CHICAGO JOURNAL OF THEORETICAL COMPUTER SCIENCE 2008, Article 3, pages 1–24
http://cjtcs.cs.uchicago.edu/

Syntactic Characterizations of Polynomial Time Optimization Classes

Prabhu Manyem

Received: January 11, 2007; published: May 22, 2008.

Abstract: The characterization of important complexity classes by logical descriptions has been an important and prolific area of Descriptive complexity. However, the central focus of the research has been the study of classes like **P**, **NP**, **L** and **NL**, corresponding to decision problems (e.g. the characterization of **NP** by Fagin [5] and of **P** by Grädel [7]). In contrast, optimization problems have received much less attention. Optimization problems corresponding to the **NP** class have been characterized in terms of logic expressions by Papadimitriou and Yannakakis, Panconesi and Ranjan, Kolaitis and Thakur, Khanna et al, and by Zimand. In this paper, we attempt to characterize the optimization versions of **P** via expressions in second order logic, many of them using universal Horn formulae with successor relations. These results nicely complement those of Kolaitis and Thakur [13] for polynomially bounded NP-optimization problems. The polynomially bounded versions of maximization and minimization problems are treated first, and then the maximization problems in the "not necessarily polynomially bounded" class.

Key words and phrases: Descriptive Complexity, Finite Model Theory, Syntactic Hierarchy, Optimization, Quantifier Complexity, Quantifier Alternation, Existential Second Order Logic

1 Introduction

Though there has been abundant research in Descriptive Complexity since Fagin's 1974 theorem [5] (which captures the class **NP** as precisely the set of properties that can be represented in existential second order logic), the application of this area to approximation complexity has been limited. Approximation complexity measures how well an NP-hard optimization problem can be approximated, or how far is the value of a (possible) heuristic solution from that of an optimal solution.

DOI: 10.4086/cjtcs.2008.003

A few attempts to characterize approximation classes in terms of logic are: Papadimitriou and Yannakakis in 1991 [17], Panconesi and Ranjan in 1993 [15], Kolaitis and Thakur in 1994 and 1995 [13, 14], and Khanna et al in 1998 [11].

The approximation complexity of a problem P is usually measured by the *approximation ratio* that a heuristic H for P can guarantee, over all instances of P. The approximation ratio $R_H(I)$ obtained by H for a given instance I of a maximization problem P is given by

$$R_H(I) = \frac{\text{value obtained by } H \text{ on } I}{\text{value of an optimal solution for } I}. \tag{1.1}$$

For a minimization problem, $R_H(I)$ is the reciprocal of the formula in (1.1). Note that regardless of whether the problem is maximization or minimization, $R_H(I) \geq 1$ always.

In [13, 15, 17], the authors characterize approximation hardness in terms of quantifier complexity — the number and types of quantifiers that appear at the beginning of a second-order formula in prenex normal form (PNF). For a formula in PNF, all quantifiers appear at the beginning, followed by a quantifier-free formula.

Grädel [7] presented a syntactic characterization of **P** (given below in Theorem 1.12). Results complementing those in this paper have appeared in Bueno and Manyem [2].

1.1 Contributions

In this paper, we first present a logical representation of a subclass of $\mathbf{P_{opt}}$, which is the class of optimization problems that can be solved to optimality within polynomial time[1] (see Table 1 and Section 1.2 for all notation and definitions). The class of *decision problems* corresponding to $\mathbf{P_{opt}}$ is the well-studied class **P**. The particular subclass $\mathbf{P^{pb}_{opt}}$ (of $\mathbf{P_{opt}}$) that we focus on includes only *polynomially bounded optimization problems*, defined below in Definition 1.4. In particular, we

- provide syntactic characterizations for both maximization and minimization problems in $\mathbf{P^{pb}_{opt}}$ (Section 2). Results are summarized in Table 3;
- provide expressions for problems in the class **P** that have a maximization counterpart (Section 2.1.1);
- give examples of characterizations, namely MAXFLOW$_{\text{PB}}$ for maximization, and SHORTEST PATH$_{\text{PB}}$ minimization (Sections 2.1.2 and 2.2.1). The subscript *PB* refers to the *polynomially bounded* versions of these two problems, i.e. the versions where the optimal solution value is polynomially bounded in the size of the input;
- show that MAXFLOW$_{\text{PB}}$ is complete for the maximization subclass of $\mathbf{P^{pb}_{opt}}$ (Section 2.1.3);

[1]Strictly speaking, in Turing machine terminology, $\mathbf{P_{opt}}$ is the set of languages where, if an instance I of an optimization problem $P \in \mathbf{P_{opt}}$ is encoded as an input string x in some alphabet Σ, a deterministic Turing machine will compute the optimal solution (which is again a string) within $\Theta(|x|^k)$ steps, where k is some constant and $|x|$ is the length of the input string.

σ	vocabulary or signature
$\mathbf{A}$	a structure defined over σ (captures an instance of an optimization problem)
θ	a quantifier-free first order (FO) conjunction of *Horn* clauses. (Recall that a Horn clause contains at most one positive literal.)
$\mathbf{x}$	an $m-$tuple of first order (FO) variables
$\mathbf{S}$	a sequence of second-order variables (SO) (predicate symbols) (captures a solution to the optimization problem)
P (**NP**)	computational class of *decision* problems, decidable in polynomial time by a deterministic (non-deterministic) Turing machine
$\mathbf{P_{opt}}$ ($\mathbf{NP_{opt}}$)	**P**-optimization (**NP**-optimization) problems. See Definition 1.1 (1.2).
$\mathbf{P^{pb}_{opt}}$ ($\mathbf{NP^{pb}_{opt}}$)	Polynomially bounded **P**-optimization (**NP**-optimization) problems. See Definition 1.4.
ESO	Existential Second Order Logic
PNF	Prenex Normal Form

Table 1: Notation

- present characterizations in Section 3 for maximization problems in $\mathbf{P_{opt}}$ (defined in Table 1 — problems not necessarily polynomially bounded), as well as an example for a problem in this class (MAXIMUM MATCHING). This is the most significant contribution in this paper, and is a considerable departure from the treatment in Zimand [18]. Zimand's paper has a similar motivation, but it studies $\mathbf{NP_{opt}}$, while we characterize maximization problems in $\mathbf{P_{opt}}$.

1.2 Notation and Definitions

All notation is defined in Table 1, as a one-stop reference point. For the same reason, all definitions are provided in this subsection. The interested reader can also refer to standard texts on Logic such as [4] or [3]. Ausiello et al [1] is an excellent reference on optimization problems and their approximability.

Definition 1.1. A **P-optimization** problem Q is a set $Q = \{I_{\mathrm{Q}}, F_{\mathrm{Q}}, f_{\mathrm{Q}}, opt_{\mathrm{Q}}\}$, where

(i) I_{Q} is a set of instances to Q,

(ii) $F_{\mathrm{Q}}(I)$ is the set of feasible solutions to instance I (a feasible solution is one which obeys all the constraints to an optimization problem — but not necessarily constraints of the type $f_{\mathrm{Q}}(I,S) \geq k$ or $f_{\mathrm{Q}}(I,S) \leq k$ referred to in (v) below),

(iii) $f_{\mathrm{Q}}(I,S)$ is the *objective function* value to a solution $S \in F_{\mathrm{Q}}(I)$ of an instance $I \in I_{\mathrm{Q}}$. It is a function $f : \bigcup_{I \in I_{\mathrm{Q}}}[I \times F_{\mathrm{Q}}(I)] \rightarrow R^+$ (non-negative reals), computable in time polynomial in $|I|$, where $|I|$ is the length of the representation of I,

(iv) For an instance $I \in I_{\mathrm{Q}}$, $\mathrm{opt}_{\mathrm{Q}}(I)$ is either the minimum or maximum possible value that can be obtained for the objective function, taken over all feasible solutions in $F_{\mathrm{Q}}(I)$.

$\mathrm{opt}_{\mathrm{Q}}(I) = \max_{S \in F_{\mathrm{Q}}(I)} f_{\mathrm{Q}}(I,S)$ (for P-maximization problems),

$\mathrm{opt}_{\mathrm{Q}}(I) = \min_{S \in F_{\mathrm{Q}}(I)} f_{\mathrm{Q}}(I,S)$ (for P-minimization problems),

(v) The following decision problem is in the class **P**: *Given an instance I and a non-negative constant k, is there a feasible solution $S \in F_{\mathrm{Q}}(I)$, such that $f_{\mathrm{Q}}(I,S) \geq k$ (for a P-maximization problem), or $f_{\mathrm{Q}}(I,S) \leq k$ (in the case of a P-minimization problem)?*

And finally,

(vi) **(condition to be removed ??)** An optimal solution $S_{\mathrm{opt}}(I)$ for a given instance I can be computed[2] in time polynomial in $|I|$, where $\mathrm{opt}_{\mathrm{Q}}(I) = f_{\mathrm{Q}}(I, S_{\mathrm{opt}}(I))$.

The set of all such **P**-optimization problems is the $\mathbf{P_{opt}}$ class.

A similar definition, for NP-optimization problems, appeared in Panconesi and Ranjan (1990) [15]:

Definition 1.2. An **NP-optimization** problem is defined as follows. Points *(i)-(iv)* in Definition 1.1 above apply to NP-optimization problems, whereas *(vi)* does not. Point *(v)* is modified as follows:

(v) The following decision problem is a member of **NP**: *Given an instance I and a non-negative constant k, is there a feasible solution $S \in F_{\mathrm{Q}}(I)$, such that $f_{\mathrm{Q}}(I,S) \geq k$ (for an NP-maximization problem), or $f_{\mathrm{Q}}(I,S) \leq k$ (in the case of an NP-minimization problem)?*

The set of all such **NP**-optimization problems is the $\mathbf{NP_{opt}}$ class, and $\mathbf{P_{opt}} \subseteq \mathbf{NP_{opt}}$.

