UPPSALA
UNIVERSITET

$\mathcal{P}$ is not equal to $\mathcal{NP}$*

Sten-Åke Tärnlund

Division of Information Systems
Research Report 2013:5

Uppsala University
Department of Informatics and media
P.O. Box 513
SE-751 20 Uppsala, Sweden

ISSN 1403-7572

$\mathcal{P}$ is not equal to $\mathcal{NP}$*

Sten-Åke Tärnlund[†,‡]

November 4, 2013

Abstract

The problem of computing whether any formula of propositional logic is satisfiable is not in $\mathcal{P}$. Therefore, $\mathcal{P}$ is not equal to $\mathcal{NP}$. The proofs are informal about formal proofs in a first-order theory **B** axiomatizing Turing's theory of computing. However, the informal proofs can be converted into formal proofs in Hilbert's proof theory, and proved using a theorem prover.

1 Introduction

Let SAT be the set of satisfiable formulas of propositional logic. The satisfiability problem for whether $p \in SAT$, all propositional formulas p. Let $\mathcal{P}$ be the set of problems with a solution of a deterministic Turing machine in polynomial computing time. Let $\mathcal{NP}$ be the set of problems with a solution of a nondeterministic Turing machine in polynomial computing time. $SAT \in \mathcal{P}$ for the satisfiability problem is in $\mathcal{P}$.

Theorem 1: $SAT \notin \mathcal{P}$, gives a proof of Theorem 2: $\mathcal{P} \neq \mathcal{NP}$, in addition, to the ones given by Tärnlund [10, 11, 12, 13].

Axiom B, cf. Tärnlund [10],[1] characterizes a universal Turing machine, thus, defines computing. It connects computing with foundations of mathematics e.g., proof theory. As a consequence, textbook methods, cf. Kleene [5], give Lemma 1: if $SAT \in \mathcal{P}$ then each sufficiently large tautology F (on DNF) has a formal propositional proof of polynomial size (in the size of F) in Robinson [7] resolution systems. Then, Theorem 1 follows from Lemma 1, and Corollary 3: it is not the case that each sufficiently large tautology F (on DNF) has a proof of polynomial size (in the size of F) in Robinson resolution systems. The latter follows from Haken's [2] theorem.

The proofs are informal about formal proofs in a first-order theory **B** axiomatizing Turing's [14] theory of computing. However, they can be converted into formal proofs in Hilbert's [4] proof theory, cf. Kleene [5], and proved using a theorem prover, in the manner of the proofs in Tärnlund [12].

Conceptually, Axiom B is a logic program, cf. Kowalski [6]. Formal proving from axiom B is computing, any computation can be proved from B (Turing's thesis). The size of a computation (proof) is the number of symbols in the proof.

---

*Fourth edition. The idea of the proofs are similar to those in the former editions, but simplified. Corollary 3 has been restored, similar to Corollary 8 of Tärnlund [11].

‡ gmail: stenake.

[1] Axiom B is defined in Axiom 1, a slight simplification of the Horn clause in Tärnlund [9].

Number theory, cf. Kleene [5], and a theory of data and programs, cf. Clark and Tärnlund [1], are used to prove properties of computations.

2 Axiom B a universal Turing machine

Theory **B** has an Axiom B. The U predicate of B defines a predicate $T(i, a, u)$, i.e., Turing machine i for input a computes output u. U also characterizes a universal Turing machine that defines computing. Axiom B comprises five parts: a definition of $T(i, a, u)$ (1)–(2), a halt statement (3), a left or right move of the head of Turing machine i (4)–(5), and a search for a quintuple of i (6). (For more explanations of theory **B** cf. Tärnlund [12].)

Axiom 1 *B for*

$$T(i,\, a,\, u) \supset U(\emptyset,\, \emptyset,\, a\,.\,\emptyset,\, 1,\, i,\, i,\, u) \quad i \in M\ a \in R\ u \in L. \tag{1}$$

$$U(\emptyset,\, \emptyset,\, a\,.\,\emptyset,\, 1,\, i,\, i,\, u) \supset T(i,\, a,\, u) \quad i \in M\ a \in R\ u \in L. \tag{2}$$

$$U(x,\, s,\, z,\, 0,\, i,\, i,\, x) \quad x \in L\ s \in S\ z \in R\ i \in M. \tag{3}$$

