On #P-completeness of Some Counting Problems

Nadia CREIGNOU
Département de Mathématiques
Université de Caen
14032 Caen, France

Miki HERMANN*
CRIN (CNRS) and INRIA-Lorraine
BP 239
54506 Vandœuvre-lès-Nancy, France

Résumé

Nous prouvons que les problèmes #1-in-3Sat, #Not-All-Equal 3Sat et #3-Colorabilité, dont les problèmes de décision correspondants font partie des problèmes les plus fréquemment utilisés pour prouver la NP-complétude de nouveaux problèmes, sont #P-complets. D'une part, la preuve explicite de la #P-complétude de #1-in-3Sat pourrait être utile dans le cadre des preuves de complexité en unification équationnelle. D'autre part, le fait que la #3-Colorabilité est #P-complète nous permet de déduire immédiatement que de nombreux problèmes NP-complets ont une version énumérative #P-complète.

De plus, ce travail met une nouvelle fois en évidence l'intérêt d'exhiber des réductions linéaires entre problèmes de la classe NP.

Abstract

We prove that the counting problems #1-in-3Sat, #Not-All-Equal 3Sat and #3-Colorability, whose decision counterparts have been the most frequently used in proving NP-hardness of new decision problems, are #P-complete. On one hand, the explicit #P-completeness proof of #1-in-3Sat could be useful to prove complexity results within unification theory. On the other hand, the fact that #3-Colorability is #P-complete allows us to deduce immediately that the enumerative versions of a large class of NP-complete problems are #P-complete.

Moreover, our proofs shed some new light on the interest of exhibiting linear reductions between NP problems.

Keywords: counting class, counting problem, #P-completeness, parsimonious reduction, satisfiability.

*Partially supported by *Institut National Polytechnique de Lorraine* grant 910 0146 R1.

1 Introdution

Counting problems represent the quantitative counterpart to decision problems. The complexity class NP and NP-complete decision problems have been exhaustively studied in the literature (see [GJ79] for an overview). Valiant [Val79a, Val79b] introduced the complexity class #P and proved several counting problems to be #P-complete. The class #P is defined as the class of counting problems whose decision problem is in NP, that is, $f \in \#P$ if and only if there is a nondeterministic Turing machine M that runs in polynomial time with the property that $f(x)$ equals the number of accepting computation paths of M on input x. If A is a decision problem, let us denote the corresponding counting problem by #A.

The counting problem #Sat (the number of satisfying assignments of a propositional CNF formula) is known to be #P-complete, mainly because Cook's generic transformation is parsimonious in the sense that the number of satisfying assignments to the Boolean formula corresponds exactly to the number of accepting computations of the nondeterministic Turing machine being simulated. The reduction from Sat to 3Sat can be made parsimonious [Koz92], thus proving the #P-hardness part of #3Sat. Intuitively, it seems that almost for each NP-complete problem the corresponding counting problem is #P-complete. This is often repeated in the literature [Joh90, page 107][PB83, page 779][Sim77, page 484], but the explicite proofs are omitted. In fact, it is sometimes not straightforward to find an appropriate reduction [Val79b, Gal74]. Moreover, not all NP decision problems have counting counterparts that belong to the class #P. For example, AC-unification as a decision problem is NP-complete [KN92] but there are AC-unification problems whose minimal complete set of unifiers (see [FH86] for the definition) has double exponential cardinality [Dom92]. This situation is due to the fact that the decision problem asks for the existence of a unifier, not necessarily a member of the minimal complete set. On the other hand, computing the (minimal) complete set of unifiers is a crucial problem in automated deduction in particular and artificial intelligence in general to guarantee the completeness of many procedures or algorithms which use AC-unification. Thus, the AC-unification counting problem cannot belong to the counting class #P, since each counting problem belonging to #P contains, by definition, only a simple exponential number of different solutions.

