Quelques mystères de la complexité algébique

Pascal Koiran
LIP, Ecole Normale Supérieure de Lyon

ENS Cachan, 10 septembre 2013

Qu'est-ce que la complexité algébrique ?

- But de la **complexité algorithmique :**
  déterminer le coût asymptotique du "meilleur algorithme"
  pour un problème donné.
  Le coût :
  temps de calcul, espace mémoire, communication, énergie...
- **En complexité algébrique :**
  coût = nombre d'opérations arithmétiques.

Qu'est-ce que la complexité algébrique ?

- But de la **complexité algorithmique :**
  déterminer le coût asymptotique du "meilleur algorithme"
  pour un problème donné.
  Le coût :
  temps de calcul, espace mémoire, communication, énergie...
- **En complexité algébrique :**
  coût = nombre d'opérations arithmétiques.

Qu'est-ce que la complexité algébrique ?

- But de la **complexité algorithmique :**
  déterminer le coût asymptotique du "meilleur algorithme"
  pour un problème donné.
  Le coût :
  temps de calcul, espace mémoire, communication, énergie...
- **En complexité algébrique :**
  coût = nombre d'opérations arithmétiques.

Premier mystère :
la complexité de la multiplication de matrices

Soient A et B des matrices $n \times n$ et $C = AB$.

$$c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}$$

n^3 multiplications, $n^2(n-1)$ additions.

Multiplication de matrices de taille 2 : algorithme de Strassen

$$\begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix}.$$

On calcule les 7 produits :

$p_1 = (a_{11} + a_{22})(b_{11} + b_{22})$
$p_2 = (a_{21} + a_{22})b_{11}$
$p_3 = a_{11}(b_{12} - b_{22})$
$p_4 = a_{22}(-b_{11} + b_{21})$
$p_5 = (a_{11} + a_{12})b_{22}$
$p_6 = (-a_{11} + a_{21})(b_{11} + b_{12})$
$p_7 = (a_{12} - a_{22})(b_{21} + b_{22})$.

$$\begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{pmatrix} = \begin{pmatrix} p_1 + p_4 - p_5 + p_7 & p_3 + p_5 \\ p_2 + p_4 & p_1 + p_3 - p_2 + p_6 \end{pmatrix}$$

Multiplication de matrices de taille 2, suite

Vérification :

$$c_{12} = p_3 + p_5 = a_{11}(b_{12} - b_{22}) + (a_{11} + a_{12})b_{22} = a_{11}b_{12} + a_{12}b_{22}.$$

Bilan : 7 multiplications, 18 additions
(8 et 4 pour la méthode usuelle).
Intérêt ?
Les entrées de A et B peuvent appartenir à un anneau,
même non commutatif, par exemple un anneau de matrices !
Dans ce cas, une multiplication coûte plus cher que 14 additions.

Remarques :

- Comme l'algorithme usuel, cet algorithme est bilinéaire
  (on ne multiplie les entrées d'une matrice
  qu'avec les entrées de l'autre matrice).
- Contrairement à l'algorithme usuel, il y a des annulations.

Multiplication de matrices de taille 2, suite

Vérification :

$$c_{12} = p_3 + p_5 = a_{11}(b_{12} - b_{22}) + (a_{11} + a_{12})b_{22} = a_{11}b_{12} + a_{12}b_{22}.$$

Bilan : 7 multiplications, 18 additions
(8 et 4 pour la méthode usuelle).
Intérêt ?
Les entrées de A et B peuvent appartenir à un anneau,
même non commutatif, par exemple un anneau de matrices !
Dans ce cas, une multiplication coûte plus cher que 14 additions.

Remarques :

- Comme l'algorithme usuel, cet algorithme est bilinéaire
  (on ne multiplie les entrées d'une matrice
  qu'avec les entrées de l'autre matrice).
- Contrairement à l'algorithme usuel, il y a des annulations.

Multiplication de matrices de taille 2, suite

Vérification :

$$c_{12} = p_3 + p_5 = a_{11}(b_{12} - b_{22}) + (a_{11} + a_{12})b_{22} = a_{11}b_{12} + a_{12}b_{22}.$$

Bilan : 7 multiplications, 18 additions
(8 et 4 pour la méthode usuelle).
Intérêt ?
Les entrées de A et B peuvent appartenir à un anneau,
même non commutatif, par exemple un anneau de matrices !
Dans ce cas, une multiplication coûte plus cher que 14 additions.

Remarques :

- Comme l'algorithme usuel, cet algorithme est bilinéaire
  (on ne multiplie les entrées d'une matrice
  qu'avec les entrées de l'autre matrice).
- Contrairement à l'algorithme usuel, il y a des annulations.

Multiplication de matrices de taille n : algorithme de Strassen

$$\begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix}.$$

Les matrices A_{ij}, B_{ij}, C_{ij} sont de taille $n/2$, a coefficients dans R.
Soit $T(n) =$ nombre d'opérations arithmétiques dans R.

$$T(n) = 18(n/2)^2 + 7T(n/2)$$

On trouve $T(n) = O(n^{\log_2 7}) \sim n^{2,807}$.

Après Strassen

Soit ω le meilleur exposant possible.
$\omega \leq 2,807$ (Strassen, 1968)
2,781 (Pan, 1978)
2,780 (Bini et al., 1979)
2,548 (Schönhage, 1979)
2,522 (Pan, 1979)
2,498 (Coppersmith - Winograd, 1980)
2,479 (Strassen, 1986)
2,375 (Coppersmith-Winograd, 1986)

Après Strassen

Soit ω le meilleur exposant possible.
$\omega \leq 2,807$ (Strassen, 1968)
2,781 (Pan, 1978)
2,780 (Bini et al., 1979)
2,548 (Schönhage, 1979)
2,522 (Pan, 1979)
2,498 (Coppersmith - Winograd, 1980)
2,479 (Strassen, 1986)
2,375 (Coppersmith-Winograd, 1986)

2,3736 (Stothers, 2010)
2,3727 (Williams, 2011)

$\omega = 2$ n'est pas impossible.

Quelques bornes inférieures

On ne sait pas montrer que $\omega > 2$ mais...

