$\mathbf{P} \neq \mathbf{NP}$ for all infinite Boolean rings

after M. Prunescu

Slides prepared by Iddo Bentov
for the Technion advanced course CS-238900 (2013)
The question $\mathbf{P} = \mathbf{NP}$ on arbitrary structures

given by J.A. Makowsky

```
@article{DBLP:journals/mlq/Prunescu03,
  author    = {Mihai Prunescu},
  title     = {P != NP for all infinite Boolean algebras},
  journal   = {Math. Log. Q.},
  volume    = {49},
  number    = {2},
  year      = {2003},
  pages     = {210-213},
  ee        = {http://dx.doi.org/10.1002/malq.200310020},
  bibsource = {DBLP, http://dblp.uni-trier.de}
}
```

$\mathbf{P} \neq \mathbf{NP}$ for all infinite Boolean rings

A ring $\mathcal{R}$ (with 1) is a Boolean ring if $\forall x \in \mathcal{R} : x^2 = x$.

Theorem:(Marshall H. Stone) Let $\mathcal{R}$ be a boolean ring. There exists a set S and a ring $\mathcal{R}' \subseteq 2^S$ with the operations

- $A + B \triangleq (A \cap \overline{B}) \cup (\overline{A} \cap B)$
- $A \cdot B \triangleq A \cap B$
- $0_{\mathcal{R}'} \triangleq \emptyset$ and $1_{\mathcal{R}'} \triangleq S$

such that $\mathcal{R} \simeq \mathcal{R}'$. We refer to $\mathcal{R}'$ as a **Boolean algebra**.

For a Boolean ring $\mathcal{R}$, we assume that an $\mathcal{R}$-machine utilizes only the operations $\{\bar{\ }, \cap\}$, and branches only via tests of the form $A = 0$. This is w.l.o.g., since de Morgan's rule $A \cup B = \overline{\overline{A} \cap \overline{B}}$ allows us to translate the $\cup$ operation in constant time, and $A = B$ iff $(\overline{A} \cap B = 0) \wedge (A \cap \overline{B} = 0)$.

The Zero-Divisors problem

For a set X, let $X^0 \triangleq X$ and $X^1 \triangleq \overline{X}$.

For a Boolean algebra $\mathcal{R}$ we define the Zero-Divisors problem $ZD_{\mathcal{R}} \subseteq \mathcal{R}^{\infty}$, as follows:

$$ZD_{\mathcal{R}} = \{(x_1, \ldots, x_n) : n \in \mathbb{N} \text{ and } \exists \varepsilon_1, \ldots, \varepsilon_n \in \{0,1\} \text{ s.t. } x_1^{\varepsilon_1} \cap \cdots \cap x_n^{\varepsilon_n} = 0\}$$

To take an example, if $\mathcal{R} = 2^{\{a,b,c,d\}}$ then $(\{a,b\},\{a,c\}) \notin ZD_{\mathcal{R}}$.

Observe that:

- $ZD_{\mathcal{R}} \in \mathbf{NP}_{\mathcal{R}}$ via a parameter-free $\mathcal{R}$-machine that makes Boolean guesses.
- $ZD_{\mathcal{R}}$ is computable in exponential time deterministically.

Prunescu's theorem for infinite Boolean rings

Theorem:(Mihai Prunescu) For any infinite Boolean ring $\mathcal{R}$, there does not exist a deterministic $\mathcal{R}$-machine using arbitrary fixed constants $c_1, \ldots, c_k \in \mathcal{R}$ that can decide $ZD_{\mathcal{R}}$ in polynomial time.

- Consequently, $\mathbf{P}_{\mathcal{R}} \neq \mathbf{NP}_{\mathcal{R}}$ for any Boolean ring $\mathcal{R}$.

We shall initially prove this theorem for a parameter-free $\mathcal{R}$-machine.

Boolean algebras: preliminaries

Definition: $x \in \mathcal{R}$ is called an *atom* of $\mathcal{R}$ if $x \neq 0 \wedge \forall y : y \subseteq x \rightarrow (y = 0 \vee y = x)$. A Boolean algebra is called *atomic* if every element contains an atom.

