Interpolation in Valiant's theory

Pascal Koiran   Sylvain Perifel

LIP, ENS Lyon

Paderborn, October 30, 2007

Introduction

Two ways of computing a polynomial with integer coefficients

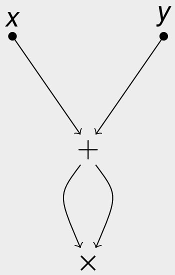
- Algorithm that evaluates the polynomial at an integer point.
  Example: $P(x, y) = (x + y)^2$ on input $(1, 3) \rightarrow 16$.

Introduction

Two ways of computing a polynomial with integer coefficients

- Algorithm that evaluates the polynomial at an integer point.
  Example: $P(x, y) = (x + y)^2$ on input $(1, 3) \to 16$.

- Arithmetic circuit that computes the polynomial.
  Example:



A question of Papadimitriou

Question ♣

If a polynomial P can be evaluated by a polynomial-time algorithm, is it true that it is computable by an arithmetic circuit of polynomial size?

A question of Papadimitriou

Question ♣

If a polynomial P can be evaluated by a polynomial-time algorithm, is it true that it is computable by an arithmetic circuit of polynomial size?

In other words, does the use of boolean operations other than $+$ and $\times$ enable a superpolynomial speed-up in the computation?

A question of Papadimitriou

Question ♣

If a polynomial P can be evaluated by a polynomial-time algorithm, is it true that it is computable by an arithmetic circuit of polynomial size?

In other words, does the use of boolean operations other than $+$ and $\times$ enable a superpolynomial speed-up in the computation?

- The use of families of polynomials makes these questions meaningful.

Divisions

- Strassen: positive answer for divisions if the polynomial has a polynomial degree.
- Idea: replace $\frac{1}{1-x}$ by $1 + x + x^2 + \cdots + x^{p(n)}$.

Divisions

- Strassen: positive answer for divisions if the polynomial has a polynomial degree.
- Idea: replace $\frac{1}{1-x}$ by $1 + x + x^2 + \cdots + x^{p(n)}$.

- What if the degree is not polynomial ?

Discussion

- In order to show that question ♣ has a negative answer, one looks for a polynomial P that can be evaluated in polynomial time but cannot be computed by polynomial-size circuits.

Discussion

- In order to show that question ♣ has a negative answer, one looks for a polynomial P that can be evaluated in polynomial time but cannot be computed by polynomial-size circuits.

  But lack of candidates (usual examples don't work: determinant, permanent, etc.).

Discussion

- In order to show that question ♣ has a negative answer, one looks for a polynomial P that can be evaluated in polynomial time but cannot be computed by polynomial-size circuits.

  But lack of candidates (usual examples don't work: determinant, permanent, etc.).

- In order to show that question ♣ has a positive answer, one wants to transform an evaluation algorithm into an arithmetic circuit.

Main result

If question ♣ has a negative answer, then $\mathrm{VP} \neq \mathrm{VNP}$.

Outline

1. Valiant's classes

2. The counting hierarchy

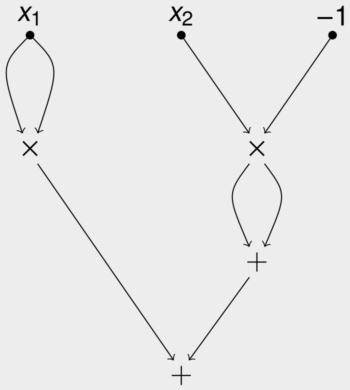
3. Interpolation

4. Consequences

Arithmetic circuits

Arithmetic circuits:

- gates $+$ and $\times$
- inputs $x_1, \ldots, x_n$ and the constant -1
- $\rightarrow$ multivariate polynomials with integer coefficients.



Disclaimer

We will skip the problem of constants and of uniformity...

P and NP in Valiant's model

- Family of polynomials (f_n): one circuit C_n per polynomial $f_n \in \mathbb{Z}[x_1, \ldots, x_{u(n)}]$.

