EasyChair Preprint

№ 7447

Short Note on P vs NP

Frank Vega

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

February 11, 2022

Short Note on P vs NP

Frank Vega
CopSonic, 1471 Route de Saint-Nauphary 82000 Montauban, France

Abstract

Under the assumption of certain hypothesis, we show that $P \neq NP$. In this way, we provide another possible tool to prove the P versus NP problem.

2012 ACM Subject Classification Theory of computation Complexity classes; Theory of computation Problems, reductions and completeness

Keywords and phrases complexity classes, graph, completeness, polynomial time

1 Result

A principal *NP–complete* problem is SAT [2]. An instance of SAT is a Boolean formula ϕ which is composed of:

1. Boolean variables: $x_1, x_2, \ldots, x_n$;
2. Boolean connectives: Any Boolean function with one or two inputs and one output, such as $\wedge$(AND), $\vee$(OR), $\neg$(NOT), $\Rightarrow$(implication), $\Leftrightarrow$(if and only if);
3. and parentheses.

A truth assignment for a Boolean formula ϕ is a set of values for the variables in ϕ. A satisfying truth assignment is a truth assignment that causes ϕ to be evaluated as true. A Boolean formula with a satisfying truth assignment is satisfiable. The problem SAT asks whether a given Boolean formula is satisfiable [2]. We define a CNF Boolean formula using the following terms:

A literal in a Boolean formula is an occurrence of a variable or its negation [1]. A Boolean formula is in conjunctive normal form, or CNF, if it is expressed as an AND of clauses, each of which is the OR of one or more literals [1]. A Boolean formula is in 3-conjunctive normal form or $3CNF$, if each clause has exactly three distinct literals [1]. For example, the Boolean formula:

$$(x_1 \vee \neg x_1 \vee \neg x_2) \wedge (x_3 \vee x_2 \vee x_4) \wedge (\neg x_1 \vee \neg x_3 \vee \neg x_4)$$

is in $3CNF$. The first of its three clauses is $(x_1 \vee \neg x_1 \vee \neg x_2)$, which contains the three literals x_1, $\neg x_1$, and $\neg x_2$. We state the following Hypothesis on Boolean formulas in $3CNF$:

▶ **Hypothesis 1.** *There is a general fixed constant c for all set of variables $X = \{x_1, x_2, \ldots, x_n\}$ and a set of truth assignments T_X assigned to X such that there exists a satisfiable Boolean formula ϕ in $3CNF$ using a set of variables Y with at most n^c variables and $X \subseteq Y$. For each satisfying truth assignment T in ϕ, we have there is at least a truth assignment $T' \in T_X$ such that $T' \subseteq T$, which means T' is mapped into the variables in X. For every truth assignment $T' \in T_X$, there exists at least a satisfying truth assignment T in ϕ such that $T' \subseteq T$. Moreover, there is no a satisfying truth assignment T in ϕ such that a truth assignment T' is mapped into the variables in X, $T' \subseteq T$ and $T' \notin T_X$.*

A graph $G = (V, E)$ has V as the set of vertices and E as the set of edges, each edge being a pair of vertices [1]. We say $(u, v) \in E$ is an edge in a graph $G = (V, E)$ where u and v are vertices: We say that u and v are adjacent. For a graph $G = (V, E)$, a simple path in G is a sequence of distinct vertices $\langle v_0, v_1, v_2, ..., v_k \rangle$ such that $(v_{i-1}, v_i) \in E$ for $i = 1, 2, ..., k$ [1]. A

Hamilton path is a simple path of a graph which contains all the vertices of the graph [1]. Interestingly, a linear order P on the nodes of G describes the existence of a Hamilton path, that is, a binary relationship isomorphic to $<$ on the nodes of G (without loss of generality, these nodes are $\{0, 1, 2, \ldots, n-1\}$) such that consecutive nodes are connected in G [3]. The properties of P require several things. We say that a tuple (x, y) is appropriated for the binary relation P when (x, y) belongs to P. First, all distinct nodes of G are comparable by P [3]:

$$\forall x \forall y ((P(x, y) \vee P(y, x)) \vee x = y).$$

Next, P must be transitive but not reflexive [3]:

$$\forall x \forall y \forall z ((\neg P(x, x)) \wedge ((P(x, y) \wedge P(y, z)) \Rightarrow P(x, z))).$$

Finally, any two consecutive nodes in P must be adjacent in G [3]:

$$\forall x \forall y ((P(x, y) \wedge \forall z (\neg P(x, z) \vee \neg P(z, y))) \Rightarrow G(x, y))$$

where $G(x, y)$ means that (x, y) is an edge on G. The existence of such linear order P with these properties guarantee the existence of a Hamilton path on G [3].

In computational complexity theory, *SUCCINCT HAMILTON PATH* is a well-known problem in *NEXP–complete* [3]. A succinct representation of a graph with n nodes, where $n = 2^b$ is a power of two, is a Boolean circuit C with $2 \times b$ input gates [3]. The graph represented by C, denoted G_C, is defined as follows: The nodes of G_C are $\{0, 1, 2, \ldots, n-1\}$ and (i, j) is an edge of G_C if and only if C accepts the binary representations of the b-bits integers i, j as input [3].

