P $\neq$ NP for all infinite Boolean rings

Mihai Prunescu [*]

Abstract

We prove that all infinite Boolean rings (algebras) have the property P $\neq$ NP according to the digital (binary) nondeterminism.
A.M.S. Classification: 06E99, 03B70.

1 Introduction

Given a ring R, we call **input** over R a finite non-empty sequence of elements of R. Let R^∞ be the set of all inputs. A **problem** $\mathfrak{P}$ over R is any set of inputs ($\mathfrak{P} \subset R^\infty$). An R**-machine** is a computation system given by a finite description and able to work out inputs of arbitrary length according to a program. Finitely many fixed constants in the ring may occur in the program as machine's parameters. If excepting 0 and 1 no other parameter occurs, we say that the machine is **parameter-free**. The length of an input is the measure of its (algebraic) complexity. By **polynomial time** we mean that the time of computation has at most a polynomial increment rate in the length of the input.

In the **binary** (called also boolean -, ramification -, or digital -) nondeterminism are allowed situations in which the machine can continue its computation in two different ways, and the decision is taken arbitrarily. The second kind of nondeterministic machines have **guess** instructions, assigning to some register any value picked up arbitrarily from the ring. If one algebraic structure contains at least two elements and possess equality one can simulate any binary nondeterministic machine using a guess-nondeterministic one.

If we interpret the structure above as a model of computation, we can define the class P_R of problems decided by deterministic machines in polynomial time and the classes NBP_R and NP_R of problems recognized by the eventually halting of digital nondeterministic machines, respectively guess-nondeterministic machines, in polynomial time. As we have seen, $P_R \subseteq NBP_R \subseteq NP_R$, so $P \neq NBP$ implies $P \neq NP$ over R.

This approach to algebraic complexity for rings has been started by Blum, Shub and Smale in [3] and continued in [2]. Goode put in [6] the bases of a general theory of computation over algebraic structures compatible with [3] and introduced the class NBP. Poizat continued this work in his book [8]. His examples concern mainly the class NBP.

It is quite easy to find rings with $P \neq NP$: choose a ring which has not quantifier elimination in the given language. However, it is more difficult to prove that a ring has the property $P \neq NBP$ or even to show that a ring admitting quantifier elimination has $P \neq NP$.

The unique example of ring known before by the author to have $P \neq NBP$ occurred in the literature in a somehow hidden way. Poizat proved in [8] that all atomless Boolean algebras have this property, but he didn't emphasize that this automatically gives an example for rings too, every Boolean algebra being a Boolean ring. In spite of this fact it continued to be widely believed that no example of ring with $P \neq NBP$ is known.

The aim of this paper is to generalize Poizat's result to all infinite Boolean algebras.

[*] Universität Greifswald, Germany; and IMAR Bucharest, Romania. e-Mail: prunescu@mail.uni-greifswald.de.

2 Boolean rings

In this Section we present general things about Boolean algebras and rings. For more information, see [1], [5], [7], [9] or every specialized monograph.

A Boolean ring is an associative ring with 1 that models the supplementary axiom $\forall x \; x^2 = x$.

Definition: Let S be a set. The power-set 2^S is a Boolean ring with the operations $A + B := (A \cap \bar{B}) \cup (\bar{A} \cap B)$, $AB := A \cap B$, $0 = \emptyset$, $1 = S$. We call any sub-ring of 2^S a **ring of sets**. All rings of sets are Boolean. The reciprocal is true according to the following Representation Theorem.

Theorem 2.1 *Let R be a Boolean ring. There is a set S and a sub-ring $\mathcal{R} \subseteq 2^S$ such that $R \simeq \mathcal{R}$.*

Proof: Immediate consequences of the axioms are: R is commutative, has characteristic 2 and 1 is the only one unit (invertible element) in R. Let $\mathfrak{m}$ be some maximal ideal of R. $R/\mathfrak{m}$ is a field of characteristic 2 which models $\forall x \; x^2 = x$, and the unique such field is $\mathbb{F}_2$. Let S be the set of all maximal ideals of R. Consider the canonical homomorphism of rings:

$$\Phi : R \longrightarrow \prod_{\mathfrak{m} \in S} R/\mathfrak{m} \simeq \prod_{S} \mathbb{F}_2.$$

The kern of Φ is the Jacobson radical $J(R)$ and consists of all $a \in R$ such that $1 + a$ is a unit. But 1 is the only one unit so Φ is an embedding of R into 2^S. □