A polynomial reduction from 1-in-3Sat is the NP-hardness proof of many interesting problems, especially in automated deduction. Unification modulo idempotence [KN92] and modulo unit [TA87] are proved NP-complete by a reduction from 1-in-3Sat. Within the unification theory, we are interested not only in decision problems (whether two terms are unifiable in a given theory) but also in counting problems (how many substitutions contains the minimal complete set of unifiers), therefore the interest in counting classes. We could perhaps adapt the polynomial reductions from NP-hardness proofs to parsimonious ones from the counting problems #1-in-3Sat or #Mono 1-in-3Sat, proving this way the #P-hardness of the corresponding counting problems, provided #1-in-3Sat and #Mono 1-in-3Sat are proved #P-complete. The decision problem 1-in-3Sat was proved to be NP-complete by Schaefer [Sch78] but the corresponding counterpart was not proved to be #P-complete.

The purpose of our paper is to prove the #P-completeness of several counting problems such as #1-in-3Sat, #Not-All-Equal 3Sat (and their variants) and #3-Colorability. In this way we give an effective proof of the #P-completeness of the counting versions of some

decision problems that are among those that have been most frequently used to get proofs of NP-completeness. Finally, a prior work on the class of problems that are linearly equivalent to Satisfiability enables us to prove that a the counting versions of a large class of problems are #P-complete.

2 Counting problems

We consider the following counting problems.

#1-in-3Sat (#Mono-1-in-3Sat)
Instance: Set V of Boolean variables, a Boolean formula $\mathcal{B}$ over V in conjunctive normal form where each clause of $\mathcal{B}$ has exactly three (all positive or all negative) literals.
Question: How many truth assignments for V satisfy $\mathcal{B}$ with exactly *one* true literal in each clause ?

The problem #2-in-3Sat (respectively #Mono-2-in-3Sat) requires exactly *two* true literals in each clause.

#Not-Exactly-One 3Sat (#Not-Exactly-One Horn-3Sat)
Instance: Set V of Boolean variables, a Boolean formula $\mathcal{B}$ over V in conjunctive normal form where each clause of $\mathcal{B}$ has exactly three literals (and at most one unegated variable).
Question: How many truth assignments for V satisfy $\mathcal{B}$ with no clause having exactly *one* true literal ?

#Not-All-Equal 3Sat (#Mono-Not-All-Equal 3Sat)
Instance: Set V of Boolean variables, a Boolean formula $\mathcal{B}$ over V in conjunctive normal form where each clause of $\mathcal{B}$ has exactly three (all positive or all negative) literals.
Question: How many truth assignments for V satisfy $\mathcal{B}$ with at least one true and one false literal in each clause ?

#3-Colorability
Instance: Graph $G = (V, E)$.
Question: How many 3-colorings defined on V exist for G ? In another words, how many functions $col: V \to \{1, 2, 3\}$ exist such that $col(u) \neq col(v)$ whenever $(u, v) \in E$?

Let us recall also the counting problems which were proved #P-complete before and which will be reduced to previously mentioned problems.

Positive 2Sat **(#Pos-2Sat)** proved #P-complete (called *monotone*) by Valiant [Val79b]
Instance: Set V of Boolean variables, a Boolean formula $\mathcal{B}$ over V in conjunctive normal form where each clause of $\mathcal{B}$ has exactly two positive literals.
Question: How many truth assignments for V satisfy $\mathcal{B}$?

Bipartite positive 2Sat **(#BPos-2Sat)** proved #P-complete by Provan and Ball [PB83]
Instance: Two disjoint sets of variables $X = \{x_1, \ldots, x_k\}$, $Y = \{y_1, \ldots, y_l\}$ and the Boolean formula $\mathcal{B} = (x_{i1} \vee y_{j1}) \wedge \cdots \wedge (x_{in} \vee y_{jn})$.
Question: How many truth assignments for V satisfy $\mathcal{B}$?

Implicative 2Sat **(#Impl-2Sat)** proved #P-complete by Linial [Lin86]
Instance: Set V of Boolean variables, a Boolean formula $\mathcal{B}$ over V in conjunctive normal form where each clause of $\mathcal{B}$ has exactly one positive and one negative literal.
Question: How many truth assignments for V satisfy $\mathcal{B}$?