Definition 1.3. *Problems in* **P** *(***NP***) that have an optimization counterpart* are the ones described in item *(v)* in Definition 1.1 (Definition 1.2). Call this class as $\mathbf{P_{oc}}$ ($\mathbf{NP_{oc}}$). There is a one-to-one correspondence between the decision problems in $\mathbf{P_{oc}}$ ($\mathbf{NP_{oc}}$) and the optimization problems in $\mathbf{P_{opt}}$ ($\mathbf{NP_{opt}}$).

Definition 1.4. An optimization problem Q is said to be **polynomially bounded** if the value of an optimal solution to every instance I of Q is bound by a polynomial in the size of I. In other words, there exists a polynomial p such that

$$\mathrm{opt}_{\mathrm{Q}}(I) \leq p(|I|), \tag{1.2}$$

for every instance I of Q. $\mathbf{P^{pb}_{opt}}$ ($\mathbf{NP^{pb}_{opt}}$) is the set of polynomially-bounded **P**-optimization (**NP**-optimization) problems. Naturally, $\mathbf{P^{pb}_{opt}} \subseteq \mathbf{P_{opt}}$ and $\mathbf{NP^{pb}_{opt}} \subseteq \mathbf{NP_{opt}}$.

$\mathbf{P^{pb}_{opt}}$ consists of two subclasses: maximization and minimization problems. Since we will show in the next section that both these subclasses can be represented by formulas where the first order part is Horn Π_1, we can name these two subclasses as $MAX_{\mathrm{P}}\Pi_1$ and $MIN_{\mathrm{P}}\Pi_1$ respectively.

[2]Suppose an algorithm $\mathcal{A}$ that computes an optimal solution $S_{\mathrm{opt}}(I)$, visits other solutions before arriving at $S_{\mathrm{opt}}(I)$. Then $\mathcal{A}$ must be able to verify the feasibility of such solutions in time polynomial in $|I|$. That is, for any solution T visited, $\mathcal{A}$ must be able to verify whether $T \in F_Q(I)$ in time polynomial in $|I|$.

Definition 1.5. First order logic consists of a *vocabulary* (alias signature) σ, and *structures* defined on the vocabulary. In its simplest form, a vocabulary consists of a set of variables, and a set of relation symbols $R_j(1 \leq j \leq J)$, each of arity r_j. A structure M consists of a universe U whose elements are the values that variables can take — M also instantiates each relation symbol $R_j \in \sigma$ with tuples from $U^{(r_j)}$. Once a structure $\mathcal{A}$ satisfies a formula ϕ (written as $\mathcal{A} \models \phi$), $\mathcal{A}$ is said to *model* ϕ, or, $\mathcal{A}$ is a *model* for ϕ.

For example, a structure **G** in graph theory may have the set of vertices $G = \{1, 2, \cdots 10\}$ as its universe (assuming that the graph has 10 vertices), and a single binary relation E where $E(i, j)$ is true iff (i, j) is an edge in the graph **G**. A structure represents an instance of an optimization problem.

Definition 1.6. For a formula to be in **prenex normal form (PNF)**, all quantifiers appear at the beginning, followed by a quantifier-free formula.

Definition 1.7. A **$\blacksquare_1$ ($\blacksquare_1$) first order formula in PNF** only has universal (existential) quantifiers that range over first order variables.

Definition 1.8. A **Horn clause** is a disjunction of one or more literals, at most one of which is positive. For example, $x_1, \bar{x}_1$ and $(\bar{x}_1 \vee x_2)$ are all Horn clauses, whereas $(x_1 \vee x_2)$ is not.

Definition 1.9. An **existential second-order (ESO) Horn** expression is of the form $\exists \mathbf{S} \psi$, where ψ is a first order formula, and $\mathbf{S} = (S_1, \; \cdots \; S_p)$ is a sequence of predicate symbols not in the vocabulary of ψ. The formula ψ can be written in Π_1 form as

$$\psi = \forall x_1 \forall x_2 \cdots \forall x_k \eta = \forall \mathbf{x} \; \eta. \tag{1.3}$$

where η is a conjunction of Horn clauses (η is, of course, quantifier-free), and x_i $(1 \leq i \leq k)$ are first order variables. Each clause in η contains at most one positive occurrence of any of the second order predicates S_i $(1 \leq i \leq p)$.

A general **ESO** formula is the same as an ESO Horn expression, except that η can now be any quantifier-free first order formula.

Definition 1.10. A **$\blacksquare_2$ ($\blacksquare_2$) formula in PNF** is one that has the following form:

$$\phi = \forall x_1 \cdots \forall x_a \, \exists y_1 \cdots \exists y_b \; \eta \quad (\phi = \exists y_1 \cdots \exists y_b \, \forall x_1 \cdots \forall x_a \; \eta), \tag{1.4}$$

where η is quantifier-free, $a, b \geq 1$, and the x's and y's are first-order variables.

Definition 1.11. A **successor relation** $succ(a, b)$, where $a \neq b$, denotes that
(i) a immediately precedes b (or b immediately succeeds a) in A, where A = universe of a structure **A**,
(ii) $\forall c \in A$, where a, b and c are distinct, $\neg succ(a, c) \wedge \neg succ(c, b)$, and
(iii) $\forall c \in A, \neg succ(c, c)$.

Informally, b occurs "next" to a in A according to the above definition — a and b are "adjacent" to each other.

An **ordering** $(<)$ is different from a successor relation. Ordering (a,b) only denotes that a occurs "before" b in the universe — it does not imply that b immediately follows a. However, given a complete ordering between every pair of elements in A, successor relations between any pair of elements can be constructed:

$$\begin{aligned} succ(x,y) &\equiv \forall z\ (x<y) \wedge \neg(x<z \wedge z<y) \\ &\equiv \forall z\ (x<y) \wedge [\neg(x<z) \vee \neg(z<y)]. \end{aligned} \tag{1.5}$$

We assume that the structures we work with are ordered, and come with the explicit successor relation $succ(x,y)$ in their first order vocabularies. Hence we will not need expression (1.5) above. The following theorem is due to Grädel [7]. It characterizes **P** as the class of decision problems definable by ESO universal Horn formulae. This is the deterministic counterpart of Fagin's theorem [5] which characterized the class **NP** as precisely the set of properties that can be represented in ESO logic.

Theorem 1.12. *For any ESO Horn expression as defined in Definition 1.9, the corresponding decision problem is a member of* **P**.

The converse is also true — if a problem P is a member of **P**, *then it can be expressed in ESO Horn form — but only if a successor relation is allowed to be included in the vocabulary of the first-order formula ψ.*

2 Polynomially Bounded P-Optimization Problems

We will study maximization problems first, and then the minimization problems. Unless otherwise mentioned, we will denote decision problems by capital letters, and their optimization versions by a prime. Thus Q' is the optimization problem corresponding to the decision problem Q.

2.1 Polynomially Bounded P-Maximization Problems

For polynomially bounded NP-maximization problems (the ones in $\mathbf{NP}^{\mathbf{pb}}_{\mathbf{opt}}$, see Table 1 for definitions), Kolaitis and Thakur [13] proved the following:

Theorem 2.1. *A polynomially bounded NP-maximization problem $Q' \in \mathbf{NP}^{\mathbf{pb}}_{\mathbf{opt}}$ if and only if there exists a Π_2 first order formula $\phi(\mathbf{w},\mathbf{S})$ with predicate symbols from the vocabulary σ (of ϕ) and the sequence* **S**, *such that for every instance* **A** *of Q', the optimal solution value is given by*

$$\mathrm{opt}_{Q'}(\mathbf{A}) = \max_{\mathbf{S}} |\{\mathbf{w} : (\mathbf{A},\mathbf{S}) \models \phi(\mathbf{w},\mathbf{S})\}|. \tag{2.1}$$

In other words, polynomially bounded NP-maximization problems fall in what is called the MAX Π_2 class. We can show a similar result for the polynomial-time counterpart of $\mathbf{NP}^{\mathbf{pb}}_{\mathbf{opt}}$, that is, polynomially bounded P-maximization problems (the ones in $\mathbf{P}^{\mathbf{pb}}_{\mathbf{opt}}$):

Theorem 2.2. *Let Q' be a maximization problem in the class $\mathbf{P}^{\mathbf{pb}}_{\mathbf{opt}}$. Also, let $\mathbf{A}$ be a structure (instance) of Q' defined over vocabulary σ. Then the value of an optimal solution to $\mathbf{A}$ can be represented by*

$$\mathrm{opt}_{Q'}(\mathbf{A}) = \max_{\mathbf{S}} |\{\mathbf{w} : (\mathbf{A},\mathbf{S}) \models \forall \mathbf{x}\ \eta(\mathbf{w},\mathbf{x},\mathbf{S})\}| \tag{2.2}$$

where $\mathbf{x}$, $\mathbf{A}$, $\mathbf{S}$ *and* η *are defined in Table 1.*

Before we embark on proving Theorem 2.2, we should first observe that the problems described in Theorems 2.1 and 2.2 are necessarily polynomially bounded.

