Approximation and Existential Second-Order Logic [*][†]

Edmund Ihler[‡]

Abstract

The approximation properties of NP-complete optimization problems vary considerably. While for some problems, the solution cannot be approximated with constant accuracy unless P=NP, others can be approximated to any arbitrary degree of accuracy. In between there are problems for which approximation with some constant accuracy exists; for others, no constant accuracy approximation is known.

In this paper, we aim at capturing these kinds of approximation properties by introducing a natural hierarchy of logically defined classes of optimization problems. Our hierarchy makes use of the full expressive power of existential second-order logic; it contains the classes MAX SNP, MAX NP [PY88], and MAX Π_1 [PR90] as lowest classes.

1 Introduction

Approximate solutions for NP-hard optimization problems are important in practice and appealing in the theory of algorithms and complexity. In particular, we are interested in the accuracy with which an optimal solution can be approximated, and in the complexity of a suitable algorithm. There is a variety of results in this field: some optimization problems can be approximated with arbitrary accuracy (e.g. Max Knapsack), while for others no constant accuracy approximation exists (e.g. TSP), unless P=NP [GJ79]. For most optimization problems only weaker results are known: many problems can be approximated with *some* constant accuracy (e.g. $\triangle$TSP, Steiner Tree),

[*]This work was supported by the Deutsche Forschungsgemeinschaft under grant Wid 810/5.

[†]Appeared as Technical Report: Bericht Nr. 26, Institut für Informatik, Universität Freiburg, 1990.

[‡]Institut für Informatik, Universität Freiburg, Rheinstraße 10 - 12, D - 79104 Freiburg, Germany, e-mail: ihler@informatik.uni-freiburg.de

but it is not known whether *arbitrary* accuracy can be achieved. For many optimization problems, not even a constant accuracy approximation is known (e.g. MAXIMUM INDEPENDENT SET, GROUP STEINER PROBLEM [Ihl90]). In order to obtain a general theory of approximability, several classifications of optimization problems have been proposed. Most classes have been defined through the concept of Turing machines (e.g. [Kre88, BS89]); they did not succeed in separating the problems according to their approximation properties.

A new and promising approach has been proposed by Papadimitriou and Yannakakis in [PY88]. They captured optimization problems with constant approximation by defining a class MAX NP of problems based on existential second-order formulas. For a subclass MAX SNP, they identified complete problems for which no arbitrary accuracy approximation is known; such an approximation for any complete problem, however, implies such an approximation for the whole class. As a consequence of the large number of MAX SNP-complete problems, MAX SNP-hardness is a meaningful criterion for the approximability of a problem; in fact, the famous **TSP** and STEINER TREE problems have been shown to be MAX SNP-hard ([PY89, BP89]). Recently, however, it has been proved [PR90] that some important problems do not belong to MAX NP (e.g. MAX CLIQUE); a similar class, MAX Π_1, and subclasses RMAX(k), have been introduced for these problems. Complete problems have not been found for MAX Π_1, but only for a restricted version.

The motivation for syntactically defined classes of optimization problems lies in Fagin's result [Fag74] that, roughly speaking, any problem of NP is equivalent to a sentence in existential second-order logic. The problem instances and solutions are encoded in finite structures G and S (see e.g. [Imm89] for a discussion of the relationship between logic and computational complexity). In [PY88, PR90], an existential second-order formula $\exists \mathcal{S} \Phi(x, \mathcal{G}, \mathcal{S})$ is interpreted as a maximization problem. Here we ask for a structure S that maximizes the number of values u of the tuple x of the free variables for which $\Phi(x, \mathcal{G}, \mathcal{S})$ holds. Here, specific conditions determine the form of Φ.

In this paper, we introduce a class SO$\exists$-OPT *without any* restrictions on the form of the first-order formula Φ, thereby making use of the full expressive power of existential second-order logic. We present natural complete problems for SO$\exists$-OPT. Furthermore, we cover SO$\exists$-OPT with hierarchies Π_i-OPT and Σ_i-OPT; the classes MAX SNP, MAX NP, and MAX Π_1 coincide with our classes for $i \in \{0, 1\}$. Complete problems in MAX Π_1 have been shown not to be approximable [PR90], unless P=NP. No such proof is known for MAXIMUM INDEPENDENT SET. For accomodating problems like this as complete problems, we introduce a subhierarchy for each class of our hierarchies. The subhierarchy of Π_1-MAX turns out to be a generalized version of RMAX(k). For problems complete in $\Pi_{1,1}$-MAX, as well as for those complete in RMAX(2), approximability for *some* constant implies approximability for an *arbitrary* constant. This is due to the fact that Maximum Independent Set has this property [GJ76] and is complete in $\Pi_{1,1}$-MAX. We try to separate this property from pure constant approximability by generalizing our subhierarchies appropriately. For instance, Maximum Independent Set is complete in $\Pi^{+}_{1,1}$-MAX, but we conjecture that nevertheless this does not imply arbitrary approximability for all other complete problems in $\Pi^{+}_{1,1}$-MAX.

In addition, our formalism works equally well for minimization problems; many of our results hold for minimization, too.

2 Basic definitions

2.1 Optimization and approximation

In line with [GJ79], we describe an *optimization problem* Π by a triple $(\mathbf{I}, \mathbf{S}, /\cdot,\cdot/)$. $\mathbf{I}$ is the set of *instances* of Π, $\mathbf{S}$ is a mapping, producing a finite set $\mathbf{S}(I)$ of *solutions* for each instance $I \in \mathbf{I}$, and $/\cdot,\cdot/$ is a *solution value function*, mapping the pairs (I, S) with $I \in \mathbf{I}$ and $S \in \mathbf{S}(I)$ into the positive integers. The set of all optimization problems Π is called problem class *OPT*. $S^* \in \mathbf{S}(I)$ denotes an *optimum solution*, i.e., a solution with the optimum value $/I, S/$.

If I is understood, we will often write $/S/$ instead of $/(I, S)/$. OPT can be specified to denote a class MAX of maximization problems, or a class MIN of minimization problems; we use OPT to denote any one of both. To make maximization and minimization explicit, we write Π_{max} and Π_{min}.

A *polynomial time* algorithm A is an *approximation algorithm (PTA)* for $\Pi \in$ OPT, if A finds for each instance $I \in \mathbf{I}$ a solution $A(I) \in \mathbf{S}(I)$.

A PTA is called a $(1+\epsilon(I))$-*PTA* for Π, if ϵ is a function $\mathbf{I} \to [0, \infty)$, and for each $I \in \mathbf{I}$:

$$\begin{array}{rcll} /S^*/ & \leq & (1+\epsilon(I)) \;\; /A(I)/ & ;\text{if } \Pi \in \text{MAX}, \\ /A(I)/ & \leq & (1+\epsilon(I)) \;\;\; /S^*/ & ;\text{if } \Pi \in \text{MIN}. \end{array}$$

A $(1+\epsilon)$-PTA, where ϵ is an instance-independent constant, is a $(1+\epsilon)$-*CPTA*, or *CPTA*.

