The incredible shrinking algebraic model of computation

Pascal Koiran
Ecole Normale Supérieure de Lyon

14th International Summer School-Conference:
"Problems Allied to Universal Algebra and Model Theory"
June 2021

What is so special about 2021?

What is so special about 2021?

- Bruno Poizat 75th birthday.

What is so special about 2021?

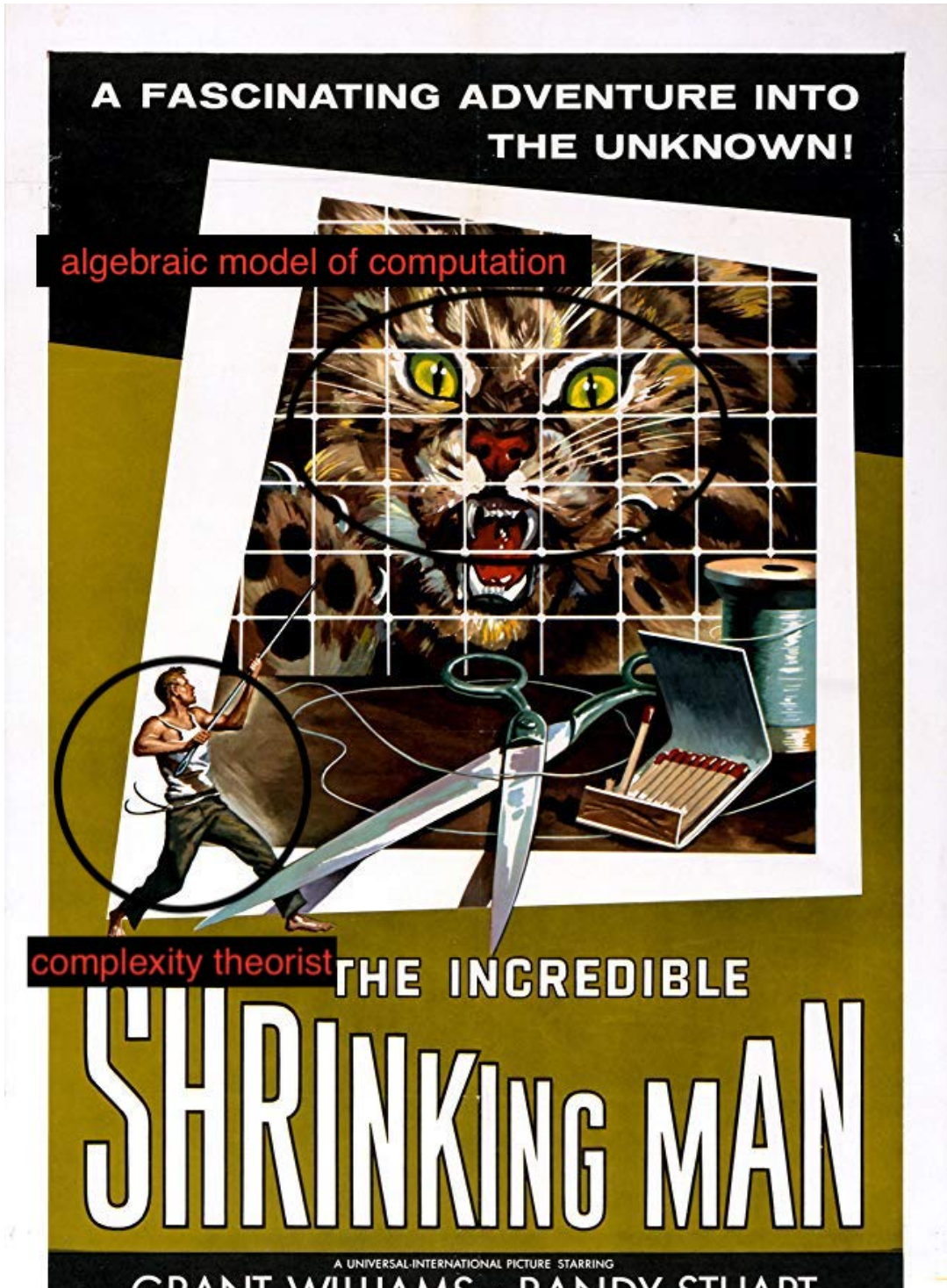
- Bruno Poizat 75th birthday.
- 50th anniversary of NP-completeness and the P vs NP problem (Cook'71, Levin'73).

What is so special about 2021?

- Bruno Poizat 75th birthday.
- 50th anniversary of NP-completeness and the P vs NP problem (Cook'71, Levin'73).

This talk: a (biased) survey on algebraic models of computation, including algebraic versions of P versus NP.

In this old (1957) black-and-white movie, a businessman is exposed to radioactive dust during a boating trip. He begins to shrink while researchers try and fail to stop that process.


A FASCINATING ADVENTURE INTO THE UNKNOWN!
algebraic model of computation
complexity theorist
THE INCREDIBLE
SHRINKING MAN
A UNIVERSAL-INTERNATIONAL PICTURE STARRING

The Blum-Shub-Smale model

- Formalization of "real RAM" model:
  exact arithmetic $(+, -, \times, \leq)$ on real numbers.
- Versions of "P=NP?" problem over $\mathbb{R}$, $\mathbb{C}$, finite fields...
- Initial paper: L. Blum, M. Shub and S. Smale. On a theory of computation and complexity over the real numbers: NP-completeness, recursive functions and universal machines, Bulletin AMS, 1989.
- Their book: L. Blum, F. Cucker, M. Shub, S. Smale. *Complexity and Real Computation*, Springer, 1998. Includes lots of material on numerical algorithms.
- B. Poizat's book: Les petits cailloux. Aléas, 1995. Model theoretic point of view.

The Blum-Shub-Smale model

- Formalization of "real RAM" model:
  exact arithmetic $(+, -, \times, \leq)$ on real numbers.
- Versions of "P=NP?" problem over $\mathbb{R}$, $\mathbb{C}$, finite fields...
- Initial paper: L. Blum, M. Shub and S. Smale. On a theory of computation and complexity over the real numbers: NP-completeness, recursive functions and universal machines, Bulletin AMS, 1989.
- Their book: L. Blum, F. Cucker, M. Shub, S. Smale. *Complexity and Real Computation*, Springer, 1998. Includes lots of material on numerical algorithms.
- B. Poizat's book: Les petits cailloux. Aléas, 1995. Model theoretic point of view.

The Blum-Shub-Smale model

- Formalization of "real RAM" model:
  exact arithmetic $(+, -, \times, \leq)$ on real numbers.
- Versions of "P=NP?" problem over $\mathbb{R}$, $\mathbb{C}$, finite fields...
- Initial paper: L. Blum, M. Shub and S. Smale. On a theory of computation and complexity over the real numbers:
  NP-completeness, recursive functions and universal machines,
  Bulletin AMS, 1989.
- Their book: L. Blum, F. Cucker, M. Shub, S. Smale.
  Complexity and Real Computation, Springer, 1998.
  Includes lots of material on numerical algorithms.
- B. Poizat's book: Les petits cailloux. Aléas, 1995.
  Model theoretic point of view.