Let $|A|$ be the size of $\mathbf{A}$'s universe. The arity of the tuple $\mathbf{w}$ is some fixed constant, say $\mathcal{R}$. This means that the number of possible tuples $\mathbf{w}$ has an upper bound of $|A|^{\mathcal{R}}$. Consequently, the optimal values in (2.1) and (2.2) have an upper bound of $|A|^{\mathcal{R}}$. But $|A|^{\mathcal{R}}$ is a polynomial in $|A|$. Hence the corresponding optimization problems have to be polynomially bounded.

Proof. (of Theorem 2.2)

As usual, let Q and Q' be the decision and optimization versions respectively.

We should show that if $Q' \in \mathbf{P}^{\mathbf{pb}}_{\mathbf{opt}}$, then the optimal solution value to an instance $\mathbf{A}$ of Q' can be represented by equation (2.2).

Refer to Grädel's theorem (Theorem 1.12). The decision problem Q, as described in item (vi) in Definition 1.1, can be written as an ESO Horn expression $\exists \mathbf{S} \psi$, except that now, ψ should include a successor relation in its vocabulary, in addition to being a Horn first order formula. Problem Q can be posed as: Given an instance (a finite structure) $\mathbf{A}$, is there a feasible solution $\mathbf{S}$ such that $f(\mathbf{A},\mathbf{S}) \geq K$, where K is a certain integer?
(Here f is the value of the objective function to solution $\mathbf{S}$ for the optimization problem. Assume that we deal only with problems with integer-valued objective functions.)

A feasible solution $\mathbf{S}$ could consist of several relations $S_1, S_2, \cdots S_p$ of arities $r_1, r_2, \cdots r_p$. The formula ψ should be able to express $f(\mathbf{A},\mathbf{S}) \geq K$.

Let $g(\mathbf{A}) = f(\mathbf{A},\mathbf{S})$. Each of the $g(\mathbf{A})$ number of entities can be considered to be a tuple $\mathbf{w_i}$, and thus we need at least K such tuples. These tuples, at least K in number, can be defined to form a new relation F (on the universe A of $\mathbf{A}$) of arity k. Thus we want $|F|$, the number of tuples $\mathbf{w}$ that satisfy $F(\mathbf{w})$, to be at least K.

Digression to discuss arity k. As examples, setting $k = 2$ will suffice for the LONGEST PATH problem where the number of arcs in a path is to be maximized, and $k = 1$ in a MAXSAT problem where the number of satisfying clauses is to be maximized. However, this can handle only up to very small values of the objective function. In the LONGEST PATH case, we can only count $|A|^2$ tuples at most (where $|A|$ is the number of vertices in the graph). However, if arc lengths are higher than one, but still polynomially bound in $|A|$, the length of the longest path — though still polynomially bound in $|A|$ — could be well above $|A|^2$, and this length cannot be handled by an arity of $k = 2$ — a higher arity is required. Hence it would be safest to increase the arity to $\mathcal{R}$, since $|A|^{\mathcal{R}}$ is the upper bound on the objective function value. A similar argument applies to the weighted MAXSAT problem with polynomially bound weights — and to all polynomially bounded NP-maximization problems in general.

Recall from Theorem 1.12 that ψ is in Π_1 form, where the quantifier-free part of ψ is a conjunction of Horn clauses, each of which contains at most one positive occurrence of *any* of the relation symbols S_i. (For example, if S_1 and S_2 are second order predicates, then a Horn clause cannot contain both S_1 and S_2 as positive literals.) Hence ψ can be written as $\forall x_1, \cdots \forall x_m \hat{\eta}$, where $\hat{\eta}$ is an expression consisting of variables $x_1, \cdots x_m$, all predicates from $\mathbf{S}$, and the the relation F. That is,

$$\psi = \forall x_1 \cdots \forall x_m \; \hat{\eta}(x_1, \cdots, x_m, F, \mathbf{S}) = \forall \mathbf{x} \; \hat{\eta}(\mathbf{x}, F, \mathbf{S}), \tag{2.3}$$

where $\hat{\eta}$ is a conjunction of Horn clauses ($\hat{\eta}$ captures the feasibility of solution $\mathbf{S}$) and $\mathbf{x} = (x_1, \cdots, x_m)$. Furthermore, $\hat{\eta}$ needs to capture two types of conditions (an example with such conditions is provided in Section 2.1.3):

(a) *Global* conditions: Those that apply over all $\mathbf{w}$ tuples. These follow from item *(vi)* in Definition 1.1. Such conditions express the fact that (i) the solution ($\mathbf{S}$, F) as a whole is a feasible solution to $\mathbf{A}$, and that (ii) $|F| \geq K$.

(b) *Local* conditions: The ones that are specific to a given $\mathbf{w}$ — if $F(\mathbf{w})$ is true, that is.

Thus $\hat{\eta}$ is a conjunction of these two types of conditions. The global conditions[3] can be written as $\hat{\eta}_1$, and the local conditions as $\forall \mathbf{w} \; F(\mathbf{w}) \longrightarrow \hat{\eta}_2(\mathbf{x}, \mathbf{w}, \mathbf{S})$. Hence Q, the decision problem, can be written as $\exists \mathbf{S} \; \exists F \; \forall \mathbf{x} \; \hat{\eta}_1 \wedge [\forall \mathbf{w} \; F(\mathbf{w}) \longrightarrow \hat{\eta}_2(\mathbf{x}, \mathbf{w}, \mathbf{S})]$. In prenex normal form,

$$\begin{aligned} Q &\equiv \exists \mathbf{S} \; \exists F \; \forall \mathbf{w} \; \forall \mathbf{x} \; \hat{\eta}_1 \wedge [F(\mathbf{w}) \longrightarrow \hat{\eta}_2(\mathbf{x}, \mathbf{w}, \mathbf{S})] \\ &\equiv \exists \mathbf{S} \; \exists F \; \forall \mathbf{w} \; \forall \mathbf{x} \; \hat{\eta}_1 \wedge [\neg F(\mathbf{w}) \vee \hat{\eta}_2(\mathbf{x}, \mathbf{w}, \mathbf{S})] \end{aligned} \tag{2.4}$$

If $\hat{\eta}_1$ and $\hat{\eta}_2$ are each a conjunction of Horn clauses, then so is the formula in (2.4). If we let $\eta(\mathbf{x}, \mathbf{w}, \mathbf{S}, F) = \hat{\eta}_1 \wedge [\neg F(\mathbf{w}) \vee \hat{\eta}_2(\mathbf{x}, \mathbf{w}, \mathbf{S})]$, then (2.4) can be rewritten in ESO Horn Π_1 form as

$$Q \equiv \exists \mathbf{S} \; \exists F \; \forall \mathbf{w} \; \forall \mathbf{x} \; [\eta(\mathbf{x}, \mathbf{w}, \mathbf{S}, F)]. \tag{2.5}$$

To express the optimal solution value for Q', we maximize over all feasible solutions $\mathbf{S}$ — and for each solution, count the number of $\mathbf{w}$ tuples for which the relation $F(\mathbf{w})$ and $\psi(\mathbf{w}, \mathbf{S})$ hold[4]:

$$\text{opt}_{Q'}(\mathbf{A}) = \max_{\mathbf{S}, F} |\{\mathbf{w} : (\mathbf{A}, \mathbf{S}, F) \models [\forall \mathbf{x} \; \hat{\eta}(\mathbf{x}, \mathbf{w}, \mathbf{S})] \wedge F(\mathbf{w})\}|. \tag{2.6}$$

In (2.6), $\forall \mathbf{x} \; \hat{\eta}(\mathbf{x}, \mathbf{w}, \mathbf{S})$ represents the feasibility of the given instance $\mathbf{A}$.

[3]Observe that $\hat{\eta}_1$ captures the cardinality condition $|F| \geq K$. To represent this, we can define a first-order relation G of arity k over the universe A of $\mathbf{A}$ such that $|G| = K$ and $F(\mathbf{w})$ is true whenever $G(\mathbf{w})$ is. Then we need to represent the fact that $|F| \geq |G|$, which can be characterized as $\forall \mathbf{w} \; G(\mathbf{w}) \longrightarrow F(\mathbf{w})$.

[4]Out of the four possible cases (i) $(\forall \mathbf{x} \hat{\eta}) \wedge F(\mathbf{w})$, (ii) $(\forall \mathbf{x} \hat{\eta}) \wedge \neg F(\mathbf{w})$, (iii) $(\neg \forall \mathbf{x} \hat{\eta}) \wedge F(\mathbf{w})$, and (iv) $(\neg \forall \mathbf{x} \hat{\eta}) \wedge \neg F(\mathbf{w})$, cases (ii) and (iv) must be disregarded since F is false. Case (iii) should also be disregarded since it violates the feasibility condition $\forall \mathbf{x} \hat{\eta}$.

If $\mathbf{S}$ and F can be represented by a single sequence of relations $\mathbf{T} = (S_1, S_2, \cdots S_p, F)$, the optimal solution value can be expressed as

$$\mathrm{opt}_{Q'}(\mathbf{A}) = \max_{\mathbf{T}} |\{\mathbf{w} : (\mathbf{A}, \mathbf{T}) \models \forall \mathbf{x}\ \eta(\mathbf{x}, \mathbf{w}, \mathbf{T})\}| \tag{2.7}$$

where $\eta(\mathbf{x}, \mathbf{w}, \mathbf{T}) = \hat{\eta}(\mathbf{x}, \mathbf{w}, \mathbf{S}) \wedge F(\mathbf{w})$. ($\hat{\eta}$ and η are quantifier-free.) Since $\hat{\eta}$ is Horn, so is η.

