

# **Recent Advances in the Blum-Shub-Smale Model**

Peter Bürgisser

Universität Paderborn

Spring School in Montagnac, June 2005

# Contents

- I. General introduction to BSS-model
- II. Additive BSS-model
- III. Unrestricted BSS-model over  $\mathbb{C}$
- IV. Unrestricted BSS-model over  $\mathbb{R}$

# **I. The Blum-Shub-Smale Model**

# **I. The Blum-Shub-Smale Model**

Motivation: to provide a theoretical foundation for numerical computations with real numbers in the sense of theoretical computer science

# I. The Blum-Shub-Smale Model

Motivation: to provide a theoretical foundation for numerical computations with real numbers in the sense of theoretical computer science

- extension of classical theory of NP-completeness to computations over  $\mathbb{R}$

# I. The Blum-Shub-Smale Model

Motivation: to provide a theoretical foundation for numerical computations with real numbers in the sense of theoretical computer science

- extension of classical theory of NP-completeness to computations over  $\mathbb{R}$
- real numbers are basic entities (infinite precision assumed)

# I. The Blum-Shub-Smale Model

Motivation: to provide a theoretical foundation for numerical computations with real numbers in the sense of theoretical computer science

- extension of classical theory of NP-completeness to computations over  $\mathbb{R}$
- real numbers are basic entities (infinite precision assumed)
- basic computational step: arithmetic operation or  $\leq$ -test at unit cost

# I. The Blum-Shub-Smale Model

Motivation: to provide a theoretical foundation for numerical computations with real numbers in the sense of theoretical computer science

- extension of classical theory of NP-completeness to computations over  $\mathbb{R}$
- real numbers are basic entities (infinite precision assumed)
- basic computational step: arithmetic operation or  $\leq$ -test at unit cost
- inputs and outputs are vectors in  $\mathbb{R}^\infty$ ; if  $x \in \mathbb{R}^n$  then  $\text{size}(x) = n$

# I. The Blum-Shub-Smale Model

Motivation: to provide a theoretical foundation for numerical computations with real numbers in the sense of theoretical computer science

- extension of classical theory of NP-completeness to computations over  $\mathbb{R}$
- real numbers are basic entities (infinite precision assumed)
- basic computational step: arithmetic operation or  $\leq$ -test at unit cost
- inputs and outputs are vectors in  $\mathbb{R}^\infty$ ; if  $x \in \mathbb{R}^n$  then  $\text{size}(x) = n$
- decision problems formalized as subsets of  $\mathbb{R}^\infty = \sqcup_{n \geq 0} \mathbb{R}^n$ ; uniform model

# I. The Blum-Shub-Smale Model

Motivation: to provide a theoretical foundation for numerical computations with real numbers in the sense of theoretical computer science

- extension of classical theory of NP-completeness to computations over  $\mathbb{R}$
- real numbers are basic entities (infinite precision assumed)
- basic computational step: arithmetic operation or  $\leq$ -test at unit cost
- inputs and outputs are vectors in  $\mathbb{R}^\infty$ ; if  $x \in \mathbb{R}^n$  then  $\text{size}(x) = n$
- decision problems formalized as subsets of  $\mathbb{R}^\infty = \sqcup_{n \geq 0} \mathbb{R}^n$ ; uniform model
- instead of  $\mathbb{R}$  can take any field; if unordered allow only  $=$ -test

# I. The Blum-Shub-Smale Model

Motivation: to provide a theoretical foundation for numerical computations with real numbers in the sense of theoretical computer science

- extension of classical theory of NP-completeness to computations over  $\mathbb{R}$
- real numbers are basic entities (infinite precision assumed)
- basic computational step: arithmetic operation or  $\leq$ -test at unit cost
- inputs and outputs are vectors in  $\mathbb{R}^\infty$ ; if  $x \in \mathbb{R}^n$  then  $\text{size}(x) = n$
- decision problems formalized as subsets of  $\mathbb{R}^\infty = \sqcup_{n \geq 0} \mathbb{R}^n$ ; uniform model
- instead of  $\mathbb{R}$  can take any field; if unordered allow only  $=$ -test
- over  $\mathbb{F}_2$  classical theory is recovered (Turing model)

# I. The Blum-Shub-Smale Model

Motivation: to provide a theoretical foundation for numerical computations with real numbers in the sense of theoretical computer science

- extension of classical theory of NP-completeness to computations **over**  $\mathbb{R}$
- real numbers are basic entities (infinite precision assumed)
- basic computational step: arithmetic operation or  $\leq$ -test at unit cost
- inputs and outputs are vectors in  $\mathbb{R}^\infty$ ; if  $x \in \mathbb{R}^n$  then  $\text{size}(x) = n$
- decision problems formalized as subsets of  $\mathbb{R}^\infty = \sqcup_{n \geq 0} \mathbb{R}^n$ ; uniform model
- instead of  $\mathbb{R}$  can take any field; if unordered allow only  $=$ -test
- **over**  $\mathbb{F}_2$  classical theory is recovered (Turing model)
- situation **over field**  $\mathbb{C}$  is mathematically interesting

# I. The Blum-Shub-Smale Model

Motivation: to provide a theoretical foundation for numerical computations with real numbers in the sense of theoretical computer science

- extension of classical theory of NP-completeness to computations **over**  $\mathbb{R}$
- real numbers are basic entities (infinite precision assumed)
- basic computational step: arithmetic operation or  $\leq$ -test at unit cost
- inputs and outputs are vectors in  $\mathbb{R}^\infty$ ; if  $x \in \mathbb{R}^n$  then  $\text{size}(x) = n$
- decision problems formalized as subsets of  $\mathbb{R}^\infty = \sqcup_{n \geq 0} \mathbb{R}^n$ ; uniform model
- instead of  $\mathbb{R}$  can take any field; if unordered allow only  $=$ -test
- **over**  $\mathbb{F}_2$  classical theory is recovered (Turing model)
- situation **over field**  $\mathbb{C}$  is mathematically interesting
- **Additive Model**: no multiplications or divisions allowed

# Algebraic Circuits (1)

## Algebraic Circuits (1)

- Originally: formalization by machines over  $\mathbb{R}$

## Algebraic Circuits (1)

- Originally: formalization by [machines over  \$\mathbb{R}\$](#)
- Poizat: characterization by P-uniform families of algebraic circuits

## Algebraic Circuits (1)

- Originally: formalization by [machines over  \$\mathbb{R}\$](#)
- Poizat: characterization by P-uniform families of algebraic circuits
- An [algebraic circuit](#)  $C$  is an acyclic digraph with the following type of nodes:  
input, output, constant, arithmetic, selection (von zur Gathen)

## Algebraic Circuits (1)

- Originally: formalization by [machines over  \$\mathbb{R}\$](#)
- Poizat: characterization by P-uniform families of algebraic circuits
- An [algebraic circuit](#)  $C$  is an acyclic digraph with the following type of nodes: input, output, constant, arithmetic, selection (von zur Gathen)
- Semantic of selection node: on input  $(s, x, y)$  return  $x$  if  $s \geq 0$  and  $y$  otherwise

## Algebraic Circuits (1)

- Originally: formalization by machines over  $\mathbb{R}$
- Poizat: characterization by P-uniform families of algebraic circuits
- An algebraic circuit  $C$  is an acyclic digraph with the following type of nodes: input, output, constant, arithmetic, selection (von zur Gathen)
- Semantic of selection node: on input  $(s, x, y)$  return  $x$  if  $s \geq 0$  and  $y$  otherwise
- The size of  $C$  is the number of its nodes

## Algebraic Circuits (1)

- Originally: formalization by machines over  $\mathbb{R}$
- Poizat: characterization by P-uniform families of algebraic circuits
- An algebraic circuit  $C$  is an acyclic digraph with the following type of nodes: input, output, constant, arithmetic, selection (von zur Gathen)
- Semantic of selection node: on input  $(s, x, y)$  return  $x$  if  $s \geq 0$  and  $y$  otherwise
- The size of  $C$  is the number of its nodes
- The depth of  $C$  is the maximal length of a directed path in  $C$

## Algebraic Circuits (1)

- Originally: formalization by machines over  $\mathbb{R}$
- Poizat: characterization by P-uniform families of algebraic circuits
- An algebraic circuit  $C$  is an acyclic digraph with the following type of nodes: input, output, constant, arithmetic, selection (von zur Gathen)
- Semantic of selection node: on input  $(s, x, y)$  return  $x$  if  $s \geq 0$  and  $y$  otherwise
- The size of  $C$  is the number of its nodes
- The depth of  $C$  is the maximal length of a directed path in  $C$
- Apart from the labelling of the constant nodes with reals, algebraic circuits are discrete objects!

## **Algebraic Circuits (2)**

## Algebraic Circuits (2)

- A family  $\mathcal{C} := (C_n)_{n \geq 0}$  of algebraic circuits is called **P-uniform** if there exists a fixed number  $m$  of real constants  $\alpha_1, \dots, \alpha_m$  such that each  $C_n$  has exactly  $m$  constant nodes labelled by these constants and such that there exists a poly time Turing machine that on input  $(n, i)$  computes the  $i$ -th node of  $C_n$

## Algebraic Circuits (2)

- A family  $\mathcal{C} := (C_n)_{n \geq 0}$  of algebraic circuits is called **P-uniform** if there exists a fixed number  $m$  of real constants  $\alpha_1, \dots, \alpha_m$  such that each  $C_n$  has exactly  $m$  constant nodes labelled by these constants and such that there exists a poly time Turing machine that on input  $(n, i)$  computes the  $i$ -th node of  $C_n$
- $\mathcal{C}$  is called **constant-free** if 0 and 1 are the only constants

## Algebraic Circuits (2)

- A family  $\mathcal{C} := (C_n)_{n \geq 0}$  of algebraic circuits is called **P-uniform** if there exists a fixed number  $m$  of real constants  $\alpha_1, \dots, \alpha_m$  such that each  $C_n$  has exactly  $m$  constant nodes labelled by these constants and such that there exists a poly time Turing machine that on input  $(n, i)$  computes the  $i$ -th node of  $C_n$
- $\mathcal{C}$  is called **constant-free** if 0 and 1 are the only constants
- A family  $\mathcal{C}$  **computes** a function  $f: \mathbb{R}^\infty \rightarrow \mathbb{R}^\infty$  (if no division by zero occurs)

## Algebraic Circuits (2)

- A family  $\mathcal{C} := (C_n)_{n \geq 0}$  of algebraic circuits is called **P-uniform** if there exists a fixed number  $m$  of real constants  $\alpha_1, \dots, \alpha_m$  such that each  $C_n$  has exactly  $m$  constant nodes labelled by these constants and such that there exists a poly time Turing machine that on input  $(n, i)$  computes the  $i$ -th node of  $C_n$
- $\mathcal{C}$  is called **constant-free** if 0 and 1 are the only constants
- A family  $\mathcal{C}$  **computes** a function  $f: \mathbb{R}^\infty \rightarrow \mathbb{R}^\infty$  (if no division by zero occurs)
- The family  $\mathcal{C}$  **decides**  $S \subseteq \mathbb{R}^\infty$  if it computes its characteristic function

## Algebraic Circuits (2)

- A family  $\mathcal{C} := (C_n)_{n \geq 0}$  of algebraic circuits is called **P-uniform** if there exists a fixed number  $m$  of real constants  $\alpha_1, \dots, \alpha_m$  such that each  $C_n$  has exactly  $m$  constant nodes labelled by these constants and such that there exists a poly time Turing machine that on input  $(n, i)$  computes the  $i$ -th node of  $C_n$
- $\mathcal{C}$  is called **constant-free** if 0 and 1 are the only constants
- A family  $\mathcal{C}$  **computes** a function  $f: \mathbb{R}^\infty \rightarrow \mathbb{R}^\infty$  (if no division by zero occurs)
- The family  $\mathcal{C}$  **decides**  $S \subseteq \mathbb{R}^\infty$  if it computes its characteristic function
- A relation  $R \subseteq \mathbb{R}^\infty \times \mathbb{R}^\infty$  is called **balanced** if there exists a polynomial  $p$  such that, for all  $x \in \mathbb{R}^n$ ,  $(x, y) \in R$  implies  $y \in \mathbb{R}^{p(n)}$

# Complexity Classes

## Complexity Classes

- The class  $P_{\mathbb{R}}$  consists of all  $S \subseteq \mathbb{R}^{\infty}$  decidable by a P-uniform family of algebraic circuits  $(C_n)_{n \geq 0}$  of size polynomial in  $n$

## Complexity Classes

- The class  $P_{\mathbb{R}}$  consists of all  $S \subseteq \mathbb{R}^{\infty}$  decidable by a P-uniform family of algebraic circuits  $(C_n)_{n \geq 0}$  of size polynomial in  $n$
- The class  $NP_{\mathbb{R}}$  consists of all  $S \subseteq \mathbb{R}^{\infty}$  for which there exists a balanced relation  $R \subseteq \mathbb{R}^{\infty} \times \mathbb{R}^{\infty}$  in  $P_{\mathbb{R}}$  such that

$$\forall x \in \mathbb{R}^{\infty} (x \in S \Leftrightarrow \exists y \in \mathbb{R}^{\infty} (x, y) \in R)$$

## Complexity Classes

- The class  $P_{\mathbb{R}}$  consists of all  $S \subseteq \mathbb{R}^{\infty}$  decidable by a P-uniform family of algebraic circuits  $(C_n)_{n \geq 0}$  of size polynomial in  $n$
- The class  $NP_{\mathbb{R}}$  consists of all  $S \subseteq \mathbb{R}^{\infty}$  for which there exists a balanced relation  $R \subseteq \mathbb{R}^{\infty} \times \mathbb{R}^{\infty}$  in  $P_{\mathbb{R}}$  such that

$$\forall x \in \mathbb{R}^{\infty} (x \in S \Leftrightarrow \exists y \in \mathbb{R}^{\infty} (x, y) \in R)$$

