

Algoritmia y Complejidad

Titulación: Grado en Ingeniería Informática

Curso: 2023-2024

Trabajo: Ejercicios Sipser. Ejercicio 7.24

Autores: Daniel Sánchez Triviño

Rocío Sánchez Cerván

Índice

- Presentación del problema
- Idea previa
- Construcción grafo G
- Equivalencia G y Φ
- Valores de verdad de Φ
- Cálculo complejidad temporal
- Conclusión

Presentación del problema

Demostrar que **2SAT** \in P

2SAT = $\{ \langle \Phi \rangle \mid \Phi \text{ es una fórmula satisfacible en forma normal conjuntiva (FNC) con dos literales} \}$

Las fórmulas Φ serán de la forma:

$(a_1 \vee b_1) \wedge (a_2 \vee b_2) \wedge \dots \wedge (a_l \vee b_l)$ l cláusulas

Los literales de estas serán:

$x_1, x_2, x_3, \dots, x_m$ m posibles literales

Idea previa

Toda cláusula de una fórmula en FNC con dos literales, se puede expresar como una implicación

$$(x \rightarrow y) \equiv (\neg x \vee y)$$

A	B	$A \Rightarrow B$
V	V	V
V	F	F
F	V	V
F	F	V

Si queremos las clausulas como $(x \vee y)$:

$$(\neg x \rightarrow y) \equiv (\neg \neg x \vee y) \equiv (x \vee y)$$

$$(\neg y \rightarrow x) \equiv (\neg \neg y \vee x) \equiv (y \vee x) \equiv (x \vee y)$$

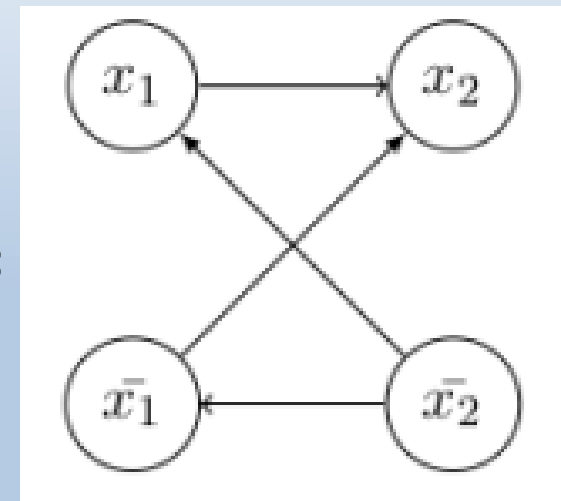
Construcción grafo G

Teniendo en cuenta las equivalencias, se construye un grafo dirigido G como:

- Dibujar un nodo por cada literal de la fórmula junto con sus complementarios.
- Para cada clausula $(x_i \vee x_j)$, dibujar un arco entre $\neg x_i x_j$ y $\neg x_j x_i$, debido a la equivalencia con la implicación.

$$\Phi_1 = (x_1 \vee x_2) \wedge (\neg x_1 \vee x_2)$$

$$G_1 =$$

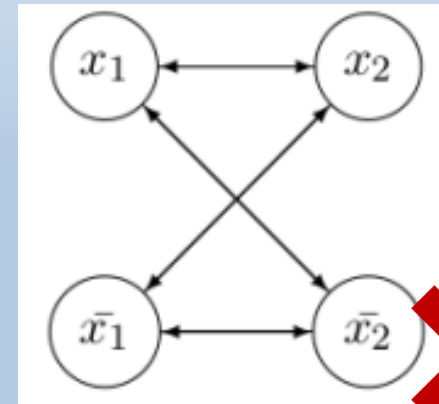
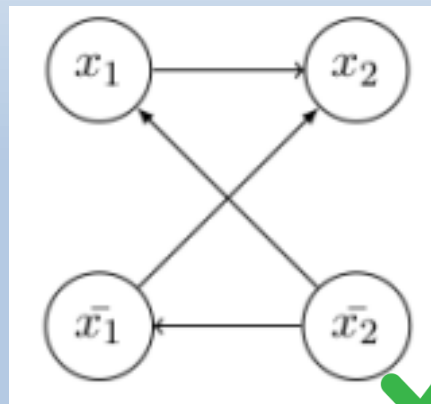


Equivalencia G y Φ

- G no contiene ningún ciclo con las variables x_i y $\neg x_i$ para cualquier $i \rightarrow \Phi$ es satisfacible
- G contiene algún ciclo con las variables x_i y $\neg x_i$ para algún $i \rightarrow \Phi$ es insatisfacible

Explicación: G es dirigido, simula las implicaciones. Si un ciclo contiene x_i y $\neg x_i$, significa $x_i \leftrightarrow \neg x_i$, que es contradictorio.

p	q	$p \leftrightarrow q$
T	T	T
T	F	F
F	T	F
F	F	T

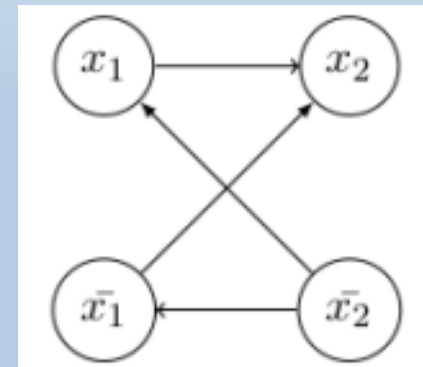


Valores de verdad de Φ

- Selecciono una variable x_i
- Selecciono el literal x_i si no puedo ir de x_i a $\neg x_i$, sino elijo $\neg x_i$
- Asigno TRUE a todos los literales alcanzables desde el literal seleccionado
- Elimino de G todos los nodos etiquetados con literales ya asignados, y sus complementarios. Repito hasta que todas las variables estén asignadas.

$$\Phi_1 = (x_1 \vee x_2) \wedge (\neg x_1 \vee x_2)$$

$$\mathbf{x_1 = TRUE, x_2 = TRUE}$$



Cálculo complejidad temporal

Φ supongo tamaño n . Condiciona la construcción de G .

Pasos:

- Recorrer Φ para dibujar los nodos $\approx O(n)$
- Colocar arcos $\approx O(n^2)$
- Comprobar ciclos $\approx O(n^2)$
- Seleccionar variable $\approx O(1)$
- Comprobar llegar a complementario desde la variable $\approx O(n)$
- Asignar TRUE a los literales alcanzables $\approx O(n)$
- Eliminar nodos de literales asignados $\approx O(n)$

Se repite todo k veces con $k \geq 1$

$$T(n) = O(k * n^2)$$

Conclusión

2SAT \in P

Existe un algoritmo en tiempo polinómico que de manera determinista decide si una fórmula de 2 literales en FNC es satisfacible o no.

