Complejidad de la Comunicación

Juan Sebastian Navarro

08 de Mayo de 2012

Introducción

Objetivo

Determinar la menor cantidad de comunicación necesaria dentro de un sistema, para realizar una cierta acción (normalmente, computar una función).
Se utiliza un protocolo preacordado.

Protocolo

Especificación que regula quien y que se transmite en cada turno.

El Modelo

- Hay dos partes (Alice y Bob), con capacidad computacional ilimitada.
- Hay un canal de comunicación perfecto.
- Al terminar ambas partes deben conocer el valor de la función.
- Cada parte puede transmitir una cantidad arbitrariamente larga de bits antes de que transmita la otra parte.

Notación

- La función se define desde $X \times Y$ en Z, donde X es el conjunto correspondiente a Alice, e Y el correspondiente a Bob.
- log es el logaritmo en base 2.
- Si S es un conjunto, entonces $|S|$ representa la cardinalidad de S.
- $n = \text{mín}(\{\lceil \log(|X|) \rceil, \lceil \log(|Y|) \rceil\})$.
- El término "complejidad" se referirá a la complejidad de la comunicación.
- Un input (x,y) representa que el elemento de Alice es x, y el de Bob es y.

Función como matriz

Ejemplo:

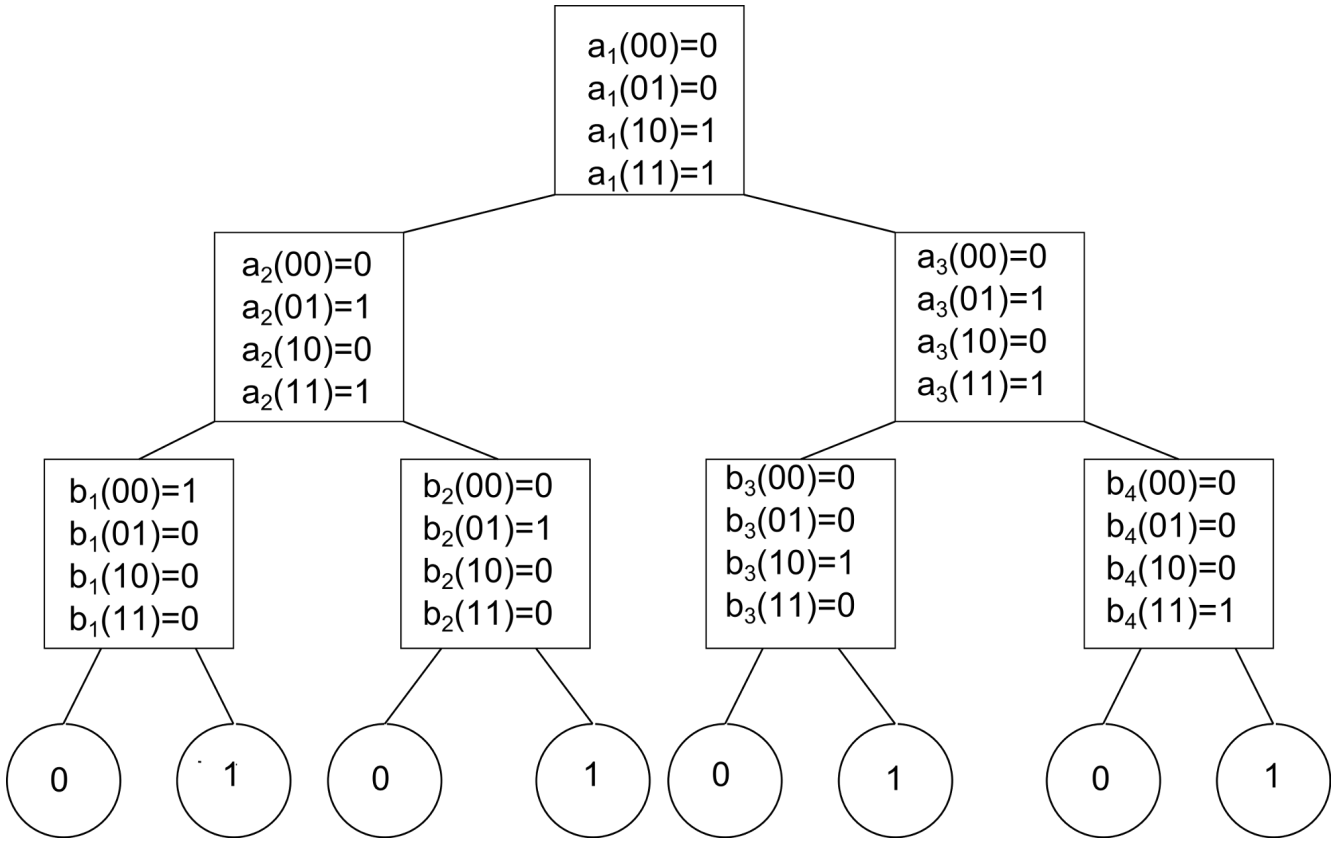
	00	01	10	11
00	1	1	1	1
01	1	0	1	0
10	1	1	0	0
11	1	0	0	0

