

Motivación: la complejidad de algunos problemas

IIC3810

El problema de factorización de números naturales

Factorizar un número natural n consiste en encontrar un divisor de n

- ▶ Este divisor debe ser no trivial, vale decir, mayor que 1 y menor que n

Es importante considerar que el problema de verificar si un número es primo se puede resolver en tiempo polinomial.

- ▶ Al factorizar un número n primero verificamos si es primo. Si n es primo indicamos que no hay divisores no triviales, en caso contrario debemos entregar uno de estos divisores

El problema de factorización de números enteros

Como problema de decisión el problema de factorización mostrado en la transparencia anterior es modelado de la siguiente forma:

$$\text{FACT} = \{(n, k) \mid n \text{ y } k \text{ son números naturales y} \\ \text{existe } \ell \text{ tal que } 1 < \ell \leq k \text{ y } \ell \text{ divide a } n\}$$

¿Por qué FACT se considera equivalente al problema de factorización?

- ¿Es cierto que si uno puede resolver FACT en tiempo polinomial también puede resolver el problema de factorización en tiempo polinomial? ¿Es cierta la dirección opuesta?

La complejidad exacta de FACT

Ejercicio

Demuestre que FACT está en NP.

Ejercicio

Demuestre que FACT está en co-NP.

¿Puede ser FACT un problema NP-completo?

- ▶ ¿Qué consecuencias tendría esto?

El problema de isomorfismo de grafos

Dados: grafos $G_1 = (N_1, A_1)$ y $G_2 = (N_2, A_2)$

Una función $f : N_1 \rightarrow N_2$ es un isomorfismo de G_1 en G_2 si

1. f es una biyección
2. Para cada par $(a, b) \in N_1 \times N_1$, se tiene que $(a, b) \in A_1$ si y sólo si $(f(a), f(b)) \in A_2$

El problema de isomorfismo de grafos

El problema de isomorfismo de grafos se define de la siguiente forma:

$$\text{GRAPH-ISO} = \{(G_1, G_2) \mid G_1 \text{ y } G_2 \text{ son grafos y} \\ \text{existe un isomorfismo de } G_1 \text{ en } G_2\}$$

Ejercicio

Demuestre que GRAPH-ISO está en NP.

¿Qué más sabemos sobre la complejidad de este problema?

- ▶ ¿Es GRAPH-ISO un problema NP-completo?
- ▶ ¿Está GRAPH-ISO en co-NP?

El problema de coloración de un grafo

Una grafo $G = (N, A)$ se dice k -coloreable si existe una función $f : N \rightarrow \{1, \dots, k\}$ tal que:

- ▶ Si $(a, b) \in A$, entonces $f(a) \neq f(b)$

El número cromático de un grafo G , denotado como $\chi(G)$, se define como el menor número k tal que G es k -coloreable.

- ▶ Vale decir, G es k -coloreable y no $(k - 1)$ -coloreable

El problema de coloración de un grafo

Considere los siguientes problemas relacionados con el cálculo del número cromático de un grafo:

$$\text{CROM}^{\leq} = \{(G, k) \mid G \text{ es un grafo,} \\ k \text{ es un número natural y } \chi(G) \leq k\}$$

$$\text{CROM}^{\geq} = \{(G, k) \mid G \text{ es un grafo,} \\ k \text{ es un número natural y } \chi(G) \geq k\}$$

Ejercicio

Demuestre que $\text{CROM}^{\leq}$ es NP-completo y $\text{CROM}^{\geq}$ es co-NP-completo.

Cálculo del número cromático

Considere ahora el problema de decisión asociado al cálculo exacto del número cromático de un grafo:

$$\text{CROM} = \{(G, k) \mid G \text{ es un grafo,} \\ k \text{ es un número natural y } \chi(G) = k\}$$

¿Cuál es la complejidad de CROM?

- ¿Está este problema en NP? ¿Está en co-NP? ¿Es completo para alguna de estas clases?

Podemos caracterizar CROM en términos de dos lenguajes en NP.

Ejercicio

Encuentre lenguajes L_1 y L_2 en NP tales que $\text{CROM} = L_1 \setminus L_2$

CROM no está “lejos” de NP

- ▶ De hecho, si $\text{PTIME} = \text{NP}$, entonces CROM está en PTIME
- ▶ CROM puede ser solucionado resolviendo dos problemas en NP. Vemos a estos problemas como sub-rutinas u oráculos

Evaluación de expresiones aritméticas

Una expresión aritmética es definida recursivamente por las siguientes reglas:

- ▶ Si $n \in \mathbb{N}$, entonces n es una expresión aritmética
- ▶ Si e_1 y e_2 son expresiones aritméticas, entonces $(e_1 + e_2)$ y $(e_1 \cup e_2)$ son expresiones aritméticas

Ejemplo

5, $(5 + 7)$ y $((5 + 7) \cup (4 \cup (6 + 11)))$ son expresiones aritméticas.

El lenguaje $L(e)$ definido por una expresión aritmética e es definido recursivamente por las siguiente reglas:

- ▶ Si $e = n$ con $n \in \mathbb{N}$, entonces $L(e) = \{n\}$
- ▶ Si $e = (e_1 + e_2)$, entonces $L(e) = \{n + m \mid n \in L(e_1) \text{ y } m \in L(e_2)\}$
- ▶ Si $e = (e_1 \cup e_2)$, entonces $L(e) = L(e_1) \cup L(e_2)$

Ejemplo

El lenguaje definido por algunas expresiones aritméticas:

- ▶ $L(5) = \{5\}$ y $L(7) = \{7\}$
- ▶ Si $e = (5 + 7)$, entonces $L(e) = \{n + m \mid n \in L(5) \text{ y } m \in L(7)\} = \{12\}$
- ▶ Si $e = (6 + 11)$, entonces $L(e) = \{17\}$
- ▶ Si $e = (4 \cup (6 + 11))$, entonces $L(e) = L(4) \cup L((6 + 11)) = \{4\} \cup \{17\} = \{4, 17\}$
- ▶ Si $e = (1 \cup 3) + (7 \cup 13)$, entonces $L(e) = \{8, 10, 14, 16\}$

Evaluación de expresiones aritméticas

El problema de evaluación de expresiones aritméticas es definido de la siguiente forma:

$$\text{EVAL} = \{(e, k) \mid e \text{ es una expresión aritmética,} \\ k \text{ es un número natural y } k \in L(e)\}$$

Ejercicio

Demuestre que EVAL es un problema NP-completo.

Equivalencia de expresiones aritméticas

Considere el siguiente problema sobre expresiones aritméticas:

$$\text{EQUIV} = \{(e_1, e_2) \mid e_1 \text{ y } e_2 \text{ son expresiones aritméticas tales que } L(e_1) = L(e_2)\}$$

¿Cuál es la complejidad de EQUIV?

- ▶ ¿Está en NP o co-NP?
- ▶ ¿Puede encontrar dos lenguajes L_1 y L_2 en NP tales que $\text{EQUIV} = L_1 \setminus L_2$?
- ▶ ¿Puede al menos demostrar que si $\text{PTIME} = \text{NP}$, entonces EQUIV está en PTIME?
 - ▶ ¿Tiene sentido la idea de sub-rutina (u oráculo) en este caso?