Valiant's model: from exponential sums to exponential products

Pascal Koiran   Sylvain Perifel

LIP, ENS Lyon

MFCS, August 31, 2006

Introduction

- Computation of sequences of polynomials by families of arithmetic circuits.
- Polynomial-size circuits: Valiant's class VP.
- Exponential sums of VP families: Valiant's class VNP.

Introduction

- Computation of sequences of polynomials by families of arithmetic circuits.
- Polynomial-size circuits: Valiant's class VP.
- Exponential sums of VP families: Valiant's class VNP.
- What about exponential products? $\longrightarrow \mathrm{V\Pi P}$.

Introduction

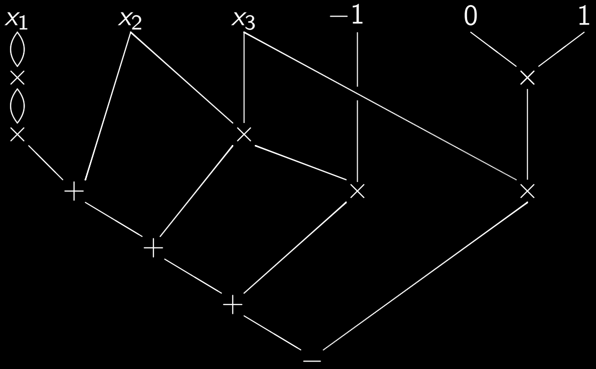
- Computation of sequences of polynomials by families of arithmetic circuits.
- Polynomial-size circuits: Valiant's class VP.
- Exponential sums of VP families: Valiant's class VNP.
- What about exponential products? $\longrightarrow \mathrm{V\Pi P}$.
- What if $\mathrm{V\Pi P}$ has small circuits (i.e. $\mathrm{VP} = \mathrm{V\Pi P}$)?

Outline

1. Arithmetic circuits, Valiant's classes.
2. $\mathrm{V\Pi P}$, definition and first results.
3. Algebraic complexity: BSS classes.
4. Main result:

$$\text{if } \mathrm{VP} = \mathrm{V\Pi P} \text{ then } \mathrm{NP}_{(K,+,-,=)} \text{ has small circuits.}$$

Arithmetic circuits



Variables and constants of K as inputs, $+$, $-$ and $\times$ gates:
a circuit computes a polynomial $f \in K[x_1, \ldots, x_n]$.

Valiant's classes

Definition

- VP: family (f_n) of polynomials computed by a family of polynomial-size arithmetic circuits, and of polynomial degree.

Valiant's classes

Definition

- VP: family (f_n) of polynomials computed by a family of polynomial-size arithmetic circuits, and of polynomial degree.
- VNP: family (g_n) such that there exists $(f_n(\bar{x}, \bar{y})) \in \mathrm{VP}$ satisfying
$$g_n(\bar{x}) = \sum_{\bar{\epsilon}} f_n(\bar{x}, \bar{\epsilon})$$
where the summation is taken over $\bar{\epsilon} \in \{0, 1\}^{p(n)}$.

Example of VNP family:

$$\mathrm{per}_n(x_{1,1}, x_{1,2}, \ldots, x_{n,n}) = \sum_{\sigma \in S_n} \prod_{i=1}^{n} x_{i,\sigma(i)}.$$

Definition

- VP: family (f_n) of polynomials computed by a family of polynomial-size arithmetic circuits, and of polynomial degree.
- VNP: family (g_n) such that there exists $(f_n(\bar{x}, \bar{y})) \in \mathrm{VP}$ satisfying

$$g_n(\bar{x}) = \sum_{\bar{\epsilon}} f_n(\bar{x}, \bar{\epsilon})$$

where the summation is taken over $\bar{\epsilon} \in \{0,1\}^{p(n)}$.

Example of VNP family:

$$\mathrm{per}_n(x_{1,1}, x_{1,2}, \ldots, x_{n,n}) = \sum_{\sigma \in S_n} \prod_{i=1}^{n} x_{i,\sigma(i)}.$$

Guillaume Malod 2003: no bound on the degree.

From now on, VP *designates Malod's version.*

Definition ($\mathrm{V\Pi P}$)

Family (g_n) such that there exists $(f_n(\bar{x}, \bar{y})) \in \mathrm{VP}$ satisfying

$$g_n(\bar{x}) = \prod_{\bar{\epsilon}} f_n(\bar{x}, \bar{\epsilon})$$

where the product is taken over $\bar{\epsilon} \in \{0, 1\}^{p(n)}$.