A *polynomial time approximation scheme (PTAS)* for a problem Π is a family of CPTAs, containing for each $\epsilon > 0$ a $(1+\epsilon)$-CPTA for Π.

For $\Pi, \Pi' \in$ OPT, we call a triple of polynomial transformations (f, g, ϵ) with $f : \mathbf{I} \to \mathbf{I}'$, $g : \mathbf{I} \times \mathbf{I}' \times \mathbf{S}'(\mathbf{I}') \to \mathbf{S}(\mathbf{I})$, and $\epsilon : [0, \infty) \to [0, \infty)$ a *C-reduction*, denoted by $\Pi \leq_C \Pi'$, if whenever A is a $(1+\epsilon')$-CPTA for Π', then $g \circ (\cdot, f, A \circ f)$ is a $(1+\epsilon(\epsilon'))$-CPTA for Π. If an entire PTAS of Π' is in this way transformed to a PTAS of Π, then we call (f, g, ϵ) an *S-reduction*, writing $\Pi \leq_S \Pi'$.

That is, a C-reduction preserves CPTAs and an S-reduction preserves PTASs. For example, the linear *L-reduction* introduced in [PY88] is an S-reduction. A reduction with $\epsilon(\epsilon') = \epsilon'$, is called $\mathrm{S}^=$-reduction.

Lemma 1 *S-reduction (resp. C-reduction) composes.* $\Box$

For a class $\mathrm{M} \subseteq \mathrm{OPT}$ of optimization problems, $\mathrm{M}\text{-c}_S := \{\Pi : \Pi \in \mathrm{M} \wedge (\forall \Pi' \in \mathrm{M} : \Pi' \leq_S \Pi)\}$ is the class of problems that are *M-complete under S-reduction*. Similary, $\mathrm{M}\text{-c}_C$ is the class of *M-complete problems under C-reduction*.

2.2 Mathematical logic

A *finite similarity type* is a finite set of *function* and *predicate symbols*. Each function (resp. predicate) symbol has a nonnegativ integer, called *arity*, related to it.

We call a symbol *k-ary*, if its arity is k. Writing $\mathcal{F}^{(k)}$ resp. $\mathcal{P}^{(k)}$ means $\mathcal{F}$ is a k-ary function symbol resp. $\mathcal{P}$ is a k-ary predicate symbol. We will denote finite similarity types by $\mathcal{G}$ and $\mathcal{S}$. In addition we have a set of *individual variables* and the *logical symbols* $\neg, \wedge, \forall$, and the *equality symbol* $=$. Let the set of *terms*, of a finite similarity type $\mathcal{G}$, be the smallest set, containing the individual variables and $\mathcal{F}t_1 \cdots t_k$, whenever $\mathcal{F}$ is a k-ary function symbol of $\mathcal{G}$ and $t_1, \ldots, t_k$ are yet terms of $\mathcal{G}$. An *atomic formula* of $\mathcal{G}$ is a string $t_1 = t_2$ or $\mathcal{P}t_1 \cdots t_k$, if $\mathcal{P}$ is a predicate symbol and $t_1, \ldots, t_k$ are terms of $\mathcal{G}$. An *atomic literal* is a formula ϕ or $\neg\phi$, when ϕ is an atomic formula. A *first-order $\mathcal{G}$-formula* is a member of the smallest set, containing each atomic formula of $\mathcal{G}$, and $\neg\phi$, $\phi \wedge \psi$, $\forall x \phi$, whenever x is an individual variable and ϕ, ψ are yet first-order $\mathcal{G}$-formulas. The formulas $\phi \vee \psi$, $\exists x \phi$, $\exists! x \phi$, $\phi \Rightarrow \psi$, $\exists xyz \phi$, $\bigvee_{i=1}^{k} \phi_i$, and so on, are defined in the usual way. An *existential second-order $\mathcal{G}, \mathcal{S}$-formula* (in prenex normal form) is a string $\exists \mathcal{S} \phi$ where ϕ is a first-order $\mathcal{G} \cup \mathcal{S}$-formula, and $\exists \mathcal{S}$ stands for $\exists \mathcal{S}_1 \cdots \exists \mathcal{S}_N$, where $\mathcal{S} := \{\mathcal{S}_1, \ldots, \mathcal{S}_N\}$. The set of all first-order $\mathcal{G} \cup \mathcal{S}$-formulas is called $\mathrm{FO}_{\mathcal{G} \cup \mathcal{S}}$. $\mathcal{S}_\nu$ are viewed here as second-order *function* (resp. *predicate*) *variables*. Let $SO\exists_{\mathcal{G},\mathcal{S}}$ be the set of all existential second-order $\mathcal{G}, \mathcal{S}$-formulas. We do not distinguish between function (resp. predicate) variables and function (resp. predicate) symbols. We assume, that it is clear, what we mean by a *free variable* and by a *quantifier free* formula. Let the set of free variables of a formula ϕ be $Fr(\phi)$. We define a *sentence* as a formula with no free variable.

Up to here, formulas and sentences are only strings of symbols. Now we introduce some meaning to it, by what we call *structures*. Suppose $\mathcal{G}$ to be a set of distinct symbols $\mathcal{G}_1, \ldots, \mathcal{G}_M$, then we call $G := (U; G_\mu)$ a *finite $\mathcal{G}$-structure*, if we have a nonempty finite set U, the *universe*, for each k-ary function symbol $\mathcal{G}_\mu$ a function $G_\mu : U^k \rightarrow U$, and for each k-ary predicate symbol $\mathcal{G}_\mu$ a predicate $G_\mu : U^k \rightarrow \{true, false\}$. We only want to give an informal definition of the following: A finite $\mathcal{G}$-structure gives a meaning to each first-order $\mathcal{G}$-sentence by making it "true" or "false". To determine which truth value is right, we have to set in the functions for the function symbols and the predicates for the predicate symbols and then we have to treat the logical symbols for a truth evaluation as usually, where the individual variables are variables for the elements of the universe. For an existential second-order $\mathcal{G}, \mathcal{S}$-sentence, we have a given $\mathcal{G}$-structure $G = (U; G_\mu)$ and have to ask, whether there is some $\mathcal{S}$-structure $S = (U; S_\nu)$ making its first-order part Φ "true". Remember, $\mathcal{S}$ in the second-order quantifier $\exists \mathcal{S}$ is a variable tuple, standing for the functions and predicates of some $\mathcal{S}$-structure.

