Reprinted from JOURNAL OF COMPUTER AND SYSTEM SCIENCES
All Rights Reserved by Academic Press, New York and London

Vol. 53, No. 2, October 1996
*Printed in Belgium*

# Taking It to the Limit: On Infinite Variants of NP-Complete Problems

Tirza Hirst and David Harel*

*Department of Applied Mathematics & Computer Science, The Weizmann Institute of Science, Rehovot 76100, Israel*

Received November 1, 1993

We define infinite, recursive versions of NP optimization problems. For example, MAX CLIQUE becomes the question of whether a recursive graph contains an infinite clique. The present paper was motivated by trying to understand what makes some NP problems highly undecidable in the infinite case, while others remain on low levels of the arithmetical hierarchy. We prove two results; one enables using knowledge about the infinite case to yield implications to the finite case, and the other enables implications in the other direction. Moreover, taken together, the two results provide a method for proving (finitary) problems to be outside the syntactic class MAX NP, and, hence, outside MAX SNP too, by showing that their infinite versions are $\Sigma_1^1$-complete. We illustrate the technique with many examples, resulting in a large number of new $\Sigma_1^1$-complete problems. © 1996 Academic Press, Inc.

## 1. INTRODUCTION

An infinite recursive graph can be thought of simply as a recursive binary relation over some recursive countable set—the natural numbers, for instance. Since recursive graphs can be represented by the Turing machines that recognize their edge sets, one can investigate the complexity of problems concerning them. Indeed, a significant amount of work has been carried out in recent years regarding such problems. Beigel and Gasarch [BG1, BG2] have shown that many problems on recursive graphs reside on low levels of the arithmetical hierarchy. For example, determining whether a recursive graph is 3-colorable is $\Pi_1^0$-complete. On the other hand, in [H2] it was shown that determining whether a recursive graph has a Hamiltonian path is outside the arithmetical hierarchy, and is, in fact, $\Sigma_1^1$-complete.[1] The present work was motivated by trying to understand what makes some NP-complete problems on graphs highly undecidable in the infinite case, while others remain on low levels of the arithmetical hierarchy.

We first set up a general way of obtaining infinite versions of NP maximization and minimization problems. If $P$ is the problem that asks for a maximum by

$$\max_S |\{\bar{w}: A \models \phi(\bar{w}, S)\}|,$$

where $\phi(\bar{w}, S)$ is first-order and $A$ is a finite structure (say, a graph), we define $P^\infty$ as the problem that asks whether there is an $S$, such that the set $\{\bar{w}: A^\infty \models \phi(w, S)\}$ is infinite, where $A^\infty$ is an infinite recursive structure. Thus, for example, MAX CLIQUE becomes the question of whether a recursive graph contains an infinite clique. Often, the infinitary problem becomes trivial; the generalization of MAX SAT, for instance, becomes the following problem: Given an infinite set of clauses $C$, does there exist some assignment $S$ that satisfies an infinite subset of $C$? The answer is always in the affirmative, when we consider only satisfiable clauses.

In this paper we prove two results. One enables using information about the infinite case to yield implications to the finite case, and the other enables implications in the other direction. Moreover, taken together, the two results provide a method for proving (finitary) problems to be outside the syntactic class MAX NP, and, hence, outside MAX SNP too,[2] while at the same time showing infinitary problems to be $\Sigma_1^1$-hard. The classes MAX NP and MAX SNP have been the subject of recent renewed interest, following the developments that show that, whereas all problems in MAX NP are approximable in polynomial time to within *some* constant, the ones that are hard for MAX SNP have no polynomial-time approximation scheme [ALMSS, AS]. These latter sets are closed under special approximation-preserving transformations and are, thus, different from the syntactic sets we refer to here. In fact, our results appear to have no direct relevance to issues of approximability.

Our first result, proved in Section 3, is that the infinite version of any problem in MAX NP is arithmetical; specifically, if $P \in$ MAX NP then $P^\infty \in \Pi_2^0$. Moreover, if $P \in$ MAX

* E-mail: {tirza, harel}@wisdom.weizmann.ac.IL.

[1] A similar result for perfect matching in recursive graphs can be established using a proof technique appearing in [AMS].

[2] These classes, which appear in [PY, PR, KT], are defined in Section 2.



0022-0000/96 $18.00

NP then every instance of $P^\infty$ has either an infinite solution or a maximal finite solution (i.e., no instance of $P^\infty$ has a solution of size $k$, for arbitrarily large $k$, without having an infinite one too). We like to view this fact in its dual formulation: The finitary version of any problem whose infinite version is higher than $\Pi_2^0$ must be outside MAX NP, and, hence, outside MAX SNP too. The same is true for problems that have an instance containing no infinite solution, but containing a solution of size $k$, for arbitrarily large $k$.

For the second result, proved in Section 4, we define a special kind of monotonic transformation between NP optimization problems, which we call an *M-reduction*. The idea is, essentially, that (in a maximization problem) enriching the structure in one problem enriches it in the other, as well as making the objective functions grow. We prove that $M$-reductions between conventional finitary problems become $\Sigma_1^1$-reductions when "lifted up" to the infinite case. This enables one to prove $\Sigma_1^1$-hardness of infinitary problems by examining, and sometimes modifying, reductions between their finitary versions.

Indeed, in Section 5 we use our second result to prove the $\Sigma_1^1$-hardness of many additional problems. Moreover, by our first result, the finitary versions of these must all be outside MAX NP. Here is a partial list of the problems for which these two properties are established: MAX CLIQUE, MAX IND SET, MAX HAM PATH, MAX SUBGRAPH, MAX COMMON SUBSEQUENCE, MAX COLOR, MAX EXACT COVER BY PAIRS, MAX TILING.

## 2. BACKGROUND AND PRELIMINARIES

Our approach to optimization problems is to focus on their descriptive complexity, via logical definability, an idea that started with Fagin's [F] characterization of NP in terms of definability in second-order logic on finite structures. (The following paragraphs are adapted from [KT].)

An *existential second-order formula* is an expression of the form $(\exists S)\,\phi(S)$, where $S$ is a sequence of second-order variables that can contain relations (predicates), and $\phi(S)$ is a first-order formula over some vocabulary $\sigma$. The formula is finitary, so the number of variables in $S$, $\bar{x}$, and $\bar{y}$ is some fixed finite constant. Fagin's theorem [F] asserts that a collection $C$ of finite structures over some vocabulary $\sigma$ is NP-computable if and only if there is a quantifier-free formula $\psi(\bar{x}, \bar{y}, S)$ over $\sigma$, such that for any finite structure $A$ we have

$$A \in C \Leftrightarrow A \models (\exists S)(\forall \bar{x})(\exists \bar{y})\,\psi(\bar{x}, \bar{y}, S).$$

Papadimitriou and Yannakakis [PY] introduced the class MAX NP of maximization problems, whose optimum can be defined by

$$\max_S |\{\bar{x}: A \models (\exists \bar{y})\,\psi(\bar{x}, \bar{y}, S)\}|,$$

for quantifier-free $\psi$. MAX SAT is the canonical example of a problem in MAX NP. They also considered the subclass MAX SNP of MAX NP, consisting of those maximization problems that are defined by quantifier-free formulas, i.e, the optimum of such problems can be defined by

$$\max_S |\{\bar{x}: A \models \psi(\bar{x}, S)\}|$$

for quantifier-free $\psi$. MAX 3SAT is easily seen to be in MAX SNP. Actually, Papadimitriou and Yannakakis [PY] meant that the classes MAX NP and MAX SNP contain also their closures under $L$-reductions, which preserve polynomial-time approximation schems. We do not. Kolaitis and Thakur use the names MAX $\Sigma_0$ and MAX $\Sigma_1$, for MAX SNP and MAX NP, and they too talk about the "pure" syntactic classes. To avoid confusion, we shall follow this terminology in the rest of the paper.

More recently, in a paper by Panconesi and Ranjan [PR], Kozen showed that MAX CLIQUE does not belong to MAX $\Sigma_1$. MAX $\Pi_1$ was introduced in [PR] as the class of maximization problems whose optimum can be defined by

$$\max_S |\{\bar{w}: A \models (\forall \bar{x})\,\psi(\bar{w}, \bar{x}, S)\}|$$

for quantifier-free $\psi$.

Kolaitis and Thakur [KT] then took a broader view. They examined the class of all maximization problems whose optimum is definable using first-order formulas, i.e., by

$$\max_S |\{\bar{w}: A \models \psi(\bar{w}, S)\}|,$$

where $\psi(\bar{w}, S)$ is an arbitrary first-order formula. They first showed that this class coincides with the collection of polynomially bounded NP-maximization problems on finite structures, i.e., those problems whose optimum value is bounded by a polynomial in the input size. They then proceeded to show that these problems form a proper hierarchy, with exactly four levels:

$$\text{MAX } \Sigma_0 \subset \text{MAX } \Sigma_1 \subset \text{MAX } \Pi_1 \subset \text{MAX } \Pi_2 = \bigcup_{i \geq 2} \text{MAX } \Pi_i.$$

Here, MAX $\Pi_1$ is defined just like MAX $\Sigma_1$ (i.e., MAX NP), but with a universal quantifier, and MAX $\Pi_2$ uses a universal followed by an existential quantifier, and corresponds to Fagin's general result stated above. The three containments are strict: It is shown in [KT] that MAX CONNECTED COMPONENT is in MAX $\Pi_2$ but not in MAX $\Pi_1$, while MAX CLIQUE is in MAX $\Pi_1$ but not in MAX $\Sigma_1$ (the latter fact was mentioned above and appears in [PR]), and MAX SAT is in MAX $\Sigma_1$ but not in MAX $\Sigma_0$ (this is from [PY]).