Observation: Every finite Boolean algebra is atomic, and isomorphic to the power-set of its atoms.

Proof sketch:

- Let $\mathcal{R}_n$ be a finite Boolean algebra with n elements. Pick any $x \in \mathcal{R}_n$.
- If x is not an atom then $\exists y \in \mathcal{R}_n$ such that $0 \neq y \subsetneq x$.
- If y is not an atom then $\exists z \in \mathcal{R}_n$ such that $0 \neq z \subsetneq y$.
- After at most $n-1$ such iterations, we conclude that x contains an atom.
- Hence $\mathcal{R}_n$ is atomic, and contains the atoms $a_1, \ldots, a_k$ for some k.
- It is a simple exercise to see that $2^{\{a_1,\ldots,a_k\}} \simeq \mathcal{R}_n$.

Boolean algebras: preliminaries (cont'd)

Definition: A subalgebra $\mathcal{R}'$ of $\mathcal{R}$ is a Boolean algebra $\mathcal{R}' \subseteq \mathcal{R}$ with $1_{\mathcal{R}'} = 1_{\mathcal{R}}, 0_{\mathcal{R}'} = 0_{\mathcal{R}}$, and $\{\bar{}, \cup, \cap\}$ operations induced by $\mathcal{R}$.

Note: in other words, $\mathcal{R}'$ is a subring of a Boolean ring (containing 1) $\mathcal{R}$.

The following lemma will be instrumental in proving Prunescu's theorem:

Lemma 1
Let $\mathcal{R}$ be an infinite Boolean algebra. For every finite algebra $\mathcal{B}_n$ with n atoms, $\mathcal{R}$ contains a subalgebra that is isomorphic to $\mathcal{B}_n$.

Boolean algebras: preliminaries (cont'd)

Definition: A Boolean algebra $\mathcal{R}$ is called an internal product of two Boolean algebras B and C, denoted by $\mathcal{R} = B \otimes C$, if:

- $B \subseteq \mathcal{R}$ and $C \subseteq \mathcal{R}$ (as sets, not as subrings)
- Every element $r \in \mathcal{R}$ can be written in exactly one way as $r = b \cup c$ with $b \in B$ and $c \in C$
- $(b_1 \cup c_1) \cap (b_2 \cup c_2) = (b_1 \cap b_2) \cup (c_1 \cap c_2)$ for all $b_1, b_2 \in B$ and $c_1, c_2 \in C$
- $(b_1 \cup c_1) \cup (b_2 \cup c_2) = (b_1 \cup b_2) \cup (c_1 \cup c_2)$ for all $b_1, b_2 \in B$ and $c_1, c_2 \in C$
- $\overline{b \cup c} = \overline{b} \cup \overline{c}$ for all $b \in B$ and $c \in C$

Note that with this notation we mean that for example $b_1 \cap b_2$ and $\overline{b}$ are formed in B, hence $B \otimes C$ is regarded as an internal decomposition of $\mathcal{R}$.

Boolean algebras: preliminaries (cont'd)

Definition: The relativization of a Boolean algebra $\mathcal{R}$ to an element $x \in \mathcal{R}$ is $D_{\mathcal{R}}(x) \triangleq \{x \cap r : r \in \mathcal{R}\}$.

Lemma 2

Any nontrivial Boolean algebra $\mathcal{R} \neq \{0, 1\}$ is internally decomposable.

Proof: pick $x \in \mathcal{R} \setminus \{0, 1\}$, $B = D_{\mathcal{R}}(x)$, $C = D_{\mathcal{R}}(\overline{x})$.

