Complejidad probabilística

IIC3810

Un primer ejemplo: equivalencia de polinomios

Consideramos polinomios en $\mathbb{Q}$

Un polinomio es una expresión de la forma:

$$p(x) = \sum_{i=1}^{k} \prod_{j=1}^{\ell_i} (x + a_{i,j})$$

donde cada $a_{i,j} \in \mathbb{Q}$

La forma canónica de $p(x)$ es una expresión de la forma:

$$p(x) = \sum_{i=0}^{\ell} c_i x^i$$

donde cada $c_i \in \mathbb{Q}$ y $\ell = \text{máx}\{\ell_1, \ldots, \ell_k\}$

El grado de $p(x)$ es ℓ.

Un primer ejemplo: equivalencia de polinomios

Evaluamos los polinomios sobre elementos de $\mathbb{Q}$

- La operación más cara en este proceso es la multiplicación, por eso nos concentramos en ella

Ejercicio

1. De un algoritmo que, dado un polinomio $p(x)$ y $a \in \mathbb{Q}$, calcule $p(a)$.
2. De un algoritmo que, dado un polinomio $p(x)$ en su forma canónica y $a \in \mathbb{Q}$, calcule $p(a)$.

Los dos algoritmos deben realizar $O(\ell)$ multiplicaciones, donde ℓ es el grado de $p(x)$.

Un primer ejemplo: equivalencia de polinomios

Dados dos polinomios $p(x)$ y $q(x)$, queremos verificar si son idénticos.

- Para cada $a \in \mathbb{Q}$, se tiene que $p(a) = q(a)$

Definimos entonces el siguiente lenguaje:

$$\text{EQUIV-POL} \quad = \quad \{(p(x), q(x)) \mid p(x) \text{ y } q(x) \text{ son polinomios idénticos}\}$$

¿Cómo podemos resolver EQUIV-POL?

Un algoritmo para EQUIV-POL

Dados dos polinomios $p(x)$ y $q(x)$

1. Determine los grados k y ℓ de $p(x)$ y $q(x)$, respectivamente
2. Si $k = \ell$, entonces vaya al paso 3, sino retorne **no**
3. Transforme $p(x)$ y $q(x)$ en sus formas canónicas:

$$p(x) = \sum_{i=0}^{k} c_i x^i$$

$$q(x) = \sum_{i=0}^{k} d_i x^i$$

4. Verifique si $c_i = d_i$ para cada $i \in \{0, \ldots, k\}$. Si es así retorne **sí**, sino retorne **no**

Un algoritmo para EQUIV-POL

Ejercicio

1. Muestre que el algoritmo anterior es correcto.
2. Muestre además que realiza $O(k^2)$ multiplicaciones.

¿Es posible resolver este problema utilizando un menor número de multiplicaciones?

Un algoritmo *probabilístico* para EQUIV-POL

Dados dos polinomios $p(x)$ y $q(x)$

1. Determine los grados k y ℓ de $p(x)$ y $q(x)$, respectivamente
2. Si $k = \ell$, entonces vaya al paso 3, sino retorne **no**
3. Escoja al azar y uniformemente un elemento a del conjunto de números naturales $\{1, ..., 100k\}$
4. Verifique si $p(a) = q(a)$. Si es así retorne **sí**, sino retorne **no**

Un algoritmo *probabilístico* para EQUIV-POL

El algoritmo sólo necesita realizar $O(k)$ multiplicaciones

- Ya que necesita evaluar dos polinomios de grado k

Pero el algoritmo puede dar una respuesta equivocada

- ¿Cuál es la probabilidad de error?

Calculando la probabilidad de error

Sean $p(x)$ y $q(x)$ dos polinomios de grado k

- Si $(p(x), q(x)) \in$ EQUIV-POL, entonces el algoritmo responde **sí** sin cometer error

- Si $(p(x), q(x)) \notin$ EQUIV-POL, el algoritmo puede responder **sí** al sacar al azar un elemento $a \in \{1, \ldots, 100k\}$ tal que $p(a) = q(a)$

  Esto significa que a es una raíz del polinomio $r(x) = p(x) - q(x)$

Calculando la probabilidad de error

$r(x)$ no es el polinomio nulo y es de grado a lo más k

- Por lo tanto $r(x)$ tiene a lo más k raíces en $\mathbb{Q}$

Concluimos que:

$$\begin{aligned} \Pr(a \text{ sea una raíz de } r(x)) &\leq \frac{k}{100k} \\ &= \frac{1}{100} \end{aligned}$$

Un mejor algoritmo probabilístico

La probabilidad de error del algoritmo está acotada por $\frac{1}{100}$

- ¿Es aceptable esta probabilidad?

