

La clase de complejidad de funciones $\#P$

IIC3810

Marcelo Arenas y Luis Alberto Croquevielle

Calculando una función

Dado un alfabeto Σ , consideramos funciones $f : \Sigma^* \rightarrow \mathbb{N}$

Representamos cada número natural como un string sobre el alfabeto $\{0, \dots, 9\}$

- ▶ Si una MTC calcula una función $f : \Sigma^* \rightarrow \mathbb{N}$, entonces para cada $w \in \Sigma^*$ suponemos que $f(w)$ es retornado en la cinta de salida como un string sobre el alfabeto $\{0, \dots, 9\}$

Las funciones computables en tiempo polinomial

Definición

Una función $f : \Sigma^ \rightarrow \mathbb{N}$ está en la clase **FP** si y sólo si existe una MTC que funciona en tiempo polinomial y calcula f*

Las funciones computables en tiempo polinomial

Definición

Una función $f : \Sigma^ \rightarrow \mathbb{N}$ está en la clase **FP** si y sólo si existe una MTC que funciona en tiempo polinomial y calcula f*

¿Qué funciones son difíciles de computar?

Las funciones computables en tiempo polinomial

Definición

Una función $f : \Sigma^ \rightarrow \mathbb{N}$ está en la clase **FP** si y sólo si existe una MTC que funciona en tiempo polinomial y calcula f*

¿Qué funciones son difíciles de computar?

- Consideramos funciones tales que si son computables en tiempo polinomial, entonces nos dan algoritmos polinomiales para resolver problemas NP-completos

La clase de funciones $\#P$

Sea M una MT no determinista con alfabeto de entrada Σ

La clase de funciones $\#P$

Sea M una MT no determinista con alfabeto de entrada Σ

Supuesto

Si decimos que M funciona en tiempo polinomial, entonces suponemos que existe un polinomio $p(n)$ tal que para cada $w \in \Sigma^*$ y cada ejecución de M con entrada w , el número de pasos en la ejecución es menor o igual a $p(|w|)$

La clase de funciones $\#P$

Sea M una MT no determinista con alfabeto de entrada Σ

Supuesto

Si decimos que M funciona en tiempo polinomial, entonces suponemos que existe un polinomio $p(n)$ tal que para cada $w \in \Sigma^*$ y cada ejecución de M con entrada w , el número de pasos en la ejecución es menor o igual a $p(|w|)$

- ¿Por qué podemos realizar este supuesto sin pérdida de generalidad?

La clase de funciones $\#P$

Sea M una MT no determinista con alfabeto de entrada Σ

Supuesto

Si decimos que M funciona en tiempo polinomial, entonces suponemos que existe un polinomio $p(n)$ tal que para cada $w \in \Sigma^*$ y cada ejecución de M con entrada w , el número de pasos en la ejecución es menor o igual a $p(|w|)$

- ¿Por qué podemos realizar este supuesto sin pérdida de generalidad?

Dado $w \in \Sigma^*$, definimos $\text{accept}_M(w)$ como el número de ejecuciones de M con entrada w que se detienen en un estado final.

- En particular, $w \in L(M)$ si y sólo si $\text{accept}_M(w) > 0$

La clase de funciones $\#P$

Definición

Una función $f : \Sigma^* \rightarrow \mathbb{N}$ está en $\#P$ si y sólo si existe una MT no determinista M con alfabeto de entrada Σ , que funciona en tiempo polinomial y tal que para cada $w \in \Sigma^*$: $f(w) = \text{accept}_M(w)$

La clase de funciones $\#P$

Definición

Una función $f : \Sigma^* \rightarrow \mathbb{N}$ está en $\#P$ si y sólo si existe una MT no determinista M con alfabeto de entrada Σ , que funciona en tiempo polinomial y tal que para cada $w \in \Sigma^*$: $f(w) = \text{accept}_M(w)$

Ejercicio

Sea $\#SAT$ una función tal que, dada una fórmula proposicional φ , retorna el número de valuaciones que satisfacen φ . Demuestre que $\#SAT \in \#P$

Para pensar ...

Ejercicios

1. Demuestre que si $f, g \in \#P$, entonces $f + 1 \in \#P$ y $f + g \in \#P$
2. Demuestre que si $f, g \in \#P$, entonces $f \cdot g \in \#P$
3. Demuestre que $FP \subseteq \#P$

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 $w \in \Sigma^*$:

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

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?

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

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 $f \in \text{FP}^g$ si existe una MTC M^g que funciona en tiempo polinomial y calcula f

Una noción de reducción más general para funciones

Dadas funciones $f, g : \Sigma^* \rightarrow \mathbb{N}$, 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 $f \in FP^g$*

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 $g \leq_T^P f$. Si adicionalmente $f \in \#P$, entonces f se dice $\#P$ -completo.*

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**

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 $g \leq_T^P f$. Si 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 $\# \text{CNF-SAT}$ una función que, dada una formula proposicional φ en CNF, retorna el número de valuaciones que satisfacen φ

Un segundo problema $\#P$ -completo

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

Teorema

$\#CNF\text{-SAT}$ 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 $\#DNF\text{-SAT}$ 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

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:

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

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?

