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

Sten-Åke Tärnlund†

January 20, 2016

Abstract

A new axiom of Turing's theory of computing gives: *SAT is not in* $\mathcal{P}$. Thus: $\mathcal{P}$ *is not equal to* $\mathcal{NP}$. A theorem-prover has verified the results (a computer-oriented version) elsewhere.

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. Equivalently, 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 $\mathcal{P} \neq \mathcal{NP}$,[1] cf. Cook [2].

The results follow, chiefly, from the new Axiom 1 of Turing's theory of computing [16], cf. Tärnlund [10]. It opens into a proof of the crucial Lemma 1: If $SAT \in \mathcal{P}$ then all sufficiently large tautologies F on disjunctive normal form (DNF) have a propositional proof of polynomial size (in the size of F) in Robinson resolution [8]. Thus: $SAT \notin \mathcal{P}$, using Haken's theorem[2] [3].

The human-oriented proofs are informal, in Hilbert's proof theory [5], about formal proofs in Robinson resolution systems. These results have been converted into a computer-oriented version that are verified, Tärnlund [12], using a theorem prover Vampire of Riazanov and Voronkov [7].

The postulates of predicate logic (including propositional logic), a consistent subset of Theory **B**, Section 2, and number theory are used.

2 Axiom 1

Axiom 1 is a new single axiom of Turing's theory of computing.[3] It defines a universal Turing machine using a U-predicate — a relationship of a two-way tape x, s, z (the head is at symbol s), a state q, Turing machines j, i, and output u.

*Sixth edition — with changes of style of the Fifth edition, cf. Tärnlund [14].

[1] The proof ideas are similar in all editions, Tärnlund [10, 11, 12, 13, 14, 15].

[2] The tautology PF_n (on DNF) for 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.

[3] Axiom 1 is a slight simplification of a formula in Tärnlund [9].

Axiom 1, in predicate calculus, has five parts: (1) defines $T(i, a, u)$, i.e., Turing machine i — a list of quintuples — computes u for input a; (2) halts at state 0 computing x; (3) reads s, writes r, moves the head one element to the left, goes from state q to state p, and resets j to i; (4) is similar to (3), but the head moves to the right; (5) searches for a quintuple of j given s and q. The intended domains are: a set S of symbols, a set Q of states, a set D of head moves, a set R of right tapes, a set L of left tapes, and a set M of Turing machines, $i\,j \in M\ a\,z \in R\ u\,x \in L\ s\,v\,r\,s' \in S\ p\,q\,q' \in Q\ \ d \in D$.

Axiom 1 *B for*

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

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

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

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

$$U(x,\, s,\, z,\, q,\, j,\, i,\, u) \supset U(x,\, s,\, z,\, q,\, q'\,.\,s'\,.\,p\,.\,r\,.\,d\,.\,j,\, i,\, u). \tag{5}$$

The free variables have the generality interpretation. $\emptyset$, 0, *and* 1 *are constants and* . *is an infix term for lists.*

3 Complexity measures

Let $T(i, a, u)$ in z be Turing machine i computes u for a in z number of moves of the head of i (the computing time), $z \in\ Z^+$, cf. Hartmanis and Stearns [4]. Let $\vdash_{_R} B \to T(i, a, u)$ in z be there exists a proof of $T(i, a, u)$ from B in computing time z in Robinson resolution systems.

Then, by Axiom 1 and induction on the computing times,

Corollary 1 *If $T(i, a, u)$ in z then $\vdash_{_R} B \to T(i, a, u)$ 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 the output 0 — or not — writing the output 1 — for all propositional formulas G.

This input-output relationship of Turing machine $i \in W$ is formalized next (the theory is extended over propositions), cf. Clark and Tärnlund [1].