3 The class SO$\exists$-OPT

SO$\exists$-OPT is the class of problems that can be expressed as $opt_{\mathcal{S}}|\{x : \Phi(x, \mathcal{G}, \mathcal{S})\}|$ with a set $\mathbf{I}$ of structures G, and with $\Phi(x, \mathcal{G}, \mathcal{S}) \in \mathrm{FO}_{\mathcal{G} \cup \mathcal{S}}$, where x is the tuple of the free individual variables. Again, *opt* represents any of max or min.

Two problems $(\mathbf{I_1}, \mathbf{\Phi_1}(x, \mathcal{G}_1, \mathcal{S}_1))$ and $(\mathbf{I_2}, \mathbf{\Phi_2}(x, \mathcal{G}_2, \mathcal{S}_2))$ of SO$\exists$-OPT are *equivalent*, if $\mathcal{G}_1 = \mathcal{G}_2$, $\mathcal{S}_1 = \mathcal{S}_2$, $\mathbf{I_1} = \mathbf{I_2}$, and $\mathbf{\Phi_1} \Leftrightarrow \mathbf{\Phi_2}$.

Remember that we do not allow, for technical reasons, solutions S with $/S/ = 0$. This implies $\exists x \Phi(x, G, S)$ for maximization problems for any solution S and we analogous require $\exists x \neg \Phi(x, G, S)$ for minimization problems for any solution S. It can be seen, that SO$\exists$-OPT $\subseteq$ OPT by formulating each $(\mathbf{I}, \Phi(x))_{max}$ as $(\mathbf{I}, \{S : \exists x \Phi(x)\}, |\{x : \Phi(x)\}|)_{max}$ and $(\mathbf{I}, \Phi(x))_{min}$ as $(\mathbf{I}, \{S : \exists x \neg \Phi(x)\}, |\{x : \Phi(x)\}|)_{min}$. Now consider a problem $(\mathbf{I}, \mathbf{S}, /\cdot,\cdot/)$, where $\mathbf{S}$ and $/\cdot,\cdot/$ are defined using a sentence $\Phi^{Sol}(\mathcal{G}, \mathcal{S})$ and a formula $\Phi^{Val}(x, \mathcal{G}, \mathcal{S}) \in \mathrm{FO}_{\mathcal{G},\mathcal{S}}$ where x is the tuple of free variables of Φ^{Val} as $\mathbf{S} = \{S : \Phi^{Sol}(G, S)\}$ and $/G, S/ = |\{x : \Phi^{Val}(x, G, S)\}|$. We denote these problems by $(\mathbf{I}, \Phi^{Sol}, \Phi^{Val}(x))$. Hence, any $(\mathbf{I}, \Phi^{Sol}, \Phi^{Val}(x))_{max}$ can be expressed as $(\mathbf{I}, \Phi^{Sol} \wedge \Phi^{Val}(x))_{max}$, and any $(\mathbf{I}, \Phi^{Sol}, \Phi^{Val}(x))_{min}$ as $(\mathbf{I}, \neg \Phi^{Sol} \vee \Phi^{Val}(x))_{min}$. Note that $(\mathbf{I}, \Phi^{Sol}, \Phi^{Val}(x))_{max}$ is equivalent to $(\mathbf{I}, \Phi^{Sol}, \neg \Phi^{Val}(x))_{min}$.

3.1 The hierarchy of SO$\exists$-OPT

Let us now define a natural hierarchy of subclasses of SO$\exists$-OPT, such that MAX SNP, MAX NP [PY88], and MAX Π_1 [PR90] are classes of this hierarchy. To this end, we use a prefix classification of FO-formulas.

Let Π_i (resp. Σ_i) be the class of all formulas of FO, having an equivalent prenex formula, where this prenex formula uses predicate symbols only, and where the prefix consists of i alternating quantifiers starting with $\forall$ (resp. $\exists$).

Π_i-*OPT* (resp. Σ_i-*OPT*) is the class of problems $(\mathbf{I}, \Phi(x)) \in$ SO$\exists$-OPT, where $\Phi(x) \in \Pi_i$ (resp. Σ_i).

The following lemma is an immediate consequence of this definition.

Lemma 2 *$\Pi_{\mathbf{0}}$-OPT $=$ $\Sigma_{\mathbf{0}}$-OPT; $\Pi_{\mathbf{0}}$-OPT $\subseteq \cdots \subseteq$ Π_∞-OPT; $\Sigma_{\mathbf{0}}$-OPT $\subseteq \cdots \subseteq \Sigma_\infty$-OPT; Π_i-OPT $\subseteq \Sigma_j$-OPT and Σ_i-OPT $\subseteq \Pi_j$-OPT for $0 \leq i < j$.* $\square$

We can indeed identify $\Pi_{\mathbf{0}}$-MAX ($= \Sigma_{\mathbf{0}}$-MAX) $=$ MAX SNP, and $\Sigma_{\mathbf{1}}$-MAX $=$ MAX NP, and $\Pi_{\mathbf{1}}$-MAX $=$ MAX $\Pi_{\mathbf{1}}$ [PY88, PR90].

In [PR90], subclasses RMAX(k) of MAX $\Pi_{\mathbf{1}}$ are defined by certain restrictions. The kind of restriction is best shown in the $(\mathbf{I}, \Phi^{Sol}, \Phi^{Val}(x))$ characterization of problems. Besides considering only one-predicate solution structures, it is requested that $\Phi^{Val}(x) = \mathcal{S}(x)$ and $\Phi^{Sol} = \forall y \phi(y)$, where $\phi(y)$ is a quantifier-free CNF formula, in which atomic formulas of $\mathcal{S}$ appear only as negated literals, at most k times per disjunction.

We will now introduce subhierarchies of the classes Π_i-OPT and Σ_i-OPT of our hierarchy of SO$\exists$-OPT, such that RMAX(k) is included as a restriction, and RMAX(2)-complete problems shown in [PR90] are complete, too.

3.2 The subhierarchies of SO$\exists$-OPT

To define a subhierarchy for each class of our hierarchy, we further classify FO-formulas.

Let $\mathbf{P}(x)$ be a prefix of a FO-formula in prenex normal form, quantifying the individual variables of the tuple x, and let $y := (y^1, \ldots, y^k)$ be a tuple of tuples of different individual variables. Consider all formulas of FO$_{\mathcal{G}\cup\mathcal{S}}$ of the form $\mathbf{P}(x)\forall y \phi(x, y, \mathcal{G}, \mathcal{S})$, where ϕ is of the form

$$\bigvee_{\lambda=1}^{k} \bigwedge_{\mu=1}^{h^\lambda} (\bigvee_{\nu=1}^{s_\mu^\lambda} \sigma_{\mu\nu}^\lambda(x, y^\lambda) \vee \bigvee_{\nu=1}^{g_\mu^\lambda} \gamma_{\mu\nu}^\lambda(x, y^\lambda))$$

(resp. of the dual form for $\mathbf{P}(x)\exists y \phi(x, y, \mathcal{G}, \mathcal{S})$), each $s_\mu^\lambda \leq j$, each $\gamma_{\mu\nu}^\lambda$ is an atomic literal of $\mathcal{G}$, and each $\sigma_{\mu\nu}^\lambda$ is an atomic literal of $\mathcal{S}$ with individual variables of the tuples x, y^λ only. Using these restrictions for the required equivalent prenex formula of a formula in our classes Π_i,Σ_i, we get families of new classes $\Pi_{i,j}^+$, and $\Sigma_{i,j}^+$. Not allowing the disjunctor $\bigvee_{\lambda=1}^k$ (resp. the conjunctor $\bigwedge_{\lambda=1}^k$) restricts $\Pi_{i,j}^+$ to $\Pi_{i,j}$, and $\Sigma_{i,j}^+$ to $\Sigma_{i,j}$.