DEFINITION 1 [PR]. An NPO problem is a tuple $F = (\mathcal{I}_F, S_F, m_F, \text{opt})$, where

- $\mathcal{I}_F$ is the space of *input instances*, which are finite structures over some vocabulary $\sigma$ and is recognizable in time that is polynomial in the number of elements of the domain.
- $S_F(I)$ is the space of *feasible solutions* on input $I \in \mathcal{I}_F$. The only requirement on $S_F$ is that there exists a polynomial $q$ and a polynomial time computable predicate $p$, both depending only on $F$, such that $\forall I \in \mathcal{I}_F$, $S_F(I) = \{S: |S| \leqslant q(|I|) \wedge p(I, S)\}$.
- $m_F: \mathcal{I}_F \times \Sigma^* \to \mathbb{N}$, the *objective function*, is a polynomial time computable function. $m_F(I, S)$ is defined only when $S \in S_F(I)$.
- $\text{opt} \in \{\max, \min\}$ indicates whether $F$ is a maximization or minimization problem.
- The following decision problem is in NP: Given $I \in \mathcal{I}_F$ and an integer $k$, is there a feasible solution $S \in S_F(I)$, such that $m_F(I, S) \geqslant k$ when $\text{opt} = \max$ (or $m_F(I, S) \leqslant k$, when $\text{opt} = \min$)?

*Note.* We turn some minimization problems into maximization problems by considering the complements of the solutions. For example, $\overline{\text{MIN VERTEX COVER}}$ will be the problem of finding the maximal set of vertices in a graph such that the complement is a vertex cover. This does not contradict the fact that minimization problems can be very different from maximization ones (see [KT]).

The above definition is broad enough to encompass most known optimization problems arising in the theory of NP-completeness. We now restrict attention to polynomially bounded NP optimization problems [BJY, LM], in which the value of the objective function for every feasible solution is bounded by a polynomial in the length of the corresponding instance.

DEFINITION 2 [PY, PR, KT]. MAX $\Sigma_0$ (MAX $\Sigma_1$, MAX $\Pi_1$, MAX $\Pi_2$, respectively) is the class of NPO problems $F$, such that

$$\text{opt}_F(I) = \max_S |\{\bar{x}: \phi_F(I, S, \bar{x})\}|$$

$$(\text{opt}_F(I) = \max_S |\{\bar{x}: (\exists \bar{y})\, \phi_F(I, S, \bar{x}, \bar{y})\}|,$$

$$\text{opt}_F(I) = \max_S |\{\bar{x}: (\forall \bar{y})\, \phi_F(I, S, \bar{x}, \bar{y})\}|,$$

$$\text{opt}_F(I) = \max_S |\{\bar{x}: (\forall \bar{y})(\exists \bar{z})\, \phi_F(I, S, \bar{x}, \bar{y}, \bar{z})\}|,$$

respectively),

where $\phi_F$ is quantifier-free.

PROPOSITION 1 [KT]. *$F$ is a polynomially bounded* NP *maximization problem iff $F \in$* MAX $\Pi_2$.

DEFINITION 3 [PR]. A problem $F \in \text{RMAX}(k)$ if its optimization function can be expressed as

$$\text{opt}_F(I) = \max_S \{|S|: (\forall \bar{y})\, \phi(I, S, \bar{y})\},$$

where $\phi$ is a quantifier-free CNF formula with all the occurrences of $S$ in $\phi$ being negative, $S$ is a single predicate appearing at most $k$ times in each clause, and $|S|$ denotes $|\{\bar{x}: S(\bar{x})\}|$.

DEFINITION 4. Let $A_l = (D_1, R_1^1, ..., R_m^1)$, $A_2 = (D_2, R_1^2, ..., R_m^2)$ be (possibly infinite) structures over the same similarity type: $\{P_1, ..., P_m\}$. We say that $A_1$ is a *substructure* of $A_2$, denoted $A_1 \leqslant A_2$, if $D_1 \subseteq D_2$ and $R_i^1$, $1 \leqslant i \leqslant m$, is the restriction of $R_i^2$ to $D_1$. (If the elements of the domain are ordered, then $D_1$ has to be a prefix of $D_2$.)

We now restrict the class of NPO problems somewhat.

DEFINITION 5. NPM is the class of NPO problems $F = (\mathcal{I}_F, S_F, m_F, \text{opt})$, for which $\text{opt} = \max$, $\mathcal{I}_F$ contains finite structures over a vocabulary $\sigma$, and the objective function is given by

$$(\forall I \in \mathcal{I}_F)(\forall S \in S_F(I)) \quad (m_F(I, S) = |\{\bar{x}: \psi_F(I, S, \bar{x})\}|),$$

where $\psi_F$ is a $\Pi_2$-formula. We require that $\psi_F$ satisfies the following additional condition: If for some infinite structure $I^\infty$ and for some $S$ and $\bar{x}$, $\psi_F(I^\infty, S, \bar{x})$ is true, then there exists a finite substructure $I$ of $I^\infty$ containing $\bar{x}$, such that for each $I \leqslant I' \leqslant I^\infty$, $\psi_F(I', S', \bar{x})$ is also true, where $S'$ is the restriction of $S$ to the domain of $I'$.

Note that the addition does not sacrifice generality in the case of $\Sigma_0$, $\Sigma_1$, and $\Pi_1$, since such formulas satisfy the condition anyway.

We, now "lift up" NP maximization problems, resulting in versions that apply to infinite recursive structures. We do this simply by requiring an infinite solution instead of a maximal one. Minimization problems can be similarly generalized; by requiring that the complement of the solution should be infinite.

DEFINITION 6. Let $F = (\mathcal{I}_F, S_F, m_F)$ be an NPM problem. Define $F^\infty$, the *infinitary version of* $F$, as follows: $F^\infty = (\mathcal{I}_F^\infty, S_F^\infty, m_F^\infty)$, where

- $\mathcal{I}_F^\infty$ is the space of *input instances*, which are infinite recursive structures over the vocabulary $\sigma$.
- $S_F^\infty(I^\infty)$ is the space of *feasible solutions* on input $I^\infty \in \mathcal{I}_F^\infty$.

• $m_F^\infty : \mathscr{I}_F^\infty \times S_F^\infty \to \mathbb{N} \cup \{\infty\}$ is the *objective function* and satisfies

$$(\forall I^\infty \in \mathscr{I}_F^\infty)(\forall S \in S_F^\infty(I^\infty)) \quad (m_F^\infty(I^\infty, S) = |\{\bar{x}: \psi_F(I^\infty, S, \bar{x})\}|)$$

where $\psi_F$ is the $\Pi_2$-formula of $F$.

• The decision problem is: Given $I^\infty \in \mathscr{I}_F^\infty$, does there exist $S \in S_F^\infty(I^\infty)$ such that $m_F^\infty(I^\infty, S) = \infty$? Put another way,

$$F^\infty(I^\infty) = \text{TRUE} \qquad \text{iff} \quad \exists S(|\{\bar{x}: \psi_F(I^\infty, S, \bar{x})\} = \infty).$$

Due to the condition in Definition 5 of an NPM, $F^\infty$ does not depend on the $\Pi_2$-formula representing $m_F$. Otherwise, if some finite problem $F$ could be defined by two different formulas $\psi_1$ and $\psi_2$ satisfying the condition, which yield different infinite problems, we could construct a finite structure for which $\psi_1$ and $\psi_2$ determine different solutions.

## 3. FROM THE INFINITE TO THE FINITE

PROPOSITION 2. *If $F \in$ NPM then $F^\infty \in \Sigma_1^1$.*

*Proof.* Let $F = (\mathscr{I}_F, S_F, m_F) \in$ NPM. We have to express $F^\infty$ by an existential second-order formula over some recursive predicate. (The formula need not necessarily be over $F$'s vocabulary.)

$F^\infty$ can be described as

$$(\exists S)(\forall \bar{x}_1)(\exists \bar{x}_2) \qquad (\bar{x}_1 < \bar{x}_2 \wedge \psi_F(I^\infty, S, \bar{x}_2)),$$

where $\psi_F(I, S, \bar{x})$ is the first-order formula appearing in the definition of $F$. The relation $<$ is not part of the vocabulary; rather, it is the lexicographic extension of some ordering that can be computed from a Turing Machine that recognizes the domain of $I^\infty$. (There is such a Turing Machine since $I^\infty$ is recursive.) ▮

The following lemma is needed for the proof of Theorem 1. It states that in order to decide whether an instance $I^\infty$ of a problem in MAX $\Sigma_1$ contains an infinite solution, it suffices to check whether there are infinitely many $\bar{x}$'s for which there is an $S$ etc., instead of checking if there exists an $S$ for which there are infinitely many $\bar{x}$'s. For example, consider MAX SAT$^\infty$, which is the problem of determining that there is an assignment that satisfies infinitely many of the clauses in an infinite recursive set of clauses. MAX SAT$^\infty(P,N) = \text{TRUE}$ iff $(\exists S)|\{c: (\exists y)(P(y,c) \wedge S(y)) \vee (N(y,c) \wedge \neg S(y))\}| = \infty$, where $S$ is a second-order variable that ranges over truth assignments, and $P$ and $N$ are two recursive binary predicates; $P(y,c) = \text{TRUE}$ iff variable $y$ appears unnegated in clause $c$, and $N(y,c) = \text{TRUE}$ iff variable $y$ appears negated in clause $c$. The lemma says that it suffices to check whether there are infinitely many clauses that are satisfiable, by possibly different assignments. Here, of course, the answer is always true.

LEMMA 1. *Let $F \in$ MAX $\Sigma_1$, where $\text{opt}_F(I) = \max_S |\{\bar{x}: (\exists \bar{y})\, \phi(\bar{x}, \bar{y}, I, S)\}|$, for quantifier free $\phi$. Then, for each instance $I^\infty$,*

$$F^\infty(I^\infty) = \text{TRUE} \quad \text{iff} \quad |\{\bar{x}: (\exists S)(\exists \bar{y})\, \phi(\bar{x}, \bar{y}, I^\infty, S)\}| = \infty.$$

*Proof.* Let $I^\infty$ be a recursive infinite structure, which is an instance of $F^\infty$. According to Definition 6,

$$F^\infty(I^\infty) = \text{TRUE} \quad \text{iff} \quad (\exists S)(|\{\bar{x}: (\exists \bar{y})\, \phi(\bar{x}, \bar{y}, I^\infty, S)\}| = \infty).$$

The "only-if" direction is clear, since we are simply pushing the existential quantifier into the set.