P and NP in Valiant's model

- Family of polynomials (f_n): one circuit C_n per polynomial $f_n \in \mathbb{Z}[x_1, \ldots, x_{u(n)}]$.
- VP: families of polynomials of polynomial degree computed by arithmetic circuits of polynomial size.
  Example: the determinant

$$\det_n(x_{1,1}, \ldots, x_{1,n}, x_{2,1}, \ldots, x_{n,n}) = \sum_{\sigma \in \mathcal{S}_n} \epsilon(\sigma) \prod_{i=1}^{n} x_{i,\sigma(i)}.$$

P and NP in Valiant's model

- VNP: exponential sum of a VP family. If $(f_n(x_1, \ldots, x_{u(n)}, y_1, \ldots, y_{p(n)})) \in \mathrm{VP}$,

$$g_n(x_1, \ldots, x_{u(n)}) = \sum_{\bar{\epsilon} \in \{0,1\}^{p(n)}} f_n(\bar{x}, \bar{\epsilon})$$

  Example: the permanent (VNP-complete)

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

Counting classes

- Languages (PP) or functions ($\sharp\mathrm{P}$). We will focus on languages.

Counting classes

- Languages (PP) or functions ($\sharp\mathrm{P}$). We will focus on languages.
- A language A is in PP if there exists a polynomial-time nondeterministic Turing machine such that $x \in A$ iff more than half of the computation paths are accepting.

Counting classes

- Languages (PP) or functions ($\sharp\mathrm{P}$). We will focus on languages.
- A language A is in PP if there exists a polynomial-time nondeterministic Turing machine such that $x \in A$ iff more than half of the computation paths are accepting.
- A function $f : \{0,1\}^* \to \mathbb{N}$ is in $\sharp\mathrm{P}$ if it counts the number of accepting paths of a polynomial-time nondeterministic Turing machine.

Counting hierarchy

- Counting hierarchy: $\mathrm{CH} = \mathrm{PP} \cup \mathrm{PP}^{\mathrm{PP}} \cup \mathrm{PP}^{\mathrm{PP}^{\mathrm{PP}}} \cup \dots$ (similarity with the polynomial hierarchy).

Counting hierarchy

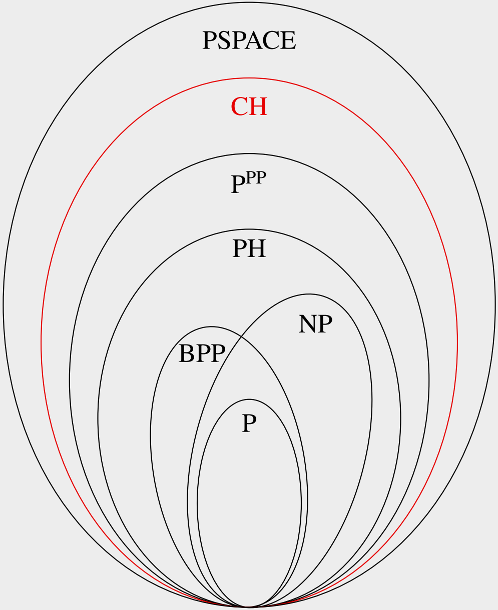
- Counting hierarchy: $\mathrm{CH} = \mathrm{PP} \cup \mathrm{PP}^{\mathrm{PP}} \cup \mathrm{PP}^{\mathrm{PP}^{\mathrm{PP}}} \cup \dots$ (similarity with the polynomial hierarchy).

- Majority operator **C**: if $\mathcal{C}$ is a complexity class, **C**.$\mathcal{C}$ is the set of languages A such that there exists a language $B \in \mathcal{C}$ satisfying:

$$x \in A \iff \#\{y \in \{0,1\}^{p(|x|)} \mid (x,y) \in B\} \geq 2^{p(|x|)-1}.$$

- $C_0P = \mathrm{P}$ et $C_{i+1}P = \mathbf{C}.C_iP$. Then $\mathrm{CH} = \cup_i C_iP$.

Some inclusions



A central lemma

Lemma

If $\mathrm{VP} = \mathrm{VNP}$ *then* $\mathrm{CH} = \mathrm{P}$*.*

Proof (idea)

If $\mathrm{VP} = \mathrm{VNP}$ then the permanent has polynomial-size arithmetic circuits. Then it can be evaluated in polynomial time. Since the permanent is $\sharp\mathrm{P}$-complete, it yields $\mathrm{PP} = \mathrm{P}$, hence $\mathrm{CH} = \mathrm{P}$. □

Sequences of integers

Definition

A sequence of integers $(a_{n,k})_{k \le 2^{p(n)}}$ of exponential bitsize is computable in CH if

$$\{(1^n, k, j, b) \mid \text{the } j\text{-th bit of } a_{n,k} \text{ is } b\} \in \mathrm{CH}.$$

Some results of Bürgisser

Theorem (Bürgisser)

If $(a_{n,k})$ is computable in CH, then it is also the case of

$$c_n = \sum_{k=0}^{2^{p(n)}} a(n,k) \quad \textit{and} \quad d_n = \prod_{k=0}^{2^{p(n)}} a(n,k).$$

Proof (idea)

Key ingredient: iterated addition and multiplication are in LOGTIME-uniform TC^0 (recent result of Hesse, Allender and Barrington for the multiplication). Then scaling up to obtain the result on the counting hierarchy.