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:

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

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-SAT

Proposición

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

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.

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.

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

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

¿Cómo podemos calcular una función $\#P$ -completa?

Sea $f : \Sigma^* \rightarrow \mathbb{N}$ una función en $\#P$

Si la función f es $\#P$ -completa entonces no esperamos tener un algoritmo polinomial para calcularla.

¿Cómo podemos calcular una función #P-completa?

Sea $f : \Sigma^* \rightarrow \mathbb{N}$ una función en #P

Si la función f es #P-completa entonces no esperamos tener un algoritmo polinomial para calcularla.

Pero el hecho de que f sea #P-completa no implica que el valor de f no pueda ser aproximado de manera eficiente.

¿Cómo podemos calcular una función $\#P$ -completa?

Sea $f : \Sigma^* \rightarrow \mathbb{N}$ una función en $\#P$

Si la función f es $\#P$ -completa entonces no esperamos tener un algoritmo polinomial para calcularla.

Pero el hecho de que f sea $\#P$ -completa no implica que el valor de f no pueda ser aproximado de manera eficiente.

- Vamos a estudiar una noción de aproximación aleatorizada para funciones en $\#P$

Aproximación aleatorizada de funciones

Sea $f : \Sigma^* \rightarrow \mathbb{N}$

Aproximación aleatorizada de funciones

Sea $f : \Sigma^* \rightarrow \mathbb{N}$

Definición

Un algoritmo aleatorizado $\mathcal{A} : \Sigma^* \times (0, 1) \rightarrow \mathbb{N}$ es un **fully polynomial randomized approximation scheme (FPRAS)** para f si existe un polinomio $p(x, y)$ tal que para cada $w \in \Sigma^*$ y $\varepsilon \in (0, 1)$:

1. El número de pasos ejecutados por $\mathcal{A}(w, \varepsilon)$ es menor o igual a $p(|w|, \frac{1}{\varepsilon})$
2. $\Pr(|\mathcal{A}(w, \varepsilon) - f(w)| \leq \varepsilon \cdot f(w)) \geq \frac{3}{4}$

Un enfoque general para construir un FPRAS

Dada una función $f : \Sigma^* \rightarrow \mathbb{N}$, $w \in \Sigma^*$ y $\varepsilon \in (0, 1)$, definimos una variable aleatoria X tal que $\mathbf{E}[X] = f(w)$

- Para tener una estimación de $f(w)$ nos basta muestrear la variable aleatoria X

Un enfoque general para construir un FPRAS

Dada una función $f : \Sigma^* \rightarrow \mathbb{N}$, $w \in \Sigma^*$ y $\varepsilon \in (0, 1)$, definimos una variable aleatoria X tal que $\mathbf{E}[X] = f(w)$

- Para tener una estimación de $f(w)$ nos basta muestrear la variable aleatoria X

Debe ser posible muestrear X en tiempo polinomial en $|w|$ y $\frac{1}{\varepsilon}$, y además debemos tener que $\mathbf{Pr}(|X - f(w)| \leq \varepsilon \cdot f(w)) \geq \frac{3}{4}$

- Debemos entonces acotar inferiormente $\mathbf{Pr}(|X - \mathbf{E}[X]| \leq \varepsilon \cdot \mathbf{E}[X])$

Un enfoque general para construir un FPRAS

Dada una función $f : \Sigma^* \rightarrow \mathbb{N}$, $w \in \Sigma^*$ y $\varepsilon \in (0, 1)$, definimos una variable aleatoria X tal que $\mathbf{E}[X] = f(w)$

- Para tener una estimación de $f(w)$ nos basta muestrear la variable aleatoria X

Debe ser posible muestrear X en tiempo polinomial en $|w|$ y $\frac{1}{\varepsilon}$, y además debemos tener que $\mathbf{Pr}(|X - f(w)| \leq \varepsilon \cdot f(w)) \geq \frac{3}{4}$

- Debemos entonces acotar inferiormente $\mathbf{Pr}(|X - \mathbf{E}[X]| \leq \varepsilon \cdot \mathbf{E}[X])$

Vamos a estudiar entonces algunas desigualdades útiles para obtener una cota inferior para $\mathbf{Pr}(|X - \mathbf{E}[X]| \leq \varepsilon \cdot \mathbf{E}[X])$

La desigualdad de Markov

Teorema

Sea X una variable aleatoria no negativa. Para cada $a \in \mathbb{R}^+$ se tiene que:

$$\Pr(X \geq a) \leq \frac{\mathbf{E}[X]}{a}$$

Una demostración de la desigualdad de Markov

Suponemos que el recorrido de X es un conjunto finito $\Omega \subseteq \mathbb{R}_0^+$:

$$\begin{aligned}\mathbf{E}(X) &= \sum_{r \in \Omega} r \cdot \mathbf{Pr}(X = r) \\&= \left(\sum_{r \in \Omega : r < a} r \cdot \mathbf{Pr}(X = r) \right) + \left(\sum_{s \in \Omega : s \geq a} s \cdot \mathbf{Pr}(X = s) \right) \\&\geq \sum_{s \in \Omega : s \geq a} s \cdot \mathbf{Pr}(X = s) \\&\geq \sum_{s \in \Omega : s \geq a} a \cdot \mathbf{Pr}(X = s) \\&= a \cdot \left(\sum_{s \in \Omega : s \geq a} \mathbf{Pr}(X = s) \right) \\&= a \cdot \mathbf{Pr}(X \geq a)\end{aligned}$$