Protocolo como árbol

Un protocolo puede representarse como un árbol binario, donde:

- Cada nodo interno contiene una función $a_i : X \to \{0, 1\}$, o $b_i : Y \to \{0, 1\}$.
- Cada hoja contiene un elemento de Z.
- La ejecución comienza en la raíz, y si el valor de la función es 0, pasa al hijo izquierdo, y si es 1, pasa al hijo derecho.
  El protocolo termina al alcanzar una hoja, la cual contiene el resultado.

Ejemplo (función EQ, donde EQ(x,y)=1 ssi x=y):



Complejidad determinística

Complejidad determinística de un protocolo $\mathcal{P}$, con input (x, y)

Cantidad de bits totales transmitidos por el protocolo $\mathcal{P}$, con input (x, y). Se denota por $s_{\mathcal{P}}(x, y)$.
En el ejemplo anterior, $s_{\mathcal{P}}(00, 10) = 3$.

Complejidad determinística de un protocolo $\mathcal{P}$

Cantidad de bits totales transmitidos por $\mathcal{P}$, en el peor caso:

$$D(\mathcal{P}) = \text{máx}\{s_{\mathcal{P}}(x, y) \mid (x, y) \in X \times Y\}$$

Es igual a la altura del árbol correspondiente.
En el ejemplo anterior, es 3.

Complejidad determinística de una función f

Complejidad del protocolo con menor complejidad, que resuelve f.

$$D(f) = \text{mín}\{D(\mathcal{P}) \mid \mathcal{P} \text{ resuelve } f\}$$

Rectángulos combinatoriales

Definición

Un rectángulo R se define como un conjunto de pares $(x, y) \in X \times Y$, tales que si $(x_1, y_1) \in R$, y $(x_2, y_2) \in R$, entonces $(x_1, y_2) \in R$ y $(x_2, y_1) \in R$.
Equivalentemente, como $R = X' \times Y'$, donde $X' \subseteq X$, e $Y' \subseteq Y$.

Rectángulo monocromático

Dada una función $f : X \times Y \to Z$, un rectángulo R se dice z-monocromático, si para todo elemento $(x, y) \in R$, se cumple que $f(x, y) = z$.

Ejemplo

Un rectángulo 1-monocromático:

	00	01	10	11
00	1	1	1	1
01	1	0	1	0
10	1	1	0	0
11	1	0	0	0

Definición

Al conjunto de pares (x, y), tales que al seguir un protocolo $\mathcal{P}$ con inputs (x, y), este pase por un nodo v, se denomina R_v.

Rectángulos combinatoriales

Lema

Para toda hoja l, se cumple que R_l es un rectángulo monocromático.

Demostración

Primero se demuestra que es un rectángulo, demostrando que para todo nodo v, R_v es un rectángulo.
Caso base: altura 0: $R_{root} = X \times Y$.
Paso inductivo: Si se cumple la propiedad para altura n, entonces si un nodo a altura n es m, con función a_v, y cuyo hijo izquierdo es p, entonces (los demás casos son análogos):

Demostración (cont.)

$$R_p = R_m \cap \{(x, y) \mid a_v(x) = 0\}$$

Por hipótesis de inducción, existen $X' \subseteq X$, y $Y' \subseteq Y$ tales que $R_m = X' \times Y'$.

$$R_p = (X' \times Y') \cap \{(x, y) \mid a_v(x) = 0\}$$

$$R_p = (X' \times Y') \cap (\{x \in X \mid a_v(x) = 0\} \times Y)$$

$$R_p = (X' \cap \{x \in X \mid a_v(x) = 0\}) \times (Y' \cap Y)$$

pero $(X' \cap \{x \in X \mid a_v(x) = 0\}) \subseteq X$, y $Y' \subseteq Y$, por lo tanto R_p es un rectángulo.

Además, para todos los inputs (x, y) cuya ejecución termina en una hoja l, se cumple que $f(x, y)$ es igual.

Teorema

Todo protocolo $\mathcal{P}$, para una función f, induce una partición de la matriz de f en rectángulos monocromáticos.