The partial order $\subseteq$ does not depend of representation because it is definable by $a \subseteq b$ iff $ab = a$. One gets back the set-theoretic operations by $a \cap b := ab$, $a \cup b := a + b + ab$, $\bar{a} := 1 + a$. This shows that the abstract Boolean algebras, the abstract Boolean rings, the rings of sets and the Boolean algebras of sets are the same objects. If we add de Morgan's Rule $a \cup b = \overline{(\bar{a} \cap \bar{b})}$, we conclude that: $\{+, \cdot\}$ - circuits, $\{\cup, \cap, ^-\}$ - circuits and $\{\cap, ^-\}$ - circuits are equivalent and can be effectively translated one in another within linear consume of time and space. Moreover, it is sufficient for computing to dispose of tests $a = 0$, because $a = 1$ iff $\bar{a} = 0$ and $a = b$ iff $\bar{a} \cap b = 0 \;\wedge\; a \cap \bar{b} = 0$.

Definition: Let $\mathcal{R} \subseteq 2^S$ be a represented Boolean algebra and $Y \subset S$ be a proper subset. We define the restricted Boolean algebra:

$$\mathcal{R} \,|\, Y := \{ r \cap Y \,|\, r \in \mathcal{R} \} \subseteq 2^Y.$$

$\mathcal{R} \,|\, Y$ is never a proper sub-algebra of $\mathcal{R}$ because it has $1 = Y$, but is a principal ideal of $\mathcal{R}$ if $Y \in \mathcal{R}$. In the last case the isomorphism class of $\mathcal{R} \,|\, Y$ does not depend of the representation.

Definition: $x \in \mathcal{R}$ is called an **atom** of $\mathcal{R}$ if $\mathcal{R} \,|\, x$ is isomorphic with the two-element Boolean algebra $\mathbb{F}_2$. A Boolean algebra is called **atomic** if every element contains an atom. 2^S is an atomic algebra.

Finite Boolean algebras are always atomic and are isomorphic with the power-sets of their atoms. If F_m is the finite Boolean algebra with m atoms then F_m has 2^m elements. We can embed F_m into F_{m+1} in many ways, one possible embedding is defined by the following action on atoms: $\{i\} \rightsquigarrow \{i\}$ for $i = 1, \dots, m-1$ and $\{m\} \rightsquigarrow \{m, m+1\}$.

Let $\mathcal{F}_m$ be the Boolean algebra freely generated by m propositional variables $X_1, \dots, X_m$. For $\varepsilon \in \{0, 1\}$ we denote by $X^\varepsilon := X + \varepsilon$, such that $X^0 = X$ and $X^1 = \bar{X}$. With the m independent variables we can construct 2^m atoms:

$$X_1^{\varepsilon_1} \cap \dots \cap X_m^{\varepsilon_m} \neq 0.$$

It follows that $\mathcal{F}_m$ has 2^{2^m} elements and that $\mathcal{F}_m \simeq F_{2^m}$. To find an isomorphism take for all $i = 1, \dots, 2^m$ and $k = 1, \dots, m$: $i \in X_k \Leftrightarrow 2^{k-1}$ occurs in the shortest sum of two-powers representing i.

Definition: Let $\mathcal{R}$ be a Boolean algebra. The problem Dependent $\mathfrak{D}(\mathcal{R})$ is the set of all finite strings $x_1, \dots, x_n \in \mathcal{R}$ such that there exist $\varepsilon_1, \dots, \varepsilon_n \in \{0, 1\}$ with:

$$x_1^{\varepsilon_1} \cap \dots \cap x_n^{\varepsilon_n} = 0.$$

More generally, for a ring R we define the problem Zero-divisor $\mathfrak{Z}(R)$ consisting of all finite strings $x_1, \dots, x_n \in R$ such that there exist $\varepsilon_1, \dots, \varepsilon_n \in \{0, 1\}$ with:

$$(x_1 + \varepsilon_1) \dots (x_n + \varepsilon_n) = 0.$$

If R is a Boolean ring then $\mathfrak{Z}(R) = \mathfrak{D}(R)$. Both problems are always NBP over the given structures.

3 Main result

Theorem 3.1 *There does not exist any infinite Boolean ring (algebra) R such that a deterministic R-machine using arbitrary fixed constants $c_1, \dots, c_k \in R$ could decide $\mathfrak{D}(R)$ in polynomial time. Consequently, infinite Boolean rings satisfy $P \neq NBP$ and $P \neq NP$.*

The proof is divided in three Lemmas. The first Lemma eliminates the constants like in [10], the second Lemma reduces the problem to the existence of some uniform algorithm for all finite Boolean algebras and the third Lemma shows that such an algorithm cannot exist. The last two Lemmas are practically taken out from Poizat's proof over atomless algebras. An earlier version of the proof used the fact that every infinite Boolean algebra has a saturated extension that contains an atomless Boolean algebra as a sub-algebra. This has been finally eliminated as superfluous.