▶ **Definition 2.** ***SUCCINCT HAMILTON PATH***
INSTANCE: A succinct representation C of a graph G_C with n nodes.
QUESTION: Does G_C have a Hamilton path?
REMARKS: We know that SUCCINCT HAMILTON PATH $\in$ NEXP–complete [3].

Given a succinct representation C of a graph G_C with n nodes, where $n = 2^b$ is a power of two, if the Hypothesis 1 is true and $C \in$ *SUCCINCT HAMILTON PATH*, then there exists a Boolean formula Q in $3CNF$ bounded by less than $(3 \times b)^c$ variables and $(3 \times b)^{4 \times c}$ clauses. $Q(x, y)$ means the remaining formula after evaluating Q in the first $2 \times b$ variables that correspond to the bits of the b-bits integers x, y. In addition, Q could represent a linear order P such that $P(x, y)$ holds if and only if the Boolean formula $Q(x, y)$ is satisfiable. Similarly, we say that $C(x, y)$ accepts when the Boolean circuit C has been evaluated in the binary representations of the b-bits integers x, y and the output is 1 (or simply true). Moreover, this linear order P that represents Q could comply the properties mentioned above when G_C has a Hamilton path and thus, we can confirm that $C \in$ *SUCCINCT HAMILTON PATH*.

We can apply the Hypothesis 1 and obtain the formula Q, because the linear order P is a binary relation between integers represented by a set of variables $X = \{x_1, x_2, \ldots, x_{2 \times b}\}$ and a set of truth assignments T_X assigned to X, where T_X contains the truth assignments for the $2 \times b$ variables that correspond to the bits of the b-bits integers x, y when (x, y) belongs to P. Since the set X has a cardinality of $2 \times b$, the set of variables in Q has at most $(2 \times b)^c$ elements (this is bounded by the amount of $(3 \times b)^c$). Since every clause of a formula in $3CNF$ has exactly 3 literals, then we would obtain at most a combination of $2 \times (2 \times b)^c$ literals within sets of three elements (this is bounded by the amount of $(3 \times b)^{4 \times c}$). Note that, the set X corresponds to the first $2 \times b$ variables in Q and so, every appropriated tuple (x, y) in the binary relation P would be a truth assignment to the variables in X that will be contained into a satisfying truth assignment of Q. Indeed, $Q(x, y)$ will be a satisfiable formula if and only if the pair (x, y) belongs to P, because of the Hypothesis 1 which guarantee the existence of such Boolean formula Q and its constraints.

Basically, we could represent an appropriated tuple (x, y) of the linear order P if and only if $Q(x, y)$ is satisfiable. In this way, we could represent the first property of P:

$$\forall x \forall y ((P(x, y) \vee P(y, x)) \vee x = y)$$

as the computational problem of solving the Boolean formula with quantified variables,

$$\forall x \forall y ((Q(x, y) \vee Q(y, x)) \vee \psi(x, y))$$

where the Boolean formula ψ is satisfied when $x = y$. We can see that, the other variables in Q, which are not in the set X, remain as free variables inside of this kind of Boolean formula. In addition, we could represent the other properties:

$$\forall x \forall y \forall z ((\neg P(x, x)) \wedge ((P(x, y) \wedge P(y, z)) \Rightarrow P(x, z))).$$

and

$$\forall x \forall y ((P(x, y) \wedge \forall z (\neg P(x, z) \vee \neg P(z, y))) \Rightarrow G(x, y))$$

as the computational problems of solving the Boolean formulas with quantified variables,

$$\forall x \forall y \forall z ((\neg Q(x, x)) \wedge ((Q(x, y) \wedge Q(y, z)) \Rightarrow Q(x, z))).$$

and

$$\forall x \forall y ((Q(x, y) \wedge \forall z (\neg Q(x, z) \vee \neg Q(z, y))) \Rightarrow F(x, y))$$

where F is the Boolean function that represents the circuit C ($F(x, y)$ is satisfied if and only if $C(x, y)$ accepts). We know the bit-length of the formulas $Q(x, y)$, $\psi(x, y)$ and $F(x, y)$ are polynomially bounded by the bit-length of the circuit C according to the Hypothesis 1 since all problems in P have polynomial circuits such as checking whether two sequences of bits are equals or whether a Boolean circuit accepts after being evaluated all its input gates [3].

Note also that, solving those Boolean formulas with quantified variables signifies the necessity of computing instances of problems that can be solved in polynomial time when $P = NP$ [3]. Under the assumption that $P = NP$, we would have a succinct certificate for the instance $C \in$ *SUCCINCT HAMILTON PATH* that could be the formula Q, where we should be able to check the existence of the Hamilton path using Q by a deterministic Turing machine in polynomial time. However, this is exactly the definition of NP. If there is any single problem in *NEXP–complete* that it is also in NP, then $NP = NEXP$. However, $NP \neq NEXP$ is a previous known result [3]. If we assume that $P = NP$ and the Hypothesis 1 is true, then this implies that *SUCCINCT HAMILTON PATH* should be in NP which is trivial contradiction. Consequently, we obtain that necessarily $P \neq NP$ under the assumption that the Hypothesis 1 is true.

References

1 Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms.* The MIT Press, 3rd edition, 2009.

2 Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* San Francisco: W. H. Freeman and Company, 1 edition, 1979.

3 Christos H. Papadimitriou. *Computational complexity.* Addison-Wesley, 1994.