In the sequel, a k-clause means a clause containing k literals. An *implicative clause* means a clause containing exactly one positive and one negative literal.

3 Counting class #P

Let Σ and Γ be nonempty alphabets. Let $w\colon \Sigma^* \to \mathcal{P}(\Gamma^*)$, where $\mathcal{P}(\Gamma^*)$ denotes the power set of Γ^*, and let $x \in \Sigma^*$. We refer to the elements of $w(x)$ as *witnesses* for x.

For satisfiability problem, let $x \in \Sigma^*$ be an encoding of the Boolean formula $\mathcal{B}$ and $y \in \Gamma^*$ an encoding of a truth assignment. The witness set is

$$w(x) \;=\; \{\text{truth assignments } y \text{ satisfying } x \text{ and } Q(x,y)\}$$

where $Q(x,y)$ is a predicate on x and y, expressing e.g. that each clause of $\mathcal{B}$ has exactly three literals, or that each satisfying assignment for $\mathcal{B}$ has exactly one true value, etc.

For #3-Colorability, let $x \in \Sigma^*$ be an encoding of the graph $G = (V, E)$ and $y \in \Gamma^*$ an encoding of a coloring function $col\colon V \to \{0,1,2\}$. The witness set is

$$w(G) \;=\; \{col\colon V \to \{0,1,2\} \mid (col(u) \neq col(v)) \equiv ((u,v) \in E)\}$$

N denotes the natural numbers, $|x|$ is the size of the string x, and $|S|$ is the cardinality of a set S.

Definition 3.1 ([Koz92]) *The class #P is the class of witness functions w such that:*

1. *there is a polynomial-time algorithm to determine, for given x and y, if $y \in w(x)$;*
2. *there exists a constant $k \in N$ such that for all $y \in w(x)$, $|y| \leq |x|^k$.*
   (The constant k can depend on w).

Using this definition it is clear that all the problems proposed here are in the counting class #P.

4 Parsimonious reductions and #P-complete problems

To prove #P-hardness, we need reductions from already known #P-complete problems. It is necessary to observe how counting problems v and w are related under the process of reduction. The notion of *counting reductions* and *parsimonious reductions* have been introduced for this purpose.

Definition 4.1 *Let $w\colon \Sigma^* \to \mathcal{P}(\Gamma^*)$ and $v\colon \Pi^* \to \mathcal{P}(\Delta^*)$ be counting problems. A* weakly parsimonious reduction *from w to v consists of a pair of polynomial-time computable functions $\sigma\colon \Sigma^* \to \Pi^*$ and $\tau\colon \Sigma^* \times N \to N$ such that $|w(x)| = \tau(x, |v(\sigma(x))|)$.*

A counting reduction σ is parsimonious *if $|w(x)| = |v(\sigma(x))|$.*

Let w and v be counting problems. Note $w \leq_! v$ and $w \leq_{!!} v$ if there is a *weakly parsimonious* and *parsimonious* reduction from v to w, respectively.

The #P-complete problems are the most difficult problems in the class #P.

Definition 4.2 *A counting problem w is* **#P-hard** *if $v \leq_! w$ for all problems $v \in \#P$. A counting problem w is* **#P-complete** *if it is #P-hard and $w \in \#P$.*

If w is #P-complete and there is a weakly parsimonious reduction from w to v then v is #P-hard.

The reductions in the #P-hardness proofs must preserve the number of solutions, possibly with a certain factor. Therefore the necessity to look for parsimonious or weakly parsimonious reductions. Note that a composition of (weakly) parsimonious reductions is a (weakly) parsimonious reduction.

Theorem 4.3 *The counting problems #1-in-3Sat and #Mono 1-in-3Sat are #P-complete.*

Proof: It is clear that these problems are in the class #P. To show that they are #P-hard it is sufficient to prove that prove that #Mono-1-in-3Sat is #P-hard. For this we first show that there is a parsimonious reduction from #Impl-2Sat to #Mono-2-in-3Sat.

