G_\phi = (V, E, L, m)}$ con el etiquetado $\boldsymbol{m_\phi = L \to \mathbb{N}}$ donde:
  - $\boldsymbol{V = \{v_i, x_i^t, x_i^t \mid i \leq n\} \sqcup \{c_j, c_j^1, c_j^2 \mid j \leq s\}}$
  - $\boldsymbol{E = \{\{v_i, x_i^t\}, \{v_i, x_i^f\} \mid i \leq n \cup}$
    $\boldsymbol{\{\{c_j, c_j^1\}, \{c_j, xc_j^2\} \mid j \leq s \cup}$
    $\boldsymbol{\{\{x_i, c_j\}}$ | $\boldsymbol{x_i}$ es un literal en $\boldsymbol{c_j}$ como *true* $\boldsymbol{(x_i^t)}$ o *false* $\boldsymbol{(x_i^f)}$
  - $\boldsymbol{L = \{v_i \mid i \leq n\} \cup \{c_j \mid j \leq s\}}$, donde $\boldsymbol{m_\phi(v_i) = 1}$ y $\boldsymbol{m_\phi(c_j) = 3}$

- La conversión se computa en tiempo polinomial ya que tenemos que:
  - $\boldsymbol{3n + 3s}$ vértices.
  - $\boldsymbol{2n + 5s}$ aristas.
  - $(\boldsymbol{n + s})$ etiquetas.
- Obtenemos $\langle \boldsymbol{G_\phi, m_\phi} \rangle$ en tiempo polinómico para el tamaño de entrada de $\boldsymbol{\phi}$.

## Afirmación ($\rightarrow$)

Si hay una asignación satisfacible para $\phi$, hay una asignación de minas consistente para $G_\phi$.

Demostración: Consideremos a $M$ como el conjunto de los literales que se evalúan a $true$ en $\phi$. Solo se seleccionará uno de entre $\{x_i^t, x_i^f\}$ para cada $i$, así que en los vecinos de $v_i$ habrá exactamente $1 = m(v_i)$ minas.

Además, en los vecinos de cada $c_j$ habrá al menos un literal que se evalúe a $true$ (ya que la asignación es satisfacible) y como máximo tres. Por tanto, podemos añadir $c_j^1$ o $c_j^2$ a $M$ para que $c_j$ sea adyacente a $3 = m(c_j)$ minas exactamente.

## Afirmación ($\leftarrow$)

Si existe una asignación de minas consistente para $G_\phi$, hay una asignación satisfacible para $\phi$.

Demostración: Sea $M$ la asignación de minas y consideremos $M \cap \{x_i^t, x_i^f \mid i \leq n\}$. Como cada $v_i$ es adyacente a una sola mina, contendrá uno de entre $\{x_i^t, x_i^f\}$.

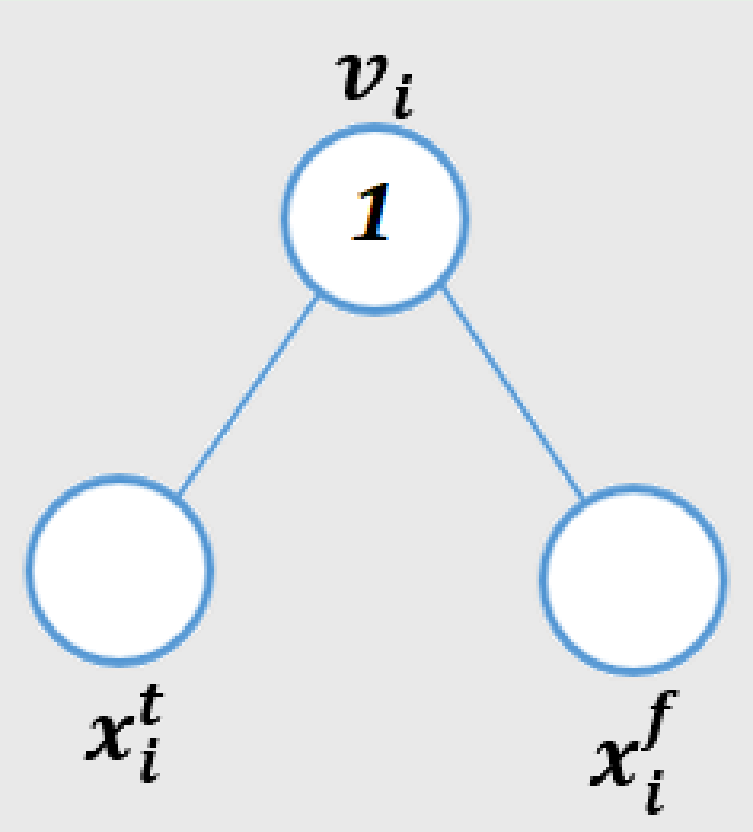
Como cada $c_j$ es adyacente a exactamente tres minas, y solo dos de sus vecinos no son literales, debe ser adyacente a algún literal que es una mina ($true$ en la evaluación). Por tanto, la evaluación es $true$ para toda cláusula, obteniendo una asignación que satisface a $\phi$.