$\Pi_{i,j}^{(+)}$-*OPT* (resp. $\Sigma_{i,j}^{(+)}$-*OPT*) is the class of problems $(\mathbf{I}, \Phi(x)) \in$ SO$\exists$-OPT, where $\Phi(x) \in \Pi_{i,j}^{(+)}$ (resp. $\Sigma_{i,j}^{(+)}$).

Lemmas analogous to Lemma 2 can be derived. It is clear, that RMAX(k) $\subseteq \Pi_{1,k}$-MAX $\subseteq \Pi^+_{1,k}$-MAX. We will prove in addition an interesting inclusion-relationship.

Lemma 3 *Π_{i-1}-OPT $\subseteq \Pi^+_{i,j}$-OPT and Σ_{i-1}-OPT $\subseteq \Sigma^+_{i,j}$-OPT for all $i, j \geq 1$.*

Proof: Assume we have a problem $(\mathbf{I}, \Phi(x, \mathcal{G}, \mathcal{S}))$ where the formula $\Phi(x, \mathcal{G}, \mathcal{S}) = \mathbf{P}(x)\phi(x, \mathcal{G}, \mathcal{S})$ is in Π_{i-1}, if $i-1$ is even (in Σ_{i-1}, if $i-1$ is odd). Then we can write the quantifier-free part ϕ in DNF (disjunctive normal form) $\phi \Leftrightarrow \bigvee_{\lambda=1}^{k} \bigwedge_{\mu=1}^{h^\lambda} \varphi^\lambda_\mu(x, \mathcal{G}, \mathcal{S})$, for some constants k, h^λ. Each φ^λ_μ is an atomic literal of $\mathcal{G}$ or $\mathcal{S}$, so we have changed this formula into an equivalent one of $\Pi^+_{i,j}$ (resp. $\Sigma^+_{i,j}$) without making use of the allowed additional quantifier $\forall y$ (resp. $\exists y$), and so no atomic literal is dependent on a tuple y^λ of y. A similary proof is possible for the formulas $\mathbf{P}(x)\phi(x, \mathcal{G}, \mathcal{S})$ of Π_{i-1}, if $i-1$ is odd (resp. of Σ_{i-1}, if $i-1$ is even), using CNF (conjunctive normal form). $\Box$

4 Completeness results

4.1 A complete problem of $\Sigma^{(+)}_{2,1}$-OPT

Our first example of a complete problem is Optimum Group Satisfiability (OPT GSAT), a generalized version of MAX SAT.

OPT GSAT, OPT k-GSAT: Given a set of Boolean DNF formulas, called *clauses*, find a truth assignment optimizing the number of satisfied clauses. For OPT k-GSAT, the number of conjunctions, called *groups*, in each clause is bounded by k.

Lemma 4 *OPT GSAT $\in \Sigma_{2,1}$-OPT.*

Proof: Let the universe consist of a symbol for each clause, each group, and each Boolean variable. Encode by Gxy that clause x contains group y, and by Pyz (resp. Nyz) that variable z appears positively (resp. negatively) in group y. The problem can now be formulated by: $\text{opt}_{\mathcal{T}}|\{x : \exists y \forall z (\mathcal{G}xy \wedge ((\mathcal{P}yz \Rightarrow \mathcal{T}z) \wedge (\mathcal{N}yz \Rightarrow \neg\mathcal{T}z)))\}|$. $\Box$

We profited here from the fact, that for this problem $\Phi^{Sol} =$ "true" and we therefore have $\Pi = (\mathbf{I}, \Phi^{Val}(x))$.

Theorem 1 *OPT GSAT is $\Sigma^{(+)}_{2,1}$-OPT-complete under $S^=$-reduction.*

Proof: Assume a problem $\text{opt}_{\mathcal{S}}|\{x : \Phi(x, y, z, \mathcal{G}, \mathcal{S})\}|$ where $\Phi := \exists y \forall z \phi(x, y, z, \mathcal{G}, \mathcal{S})$, and $\phi \in \Sigma^+_{2,1}$ is quantifier-free. We will denote the atomic literals of $\mathcal{G}$ by $\gamma(x, y, z^\lambda)$ and those of $\mathcal{S}$ by $\sigma(x, y, z^\lambda)$, not specifying, which x_i, y_j and z_l^λ are really arguments; x is a m-tuple, y is a n-tuple, and z^λ are r^λ-tuples.

Now we will describe a pair of transformations (f, g) for an S$^=$-reduction. We know, that we can change ϕ into an equivalent formula of the required type. This can be done in a constant number of steps, because Φ is a fixed formula.

In our input structure G of Π, the number of possible values $u \in U^m$ for x is $|U|^m$ (m independent of G) and we derive a formula $\Phi(u, y, z)$ for each possible u, no longer dependent on x. In the next step, we want to write for each of the $|U|^n$ possible values $v \in U^n$ for y a different formula and we can verify (remember the definition of $\forall z$):

$$\Phi(u, y, z) \Leftrightarrow \bigvee_{v \in U^n} \forall z^1 \cdots \forall z^k \bigvee_{\lambda=1}^{k} \bigwedge_{\mu=1}^{h^\lambda} (\sigma_\mu^\lambda(u, v, z^\lambda) \vee \bigvee_{\nu=1}^{g_\mu^\lambda} \gamma_{\mu\nu}^\lambda(u, v, z^\lambda)).$$

We will use the fact, that for each $\lambda_0 \leq k$ we know that the atomic literals $\sigma_\mu^{\lambda_0}$ and $\gamma_{\mu\nu}^{\lambda_0}$ are independent of each variable tuple z^λ where $\lambda \neq \lambda_0$. Further, we will replace the $\forall z^\lambda$ quantifiers by conjunctions over all $|U|^{r^\lambda}$ possible values $w^\lambda \in U^{r^\lambda}$ and derive

$$\Phi(u, y, z) \Leftrightarrow \bigvee_{v \in U^n} \bigvee_{\lambda=1}^{k} \bigwedge_{w^\lambda \in U^{r^\lambda}} \bigwedge_{\mu=1}^{h^\lambda} (\sigma_\mu^\lambda(u, v, w^\lambda) \vee \bigvee_{\nu=1}^{g_\mu^\lambda} \gamma_{\mu\nu}^\lambda(u, v, w^\lambda)).$$

As G is given, the truth value of each $\gamma_{\mu\nu}^\lambda(u, v, w^\lambda)$ is in {true, false} and therefore constant. We will remove all these constants and call the resulting formulas Φ_u. This removal can be done in a polynomial number of steps, because the total number of atomic literals is polynomially bounded. All remaining atomic literals in Φ_u are literals of the form $\sigma_\mu^\lambda(u, v, w^\lambda)$ and we can view these literals as literals of a set B of Boolean variables. Each S describes a truth assignment to these Boolean variables. Altogether, this describes the transformation f, where $f(G) := (B, F)$ is an instance of OPT GSAT, and the set of Boolean formulas is $F := \{\Phi_u : u \in U^m\}$.