Hence the proof. □

Notes. Theorem 2.2 does NOT provide a polynomial-time algorithm to obtain an optimal solution — the maximum in (2.7) is taken over ALL solutions $\mathbf{S}$, which could be exponential in number. Though the number of *solution values* is at most $|A|^{\mathcal{R}}$, which is polynomial in the size of the instance, the number of *solutions* need not be. Examining each solution value between 1 and $|A|^{\mathcal{R}}$ does not guarantee finding a solution in polynomial time. (The difficulty arises in inverse-mapping solution values to solutions.)

2.1.1 Syntactic Expression for a P-maximization Problem's Decision Version with Ordered Structures

Consider a problem P as defined in item *(v)* of Definition 1.1: *Given an instance I and a non-negative integer k, is there a feasible solution $S \in F_Q(I)$, such that $f_Q(I, S) \geq k$?*
(we assume that the solution values are non-negative integers)

Recall item *(vi)* of that definition, which says that an optimal solution $S_{\mathrm{opt}}(I)$ for a given instance I can be computed in time polynomial in $|I|$. We assume that the structure we use, $\mathbf{A}$, is ordered, and comes with an explicit successor relation $succ(x, y)$ (see the discussion after Definition 1.11).

The expression for the problem will consist of two types of conditions: (i) The feasibility of a solution $\mathbf{S}$, and (ii) the fact that the objective function value is at least a given integer k.

Conditions (i) and (ii) relate to the *constraints* and the *objective function* respectively in a classical mathematical programming formulation of an optimization problem. Here we only concern ourselves with condition (ii), which we will show is a universal Horn formula.

We will phrase our query along the same lines as (2.7): Is there a solution $\mathcal{S}$ (represented by a sequence of relations $\mathbf{S}$) such that the number of $\mathbf{w}$ tuples that satisfy ψ_1 in (2.7) is equal to or higher than a certain k ? (Does there exist an integer[5] L, such that $L \geq k$?)

We first create an $m-$ary artificial predicate F to hold the $\mathbf{w}$ tuples that satisfy the feasibility condition ψ_1 in (2.7). That is, $F(\mathbf{w})$ is true iff $\mathbf{w}$ satisfies ψ_1. In addition, assume that F is in lexicographic (lex) order; recall that the structure $\mathbf{A}$ is ordered — hence F can be considered as a *sequence*. Since F has a lex order, it has a *smallest* or *first* tuple which occurs earlier than any other tuple in F, and a *highest* or *last* tuple that occurs after all other tuples in the sequence F.

[5]That the L number of tuples $\mathbf{w_1}, \mathbf{w_2}, \cdots, \mathbf{w_L}$ form a meaningful solution is taken care of by the sequence of predicates $\mathbf{S}$. For example, in the LONGEST PATH problem, the $\mathbf{w}$'s would be two-dimensional tuples (the edges), and it is up to $\mathbf{S}$ to ensure that the $\mathbf{w}$'s form a path from origin to destination.

$$\begin{aligned}
& \mathbf{x} = (x_1, x_2, \cdots, x_m)\text{, and similarly for } \mathbf{y} \text{ and } \mathbf{z}\text{. (These are tuples in } A^m\text{.)} \\
& \forall \mathbf{x} \equiv \forall x_1 \forall x_2 \cdots \forall x_m\text{; and similarly for } \forall \mathbf{y} \text{ and } \forall \mathbf{z}. \\
\eta_1 \equiv\ & \forall x \forall \mathbf{y}\ g(x,\mathbf{y}) \to [F(\mathbf{y}) \wedge C(x)] \ \equiv\ \forall x \forall \mathbf{y}\ \neg g(x,\mathbf{y}) \vee [F(\mathbf{y}) \wedge C(x)] \\
\equiv\ & \forall x \forall \mathbf{y}\ [\neg g(x,\mathbf{y}) \vee F(\mathbf{y})] \wedge [\neg g(x,\mathbf{y}) \vee C(x)]. \\
& [g \text{ is a binary relation. For } g\text{, the first argument } x \in I_k\text{, and the second argument } \mathbf{y} \in F.] \\
\eta_2 \equiv\ & \forall \mathbf{y}\ F(\mathbf{y}) \to [\textstyle\bigwedge_{i=1}^m \neg C(y_i) \wedge \forall \mathbf{x}\ \eta(\mathbf{w},\mathbf{x},\mathbf{S})] \\
\equiv\ & \forall \mathbf{y} \forall \mathbf{x}\ \neg F(\mathbf{y}) \vee [\textstyle\bigwedge_{i=1}^m \neg C(y_i) \wedge \eta(\mathbf{w},\mathbf{x},\mathbf{S})] \\
\equiv\ & \forall \mathbf{y} \forall \mathbf{x}\ \textstyle\bigwedge_{i=1}^m [\neg F(\mathbf{y}) \vee \neg C(y_i)] \wedge [\neg F(\mathbf{y}) \vee \eta(\mathbf{w},\mathbf{x},\mathbf{S})]. \\
& [\text{If } \mathbf{y} \in F\text{, then } \psi_1 \equiv \forall \mathbf{x}\ \eta(\mathbf{w},\mathbf{x},\mathbf{S}) \text{ is true, and } y_i \notin I_k \text{ for } 1 \le i \le m.] \\
\eta_3 \equiv\ & \forall x \forall \mathbf{y} \forall \mathbf{z}\ [g(x,\mathbf{y}) \wedge g(x,\mathbf{z})] \to (\mathbf{y} =_{\mathbf{m}} \mathbf{z}) \ \equiv\ \forall x \forall \mathbf{y} \forall \mathbf{z}\ \neg g(x,\mathbf{y}) \vee \neg g(x,\mathbf{z}) \vee (\mathbf{y} =_{\mathbf{m}} \mathbf{z}) \\
\equiv\ & \forall x \forall \mathbf{y} \forall \mathbf{z}\ \neg g(x,\mathbf{y}) \vee \neg g(x,\mathbf{z}) \vee \textstyle\bigwedge_{i=1}^m (y_i = z_i) \\
\equiv\ & \forall x \forall \mathbf{y} \forall \mathbf{z}\ \textstyle\bigwedge_{i=1}^m [\neg g(x,\mathbf{y}) \vee \neg g(x,\mathbf{z}) \vee (y_i = z_i)]. \quad [g \text{ is a function.}]
\end{aligned}$$

Table 2: Expressing g as a Function

So the question to be answered now is, is there such a predicate F with a cardinality of at least k ? Is $|F| \ge k$?

We can answer this question positively, if we can show that there is a total injective function g from the set $I_k = \{1, 2, 3, \cdots, k\}$, to the *first* k tuples in F — from this, we will be able to conclude that F has at least k elements. Note that $F \subseteq A^m$. The function g should map each number $i \in I_k$ to the i-th tuple in F. I_k can be merged into A to form an expanded universe $U = A \cup I_k$. Assume w.l.o.g. that A and I_k are disjoint (if they are not, then rename the corresponding elements in A). First, we create a unary switch relation $C(x)$ which is true iff $x \in I_k$.

Expressing g as a function is done in Table 2. Since F is a sequence, for two tuples $\mathbf{y}$, $\mathbf{z} \in F$, the fact that $\mathbf{y}$ appears before $\mathbf{z}$ can be defined as a $(2m)$−ary first order relation $(\mathbf{y} <_{\mathrm{m}} \mathbf{z})$:

$$\begin{aligned}
(\mathbf{y} <_{\mathrm{m}} \mathbf{z}) \equiv \textstyle\bigvee_{i=1}^m \phi_i, \text{ where } \phi_i \ &\equiv\ \textstyle\bigwedge_{j=1}^m (y_i < z_i) \wedge [(j < i) \to y_i = z_j], \\
\text{Or, } \phi_i \ &\equiv\ \textstyle\bigwedge_{j=1}^m (y_i < z_i) \wedge [\neg(j < i) \vee y_i = z_j].
\end{aligned} \tag{2.8}$$

For any two formulae ϕ_i and ϕ_j in (2.8) such that $i \neq j$, $\phi_i \wedge \phi_j$ is false (pairwise disjunct). Hence, if $(\mathbf{y} <_{\mathrm{m}} \mathbf{z})$ is true, then exactly one of the ϕ_i's in (2.8) is true and the rest are false. Expression (2.8) above is provided only a reference; we will not need it in our final expression, since $<_{\mathrm{m}}$ is treated as a first order predicate (part of the input instance) — see below. Thus the tuples in F can be sequenced lexicographically.

Also note that $<_{\mathrm{m}}$ is a first order predicate, not second order; given the values of $\mathbf{y}$ and $\mathbf{z}$ from the input A, one can evaluate $(\mathbf{y} <_{\mathrm{m}} \mathbf{z})$ from input information for the problem instance, such as $(y_i < z_i)$, in at most m steps, where m is a constant. Similarly, scalar equality ($=$) is a binary first order predicate and tuple equality ($=_{\mathrm{m}}$) is a $(2m)-$ary first order predicate. The proposition $(\mathbf{y} =_{\mathbf{m}} \mathbf{z})$ is shorthand for $\bigwedge_{i=1}^{m}(y_i = z_i)$.