TC^0: polynomial-size circuits of constant depth with majority gates.

LOGTIME-uniform: very strong uniformity condition. $\square$

Main result (bis)

If question ♣ has a negative answer, then $\mathrm{VP} \neq \mathrm{VNP}$.

Main result (bis)

If question ♣ has a negative answer, then $\mathrm{VP} \neq \mathrm{VNP}$.

In other words, if $\mathrm{VP} = \mathrm{VNP}$ then question ♣ has a positive answer: we know how to transform an evaluation algorithm into an arithmetic circuit.

Some tools from Lagrange

Going from the evaluation at integer points to the computation:
Lagrange interpolation.

Some tools from Lagrange

Going from the evaluation at integer points to the computation: Lagrange interpolation.

Lemma (Lagrange interpolation)

Let $p(x)$ be a polynomial in one variable and of degree $\leq d$. Then

$$p(x) = \sum_{i=0}^{d} p(i) \prod_{j \neq i} \frac{x-j}{i-j},$$

where the integer j ranges from 0 to d.

Proof

Both polynomials are of degree $\leq d$ and coincide on $d+1$ points. $\square$

Lagrange interpolation

Lemma

Let $p(x_1, \ldots, x_n)$ be a polynomial of degree $\leq d$. Then

$$p(x_1, \ldots, x_n) = \sum_{0 \leq i_1, \ldots, i_n \leq d} p(i_1, \ldots, i_n) \prod_{k=1}^{n} \Bigg(\prod_{j_k \neq i_k} \frac{x_k - j_k}{i_k - j_k} \Bigg),$$

where the integers j_k range from 0 to d.

Main result (ter)

Definition

Let $(f_n(x_1, \ldots, x_{u(n)}))$ be a family of polynomials. We say that (f_n) *can be evaluated in* CH *at integer points* if

$$\{(1^n, i_1, \ldots, i_{u(n)}, j, b) \mid \text{the } j\text{-th bit of } f_n(i_1, \ldots, i_{u(n)}) \text{ is } b\} \in \mathrm{CH}.$$

Main result (ter)

Definition

Let $(f_n(x_1, \ldots, x_{u(n)}))$ be a family of polynomials. We say that (f_n) *can be evaluated in* CH *at integer points* if

$$\{(1^n, i_1, \ldots, i_{u(n)}, j, b) \mid \text{the } j\text{-th bit of } f_n(i_1, \ldots, i_{u(n)}) \text{ is } b\} \in \mathrm{CH}.$$

What we will show:

(if $\mathrm{VP} = \mathrm{VNP}$ and f can be evaluated in CH at integer points)
then f has a polynomial-size circuit.

Valiant's criterion

Definition of $\mathrm{VP_{nb}}$: idem VP but without the polynomial constraint on the degree
$\longrightarrow$ families of polynomials computed by arithmetic circuits of polynomial size.

Valiant's criterion

Definition of $\mathrm{VP_{nb}}$: idem VP but without the polynomial constraint on the degree
$\longrightarrow$ families of polynomials computed by arithmetic circuits of polynomial size.

Lemma

Let

$$f_n(x_1, \ldots, x_n) = \sum_{\alpha^{(1)}, \ldots, \alpha^{(n)}} a(n, \alpha^{(1)}, \ldots, \alpha^{(n)}) x_1^{\alpha^{(1)}} \cdots x_n^{\alpha^{(n)}},$$

where $a(n, \alpha^{(1)}, \ldots, \alpha^{(n)})$ *is a sequence of integers computable in* CH.
If $\mathrm{VP} = \mathrm{VNP}$ *then* $(f_n) \in \mathrm{VP_{nb}}$.