Transformation g is obvious: A solution for $f(G)$ is a truth assignment T, and we use set B and T to define a structure S by setting the corresponding atomic formula of $\sigma_\mu^\lambda(u, v, w^\lambda)$ to "true", if the corresponding Boolean variable is in T, otherwise to "false". It is clear by our construction that f and g are both polynomial time algorithms, and (f, g) is an S$^=$-reduction, as for each instance G, $/S/ = /T/$. Lemma 4 completes the proof.□

4.2 Complete problems of $\Pi_{1,1}^{(+)}$-OPT

We will show that MAX-k-GSAT and MAXIMUM INDEPENDENT SET (MIS), as well as RMAX(2)-complete problems presented also in [PR90], are complete in $\Pi^{(+)}_{1,1}$-MAX.

MIS: An instance is an undirected graph. Find a subset I of the nodes of maximum cardinality, called *independent set*, where no pair of nodes of I is adjacent.

Max k-SC, Max SC: Given a collection of sets, find a subcollection of pairwise nonintersecting sets, such that the number of these sets is maximum. For Max k-SC, each set has cardinality k.

Max k-DM, Max DM: Given a collection of tuples, find a subcollection of tuples, component-wise different in pairs, such that the number of these tuples is maximum. Max k-DM is the restriction to k-tuples.

Max Graph k-Coloring: Given an undirected graph, find a maximum subset of nodes and a color-assignment to it, so that adjacent nodes of this subset have different color. The number of allowed colors is k.

Max k-Anlsat: Given a Boolean CNF formula of negative literals only and at most k literals per clause, find a satisfying truth assignment maximizing the number of variables which are set to true.

Max k-Hyperclique, Max Clique: Given a hypergraph with rank bounded by k. Find a subset of nodes with maximum cardinality, so that for each set of at most k nodes in the subset, there is a hyperedge. Max Clique is the restriction to rank 2.

Lemma 5 *Max k-GSAT, MIS, Max k-SC, Max SC, Max k-DM, Max DM, Max Clique, Max Graph k-Coloring $\in \Pi_{1,1}$-MAX for any k, and Max k-Anlsat $\in \Pi_{1,k}$-MAX and, Max k-Hyperclique $\in \Pi_{1,k-1}$-MAX.*

Proof: For MAX k-GSAT, let the universe consist of a symbol for each group and each Boolean variable. $Cx_1 \cdots x_k$ is "true" if $x_1 \cdots x_k$ are groups, building a clause (respect an arbitrary but fixed ordering of the groups in a clause). Px_iz (resp. Nx_iz) encodes that variable z appears positively (resp. negatively) in group x_i. Then express MAX k-GSAT as $\max_{\mathcal{T}} |\{x_1 \cdots x_k : \forall z(\mathcal{C}x_1 \cdots x_k \wedge \bigvee_{i=1}^{k}((\mathcal{P}x_iz \Rightarrow \mathcal{T}z) \wedge (\mathcal{N}x_iz \Rightarrow \neg\mathcal{T}z)))\}|$. Building the CNF of the quantifier-free part, we can verify this to be of the required form.

For MIS, let the universe contain a symbol for each node, and let Exy encode adjacency: $\max_{\mathcal{I}} |\{x : \forall y(\mathcal{I}x \wedge (\mathcal{I}y \Rightarrow \neg\mathcal{E}xy)\}|$.

If we encode Max k-SC, Max SC, the universe containes a symbol for each set and each element, Sx encodes the sets, $y \in x$ that y is an element of set x. So we can write $\max_{\mathcal{C}} |\{x_1 : \mathcal{C}x_1 \wedge \mathcal{S}x_1 \wedge \forall x_2 y((\mathcal{C}x_2 \wedge \mathcal{S}x_2) \Rightarrow \neg(y \in x_1 \wedge y \in x_2))\}|$.

For MAX k-DM, MAX DM, let the universe contain a symbol for each tuple, for each component of each tuple, and for each component number starting from 1 to the maximal number of the largest input tuple. Tx encodes the tuples, Pi the component numbers, $y_1 \neq y_2$ the nonequality of the components, and $Cixy$ is "true", if the i-th component of x is y. Hence, we get $\max_{\mathcal{M}} |\{x_1 : \mathcal{M}x_1 \wedge \mathcal{T}x_1 \wedge \forall x_2 y_1 y_2 i((\mathcal{M}x_2 \wedge \mathcal{T}x_2 \wedge \mathcal{P}i \wedge \mathcal{C}ix_1y_1 \wedge \mathcal{C}ix_2y_2) \Rightarrow y_1 \neq y_2)\}|$.

To encode MAX GRAPH k-COLORING, let the universe contain a symbol for each node and each color. Pxy means x is a node and y is a color, the edges are given by Ex_1x_2, and $y_1 \neq y_2$ represents the nonequality of colors. Then we get $\max_{\mathcal{A}} |\{x_1 : \mathcal{A}x_1y_1 \wedge \mathcal{P}x_1y_1 \wedge \forall x_2y_2((y_1 \neq y_2 \Rightarrow \neg\mathcal{A}x_1y_2) \wedge ((\mathcal{A}x_2y_2 \wedge \mathcal{P}x_2y_2 \wedge \mathcal{E}x_1x_2) \Rightarrow y_1 \neq y_2))\}|$.

For MAX k-ANLSAT, we use $Cx_1 \cdots x_k$ to encode that $x_1 \cdots x_k$ are variables building a clause, and we repeat a variable if necessary, yielding $\max_{\mathcal{T}} |\{x : \mathcal{T}x \wedge \forall x_1 \cdots x_k(\mathcal{C}x_1 \cdots x_k \Rightarrow (\neg\mathcal{T}x_1 \vee \cdots \vee \neg\mathcal{T}x_k))\}|$.