Introduce a second-order successor relation $succT$ for $(2m)-$ary tuples, similar to expression (1.11) for elements in A:

$$\begin{aligned} \eta_4 &\equiv \forall\mathbf{u}\forall\mathbf{v}\, succT(\mathbf{u},\mathbf{v}) \to \phi_0 \;\equiv\; \forall\mathbf{u}\forall\mathbf{v}\, \neg succT(\mathbf{u},\mathbf{v}) \vee \phi_0, \text{ where,} \\ \phi_0 &\equiv F(\mathbf{u}) \wedge F(\mathbf{v}) \wedge \mathbf{u} <_{\mathrm{m}} \mathbf{v} \wedge \neg[\exists\mathbf{w}\, F(\mathbf{w}) \wedge \mathbf{u} <_{\mathrm{m}} \mathbf{w} \wedge \mathbf{w} <_{\mathrm{m}} \mathbf{v}] \\ &\equiv F(\mathbf{u}) \wedge F(\mathbf{v}) \wedge \mathbf{u} <_{\mathrm{m}} \mathbf{v} \wedge \forall\mathbf{w}[\neg F(\mathbf{w}) \vee \neg(\mathbf{u} <_{\mathrm{m}} \mathbf{w}) \vee \neg(\mathbf{w} <_{\mathrm{m}} \mathbf{v})]. \end{aligned} \tag{2.9}$$

Hence η_4 can be rewritten as $\eta_4 \equiv \forall\mathbf{u}\forall\mathbf{v}\forall\mathbf{w}\ \phi_1 \wedge \phi_2 \wedge \phi_3 \wedge \phi_4$, where,

$$\begin{aligned} \phi_1 &\equiv \neg succT(\mathbf{u},\mathbf{v}) \vee F(\mathbf{u}), \\ \phi_2 &\equiv \neg succT(\mathbf{u},\mathbf{v}) \vee F(\mathbf{v}), \\ \phi_3 &\equiv \neg succT(\mathbf{u},\mathbf{v}) \vee (\mathbf{u} <_{\mathrm{m}} \mathbf{v}), \text{ and} \\ \phi_4 &\equiv \neg succT(\mathbf{u},\mathbf{v}) \vee \neg F(\mathbf{w}) \vee \neg(\mathbf{u} <_{\mathrm{m}} \mathbf{w}) \vee \neg(\mathbf{w} <_{\mathrm{m}} \mathbf{v}). \end{aligned} \tag{2.10}$$

The fact that a certain tuple $\mathbf{u} \in A^m$ is the *first* tuple in F is defined by the following $m-$ary second-order relation min:

$$\begin{aligned} \eta_5 &\equiv \forall\mathbf{u}\ \min(\mathbf{u}) \to \{F(\mathbf{u}) \wedge \forall\mathbf{v}[F(\mathbf{v}) \to \neg(\mathbf{v} <_{\mathrm{m}} \mathbf{u})]\} \\ &\equiv \forall\mathbf{u}\ \neg\min(\mathbf{u}) \vee \{F(\mathbf{u}) \wedge \forall\mathbf{v}[\neg F(\mathbf{v}) \vee \neg(\mathbf{v} <_{\mathrm{m}} \mathbf{u})]\} \\ &\equiv \forall\mathbf{u}\forall\mathbf{v}\ [\neg\min(\mathbf{u}) \vee F(\mathbf{u})] \wedge [\neg\min(\mathbf{u}) \vee \neg F(\mathbf{v}) \vee \neg(\mathbf{v} <_{\mathrm{m}} \mathbf{u})]. \end{aligned} \tag{2.11}$$

We should express that g maps the first element in I_k to the first element in F:

$$\eta_6 \equiv \forall\mathbf{u}\ \min(\mathbf{u}) \to g(1,\mathbf{u}) \;\equiv\; \forall\mathbf{u}\ \neg\min(\mathbf{u}) \vee g(1,\mathbf{u}) \tag{2.12}$$

As long as F is non-empty, it will contain a "minimum" (first) tuple $\mathbf{u}$, and hence there will be a $g(1,\mathbf{u})$ map. The inductive formula below states that if g maps a number x to an element $\mathbf{u} \in F$, then g also maps $x+1 = y$ (the successor of x) to $\mathbf{u}$'s successor in F:

$$\begin{aligned} \eta_7 &\equiv \forall x\forall y\forall\mathbf{u}\forall\mathbf{v}\ [g(x,\mathbf{u}) \wedge (x<k) \wedge succ(x,y) \wedge succT(\mathbf{u},\mathbf{v})] \to g(y,\mathbf{v}) \\ &\equiv \forall x\forall y\forall\mathbf{u}\forall\mathbf{v}\ \neg g(x,\mathbf{u}) \vee \neg(x<k) \vee \neg succ(x,y) \vee \neg succT(\mathbf{u},\mathbf{v}) \vee g(y,\mathbf{v}). \end{aligned} \tag{2.13}$$

This takes care of the requirement that $g : I_k \to F$ is a *total injective* function.

Thus the fact that $|F| \geq k$ can be expressed as (placing the three second-order existential quantifiers at the beginning):

$$\Phi = (\exists F \exists g)(\exists succT)(\exists \min) \bigwedge_{i=1}^{7} \eta_i, \tag{2.14}$$

provided the quantifiers in each of the η_i formulae are distinct. This can easily be achieved by renaming the variables. For example, the quantifiers (and hence the variables) in η_1 can be renamed as $\forall x_1$ and $\forall \mathbf{y_1}$, those in η_2 as $\forall \mathbf{y_2}$, those in η_3 as $\forall x_3 \forall \mathbf{y_3} \forall \mathbf{z_3}$, and so on. Placing all quantifiers at the beginning, Φ can be rewritten as

$$\Phi = (\exists F \exists g)(\exists succT)(\exists \min)\ \forall x_1 \forall \mathbf{y_1} \forall \mathbf{y_2} \cdots \forall \mathbf{u_7} \forall \mathbf{v_7} \bigwedge_{i=1}^{7} \theta_i, \tag{2.15}$$

where θ_i is the quantifier-free part of η_i. Two important observations:

1. All quantifiers in η_1 through η_7 are universal, and

2. All clauses in η_1 through η_7 are Horn clauses, hence $\bigwedge_{i=1}^{7} \theta_i$ is a conjunction of Horn clauses.

These two conditions satisfy those of Grädel's theorem (Theorem 1.12).

We have been able to represent the problem P, which is a decision version of a maximization problem, as a universal Horn formula. We have expressed both conditions: (i) The one with respect to the maximization problem's objective function, as well as (ii) the feasibility constraint ψ_1 in (2.7).

Hence, in this section, we have shown that

Theorem 2.3. *Let P be a decision problem (in the class $\mathbf{P_{oc}}$) as defined in item (v) of Definition 1.1. Then P can be represented in Existential Second Order logic, where the first order part is a universal Horn formula in prenex normal form, provided the structures representing instances of P are (i) ordered, and (ii) come with an explicit successor relation defined in the vocabulary of the structure.*

Remark 2.4. Theorem 2.3 is of course, a special case of Theorem 1.12. However, the fact that $\mathbf{P_{oc}} \subset \mathbf{P}$ does not imply that an ESO universal Horn formula can be obtained in a straightforward manner for a problem $P \in \mathbf{P_{oc}}$ using Theorem 1.12. Unless one exploits special structures of the problems in $\mathbf{P_{oc}}$ (as we have done above), expressing a generic problem in $\mathbf{P}$ using Theorem 1.12 can be an arduous task, such as having to represent problem instances as input strings for a deterministic Turing machine.

2.1.2 Example: Polynomially Bounded Maximum Flow (Unit Capacities)

In this section, we will see how the MAXFLOW problem with unit capacities can be expressed in ESO, in Π_1 first order form. Given a source s and a sink t, and a network G containing directed edges with unit capacity, we want to find the maximum flow that can be sent through the network from s to t. In other words, we seek the maximum number of edge-disjoint paths from s to t. Call this (polynomially bounded) problem $\text{MAXFLOW}_{\text{PB}}$.

Assume that the given network has an s-t maximum flow, where the flow along each edge is either zero or one. Consider edges in the network of the form (s,w) that are outgoing from the source s. Observe that maximizing the s-t flow is the same as maximizing the number of such edges (s,w) that carry a unit flow.

All vertices in the network adjacent to s can be considered as unary tuples of the form (w). Thus, maximizing the s-t flow is equivalent to maximizing the number of tuples (w) that receive a unit flow from s.

Every *s*-*t* edge-disjoint path can be considered as a partial order on the set of vertices. We will use ideas similar to those used in the expression for the REACHABILITY problem (see [16], Section 5.7, Pages 114-115).

To represent the partial orders, introduce a second-order ternary predicate $P(x,y,w)$ which holds iff $x \neq y$ and there is an edge-disjoint path from s to w to x to y, in the feasible solution — the path from s to w, is of course, just a single edge. (The "main arguments" for P are x and y — w is just an additional reference.) Thus we seek the maximum number of w's such that $P(w,t,w)$ is true. The following expressions capture the properties of a feasible solution.