$$U(x,\, v,\, r\,.\,z,\, p,\, i,\, i,\, u) \supset U(x\,.\,v,\, s,\, z,\, q,\, q\,.\,s\,.\,p\,.\,r\,.\,0\,.\,j,\, i,\, u) \quad x\,u \in L \tag{4}$$
$$v\,r\,s \in S\ \ z \in R\ \ p\,q \in Q\ \ i\,j \in M.$$

$$U(x\,.\,r,\, v,\, z,\, p,\, i,\, i,\, u) \supset U(x,\, s,\, v\,.\,z,\, q,\, q\,.\,s\,.\,p\,.\,r\,.\,1\,.\,j,\, i,\, u) \quad x\,u \in L \tag{5}$$
$$v\,r\,s \in S\ \ z \in R\ \ p\,q \in Q\ \ i\,j \in M.$$

$$U(x,\, s,\, z,\, q,\, j,\, i,\, u) \supset U(x,\, s,\, z,\, q,\, q'\,.\,s'\,.\,p\,.\,r\,.\,d\,.\,j,\, i,\, u) \quad x\,u \in L \tag{6}$$
$$r\,s\,s' \in S\ \ z \in R\ \ p\,q\,q' \in Q\ \ d \in D\ \ i\,j \in M.$$

Here, $\emptyset$, 0 and 1 are constants, and $.$ an infix term for lists.

The free variables have the generality interpretation. The domains of B are: the set S of symbols, the set Q of states, the set D of head moves, the set R of right tapes, the set L of left tapes, and the set M of codes of Turing machines.

3 Complexity measures

Let the computing time be the number of moves of the head of a Turing machine in a computation, cf. Sipser [8], and Hartmanis and Stearns [3].

Then, the computing time in theory **B** is the number of moves of the head of a Turing machine in a formal deduction, cf. Tärnlund [10].

The notion: there exists a formal deduction of $\exists\, u\, T(i, a, u)$ from axiom B in a computing time that is less than or equal to z, is written as follows using a Kleene G4 system, $i \in M\ a \in R\ z \in Z^+$.[2]

Definition 1 $\vdash B \rightarrow \exists\, u\, T(i, a, u)$ *in z if and only if there exists a formal deduction of $\exists\, u\, T(i, a, u)$ from B in a computing time that is less than or equal to z in Kleene G4 systems all $i \in M\ \ a \in R\ \ z \in Z^+$.*

[2] Kleene's G4 system is a choice of convenience, other systems can be used, e.g., Robinson resolution systems, cf. Proposition 1, footnote 4.

Let U be the nonempty set of deterministic Turing machines that compute whether G is satisfiable, with output $\emptyset\,.\,0$, or not, with output $\emptyset\,.\,1$, for all propositions G.

$$\begin{gathered}\text{Assume that, there is a deterministic Turing machine with the name } i, \\ i \in U, \text{ that computes the output in computing time } c \cdot |G|^n \\ \text{some } c\ n \in Z^+ \text{ all propositions } G.\end{gathered} \quad (7)$$

If $SAT \in \mathcal{P}$ then the deterministic Turing machine i for input G computes output u if and only if the negation of G is not valid and $u = \emptyset\,.\,0$ (G is satisfiable), or the negation of G is valid and $u = \emptyset\,.\,1$ (G is unsatisfiable) for all propositions G. This input-output relationship of i, in Theory **B**, is formalized next, using Axiom B.

Definition 2 *If* $SAT \in \mathcal{P}$ *then* $T(i, G\,.\,\emptyset, u) \equiv (\not\models \neg\, G \wedge u = \emptyset\,.\,0) \vee (\models \neg\, G \wedge u = \emptyset\,.\,1)$ *all propositional formulas* G*.*

Corollary 1 *If* $SAT \in \mathcal{P}$ *then* $T(i, \neg\, F\,.\,\emptyset, \emptyset\,.\,1) \supset F$ *all tautologies* F*.*

If $SAT \in \mathcal{P}$ then i for input $\neg F$ has a proof (computation) of the output $\emptyset\,.\,1$ from axiom B, in polynomial computing time $c \cdot |F|^n$, in Kleene G4 systems for all tautologies F. This definition of $SAT \in \mathcal{P}$ is formalized next, using Definition 1.

Definition 3 *If* $SAT \in \mathcal{P}$ *then* $\vdash B \to T(i, \neg\, F\,.\,\emptyset, \emptyset\,.\,1)$ *in* $c \cdot |F|^n$ *some* $c\ n \in Z^+$ *all tautologies* F*.*

Let c and n be names of the positive integers in Definition 3.

