Problemas en P y NP

Marcos Kiwi

U. Chile

Semestre Otoño 2012

Problemas en P

$$
\begin{aligned}
\mathrm{Path} &= \{\langle G, s, t\rangle : \text{Existe un dicamino de } s \text{ a } t \text{ en el digrafo } G\} \\
\mathrm{Conexo} &= \{\langle G\rangle : G \text{ grafo conexo}\} \\
\mathrm{PL} &= \left\{\langle A, b, c, k\rangle : \begin{array}{c} A \in \mathbb{Q}^{m\times n},\ b \in \mathbb{Q}^m,\ c \in \mathbb{Q}^n \text{ y } k \in \mathbb{Q}, \\ \exists x \in \mathbb{Q}^n, \text{ tal que } c^T x \leq k \text{ y } Ax \leq b,\ x \geq 0 \end{array}\right\} \\
\mathrm{Mult} &= \{\langle a, b, c\rangle : a, b, c \in \mathbb{Q} \text{ tal que } ab = c\} \\
\mathrm{Sum} &= \{\langle a, b, c\rangle : a, b, c \in \mathbb{Q} \text{ tal que } a + b = c\} \\
\mathrm{MCD} &= \{\langle a, b, d\rangle : a, b, d \in \mathbb{N} \setminus \{0\} \text{ tal que } \mathrm{mcd}(a, b) = d\} \\
\mathrm{Expon} &= \{\langle a, n, b\rangle : a, b \in \mathbb{Z},\ n \in \mathbb{N} \text{ en unario tal que } a^n = b\} \\
\mathrm{ModExp} &= \left\{\langle a, b, c, m\rangle : \begin{array}{c} a, c \in \mathbb{Z},\ b, m \in \mathbb{N},\ m \neq 0, \\ \text{tal que } a^b = c \ (\mathrm{mód}\ m) \end{array}\right\}
\end{aligned}
$$

Algorithm 1: Exponenciación Modular

input : a, b, m tal que $a \in \mathbb{N}$, $b, m \in \mathbb{N}$, $m \neq 0$.

output: el resto $c \in \{0, \ldots, m-1\}$ de la división de a^b por m.

$c \leftarrow 1$;

while $b > 0$ **do**

  if b *es impar* **then**

    $c \leftarrow (c \cdot a) \mod m$;

  $b \leftarrow \lfloor b/2 \rfloor$;

  $a \leftarrow a^2 \mod m$;

return(c)

Problemas en $\mathrm{NP} \cap \mathrm{coNP}$

$$\begin{aligned} \mathrm{Compst} &= \{\langle n \rangle : n \in \mathbb{N} \text{ es compuesto}\} \\ \mathrm{Primos} &= \{\langle p \rangle : p \in \mathbb{N} \text{ es primo}\} \end{aligned}$$

De hecho, ¡ambos problemas de decisión están en P!

Problemas en NP

$$
\begin{aligned}
\text{Clique} &= \left\{ \langle G, k \rangle : \begin{array}{c} G \text{ es un grafo que posee un clique} \\ \text{de tamaño al menos } k \in \mathbb{N} \end{array} \right\} \\
\text{VC} &= \left\{ \langle G, k \rangle : \begin{array}{c} G \text{ es un grafo que posee un recubrimiento} \\ \text{de nodos de tamaño a lo más } k \in \mathbb{N} \end{array} \right\} \\
\text{3Color} &= \{ \langle G \rangle : G \text{ es grafo } 3 \text{ coloreable} \} \\
\text{HamCycle} &= \{ \langle G \rangle : G \text{ es grafo que posee un circuito Hamiltoniano} \} \\
\text{HamPath} &= \{ \langle G \rangle : G \text{ es grafo que posee un camino Hamiltoniano} \} \\
\text{MaxCut} &= \{ \langle G, k \rangle : G \text{ es grafo que posee un corte } S \text{ tal que } |\delta(S)| \geq k \}
\end{aligned}
$$

Problemas en NP (cont.)

$$\text{SubSum} = \left\{ \langle U, t \rangle : U \subset \mathbb{N},\ t \in \mathbb{N} \text{ tales que } \exists S \subseteq U,\ \sum_{s \in S} s = t \right\}$$

$$\text{Mochila} = \left\{ \langle U, \omega, v, W, V \rangle : \begin{array}{r} \omega : U \to \mathbb{N},\ v : U \to \mathbb{N},\ W, V \in \mathbb{N}, \text{ y} \\ \exists S \subseteq U \text{ tal que } \sum_{s \in S} \omega(s) \leq W, \\ \sum_{s \in S} v(s) \geq V \end{array} \right\}$$

$$\text{TSP} = \left\{ \langle G, \omega, k \rangle : \begin{array}{c} G \text{ grafo, } \omega : E(G) \to \mathbb{N},\ k \in \mathbb{N} \text{ tal que existe} \\ \text{un tour } T \text{ de } G \text{ tal que } \sum_{e \in E(T)} \omega(e) \leq k \end{array} \right\}$$

$$\text{BinPack} = \left\{ \langle U, \omega, k \rangle : \begin{array}{c} \omega : U \to \mathbb{Q} \cap [0, 1],\ k \in \mathbb{N} \text{ tal que existe} \\ \text{una partición } S_1, \ldots, S_k \text{ de } U \text{ para la que} \\ \sum_{s \in S_i} \omega(s) \leq 1 \text{ para todo } i = 1, \ldots, k \end{array} \right\}$$

Fórmulas Booleanas

Una fórmula Booleana φ sobre las variables $x_1, \ldots, x_n$, denotado $\varphi = \varphi(x_1, \ldots, x_n)$ se define recursivamente como:

- x_i es una fórmula Booleana,
- $\neg(\phi)$ es fórmula Booleana si ϕ es fórmula Booleana, y
- $(\phi') \star (\phi'')$ es fórmula Booleana si ϕ' y ϕ'' lo son, donde $\star$ es un operador lógico binario.