(1) If $P(x_1,x_2,w)$ holds, then so does $G(s,w)$ — that is, the edge (s,w) is defined in G:

$$\begin{aligned} \phi_1 &\equiv \forall x_1 \forall x_2 \forall w\ P(x_1,x_2,w) \longrightarrow G(s,w) \\ &\equiv \forall x_1 \forall x_2 \forall w\ \neg P(x_1,x_2,w) \vee G(s,w). \end{aligned} \tag{2.16}$$

(2) An edge (i,j) can be a part of only one $s-t$ disjoint path (equivalently, only one $w-t$ edge disjoint path):

$$\begin{aligned} \phi_2 &\equiv \forall i \forall j \forall w_1 \forall w_2\ P(i,j,w_1) \wedge P(i,j,w_2) \wedge G(i,j) \longrightarrow (w_1 = w_2) \\ &\equiv \forall i \forall j \forall w_1 \forall w_2\ \neg P(i,j,w_1) \vee \neg P(i,j,w_2) \vee \neg G(i,j) \vee (w_1 = w_2). \end{aligned} \tag{2.17}$$

(3) P is non-reflexive:

$$\phi_3 \equiv \forall y_1 \forall y_2\ \neg P(y_1,y_1,y_2). \tag{2.18}$$

(4) P is transitive:

$$\begin{aligned} \phi_4 &\equiv \forall u_1 \forall u_2 \forall u_3 \forall w_3\ P(u_1,u_2,w_3) \wedge P(u_2,u_3,w_3) \longrightarrow P(u_1,u_3,w_3). \\ &\equiv \forall u_1 \forall u_2 \forall u_3 \forall w_3\ \neg P(u_1,u_2,w_3) \wedge \neg P(u_2,u_3,w_3) \vee P(u_1,u_3,w_3). \end{aligned} \tag{2.19}$$

(5) And finally, any two adjacent vertices in P should also be adjacent in G:

$$\begin{aligned} \phi_5 &\equiv \forall z_1 \forall z_2 \forall w_4\ P(z_1,z_2,w_4) \wedge\ \forall z_3\ \neg[P(z_1,z_3,w_4) \wedge P(z_3,z_2,w_4)] \longrightarrow G(z_1,z_2) \\ &\equiv \forall z_1 \forall z_2 \forall z_3 \forall w_4\ P(z_1,z_2,w_4) \wedge \neg[P(z_1,z_3,w_4) \wedge P(z_3,z_2,w_4)] \longrightarrow G(z_1,z_2) \\ &\equiv \forall z_1 \forall z_2 \forall z_3 \forall w_4\ \neg P(z_1,z_2,w_4) \vee [P(z_1,z_3,w_4) \wedge P(z_3,z_2,w_4)] \vee G(z_1,z_2) \\ &\equiv \forall z_1 \forall z_2 \forall z_3 \forall w_4\ [\neg P(z_1,z_2,w_4) \vee P(z_1,z_3,w_4) \vee G(z_1,z_2)] \\ &\quad \wedge [\neg P(z_1,z_2,w_4) \vee P(z_3,z_2,w_4) \vee G(z_1,z_2)]. \end{aligned} \tag{2.20}$$

Let $\Phi = \bigwedge_{i=1}^{5} \phi_i$.

Observe that each ϕ_i $(1 \leq i \leq 5)$ is a Π_1 Horn formula, as required by Theorem 2.2. The optimal solution value to the given instance (network G, represented by a structure $\mathbf{A}$), is given by

$$\mathrm{opt}_Q(\mathbf{A}) = \max_P |\{w : (\mathbf{A},P) \models\ P(w,t,w) \wedge \Phi\}|. \tag{2.21}$$

Remark 2.5. In most such expressions as above, there are two sets of conditions to be expressed:
(1) *Global* conditions (those that apply over all $\mathbf{w}$ tuples), such as the expression Φ above. This set corresponds to the *constraints* in a classical mathematical programming framework; and
(2) *Local* conditions (the ones that are specific to a given $\mathbf{w}$), such as $P(w,t,w)$ above. This set corresponds to the *objective function* in mathematical programming.

It is clear that MAXFLOW$_{\text{PB}}$ can be expressed with neither Σ_0 nor Σ_1 formulae. In particular, without universal quantifiers, none of the five properties — expressions (2.16) to (2.20) — can be expressed independently of the size of the instance.

Consider property ϕ_2, for instance. Using only existential quantifiers, one should enumerate the property individually for each edge. However, this will make the length of ϕ_2 dependent on the number of edges in the graph. Hence we conjecture that

Conjecture 2.6. *The property, that an edge belongs at most one edge-disjoint $s-t$ path in a solution, (and hence the* MAXFLOW$_{\text{PB}}$ *problem) can be expressed with a Π_1 formula, but not with a Σ_1 formula.*

2.1.3 MAXFLOW$_{\text{PB}}$ is Complete for Polynomially Bounded Maximization

The analysis in this subsection and the next two (Sec.2.1.4 and Sec.2.1.5) are purely to obtain a hierarchy in terms of syntactic expressibility. Separation in terms of computation time or approximability are not issues here, since all problems considered are members of $\mathbf{P^{pb}_{opt}}$.

We can show that the MAXFLOW$_{\text{PB}}$ problem is complete for the class of polynomially bounded maximization problems by reducing an instance I of a general problem Q' in this class to an instance $\mathfrak{I}$ of MAXFLOW$_{\text{PB}}$. If I is represented by a structure $\mathbf{A}$, the optimal solution value to I is given by Theorem 2.2:

$$\text{opt}_{Q'}(\mathbf{A}) = \max_{\mathbf{S}} |\{\mathbf{w} : (\mathbf{A},\mathbf{S}) \models \Phi\}|, \quad \Phi = \forall \mathbf{x}\ \eta(\mathbf{w},\mathbf{x},\mathbf{S}), \tag{2.22}$$

if $Q' \in \mathbf{P^{pb}_{opt}}$, where $\mathbf{x}$, $\mathbf{A}$, $\mathbf{S}$ and η are defined in Table 1. (Recall that η is a conjunction of Horn clauses and quantifier-free, and $\mathbf{S}$ is a sequence of second order predicate symbols.)

Let the arity of $\mathbf{w}$ ($\mathbf{x}$) be k (m) respectively. The different possible $\mathbf{w}$ ($\mathbf{x}$) tuples are $\mathbf{w_i}$, $1 \le i \le n^k$ ($\mathbf{x_j}$, $1 \le j \le n^m$), where n is the cardinality of the universe A of $\mathbf{A}$. For a given $\mathbf{w_i}$, the expression for Φ in (2.22) can be rewritten as

$$\Phi(\mathbf{w_i}) = \forall \mathbf{x}\ \eta(\mathbf{w_i},\mathbf{x},\mathbf{S}) = \bigwedge_{j=1}^{n^m} \eta(\mathbf{w_i},\mathbf{x_j},\mathbf{S}). \tag{2.23}$$

Instance $\mathfrak{I}$ consists of n^k+2 vertices — one for each $\mathbf{w_i}$ tuple, as well as two additional vertices s and t. Add a directed edge with unit capacity from s to each $\mathbf{w_i}$ vertex. Add a directed edge with unit capacity from each $\mathbf{w_i}$ vertex to t iff $\Phi(\mathbf{w_i})$ holds.

The reduction is polynomial time — $O(n^k)$ time to create the vertices, and $O(n^{k+m})$ time to add the edges. It is clear that instance $\mathfrak{I}$ of MAXFLOW$_{\text{PB}}$ has a maximum flow of α units from source s to sink t iff the optimal solution value to I in (2.22) is also α.

2.1.4 A Problem in MAX$_{\mathrm{P}}$■$_0$?

Within the class $\mathbf{P}^{\mathbf{pb}}_{\mathbf{opt}}$ (see Table 1), let us define the class MAX$_{\mathrm{P}}\Sigma_0$ (MAX$_{\mathrm{P}}\Pi_1$) as the class of polynomially bounded maximization problems that can be expressed by a Σ_0 (Π_1) formula.

Kolaitis and Thakur showed that MAX3SAT is a member of MAX$_{\mathrm{NP}}\Sigma_0$ (defined similarly to MAX$_{\mathrm{P}}\Sigma_0$), a subset of $\mathbf{NP}^{\mathbf{pb}}_{\mathbf{opt}}$.

On the other hand, MAX2SAT is not known to be polynomially solvable [9], though the decision version, as to whether all clauses are satisfiable, is well-known to be in **P** ([6], Page 259). Results similar to MAX2SAT for both the maximization and decision versions are also known for HORNSAT (where every clause is required to be a Horn clause) [12].

Towards the goal of obtaining a hierarchy within the polynomially bounded P-maximization class, we need to exhibit a problem in MAX$_{\mathrm{P}}\Sigma_0$. However, we have been unable to find such problem so far. We conjecture that

Conjecture 2.7. *As long as a successor relationship or a linear ordering on the universe of a structure is necessary, a problem cannot be expressed in MAX*$_{\mathrm{P}}\Sigma_0$ *(since this will require a* Π_1 *expression).*

To express linear ordering of arbitrary length in first order Σ_0, one would need an arbitrary number of variables, and hence a formula of arbitrary length.

2.1.5 Hierarchy Within Maximization

We state the hierarchy within the polynomially bounded maximization class without a formal proof (since it is clear from the argument below): If a maximization problem exists in the Σ_0 class (that is, the Π_0 class), then the Σ_0 class is strictly contained within the Π_1 class.

It is clear that the problem considered in Section 2.1.2, MAXFLOW$_{\mathrm{PB}}$, cannot be expressed with a Σ_0 formula. In particular, without quantifiers, none of the five properties in Section 2.1.2 — expressions (2.16) to (2.20) — can be expressed.

From Section 2.1.3, clearly MAXFLOW$_{\mathrm{PB}}$ serves as a complete problem for the MAX$_{\mathrm{P}}\Pi_1$ class. It would be desirable to obtain a complete problem for the MAX$_{\mathrm{P}}\Sigma_0$ class. An interesting observation is that the decision version of the weighted MAXFLOW problem (where arc capacity can be any non-negative integer) is complete for the class **P** (Page 152 of [8] and Page 30 of [10]).

Another question to be answered is, is there a class MAX$_{\mathrm{P}}\Sigma_1$ which is between MAX$_{\mathrm{P}}\Pi_1$ and MAX$_{\mathrm{P}}\Sigma_0$?