On sait que le nombre de multiplications nécessaire est au moins :

- $2n^2 - 1$ (Lafon-Winograd, 1978)
- $2n^2 + n - 3$ (Bläser, 1999)

Avec un **algorithme bilinéaire** il en faut au moins :

- $2n^2 - 1$ (Brockett-Dobkin, 1978)
- $\frac{5}{2}n^2 - 3n$ (Bläser, 1999)
- $3n^2 - 4n^{3/2} + n$ (Landsberg, 2012)

Sans soustractions, pas d'annulations :

- Il faut n^3 multiplications (Paterson, 1975).

Si on ne multiplie que par des constantes ≤ 1 en valeur absolue :

- Il faut $\Omega(n^2 \log n)$ opérations arithmétiques (Raz, 2002)

Remarque : Pour montrer de telles bornes inférieures,
il faut se donner un **modèle de calcul** bien défini.

Quelques bornes inférieures

On ne sait pas montrer que $\omega > 2$ mais...
On sait que le nombre de multiplications nécessaire est au moins :

- $2n^2 - 1$ (Lafon-Winograd, 1978)
- $2n^2 + n - 3$ (Bläser, 1999)

Avec un **algorithme bilinéaire** il en faut au moins :

- $2n^2 - 1$ (Brockett-Dobkin, 1978)
- $\frac{5}{2}n^2 - 3n$ (Bläser, 1999)
- $3n^2 - 4n^{3/2} + n$ (Landsberg, 2012)

Sans soustractions, pas d'annulations :

- Il faut n^3 multiplications (Paterson, 1975).

Si on ne multiplie que par des constantes ≤ 1 en valeur absolue :

- Il faut $\Omega(n^2 \log n)$ opérations arithmétiques (Raz, 2002)

Remarque : Pour montrer de telles bornes inférieures,
il faut se donner un **modèle de calcul** bien défini.

Quelques bornes inférieures

On ne sait pas montrer que $\omega > 2$ mais...
On sait que le nombre de multiplications nécessaire est au moins :

- $2n^2 - 1$ (Lafon-Winograd, 1978)
- $2n^2 + n - 3$ (Bläser, 1999)

Avec un **algorithme bilinéaire** il en faut au moins :

- $2n^2 - 1$ (Brockett-Dobkin, 1978)
- $\frac{5}{2}n^2 - 3n$ (Bläser, 1999)
- $3n^2 - 4n^{3/2} + n$ (Landsberg, 2012)

Sans soustractions, pas d'annulations :

- Il faut n^3 multiplications (Paterson, 1975).

Si on ne multiplie que par des constantes ≤ 1 en valeur absolue :

- Il faut $\Omega(n^2 \log n)$ opérations arithmétiques (Raz, 2002)

Remarque : Pour montrer de telles bornes inférieures,
il faut se donner un **modèle de calcul** bien défini.

Quelques bornes inférieures

On ne sait pas montrer que $\omega > 2$ mais...
On sait que le nombre de multiplications nécessaire est au moins :

- $2n^2 - 1$ (Lafon-Winograd, 1978)
- $2n^2 + n - 3$ (Bläser, 1999)

Avec un **algorithme bilinéaire** il en faut au moins :

- $2n^2 - 1$ (Brockett-Dobkin, 1978)
- $\frac{5}{2}n^2 - 3n$ (Bläser, 1999)
- $3n^2 - 4n^{3/2} + n$ (Landsberg, 2012)

Sans soustractions, pas d'annulations :

- Il faut n^3 multiplications (Paterson, 1975).

Si on ne multiplie que par des constantes ≤ 1 en valeur absolue :

- Il faut $\Omega(n^2 \log n)$ opérations arithmétiques (Raz, 2002)

Remarque : Pour montrer de telles bornes inférieures,
il faut se donner un **modèle de calcul** bien défini.

Quelques bornes inférieures

On ne sait pas montrer que $\omega > 2$ mais...
On sait que le nombre de multiplications nécessaire est au moins :

- $2n^2 - 1$ (Lafon-Winograd, 1978)
- $2n^2 + n - 3$ (Bläser, 1999)

Avec un **algorithme bilinéaire** il en faut au moins :

- $2n^2 - 1$ (Brockett-Dobkin, 1978)
- $\frac{5}{2}n^2 - 3n$ (Bläser, 1999)
- $3n^2 - 4n^{3/2} + n$ (Landsberg, 2012)

Sans soustractions, pas d'annulations :

- Il faut n^3 multiplications (Paterson, 1975).

Si on ne multiplie que par des constantes ≤ 1 en valeur absolue :

- Il faut $\Omega(n^2 \log n)$ opérations arithmétiques (Raz, 2002)

Remarque : Pour montrer de telles bornes inférieures,
il faut se donner un **modèle de calcul** bien défini.

Quelques bornes inférieures

On ne sait pas montrer que $\omega > 2$ mais...
On sait que le nombre de multiplications nécessaire est au moins :

- $2n^2 - 1$ (Lafon-Winograd, 1978)
- $2n^2 + n - 3$ (Bläser, 1999)

Avec un **algorithme bilinéaire** il en faut au moins :

- $2n^2 - 1$ (Brockett-Dobkin, 1978)
- $\frac{5}{2}n^2 - 3n$ (Bläser, 1999)
- $3n^2 - 4n^{3/2} + n$ (Landsberg, 2012)

Sans soustractions, pas d'annulations :

- Il faut n^3 multiplications (Paterson, 1975).

Si on ne multiplie que par des constantes ≤ 1 en valeur absolue :

- Il faut $\Omega(n^2 \log n)$ opérations arithmétiques (Raz, 2002)

Remarque : Pour montrer de telles bornes inférieures,
il faut se donner un **modèle de calcul** bien défini.

De très nombreux modèles de calcul...

pour étudier les nombreuses facettes du calcul.
Exemples :
Machines de Turing, circuits booléens, modèles quantiques,
circuits arithmétiques.

- Les entrées sont des variables $X_1, \ldots, X_n$,
  ou des constantes d'un corps K.
- A l'étape i on calcule $R_i = P_i + Q_i$ ou $R_i = P_i \times Q_i$
  avec P_i, Q_i : entrées ou résultats déjà calculés.

