Unidad I: Lógica proposicional

Lógica proposicional: Satisfacibilidad

Clase 03 - Matemáticas Discretas (IIC1253)

Prof. Miguel Romero

Fórmulas satisfacibles

Definición:

Una fórmula φ es **satisfacible** si **existe** una valuación σ tal que $\sigma(\varphi) = 1$.

Ejercicio:

¿Cuáles de las siguientes fórmulas son satisfacibles?

- $p \wedge \neg p$ ✗
- $p \wedge (p \rightarrow q)$ ✓
- $p \wedge (p \rightarrow q) \wedge \neg q$ ✗
- $(x \vee \neg y) \wedge (y \vee \neg z) \wedge (z \vee \neg x)$ ✓
- $(x \wedge \neg z \wedge z) \vee (y \wedge \neg x \wedge \neg z)$ ✓

¿Cómo se ve la tabla de verdad de una fórmula satisfacible?

El problema de satisfacibilidad

Problema:

Dada una fórmula proposicional φ, verificar que φ es satisfacible.

¿Puede dar un algoritmo para resolver este problema?

- ¿Cuántas operaciones realiza su algoritmo para una fórmula con n variables?
- ¿Puede dar un algoritmo que realice n^k operaciones, donde k es una constante?
  - A esto se le llama un algoritmo de tiempo polinomial.

No se sabe si existe un algoritmo polinomial para este problema.

- Este problema es considerado el problema abierto más importante en ciencia de la computación.
  - También es llamado el problema **P vs NP**.
- Este problema también es fundamental en matemáticas.

https://www.claymath.org/millennium-problems/

El problema de satisfacibilidad y el poder expresivo de la lógica proposicional

¿Por qué es tan importante el problema de satisfacibilidad?

- Muchos problemas en ciencia de la computación y otras disciplinas se pueden resolver utilizando del problema de satisfacibilidad.

La lógica proposicional es un lenguaje muy expresivo!

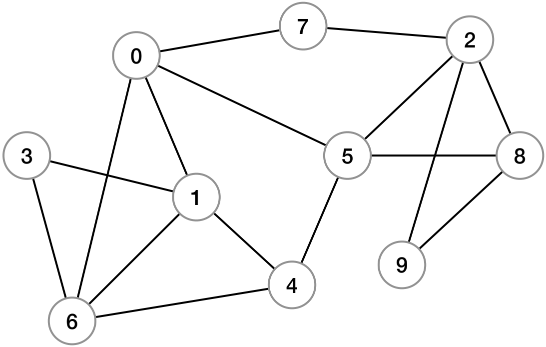
Un problema de asignación de salas

Considere el siguiente problema:

- Debemos repartir a un grupo de alumnos entre 3 salas.
- Algunas parejas de alumnos no se llevan muy bien, luego, deben ir a salas distintas.
  - Hay una lista con las parejas prohibidas, es decir, parejas de alumnos que deben ir a salas distintas.
- ¿Existe alguna forma de asignarle salas a todos los alumnos?

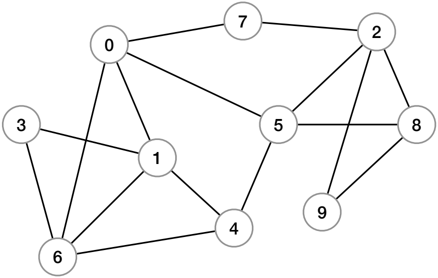
¿Puede dar un algoritmo pata este problema?
¿Cuántas operaciones hace su algoritmo para un problema con n alumnos?

Un ejemplo con 10 alumnos:



Los links entre alumnos representan la lista de parejas prohibidas.

¿Existe una solución? ¿Cómo expresamos esto en lógica proposicional?



Variables proposicionales:

- $p_{i,j}$: los alumnos i y j están en la lista de parejas prohibidas, donde $0 \leq i < j \leq 9$.
- $x_{i,c}$: el alumno i va a la sala c, donde $0 \leq i \leq 9$ y $1 \leq c \leq 3$.

Usamos fórmulas para modelar las restricciones de nuestro problema:

- A cada alumno i se le debe asignar una sala:

$$x_{i,1} \vee x_{i,2} \vee x_{i,3}$$

- No se le puede asignar más de una sala a un alumno i:

$$\neg(x_{i,1} \wedge x_{i,2}) \wedge \neg(x_{i,1} \wedge x_{i,3}) \wedge \neg(x_{i,2} \wedge x_{i,3})$$

Usamos fórmulas para modelar las restricciones de nuestro problema:

- Si los alumnos i y j están en la lista de parejas prohibidas, entonces se les asigna salas distintas:

$$(p_{i,j} \to \neg(x_{i,1} \wedge x_{j,1})) \wedge (p_{i,j} \to \neg(x_{i,2} \wedge x_{j,2})) \wedge (p_{i,j} \to \neg(x_{i,3} \wedge x_{j,3}))$$

- Si los alumnos i y j están en la lista de parejas prohibidas, entonces $p_{i,j}$ debe ser verdadero:

$$\bigwedge_{\substack{i<j \\ i \text{ y } j \text{ están en la lista prohibida}}} p_{i,j}$$