Demostración

Del lema se tiene que a cada hoja le corresponde un rectángulo monocromático, estos son disjuntos, ya que un input lleva al protocolo a una única hoja, y cubren toda la matriz ya que para todo input, el protocolo debe entregar algún resultado.

Cotas para la complejidad determinística

- Para toda función $f : X \times Y \to Z$, se tiene que $D(f) \leq n + \lceil \log(|Z|) \rceil$
  Demostración: Siempre es posible utilizar el siguiente protocolo:
  - Una parte envía su input completo (n bits).
  - la otra parte calcula el valor de f, y responde con este valor ($\lceil \log(|Z|) \rceil$ bits).
- Si toda partición de la matriz de una función f en rectángulos monocromáticos, tiene al menos t rectángulos, entonces $D(f) \geq \lceil \log(t) \rceil$
  Demostración:
  Todo protocolo induce una partición, luego si todas las particiones tienen al menos t rectángulos, entonces para todo protocolo, su árbol correspondiente tiene al menos t hojas, y como es un árbol binario, entonces tiene altura al menos $\lceil \log(t) \rceil$.

Fooling set

Definición

Un fooling set S, sobre una función f, se define como un subconjunto de $X \times Y$, tal que

- si $(x_1, y_1), (x_2, y_2) \in S$, entonces $f(x_1, y_1) = f(x_2, y_2) = z$.
- si $(x_1, y_1), (x_2, y_2) \in S$, y son distintos, entonces $f(x_1, y_2) \neq z$ o $f(x_2, y_1) \neq z$.

Fooling set

Teorema

Si una función f tiene un fooling set de cardinalidad t, entonces $D(f) \geq \lceil \log(t) \rceil$.

Demostración

Por definición 2 elementos distintos de S no pueden estar en el mismo rectángulo monocromático, luego toda partición tiene al menos t rectángulos, y por lo tanto $\lceil \log(t) \rceil \leq D(f)$.

Ejemplo

Dar una cota inferior para $D(EQ)$, donde
$EQ : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ toma valor 1, ssi sus parámetros son iguales.
Un fooling set de cardinalidad 2^n es $S = \{(w, w) \mid w \in \{0,1\}^n\}$, luego $D(EQ) \geq n$

Una generalización de lo anterior

Dada una distribución de probabilidad μ sobre $X \times Y$, tal que todo rectángulo monocromático R, sobre una función f, tenga medida $\mu(R) \leq \delta$, entonces $D(f) \geq \log(1/\delta)$.

Demostración

Como $\mu(X \times Y) = 1$, entonces hay al menos $1/\delta$ rectángulos monocromáticos en cualquier partición de la matriz.

Rango

Teorema

Si A es la matriz correspondiente a la función f, entonces

$$D(f) \geq \lceil \log(rank(A)) \rceil$$

Demostración

Se considera una partición óptima (cantidad de rectángulos mínima) con t rectángulos, y para cada rectángulo i se construye una matriz A_i donde se mantienen los valores del rectángulo i, y en las demás posiciones se colocan ceros. Luego se tiene que

$$A = \sum_{i=1}^{t} A_i$$

Rango

y además $rank(A_i)$ es 0 o 1, luego:

$$rank(A) \leq \sum_{i=1}^{t} rank(A_i) \leq t$$

y por lo tanto:

$$\lceil \log(rank(A)) \rceil \leq \lceil \log(t) \rceil \leq D(f)$$

Corolario

Si $Z = \{0, 1\}$, se tiene que:

$$D(f) \geq \lceil \log(rank(A) + rank(\bar{A})) \rceil$$

donde $\bar{A}$ es la matriz A, tras intercambiar ceros con unos.

Demostración

Se considera una partición óptima de la función con t rectángulos. Si esta tiene Z 0-rectángulos, y Q 1-rectángulos, entonces por el mismo argumento anterior, se tiene que:

$$rank(A) \leq Q$$

y que:

$$rank(\bar{A}) \leq Z$$

luego, como $t = Z + Q$, entonces:

$$rank(\bar{A}) + rank(A) \leq t$$

$$\log(rank(\bar{A}) + rank(A)) \leq \log(t) \leq D(f)$$

Ejercicio

Acotar el valor de $D(\text{DISJ})$, donde DISJ es definida como sigue: Dado un conjunto A de n elementos, entonces
DISJ : $2^A \times 2^A \to \{0,1\}$ cumple que $\text{DISJ}(x,y) = 1$ si y solo si x e y son disjuntos.