For the "if" direction, we have to show that if there are infinitely many $\bar{x}$'s for which there is an $S$, etc., then there is a single $S$ for which there are infinitely many $\bar{x}$'s, etc. Let us consider the sequence consisting of the following formulas in some order: $\phi(\bar{x}_i, \bar{y}_j, I^\infty, S)$ for $i, j \geqslant 1$, where $\{\bar{x}_1, \bar{x}_2, ...\}$ and $\{\bar{y}_1, \bar{y}_2, ...\}$ are all the feasible values of $\bar{x}$ and $\bar{y}$. We may view $\phi$ as a Boolean formula with "variables" of the form $S(\bar{z})$, where $\bar{z}$ is a projection of $\bar{x}$ and $\bar{y}$. The formula also contains terms of the form $I^\infty(\bar{w})$ (with $(\bar{w})$ a similar projection), but these are fixed, since $I^\infty$ is given. By the assumption, there are infinitely many $\bar{x}$'s that have an $S$ and $\bar{y}$ satisfying $\phi$. We may thus take a sequence of $\phi$'s with corresponding $\bar{y}$'s—one for each $\bar{x}$—that have satisfying $S$'s. Let us denote them by $\phi_1, \phi_2, \phi_3, ...$, with $S_i$ satisfying $\phi_i$.

We will use $k$ to denote the (constant) number of variables of the form $S(\bar{z})$ in $\phi$. We proceed by induction on $k$. If $k = 0$, then $\phi$ has no variables, and each $\phi_i$ is satisfiable by $S_i = \varnothing$. Hence, we have our infinitely many $\bar{x}$'s. (Actually, $\bigwedge_{i=1}^\infty \phi_i$ is a tautology.)

Assume that whenever $\phi$ has $k-1$ variables there is an $S$ that satisfies infinitely many $\phi_i$'s, and let our $\phi$ contain $k$ variables. If there exists some variable $S(\bar{z})$ that appears in infinitely many $\phi_i$'s, then it appears positively (or negatively, respectively) in the satisfying assignment of an infinite subset of the $\phi_i$'s. Assign $S(\bar{z})$ true (or false, respectively) and assign truth values to the other $k-1$ variables by the inductive hypothesis. In this case we are done. If each variable appears in only finitely many $\phi_i$'s, we proceed as follows. First, assign the values of $S_1$ (the satisfying assignment of $\phi_1$) to $\phi_1$'s variables. Next, repeatedly choose a new $\phi_i$ containing only new variables, and satisfy it by using the values of $S_i$. Continuing this process yields an infinite set from among the $\phi_i$'s, that are all satisfied by the single assignment $S$ obtained by collecting the values used at each stage. ∎

THEOREM 1. *If* $F \in$ MAX $\Sigma_1$ *then*

1. $F^\infty \in \Pi_2^0$.
2. *For each recursive structure* $I^\infty \in \mathscr{I}_F^\infty$,

$$F^\infty(I^\infty) = \text{TRUE} \qquad \textit{iff} \quad (\forall n)(\exists S)(m_F^\infty(I^\infty, S) \geq n).$$

*Proof.* Let $F$ be a problem in MAX $\Sigma_1$, such that

$$\text{opt}_F(I) = \max_S |\{\bar{x}: (\exists \bar{y})\, \phi(\bar{x}, \bar{y}, I, S)\}|,$$

for quantifier-free $\phi$. According to Lemma 1, for each $I^\infty$,

$$F^\infty(I^\infty) = \text{TRUE} \qquad \text{iff} \quad \{\bar{x}: (\exists S)(\exists \bar{y})\, \phi(\bar{x}, \bar{y}, I^\infty, S)\}| = \infty.$$

It follows that we can express the problem $F^\infty$ as

$$(\forall \bar{x}_1)(\exists \bar{x}_2)(\exists \bar{y}) \quad (\bar{x}_1 < \bar{x}_2 \wedge \phi(\bar{x}_2, \bar{y}, I^\infty, S) \text{ is satisfiable}).$$

Now, since there is some recursive order on the domain (because $I^\infty$ is recursive) and since checking satisfiability of $\phi$ is recursive (because $\phi$ is a Boolean formula with only $k$ variables), $F^\infty$ is in $\Pi_2^0$. This completes the proof of the first clause of the theorem. As to the second, the "only if" direction is clear. The "if" direction follows from Lemma 1, since if $(\forall n)(\exists S)\, |\{\bar{x}: (\exists \bar{y})\, \phi(\bar{x}, \bar{y}, I^\infty, S)\}| \geq n$, then there are clearly infinitely many $\bar{x}$'s for which there is an $S$, etc. Hence, $F^\infty(I^\infty) = \text{TRUE}$. ∎

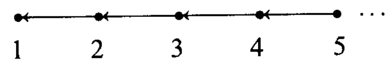
We like to view the theorem in its dual formulation, whereby information about an infinitary problem bears upon the status of its finitary version:

COROLLARY 1. *For any* NPM *problem* $F$, *if* $F^\infty$ *is* $\Sigma_1^1$*-complete then* $F$ *is not in* MAX $\Sigma_1$.

It follows that Hamiltonicity, which is $\Sigma_1^1$-complete [H2], is not in MAX $\Sigma_1$ (and the same applies to perfect matching, following [AMS]). We shall see many more such problems in Section 5. Obviously, the corollary is valid not only for $\Sigma_1^1$-complete problems, but for all problems that are outside $\Pi_2^0$. For example, detecting the existence of an Eulerian path in a recursive graph is $\Pi_3^0$-complete [BG2], hence its finite variant cannot be in MAX $\Sigma_1$.

COROLLARY 2. *For any* NPM *problem* $F$, *if there is a recursive structure* $I^\infty \in \mathscr{I}_F^\infty$ *for which* $(\forall n)(\exists S)$ $(m_F^\infty(I^\infty, S) \geq n)$ *but* $\neg(\exists S)(m_F^\infty(I^\infty, S) = \infty)$ *then* $F$ *is not in* MAX $\Sigma_1$.

We can now easily show many problems to be outside MAX $\Sigma_1$. For example, for MAX CLIQUE and for MAX CONNECTED COMPONENT (MCC), consider the recursive graph containing isolated cliques of size $n$, for each $n$. Also, for Hamiltonicity and for Eulerian paths, consider the following graph:




By Corollary 2, these are outside MAX $\Sigma_1$.

We note that the problems that appeared in [PY] as examples for MAX $\Sigma_0$ and MAX $\Sigma_1$, such as INDEPENDENT SET-B and MAX SAT (for which the given clauses are satisfiable), become trivial in the infinite case: There is always an infinite solution. The reason is that in these problems the appropriate $\phi$'s are always satisfiable, and, hence, one can always find an assignment that satisfies infinitely many $\phi$'s. However, there are problems in these classes whose infinite variants are nontrivial. Here is an example:

MAX IND SET-B-2. Given a graph with degree bounded by $B$ and $m < B$, find the largest independent set of nodes with degree $\leq m$. (This is similar to INDEPENDENT SET-B of [PY].) To provide an economic logical definition of this problem, we represent a graph of degree $B$ by a $(B+1)$-ary relation $A$, encoding the adjacency lists of the $n$ nodes (which we may assume to be $\{1, 2, ..., n\}$). For each node $u$, the tuple $(u, v_l, ..., v_B)$ lists its neighbors $v_l, ..., v_B$, and if $u$ has less than $B$ neighbors, the remaining places will contain zeros. The problem can now be expressed as

$$\max_S |\{(u, v_1, ..., v_B) \in A: u \in S \wedge v_1, ..., v_B \notin S \wedge v_{m+1} = 0\}|.$$

In contrast to INDEPENDENT SET-B$^\infty$, the problem MAX IND SET-B-2$^\infty$ is nontrivial. Some bounded graphs have an infinite independent set of vertices with degree smaller than $m$, but others do not.

COROLLARY 3. *Let* $F \in$ MAX $\Sigma_0$ *(respectively,* $F \in$ MAX $\Sigma_1$*). If for each* $I \in \mathscr{I}_F$*, and for each* $\bar{x}$*, the formula* $\phi(\bar{x}, S, I)$ *(respectively,* $(\exists \bar{y})\, \phi(\bar{x}, \bar{y}, S, I)$*) is satisfiable, then* $F^\infty$ *always has a positive answer.*

This corollary is not necessarily valid for MAX $\Pi_1$ and MAX $\Pi_2$. Consider, for example, MAX CLIQUE. Every vertex can be contained in some clique, but there need not be an infinite clique. Also, in MCC, each vertex of the input graph is contained in some component, but there need not be an infinite one.

## 4. FROM THE FINITE TO THE INFINITE

In this section, we define a special kind of *monotonic* reduction between finitary NPM problems, which we call an *M-reduction.* We then show that $M$-reductions preserve the $\Sigma_1^1$-hardness of infinitary variants. It is worth noting that about half of the reductions needed for the results in

Section 5 are taken from [GJ], and are already monotonic there. These are usually the simpler ones. Among the others, some are monotonic modifications of reductions from [GJ], but others required more work on our part. We also have a few monotonic reductions from polynomial-time problems, for which [GJ] is irrelevant.

DEFINITION 7. Let $\mathscr{A}$ and $\mathscr{B}$ be sets of structures. A function $f: \mathscr{A} \to \mathscr{B}$ is *monotonic* if $\forall A, B \in \mathscr{A}$ $(A \leqslant B \Rightarrow f(A) \leqslant f(B))$. (Here, $\leqslant$ denotes the substructure relationship of Definition 4.)