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

Definition 2 *If $SAT \in \mathcal{P}$ then $\exists u\, T(i,\, G\,.\,\emptyset,\, u)$ in $c{\cdot}|G|^n$ some $c\ n \in\ Z^+\ i \in\ W$ all propositions G.*

Let b be the name of some $i \in W$, assumed to exist, in Definition 2. Let $TAUT$ be the set of propositional tautologies. Then, by Definition 1,

Corollary 2 *$T(b,\, \neg\, F\,.\,\emptyset,\, \emptyset\,.\,1)\ \supset\ F$ all $F \in TAUT$.*

Corollaries 1–2 and Definition 2 give the following result.

[4] The input G and the output 0 and 1 are coded as lists, i.e., $G\,.\,\emptyset$, $\emptyset\,.\,0$, and $\emptyset\,.\,1$.

Corollary 3 *If* $SAT \in \mathcal{P}$ *then* $\vdash_R B, T(b, \neg F . \emptyset, \emptyset . 1) \supset F \to F$ *in* $c \cdot |F|^n$ *some* $c\ n \in Z^+$ *all* $F \in TAUT$ *on DNF.*

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

Definition 3 $|\vdash_R F| \in O(|F|^m)$ *if and only if there is a Robinson resolution proof of* F *in size* $O(|F|^m)$ *all* $m \in Z^+$ *sufficiently large* $F \in TAUT$ *on DNF.*

4 Lemma 1 and a proof

Writing Lemma 1 using Definition 3, cf. Section 1.

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

Proof.

$$\text{Assume that } SAT \in \mathcal{P}. \tag{6}$$

Thus, by Corollary 3,

$$\vdash_R B, T(b, \neg F . \emptyset, \emptyset . 1) \supset F \to F \text{ in } c \cdot |F|^n. \tag{7}$$

A proof in Robinson resolution systems that exists, by (7), is shown on Kleene G4 style in Proof-tree 1. The following notation is used.

Let $U_{j\,k}$ be the short name for a propositional predicate U from Axiom 1. Let j be the computing time and, for each j, let k count the inference steps to find a quintuple of b, $j\,k \in N$. Let T be $T(b, \neg F . \emptyset, \emptyset . 1)$. Then,[5]

Proof-tree 1 *A resolution proof of* F *in computing time* $j + 1 \leq c \cdot |F|^n$.

$$B, U_{j\,k_j} \overset{\times}{\to} U_{j\,k_j} \qquad B, U_{(j+1)\,0} \overset{\times}{\to} U_{(j+1)\,0} \tag{8}$$

$$\ddots \qquad \vdots$$

$$B, U_{0\,0} \overset{\times}{\to} U_{0\,0} \qquad B \to U_{1\,0} \tag{9}$$

$$\backslash \qquad |$$

$$B, T \overset{\times}{\to} T \qquad B, U_{1\,0} \supset U_{0\,0} \to U_{0\,0} \tag{10}$$

$$\backslash \qquad |$$

$$B, F \overset{\times}{\to} F \qquad B, U_{0\,0} \supset T \to T \tag{11}$$

$$\backslash \qquad |$$

$$\vdash B, T \supset F \to F \text{ in } c \cdot |F|^n \tag{12}$$

There is a propositional Robinson resolution proof of F (and its instances)[6] in computing time $c \cdot |F|^n$ in Proof-tree 1. It can be written,

$$\begin{gathered} U_{(j+1)\,0}, U_{(j+1)\,0} \supset U_{j\,k_j}, U_{j\,k_j}, U_{j\,k_j} \supset U_{j\,(k_j-1)}, \ldots, U_{j\,0} \supset U_{(j-1)\,(k_{j-1})}, \\ \ldots, U_{2\,0}, U_{2\,0} \supset U_{1\,k_1}, U_{1\,k_1}, \ldots, U_{1\,1}, U_{1\,1} \supset U_{1\,0}, \\ U_{1\,0}, U_{1\,0} \supset U_{0\,0}, U_{0\,0} \supset T, T, T \supset F, F. \end{gathered} \tag{14}$$

[5] $B, T \supset F, \ldots, U_{(j+1)\,0}$ are logic programs; $F, \ldots, U_{(j+1)\,0}$ are goals, cf. Kowalski [6].