Una demostración de la desigualdad de Markov

Suponemos que el recorrido de X es un conjunto finito $\Omega \subseteq \mathbb{R}_0^+$:

$$\begin{aligned}\mathbf{E}(X) &= \sum_{r \in \Omega} r \cdot \mathbf{Pr}(X = r) \\ &= \left(\sum_{r \in \Omega : r < a} r \cdot \mathbf{Pr}(X = r) \right) + \left(\sum_{s \in \Omega : s \geq a} s \cdot \mathbf{Pr}(X = s) \right) \\ &\geq \sum_{s \in \Omega : s \geq a} s \cdot \mathbf{Pr}(X = s) \\ &\geq \sum_{s \in \Omega : s \geq a} a \cdot \mathbf{Pr}(X = s) \\ &= a \cdot \left(\sum_{s \in \Omega : s \geq a} \mathbf{Pr}(X = s) \right) \\ &= a \cdot \mathbf{Pr}(X \geq a)\end{aligned}$$

Concluimos que $\mathbf{Pr}(X \geq a) \leq \frac{\mathbf{E}(X)}{a}$



La desigualdad de Chebyshev

Teorema

$$\Pr(|X - \mathbf{E}[X]| \geq a) \leq \frac{\mathbf{Var}[X]}{a^2}$$

La desigualdad de Chebyshev

Teorema

$$\Pr(|X - \mathbf{E}[X]| \geq a) \leq \frac{\mathbf{Var}[X]}{a^2}$$

Demostración. Utilizando la desigualdad de Markov obtenemos:

$$\begin{aligned} \Pr(|X - \mathbf{E}[X]| \geq a) &= \Pr((X - \mathbf{E}[X])^2 \geq a^2) \\ &\leq \frac{\mathbf{E}[(X - \mathbf{E}[X])^2]}{a^2} \\ &= \frac{\mathbf{Var}[X]}{a^2} \end{aligned}$$



Utilizando la desigualdad de Chebyshev

A partir de la desigualdad de Chebyshev obtenemos:

$$\Pr(|X - \mathbf{E}[X]| \leq \varepsilon \cdot \mathbf{E}[X]) \geq 1 - \frac{\mathbf{Var}[X]}{\varepsilon^2 \cdot \mathbf{E}[X]^2}$$

Utilizando la desigualdad de Chebyshev

A partir de la desigualdad de Chebyshev obtenemos:

$$\Pr(|X - \mathbf{E}[X]| \leq \varepsilon \cdot \mathbf{E}[X]) \geq 1 - \frac{\mathbf{Var}[X]}{\varepsilon^2 \cdot \mathbf{E}[X]^2}$$

Es posible mejorar esta cota inferior considerando el promedio de $k \geq 2$ muestras de X obtenidas de manera independiente.

- ▶ Así reducimos el impacto de las muestras alejadas de $\mathbf{E}[X]$

Utilizando la desigualdad de Chebyshev

Por la desigualdad de Chebyshev:

$$\Pr\left(\left|\frac{1}{k} \cdot \sum_{i=1}^k X_i - \mathbf{E}\left[\frac{1}{k} \cdot \sum_{i=1}^k X_i\right]\right| \geq \varepsilon \cdot \mathbf{E}\left[\frac{1}{k} \cdot \sum_{i=1}^k X_i\right]\right) \leq \frac{\mathbf{Var}\left[\frac{1}{k} \cdot \sum_{i=1}^k X_i\right]}{\varepsilon^2 \cdot \mathbf{E}\left[\frac{1}{k} \cdot \sum_{i=1}^k X_i\right]^2}$$

Además, tenemos que:

$$\begin{aligned}\mathbf{E}\left[\frac{1}{k} \cdot \sum_{i=1}^k X_i\right] &= \frac{1}{k} \cdot \mathbf{E}\left[\sum_{i=1}^k X_i\right] \\ &= \frac{1}{k} \cdot \sum_{i=1}^k \mathbf{E}[X_i] \\ &= \frac{1}{k} \cdot \sum_{i=1}^k \mathbf{E}[X] \\ &= \mathbf{E}[X]\end{aligned}$$

Utilizando la desigualdad de Chebyshev

$$\begin{aligned}\mathbf{Var}\left[\frac{1}{k} \cdot \sum_{i=1}^k X_i\right] &= \frac{1}{k^2} \cdot \mathbf{Var}\left[\sum_{i=1}^k X_i\right] \\ &= \frac{1}{k^2} \cdot \sum_{i=1}^k \mathbf{Var}[X_i] \\ &= \frac{1}{k^2} \cdot \sum_{i=1}^k \mathbf{Var}[X] \\ &= \frac{1}{k} \cdot \mathbf{Var}[X]\end{aligned}$$

