VPSPACE and a transfer theorem over the complex numbers

The question "$\mathrm{P} = \mathrm{PSPACE}$?" in algebraic complexity

Pascal Koiran   Sylvain Perifel

LIP, ENS Lyon

Český Krumlov, August 30, 2007

Introduction

- Decision problems
  Languages (over $\mathbb{C}$), Blum-Shub-Smale model
  Example: decide whether a system of multivariate polynomials has a solution ($\mathrm{NP}_{\mathbb{C}}$-complete)

Introduction

- Decision problems
  Languages (over $\mathbb{C}$), Blum-Shub-Smale model
  Example: decide whether a system of multivariate polynomials has a solution ($\mathrm{NP}_{\mathbb{C}}$-complete)

- Evaluation problems
  Families of polynomials, Valiant's model
  Example: compute the permanent of a matrix (VNP-complete)

Outline

1. P and PSPACE (boolean case)

2. P and PSPACE in BSS model

3. P and PSPACE in Valiant's model

4. Sign condition

if $\mathrm{VP} = \mathrm{VPSPACE}$ then $\mathrm{P}_{\mathbb{C}} = \mathrm{PAR}_{\mathbb{C}}$

P and PSPACE (boolean case)

- P: languages over $\{0,1\}$ recognized in polynomial time by a Turing machine.
- PSPACE: languages over $\{0,1\}$ recognized in polynomial space by a Turing machine.

P and PSPACE (boolean case)

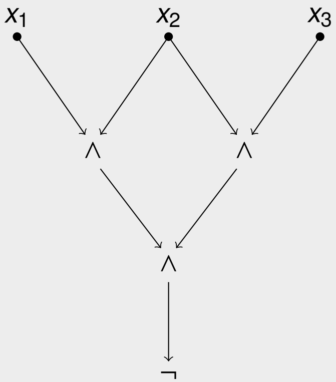
- P: languages over $\{0, 1\}$ recognized in polynomial time by a Turing machine.
- PSPACE: languages over $\{0, 1\}$ recognized in polynomial space by a Turing machine.

Turing machines

$\longleftrightarrow$

boolean circuits

(gates $\wedge$, $\vee$, $\neg$).



P and PSPACE (boolean case)

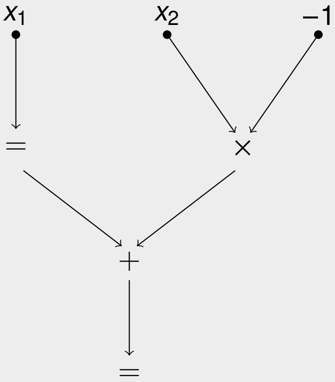
- Language recognition: one circuit per input length.

P and PSPACE (boolean case)

- Language recognition: one circuit per input length.
- P: languages recognized by boolean circuits of polynomial size (+ uniformity).
- PSPACE: languages recognized by boolean circuits of polynomial *depth* (of possibly exponential size) (+ uniformity).

P and PSPACE in BSS model

Algebraic circuits: gates $+$, $-$, $\times$ and $=$.



P and PSPACE in BSS model

- Languages over $\mathbb{C}$: sets of words over the alphabet $\mathbb{C}$, that is, $A \subseteq \cup_{n \geq 0} \mathbb{C}^n$.
- Language recognition over $\mathbb{C}$: one circuit per input length.

P and PSPACE in BSS model

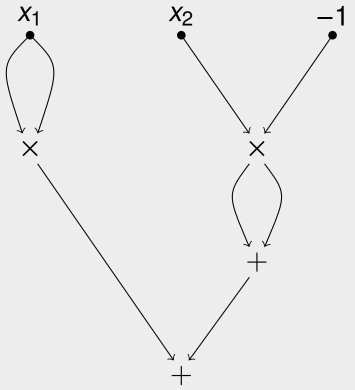
- Languages over $\mathbb{C}$: sets of words over the alphabet $\mathbb{C}$, that is, $A \subseteq \cup_{n \geq 0} \mathbb{C}^n$.
- Language recognition over $\mathbb{C}$: one circuit per input length.
- $\mathrm{P}_{\mathbb{C}}$: languages over $\mathbb{C}$ recognized by algebraic circuits of polynomial size (+ uniformity).