The Blum-Shub-Smale model

- Formalization of "real RAM" model:
  exact arithmetic $(+, -, \times, \leq)$ on real numbers.
- Versions of "P=NP?" problem over $\mathbb{R}$, $\mathbb{C}$, finite fields...
- Initial paper: L. Blum, M. Shub and S. Smale. On a theory of computation and complexity over the real numbers:
  NP-completeness, recursive functions and universal machines,
  Bulletin AMS, 1989.
- Their book: L. Blum, F. Cucker, M. Shub, S. Smale.
  Complexity and Real Computation, Springer, 1998.
  Includes lots of material on numerical algorithms.
- B. Poizat's book: Les petits cailloux. Aléas, 1995.
  Model theoretic point of view.

Bruno POIZAT

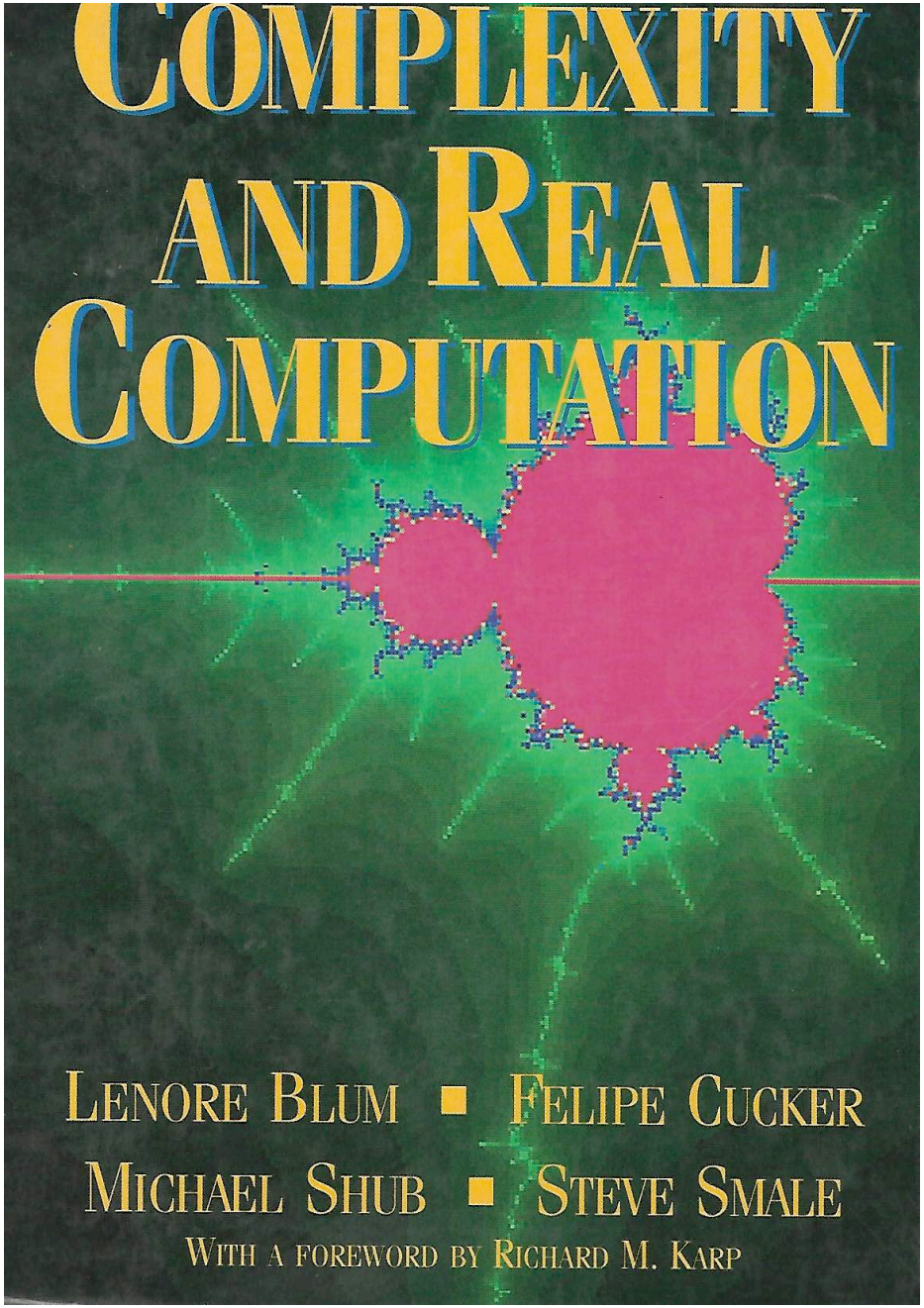
LES PETITS CAILLOUX

Une approche
modèle-théorique
de l'Algorithmie

نور المنطق والمعرفة

NUR AL-MANTIQ WAL-MA'RIFAH

ALÉAS


COMPLEXITY
AND REAL
COMPUTATION
LENORE BLUM ■ FELIPE CUCKER
MICHAEL SHUB ■ STEVE SMALE
WITH A FOREWORD BY RICHARD M. KARP

Computation in first-order structures

- Complexity classes $\mathrm{P}_M, \mathrm{NP}_M$ for any structure M.
  When M is finite, we obtain the usual classes P and NP.
- Model of computation: multi-tape Turing machines over M.
  Each cell contains an element of M.
  Machine can apply functions and relations of M.
  The input is a finite sequence of elements of M.
- We can also define P_M using uniform circuit families over M.
  These generalize Boolean circuits.