Ejercicio

De un algoritmo que resuelva EQUIV-POL, realice $O(k)$ multiplicaciones y tenga una probabilidad de error acotada por $\frac{1}{100^{10}}$

¿Confiaría en este algoritmo?

Algoritmos probabilísticos y Máquinas de Turing

¿Cómo podemos formalizar la idea de un algoritmo probabilístico utilizando la noción de MT?

¿Podemos definir clases de complejidad basados en los algoritmos probabilísticos?

Vamos a responder a esta preguntas en las siguientes transparencias.

MT probabilística

Definición

Una MT probabilística es una tupla $M = (Q, \Sigma, \Gamma, q_0, \delta, F)$ *tal que:*

- Q *es un conjunto finito de estados*
- Σ *es un alfabeto finito tal que* $\vdash, \mathtt{B} \notin \Sigma$
- Γ *es un alfabeto finito tal que* $\Sigma \cup \{\vdash, \mathtt{B}\} \subseteq \Gamma$
- $q_0 \in Q$ *es el estado inicial*
- $F \subseteq Q$ *es un conjunto de estados finales*
- δ *es una función parcial:*

$$\delta \quad : \quad Q \times \Gamma \times \{0, 1\} \to Q \times \Gamma \times \{\leftarrow, \square, \rightarrow\}$$

MT probabilística: Funcionamiento

La entrada de una MT probabilística M consiste de un string $w \in \Sigma^*$ y un string $s \in \{0,1\}^\omega$

- w es el input que se quiere aceptar o rechazar
- s es un string infinito de símbolos 0 y 1, el cual es considerado como un string de bits aleatorios

En el estado inicial:

- M tiene en la primera cinta $\vdash w\mathtt{B}\cdots$ y en la segunda cinta $\vdash s$
- M está en el estado q_0
- Las cabezas lectoras de ambas cintas están en la posición 1

MT probabilística: Funcionamiento

En cada instante la máquina se encuentra en un estado q y sus cabezas lectoras están en posiciones p_1 y p_2

- Si el símbolo en la posición p_i $(i = 1, 2)$ es a_i y $\delta(q, a_1, a_2) = (q', b, X)$, entonces:
  - La máquina escribe el símbolo b en la posición p_1 de la primera cinta
  - Cambia de estado desde q a q'
  - Mueve la cabeza lectora de la primera cinta a la posición $p_1 - 1$ si X es $\leftarrow$, y a la posición $p_1 + 1$ si X es $\rightarrow$. Si X es $\square$, entonces esta cabeza lectora permanece en la posición p_1
  - Mueve la cabeza lectora de la segunda cinta a la posición $p_2 + 1$

El tiempo de ejecución de una MT probabilística

La entrada de una MT probabilística M con alfabeto Σ consiste de dos strings $w \in \Sigma^*$ y $s \in \{0,1\}^\omega$

- Utilizamos la notación $M(w,s)$ para indicar las entradas de M
- Decimos que $M(w,s)$ acepta si M con entrada (w,s) se detiene en un estado final
  - El caso en que $M(w,s)$ rechaza se define de forma similar

Primer supuesto

Consideramos una MT probabilística M que se detiene en todas sus entradas (w,s)

El tiempo de ejecución de una MT probabilística

Un paso de una MT probabilística M consiste en ejecutar una instrucción de la función de transición

- Definimos $tiempo_M(w, s)$ como el número de pasos ejecutados por M con entrada (w, s)

Segundo supuesto

Existe una función $f : \Sigma^* \to \mathbb{N}$ tal que para cada $w \in \Sigma^*$ y $s \in \{0,1\}^\omega$:

$$tiempo_M(w, s) \quad \leq \quad f(w)$$

Vale decir, hay una cantidad máxima de bits aleatorios que deben ser utilizados con entrada w, la cual sólo depende de w

