Algorithmes Distribués

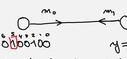
Détecter des C_3 en C_4.

Algorithme LOCAL CONGEST
- synchrone
- messages à $O(\log n)$ bits

$\Rightarrow$ $O(\sqrt{n})$ rondes

Est-ce le meilleur possible ?

Complexité de Communication



$x = 01100100$ $\quad y = 11000001$

$f(x,y) \in \{0,1\}$ $\quad |x| = |y| = n = 7$

$f(x,y) = 1 \Leftrightarrow \exists$ une position $i \in \{0 \dots n-1\}$ tq $x[i] = y[i] = 1$

P calculer $f(x,y)$, ie appliquant un "protocole de communication"

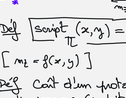
Règle :

1. Alice qui commence elle prépare un message $\rightarrow$ Bob.
2. À tour de rôle Bob $\rightarrow$ Alice $\rightarrow$ Bob $\rightarrow \dots$

ogr Le dernier message est $f(x,y)$.

Ex: Protocole pour $f(x,y)$ = "bit en commun"

Π = "On envoie sa position avec un '1' son ordre croissant jusqu'à ce qu'on trouve décide"



Déf $\boxed{\text{script}_\Pi(x,y) = (m_0, m_1, \dots, m_k)}$

$[\, m_k = f(x,y) \,]$

Déf Coût d'un protocole Π est la somme (en bits) des messages échangés, dans le pire des cas (des pires entrées possible)

Exercice Calculer le coût du protocole Π de l'exemple pour $x, y \in \{0,1\}^n$?

Réponse Alice et Bob vont s'envoyer des positions écrites en binaires $i \in \{0 \dots n-1\}$, $\lceil \log_2(i+1) \rceil \in \lceil \log_2 n \rceil$ bits

Les position envoyés : $(i_1, i_2, \dots, i_k, f(x,y))$ avec $i_1 < i_2 < \dots < i_k$

coût $(\Pi) \leq n \lceil \log_2 n \rceil + 1$ $\quad k \leq n$.

Rem. Il existe un protocole (trivial) qui a un coût de $\leq n+1$ bits, $\forall f$.

Π_t = Alice envoie x à Bob, Bob calcule $f(x,y)$ et l'envoie à Alice.

coût $(\Pi_t) \leq n+2$.

On va se concentrer sur la fonction SET-DISJOINTNESS

$\text{Disj}_n(x,y) = \bigvee_{i=0}^{n-1} x[i] \wedge y[i] = \begin{cases} 1 \Leftrightarrow x,y \text{ disjoint} \\ 0 \Leftrightarrow x,y \text{ intersectent} \end{cases}$

(ie Ne partagent aucun bit à 1)

$A, B \subseteq \{0, \dots, n-1\}$ $\quad A = \{2,5\}$ $\quad B = \{1,4,5,6\}$

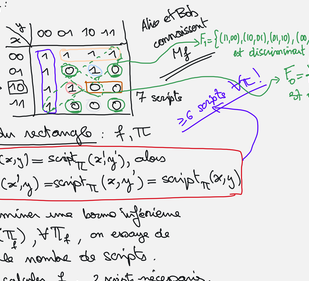
$x = 00\underline{1}00\underline{1}00$ $\quad y = 11\underline{0}11\underline{0}01$

$A \cap B = \emptyset$ ou pas ?

coût d'un protocole qui calcule Disj_n ?

Pour une fonction $f : \{0,1\}^n \times \{0,1\}^n \longrightarrow \{0,1\}$

Matrice de f :



Propriété du rectangle : f, Π

Si $\text{script}_\Pi(x,y) = \text{script}_\Pi(x',y')$, alors $\text{script}_\Pi(x',y) = \text{script}_\Pi(x,y') = \text{script}_\Pi(x,y)$

Pour déterminer une bonne référence sur coût (Π_f), $\forall \Pi_f$, on essaye de minorer le nombre de scripts.

Ex: Pour calculer f, 2 scripts nécessaires.

$S_1 = (\underbrace{\text{xxxxxxxx}}_{n}, \underbrace{m_2}_{\log n}) \Big\}$ coût$_1$ = max(|s|,|s|)

$S_2 = (\underbrace{\text{xxxxxxxx}}_{n}, m_2', \dots, m_k)$

Si il y a 2 scripts alors un autre protocole aurait pu avoir un coût $\leq$ coût$_1$.

si la 1ère différence est à une position paire, alors Alice envoie m_2 ou m_2'.

sinon, c'est Bob qui va déterminer si c'est s_1 ou s_2 (et donc m_2 ou m_2').

Alice envoie un premier message "0"

$\Rightarrow$ 2 bits suffisent.

Intuitivement, si pour tout protocole Π, 2^k scripts sont nécessaires, c'est qu'il faut au moins k bits transmettre.

Idée de la preuve : (H) $\text{script}_\Pi(x,y) = \text{script}_\Pi(x',y')$

Alice $m_0 = P_0(x)$, $\forall y$.

$\text{script}_\Pi(x,y)[0] = \text{script}_\Pi(x,y')[0]$ car le 1er message ne dépend pas de l'entrée y/y' de Bob.

$\text{script}_\Pi(x,y)[0] = \text{script}_\Pi(x',y')[0]$ (H).

$\text{script}_\Pi(x',y')[0] = \text{script}_\Pi(x',y)$

$\boxed{\text{script}_\Pi(x',y)[0] = \text{script}_\Pi(x,y)[0] = \text{script}_\Pi(x,y)[0]}$

Par induction sur $\text{script}_\Pi(x,y)[0 \dots i]$

$\vdots$ ❑

Ensemble discriminant

C'est un ensemble d'entrées $\{\dots (x,y) \dots\}$ qui violent 2 à 2 la propriété de rectangle.

Soit $v \in \{0,1\}$, $F_v \subseteq \{0,1\}^n \times \{0,1\}^n$

F_v est un ensemble discriminant pour v et f.

$\forall (x,y)(x',y') \in F_v$ $\quad x' \dots \overset{j}{\overline{x}} \dots \overline{x'}^{(j)}$ $\quad x \dots \underline{x_{(j)}} \dots \overset{j}{\overline{x}}$

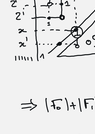
$f(x,y) = f(x',y') = v$

et $\overline{v} = f(x',y), f(x,y')\}$

Proposition Si F_0 et F_1 sont des ensemble discriminants pour f pour la valeurs 0 et 1, alors le nombre de scripts de TOUT protocole qui calcule f est au moins $\geq |F_0| + |F_1|$.

Proposition Tout protocole de communication qui calcul Disj_n possède au moins $2^n + n$ scripts différents.

Preuve : Disj_n



$F_1 = \{(x,\overline{x}) : x \in \{0,1\}^n\}$ discriminant pour $v=1$

$|F_1| = 2^n$

$F_0 = \{(2^i, 2^i) : i \in \{0 \dots n-1\}\}$ discriminant pour $v=0$

$|F_0| = n$

$\Rightarrow |F_0| + |F_1|$ ❑