Algoritmos Aleatorizados

Parte II

Segundo semestre 2022

IIC2283

Prof. Nicolás Van Sint Jan

Recordatorio: Equivalencia de polinomios

Suponga que:

$$\begin{aligned} p(x) &= \sum_{i=1}^{k} \prod_{j=1}^{r_i} (a_{i,j} x + b_{i,j}) \\ q(x) &= \sum_{i=1}^{\ell} \prod_{j=1}^{s_i} (c_{i,j} x + d_{i,j}) \end{aligned}$$

Y queremos resolver el problema de decidir si $p(x) = q(x)$.

Utilizamos el siguiente algoritmo aleatorizado que realiza $\mathcal{O}(n)$ siendo $n = |p(x)| + |q(x)|$:

EquivPolAleatorizado($p(x)$, $q(x)$)

$K := 1 + \max\{r_1, \ldots, r_k, s_1, \ldots, s_\ell\}$

escoja al azar y con distribución uniforme un elemento a del conjunto de números naturales $\{1, ..., 100 \cdot K\}$

if $p(a) = q(a)$ **then return** sí

else return no

Recordatorio: Calculando la probabilidad de error

Sean $p(x)$ y $q(x)$ dos polinomios dados como entrada a **EquivPolAleatorizado**.

- Si los polinomios $p(x)$ y $q(x)$ son equivalentes, entonces el algoritmo responde **SÍ** sin cometer error.

- Si los polinomios $p(x)$ y $q(x)$ no son equivalentes, el algoritmo puede responder **SÍ** al sacar al azar un elemento $a \in \{1, \ldots, 100 \cdot K\}$ tal que $p(a) = q(a)$.

  Esto significa que a es una **raíz** del polinomio

$$r(x) = p(x) - q(x)$$

Recordatorio: Calculando la probabilidad de error

Sabemos que $r(x)$ no es el polinomio nulo y que es de grado a lo más K.

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

Concluimos que:

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

¿ Es **aceptable** esta probabilidad ?

Recordatorio: Mejorar la probabilidad de error

Suponga que $p(x)$ y $q(x)$ son de la forma definida en la versión anterior de **EquivPolAleatorizado**.

El siguiente algoritmo resuelve el problema de equivalencia entre polinomios en $\mathcal{O}(n)$ y tiene una probabildad de error acotada por

$$\Pr(\text{obtener una respuesta incorrecta}) \leq \frac{1}{100^{10}}$$

EquivPolAleatorizado($p(x)$, $q(x)$)

$K := 1 + \max\{r_1, \ldots, r_k, s_1, \ldots, s_\ell\}$
$A := \{1, ..., 100 \cdot K\}$
$total := 0$
for $i := 1$ **to** 10 **do**
  escoja al azar y con distribución uniforme un elemento a en A
  if $p(a) = q(a)$ **then** $total = total + 1$
if $total = 10$ **then return** sí
return no

Outline

Polinomios en varias variables

Outline

Una definición general de polinomios

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 $cx_1^{\ell_1} \cdots x_n^{\ell_n}$ se dice **nulo** si $c = 0$.

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

Una definición general de polinomios

Definición

Un **polinomio en varias variables** es una expresión de la forma:

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

donde cada $a_{i,j,k} \in \mathbb{Q}$ y cada $b_{i,j} \in \mathbb{Q}$.

Una definición general de polinomios

Proposición

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 $i \in \{1, \ldots, n\}$.

Definición

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

Definición

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:

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

¿ Podemos verificar en tiempo polinomial si dos polinomios en varias variables son **equivalentes** ?

Equivalencia de polinomios en varias variables

Problema: Calcular la forma canónica de un polinomio toma **tiempo exponencial**.

Sin embargo existe un **algoritmo aleatorizado eficiente** para este problema.

- Esto no es trivial ya que un polinomio $p(x_1, \ldots, x_n)$ puede tener **una cantidad infinita de raíces**.

Ejemplo

$$p(x_1, x_2) = (x_1 - 1)(x_2 - 3)$$

- El ingrediente esencial es el **lema de Schwartz-Zippel**.

