Fundamentos de la Computación
TC4001

Problemas NP-Completos

Centro de Manufactura / Centro de Sistema Inteligentes

ITESM

Reducción: Idea

Suponga que se desea resolver el problema **P** y que se tiene un algoritmo para resolver el problema **Q**.

Reducción: Idea

Suponga que se desea resolver el problema **P** y que se tiene un algoritmo para resolver el problema **Q**. Suponga que tiene un procedimiento o función T que toma una x entrada para **P** y produce $T(x)$, una entrada para **Q**, muy especial:

Reducción: Idea

Suponga que se desea resolver el problema **P** y que se tiene un algoritmo para resolver el problema **Q**. Suponga que tiene un procedimiento o función T que toma una x entrada para **P** y produce $T(x)$, una entrada para **Q**, muy especial: Que la respuesta para **P** con x sería **sí** si y sólo si la respuesta para **Q** con $T(x)$ es **sí**.

Reducción: Idea

Suponga que se desea resolver el problema **P** y que se tiene un algoritmo para resolver el problema **Q**. Suponga que tiene un procedimiento o función T que toma una x entrada para **P** y produce $T(x)$, una entrada para **Q**, muy especial: Que la respuesta para **P** con x sería **sí** si y sólo si la respuesta para **Q** con $T(x)$ es **sí**.
Entonces componiento el procedimiento T y el algoritmo de solución a **Q** se tiene un algoritmo de solución para **P**.

Sea T una función que establece una correspondencia entre el conjunto de entrada de un problema de decisión **P** y el conjunto de entradas de un problema de decisión **Q**.

Sea T una función que establece una correspondencia entre el conjunto de entrada de un problema de decisión **P** y el conjunto de entradas de un problema de decisión **Q**. Se dice que T es una reducción o transformación polinómica de **P** a **Q** si se cumplen:

Sea T una función que establece una correspondencia entre el conjunto de entrada de un problema de decisión **P** y el conjunto de entradas de un problema de decisión **Q**. Se dice que T es una reducción o transformación polinómica de **P** a **Q** si se cumplen:

- T se puede calcular en tiempo polinómicamente acotado.

Sea T una función que establece una correspondencia entre el conjunto de entrada de un problema de decisión **P** y el conjunto de entradas de un problema de decisión **Q**. Se dice que T es una reducción o transformación polinómica de **P** a **Q** si se cumplen:

- T se puede calcular en tiempo polinómicamente acotado.
- Para toda cadena x, si x es una entrada de **sí** para **P**, entonces $T(x)$ es una entrada de **sí** para **Q**.

Reducción

Sea T una función que establece una correspondencia entre el conjunto de entrada de un problema de decisión **P** y el conjunto de entradas de un problema de decisión **Q**. Se dice que T es una reducción o transformación polinómica de **P** a **Q** si se cumplen:

- T se puede calcular en tiempo polinómicamente acotado.
- Para toda cadena x, si x es una entrada de **sí** para **P**, entonces $T(x)$ es una entrada de **sí** para **Q**.
- Para toda cadena x, si $T(x)$ es una entrada de **sí** para **Q**, entonces x debe ser una entrada de **sí** para **P**.

Problema Reducible

Una problema **P** se dice polinómicamente reducible a **Q** si existe una transformación polinómica de **P** a **Q**.

Problema Reducible

Una problema **P** se dice polinómicamente reducible a **Q** si existe una transformación polinómica de **P** a **Q**. Se dirá **P** se puede reducir a **Q**.

Problema Reducible

Una problema **P** se dice polinómicamente reducible a **Q** si existe una transformación polinómica de **P** a **Q**. Se dirá **P** se puede reducir a **Q**.

Notación:

$$\boxed{\mathbf{P} \leq_p \mathbf{Q}}$$

Problema Reducible

Una problema **P** se dice polinómicamente reducible a **Q** si existe una transformación polinómica de **P** a **Q**. Se dirá **P** se puede reducir a **Q**.

Notación:

$$\boxed{\mathbf{P} \leq_p \mathbf{Q}}$$

Idea:

Q es al menos tan difícil como **P**.

Problema Reducible

Una problema **P** se dice polinómicamente reducible a **Q** si existe una transformación polinómica de **P** a **Q**. Se dirá **P** se puede reducir a **Q**.