[6] Assuming $\neg F$ gives, $\forall_R B \to T(b, \neg F . \emptyset, \emptyset . 1)$ in computing time $c \cdot |F|^n$. Therefore, $\neg T(b, \neg F . \emptyset, \emptyset . 1)$. Thus, instances have Robinson resolution proofs of F, by (11) and (14),

$$U_{(j+1)\,0}, \ldots, T, \neg F, \neg T, \sqcup, F. \tag{13}$$

Next, it is proved that (7) gives (14), called $\Delta(F)$.

For $b \in W$, let $f : TAUT \to Z^+$ be the map whose value at F is the computing time $z+1$; let r be the map whose value at $z+1$ is the Robinson resolution proof of F; let h be the map whose value at $z+1$ is a partial proof. Then, the map Δ can be defined, e.g., $\Delta(F)$ is (14).

Definition 4 $\Delta = r \circ f$.
$r(1) = \langle U_{1\,0}, U_{1\,0} \supset U_{0\,0}, U_{0\,0} \supset T, T, T \supset F, F \rangle$.
$r(z+1) = \langle h(z+1), r(z) \rangle$.
$h(z+1) = \langle U_{(z+1)\,0}, U_{(z+1)\,0} \supset U_{z\,k_z}, U_{z\,k_z}, U_{z\,(k_z-q)} \supset U_{z\,(k_z-q-1)} \rangle$ *for* $0 \leq q < k_z$ *all* $z \in N$ *some* $k_z \in Z^+$ $q \in N$.

First, by Definition 4, induction on the computing times, and number theory,

$$\text{If } \vdash_R B, T \supset F \to F \text{ in } c \cdot |F|^n \text{ then } \Delta(F). \tag{15}$$

Hence, using (7),

$$\Delta(F). \tag{16}$$

Induction on the computing times, and the formal definition of a propositional Robinson resolution proofs, cf. Tärnlund [10], give,

$$\Delta(F) \text{ is a propositional Robinson resolution proof of } F \text{ on DNF.} \tag{17}$$

Therefore,

$$\vdash_R F \text{ all } F \in TAUT \text{ on DNF.} \tag{18}$$

Second, for each size of b there are sufficiently large $F \in TAUT$ such that $|b| < |F|$. The size of each propositional predicate of $\Delta(F)$ has a polynomial upper bound $c \cdot |F|^n$, $k_z < |F|$. The computing time $z+1 \leq c \cdot |F|^n$.

Thus, by Axiom 1, Definition 4, (16), and number theory,

$$|\Delta(F)| \in O(|F|^{2 \cdot n+1}). \tag{19}$$

Hence, by Definition 3 and (16)–(19),

$$|\vdash_R F| \in O(|F|^{2 \cdot n+1}). \tag{20}$$

Discharging the assumption (6) ends the proof. Therefore,

$$\text{If } SAT \in \mathcal{P} \text{ then } |\vdash_R F| \in O(|F|^n) \text{ some } n \in Z^+ \text{ all sufficiently large } F \in TAUT \text{ on DNF.} \tag{21}$$

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

Lemma 1 and Haken's theorem[2] 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, and the AI course 17 October 2014 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.

[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. Republished 1972; English translation by Lewis Hammond et al., Principles of Mathematical Logic, Chelsea, New York, 1950.

[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. North-Holland.

[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. Also, TRITA-IBADB-1034, 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. $\mathcal{P}$ is not equal to $\mathcal{NP}$. *DiVA e-prints*, November 2013. Fourth edition.

[14] Sten-Åke Tärnlund. $\mathcal{P}$ is not equal to $\mathcal{NP}$. *DiVA e-prints*, July 2015. Fifth edition.

[15] Sten-Åke Tärnlund. AI and theorem-proving using a proof of $\mathcal{P}$ is not equal to $\mathcal{NP}$. *Lecture October 17 2014*, August 2015. On YouTube.

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