El ingrediente principal

Lema de Schwartz-Zippel

Sea $p(x_1, \ldots, x_n)$ 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

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

Un algoritmo aleatorizado para la equivalencia de polinomios en varias variables

Vamos a dar un **algoritmo aleatorizado eficiente** para el problema de verificar si dos polinomios en varias variables son equivalentes.

Suponga que la entrada del algoritmo está dada por los siguientes polinomios:

$$
\begin{aligned}
p(x_1, \ldots, x_n) &= \sum_{i=1}^{\ell} \prod_{j=1}^{r_i} \left(\sum_{k=1}^{n} a_{i,j,k} x_k + b_{i,j} \right) \\
q(x_1, \ldots, x_n) &= \sum_{i=1}^{m} \prod_{j=1}^{s_i} \left(\sum_{k=1}^{n} c_{i,j,k} x_k + d_{i,j} \right)
\end{aligned}
$$

Un algoritmo aleatorizado para la equivalencia de polinomios en varias variables

EquivPolAleatorizado$(p(x_1, \ldots, x_n),\ q(x_1, \ldots, x_n))$

$K := 1 + \max\{r_1, \ldots, r_\ell, s_1, \ldots, s_m\}$

$A := \{1, ..., 100 \cdot K\}$

sea $a_1, \ldots, a_n$ una secuencia de números elegidos de manera uniforme e independiente desde A

if $p(a_1, \ldots, a_n) = q(a_1, \ldots, a_n)$ **then return** sí

else return no

Utilizando el lema de Schwartz-Zippel

Vamos a calcular la **probabilidad de error** del algoritmo:

- Si los polinomios $p(x_1, \ldots, x_n)$ y $q(x_1, \ldots, x_n)$ son equivalentes, entonces el algoritmo responde **SÍ** sin cometer error

- Si los polinomios $p(x_1, \ldots, x_n)$ y $q(x_1, \ldots, x_n)$ no son equivalentes, el algoritmo puede responder **SÍ** al escoger una secuencia de números $a_1, \ldots, a_n$ desde A tales que $p(a_1, \ldots, a_n) = q(a_1, \ldots, a_n)$
    - Donde $A = \{1, \ldots, 100 \cdot K\}$

  Esto significa que $(a_1, \ldots, a_n)$ es una raíz del polinomio $r(x_1, \ldots, x_n) = p(x_1, \ldots, x_n) - q(x_1, \ldots, x_n)$

Utilizando el lema de Schwartz-Zippel

$r(x_1, \ldots, x_n)$ no es el polinomio nulo y es de grado t con $t < K$

- Dado que $K = 1 + \max\{r_1, \ldots, r_\ell, s_1, \ldots, s_m\}$

Utilizando el lema de Schwartz-Zippel obtenemos:

$$\Pr(r(a_1, \ldots, a_n) = 0) \ \leq \ \frac{t}{|A|} \ < \ \frac{K}{|A|} \ = \ \frac{K}{100 \cdot K} \ = \ \frac{1}{100}$$

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

Demostración del lema de Schwartz-Zippel

Demostración

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

Demostración

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

Demostración

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

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

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 $m \leq k - \ell$

Sea A un subconjunto finito y no vacío de $\mathbb{Q}$, y sea $a_1, \ldots, a_{n+1}$ una secuencia de números elegidos de manera uniforme e independiente desde A

Demostración del lema de Schwartz-Zippel

Demostración

Por hipótesis de inducción tenemos que:

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

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:

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

Demostración del lema de Schwartz-Zippel

Demostración

Concluimos que:

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

$\square$

Un mejor algoritmo aleatorizado para el problema general

Ejercicio

De un algoritmo aleatorizado que resuelva el problema de equivalencia de polinomios en varias variables.

- La probabilidad de error del algoritmo debe estar acotada por $\frac{1}{100^{10}}$
- Debe existir una constante c tal que el algoritmo en el peor caso es $\mathcal{O}(m^c)$, donde m es el tamaño de la entrada
  - Si consideramos $p(x_1, \ldots, x_n)$ y $q(x_1, \ldots, x_n)$ como palabras sobre un cierto alfabeto, entonces $m = |p(x_1, \ldots, x_n)| + |q(x_1, \ldots, x_n)|$
  - Recuerdo que la operación básica a contar es la suma y multiplicación de números racionales.

Una aplicación: polinomios como circuitos