If $SAT \in \mathcal{P}$ then there exists a proof of F from B, in polynomial computing time (in the size of F), in Kleene G4 systems for all tautologies F on disjunctive normal form (DNF), by Corollary 1 and Definition 3.

Corollary 2 *If* $SAT \in \mathcal{P}$ *then* $\vdash B \to F$ *in* $c \cdot |F|^n$ *all tautologies* F *on DNF.*

Let the size of a proof be the number of symbols in the proof.

Next, a polynomial upper bound is introduced on the size $|\vdash_R F|$ of $\vdash_R F$, i.e., on the size of a formal proof of F that exists in Robinson resolution systems for all sufficiently large tautologies F on DNF. It is formalized as follows.

Definition 4 $|\vdash_R F| \in O(|F|^m)$ *if and only if the size of a formal proof of* F *that exists, in Robinson resolution system, has a polynomial upper bound* $a \cdot |F|^m$ *some* $a \in Z^+$ *all* $m \in Z^+$ *all sufficiently large tautologies* F *on DNF.*

4 Lemma 1 and a proof

If $SAT \in \mathcal{P}$ then the size of a proof of F that exists in Robinson resolution systems has a polynomial upper bound (in the size of F) for all sufficiently large tautologies F on DNF. This is written more formally using Definition 4.

Lemma 1 *If* $SAT \in \mathcal{P}$ *then* $|\vdash_R F| \in O(|F|^{2 \cdot n+1})$ *all sufficiently large tautologies* F *on DNF.*

Proof.

$$\text{Assume that } SAT \in \mathcal{P}. \tag{8}$$

Then, there is a proof of F from B in computing time $c \cdot |F|^n$, all tautologies F on DNF in Kleene G4 systems, by Corollary 2. Therefore,

$$\vdash B \to F \text{ in } c \cdot |F|^n \text{ all tautologies } F \text{ on DNF.} \tag{9}$$

A proof tree of the predicate calculus sequent $B \to F$ is computable (breadth-first) in polynomial $c \cdot |F|^n$ computing time. For reasons of space, the propositional atoms $U_{j\,k}$ and T are used as short names for propositional instantiations of the U and T predicates of axiom B.[3] The names are enumerated by the indexes, where j is the computing time, and k is the searching time for a quintuple of the code, of the deterministic Turing machine i.

The proof tree is computable in polynomial computing time, i.e., in at most polynomially many head moves of i (in the size of F) along a branch. The search time for a quintuple is less than the size $|i|$ of i.

Proof tree 1 *A (condensed) proof of the sequent $B \to F$ in polynomial computing time z, $z \leq c \cdot |F|^n$, follows, $F \in TAUT$ on DNF.*[4]

$$\begin{array}{ccc}
B, U_{(z-1)\,k_z} \overset{\times}{\to} U_{(z-1)\,k_z} & & B, U_{z\,0} \overset{\times}{\to} U_{z\,0} \\
 & \ddots & \vdots \\
B, U_{0\,0} \overset{\times}{\to} U_{0\,0} & & B \to U_{0\,1} \\
 & \backslash & | \\
B, T \overset{\times}{\to} T & & B, U_{0\,1} \supset U_{0\,0} \to U_{0\,0} \\
 & \backslash & | \\
B, F \overset{\times}{\to} F & & B, U_{0\,0} \supset T \to T, \ \textit{by} \supset\to \ \forall\to \\
 & \backslash & | \\
 & & B,\ T(i,\, \neg F\,.\,\emptyset,\, \emptyset\,.\,1) \supset F \ \to F, \ \textit{Corollary 1} \\
 & & | \\
 & & B \to F
\end{array}$$

The proof tree is computable breadth-first, i.e., all choices of axiom B are explored with no backtracking on the logic program B. The entire proof tree is closed ($\times$) at computing time $z \leq c \cdot |F|^n$ by the proposition $U_{z\,0}$.

A formal propositional proof $\Delta(c \cdot |F|^n)$ of F is computable from $U_{z\,0}$, of the top leaf node $B, U_{z\,0} \overset{\times}{\to} U_{z\,0}$, downward in the tree. However, the proof path from $U_{z\,0}$ to F is not decidable until $U_{z\,0}$ is computed, in computing time $c \cdot |F|^n$.

