Una primera noción de reducción para funciones

Definición

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

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

Una primera noción de reducción para funciones

Definición

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

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

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?

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

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

Una noción de reducción más general para funciones

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

Definición

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

Una noción de reducción más general para funciones

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

Definición

Dadas funciones $f, g : \Sigma^* \to \mathbb{N}$, 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 $f : \Sigma^* \to \mathbb{N}$ 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.

Una noción de completitud para #P

Definición

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

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

Una noción de completitud para $\#$P

Definición

Una función $f : \Sigma^* \to \mathbb{N}$ 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.

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

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.

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-SAT una función que, dada una formula proposicional φ en DNF, retorna el número de valuaciones que satisface φ

Un tercer problema #P-completo

Sea #DNF-SAT una función que, dada una formula proposicional φ en DNF, retorna el número de valuaciones que satisface φ

Teorema

#DNF-SAT es #P-completo

Un tercer problema #P-completo

Sea #DNF-SAT una función que, dada una formula proposicional φ en DNF, retorna el número de valuaciones que satisface φ

Teorema

#DNF-SAT es #P-completo

Demostración: vamos a demostrar que #CNF-SAT $\leq_T^p$ #DNF-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 $\text{nv}(\cdot)$ como una función que retorna el número de variables de una fórmula proposicional φ

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

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

La demostración del teorema

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

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

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

Tenemos que:

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

La demostración del teorema

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

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

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

Tenemos que:

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

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

- Por lo tanto, $\#\text{CNF-SAT} \leq_T^p \#\text{DNF-SAT}$ $\square$

Sobre la completitud de #DNF-SAT

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

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 $f : \Sigma^* \to \mathbb{N}$ el siguiente problema de decisión:

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

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 $f : \Sigma^* \to \mathbb{N}$ el siguiente problema de decisión:

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

Ejemplo

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

Sobre la completitud de #DNF-SAT

Proposición

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

Sobre la completitud de #DNF-SAT

Proposición

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

Ejercicio

Demuestre la proposición.

Sobre la completitud de #DNF-SAT

Proposición

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

Ejercicio

Demuestre la proposición.

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

- Puesto que si #SAT $\leq^p_{par}$ #DNF-SAT, entonces SAT $\in$ PTIME