El tiempo de ejecución de una MT probabilística

Para estudiar el peor caso necesitamos la siguiente definición:

$$tiempo_M(w) \quad = \quad \text{máx}\{tiempo_M(w, s) \mid s \in \{0,1\}^\omega\}$$

Con esto tenemos que el tiempo de funcionamiento de M en el peor caso es definido por la función t_M:

$$t_M(n) \quad = \quad \text{máx}\{tiempo_M(w) \mid w \in \Sigma^* \text{ y } |w| = n\}$$

La probabilidad de aceptar en una MT probabilística

Tercer supuesto

Si para una MT probabilística M con alfabeto Σ se tiene que $t_M(n) \leq g(n)$ para todo $n \in \mathbb{N}$, entonces suponemos que las entradas de M son de la forma (w, s) con $w \in \Sigma^*$, $s \in \{0,1\}^*$ y $|s| = g(n)$.

Dado el tiempo de ejecución de M no podemos usar más de $g(n)$ bits aleatorios para una entrada w de largo n.

La probabilidad de aceptar en una MT probabilística

Sea M una MT probabilística con alfabeto Σ y tal que $t_M(n) \leq g(n)$ para todo $n \in \mathbb{N}$.

Definición

Para cada $w \in \Sigma^*$ tal que $|w| = n$, la probabilidad de que M acepte w es definida de la siguiente forma:

$$Pr(M \text{ acepte } w) = \frac{|\{s \in \{0,1\}^* \mid |s| = g(n) \text{ y } M(w,s) \text{ acepta}\}|}{2^{g(n)}}$$

Clases de complejidad probabilísticas

Vamos a definir una primera clase de complejidad considerando los algoritmos probabilísticos

- Esto nos va a permitir decir cuando un lenguaje es *aceptado* por una MT probabilística

Definición

Sea L un lenguaje sobre un alfabeto Σ. Entonces L está en RP si existe una MT probabilística M tal que $t_M(n)$ es $O(n^k)$ y para cada $w \in \Sigma^*$:

- *Si $w \in L$, entonces $Pr(M \text{ acepte } w) \geq \frac{3}{4}$*
- *Si $w \notin L$, entonces $Pr(M \text{ acepte } w) = 0$*

Vale decir, para los lenguaje en RP tenemos algoritmos probabilísticos que pueden cometer errores sólo para los elementos que están en L

La clase RP: un primer ejemplo

Ejercicio

Muestre que $\overline{\text{EQUIV-POL}} \in \text{RP}$

¿Por qué utilizamos la probabilidad $\frac{3}{4}$?

El valor $\frac{3}{4}$ es arbitrario

- Podemos utilizar valores arbitrariamente más pequeños

Lema de amplificación

Sea L un lenguaje sobre un alfabeto Σ. Si $L \in \text{RP}$, entonces para cada $\ell \in \mathbb{N}$, existe una MT probabilística M tal que $t_M(n)$ es $O(n^k)$ y para cada $w \in \Sigma^*$:

- Si $w \in L$, entonces $\Pr(M \text{ acepte } w) \geq 1 - \frac{1}{4^\ell}$
- Si $w \notin L$, entonces $\Pr(M \text{ acepte } w) = 0$

Ejercicio

Demuestre el lema de amplificación

¿Dónde está la clase RP?

Teorema

$PTIME \subseteq RP \subseteq NP$

Ejercicio

Demuestre el teorema.

Corolario

$PTIME \subseteq co\text{-}RP \subseteq co\text{-}NP$

¿Qué sabemos sobre RP y co-RP?

Son problemas abiertos si PTIME = RP o RP = co-RP

Pero se cree que PTIME = RP

- Puesto que si $L \in$ RP, entonces hay un algoritmo para resolver L puede ser usado en la *práctica* como un algoritmo de tiempo polinomial
- De esto se concluiría que RP = co-RP = PTIME

¿Hay algún ejemplo de un problema que está en RP y para el cual no se sabe si está en PTIME?

- $\overline{\text{EQUIV-POL}}$ no nos sirve como ejemplo ya que $\overline{\text{EQUIV-POL}} \in$ PTIME

En las siguientes transparencias vamos a ver este ejemplo.

Polinomios en varias variables

Consideramos polinomios en varias variables en $\mathbb{Q}$

Un monomio es una expresión de la forma $cx_1^{\ell_1} \cdots x_n^{\ell_n}$, donde $c \in \mathbb{Q}$ y cada $\ell_i \in \mathbb{N}$.

