

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

Sten-Åke Tärnlund[†]

February 1, 2016

Abstract

A new axiom of Turing's theory of computing yields *SAT is not in \mathcal{P}* . Therefore, *\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. 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].

The results follow, chiefly, from the new Axiom 1 of Turing's theory of computing [17].¹ 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 theorem² [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 .

^{*}Seventh edition — with changes of style of the Sixth edition cf. Tärnlund [16]

[†]Copyright © 2015, Sten-Åke Tärnlund, Stockholm Sweden. gmail: stenake.

¹The proof ideas are similar in all editions, Tärnlund [10, 11, 12, 13, 14, 16, 15]

²The 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.

³Axiom 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

$$U(\emptyset, \emptyset, a.\emptyset, 1, i, i, u) \supset T(i, a, u). \quad (1)$$

$$U(x, s, z, 0, i, i, x). \quad (2)$$

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

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

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

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 input a in z number of moves of the head of i (the computing time), $z \in Z^+$, cf. Hartmanis and Stearns [4]. Let $\vdash_R B \rightarrow T(i, a, u)$ 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 $\vdash_R B \rightarrow T(i, a, u)$ 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 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 Corollary 1 and Definition 2,

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

Thus, cf. Definition 1,

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 $\neg T(b, G, \emptyset, \emptyset, 1)$ in $c \cdot |G|^n$ some $c, n \in \mathbb{Z}^+$ all satisfiable propositions G .

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 a Robinson resolution proof of F exists in size $O(|F|^m)$ all $m \in \mathbb{Z}^+$ sufficiently large $F \in \text{TAUT}$ on DNF.

4 Lemma 1 and a proof

Writing the more informal Lemma 1 from Section 1 using Definition 3.

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

Proof.

$$\text{Assume that } SAT \in \mathcal{P}. \quad (6)$$

So, by Corollary 2,

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

Then, a Robinson resolution proof exists, and shown on Kleene G4 and Kowalski [6] logic programming 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 , and let k count the inference steps to find a quintuple of $b, j, k \in \mathbb{N}$. Let T be $T(b, \neg F, \emptyset, \emptyset, 1)$. Thus,

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

$$B, \neg F, U_{j,k_j} \xrightarrow{\times} U_{j,k_j} \quad B, \neg F, U_{(j+1)_0} \xrightarrow{\times} U_{(j+1)_0} \quad (8)$$

$$\begin{array}{ccc} & \vdots & \\ & \vdots & \\ B, \neg F, U_{0,0} \xrightarrow{\times} U_{0,0} & & B, \neg F \rightarrow U_{1,0} \end{array} \quad (9)$$

$$\begin{array}{ccc} & \backslash & | \\ B, \neg F, T \xrightarrow{\times} T & & B, \neg F, U_{1,0} \supset U_{0,0} \rightarrow U_{0,0} \end{array} \quad (10)$$

$$\begin{array}{ccc} & \backslash & | \\ & & B, \neg F, U_{0,0} \supset T \rightarrow T \end{array} \quad (11)$$

$$\begin{array}{ccc} & | & \\ & B, \neg F, \neg T \rightarrow & \end{array} \quad (12)$$

$\neg F$ is assumed and Corollary 3 is used in (12). Now, there is a propositional Robinson resolution proof of F in computing time $c \cdot |F|^n$. It can be written,

$$\begin{array}{l} U_{(j+1)_0}, U_{(j+1)_0} \supset U_{j,k_j}, U_{j,k_j}, U_{j,k_j} \supset U_{j,(k_j-1)}, \dots, U_{j,0} \supset U_{(j-1),(k_{j-1})}, \\ \dots, U_{2,0}, U_{2,0} \supset U_{1,k_1}, U_{1,k_1}, \dots, U_{1,1}, U_{1,1} \supset U_{1,0}, \\ U_{1,0}, U_{1,0} \supset U_{0,0}, U_{0,0} \supset T, T, \neg F, \neg T, \sqcup, F. \end{array} \quad (13)$$

It will be proved that (7) gives (13), called $\Delta(F)$, using next definition.

For $b \in W$, let $f : \text{TAUT} \rightarrow \mathbb{Z}^+$ 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 (13).

Definition 4 $\Delta = r \circ f$.

$$r(1) = \langle U_{10}, U_{10} \supset U_{00}, U_{00} \supset T, T, \neg F, \neg T, \sqcup, F \rangle.$$

$$r(z+1) = \langle h(z+1), r(z) \rangle.$$

$$h(z+1) = \langle U_{(z+1)0}, U_{(z+1)0} \supset U_{zk_z}, U_{zk_z}, U_{z(k_z-q)} \supset U_{z(k_z-q-1)} \rangle \text{ for } 0 \leq q < k_z$$

all $z \in \mathbb{N}$ some $k_z \in \mathbb{Z}^+$ $q \in \mathbb{N}$.

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

$$\text{If } \vdash_R B \rightarrow T \text{ in } c \cdot |F|^n \text{ then } \Delta(F). \quad (14)$$

Hence, using (7),

$$\Delta(F). \quad (15)$$

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

$$\Delta(F) \text{ is a propositional Robinson resolution proof of } F \text{ on DNF.} \quad (16)$$

Therefore,

$$\vdash_R F \text{ all } F \in \text{TAUT on DNF.} \quad (17)$$

Second, for each size of b there are sufficiently large $F \in \text{TAUT}$ 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, Corollary 3, Definition 4, (15), and number theory,

$$|\Delta(F)| \in O(|F|^{2 \cdot n+1}). \quad (18)$$

Hence, by Definition 3 and (15)–(18),

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

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

$$\text{If } SAT \in \mathcal{P} \text{ then } |\vdash_R F| \in O(|F|^n) \text{ some } n \in \mathbb{Z}^+ \text{ all sufficiently} \quad (20)$$

large $F \in \text{TAUT on DNF.}$

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

Lemma 1 and Haken's theorem² 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 $\text{TAUT} \notin \mathcal{P}$.

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. *Trans. Amer. Math. Soc.*, 117:285–306, 1965.
- [5] David Hilbert and Wilhelm Ackermann. *Grundzüge der theoretischen Logik*. Springer-Verlag, 1928. English translation by Lewis Hammond et al., Principles of Mathematical Logic, Chelsea, New York, 1950, 1972.
- [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. TRITA-IBADB-1034, Royal Institute of Technology 1975, Sweden. 1st ed.
- [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 ed.
- [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 ed.
- [14] Sten-Åke Tärnlund. \mathcal{P} is not equal to \mathcal{NP} . [DiVA e-prints](#), July 2015. 5th ed.
- [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] Sten-Åke Tärnlund. \mathcal{P} is not equal to \mathcal{NP} . [DiVA e-prints](#), January 2016. 6th ed.
- [17] Alan M. Turing. On Computable Numbers with an Application to the Entscheidungsproblem. *Proc. London Math. Soc.*, 2/42:230–265, 1936.