Como $\langle \phi \rangle \in 3SAT$ sii $\langle G_\phi \rangle \in MineConsistency$ y la función $\langle \phi \rangle \mapsto \langle G_\phi \rangle$, concluimos que $MineConsistency \in$ NP-completo ✓.

# Ejemplo

$$\phi = (x_1 \vee \bar{x}_2 \vee x_3)$$

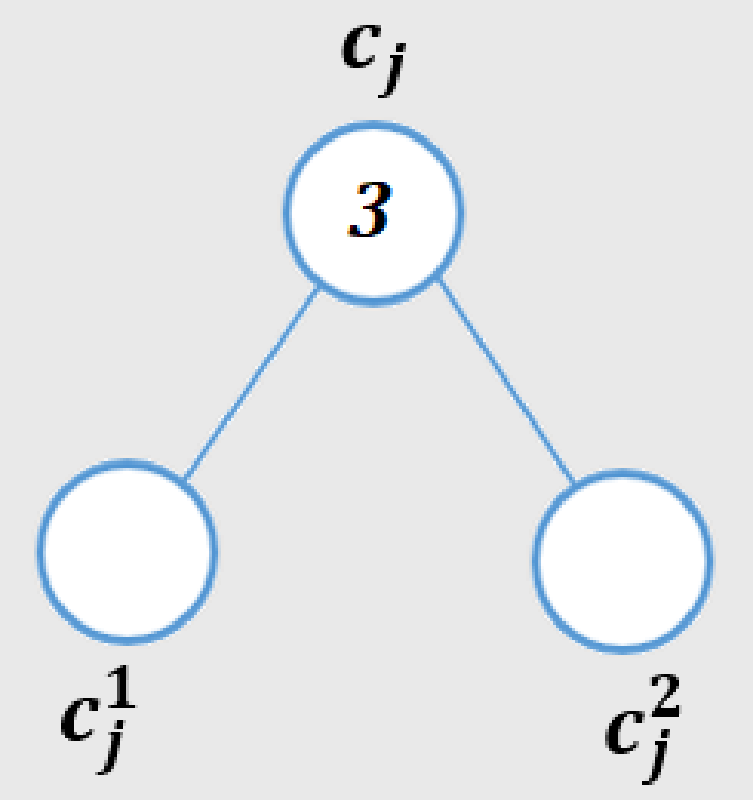