De lo cual concluimos que:

$$\mathbf{Pr}\left(\left|\frac{1}{k} \cdot \sum_{i=1}^k X_i - \mathbf{E}[X]\right| \geq \varepsilon \cdot \mathbf{E}[X]\right) \leq \frac{\mathbf{Var}[X]}{k \cdot \varepsilon^2 \cdot \mathbf{E}[X]^2}$$

Utilizando la desigualdad de Chebyshev

Realizando k muestras independientes de la variable aleatoria X obtenemos entonces:

$$\Pr\left(\left|\frac{1}{k} \cdot \sum_{i=1}^k X_i - \mathbf{E}[X]\right| \leq \varepsilon \cdot \mathbf{E}[X]\right) \geq 1 - \frac{\mathbf{Var}[X]}{k \cdot \varepsilon^2 \cdot \mathbf{E}[X]^2}$$

Utilizando la desigualdad de Chebyshev

Realizando k muestras independientes de la variable aleatoria X obtenemos entonces:

$$\Pr\left(\left|\frac{1}{k} \cdot \sum_{i=1}^k X_i - \mathbf{E}[X]\right| \leq \varepsilon \cdot \mathbf{E}[X]\right) \geq 1 - \frac{\mathbf{Var}[X]}{k \cdot \varepsilon^2 \cdot \mathbf{E}[X]^2}$$

Podemos entonces ajustar el valor de k para obtener un FPRAS.

- El valor de $\mathbf{Var}[X]$ debe ser polinomial en el tamaño de la entrada para obtener un FPRAS.

Construyendo un FPRAS para $\#DNF\text{-SAT}$

Sea $\varphi = D_1 \vee \cdots \vee D_m$ una formula proposicional en DNF

- ▶ Cada D_i es una conjunción de literales
- ▶ φ menciona n variables proposicionales, vale decir, $nv(\varphi) = n$

Suponemos que cada D_i no tiene literales repetidos ni complementarios

- ▶ ¿Por qué podemos suponer esto?
- ▶ Sea ℓ_i el número de literales mencionados en D_i

Un FPRAS para $\#DNF\text{-SAT}$: primer intento

Suponga que las asignaciones de valores de verdad σ para φ son escogidas con distribución uniforme:

$$\mathbf{Pr}(\sigma) = \frac{1}{2^n}$$

Y considere la siguiente variable aleatoria:

$$X(\sigma) = \begin{cases} 2^n & \sigma(\varphi) = 1 \\ 0 & \sigma(\varphi) = 0 \end{cases}$$

Un FPRAS para #DNF-SAT: primer intento

Tenemos que:

$$\begin{aligned}\mathbf{E}[X] &= \sum_{\sigma} X(\sigma) \cdot \mathbf{Pr}(\sigma) \\ &= \sum_{\sigma : \sigma(\varphi)=1} 2^n \cdot \frac{1}{2^n} \\ &= \sum_{\sigma : \sigma(\varphi)=1} 1 \\ &= \#\text{DNF-SAT}(\varphi)\end{aligned}$$

Un FPRAS para #DNF-SAT: primer intento

Tenemos que:

$$\begin{aligned}\mathbf{E}[X] &= \sum_{\sigma} X(\sigma) \cdot \mathbf{Pr}(\sigma) \\ &= \sum_{\sigma : \sigma(\varphi)=1} 2^n \cdot \frac{1}{2^n} \\ &= \sum_{\sigma : \sigma(\varphi)=1} 1 \\ &= \#\text{DNF-SAT}(\varphi)\end{aligned}$$

Tenemos entonces que $\mathbf{E}[X] = \#\text{DNF-SAT}(\varphi)$, el primer requisito para la variable aleatoria que queremos construir

- ▶ Además se puede muestrear X en tiempo polinomial en n
 - ▶ ¿Cómo se hace esto?

Un FPRAS para $\#DNF\text{-SAT}$: primer intento

Nos falta calcular $\mathbf{Var}[X]$ para saber si podemos obtener un FPRAS.

Un FPRAS para $\#$ DNF-SAT: primer intento

Nos falta calcular $\mathbf{Var}[X]$ para saber si podemos obtener un FPRAS.

Tenemos que:

$$\begin{aligned}\mathbf{E}[X^2] &= \sum_{\sigma} X^2(\sigma) \cdot \mathbf{Pr}(\sigma) \\ &= \sum_{\sigma : \sigma(\varphi)=1} 2^{2 \cdot n} \cdot \frac{1}{2^n} \\ &= \sum_{\sigma : \sigma(\varphi)=1} 2^n \\ &= 2^n \cdot \# \text{DNF-SAT}(\varphi)\end{aligned}$$

Un FPRAS para $\# \text{DNF-SAT}$: primer intento

Nos falta calcular $\mathbf{Var}[X]$ para saber si podemos obtener un FPRAS.