Notación:

$$\boxed{\mathbf{P} \leq_p \mathbf{Q}}$$

Idea:

Q es al menos tan difícil como **P**.

Resultado

Si $\mathrm{P} \leq_p \mathrm{Q}$ y Q está en P, entonces P está en P.

NP-Difícil y NP-Completo

- Un problema **Q** se dice NP-Difícil si todo problema **P** de NP se puede reducir a **Q**.

NP-Difícil y NP-Completo

- Un problema **Q** se dice NP-Difícil si todo problema **P** de NP se puede reducir a **Q**. **Q** no necesariamente está en NP.

NP-Difícil y NP-Completo

- Un problema **Q** se dice NP-Difícil si todo problema **P** de NP se puede reducir a **Q**.
  Q no necesariamente está en NP.
- Un problema **Q** se dice NP-Completo si **Q** está en NP y es NP-Difícil.

NP-Difícil y NP-Completo

- Un problema **Q** se dice NP-Difícil si todo problema **P** de NP se puede reducir a **Q**.
  Q no necesariamente está en NP.
- Un problema **Q** se dice NP-Completo si **Q** está en NP y es NP-Difícil.

Consecuencias:

- Si **P** es NP-Completo y se puede encontrar un algoritmo polinomial para **P**, entonces cualquier problema de la clase NPC se puede resolver polinomialmente.

NP-Difícil y NP-Completo

- Un problema **Q** se dice NP-Difícil si todo problema **P** de NP se puede reducir a **Q**.
  Q no necesariamente está en NP.
- Un problema **Q** se dice NP-Completo si **Q** está en NP y es NP-Difícil.

Consecuencias:

- Si **P** es NP-Completo y se puede encontrar un algoritmo polinomial para **P**, entonces cualquier problema de la clase NPC se puede resolver polinomialmente.
- Si $\mathbf{P}$ está en NPC, $\mathbf{Q}$ está en NP, y $\mathbf{P} \leq_p \mathbf{Q}$ entonces $\mathbf{Q}$ está en NPC.

SAT está en NPC

Teorema(Teorema de Cook, 1971)

- SAT está en NP
- Cualquier problema en la clase NP se puede convertir polinomialmente en una instancia SAT

3COL $\leq_p$ SAT

Dado un grafo $G = (V, E)$ digamos que $n = |V|$ y $m = |E|$, y consideremos la pregunta : ¿G puede colorearse con colores rojo, azul y blanco?

3COL $\leq_p$ SAT

Dado un grafo $G = (V, E)$ digamos que $n = |V|$ y $m = |E|$, y consideremos la pregunta : ¿G puede colorearse con colores rojo, azul y blanco? Indiquemos un procedimiento para partiendo de G se construya una instancia para SAT.

3COL $\leq_p$ SAT

Dado un grafo $G = (V, E)$ digamos que $n = |V|$ y $m = |E|$, y consideremos la pregunta : ¿G puede colorearse con colores rojo, azul y blanco? Indiquemos un procedimiento para partiendo de G se construya una instancia para SAT. Para cada vértice v_i de V se definen 3 variables booleanas R_i, A_i y W_i.

3COL $\leq_p$ SAT

Dado un grafo $G = (V, E)$ digamos que $n = |V|$ y $m = |E|$, y consideremos la pregunta : ¿G puede colorearse con colores rojo, azul y blanco? Indiquemos un procedimiento para partiendo de G se construya una instancia para SAT. Para cada vértice v_i de V se definen 3 variables booleanas R_i, A_i y W_i. La variable con valor booleano verdadero indicará el color del nodo.

Dado un grafo $G = (V, E)$ digamos que $n = |V|$ y $m = |E|$, y consideremos la pregunta : ¿G puede colorearse con colores rojo, azul y blanco? Indiquemos un procedimiento para partiendo de G se construya una instancia para SAT. Para cada vértice v_i de V se definen 3 variables booleanas R_i, A_i y W_i. La variable con valor booleano verdadero indicará el color del nodo. Construyamos las cláusulas de la expresión en su CNF, será un AND entre:

3COL $\leq_p$ SAT

Dado un grafo $G = (V, E)$ digamos que $n = |V|$ y $m = |E|$, y consideremos la pregunta : ¿G puede colorearse con colores rojo, azul y blanco? Indiquemos un procedimiento para partiendo de G se construya una instancia para SAT. Para cada vértice v_i de V se definen 3 variables booleanas R_i, A_i y W_i. La variable con valor booleano verdadero indicará el color del nodo. Construyamos las cláusulas de la expresión en su CNF, será un AND entre:

- Para cada $i = 1, \ldots, n$:

$$(R_i \vee A_i \vee B_i)$$

Dado un grafo $G = (V, E)$ digamos que $n = |V|$ y $m = |E|$, y consideremos la pregunta : ¿G puede colorearse con colores rojo, azul y blanco? Indiquemos un procedimiento para partiendo de G se construya una instancia para SAT. Para cada vértice v_i de V se definen 3 variables booleanas R_i, A_i y W_i. La variable con valor booleano verdadero indicará el color del nodo. Construyamos las cláusulas de la expresión en su CNF, será un AND entre:

- Para cada $i = 1, \ldots, n$:

$$(R_i \vee A_i \vee B_i), (\bar{R}_i \vee \bar{A}_i)$$

Dado un grafo $G = (V, E)$ digamos que $n = |V|$ y $m = |E|$, y consideremos la pregunta : ¿G puede colorearse con colores rojo, azul y blanco? Indiquemos un procedimiento para partiendo de G se construya una instancia para SAT. Para cada vértice v_i de V se definen 3 variables booleanas R_i, A_i y W_i. La variable con valor booleano verdadero indicará el color del nodo. Construyamos las cláusulas de la expresión en su CNF, será un AND entre:

- Para cada $i = 1, \ldots, n$:

$$(R_i \vee A_i \vee B_i), (\bar{R}_i \vee \bar{A}_i), (\bar{R}_i \vee \bar{B}_i)$$

Dado un grafo $G = (V, E)$ digamos que $n = |V|$ y $m = |E|$, y consideremos la pregunta : ¿G puede colorearse con colores rojo, azul y blanco? Indiquemos un procedimiento para partiendo de G se construya una instancia para SAT. Para cada vértice v_i de V se definen 3 variables booleanas R_i, A_i y W_i. La variable con valor booleano verdadero indicará el color del nodo. Construyamos las cláusulas de la expresión en su CNF, será un AND entre:

- Para cada $i = 1, \ldots, n$:

$$(R_i \vee A_i \vee B_i), (\bar{R}_i \vee \bar{A}_i), (\bar{R}_i \vee \bar{B}_i), (\bar{A}_i \vee \bar{B}_i)$$

- Para cada lado (i, j) de E:

$$(\bar{R}_i \vee \bar{R}_j), (\bar{A}_i \vee \bar{A}_j), (\bar{B}_i \vee \bar{B}_j)$$

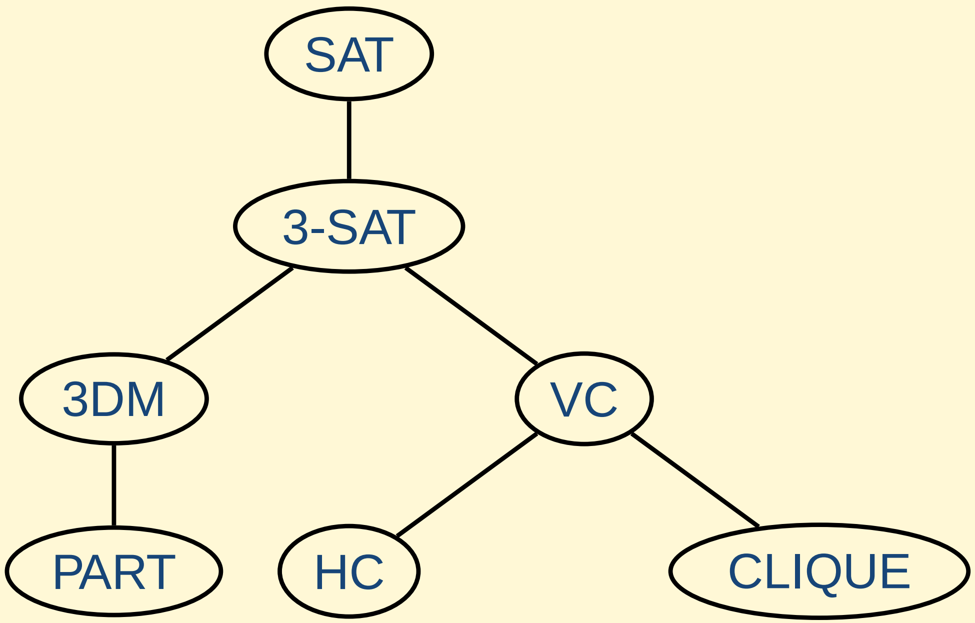
- Para cada lado (i,j) de E:

