On axiom of Archimedes

Andrzej Salwicki
Dombrova Research, Salwicki at mimuw dot edu dot pl
Partyzantów 19, Łomianki, PL 05-092 POLAND

Abstract. We prove that the following statement (known as axiom of Archimedes) is a theorem of algorithmic theory of natural numbers $\mathcal{ATN}$.

For every two positive natural numbers $0 < x < y$,
there exists a natural number n such, that the inequality $n \cdot x > y$ holds.

We are offering 1°) an algorithmic formula (Archi) that states: *the structure* $\mathfrak{N}$ *of natural numbers is archimedean*, 2°) the proof the this formula.
Therefore, the structure $\mathfrak{N}$ enjoys the *archimedean property*.

1. Introduction

Everybody believes that the data type of natural numbers (*aka* unsigned integers) enjoys the *Archimedean property.*. But, where is a proof? And above all, how to express this property?

In many mathematical proofs, as well as in almost every proof in algorithmics, one can find algorithmic properties are interspersed with elementary properties. (Elementary properties are expressed as formulas of predicate calculus. Algorithmic properties utilize the richer language, in which algorithmic formulas occur.)
An analysis of correctness' property of the Euclid's algorithm may serve as an illustration of our remark. The proof of the halting property of Euclid's algorithm assumes the Archimedean property [1] of natural numbers.
We can repeat our remark in a larger context. Many ordered fields enjoy the Archimedean property. The belief that every ordered field has the Archimedean property was shared by the esearchers for many years. Till the paper of Levi-Civitta broght a counter-example.

Eudoksus of Knidos (IVcentury BC) formulated the *principle of exhaustion*. The principle allowed to calculate the fields of geometrical figures. In fact, the Edoxus principle was the origin of integral calculus.
This principle is quoted by Euclid in "The Elements". The method of calculating surface area was simplified and used by Archimedes to calculate the volume of solids. You can read more about the cientific achievements of these outstanding scientists, in the book by Marek Kordos, "*Lectures from the history of mathematics*"[Kor05].

David Hilbert remarked that the axiomatization of the structure of natural numbers proposed by Giuseppe Peano should be completed by adding an appropriate axiom.[2]
And here a paradox appears, for the axiom of Archimedes can not be written as an elementary formula – this observation was made by Carol Karp in [Kar64] [1964]. This is a consequence of Goedel incompleteness theorem.
On the other hand, in many proofs the authors do use the assumption that natural numbers enjoy the archimedean

[1] But, who have seen the axiom and a verification of archimedean property of the structure natural numbers?
[2] The name '*axiom of Archimedes*' was coined by Otto Stolz around 1880.

property.

Erwin Engeler [Eng67] have shown that termination property of program is expressible as an infinite disjunction of quantifier-free formulas. He also proved an important theorem on termination property of algorithms running in the field of real numbers.
Let P denote the conjunction of the ordered field axioms and Archimedes' axiom[3]. Engeler showed that an algorithmic formula φ is true in the ordered field $\mathfrak{RO}$ of real numbers if and only if the implication $P \implies \varphi$ is a tautology.

$$\mathfrak{RO} \models \varphi \qquad \text{iff} \qquad \vdash P \implies \varphi$$

The Engeler's theorem have inspired us to create a calculus of programs (known as algorithmic logic $\mathcal{AL}$) [MS87]. The language of $\mathcal{AL}$ admits algorithms (i.e. programs) besides the terms and formulas. The set of formulas admits algorithmic formulas e.g. $K\alpha$ where a formula α is preceded by an algorithm K. If a formula of this form is satisfied then we say: *formula $K\alpha$ is the weakest precondion of the formula α with respect to the algorithm K*. Hence, the termination property of a program and the correctness of of a program can be stated as formulas and proved or falsified by means of the calculus of programs.
The paper *Programmability in fields* [Kre77] by Antoni Kreczmar contains many interesting results on algorithmic properties of programs executed in fields (mostly in achimedean fields).
The present note contributes to the paper of Kreczmar and proves that a formula that expresses the archimedean property is a theorem of algorithmic theory of numbers $\mathcal{ATN}$.. Hence, we show, that the structure of natural numbers enjoys the archimedean property, although it is not a field.