Un monomio $cx_1^{\ell_1} \cdots x_n^{\ell_n}$ es nulo si $c = 0$

- No es nulo si $c \neq 0$

El grado de un monomio $cx_1^{\ell_1} \cdots x_n^{\ell_n}$ no nulo es $\ell_1 + \cdots + \ell_n$.

Polinomios en varias variables

Un polinomio es una expresión de la forma:

$$p(x_1, \ldots, x_n) \quad = \quad \sum_{i=1}^{\ell} \prod_{j=1}^{m_i} \left(\sum_{k=1}^{n} a_{i,j,k} x_k + a_{i,j,n+1} \right)$$

donde cada $a_{i,j,k} \in \mathbb{Q}$ y cada $a_{i,j,n+1} \in \mathbb{Q}$

Polinomios en varias variables

La forma canónica de un polinomio $p(x_1, \ldots, x_n)$ es única, y es igual a 0 o a una suma de monomios que satisface las siguiente propiedades:

- cada monomio en la forma canónica es de la forma $cx_1^{\ell_1} \cdots x_n^{\ell_n}$ con $c \neq 0$
- si $cx_1^{\ell_1} \cdots x_n^{\ell_n}$ y $dx_1^{m_1} \cdots x_n^{m_n}$ son dos monomios distintos en la forma canónica, entonces $\ell_i \neq m_i$ para algún $i \in \{1, \ldots, n\}$

Un polinomio $p(x_1, \ldots, x_n)$ es nulo si su forma canónica es 0

El grado de un polinomio $p(x_1, \ldots, x_n)$ no nulo es el mayor grado de los monomios en su forma canónica.

Equivalencia de polinomios en varias variables

Dos polinomios $p(x_1, \ldots, x_n)$ y $q(x_1, \ldots, x_n)$ son idénticos si para cada secuencia $a_1, \ldots, a_n \in \mathbb{Q}$ se tiene que:

$$p(a_1, \ldots, a_n) \quad = \quad q(a_1, \ldots, a_n)$$

Nuevamente queremos verificar si dos polinomios son idénticos, para lo cual definimos el siguiente lenguaje:

$$\text{EQUIV-POL-V} = \{(p(x_1, \ldots, x_n), q(x_1, \ldots, x_n)) \mid p(x_1, \ldots, x_n) \text{ y } q(x_1, \ldots, x_n) \text{ son polinomios idénticos}\}$$

Equivalencia de polinomios en varias variables

¿Podemos resolver EQUIV-POL-V en tiempo polinomial?

- Nótese que ahora estamos considerando todas las operaciones, no sólo la multiplicación

Tenemos un problema: calcular la forma canónica de un polinomio toma tiempo exponencial

Pero podemos demostrar que $\overline{\text{EQUIV-POL-V}} \in \text{RP}$

- Esto no es trivial ya que un polinomio $p(x_1, \ldots, x_n)$ puede tener una cantidad infinita de raíces
  - Por ejemplo: $p(x_1, x_2) = (x_1 - 1)(x_2 - 3)$
- El ingrediente principal es el lema de Schwartz-Zippel

El ingrediente principal

Lema de Schwartz-Zippel

Sea $p(x_1, \ldots, p_n)$ un polinomio no nulo de grado k, y sea A un subconjunto finito y no vacío de $\mathbb{Q}$. Si $a_1, \ldots, a_n$ son elegidos de manera uniforme e independiente desde A, entonces

$$\Pr(p(a_1, \ldots, a_n) = 0) \quad \leq \quad \frac{k}{|A|}$$

Demostración: Por inducción en n.

Ya habíamos demostrado el caso $n = 1$.

Suponemos que la propiedad se cumple para todo polinomio en n variables, y consideramos un polinomio $p(x_1, x_2, \ldots, x_{n+1})$ de grado k.

Demostración del lema de Schwartz-Zippel

Si $p(x_1, x_2, \ldots, x_{n+1})$ en su forma canónica es igual a $c \in (\mathbb{Q} \setminus \{0\})$, entonces el lema se cumple trivialmente ya que

$$\Pr(p(a_1, a_2, \ldots, a_{n+1}) = 0) \quad = \quad 0$$

Suponemos entonces que $p(x_1, x_2, \ldots, x_{n+1})$ en su forma canónica no es igual a $c \in \mathbb{Q}$.