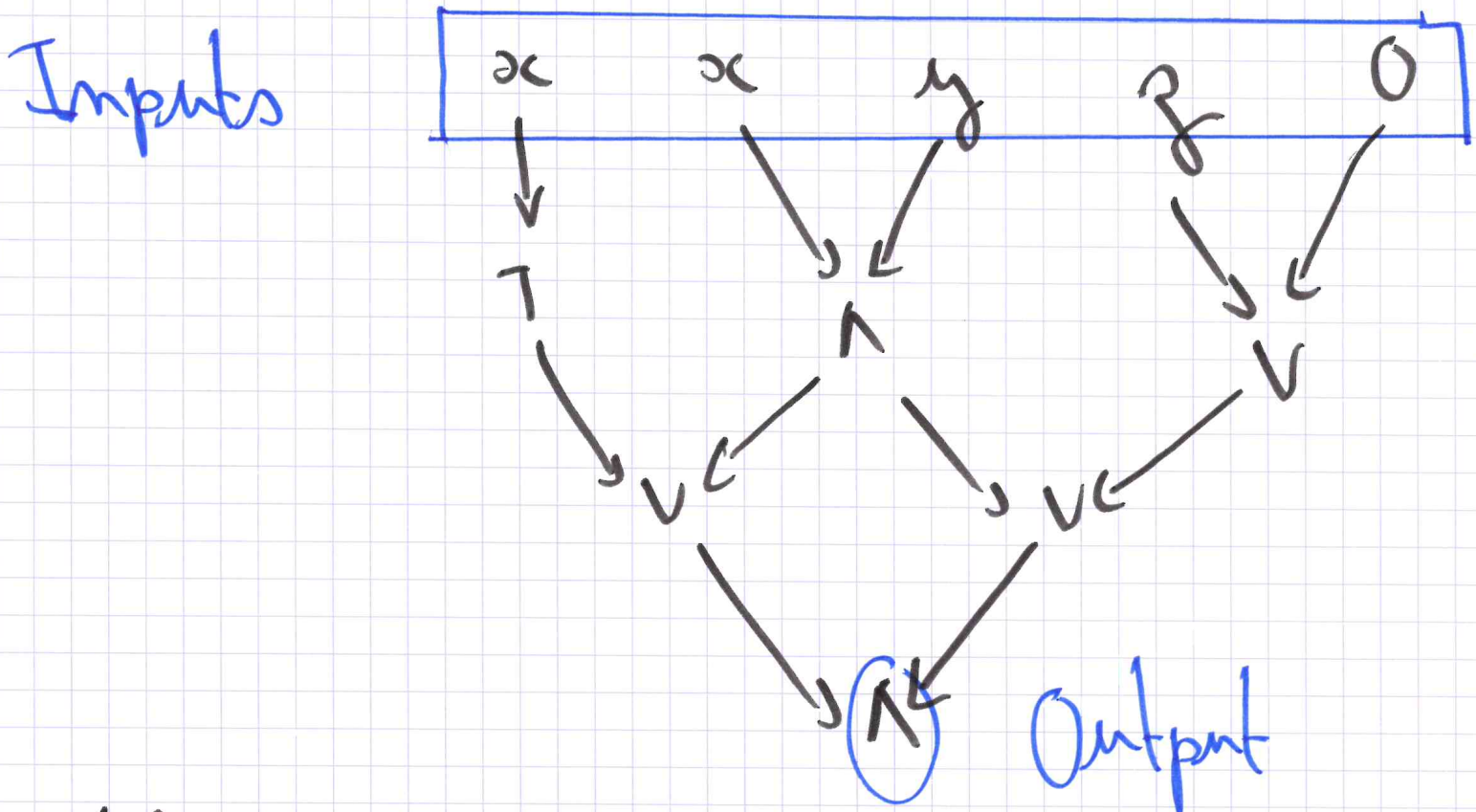
Computation in first-order structures

- Complexity classes $\mathrm{P}_M, \mathrm{NP}_M$ for any structure M.
  When M is finite, we obtain the usual classes P and NP.
- Model of computation: multi-tape Turing machines over M.
  Each cell contains an element of M.
  Machine can apply functions and relations of M.
  The input is a finite sequence of elements of M.
- We can also define P_M using uniform circuit families over M.
  These generalize Boolean circuits.

Computation in first-order structures

- Complexity classes $\mathrm{P}_M, \mathrm{NP}_M$ for any structure M.
  When M is finite, we obtain the usual classes P and NP.
- Model of computation: multi-tape Turing machines over M.
  Each cell contains an element of M.
  Machine can apply functions and relations of M.
  The input is a finite sequence of elements of M.
- We can also define P_M using uniform circuit families over M.
  These generalize Boolean circuits.

Boolean circuits



With circuit family $C_n(x_1, \ldots, x_n)$:
we recognize a language $L \subseteq \{0,1\}^*$

Circuits over M

- We have gates for functions and relations of M.
- For instance: $+, \times, \leq$ over $\mathbb{R}$.
  if-then-else statement: $s(x, y, z) = xy + (1 - x)z$.
- Formulas have a tree structure,
  but circuits are directed acyclic graphs $\Rightarrow$
  Circuits are compact representations of quantifier-free formulas.

Circuits over M

- We have gates for functions and relations of M.
- For instance: $+, \times, \leq$ over $\mathbb{R}$.
  if-then-else statement: $s(x, y, z) = xy + (1 - x)z$.
- Formulas have a tree structure,
  but circuits are directed acyclic graphs $\Rightarrow$
  Circuits are compact representations of quantifier-free formulas.

NP and quantifier elimination

For a problem $A \in \mathrm{NP}_M$:

$$\overline{x} \in A \Leftrightarrow \exists \overline{y} \in M^{p(n)} \langle \overline{x}, \overline{y} \rangle \in B$$

where $\overline{x} \in M^n$ is the input, $B \in \mathrm{P}$ and p is a polynomial.
$\overline{y}$ is the "certificate" of membership to A.
Suppose M admits quantifier elimination.

- For any quantifier-free formula F:

$$\exists \overline{y} F(\overline{x}, \overline{y}) \Leftrightarrow G(\overline{x})$$

  where G is quantifier free.

- $\mathrm{P}_M = \mathrm{NP}_M$ means:
  G can be constructed in polynomial time from F,
  if we represent G by a circuit over M.

NP and quantifier elimination

For a problem $A \in \mathrm{NP}_M$:

$$\overline{x} \in A \Leftrightarrow \exists \overline{y} \in M^{p(n)} \langle \overline{x}, \overline{y} \rangle \in B$$

where $\overline{x} \in M^n$ is the input, $B \in \mathrm{P}$ and p is a polynomial.
$\overline{y}$ is the "certificate" of membership to A.
Suppose M admits quantifier elimination.

- For any quantifier-free formula F:

  $$\exists \overline{y} F(\overline{x}, \overline{y}) \Leftrightarrow G(\overline{x})$$

  where G is quantifier free.
- $\mathrm{P}_M = \mathrm{NP}_M$ means:
  G can be constructed in polynomial time from F,
  if we represent G by a circuit over M.

The BSS model today

- Research area no longer so active.
- Some exceptions: $\exists\mathbb{R}$-complete problems ($\simeq \mathrm{NP}_{\mathbb{R}}$-complete): Nash equilibria [Schaefer-Štefankovič], graph drawing, tensor rank [SS, Shitov], covering of polygons [Abrahamsen'21]...
  Tensor rank is $\mathrm{NP}_{\mathbb{F}}$-complete for any field $\mathbb{F}$ [SS'18].

$\mathrm{P}_{\mathbb{R}}$ vs $\mathrm{NP}_{\mathbb{R}}$, $\mathrm{P}_{\mathbb{C}}$ vs $\mathrm{NP}_{\mathbb{C}}$ are natural problems,
closely connected to complexity of quantifier elimination.
They should be solved someday!

The BSS model today

- Research area no longer so active.
- Some exceptions: $\exists\mathbb{R}$-complete problems ($\simeq \mathrm{NP}_{\mathbb{R}}$-complete): Nash equilibria [Schaefer-Štefankovič], graph drawing, tensor rank [SS, Shitov], covering of polygons [Abrahamsen'21]...
  Tensor rank is $\mathrm{NP}_{\mathbb{F}}$-complete for any field $\mathbb{F}$ [SS'18].