DEFINITION 8. Given two NPM problems: $F = (\mathscr{I}_F, S_F, m_F)$, $G = (\mathscr{I}_G, S_G, m_G)$. An *M-reduction* $g$ from $F$ to $G$ (denoted $F \propto_M G$) is a tuple $g = (t_1, t_2, t_3)$:

1. $t_1, t_2, t_3$ are polynomial time computable functions.
2. $t_1: \mathscr{I}_F \to \mathscr{I}_G$, $t_2: \mathscr{I}_F \times S_F \to S_G$, and $t_3: \mathscr{I}_G \times S_G \Rightarrow S_F$.[3]
3. $t_1$ is monotonic, in the sense of Definition 7.
4. $t_2$ and $t_3$ are partially monotonic; i.e., $\forall I_1, I_2 \in \mathscr{I}_F$

$$(I_1 < I_2) \Rightarrow (\forall S \in S_F(I_2)\ \ t_2(I_1, S') \leqslant t_2(I_2, S)$$
$$\wedge\ \forall S \in S_G(t_1(I_2))\ \ t_3(t_1(I_1), S') \leqslant t_3(t_1(I_2), S)),$$

where $S'$ is the restriction of $S$ to the domain of $I_1$ (or resp. $t_1(I_1)$).

5. Let $\{I_i \in \mathscr{I}_F\}_{i=1}^{\infty}$, $\{S_i \in S_F(I_i)\}_{i=1}^{\infty}$, $\{X_i \subseteq \{\bar{x}: \psi_F(I_i, S_i, \bar{x})\}\}_{i=1}^{\infty}$, such that $I_1 \leqslant I_2 \leqslant \cdots$ and $S_1 \leqslant S_2 \leqslant \cdots$. If $X_1 \subset X_2 \subset X_3 \cdots$ then there exist $\{Y_j \subseteq \{\bar{x}: \psi_G(t_1(I_{i_j}), t_2(S_{i_j}), \bar{x})\}\}_{j=1}^{\infty}$, such that $\forall j\ i_j > i_{j+1}$ and $Y_1 \subset Y_2 \subset Y_3 \cdots$ (the containments, denoted by $\subset$, are strict.) The same is required for the other direction.

*Note.* All the problems appearing in Section 5 satisfy

$$\forall I_1, I_2 \in \mathscr{I}_F \quad \forall S_1 \in S_F(I_1) \quad \forall S_2 \in S_F$$
$$(I_1 \leqslant I_2 \wedge S_1 \leqslant S_2)$$
$$\Rightarrow \{\bar{x}: \psi(I_1, S_1, \bar{x})\} \subseteq \{\bar{x}: \psi(I_2, S_2, \bar{x})\}.$$

Moreover, the reductions we show there satisfy

$$\forall I_1 \in \mathscr{I}_F \quad \forall S_1 \in S_F(I_1) \quad \exists k \quad \forall I_2 \in \mathscr{I}_F \quad \forall S_2 \in S_F(I_2)$$
$$((I_1 \leqslant I_2 \wedge S_1 \leqslant S_2 \wedge (m_F(I_1, S_1) + k) < m_F(I_2, S_2))$$
$$\Rightarrow (m_G(t_1(I_1), t_2(S_1)) < m_G(t_1(I_2), t_2(S_2))))$$

and similarly for the second direction. Hence, constraint 5 is satisfied for these problems.

[3] Although $t_2$ and $t_3$ are two-place functions, we shall sometimes omit their first argument, which will be clear from the context (as in, e.g., clause 5 below).

For example, let us show that MAX CLIQUE $\propto_M$ MAX IND SET. The instances (i.e., the $I$'s) for these two problems are graphs, the feasible solutions are all the cliques or independent sets in the graph, and the objective function yields the size of the solution.

Let $G = (V, E)$ be an instance of MAX CLIQUE. Define $g = (t_1, t_2, t_3)$ to be:

- $t_1(G) = \bar{G} = (V, \bar{E})$.
- If $Q \in S(G)$, then let $t_2(G, Q) = Q \in S(\bar{G})$.
- If $Q \in S(\bar{G})$, then let $t_3(\bar{G}, Q) = Q \in S(G)$.
- $t_1$ is monotonic. Let $G_1 = (V_1, E_1)$, $G_2 = (V_2, E_2)$, where $V_1 = \{v_1, ..., v_k\}$, $V_2 = \{v_1, ..., v_k, ..., v_n\}$, and such that $G_1 < G_2$. According to the definition of a substructure, the restriction of $G_2$ to $\{v_1, ..., v_k\}$ is isomorphic to $G_1$. Thus, if $v, u \in V_1$, we have

$$(v, u) \notin t_1(E_2) \Leftrightarrow (v, u) \in E_2 \Leftrightarrow (v, u) \in E_1 \Leftrightarrow (v, u) \notin t_1(E_1),$$

and, hence, $t_1(G_1) \leqslant t_1(G_2)$.

Clearly, the other requirements are satisfied too.

THEOREM 2. *Let $F$ and $G$ be two* NPM *problems, with $F \propto_M G$. If $F^{\infty}$ is $\Sigma_1^1$-hard, then $G^{\infty}$ is $\Sigma_1^1$-hard too.*

*Proof.* Let $F, G \in$ NPM, where $F^{\infty} = (\mathscr{I}_F^{\infty}, S_F^{\infty}, m_F^{\infty})$ is $\Sigma_1^1$-hard and let $g = (t_1, t_2, t_3)$ be an $M$-reduction from $F$ to $G$. In order to prove $\Sigma_1^1$-hardness of $G^{\infty}$, we will exhibit a recursive reduction $f$ from $F^{\infty}$ to $G^{\infty}$, such that any $I^{\infty} \in \mathscr{I}_F^{\infty}$ will have an infinite solution iff $f(I^{\infty}) \in \mathscr{I}_G^{\infty}$ has one.

Let $I^{\infty} \in \mathscr{I}_F^{\infty}$. Assume, without loss of generality, that $I^{\infty}$ is a structure over the domain $\mathbb{N}$. For each $i$, let $I_i$ be the restriction of $I^{\infty}$ to $\{1, 2, ..., i\}$. By Definition 4 we have $I_1 \leqslant I_2 \leqslant \cdots \leqslant I^{\infty}$. Since $t_1$ is monotonic,

$$t_1(I_1) \leqslant t_1(I_2) \leqslant \cdots.$$

Define $f(I^{\infty}) = \bigcup_{i=1}^{\infty} t_1(I_i)$. (This is well defined even if there is no order on the domain, because of the monotonicity of $t_1$.) $f(I^{\infty})$ is recursive since in order to check if a tuple $u$ is in some relation $R \in f(I^{\infty})$ it suffices to check if $u$ is in the appropriate relation in $t_1(I_i)$, for large enough $i$, that contains the elements in $u$.

We now show that $I^{\infty}$ has an infinite solution iff $f(I^{\infty})$ has one. Assume that $S \in S_F^{\infty}(I^{\infty})$, and

$$m_F^{\infty}(I^{\infty}, S) = |\{\bar{x}: \psi_F(I^{\infty}, S, \bar{x})\}| = |\{\bar{x}_1, \bar{x}_2, ...\}| = \infty.$$

Due to the constraints of NPM, we can construct a subsequence $\{A_j\}_{j=1}^{\infty}$ of $\{I_i\}_{i=1}^{\infty}$ such that for all $j$, $A_j$ contains $\{\bar{x}_1, ..., \bar{x}_j\}$, and $\psi_F(A_j, S_j, \bar{x})$ is true for each $\bar{x} \in \{\bar{x}_1, ..., \bar{x}_j\}$, where $S_j$ is the restriction of $S$ to the domain of $A_j$.

By the monotonicity of $g$, for each $j \geqslant 1$ we have

$$t_2(A_j, S_j) \leqslant t_2(A_{j+1}, S_{j+1}).$$

Define $\hat{S} = \bigcup_{j=1}^{\infty} t_2(A_j, S_j)$. Consider now the sets:

$$\{B_j = \{\bar{y}: \psi_G(t_1(A_j), t_2(A_j, S_j), \bar{y})\}\}_{j=1}^{\infty}.$$

Due to Condition 5 of the monotonicity of $g$, there are infinitely many $\bar{y}$'s that are contained in infinitely many $B_j$'s. Since $\psi_G$ is a $\Pi_2$-formula, these $\bar{y}$'s must satisfy $\psi_G(f(I^\infty), \hat{S}, \bar{y})$. Hence, $m_G^\infty(f(I^\infty), \hat{S}) = \infty$.

The other direction is similar, but uses $t_3$ instead of $t_2$. ∎

We already know that there are problems in MAX $\Pi_1$ whose infinite versions are $\Sigma_1^1$-complete, e.g., Hamiltonicity [H2]. We can also show that there are problems in its *sub*class RMAX(2) that have the same property, e.g., MAX CLIQUE.

COROLLARY 4. *Let $F \in$ NPM. If $F$ is hard for* MAX $\Pi_1$ *(or even for* RMAX(2)*) with respect to M-reductions, then $F^\infty$ is $\Sigma_1^1$-complete.*

For example, Panconesi and Ranjan [PR], proved the MAX $\Pi_1$ completeness of MAX NSF[4] with respect to approximation-preserving reductions. Since the reduction they used is also an $M$-reduction, MAX NSF$^\infty$ is $\Sigma_1^1$-complete.

## 5. APPLICATIONS

We start by listing several problems in NPM:

1. MAX PATH IN TREES. $I$ is a tree $T = (N, P, 0)$, where 0 is the root. $N = \{0, 00, 01, 000, 001, 010, 011, 02, ..., d\}$.

$$S(T) = \{\bar{p}: \bar{p} \text{ is a path in } T\}$$

$$m_1(T, \bar{p}) = |\bar{p}|$$

$$m_2(T, \bar{p}) = \begin{cases} |\bar{p}| & \text{if } p_1 = 0 \\ 0 & \text{if } p_1 \neq 0 \end{cases}$$

$$\max_{\bar{p}} |\{l: 1 \leqslant l \leqslant |\bar{p}|, \forall i, p_i \in N, p_1 = 0, \forall i, \quad 1 \leqslant i < |\bar{p}|, (p_i, p_{i+1}) \in P\}|.$$

MAX PATH IN TREES$^\infty$. $I^\infty$ is a recursive tree $T$.

Q. Does $T$ contain an infinite path?

This is the non-well-foundedness of recursive trees with possibly infinite out-degree—the quintessential $\Sigma_1^1$-complete problem [R].

