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

Sten-Åke Tärnlund†

May 9, 2017

Abstract

SAT is not in $\mathcal{P}$ is proved, in a first-order theory, with a new single finite axiom of Turing's theory of computing. So, $\mathcal{P}$ is not equal to $\mathcal{NP}$.

1 Introduction

It is proved that computing whether each propositional formula is satisfiable is not in $\mathcal{P}$, the set of problems having a solution in polynomial computing time (in the size of the input) using a deterministic Turing machine. Thus, Theorem 1 $SAT \notin \mathcal{P}$. But $SAT \in \mathcal{NP}$, the set of problems having a solution in polynomial computing time (in the size of the input) using a nondeterministic Turing machine. Therefore, Theorem 2 $\mathcal{P} \neq \mathcal{NP}$, cf. Cook [2].

Theorem 1 follows, essentially, from the new single finite Axiom 1 of Turing's theory of computing [15] by Lemma 1,[5] using Haken's theorem [3].[6]

The, human-oriented, proofs are informal in Hilbert's proof theory [5], about formal proofs in Robinson resolution [8],[1] [2] using number theory, and a consistent subset of Theory **B**, introduced next.

2 The sole finite axiom of Theory B

Axiom 1 characterizes a universal Turing machine in first-order logic.[3] Roughly, (1) if U with the initial two-way tape computes u for input a by Turing machine i, a list of quintuples, then i computes u for a; (2) U at state 0 computes output x and halts; (3) if U with the tape x, v, $p, r\,.\,z \rightsquigarrow r\,.\,z$, u (r is printed, the tape-head moves to the left, i.e., to v, p is a new state, reads the tape $r\,.\,z$ starting at $r\,.\,z$, i.e., $r\,.\,z \rightsquigarrow r\,.\,z$) computes u then U with the, previous, tape $x\,.\,v$, s, q, $z \rightsquigarrow q\,.\,s\,.\,p\,.\,r\,.\,0\,.\,j$, u (read s in state q, reads the tape z starting at $q\,.\,s\,.\,p\,.\,r\,.\,0\,.\,j$) computes u; (4) is similar to (3), but the head moves to the right; (5) if U with the tape x, s, q, $z \rightsquigarrow j$, u computes u then U with the tape x, s, $q, z \rightsquigarrow v'\,.\,j$, u computes u. The sets of symbols S, states Q, head-moves D, right-tapes R, left-tapes L, and Turing machines M are the intended domains, $i\,j \in M\ a\,z \in R\ u\,x \in L\ s\,v\,r\,s'\,v' \in S\ p\,q \in Q\ d \in D\ v' \in z$. Formally,

*Eighth edition, minor simplifications of the Seventh edition, cf. Tärnlund [14].

[1] The proof ideas are similar to earlier editions, Tärnlund [10, 11, 13, 14].

[2] In addition, there is verified a computer-oriented version, Tärnlund [12], using a theorem-prover Vampire of Riazanov and Voronkov [7].

[3] Axiom 1 is a simplification of Tärnlund [9], and a slight change of Tärnlund [14].

Axiom 1

$$U(\emptyset,\, \emptyset,\, 1,\, a\,.\,i \rightsquigarrow a\,.\,i,\, u) \supset T(i,\, a,\, u). \tag{1}$$

$$U(x,\, s,\, 0,\, z \rightsquigarrow z,\, x). \tag{2}$$

$$U(x,\, v,\, p, r\,.\,z \rightsquigarrow r\,.\,z,\, u) \supset U(x\,.\,v,\, s,\, q,\, z \rightsquigarrow q\,.\,s\,.\,p\,.\,r\,.\,0\,.\,j,\, u). \tag{3}$$

$$U(x\,.\,r,\, v,\, p,\, z \rightsquigarrow z,\, u) \supset U(x,\, s,\, q,\, v\,.\,z \rightsquigarrow q\,.\,s\,.\,p\,.\,r\,.\,1\,.\,j,\, u). \tag{4}$$

$$U(x,\, s,\, q,\, z \rightsquigarrow j,\, u) \supset U(x,\, s,\, q,\, z \rightsquigarrow v'\,.\,j,\, u). \tag{5}$$

Here, $\emptyset$, 0 *and* 1 *are constants. The infix binary terms* . *and* $\rightsquigarrow$, *for lists, are identity maps. The free variables have the generality interpretation.*

3 Complexity of Computing

For short, writing B for Axiom 1. (6)

Definition 1 *Let* $T(i, a, u)$ *in* z *be Turing machine* i *computes* u *for input* a *within* z *head-moves (state-symbol computing cycles) i.e., computing time Hartmanis and Stearns [4]. Let* $\vdash B \rightarrow T(i, a, u)$ *in* z *be there exists a proof of* $T(i, a, u)$ *from* B *within* z *head-moves by (3)–(4) all* $i \in M$ $a \in R$ $u \in L$ $z \in Z^+$.