Un circuit calcule un ou plusieurs polynômes de $K[X_1, \ldots, X_n]$.

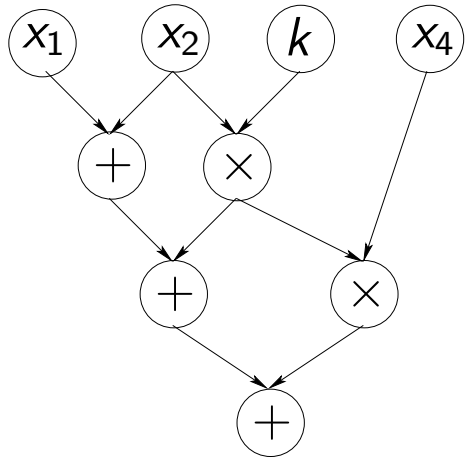
De très nombreux modèles de calcul...

pour étudier les nombreuses facettes du calcul.
Exemples :
Machines de Turing, circuits booléens, modèles quantiques,
circuits arithmétiques.

- Les entrées sont des variables $X_1, \ldots, X_n$,
  ou des constantes d'un corps K.
- A l'étape i on calcule $R_i = P_i + Q_i$ ou $R_i = P_i \times Q_i$
  avec P_i, Q_i : entrées ou résultats déjà calculés.

Un circuit calcule un ou plusieurs polynômes de $K[X_1, \ldots, X_n]$.

Représentation graphique d'un circuit arithmétique



Circuit

Taille : 9

Profondeur : 3

Ce modèle permet d'exprimer de nombreux algorithmes naturels, et de prouver des bornes inférieures.

Importance de la multiplication de matrices

- Un problème de base de l'algèbre linéaire.
- L'exposant optimal ω est le même pour de nombreux problèmes (inversion, déterminant, décomposition $LU\ldots$).

Une question plus simple :
calcul du déterminant par des circuits de taille polynomiale ?

Elimination de Gauss

$O(n^3)$ opération, y compris des divisions !
On obtient :

$$\det A = \frac{P(a_{ij})}{Q(a_{ij})}$$

avec P, Q des polynômes calculés par des circuits arithmétiques de taille $O(n^3)$.

Elimination des divisions (Strassen) :
pour diviser par $1 - R$ on remarque que

$$\frac{1}{1-R} = \sum_{k=0}^{+\infty} R^k.$$

Pour R sans terme constant on peut tronquer au degré du résultat (n pour $\det A$).

Elimination de Gauss

$O(n^3)$ opération, y compris des divisions !
On obtient :

$$\det A = \frac{P(a_{ij})}{Q(a_{ij})}$$

avec P, Q des polynômes calculés par des circuits arithmétiques de taille $O(n^3)$.
Elimination des divisions (Strassen) :
pour diviser par $1 - R$ on remarque que

$$\frac{1}{1-R} = \sum_{k=0}^{+\infty} R^k.$$

Pour R sans terme constant on peut tronquer au degré du résultat (n pour $\det A$).

Elimination de Gauss

$O(n^3)$ opération, y compris des divisions !
On obtient :

$$\det A = \frac{P(a_{ij})}{Q(a_{ij})}$$

avec P, Q des polynômes calculés par des circuits arithmétiques de taille $O(n^3)$.
Elimination des divisions (Strassen) :
pour diviser par $1 - R$ on remarque que

$$\frac{1}{1-R} = \sum_{k=0}^{+\infty} R^k.$$

Pour R sans terme constant on peut tronquer au degré du résultat (n pour $\det A$).

Détection des triangles dans un graphe

Dans un graphe $G = (V, E)$, 3 sommets distincts $\{i, j, k\} \subseteq V$ forment un triangle si $\{ij, jk, ki\} \subseteq E$.

Problème : G contient-il un triangle ?

Algorithme naif : tester les $\binom{|V|}{3}$ triplets de sommets.
Peut-on faire mieux ?

Le retour de la multiplication

Matrice d'adjacence de G :
Pour chaque couple de sommets (i, j),
$A_{ij} = 1$ si $ij \in E$, $A_{ij} = 0$ sinon.
C'est une matrice symétrique.

$(A^2)_{ij} = \sum_{k=1}^{n} A_{ik} A_{kj} \geq 1$ ssi il existe k tel que $ik \in E$ et $kj \in E$.

Algorithme :

1. Calculer A^2 par multiplication rapide.
2. Pour chaque arête $ij \in E$: vérifier si $(A^2)_{ij} \geq 1$;
   si oui, il y a un triangle.

Le retour de la multiplication

Matrice d'adjacence de G :
Pour chaque couple de sommets (i, j),
$A_{ij} = 1$ si $ij \in E$, $A_{ij} = 0$ sinon.
C'est une matrice symétrique.

$(A^2)_{ij} = \sum_{k=1}^{n} A_{ik} A_{kj} \geq 1$ ssi il existe k tel que $ik \in E$ et $kj \in E$.

Algorithme :

1. Calculer A^2 par multiplication rapide.
2. Pour chaque arête $ij \in E$: vérifier si $(A^2)_{ij} \geq 1$;
   si oui, il y a un triangle.

Borne inférieure pour la détection de triangle : le modèle approprié ?

Données booléennes (matrice d'adjacence).

1. **Circuits arithmétiques** sur $\mathbb{Z}/2\mathbb{Z}$.
2. **Circuits booléens :**
   les opérations sont $\vee, \wedge, \neg$ au lieu de $+, \times$.

Ces deux modèles sont équivalents :
par exemple, $x \wedge y = xy$, $x \vee y = x + y + xy$, $\neg x = x + 1$.

3. **Circuits booléens monotones :** pas de $\neg$.

Dans ce modèle : borne inférieure en $n^3/(\log n)^6$ (Razborov, 1985).

Borne inférieure pour la détection de triangle : le modèle approprié ?

Données booléennes (matrice d'adjacence).

1. **Circuits arithmétiques** sur $\mathbb{Z}/2\mathbb{Z}$.
2. **Circuits booléens :**
   les opérations sont $\vee, \wedge, \neg$ au lieu de $+, \times$.

Ces deux modèles sont équivalents :
par exemple, $x \wedge y = xy$, $x \vee y = x + y + xy$, $\neg x = x + 1$.