Below, we give a short and incomplete list of theorems and algorithms that use the axiom of Archimedes in the proofs.

Theorem on termination of Euclid's algorithm. Fundamental theorem of arithmetic. Chinese theorem on remainders and congruences. Theorem on bisection algorithm. Termination of binary power algorithm.

2. Proof of axiom of Archimedes

We need to formulate the property of beeing Archimedean before we prove it.
We recall that the property known as axiom of Archimedes can not be expressed by any first-order formula.

2.1. The formula

Carol Karp [Kar64] and Raphael M. Robinson [TMR65] noted that the Archimedean property (i.e. Archimedes' axiom) is not expressed by any first-order formula (or by a set of such formulas). C. Karp wrote this property as an infinite alternative of quantifier-free formulas.
Notice, that the following algorithmic statement (Archi) is valid in an algebraic structure $\mathfrak{A}$ iff *for every pair of non-zero elements x, y exists a natural number n such that the following inequality holds* $\underbrace{x + x + \cdots + x}_{n\times} > y$.

$$\forall_{x,y} 0 < x < y \implies \left\{ \begin{array}{l} a := x; \\ \textbf{while } a \leq y \textbf{ do} \\ \quad a := a + x \\ \textbf{od} \end{array} \right\} (a > y) \qquad \text{(Archi)}$$

is valid in a given algebraic structure $\mathfrak{A}$ iff *for every pair of non-zero elements x and y exists a natural number n such that the following inequality* $\underbrace{x + x + \cdots + x}_{n\times} > y$ *holds in* $\mathfrak{A}$.
Note, the number n does not appear in the formula above.

[3] Archimedes' axiom can be expressed as an infinite disjunction.

2.2. The proof

Consider the case when $x = 1$ and $y > x$.
We begin with the algorithmic axiom (R) of natural numbers that states that every natural number is reachable.

$$\{x := 1;\ \mathbf{while}\ x \neq y\ \mathbf{do}\ x := x + 1\ \mathbf{od}\}\,(x = y) \tag{R}$$

and obtain

$$\{a := 1;\ \mathbf{while}\ a \neq y\ \mathbf{do}\ a := a + 1\ \mathbf{od};\ a := a + 1\}\,(a > y) \tag{1}$$

In the case when $x \geq 1$ we shall consider the following program { N}

$$\left\{\begin{array}{l} u := 0; a := 1; \\ n := 0; \\ \mathbf{while}\ a \leq y\ \mathbf{do} \\ \quad a := a + 1;\ u := u + 1; \\ \quad \mathbf{if}\ u = x\ \mathbf{then}\ n := n + 1;\ u := 0\ \mathbf{fi} \\ \mathbf{od}; \\ n := n + 1 \end{array}\right\} \tag{N}$$

1. The program { N} terminates, for the following implication (2) is a theorem of calculus of programs

$$\vdash \left\{\begin{array}{l} x := 1; \\ \mathbf{while}\ x \neq y\ \mathbf{do} \\ \quad x := x + 1 \\ \mathbf{od} \end{array}\right\} \big(x = y\big) \implies \left\{\begin{array}{l} u := 0; a := 1; \\ n := 0; \\ \mathbf{while}\ a \neq y\ \mathbf{do} \\ \quad a := a + 1;\ u := u + 1; \\ \quad \mathbf{if}\ u = x\ \mathbf{then} \\ \quad\quad n := n + 1;\ u := 0 \\ \quad \mathbf{fi} \\ \mathbf{od}; \\ n := n + 1 \end{array}\right\} \big(a = y\big) \tag{2}$$

2. The final value of the variable n is the desired natural number. Because, the final valuation of variables does satisfy the inequality

$$\{N\}(n * x > y \tag{3}$$

□

2.3. If the computation of the program N can be prolongued at will

I.e. if the computation of the program N is infinite, then the element y is not a natural number.
This remark shows that the reachability axiom of natural numbers Ax_2 and axiom of Archimedes are mutually replaceable. For the convenience of the reader the subsection 4.1 contains a set of axioms of $\mathcal{ATN}$ theory. The reachability axiom Ax_2 can be replaced by the Archimedes' axiom $Archi$. Both theories are algorithmically equivalent. In particular, both theories enjoy the cathegoricity property.