By Axiom 1, induction on the computing times there is a Robinson resolution proof, written $\vdash_R$, using Kleene G4 [6] notation.

Corollary 1 *If* $T(i,\, a,\, u)$ *in* z *then* $\vdash_R B \rightarrow T(i, a, u)$ *in* z *all* $i \in M$ $a \in R$ $u \in L$ $z \in Z^+$.

Let W be the nonempty set of all, halting, deterministic Turing machines computing whether G is satisfiable, writing 0 or 1 as output, for all propositional formulas G. (7)

Formally, cf. Clark and Tärnlund [1].

Definition 2 $T(i,\, G,\, \emptyset\,.\,0) \equiv \not\models \neg G$ *and* $T(i,\, G,\, \emptyset\,.\,1) \equiv \models \neg G$ *all* $i \in W$ *propositions* G.[4]

Definition 3 *If* $SAT \in \mathcal{P}$ *then* $\exists u\, T(i,\, G,\, u)$ *in* $c{\cdot}|G|^n$ *some* $i \in W$ $c \;\; n \;\in\; Z^+$ *all propositions* G.

Let d be the name of some $i \in W$, assumed to exist, by Definition 3. (8)

Let $TAUT$ be the set of propositional tautologies. By Corollary 1,

Corollary 2 *If* $SAT \;\in\; \mathcal{P}$ *then* $\vdash_R \; B \;\rightarrow\; T(d,\, \neg F,\, \emptyset\,.\,1)$ *in* $c \cdot |F|^n$ *some* $c\; n \;\in\; Z^+$ *all* $F \in TAUT$ *on DNF.*

Then, by Definitions 2–3 and a terminated unsuccessful computation,

Corollary 3 *If* $SAT \in \mathcal{P}$ *then* $\neg T(d,\, G,\, \emptyset\,.\,1)$ *in* $c \cdot |G|^n$ *some* $c\; n \in Z^+$ *all satisfiable propositions* G.

Let the size of a proof be the number of symbols in the proof. (9)

Definition 4 $|\vdash_{R_p} F| \in O(|F|^n)$ *if there is a propositional Robinson resolution proof of* F *of size* $O(|F|^n)$ *all* $n \in Z^+$ $F \in TAUT$ *on DNF.*

[4] The numbers 0 and 1 are written as sole elements on a list.

4 Lemma 1 and a proof

Formally,[5]

Lemma 1 *If* $SAT \in \mathcal{P}$ *then* $|\vdash_{R_p} F| \in O(|F|^n)$ *some* $n \in Z^+$ *all sufficiently large* $F \in TAUT$ *on DNF.*

Proof.

$$\text{Assume, } SAT \in \mathcal{P}. \tag{10}$$

Thus, by Corollary 2,

$$\begin{array}{c} \vdash_R B \to T(d, \neg F, \emptyset\,.\,1) \text{ in } c \cdot |F|^n \\ c\ n \in Z^+ \quad F \in TAUT \text{ on DNF.} \end{array} \tag{11}$$

Let $U_{j\,k}$ be a short name for the predicate U of Axiom 1. The indexes enumerate the computing time – j –, by (3)–(4), and the search – k – for a quintuple of Turing machine d, by (5), $j\,k \in N$. Let T be $T(d, \neg F, \emptyset\,.\,1)$.