Lemma 3.2 *Let $\mathcal{R}$ be an infinite Boolean algebra. If there are constants $c_1, \dots, c_k$ in $\mathcal{R}$ and some $\mathcal{R}$-machine $\mathcal{M}$ using these constants able to decide the problem $\mathfrak{D}(\mathcal{R})$ in uniform polynomial time $p(n)$, then there exists an infinite Boolean algebra $\mathcal{R}_1$ and a parameter-free deterministic $\mathcal{R}_1$-machine $\mathcal{M}_1$ able to decide the problem $\mathfrak{D}(\mathcal{R}_1)$ in uniform polynomial time $2^{k+1}p(n) + n + 1$. (Recall that k is a constant.)*

Proof: We represent $\mathcal{R}$ as a sub-algebra of 2^S for some infinite set S. Let C be the finite sub-algebra of $\mathcal{R}$ generated by the constants $c_1, \dots, c_k$. Let $\alpha_1, \dots, \alpha_l$ be the atoms of C, $l \leq 2^k$. One sees that $\alpha_1 \cup \dots \cup \alpha_k = S$ and this is a partition. There must be an α_i, say α_1, such that the restricted algebra $\mathcal{R}_1 := \mathcal{R} \,|\, \alpha_1$ with $0 = \emptyset$ and $1 = \alpha_1$ is infinite.

Let $x_1, \dots, x_n \in \mathcal{R}_1$ be an input from $\mathcal{R}_1$. We observe:

$$\vec{x} \in \mathfrak{D}(\mathcal{R}_1) \Leftrightarrow \vec{x} \in \mathfrak{D}(\mathcal{R}) \,\vee\, \mathcal{R} \models \; \bar{x}_1 \cap \dots \cap \bar{x}_n \cap \alpha_1 = 0.$$

Indeed, if $x_1^{\varepsilon_1} \cap \dots \cap x_n^{\varepsilon_n} = 0$ calculated in $\mathcal{R}_1$ and at least one $\varepsilon_i = 0$ then $x_1^{\varepsilon_1} \cap \dots \cap x_n^{\varepsilon_n}$ calculated in $\mathcal{R}$ is a subset of $x_i \subseteq \alpha_1$ so $x_1^{\varepsilon_1} \cap \dots \cap x_n^{\varepsilon_n} = 0$ in $\mathcal{R}$. If all $\varepsilon_i = 1$ then $\bar{x}_1 \cap \dots \cap \bar{x}_n = \alpha_2 \cup \dots \cup \alpha_l$ in $\mathcal{R}$ but in this case $\bar{x}_1 \cap \dots \cap \bar{x}_n \cap \alpha_1 = 0$ in $\mathcal{R}$.

Step 1: we construct a $\mathcal{R}$-machine $\mathcal{M}'$ with constants $c_1, \dots, c_k$ working over $\mathcal{R}$ and deciding $\mathfrak{D}(\mathcal{R}_1)$ in polynomial time $p(n) + n + 1$. $\mathcal{M}'$ has the program: `Input` $x_1, \dots, x_n \in \mathcal{R}_1$`; if` $\mathcal{M}(x_1, \dots, x_n) =$ `yes then output yes and stop else if` $\bar{x}_1 \cap \dots \cap \bar{x}_n \cap \alpha_1 = 0$ `then output yes and stop else output no and stop`.

Step 2: we modify $\mathcal{M}'$ in order to get a parameter-free $\mathcal{R}_1$-machine $\mathcal{M}_1$ deciding $\mathfrak{D}(\mathcal{R}_1)$ in time $2^{k+1}p(n) + n + 1$. Let $C_1 := C \,|\, \alpha_2 \cup \dots \cup \alpha_l$. Every element occurring during a calculation of $\mathcal{M}'$ with inputs in $\mathcal{R}_1$ has the form $x = x' \cup x''$ where $x' \in \mathcal{R}_1$ and $x'' \in C_1$. We see that $\mathcal{M}'$ cannot quit the algebra $\mathcal{R}_1 \times C_1$. In the program of $\mathcal{M}'$ we replace the instruction `input` $x_1, \dots, x_n$ with $x'_1 := x_1; \dots; x'_n := x_n$ $x''_1 := 0; \dots; x''_n := 0$. We replace $x := y \cap z$ by $x' := y' \cap z'$; $x'' := y'' \cap z''$

and $x := \bar{z}$ by $x' := \bar{z}'$; $x'' := \bar{z}''$. We replace $x := x \cap c_i$ by $x' := 0$; $x'' := x'' \cap c_i$ if $\alpha_1 c_i = 0$ and by $x'' := x'' \cap (c_i \setminus \alpha_1)$ if $\alpha_1 c_i = \alpha_1$. Finally, replace `if` $x = 0$ `then ... else ...` by `if` $x' = 0$ `and` $x'' = 0$ `then ... else ....`

