Information Processing Letters 109 (2009) 259–261

Contents lists available at ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

A note on $\#\mathcal{P}$-completeness of NP-witnessing relations

Noam Livne [1]

Department of Computer Science and Applied Mathematics, Weizmann Institute of Science, Rehovot, Israel

ARTICLE INFO

Article history:
Received 30 March 2008
Received in revised form 10 October 2008
Accepted 23 October 2008
Available online 30 October 2008
Communicated by L.A. Hemaspaandra

Keywords:
Computational complexity
$\#\mathcal{P}$-completeness
NP

ABSTRACT

In this note, we study under which conditions various sets (even easy ones) can be associated with a witnessing relation that is $\#\mathcal{P}$ complete. We show a sufficient condition for an $\mathcal{NP}$ set to have such a relation. This condition applies also to many $\mathcal{NP}$-complete sets, as well as to many sets in $\mathcal{P}$.

1. Introduction

By definition, every set in $\mathcal{NP}$ has a polynomial time decidable witnessing relation that defines it (see definitions in Section 1.1). In fact, every $\mathcal{NP}$ set has infinitely many such witnessing relations. Given such a relation, its counting version is the function that assigns to every x the number of y's such that (x, y) is in the relation. The class $\#\mathcal{P}$ consists of all such functions (i.e., arising from all witnessing relations of all $\mathcal{NP}$ sets). A relation is $\#\mathcal{P}$ complete if its counting version is in $\#\mathcal{P}$, and if every function in $\#\mathcal{P}$ can be computed by a polynomial time Turing machine with oracle access to the counting version of that relation. To date, all known $\mathcal{NP}$ complete sets have a defining relation which is $\#\mathcal{P}$ complete (for example, counting the number of satisfying assignments for Boolean formulas, or counting the number of Hamiltonian cycles in a graph, are both $\#\mathcal{P}$ complete). However, an $\mathcal{NP}$ set does not have to be hard in order to have a defining relation which is $\#\mathcal{P}$ complete. One example is the celebrated result that counting the number of perfect matchings in a bipartite graph is $\#\mathcal{P}$-complete [3], whereas deciding whether there exists such a matching is in $\mathcal{P}$.

Another (albeit unnatural) example is the following proof that even extremely easy sets can have a witnessing relation that is $\#\mathcal{P}$-complete. Let R_{SAT} be the natural witnessing relation for SAT (consisting of all pairs of a Boolean formula and an assignment that satisfies it), and consider the relation $R_{\mathrm{SAT}} \cup (\{0,1\}^* \times \{\lambda\})$ (where λ is the empty string). Then, the set defined by this relation is $\{0,1\}^*$, yet this relation is clearly $\#\mathcal{P}$-complete.

In this note, we study under which conditions "easy" $\mathcal{NP}$ sets (for example, sets in $\mathcal{P}$) have a witnessing relation that is $\#\mathcal{P}$ complete. We show a sufficient condition for an $\mathcal{NP}$ set to have such witnessing relation. In particular, the condition holds for every set that is "markable", as defined by Hartmanis and Berman [2] (see definition in Section 2).[2] This condition applies also to certain sets in $\mathcal{P}$.

In the rest of this section we present relevant definitions, and discuss related previous results. In Section 2 we prove our result, which consists of a sufficient condition for the aforementioned question.

1.1. Definitions

For a string x, we denote by $|x|$ the length of x. Given a set of strings S, we denote by $\bar{S}$ the set $\{0,1\}^* \setminus S$. Given

E-mail address: noam.livne@weizmann.ac.il.

[1] Partially supported by the Israel Science Foundation (grant No. 460/05).

[2] This notion was defined by Hartmanis and Berman, but was not given a name in [2].

0020-0190/$ – see front matter
doi:10.1016/j.ipl.2008.10.009

a function f we say that it is *honest* if there exists some polynomial q such that $|x| \leqslant q(|f(x)|)$ for all x. Given a function f we denote by $f|_A$ the restriction of f to the elements of A. When defining strings in the form $(\cdot)$, $(\cdot,\cdot)$, etc., we implicitly assume some 1–1, efficient, efficiently invertible encoding from $\bigcup_{n\in\mathbb{N}}(\{0,1\}^*)^n$ to $\{0,1\}^*$.

1.1.1. $\mathcal{NP}$-witnessing relations and witnesses

By definition, for every set L in $\mathcal{NP}$, there exists an algorithm V such that:

- $x \in L$ if and only if there exists y such that $V(x,y) = 1$.
- There exists a polynomial q such that if $V(x,y) = 1$ then $|y| \leqslant q(|x|)$.
- The running time of V is polynomial in its input.