- We claim that $\mathcal{R} = B \otimes C$.
- Consider an arbitrary $r \in \mathcal{R}$, hence $r \cap x = b \in B$ and $r \cap \overline{x} = c \in C$.
- $r = r \cap 1 = r \cap (x \cup \overline{x}) = (r \cap x) \cup (r \cap \overline{x}) = b \cup c$, and if $r = b' \cup c'$ then $b = r \cap x = (b' \cup c') \cap x = (b' \cap x) \cup (c' \cap x) = b' \cup 0 = b'$. Similarly, $c = c'$.
- Via distributive law, $\forall b_1, b_2 \in B, c_1, c_2 \in C : (b_1 \cup c_1) \cap (b_2 \cup c_2) = [(b_1 \cap b_2) \cup (c_1 \cap b_2)] \cup [(b_1 \cap c_2) \cup (c_1 \cap c_2)] = [(b_1 \cap b_2) \cup 0] \cup [0 \cup (c_1 \cap c_2)] = (b_1 \cap b_2) \cup (c_1 \cap c_2)$.
- $\forall b \in B, c \in C : \overline{b \cup c} = (\overline{b} \cap \overline{c}) \cap (x \cup \overline{x}) = (\overline{b} \cap \overbrace{\overline{c} \cap x}^{=x}) \cup (\overline{c} \cap \overbrace{\overline{b} \cap \overline{x}}^{=\overline{x}}) = (\overline{b} \cap x) \cup (\overline{c} \cap \overline{x}).\square$

Boolean algebras: Lemma 1

Reminder: Lemma 1 states that any infinite Boolean algebra $\mathcal{R}$ contains subalgebras isomorphic to $\mathcal{B}_n$, for all finite subalgebras $\mathcal{B}_n$ with n atoms.

Proof of Lemma 1: By induction on n.

- If $n = 1$ then the subalgebra $\{0_\mathcal{R}, 1_\mathcal{R}\} \subset \mathcal{R}$ is isomorphic to $\mathcal{B}_1$.
- Assume that the lemma holds for an integer n.
- Since $\mathcal{R}$ is nontrivial, by lemma 2 we can write $\mathcal{R} = C \otimes D$.
- $\mathcal{R}$ is infinite, so at least one of the algebras C, D, say D, is infinite.
- $\mathcal{B}_1 \simeq \{0_C, 1_C\} = C' \subseteq C$.
- By the induction hypothesis $\exists D' \subset D$ such that $D' \simeq \mathcal{B}_n$.
- $C' \otimes D' \simeq \mathcal{B}_1 \otimes \mathcal{B}_n \simeq \mathcal{B}_1 \otimes \mathcal{B}_1^n \simeq \mathcal{B}_1^{n+1} \simeq \mathcal{B}_{n+1}$. (note: $1_{C' \otimes D'} = 1_\mathcal{R}$)
- $\Rightarrow \exists \mathcal{R}' = C' \otimes D' \subset \mathcal{R}$ for which $\mathcal{R}' \simeq \mathcal{B}_{n+1}$. $\square$

Prunescu's proof for the parameter-free case

Lemma 3
If there exist an infinite Boolean ring $\mathcal{R}$ and a parameter-free $\mathcal{R}$-machine $\mathcal{M}$ that deterministically decides $ZD_{\mathcal{R}}$ in some polynomial time $q(t)$, then $\mathcal{M}$ decides $ZD_{\mathcal{B}_n}$ in the same deterministic polynomial time $q(t)$, uniformly in n for all finite Boolean algebras $\mathcal{B}_n$.

Proof:

- By lemma 1, every finite algebra $\mathcal{B}_n$ is isomorphic to a subalgebra $\mathcal{R}_n \subset \mathcal{R}$.
- $\mathcal{M}$ is parameter-free $\Rightarrow$ for inputs from $\mathcal{R}_n^{\infty}$, all the computations remain in the subalgebra $\mathcal{R}_n$. $\square$

Prunescu's proof for the parameter-free case (cont'd)

Lemma 4
There is no parameter-free deterministic machine $\mathcal{M}$ that can decide $ZD_{\mathcal{B}_n}$ in some uniform polynomial time $q(k)$ over all finite Boolean algebras $\mathcal{B}_n$.