P and PSPACE in BSS model

- Languages over $\mathbb{C}$: sets of words over the alphabet $\mathbb{C}$, that is, $A \subseteq \cup_{n \geq 0} \mathbb{C}^n$.
- Language recognition over $\mathbb{C}$: one circuit per input length.
- $\mathrm{P}_{\mathbb{C}}$: languages over $\mathbb{C}$ recognized by algebraic circuits of polynomial size (+ uniformity).
- $\mathrm{PAR}_{\mathbb{C}}$: languages over $\mathbb{C}$ recognized by algebraic circuits of polynomial *depth* (of possibly exponential size) (+ uniformity).

P and PSPACE in Valiant's model

Arithmetic circuits: gates $+$, – and $\times$, inputs $x_1, \ldots, x_n$ and constant 1 $\longrightarrow$ multivariate polynomial with integer coefficients.



P and PSPACE 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 PSPACE 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 computed by arithmetic circuits of polynomial size (+ uniformity).

$(= \text{Uniform } \mathrm{VP}^0_{\mathrm{nb}})$

P and PSPACE 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 computed by arithmetic circuits of polynomial size (+ uniformity).

  $(= \text{Uniform VP}^0_{\text{nb}})$
- VPSPACE: families of polynomials computed by arithmetic circuits of polynomial *depth* (+ uniformity).

Recapitulation

- Decision problems over $\{0, 1\}$: boolean circuits (gates $\wedge$, $\vee$ et $\neg$).
- Decision problems over $\mathbb{C}$ (BSS): algebraic circuits (gates $+$, $-$, $\times$, $=$).
- Evaluation problems (Valiant): arithmetic circuits (gates $+$, $-$, $\times$).

Recapitulation

- Decision problems over $\{0, 1\}$: boolean circuits (gates $\wedge$, $\vee$ et $\neg$).
- Decision problems over $\mathbb{C}$ (BSS): algebraic circuits (gates $+$, $-$, $\times$, $=$).
- Evaluation problems (Valiant): arithmetic circuits (gates $+$, $-$, $\times$).

- P: circuits of polynomial size.
- PSPACE: circuits of polynomial depth.

Other characterizations of $\mathrm{VPSPACE}$

- Original definition: coefficient function in PSPACE.

$$f_n(\bar{x}) = \sum_{\alpha} a(\alpha)\bar{x}^{\alpha}$$

Function $a : \{0,1\}^* \rightarrow \mathbb{Z}$ computable bit by bit in polynomial space.

Other characterizations of VPSPACE

- Original definition: coefficient function in PSPACE.

$$f_n(\bar{x}) = \sum_{\alpha} a(\alpha)\bar{x}^{\alpha}$$

Function $a : \{0,1\}^* \to \mathbb{Z}$ computable bit by bit in polynomial space.

- Poizat: circuits of polynomial size endowed with exponential summation gates

Other characterizations of VPSPACE

- Original definition: coefficient function in PSPACE.

$$f_n(\bar{x}) = \sum_{\alpha} a(\alpha)\bar{x}^{\alpha}$$

  Function $a : \{0,1\}^* \to \mathbb{Z}$ computable bit by bit in polynomial space.
- Poizat: circuits of polynomial size endowed with exponential summation gates
  or gates of evaluation at 0 and 1.

Other characterizations of VPSPACE

- Original definition: coefficient function in PSPACE.

$$f_n(\bar{x}) = \sum_{\alpha} a(\alpha)\bar{x}^{\alpha}$$

  Function $a : \{0,1\}^* \to \mathbb{Z}$ computable bit by bit in polynomial space.

- Poizat: circuits of polynomial size endowed with exponential summation gates
  or gates of evaluation at 0 and 1.
- Example: multivariate resultant of a system of polynomials.

Other characterizations of VPSPACE

- Original definition: coefficient function in PSPACE.

$$f_n(\bar{x}) = \sum_{\alpha} a(\alpha)\bar{x}^{\alpha}$$

  Function $a : \{0,1\}^* \to \mathbb{Z}$ computable bit by bit in polynomial space.
- Poizat: circuits of polynomial size endowed with exponential summation gates
  or gates of evaluation at 0 and 1.
- Example: multivariate resultant of a system of polynomials.

- Proposition: $\mathrm{VPSPACE} = \mathrm{VP} \Longrightarrow \mathrm{PSPACE} = \mathrm{P}$.