Main theorem

Theorem

Let $(f_n(x_1, \ldots, x_{u(n)}))$ *be a family of multivariate polynomials. Suppose* (f_n) *can be evaluated in* CH *at integer points. If* $\mathrm{VP} = \mathrm{VNP}$ *then* $(f_n) \in \mathrm{VP}_{\mathrm{nb}}$*.*

Main theorem

Theorem

Let $(f_n(x_1, \ldots, x_{u(n)}))$ *be a family of multivariate polynomials. Suppose* (f_n) *can be evaluated in* CH *at integer points. If* $\mathrm{VP} = \mathrm{VNP}$ *then* $(f_n) \in \mathrm{VP}_{\mathrm{nb}}$.

Proof (idea)

- By the results of Bürgisser, the coefficients of the interpolation polynomial are computable in CH.
- By Valiant's criterion, if $\mathrm{VP} = \mathrm{VNP}$ then $(f_n) \in \mathrm{VP}_{\mathrm{nb}}$. $\square$

Summary

- Under the hypothesis $\mathrm{VP} = \mathrm{VNP}$, we aim at showing that a family of polynomials that can be "easily evaluated" has polynomial-size circuits.

Summary

- Under the hypothesis $\mathrm{VP} = \mathrm{VNP}$, we aim at showing that a family of polynomials that can be “easily evaluated” has polynomial-size circuits.
- Idea: use Lagrange interpolation (enables to go from the evaluation to the polynomial itself).

Summary

- Under the hypothesis $\mathrm{VP} = \mathrm{VNP}$, we aim at showing that a family of polynomials that can be "easily evaluated" has polynomial-size circuits.
- Idea: use Lagrange interpolation (enables to go from the evaluation to the polynomial itself).
- Technical points:
  - Valiant's criterion: if the coefficients are computable in CH, then the polynomial has polynomial-size circuits (under the hypothesis that $\mathrm{VP} = \mathrm{VNP}$)
  - the results of Bürgisser enable to compute in CH the coefficients of the interpolation polynomial.

Consequence for question ♣

Theorem

If question ♣ has a negative answer, then $\mathrm{VP} \neq \mathrm{VNP}$.

Consequence for question ♣

Theorem

If question ♣ has a negative answer, then $\mathrm{VP} \neq \mathrm{VNP}$*.*

Remark: if question ♣ has a *positive* answer, then
$\mathrm{P} = \mathrm{PP} \Rightarrow \mathrm{VP} = \mathrm{VNP}$.

Bounded and unbounded versions

Theorem

(In a constant-free context)

$$\mathrm{VP} = \mathrm{VNP} \Rightarrow \mathrm{VP}_{\mathrm{nb}} = \mathrm{VNP}_{\mathrm{nb}}.$$

Remark: on fields of positive characteristic, this result was shown by Malod (2003).

Transfer toward BSS

- Algebraic versions of P and NP: Blum-Shub-Smale model.
- On a field K of characteristic zero, operations $+$, $\times$ and $=$.

Transfer toward BSS

- Algebraic versions of P and NP: Blum-Shub-Smale model.
- On a field K of characteristic zero, operations $+$, $\times$ and $=$.
- Separation of P_K and NP_K thanks to problems in $\mathrm{NP}_{(K,+,=)}$? (Twenty Questions, Subset Sum, ...)

Transfer toward BSS

- Algebraic versions of P and NP: Blum-Shub-Smale model.
- On a field K of characteristic zero, operations $+$, $\times$ and $=$.
- Separation of P_K and NP_K thanks to problems in $\mathrm{NP}_{(K,+,=)}$? (Twenty Questions, Subset Sum, ...)

Theorem

$\mathrm{VP} = \mathrm{VNP} \Rightarrow \mathrm{NP}_{(K,+,=)} \subseteq \mathrm{P}_{(K,+,\times,=)}$.

We use exponential-size products as an intermediate step.

Conclusion

- Question ♣ is central but difficult: if the answer is positive, we obtain a transfer result; otherwise we obtain the separation of VP and VNP.

Conclusion

- Question ♣ is central but difficult: if the answer is positive, we obtain a transfer result; otherwise we obtain the separation of VP and VNP.
- Little intuition on the answer.

Conclusion

- Question ♣ is central but difficult: if the answer is positive, we obtain a transfer result; otherwise we obtain the separation of VP and VNP.
- Little intuition on the answer.
- Candidates for a negative answer? (polynomials that can be easily evaluated but that do not have polynomial-size circuits)

Outline

1. Valiant's classes
2. The counting hierarchy
3. Interpolation
4. Consequences