Suppose that we are given an instance of the #Impl-2Sat problem with the variables V and the implicative clauses $C = \{c_1, \ldots, c_n\}$, with the Boolean formula $\mathcal{B} = c_1 \wedge \cdots \wedge c_n$. Let $X = \{x_{12}, x_{13}, \ldots, x_{n2}, x_{n3}\}$ and $Y = \{y_{12}, y_{13}, y_{14}, \ldots, y_{n2}, y_{n3}, y_{n4}\}$ be sets of new variables. Let $V' = V \cup X \cup Y$.

To each implicative clause $c_i = \bar{v}_{i1} \vee v_{i2}$ we associate four new clauses

$$\begin{array}{lcl} c_{i1} & = & v_{i1} \vee x_{i2} \vee x_{i3} \\ c_{i2} & = & v_{i2} \vee x_{i2} \vee y_{i2} \\ c_{i3} & = & v_{i2} \vee x_{i3} \vee y_{i3} \end{array} \qquad\qquad c_{i4} \;=\; x_{i3} \vee y_{i2} \vee y_{i4}$$

Let $\mathcal{B}'$ be the Boolean formula constructed from $\mathcal{B}$ by replacing each clause c_i by the conjunction $c_{i1} \wedge c_{i2} \wedge c_{i3} \wedge c_{i4}$.

First, we show that for each truth assignment for V satisfying $\mathcal{B}$ there exists a unique enlargement to a truth assignment for V' satisfying $\mathcal{B}'$ with exactly *two* true literals in each clause.

v_{i1}	v_{i2}	x_{i2}	x_{i3}	y_{i2}	y_{i3}	y_{i4}
0	0	1	1	1	1	0
0	1	1	1	0	0	1
1	1	0	1	1	0	0

Figure 1: Enlargements of truth assignments from $\mathcal{B}$ to $\mathcal{B}'$.

Each truth assignement satisfying $\mathcal{B}$ evaluates the variables v_{i1} and v_{i2} in each clause c_i in one of the following ways: (1) v_{i1} false and v_{i2} false; (2) v_{i1} false and v_{i2} true; (3) v_{i1} true and v_{i2} true.

Case 1: If v_{i1} and v_{i2} are evaluated as false, then the variables x_{i2}, x_{i3}, y_{i2} and y_{i3} must be true. The variables x_{i3} and y_{i2} are true, thus y_{i4} must be false. All variables in the clauses c_{i1}, ..., c_{i4} have been determined.

Case 2: If v_{i1} is evaluated as false and v_{i2} as true. The variable v_{i1} is false, thus the variables x_{i2} and x_{i3} are true following clause c_{i1}. The variables v_{i2}, x_{i2}, and x_{i3} are true, thus y_{i2} and y_{i3} are false. The variable x_{i3} is true and y_{i2} is false, thus y_{i4} is true. All variables in the clauses c_{i1}, ..., c_{i4} have been determined.

Case 3: If v_{i1} and v_{i2} are evaluated as true. Suppose that x_{i2} is true, then y_{i2} must be false since v_{i2} is already true. If v_{i1} and x_{i2} are true then x_{i3} must be false following clause c_{i1}. But we have now the situation where x_{i3} is false and y_{i2} is false, thus the clause c_{i4} cannot have two true literals. Hence, x_{i2} must be false. The variable v_{i1} is true and x_{i2} is false, thus x_{i3} is true. The variables v_{i2} and x_{i3} are true, thus y_{i3} must be false. The variable v_{i2} is true and x_{i2} is false, thus y_{i2} is true. The variables x_{i3} and y_{i2} are true, thus y_{i4} is false. All variables in the clauses c_{i1}, ..., c_{i4} have been determined.

We proved that each truth assignment satisfying $\mathcal{B}$ determines a unique enlargement satisfying $\mathcal{B}'$ with exactly two true literals in each clause. Figure 1 summarizes the truth assignments.