[3] The sizes of the instantiations of the predicates U and T of axiom B are used for computing the size of a proof, not the sizes of the short names.

[4] Proof tree 1 is computable in computing time $c \cdot |F|^n$ from axiom B, using Robinson resolution systems, by induction on the computing time.

Generally, by induction on the computing time.

Proposition 1 *If Turing machine i for input a computes output u in polynomial computing time (in the size of a) then $B \vdash_R T(i,\, a,\, u)$ in polynomial computing time (in the size of a) all $i \in M$ $a \in R$ $u \in L$.*

In summary, $\Delta(c \cdot |F|^n)$ is the sequence of the blue propositions, instantiations of axiom B, and F of Corollary 1, it begins and ends as follows.

$$U_{z\,0},\, U_{z\,0} \supset U_{(z-1)\,k_z},\, U_{(z-1)\,k_z},\, U_{(z-1)\,k_z} \supset U_{(z-1)\,(k_z-1)}, \ldots,$$
$$U_{1\,0},\, U_{1\,0} \supset U_{0\,k_1}, \ldots, U_{0\,1} \supset U_{0\,0},\, U_{0\,0},$$
$$U_{0\,0} \supset T,\, T,\, T(t,\, F\,.\,\emptyset,\, \emptyset\,.\,0) \supset F,\, F. \tag{10}$$

More precisely, a formal propositional proof $\Delta(z)$ of F in computing time z is defined as follows, using (enumerations of) the propositional atoms T and U for instantiations of axiom B, cf. (10), $z \in Z^+$ $F \in TAUT$ on DNF.[5]

First, a proof $h(z)$ of $U_{(z-1)\,0}$ in computing time from z to $(z-1)$.

Definition 5 $h(z) = < U_{z\,0}, U_{z\,0} \supset U_{(z-1)\,k_z}, U_{(z-1)\,k_z}, U_{(z-1)\,(k_z-q)} \supset U_{(z-1)\,(k_z-q-1)}, U_{(z-1)\,(k_z-q-1)} >$ *for* $0 \leq q < k_z$ *all* $z \in Z^+$ *some* $k_z \in N$.

Second, a proof $\Delta(z)$ of F in computing time z, defined inductively.

Definition 6 $\Delta(z) =< h(z),\, \Delta(z-1) >$ *all* $z \in Z^+$.

$\Delta(0) =< U_{0\,0} \supset T, T, T(t, F\,.\,\emptyset, \emptyset\,.\,0) \supset F, F >$ *all tautologies* F *on DNF.*

If (9) then $\Delta(c \cdot |F|^n)$, by Definition 6, and induction on the computing time. Hence,

$$\Delta(c \cdot |F|^n) \text{ all tautologies } F \text{ on DNF.} \tag{11}$$

$\Delta(c \cdot |F|^n)$ is a formal propositional proof of F (on DNF) in Robinson resolution systems, by induction on the computing time. Thus,

if (11) then $\vdash_R F$ all tautologies F on DNF. Therefore,

$$\vdash_R F \text{ all tautologies } F \text{ on DNF.} \tag{12}$$

It is sufficient to compute an upper bound on the size $|\Delta(z)|$ of the proof $\Delta(z)$ of F for sufficiently large tautologies F, in Robinson resolution systems.

For any size of the nondeterministic Turing machine i, there are sufficiently large tautologies F such that $|i| < |F|$. The size of each atomic propositional formula of $\Delta(z)$ has a polynomial upper bound $c \cdot |F|^n$. Moreover, $z \leq c \cdot |F|^n$ and $k_r < |F|$ for $0 \leq r \leq z$. Therefore, by Definition 6,

$$|\Delta(c \cdot |F|^n)| \in O(|F|^{2 \cdot n+1}) \text{ all sufficiently large tautologies } F \text{ on DNF.} \tag{13}$$

There is a resolution proof of F of polynomial size (in the size of F) for all sufficiently large tautologies F, by (12)–(13) and Definition 4. Therefore,

$$|\vdash_R F| \in O(|F|^{2 \cdot n+1}) \text{ all sufficiently large tautologies } F \text{ on DNF.} \tag{14}$$

Discharging the assumption (8) ends the proof of Lemma 1.

$$\text{If } SAT \in \mathcal{P} \text{ then } |\vdash_R F| \in O(|F|^{2 \cdot n+1}) \text{ all sufficiently large tautologies } F \text{ on DNF.} \tag{15}$$

[5] $\Delta(c \cdot |F|^n)$ is computable in computing time $c \cdot |F|^n$ from axiom B, using Robinson resolution systems, by induction on the computing time, cf. Proposition 1 in footnote 4.