$\mathrm{P}_{\mathbb{R}}$ vs $\mathrm{NP}_{\mathbb{R}}$, $\mathrm{P}_{\mathbb{C}}$ vs $\mathrm{NP}_{\mathbb{C}}$ are natural problems,
closely connected to complexity of quantifier elimination.
They should be solved someday!

The BSS model today

- Research area no longer so active.
- Some exceptions: $\exists\mathbb{R}$-complete problems ($\simeq \mathrm{NP}_{\mathbb{R}}$-complete):
  Nash equilibria [Schaefer-Štefankovič], graph drawing,
  tensor rank [SS, Shitov], covering of polygons
  [Abrahamsen'21]...
  Tensor rank is $\mathrm{NP}_{\mathbb{F}}$-complete for any field $\mathbb{F}$ [SS'18].

$\mathrm{P}_{\mathbb{R}}$ vs $\mathrm{NP}_{\mathbb{R}}$, $\mathrm{P}_{\mathbb{C}}$ vs $\mathrm{NP}_{\mathbb{C}}$ are natural problems,
closely connected to complexity of quantifier elimination.
They should be solved someday!

BSS machines are very powerful:

+ good expressive power.
- proving lower bounds is difficult.

We still have the lower bounds:

BSS machines are very powerful:

+ good expressive power.
- proving lower bounds is difficult.

We still have the lower bounds:

- $\mathrm{NC}_{\mathbb{R}} \subsetneq \mathrm{P}_{\mathbb{R}} \subsetneq \mathrm{EXP}_{\mathbb{R}}$
  (Cucker'92, degree argument).
- For fixed polynomial time: $\mathrm{TIME}_{\mathbb{R}}(n^d) \neq \mathrm{NTIME}_{\mathbb{R}}(n^d)$
  (Cucker-Shub'96, number of connected components).

BSS machines are very powerful:

+ good expressive power.
- proving lower bounds is difficult.

We still have the lower bounds:

- $\mathrm{NC}_{\mathbb{R}} \subsetneq \mathrm{P}_{\mathbb{R}} \subsetneq \mathrm{EXP}_{\mathbb{R}}$
  (Cucker'92, degree argument).
- For fixed polynomial time: $\mathrm{TIME}_{\mathbb{R}}(n^d) \neq \mathrm{NTIME}_{\mathbb{R}}(n^d)$
  (Cucker-Shub'96, number of connected components).

And that's it!

BSS machines are very powerful:

+ good expressive power.
- proving lower bounds is difficult.

We still have the lower bounds:

- $\mathrm{NC}_{\mathbb{R}} \subsetneq \mathrm{P}_{\mathbb{R}} \subsetneq \mathrm{EXP}_{\mathbb{R}}$
  (Cucker'92, degree argument).
- For fixed polynomial time: $\mathrm{TIME}_{\mathbb{R}}(n^d) \neq \mathrm{NTIME}_{\mathbb{R}}(n^d)$
  (Cucker-Shub'96, number of connected components).

And that's it!

A "solution" to this problem:

BSS machines are very powerful:

+ good expressive power.
- proving lower bounds is difficult.

We still have the lower bounds:

- $\mathrm{NC}_{\mathbb{R}} \subsetneq \mathrm{P}_{\mathbb{R}} \subsetneq \mathrm{EXP}_{\mathbb{R}}$
  (Cucker'92, degree argument).
- For fixed polynomial time: $\mathrm{TIME}_{\mathbb{R}}(n^d) \neq \mathrm{NTIME}_{\mathbb{R}}(n^d)$
  (Cucker-Shub'96, number of connected components).

And that's it!

A "solution" to this problem: let's shrink the model!

BSS machines are very powerful:

+ good expressive power.
- proving lower bounds is difficult.

We still have the lower bounds:

- $\mathrm{NC}_{\mathbb{R}} \subsetneq \mathrm{P}_{\mathbb{R}} \subsetneq \mathrm{EXP}_{\mathbb{R}}$
  (Cucker'92, degree argument).
- For fixed polynomial time: $\mathrm{TIME}_{\mathbb{R}}(n^d) \neq \mathrm{NTIME}_{\mathbb{R}}(n^d)$
  (Cucker-Shub'96, number of connected components).

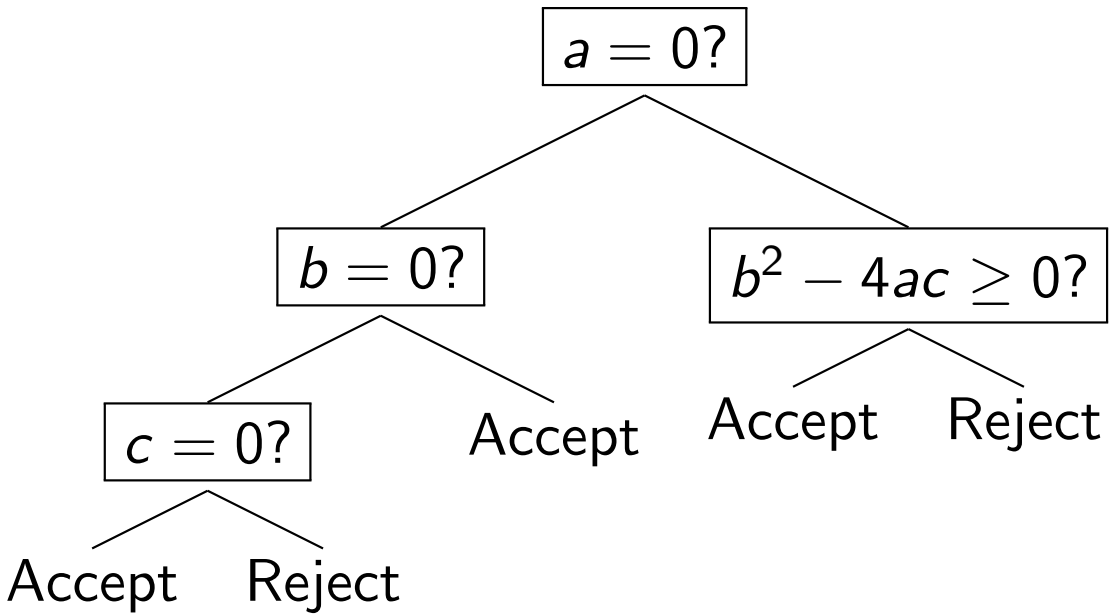
And that's it!

A "solution" to this problem: let's shrink the model!

For instance: $\mathrm{P} \neq \mathrm{NP}$ over $(\mathbb{R}, +, -, =)$ (Meer, 1992)

Decision and computation trees

$$\exists x \in \mathbb{R}\ ax^2 + bx + c = 0\ ?$$



For polynomials of bounded degree (e.g., $d = 1, d = 2$):
Complexity $\equiv$ tree depth.