3. **Circuits booléens monotones :** pas de $\neg$.

Dans ce modèle : borne inférieure en $n^3/(\log n)^6$ (Razborov, 1985).

Borne inférieure pour la détection de triangle : le modèle approprié ?

Données booléennes (matrice d'adjacence).

1. **Circuits arithmétiques** sur $\mathbb{Z}/2\mathbb{Z}$.
2. **Circuits booléens :**
   les opérations sont $\vee, \wedge, \neg$ au lieu de $+, \times$.

Ces deux modèles sont équivalents :
par exemple, $x \wedge y = xy$, $x \vee y = x + y + xy$, $\neg x = x + 1$.

3. **Circuits booléens monotones :** pas de $\neg$.

Dans ce modèle : borne inférieure en $n^3/(\log n)^6$ (Razborov, 1985).

Borne inférieure pour la détection de triangle : le modèle approprié ?

Données booléennes (matrice d'adjacence).

1. **Circuits arithmétiques** sur $\mathbb{Z}/2\mathbb{Z}$.
2. **Circuits booléens :**
   les opérations sont $\vee, \wedge, \neg$ au lieu de $+, \times$.

Ces deux modèles sont équivalents :
par exemple, $x \wedge y = xy$, $x \vee y = x + y + xy$, $\neg x = x + 1$.

3. **Circuits booléens monotones :** pas de $\neg$.

Dans ce modèle : borne inférieure en $n^3/(\log n)^6$ (Razborov, 1985).

Puissances de matrices

- $(A^2)_{ij}$: nombre de chemins de longueur 2 de i à j.
- $(A^k)_{ij}$: nombre de chemins de longueur k de i à j.
- Se généralise aux graphes orientés
  (matrice d'adjacence non nécessairement symétrique).
- Se généralise aux graphes avec poids sur les arêtes
  (poids d'un chemin = produit du poids de ses arêtes).

Puissances de matrices

- $(A^2)_{ij}$: nombre de chemins de longueur 2 de i à j.
- $(A^k)_{ij}$: nombre de chemins de longueur k de i à j.
- Se généralise aux graphes orientés (matrice d'adjacence non nécessairement symétrique).
- Se généralise aux graphes avec poids sur les arêtes (poids d'un chemin = produit du poids de ses arêtes).

Puissances de matrices

- $(A^2)_{ij}$: nombre de chemins de longueur 2 de i à j.
- $(A^k)_{ij}$: nombre de chemins de longueur k de i à j.
- Se généralise aux graphes orientés
  (matrice d'adjacence non nécessairement symétrique).
- Se généralise aux graphes avec poids sur les arêtes
  (poids d'un chemin = produit du poids de ses arêtes).

Puissances de matrices

- $(A^2)_{ij}$: nombre de chemins de longueur 2 de i à j.
- $(A^k)_{ij}$: nombre de chemins de longueur k de i à j.
- Se généralise aux graphes orientés
  (matrice d'adjacence non nécessairement symétrique).
- Se généralise aux graphes avec poids sur les arêtes
  (poids d'un chemin = produit du poids de ses arêtes).

Second mystère :
la complexité du calcul du permanent

Un petit cousin du déterminant :

$$\mathrm{per}(X) = \sum_{\sigma \in S_n} \prod_{i=1}^{n} X_{i\sigma(i)}$$

Un autre problème de borne inférieure :

Conjecture de Valiant : Le permanent n'est pas calculable par des circuits arithmétiques de taille polynomiale en n (sauf dans un corps de caractéristique 2).

C'est un analogue algébrique de la conjecture P $\neq$ NP.

Second mystère :
la complexité du calcul du permanent

Un petit cousin du déterminant :

$$\mathrm{per}(X) = \sum_{\sigma \in S_n} \prod_{i=1}^{n} X_{i\sigma(i)}$$

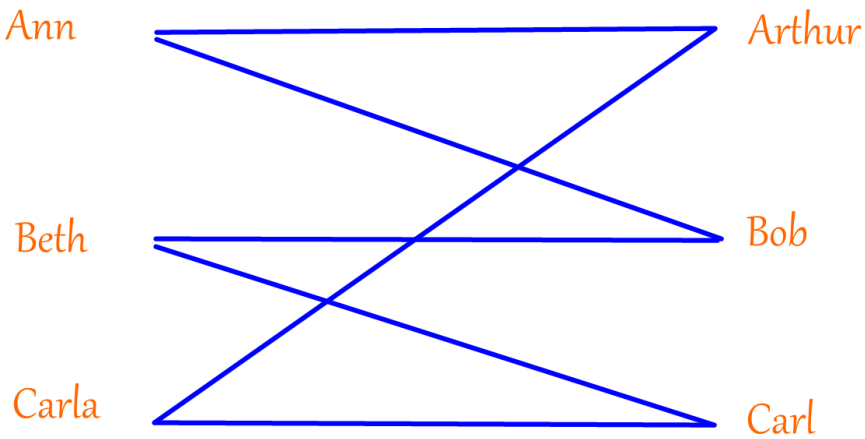
Un autre problème de borne inférieure :

Conjecture de Valiant : Le permanent n'est pas calculable par des circuits arithmétiques de taille polynomiale en n (sauf dans un corps de caractéristique 2).

C'est un analogue algébrique de la conjecture P $\neq$ NP.

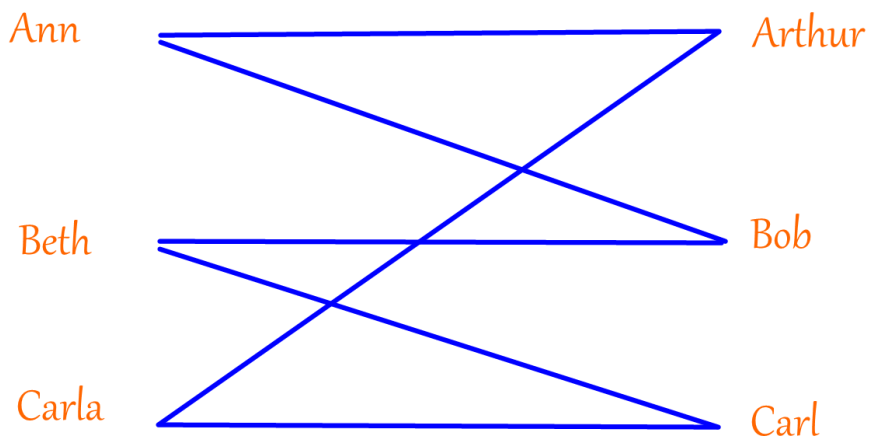
Mariages parfaits

Etant donnés n hommes et n femmes qui peuvent se connaître (ou non),
peut-on marier les n femmes à n hommes qui les connaissent ?