Note: it may still be possible that for every given finite algebra $\mathcal{B}_n$, there exists a deterministic $\mathcal{B}_n$-machine that decides $ZD_{\mathcal{B}_n}$ in polynomial time. In fact, the claim that $\mathbf{P}_{\mathcal{B}_n} = \mathbf{NP}_{\mathcal{B}_n}$ holds for every finite Boolean algebra $\mathcal{B}_n$ is equivalent to the classical $\mathbf{P}$ versus $\mathbf{NP}$ question.

Freely generated finite Boolean algebras

Definition: $\mathcal{F}_n$ is a Boolean algebra that is freely generated by $X_1, \ldots, X_n$:

- Each atom of $\mathcal{F}_n$ is of the form $X_1^{\varepsilon_1} \cap \cdots \cap X_n^{\varepsilon_n} \neq 0$ for $\varepsilon_1, \ldots, \varepsilon_n \in \{0, 1\}$
- $\Rightarrow \mathcal{F}_n$ has 2^n atoms, hence $|\mathcal{F}_n| = 2^{2^n}$ and $\mathcal{F}_n \simeq \mathcal{B}_{2^n}$

Sidenote: the countably infinite freely generated Boolean algebra is atomless.

To obtain an isomorphism $\mathcal{B}_{2^n} \simeq \mathcal{F}_n$, define for $i = 1, \ldots, 2^n$ and $t = 1, \ldots, n$:

- $i \in X_t \iff 2^{t-1}$ occurs in the sum of two-powers representing i

Claim: $X_1^{\varepsilon_1} \cap \cdots \cap X_n^{\varepsilon_n} = \{i\}$ for some atom $\{i\} \in \mathcal{B}_{2^n}$

Freely generated finite Boolean algebras (cont'd)

To see this, denote by $(b_1, \ldots, b_n)$ the bit-representation of an n-bit number, where b_1 is the least significant bit:

- $X_t^0 = X_t = \{i : 1 \leq i < 2^n, i = (b_1, \ldots, b_n), b_t = 1\}$
- $X_t^1 = \overline{X_t} = \{i : 1 \leq i < 2^n, i = (b_1, \ldots, b_n), b_t = 0\} \cup \{2^n\}$
- $\sum_{j=1}^n (1 - \varepsilon_j) \neq 0 \Rightarrow X_1^{\varepsilon_1} \cap \cdots \cap X_n^{\varepsilon_n} = \{i : i = (1 - \varepsilon_1, \ldots, 1 - \varepsilon_n)\}$
- $\sum_{j=1}^n (1 - \varepsilon_j) = 0 \Rightarrow X_1^{\varepsilon_1} \cap \cdots \cap X_n^{\varepsilon_n} = \{2^n\}$

Therefore:

- $(\varepsilon_1, \ldots, \varepsilon_n) \neq (\varepsilon_1', \ldots, \varepsilon_n') \Rightarrow X_1^{\varepsilon_1} \cap \cdots \cap X_n^{\varepsilon_n} \neq X_1^{\varepsilon_1'} \cap \cdots \cap X_n^{\varepsilon_n'}$
- $|X_1^{\varepsilon_1} \cap \cdots \cap X_n^{\varepsilon_n}| = 1$

For example, if $n = 3$ then $X_1 = \{1, 3, 5, 7\}, \overline{X_1} = \{2, 4, 6, 8\}, X_2 = \{2, 3, 6, 7\}, \ldots$

Prunescu's proof for the parameter-free case (cont'd)

Proof of Lemma 4: Assume that there exists a deterministic machine $\mathcal{M}$ with a polynomial time bound $q(k)$ that decides $ZD_{\mathcal{B}_n}$ over all $\mathcal{B}_n$ uniformly:

- pick k for which $2^k > q(k)$, and consider inputs from $\mathcal{F}_k$.
- Let $X_1, \ldots, X_k$ be the elements that freely generate $\mathcal{F}_k$.
- Consider the input tuple $\vec{X} = (X_1, \ldots, X_k)$.
- Notice that $\vec{X} \notin ZD_{\mathcal{F}_k}$, thus if our assumption holds then $\mathcal{M}$ rejects $\vec{X}$.
- $\mathcal{M}$ tested $\leq q(k)$ atoms $X_1^{\varepsilon_1} \cap \cdots \cap X_k^{\varepsilon_k} \overset{?}{=} 0$ along its computation path.
- Since $2^k > q(k)$, $\mathcal{M}$ did not test at least one atom $\{i_0\} \in \{\{1\}, \ldots, \{2^k\}\}$.
- Let $Y_j = X_j \setminus \{i_0\}$, and consider the input tuple $\vec{Y} = (Y_1, \ldots, Y_k) \in \mathcal{B}^{\infty}_{2^k-1}$.