VΠP

Definition (VΠP)

Family (g_n) such that there exists $(f_n(\bar{x}, \bar{y})) \in \mathrm{VP}$ satisfying

$$g_n(\bar{x}) = \prod_{\bar{\epsilon}} f_n(\bar{x}, \bar{\epsilon})$$

where the product is taken over $\bar{\epsilon} \in \{0, 1\}^{p(n)}$.

Example

$$g_n(X) = \prod_{i=0}^{2^n-1} (X - i)$$

Then $g_n(X) = \prod_{\bar{\epsilon} \in \{0,1\}^n} f_n(X, \bar{\epsilon})$, where

$$f_n(X, \bar{\epsilon}) = X - \sum_{i=1}^{n} \epsilon_i 2^i.$$

Does $\mathrm{V\Pi P}$ equal VP?

Theorem

If $\mathrm{V\Pi P}^0 = \mathrm{VP}^0$ *(constant-free classes)*
then $\mathrm{P/poly} = \mathrm{NP/poly}$.

Does $\mathrm{V\Pi P}$ equal VP?

Theorem

If $\mathrm{V\Pi P}^0 = \mathrm{VP}^0$ *(constant-free classes)*
then $\mathrm{P/poly} = \mathrm{NP/poly}$.

Proof.

Take A in $\mathrm{NP/poly}$: family (C_n) of polynomial-size boolean circuits such that

$$x \in A \Longleftrightarrow \exists y \in \{0,1\}^{p(n)}(C_n(x,y) = 0).$$

Simulate C_n by an arithmetic circuit $D_n \longrightarrow$ family VP.
$x \in A \Longleftrightarrow \prod_y D_n(x,y) = 0 \longrightarrow$ testing a VP^0 family to zero.
Done in BPP (Schwartz 1980),
thus in P/poly (Adleman 1978). □

BSS complexity

- Computation over arbitrary fields K, languages $A \subseteq (\bigcup_n K^n)$.
- Allowed operations: $+$, $-$, $\times$ and equality tests.

BSS complexity

- Computation over arbitrary fields K, languages $A \subseteq (\bigcup_n K^n)$.
- Allowed operations: $+$, $-$, $\times$ and equality tests.
- P_K: languages recognized by a family of polynomial-size algebraic circuits.
- NP_K: existential version, i.e. there exists $B \in \mathrm{P}_K$ such that

$$x \in A \iff \exists y \in K^{p(n)} (x, y) \in B.$$

BSS complexity

- Computation over arbitrary fields K, languages $A \subseteq (\bigcup_n K^n)$.
- Allowed operations: $+$, $-$, $\times$ and equality tests.
- P_K: languages recognized by a family of polynomial-size algebraic circuits.
- NP_K: existential version, i.e. there exists $B \in \mathrm{P}_K$ such that

$$x \in A \iff \exists y \in K^{p(n)} (x, y) \in B.$$

- *Twenty questions* (Shub and Smale): decide whether the input x is in $\{0, 1, \ldots, 2^n - 1\}$. This problem is in $\mathrm{NP}_{(\mathbf{C},+,-,=)}$ but suspected to be outside of $\mathrm{P}_{\mathbf{C}}$.
If $\mathrm{V\Pi P} = \mathrm{VP}$, it is in $\mathrm{P}_{\mathbf{C}}$ by computing $\prod_{i=0}^{2^n-1}(X - i)$.

Transfer theorem

Theorem

Any problem in $\mathrm{NP}_{(K,+,-,=)}$ *is solved by a family of polynomial-size circuits with* $+$, $-$, $\times$, $=$ *and* $\mathrm{V\Pi P}$ *gates.*

Corollary

If $\mathrm{V\Pi P} = \mathrm{VP}$ *then any problem in* $\mathrm{NP}_{(K,+,-,=)}$ *is solved by a family of polynomial-size circuits over the field* K*.*

Transfer theorem

Theorem

Any problem in $\mathrm{NP}_{(K,+,-,=)}$ *is solved by a family of polynomial-size circuits with* $+$, $-$, $\times$, $=$ *and* $\mathrm{V\Pi P}$ *gates.*

Corollary

If $\mathrm{V\Pi P} = \mathrm{VP}$ *then any problem in* $\mathrm{NP}_{(K,+,-,=)}$ *is solved by a family of polynomial-size circuits over the field* K*.*

Proof (of the theorem).