- The **counting class**  $\#P_{\mathbb{R}}$  (Meer) consists of all corresponding functions  $f: \mathbb{R}^{\infty} \rightarrow \mathbb{N} \cup \{\infty\}$  such that

$$f(x) = |\{y \in \mathbb{R}^{\infty} \mid (x, y) \in R\}|$$

## Complexity Classes

- The class  $\mathbf{P}_{\mathbb{R}}$  consists of all  $S \subseteq \mathbb{R}^{\infty}$  decidable by a P-uniform family of algebraic circuits  $(C_n)_{n \geq 0}$  of size polynomial in  $n$
- The class  $\mathbf{NP}_{\mathbb{R}}$  consists of all  $S \subseteq \mathbb{R}^{\infty}$  for which there exists a balanced relation  $R \subseteq \mathbb{R}^{\infty} \times \mathbb{R}^{\infty}$  in  $\mathbf{P}_{\mathbb{R}}$  such that

$$\forall x \in \mathbb{R}^{\infty} (x \in S \Leftrightarrow \exists y \in \mathbb{R}^{\infty} (x, y) \in R)$$

- The **counting class**  $\#\mathbf{P}_{\mathbb{R}}$  (Meer) consists of all corresponding functions  $f: \mathbb{R}^{\infty} \rightarrow \mathbb{N} \cup \{\infty\}$  such that

$$f(x) = |\{y \in \mathbb{R}^{\infty} \mid (x, y) \in R\}|$$

- The class  $\mathbf{PAR}_{\mathbb{R}}$  (Cucker) consists of all  $S \subseteq \mathbb{R}^{\infty}$  decidable by a P-uniform family  $(C_n)_{n \geq 0}$  of algebraic circuits of depth polynomial in  $n$

# **Analogy with Classical Complexity Classes**

## Analogy with Classical Complexity Classes

- The definitions of  $P_{\mathbb{R}}$ ,  $NP_{\mathbb{R}}$ , and  $PAR_{\mathbb{R}}$  are analogous to the definitions of  $P$ ,  $NP$ , and  $PSPACE$  in terms of Boolean circuits

## Analogy with Classical Complexity Classes

- The definitions of  $P_{\mathbb{R}}$ ,  $NP_{\mathbb{R}}$ , and  $PAR_{\mathbb{R}}$  are analogous to the definitions of  $P$ ,  $NP$ , and  $PSPACE$  in terms of Boolean circuits
- The classical counting complexity class  $\#P$  was introduced by Valiant in 1979

## Analogy with Classical Complexity Classes

- The definitions of  $P_{\mathbb{R}}$ ,  $NP_{\mathbb{R}}$ , and  $PAR_{\mathbb{R}}$  are analogous to the definitions of  $P$ ,  $NP$ , and  $PSPACE$  in terms of Boolean circuits
- The classical counting complexity class  $\#P$  was introduced by Valiant in 1979
- The canonical  $\#P$ -complete problem is **#SAT**: Given a Boolean formula, count the number of its satisfying assignments

## Analogy with Classical Complexity Classes

- The definitions of  $P_{\mathbb{R}}$ ,  $NP_{\mathbb{R}}$ , and  $PAR_{\mathbb{R}}$  are analogous to the definitions of  $P$ ,  $NP$ , and  $PSPACE$  in terms of Boolean circuits
- The classical counting complexity class  $\#P$  was introduced by Valiant in 1979
- The canonical  $\#P$ -complete problem is  **$\#SAT$** : Given a Boolean formula, count the number of its satisfying assignments
- Important  $\#P$ -hard problems are known in different areas, such as geometry (volume of polyhedra), knot theory (Jones polynomial), statistical physics (partition function), and network reliability

# Basic Completeness Results

## Basic Completeness Results

- Efficient algorithms of real algebraic geometry imply that

$$P_{\mathbb{R}} \subseteq NP_{\mathbb{R}} \subseteq \#P_{\mathbb{R}} \subseteq FPAR_{\mathbb{R}}$$

## Basic Completeness Results

- Efficient algorithms of real algebraic geometry imply that

$$P_{\mathbb{R}} \subseteq NP_{\mathbb{R}} \subseteq \#P_{\mathbb{R}} \subseteq FPAR_{\mathbb{R}}$$

- We introduce notions of polynomial **reductions** (many-one/Turing) between computational problems and obtain the corresponding notions of completeness

## Basic Completeness Results

- Efficient algorithms of real algebraic geometry imply that

$$P_{\mathbb{R}} \subseteq NP_{\mathbb{R}} \subseteq \#P_{\mathbb{R}} \subseteq FPAR_{\mathbb{R}}$$

- We introduce notions of polynomial **reductions** (many-one/Turing) between computational problems and obtain the corresponding notions of completeness
- **FEAS<sub>ℝ</sub>**: Given a multivariate real polynomial, decide whether it has a real root

## Basic Completeness Results

- Efficient algorithms of real algebraic geometry imply that

$$P_{\mathbb{R}} \subseteq NP_{\mathbb{R}} \subseteq \#P_{\mathbb{R}} \subseteq FPAR_{\mathbb{R}}$$

- We introduce notions of polynomial **reductions** (many-one/Turing) between computational problems and obtain the corresponding notions of completeness
- **FEAS<sub>ℝ</sub>**: Given a multivariate real polynomial, decide whether it has a real root
- **#FEAS<sub>ℝ</sub>**: Given a multivariate real polynomial, count its real roots (returning  $\infty$  if this number is not finite)

## Basic Completeness Results

- Efficient algorithms of real algebraic geometry imply that

$$P_{\mathbb{R}} \subseteq NP_{\mathbb{R}} \subseteq \#P_{\mathbb{R}} \subseteq FPAR_{\mathbb{R}}$$

- We introduce notions of polynomial **reductions** (many-one/Turing) between computational problems and obtain the corresponding notions of completeness
- **FEAS<sub>ℝ</sub>**: Given a multivariate real polynomial, decide whether it has a real root
- **#FEAS<sub>ℝ</sub>**: Given a multivariate real polynomial, count its real roots (returning  $\infty$  if this number is not finite)
- **Theorem [BSS 89]** FEAS<sub>ℝ</sub> is NP<sub>ℝ</sub>-complete

## Basic Completeness Results

- Efficient algorithms of real algebraic geometry imply that

$$P_{\mathbb{R}} \subseteq NP_{\mathbb{R}} \subseteq \#P_{\mathbb{R}} \subseteq FPAR_{\mathbb{R}}$$

- We introduce notions of polynomial **reductions** (many-one/Turing) between computational problems and obtain the corresponding notions of completeness
- **FEAS<sub>ℝ</sub>**: Given a multivariate real polynomial, decide whether it has a real root
- **#FEAS<sub>ℝ</sub>**: Given a multivariate real polynomial, count its real roots (returning  $\infty$  if this number is not finite)
- **Theorem [BSS 89]** FEAS<sub>ℝ</sub> is NP<sub>ℝ</sub>-complete
- **Theorem [BC 04]** #FEAS<sub>ℝ</sub> is #P<sub>ℝ</sub>-complete

## Basic Completeness Results

- Efficient algorithms of real algebraic geometry imply that

$$P_{\mathbb{R}} \subseteq NP_{\mathbb{R}} \subseteq \#P_{\mathbb{R}} \subseteq FPAR_{\mathbb{R}}$$

- We introduce notions of polynomial **reductions** (many-one/Turing) between computational problems and obtain the corresponding notions of completeness
- **FEAS<sub>ℝ</sub>**: Given a multivariate real polynomial, decide whether it has a real root
- **#FEAS<sub>ℝ</sub>**: Given a multivariate real polynomial, count its real roots (returning  $\infty$  if this number is not finite)
- **Theorem [BSS 89]** FEAS<sub>ℝ</sub> is NP<sub>ℝ</sub>-complete
- **Theorem [BC 04]** #FEAS<sub>ℝ</sub> is #P<sub>ℝ</sub>-complete
- Only few completeness results are known

## **II. Additive Model**

## II. Additive Model

- No multiplications or divisions allowed: algebraic circuits specialize to **additive circuits**; we define additive complexity classes as before

## II. Additive Model

- No multiplications or divisions allowed: algebraic circuits specialize to **additive circuits**; we define additive complexity classes as before
- $P_{\text{add}} \subseteq NP_{\text{add}} \subseteq \#P_{\text{add}} \subseteq FPAR_{\text{add}}$

## II. Additive Model

- No multiplications or divisions allowed: algebraic circuits specialize to **additive circuits**; we define additive complexity classes as before
- $P_{\text{add}} \subseteq NP_{\text{add}} \subseteq \#P_{\text{add}} \subseteq FPAR_{\text{add}}$
- The set  $S_C$  accepted by an additive circuit  $C$  is **semilinear**, that is, derived from (closed) halfspaces by finitely many Boolean operations

## II. Additive Model

- No multiplications or divisions allowed: algebraic circuits specialize to **additive circuits**; we define additive complexity classes as before
- $P_{\text{add}} \subseteq NP_{\text{add}} \subseteq \#P_{\text{add}} \subseteq FPAR_{\text{add}}$
- The set  $S_C$  accepted by an additive circuit  $C$  is **semilinear**, that is, derived from (closed) halfspaces by finitely many Boolean operations
- **CSAT<sub>add</sub>**: Given an additive circuit  $C$ , decide whether  $S_C$  is nonempty

## II. Additive Model

- No multiplications or divisions allowed: algebraic circuits specialize to **additive circuits**; we define additive complexity classes as before
- $P_{\text{add}} \subseteq NP_{\text{add}} \subseteq \#P_{\text{add}} \subseteq FPAR_{\text{add}}$
- The set  $S_C$  accepted by an additive circuit  $C$  is **semilinear**, that is, derived from (closed) halfspaces by finitely many Boolean operations
- **CSAT<sub>add</sub>**: Given an additive circuit  $C$ , decide whether  $S_C$  is nonempty
- **Proposition** CSAT<sub>add</sub> is NP<sub>add</sub>-complete

## II. Additive Model

- No multiplications or divisions allowed: algebraic circuits specialize to **additive circuits**; we define additive complexity classes as before
- $P_{\text{add}} \subseteq NP_{\text{add}} \subseteq \#P_{\text{add}} \subseteq FPAR_{\text{add}}$
- The set  $S_C$  accepted by an additive circuit  $C$  is **semilinear**, that is, derived from (closed) halfspaces by finitely many Boolean operations
- **CSAT<sub>add</sub>**: Given an additive circuit  $C$ , decide whether  $S_C$  is nonempty
- **Proposition** CSAT<sub>add</sub> is NP<sub>add</sub>-complete

More natural complete problems?

## II. Additive Model

- No multiplications or divisions allowed: algebraic circuits specialize to **additive circuits**; we define additive complexity classes as before
- $P_{\text{add}} \subseteq NP_{\text{add}} \subseteq \#P_{\text{add}} \subseteq FPAR_{\text{add}}$
- The set  $S_C$  accepted by an additive circuit  $C$  is **semilinear**, that is, derived from (closed) halfspaces by finitely many Boolean operations
- **CSAT<sub>add</sub>**: Given an additive circuit  $C$ , decide whether  $S_C$  is nonempty
- **Proposition** CSAT<sub>add</sub> is NP<sub>add</sub>-complete

More natural complete problems?

Connection to discrete Turing model?

# Boolean Parts

## Boolean Parts

- Boolean part of a class  $\mathcal{C}$ :

$$\text{BP}(\mathcal{C}) := \{S \cap \{0, 1\}^\infty \mid S \in \mathcal{C}\}$$

## Boolean Parts

- Boolean part of a class  $\mathcal{C}$ :

$$\text{BP}(\mathcal{C}) := \{S \cap \{0, 1\}^\infty \mid S \in \mathcal{C}\}$$

- Theorem [Koiran 94] (i)  $\text{BP}(\text{P}_{\text{add}}) = \text{P/poly}$  (ii)  $\text{BP}(\text{NP}_{\text{add}}) = \text{NP/poly}$

## Boolean Parts

- Boolean part of a class  $\mathcal{C}$ :

$$\text{BP}(\mathcal{C}) := \{S \cap \{0, 1\}^\infty \mid S \in \mathcal{C}\}$$

- Theorem [Koiran 94] (i)  $\text{BP}(\text{P}_{\text{add}}) = \text{P/poly}$  (ii)  $\text{BP}(\text{NP}_{\text{add}}) = \text{NP/poly}$
- Idea for  $\subseteq$ :

## Boolean Parts

- **Boolean part** of a class  $\mathcal{C}$ :

$$\text{BP}(\mathcal{C}) := \{S \cap \{0, 1\}^\infty \mid S \in \mathcal{C}\}$$

- **Theorem [Koiran 94]** (i)  $\text{BP}(\text{P}_{\text{add}}) = \text{P/poly}$  (ii)  $\text{BP}(\text{NP}_{\text{add}}) = \text{NP/poly}$
- Idea for  $\subseteq$ :
  - (i) replace the constant vector by a rational vector serving as polynomial advice

## Boolean Parts

- **Boolean part** of a class  $\mathcal{C}$ :

$$\text{BP}(\mathcal{C}) := \{S \cap \{0, 1\}^\infty \mid S \in \mathcal{C}\}$$

- **Theorem [Koiran 94]** (i)  $\text{BP}(\text{P}_{\text{add}}) = \text{P/poly}$  (ii)  $\text{BP}(\text{NP}_{\text{add}}) = \text{NP/poly}$

- Idea for  $\subseteq$ :

(i) replace the constant vector by a rational vector serving as polynomial advice