For MAX k-HYPERCLIQUE, let $Ex_1 \cdots x_k$ be true, if there is a hyperedge connecting exactly the tuple of nodes $x_1 \cdots x_k$; this leads to $\max_{\mathcal{C}} |\{x_1 : \mathcal{C}x_1 \wedge \forall x_2 \cdots x_k((\mathcal{C}x_2 \wedge \cdots \wedge \mathcal{C}x_k) \Rightarrow \mathcal{E}x_1 \cdots x_k)\}|$. □

Lemma 6 *For each positive k, MAX k-GSAT $\leq_C$ MIS, MAX 1-GSAT $\leq_{S^=}$ MIS, and MIS $\leq_{S^=}$ MAX k-GSAT.*

Proof: (" $\leq_C$ ") Let $C := (B, F)$ be an instance of MAX k-GSAT. We will give a C-reduction (f, g, ϵ) and define $G := (V, E) := f(C)$ and $T_I := g(C, G, I)$.

We describe transformation f: First a new formula F' is constructed by deleting each group of F for which there is a Boolean variable with the positive and the negative literal appearing simultaneously. Now consider groups of F' only.

For each group g_i, a different node v_{g_i} is constructed. Let V be the set of all these nodes. For each pair of groups (g_i, g_j), we compare literals. If there is a literal in group g_i, appearing negated in group g_j, then we get an edge (v_{g_i}, v_{g_j}) between the related nodes v_{g_i} and v_{g_j}. Otherwise we have no edge, saying, that both groups *agree*. All these edges form the set E.

Transformation g works as follows: Starting from an independent set I, consider all groups g_i, where the related node v_{g_i} is in I. Let T_I be the set of all Boolean variables appearing as a positive literal in at least one of these groups g_i.

Given an independent set I, the related group g_i of each node $v_{g_i} \in I$ is satisfied by T_I, because all these groups have to agree in pairs (see the construction of E), so the groups agree as a whole. As each group corresponds to one and only one node, each node $v_{g_i} \in I$ corresponds to one satisfied group g_i. One satisfied group can satisfy a whole clause, but in the worst case, we do need k satisfied groups for each clause.

Let I_T be the set of all nodes v_{g_i}, where g_i is a group satisfied by T. I_T is an independent set, because each pair of satisfied groups (g_i, g_j) has to agree and so there is no edge between v_{g_i} and v_{g_j}. For each satisfied clause, there is at least one satisfied group, and for each satisfied group we have a different node in I.

Altogether this implies $/T^*/ \leq /I_{T^*}/ \leq /I^*/$ and $/I/ \leq k/T_I/$ and thereby $\frac{/T^*/}{/T_I/} \leq \frac{/I^*/}{\frac{1}{k}/I/} \leq k(1+\epsilon') = \epsilon(\epsilon')$. Because f, g, ϵ are computable in polynomial time, this yields MAX k-GSAT $\leq_C$ MIS and especially MAX 1-GSAT $\leq_{S^=}$ MIS.

(" $\leq_{S^=}$ ") We reduce MIS to MAX 1-GSAT by a new (f, g). Let $C := (B, F) := f(G)$ and $I_T := g(G, C, T)$.

For transformation f, let B contain for each node v of G one variable u_v. Furthermore for each node $v \in V$ one group g_v is built by $g_v := u_v \wedge \bar{u}_{v_1} \wedge \cdots \wedge \bar{u}_{v_{deg(v)}}$, where $v_1, \ldots, v_{deg(v)}$ are the adjacent nodes of v and $deg(v)$ is the number of adjacent nodes of v. In MAX 1-GSAT, each clause consists of exactly one group, so $F := \{g_v : v \in V\}$ is of the proper type.

For transformation g, the desired independent set I_T is the set of nodes v with its related variable u_v in the truth assignment.

It can easily be verified that (f, g) is an S$^=$-reduction. This completes the proof, as MAX 1-GSAT can be S$^=$-reduced trivially to each MAX k-GSAT with $k > 1$. □

Theorem 2 *The following problems are* $\Pi^+_{1,1}$*-MAX-complete under C-reduction and* $\Pi_{1,1}$*-MAX-complete under S*$^=$*-reduction for any k:* MAX k-GSAT*,* MIS*,* MAX SC*,* MAX DM*,* MAX CLIQUE*, and* MAX GRAPH k-COLORING*.*

Proof: As a corollary of Theorem 1, for each $\Pi \in \Pi^+_{1,1}$-MAX there is a $k \in \{1, 2, \ldots\}$, so that $\Pi \leq_{S^=}$ MAX k-GSAT; especially for $\Pi \in \Pi_{1,1}$-MAX we have $k = 1$. Using Lemma 6, the proposed completeness results for **MAX k-GSAT** and **MIS** are proved. MIS $\leq_{S^=}$ MAX CLIQUE is clear by using the co-graph. For the remaining S$^=$-reductions see [PR90]. □

Corollary 1 MIN k-GSAT *for any k is* $\Pi_{1,1}$*-MIN-complete under S*$^=$*-reduction.* □

5 Complete Problems of $\Pi_{i,j}$-OPT and $\Sigma_{i,j}$-OPT

To derive a complete problem for each class of each subhierarchy, let us now look at a more general version of OPT-GSAT.

OPT-$\Pi_{i,j}$-SAT, OPT-$\Sigma_{i,j}$-SAT: Assume arbitrary nonempty recursively defined index set, disjoint in pairs, I, I_ν $(\forall \nu \in I)$, I_{μ_1} $(\forall \mu_1 \in \bigcup_\nu I_\nu)$, $\ldots$, I_{μ_i} $(\forall \mu_i \in \bigcup_{\mu_{i-1}} I_{\mu_{i-1}})$, where only the cardinality of the last sets I_{μ_i} is bound by the problem parameter j. Let be $B = \{u_1, \ldots, u_m\}$ a finite set of Boolean variables, and $L = \{u_1, \bar{u}_1, \ldots, u_m, \bar{u}_m\}$ its set of literals. Let further $l_\lambda \in L$ $(\forall \lambda \in \bigcup_{\mu_i} I_{\mu_i})$. Then an instance of OPT-$\Pi_{i,j}$-SAT is a pair (B, F) with $F = \{c_\nu : \nu \in I\}$, where for problems with even i:

$$c_\nu := \bigwedge_{\mu_1 \in I_\nu} \cdots \bigvee_{\mu_i \in I_{\mu_{i-1}}} \bigwedge_{\lambda \in I_{\mu_i}} l_\lambda,$$

resp. with odd i:

$$c_\nu := \bigwedge_{\mu_1 \in I_\nu} \bigvee_{\mu_2 \in I_{\mu_1}} \cdots \bigwedge_{\mu_i \in I_{\mu_{i-1}}} \bigvee_{\lambda \in I_{\mu_i}} l_\lambda,$$

i.e. F is a set of Boolean formulas, so called *clauses*, with an alternation of junctors. We denote this clauses by the elements of I. Each set I_{μ_i} is called *subgroup* μ_i, i.e. the literals captured by the junctor with the index λ, and each set $I_{\mu_{i-1}}$ is called *group* μ_{i-1}, i.e. the subgroups captured by the junctor with the index μ_i.