La fórmula completa para nuestro ejemplo sería:

$$\bigwedge_{i=0}^{9} (x_{i,1} \lor x_{i,2} \lor x_{i,3}) \land$$

$$\bigwedge_{i=0}^{9} \Big(\neg(x_{i,1} \land x_{i,2}) \land \neg(x_{i,1} \land x_{i,3}) \land \neg(x_{i,2} \land x_{i,3})\Big) \land$$

$$\bigwedge_{0 \le i < j \le 9} \Big((p_{i,j} \to \neg(x_{i,1} \land x_{j,1})) \land (p_{i,j} \to \neg(x_{i,2} \land x_{j,2})) \land (p_{i,j} \to \neg(x_{i,3} \land x_{j,3}))\Big) \land$$

$$\Big(p_{0,1} \land p_{0,5} \land p_{0,6} \land p_{0,7} \land p_{1,3} \land p_{1,4} \land p_{1,6} \land p_{2,5} \land p_{2,7} \land p_{2,8} \land p_{2,9} \land p_{3,6} \land$$

$$p_{4,5} \land p_{4,6} \land p_{5,8} \land p_{8,9}\Big)$$

¿Y ahora qué hacemos con esta fórmula?

Podemos usar un SAT solver:

- Un programa para verificar si una fórmula proposicional es satisfacible.
- Esta tecnología funciona muy bien en la práctica!

Usemos el SAT solver **Z3** para buscar una solución a nuestro problema.

Paréntesis: producto cartesiano

Dado dos conjuntos A y B, se define:

$$A \times B = \{(a, b) \mid a \in A \text{ y } b \in B\}$$

Ejemplo:

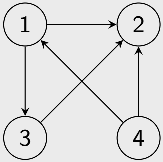
Si $A = \{1, 2\}$ y $B = \{1, 2, 4\}$, entonces:

$$A \times B = \{(1, 1), (1, 2), (1, 4), (2, 1), (2, 2), (2, 4)\}$$

Un **grafo** es un par $G = (V, E)$, donde:

- V es el conjunto de nodos.
- $E \subseteq V \times V$ es el conjunto de arcos.

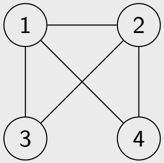
Ejemplo:



$V = \{1, 2, 3, 4\}$
$E = \{(1, 2), (1, 3), (3, 2), (4, 1), (4, 2)\}$

Un grafo $G = (V, E)$ es un **grafo no dirigido** si para cada $(u, v) \in E$, se tiene que $(v, u) \in E$.

Ejemplo:



$V = \{1, 2, 3, 4\}$

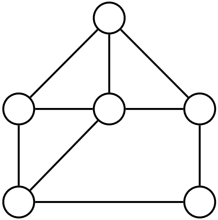
$E = \{(1, 2), (2, 1), (1, 3), (3, 1), (3, 2), (2, 3), (4, 1), (1, 4), (4, 2), (2, 4)\}$

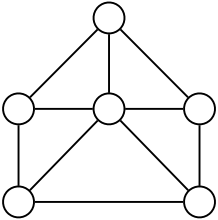
Un grafo no dirigido $G = (V, E)$ es ***k*-coloreable** si **existe** una función $c : V \to \{1, \ldots, n\}$ tal que:

$$\text{para cada } (u, v) \in E, \text{ se tiene que } c(u) \neq c(v).$$

Es decir: nodos adyacentes recibe colores distintos.

Ejemplo: ¿Cuáles de estos grafos tienen un 3-coloreo?





¿En qué se parece el problema de coloración en grafos al problema anterior?

- El problema anterior se puede escribir como un problema de 3-coloreo en grafos (¿cierto?)

Veamos como expresar en general el problema de k-coloreo en lógica proposicional.

Supongamos que $G = (V, E)$, donde $V = \{1, \ldots, n\}$.

Variables proposicionales:

- $a_{i,j}$: hay un arco entre i y j, donde $1 \leq i < j \leq n$.
- $x_{i,c}$: el nodo i recibe el color c, donde $1 \leq i \leq n$ y $1 \leq c \leq k$.

Usamos las siguientes fórmulas para expresar el problema de k-coloreo:

- Cada nodo i recibe un único color:

$$\bigwedge_{i=1}^{n} \bigvee_{c=1}^{k} \Big(x_{i,c} \wedge \bigwedge_{d \neq c} \neg x_{i,d} \Big)$$

- Si hay un arco entre los nodos i y j, entonces reciben colores distintos:

$$\bigwedge_{i=1}^{n} \bigwedge_{j=1}^{n} \bigwedge_{c=1}^{k} \Big((a_{i,j} \wedge x_{i,c}) \to \neg x_{j,c} \Big)$$

- Las variables $a_{i,j}$ se deben hacer verdaderas cuando (i,j) es un arco:

$$\bigwedge_{(i,j) \in E} a_{i,j}$$

La fórmula final φ es la conjunción de las fórmulas anteriores.

G es k-coloreable si y sólo si φ es satisfacible.

Un **clique** de un grafo no dirigido $G = (V, E)$ es un subconjunto de nodos $C \subseteq V$ tal que:

para cada par de nodos $u \neq v$ en C, se tiene que $(u, v) \in E$

Es decir: Todos los pares de nodos en C están conectados por un arco.

Problema:
Dado un grafo no dirigido $G = (V, E)$ y un número $k \geq 0$,
verificar si G tiene un clique C con k nodos.

¿Cómo representaría este problema en lógica proposicional?