(ii) instead of guessing a real vector guess a small rational point

## Boolean Parts

- **Boolean part** of a class  $\mathcal{C}$ :

$$\text{BP}(\mathcal{C}) := \{S \cap \{0, 1\}^\infty \mid S \in \mathcal{C}\}$$

- **Theorem [Koiran 94]** (i)  $\text{BP}(\text{P}_{\text{add}}) = \text{P/poly}$  (ii)  $\text{BP}(\text{NP}_{\text{add}}) = \text{NP/poly}$

- Idea for  $\subseteq$ :

(i) replace the constant vector by a rational vector serving as polynomial advice

(ii) instead of guessing a real vector guess a small rational point

- **Corollary (Transfer result):**  $\text{P}_{\text{add}} = \text{NP}_{\text{add}}$  implies  $\text{P/poly} = \text{NP/poly}$

# **The Power of Discrete Oracles**

## The Power of Discrete Oracles

- Theorem [Fournier and Koiran 2000] Every problem in  $\text{NP}_{\text{add}}$  can be solved in polynomial time by an additive machine making oracle calls to a problem in NP, i.e.,

$$\text{NP}_{\text{add}} \subseteq \text{P}_{\text{add}}^{\text{NP}}$$

## The Power of Discrete Oracles

- Theorem [Fournier and Koiran 2000] Every problem in  $\text{NP}_{\text{add}}$  can be solved in polynomial time by an additive machine making oracle calls to a problem in  $\text{NP}$ , i.e.,

$$\text{NP}_{\text{add}} \subseteq \text{P}_{\text{add}}^{\text{NP}}$$

- Corollary If  $S \in \text{NP}_{\text{add}}$  and  $S$  is  $\text{NP}$ -hard, then  $S$  is  $\text{NP}_{\text{add}}$ -complete (with respect to Turing reductions)

## The Power of Discrete Oracles

- Theorem [Fournier and Koiran 2000] Every problem in  $\text{NP}_{\text{add}}$  can be solved in polynomial time by an additive machine making oracle calls to a problem in  $\text{NP}$ , i.e.,

$$\text{NP}_{\text{add}} \subseteq \text{P}_{\text{add}}^{\text{NP}}$$

- Corollary If  $S \in \text{NP}_{\text{add}}$  and  $S$  is  $\text{NP}$ -hard, then  $S$  is  $\text{NP}_{\text{add}}$ -complete (with respect to Turing reductions)
- This allows to identify many  $\text{NP}_{\text{add}}$ -complete problems, e.g., the real knapsack problem

$$\{x \in \mathbb{R}^{\infty} \mid \exists n \in \mathbb{N} x \in \mathbb{R}^n \exists S \subseteq \{1, \dots, n\} \sum_{i \in S} x_i = 1\}$$

## The Power of Discrete Oracles

- Theorem [Fournier and Koiran 2000] Every problem in  $\text{NP}_{\text{add}}$  can be solved in polynomial time by an additive machine making oracle calls to a problem in  $\text{NP}$ , i.e.,

$$\text{NP}_{\text{add}} \subseteq \text{P}_{\text{add}}^{\text{NP}}$$

- Corollary If  $S \in \text{NP}_{\text{add}}$  and  $S$  is  $\text{NP}$ -hard, then  $S$  is  $\text{NP}_{\text{add}}$ -complete (with respect to Turing reductions)
- This allows to identify many  $\text{NP}_{\text{add}}$ -complete problems, e.g., the real knapsack problem

$$\{x \in \mathbb{R}^{\infty} \mid \exists n \in \mathbb{N} x \in \mathbb{R}^n \exists S \subseteq \{1, \dots, n\} \sum_{i \in S} x_i = 1\}$$

- Corollary  $\text{P/poly} = \text{NP/poly}$  implies  $\text{P}_{\text{add}} = \text{NP}_{\text{add}}$

# Point location in arrangements of hyperplanes

## Point location in arrangements of hyperplanes

- We define  $\mathcal{H}_n$  to be the set of linear polynomials  $a_0 + \sum_{i=1}^n a_i X_i$  with integer coefficients  $a_i$  such that  $\sum_{i=0}^n |a_i| \leq 2^{t(n)}$ , for a fixed polynomial  $t$

## Point location in arrangements of hyperplanes

- We define  $\mathcal{H}_n$  to be the set of linear polynomials  $a_0 + \sum_{i=1}^n a_i X_i$  with integer coefficients  $a_i$  such that  $\sum_{i=0}^n |a_i| \leq 2^{t(n)}$ , for a fixed polynomial  $t$
- The space  $\mathbb{R}^n$  is the disjoint union of all cells  $F$ , defined as

$$F = \bigcap_{f \in \mathcal{H}_n} \{x \in \mathbb{R}^n \mid f(x) = \sigma(f)\},$$

for some sign function  $\sigma: \mathcal{H}_n \rightarrow \{-1, 0, 1\}$

## Point location in arrangements of hyperplanes

- We define  $\mathcal{H}_n$  to be the set of linear polynomials  $a_0 + \sum_{i=1}^n a_i X_i$  with integer coefficients  $a_i$  such that  $\sum_{i=0}^n |a_i| \leq 2^{t(n)}$ , for a fixed polynomial  $t$
- The space  $\mathbb{R}^n$  is the disjoint union of all cells  $F$ , defined as

$$F = \bigcap_{f \in \mathcal{H}_n} \{x \in \mathbb{R}^n \mid f(x) = \sigma(f)\},$$

for some sign function  $\sigma: \mathcal{H}_n \rightarrow \{-1, 0, 1\}$

- The **point location problem** is the problem of computing, for a given input  $x \in \mathbb{R}^n$ , a rational point of bit-size  $n^{\mathcal{O}(1)}$  of the uniquely determined cell  $F$  in which  $x$  lies

## Point location in arrangements of hyperplanes

- We define  $\mathcal{H}_n$  to be the set of linear polynomials  $a_0 + \sum_{i=1}^n a_i X_i$  with integer coefficients  $a_i$  such that  $\sum_{i=0}^n |a_i| \leq 2^{t(n)}$ , for a fixed polynomial  $t$
- The space  $\mathbb{R}^n$  is the disjoint union of all cells  $F$ , defined as

$$F = \bigcap_{f \in \mathcal{H}_n} \{x \in \mathbb{R}^n \mid f(x) = \sigma(f)\},$$

for some sign function  $\sigma: \mathcal{H}_n \rightarrow \{-1, 0, 1\}$

- The **point location problem** is the problem of computing, for a given input  $x \in \mathbb{R}^n$ , a rational point of bit-size  $n^{\mathcal{O}(1)}$  of the uniquely determined cell  $F$  in which  $x$  lies
- **Theorem [Meyer auf der Heide 84]** For fixed  $n$ , the point location problem can be solved by a linear decision tree of depth  $n^{\mathcal{O}(1)}$

## **Point location in arrangements of hyperplanes (2)**

## Point location in arrangements of hyperplanes (2)

- Fournier and Koiran observed that the nonuniformity in Meyer auf der Heide's construction can be eliminated provided an NP-oracle is available

## Point location in arrangements of hyperplanes (2)

- Fournier and Koiran observed that the nonuniformity in Meyer auf der Heide's construction can be eliminated provided an NP-oracle is available
- Thus: The point location problem can be solved in polynomial time by an additive machine making oracle calls to a problem in NP

## Point location in arrangements of hyperplanes (2)

- Fournier and Koiran observed that the nonuniformity in Meyer auf der Heide's construction can be eliminated provided an NP-oracle is available
- Thus: The point location problem can be solved in polynomial time by an additive machine making oracle calls to a problem in NP
- Let  $S \in \text{NP}_{\text{add}}$  be decided by nondeterministic machine  $M$  with time bound  $t$

## Point location in arrangements of hyperplanes (2)

- Fournier and Koiran observed that the nonuniformity in Meyer auf der Heide's construction can be eliminated provided an NP-oracle is available
- Thus: The point location problem can be solved in polynomial time by an additive machine making oracle calls to a problem in NP
- Let  $S \in \text{NP}_{\text{add}}$  be decided by nondeterministic machine  $M$  with time bound  $t$
- Assume  $M$  constant-free

## Point location in arrangements of hyperplanes (2)

- Fournier and Koiran observed that the nonuniformity in Meyer auf der Heide's construction can be eliminated provided an NP-oracle is available
- Thus: The point location problem can be solved in polynomial time by an additive machine making oracle calls to a problem in NP
- Let  $S \in \text{NP}_{\text{add}}$  be decided by nondeterministic machine  $M$  with time bound  $t$
- Assume  $M$  constant-free
- W.l.o.g only digital guesses

## Point location in arrangements of hyperplanes (2)

- Fournier and Koiran observed that the nonuniformity in Meyer auf der Heide's construction can be eliminated provided an NP-oracle is available
- Thus: The point location problem can be solved in polynomial time by an additive machine making oracle calls to a problem in NP
- Let  $S \in \text{NP}_{\text{add}}$  be decided by nondeterministic machine  $M$  with time bound  $t$
- Assume  $M$  constant-free
- W.l.o.g only digital guesses
- Locate input  $x \in \mathbb{R}^n$  in poly time in a cell  $F$  of the arrangement  $\mathcal{H}_n$  (corresponding to  $t$ ) obtaining a rational point  $x'$  of  $F$

## Point location in arrangements of hyperplanes (2)

- Fournier and Koiran observed that the nonuniformity in Meyer auf der Heide's construction can be eliminated provided an NP-oracle is available
- Thus: The point location problem can be solved in polynomial time by an additive machine making oracle calls to a problem in NP
- Let  $S \in \text{NP}_{\text{add}}$  be decided by nondeterministic machine  $M$  with time bound  $t$
- Assume  $M$  constant-free
- W.l.o.g only digital guesses
- Locate input  $x \in \mathbb{R}^n$  in poly time in a cell  $F$  of the arrangement  $\mathcal{H}_n$  (corresponding to  $t$ ) obtaining a rational point  $x'$  of  $F$
- Guess the digital  $y$  and check in P whether  $M$  accepts  $(x', y)$

# **Power of Discrete Oracles: Extensions**

## Power of Discrete Oracles: Extensions

- For the complexity classes  $C \in \{\Sigma^k, \text{PH}, \text{PSPACE}\}$ , one can show in a similar way that

$$C_{\text{add}} \subseteq P_{\text{add}}^C$$

## Power of Discrete Oracles: Extensions

- For the complexity classes  $C \in \{\Sigma^k, \text{PH}, \text{PSPACE}\}$ , one can show in a similar way that

$$C_{\text{add}} \subseteq P_{\text{add}}^C$$

Similarly [BC 03],  $\#P_{\text{add}} \subseteq \text{FP}_{\text{add}}^{\#P}$

## Power of Discrete Oracles: Extensions

- For the complexity classes  $C \in \{\Sigma^k, \text{PH}, \text{PSPACE}\}$ , one can show in a similar way that

$$C_{\text{add}} \subseteq P_{\text{add}}^C$$

Similarly [BC 03],  $\#P_{\text{add}} \subseteq \text{FP}_{\text{add}}^{\#P}$

- The famous Toda Theorem states that  $\text{PH} \subseteq P^{\#P}$

## Power of Discrete Oracles: Extensions

- For the complexity classes  $C \in \{\Sigma^k, \text{PH}, \text{PSPACE}\}$ , one can show in a similar way that

$$C_{\text{add}} \subseteq P_{\text{add}}^C$$

Similarly [BC 03],  $\#P_{\text{add}} \subseteq \text{FP}_{\text{add}}^{\#P}$

- The famous Toda Theorem states that  $\text{PH} \subseteq P^{\#P}$
- Hence Toda's Theorem is also true in the additive model [BC 03]:

$$\text{PH}_{\text{add}} \subseteq P_{\text{add}}^{\text{PH}} \subseteq P_{\text{add}}^{\#P} = P_{\text{add}}^{\#P_{\text{add}}}$$

# **Complexity of Semilinear Problems in Succinct Representation**

## Complexity of Semilinear Problems in Succinct Representation

- Let the semilinear set  $S_C \subseteq \mathbb{R}^n$  be given by an additive circuit  $C$  that accepts  $S_C$

## Complexity of Semilinear Problems in Succinct Representation

- Let the semilinear set  $S_C \subseteq \mathbb{R}^n$  be given by an additive circuit  $C$  that accepts  $S_C$
- We ask about deciding various properties of  $S_C$  in the Euclidean topology, like closedness, compactness, connectedness, or irreducibility (Zariski topology)

## Complexity of Semilinear Problems in Succinct Representation

- Let the semilinear set  $S_C \subseteq \mathbb{R}^n$  be given by an additive circuit  $C$  that accepts  $S_C$
- We ask about deciding various properties of  $S_C$  in the Euclidean topology, like closedness, compactness, connectedness, or irreducibility (Zariski topology)
- and about computing numerical invariants like (local) dimension, topological Euler characteristic, Betti numbers

## Complexity of Semilinear Problems in Succinct Representation

- Let the semilinear set  $S_C \subseteq \mathbb{R}^n$  be given by an additive circuit  $C$  that accepts  $S_C$
- We ask about deciding various properties of  $S_C$  in the Euclidean topology, like closedness, compactness, connectedness, or irreducibility (Zariski topology)
- and about computing numerical invariants like (local) dimension, topological Euler characteristic, Betti numbers
- We obtain completeness results in the additive model for all these problems

## Complexity of Semilinear Problems in Succinct Representation

- Let the semilinear set  $S_C \subseteq \mathbb{R}^n$  be given by an additive circuit  $C$  that accepts  $S_C$
- We ask about deciding various properties of  $S_C$  in the Euclidean topology, like closedness, compactness, connectedness, or irreducibility (Zariski topology)
- and about computing numerical invariants like (local) dimension, topological Euler characteristic, Betti numbers
- We obtain completeness results in the additive model for all these problems
- Constant-free additive circuits can be taken as input of an ordinary Turing machine