Transfer theorem

If $\mathrm{VPSPACE} = \mathrm{VP}$ then $\mathrm{PAR}_{\mathbb{C}} = \mathrm{P}_{\mathbb{C}}$.

Outline of the proof:

- Goal: for $A \in \mathrm{PAR}_{\mathbb{C}}$, decide in polynomial time (with $\mathrm{VPSPACE}$ tests) whether $\bar{x} \in A$.
- Find the sign condition of $\bar{x}$
- Simulate the circuit on this sign condition.

Transfer theorem

If $\mathrm{VPSPACE} = \mathrm{VP}$ then $\mathrm{PAR}_{\mathbb{C}} = \mathrm{P}_{\mathbb{C}}$.

Outline of the proof:

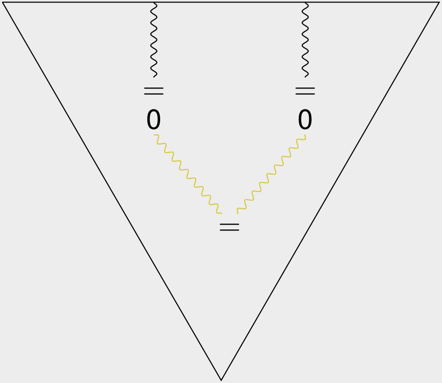
- Goal: for $A \in \mathrm{PAR}_{\mathbb{C}}$, decide in polynomial time (with $\mathrm{VPSPACE}$ tests) whether $\bar{x} \in A$.
- Find the sign condition of $\bar{x}$
  - enumeration of the satisfiable sign conditions (Fichtas, Galligo, Morgenstern);
  - binary search.
- Simulate the circuit on this sign condition.

Polynomials tested by a circuit

Test gate: $f(\bar{x}) = 0$?

If the results of the preceding tests are fixed, f is a polynomial.

→ enumeration of all possible polynomials (polynomial space): family $f_1, \ldots, f_s$.

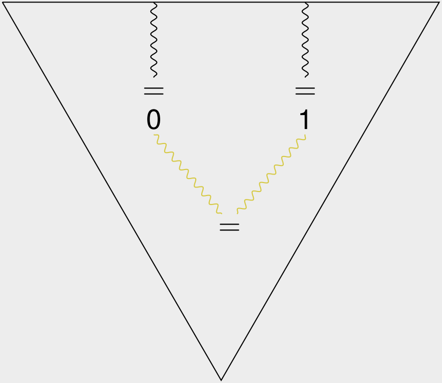


Polynomials tested by a circuit

Test gate: $f(\bar{x}) = 0$?

If the results of the preceding tests are fixed, f is a polynomial.

$\rightarrow$ enumeration of all possible polynomials (polynomial space): family $f_1, \ldots, f_s$.



Sign conditions

- Sign condition $S \in \{0, 1\}^s$: "sign" of the polynomials $f_1, \ldots, f_s$, i.e. 0 if $f_i(\bar{x}) = 0$ and 1 otherwise.
- Sign condition of $\bar{x}$: $(\mathrm{sign}(f_1(\bar{x})), \ldots, \mathrm{sign}(f_s(\bar{x})))$.

Sign conditions

- Sign condition $S \in \{0,1\}^s$: "sign" of the polynomials $f_1, \ldots, f_s$, i.e. 0 if $f_i(\bar{x}) = 0$ and 1 otherwise.
- Sign condition of $\bar{x}$: $(\mathrm{sign}(f_1(\bar{x})), \ldots, \mathrm{sign}(f_s(\bar{x})))$.

- If $\bar{x}$ and $\bar{y}$ have the same sign condition then every test gives the same result $\longrightarrow$ $\bar{x}$ and $\bar{y}$ are simultaneously in the language or outside of the language.
- It is enough to study the sign condition (boolean object).

Satisfiable sign conditions

- Sign condition $S \in \{0, 1\}^s$: sign of the polynomials $f_1, \ldots, f_s$.
- A sign condition is not necessarily satisfiable.
- Example: x and $x + 1$ cannot be both zero, hence $(0, 0)$ is not satisfiable.