For unbounded degree:
complexity of polynomial evaluation should be taken into account
(We'll have branch nodes and computation nodes).

The unreasonable power of trees

Upper bounds:

- Tree depth $\leq$ Computation time of BSS machine.
- Depth n for any Boolean function on n variables.
- Depth $\tilde{O}(n^4)$ for Knapsack (Meyer auf der Heide'84, '88); Point location in arrangements of hyperplanes (Meiser'93).

Lower bounds:

- $\Omega(n^2)$ for Knapsack (Dobkin-Lipton'76) with linear decision trees ($d = 1$).
- Many other results by Yao, Grigoriev, Karpinski, Gabrielov, Vorobjov...

The unreasonable power of trees

Upper bounds:

- Tree depth $\leq$ Computation time of BSS machine.
- Depth n for any Boolean function on n variables.
- Depth $\tilde{O}(n^4)$ for Knapsack (Meyer auf der Heide'84, '88); Point location in arrangements of hyperplanes (Meiser'93).

Lower bounds:

- $\Omega(n^2)$ for Knapsack (Dobkin-Lipton'76) with linear decision trees ($d = 1$).
- Many other results by Yao, Grigoriev, Karpinski, Gabrielov, Vorobjov...

Machines over the reals with addition and order

Let $M = (\mathbb{R}, +, -, <)$.
Transfer "Theorem" [Fournier-Koiran'00]:
$\mathsf{P} = \mathsf{NP} \Leftrightarrow \mathsf{P}_M = \mathsf{NP}_M$.
Proof of $\Rightarrow$: Let A be an arbitrary problem in NP_M.

- A can be solved unconditionally in depth poly(n).
  Proof is not (efficiently) constructive.
- Construct "on the fly" the path followed by input $x \in \mathbb{R}^n$ from root to leaf with help of an oracle in NP.

Can we shrink the BSS model in another way?

Machines over the reals with addition and order

Let $M = (\mathbb{R}, +, -, <)$.
Transfer "Theorem" [Fournier-Koiran'00]:
$\mathrm{P} = \mathrm{NP} \Leftrightarrow \mathrm{P}_M = \mathrm{NP}_M$.
Proof of $\Rightarrow$: Let A be an arbitrary problem in NP_M.

- A can be solved unconditionally in depth poly(n).
  Proof is not (efficiently) constructive.
- Construct "on the fly" the path followed by input $x \in \mathbb{R}^n$ from root to leaf with help of an oracle in NP.

Can we shrink the BSS model in another way?

Machines over the reals with addition and order

Let $M = (\mathbb{R}, +, -, <)$.
Transfer "Theorem" [Fournier-Koiran'00]:
$\mathrm{P} = \mathrm{NP} \Leftrightarrow \mathrm{P}_M = \mathrm{NP}_M$.
Proof of $\Rightarrow$: Let A be an arbitrary problem in NP_M.

- A can be solved unconditionally in depth poly(n).
  Proof is not (efficiently) constructive.
- Construct "on the fly" the path followed by input $x \in \mathbb{R}^n$ from root to leaf with help of an oracle in NP.

Can we shrink the BSS model in another way?

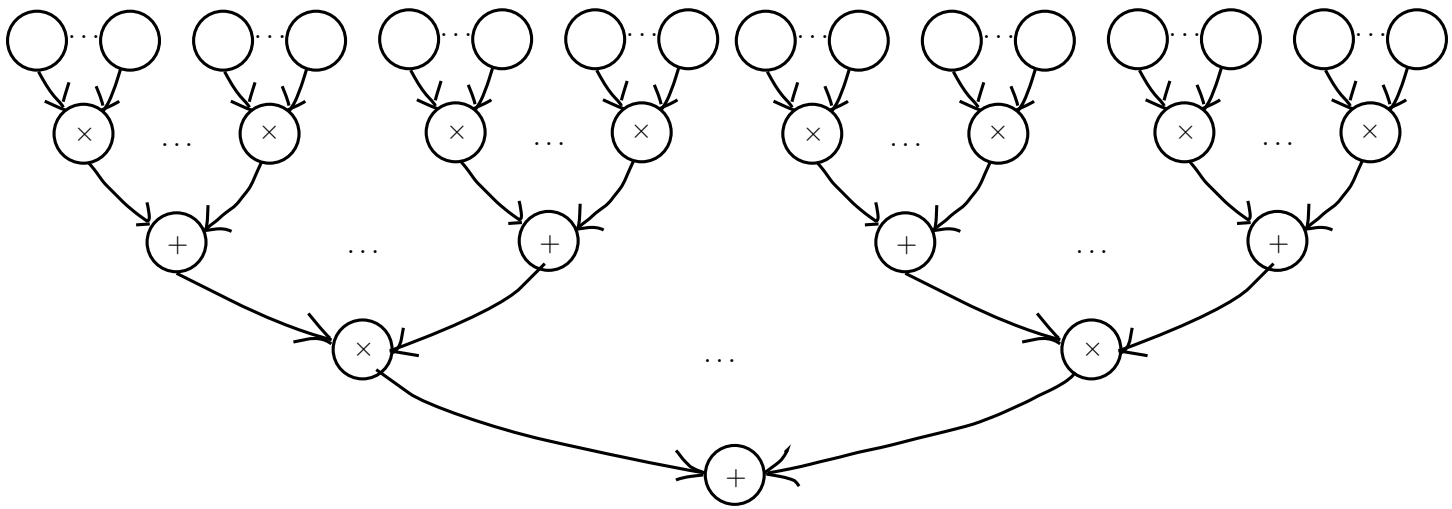
Machines over the reals with addition and order

Let $M = (\mathbb{R}, +, -, <)$.
Transfer "Theorem" [Fournier-Koiran'00]:
$\mathrm{P} = \mathrm{NP} \Leftrightarrow \mathrm{P}_M = \mathrm{NP}_M$.
Proof of $\Rightarrow$: Let A be an arbitrary problem in NP_M.

- A can be solved unconditionally in depth poly(n).
  Proof is not (efficiently) constructive.
- Construct "on the fly" the path followed by input $x \in \mathbb{R}^n$ from root to leaf with help of an oracle in NP.

Can we shrink the BSS model in another way?
Instead of removing multiplication,
we can remove branching.

Arithmetic circuits / straight-line programs



A depth 4 circuit.
The inputs are variables $(x_1, x_2, \ldots)$ or constants $(-1, 2, \sqrt{2}, \ldots)$,
the output is a polynomial.

VP = VNP?

- The permanent polynomial

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

  is VNP-complete if $\mathrm{char}(K) \neq 2$ (Valiant'79, Bürgisser'00).
- As a result,
  $\mathsf{VP}_K = \mathsf{VNP}_K \Leftrightarrow$ perm has polynomial size arithmetic circuits.