We call V a *verification algorithm* for L. Note that V well-defines L (i.e., $L = \{x \mid \exists y V(x,y) = 1\}$), thus, we say that *$L$ is the set defined by V*. Such a verification algorithm is not unique. In fact, every set L in $\mathcal{NP}$ has infinitely many verification algorithms. Every such algorithm induces a relation R: the set of pairs that this algorithm accepts. This relation, too, well-defines L (i.e., $L = \{x \mid \exists y (x,y) \in R\}$). We call such a relation an *$\mathcal{NP}$-witnessing relation*, or briefly a *witnessing relation*, and say that *L is the set defined by R*.

Given a witnessing relation R and $(x,y) \in R$ we say that y is a *witness, or a solution, for x with respect to R*.

1.1.2. $\#\mathcal{P}$-completeness of $\mathcal{NP}$-witnessing relations

Given a relation R we define the function $\#R$ by $\#R(x) = |\{y\colon (x,y) \in R\}|$. We call $\#R$ *the counting version of R*. We define $\#\mathcal{P}$ as $\{\#R\colon R \text{ is a witnessing relation}\}$. We say that a relation $R \in \#\mathcal{P}$ is *$\#\mathcal{P}$-complete* if every function in $\#\mathcal{P}$ can be computed by a polynomial time oracle machine with oracle access to $\#R$. (Note that the oracle to $\#R$ is a function oracle.)

1.2. Related work

In a previous work, Fischer et al. [1] studied under what conditions $\mathcal{NP}$-complete sets have a defining relation that is $\#\mathcal{P}$-complete. The following theorem is a direct consequence of Theorem 3.9 in [1]:

Theorem 1.1. *Let f be a Karp-reduction (i.e., polynomial-time many-to-one reduction) of* SAT *to $L \in \mathcal{NP}$, and suppose that f meets the following conditions:*

1. $f|_{\mathrm{SAT}}$ *is* 1–1.
2. $f|_{\mathrm{SAT}}$ *is honest.*
3. *There exists a set $S \in \mathcal{NP}$ such that $L \setminus \mathrm{image}(f) \subseteq S$ and $\mathrm{image}(f) \cap L \subseteq \overline{S}$.*

Then, L has a witnessing relation that is $\#\mathcal{P}$-complete.

Note that L must be an $\mathcal{NP}$-complete set in order to meet the hypothesis of the theorem (i.e., SAT is reduced to L). We mention that SAT is merely a set that has a $\#\mathcal{P}$-complete witnessing relation. Indeed, the use of SAT in the statement of the theorem is arbitrary, and any other set that has a $\#\mathcal{P}$-complete witnessing relation will do.

2. Our result

We prove a sufficient condition for an $\mathcal{NP}$ set to have a $\#\mathcal{P}$-complete witnessing relation. The condition is applicable also to sets that are not $\mathcal{NP}$-complete.

Theorem 2.1. *Let L be some set in $\mathcal{NP}$. Suppose there exists a polynomial time computable function $f:\{0,1\}^* \mapsto \{0,1\}^*$ such that:*

1. $\mathrm{image}(f) \subseteq L$.
2. *f is* 1–1.
3. *f is honest.*
4. *There exists a set $S \in \mathcal{NP}$ such that $L \setminus \mathrm{image}(f) \subseteq S$ and $\mathrm{image}(f) \subseteq \overline{S}$.*

Then, L has a witnessing relation that is $\#\mathcal{P}$-complete.

We stress that, as opposed to Theorem 1.1, f is *not* a reduction of some $\#\mathcal{P}$-complete set to L. Thus, for the conditions to hold, L does not necessarily have to be $\mathcal{NP}$-complete. It is easy to come-up with sets in $\mathcal{P}$ that meet the conditions (see discussion following the proof).

Proof. We construct a new verification algorithm for L, that essentially "embeds" R_{SAT} (the natural witnessing relation for SAT), in the witnessing relation induced by this verification algorithm (while still defining L). This will enable reducing $\#R_{\mathrm{SAT}}$ to the counting version of the induced relation. Since R_{SAT} is $\#\mathcal{P}$-complete, the theorem follows.

Let V_L and V_S be verification algorithms for L and S, respectively. We define the following verification algorithm V' for L: accept w as a witness for x if and only if one of the following conditions hold:

1. $w = (\phi)$ where $f(\phi) = x$.
2. $w = (\phi,\tau)$ where $f(\phi) = x$ and τ is a satisfying assignment for ϕ.
3. $w = (y,z)$ where $V_S(x,y) = 1$ and $V_L(x,z) = 1$.