Solución

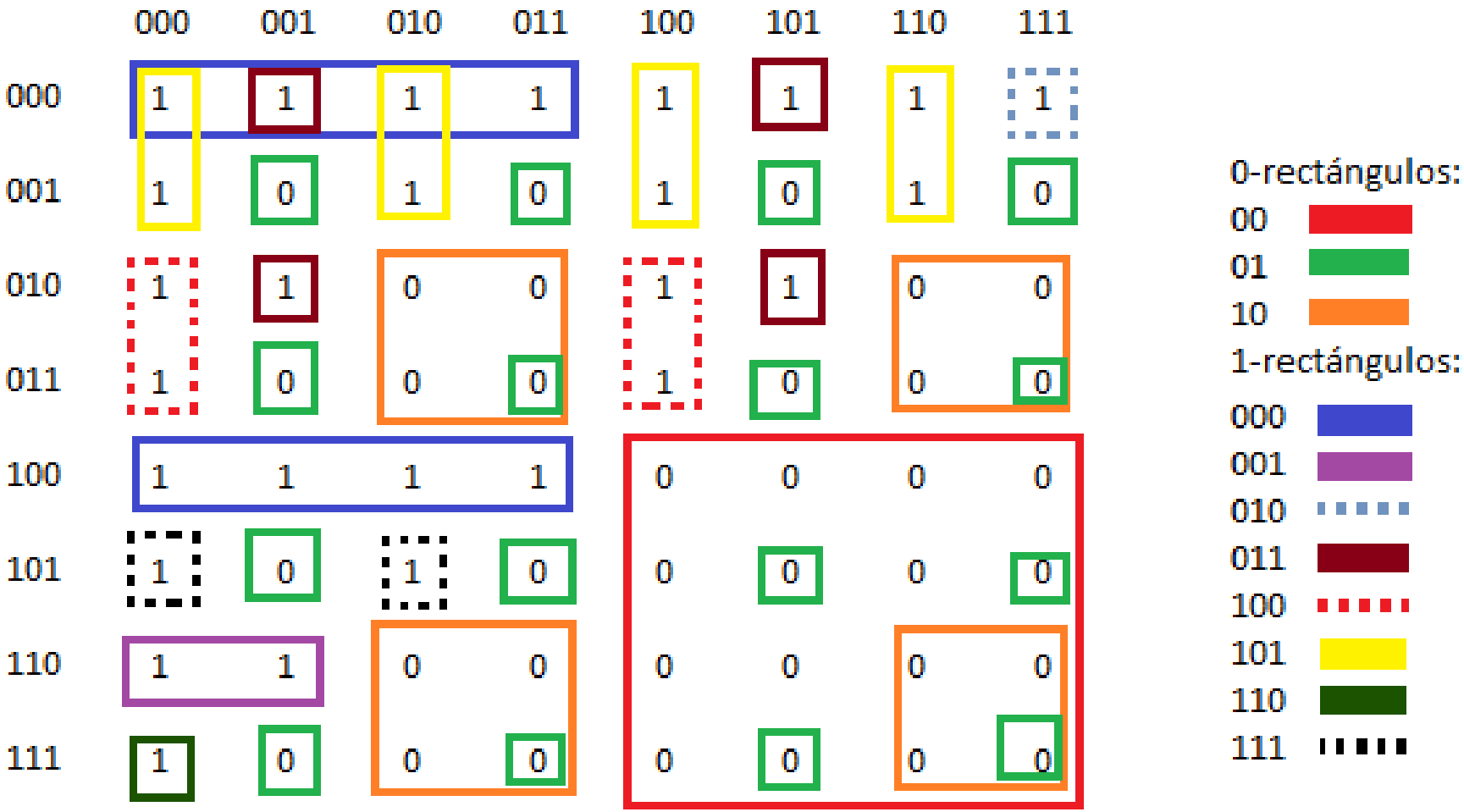
Un fooling set es $S = \{(x, \bar{x}) \mid x \in \{0,1\}^n\}$, el cual tiene cardinalidad 2^n, luego $D(DISJ) \geq n$.

Definiciones

Para una función f, con $Z = \{0, 1\}$, se definen:

- El número de partición por protocolo, $C^P(f)$, es la cantidad mínima de hojas de entre todos los árboles correspondientes a protocolos para f.
- El número de partición, $C^D(f)$, es la menor cantidad de rectángulos monocromáticos en los que se puede particionar la matriz de f.
- El número de cubrimiento de f, $C(f)$, es el menor número de rectángulos monocromáticos necesarios para cubrir la matriz de f (posiblemente de forma no disjunta).
- Para $z \in \{0, 1\}$, se define $C^z(f)$ como el menor numero de rectángulos necesarios para cubrir las entradas iguales a z de la matriz de f.

