

Una primera noción de reducción para funciones

Definición

Dadas funciones $f, g : \Sigma^* \rightarrow \mathbb{N}$, decimos que f se puede reducir a g de forma parsimoniosa, denotado como $f \leq_{par}^p g$, si existe una función $h : \Sigma^* \rightarrow \Sigma^*$ tal que h puede ser calculada en tiempo polinomial y para todo $w \in \Sigma^*$:

$$f(w) = g(h(w))$$

Ejercicio

Demuestre que si $f_1 \leq_{par}^p f_2$ y $f_2 \leq_{par}^p f_3$, entonces $f_1 \leq_{par}^p f_3$

Una noción de reducción más general para funciones

Al igual que para las máquinas de Turing con oráculos para problemas de decisión, podemos definir las máquinas de Turing con oráculos para funciones

- ▶ ¿Cómo debería funcionar la cinta de consulta en este caso?

Dada una función $f : \Sigma^* \rightarrow \mathbb{N}$, denotamos como M^f a una MT que tiene a f como oráculo

- ▶ M^f también puede ser una MTC

Una noción de reducción más general para funciones

Dadas funciones $f, g : \Sigma^* \rightarrow \mathbb{N}$, decimos que $f \in FP^g$ si existe una MTC M^g que funciona en tiempo polinomial y calcula f

Definición

Dadas funciones $f, g : \Sigma^ \rightarrow \mathbb{N}$, decimos que f se puede reducir a g en tiempo polinomial, denotado como $f \leq_T^p g$, si $f \in FP^g$*

Ejercicio

Demuestre que si $f_1 \leq_T^p f_2$ y $f_2 \leq_T^p f_3$, entonces $f_1 \leq_T^p f_3$

Una noción de completitud para $\#P$

Definición

Una función $f : \Sigma^ \rightarrow \mathbb{N}$ es $\#P$ -hard si para cada $g \in \#P$ se tiene que $g \leq_T^P f$. Sí adicionalmente $f \in \#P$, entonces f se dice $\#P$ -completo.*

Si $\leq_T^P$ es reemplazado por $\leq_{par}^P$ en la definición anterior, entonces decimos que f es $\#P$ -hard o $\#P$ -completo **bajo reducciones parsimoniosas**

- Si f es $\#P$ -hard bajo reducciones parsimoniosas entonces también es $\#P$ -hard, puesto que $f \leq_{par}^P g$ implica $f \leq_T^P g$

Un primer problema $\#P$ -completo

Teorema

$\#SAT$ es $\#P$ -completo bajo reducciones parsimoniosas.

Ejercicio

Demuestre el teorema.

Un segundo problema #P-completo

Sea $\#CNF-SAT$ una función que, dada una formula proposicional φ en CNF, retorna el número de valuaciones que satisfacen φ

Teorema

$\#CNF-SAT$ es #P-completo bajo reducciones parsimoniosas.

Ejercicio

Demuestre el teorema.

Un tercer problema $\#P$ -completo

Sea $\#DNF\text{-SAT}$ una función que, dada una fórmula proposicional φ en DNF, retorna el número de valuaciones que satisface φ

Teorema

$\#DNF\text{-SAT}$ es $\#P$ -completo

Demostración: vamos a demostrar que $\#CNF\text{-SAT} \leq_T^P \#DNF\text{-SAT}$

Sea φ una fórmula proposicional en CNF, y sea ψ una fórmula proposicional en DNF que es equivalente a $\neg\varphi$

- ψ puede ser construida en tiempo polinomial a partir de φ

La demostración del teorema

Definimos $nv(\cdot)$ como una función que retorna el número de variables de una fórmula proposicional φ

- ▶ Por ejemplo, $nv((p \vee q) \wedge \neg r) = 3$ y $nv((r \vee q) \wedge \neg r) = 2$

La función $nv(\cdot)$ es computable en tiempo polinomial

Tenemos que:

$$\#CNF-SAT(\varphi) = 2^{nv(\psi)} - \#DNF-SAT(\psi)$$

Concluimos entonces que $\#CNF-SAT \in FP^{\#DNF-SAT}$

- ▶ Por lo tanto, $\#CNF-SAT \leq_T^p \#DNF-SAT$



Sobre la completitud de #DNF-SAT

¿Se puede demostrar que #DNF-SAT es #P-completo bajo reducciones parsimoniosas?

Para responder esta pregunta, defina para cada función $f : \Sigma^* \rightarrow \mathbb{N}$ el siguiente problema de decisión:

$$L_f = \{w \in \Sigma^* \mid f(w) > 0\}$$

Ejemplo

Tenemos que $L_{\#SAT} = SAT$

Sobre la completitud de $\#DNF\text{-SAT}$

Proposición

Si $f \leq_{par}^p g$ y $L_g \in PTIME$, entonces $L_f \in PTIME$

Ejercicio

Demuestre la proposición.

Concluimos que $\#DNF\text{-SAT}$ no puede ser $\#P$ -completo bajo reducciones parsimoniosas, a menos que $PTIME = NP$

- Puesto que si $\#SAT \leq_{par}^p \#DNF\text{-SAT}$, entonces $SAT \in PTIME$