## Complexity of Semilinear Problems in Succinct Representation

- Let the semilinear set  $S_C \subseteq \mathbb{R}^n$  be given by an additive circuit  $C$  that accepts  $S_C$
- We ask about deciding various properties of  $S_C$  in the Euclidean topology, like closedness, compactness, connectedness, or irreducibility (Zariski topology)
- and about computing numerical invariants like (local) dimension, topological Euler characteristic, Betti numbers
- We obtain completeness results in the additive model for all these problems
- Constant-free additive circuits can be taken as input of an ordinary Turing machine
- When restricting  $C$  to constant-free additive circuits, we also obtain completeness results for the corresponding discrete complexity class

## Some Completeness Results [BC 03, BC-de Naurois 05]

Problems	Complete in	Discrete version complete in
testing closedness	$\text{coNP}_{\text{add}}$	$\text{coNP}$
testing compactness	$\text{coNP}_{\text{add}}$	$\text{coNP}$
(loc) dimension at least $d$ ?	$\text{NP}_{\text{add}}$	$\text{NP}$
testing irreducibility	$\text{P}_{\text{add}}^{\text{NP}_{\text{add}}[\log]}$	$\text{P}^{\text{NP}[\log]}$
is there an isolated point?	$\Sigma_{\text{add}}^2$	$\Sigma_2\text{P}$
count the isolated points	$\text{FP}_{\text{add}}^{\#\text{P}_{\text{add}}}(\text{T})$	$\text{FP}^{\#\text{P}}(\text{T})$
count the irreducible components	$\text{FP}_{\text{add}}^{\#\text{P}_{\text{add}}}(\text{T})$	$\text{FP}^{\#\text{P}}(\text{T})$
compute the topological Euler characteristic	$\text{FP}_{\text{add}}^{\#\text{P}_{\text{add}}}(\text{T})$	$\text{FP}^{\#\text{P}}(\text{T})$
decide connectedness	$\text{PAR}_{\text{add}}(\text{T})$	$\text{PSPACE}$
compute the $k$ -th Betti number	$\text{FPAR}_{\text{add}}(\text{T})$	$\text{FPSPACE}$

# **Some Ingredients of Proofs**

## Some Ingredients of Proofs

- $\text{NP}_{\text{add}}$  hardness proofs by reducing the  $\text{NP}_{\text{add}}$ -complete **Circuit Boolean Satisfiability** to the problem under question

## Some Ingredients of Proofs

- $NP_{\text{add}}$  hardness proofs by reducing the  $NP_{\text{add}}$ -complete **Circuit Boolean Satisfiability** to the problem under question
- $\#P_{\text{add}}$ -hardness and  $PAR_{\text{add}}$ -hardness by showing  $\#P$ -hardness of PSPACE-hardness and applying “**power of discrete oracles**”

## Some Ingredients of Proofs

- $\text{NP}_{\text{add}}$  hardness proofs by reducing the  $\text{NP}_{\text{add}}$ -complete **Circuit Boolean Satisfiability** to the problem under question
- $\#\text{P}_{\text{add}}$ -hardness and  $\text{PAR}_{\text{add}}$ -hardness by showing  $\#\text{P}$ -hardness of PSPACE-hardness and applying “**power of discrete oracles**”
- Toda and Watanabe’s result  $\#\text{PH} \subseteq \text{P}\#\text{P}$

## Some Ingredients of Proofs

- $\text{NP}_{\text{add}}$  hardness proofs by reducing the  $\text{NP}_{\text{add}}$ -complete **Circuit Boolean Satisfiability** to the problem under question
- $\#\text{P}_{\text{add}}$ -hardness and  $\text{PAR}_{\text{add}}$ -hardness by showing  $\#\text{P}$ -hardness of PSPACE-hardness and applying “**power of discrete oracles**”
- **Toda and Watanabe’s result**  $\#\text{PH} \subseteq \text{P}\#\text{P}$
- Space-efficient linear algebra (for upper bound Betti)

### **III. Unrestricted BSS-Model over $\mathbb{C}$**

### III. Unrestricted BSS-Model over $\mathbb{C}$

- Efficient algorithms of complex algebraic geometry imply that

$$P_{\mathbb{C}} \subseteq NP_{\mathbb{C}} \subseteq \#P_{\mathbb{C}} \subseteq FPAR_{\mathbb{C}}$$

### III. Unrestricted BSS-Model over $\mathbb{C}$

- Efficient algorithms of complex algebraic geometry imply that

$$P_{\mathbb{C}} \subseteq NP_{\mathbb{C}} \subseteq \#P_{\mathbb{C}} \subseteq FPAR_{\mathbb{C}}$$

- $HN_{\mathbb{C}}$ : Given a finite set of multivariate complex polynomials, decide whether they have a common complex root

### III. Unrestricted BSS-Model over $\mathbb{C}$

- Efficient algorithms of complex algebraic geometry imply that

$$P_{\mathbb{C}} \subseteq NP_{\mathbb{C}} \subseteq \#P_{\mathbb{C}} \subseteq FPAR_{\mathbb{C}}$$

- $HN_{\mathbb{C}}$ : Given a finite set of multivariate complex polynomials, decide whether they have a common complex root
- $\#HN_{\mathbb{C}}$ : Given a finite set of multivariate complex polynomials, count its complex common roots (returning  $\infty$  if this number is not finite)

### III. Unrestricted BSS-Model over $\mathbb{C}$

- Efficient algorithms of complex algebraic geometry imply that

$$P_{\mathbb{C}} \subseteq NP_{\mathbb{C}} \subseteq \#P_{\mathbb{C}} \subseteq FPAR_{\mathbb{C}}$$

- $HN_{\mathbb{C}}$ : Given a finite set of multivariate complex polynomials, decide whether they have a common complex root
- $\#HN_{\mathbb{C}}$ : Given a finite set of multivariate complex polynomials, count its complex common roots (returning  $\infty$  if this number is not finite)
- **Theorem [BSS 89]**  $HN_{\mathbb{C}}$  is  $NP_{\mathbb{C}}$ -complete

### III. Unrestricted BSS-Model over $\mathbb{C}$

- Efficient algorithms of complex algebraic geometry imply that

$$P_{\mathbb{C}} \subseteq NP_{\mathbb{C}} \subseteq \#P_{\mathbb{C}} \subseteq FPAR_{\mathbb{C}}$$

- $HN_{\mathbb{C}}$ : Given a finite set of multivariate complex polynomials, decide whether they have a common complex root
- $\#HN_{\mathbb{C}}$ : Given a finite set of multivariate complex polynomials, count its complex common roots (returning  $\infty$  if this number is not finite)
- Theorem [BSS 89]  $HN_{\mathbb{C}}$  is  $NP_{\mathbb{C}}$ -complete
- Theorem [BC 04]  $\#HN_{\mathbb{C}}$  is  $\#P_{\mathbb{C}}$ -complete

## Boolean Parts: $P_C$

## Boolean Parts: $P_C$

- Theorem  $P \subseteq BP(P_C) \subseteq P^{RP}$

## Boolean Parts: $P_{\mathbb{C}}$

- Theorem  $P \subseteq BP(P_{\mathbb{C}}) \subseteq P^{RP}$
- Idea:
  - W.l.o.g. constants algebraically independent

## Boolean Parts: $P_{\mathbb{C}}$

- Theorem  $P \subseteq BP(P_{\mathbb{C}}) \subseteq P^{RP}$
- Idea:
  - W.l.o.g. constants algebraically independent
  - substitute them by huge integers

## Boolean Parts: $P_{\mathbb{C}}$

- Theorem  $P \subseteq BP(P_{\mathbb{C}}) \subseteq P^{RP}$
- Idea:
  - W.l.o.g. constants algebraically independent
  - substitute them by huge integers
  - compute modulo random primes

## Boolean Parts: $P_{\mathbb{C}}$

- **Theorem**  $P \subseteq BP(P_{\mathbb{C}}) \subseteq P^{RP}$
- **Idea:**
  - W.l.o.g. constants algebraically independent
  - substitute them by huge integers
  - compute modulo random primes
- **Corollary (Transfer result):**  $P_{\mathbb{C}} = NP_{\mathbb{C}}$  implies  $NP \subseteq P^{RP} \subseteq BPP$

## **Boolean Parts: $NP_C$ (1)**

## Boolean Parts: $NP_{\mathbb{C}}$ (1)

- Theorem [Koiran 96]  $NP \subseteq BP(NP_{\mathbb{C}}) \subseteq RP^{NP}$  under (GRH)

## Boolean Parts: $NP_{\mathbb{C}}$ (1)

- Theorem [Koiran 96]  $NP \subseteq BP(NP_{\mathbb{C}}) \subseteq RP^{NP}$  under (GRH)
- The proof is sophisticated. Rough Outline:

## Boolean Parts: $\text{NP}_{\mathbb{C}}$ (1)

- Theorem [Koiran 96]  $\text{NP} \subseteq \text{BP}(\text{NP}_{\mathbb{C}}) \subseteq \text{RP}^{\text{NP}}$  under (GRH)
- The proof is sophisticated. Rough Outline:
- Enough to show that the integer version  $\text{HN}_{\mathbb{C}}^{\mathbb{Z}}$  is in  $\text{RP}^{\text{NP}}$

## Boolean Parts: $\text{NP}_{\mathbb{C}}$ (1)

- Theorem [Koiran 96]  $\text{NP} \subseteq \text{BP}(\text{NP}_{\mathbb{C}}) \subseteq \text{RP}^{\text{NP}}$  under (GRH)
- The proof is sophisticated. Rough Outline:
- Enough to show that the integer version  $\text{HN}_{\mathbb{C}}^{\mathbb{Z}}$  is in  $\text{RP}^{\text{NP}}$
- Let  $(S)$  be a given system of polynomial equations over  $\mathbb{Z}$  of degree at most  $d$  and weight at most  $w$

## Boolean Parts: $\text{NP}_{\mathbb{C}}$ (1)

- Theorem [Koiran 96]  $\text{NP} \subseteq \text{BP}(\text{NP}_{\mathbb{C}}) \subseteq \text{RP}^{\text{NP}}$  under (GRH)
- The proof is sophisticated. Rough Outline:
- Enough to show that the integer version  $\text{HN}_{\mathbb{C}}^{\mathbb{Z}}$  is in  $\text{RP}^{\text{NP}}$
- Let  $(S)$  be a given system of polynomial equations over  $\mathbb{Z}$  of degree at most  $d$  and weight at most  $w$
- Consider the “density”  $\pi_S(x) := |\{p \leq x \mid p \text{ prime, } (S) \text{ solvable in } \mathbb{F}_p\}|$

## Boolean Parts: $\text{NP}_{\mathbb{C}}$ (1)

- Theorem [Koiran 96]  $\text{NP} \subseteq \text{BP}(\text{NP}_{\mathbb{C}}) \subseteq \text{RP}^{\text{NP}}$  under (GRH)
- The proof is sophisticated. Rough Outline:
- Enough to show that the integer version  $\text{HN}_{\mathbb{C}}^{\mathbb{Z}}$  is in  $\text{RP}^{\text{NP}}$
- Let  $(S)$  be a given system of polynomial equations over  $\mathbb{Z}$  of degree at most  $d$  and weight at most  $w$
- Consider the “density”  $\pi_S(x) := |\{p \leq x \mid p \text{ prime, } (S) \text{ solvable in } \mathbb{F}_p\}|$
- Using the effective arithmetic Nullstellensatz show that
  - if  $(S)$  has no solution over  $\mathbb{C}$  then  $\pi_S(x) \leq A(n, d, w)$  for all  $x$

## Boolean Parts: $\text{NP}_{\mathbb{C}}$ (1)

- Theorem [Koiran 96]  $\text{NP} \subseteq \text{BP}(\text{NP}_{\mathbb{C}}) \subseteq \text{RP}^{\text{NP}}$  under (GRH)
- The proof is sophisticated. Rough Outline:
- Enough to show that the integer version  $\text{HN}_{\mathbb{C}}^{\mathbb{Z}}$  is in  $\text{RP}^{\text{NP}}$
- Let  $(S)$  be a given system of polynomial equations over  $\mathbb{Z}$  of degree at most  $d$  and weight at most  $w$
- Consider the “density”  $\pi_S(x) := |\{p \leq x \mid p \text{ prime, } (S) \text{ solvable in } \mathbb{F}_p\}|$
- Using the effective arithmetic Nullstellensatz show that
  - if  $(S)$  has no solution over  $\mathbb{C}$  then  $\pi_S(x) \leq A(n, d, w)$  for all  $x$
  - if  $(S)$  is solvable over  $\mathbb{C}$ , then for all  $x$  (under (GRH))

$$\pi_S(x) \geq \frac{\pi(x)}{d^{\mathcal{O}(n)}} - x^{1/2} \log(wx)$$

## Boolean Parts: $NP_C$ (2)

## Boolean Parts: $\text{NP}_{\mathbb{C}}$ (2)

- On input  $(S)$  and  $x$  (in binary) one can compute  $\pi_S(x)$  in  $\#\text{NP}$  and hence, by Toda-Watanabe, in  $\text{FP}^{\#\text{P}}$

## Boolean Parts: $\text{NP}_{\mathbb{C}}$ (2)

- On input  $(S)$  and  $x$  (in binary) one can compute  $\pi_S(x)$  in  $\#\text{NP}$  and hence, by Toda-Watanabe, in  $\text{FP}^{\#\text{P}}$
- However, for deciding solvability of  $(S)$  over  $\mathbb{C}$ , only approximate counting is required

## Boolean Parts: $\text{NP}_{\mathbb{C}}$ (2)

- On input  $(S)$  and  $x$  (in binary) one can compute  $\pi_S(x)$  in  $\#\text{NP}$  and hence, by Toda-Watanabe, in  $\text{FP}^{\#\text{P}}$
- However, for deciding solvability of  $(S)$  over  $\mathbb{C}$ , only approximate counting is required
- Stockmeyer implies  $\text{HN}_{\mathbb{C}}^{\mathbb{Z}} \in \text{PH}$