Conversely, we must prove that the restriction to the variables V of each 2-in-3 truth assignment satisfying $\mathcal{B}'$ satisfies also the original formula $\mathcal{B}$. Suppose that $\mathcal{B}'$ is evaluated as true, thus each clause c_{ij} is true. One of the literals in the clause c_{i1} must be false. If x_{ik} is false then the two other literals in the clause c_{ik}, in particular v_{i2}, must be true. If v_{i1} is false then $\bar{v}_{i1}$ is true. Thus, at least one of the literals $\bar{v}_{i1}$ and v_{i2} in each clause c_i is true. Hence, $\mathcal{B}$ is true.

We proved that the reduction from #Impl-2Sat to #2-in-3Sat is parsimonious. It is sufficient now to negate all literals in each clause to get a parsimonious reduction from #Mono-2-in-3Sat to #Mono-1-in-3Sat. The #P-hardness of #1-in-3Sat follows from the #P-completeness of #Mono 1-in-3Sat. □

Theorem 4.4 *The counting problem #Not-Exactly-one 3Sat is #P-complete.*

Proof: It is clear that this problem belongs to #P. To show that it is #P-hard we show that there is a parsimonious reduction from #Pos-2Sat.

Suppose that we are given an instance of the #Pos-2Sat problem with the variables V and the 2-clauses $C = \{c_1, \ldots, c_n\}$, where $\mathcal{B} = c_1 \wedge \cdots \wedge c_n$. Let $V' = V \cup \{x, y\}$, where x and y are new variables.

To each clause $c_i = v_{i1} \vee v_{i2}$ we associate the clause $c'_i = v_{i1} \vee v_{i2} \vee x$. Let us consider the new clauses $c' = x \vee \bar{x} \vee y$ and $d' = y \vee \bar{y} \vee x$ and the new formula

$$\mathcal{B}' \quad = \quad c'_1 \wedge \cdots \wedge c'_n \wedge c' \wedge d'$$

The case analysis with constraints for evaluating logical variables proves that each truth assignment satisfying $\mathcal{B}$ determines a unique enlargment on V' with no clause in $\mathcal{B}'$ having exactly one true literal. The literals x and y must be evaluated as true following the clauses c' and d' and conversely, the restriction to the variables V of each truth assignment on V', with no clause in $\mathcal{B}'$ having exactly one true literal, satisfies the original formula $\mathcal{B}$.

Thus, the reduction from #Pos-2Sat to #Not-Exactly-One 3Sat is parsimonious. □

Remark. Let us notice that this reduction, in negating all the literals, also provides a proof of the #P-completeness of the same problem restricted to Horn 3-clauses. It is interesting to note that the decision problem Not-Exactly-One Horn-3Sat is in P (see [Sch78]). Thus, in this particular case we show that the corresponding counting problem is #P-complete, i.e. considerably more difficult than the original decision problem, provided P $\neq$ NP.

Theorem 4.5 *The counting problems #Not-All-Equal 3Sat and #Mono-Not-All-Equal 3Sat are #P-complete.*

Proof: It is clear that these problems belong to #P. To show that there are #P-hard we show that there are weakly parsimonious reductions from #1-in-3Sat and #Mono-1-in-3Sat respectively.

Suppose that we are given an instance of the #1-in-3Sat problem with the variables V and the clauses $C = \{c_1, \ldots, c_n\}$, where $\mathcal{B} = c_1 \wedge \cdots \wedge c_n$. Let $V' = V \cup \{t\}$, where t is a new variable. To each clause $c_i = l_{i1} \vee l_{i2} \vee l_{i3}$ we associate four new clauses

$$\begin{array}{lcl} c_{i1} & = & l_{i1} \vee l_{i2} \vee t \\ c_{i2} & = & l_{i2} \vee l_{i3} \vee t \\ c_{i3} & = & l_{i3} \vee l_{i1} \vee t \end{array} \qquad\qquad c_{i4} \quad = \quad l_{i1} \vee l_{i2} \vee l_{i3}$$

Let $\mathcal{B}'$ be the Boolean formula constructed from $\mathcal{B}$ by replacing each clause c_i by the conjunction $c_{i1} \wedge c_{i2} \wedge c_{i3} \wedge c_{i4}$.