2. MAX CLIQUE. $I$ is an undirected graph, $G = (V, E)$.

$$S(G) = \{Y: Y \subseteq V, \forall y, z \in Y\,(y \neq z \Rightarrow (y, z) \in E)\}$$

$$m(G, Y) = |Y|.$$

[4] MAX NSF is the problem of finding the maximum number of satisfiable formulas in a given set of CNF formulas.

The maximization version is

$$\max_{Y \subseteq V} |\{x: x \in Y \wedge \forall y, z \in Y\,(y \neq z \Rightarrow (y, z) \in E)\}|.$$

MAX CLIQUE$^\infty$. $I^\infty$ is a recursive graph $G$.

Q. Does $G$ contain an infinite clique?

3. MAX IND SET. $I$ is an undirected graph $G = (V, E)$.

$$S(G) = \{Y: Y \subseteq V, \forall y, z \in Y\,(y, z) \notin E\}$$

$$m(G, Y) = |Y|$$

$$\max_{Y \subseteq V} |\{x: x \in Y \wedge \forall y, z \in Y\,(y, z) \notin E\}|.$$

MAX IND SET$^\infty$. $I^\infty$ is a recursive graph $G$.

Q. Does $G$ contain an infinite independent set?

4. MIN VERTEX COVER. $I$ is a graph $G = (V, E)$.

$$S(G) = \{Y \subseteq V: \forall (u, v) \in E\,(u \in V - Y \text{ or } v \in V - Y)\}$$
$$= \{Y \subseteq V: \forall u, v \in Y\,(u, v) \notin E\}$$

$$m(G, Y) = |Y|.$$

(This problem is identical to MAX IND SET.)

$\overline{\text{MIN VERTEX COVER}}^\infty$. $I^\infty$ is a recursive graph $G$.

Q. Is there a vertex-cover of $G$ whose complement is infinite?

5. MAX SET PACKING. $I$ is a collection $C$ of finite sets, represented by pairs $(i, j)$, where the set $i$ contains $j$.

$$S(C) = \{Y \subseteq C: \forall A, B \in Y\,(A \neq B \Rightarrow A \cap B = \emptyset)\}$$

$$m(C, Y) = |Y|.$$

MAX SET PACKING$^\infty$. $I^\infty$ is a recursive collection of infinite sets $C$.

Q. Does $C$ contains infinitely many disjoint sets?

6. $\overline{\text{MIN SET COVER}}$. $I$ is a set $A = \{a_1, ..., a_n\}$ and a set $C$ of subsets of $A$.

$$S(A, C) = \{Y \subseteq C: \forall a \in A\, \exists S \in C - Y \text{ such that } a \in S\}$$

$$m((A, C), Y) = |Y|.$$

$\overline{\text{MIN SET COVER}}^\infty$. $I^\infty$ is a recursive set $A$, and a recursive collection $C$ of subsets of $A$.

Q. Is there a set-covering of $A$ from $C$, whose complement is infinite?

7. MAX SUBGRAPH. $I$ is a pair of graphs, $G = (V_1, E_1)$ and $H = (V_2, E_2)$, with $V_2 = \{v_1, ..., v_n\}$.

$$S(G, H) = \{Y: Y \subseteq V_1 \times V_2, \forall (u, v), (x, y) \in Y (u \neq x \wedge v \neq y \wedge (u, x) \in E_1 \Leftrightarrow (v, y) \in E_2)\}$$

$m((G, H), Y) = k$ iff $v_1, ..., v_k$ appear in $Y$, but $v_{k+1}$ does not appear in $Y$.

MAX SUBGRAPH$^\infty$. $I^\infty$ is a pair of recursive graphs, $H$ and $G$.

Q. Is $H$ a subgraph of $G$?

The finite problem is defined so as to yield the desired infinite one. Note that if we were to define $m((G, H), Y)$ simply to be $|Y|$, then MAX SUBGRAPH$^\infty$ would become the problem of finding a common infinite subgraph of $H$ and $G$.

8. MAX COLOR. $I$ is a graph $G = (V, E)$ with $V = \{v_1, ..., v_n\}$, and a set $C = \{c_0, ..., c_m\}$ of colors.

$$S(G, C) = \{\bar{y}: \forall i, 1 \leqslant i \leqslant |\bar{y}|, y_i \in C \wedge \forall i, j \leqslant |\bar{y}| ((v_i, v_j) \in E \Rightarrow y_i \neq y_j)\}$$

$m((G, C), \bar{y}) = k$ iff $c_0$ appears $k$ times in $\bar{y}$.

$$\max_{\bar{y} \subseteq C} |\{k: y_k = c_0 \wedge \forall i, j \leqslant |\bar{y}| ((v_i, v_j) \in E \Rightarrow y_i \neq y_j)\}|.$$

MAX COLOR$^\infty$. $I^\infty$ is a recursive graph $G$, and a recursive set of colors.

Q. Is there a coloring of $G$ in which the first color, $c_0$, appears infinitely often?

9. LARGEST COMMON SUBSEQUENCE (LCS). $I$ is a finite alphabet $\Sigma$ and a finite set $R$ of strings from $\Sigma^*$:

$$S(\Sigma, R) = \{x: \forall w \in R, x \text{ is a subsequence of } w\}$$

$$m((\Sigma, R), x) = |x|.$$

(Note. *ab* is a subsequence of *cacbc*.)

LCS$^\infty$. $I^\infty$ is an infinite alphabet $\Sigma$ and an infinite recursive set $R$ of infinite strings over $\Sigma^\omega$. (Instances can be represented by triples $(i, j, a)$, where the $j$th character of the $i$th string is $a$. This case is a natural generalization of triple representation in the finitary case.)

Q. Is there a common infinite subsequence of all the strings in $R$?

10. MAX EXACT COVER BY PAIRS (MAX 2XC; this is essentially PERFECT MATCHING). $I$ is a set $X = \{x_1, ..., x_{2q}\}$, and a collection $C$ of unordered pairs from $X$.

$$S(X, C) = \{Y: Y \subseteq C, \forall c, d \in Y (c \cap d = \varnothing)\}$$

$m((X, C), Y) = k$ iff $x_1, ..., x_k$ appear (in some pairs) in $Y$, but $x_{k+1}$ does not appear in $Y$.

MAX 2XC$^\infty$. $I^\infty$ is an infinite set $X$, and an infinite recursive collection $C$ of pairs from $X$.

Q. Is there an infinite subset $C'$ of $C$ that is an exact cover of $X$?

11. MAX HAM PATH. $I$ is a directed graph $G = (V, E)$, $V = \{v_1, ..., v_n\}$.

$$S(G) = \{\bar{y}: \forall i, 1 \leqslant i < |\bar{y}|, y_i \in V \wedge (y_i, y_{i+1}) \in E \wedge \forall i, j, 1 \leqslant i, j \leqslant |\bar{y}|, i \neq j \Rightarrow y_i \neq y_j\}$$

$$m(G, \bar{y}) = k \quad \text{iff} \quad v_1, v_2, ..., v_k \in \bar{y} \quad \text{and} \quad v_{k+1} \notin \bar{y}.$$

$$\max_{\bar{y} \subseteq V} |\{k: \forall i, 1 \leqslant i \leqslant k, v_i \in \bar{y} \wedge \forall i, 1 \leqslant i < |\bar{y}|, (y_i, y_{i+1}) \in E \wedge \forall i, j, 1 \leqslant i, j \leqslant |\bar{y}|, \quad (i \neq j \Rightarrow y_i \neq y_j)\}|.$$

MAX HAM PATH$^\infty$. $I^\infty$ is a recursive graph $G$.

Q. Does $G$ contain a Hamiltonian path?

MAX PLANAR HAM PATH and MAX UNDIRECTED PLANAR HAM PATH are the problems of detecting the existence of a Hamiltonian path in directed and undirected planar graphs, respectively.

12. MAX SAT2. $I$ is a set of variables $U = \{\sigma_1, \neg\sigma_1, \sigma_2, \neg\sigma_2, ..., \sigma_m, \neg\sigma_m\}$ and a collection of clauses represented by triples $C = \{(i, j, \sigma): \sigma \in U$ appears in location $j$ in clause $I\}$.

$$S(U, C) = \{\bar{y}: \forall i, 1 \leqslant i \leqslant |\bar{y}|, \exists j \ (i, j, y_i) \in C \wedge \forall i, j \ (y_i \neq \neg y_j)\}$$

$$m((U, C), \bar{y}) = |\bar{y}|$$

$$\max_{\bar{y}} |\{l: 1 \leqslant l \leqslant |\bar{y}|, \forall i, 1 \leqslant i \leqslant |\bar{y}|, \exists j \ (i, j, y_i) \in C \wedge \forall i, j \ (y_i \neq \neg y_j)\}|.$$

(Note. This is not the same as MAX SAT from [PY], whose infinite version, as mentioned in Section 1, is trivial.)

MAX SAT2$^\infty$. $I^\infty$ is a recursive set of variables, and a recursive collection $C$ of clauses represented by triples.

Q. Is there a truth assignment that satisfies all the clauses in $C$?

13. MAX TILING. $I$ is a grid $D$ of size $n \times n$, and a set of tiles $T = \{t_1, ..., t_m\}$. (We assume the reader is familiar with the rules of tiling problems. See, e.g., [H1].)

$$S(D, T) = \{Y: Y \text{ is a legal tiling of some portion of } D \text{ with tiles from } T\}$$

$m((D, T), Y) = k$ iff $Y$ contains a tiling of a full $k \times k$ subgrid of $D$.

MAX TILING$^\infty$. $I^\infty$ is a recursive set of tiles $T$.