## Boolean Parts: $\text{NP}_{\mathbb{C}}$ (2)

- On input  $(S)$  and  $x$  (in binary) one can compute  $\pi_S(x)$  in  $\#\text{NP}$  and hence, by Toda-Watanabe, in  $\text{FP}^{\#\text{P}}$
- However, for deciding solvability of  $(S)$  over  $\mathbb{C}$ , only approximate counting is required
- Stockmeyer implies  $\text{HN}_{\mathbb{C}}^{\mathbb{Z}} \in \text{PH}$
- The universal hashing argument yielding  $\text{RP} \subseteq \Sigma_2$  shows more precisely that  $\text{HN}_{\mathbb{C}}^{\mathbb{Z}} \in \text{RP}^{\text{NP}}$

**Boolean Parts:  $\#P_{\mathbb{C}}$**

## Boolean Parts: $\#P_{\mathbb{C}}$

- We call the Boolean part of  $\#P_{\mathbb{C}}$  the **class of geometric counting complex problems** and denote it by **GCC**

## Boolean Parts: $\#P_{\mathbb{C}}$

- We call the Boolean part of  $\#P_{\mathbb{C}}$  the **class of geometric counting complex problems** and denote it by **GCC**
- Efficient algorithms in algebraic geometry show that

$$\#P \subseteq \text{GCC} \subseteq \text{FPSPACE}$$

## Boolean Parts: $\#P_{\mathbb{C}}$

- We call the Boolean part of  $\#P_{\mathbb{C}}$  the **class of geometric counting complex problems** and denote it by **GCC**
- Efficient algorithms in algebraic geometry show that

$$\#P \subseteq \text{GCC} \subseteq \text{FPSPACE}$$

- **Tantalizing question: Is  $\text{GCC} \subseteq \text{FP}^{\#P}$ ?**

## Boolean Parts: $\#P_{\mathbb{C}}$

- We call the Boolean part of  $\#P_{\mathbb{C}}$  the **class of geometric counting complex problems** and denote it by **GCC**
- Efficient algorithms in algebraic geometry show that

$$\#P \subseteq \text{GCC} \subseteq \text{FPSPACE}$$

- **Tantalizing question: Is  $\text{GCC} \subseteq \text{FP}^{\#P}$ ?**
- Bernstein & Kushnirenko: the number of complex solutions in  $(\mathbb{C}^*)^n$  of a **regular** system equals the mixed volume of the Newton polytopes of the equations; the latter can be computed in  $\text{FP}^{\#P}$

# Dimension

## Dimension

- **DIM $\mathbb{C}$** : Given a finite set of multivariate complex polynomials with affine zero set  $Z \subseteq \mathbb{C}^n$  and  $d \in \mathbb{N}$ , decide whether  $\dim Z \geq d$

## Dimension

- **DIM $\mathbb{C}$** : Given a finite set of multivariate complex polynomials with affine zero set  $Z \subseteq \mathbb{C}^n$  and  $d \in \mathbb{N}$ , decide whether  $\dim Z \geq d$
- **Theorem [Koiran 97]** **DIM $\mathbb{C}$  is NP $\mathbb{C}$ -complete**

## Dimension

- **DIM $\mathbb{C}$** : Given a finite set of multivariate complex polynomials with affine zero set  $Z \subseteq \mathbb{C}^n$  and  $d \in \mathbb{N}$ , decide whether  $\dim Z \geq d$
- **Theorem [Koiran 97]** **DIM $\mathbb{C}$  is NP $\mathbb{C}$ -complete**
- The NP $\mathbb{C}$ -hardness of DIM $\mathbb{C}$  follows by a trivial reduction from HN $\mathbb{C}$  to DIM $\mathbb{C}$

## Dimension

- **DIM $\mathbb{C}$** : Given a finite set of multivariate complex polynomials with affine zero set  $Z \subseteq \mathbb{C}^n$  and  $d \in \mathbb{N}$ , decide whether  $\dim Z \geq d$
- **Theorem [Koiran 97]** **DIM $\mathbb{C}$  is NP $\mathbb{C}$ -complete**
- The NP $\mathbb{C}$ -hardness of DIM $\mathbb{C}$  follows by a trivial reduction from HN $\mathbb{C}$  to DIM $\mathbb{C}$
- We parametrize affine subspaces  $A_a \subseteq \mathbb{C}^n$  of codimension  $d$  by the coefficient matrix  $a \in \mathbb{C}^{d(n+1)}$  of a system of linear equations describing  $A_a$

## Dimension

- **DIM $\mathbb{C}$** : Given a finite set of multivariate complex polynomials with affine zero set  $Z \subseteq \mathbb{C}^n$  and  $d \in \mathbb{N}$ , decide whether  $\dim Z \geq d$
- **Theorem [Koiran 97]** **DIM $\mathbb{C}$  is NP $\mathbb{C}$ -complete**
- The NP $\mathbb{C}$ -hardness of DIM $\mathbb{C}$  follows by a trivial reduction from HN $\mathbb{C}$  to DIM $\mathbb{C}$
- We parametrize affine subspaces  $A_a \subseteq \mathbb{C}^n$  of codimension  $d$  by the coefficient matrix  $a \in \mathbb{C}^{d(n+1)}$  of a system of linear equations describing  $A_a$
- To prove the upper bound, we use

$$\dim Z \geq d \iff \forall^* a \ Z \cap A_a \neq \emptyset$$

## Dimension

- **DIM $\mathbb{C}$** : Given a finite set of multivariate complex polynomials with affine zero set  $Z \subseteq \mathbb{C}^n$  and  $d \in \mathbb{N}$ , decide whether  $\dim Z \geq d$
- **Theorem [Koiran 97]** **DIM $\mathbb{C}$  is NP $\mathbb{C}$ -complete**
- The NP $\mathbb{C}$ -hardness of DIM $\mathbb{C}$  follows by a trivial reduction from HN $\mathbb{C}$  to DIM $\mathbb{C}$
- We parametrize affine subspaces  $A_a \subseteq \mathbb{C}^n$  of codimension  $d$  by the coefficient matrix  $a \in \mathbb{C}^{d(n+1)}$  of a system of linear equations describing  $A_a$
- To prove the upper bound, we use

$$\dim Z \geq d \iff \forall^* a \ Z \cap A_a \neq \emptyset$$

- This suggests an obvious randomized algorithm

# Generic quantifiers

## Generic quantifiers

- In the following we describe a general formal complexity framework for the analysis of general position arguments

## Generic quantifiers

- In the following we describe a general formal complexity framework for the analysis of general position arguments
- We show that it is possible to compute in deterministic polynomial time a list of candidates  $\alpha_1, \dots, \alpha_{2p+1}$  for generic parameters, among which the majority satisfies the genericity condition

## Generic quantifiers

- In the following we describe a general formal complexity framework for the analysis of general position arguments
- We show that it is possible to compute in deterministic polynomial time a list of candidates  $\alpha_1, \dots, \alpha_{2p+1}$  for generic parameters, among which the majority satisfies the genericity condition
- $\mathcal{F}_{\mathbb{R}}$  denotes the set of first order formulas over the language of the theory of ordered fields with constant symbols for real numbers

## Generic quantifiers

- In the following we describe a general formal complexity framework for the analysis of general position arguments
- We show that it is possible to compute in deterministic polynomial time a list of candidates  $\alpha_1, \dots, \alpha_{2p+1}$  for generic parameters, among which the majority satisfies the genericity condition
- $\mathcal{F}_{\mathbb{R}}$  denotes the set of first order formulas over the language of the theory of ordered fields with constant symbols for real numbers
- Let  $F \in \mathcal{F}_{\mathbb{R}}$  have free variables  $a_1, \dots, a_k$

## Generic quantifiers

- In the following we describe a general formal complexity framework for the analysis of general position arguments
- We show that it is possible to compute in deterministic polynomial time a list of candidates  $\alpha_1, \dots, \alpha_{2p+1}$  for generic parameters, among which the majority satisfies the genericity condition
- $\mathcal{F}_{\mathbb{R}}$  denotes the set of first order formulas over the language of the theory of ordered fields with constant symbols for real numbers
- Let  $F \in \mathcal{F}_{\mathbb{R}}$  have free variables  $a_1, \dots, a_k$
- $F$  is called **Zariski-generically true** if the set of values  $a \in \mathbb{R}^k$  not satisfying  $F(a)$  has dimension strictly less than  $k$ . We express this fact by writing  $\forall^* a F(a)$  using the **generic universal quantifier**  $\forall^*$

## Generic quantifiers

- In the following we describe a general formal complexity framework for the analysis of general position arguments
- We show that it is possible to compute in deterministic polynomial time a list of candidates  $\alpha_1, \dots, \alpha_{2p+1}$  for generic parameters, among which the majority satisfies the genericity condition
- $\mathcal{F}_{\mathbb{R}}$  denotes the set of first order formulas over the language of the theory of ordered fields with constant symbols for real numbers
- Let  $F \in \mathcal{F}_{\mathbb{R}}$  have free variables  $a_1, \dots, a_k$
- $F$  is called **Zariski-generically true** if the set of values  $a \in \mathbb{R}^k$  not satisfying  $F(a)$  has dimension strictly less than  $k$ . We express this fact by writing  $\forall^* a F(a)$  using the **generic universal quantifier**  $\forall^*$
- Define this similarly for formulas in  $\mathcal{F}_{\mathbb{C}}$

# Partial Witness Sequences

## Partial Witness Sequences

- Let  $F(u, a) \in \mathcal{F}_{\mathbb{C}}$  with free variables  $u \in \mathbb{C}^p$  and  $a \in \mathbb{C}^k$ . We call a sequence  $\alpha = (\alpha_1, \dots, \alpha_{2p+1}) \in (\mathbb{C}^k)^{2p+1}$  a **witness sequence** for  $F$  iff

$$\forall u \in \mathbb{C}^p \left( \forall^* a \in \mathbb{C}^k F(u, a) \iff |\{i \in [2p+1] \mid F(u, \alpha_i)\}| > p \right).$$

## Partial Witness Sequences

- Let  $F(u, a) \in \mathcal{F}_{\mathbb{C}}$  with free variables  $u \in \mathbb{C}^p$  and  $a \in \mathbb{C}^k$ . We call a sequence  $\alpha = (\alpha_1, \dots, \alpha_{2p+1}) \in (\mathbb{C}^k)^{2p+1}$  a **witness sequence** for  $F$  iff

$$\forall u \in \mathbb{C}^p \left( \forall^* a \in \mathbb{C}^k F(u, a) \iff |\{i \in [2p+1] \mid F(u, \alpha_i)\}| > p \right).$$

- Let  $F(u, a) \in \mathcal{F}_{\mathbb{R}}$  with free variables  $u \in \mathbb{R}^{2p}$  and  $a \in \mathbb{R}^k$ . A sequence  $\alpha = (\alpha_1, \dots, \alpha_{4p+1}) \in (\mathbb{R}^k)^{4p+1}$  is called a **partial witness sequence** for  $F$  iff

$$\forall u \in \mathbb{R}^{2p} \left( \forall^* a \in \mathbb{R}^k F(u, a) \implies |\{i \in [4p+1] \mid F(u, \alpha_i)\}| > 2p \right).$$

# Derandomizing Genericity Conditions

## Derandomizing Genericity Conditions

- The set of (partial) witness sequences is Zariski dense

## Derandomizing Genericity Conditions

- The set of (partial) witness sequences is Zariski dense
- A (partial) witness sequence of  $F$  can be computed by a machine over  $\mathbb{C}$  in polynomial time under some mild assumption:  
the formula should be expressible in the polynomial hierarchy over  $\mathbb{R}$  (essential: number of alternating quantifier blocks is fixed)

## Derandomizing Genericity Conditions

- The set of (partial) witness sequences is Zariski dense
- A (partial) witness sequence of  $F$  can be computed by a machine over  $\mathbb{C}$  in polynomial time under some mild assumption:  
the formula should be expressible in the polynomial hierarchy over  $\mathbb{R}$  (essential: number of alternating quantifier blocks is fixed)
- The machine essentially produces huge numbers by repeated squaring

## Derandomizing Genericity Conditions

- The set of (partial) witness sequences is Zariski dense
- A (partial) witness sequence of  $F$  can be computed by a machine over  $\mathbb{C}$  in polynomial time under some mild assumption:  
the formula should be expressible in the polynomial hierarchy over  $\mathbb{R}$  (essential: number of alternating quantifier blocks is fixed)
- The machine essentially produces huge numbers by repeated squaring
- The analysis is based on the full machinery of efficient quantifier elimination over the reals

## **Dimension (continued)**

## Dimension (continued)

- Let  $Z_u$  be the zero set of polynomials with coefficient vector  $u \in \mathbb{C}^p$

## Dimension (continued)

- Let  $Z_u$  be the zero set of polynomials with coefficient vector  $u \in \mathbb{C}^p$
- Recall

$$\dim Z_u \geq d \iff \forall^* a \ Z_u \cap A_a \neq \emptyset$$

## Dimension (continued)

- Let  $Z_u$  be the zero set of polynomials with coefficient vector  $u \in \mathbb{C}^p$
- Recall

$$\dim Z_u \geq d \iff \forall^* a \ Z_u \cap A_a \neq \emptyset$$

- Let  $F(u, a) \in \mathcal{F}_{\mathbb{C}}$  be the obvious parametrized formula expressing that  $Z_u \cap A_a \neq \emptyset$

## Dimension (continued)

- Let  $Z_u$  be the zero set of polynomials with coefficient vector  $u \in \mathbb{C}^p$
- Recall

$$\dim Z_u \geq d \iff \forall^* a \ Z_u \cap A_a \neq \emptyset$$

- Let  $F(u, a) \in \mathcal{F}_{\mathbb{C}}$  be the obvious parametrized formula expressing that  $Z_u \cap A_a \neq \emptyset$
- Compute a witness sequence  $A_1, \dots, A_{2p+1}$  for  $F(u, a)$  in polynomial time