Tenemos que:

$$\begin{aligned}\mathbf{E}[X^2] &= \sum_{\sigma} X^2(\sigma) \cdot \mathbf{Pr}(\sigma) \\ &= \sum_{\sigma : \sigma(\varphi)=1} 2^{2 \cdot n} \cdot \frac{1}{2^n} \\ &= \sum_{\sigma : \sigma(\varphi)=1} 2^n \\ &= 2^n \cdot \# \text{DNF-SAT}(\varphi)\end{aligned}$$

Por lo tanto:

$$\begin{aligned}\mathbf{Var}[X] &= \mathbf{E}[X^2] - \mathbf{E}[X]^2 \\ &= 2^n \cdot \# \text{DNF-SAT}(\varphi) - \# \text{DNF-SAT}(\varphi)^2 \\ &= (2^n - \# \text{DNF-SAT}(\varphi)) \cdot \# \text{DNF-SAT}(\varphi)\end{aligned}$$

Un FPRAS para #DNF-SAT: primer intento

Realizando k muestras independientes de X obtenemos entonces:

$$\begin{aligned}\Pr\left(\left|\frac{1}{k} \cdot \sum_{i=1}^k X_i - \mathbf{E}[X]\right| \geq \varepsilon \cdot \mathbf{E}[X]\right) &\leq \frac{\mathbf{Var}[X]}{k \cdot \varepsilon^2 \cdot \mathbf{E}[X]^2} \\ &= \frac{(2^n - \#\text{DNF-SAT}(\varphi)) \cdot \#\text{DNF-SAT}(\varphi)}{k \cdot \varepsilon^2 \cdot \#\text{DNF-SAT}(\varphi)^2} \\ &= \frac{2^n - \#\text{DNF-SAT}(\varphi)}{k \cdot \varepsilon^2 \cdot \#\text{DNF-SAT}(\varphi)}\end{aligned}$$

Un FPRAS para #DNF-SAT: primer intento

Realizando k muestras independientes de X obtenemos entonces:

$$\begin{aligned}\Pr\left(\left|\frac{1}{k} \cdot \sum_{i=1}^k X_i - \mathbf{E}[X]\right| \geq \varepsilon \cdot \mathbf{E}[X]\right) &\leq \frac{\mathbf{Var}[X]}{k \cdot \varepsilon^2 \cdot \mathbf{E}[X]^2} \\ &= \frac{(2^n - \#\text{DNF-SAT}(\varphi)) \cdot \#\text{DNF-SAT}(\varphi)}{k \cdot \varepsilon^2 \cdot \#\text{DNF-SAT}(\varphi)^2} \\ &= \frac{2^n - \#\text{DNF-SAT}(\varphi)}{k \cdot \varepsilon^2 \cdot \#\text{DNF-SAT}(\varphi)}\end{aligned}$$

Por lo tanto, para obtener $\Pr(|\frac{1}{k} \cdot \sum_{i=1}^k X_i - \mathbf{E}[X]| \geq \varepsilon \cdot \mathbf{E}[X]) \leq \frac{1}{4}$ necesitamos imponer la siguiente restricción sobre k :

$$\frac{4 \cdot (2^n - \#\text{DNF-SAT}(\varphi))}{\varepsilon^2 \cdot \#\text{DNF-SAT}(\varphi)} \leq k$$

Un FPRAS para #DNF-SAT: primer intento

Tomando entonces:

$$k = \left\lceil \frac{4 \cdot (2^n - \# \text{DNF-SAT}(\varphi))}{\varepsilon^2 \cdot \# \text{DNF-SAT}(\varphi)} \right\rceil$$

obtenemos la cota superior deseada para la probabilidad de error.

Un FPRAS para #DNF-SAT: primer intento

Tomando entonces:

$$k = \left\lceil \frac{4 \cdot (2^n - \# \text{DNF-SAT}(\varphi))}{\varepsilon^2 \cdot \# \text{DNF-SAT}(\varphi)} \right\rceil$$

obtenemos la cota superior deseada para la probabilidad de error.

¡Pero el valor de k puede ser exponencial en n , lo cual significa que el algoritmo aleatorizado resultante es de tiempo exponencial!

- ▶ Por ejemplo, si $\# \text{DNF-SAT}(\varphi)$ es polinomial en n

Un FPRAS para #DNF-SAT: primer intento

Tomando entonces:

$$k = \left\lceil \frac{4 \cdot (2^n - \# \text{DNF-SAT}(\varphi))}{\varepsilon^2 \cdot \# \text{DNF-SAT}(\varphi)} \right\rceil$$

obtenemos la cota superior deseada para la probabilidad de error.

¡Pero el valor de k puede ser exponencial en n , lo cual significa que el algoritmo aleatorizado resultante es de tiempo exponencial!

- Por ejemplo, si $\# \text{DNF-SAT}(\varphi)$ es polinomial en n

Vamos a estudiar un segundo enfoque que sí da el resultado deseado.