We define OPT-$\Sigma_{i,j}$-SAT analogous: The clauses have to start with a disjunctor $\bigvee_{\mu_1}$, and the definition of the odd case for this of the even case is exchanged; i.e. the definitions are of the dual form.

It is interesting to identify MAX 3SAT = MAX-$\Sigma_{0,3}$-SAT, MAX SAT = MAX-$\Sigma_{0,\infty}$-SAT (= MAX-$\Sigma_{1,1}$-SAT) [Joh74], MAX GSAT(k) = MAX-$\Sigma_{1,k}$-SAT, MAX GSAT = MAX-$\Sigma_{1,\infty}$-SAT (= MAX-$\Sigma_{2,1}$-SAT) [PY88], MAX 1-GSAT = MAX-$\Pi_{0,\infty}$-SAT (= MAX-$\Pi_{1,1}$-SAT), and MAX NSF = MAX-$\Pi_{1,3}$-SAT [PR90].

Lemma 7 *OPT-$\Pi_{i,j}$-SAT $\in \Pi_{i,j}$-OPT, and OPT-$\Sigma_{i,j}$-SAT $\in \Sigma_{i,j}$-OPT, $i \geq 0$, $j \geq 1$.*

Proof: To encode an instance, use a finite $\{\mathcal{C}^{(1)}, \mathcal{S}^{(i)}, \mathcal{A}_0^{(j+1)}, \ldots, \mathcal{A}_{2^j-1}^{(j+1)}\}$-structure $(U; C, S, A_0, \ldots, A_{2^j-1})$ defined as follows:

$$U := B \cup I \cup \bigcup_\nu I_\nu \cup \bigcup_{\mu_i} I_{\mu_1} \cup \cdots \cup \bigcup_{\mu_{i-1}} I_{\mu_{i-1}}$$

For each subgroup μ_i, put its literals in an arbitrary but fixed linear order. If $|I_{\mu_i}| < j$, then repeat the last literal, to get a string of length j, called *characteristic string* of subgroup μ_i. Let $a_\alpha(\beta, z)$ (for appearance) of a Boolean variable z and integers $\alpha > 0$ and $\beta \geq 0$ be z if in the binary representation of β the α-th bit is 1, and $\bar{z}$ if this bit is 0 (count from the right to the left, starting with 1). Let $n_\alpha(\beta)\psi$ (for negation) for a formula ψ be ψ if in the binary representation of β the α-th bit is 1, and $\neg\psi$ if the bit is 0. We call a sequence $xy_1\cdots y_{i-1}$ of elements of the universe U *valid*, if $x, y_1, \ldots, y_{i-1} \in U - B$ and $xy_1\cdots y_{i-1} \in I \times I_x \times I_{y_1} \times \cdots \times I_{y_{i-2}}$. So let

$$\begin{array}{lcl} Cx & := & \text{"}x \text{ is a clause "}, \\ Sxy_1\cdots y_{i-1} & := & \text{"}xy_1\cdots y_{i-1} \text{ is a valid sequence "}, \\ A_\beta y_{i-1}z_1\cdots z_j & := & \text{"}y_{i-1} \text{ is a group } \wedge z_1,\ldots,z_j \in B \\ & & \wedge \text{ there is a subgroup of } y_{i-1}, \\ & & \text{with characteristic string } a_1(\beta,z_1)\cdots a_j(\beta,z_j)\text{"} \\ & & \text{ for } 0 \le \beta \le 2^j - 1. \end{array}$$

The truth assignments are obviously represented by a finite $\{\mathcal{T}^{(1)}\}$-structure T. As a further definition we use

$$T^\wedge_\beta z_1\cdots z_j := n_1(\beta)Tz_1 \wedge \cdots \wedge n_j(\beta)Tz_j,$$

and analogous $T^\vee_\beta$, using $\vee$ for $\wedge$. Now we are able to denote OPT-$\Pi_{i,j}$-SAT by $(\mathbf{I}, \Phi(x, \{\mathcal{C}^{(1)}, \mathcal{S}^{(i)}, \mathcal{A}_0^{(j+1)}, \ldots, \mathcal{A}_{2^j-1}^{(j+1)}\}, \{\mathcal{T}^{(1)}\}))$, where for problems with even i:

$$\begin{array}{rcl} \Phi & := & \forall y_1\cdots\exists y_{i-2}\forall y_{i-1}\exists z_1\cdots z_j(\mathcal{C}x \\ & \wedge & (\mathcal{S}xy_1\cdots y_{i-1} \Rightarrow \bigvee_{\beta=0}^{2^j-1}(\mathcal{A}_\beta y_{i-1}z_1\cdots z_j \wedge \mathcal{T}^\wedge_\beta z_1\cdots z_j))), \end{array}$$

resp. with odd i:

$$\begin{array}{rcl} \Phi & := & \forall y_1\cdots\exists y_{i-1}\forall z_1\cdots z_j(\mathcal{C}x \\ & \wedge & (\mathcal{S}xy_1\cdots y_{i-1} \Rightarrow \bigwedge_{\beta=0}^{2^j-1}(\mathcal{A}_\beta y_{i-1}z_1\cdots z_j \Rightarrow \mathcal{T}^\vee_\beta z_1\cdots z_j))), \end{array}$$

i.e. Φ has a prefix of alternate quantifiers. To see that for even i OPT-$\Pi_{i,j}$-SAT $\in \Pi_{i,j}$-OPT, note that we can change the quantifier-free part of Φ to disjunctive normal form, where no conjunction can contain more than j atomic formulas of the finite similarity type $\{\mathcal{T}^{(1)}\}$. For odd i, analogous argumentation with conjunctive normal form is possible.

For OPT-$\Sigma_{i,j}$-SAT a similar formulation is possible, where Φ starts with an $\exists y_i$ quantifier and exchanging the definition of the odd case for this of the even case. $\square$

Theorem 3 *The following problems are complete:*
OPT-$\Pi_{i,j}$-SAT $\in \Pi^+_{i,j}$-OPT-$c_{S=}$ and OPT-$\Sigma_{i,j}$-SAT $\in \Sigma^+_{i,j}$-OPT-$c_{S=}$, for $i \ge 2, j \ge 1$, and
OPT-$\Pi_{i,j}$-SAT $\in \Pi_{i,j}$-OPT-$c_{S=}$ and OPT-$\Sigma_{i,j}$-SAT $\in \Sigma_{i,j}$-OPT-$c_{S=}$, for $i \ge 1, j \ge 1$.