Q. Can $T$ tile the positive quadrant of the infinite integer grid?

We now prove that the infinitary versions of these problems are all $\Sigma_1^1$-complete. From Theorem 1 it will then

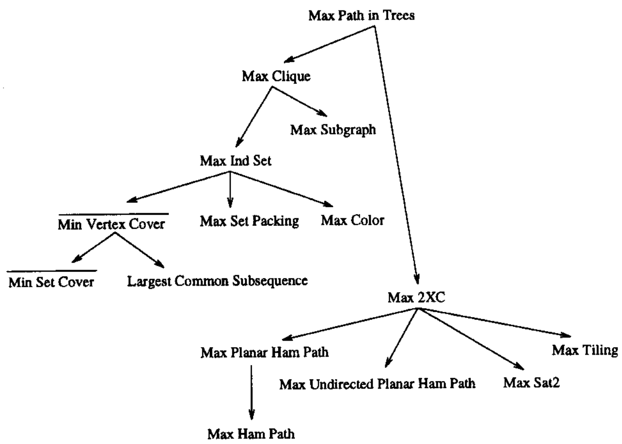



**FIG. 1.** Scheme of the $M$-reductions.

follow that the finitary versions must be outside MAX $\Sigma_1$. By Proposition 2 it suffices to show $\Sigma_1^1$-hardness, for which we shall exhibit appropriate $M$-reductions between finitary versions, and employ Theorem 2.

First note that MAX PATH IN TREES$^\infty$ is $\Sigma_1^1$-complete. This is simply Kleene's result (see [R, p. 396]). We now prove all the others $\Sigma_1^1$-hard by exhibiting $M$-reductions. See Fig. 1.

PROPOSITION 3. MAX PATH IN TREES $\propto_M$ MAX CLIQUE.

*Proof.* Let $T=(N, P, O)$ be an instance of MAX PATH IN TREES, with $m_1$ as the objective function.

Define a monotonic reduction $g=(t_1, t_2, t_3)$, as follows: $t_l(T)=G=(N, E)$, where $E$ contains all the edges of $T$ (but undirected), and edges between node and all its ancestors:

For each $\bar{p}=\langle p_1, ..., p_k\rangle \in S(T)$, let $t_2(\bar{p})=\{p_1, ..., p_k\} \in S(t_1(T))$. For each $Q\in S(t_1(T))$, let $t_3(Q)=\{q: q\in Q\}$ that are ordered according to their distance from the root. ▮

PROPOSITION 4. MAX CLIQUE $\propto_M$ MAX IND SET.

*Proof.* Appears in Section 4. ▮

PROPOSITION 5. MAX IND SET $\propto_M$ $\overline{\text{MIN VERTEX COVER}}$.

*Proof.* These are actually the same problem, so that the trivial reduction is fine. ▮

PROPOSITION 6. (a) MAX IND SET $\propto_M$ MAX SET PACKING [K].

(b) $\overline{\text{MIN VERTEX COVER}}$ $\propto_M$ $\overline{\text{MIN SET COVER}}$.

*Proof.* Let $G=(V, E)$, where $V=\{v_l, ..., v_n\}$, be an instance of MAX IND SET (or $\overline{\text{MIN VERTEX COVER}}$). Define a monotonic reduction $g=(t_1, t_2, t_3)$, as

$$t_1(G)=C=\{S_1, ..., S_n\},$$
$$\text{where} \quad S_i=\{(i, j): (v_i, v_j)\in E\}.$$

This is for MAX SET PACKING. For $\overline{\text{MIN SET COVER}}$ let $t_1(G)$ be $C$ and $A=\{(i, j): (v_i, v_j)\in E\}$.

For each $Y\in S(G)$, let $t_2(Y)=\{S_j: v_j\in Y\}\in S(t_1(G))$. For each $Z\in S(t_1(G))$, let $t_3(Z)=\{v_j: S_j\in Z\}\in S(G)$. ▮

PROPOSITION 7. MAX CLIQUE $\propto_M$ MAX SUBGRAPH.

*Proof.* Let $G=(V, E)$ with $|V|=n$ be an instance of MAX CLIQUE. Define $g=(t_1, t_2, t_3)$ as follows: $t_1(G)=(G, Q)$, where $Q$ is a clique with $n$ vertices $\{u_1, ..., u_n\}$.

For each $Y=\{y_1, ..., y_k\}\in S(G)$, let $t_2(Y)=\{y_i, u_i):1\leqslant i\leqslant k\}$. For each $Z\in S(t_1(G))$, let $t_3(Z)=\{y_i$: for each $1\leqslant j\leqslant i, (y_j, u_j)\in Z\}$. ▮

PROPOSITION 8. MAX IND SET $\propto_M$ MAX COLOR.

*Proof.* Let $G=(V, E)$, with $V=\{v_1, ..., v_n\}$, be an instance of MAX IND SET. Define $g=(t_1, t_2, t_3)$ as follows: $t_1(G)=(G, C)$, where $C=\{c_0, ..., c_n\}$.

For each $Y\in S(G)$, let $t_2(Y)=\bar{y}$, where

$$y_i=\begin{cases} c_0 & \text{if} \quad v_i\in Y\\ c_i & \text{if} \quad v_i\notin Y.\end{cases}$$

For each $\bar{y}\in S(t_1(G))$, let $t_3(\bar{y})=\{v_i: 1\leqslant i\leqslant n, y_i=c_0\}$. ▮

PROPOSITION 9 (Based on [M]). $\overline{\text{MIN VERTEX COVER}}$ $\propto_M$ LCS.

*Proof.* Let $G=(V, E)$, where $V=\{v_1, ..., v_n\}$ and $E=\{e_1, ..., e_r\}$, be an instance of $\overline{\text{MIN VERTEX COVER}}$. Define $g=(t_1, t_2, t_3)$ as follows: $t_1(G)=R$, a set of $r+1$ sequences over $V$. $R$ consists of the sequence $\langle v_1v_2\cdots v_n\rangle$, and for each $e_i=(v_j, v_m)\in E$, where $j<m$, the sequences

$$s_i=\langle v_1v_2\cdots v_{j-1}v_{j+1}\cdots v_mv_1v_2\cdots v_{m-1}v_{m+1}\cdots v_n\rangle.$$

For each $Y\in S(G)$, let $t_2(Y)=\bar{y}$, where $\bar{y}$ is a sequence of the vertices in $Y$, in ascending order. Now, $\bar{y}\in S(t_1(G))$ is a common sequence, because for each $e_i=(v_j, v_m)\in E$ $(j<m)$ it is not the case that $v_j$ and $v_m$ are in $Y$, which is the *only* case for which an ascending sequence is not a subsequence of $s_i$:

For each $\bar{y}\in S(t_1(G))$, let $t_3(\bar{y})=\{v: v\in\bar{y}\}=Y$.

$(V-Y)$ is a vertex-cover, because for each $e_i=(v_j, v_m)\in E$, at least $v_j$ or $v_m$ is not in $\bar{y}$, and thus in $(V-Y)$. Otherwise, $\bar{y}$ could not be a subsequence of $s_i$.

If $R$ is represented by triples of the form $(i, j, k)$, meaning that $v_k\in\Sigma$, is the $j$th character in sequence $i$, then the reduction is monotonic. ▮

PROPOSITION 10. MAX PATH IN TREES $\propto_M$ MAX 2XC.

*Proof.* Let $T=(N, P, 0)$, where $N=\{0, d_1, d_2, ..., d_n\}$, be an instance of MAX PATH IN TREES, with $m_2$ as the objective function.

Define a monotonic reduction $g=(t_1, t_2, t_3)$, as $t_1(T)=G=(X, C)$, where $X=\{0, \hat{d}_1, d_1, \hat{d}_2, d_2, ..., \hat{d}_n, d_n\}$, and $C=\{(u, \hat{v}): (u, v)\in P\}\cup\{(u, \hat{u}): u\neq 0\}$.

For each $\bar{p} = \langle p_1, ..., p_k \rangle \in S(T)$, let

$$t_2(\bar{p}) = \begin{cases} B & \text{if } p_1 = 0 \\ \varnothing & \text{if } p_1 \neq 0, \end{cases}$$

where

$$B = \{(p_1, \hat{p}_2), (p_2, \hat{p}_3), ..., (p_{k-1}, \hat{p}_k)\} \cup \{(u, \hat{u}): u \notin \bar{p} \wedge u < \max\{p_i: 1 \leqslant i \leqslant |\bar{p}|\}\}.$$

For each $B \in S(t_1(T))$, let $t_3(B) = \langle 0, d_1, d_2, ..., d_k \rangle$, where $\{0, \hat{d}_1, d_1, \hat{d}_2, d_2, ..., \hat{d}_k\}$ are covered by $B$, but $d_k$ is not covered by $B$. ▮

In [H2] there is a direct proof of the $\Sigma_1^1$-hardness of detecting Hamiltonicity even in (directed or undirected) highly recursive graphs[5] of degree 3. However, the proof in [H2] does not work if the graphs are to be planar. Here we prove the result for *planar* recursive graphs, by exhibiting a monotonic reduction from MAX 2XC.

PROPOSITION 11. MAX 2XC $\propto_M$ MAX PLANAR HAM PATH.

*Proof.* The reduction is based on modifying the nonmonotonic reduction that appears in [GJS], to be monotonic. We do not repeat the entire description appearing in [GJS] and assume that the reader is familiar with it. In order to follow our modification, it helps to have Figs. 7 and 8 of [GJS] available.

Let there be given a set $X = \{a_1, ..., a_t\}$, and a collection $S = \{S_1, S_2, ..., S_n\}$ of pairs from $X$. In [GJS], a planar directed graph $G$ was constructed, which has a Hamiltonian path iff $S$ contains an exact cover for $X$. A node $f_i$ and a sequence of $5 \times t$ nodes is associated therein with each set $S_i$ (see Fig. 7 of [GJS, p. 256]). Any Hamiltonian path $P$ must begin from the first line in the figure. Thereafter, for each $k$, $1 \leqslant k \leqslant n$, $P$ reaches $f_k$, and "turns" right or left according to whether or not $S_k$ is in the cover. It then proceeds along line $k$ and updates the line, finally arriving at $f_{k+l}$. Passing along line $k$ from the right is possible only if the elements in $S_k$ were not previously chosen, and passing the last line is possible only if all the elements in $X$ were already chosen.

