Un résultat de transfert entre modèles de Valiant et BSS

Pascal Koiran Sylvain Perifel

Le 24 mars 2005

Introduction

— Calcul de suites de polynômes.

— Classes de Valiant : VP et VNP.

— Question ouverte « VP = VNP ? ».

— Plus simple : résultats de transfert
(de la forme « si VP = VNP alors P = NP »).

Plan

1. Complexité, Valiant
2. Séparation d'hypersurfaces
3. Tester l'appartenance à une variété complexe
4. Résultat de transfert

Modèle BSS

- Circuits arithmétiques avec tests sur $(\mathbf{C}, +, \times, =)$:
  entrées x_i, paramètres $\in \mathbf{C}$
  portes $+$, $\times$, tests « $= 0$ » (réponse 0 ou 1).
- Une famille de circuits (C_n) reconnaît un langage $X \subseteq \cup_n \mathbf{C}^n$:
  si $|x| = n$, alors [$x \in X$ ssi $C_n(x) = 1$].
- $\mathrm{P}_{\mathbf{C}}$: $|C_n| \leq p(n)$
  $\mathrm{PAR}_{\mathbf{C}}$: profondeur polynomiale (+ uniformité).
- Remarque : $\mathrm{NP}_{\mathbf{C}} \subseteq \mathrm{PAR}_{\mathbf{C}}$.

Modèle de Valiant

— Circuits arithmétiques *sans* test sur $(\mathbf{C}, +, \times, =)$:
variables x_i, constantes $\in \{-1, 0, 1\}$
portes $+$, $\times$ $\longrightarrow$ polynômes $\in \mathbf{Z}[x_1, \ldots, x_n]$

— Une famille de circuits (C_n) calcule une *famille de polynômes.*

— VP : familles de polynômes calculables par circuits de taille polynomiale
VPSPACE : fonction coefficient PSPACE $(f_n(\bar{x}) = \sum_{\bar{\epsilon}} a(n, \bar{\epsilon})\bar{x}^{\bar{\epsilon}})$
($\equiv$ circuits profondeur polynomiale + uniformité).

— Résultat :

$$\mathrm{VP} = \mathrm{VPSPACE} \Longrightarrow \mathrm{P}_{\mathbf{C}} = \mathrm{PAR}_{\mathbf{C}}$$

Séparation d'hypersurfaces : problème

- Polynômes $f_1, \dots, f_s \in \mathbf{Z}[x_1, \dots, x_n]$.
- Relation $x \sim y$ ssi $\forall i (f_i(x) = 0 \Leftrightarrow f_i(y) = 0)$.
- Cellule : classe d'équivalence (appartenir aux mêmes hypersurfaces).
- But : algorithme avec tests VPSPACE, à sorties booléennes, tel que

$$x \not\sim y \Rightarrow C(x) \neq C(y)$$

Séparation d'hypersurfaces : pourquoi ?

- Circuit BSS $\longrightarrow$ famille de tous les polynômes $f_1, \ldots, f_s$ qui peuvent être calculés lors de l'exécution du circuit
- $x \sim y \Rightarrow C(x) = C(y)$ (résultats des tests identiques).
- Il suffit de tester si la cellule de x est dans le langage (un s-uple booléen plutôt qu'un n-uple complexe).

Conditions de signe

- Cellule pour $f_1, \ldots, f_s \longrightarrow$ condition de signe $S \in \{0,1\}^s$ (2^s conditions de signe).
- Condition de signe de x : $S_i = 1$ ssi $f_i(x) = 0$.
- Condition de signe *satisfaisable* : cs de x pour un certain $x \in \mathbf{C}^n$.
- Seulement $(sd)^{O(n)}$ cs satisfaisables + algo PSPACE pour les énumérer : $a(i,j) = i$-ème coordonnée de la j-ème cs satisfaisable.
- But : trouver la condition de signe de l'entrée $x \in \mathbf{C}^n$.

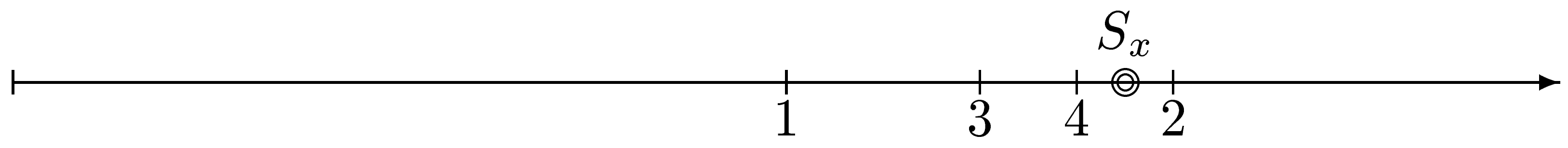
Algorithme pour la cs

— Sur les cs : ordre total compatible avec l'inclusion.

— S *candidat* pour x si $\forall i, S_i = 1 \Rightarrow f_i(x) = 0$.