## *Gadget* para cada variable $v_i$



# Ejemplo

$$\phi = (x_1 \vee \bar{x}_2 \vee x_3)$$

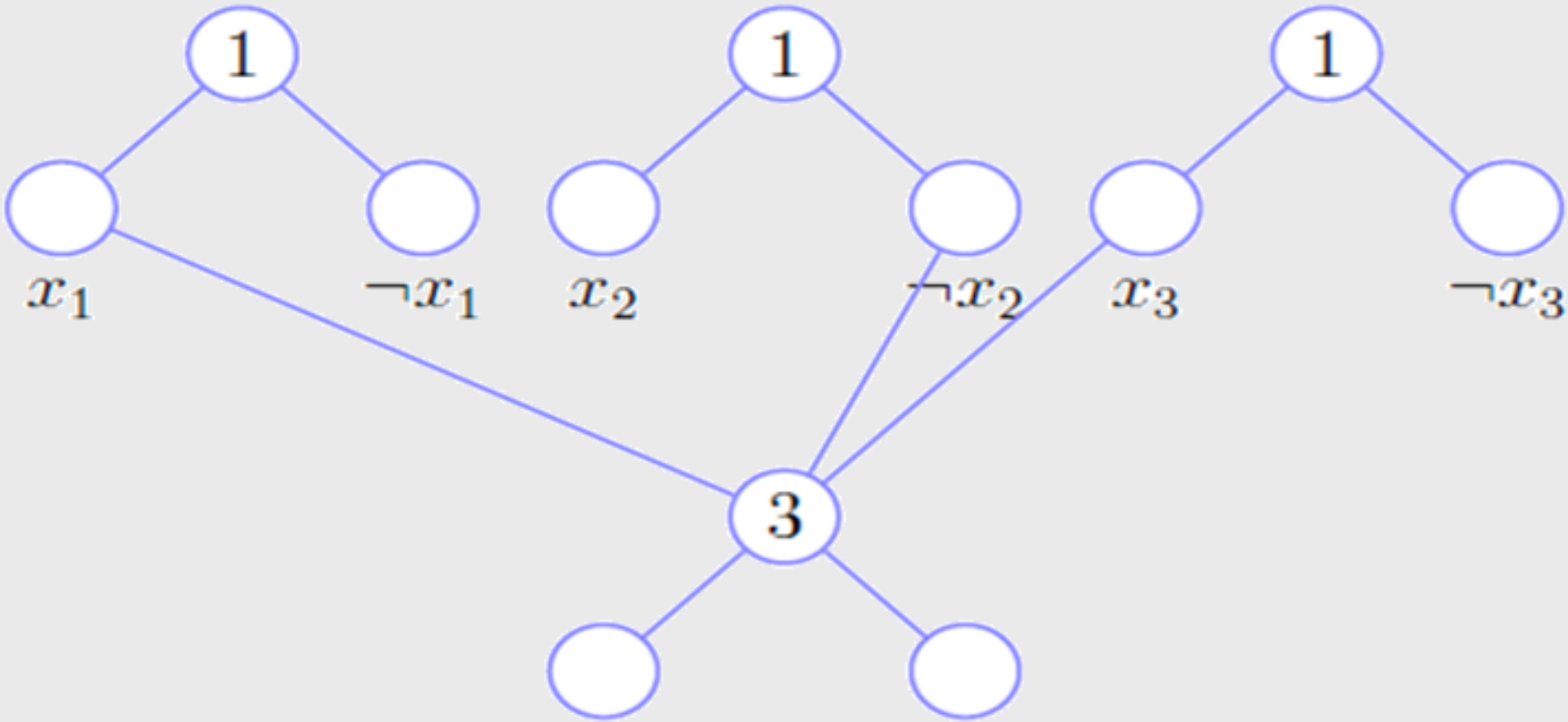
## *Gadget* para cada cláusula $c_j$



# Ejemplo

$$\phi = (x_1 \vee \bar{x}_2 \vee x_3)$$

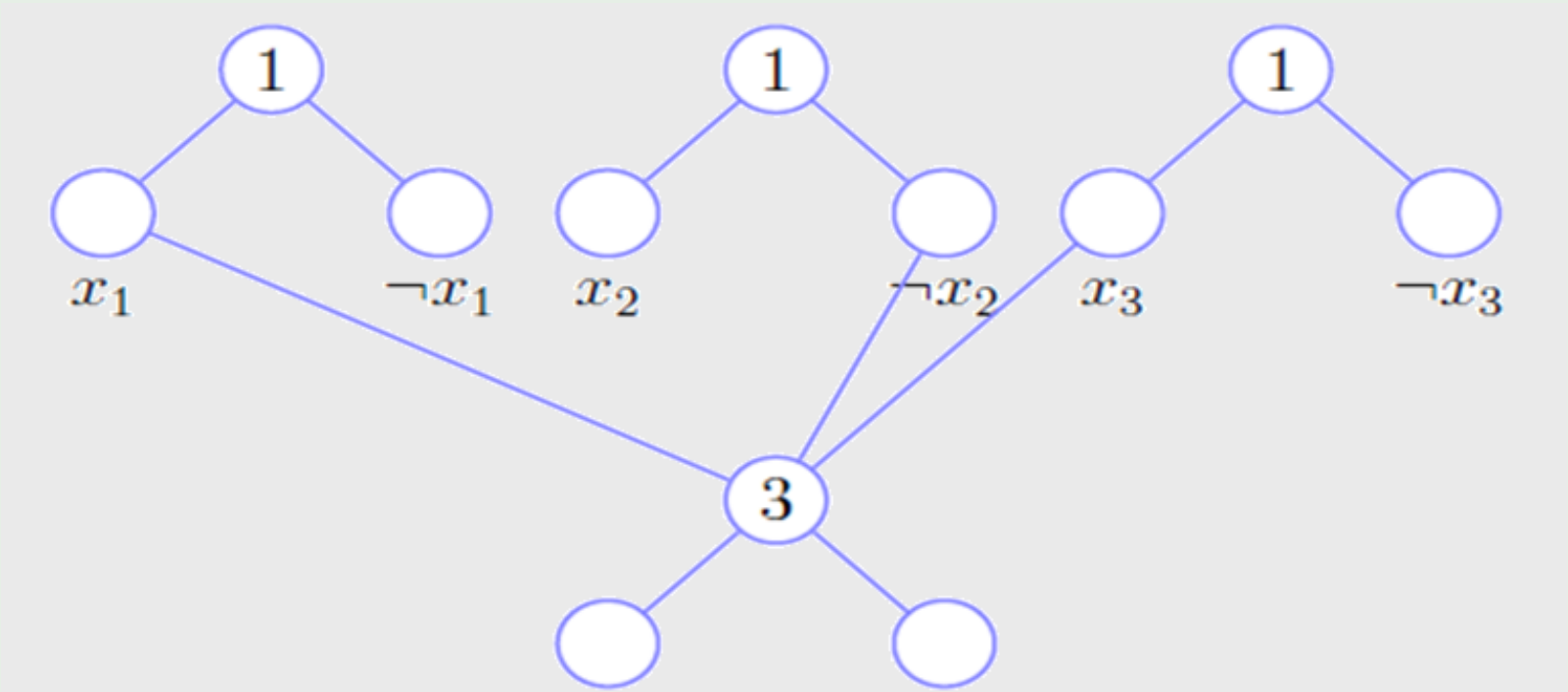
## Aristas desde $c_j$ hasta los nodos de los tres literales de $c_j$



# Ejemplo

- Dada una asignación que satisface a $\phi$,
  - Si $x_i = true$, ponemos una mina en $x_i$
  - Si $x_i = false$, ponemos una mina en $\bar{x}_i$
- Los nodos etiquetados con el número $'1'$ fuerzan a que solo uno de entre $\{x_i, \bar{x}_i\}$ tengan una mina.