In Robinson resolution, T follows from B in computing time $c \cdot |F|^n$, i.e., (16)–(13) in Proof-tree 1, written on Kleene G4 style.

$$\text{Assume further, } \neg F. \text{ Thus, } \neg T, \text{ by Corollary 3, so, (17). Then,} \tag{12}$$

Proof tree 1 $\vdash_R B \to F$ *in* $j + 1 \leq c \cdot |F|^n$ *by (11)–(12).*

$$B, U_{j\,k_j} \overset{\times}{\to} U_{j\,k_j} \qquad B, U_{(j+1)\,0} \overset{\times}{\to} U_{(j+1)\,0} \tag{13}$$

$$\ddots \qquad \vdots$$

$$B, U_{0\,k_0} \overset{\times}{\to} U_{0\,k_0} \qquad B \to U_{1\,0} \tag{14}$$

$$\backslash \qquad |$$

$$B, T \overset{\times}{\to} T \qquad B, U_{1\,0} \supset U_{0\,k_0} \to U_{0\,k_0} \tag{15}$$

$$\backslash \qquad \vdots$$

$$B, U_{0\,0} \supset T \to T \tag{16}$$

$$|$$

$$B, \neg F, \neg T \to \tag{17}$$

$$|$$

$$B \to F \tag{18}$$

The unification algorithm of Robinson resolution computes a propositional proof of F, from (13) to (18). So, by induction on the computing times,

$$\vdash_{R_p} F. \tag{19}$$

$$\text{Let } \Delta(F) \text{ be the propositional Robinson resolution proof of (19) of } F. \tag{20}$$

[5]Informally, if $SAT \in \mathcal{P}$ then all sufficiently large tautologies F on disjunctive normal form (DNF) have a propositional Robinson resolution proof of polynomial size – in the size of F.

By Proof-tree 1, $j+1 \leq c \cdot |F|^n$ and,

$$\begin{aligned} \Delta(F) \quad = \quad & U_{(j+1)\,0}, U_{(j+1)\,0} \supset U_{j\,k_j}, U_{j\,k_j}, \ldots, U_{j\,0} \supset U_{j-1\,(k_{j-1})}, \ldots, \\ & U_{1\,0}, U_{1\,0} \supset U_{0\,k_0}, U_{0\,k_0}, \ldots, U_{0\,0}, U_{0\,0} \supset T, T, \neg F, \neg T, \Box, F. \end{aligned} \tag{21}$$

There are sufficiently large $F \in TAUT$ such that $|d| < |F|$, for each Turing machine d. The size of each propositional predicate $U_{j\,k}$ and T of $\Delta(F)$ have upper bounds $c \cdot |F|^n$ and $c' \cdot |F|$. The indexes of $U_{j\,k}$ have upper bounds, $j+1, k \leq c \cdot |F|^n$. So, the size $|\Delta(F)|$ of $\Delta(F)$ is,

$$|\Delta(F)| \in O(|F|^{3n}) \text{ sufficiently large } F. \tag{22}$$

So, Definition 4, (22) and discharging (10) give (23), and Lemma 1.

$$\text{If } SAT \in \mathcal{P} \text{ then } |\vdash_{R_p} F| \in O(|F|^{3n}) \text{ sufficiently large } F. \ \Box \tag{23}$$

5 $SAT \notin \mathcal{P}$ and $\mathcal{P} \neq \mathcal{NP}$

Haken's [3] theorem,[6] and Lemma 1 give, by reductio ad absurdum,

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

However, $SAT \in \mathcal{NP}$. Therefore,

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

Corollary 4 *$TAUT \notin \mathcal{P}$.*

Acknowledgment

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, thank you all.

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] Stephen Cook. The complexity of theorem-proving procedures. In *Third Annual ACM Symposium on Theory of Computing*, pages 151–158, New York, NY, USA, 1971. ACM Press.

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

[6] Writing PF_n for, a tautology (on DNF), n pigeons in $n+1$ holes have an empty hole.
Theorem. (Haken) There exists a constant $c, c > 1$, so that, for sufficiently large n, every resolution proof of PF_n, contains at least c^n different clauses.

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

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

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

[7] Alexandre Riazanov and Andrei Voronkov. The design and implementation of VAMPIRE. *AI Communications*, 15(2-3):91–110, 2002.

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

[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. 2nd edition.

[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. $\mathcal{P}$ is not equal to $\mathcal{NP}$. *DiVA e-prints*, November 2013. 4th edition.

[14] Sten-Åke Tärnlund. $\mathcal{P}$ is not equal to $\mathcal{NP}$. *DiVA e-prints*, February 2016. 7th edition.

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