2.2 Polynomially Bounded P-Minimization Problems

For minimization problems in $\mathbf{NP}^{\mathbf{pb}}_{\mathbf{opt}}$, Kolaitis and Thakur [13] proved the following (see Table 1 for a definition of $\mathbf{NP}^{\mathbf{pb}}_{\mathbf{opt}}$ and Definition 1.10 regarding Σ_2 formulae):

Theorem 2.8. *A minimization problem $Q' \in \mathbf{NP}^{\mathbf{pb}}_{\mathbf{opt}}$ if and only if there exists a Σ_2 first order formula $\phi(\mathbf{w},\mathbf{S})$ with predicate symbols from the vocabulary σ (of ϕ) and the sequence $\mathbf{S}$, such that for every instance $\mathbf{A}$ of Q',*

$$\mathrm{opt}_{Q'}(\mathbf{A}) = \min_{\mathbf{S}} |\{\mathbf{w} : (\mathbf{A},\mathbf{S}) \models \phi(\mathbf{w},\mathbf{S})\}|. \tag{2.24}$$

In other words, they showed that all polynomially bounded NP-minimization problems fall in what can be called the MIN Σ_2 class. In the same paper, they also showed that this class is equivalent to the MIN Π_1 class.

A similar result can be shown for minimization problems in $\mathbf{P}^{\mathbf{pb}}_{\mathbf{opt}}$ (the polynomial-time equivalent of $\mathbf{NP}^{\mathbf{pb}}_{\mathbf{opt}}$):

Theorem 2.9. *Let $\mathbf{A}$ be a structure (instance) defined over σ. If Q' is a minimization problem in $\mathbf{P}^{\mathbf{pb}}_{\mathbf{opt}}$, then the value of an optimal solution to an instance $\mathbf{A}$ of Q' can be represented by*

$$\mathrm{opt}_{Q'}(\mathbf{A}) = \min_{\mathbf{S},F} |\{\mathbf{w} : (\mathbf{A},\mathbf{S},F) \models \forall \mathbf{x}\ \tau\}| \tag{2.25}$$

where $\tau = \eta(\mathbf{w},\mathbf{x},\mathcal{S}) \wedge F(\mathbf{w})$, and $\mathbf{x}$, $\mathbf{A}$, $\mathbf{S}$, η are defined as in Table 1. (The symbol F is a $k-$ary relation defined on the universe $|A|$ of $\mathbf{A}$, since each $\mathbf{w}$ is $k-$dimensional.)

Proof. The proof that Q' is polynomially bounded is the same as in Theorem 2.2.

We start with Grädel's Theorem (Theorem 1.12). The decision problem can be represented by an ESO Horn expression $\exists \mathbf{S} \psi$ where $\mathbf{S}$ is a sequence of predicate symbols, and ψ is a Π_1 first order Horn expression where a successor relation is included in the vocabulary of ψ.

The analysis for the decision problem Q is similar to the maximization case, except that one looks for at most K tuples that satisfy the feasibility condition $\eta(\mathbf{x},\mathbf{W},\mathbf{S})$. (This is the same as looking for at least $n^k - K + 1$ tuples that do not satisfy $\eta(\mathbf{x},\mathbf{W},\mathbf{S})$.)

Thus for the minimization version of the problem, an optimal value to an instance $\mathbf{A}$ can be written as

$$\mathrm{opt}_{Q'}(\mathbf{A}) = \min_{\mathbf{S}} |\{\mathbf{w} : (\mathbf{A},\mathbf{S}) \models\ \phi(\mathbf{w},\mathbf{S})\}| \tag{2.26}$$

where $\phi(\mathbf{w},\mathbf{S}) = \forall \mathbf{x}\ \eta(\mathbf{x},\mathbf{w},\mathbf{S})$.

The $\mathbf{w}$ tuples can be considered as a $k-$ary relation F such that $F(\mathbf{w})$ is true if and only if $\mathbf{w} \in F$. Hence $\phi(\mathbf{w},\mathbf{S})$ in (2.26) should be modified to $\forall \mathbf{x}\ \eta(\mathbf{x},\mathbf{w},\mathbf{S}) \wedge F(\mathbf{w})$. The number of tuples $|F|$ in F should be minimized.

Again, out of the the four cases (i) $(\forall \mathbf{x}\eta) \wedge F(\mathbf{w})$, (ii) $(\forall \mathbf{x}\eta) \wedge \neg F(\mathbf{w})$, (iii) $(\neg\forall \mathbf{x}\eta) \wedge F(\mathbf{w})$, and (iv) $(\neg\forall \mathbf{x}\eta) \wedge \neg F(\mathbf{w})$, cases (ii) and (iv) should be disregarded since F is false, and (iii) violates the feasibility condition $\forall \mathbf{x}\eta$. This leaves us with the following modification of (2.26):

$$\mathrm{opt}_{Q'}(\mathbf{A}) = \min_{\mathbf{S},F} |\{\mathbf{w} : (\mathbf{A},\mathbf{S}) \models\ (\forall \mathbf{x}\eta) \wedge F(\mathbf{w})\}| \tag{2.27}$$

	Maximization	Minimization
NP	$\mathrm{MAX_{NP}}\Pi_2$: ESO (Π_2 first order)	$\mathrm{MIN_{NP}}\Pi_1$: ESO (Π_1 first order)
P	$\mathrm{MAX_{P}}\Pi_1$: ESO (Horn Π_1 first order)	$\mathrm{MIN_{P}}\Pi_1$: ESO (Horn Π_1 first order)

Table 3: Syntactic Characterizations of Polynomially Bounded Optimization Problems

Note that $(\forall \mathbf{x}\eta) \wedge F(\mathbf{w}) = \forall \mathbf{x}(F(\mathbf{w}) \wedge \eta)$. Since η is Horn, so is $(F(\mathbf{w}) \wedge \eta)$.

It may appear that the minimization in (2.27) will always result in an optimal value $|F|$ of zero, but since the minimum value is taken only over all *feasible* solutions $(\mathbf{S}, F)$, the value obtained in (2.27) is correct. Hence the proof. (The minimization over "only feasible solutions $(\mathbf{S}, F)$" needs further illustration and is provided below.) □

Illustration of Minimization over "only feasible solutions $(\mathbf{S}, F)$". To illustrate this point, consider the SHORTEST PATH problem in Section 2.2.1. We attempt to minimize the number of edges in a path from the source s to the sink t. If we minimize over **any** ($\mathbf{S}$, F) combination, obviously this minimum number would be zero — however, this would violate the feasibility condition ϕ_1 that there exists a path from s to t. Hence this "zero" solution is obviously infeasible.

Consider another example, MIN SET COVER. We are given a *ground* set X and several subsets Y_1, Y_2, $\cdots$, Y_q of X. Let $C = \{Y_1, Y_2, \cdots, Y_q\}$. The problem is select a few (minimum number of) subsets Y_i from C, such that the union of the selected subsets is X. We can associate a unary tuple w_i to each Y_i, such that the number of such w tuples in a solution is to be minimized. Let a second order predicate $S(w_i)$ determine if a certain subset Y_i is chosen in a solution S. Obviously if we minimize over *all possible* S, the minimum number of Y_i subsets selected will be zero — but then, such a solution is clearly infeasible, since the union of the selected subsets (zero of them!) is not equal to the ground set X.

Remark 2.10. From Theorems 2.1, 2.2, 2.8 and 2.9, the following can be observed in the case of *polynomially bounded* optimization problems:

While second-order expressions are able to distinguish clearly between NP-maximization and P-maximization problems (Π_2 for the former and Horn Π_1 for the latter), the distinction is less clear between NP-minimization and P-minimization problems (Π_1 formulae in both cases, the only distinction being the Horn clause requirement in the P-minimization case).

Looking at this from another angle, while the relationship is asymmetric in the case of NP-optimization problems (the first order part is Π_2 for maximization and Π_1 for minimization), it is symmetric for P-optimization problems (the first order part is Horn Π_1 for both maximization and minimization).

The results are summarized in Table 3.

2.2.1 Example: Shortest Path

We now provide an example of a polynomially bounded P-minimization problem, SHORTEST $\mathrm{PATH_{PB}}$. Assume that the edges have unit weight and are directed. The number of edges in the shortest path is to be minimized. The decision version of this problem is easily represented as a Σ_1 formula:

$$\exists x_1 \exists x_2 \cdots \exists x_k \; G(s,x_1) \wedge G(x_1,x_2) \wedge \cdots \wedge G(x_{k-1},x_k) \wedge G(x_k,t) \tag{2.28}$$

where s is the origin and t is the destination. The above formula says that there is a path from s to t of length $k+1$, and it is a Horn formula. ($G(x,y)$ is true if there exists an arc (x,y) in graph G.) We have not used any second-order variables in (2.28), hence the decision version is *FO (first order) expressible*.

Minimization version. A shortest path (or any path from origin to destination) represents a partial order P on the universe (the set of vertices) — P is represented by a second order (SO) binary predicate. Another SO binary predicate S chooses which arcs in the network are in the required path. Again, we will use ideas similar to those used for REACHABILITY ([16], Section 5.7, Pages 114-115). The following formulae express the properties of P and S:

$\phi_1 \equiv$ $P(s,t) \equiv \eta_1$ (there exists a path from s to t).

$\phi_2 \equiv$ $\forall x\, \forall y\, \forall z\; \eta_2,\;\; \eta_2 \equiv [(P(x,y) \wedge P(y,z)) \rightarrow P(x,z)]$ (P is transitive.)