- El primer enfoque no utilizaba el hecho de que φ está en DNF

Un FPRAS para $\#DNF\text{-SAT}$: segundo intento

Recuerde que $\varphi = D_1 \vee \cdots \vee D_m$

- Donde cada D_i menciona ℓ_i literales, y no contiene literales repetidos ni complementarios

Para cada $i \in \{1, \dots, m\}$ sea $S_i = |\{\sigma \mid \sigma(D_i) = 1\}|$, y sea $M = \sum_{i=1}^m S_i$

- Tenemos que $S_i = 2^{n-\ell_i}$

Además, para cada valuación σ sea $d(\sigma) = |\{i \in \{1, \dots, m\} \mid \sigma(D_i) = 1\}|$

- Tenemos que $\sigma(\varphi) = 1$ si y sólo si $d(\sigma) \geq 1$

Un FPRAS para $\#DNF\text{-SAT}$: segundo intento

Suponga que las asignaciones de valores de verdad σ para φ son escogidas de manera que:

$$\mathbf{Pr}(\sigma) = \frac{d(\sigma)}{M}$$

Nótese que esta probabilidad está bien definida puesto que $1 \leq M$ y $d(\sigma) \leq M$

Y considere la siguiente variable aleatoria:

$$Y(\sigma) = \begin{cases} \frac{M}{d(\sigma)} & \sigma(\varphi) = 1 \\ 0 & \sigma(\varphi) = 0 \end{cases}$$

Un FPRAS para $\#$ DNF-SAT: segundo intento

Tenemos que:

$$\begin{aligned}\mathbf{E}[Y] &= \sum_{\sigma} Y(\sigma) \cdot \mathbf{Pr}(\sigma) \\ &= \sum_{\sigma : \sigma(\varphi)=1} \frac{M}{d(\sigma)} \cdot \frac{d(\sigma)}{M} \\ &= \sum_{\sigma : \sigma(\varphi)=1} 1 \\ &= \# \text{DNF-SAT}(\varphi)\end{aligned}$$

Un FPRAS para #DNF-SAT: segundo intento

Tenemos que:

$$\begin{aligned}\mathbf{E}[Y] &= \sum_{\sigma} Y(\sigma) \cdot \mathbf{Pr}(\sigma) \\ &= \sum_{\sigma : \sigma(\varphi)=1} \frac{M}{d(\sigma)} \cdot \frac{d(\sigma)}{M} \\ &= \sum_{\sigma : \sigma(\varphi)=1} 1 \\ &= \# \text{DNF-SAT}(\varphi)\end{aligned}$$

Tenemos entonces que $\mathbf{E}[Y] = \# \text{DNF-SAT}(\varphi)$, el primer requisito que debemos cumplir

- ¿Cómo podemos muestrear Y en tiempo polinomial en el tamaño de φ ?

Muestreando Y en tiempo polinomial

Para obtener un valor de Y realizamos los siguientes pasos:

1. Escoja $i \in \{1, \dots, m\}$ con probabilidad $\frac{S_i}{M}$
2. Escoja σ tal que $\sigma(D_i) = 1$ con distribución uniforme
3. Retorne $Y(\sigma)$

Muestreando Y en tiempo polinomial

Para obtener un valor de Y realizamos los siguientes pasos:

1. Escoja $i \in \{1, \dots, m\}$ con probabilidad $\frac{S_i}{M}$
2. Escoja σ tal que $\sigma(D_i) = 1$ con distribución uniforme
3. Retorne $Y(\sigma)$

¿Cómo se realiza el paso 1 en tiempo polinomial en el tamaño de φ ?

Muestreando Y en tiempo polinomial

Para obtener un valor de Y realizamos los siguientes pasos:

1. Escoja $i \in \{1, \dots, m\}$ con probabilidad $\frac{S_i}{M}$
2. Escoja σ tal que $\sigma(D_i) = 1$ con distribución uniforme
3. Retorne $Y(\sigma)$

¿Cómo se realiza el paso 1 en tiempo polinomial en el tamaño de φ ?

- ¿Por qué no se puede hacer algo similar y escoger directamente σ con probabilidad $\frac{d(\sigma)}{M}$?

Muestreando Y en tiempo polinomial

Nos falta calcular la probabilidad de escoger σ en los pasos 1 y 2

Muestreando Y en tiempo polinomial

Nos falta calcular la probabilidad de escoger σ en los pasos 1 y 2

Tenemos que:

$$\begin{aligned}\Pr(\sigma) &= \sum_{i=1}^m \Pr(\sigma \mid D_i) \cdot \Pr(D_i) \\ &= \sum_{i: \sigma(D_i)=1} \Pr(\sigma \mid D_i) \cdot \Pr(D_i) \\ &= \sum_{i: \sigma(D_i)=1} \frac{1}{S_i} \cdot \frac{S_i}{M} \\ &= \sum_{i: \sigma(D_i)=1} \frac{1}{M} \\ &= \frac{1}{M} \cdot \sum_{i: \sigma(D_i)=1} 1 \\ &= \frac{d(\sigma)}{M}\end{aligned}$$

Un FPRAS para $\#DNF\text{-SAT}$: segundo intento

Nos falta calcular $\mathbf{Var}[Y]$ para saber si podemos obtener un FPRAS.

Un FPRAS para $\#DNF\text{-SAT}$: segundo intento

Nos falta calcular $\mathbf{Var}[Y]$ para saber si podemos obtener un FPRAS.