Let us first show that V' defines L. To see this, note that every instance in L is either in image(f) or in S, and thus will be accepted by conditions 1 or 3 of V', respectively; and every instance not in L is not in image(f) and thus cannot be accepted by conditions 1 and 2 of V', while condition 3 accepts only instances in L.

To complete the proof, we show that $\#R_{\mathrm{SAT}}(\phi) = \#R_{V'}(f(\phi)) - 1$ where $R_{V'}$ is the relation induced by V'. To see this, note that for every unsatisfiable ϕ, the L-instance $f(\phi)$ is accepted by condition 1, and only by it. Since f is 1–1 such $f(\phi)$ will have exactly one witness under $R_{V'}$ (i.e., $w = (\phi)$). For every satisfiable ϕ, every satisfying assignment contributes exactly one witness to the L-instance $f(\phi)$ (by condition 2 of V'), and since f is 1–1, no other formula is mapped to $f(\phi)$, thus there are no other witnesses contributed by condition 2 of V'. The first condition contributes exactly one more witness (again, since f is 1–1). Finally, condition 3 contributes no witness (since $f(\phi)$ is not in S). $\square$

We show, that the sufficient condition is met for every "markable set". First, we define this notion:

Definition 2.2 *(Markable Sets [2]).* A set $L \subseteq \{0,1\}^*$ is *markable* if it is nonempty, and if there exists a marking function $E : \{0,1\}^* \times \{0,1\}^* \mapsto \{0,1\}^*$ and a decoding function $D : \{0,1\}^* \mapsto \{0,1\}^*$ such that:

- E and D are polynomial-time computable.
- For every $p, x \in \{0,1\}^*$ it holds that $E(p,x) \in L$ if and only if $x \in L$.
- For every $p, x \in \{0,1\}^*$ it holds that $D(E(p,x)) = p$.

Corollary 2.3. *Every markable set has a witnessing relation that is $\#\mathcal{P}$-complete.*

Proof. We show that every markable set meets the sufficient condition of Theorem 2.1. Let L be a markable set, and E and D as above. Let a be an arbitrary string in L (L is nonempty by the hypothesis). Then, we define $f(x) = E(x,a)$ and $S = \overline{\text{image}(f)}$.

We show that f and S meet the conditions in the hypothesis of Theorem 2.1. From the second condition of Definition 2.2 it is straightforward that $\text{image}(f) \subseteq L$. We show that the function f is 1–1: Suppose $f(x) = f(x')$. Then $x = D(E(x,a)) = D(f(x)) = D(f(x')) = D(E(x',a)) = x'$. Next, we show that the function f is honest: Let q be a polynomial that bounds the running time of D. Then, since $D(f(x)) = D(E(x,a)) = x$, it follows that $|x| \leqslant q(|f(x)|)$. As for the conditions $L \setminus \text{image}(f) \subseteq S$ and $\text{image}(f) \subseteq \overline{S}$, they follow trivially from the definition of S. Lastly, in order to show that S is in $\mathcal{NP}$ we will show an algorithm that efficiently decides S (thus showing that in fact $S \in \mathcal{P}$): given a string y, the algorithm rejects if $y = f(D(y))$, else it accepts. Now, if $y \in S$ then $y \notin \text{image}(f)$, so it cannot be that $y = f(D(y))$ and the algorithm accepts. On the other hand, if $y \notin S$, then $y \in \text{image}(f)$, so there exists x such that $f(x) = y$, so $f(D(y)) = f(D(f(x))) = f(D(E(x,a))) = f(x) = y$, so the algorithm rejects as required. □

Note that the construction in the proof does not make use of the full computational power of Turing reductions. Rather, the constructed set L has a witnessing relation that is complete for $\#\mathcal{P}$ under Krentel's metrical reductions (i.e., 1-tt-reductions) [4].

Acknowledgements

We thank Lance Fortnow for helpful comments, Oded Goldreich for helpful discussions, and Lane Hemaspaandra and Leen Torenvliet for pointing us to related material.

References

[1] S. Fischer, L. Hemaspaandra, L. Torenvliet, Witness-isomorphic reductions and local search, in: Complexity, Logic, and Recursion Theory, Marcel Dekker, Inc., 1997, pp. 207–223.

[2] J. Hartmanis, L. Berman, On isomorphisms and density of NP and other complete sets, in: STOC'76: Proceedings of the Eighth Annual ACM Symposium on Theory of Computing, ACM, New York, NY, USA, 1976, pp. 30–40.

[3] L.G. Valiant, The complexity of computing the permanent, Theoret. Comput. Sci. 8 (1979) 189–201.

[4] M.W. Krentel, The complexity of optimization problems, J. Comput. System Sci. 36 (3) (1988) 490–509.