First, we show that for each truth assignment for V, satisfying $\mathcal{B}$ with exactly *one* true literal in each clause, there exists a unique enlargement to a truth assignment for V' satisfying $\mathcal{B}'$ with at least *one* true and *one* false literal in each clause.

Let us consider a truth assignment for V satisfying $\mathcal{B}$ with exactly *one* true literal in each clause. Since two of the literals l_{i1}, l_{i2}, and l_{i3} are false, the variable t must be true following one of the clauses c_{i1}, c_{i2} or c_{i3}.

Conversely, let s' be a truth assignment satisfying $\mathcal{B}'$ with at least *one* true and *one* false literals in each clause. Such a truth assignment evaluates t either as true or false.

If the variable t is evaluated as true, then at least one of the two literals in each pair (l_{i1}, l_{i2}), (l_{i2}, l_{i3}), and (l_{i3}, l_{i1}) is false following the clauses c_{i1}, c_{i2}, and c_{i3}. But following clause c_{i4}, one of the literals l_{i1}, l_{i2} or l_{i3} must be true. Therefore, exactly one of the three literals l_{i1}, l_{i2} and l_{i3} is true. Thus, the restriction to the variables V of the truth assignment s' satisfies also the original formula $\mathcal{B}$ with exactly *one* true literal in each clause.

If the variable t is evaluated as false, then at least one of the two literals in each pair (l_{i1}, l_{i2}), (l_{i2}, l_{i3}), and (l_{i3}, l_{i1}) is true following the clauses c_{i1}, c_{i2}, and c_{i3}. But following clause c_{i4}, one of the literals l_{i1}, l_{i2} or l_{i3} must be false. Therefore, exactly one of these three literals is false. Observe that the number of satisfying assignments for $\mathcal{B}'$ with exactly one true literal in each clause is equal to the number of satisfying assignments for $\mathcal{B}'$ with exactly two true literals in each clause. If s is a satisfying assignment with exactly one true literal in each clause then $\bar{s}$, where $\bar{s}(v) = 1 - s(v)$ for each $v \in V$, is a satisfying assignment with exactly two true literals in each clause.

We showed that if t is evaluated as true then exactly one literal in each clause in $\mathcal{B}$ is evaluated as true, and if t is evaluated as false then exactly two literals in each clause in $\mathcal{B}$ are evaluated as true. Hence, each restriction of a satisfying assignment for $\mathcal{B}'$ to the variables V is either 1-in-3 or 2-in-3. There is no restriction in which one clause of $\mathcal{B}$ has exactly one true literal and another clause of $\mathcal{B}$ has exactly two true literals.

From these observations, we conclude that the number of truth assignment satisfying $\mathcal{B}'$ with at least one true and one false literals in each clause equals two times the number of truth assignment satisfying the original formula $\mathcal{B}$ with exactly one true literal in each clause.

We showed that the reduction from #1-in-3Sat to #Not-All-Equal 3Sat is weakly parsimonious, thus completing the proof of the #P-hardness of #Not-All-Equal 3Sat. The same reduction from #Mono-1-in 3Sat proves that #Mono-Not-All-Equal 3Sat is also #P-complete. □

Although the following theorem is subsumed by the Linial's result that the 3-coloring of a bipartite graph is #P-complete [Lin86], we present it here since we prove its #P-hardness by a weakly parsimonious reduction from #Not-All-Equal 3Sat.

Theorem 4.6 *The counting problem #3-Colorability is #P-complete.*

Proof: It is easy to see that this problem belongs to the class #P. The classical linear reduction [Dew82] from Not-All-Equal 3Sat to 3-Colorability is suitable for proving that #3-Colorability is #P-hard. Let us remind the reader of this transformation and prove that it is weakly parsimonious.