Prunescu's proof for the parameter-free case (cont'd)

- Notice that $\forall i \neq i_0 : \{i\} = X_1^{\varepsilon_1} \cap \cdots \cap X_k^{\varepsilon_k} = Y_1^{\varepsilon_1} \cap \cdots \cap Y_k^{\varepsilon_k}$, therefore $Y_1, \ldots, Y_k$ indeed generate $\mathcal{B}_{2^k-1}$.

- Hence we obtained that $\vec{Y} \in ZD_{\mathcal{B}_{2^k-1}}$, because the atom $\{i_0\} = X_1^{\varepsilon_1} \cap \cdots \cap X_k^{\varepsilon_k}$ of $\mathcal{F}_k$ corresponds to $Y_1^{\varepsilon_1} \cap \cdots \cap Y_k^{\varepsilon_k} = 0$ in $\mathcal{B}_{2^k-1}$.

- $\mathcal{M}(\vec{X})$ branched by testing the emptiness of sets that either contained at least two elements or did not contain i_0, therefore $\mathcal{M}$ also rejects $\vec{Y}$.

- Even if $\mathcal{M}(\vec{X})$ tests the emptiness of $\emptyset$, $\mathcal{M}(\vec{Y})$ will follow the same computation path, but no such tests occur if $\mathcal{M}$ runs a minimal program.

- Thus the positive input $\vec{Y}$ is rejected by $\mathcal{M}$, which is a contradiction. $\square$

Prunescu's proof for a machine with fixed constants

Lemma 5
Let $\mathcal{R}$ be an infinite Boolean algebra. If there exist constants $c_1, \ldots, c_t \in \mathcal{R}$ and a deterministic $\mathcal{R}$-machine $\mathcal{M}$ that utilizes these constants to decide $ZD_{\mathcal{R}}$ in some polynomial time $q(k)$, then there exist an infinite Boolean algebra $\mathcal{R}_1$ and a parameter-free deterministic $\mathcal{R}_1$-machine $\mathcal{M}_1$ that decides $ZD_{\mathcal{R}_1}$ in $\mathcal{O}(2^{t+1} \cdot q(k))$ time.

Proof: Let S be an infinite set and $\mathcal{R} \subseteq 2^S$.

- Let $\mathcal{C}$ be the subalgebra of $\mathcal{R}$ generated by $c_1, \ldots, c_t$, note that $\mathcal{C}$ is **finite**.
- Let $a_1, \ldots, a_d$ be the atoms of $\mathcal{C}$, hence $d \leq 2^t$.
- It is simple to verify that $a_1 \cup \cdots \cup a_d = S$, and that this is a partition.
- There must be an a_i, say a_1, such that the relativized Boolean algebra $\mathcal{R}_1 = D_{\mathcal{R}}(a_1) = \{a_1 \cap r : r \in \mathcal{R}\}$ (with $0_{\mathcal{R}_1} = \emptyset$ and $1_{\mathcal{R}_1} = a_1$) is infinite.