Decimos que $\varphi = \varphi(x_1, \ldots, x_n)$ es fórmula Booleana que se puede satisfacer si $\exists a_1, \ldots, a_n \in \{0, 1\}$ tal que $\varphi(a_1, \ldots, a_n) = 1$.

SAT

Se define:

$$\mathrm{SAT} \quad = \quad \{\langle \varphi \rangle : \varphi \text{ es fórmula Booleana que se puede satisfacer}\} \ .$$

Theorem (Cook (1971) & Levin (1973))

SAT*es* NP*-completo.*

Circuitos Booleanos

Un digrafo acíclico $C = (V, E)$ se dice circuito Booleano si:

- Cada *puertas lógica* (i.e. los nodo $v \in V$ de grado de entrada $i_v > 0$) está etiquetada por una función Booleana de aridad i_v.
- Existe una única puerta lógica con grado de salida igual a 0 y que se denomina *salida*.
- Las *entradas* (i.e. los nodos $v \in V$ con grado de entrada igual a 0) están etiquetados por 0, 1, o por variables Booleanas que se denominan *variables de entrada*.

Salvo que se diga lo contrario, las puertas lógicas de un circuito Booleano estarán etiquetados por las funciones OR (denotada $\vee$), o AND (denotada $\wedge$), o NOT (denotada $\neg$) en cuyo caso la puerta lógica debe necesariamente tener grado de entrada 1.

Definiciones adicionales relativas a circuitos Booleanos

Un circuito Booleano C de n entradas tiene asociada de forma natural una función Booleana de $\{0,1\}^n \to \{0,1\}$ que se denotará (abusando notación) por C.

El tamaño de un circuito Booleano C es el número de puertas lógicas de C, y se denota por $|C|$.

Se denomina *profundidad* del circuito Booleano C al mayor largo de un dicamino que va desde una entrada hasta la salida de C.

Evaluación de circuitos Booleanos

Se define,

$$\mathrm{CircEval} = \{\langle C, \omega\rangle : C \text{ es circuito Booleano y } C(\omega) = 1\}.$$

Theorem

$\mathrm{CircEval}$ *está en* P.

De hecho, más adelante veremos que $\mathrm{CircEval}$ es de los problemas de decisión más difíciles en P.

Algorithm 2: Evaluación de Circuitos Booleanos

input : $\omega \in \{0,1\}^n$ y $C = (V, E)$ circuito Booleano con entradas $v_1, \ldots, v_n$ y salida v_{sal}.

output: $C(\omega)$.

```
/* Inicialización                                                    */
```

for $v \in V$ **do**
  $prov_v \leftarrow 0$;

for $i = 1, \ldots, |V|$ **do**
  for $uv \in E$ **do**
    $prov_v \leftarrow \max\{prof_v, prof_u + 1\}$;

for $i = 1, \ldots, n$ **do**
  $val_{v_i} \leftarrow \omega_i$;

for $i = 1, \ldots, prof_{v_{sal}}$ **do**
  for $v \in V$ **do**
    if $prof_v = i$ **then**
      /* f_v etiqueta de v y $u_1 v, \ldots, u_{i_v} v \in E$ */
      $val_v \leftarrow f_v(val_{u_1}, \ldots, val_{u_{i_v}})$;

return$(val_{v_{sal}})$

Idea de la demostración del Teorema de Cook-Levin

Theorem

Sea $t : \mathbb{N} \to \mathbb{N}$ *tal que* $t(n) \geq n$. *Si* $L \in \mathrm{DTIEMPO}(t(n))$, *entonces existe una familia de circuitos Booleanos* $(C_n)_{n\in\mathbb{N}}$, $|C_n| = O(t^2(n))$, *tal que para todo* $\omega \in \Sigma_L^*$ *se tiene que*

$$\omega \in L \iff C_n(\omega) = 1, \text{ donde } n = |\omega| .$$

Más aún, la familia $(C_n)_{n\in\mathbb{N}}$ *es constructible en tiempo* $O(t^2(n))$ *(es decir, existe una máquina de Turing que en la entrada* n *en unario, entrega como salida una codificación de* C_n *en tiempo* $O(t^2(n))$*.*

Corollary

$L \in \mathrm{P}$ *si y solo si existe una familia de circuitos* $(C_n)_{n\in\mathbb{N}}$ *constructible en tiempo polinomial tal que*

$$\omega \in L \iff C_n(\omega) = 1, \text{ donde } n = |\omega| .$$

Idea de la demostración del Teorema de Cook-Levin (cont.)

Corollary

$L \in \mathrm{NP}$ si y solo si existe una familia de circuitos $(C_n(\cdot,\cdot))_{n\in\mathbb{N}}$ constructible en tiempo polinomial y un polinomio p tales que

$$\omega \in L \iff \exists \pi_\omega \in \{0,1\}^{p(n)} \text{ tal que } C_n(\omega, \pi_\omega) = 1, \text{ donde } n = |\omega| .$$

Se dice que un circuito Booleano C en n entradas se puede *satisfacer*, si existe $x \in \{0,1\}^n$ tal que $C(x) = 1$. Se define,

$$\mathrm{CircSat} = \{\langle C\rangle : C \text{ es circuito Booleano y se puede satisfacer}\} .$$

Corollary

$\mathrm{CircSat}$ *es* NP*-completo.*

Corollary

$\mathrm{CircSat} \leq_P \mathrm{SAT}$ *y por lo tanto* SAT *es* NP*-completo.*