Let $A \in \mathrm{NP}_{(K,+,-,=)}$: there is $B \in \mathrm{P}_{(K,+,-,=)}$ such that

$$x \in A \Longleftrightarrow \exists y \in \{0,1\}^{p(n)}((x,y) \in B) \quad \text{(Koiran 1994).}$$

B is recognized by a family (C_n) of circuits with $+$, $-$ and $=$ gates. Tests made by $C_n(x,y)$ are of the form $\sum \lambda_i x_i = \sum \mu_i y_i + \gamma$. Coefficients $< 2^{poly(n)}$ in absolute value.

Therefore if x and x' belong to exactly the same hyperplanes with polynomial-size coefficients, they are both in A or both outside of A. $\longrightarrow$ Arrangement of hyperplanes.

$$\text{The cell of } x\text{:}\quad F = \Big(\bigcap_{x \in H} H\Big) \setminus \Big(\bigcup_{x \notin H'} H'\Big).$$

Goal: decide whether the cell F of the input x is in A.

Therefore if x and x' belong to exactly the same hyperplanes with polynomial-size coefficients, they are both in A or both outside of A. $\longrightarrow$ Arrangement of hyperplanes.

$$\text{The cell of } x\text{:}\quad F = \Big(\bigcap_{x\in H} H\Big) \setminus \Big(\bigcup_{x\notin H'} H'\Big).$$

Goal: decide whether the cell F of the input x is in A.

First step: Find F.

Algorithm: maintain a search space E containing x.

- $E \leftarrow K^n$.
- Repeat (while H exists):
  - by binary search, find the first hyperplane H such that $x \in H$ and $E \cap H \neq E$ (VΠP test: $\prod_{H/E\not\subseteq H} \varphi_H(x) = 0$?);
  - $E \leftarrow E \cap H$.
- Output E.

Second step: Decide whether $F \subseteq A$ or $F \subseteq K^n \setminus A$.

Algorithm:

- Find a “small” rational point q in F;
- decide whether $q \in A$.

Second step: Decide whether $F \subseteq A$ or $F \subseteq K^n \setminus A$.

Algorithm:

- Find a "small" rational point q in F;
- decide whether $q \in A$.

The first point is easy from the list of hyperplanes defining F.

Second step: Decide whether $F \subseteq A$ or $F \subseteq K^n \setminus A$.
Algorithm:

- Find a "small" rational point q in F;
- decide whether $q \in A$.

The first point is easy from the list of hyperplanes defining F.

The second point is done thanks to a $\mathrm{V\Pi P}$ test. Indeed,

$$q \in A \iff \exists y \in \{0,1\}^{p(n)} (q,y) \in B.$$

$(q,y) \in B$ is decided by boolean circuit C_n. The family (C_n) is simulated by a VP family (g_n), hence:

$$q \in A \iff \prod_{y \in \{0,1\}^{p(n)}} g_n(q,y) = 0. \quad \square$$

Current and future work

- What about the other direction: $\mathrm{P}_K = \mathrm{NP}_K \Rightarrow \mathrm{VP} = \mathrm{V\Pi P}$?

- What about the other direction: $\mathrm{P}_K = \mathrm{NP}_K \Rightarrow \mathrm{VP} = \mathrm{V\Pi P}$?
- One can define a whole hierarchy by alternating $\sum$ and $\prod$, and a class VPSPACE containing it.

$$\begin{tikzcd}
 & \mathsf{V\Sigma P} \arrow[r] \arrow[dr] & \mathsf{V\Pi\Sigma P} \arrow[r] \arrow[dr] & \ldots \arrow[dr] & \\
\mathsf{VP} \arrow[ur] \arrow[dr] & & & & \mathsf{VPSPACE} \\
 & \mathsf{V\Pi P} \arrow[r] \arrow[ur] & \mathsf{V\Sigma\Pi P} \arrow[r] \arrow[ur] & \ldots \arrow[ur] &
\end{tikzcd}$$

- VPSPACE enables to manipulate hypersurfaces instead of hyperplanes, thus taking $\times$ into account:

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

Thank you!

1. Arithmetic circuits, Valiant's classes.
2. $\mathrm{V\Pi P}$, definition and first results.
3. Algebraic complexity: BSS classes.
4. Main result:

$$\text{if } \mathrm{VP} = \mathrm{V\Pi P} \text{ then } \mathrm{NP}_{(K,+,-,=)} \text{ has small circuits.}$$