Prunescu's proof for a machine with fixed constants (cont'd)

- Consider an input $\vec{x} = (x_1, \ldots, x_k) \in \mathcal{R}_1^\infty$.
- **Claim:** $\vec{x} \in ZD_{\mathcal{R}_1} \iff \vec{x} \in ZD_{\mathcal{R}} \vee \mathcal{R} \models \overline{x_1} \cap \cdots \cap \overline{x_k} \cap a_1 = 0$.
- This holds because if $\exists \varepsilon_i = 0$ then $x_1^{\varepsilon_1} \cap \cdots \cap x_k^{\varepsilon_k} \subseteq x_i \subseteq a_1$ and therefore $x_1^{\varepsilon_1} \cap \cdots \cap x_k^{\varepsilon_k} = 0$ in $\mathcal{R}$, otherwise $\overline{x_1} \cap \cdots \cap \overline{x_k} = a_2 \cup \cdots \cup a_d$ in $\mathcal{R}$ and then $\overline{x_1} \cap \cdots \cap \overline{x_k} \cap a_1 = 0$ in $\mathcal{R}$.
- **Claim:** we can construct an $\mathcal{R}$-machine $\mathcal{M}'$ with constants $c_1, \ldots, c_t$ that works over $\mathcal{R}$ and decides $ZD_{\mathcal{R}_1}$ in deterministic time $q(k) + \mathcal{O}(k)$.
- $\mathcal{M}'$ on input $\vec{x} = (x_1, \ldots, x_k) \in \mathcal{R}_1^\infty$ will first invoke $\mathcal{M}(\vec{x})$ and accept if $\mathcal{M}$ accepted. Otherwise, $\mathcal{M}'$ tests $\overline{x_1} \cap \cdots \overline{x_k} \cap a_1 \stackrel{?}{=} 0$, and accepts iff the test result was positive.
- Notice that a_1 is generated by $c_1, \ldots, c_t$, therefore $\mathcal{M}'$ can form a_1 in constant time via some computation $a_1 = c_1^{\varepsilon_1} \cap \cdots \cap c_t^{\varepsilon_t}$.

Prunescu's proof for a machine with fixed constants (cont'd)

- Let $\mathcal{C}_1 = D_{\mathcal{C}}(a_2 \cup \cdots \cup a_d)$
- When $\mathcal{M}'$ runs with an input from $\mathcal{R}_1^\infty$ and utilizes the constants $c_1, \ldots, c_t$, every element that occurs during its calculations has the form $x = x' \cup x''$ where $x' \in \mathcal{R}_1$ and $x'' \in \mathcal{C}_1$.
- In other words, $\mathcal{M}'$ cannot quit the internal product $\mathcal{R}_1 \otimes \mathcal{C}_1$.
- We can modify $\mathcal{M}'$ to construct a parameter-free $\mathcal{R}_1$-machine $\mathcal{M}_1$ that decides $ZD_{\mathcal{R}_1}$.
- $\mathcal{M}_1$ replaces the input $x_1, \ldots, x_k$ with $x_1' = x_1, x_1'' = 0, \ldots, x_k' = x_k, x_k'' = 0$.
- $\mathcal{M}_1$ simulates $\mathcal{M}'$ and replaces $x := y \cap z$ with $x' := y' \cap z', x'' := y'' \cap z''$, replaces $x := \overline{y}$ with $x' := \overline{y'}, x'' := \overline{y''}$, and replaces $x \stackrel{?}{=} 0$ with the two tests $x' \stackrel{?}{=} 0 \wedge x'' \stackrel{?}{=} 0$.

Prunescu's proof for a machine with fixed constants (cont'd)

- For every c_i for which it holds that $c_i \cap a_1 = 0$, $\mathcal{M}_1$ replaces $x := x \cap c_i$ with $x' := 0, x'' := x'' \cap c_i$.
- For every c_i for which it holds that $c_i \cap a_1 = a_1$, $\mathcal{M}_1$ replaces $x := x \cap c_i$ with $x'' := x'' \cap (c_i \setminus a_1)$, and retains x'.
- $\Rightarrow$ All the computations in the registers x' do not use any constants.
- The computations in the registers x'' take place in the finite algebra $\mathcal{C}_1$ that has at most $2^t - 1$ atoms, and therefore at most $2^{2^t - 1}$ elements.
- We can hardcode two tables of size $\leq (2^{2^t-1})^2$ for the $\{^-, \cap\}$ operations, inside the program that $\mathcal{M}_1$ runs.
- Hence $\mathcal{M}_1$ can carry out the computations in $\mathcal{C}_1$ by querying these tables in time that is constant in k, and in time that is logarithmic in $2^{2^{t+1}}$ by computing the required index and accessing the memory location of the corresponding table entry.