Satisfiable sign conditions

- Sign condition $S \in \{0, 1\}^s$: sign of the polynomials $f_1, \ldots, f_s$.
- A sign condition is not necessarily satisfiable.
- Example: x and $x + 1$ cannot be both zero, hence $(0, 0)$ is not satisfiable.

Theorem (Fichtas, Galligo, Morgenstern 1990)

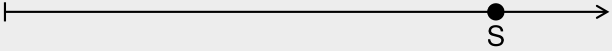
- *There are $N = (sd)^{O(n)}$ satisfiable sign conditions (s: number of polynomials, n: number of variables, d: max degree).*
- *Satisfiable sign conditions can be enumerated in* PSPACE.

Finding the sign condition

- Linear order compatible with inclusion on satisfiable sign conditions:

Finding the sign condition

- Linear order compatible with inclusion on satisfiable sign conditions:

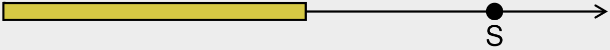


- Search of the minimal satisfiable sign condition S satisfying

$$\forall k \leq s, S_k = 0 \Longrightarrow f_k(\bar{x}) = 0.$$

Finding the sign condition

- Linear order compatible with inclusion on satisfiable sign conditions:

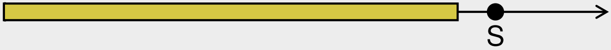


- Search of the minimal satisfiable sign condition S satisfying

$$\forall k \leq s, S_k = 0 \implies f_k(\bar{x}) = 0.$$

Finding the sign condition

- Linear order compatible with inclusion on satisfiable sign conditions:



- Search of the minimal satisfiable sign condition S satisfying

$$\forall k \leq s, S_k = 0 \Longrightarrow f_k(\bar{x}) = 0.$$

Finding the sign condition

- Linear order compatible with inclusion on satisfiable sign conditions:

- Search of the minimal satisfiable sign condition S satisfying

$$\forall k \leq s, S_k = 0 \Longrightarrow f_k(\bar{x}) = 0.$$

Finding the sign condition

- Linear order compatible with inclusion on satisfiable sign conditions:

- Search of the minimal satisfiable sign condition S satisfying

$$\forall k \leq s, S_k = 0 \implies f_k(\bar{x}) = 0.$$

- Over $\mathbb{R}$, easy thanks to $\mathrm{VPSPACE}$ tests

$$\prod_{j \leq i} \left(\sum_{S_k^{(j)}=0} f_k(\bar{x})^2 \right) = 0 \ \ \text{(true iff } S \leq i\text{)}$$

Membership to a variety

- Over $\mathbb{C}$: no "sum of squares" trick.
- We have to test with a polynomial number of tests if $\bar{x} \in V$ for a variety V given by an exponential number of polynomials.

Membership to a variety

- Over $\mathbb{C}$: no "sum of squares" trick.
- We have to test with a polynomial number of tests if $\bar{x} \in V$ for a variety V given by an exponential number of polynomials.
- Nonconstructively: use the following lemma.

Lemma

Let $V \in \mathbb{C}^n$ be a variety defined by s polynomials $f_1, \ldots, f_s$. Then V is defined by $n+1$ generic linear combinations $g_1, \ldots g_{n+1}$ of the f_i.

"generic": $g_i = \sum_{j=1}^{s} \alpha_{i,j} f_j$ where the $\alpha_{i,j}$ are algebraically independent.

Membership to a variety

- Over $\mathbb{C}$: no "sum of squares" trick.
- We have to test with a polynomial number of tests if $\bar{x} \in V$ for a variety V given by an exponential number of polynomials.
- Nonconstructively: use the following lemma.

Lemma

Let $V \in \mathbb{C}^n$ be a variety defined by s polynomials $f_1, \ldots, f_s$. Then V is defined by $n+1$ generic linear combinations $g_1, \ldots g_{n+1}$ of the f_i.

"generic": $g_i = \sum_{j=1}^{s} \alpha_{i,j} f_j$ where the $\alpha_{i,j}$ are algebraically independent.

- Problem: we can only use integers.

Constructive tests

Lemma (Nonconstructive, reminder)

Let $V \in \mathbb{C}^n$ be a variety defined by s polynomials $f_1, \ldots, f_s$. Then V is defined by $n+1$ generic linear combinations $g_1, \ldots g_{n+1}$ of the f_i.

Replace transcendant numbers by integers growing sufficiently fast.