Connection to the BSS model

Let K be the field of real or complex numbers.
Transfer "Theorem" [Koiran-Perifel'07-09]:
$(\mathrm{P} = \mathrm{PSPACE}$ and $\mathrm{VP}_K = \mathrm{VNP}_K) \Rightarrow \mathrm{P}_K = \mathrm{NP}_K$.
In fact, $(\mathrm{P} = \mathrm{PSPACE}$ and $\mathrm{VP}_K = \mathrm{VNP}_K) \Leftrightarrow \mathrm{P}_K = \mathrm{PAR}_K$.

Proof sketch of $\Rightarrow$:
Construct "on the fly" the path followed by input $x \in \mathbb{R}^n$
from root to leaf with help of oracles in PSPACE and VNP.

What next?

Connection to the BSS model

Let K be the field of real or complex numbers.
Transfer "Theorem" [Koiran-Perifel'07-09]:
$(\mathrm{P} = \mathrm{PSPACE}$ and $\mathrm{VP}_K = \mathrm{VNP}_K) \Rightarrow \mathrm{P}_K = \mathrm{NP}_K$.
In fact, $(\mathrm{P} = \mathrm{PSPACE}$ and $\mathrm{VP}_K = \mathrm{VNP}_K) \Leftrightarrow \mathrm{P}_K = \mathrm{PAR}_K$.

Proof sketch of $\Rightarrow$:
Construct "on the fly" the path followed by input $x \in \mathbb{R}^n$
from root to leaf with help of oracles in PSPACE and VNP.

What next?

Connection to the BSS model

Let K be the field of real or complex numbers.
Transfer "Theorem" [Koiran-Perifel'07-09]:
$(\mathrm{P} = \mathrm{PSPACE}$ and $\mathrm{VP}_K = \mathrm{VNP}_K) \Rightarrow \mathrm{P}_K = \mathrm{NP}_K$.
In fact, $(\mathrm{P} = \mathrm{PSPACE}$ and $\mathrm{VP}_K = \mathrm{VNP}_K) \Leftrightarrow \mathrm{P}_K = \mathrm{PAR}_K$.

Proof sketch of $\Rightarrow$:
Construct "on the fly" the path followed by input $x \in \mathbb{R}^n$
from root to leaf with help of oracles in PSPACE and VNP.

What next?

Connection to the BSS model

Let K be the field of real or complex numbers.
Transfer "Theorem" [Koiran-Perifel'07-09]:
$(\mathrm{P} = \mathrm{PSPACE}$ and $\mathrm{VP}_K = \mathrm{VNP}_K) \Rightarrow \mathrm{P}_K = \mathrm{NP}_K$.
In fact, $(\mathrm{P} = \mathrm{PSPACE}$ and $\mathrm{VP}_K = \mathrm{VNP}_K) \Leftrightarrow \mathrm{P}_K = \mathrm{PAR}_K$.

Proof sketch of $\Rightarrow$:
Construct "on the fly" the path followed by input $x \in \mathbb{R}^n$
from root to leaf with help of oracles in PSPACE and VNP.

What next?

- It seems reasonable to focus on VP versus VNP.

Connection to the BSS model

Let K be the field of real or complex numbers.
Transfer "Theorem" [Koiran-Perifel'07-09]:
$(\mathrm{P} = \mathrm{PSPACE}$ and $\mathrm{VP}_K = \mathrm{VNP}_K) \Rightarrow \mathrm{P}_K = \mathrm{NP}_K$.
In fact, $(\mathrm{P} = \mathrm{PSPACE}$ and $\mathrm{VP}_K = \mathrm{VNP}_K) \Leftrightarrow \mathrm{P}_K = \mathrm{PAR}_K$.

Proof sketch of $\Rightarrow$:
Construct "on the fly" the path followed by input $x \in \mathbb{R}^n$
from root to leaf with help of oracles in PSPACE and VNP.

What next?

- It seems reasonable to focus on VP versus VNP.
- But it's a difficult question...

Connection to the BSS model

Let K be the field of real or complex numbers.
Transfer "Theorem" [Koiran-Perifel'07-09]:
$(\mathrm{P} = \mathrm{PSPACE}$ and $\mathrm{VP}_K = \mathrm{VNP}_K) \Rightarrow \mathrm{P}_K = \mathrm{NP}_K$.
In fact, $(\mathrm{P} = \mathrm{PSPACE}$ and $\mathrm{VP}_K = \mathrm{VNP}_K) \Leftrightarrow \mathrm{P}_K = \mathrm{PAR}_K$.

Proof sketch of $\Rightarrow$:
Construct "on the fly" the path followed by input $x \in \mathbb{R}^n$
from root to leaf with help of oracles in PSPACE and VNP.

What next?

- It seems reasonable to focus on VP versus VNP.
- But it's a difficult question...
- Let's shrink the arithmetic circuit model!

Bounded depth circuits

- Model: arithmetic circuits of depth 3 ($\Sigma\Pi\Sigma$) over K, char(K)=0.
- Target polynomial: SYM_n^d, the symmetric polynomial of degree d in n variables.

Theorem [Nisan-Wigderson'96]: Any *homogeneous* depth 3 circuit computing SYM_n^{2d} requires size $\Omega((n/4d)^d)$.
Complexity measure: dimension of space of partial derivatives.

- **Remark** [Ben-Or]: Homogeneity assumption is necessary.
- Recent progress:
  shifted partial derivatives (Neeraj Kayal 2012 + many others).
  Arbitrary constant depth (Limaye, Srinivasan, Tavenas, **June 2021**).

How far to general (unbounded depth) circuit lower bounds?

Bounded depth circuits

- Model: arithmetic circuits of depth 3 ($\Sigma\Pi\Sigma$) over K, char(K)=0.
- Target polynomial: SYM_n^d, the symmetric polynomial of degree d in n variables.

Theorem [Nisan-Wigderson'96]: Any *homogeneous* depth 3 circuit computing SYM_n^{2d} requires size $\Omega((n/4d)^d)$.
Complexity measure: dimension of space of partial derivatives.

- **Remark** [Ben-Or]: Homogeneity assumption is necessary.
- Recent progress:
  shifted partial derivatives (Neeraj Kayal 2012 + many others).
  Arbitrary constant depth (Limaye, Srinivasan, Tavenas, **June 2021**).

How far to general (unbounded depth) circuit lower bounds?

Bounded depth circuits

- Model: arithmetic circuits of depth 3 ($\Sigma\Pi\Sigma$) over K, char(K)=0.
- Target polynomial: SYM_n^d, the symmetric polynomial of degree d in n variables.

Theorem [Nisan-Wigderson'96]: Any *homogeneous* depth 3 circuit computing SYM_n^{2d} requires size $\Omega((n/4d)^d)$.
Complexity measure: dimension of space of partial derivatives.

- **Remark** [Ben-Or]: Homogeneity assumption is necessary.
- Recent progress:
  shifted partial derivatives (Neeraj Kayal 2012 + many others).
  Arbitrary constant depth (Limaye, Srinivasan, Tavenas, **June 2021**).

How far to general (unbounded depth) circuit lower bounds?