- Puesto que además sabemos que $p(x_1, x_2, \ldots, x_{n+1})$ no es nulo

Demostración del lema de Schwartz-Zippel

Tenemos que $p(x_1, x_2, \ldots, x_{n+1})$ en su forma canónica contiene un monomio de la forma:

$$c x_1^{\ell_1} x_2^{\ell_2} \cdots x_{n+1}^{\ell_{n+1}}$$

donde $c \neq 0$ y $\ell_i > 0$ para algún $i \in \{1, \ldots, n+1\}$.

Sin perdida de generalidad suponemos que en el monomio anterior $\ell_1 > 0$.

Tenemos que:

$$p(x_1, x_2, \ldots, x_{n+1}) \quad = \quad \sum_{i=0}^{k} x_1^i p_i(x_2, \ldots, x_{n+1})$$

donde cada $p_i(x_2, \ldots, x_{n+1})$ es un polinomio y al menos uno de ellos no es nulo

Demostración del lema de Schwartz-Zippel

Sea $\ell = \text{máx}\{i \in \{0, \ldots, k\} \mid p_i(x_2, \ldots, x_{n+1}) \text{ no es nulo}\}$

- Tenemos que $\ell > 0$ ya que supusimos que $\ell_1 > 0$

Sea A un subconjunto finito y no vacío de $\mathbb{Q}$

Dado que el grado de $p(x_1, x_2, \ldots, x_{n+1})$ es k, tenemos que el grado de $p_\ell(x_2, \ldots, x_{n+1})$ es m con $m \leq k - \ell$.

Por hipótesis de inducción tenemos que

$$\begin{aligned} \Pr(p_\ell(a_2, \ldots, a_{n+1}) = 0) &\leq \frac{m}{|A|} \\ &\leq \frac{k - \ell}{|A|} \end{aligned}$$

Demostración del lema de Schwartz-Zippel

Si $p_\ell(a_2, \ldots, a_{n+1}) \neq 0$, entonces por definición de ℓ tenemos que $q(x_1) = p(x_1, a_2, \ldots, a_{n+1})$ es un polinomio de grado ℓ.

Por lo tanto:

$$\Pr(p(a_1, a_2, \ldots, a_{n+1}) = 0 \mid p_\ell(a_2, \ldots, a_{n+1}) \neq 0) \quad \leq \quad \frac{\ell}{|A|}$$

Demostración del lema de Schwartz-Zippel

Concluimos que:

$$
\begin{aligned}
&\Pr(p(a_1, a_2, \ldots, a_{n+1}) = 0) \; = \\
&\quad \Pr(p(a_1, a_2, \ldots, a_{n+1}) = 0 \mid p_\ell(a_2, \ldots, a_{n+1}) = 0) \cdot \Pr(p_\ell(a_2, \ldots, a_{n+1}) = 0) \; + \\
&\quad \Pr(p(a_1, a_2, \ldots, a_{n+1}) = 0 \mid p_\ell(a_2, \ldots, a_{n+1}) \neq 0) \cdot \Pr(p_\ell(a_2, \ldots, a_{n+1}) \neq 0) \; \leq \\
&\quad \Pr(p_\ell(a_2, \ldots, a_{n+1}) = 0) + \Pr(p(a_1, a_2, \ldots, a_{n+1}) = 0 \mid p_\ell(a_2, \ldots, a_{n+1}) \neq 0) \; \leq \\
&\quad \frac{k - \ell}{|A|} + \frac{\ell}{|A|} \; = \; \frac{k}{|A|}
\end{aligned}
$$

□

Un algoritmo probabilístico para EQUIV-POL-V

Ejercicio

Utilice el lema de Schwartz-Zippel para demostrar que $\overline{\text{EQUIV-POL-V}} \in \text{RP}$

- ¿Es necesario calcular los grados de los polinomios? ¿Se puede hacer esto?
- ¿Es necesario verificar si los polinomios tienen el mismo grado?
- ¿Es necesario verificar que los polinomios no son nulos?
  - El lema de Schwartz-Zippel se aplica sobre polinomios no nulos

Corolario

EQUIV-POL-V $\in$ *co-RP*