$$(\bar{R}_i \vee \bar{R}_j), (\bar{A}_i \vee \bar{A}_j), (\bar{B}_i \vee \bar{B}_j)$$

La instancia de SAT tendrá:

- $3\,n$ variables boolenas y
- la CNF tendrá $4\,n + 3\,m$ cláusulas.

Lista de Karp (1972)



Revise la liga:

http://en.wikipedia.org/wiki/Karp%27s_21_NP-complete_problems

3-SAT es NPC

- 3-SAT está en NP
- SAT $\leq_p$ 3-SAT

3-SAT es NPC

- 3-SAT está en NP
- SAT $\leq_p$ 3-SAT

Veamos cómo convertir una instancia de SAT en una instancia de 3-SAT (cada cláusula con exactamente 3 literales).

3-SAT es NPC

- 3-SAT está en NP
- SAT $\leq_p$ 3-SAT
Veamos cómo convertir una instancia de SAT en una instancia de 3-SAT (cada cláusula con exactamente 3 literales). Supongamos que la fórmula está compuesta por las cláusulas $C_1, C_2, \ldots, C_m$.

3-SAT es NPC

- 3-SAT está en NP
- SAT $\leq_p$ 3-SAT

Veamos cómo convertir una instancia de SAT en una instancia de 3-SAT (cada cláusula con exactamente 3 literales). Supongamos que la fórmula está compuesta por las cláusulas C_1,C_2,. . . ,C_m. La fórmula nueva a construir a partir de las cláusulas C_i:

3-SAT es NPC

- 3-SAT está en NP
- SAT $\leq_p$ 3-SAT

Veamos cómo convertir una instancia de SAT en una instancia de 3-SAT (cada cláusula con exactamente 3 literales). Supongamos que la fórmula está compuesta por las cláusulas C_1,C_2,...,C_m. La fórmula nueva a construir a partir de las cláusulas C_i:

- Si C_i contiene 3 literales la usamos tal cual.

3-SAT es NPC

- 3-SAT está en NP
- SAT $\leq_p$ 3-SAT

Veamos cómo convertir una instancia de SAT en una instancia de 3-SAT (cada cláusula con exactamente 3 literales). Supongamos que la fórmula está compuesta por las cláusulas C_1,C_2,...,C_m. La fórmula nueva a construir a partir de las cláusulas C_i:

- Si C_i contiene 3 literales la usamos tal cual.
- Si $C_i = z$,añadimos dos nuevas variables x_i y y_i booleanas y cambiamos C_i por las cláusulas:

$$(z \vee x_i \vee y_i), (z \vee x_i \vee \bar{y_i}), (z \vee \bar{x_i} \vee y_i), (z \vee \bar{x_i} \vee \bar{y_i})$$

$$CC = (z \vee x_i \vee y_i) \wedge (z \vee x_i \vee \bar{y}_i) \wedge (z \vee \bar{x}_i \vee y_i) \wedge (z \vee \bar{x}_i \vee \bar{y}_i)$$

z	x_i	y_i	$z \vee x_i \vee y_i$	$z \vee x_i \vee \bar{y}_i$	$z \vee \bar{x}_i \vee y_i$	$z \vee \bar{x}_i \vee \bar{y}_i$	CC
0	0	0	0	1	1	1	0
0	0	1	1	0	1	1	0
0	1	0	1	1	0	1	0
0	1	1	1	1	1	0	0
1	0	0	1	1	1	1	1
1	0	1	1	1	1	1	1
1	1	0	1	1	1	1	1
1	1	1	1	1	1	1	1

z es satisfacible $\leftrightarrow CC$ es satisfacible

- Si $C_i = z \vee w$ añadimos una nueva variable boolena x_i y cambiamos a C_i por las cláusulas:

$$(z \vee w \vee x_i), (z \vee w \vee \bar{x}_i)$$

$$CC = (z \vee w \vee x_i) \wedge (z \vee w \vee \bar{x}_i)$$

z	w	x_i	$z \vee w \vee x_i$	$z \vee w \vee \bar{x}_i$	CC	$w \vee z$
0	0	0	0	1	0	0
0	0	1	1	0	0	0
0	1	0	1	1	1	1
0	1	1	1	1	1	1
1	0	0	1	1	1	1
1	0	1	1	1	1	1
1	1	0	1	1	1	1
1	1	1	1	1	1	1