We observe now that all the computations in registers x' don't use any constants. The computations in registers x'' take place in a finite algebra C_1 with $\leq 2^k - 1$ atoms. All such possible computations can be now coded in some fixed data-basis of length $\leq 2^{2^{k+1}}$ in the program of $\mathcal{M}_1$. They are binary searched in logarithmic time according to the length of this data-basis by any new simulation done by $\mathcal{M}_1$ for a $\mathcal{M}'$-computation over C_1. $\mathcal{M}_1$ is parameter-free and deterministic.□

Lemma 3.3 *If for a parameter-free machine $\mathcal{M}_1$ there is an infinite Boolean algebra $\mathcal{R}_1$ such that $\mathcal{M}_1$ deterministically decides $\mathfrak{D}(\mathcal{R}_1)$ in some polynomial time $q(n)$, then $\mathcal{M}_1$ decides $\mathfrak{D}(F_m)$ in the same time $q(n)$ uniformly in m for all finite Boolean algebras F_m.*

Proof: $\mathcal{R}_1$ contains arbitrarily big finite sub-algebras, and because one can embed $F_m \hookrightarrow F_{m+1}$, contains $\mathcal{R}_1$ every finite Boolean algebra as a sub-algebra. $\mathcal{M}_1$ is parameter-free, so for inputs from the isomorphic image of F_m remain all computations in this sub-algebra F_m. □

Lemma 3.4 *There is no parameter-free machine able to decide the problems $\mathfrak{D}(F_m)$ in a uniform polynomial time $q(n)$ over all finite Boolean algebras F_m.*

Proof: Choose an m big enough such that $2^m > q(m)$ and consider inputs from the algebra $\mathcal{F}_m \simeq F_{2^m}$. If $X_1, \dots, X_m$ form a minimal set of free generators of $\mathcal{F}_m$ then the input $\vec{X} = (X_1, \dots, X_m) \notin \mathfrak{D}(\mathcal{F}_m)$. Along its computation path at most $q(m)$ atoms of the form $X_1^{\varepsilon_1} \cap \dots \cap X_m^{\varepsilon_m}$ have been tested if they are $= 0$ and all answers were negative. We can represent $X_1, \dots, X_m$ as subsets of a set of 2^m points. We color in red the points corresponding to the tested formal atoms and in black all other points. Because $2^m - q(m) > 0$ there is at least one black point.

If we remove a black point $i \in \{1, \dots, 2^m\}$ and consider the sets $Y_k := X_k \setminus \{i\}$ as subsets of the $2^m - 1$ set $\{1, \dots, 2^m\} \setminus \{i\}$, then is $\vec{Y} = (Y_1, \dots, Y_m) \in \mathfrak{D}(F_{2^m - 1})$ a positive input that will follow the same computation path like the negative input $\vec{X}$. This is a contradiction. □

The Theorem is now proved. We remark that all considerations work also for the non-uniform polynomial class $\mathbb{P}$.

On the other hand we emphasize that the general P - NP question in the sense of [8] for a given finite Boole algebra is equivalent with the classical P - NP Turing problem, so is sensibly more difficult than questions considered in this paper.

References

[1] **J.L. Bell, A.B. Slomson**: *Models and Ultraproducts.* North-Holland Publishing Co., 1969.

[2] **Lenore Blum, Felipe Cucker, Michael Shub, Steven Smale**: *Complexity and Real Computation.* Springer-Verlag, New-York, 1998.

[3] **Lenore Blum, Michael Shub, Steven Smale**: *On a theory of computation and complexity over the real numbers.* Bulletin A.M.S. 21, 1989.

[4] **Felipe Cucker, M. Matamala**: *On digital nondeterminism.* Mathematic Systems Theory 29, 1996, 635 - 647.

[5] **Yuri Ershov**: *Theorie der Numerierungen III.* Zeitschrift für die Mathematische Logik und Grundlagen der Mathematik, 23, 4, 1977, 289 - 371.

[6] **John B. Goode**: *Accessible telephone directories.* Journal of Symbolic Logic, 39, 1, 1994, 92 - 105.

[7] **Hans Hermes**: *Einführung in die Verbandstheorie.* Springer-Verlag, 1955.

[8] **Bruno Poizat**: *Les petits cailloux.* Aleas, Lyon 1995.

[9] **Bruno Poizat**: *A course in Model Theory.* Springer-Verlag, 2000.

[10] **Mihai Prunescu**: *A model-theoretic proof of $P \neq NP$ for all infinite Abelian groups.* Journal of Symbolic Logic, 2002.