- ▶ En lugar de hacer esto, vamos a acotar superiormente $\frac{\mathbf{Var}[Y]}{\mathbf{E}[Y]^2}$

Un FPRAS para $\#DNF\text{-SAT}$: segundo intento

Nos falta calcular $\mathbf{Var}[Y]$ para saber si podemos obtener un FPRAS.

- ▶ En lugar de hacer esto, vamos a acotar superiormente $\frac{\mathbf{Var}[Y]}{\mathbf{E}[Y]^2}$

Si $\sigma(\varphi) = 1$ tenemos que $\frac{M}{m} \leq Y(\sigma) \leq M$

- ▶ Por lo tanto $\frac{M}{m} \leq \mathbf{E}[Y] \leq M$

Concluimos que:

$$\frac{M}{m} - M \leq Y - \mathbf{E}[Y] \leq M - \frac{M}{m}$$

Un FPRAS para #DNF-SAT: segundo intento

Tenemos entonces que:

$$|Y - \mathbf{E}[Y]| \leq \frac{M}{m} \cdot (m - 1)$$

De lo cual se concluye que $(Y - \mathbf{E}[Y])^2 \leq (\frac{M}{m})^2 \cdot (m - 1)^2$

► Por lo tanto $\mathbf{Var}[Y] \leq (\frac{M}{m})^2 \cdot (m - 1)^2$

Un FPRAS para #DNF-SAT: segundo intento

Tenemos entonces que:

$$|Y - \mathbf{E}[Y]| \leq \frac{M}{m} \cdot (m - 1)$$

De lo cual se concluye que $(Y - \mathbf{E}[Y])^2 \leq \left(\frac{M}{m}\right)^2 \cdot (m - 1)^2$

► Por lo tanto $\mathbf{Var}[Y] \leq \left(\frac{M}{m}\right)^2 \cdot (m - 1)^2$

Como $\frac{M}{m} \leq \mathbf{E}[Y]$, sabemos que $\frac{1}{\mathbf{E}[Y]} \leq \frac{m}{M}$

Concluimos que: $\frac{\mathbf{Var}[Y]}{\mathbf{E}[Y]^2} \leq \left(\frac{m}{M}\right)^2 \cdot \left(\frac{M}{m}\right)^2 \cdot (m - 1)^2 = (m - 1)^2$

Un FPRAS para #DNF-SAT: segundo intento

Realizando k muestras independientes de Y obtenemos entonces:

$$\begin{aligned} \Pr\left(\left|\frac{1}{k} \cdot \sum_{i=1}^k Y_i - \mathbf{E}[Y]\right| \geq \varepsilon \cdot \mathbf{E}[Y]\right) &\leq \frac{\mathbf{Var}[Y]}{k \cdot \varepsilon^2 \cdot \mathbf{E}[Y]^2} \\ &\leq \frac{(m-1)^2}{k \cdot \varepsilon^2} \end{aligned}$$

Un FPRAS para $\#$ DNF-SAT: segundo intento

Realizando k muestras independientes de Y obtenemos entonces:

$$\begin{aligned}\Pr\left(\left|\frac{1}{k} \cdot \sum_{i=1}^k Y_i - \mathbf{E}[Y]\right| \geq \varepsilon \cdot \mathbf{E}[Y]\right) &\leq \frac{\mathbf{Var}[Y]}{k \cdot \varepsilon^2 \cdot \mathbf{E}[Y]^2} \\ &\leq \frac{(m-1)^2}{k \cdot \varepsilon^2}\end{aligned}$$

Por lo tanto, para obtener $\Pr(|\frac{1}{k} \cdot \sum_{i=1}^k Y_i - \mathbf{E}[Y]| \geq \varepsilon \cdot \mathbf{E}[Y]) \leq \frac{1}{4}$ imponemos la siguiente restricción sobre k :

$$\frac{4 \cdot (m-1)^2}{\varepsilon^2} \leq k$$

Un FPRAS para #DNF-SAT: segundo intento

Tomando entonces:

$$k = \left\lceil \frac{4 \cdot (m - 1)^2}{\varepsilon^2} \right\rceil$$

obtenemos la cota superior deseada para la probabilidad de error.

Un FPRAS para #DNF-SAT: segundo intento

Tomando entonces:

$$k = \left\lceil \frac{4 \cdot (m - 1)^2}{\varepsilon^2} \right\rceil$$

obtenemos la cota superior deseada para la probabilidad de error.

¡El algoritmo aleatorizado resultante funciona en tiempo polinomial en el tamaño de φ y $\frac{1}{\varepsilon}$!

Un FPRAS para #DNF-SAT: segundo intento

Tomando entonces:

$$k = \left\lceil \frac{4 \cdot (m-1)^2}{\varepsilon^2} \right\rceil$$

obtenemos la cota superior deseada para la probabilidad de error.

¡El algoritmo aleatorizado resultante funciona en tiempo polinomial en el tamaño de φ y $\frac{1}{\varepsilon}$!

- ▶ Debemos realizar k muestras independientes de Y , donde k depende polinomialmente de m y $\frac{1}{\varepsilon}$
- ▶ Cada una de las muestras puede ser obtenida en tiempo polinomial en el tamaño de φ

¿Toda función en $\#P$ admite un FPRAS?