$z \vee w$ es satisfacible $\leftrightarrow$ CC es satisfacible

- Si $C_i = z_1 \vee z_2 \vee \cdots \vee z_k$ tiene $k > 3$ literales añadimos $k-3$ nuevas variables x_1,x_2,. . . ,x_{k-3}, y cambiamos a C_i por las cláusulas:

$$
\begin{gathered}
z_1 \vee z_2 \vee x_1 \\
\bar{x}_1 \vee z_3 \vee x_2 \\
\bar{x}_2 \vee z_4 \vee x_3 \\
\vdots \\
\bar{x}_{k-4} \vee z_{k-2} \vee x_{k-3} \\
\bar{x}_{k-3} \vee z_{k-1} \vee z_k
\end{gathered}
$$

Para $k = 4$ (4 literales en la cláusula $\alpha = z_1 \vee z_2$ y $\beta = z_3 \vee z_4$):

$$CC = (\alpha \vee x_i) \wedge (\bar{x}_i \vee \beta)$$

α	β	x_i	$\alpha \vee x_i$	$\bar{x}_i \vee \beta$	CC	$\alpha \vee \beta$
0	0	0	0	1	0	0
0	0	1	1	0	0	0
0	1	0	0	1	0	1
0	1	1	1	1	1	1
1	0	0	1	1	1	1
1	0	1	1	0	0	1
1	1	0	1	1	1	1
1	1	1	1	1	1	1

$\alpha \vee \beta$ es satisfacible $\leftrightarrow$ CC es satisfacible

Algunos problemas NPC

Teorema

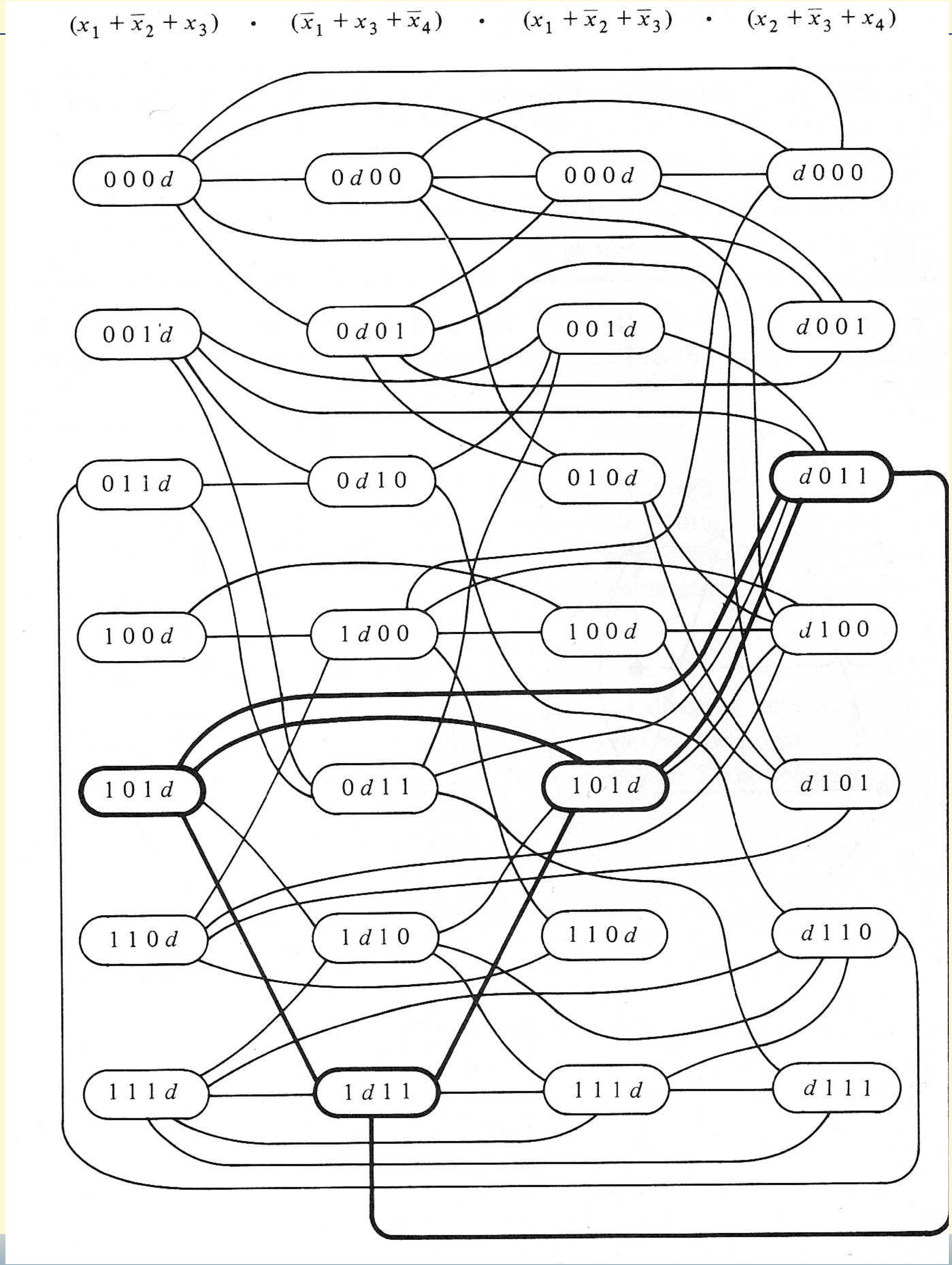
El problema de saber si un grafo $G(V,E)$ tiene un subgrafo completo de tamaño k es NP completo (CLIQUE).