Ejemplo de cubrimiento óptimo (función DISJ)



Protocolo no determinístico

Definición

Un protocolo no determinístico consta de dos partes:

- Se genera de forma no determinística un string binario s.
- Se desarrolla un protocolo determinístico donde en cada turno, la parte involucrada utiliza como input tanto su input, como s.

Correctitud

Un protocolo es correcto si las ejecuciones con resultado igual a 1, pueden interpretarse como el valor correcto de la función.

Complejidad no determinística

Complejidad no determinística de un protocolo, con input (x, y)

Dado un protocolo no determinístico $\mathcal{P}_1$, cuyo protocolo determinístico es $\mathcal{P}_2$, entonces se define la complejidad no determinística de $\mathcal{P}_1$, con input (x, y) como $\text{mín}(\{|s| + s_{\mathcal{P}_2}(x, y, s) \mid \mathcal{P}_1(x, y, s) = 1\})$.

Complejidad no determinística de un protocolo

Dado un protocolo no determinístico $\mathcal{P}_1$, entonces se define la complejidad no determinística de $\mathcal{P}_1$ como el máximo valor tomado por la complejidad de $\mathcal{P}_1$ al considerar todos los inputs (x, y) tales que exista s, que cumpla que $\mathcal{P}_1(x, y, s) = 1$.

Complejidad no determinística

Complejidad no determinística de una función f ($N^1(f)$)

La complejidad del protocolo de menor complejidad que prueba que $f(x, y) = 1$.

Complejidad co-no determinística de una función f ($N^0(f)$)

La complejidad del protocolo de menor complejidad que prueba que $f(x, y) = 0$.

$N(f)$

La complejidad del protocolo de menor complejidad que prueba tanto $f(x, y) = 1$ como $f(x, y) = 0$.

Cotas para la complejidad no determinística

Lema

$N^1(f) = \Theta(\log(C^1(f)))$
$N^0(f) = \Theta(\log(C^0(f)))$
$N(f) = \Theta(\log(C(f)))$

Demostración

Se demuestra la primera parte, las otras dos son análogas.
Si se tiene un cubrimiento óptimo de las entradas iguales a 1, con los rectángulos numerados (codificados con $\log(C^1)$ bits), se genera de forma no determinística el nombre de un rectángulo, y luego Alice y Bob verifican que sus inputs estén en él, ocupando $O(log(C^1(f))) + 2$ bits. Con esto se muestra que $N^1(f) = O(log(C^1(f)))$.
Por otra parte, si se tiene un protocolo óptimo (con complejidad $N^1(f)$), se puede construir un cubrimiento de tamaño menor a $2^{N^1(f)+1}$, con lo cual $N^1(f) = \Omega(log(C^1(f)))$.

Relación entre complejidad determinística y no determinística

Lema

$D(f) = O(N^1(f)N^0(f))$

Demostración

Se considera un conjunto de rectángulos 0-monocromáticos L, que inicialmente contiene a todos los 0-rectángulos de un cubrimiento óptimo ($C^0(f)$ rectángulos).
Se desarrolla el siguiente protocolo determinístico:

Demostración

- Alice revisa L. Si $L = \emptyset$, termina el protocolo con $f(x, y) = 1$. Si L no es vacío, busca un 1-rectángulo tal que contenga algún elemento de la fila x, y que tenga filas comunes con a lo más $|L|/2$ rectángulos de L. Si existe envía su nombre a Bob, y si no, envía a Bob algún string que represente esta situación. Si el rectángulo existe, se define L como el conjunto de los 0-rectángulos con filas comunes a el, que estaban originalmente en L.
- Bob busca un 1-rectángulo que contenga a algún elemento de la columna y, y que tenga columnas comunes con a lo más la mitad de los elementos de L. Si este existe, envía el nombre a Alice, y define L como el conjunto de los 0-rectángulos con columnas comunes a él. Si no existe, termina el protocolo con $f(x, y) = 0$.

Correctitud

Si $f(x, y) = 0$, entonces (x, y) pertenece a algún 0-rectángulo, el cual siempre permanece en L, por lo cual el protocolo terminaría retornando el valor correcto.
Si $f(x, y) = 1$, entonces siempre se encontraría un 1-rectángulo con las propiedades pedidas (el que contiene (x, y)), y L se reduciría hasta el conjunto vacío, con lo cual el protocolo termina retornando el valor correcto.