Proof: The proof is quit similar to this of Theorem 1, given for $\Sigma^+_{2,1}$-OPT. Assume a problem of $\Pi^+_{i,j}$-OPT for odd i (or a problem of $\Sigma^+_{i,j}$-OPT for even i), then the problem is of the form

$$\max_S |\{x : \Phi(x, y^1, \ldots, y^{i-1}, z, \mathcal{G}, \mathcal{S})\}|$$

$$\text{where } \Phi := \mathbf{P}(y^1, \ldots, y^{i-1})\forall z \phi(x, y^1, \ldots, y^{i-1}, z, \mathcal{G}, \mathcal{S})$$

and $\mathbf{P}$ is a prefix of $i-1$ alternating quantifiers ending with an existential quantifier. With quite analogous arguments to Theorem 1, we can change in polynomial steps each $\Phi(u, y^1, \ldots, y^{i-1}, z, \mathcal{G}, \mathcal{S})$ to an equivalent formula

$$\bigwedge_{v^1 \in U^{n^1}} (\bigvee) \cdots \bigvee_{v^{i-1} \in U^{n^{i-1}}} \bigvee_{\lambda=1}^{k} \bigwedge_{w^\lambda \in U^{r^\lambda}} \bigwedge_{\mu=1}^{h^\lambda} (\bigvee_{\nu=1}^{s^\lambda_\mu} \sigma^\lambda_{\mu\nu}(u, v, w^\lambda) \vee \bigvee_{\nu=1}^{g^\lambda_\mu} \gamma^\lambda_{\mu\nu}(u, v, w^\lambda)),$$

where all $s^\lambda_\mu \leq j$. It is obvious, how an $\mathrm{S}^=$-reduction is constructed analogous to the proof of Theorem 1.

For the remaining cases of $\Pi^+_{i,j}$-OPT and of $\Sigma^+_{i,j}$-OPT, the same proof is possible, exchanging all junctors $\wedge$ for $\vee$, and all quantifiers $\forall$ for $\exists$. $\Box$

Complete problems for $i = 0$ are not of particular interest here, because complete problems are known for the whole class MAX SNP. For $\Pi^+_{1,j}$-OPT, and $\Sigma^+_{1,j}$-OPT we would like to identify complete problems under C-reduction, similar to what we have found for $\Pi^+_{1,1}$-MAX in Theorem 2. Unfortunately, we have not been able to generalize the proof of Theorem 2 with a problem similar to MIS, on the base of generalizing MAX k-GSAT similar to MAX GSAT. Results similar to Corollary 1 can be achieved for this generalized MAX k-GSAT.

Further interesting results depend on the following. For any problem $\Pi = (\mathbf{I}, \Phi(x, \mathcal{G}, \mathcal{S}))$ of SO$\exists$-OPT, we can change the formula $\exists \mathcal{S} \Phi(x, \mathcal{G}, \mathcal{S})$ of SO$\exists$ to an equivalent formula $\exists \mathcal{S}' \Phi'(x, \mathcal{G}', \mathcal{S}')$, where Φ' is in the prefix class Π_2 and does not have function symbols and equality symbol. This can be done in a standard way by introducing Skolem functions for the existential quantifiers and then simulating all functions and the equality symbol by predicates (see e.g. [End72]). Since rewriting the formula is independent of structures, it takes only a constant number of steps. For the related problem $\Pi' = (\mathbf{I}', \Phi'(x, \mathcal{G}', \mathcal{S}'))$ of Π_2-OPT, the described transformation is an $\mathrm{S}^=$-reduction from Π to Π', as inputs G and G' as well as solutions S and S' are polynomially related, because the number of functions and predicates and its arities are instance-independent.

With Lemmas 2 and 3, this implies:

Theorem 4 *For* $i \geq 3, j \geq 1$*:*
SO$\exists$-OPT-$c_{S^=}$ $\supseteq$ $\Pi^+_{i,j}$-OPT-$c_{S^=}$ and SO$\exists$-OPT-$c_{S^=}$ $\supseteq$ $\Sigma^+_{i+1,j}$-OPT-$c_{S^=}$.

The theorem holds for completeness under C-reduction, too. This theorem is especially interesting, as it shows that the entire class SO∃-OPT containing all its hierarchies and subhierarchies does actually possess an infinite number of complete problems. (Note: Our SO∃ was defined by formulas with all its second-order quantifiers out front. This is no real restriction, as we can change in constant steps each second-order formula to an equivalent prenex normal form.)

In the other direction, we can only prove hardness results, because SO∃-OPT $\not\subseteq \Pi^+_{i,j}$-OPT (resp. $\Sigma^+_{i,j}$-OPT) for finite i and j.

In the definition of the classes of our subhierarchies, we used an equivalence relation between optimization problems of SO∃-OPT based on the equivalence of its first-order part. One might use an equivalence relation based on $S^=$-reduction to define each (sub)hierarchy. This, however, clearly forces this hierarchy to collapse; then, problems related by $S^=$-reduction, like the problems of Theorem 2, are viewed as the same problem.

References

[BP89] Marshall Bern and Paul Plassmann. The Steiner problem with edge lengths 1 and 2. *Information Processing Letters*, 32:171–176, 1989.

[BS89] Piotr Berman and Georg Schnitger. On the complexity of approximating the independent set problem. In *Proc. 6th Ann. Symp. on Theoretical Aspects of Computer Science*, pages 256–268, 1989.

[End72] Herbert B. Enderton. *A Mathematical Introduction to Logic.* Academic Press, 1972.

[Fag74] Ronald Fagin. *Generalized First-Order Spectra, and Polynomial-time Recognizable Sets, in R. Karp (ed.): Complexity of Computation.* AMS, 1974.

[GJ76] Michael R. Garey and David S. Johnson. The complexity of near optimal graph coloring. *J. ACM*, 23:43–49, 1976.

[GJ79] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman and Company, 1979.

[Ihl90] Edmund Ihler. Bounds on the quality of approximate solutions to the Group Steiner Problem. In *16th International Workshop on Graph-Theoretic Concepts in Computer Science*, 1990.

[Imm89] Neil Immerman. Descriptive and computational complexity. *Proceedings of Symposia in Applied Mathematics*, 38:75–91, 1989.

[Joh74] David S. Johnson. Approximation algorithms for combinatorial problems. *J. Comp. Sys. Sc.*, 9:256–278, 1974.

[Kre88] Mark W. Krentel. The complexity of optimization problems. *J. Comp. Sys. Sc.*, 36:490–509, 1988.

[PR90] Alessandro Panconesi and Desh Ranjan. Quantifiers and approximation. In *Proc. 22th Annual ACM Symp. on Theory of Computing*, pages 446–456, 1990.

[PY88] Christos H. Papadimitriou and Mihalis Yannakakis. Optimization, approximation, and complexity classes. In *Proc. 20th Annual ACM Symp. on Theory of Computing*, pages 229–234, 1988.

[PY89] Christos H. Papadimitriou and Mihalis Yannakakis. The traveling salesman problem with distances one and two. Manuscript, 1989.