Reduction to depth 4

Theorem [Agrawal-Vinay'08,Koiran'12,Tavenas'13]:
Let $C(x_1, \ldots, x_n)$ be a circuit of degree d and size s, where $d, s = n^{O(1)}$.
There is an equivalent depth 4 circuit of size $n^{O(\sqrt{n})}$.

- By shifted partials, this is essentially optimal [Fournier,Limaye,Malod,Srinivasan'15].
- Also, reduction to depth 3 [Gupta, Kamath, Kayal, Saptharishi'13].
  Does not preserve homogeneity.

Wanted: $n^{\omega(\sqrt{n})}$ lower bound for circuits of depth 3 or 4....

Reduction to depth 4

Theorem [Agrawal-Vinay'08,Koiran'12,Tavenas'13]:
Let $C(x_1, \ldots, x_n)$ be a circuit of degree d and size s, where $d, s = n^{O(1)}$.
There is an equivalent depth 4 circuit of size $n^{O(\sqrt{n})}$.

- By shifted partials, this is essentially optimal [Fournier,Limaye,Malod,Srinivasan'15].
- Also, reduction to depth 3 [Gupta, Kamath, Kayal, Saptharishi'13].
  Does not preserve homogeneity.

Wanted: $n^{\omega(\sqrt{n})}$ lower bound for circuits of depth 3 or 4....

Reduction to depth 4

Theorem [Agrawal-Vinay'08,Koiran'12,Tavenas'13]:
Let $C(x_1, \ldots, x_n)$ be a circuit of degree d and size s, where $d, s = n^{O(1)}$.
There is an equivalent depth 4 circuit of size $n^{O(\sqrt{n})}$.

- By shifted partials, this is essentially optimal [Fournier,Limaye,Malod,Srinivasan'15].
- Also, reduction to depth 3 [Gupta, Kamath, Kayal, Saptharishi'13].
  Does not preserve homogeneity.

Wanted: $n^{\omega(\sqrt{n})}$ lower bound for circuits of depth 3 or 4....

Reduction to depth 4

Theorem [Agrawal-Vinay'08,Koiran'12,Tavenas'13]:
Let $C(x_1, \ldots, x_n)$ be a circuit of degree d and size s, where $d, s = n^{O(1)}$.
There is an equivalent depth 4 circuit of size $n^{O(\sqrt{n})}$.

- By shifted partials, this is essentially optimal [Fournier,Limaye,Malod,Srinivasan'15].
- Also, reduction to depth 3 [Gupta, Kamath, Kayal, Saptharishi'13].
  Does not preserve homogeneity.

Wanted: $n^{\omega(\sqrt{n})}$ lower bound for circuits of depth 3 or 4....
But we are currently stuck!

The Real τ-Conjecture

Conjecture: Consider $f(X) = \sum_{i=1}^{k} \prod_{j=1}^{m} f_{ij}(X)$,
where the f_{ij} are t-sparse.
If f is nonzero, its number of **real roots** is polynomial in kmt.
Theorem: If the conjecture is true then VP $\neq$ VNP.
Remarks:

- The expression for f is a univariate circuit of depth 4.
- Case $k = 1$ of the conjecture follows from Descartes' rule (t monomials $\Rightarrow$ at most $2t - 1$ real roots).
- By expanding the products, f has at most $2kt^m - 1$ zeros.
- $k = 2$ is open. **A take-home problem:** how many real roots for $fg + 1$? Descartes' bound is $O(t^2)$ but true bound could be $O(t)$.

The Real τ-Conjecture

Conjecture: Consider $f(X) = \sum_{i=1}^{k} \prod_{j=1}^{m} f_{ij}(X)$,
where the f_{ij} are t-sparse.
If f is nonzero, its number of **real roots** is polynomial in kmt.
Theorem: If the conjecture is true then VP $\neq$ VNP.
Remarks:

- The expression for f is a univariate circuit of depth 4.
- Case $k = 1$ of the conjecture follows from Descartes' rule (t monomials $\Rightarrow$ at most $2t - 1$ real roots).
- By expanding the products, f has at most $2kt^m - 1$ zeros.
- $k = 2$ is open. **A take-home problem:** how many real roots for $fg + 1$? Descartes' bound is $O(t^2)$ but true bound could be $O(t)$.

The Real τ-Conjecture

Conjecture: Consider $f(X) = \sum_{i=1}^{k} \prod_{j=1}^{m} f_{ij}(X)$,
where the f_{ij} are t-sparse.
If f is nonzero, its number of **real roots** is polynomial in kmt.
Theorem: If the conjecture is true then VP $\neq$ VNP.
Remarks:

- The expression for f is a univariate circuit of depth 4.
- Case $k = 1$ of the conjecture follows from Descartes' rule (t monomials $\Rightarrow$ at most $2t - 1$ real roots).
- By expanding the products, f has at most $2kt^m - 1$ zeros.
- $k = 2$ is open. **A take-home problem:** how many real roots for $fg + 1$? Descartes' bound is $O(t^2)$ but true bound could be $O(t)$.

The Real τ-Conjecture

Conjecture: Consider $f(X) = \sum_{i=1}^{k} \prod_{j=1}^{m} f_{ij}(X)$,
where the f_{ij} are t-sparse.
If f is nonzero, its number of **real roots** is polynomial in kmt.
Theorem: If the conjecture is true then VP $\neq$ VNP.
Remarks:

- The expression for f is a univariate circuit of depth 4.
- Case $k = 1$ of the conjecture follows from Descartes' rule (t monomials $\Rightarrow$ at most $2t - 1$ real roots).
- By expanding the products, f has at most $2kt^m - 1$ zeros.
- $k = 2$ is open. **A take-home problem:** how many real roots for $fg + 1$? Descartes' bound is $O(t^2)$ but true bound could be $O(t)$.

The Real τ-Conjecture

Conjecture: Consider $f(X) = \sum_{i=1}^{k} \prod_{j=1}^{m} f_{ij}(X)$,
where the f_{ij} are t-sparse.
If f is nonzero, its number of **real roots** is polynomial in kmt.
Theorem: If the conjecture is true then VP $\neq$ VNP.
Remarks:

- The expression for f is a univariate circuit of depth 4.
- Case $k = 1$ of the conjecture follows from Descartes' rule (t monomials $\Rightarrow$ at most $2t - 1$ real roots).
- By expanding the products, f has at most $2kt^m - 1$ zeros.
- $k = 2$ is open. **A take-home problem:** how many real roots for $fg + 1$? Descartes' bound is $O(t^2)$ but true bound could be $O(t)$.