Complejidad

En cada ronda se transmiten $2\lceil \log(C^1(f)) \rceil + O(1)$ bits, y se reduce L por lo menos a la mitad, por lo tanto hay a lo más $\lceil \log(C^0(f)) \rceil$ rondas, luego $D(f) = O(N^1(f)N^0(f))$.

Complejidad de la función $DISJ_k$

La función

Dado un conjunto A de cardinalidad n, se tiene que $x, y \subseteq A$, y $|x| = |y| = k \leq n/2$. Luego se define $DISJ_k(x, y) = 1$ ssi x e y son disjuntos.

Se define $m = \binom{n}{k}$

Complejidad de la función $DISJ_k$

Complejidad No Determinística

Se utiliza el método probabilístico para demostrar que $N^1(DISJ_k) = O(k + \log\log n)$. Se elige un conjunto $S \subseteq A$ de forma aleatoria (cada elemento tiene probabilidad de $1/2$ de pertenecer a S), y luego se define

$$R_S = \{x \mid x \subseteq S, |x| = k\} \times \{y \mid y \subseteq \bar{S}, |y| = k\}$$

el cual es un 1-rectángulo. Para cada input (x, y) tal que x e y sean disjuntos y de cardinalidad igual a k, se tiene que $Pr_S[(x, y) \in R_S] = 2^{-2k}$. Luego se eligen $t = 2^{2k} \ln(m^2)$ rectángulos de forma aleatoria (estos son R_S con distintas elecciones para S).

Complejidad de la función $DISJ_k$

Complejidad No Determinística

La probabilidad de que el input (x, y) no esté en la unión de estos es de $(1 - 2^{-2k})^t < 1/m^2$. Luego, como no todo input (a, b) cumple con que a sea disjunto de b, entonces la probabilidad de que un 1-input no este en alguno de estos t rectángulos es menor a 1, lo que es equivalente a que existe una probabilidad positiva de que un cubrimiento aleatorio como el señalado cubra todos los 1-inputs, y por lo tanto existe un cubrimiento de tamaño t. Luego $N^1(DISJ_k) = O(\log(t)) = O(k + \log\log n)$.

Complejidad de la función $DISJ_k$

Complejidad determinística

Se demuestra que $D(DISJ_k) \geq \log(m)$ por el método de rango.
Si la matriz asociada a la función es D_k^n, y $\vec{x}$ es la fila asociada al conjunto x, se demuestra por inducción en k y n, que D_k^n tiene rango completo.
Casos base: las matrices D_0^n ([1]), y D_k^{2k} (diagonal) tienen rango completo.
Paso inductivo: Sea X_1 el conjunto de las filas de D_k^n que corresponden a conjuntos que no contienen a n, y X_2 el conjunto de las que si contienen a n.
Análogamente se definen Y_1 como el conjunto de las columnas correspondientes a conjuntos que no contienen a n, y Y_2 al conjunto de las que si lo contienen.
Luego, $X_1 \times Y_1 = D_k^{n-1}$, y $X_2 \times Y_2$ es una matriz de ceros.

Complejidad de la función $DISJ_k$

Luego se aplica una transformación lineal sobre D_k^n, de forma de que $X_2 \times Y_1$ sea la matriz de ceros, y $X_2 \times Y_2$ sea D_{k-1}^{n-1}.

La transformación

Si $\vec{x}$ es una fila en X_2, entonces se reemplaza por:

$$\hat{x} = \frac{1}{n-2k+1} \cdot \vec{v}_x - \frac{n-2k}{n-2k+1} \cdot \vec{x}$$

donde

$$\vec{v}_x = \sum_{z|x' \subseteq z, |z|=k} \vec{z}$$

y $x = x' \cup \{n\}$

Complejidad de la función $DISJ_k$

Primero se verifica que para todo $y \in Y_1$, $\hat{x}[y] = 0$. Hay dos casos:

- Si $\vec{x}[y] = 0$, entonces x no es disjunto con y, y como x no contiene n, entonces x' no es disjunto con y. Luego para todo z, se tiene que $\vec{z}[y] = 0$, y por lo tanto $\vec{v}_x[y] = 0$. Entonces $\hat{x}[y] = 0 - 0 = 0$.
- Si $\vec{x}[y] = 1$, entonces x es disjunto con y, y en particular x' también lo es. Luego los conjuntos z que contienen a n' y son disjuntos a y, son los cuyo k-ésimo elemento no esta en $x \cup y$, y por lo tanto hay $n - 2k$ conjuntos posibles. luego $\vec{v}_x[y] = n - 2k$, y por lo tanto

$$\hat{x}[y] = \frac{n-2k}{n-2k+1} - \frac{n-2k}{n-2k+1} = 0$$

Complejidad de la función $DISJ_k$

Luego se verifica que para todo $y \in Y_2$, se cumpla que $\hat{x}[y] = DISJ(x', y')$, donde $y = y' \cup \{n\}$. Hay dos casos:

- Si x' y y' no son disjuntos, entonces para todo z, z no es disjunto con y', luego, $\vec{v}_x[y] = 0$, y $\hat{x}[y] = 0 - 0 = 0$.
- Si x' y y' son disjuntos, entonces la cantidad de conjuntos z que sean disjuntos a y' es $n - 2k + 1$, y por lo tanto:

$$\hat{x}[y] = \frac{n - 2k + 1}{n - 2k + 1} - 0 = 1$$

Complejidad de la función $DISJ_k$

Por hipótesis de inducción, tanto D_k^{n-1} como D_{k-1}^{n-1} tienen rango completo, entonces D_k^n tiene rango completo, y como es una matriz de m por m, entonces $D(DISJ_k) \geq \log(m)$.
Si $k = \log(n)$, entonces $D(DISJ_k) = \Omega(\log^2(n))$, y
$N^1(DISJ_k) = O(\log(n))$, y como además para todo k, se tiene que $N^0(f) = O(\log(n))$, entonces la cota del lema anterior es ajustada.

Complejidad aleatorizada

Protocolo aleatorizado con "moneda privada"

Corresponde a un protocolo donde cada parte tiene acceso a un string aleatorio privado, y donde cada función interna depende tanto del input original, como del string aleatorio.

Notación

- $\mathcal{P}(x, y)$ es el valor retornado por $\mathcal{P}$ con input (x, y), para algún valor de los strings aleatorios.
- El string aleatorio de Alice es r_A, y el de Bob es r_B.

Tipos de error

Para un protocolo $\mathcal{P}$, y una función f hay 3 definiciones:

- $\mathcal{P}$ calcula f con error 0, si para todo (x, y):

$$Pr[\mathcal{P}(x, y) = f(x, y)] = 1$$

- $\mathcal{P}$ calcula f con error ε, si para todo (x, y):

$$Pr[\mathcal{P}(x, y) = f(x, y)] \geq 1 - \varepsilon$$

- $\mathcal{P}$ calcula f con error ε por un lado, si para todo (x, y) tal que $f(x, y) = 0$ se cumple que:

$$Pr[\mathcal{P}(x, y) = f(x, y)] = 1$$

y para todo (x, y) tal que $f(x, y) = 1$ se cumple que:

$$Pr[\mathcal{P}(x, y) = f(x, y)] \geq 1 - \varepsilon$$

Complejidad aleatorizada

Hay 2 posibilidades, considerar el peor caso, o el caso promedio, luego se definen:

Complejidad de un protocolo $\mathcal{P}$, con input (x, y) en el peor caso

La cantidad máxima de bits transmitido por $\mathcal{P}$ con input (x, y), al considerar todas las posibilidades para r_A y r_B.

Complejidad de un protocolo $\mathcal{P}$, en el peor caso

La complejidad del peor caso máxima para $\mathcal{P}$ de entre todos los inputs posibles.

Complejidad de un protocolo $\mathcal{P}$, con input (x, y) en el caso promedio

La cantidad esperada de bits transmitido por $\mathcal{P}$ con input (x, y).

Complejidad de un protocolo $\mathcal{P}$, en el caso promedio

La complejidad del caso promedio máxima para $\mathcal{P}$ de entre todos los inputs posibles.

Complejidad aleatorizada

Dada una función f, con $Z = \{0, 1\}$, se tiene que:

- $R_0(f)$ es la mínima complejidad con error cero, en el caso promedio de entre todos los protocolos para f.
- para $0 < \varepsilon < 1/2$, $R_\varepsilon(f)$ es la mínima complejidad en el peor caso con error ε, de entre todos los protocolos para f.
  Notación: $R(f)$ es $R_{1/3}(f)$
- para $0 < \varepsilon < 1$, $R^1_\varepsilon(f)$ es la mínima complejidad en el peor caso, con error por un lado igual a ε, de entre todos los protocolos para f.
  Notación: $R^1(f)$ es $R^1_{1/2}(f)$.