5 Theorem 1 and Theorem 2

Haken's theorem gives: it is not the case that each sufficiently large tautology F (on DNF) has a proof of polynomial size (in the size of F) in Robinson resolution systems. This sentence is formalized as follows.

Corollary 3 *$\neg(|\vdash_R F| \in O(|F|^m)$ some $m \in Z^+$ all sufficiently large $F \in TAUT$ on DNF).*

The sentence:SAT is not in $\mathcal{P}$, has a reductio ad absurdum proof from Lemma 1 and Corollary 3. Therefore,

Theorem 1 *$SAT \notin \mathcal{P}$.*

Thus, the Turing machine i in (7) does not exist. However, $SAT \in \mathcal{NP}$, cf. Sipser. Therefore,

Theorem 2 *$\mathcal{P} \neq \mathcal{NP}$.*

6 Conclusion

Theorem 1: $SAT \notin \mathcal{P}$, gives a proof of Theorem 2: $\mathcal{P} \neq \mathcal{NP}$. This is an alternative proof, in addition, to the ones given by Tärnlund [10, 11, 12, 13]. The informal proofs about formal proofs can be converted into formal proofs in Hilbert's proof theory, and proved using a theorem prover, in the manner of the proofs in Tärnlund [12]. The existence of such a proof of $\mathcal{P} \neq \mathcal{NP}$ is more trustworthy, of course.

Acknowledgment

Thanks to Keith Clark, Hanna-Nina Ekelund, Niklas Ekelund, Andreas Hamfelt, Torsten Palm, Bo Steinholtz, Carl-Anton Tärnlund, and the participants of The Stockholm-Uppsala Logic Seminar, 3 February 2010.

References

[1] Keith L. Clark and Sten-Åke Tärnlund. A first order theory of data and programs. In Bruce Gilchrist, editor, *Information Processing 77*, volume 7, pages 939–944, Amsterdam, The Netherlands, 1977. North-Holland.

[2] Armin Haken. The intractability of resolution (complexity). *Theoretical Computer Science*, 39:297–308, 1985. Ph D thesis University of Illinois at Urbana-Champaign 1984.

[3] J. Hartmanis and R. E. Stearns. On the computational complexity of algorithms. *Transactions of the American Mathematical Society*, 117:285–306, 1965.

[4] David Hilbert and Wilhelm Ackermann. *Grundzüge der theoretischen Logik*. Springer-Verlag, 1928. Republished 1972; English translation by Lewis Hammond et al., Principles of Mathematical Logic, Chelsea, New York, 1950.

[5] Stephen C. Kleene. *Mathematical Logic*. John Wiley and Sons, New York, USA, 1967. First corrected printing, March, 1968.

[6] Robert A. Kowalski. Predicate Logic as a Programming Language. In J.L. Rosenfeldt, editor, *Information Processing 74*, pages 569–574. Amsterdam, The Netherlands, 1974.

[7] John Alan Robinson. A Machine-Oriented Logic Based on the Resolution Principle. *Journal of the ACM*, 12(1):23–41, 1965.

[8] Michael Sipser. *Introduction to the Theory of Computation*. Thomson Course Technology, 2005. Second Edition International Edition.

[9] Sten-Åke Tärnlund. Horn clause computability. *BIT*, 17(2):215–226, 1977. TRITA-IBADB-1034, The Royal Institute of Technology 1975, Sweden.

[10] Sten-Åke Tärnlund. $\mathcal{P}$ is not equal to $\mathcal{NP}$. *arXiv e-prints*, October 2008.

[11] Sten-Åke Tärnlund. $\mathcal{P}$ is not equal to $\mathcal{NP}$. *arXiv e-prints*, July 2009. Second printing.

[12] Sten-Åke Tärnlund. Verifying that $\mathcal{P}$ is not equal to $\mathcal{NP}$ using a theorem prover. *DiVA e-prints*, December 2012.

[13] Sten-Åke Tärnlund. The tautology problem is not in $\mathcal{NP}$. *DiVA e-prints*, April 2013.

[14] Alan M. Turing. On Computable Numbers with an Application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, 2/46:230–265, 1936.