3. Final remarks

It may happen that the approach presented in this paper will be of use in the coming years when the artificial inteligence will read papers and verify the proofs of theorems on software correction.

4. Supplements

4.1. Algorithmic theory of natural numbers $\mathcal{ATN}$

The theory $\mathcal{ATN}$ is a system of three objects $\langle \mathcal{L}, \mathcal{C}, \mathcal{A} \rangle$, where $\mathcal{L}$ is the language of the theory, $\mathcal{C}$ is the operation of logical consequence , $\mathcal{A}$ is the set of axioms specific for the theory $\mathcal{ATN}$.

$\mathcal{L}$ the language of the theory is the pair $\langle A, WFF \rangle$ where $A$is an alphabet and WFF - is the set of well formed expressions.
The alphabet contains a set of variables, two nostants 0 and 1, functors $s, +, *$, predicates $=, <$. The set $WFF = T \cup F \cup P$ is the union of threesets: Tset of terms, P set of programs and F set of formulas. Note that this set is larger that the set of first-order formulas for it contains also algorithmic formulas. For details consult [MS87].

$\mathcal{C}$ is the operation of logical consequence. Let X denote a set of formulas, the set of all logical consequences of this set is denoted by C(X). In particular, the set $C(Ax)$ is the set of all theorems of the algorithmic theory of numbers $\mathcal{ATN}$, provided that $Ax = \{Ax_0, \ldots, Ax_5\}$ is the set of axioms listed below. Consult [MS87].

$\mathcal{A}$ the teory has the following axioms

$Ax_0)$ $\quad \forall_n \, s(n) \neq 0$

$Ax_1)$ $\quad \forall_{n,m} \, s(n) = s(m) \implies n = m$

$Ax_2)$ $\quad \forall_n \, \{x := 1; \, \mathbf{while} \, x \neq n \, \mathbf{do} \, x := s(x) \, \mathbf{od}\} \, (x = n)$

$$Ax_3) \quad \forall_{x,y,z} \, x + y = z \overset{df}{\Leftrightarrow} \left\{ \begin{array}{l} w := x; \, u := 0; \\ \mathbf{while} \, u \neq y \, \mathbf{do} \\ \quad w := s(w); \, u := s(u) \\ \mathbf{od} \end{array} \right\} (w = z)$$

$$Ax_4) \quad \forall_{x,y} \, x < y \overset{df}{\Leftrightarrow} \left\{ \begin{array}{l} u := 0; \, z := 0; \\ \mathbf{while} \, u \neq y \wedge z \neq x \, \mathbf{do} \\ \quad z := s(z); \, u := s(u) \\ \mathbf{od} \end{array} \right\} (x = z \wedge x \neq y)$$

$$Ax_5) \quad \forall_{x,y,z} \, x * y = z \overset{df}{\Leftrightarrow} \left\{ \begin{array}{l} w := 0; \, u := 0; \\ \mathbf{while} \, u \neq y \, \mathbf{do} \\ \quad w := w + x; \, u := s(u) \\ \mathbf{od} \end{array} \right\} (w = z)$$

4.2. A more detailed proof of formula (Archi)

The proof is easy. Note, it can be verified by an appropriated *proof-checker* (to be written!).

We begin with a tautology.

$$\vdash \left\{ \begin{array}{l} a := 1; \\ \mathbf{while} \, a \neq y \, \mathbf{do} \\ \quad a := a + 1 \\ \mathbf{od} \end{array} \right\} (a = y) \implies \left\{ \begin{array}{l} a := 1; \\ \mathbf{while} \, a \neq y \, \mathbf{do} \\ \quad a := a + 1 \\ \mathbf{od} \end{array} \right\} (a = y) \tag{4}$$