- On peut répondre à cette question en temps polynomial.
- Compter le nombre de solutions est conjecturé difficile (Valiant).

Mariage et permanent

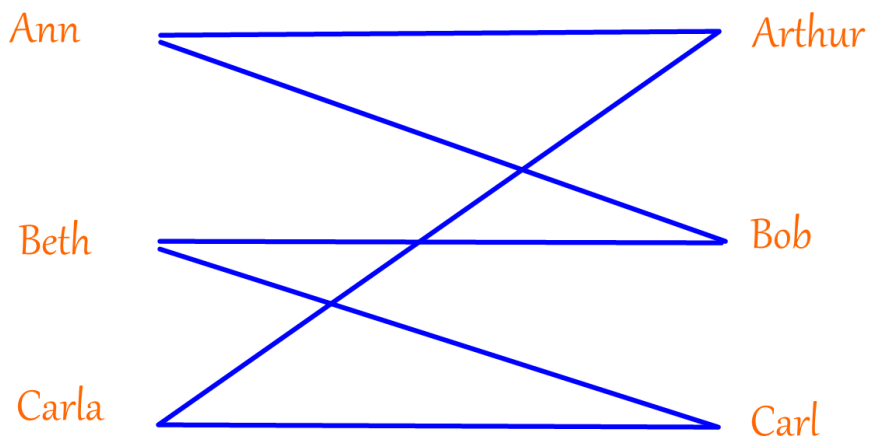


$$\#\text{ mariage parfaits} = 2 = \operatorname{per}\begin{bmatrix} \boxed{1} & 1 & 0 \\ 0 & \boxed{1} & 1 \\ 1 & 0 & \boxed{1} \end{bmatrix} \begin{matrix} \leftarrow \text{Ann} \\ \leftarrow \text{Beth} \\ \leftarrow \text{Carla} \end{matrix}$$

(C'est la matrice d'adjacence bipartie du graphe)

Entrées non booléennes $\Leftrightarrow$ poids sur les arêtes.

Mariage et permanent



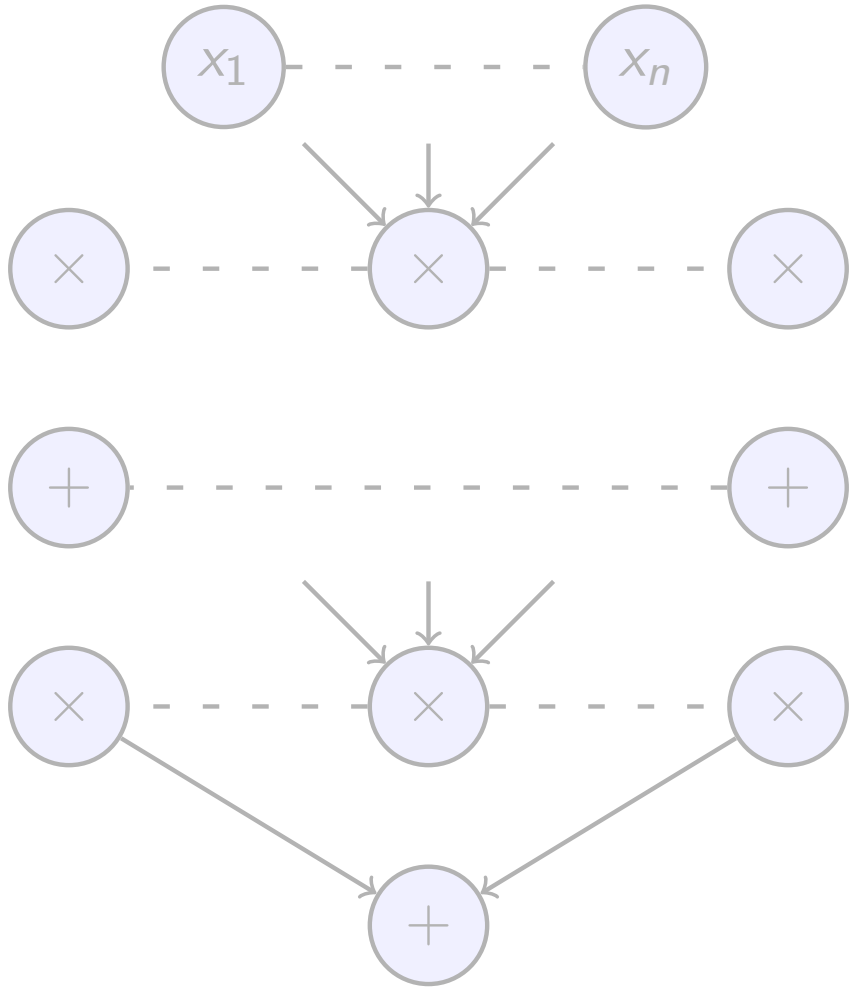
$$\#\text{ mariage parfaits} = 2 = \operatorname{per}\begin{bmatrix} 1 & \boxed{1} & 0 \\ 0 & 1 & \boxed{1} \\ \boxed{1} & 0 & 1 \end{bmatrix} \begin{matrix} \\ \leftarrow \text{Ann} \\ \leftarrow \text{Beth} \\ \leftarrow \text{Carla} \end{matrix}$$

(C'est la matrice d'adjacence bipartie du graphe)

Entrées non booléennes $\Leftrightarrow$ poids sur les arêtes.