Suppose that we are given an instance of the #Not-All-Equal 3Sat problem with the variables V and the clauses $C = \{c_1, \ldots, c_n\}$, where $\mathcal{B} = c_1 \wedge \cdots \wedge c_n$. The corresponding input for #3-Colorability is the graph $G = (V, E)$ specified as follows. Its set of vertices is $V = V_1 \cup V_2$ where,

$$\begin{aligned} V_1 &= \{control\} \cup \{v, \bar{v} \mid v \in V\} \\ V_2 &= \{s_i(l_i^0), s_i(l_i^1), s_i(l_i^2) \mid c_i = (l_0^i \vee l_2^i \vee l_2^i),\ 1 \leq i \leq n\} \end{aligned}$$

Its set of edges is $E = E_1 \cup E_2$ where,

$$\begin{aligned} E_1 &= \{\langle control, v\rangle,\ \langle control, \bar{v}\rangle,\ \langle v, \bar{v}\rangle \mid v \in V\} \\ E_2 &= \{\langle l_i^0, s_i(l_i^0)\rangle,\ \langle l_i^1, s_i(l_i^1)\rangle,\ \langle l_i^2, s_i(l_i^2)\rangle \mid 1 \leq i \leq n\} \cup \\ &\quad \{\langle s_i(l_i^0), s_i(l_i^1)\rangle,\ \langle s_i(l_i^1), s_i(l_i^2)\rangle,\ \langle s_i(l_i^2), s_i(l_i^0)\rangle \mid 1 \leq i \leq n\} \end{aligned}$$

In fact, the triangle for each i in E_2 represents the clause c_i and will be denoted by T_i.

Let I be a truth assignment satisfying $\mathcal{B}$ with at least one true and one false literal in each clause. Without loss of generality we can suppose that $I(l_0^i) = 0$ and $I(l_1^i) = 1$. So, up to isomorphism we can define a 3-coloring of G by letting $col(control) = 2$, $col(v) = I(v)$ and $col(\bar{v}) = 1 - I(v)$ for $v \in V$. Then, it is easy too see that for each triangle T_i there are exactly two different valid colorings of the vertices $s_i(l_i^0)$, $s_i(l_i^1)$, and $s_i(l_i^2)$.

Conversely, suppose that $col\colon V \to \{0, 1, 2\}$ is a valid 3-coloring of G. Up to isomorphism we can suppose that $col(control) = 2$. Thus, we can define a truth assignment on V by letting $I(v) = col(v)$. It is easy to see that this truth assignment satisfy $\mathcal{B}$ with necessarily at least one true and one false literal.

From these observations we can conclude that the number of valid 3-coloring for G equals $3! \times 2^n$ times the number of truth assignment satisfying the original formula with at least one true and one false literal.

We showed that the reduction from #Not-All-Equal 3Sat to #3-Colorability is weakly parsimonious, thus completing the proof of the #P-hardness of #3-Colorability. □

Linear equivalence to Satisfiability was studied in [Cre93, Cre92]. Among decision problems linearly equivalent to Sat we can find 3-Domatic Number, Path With Forbidden Pairs, Partition into Hamiltonian Subgraphs, 2-Partition into Perfect Matchings, Partition into Paths of Length 2, 3-Dimensional Matching, and Partition into Triangles

Corollary 4.7 *The counting versions of the decision problems linearly equivalent to Satisfiability are #P-complete.*

Proof: On one hand, it is easy to verify that all these problems belong to the class #P. On the other hand, one can easily verify that all the linear transformations proposed in [Cre93, Cre92], from 3-Colorability to these problems are in fact weakly parsimonious. Thus the corollary follows from Theorem 4.6. □

5 Conclusion

We proved here that the problems #Mono-1-in-3Sat, #Mono-Not-All-Equal 3Sat, #3-Colorability and a large class of other problems are #P-complete. Let us notice that all the reductions provided are computable in linear time. Besides to the interest of our result to obtain new #P-complete problems, we should notice that this proof strengthens the hypothesis according to which the notion of linear reductions between NP-complete problems is useful and natural (see [Cre92, Gra93]). In fact, linear time reductions allow to simplify and to standardize many proofs of NP-hardness [Cre92], they often naturally preserve the approximation algorithms for NP-complete optimization problems (observe for example that the

L-reductions proposed by Papadimitriou and Yannakakis [PY91] are linear time computable) and finally they seem to be useful to get parsimonious reductions in a natural way.