Lemma

Let $\phi(x_1, \ldots, x_n)$ be a first order formula which is true on any algebraically independent coefficients $\alpha_1, \ldots, \alpha_n$. Then $\phi(\beta_1, \ldots, \beta_n)$ is true for any integers β_i growing sufficiently fast.

Proof idea: lack of “big” roots of multivariate polynomials.

Membership tests

- V defined by $f_1, \ldots, f_s$ (s exponential). Decide $x \in V$ with a polynomial number of tests.

Membership tests

- V defined by $f_1, \ldots, f_s$ (s exponential). Decide $x \in V$ with a polynomial number of tests.
- Let $\phi(\bar{\alpha}) \equiv$ the $n + 1$ linear combinations of the f_i with coefficients $\bar{\alpha}$ also define V.
- By the first lemma, $\phi(\bar{\alpha})$ is true for all algebraically independent coefficients $\bar{\alpha}$.

Membership tests

- V defined by $f_1, \ldots, f_s$ (s exponential). Decide $x \in V$ with a polynomial number of tests.
- Let $\phi(\bar{\alpha}) \equiv$ the $n + 1$ linear combinations of the f_i with coefficients $\bar{\alpha}$ also define V.
- By the first lemma, $\phi(\bar{\alpha})$ is true for all algebraically independent coefficients $\bar{\alpha}$.
- By the second lemma, $\phi(\bar{\beta})$ is true for integers $\bar{\beta}$ growing sufficiently fast: V is then defined by the $n + 1$ linear combinations of the f_i with coefficients $\bar{\beta}$.
- Hence $n + 1$ polynomials to test to zero.

Union of varieties

- Actual tests to be performed: membership to a union of an exponential number of varieties.

Union of varieties

- Actual tests to be performed: membership to a union of an exponential number of varieties.
- Naive approach: products of the polynomials. But too many of them.
- $\rightarrow$ Divide and conquer.

Union of varieties

- Actual tests to be performed: membership to a union of an exponential number of varieties.
- Naive approach: products of the polynomials. But too many of them.
- $\rightarrow$ Divide and conquer.
- We can perform the binary search for the sign condition in polynomial time (with $\mathrm{VPSPACE}$ tests).

Recapitulation

In order to show that $\mathrm{VPSPACE} = \mathrm{VP} \Rightarrow \mathrm{PAR}_{\mathbb{C}} = \mathrm{P}_{\mathbb{C}}$:

- For $A \in \mathrm{PAR}_{\mathbb{C}}$ we want to decide in polynomial time (with $\mathrm{VPSPACE}$ tests) whether $\bar{x} \in A$.
- We enumerate all the polynomials possibly tested in the cricuit (polynomial space).
- Thanks to $\mathrm{VPSPACE}$ tests, a binary search gives the sign condition of $\bar{x}$.
- Once the sign condition of $\bar{x}$ is obtained, we can simulate the circuit and conclude.

Recapitulation

In order to show that $\mathrm{VPSPACE} = \mathrm{VP} \Rightarrow \mathrm{PAR}_{\mathbb{C}} = \mathrm{P}_{\mathbb{C}}$:

- For $A \in \mathrm{PAR}_{\mathbb{C}}$ we want to decide in polynomial time (with $\mathrm{VPSPACE}$ tests) whether $\bar{x} \in A$.
- We enumerate all the polynomials possibly tested in the cricuit (polynomial space).
- Thanks to $\mathrm{VPSPACE}$ tests, a binary search gives the sign condition of $\bar{x}$.
- Once the sign condition of $\bar{x}$ is obtained, we can simulate the circuit and conclude.

Main ideas:

1. sign conditions;
2. binary search thanks to tests of membership to varieties;
3. integers instead of transcendant numbers.

Conclusion

- Study of the question $\mathrm{P} = \mathrm{PSPACE}$ in different contexts (boolean, BSS, Valiant).
- Similar results over $\mathbb{R}$ but different techniques: we have to take into account the sign ($\rightarrow$ a vector orthogonal to roughly half a collection of vectors).
- Converse? Nullstellensatz $\Rightarrow$ work only up to a multiple.

Outline

1. P and PSPACE (boolean case)

2. P and PSPACE in BSS model

3. P and PSPACE in Valiant's model

4. Sign condition