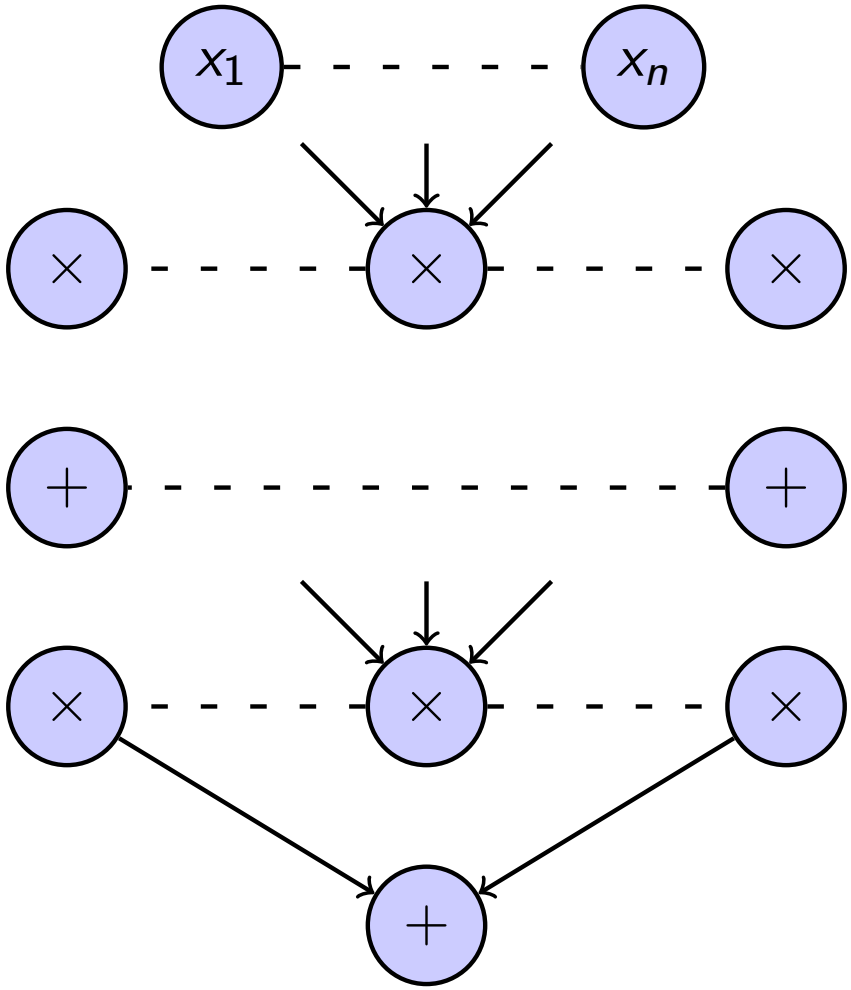
Réduction de profondeur pour les circuits arithmétiques

- Un sujet d'actualité dans les années 1970/80.
- Réduction à la profondeur 4 : circuits de la forme $\Sigma\Pi\Sigma\Pi$.
  Le degré entrant des portes $+$ et $\times$ est non borné.



Réduction de profondeur pour les circuits arithmétiques

- Un sujet d'actualité dans les années 1970/80.
- Réduction à la profondeur 4 : circuits de la forme $\Sigma\Pi\Sigma\Pi$.
  Le degré entrant des portes $+$ et $\times$ est non borné.



Réduction à la profondeur 4

Théorème (Agrawal-Vinay 2008, cas particulier) :
Un polynôme multilinéaire en m variables
avec un circuit de taille $2^{o(m)}$
est calculable par un circuit de profondeur 4 et de taille $2^{o(m)}$.

- Application potentielle (directe ou indirecte)
  aux bornes inférieures.
- Réduction à la profondeur 3 ($\Sigma\Pi\Sigma$) :
  Gupta-Kamath-Kayal-Saptharishi 2013.

Puissance de matrice en profondeur 4

Pour calculer $A^d = (A^{\sqrt{d}})^{\sqrt{d}}$:

1. Calculer $B = A^\delta$ avec $\delta = \sqrt{d}$. On a :

$$B_{ij} = \sum_{1 \leq i_2, i_3, \ldots, i_\delta \leq n} A_{i i_2} A_{i_2 i_3} \cdots A_{i_{\delta-1} i_\delta} A_{i_\delta j}.$$

2. Calculer B^δ.

Coût de l'algorithme :

1. Chaque B_{ij} : $n^{\delta-1}$ multiplications d'arités δ, une addition d'arité $n^{\delta-1}$. Profondeur 2.
2. Seconde étape : même chose.

Au total : $O(n^{\sqrt{d}+1})$ opérations arithmétiques, profondeur 4.

Puissance de matrice en profondeur 4

Pour calculer $A^d = (A^{\sqrt{d}})^{\sqrt{d}}$:

1. Calculer $B = A^\delta$ avec $\delta = \sqrt{d}$. On a :

$$B_{ij} = \sum_{1 \leq i_2, i_3, \ldots, i_\delta \leq n} A_{i i_2} A_{i_2 i_3} \cdots A_{i_{\delta-1} i_\delta} A_{i_\delta j}.$$

2. Calculer B^δ.

Coût de l'algorithme :

1. Chaque B_{ij} : $n^{\delta-1}$ multiplications d'arités δ, une addition d'arité $n^{\delta-1}$. Profondeur 2.
2. Seconde étape : même chose.

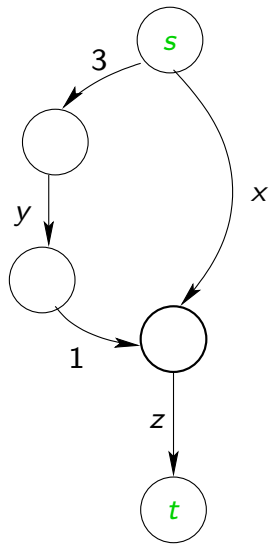
Au total : $O(n^{\sqrt{d}+1})$ opérations arithmétiques, profondeur 4.

Un autre modèle de calcul : les programmes à branchement

Graphe orienté sans cycle avec 2 sommets distingués s, t.

- Arêtes étiquetées par des variables ou des constantes.
- Poids d'un chemin = produit des poids de ses arêtes.
- sortie = somme des poids de tous les chemins de s à t.