We incorporate the following changes in order to make the reduction monotonic:

- Instead of one node $f_i$ for each set $S_i$, we order the sets in $S$ with repetitions as follows:

$$S_1, \quad S_1, S_2, S_1, S_2, \quad S_1, S_2, S_3, S_1, S_2, S_3, S_1, S_2, S_3, ....$$

We then associate a node $f_i$ with each element in this sequence. The size of the sequence is $O(n^3)$. In this manner one can choose the sets in such a way that the elements $\{a_1, a_2, ..., a_t\}$ will be covered in order. We also change the internal structure of the nodes (Fig. 8 of [GJS, p. 257]) so as to allow a set $f_i$ to be chosen only if all the elements that are smaller than those in $f_i$ were already covered.

[5] A graph is *highly recursive* if it is recursive, its degree at each node is finite, and the function listing a node's neighbors is recursive too.

- Instead of the nodes matching *all* the elements in $X$, we insert in the first line only nodes matching elements in $S_1$. Each line $i$ will contain nodes matching all the elements that are represented in line $i-1$, and those that are represented by $f_i$.
- Every line will appear twice, so that we return to $f_{i+1}$ from the same side we entered.
- We add two nodes to each edge that connects $f_i$ to its line on the right, in order to force more turns to the right. (Recall that right turns correspond to choosing elements in the cover.)

This reduction is monotonic, since the addition of sets to $S$ will increase the graph only outwards. There will be more $f_i$'s and more lines, but there will be no need to eliminate nodes or edges.

For each $Y \in S(X, S)$ which is a partial cover of $X = \{a_1, ..., a_t\}$ such that $m(Y) = k$, $t_2(Y)$ will be the path that starts from the first line and turns right from the $f_i$'s matching the sets in $Y$, such that $\{a_1, ..., a_k\}$ will be covered in order.

For each partial Hamiltonian path $H \in S(t_1(X, S))$, that starts from the first node, the associated cover will contain exactly the sets matching those $f_i$'s from which the path turns right. ▮

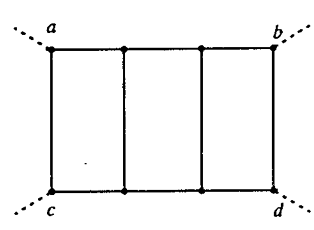
PROPOSITION 12. MAX PLANAR HAM PATH $\propto_M$ MAX HAM PATH.

*Proof.* The trivial reduction $t_1(G) = G$ works. ▮

PROPOSITION 13. MAX 2XC $\propto_M$ MAX UNDIRECTED PLANAR HAM PATH.

*Proof.* Let $X = \{a_1, ..., a_t\}$ and $S = \{S_1, ..., S_n\}$, with each $S_i \in X \times X$. We construct an undirected planar graph, which is similar to the graph $G$ described in the proof of Proposition 11. Again, it helps to have Figs. 7 and 8 of [GJS] available:

- The nodes $f_i$ and the two nodes to the right remain the same. The edges between them are now undirected.
- The following structure, taken from [GJT], replaces each node in the lines of $G$:




This structure forces the path that enters at node $a$ (respectively, $b$, $c$, $d$), to exit from node $b$ (respectively, $a$, $d$, $c$).

- Each $f_i$ is connected to node $a$ of the leftmost structure in its appropriate line and to node $d$ of the rightmost structure in the same line, in order to force the path from left to

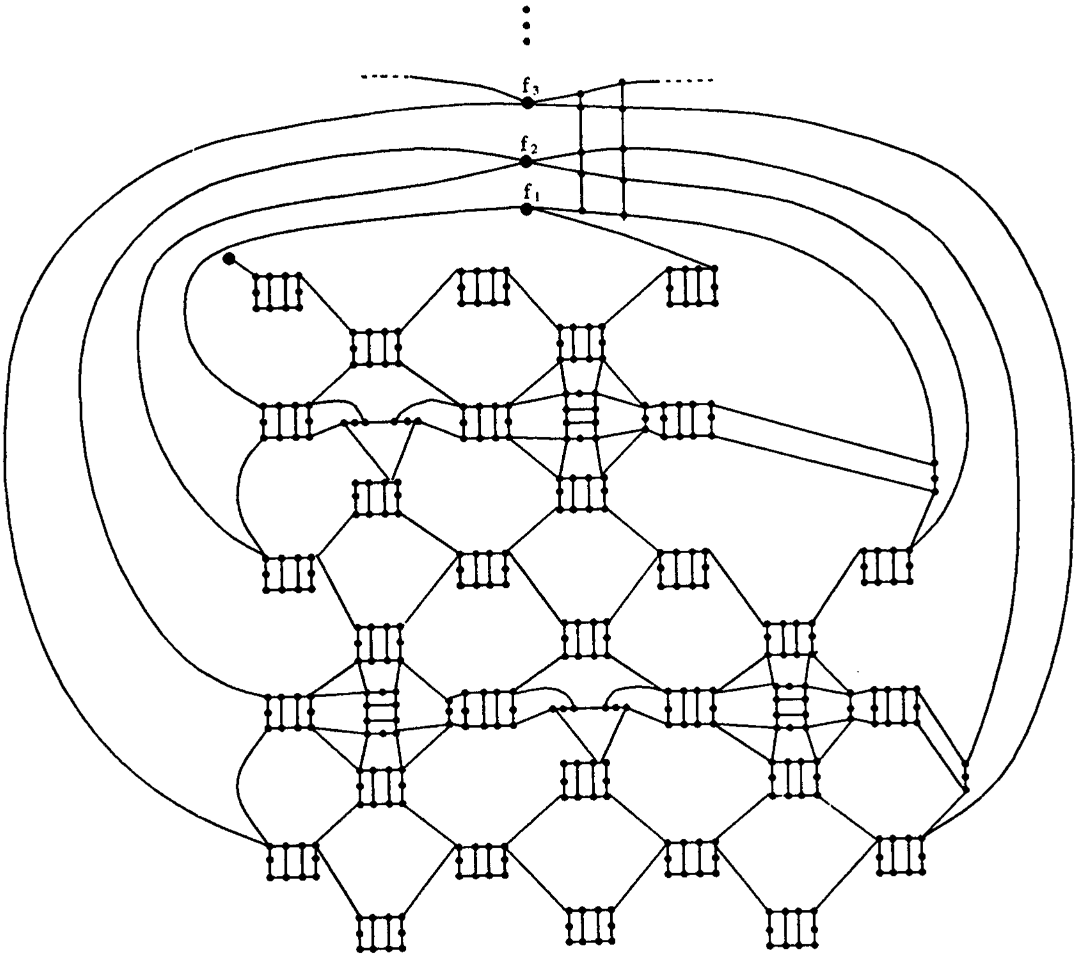
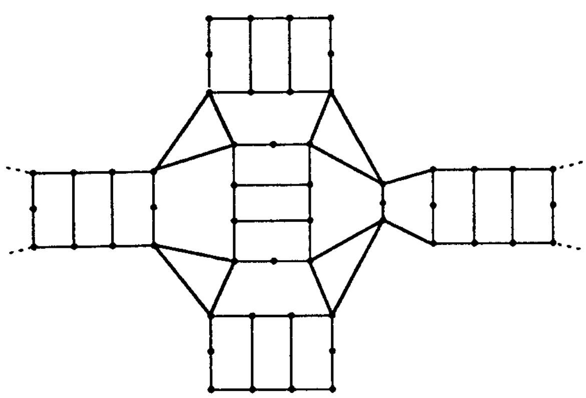


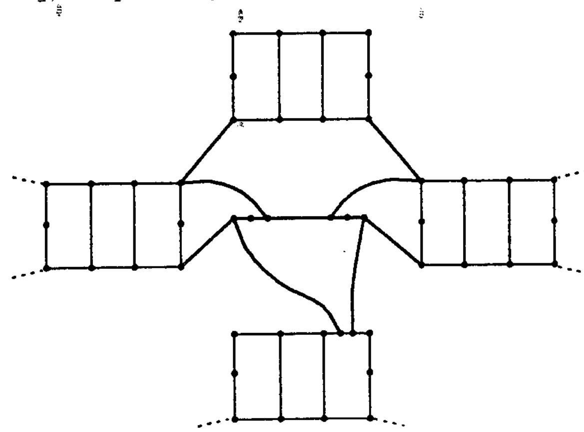

FIGURE 2

right to be along the top portions, and the path from right to left to be along the bottom portions.

- The second line of each $f_i$ is connected to the previous line and to $f_{i+1}$, such that the path along this line will be along the top portions in both directions. (Recall that the second line corresponding to each $f_i$ is used in order to direct the path to exit from the same side it entered the first line.)



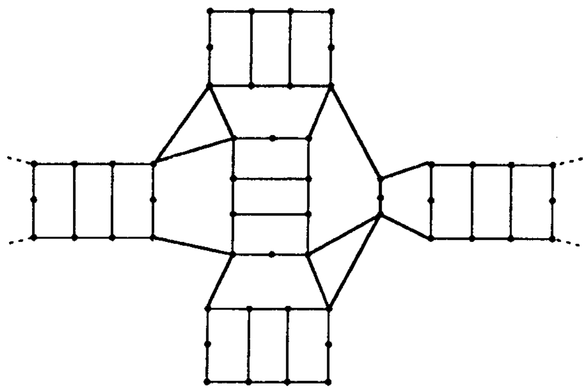
- The "copy structure" (appearing in Fig. 8A of [GJS, p. 257]) is replaced by the following structure:
- The "update structure" (appearing in Fig. 8B of [GJS, p. 257]) is replaced by the following structure:



If the path passes from left to right (through the top portions) then this structure is just a "copy structure."

Otherwise (the path passes through the bottom portions) the structure updates, i.e., it requires that the upper "node" (which is now a structure) was visited already (it means that the appropriate element was not chosen already), and it leaves the bottom node-structure nonvisited, in order to indicate that this element is chosen.

- As before, we allow a set $f_i$ to be chosen only if all the elements that are smaller than those in $f_i$ were already covered. For the appropriate smaller elements we use the following structure:



A part of such a graph thus looks like Fig. 2.

PROPOSITION 14. MAX 2XC $\propto_M$ MAX SAT2.