## Dimension (continued)

- Let  $Z_u$  be the zero set of polynomials with coefficient vector  $u \in \mathbb{C}^p$
- Recall

$$\dim Z_u \geq d \iff \forall^* a \ Z_u \cap A_a \neq \emptyset$$

- Let  $F(u, a) \in \mathcal{F}_{\mathbb{C}}$  be the obvious parametrized formula expressing that  $Z_u \cap A_a \neq \emptyset$
- Compute a witness sequence  $A_1, \dots, A_{2p+1}$  for  $F(u, a)$  in polynomial time
- $\forall^* a \ Z_u \cap A_a \neq \emptyset \iff$  the majority of the  $A_j$  intersect  $Z_u$

## Dimension (continued)

- Let  $Z_u$  be the zero set of polynomials with coefficient vector  $u \in \mathbb{C}^p$
- Recall

$$\dim Z_u \geq d \iff \forall^* a \ Z_u \cap A_a \neq \emptyset$$

- Let  $F(u, a) \in \mathcal{F}_{\mathbb{C}}$  be the obvious parametrized formula expressing that  $Z_u \cap A_a \neq \emptyset$
- Compute a witness sequence  $A_1, \dots, A_{2p+1}$  for  $F(u, a)$  in polynomial time
- $\forall^* a \ Z_u \cap A_a \neq \emptyset \iff$  the majority of the  $A_j$  intersect  $Z_u$
- the right-hand side can be tested in  $\text{NP}_{\mathbb{C}}$

## Dimension (continued)

- Let  $Z_u$  be the zero set of polynomials with coefficient vector  $u \in \mathbb{C}^p$
- Recall

$$\dim Z_u \geq d \iff \forall^* a \ Z_u \cap A_a \neq \emptyset$$

- Let  $F(u, a) \in \mathcal{F}_{\mathbb{C}}$  be the obvious parametrized formula expressing that  $Z_u \cap A_a \neq \emptyset$
- Compute a witness sequence  $A_1, \dots, A_{2p+1}$  for  $F(u, a)$  in polynomial time
- $\forall^* a \ Z_u \cap A_a \neq \emptyset \iff$  the majority of the  $A_j$  intersect  $Z_u$
- the right-hand side can be tested in  $\text{NP}_{\mathbb{C}}$
- Let  $\text{DIM}_{\mathbb{C}}^{\mathbb{Z}}$  denote the discrete version of  $\text{DIM}_{\mathbb{C}}$

## Dimension (continued)

- Let  $Z_u$  be the zero set of polynomials with coefficient vector  $u \in \mathbb{C}^p$
- Recall

$$\dim Z_u \geq d \iff \forall^* a \ Z_u \cap A_a \neq \emptyset$$

- Let  $F(u, a) \in \mathcal{F}_{\mathbb{C}}$  be the obvious parametrized formula expressing that  $Z_u \cap A_a \neq \emptyset$
- Compute a witness sequence  $A_1, \dots, A_{2p+1}$  for  $F(u, a)$  in polynomial time
- $\forall^* a \ Z_u \cap A_a \neq \emptyset \iff$  the majority of the  $A_j$  intersect  $Z_u$
- the right-hand side can be tested in  $\text{NP}_{\mathbb{C}}$
- Let  $\text{DIM}_{\mathbb{C}}^{\mathbb{Z}}$  denote the discrete version of  $\text{DIM}_{\mathbb{C}}$
- Analysis of the proof shows that  $\text{DIM}_{\mathbb{C}}^{\mathbb{Z}}$  is  $\text{BP}(\text{NP}_{\mathbb{C}})$ -complete

# Geometric Degree

## Geometric Degree

- The **geometric degree**  $\deg Z$  of an affine variety  $Z \subseteq \mathbb{C}^n$  of dimension  $d$  is defined, for almost all  $a$ , as  $\deg Z := |Z \cap A_a|$

## Geometric Degree

- The **geometric degree**  $\deg Z$  of an affine variety  $Z \subseteq \mathbb{C}^n$  of dimension  $d$  is defined, for almost all  $a$ , as  $\deg Z := |Z \cap A_a|$
- **Theorem [BC 04]** The problem to compute the geometric degree is  $\text{FP}_{\mathbb{C}}^{\#\text{P}_{\mathbb{C}}}$ -complete

## Geometric Degree

- The **geometric degree**  $\deg Z$  of an affine variety  $Z \subseteq \mathbb{C}^n$  of dimension  $d$  is defined, for almost all  $a$ , as  $\deg Z := |Z \cap A_a|$
- **Theorem [BC 04]** The problem to compute the geometric degree is  $\text{FP}_{\mathbb{C}}^{\#\text{P}_{\mathbb{C}}}$ -complete
- For the upper bound, we have to analyze **transversality**

## Geometric Degree

- The **geometric degree**  $\deg Z$  of an affine variety  $Z \subseteq \mathbb{C}^n$  of dimension  $d$  is defined, for almost all  $a$ , as  $\deg Z := |Z \cap A_a|$
- **Theorem [BC 04]** The problem to compute the geometric degree is  $\text{FP}_{\mathbb{C}}^{\#\text{P}_{\mathbb{C}}}$ -complete
- For the upper bound, we have to analyze **transversality**
  - $\forall^* a \in \mathbb{C}^k$   $A_a$  is transversal to  $Z$

## Geometric Degree

- The **geometric degree**  $\deg Z$  of an affine variety  $Z \subseteq \mathbb{C}^n$  of dimension  $d$  is defined, for almost all  $a$ , as  $\deg Z := |Z \cap A_a|$
- **Theorem [BC 04]** The problem to compute the geometric degree is  $\text{FP}_{\mathbb{C}}^{\#\text{P}_{\mathbb{C}}}$ -complete
- For the upper bound, we have to analyze **transversality**
  - $\forall^* a \in \mathbb{C}^k$   $A_a$  is transversal to  $Z$
  - $\forall a \in \mathbb{C}^k$  ( $A_a$  is transversal to  $Z \implies |Z \cap A_a| = \deg Z$ )

## Geometric Degree

- The **geometric degree**  $\deg Z$  of an affine variety  $Z \subseteq \mathbb{C}^n$  of dimension  $d$  is defined, for almost all  $a$ , as  $\deg Z := |Z \cap A_a|$
- **Theorem [BC 04]** The problem to compute the geometric degree is  $\text{FP}_{\mathbb{C}}^{\#\text{P}_{\mathbb{C}}}$ -complete
- For the upper bound, we have to analyze **transversality**
  - $\forall^* a \in \mathbb{C}^k$   $A_a$  is transversal to  $Z$
  - $\forall a \in \mathbb{C}^k$  ( $A_a$  is transversal to  $Z \implies |Z \cap A_a| = \deg Z$ )
- Express transversality by a formula  $F$  in the polynomial hierarchy **over**  $\mathbb{R}$

## Geometric Degree

- The **geometric degree**  $\deg Z$  of an affine variety  $Z \subseteq \mathbb{C}^n$  of dimension  $d$  is defined, for almost all  $a$ , as  $\deg Z := |Z \cap A_a|$
- **Theorem [BC 04]** The problem to compute the geometric degree is  $\text{FP}_{\mathbb{C}}^{\#\text{P}_{\mathbb{C}}}$ -complete
- For the upper bound, we have to analyze **transversality**
  - $\forall^* a \in \mathbb{C}^k$   $A_a$  is transversal to  $Z$
  - $\forall a \in \mathbb{C}^k$  ( $A_a$  is transversal to  $Z \implies |Z \cap A_a| = \deg Z$ )
- Express transversality by a formula  $F$  in the polynomial hierarchy **over  $\mathbb{R}$**
- Compute partial witness sequence  $A_1, \dots, A_{4p+1}$  for  $F$  ( $u \in \mathbb{C}^p = \mathbb{R}^{2p}$ )

## Geometric Degree

- The **geometric degree**  $\deg Z$  of an affine variety  $Z \subseteq \mathbb{C}^n$  of dimension  $d$  is defined, for almost all  $a$ , as  $\deg Z := |Z \cap A_a|$
- **Theorem [BC 04]** The problem to compute the geometric degree is  $\text{FP}_{\mathbb{C}}^{\#\text{P}_{\mathbb{C}}}$ -complete
- For the upper bound, we have to analyze **transversality**
  - $\forall^* a \in \mathbb{C}^k$   $A_a$  is transversal to  $Z$
  - $\forall a \in \mathbb{C}^k$  ( $A_a$  is transversal to  $Z \implies |Z \cap A_a| = \deg Z$ )
- Express transversality by a formula  $F$  in the polynomial hierarchy **over  $\mathbb{R}$**
- Compute partial witness sequence  $A_1, \dots, A_{4p+1}$  for  $F$  ( $u \in \mathbb{C}^p = \mathbb{R}^{2p}$ )
- Obtain  $N_j := |Z \cap A_j|$  by oracle calls to  $\text{HN}_{\mathbb{C}}$  and make a majority vote

## Geometric Degree

- The **geometric degree**  $\deg Z$  of an affine variety  $Z \subseteq \mathbb{C}^n$  of dimension  $d$  is defined, for almost all  $a$ , as  $\deg Z := |Z \cap A_a|$
- **Theorem [BC 04]** The problem to compute the geometric degree is  $\text{FP}_{\mathbb{C}}^{\#\text{P}_{\mathbb{C}}}$ -complete
- For the upper bound, we have to analyze **transversality**
  - $\forall^* a \in \mathbb{C}^k$   $A_a$  is transversal to  $Z$
  - $\forall a \in \mathbb{C}^k$  ( $A_a$  is transversal to  $Z \implies |Z \cap A_a| = \deg Z$ )
- Express transversality by a formula  $F$  in the polynomial hierarchy **over  $\mathbb{R}$**
- Compute partial witness sequence  $A_1, \dots, A_{4p+1}$  for  $F$  ( $u \in \mathbb{C}^p = \mathbb{R}^{2p}$ )
- Obtain  $N_j := |Z \cap A_j|$  by oracle calls to  $\text{HN}_{\mathbb{C}}$  and make a majority vote
- **$F$  over  $\mathbb{R}$** : required for expressing tangent spaces if the given ideal is not radical!

# Euler Characteristic

## Euler Characteristic

- The Euler characteristic  $\chi(X)$  of a space  $X$  is one of the most basic invariants in algebraic topology

## Euler Characteristic

- The Euler characteristic  $\chi(X)$  of a space  $X$  is one of the most basic invariants in algebraic topology
- For spaces  $X$  admitting a finite triangulation it is defined as

$$\chi(X) = \sum_{k=0}^d (-1)^k N_k,$$

where  $N_k$  is the number of  $k$ -dimensional simplices and  $d = \dim X$

## Euler Characteristic

- The Euler characteristic  $\chi(X)$  of a space  $X$  is one of the most basic invariants in algebraic topology
- For spaces  $X$  admitting a finite triangulation it is defined as

$$\chi(X) = \sum_{k=0}^d (-1)^k N_k,$$

where  $N_k$  is the number of  $k$ -dimensional simplices and  $d = \dim X$

- For a connected closed surface  $X$  of genus  $g$  we have  $\chi(X) = 2 - 2g$

## Euler Characteristic

- The Euler characteristic  $\chi(X)$  of a space  $X$  is one of the most basic invariants in algebraic topology
- For spaces  $X$  admitting a finite triangulation it is defined as

$$\chi(X) = \sum_{k=0}^d (-1)^k N_k,$$

where  $N_k$  is the number of  $k$ -dimensional simplices and  $d = \dim X$

- For a connected closed surface  $X$  of genus  $g$  we have  $\chi(X) = 2 - 2g$
- Alternative description:

$$\chi(X) = \sum_{k=0}^d (-1)^k b_k(X),$$

where  $b_0(X)$  is the number of connected components of  $X$  and the  $k$ th (topological) Betti number  $b_k(X)$  measures a “higher degree of connectedness”

# Complexity of Computing Betti Numbers

## Complexity of Computing Betti Numbers

- Consider the following problems for fixed  $k \in \mathbb{N}$ :

## Complexity of Computing Betti Numbers

- Consider the following problems for fixed  $k \in \mathbb{N}$ :

**BETTI <sub>$\mathbb{C}$</sub> ( $k$ ):** *Given a complex variety  $Z$ , compute its  $k$ -th topological Betti number*

## Complexity of Computing Betti Numbers

- Consider the following problems for fixed  $k \in \mathbb{N}$ :

**BETTI<sub>ℂ</sub>( $k$ )**: *Given a complex variety  $Z$ , compute its  $k$ -th topological Betti number*

**EULER<sub>ℂ</sub>**: *Given a complex variety  $Z$ , compute its topological Euler characteristic*

## Complexity of Computing Betti Numbers

- Consider the following problems for fixed  $k \in \mathbb{N}$ :

$\text{BETTI}_{\mathbb{C}}(k)$ : *Given a complex variety  $Z$ , compute its  $k$ -th topological Betti number*

$\text{EULER}_{\mathbb{C}}$ : *Given a complex variety  $Z$ , compute its topological Euler characteristic*

- [BC 03]  $\text{BETTI}_{\mathbb{C}}^{\mathbb{Z}}(k)$  is FPSPACE-hard

## Complexity of Computing Betti Numbers

- Consider the following problems for fixed  $k \in \mathbb{N}$ :

$\text{BETTI}_{\mathbb{C}}(k)$ : *Given a complex variety  $Z$ , compute its  $k$ -th topological Betti number*

$\text{EULER}_{\mathbb{C}}$ : *Given a complex variety  $Z$ , compute its topological Euler characteristic*

- [BC 03]  $\text{BETTI}_{\mathbb{C}}^{\mathbb{Z}}(k)$  is FPSPACE-hard
- [Basu 04]  $\text{BETTI}_{\mathbb{C}}^{\mathbb{Z}}(k) \in \text{FPSPACE}$  (?)