— S_x = candidat maximal.

— Recherche dichotomique : tests $x \in \cup_{i \geq k} V_i$
où V_i est définie par les f_j tels que $(S_i)_j = 1$.



Appartenance à une variété (1)

- V définie par $f_1, \ldots, f_s \in \mathbf{Z}[x_1, \ldots, x_n]$.
- But : pour $x \in \mathbf{C}^n$, décider si $x \in V$.
- **Lemme 1** :
  $(n+1)$ combinaisons linéaires « génériques » des f_i définissent V.
- **Lemme 2** :
  on peut remplacer les éléments transcendants par des entiers croissants suffisamment vite.
- Problème : les coefficients deviennent trop gros.
  $\longrightarrow$ réduire le nombre de polynômes étape par étape.

Appartenance à une variété (2)

$$\underbrace{f_1, \ldots, f_{2(n+1)}}_{V_1} \qquad \underbrace{f_{2(n+1)+1}, \ldots, f_{4(n+1)}}_{V_2} \qquad \cdots \qquad f_s$$

$(n+1)$ polynômes $\qquad$ $(n+1)$ polynômes

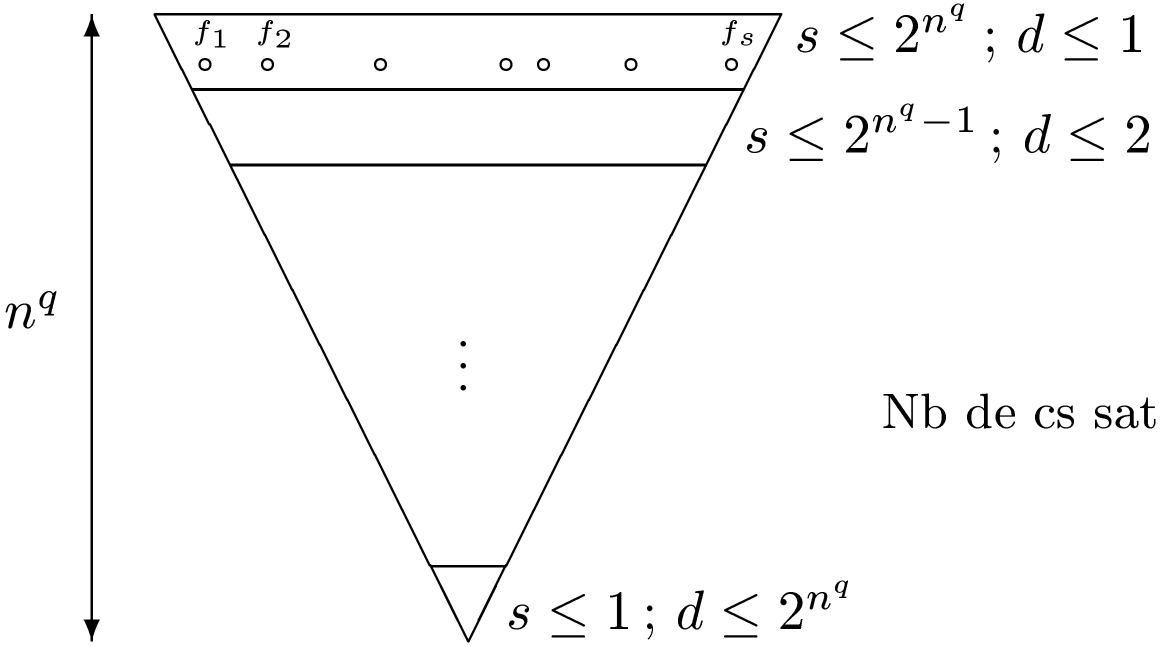
$(n+1)$ polynômes

Le résultat de transfert

Proposition :
Tout $A \in \mathrm{PAR}_{\mathbf{C}}$ est décidé par un algorithme polynomial avec tests VPSPACE.

Preuve : soit $\mathcal{C}$ le circuit BSS décidant $A^{=n}$, de profondeur n^q.

— Énumération de tous les polynômes rencontrés lors de l'exécution de $\mathcal{C}$.



Nb de cs sat. $\leq (sd)^{O(n)}$

Preuve (suite)

- Ainsi, $2^{n^{O(1)}}$ polynômes que l'on peut énumérer dans VPSPACE.
- Algorithme de séparation d'hypersurfaces $\longrightarrow S_x$ (la cs de x).
- On « simule » le circuit $\mathcal{C}$ grâce aux résultats des tests. $\square$

Corollaire :

$$\mathrm{VP} = \mathrm{VPSPACE} \Longrightarrow \mathrm{P}_{\mathbf{C}} = \mathrm{PAR}_{\mathbf{C}}$$

Conclusion

— Résultat clé : peu de cs satisfaisables, on peut les énumérer.

— Peu probable que VP = VPSPACE.

— Et sur $(\mathbf{R}, +, \times, \leq)$?