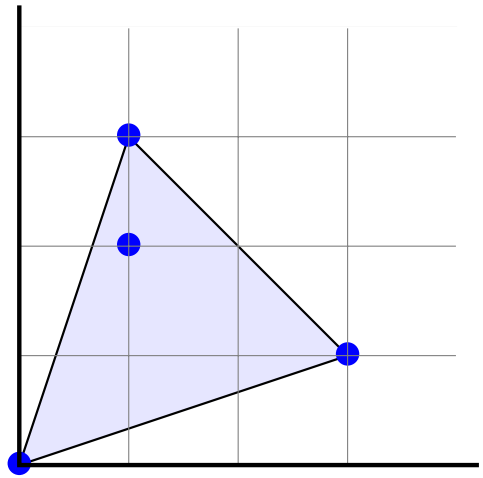
Real τ-Conjecture $\Rightarrow$ VP $\neq$ VNP

Very rough proof sketch:

1. Start with polynomial with many real roots such as: $f(X) = \prod_{i=1}^{2^n}(X - i)$.
2. Assuming VP = VNP, reduce it to depth 4: $f(X) = \sum_{i=1}^{k} \prod_{j=1}^{m} f_{ij}(X)$, where the f_{ij} are t-sparse, and $k, m, t = 2^{o(n)}$.
3. This is a contradiction with the real τ-conjecture.

τ-conjecture for Newton polygons

- Newton polygon for $f(X, Y) = 1 + 2X^3Y + XY^2 + XY^3$:



- Real roots of $f(X)$ replaced by Newton polygons of $f(X, Y)$.
- Similar conjecture with similar consequences (K., Portier, Tavenas, Thomassé'15).
- The real τ-conjecture implies the τ-conjecture for Newton polygons (Hrubes'19).
- Nontrivial $O(t^{4/3})$ bound for size (number of vertices) of the Newton polygon of $f(X, Y)g(X, Y) + 1$.

A drastic simplification: Lower bounds for univariate polynomials

Model: $f(x) = \sum_{i=1}^{s} p_i(x)^{e_i}$ where $\deg(p_i)$ is small (say, ≤ 2).

Lower bound theorem [Kayal, Koiran, Pecatte, Saha'15]:
$s = \Omega(\sqrt{d})$ for the degree d polynomials

$$\sum_{i=1}^{\sqrt{d/2}} (x - a_i)^d \text{ and } [(x - a_1)(x - a_2)]^{d/2}$$

where the a_i are distinct.

A drastic simplification: Lower bounds for univariate polynomials

Model: $f(x) = \sum_{i=1}^{s} p_i(x)^{e_i}$ where $\deg(p_i)$ is small (say, ≤ 2).

Lower bound theorem [Kayal, Koiran, Pecatte, Saha'15]:
$s = \Omega(\sqrt{d})$ for the degree d polynomials

$$\sum_{i=1}^{\sqrt{d/2}} (x - a_i)^d \text{ and } [(x - a_1)(x - a_2)]^{d/2}$$

where the a_i are distinct.

- Wanted: $s = \Omega(d)$ for some explicit family of polynomials.

A drastic simplification: Lower bounds for univariate polynomials

Model: $f(x) = \sum_{i=1}^{s} p_i(x)^{e_i}$ where $\deg(p_i)$ is small (say, ≤ 2).

Lower bound theorem [Kayal, Koiran, Pecatte, Saha'15]:
$s = \Omega(\sqrt{d})$ for the degree d polynomials

$$\sum_{i=1}^{\sqrt{d/2}} (x - a_i)^d \text{ and } [(x - a_1)(x - a_2)]^{d/2}$$

where the a_i are distinct.

- Wanted: $s = \Omega(d)$ for some explicit family of polynomials.
- This is open even for $\deg(p_i) = 1$!

A drastic simplification:
Lower bounds for univariate polynomials

Model: $f(x) = \sum_{i=1}^{s} p_i(x)^{e_i}$ where $\deg(p_i)$ is small (say, ≤ 2).

Lower bound theorem [Kayal, Koiran, Pecatte, Saha'15]:
$s = \Omega(\sqrt{d})$ for the degree d polynomials

$$\sum_{i=1}^{\sqrt{d/2}} (x - a_i)^d \text{ and } [(x - a_1)(x - a_2)]^{d/2}$$

where the a_i are distinct.

- Wanted: $s = \Omega(d)$ for some explicit family of polynomials.
- This is open even for $\deg(p_i) = 1$!
- How can we possibly shrink that model??

Shrinking the field: sums of powers of real affine functions

Model: $f(x) = \sum_{i=1}^{s} \alpha_i (x - a_i)^{e_i}$ where $a_i, \alpha_i \in \mathbb{R}$.

Lower bound theorem [Garcia-Marco, Koiran'17]:
$s = \Omega(d)$ for the degree d polynomials

$$H_1(x) = (x+1)^{d+1} - x^{d+1}, H_2(x) = \sum_{i=1}^{d/4} \alpha_i (x - a_i)^d$$

where $\alpha_i \neq 0$ and the a_i are distinct (and $e_i \leq d$ for H_1).

- Proof by Birkhoff interpolation.
- What about the complex field?

Sums of powers of complex affine functions

Open problem: Lower bound better than $\Omega(\sqrt{d})$ in the model

$$f(x) = \sum_{i=1}^{s} \alpha_i (x - a_i)^{e_i}$$

over the field of complex numbers for some explicit polynomial.

- An upper bound: If ξ is a primitive k-th root of unity,

$$\sum_{j=1}^{k} \xi^j (x + \xi^j)^d = \sum_{\substack{i \equiv -1 \pmod{k} \\ 0 \le i \le d}} k \binom{d}{i} x^{d-i}.$$

- Rules out $H_2(x) = \sum_{i=1}^{k} \alpha_i (x - a_i)^d$ as "hard polynomial" if $a_i, \alpha_i \in \mathbb{C}$ may be arbitrary.
- $H_1(x) = (x+1)^{d+1} - x^{d+1}$ may still be hard.

Small degree: sums of cubes of linear forms

$$f(x_1, \ldots, x_N) = \sum_{i=1}^{r} \ell_i(x_1, \ldots, x_N)^3$$

- Smallest possible r is the *symmetric tensor rank* of f (also known as *Waring rank*).
- It's $\Theta(N^2)$ for generic f; exact value known for any degree (Alexander-Hirschowitz theorem).
- Longstanding open problem (Strassen?): find explicit f with superlinear (symmetric) tensor rank.
- As a plausible candidate we have the matrix multiplication tensor ($N = 3n^2$):

$$f = \sum_{i,j,k=1}^{n} X_{ij} Y_{jk} Z_{ki}.$$

Small degree: sums of cubes of linear forms

$$f(x_1, \ldots, x_N) = \sum_{i=1}^{r} \ell_i(x_1, \ldots, x_N)^3$$

- Smallest possible r is the *symmetric tensor rank* of f (also known as *Waring rank*).
- It's $\Theta(N^2)$ for generic f; exact value known for any degree (Alexander-Hirschowitz theorem).
- Longstanding open problem (Strassen?): find explicit f with superlinear (symmetric) tensor rank.
- As a plausible candidate we have the matrix multiplication tensor ($N = 3n^2$):

$$f = \sum_{i,j,k=1}^{n} X_{ij} Y_{jk} Z_{ki}.$$