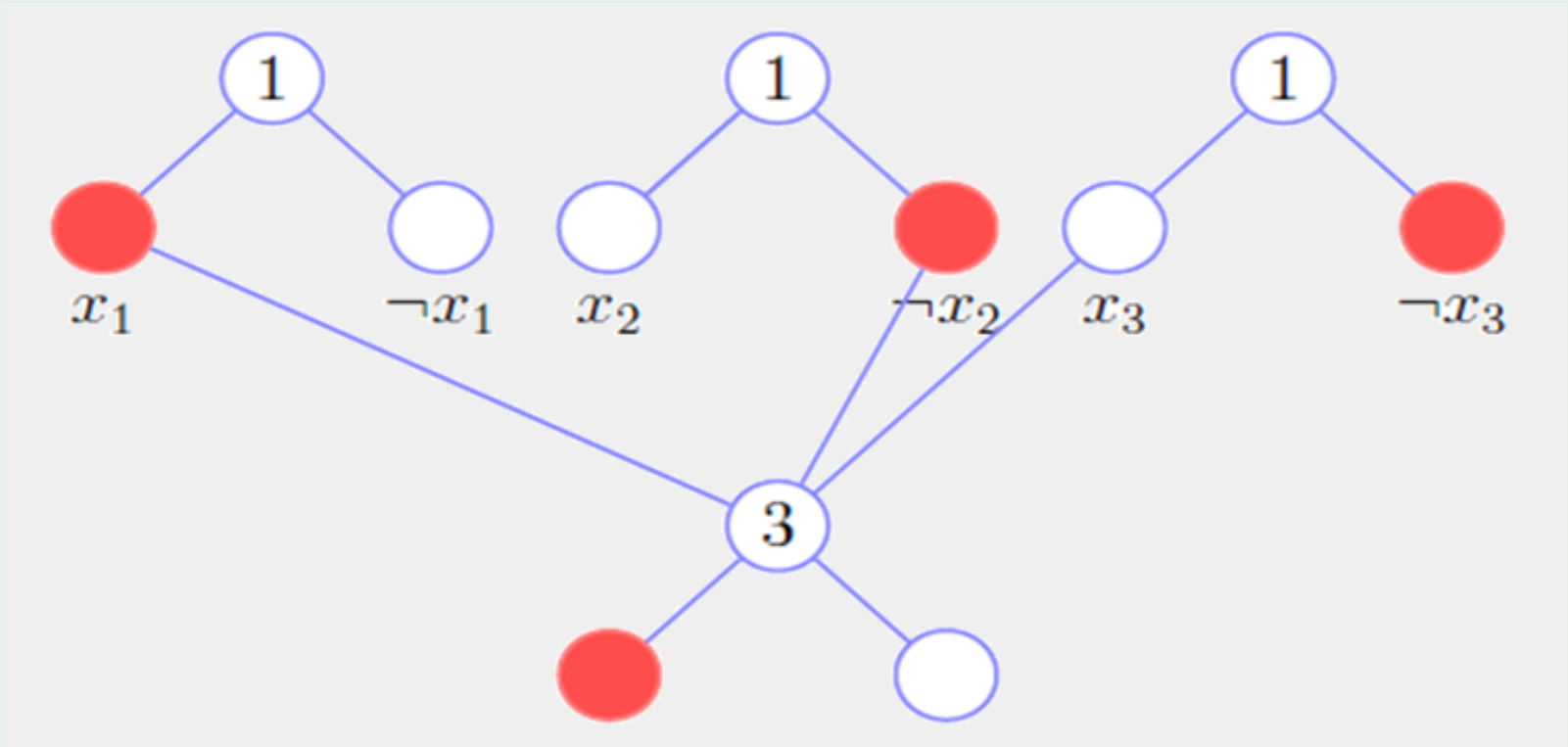
## Aristas desde $c_j$ hasta los nodos de los tres literales de $c_j$



# Ejemplo

- Dada una asignación $\{x_1 = true, x_2 = false, x_3 = false\}$, que satisface a $\phi = (x_1 \vee \bar{x}_2 \vee x_3)$, podemos encontrar una asignación satisfacible:

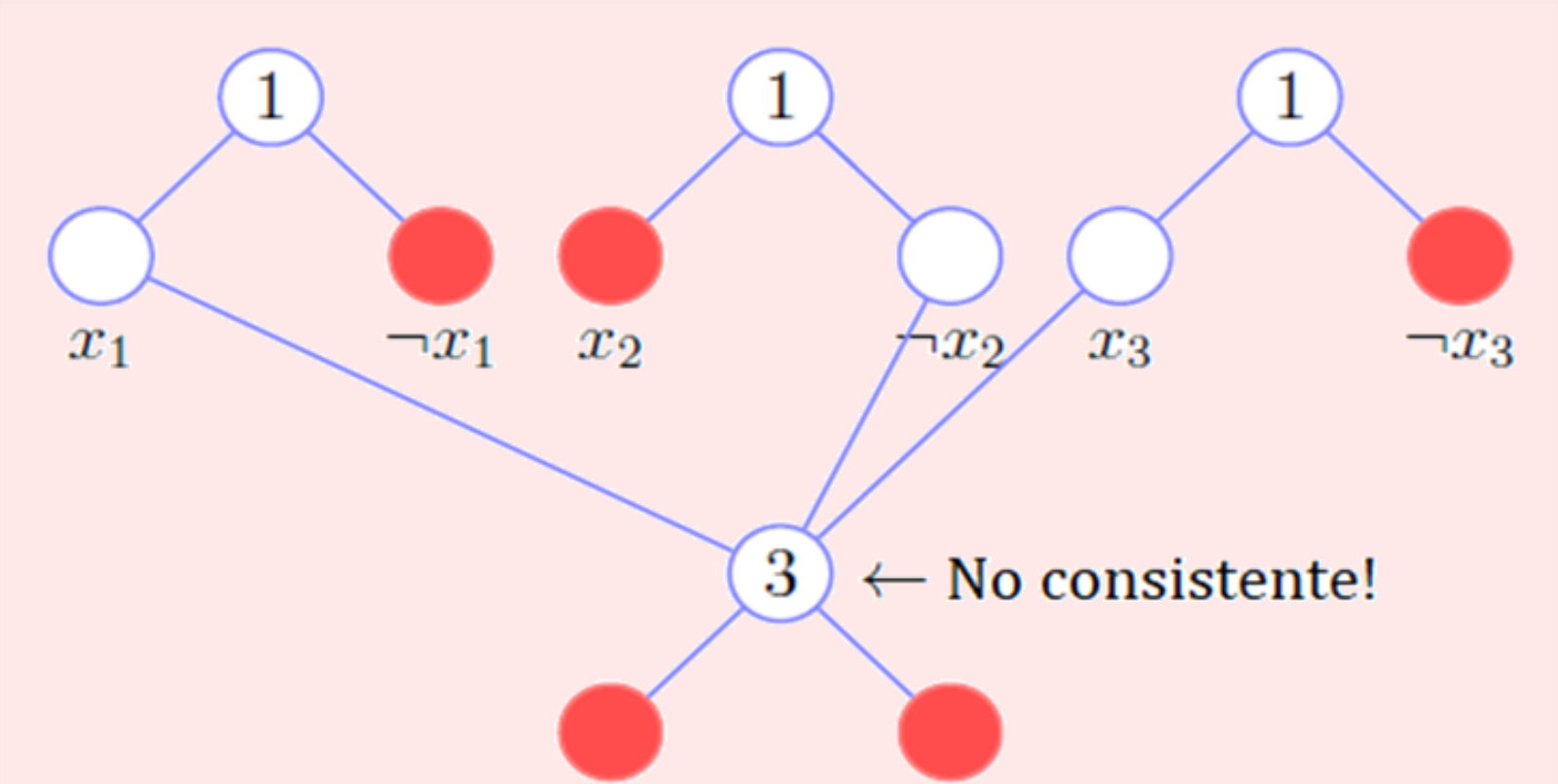
## Colocación de minas en un ejemplo consistente



# Ejemplo

- Dada una asignación $\{x_1 = false, x_2 = true, x_3 = false\}$, que no satisface a $\phi = (x_1 \vee \bar{x}_2 \vee x_3)$, no podemos encontrar una asignación satisfacible:

## Colocación de minas en un ejemplo inconsistente



# Fuentes

- CS4102 - Algorithms
- CS - CMU
- CS - MIT
- bitjoy