Ejemplo

La función EQ toma valor 1 ssi sus inputs son iguales. Calcular R(EQ).
Si $x = a_0 a_1 \cdots a_{n-1}$, e $y = b_0 b_1 \cdots b_{n-1}$, se consideran los polinomios:

$$A(t) = a_0 + a_1 t + \cdots + a_{n-1} t^{n-1} \text{ (mod p)}$$
$$B(t) = b_0 + b_1 t + \cdots + b_{n-1} t^{n-1} \text{ (mod p)}$$

sobre $GF[p]$, con p un número primo tal que $n^2 < p < 2n^2$, y luego se desarrolla el siguiente protocolo:

Ejemplo

Alice elige aleatoriamente (distribución uniforme) un elemento t de $GF[p]$, y envía a Bob t y $A(t)$, es decir $O(\log(p))$ bits, y luego Bob termina el protocolo con output 1 si $A(t) = B(t)$, y 0 en otro caso. Si $x = y$, entonces siempre $A(t) = B(t)$, y por otra parte si $x \neq y$, entonces se considera el polinomio $A(t) - B(t)$, el cual es un polinomio no idénticamente igual a cero, de orden a lo más $n - 1$, luego tiene a lo más $n - 1$ raíces, y por lo tanto la probabilidad de error es a lo más $(n-1)/p \leq n/n^2 = 1/n$, luego $R(EQ) = R_{1/n}(EQ) = R^1_{1/n}(EQ) = O(\log(n))$. En contraste, con las técnicas vistas anteriormente puede demostrarse que $D(EQ) = N(EQ) = n + 1$.

Relación con otras formas de complejidad

Proposición

Para toda función f, y todo ε entre 0 y 1:

$$R_{\varepsilon}^{1}(f) \geq N^{1}(f)$$

$$R_{0}(f) \geq N(f)$$

Idea de demostración: Se pueden generar de forma no determinística los strings aleatorios.

Lema

Para toda función f se cumple que:

$$R(f) = \Omega(\log(D(f)))$$

Idea de demostración: Simular el protocolo aleatorizado por un protocolo determinístico.

Relación entre tiempo y espacio ocupados por una MT

La idea

La ejecución de una MT puede entenderse como un proceso de comunicación entre distintos momentos de la ejecución.

Lema

Sea $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ una función, y M una MT con k cintas de trabajo, que ejecuta en tiempo $T(n)$, y espacio $S(n)$, para inputs de tamaño $m = 3n$ que acepta todas las palabras de

$$\{x0^n y \mid |x| = |y| = n, f(x,y) = 1\}$$

y rechaza todas las de

$$\{x0^n y \mid |x| = |y| = n, f(x,y) = 0\}$$

Luego $D(f) = O(T(n)S(n)/n)$.

Relación entre tiempo y espacio ocupados por una MT

Demostración

Alice y Bob simulan conjuntamente la ejecución de la MT.
Si una parte esta simulando la maquina, y necesita leer una posición que no conoce de la cinta de entrada, entonces envía toda la información necesaria a la otra parte, y esta continua la ejecución.
Toda esta información es $O(S(n))$. Además hay al menos n pasos entre cada "cambio de contexto", y como hay $T(n)$ pasos totales, la cantidad de cambios es a lo más $T(n)/n$, y por lo tanto la cantidad de bits totales intercambiados es $O(S(n)T(n)/n)$.

Ejemplo

Se considera el lenguaje de los palíndromos sobre un alfabeto binario, y de estos los de la forma:

$$L = \{x0^n x^r \mid x \in \{0,1\}^n\}$$

o equivalentemente:

$$L = \{x0^n y \mid x \in \{0,1\}^n, EQ^r(x,y) = 1\}$$

donde $EQ^r(x,y) = 1$ ssi $x = y^r$.

$$D(EQ^r) = n + 1 = O(S(n)T(n)/n)$$

por lo tanto

$$S(n)T(n) = \Omega(n^2)$$

Número de estados de un AF

Lema

Sea $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ una función, $L \subseteq \{0,1\}^*$ un lenguaje (regular), y $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ un AFD, luego si:

$$L \cap \{0,1\}^{2n} = \{xy \mid |x| = |y| = n, f(x,y) = 1\}$$

entonces, $D(f) \leq \log(|Q|) + 1$.

Ejemplo

Se considera el lenguaje $L_n = \{xx \mid |x| = n\}$, o equivalentemente:

$$L_n = \{xy \mid |x| = |y| = n, EQ(x,y) = 1\}$$

Luego como $D(EQ) = n + 1$, entonces $n + 1 \leq \log(|Q|) + 1$, y por lo tanto $|Q| \geq 2^n$.
En conclusión, el lenguaje $\{xx \mid x \in \Sigma^*\}$ no puede ser regular.