¿Toda función en $\#P$ admite un FPRAS?

Proposición

Si una función f admite un FPRAS, entonces $L_f \in BPP$

¿Toda función en $\#P$ admite un FPRAS?

Proposición

Si una función f admite un FPRAS, entonces $L_f \in BPP$

Demostración: Suponga que $f : \Sigma^* \rightarrow \mathbb{N}$, y sea $\mathcal{A} : \Sigma^* \times (0, 1) \rightarrow \mathbb{N}$ un FPRAS para f

Dado $w \in \Sigma^*$, considere el siguiente algoritmo aleatorizado $\mathcal{B}$ para verificar si $w \in L_f$:

1. Sea k el resultado de ejecutar $\mathcal{A}(w, \frac{1}{2})$
2. Si $k > 0$ entonces retorne **sí**, en caso contrario retorne **no**

Una demostración de la proposición

Necesitamos calcular la probabilidad de error del algoritmo $\mathcal{B}$

Suponga primero que $w \in L_f$, vale decir, $f(w) > 0$

Dado que $\mathcal{A}$ es un FPRAS para f , tenemos que:

$$\Pr(|\mathcal{A}(w, \frac{1}{2}) - f(w)| \leq \frac{1}{2} \cdot f(w)) \geq \frac{3}{4}$$

Lo cual es equivalente a:

$$\Pr(\frac{1}{2} \cdot f(w) \leq \mathcal{A}(w, \frac{1}{2}) \leq \frac{3}{2} \cdot f(w)) \geq \frac{3}{4}$$

Una demostración de la proposición

Concluimos entonces que:

$$\Pr\left(\frac{1}{2} \cdot f(w) \leq \mathcal{A}(w, \frac{1}{2})\right) \geq \frac{3}{4}$$

Dado que $f(w) > 0$, tenemos que $\frac{1}{2} \cdot f(w) > 0$, por lo que tenemos lo siguiente:

$$\Pr(0 < \mathcal{A}(w, \frac{1}{2})) \geq \frac{3}{4}$$

Vale decir, $\Pr(\mathcal{B} \text{ retorne sí}) \geq \frac{3}{4}$

Una demostración de la proposición

Suponga ahora que $w \notin L_f$, vale decir, $f(w) = 0$

Sabemos que:

$$\Pr\left(\frac{1}{2} \cdot f(w) \leq \mathcal{A}(w, \frac{1}{2}) \leq \frac{3}{2} \cdot f(w)\right) \geq \frac{3}{4}$$

Dado que $f(w) = 0$, concluimos entonces que $\Pr(\mathcal{A}(w, \frac{1}{2}) = 0) \geq \frac{3}{4}$

Vale decir, $\Pr(\mathcal{B} \text{ retorne no}) \geq \frac{3}{4}$

Una demostración de la proposición

Suponga ahora que $w \notin L_f$, vale decir, $f(w) = 0$

Sabemos que:

$$\Pr\left(\frac{1}{2} \cdot f(w) \leq \mathcal{A}(w, \frac{1}{2}) \leq \frac{3}{2} \cdot f(w)\right) \geq \frac{3}{4}$$

Dado que $f(w) = 0$, concluimos entonces que $\Pr(\mathcal{A}(w, \frac{1}{2}) = 0) \geq \frac{3}{4}$

Vale decir, $\Pr(\mathcal{B} \text{ retorne no}) \geq \frac{3}{4}$

De los dos casos concluimos que $\Pr(\mathcal{B} \text{ retorne un resultado incorrecto}) \leq \frac{1}{4}$ $\square$

¿Qué funciones en $\#P$ tienen FPRAS?

Por la proposición anterior no esperamos tener FPRAS para $\#SAT$ y $\#CNF-SAT$

- ▶ A menos que $NP \subseteq BPP$

¿Qué funciones en $\#P$ tienen FPRAS?

Por la proposición anterior no esperamos tener FPRAS para $\#SAT$ y $\#CNF-SAT$

- ▶ A menos que $NP \subseteq BPP$

¿Cada función $f \in \#P$ tal que $L_f \in BPP$ admite un FPRAS?

¿Qué funciones en $\#P$ tienen FPRAS?

Por la proposición anterior no esperamos tener FPRAS para $\#SAT$ y $\#CNF-SAT$

- ▶ A menos que $NP \subseteq BPP$

¿Cada función $f \in \#P$ tal que $L_f \in BPP$ admite un FPRAS?

- ▶ ¿Al menos esto se cumple para cada función $g \in \#P$ tal que $L_g \in PTIME$?

¿Qué funciones en $\#P$ tienen FPRAS?

Por la proposición anterior no esperamos tener FPRAS para $\#SAT$ y $\#CNF-SAT$

- ▶ A menos que $NP \subseteq BPP$

¿Cada función $f \in \#P$ tal que $L_f \in BPP$ admite un FPRAS?

- ▶ ¿Al menos esto se cumple para cada función $g \in \#P$ tal que $L_g \in PTIME$?

Para responder esta pregunta necesitamos considerar otros problemas $\#P$ -completos