## Complexity of Computing Betti Numbers

- Consider the following problems for fixed  $k \in \mathbb{N}$ :

$\text{BETTI}_{\mathbb{C}}(k)$ : *Given a complex variety  $Z$ , compute its  $k$ -th topological Betti number*

$\text{EULER}_{\mathbb{C}}$ : *Given a complex variety  $Z$ , compute its topological Euler characteristic*

- [BC 03]  $\text{BETTI}_{\mathbb{C}}^{\mathbb{Z}}(k)$  is FPSPACE-hard
- [Basu 04]  $\text{BETTI}_{\mathbb{C}}^{\mathbb{Z}}(k) \in \text{FPSPACE}$  (?)
- [Basu 99]  $\text{EULER}_{\mathbb{C}}^{\mathbb{Z}} \in \text{FPSPACE}$

## Complexity of Computing Betti Numbers

- Consider the following problems for fixed  $k \in \mathbb{N}$ :

$\text{BETTI}_{\mathbb{C}}(k)$ : *Given a complex variety  $Z$ , compute its  $k$ -th topological Betti number*

$\text{EULER}_{\mathbb{C}}$ : *Given a complex variety  $Z$ , compute its topological Euler characteristic*

- [BC 03]  $\text{BETTI}_{\mathbb{C}}^{\mathbb{Z}}(k)$  is FPSPACE-hard
- [Basu 04]  $\text{BETTI}_{\mathbb{C}}^{\mathbb{Z}}(k) \in \text{FPSPACE}$  (?)
- [Basu 99]  $\text{EULER}_{\mathbb{C}}^{\mathbb{Z}} \in \text{FPSPACE}$
- What is the true complexity of EULER?

# **The complexity of computing the Euler characteristic**

## **The complexity of computing the Euler characteristic**

- The Euler characteristic of a finite set equals its cardinality

## The complexity of computing the Euler characteristic

- The Euler characteristic of a finite set equals its cardinality
- One can show that  $\#HN_{\mathbb{C}}$  reduces to  $EULER_{\mathbb{C}}$

## The complexity of computing the Euler characteristic

- The Euler characteristic of a finite set equals its cardinality
- One can show that  $\#HN_{\mathbb{C}}$  reduces to  $EULER_{\mathbb{C}}$   
(difficulty: decide  $\dim Z \leq 0$  in poly time using oracle calls for  $EULER$ )

## The complexity of computing the Euler characteristic

- The Euler characteristic of a finite set equals its cardinality
- One can show that  $\#\text{HN}_{\mathbb{C}}$  reduces to  $\text{EULER}_{\mathbb{C}}$   
(difficulty: decide  $\dim Z \leq 0$  in poly time using oracle calls for  $\text{EULER}$ )
- Theorem [B-C-Lotz 04]  $\text{EULER}_{\mathbb{C}}$  is polynomial time equivalent to  $\#\text{HN}_{\mathbb{C}}$ , that is,  $\text{FP}_{\mathbb{C}}^{\#\text{P}_{\mathbb{C}}}$ -complete. A similar statement holds for the Turingmodel.

## The complexity of computing the Euler characteristic

- The Euler characteristic of a finite set equals its cardinality
- One can show that  $\#\text{HN}_{\mathbb{C}}$  reduces to  $\text{EULER}_{\mathbb{C}}$   
(difficulty: decide  $\dim Z \leq 0$  in poly time using oracle calls for  $\text{EULER}$ )
- **Theorem [B-C-Lotz 04]**  $\text{EULER}_{\mathbb{C}}$  is polynomial time equivalent to  $\#\text{HN}_{\mathbb{C}}$ , that is,  $\text{FP}_{\mathbb{C}}^{\#\text{P}_{\mathbb{C}}}$ -complete. A similar statement holds for the Turingmodel.
- **Corollary**  $\text{EULER}_{\mathbb{C}}^{\mathbb{Z}}$  is not PSPACE-complete, unless  $\text{GCC} = \text{PSPACE}$

## The Summation Lemma for $\#P_C$

## The Summation Lemma for $\#P_{\mathbb{C}}$

Let  $\varphi: \mathbb{C}^{\infty} \times \{0, 1\}^{\infty} \rightarrow \mathbb{N}$  be in  $\#P_{\mathbb{C}}$  and  $q$  be polynomial. Define for  $x \in \mathbb{C}^n$

$$\tilde{\varphi}(x) := \sum_{y \in \{0,1\}^{q(n)}} \varphi(x, y).$$

Then  $\tilde{\varphi}: \mathbb{C}^{\infty} \rightarrow \mathbb{N}$  is also in  $\#P_{\mathbb{C}}$

## Reduction from $\text{EULER}_{\mathbb{C}}$ to $\#\text{HN}_{\mathbb{C}}$ (1)

## Reduction from $\text{EULER}_{\mathbb{C}}$ to $\#\text{HN}_{\mathbb{C}}$ (1)

Note that  $\chi$  is **additive**: for  $Z_1, Z_2$  constructible and disjoint we have

$$\chi(Z_1 \cup Z_2) = \chi(Z_1) + \chi(Z_2)$$

## Reduction from $\text{EULER}_{\mathbb{C}}$ to $\#\text{HN}_{\mathbb{C}}$ (1)

Note that  $\chi$  is **additive**: for  $Z_1, Z_2$  constructible and disjoint we have

$$\chi(Z_1 \cup Z_2) = \chi(Z_1) + \chi(Z_2)$$

- Reduce to case of **projective** variety.

## Reduction from $\text{EULER}_{\mathbb{C}}$ to $\#\text{HN}_{\mathbb{C}}$ (1)

Note that  $\chi$  is **additive**: for  $Z_1, Z_2$  constructible and disjoint we have

$$\chi(Z_1 \cup Z_2) = \chi(Z_1) + \chi(Z_2)$$

- Reduce to case of **projective** variety.

Embed  $Z \subseteq \mathbb{C}^n$  into  $Z_h \subseteq \mathbb{P}^n$  (homogenize equations). By additivity

$$\chi(Z) = \chi(Z_h) - \chi(Z_h \setminus Z); \text{ note that } Z_h \setminus Z \subseteq \mathbb{P}^{n-1}$$

## Reduction from $\text{EULER}_{\mathbb{C}}$ to $\#\text{HN}_{\mathbb{C}}$ (1)

Note that  $\chi$  is **additive**: for  $Z_1, Z_2$  constructible and disjoint we have

$$\chi(Z_1 \cup Z_2) = \chi(Z_1) + \chi(Z_2)$$

- Reduce to case of **projective** variety.

Embed  $Z \subseteq \mathbb{C}^n$  into  $Z_h \subseteq \mathbb{P}^n$  (homogenize equations). By additivity

$$\chi(Z) = \chi(Z_h) - \chi(Z_h \setminus Z); \text{ note that } Z_h \setminus Z \subseteq \mathbb{P}^{n-1}$$

- Reduce to the case of a projective **(singular) hypersurface**.

## Reduction from $\text{EULER}_{\mathbb{C}}$ to $\#\text{HN}_{\mathbb{C}}$ (1)

Note that  $\chi$  is **additive**: for  $Z_1, Z_2$  constructible and disjoint we have

$$\chi(Z_1 \cup Z_2) = \chi(Z_1) + \chi(Z_2)$$

- Reduce to case of **projective** variety.

Embed  $Z \subseteq \mathbb{C}^n$  into  $Z_h \subseteq \mathbb{P}^n$  (homogenize equations). By additivity

$$\chi(Z) = \chi(Z_h) - \chi(Z_h \setminus Z); \text{ note that } Z_h \setminus Z \subseteq \mathbb{P}^{n-1}$$

- Reduce to the case of a projective **(singular) hypersurface**.

Additivity of  $\chi$  and the principle of inclusion and exclusion implies

$$\chi(\mathcal{Z}(f_1, \dots, f_r)) = \sum_{I \neq \emptyset} (-1)^{|I|-1} \chi(\mathcal{Z}(\prod_{i \in I} f_i))$$

## Reduction from $\text{EULER}_{\mathbb{C}}$ to $\#\text{HN}_{\mathbb{C}}$ (1)

Note that  $\chi$  is **additive**: for  $Z_1, Z_2$  constructible and disjoint we have

$$\chi(Z_1 \cup Z_2) = \chi(Z_1) + \chi(Z_2)$$

- Reduce to case of **projective** variety.

Embed  $Z \subseteq \mathbb{C}^n$  into  $Z_h \subseteq \mathbb{P}^n$  (homogenize equations). By additivity

$$\chi(Z) = \chi(Z_h) - \chi(Z_h \setminus Z); \text{ note that } Z_h \setminus Z \subseteq \mathbb{P}^{n-1}$$

- Reduce to the case of a projective **(singular) hypersurface**.

Additivity of  $\chi$  and the principle of inclusion and exclusion implies

$$\chi(\mathcal{Z}(f_1, \dots, f_r)) = \sum_{I \neq \emptyset} (-1)^{|I|-1} \chi(\mathcal{Z}(\prod_{i \in I} f_i))$$

The cost of the addition of exponentially many terms can be “passed to the oracle” by the Summation Lemma (**we ignore some technical difficulties**)

## Reduction from EULER to #HN (2)

## Reduction from EULER to #HN (2)

- Let  $f$  be homogeneous,  $\Sigma := \mathcal{Z}(\partial_0 f, \dots, \partial_n f)$ , and  $\Gamma \subseteq \mathbb{P}^n \times \mathbb{P}^n$  be the closure of the graph of the gradient map  $\text{grad } f: \mathbb{P}^n - \Sigma \rightarrow \mathbb{P}^n$

## Reduction from EULER to #HN (2)

- Let  $f$  be homogeneous,  $\Sigma := \mathcal{Z}(\partial_0 f, \dots, \partial_n f)$ , and  $\Gamma \subseteq \mathbb{P}^n \times \mathbb{P}^n$  be the closure of the graph of the gradient map  $\text{grad } f: \mathbb{P}^n - \Sigma \rightarrow \mathbb{P}^n$
- The **multidegree**  $d_i$  is characterized as  $d_i = \#(\Gamma \cap (L^i \times L^{n-i}))$ , where  $L^i$  denotes a linear subspace of  $\mathbb{P}^n$  of codimension  $i$  in general position

## Reduction from EULER to #HN (2)

- Let  $f$  be homogeneous,  $\Sigma := \mathcal{Z}(\partial_0 f, \dots, \partial_n f)$ , and  $\Gamma \subseteq \mathbb{P}^n \times \mathbb{P}^n$  be the closure of the graph of the gradient map  $\text{grad } f: \mathbb{P}^n - \Sigma \rightarrow \mathbb{P}^n$
- The **multidegree**  $d_i$  is characterized as  $d_i = \#(\Gamma \cap (L^i \times L^{n-i}))$ , where  $L^i$  denotes a linear subspace of  $\mathbb{P}^n$  of codimension  $i$  in general position
- Use a formula due to **Aluffi** (2003)

$$\chi(\mathcal{Z}(f)) = n + \sum_{i=1}^n (-1)^{i-1} d_{n-i}$$

## Reduction from EULER to #HN (2)

- Let  $f$  be homogeneous,  $\Sigma := \mathcal{Z}(\partial_0 f, \dots, \partial_n f)$ , and  $\Gamma \subseteq \mathbb{P}^n \times \mathbb{P}^n$  be the closure of the graph of the gradient map  $\text{grad } f: \mathbb{P}^n - \Sigma \rightarrow \mathbb{P}^n$
- The **multidegree**  $d_i$  is characterized as  $d_i = \#(\Gamma \cap (L^i \times L^{n-i}))$ , where  $L^i$  denotes a linear subspace of  $\mathbb{P}^n$  of codimension  $i$  in general position
- Use a formula due to **Aluffi** (2003)

$$\chi(\mathcal{Z}(f)) = n + \sum_{i=1}^n (-1)^{i-1} d_{n-i}$$

- If  $L^i \times L^{n-i}$  intersects  $\Gamma$  **transversally** and does not meet the exceptional subvariety  $\Gamma \cap (\Sigma \times \mathbb{P}^n)$ , then  $d_i = \#(\Gamma \cap (L^i \times L^{n-i}))$

## Reduction from EULER to #HN (2)

- Let  $f$  be homogeneous,  $\Sigma := \mathcal{Z}(\partial_0 f, \dots, \partial_n f)$ , and  $\Gamma \subseteq \mathbb{P}^n \times \mathbb{P}^n$  be the closure of the graph of the gradient map  $\text{grad } f: \mathbb{P}^n - \Sigma \rightarrow \mathbb{P}^n$
- The **multidegree**  $d_i$  is characterized as  $d_i = \#(\Gamma \cap (L^i \times L^{n-i}))$ , where  $L^i$  denotes a linear subspace of  $\mathbb{P}^n$  of codimension  $i$  in general position
- Use a formula due to **Aluffi** (2003)

$$\chi(\mathcal{Z}(f)) = n + \sum_{i=1}^n (-1)^{i-1} d_{n-i}$$

- If  $L^i \times L^{n-i}$  intersects  $\Gamma$  **transversally** and does not meet the exceptional subvariety  $\Gamma \cap (\Sigma \times \mathbb{P}^n)$ , then  $d_i = \#(\Gamma \cap (L^i \times L^{n-i}))$
- We express the transversality condition by a formula in  $\text{PH}_{\mathbb{R}}$  etc

## Reduction from EULER to #HN (2)

- Let  $f$  be homogeneous,  $\Sigma := \mathcal{Z}(\partial_0 f, \dots, \partial_n f)$ , and  $\Gamma \subseteq \mathbb{P}^n \times \mathbb{P}^n$  be the closure of the graph of the gradient map  $\text{grad } f: \mathbb{P}^n - \Sigma \rightarrow \mathbb{P}^n$
- The **multidegree**  $d_i$  is characterized as  $d_i = \#(\Gamma \cap (L^i \times L^{n-i}))$ , where  $L^i$  denotes a linear subspace of  $\mathbb{P}^n$  of codimension  $i$  in general position
- Use a formula due to **Aluffi** (2003)

$$\chi(\mathcal{Z}(f)) = n + \sum_{i=1}^n (-1)^{i-1} d_{n-i}$$

- If  $L^i \times L^{n-i}$  intersects  $\Gamma$  **transversally** and does not meet the exceptional subvariety  $\Gamma \cap (\Sigma \times \mathbb{P}^n)$ , then  $d_i = \#(\Gamma \cap (L^i \times L^{n-i}))$
- We express the transversality condition by a formula in  $\text{PH}_{\mathbb{R}}$  etc
- For rigorous proof, we introduce **generic parsimonious reductions**

# **Complexity of Computing the Hilbert Polynomial**

## Complexity of Computing the Hilbert Polynomial

- The Hilbert polynomial of a projective variety  $Z$  encodes important information about  $V$ , like its degree, dimension, arithmetic genus

## Complexity of Computing the Hilbert Polynomial

- The Hilbert polynomial of a projective variety  $Z$  encodes important information about  $V$ , like its degree, dimension, arithmetic genus
- $\text{HILBERT}_{\text{sm}}$ : *Compute the Hilbert polynomial of a given **smooth** equidimensional projective variety*

## Complexity of Computing the Hilbert Polynomial

- The Hilbert polynomial of a projective variety  $Z$  encodes important information about  $V$ , like its degree, dimension, arithmetic genus
- $\text{HILBERT}_{\text{sm}}$ : *Compute the Hilbert polynomial of a given **smooth** equidimensional projective variety*
- Theorem [B-Lotz 05]  $\text{HILBERT}_{\text{sm}}$  can be reduced in polynomial time to  $\#\text{HN}_{\mathbb{C}}$ . The analogous statement is true for the Turing model.