References

[Cre92] N. Creignou. The class of problems that are linearly equuivalent to Satisfiability or a uniform method for proving NP-completeness. In E. Börger, G. Jäger, H. Kleine Büning, S. Martini, and M.M. Richter, editors, *Proceedings 6th Workshop on Computer Science Logic (CSL'92), San Miniato (Italy)*, volume 702 of *Lecture Notes in Computer Science*, pages 115–133. Springer-Verlag, 1992. Selected papers.

[Cre93] N. Creignou. *Temps linéaire et problèmes NP-complets*. PhD thesis, Université de Caen, 1993.

[Dew82] A. K. Dewdney. Linear transformations between combinatorial problems. *International Journal of Computer Mathematics*, 11:91–110, 1982.

[Dom92] E. Domenjoud. Number of minimal unifiers of the equation $\alpha x_1 + \cdots + \alpha x_b =_{AC} \beta y_1 + \cdots + \beta y_q$. *Journal of Automated Reasoning*, 8(1):39–44, 1992.

[FH86] F. Fages and G. Huet. Complete sets of unifiers and matchers in equational theories. *Theoretical Computer Science*, 43(1):189–200, 1986.

[Gal74] Z. Galil. On some direct encodings of nondeterministic Turing machines operating in polynomial time into P-complete problems. *SIGACT News*, 6(1):19–24, January 1974.

[GJ79] M. R. Garey and D. S. Johnson. *Computers and intractability: A guide to the theory of NP-completeness*. W.H. Freeman and Co, 1979.

[Gra93] E. Grandjean. Sorting, linear time and satisfiability problem. In M. Nivat and S. Grigorieff, editors, *Annals of Mathematics and Artificial Intelligence*. J.C. Baltzer Scientific Publishing Co, Suisse, 1993. Special issue. To appear.

[Joh90] D. S. Johnson. A catalog of complexity classes. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science, Volume A: Algorithms and Complexity*, chapter 2, pages 67–161. North-Holland, Amsterdam, 1990.

[KN92] D. Kapur and P. Narendran. Complexity of unification problems with associative-commutative operators. *Journal of Automated Reasoning*, 9:261–288, 1992.

[Koz92] D. C. Kozen. *The design and analysis of algorithms*, chapter 26: Counting problems and #P, pages 138–143. Springer-Verlag, 1992.

[Lin86] N. Linial. Hard enumeration problems in geometry and combinatorics. *SIAM Journal on Algebraic and Discrete Methods*, 7(2):331–335, April 1986.

[PB83] J. S. Provan and M. O. Ball. The complexity of counting cuts and of computing the probability that a graph is connected. *SIAM Journal on Computing*, 12(4):777–788, November 1983.

[PY91] C. Papadimitriou and M. Yannakakis. Optimization, approximation, and complexity classes. *Journal of Computer and System Science*, 43:425–440, 1991.

[Sch78] T. J. Schaefer. The complexity of satisfiability problems. In *Proceedings 10th Symposium on Theory of Computing (STOC'78), San Diego (California, USA)*, pages 216–226, 1978.

[Sim77] J. Simon. On the difference between one and many. In A. Salomaa and M. Steinby, editors, *Proceedings 4th ICALP Conference, Turku (Finland)*, volume 52 of *Lecture Notes in Computer Science*, pages 480–491. Springer-Verlag, July 1977.

[TA87] E. Tidén and S. Arnborg. Unification problems with one-sided distributivity. *Journal of Symbolic Computation*, 3(1 & 2):183–202, 1987.

[Val79a] L. G. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8(2):189–201, 1979.

[Val79b] L. G. Valiant. The complexity of enumeration and reliability problems. *SIAM Journal on Computing*, 8(3):410–421, 1979.