*Proof.* Let $X = \{d_1, d_2, ..., d_n\}$ be a set, and $B$ be a set of unordered pairs from $X$. Define $g = (t_1, t_2, t_3)$ as

$$t_1(X, B) = U = \{(x, y), \neg(x, y), \text{ where } (x, y) \in B\}.$$

We view $U$ as a set of variables, and by the unorderedness we take $(x, y)$ and $(y, x)$ to be equal. The sequence of clauses $C$ is obtained by juxtaposing, in the order listed, the clauses in the following set:

$$C = \{C_1, C_{1'}, C_2, C_{2'}, ..., C_n, C_{n'}\},$$

where, for each $1 \leqslant i \leqslant n$,

$$C_i = \bigvee_{(d_i, x) \in B} (d_i, x)$$

$$C_{i'} = \{\neg(d_i, x) \vee \neg(d_i, y) \mid x \neq y \text{ and } (d_i, x), (d_i, y) \in C_i\}$$
$$\cup \{\neg(d_i, x) \vee \neg(y, x) \mid y \neq d_i \text{ and } (d_i, x) \in C_i \text{ and } (y, x) \in U\}.$$

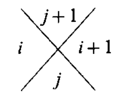
The clause $C_i$ forces a choice of some pair containing $d_i$, and the set of clauses $C_{i'}$, forbids double choices of an element.

Now, for each $Y \in S(X, B)$ which is a cover for $\{d_1, ..., d_k\}$, we let $t_2(Y) = \bar{y}$, where $\bar{y}$ contains the variables satisfying the clauses in $\{C_1, C_{1'}, ..., C_k, C_{k'}\}$. For each $\bar{y} \in S(t_1(X, B))$, we let $t_3(\bar{y})$ contain those pairs $(d_i, d_j)$ that denote the variables satisfying clauses $\{C_1, C_2, ..., C_k\}$. ∎

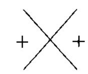
PROPOSITION 15. MAX 2XC $\propto_M$ MAX TILING.

*Proof.* Let $X = \{1, ..., n\}$, and let $C$ be a set of unordered pairs from $X$. We construct, monotonically, a tiling problem on the $n \times n$ grid, such that there is an exact cover for $\{1, 2, ..., k\}$ in $C$ iff the initial $k \times k$ subgrid can be tiled.

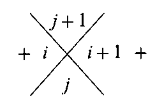
Define $g = (t_1, t_2, t_3)$ as follows: $t_1(X, C) = T$, where $T$ contains tiles with indexes (encoding colors) of the form:



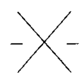

This forces each of them to be inserted only in a unique square $(i, j)$. In addition to these indexes, the tiles will contain more symbols as



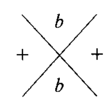

for $1 \leqslant i, j \leqslant n, i \neq 1$ (i.e., all the squares, except for the first column). Thus, they will actually appear as:



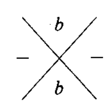

We also add symbols of the form




for $1 \leqslant i < j \leqslant n$ (i.e., the squares above the diagonal).

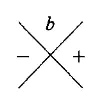



for $1 \leqslant b, i, j \leqslant n, j \neq b, i \neq 1$ (i.e., all squares except the $b$th row and the first column).

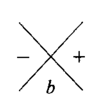



for $1 \leqslant b \leqslant n, 1 \leqslant i < j \leqslant n, j \neq b$.

Now, for each $(a, b) \in C$ (where we assume w.l.o.g. that $a < b$) we add the symbols,

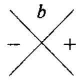



for $i = j = a$ (this will be a candidate for square $(a, a)$).

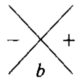



for $i = a, j = b$ (a candidate for square $(a, b)$).

For each $Y \subseteq C$ which is an exact cover for $\{1, ..., k\}$, $t_2(Y)$ is the following tiling: for each $(a, b) \in C$, where $a < b$, tile the square $(a, a)$ with

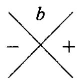



and the square $(a, b)$ with



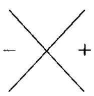

Now, the remaining squares in the $k \times k$ grid can only be tiled in a single unique manner.

Conversely, for each $S$ which is a tiling of the $k \times k$ grid, $t_3(S)$ contains all pairs $(a, b)$ such that the tile



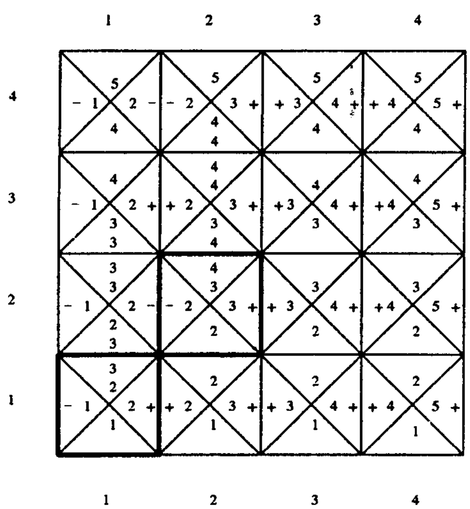

appears in square $(a, a)$.

It follows that $t_3(S)$ is an exact cover for $\{1, ..., k\}$, because of the fact that, in each line, there must be *exactly* one tile whose − and + parts of the colors are




The reduction is monotonic, since an addition of sets to $C$ will cause only the addition of tiles to $T$. ▮

EXAMPLE. $C = \{(1, 3), (2, 4), (1, 4)\}$, $D = \{1, 2, 3, 4\}$. The tiling is:



*Note.* Two additional graph problems of interest, that have not been dealt with in [BG2], are planarity and graph isomorphism. Here is a brief summary of our findings for them:

- The problem of detecting whether a recursive graph is planar is co-r.e.

- Determining whether two recursive graphs are isomorphic is arithmetical for graphs that have finite degree and contain only finitely many connected components. More precisely, this problem is in $\Pi_1^0$ for highly recursive trees; in $\Pi_3^0$ for recursive trees with finite degrees; in $\Sigma_2^0$ for connected highly recursive graphs; and in $\Sigma_4^0$ for recursive graphs with finite degrees that have finitely many connected components. As to the isomorphism problem for general recursive graphs, in the conference version of this paper,[6] we left open the question of whether the problem is arithmetical or not. Morozov [Mo] has recently proved that the problem is indeed $\Sigma_1^1$-complete.

## ACKNOWLEDGMENT

We thank the referee for several very helpful comments.

## REFERENCES

[AMS] R. Aharoni, M. Magidor, and R. A. Shore, On the strength of König's duality theorem, *J. Combin. Theory Ser. B* **54**, No. 2 (1992), 257–290.

[ALMSS] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy, Proof verification and hardness of approximation problems, *in* "Proceedings, 33rd IEEE Conf. on Found. of Comput. Sci.," Pittsburgh, PA, 1992.

[AS] S. Arora and S. Safra, Probabilistic checking of proofs, *in* "Proceedings, 33rd IEEE Conf. on Found. of Comput. Sci.," Pittsburgh, PA, 1992.

[BG1] R. Beigel and W. I. Gasarch, On the complexity of finding the chromatic number of a recursive graph, I, II, *Ann. Pure Appl. Logic* **45** (1989), 1–38, 227–247.

[BG2] R. Beigel and W. I. Gasarch, unpublished results, 1986–1990.

[BJY] D. Bruschi, D. Joseph, and P. Young, "A Structural Overview of NP Optimization Problems," Technical Report 861, Department of Computer Science, University of Wisconsin, Madison, 1989.

[F] R. Fagin, Generalized first-order spectra and polynomial-time recognizable sets, *in* "Complexity of Computations" (R. Karp, Ed.), SIAM-AMS Proceedings, Vol. 7, SIAM, Philadelphia, pp. 43–73, 1974.

[GJ] M. R. Garey and D. S. Johnson, "Computers and Intractability: A Guide to the Theory of NP-Completeness," Freeman, San Francisco, 1979.

[GJS] M. R. Garey, D. S. Johnson, and L. Stockmeyer, Some simplified NP-complete graph Problem, *Theor. Comput. Sci.* **1** (1976), 237–267.

[GJT] M. R. Garey, D. S. Johnson, and R. E. Tarjan, The planar Hamiltonian circuit problem is NP-complete, *SIAM J. Comput.* **5**, No. 4 (1976), 704–714.

[6] "Proceedings, 8th IEEE Structure in Complexity Theory Conf.," pp. 292–304, IEEE Press, New York, 1993.

[H1] D. Harel, Effective transformations on infinite trees, with applications to high undecidability, domains, and fairness, *J. Assoc. Comput. Mach.* **33** (1986), 224–248.

[H2] D. Harel, Hamiltonian paths in infinite graphs, *Israel J. Math.* **76** (1991), 317–336; preliminary version *in* "Proceedings, 23rd ACM Symp. on Theory of Computing," pp. 220–229, ACM Press, New York, 1991.

[K] R. M. Karp, Reducibility among combinatorial problems, *in* "Complexity of Computer Computations" (R. E. Miller and J. W. Thatcher, Eds.), pp. 85–103, Plenum, New York, 1972.

[KT] P. G. Kolaitis and M. N. Thakur, Logical definability of NP optimization problems, *in* "6th IEEE Conf. on Structure in Complexity Theory, 1991," pp. 353–366. *Inform. and Comput.*, to appear.

[LM] M. E. Leggett and J. Moore, Optimization problems and the poly-nomial time hierarchy, *Theor. Comput. Sci.* **15** (1981), 279–289.

[M] D. Maier, The Complexity of problems on subsequences and supersequences, *J. Assoc. Comput. Mach.* **25** (1978), 322–336.

[Mo] A. S. Morozov, Functional trees and automorphisms of models, *Algebra and Logic* **32** (1993), 28–38.

[PR] A. Panconesi and D. Ranjan, Quantifiers and approximation, *Theor. Comput. Sci.* **107** (1993), 145–163.

[PY] C. H. Papadimitriou and M. Yannakakis, Optimization, approximation, and complexity classes, *J. Comput. System Sci.* **43** (1991), 425–440.

[R] H. Rogers, "Theory of Recursive Functions and Effective Computability," McGraw–Hill, New York, 1967.

Printed by Catherine Press, Ltd., Tempelhof 41, B-8000 Brugge, Belgium