## Complexity of Computing the Hilbert Polynomial

- The Hilbert polynomial of a projective variety  $Z$  encodes important information about  $V$ , like its degree, dimension, arithmetic genus
- $\text{HILBERT}_{\text{sm}}$ : *Compute the Hilbert polynomial of a given **smooth** equidimensional projective variety*
- Theorem [B-Lotz 05]  $\text{HILBERT}_{\text{sm}}$  can be reduced in polynomial time to  $\#\text{HN}_{\mathbb{C}}$ . The analogous statement is true for the Turing model.
- It seems that even the following is new

## Complexity of Computing the Hilbert Polynomial

- The Hilbert polynomial of a projective variety  $Z$  encodes important information about  $V$ , like its degree, dimension, arithmetic genus
- $\text{HILBERT}_{\text{sm}}$ : *Compute the Hilbert polynomial of a given **smooth** equidimensional projective variety*
- Theorem [B-Lotz 05]  $\text{HILBERT}_{\text{sm}}$  can be reduced in polynomial time to  $\#\text{HN}_{\mathbb{C}}$ . The analogous statement is true for the Turing model.
- It seems that even the following is new

Corollary  $\text{HILBERT}_{\text{sm}} \in \text{FPAR}_{\mathbb{C}}$

## Complexity of Computing the Hilbert Polynomial

- The Hilbert polynomial of a projective variety  $Z$  encodes important information about  $V$ , like its degree, dimension, arithmetic genus
- $\text{HILBERT}_{\text{sm}}$ : *Compute the Hilbert polynomial of a given **smooth** equidimensional projective variety*
- Theorem [B-Lotz 05]  $\text{HILBERT}_{\text{sm}}$  can be reduced in polynomial time to  $\#\text{HN}_{\mathbb{C}}$ . The analogous statement is true for the Turing model.
- It seems that even the following is new

**Corollary**  $\text{HILBERT}_{\text{sm}} \in \text{FPAR}_{\mathbb{C}}$

- The proof is based on sophisticated concepts from enumerative geometry

# **Very Rough Outline of Proof**

## Very Rough Outline of Proof

- Let  $Z$  be an  $n$ -dimensional smooth projective subvariety of  $\mathbb{P}^m$ . Consider the **Gauss map**  $\varphi: Z \rightarrow \text{Grass}(n, m)$ ,  $x \mapsto T_x Z$  (generalizes the gradient map in the hypersurface case)

## Very Rough Outline of Proof

- Let  $Z$  be an  $n$ -dimensional smooth projective subvariety of  $\mathbb{P}^m$ . Consider the **Gauss map**  $\varphi: Z \rightarrow \text{Grass}(n, m)$ ,  $x \mapsto T_x Z$  (generalizes the gradient map in the hypersurface case)
- The homology class of the graph of  $\varphi$  in  $\mathbb{P}^m \times \text{Grass}(n, m)$  can be characterized by integers (generalizing multidegrees of the hypersurface case)

## Very Rough Outline of Proof

- Let  $Z$  be an  $n$ -dimensional smooth projective subvariety of  $\mathbb{P}^m$ . Consider the **Gauss map**  $\varphi: Z \rightarrow \text{Grass}(n, m)$ ,  $x \mapsto T_x Z$  (generalizes the gradient map in the hypersurface case)
- The homology class of the graph of  $\varphi$  in  $\mathbb{P}^m \times \text{Grass}(n, m)$  can be characterized by integers (generalizing multidegrees of the hypersurface case)
- These integers can be characterized as the number of intersection points of the graph of  $\varphi$  with a product of a linear subspace and a “Schubert variety” in general position (Schubert calculus)

## Very Rough Outline of Proof

- Let  $Z$  be an  $n$ -dimensional smooth projective subvariety of  $\mathbb{P}^m$ . Consider the **Gauss map**  $\varphi: Z \rightarrow \text{Grass}(n, m)$ ,  $x \mapsto T_x Z$  (generalizes the gradient map in the hypersurface case)
- The homology class of the graph of  $\varphi$  in  $\mathbb{P}^m \times \text{Grass}(n, m)$  can be characterized by integers (generalizing multidegrees of the hypersurface case)
- These integers can be characterized as the number of intersection points of the graph of  $\varphi$  with a product of a linear subspace and a “Schubert variety” in general position (Schubert calculus)
- From these integers one can obtain the so-called **Chern numbers** of  $Z$  (in particular, the topological Euler characteristic)

## Very Rough Outline of Proof

- Let  $Z$  be an  $n$ -dimensional smooth projective subvariety of  $\mathbb{P}^m$ . Consider the **Gauss map**  $\varphi: Z \rightarrow \text{Grass}(n, m), x \mapsto T_x Z$  (generalizes the gradient map in the hypersurface case)
- The homology class of the graph of  $\varphi$  in  $\mathbb{P}^m \times \text{Grass}(n, m)$  can be characterized by integers (generalizing multidegrees of the hypersurface case)
- These integers can be characterized as the number of intersection points of the graph of  $\varphi$  with a product of a linear subspace and a “Schubert variety” in general position (Schubert calculus)
- From these integers one can obtain the so-called **Chern numbers** of  $Z$  (in particular, the topological Euler characteristic)
- The Hilbert polynomial can be computed from the Chern numbers via the **Hirzebruch-Grothendieck-Riemann-Roch theorem**

## IV. Unrestricted BSS-model over $\mathbb{R}$

## IV. Unrestricted BSS-model over $\mathbb{R}$

- There is a notion of a **modified Euler characteristic**  $\chi^*(S)$  of a semialgebraic set  $S$  that behaves additively on finite disjoint unions

## IV. Unrestricted BSS-model over $\mathbb{R}$

- There is a notion of a **modified Euler characteristic**  $\chi^*(S)$  of a semialgebraic set  $S$  that behaves additively on finite disjoint unions
- For  $S$  compact we have  $\chi^*(S) = \chi(S)$

## IV. Unrestricted BSS-model over $\mathbb{R}$

- There is a notion of a **modified Euler characteristic**  $\chi^*(S)$  of a semialgebraic set  $S$  that behaves additively on finite disjoint unions
- For  $S$  compact we have  $\chi^*(S) = \chi(S)$
- For  $S \subseteq \mathbb{C}^n$  constructible we have  $\chi^*(S) = \chi(S)$

## IV. Unrestricted BSS-model over $\mathbb{R}$

- There is a notion of a **modified Euler characteristic**  $\chi^*(S)$  of a semialgebraic set  $S$  that behaves additively on finite disjoint unions
- For  $S$  compact we have  $\chi^*(S) = \chi(S)$
- For  $S \subseteq \mathbb{C}^n$  constructible we have  $\chi^*(S) = \chi(S)$
- **Theorem [BC 03]** The problem to compute the modified Euler characteristic of a semialgebraic set is polynomial time equivalent to  $\text{FEAS}_{\mathbb{R}}$ , that is, it is  $\text{FP}_{\mathbb{R}}^{\#\text{P}_{\mathbb{R}}}$ -complete

## IV. Unrestricted BSS-model over $\mathbb{R}$

- There is a notion of a **modified Euler characteristic**  $\chi^*(S)$  of a semialgebraic set  $S$  that behaves additively on finite disjoint unions
- For  $S$  compact we have  $\chi^*(S) = \chi(S)$
- For  $S \subseteq \mathbb{C}^n$  constructible we have  $\chi^*(S) = \chi(S)$
- **Theorem [BC 03]** The problem to compute the modified Euler characteristic of a semialgebraic set is polynomial time equivalent to  $\text{FEAS}_{\mathbb{R}}$ , that is, it is  $\text{FP}_{\mathbb{R}}^{\#\text{P}_{\mathbb{R}}}$ -complete
- Ingredients of the proof:

## IV. Unrestricted BSS-model over $\mathbb{R}$

- There is a notion of a **modified Euler characteristic**  $\chi^*(S)$  of a semialgebraic set  $S$  that behaves additively on finite disjoint unions
- For  $S$  compact we have  $\chi^*(S) = \chi(S)$
- For  $S \subseteq \mathbb{C}^n$  constructible we have  $\chi^*(S) = \chi(S)$
- **Theorem [BC 03]** The problem to compute the modified Euler characteristic of a semialgebraic set is polynomial time equivalent to  $\text{FEAS}_{\mathbb{R}}$ , that is, it is  $\text{FP}_{\mathbb{R}}^{\#\text{P}_{\mathbb{R}}}$ -complete
- Ingredients of the proof:
  - reduction of semialgebraic sets to smooth hypersurfaces (algebraic topology)

## IV. Unrestricted BSS-model over $\mathbb{R}$

- There is a notion of a **modified Euler characteristic**  $\chi^*(S)$  of a semialgebraic set  $S$  that behaves additively on finite disjoint unions
- For  $S$  compact we have  $\chi^*(S) = \chi(S)$
- For  $S \subseteq \mathbb{C}^n$  constructible we have  $\chi^*(S) = \chi(S)$
- **Theorem [BC 03]** The problem to compute the modified Euler characteristic of a semialgebraic set is polynomial time equivalent to  $\text{FEAS}_{\mathbb{R}}$ , that is, it is  $\text{FP}_{\mathbb{R}}^{\#\text{P}_{\mathbb{R}}}$ -complete
- Ingredients of the proof:
  - reduction of semialgebraic sets to smooth hypersurfaces (algebraic topology)
  - Morse theory (differential topology)

## IV. Unrestricted BSS-model over $\mathbb{R}$

- There is a notion of a **modified Euler characteristic**  $\chi^*(S)$  of a semialgebraic set  $S$  that behaves additively on finite disjoint unions
- For  $S$  compact we have  $\chi^*(S) = \chi(S)$
- For  $S \subseteq \mathbb{C}^n$  constructible we have  $\chi^*(S) = \chi(S)$
- **Theorem [BC 03]** The problem to compute the modified Euler characteristic of a semialgebraic set is polynomial time equivalent to  $\text{FEAS}_{\mathbb{R}}$ , that is, it is  $\text{FP}_{\mathbb{R}}^{\#\text{P}_{\mathbb{R}}}$ -complete
- Ingredients of the proof:
  - reduction of semialgebraic sets to smooth hypersurfaces (algebraic topology)
  - Morse theory (differential topology)
  - partial witness sequences

# Open Problems

## Open Problems

- Show completeness of the real knapsack problem in the additive model with respect to many-one reductions!

## Open Problems

- Show completeness of the real knapsack problem in the additive model with respect to many-one reductions!
- Characterize GCC in terms of known classical complexity classes!

## Open Problems

- Show completeness of the real knapsack problem in the additive model with respect to many-one reductions!
- Characterize GCC in terms of known classical complexity classes!
- Give natural examples of  $\text{FPAR}_{\mathbb{C}}$ -complete or  $\text{PAR}_{\mathbb{R}}$ -complete problems!

## Open Problems

- Show completeness of the real knapsack problem in the additive model with respect to many-one reductions!
- Characterize GCC in terms of known classical complexity classes!
- Give natural examples of  $\text{FPAR}_{\mathbb{C}}$ -complete or  $\text{PAR}_{\mathbb{R}}$ -complete problems!
- Is  $\text{FPAR}_{\mathbb{C}} = \text{P}_{\mathbb{C}}^{\text{PSPACE}}$ ?

## Open Problems

- Show completeness of the real knapsack problem in the additive model with respect to many-one reductions!
- Characterize GCC in terms of known classical complexity classes!
- Give natural examples of  $\text{FPAR}_{\mathbb{C}}$ -complete or  $\text{PAR}_{\mathbb{R}}$ -complete problems!
- Is  $\text{FPAR}_{\mathbb{C}} = \text{P}_{\mathbb{C}}^{\text{PSPACE}}$ ?
- Relate the BSS-model to Valiant's model. Could BSS-counting classes be helpful for this?