Une représentation du polynôme $z(x + 3y)$:



Puissances de matrices et programmes à branchement

Soit G un programme à branchement, A sa matrice d'adjacence.
Supposons que tous les chemins de s à t sont de longueur ℓ.
Le polynôme représenté est $(A^\ell)_{st}$.

Théorème :
Soit G un programme à branchement de taille m et de profondeur ℓ.
Il existe un circuit de profondeur 4 équivalent à G,
avec $m^2 + 1$ portes d'addition et $m^{O(\sqrt{\ell})}$ portes de multiplication.

Puissances de matrices et programmes à branchement

Soit G un programme à branchement, A sa matrice d'adjacence.
Supposons que tous les chemins de s à t sont de longueur ℓ.
Le polynôme représenté est $(A^\ell)_{st}$.

Théorème :
Soit G un programme à branchement de taille m et de profondeur ℓ.
Il existe un circuit de profondeur 4 équivalent à G,
avec $m^2 + 1$ portes d'addition et $m^{O(\sqrt{\ell})}$ portes de multiplication.

Réduction à la profondeur 4 pour les circuits arithmétiques

Théorème :
Soit C un circuit arithmétique de taille t et degré d.
Il existe un circuit de profondeur 4 équivalent de taille $t^{O(\sqrt{d}\log d)}$.

Principe de la preuve :
$C \rightarrow$ programme à branchement $\rightarrow$ circuit de profondeur 4.

Une amélioration, et une borne inférieure

Théorème (Tavenas 2013) :
Soit C un circuit arithmétique de taille t et degré d.
Il existe un circuit de profondeur 4 équivalent de taille $t^{O(\sqrt{d})}$.

Corollaire :
Si le permanent admet des circuits de taille polynomiale en n,
il admet des circuits de profondeur 4 de taille $n^{O(\sqrt{n})}$.

Théorème (Gupta-Kamath-Kayal-Saptharishi 2013) :
Tout circuit$^{(*)}$ de profondeur 4 pour le permanent
est de taille au moins $2^{\Omega(\sqrt{n})}$.

(*) Il faut supposer que le circuit est homogène et que les multiplications entre entrées sont d'arité $O(\sqrt{n})$.

Une amélioration, et une borne inférieure

Théorème (Tavenas 2013) :
Soit C un circuit arithmétique de taille t et degré d.
Il existe un circuit de profondeur 4 équivalent de taille $t^{O(\sqrt{d})}$.

Corollaire :
Si le permanent admet des circuits de taille polynomiale en n,
il admet des circuits de profondeur 4 de taille $n^{O(\sqrt{n})}$.

Théorème (Gupta-Kamath-Kayal-Saptharishi 2013) :
Tout circuit$^{(*)}$ de profondeur 4 pour le permanent
est de taille au moins $2^{\Omega(\sqrt{n})}$.

(*) Il faut supposer que le circuit est homogène et que les multiplications entre entrées sont d'arité $O(\sqrt{n})$.

Une amélioration, et une borne inférieure

Théorème (Tavenas 2013) :
Soit C un circuit arithmétique de taille t et degré d.
Il existe un circuit de profondeur 4 équivalent de taille $t^{O(\sqrt{d})}$.

Corollaire :
Si le permanent admet des circuits de taille polynomiale en n,
il admet des circuits de profondeur 4 de taille $n^{O(\sqrt{n})}$.

Théorème (Gupta-Kamath-Kayal-Saptharishi 2013) :
Tout circuit$^{(*)}$ de profondeur 4 pour le permanent
est de taille au moins $2^{\Omega(\sqrt{n})}$.

(*) Il faut supposer que le circuit est homogène et que les multiplications entre entrées sont d'arité $O(\sqrt{n})$.

Bornes inférieures et racines réelles : une approche indirecte

Principe : obtenir des bornes inférieures à partir de généralisations (conjecturées) de la règle des signes.

Règle des signes (Descartes, 1596-1650) :

- Soit $P(X) = \sum_{i=0}^{d} a_i X^i \in \mathbb{R}[X]$.
- Soit $Z^+(P)$ le nombre de racines strictement positives.
- Soit $V(P)$ le nombre de changements de signes dans la suite $[a_0, a_1, \ldots, a_d]$.

Alors $Z^+(P) \leq V(P)$.

La règle de Descartes sans les signes

Un résultat sur les polynômes creux.

Théorème :

Si $f \in \mathbb{R}[X]$ a t monômes non nuls, f a au plus $t-1$ racines strictement positives.

Preuve : Récurrence sur t. Pas de racine > 0 pour $t = 1$.

Pour $t > 1$: Soit $a_\alpha X^\alpha$ le monôme de plus petit degré.

On peut supposer que $\alpha = 0$ (sinon, diviser par X^α). Du coup :

(i) f' a $t-1$ monômes $\Rightarrow \leq t-2$ racines strictement positives.

(ii) Entre 2 racines strictement positives et consécutives de f : il existe une racine strictement positive de f' (théorème de Rolle).

Corollaire : f a au plus $2t-1$ racines réelles.

Cette borne est atteinte.

La règle de Descartes sans les signes

Un résultat sur les polynômes creux.

Théorème :

Si $f \in \mathbb{R}[X]$ a t monômes non nuls, f a au plus $t-1$ racines strictement positives.

Preuve : Récurrence sur t. Pas de racine > 0 pour $t = 1$.

Pour $t > 1$: Soit $a_\alpha X^\alpha$ le monôme de plus petit degré.

On peut supposer que $\alpha = 0$ (sinon, diviser par X^α). Du coup :

(i) f' a $t-1$ monômes $\Rightarrow \leq t-2$ racines strictement positives.

(ii) Entre 2 racines strictement positives et consécutives de f : il existe une racine strictement positive de f' (théorème de Rolle).

Corollaire : f a au plus $2t-1$ racines réelles.

Cette borne est atteinte.

La règle de Descartes sans les signes

Un résultat sur les polynômes creux.

Théorème :

Si $f \in \mathbb{R}[X]$ a t monômes non nuls, f a au plus $t-1$ racines strictement positives.

Preuve : Récurrence sur t. Pas de racine > 0 pour $t = 1$.

Pour $t > 1$: Soit $a_\alpha X^\alpha$ le monôme de plus petit degré.