Ideas

- 3-SAT se reduce a CLIQUE
- Asignación parcial de valores de verdad
- Asignaciones parciales compatibles

continuación. . .

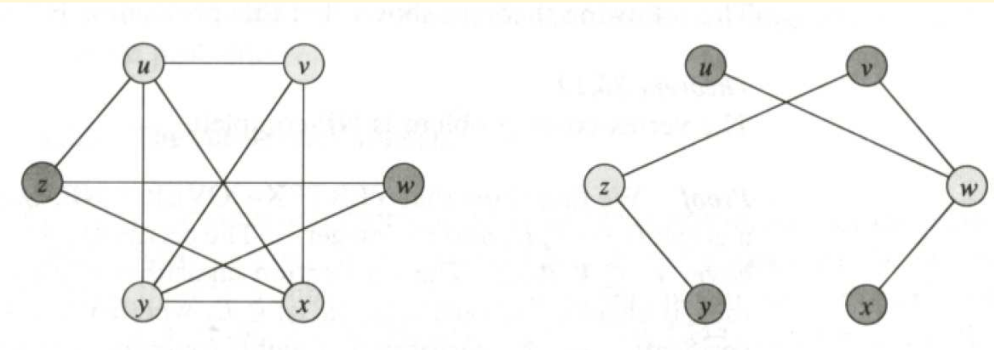
- A cada cláusula origina 7 vértices, cada uno de los cuales representa una asignación parcial: para las 3 variables presentes en una cláusula de las 8 combinaciones posibles de valores a las variables se omite el valor que la hace falsa la claúsula y a las variables no presentes se les asigna d (indefinido).
- Cada vértice se conecta con otro si las asignaciones parciales que representan los vértices son compatibles. ¿Total de lados?
- Un clique de tamaño k representa una asignación que hace verdadera las k cláusulas.
- Una asignación que hace verdadera las k cláusulas da un clique de tamaño k.


$(x_1 + \bar{x}_2 + x_3) \cdot (\bar{x}_1 + x_3 + \bar{x}_4) \cdot (x_1 + \bar{x}_2 + \bar{x}_3) \cdot (x_2 + \bar{x}_3 + x_4)$
000d
0d00
000d
d000
001d
0d01
001d
d001
011d
0d10
010d
d011
100d
1d00
100d
d100
101d
0d11
101d
d101
110d
1d10
110d
d110
111d
1d11
111d
d111

Lema

Las siguientes tres afirmaciones son equivalentes para un grafo $G = (V, E)$ y un subconjunto S de vértices de V:

1. S es un subgrafo completo máximo de G.
2. S es un subconjunto independiente del grafo complemento de G.
3. $V - S$ es una cubierta de vértices del grafo complemento de G.