Prunescu's proof for a machine with fixed constants (cont'd)

- We obtained that the parameter-free deterministic $\mathcal{M}_1$ simulates $\mathcal{M}'$ for inputs from $\mathcal{R}_1^\infty$, by carrying out all the computations in the internal product $\mathcal{R}_1 \otimes \mathcal{C}_1$.

- In particular, simulating the last step $\overline{x_1} \cap \cdots \overline{x_k} \cap a_1 \stackrel{?}{=} 0$ of $\mathcal{M}'$ becomes straightforward: the $\mathcal{R}_1$-machine $\mathcal{M}_1$ computes the complements $\overline{x_i}$ just over $\mathcal{R}_1$ into registers x_i', then intersects these x_i' registers and tests for emptiness.

- $\mathcal{M}_1$ has a polynomial time bound of $\mathcal{O}(2^{t+1} \cdot q(k))$. □

Prunescu's proof: conclusion

Reminder: Lemmas 3 and 4 imply that for any infinite Boolean algebra $\mathcal{R}$, it holds that a **parameter-free** $\mathcal{R}$-machine cannot decide $ZD_{\mathcal{R}}$ in deterministic polynomial time.

Reminder: Lemma 5 shows that for an infinite Boolean algebra $\mathcal{R}$, if a deterministic $\mathcal{R}$-machine with access to any arbitrary fixed constants of $\mathcal{R}$ can decide $ZD_{\mathcal{R}}$ in polynomial time, then there exists an infinite Boolean algebra $\mathcal{R}_1$ for which $ZD_{\mathcal{R}_1}$ can be decided in deterministic polynomial time by a **parameter-free** $\mathcal{R}_1$-machine.

Conclusion: by combining lemma 5 with lemma 4 and lemma 3, Prunescu's theorem follows, i.e. $\mathbf{P}_{\mathcal{R}} \neq \mathbf{NP}_{\mathcal{R}}$ for every infinite Boolean algebra $\mathcal{R}$. $\square$

Outline of the ESSLLI-course given by

J.A. Makowsky (Haifa) and K. Meer (Cottbus)

LECTURE 1 (JAM): Introduction INTRO (5 slides)
Turing machines over relational structures, NEWBSS, (19 slides)
Short quantifier elimination. SHORTQE (16 slides)

LECTURE 2 (JAM): Introduction to quantifier elimination QE (26 slides)
Fields, rings and other structures TABLE (incomplete, 20 slides)

LECTURE 3 (JAM): Computing with the reals: Removing order or multiplication;
Adding `Fortran`-libraries. FORTRAN (24 slides)
Comparing Poizat's Theorem with descriptive complexity FAGIN (20 slides)

LECTURE 4 (KM): Inside $\mathbf{NP}_{\mathbb{R}}$ and analogues to Ladner's Theorem, Meer-1 (149 slides)

LECTURE 5 (KM): PCP-Theorem over $\mathbb{R}$, Meer-2 (139 slides)

ADDITIONAL MATERIAL see next slide.

Additional material for the ESSLLI-course

LECTURE 6 (JAM): Quantifier elimination in algebraically closed fields, ACF-0 (21 slides)
By JAM after Kreisel and Krivine.

LECTURE 7 (JAM): $\mathbf{P}_G \neq \mathbf{NP}_G$ for all abelian groups. ABELIAN (18 slides)
By JAM After M. Prunescu

LECTURE 8 (JAM): $\mathbf{P}_G \neq \mathbf{NP}_G$ for all boolean algebras. BOOLEAN (23 slides)
By I. Bentov after M. Prunescu

LECTURE 9 (JAM): $\mathbf{P}_G \neq \mathbf{NP}_G$ for all real matrix rings. MATRIX (70 slides)
By N. Labai after A. Rybalov