insertion of assignment command $u := u + 1$ does not falsify the stop property

$$\vdash \left\{ \begin{array}{l} a := 1; \\ \textbf{while } a \neq y \textbf{ do} \\ \quad a := a + 1 \\ \textbf{od} \end{array} \right\} (a = y) \implies \left\{ \begin{array}{l} a := 1; \\ \textbf{while } a \neq y \textbf{ do} \\ \quad a := a + 1;\ u := u + 1; \\ \textbf{od} \end{array} \right\} (a = y) \tag{5}$$

we insert also the conditional instruction **if...**

$$\vdash \left\{ \begin{array}{l} a := 1; \\ \textbf{while } a \neq y \textbf{ do} \\ \quad a := a + 1 \\ \textbf{od} \end{array} \right\} (a = y) \implies \left\{ \begin{array}{l} a := 1; \\ \textbf{while } a \neq y \textbf{ do} \\ \quad a := a + 1;\ u := u + 1; \\ \quad \textbf{if } u = x \textbf{ then } n := n + 1;\ u := 0 \textbf{ fi} \\ \textbf{od} \end{array} \right\} (a = y) \tag{6}$$

the program in the succedent of implication may be preceded by a linear program $u := o;\ n := 0;$, this makes no harm to the halting prperty of the program

$$\vdash \left\{ \begin{array}{l} a := 1; \\ \textbf{while } a \neq y \textbf{ do} \\ \quad a := a + 1 \\ \textbf{od} \end{array} \right\} (a = y) \implies \left\{ \begin{array}{l} u := 0;\ n := 0;\ a := 1; \\ \textbf{while } a \neq y \textbf{ do} \\ \quad a := a + 1;\ u := u + 1; \\ \quad \textbf{if } u = x \textbf{ then } n := n + 1;\ u := 0 \textbf{ fi} \\ \textbf{od} \end{array} \right\} (a = y) \tag{7}$$

jfinally we add an assigbment $n := n + 1$ after the while instruction

$$\mathcal{ATN} \vdash \left\{ \begin{array}{l} a := 1; \\ \textbf{while } a \neq y \textbf{ do} \\ \quad a := a + 1 \\ \textbf{od} \end{array} \right\} (a = y) \implies \left\{ \begin{array}{l} u := 0;\ n := 0;\ a := 1; \\ \textbf{while } a \neq y \textbf{ do} \\ \quad a := a + 1;\ u := u + 1; \\ \quad \textbf{if } u = x \textbf{ then} \\ \quad\quad n := n + 1;\ u := 0 \\ \quad \textbf{fi} \\ \textbf{od}; \\ n := n + 1 \end{array} \right\} \big(\underbrace{(x + x + \cdots + x)}_{n \text{ razy}} > y \wedge a = y \big) \tag{8}$$

we apply modus ponens rule to obtain our formula (3)

$$\mathcal{ATN} \vdash \left\{ \begin{array}{l} u := 0;\ n := 0;\ a := 1; \\ \textbf{while } a \neq y \textbf{ do} \\ \quad a := a + 1;\ u := u + 1; \\ \quad \textbf{if } u = x \textbf{ then } n := n + 1;\ u := 0 \textbf{ fi} \\ \textbf{od}; \\ n := n + 1 \end{array} \right\} \big(\underbrace{(x + x + \cdots + x)}_{n \times} > y \big) \tag{9}$$

□

5. References

[Eng67] Erwin Engeler. Algorithmic Properties of Structures. Mathematical Systems Theory, 1:183–195, 1967.

[Kar64] Carol Karp. Languages with Expressions of Infinite Length. North Holland, 1964.

[Kor05] Marek Kordos. Wykłady z historii matematyki. Script, Warszawa, 2005.

[Kre77] Antoni Kreczmar. Programmability in Fields. Fundamenta Informaticae, I(2):195–230, 1977. [http://lem12.uksw.edu.pl/images/d/d8/Kreczmar-Program-Fields.pdf].

[MS87] Grażyna Mirkowska and Andrzej Salwicki. Algorithmic Logic. PWN and J.Reidel, Warszawa, 1987. [http://lem12.uksw.edu.pl/wiki/Algorithmic_Logic]. [Online; accessed 7-August-2017].

[TMR65] Alfred Tarski, Andrzej Mostowski, and Raphael M. Robinson. Undecidable Theories. North Holland, 1965.