(a) (b)

Algunos problemas NPC

- Coloreo de Grafos
- Circuito Hamiltoniano
- Programación de tareas con castigo (versión de decisión)
- Empacamiento (versión de decisión)
- Problema de la mochila (versión de decisión)
- Satisfactibilidad (versión de decisión)
- Agente Viajero (versión de decisión)

Revise la liga:

http://en.wikipedia.org/wiki/List_of_NP-complete_problems

Dada una 2-CNF ϕ y un entero k determinar si existe una sustitución π que hace que almenos k cláusulas de ϕ se hacen verdaderas.

Teorema

Max2SAT es NP-Completo.

- Max2SAT está NP.
- Por ver: que 3SAT se reduce polinomialmente a Max2SAT.

$$\begin{aligned} \psi(x,y,z,w) &= x \wedge y \wedge z \wedge w \wedge \\ &\quad (\bar{x} \vee \bar{y}) \wedge (\bar{y} \vee \bar{z}) \wedge (\bar{z} \vee \bar{x}) \wedge \\ &\quad (x \vee \bar{w}) \wedge (y \vee \bar{w}) \wedge (z \vee \bar{w}) \\ n &= \text{número de cláusulas en } \psi \text{ con valor 1 en una sustitución} \end{aligned}$$

w	x	y	z	x	y	z	w	$\bar{x} \vee \bar{y}$	$\bar{y} \vee \bar{z}$	$\bar{z} \vee \bar{x}$	$x \vee \bar{w}$	$y \vee \bar{w}$	$z \vee \bar{w}$	$x \vee y \vee z$	n
0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	6
1	0	0	0	0	0	0	1	1	1	1	0	0	0	0	4
0	1	0	0	1	0	0	0	1	1	1	1	1	1	1	7
1	1	0	0	1	0	0	1	1	1	1	1	0	0	1	6
0	0	1	0	0	1	0	0	1	1	1	1	1	1	1	7
1	0	1	0	0	1	0	1	1	1	1	0	1	0	1	6
0	1	1	0	1	1	0	0	0	1	1	1	1	1	1	7
1	1	1	0	1	1	0	1	0	1	1	1	1	0	1	7
0	0	0	1	0	0	1	0	1	1	1	1	1	1	1	7
1	0	0	1	0	0	1	1	1	1	1	0	0	1	1	6
0	1	0	1	1	0	1	0	1	1	0	1	1	1	1	7
1	1	0	1	1	0	1	1	1	1	0	1	0	1	1	7
0	0	1	1	0	1	1	0	1	0	1	1	1	1	1	7
1	0	1	1	0	1	1	1	1	0	1	0	1	1	1	7
0	1	1	1	1	1	1	0	0	0	0	1	1	1	1	6
1	1	1	1	1	1	1	1	0	0	0	1	1	1	1	7

Construcción

Dada una 3-CNF ϕ, construyamos una 2-CNF y un valor de k

- Para cada cláusula i de ϕ, $\alpha \vee \beta \vee \gamma$, construya la correspondiente $\psi(\alpha, \beta, \gamma, w_i)$.
  Para la 3-CNF

$$\phi(n_{\text{variables}}, m_{\text{cláusulas}})$$

  se contruye una 2-CNF

$$\varphi_\phi(n + m_{\text{variables}}, 10\, m_{\text{cláusulas}})$$

- Tome $k = 7\, m$.

Validez

- Todo asignamiento π que haga verdadera la 3-CNF ϕ se puede extender en un asignamiento que hace verdaderas $7m$ cláusulas de la 2-CNF φ_ϕ.
- Si $7m$ cláusulas de la 2-CNF φ_ϕ se pueden hacer verdaderas con una asignación π, entonces cada una de las $\psi(\alpha, \beta, \gamma, w_i)$ tiene 7 cláusulas que son satisfechas, por tanto, tal asignamiento hace verdadera a $\alpha \vee \beta \vee \gamma$ Por tanto, ϕ se satisface con π.