On peut supposer que $\alpha = 0$ (sinon, diviser par X^α). Du coup :

(i) f' a $t-1$ monômes $\Rightarrow \leq t-2$ racines strictement positives.

(ii) Entre 2 racines strictement positives et consécutives de f : il existe une racine strictement positive de f' (théorème de Rolle).

Corollaire : f a au plus $2t-1$ racines réelles.

Cette borne est atteinte.

Des généralisations souhaitables

1. Si f et g ont t monômes :
   nombres maximum de solutions réelles pour $f(x)g(x) = 1$?
   Remarque : $fg - 1$ a au plus $t^2 + 1$ monômes.
   Le nombre maximum de solutions est il quadratique en t ?
   linéaire ? intermédiaire ?
2. Idem pour $f_1 f_2 \cdots f_m = 1$.
3. **τ-conjecture réelle**
   (sur les sommes de produits de polynômes creux) :
   Soit $f(x) = \sum_{i=1}^{k} \prod_{j=1}^{m} f_{ij}(x)$.
   Le nombre de racines réelles de f est polynomial en kmt.

Des généralisations souhaitables

1. Si f et g ont t monômes :
   nombres maximum de solutions réelles pour $f(x)g(x) = 1$?
   Remarque : $fg - 1$ a au plus $t^2 + 1$ monômes.
   Le nombre maximum de solutions est il quadratique en t ?
   linéaire ? intermédiaire ?
2. Idem pour $f_1 f_2 \cdots f_m = 1$.
3. **τ-conjecture réelle**
   (sur les sommes de produits de polynômes creux) :
   Soit $f(x) = \sum_{i=1}^{k} \prod_{j=1}^{m} f_{ij}(x)$.
   Le nombre de racines réelles de f est polynomial en kmt.

Des généralisations souhaitables

1. Si f et g ont t monômes :
   nombres maximum de solutions réelles pour $f(x)g(x) = 1$?
   Remarque : $fg - 1$ a au plus $t^2 + 1$ monômes.
   Le nombre maximum de solutions est il quadratique en t ?
   linéaire ? intermédiaire ?
2. Idem pour $f_1 f_2 \cdots f_m = 1$.
3. **τ-conjecture réelle**
   (sur les sommes de produits de polynômes creux) :
   Soit $f(x) = \sum_{i=1}^{k} \prod_{j=1}^{m} f_{ij}(x)$.
   Le nombre de racines réelles de f est polynomial en kmt.

Permanent et racines réelles

Théorème : Si la τ-conjecture réelle est vraie,
pas de circuits de taille polynomiale pour le permanent.

Un ingrédient : réduction à la profondeur 4.

Circuit de profondeur 4 $(\sum\prod\sum\prod)$,
avec des entrées constantes ou de la forme X^{2^i}

$\Updownarrow$

Somme de produits de polynômes creux

Permanent et racines réelles

Théorème : Si la τ-conjecture réelle est vraie,
pas de circuits de taille polynomiale pour le permanent.

Un ingrédient : réduction à la profondeur 4.

Circuit de profondeur 4 $(\sum\prod\sum\prod)$,
avec des entrées constantes ou de la forme X^{2^i}

$\Updownarrow$

Somme de produits de polynômes creux

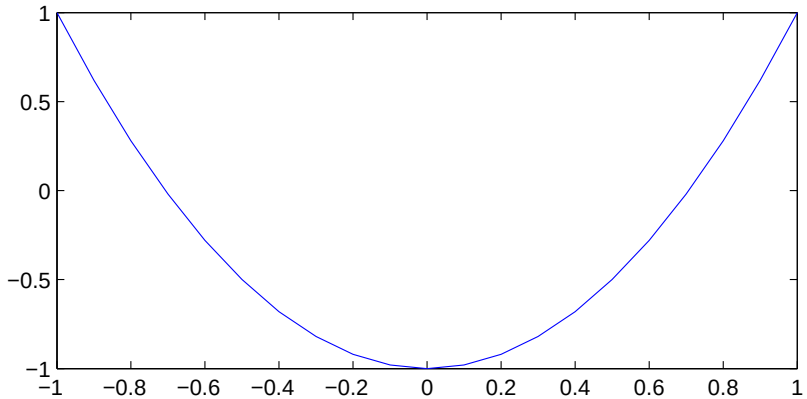
Petit circuit, beaucoup de racines réelles

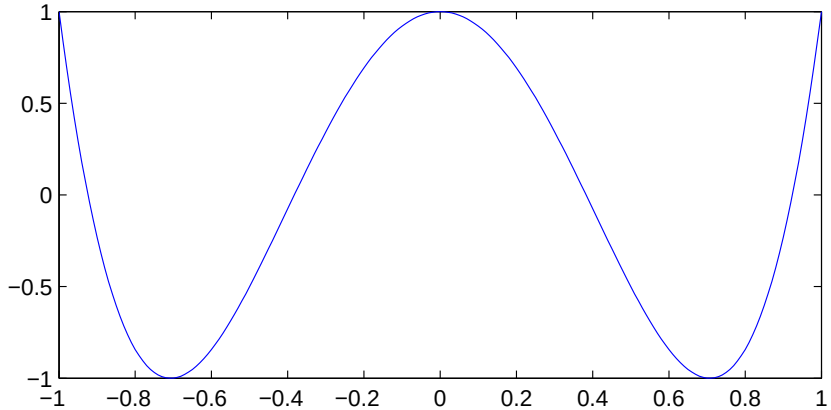
- Soit T_n le polynôme de Tchebychev d'ordre n :

$$\cos(n\theta) = T_n(\cos\theta).$$

  Par exemple : $T_1(x) = x$, $T_2(x) = 2x^2 - 1$,
  $T_4(x) = 2(2x^2 - 1)^2 - 1$.
- T_n est un polynôme de degré n avec n racines dans $[-1, 1]$.
- $T_{2^n}(x) = T_2(T_2(\cdots T_2(T_2(x))\cdots))$: T_2 itéré n fois.
  Conséquence : circuit de taille $O(n)$ pour T_{2^n}.

Graphes de T_2 et T_4 :





Un devoir à la maison

Si f et g ont t monômes :
nombres maximum de solutions réelles de l'équation $f(x)g(x) = 1$?