$\phi_3 \equiv$ $\forall x\, \forall y\; \eta_3,\;\; \eta_3 \equiv \neg P(x,x) \wedge [(P(x,y) \rightarrow \neg P(y,x)]$
(P is neither reflexive nor symmetric.)

$\phi_4 \equiv$ $\forall x\, \forall y\; \eta_4,\;\; \eta_4 \equiv S(x,y) \rightarrow [G(x,y) \wedge P(x,y)]$ (If an edge is chosen by S, then it has to be in the given graph G and in the $s-t$ path P.)

$\phi_5 \equiv$ $\forall x\, \forall y\; \hat{\eta}_5$ with $\hat{\eta}_5 \equiv P(x,y) \rightarrow [S(x,y) \vee \exists z (P(x,z) \wedge S(z,y))]$,
(Recursive definition of P — either there is an (x,y) arc, or there exists a path from x to z and a (z,y) arc.)

$\phi_6 \equiv$ $\forall x\, \forall y\, \forall z\; \eta_6,\quad \eta_6 \equiv [(S(x,y) \wedge S(z,y)) \rightarrow (x = z)]$ (Predecessor is unique, hence there is a unique path P from s to t.)

It can be shown that each η_i ($1 \le i \le 6$) above is equivalent to a Horn clause — clauses with at most one positive literal from the set of second order variables $\{P,\, S\}$ — or a conjunction of such clauses, as required by Theorems 1.12 and 2.9. (A clause such as $P(x,z) \vee S(s,t)$ cannot be a Horn clause, for instance.) The optimal solution value for instance $\mathbf{G}$ can now be written in Horn Π_1 form as

$$\text{opt}(\mathbf{G}) = \min_{P,S} \left| \left\{ (p,q) : (\mathbf{G},P,S) \models \; \forall x\, \forall y\, \forall z \bigwedge_{i=1}^{6} \eta_i \wedge S(p,q) \right\} \right| . \tag{2.29}$$

Conjecture 2.11. *Though we have not proved it,* SHORTEST PATH$_{\text{PB}}$ *could be one of those problems where the decision version can be represented in Σ_1 form, but the optimization problem is in Π_1 form. It would be interesting if this observation (hierarchy in terms of quantifier complexity) could be proven or disproven.*

3 Optimization Problems in $\mathbf{P_{opt}}$

We next turn our attention to the class $\mathbf{P_{opt}}$. This is the set of all optimization problems, *not necessarily polynomially bounded*, but the optimal solution can be computed within time polynomial in the size of the input. (These problems need not obey Equation 1.2.)

Zimand 1998 [18] generalized Theorems 2.1 and 2.8 to all NP-Optimization problems, not just those that are polynomially bounded. He showed that a Π_2 first-order formula captures the feasibility conditions for any problem in this class, while the optimal solution value can be represented by a maximization (or minimization) over weighted tuples — the tuples are similar to those used in expressions (2.7) and (2.27) for polynomially bounded problems, but now they are also assigned rational number weights. The method of attaching weights to tuples has also been discussed in Papadimitriou and Yannakakis 1991 [17].

We sketch below how Zimand's result can be extended to polynomial-time maximization problems as well. Zimand shows that for any positive integer value z for an optimal solution, we can compute a set of weights c_i that are powers of two, such that $z = \sum_i c_i$. However, for a *given* optimization problem Q', the weights on tuples are *given quantities*, such as the arc capacities in a MAXFLOW problem or the arc costs in a TRAVELLING SALESPERSON problem — the weights are part of the input. (Zimand makes no attempt to relate his computed weights to the input weights.)

Grädel's Theorem states that any decision problem $Q \in \mathbf{P}$ can be represented as

$$Q \equiv \exists \mathcal{S} \psi. \tag{3.1}$$

A decision version of a maximization problem asks if there is a solution $\mathcal{S}$ to an instance I (represented by a structure $\mathbf{A}$) such that the objective function $f(\mathbf{A}, \mathcal{S}) \geq K$ where K is a given constant.

Motivation to attach weights to tuples. If I is a YES instance to Q, then ψ must be able to express the fact that $f(\mathbf{A}, \mathcal{S}) \geq K$ using a finite structure, according to Grädel's Theorem. The quantity $f(\mathbf{A}, \mathcal{S})$, though not polynomially bound in the size of I any more, is still a finite quantity. For a structure $\mathbf{A}$ with universe A, the number of $k-$ary tuples possible is $|A|^k$, which is polynomial in the size of the instance. In other words, $f(\mathbf{A}, \mathcal{S})$ need not be polynomially bound, whereas the maximum number of $\mathbf{w}$ tuples should be — this is in contrast to the problems in Sect. 2. One way to capture a larger number ($f(\mathbf{A}, \mathcal{S})$) using a smaller one (the number of $\mathbf{w}$ tuples) is by attaching weights to the tuples.

For example, in the MIN CUT problem (dual of MAX FLOW), the tuples (arcs) are binary, and the weights of these tuples are the arc capacities. In WEIGHTED MAX3SAT, the tuples (clauses) are ternary with a weight attached to each clause. In WEIGHTED MAXSAT, it is unknown how many literals are in each clause, hence a unary tuple is commonly used [13]. In TRAVELLING SALESPERSON (TSP), the weights on the binary tuples are the arc costs. It may be undecided ahead of time how the optimal value to a problem in $\mathbf{P_{opt}}$ can be represented, as to which set of tuples and their weights will be used — for example, the set of edges used in a solution to the TSP is unknown until a solution is determined. However, the number of such sets and their tuples are finite — and the weight of each tuple is a given quantity.

Naturally, each set of tuples described above can be said to form a relation U_i over the universe of $\mathbf{A}$, and the set of all such relations can be represented by $\mathbf{U}$. For $U_i \in \mathbf{U}$, its weight $w(U_i)$ is defined as

$$w(U_i) = \sum_{\mathbf{w} \in U_i} w(\mathbf{w}), \tag{3.2}$$

where $w(\mathbf{w})$ is the given weight of tuple $\mathbf{w}$. For example, in a TSP instance with five vertices, each U_i will contain five tuples (the five arcs in the solution). However, in SHORTEST PATH, the number of tuples in U_i depends on the path (solution) used — hence the cardinality of the different U_i's is not the same, since the number of arcs in each solution can vary.

Furthermore, the universe A should consist of two *sorts* of values: one for the variables (such as vertex indices in graphs), and the other sort for the tuple weights. (This is a variant of Many-sorted Logic, see Enderton [4].)

A unary relation $C(x)$ — sometimes known as a *hidden relation* — decides if a given variable is a *basic* variable, or a weight for one of the tuples. The universe A of a structure $\mathbf{A}$ will be of the form

$$A = \{a_1, a_2, \cdots a_n, w_1, w_2, \cdots w_m\} \tag{3.3}$$

where the a_i's are possible values for the basic variables and the w_j's are possible weights for tuples of basic variables. For any variable x_i, the following expression $\phi_1(x_i)$ should hold:

$$\phi_1(x_i) \equiv [C(x_i)] \longleftrightarrow \bigvee_{j=1}^{n} [x_i = a_j]. \tag{3.4}$$

Introduce a relation $R(x_0, x_1, x_2, \cdots x_k)$ which holds true iff x_0 is a weight for the tuple $\mathbf{w} = (x_1, x_2, \cdots x_k)$ — hence $C(x_0)$ is false, and all other $C(x_i)$'s are true. In an instance, the variable x_0 is instantiated with a weight w_i from A.

3.1 Maximization Problems

Reverting to Grädel's expressibility in (3.1), since ψ is in Π_1 ESO Horn form, it can be written as (just like the case for polynomially bounded problems),

$$\psi(\mathbf{x}, \mathbf{w_i}, \mathcal{S}) = \forall x_1 \forall x_2 \cdots \forall x_m \; \hat{\eta}(x_1, x_2, \cdots x_m, \mathbf{w_i}, \mathcal{S}) = \forall \mathbf{x} \; \hat{\eta}(\mathbf{x}, \mathbf{w_i}, \mathcal{S}) \tag{3.5}$$

where $\mathbf{x} = (x_1, x_2, \cdots x_m)$ — hence $\hat{\eta}$ should express the fact that $w(U_i) \geq K$, and $\hat{\eta}$ should include expressions for every $\phi_1(x_i)$ in (3.4). If a certain relation U_i that satisfies the feasibility conditions exists, then

$$Q \equiv \exists \mathcal{S} \; \exists U_i \; \forall \mathbf{w} \; \forall \mathbf{x} \, [U_i(\mathbf{w}) \longleftrightarrow \hat{\eta}(\mathbf{x}, \mathbf{w}, \mathcal{S})], \tag{3.6}$$

where Q is the decision problem. From this, the value of the optimal solution (for the optimization problem Q') can be expressed as

$$\mathrm{opt}_{Q'}(\mathbf{A}) = \max_{\mathcal{S}, U_i} \{ w(U_i) : (\mathbf{A}, \mathcal{S}, U_i) \models \forall \mathbf{w} \forall \mathbf{x} \, [U_i(\mathbf{w}) \longleftrightarrow \hat{\eta}(\mathbf{x}, \mathbf{w}, \mathcal{S})] \}. \tag{3.7}$$

$(\mathbf{A}, \mathcal{S}, U_i)$ above also satisfies expressions where $U_i(\mathbf{w})$ and $\hat{\eta}(\mathbf{x}, \mathbf{w}, \mathcal{S})$ are false — however, since $U_i(\mathbf{w})$ is false, the weight of this tuple $\mathbf{w}$ will not be counted in $w(U_i)$.

Note that (3.7) need not be a Π_1 Horn formula any more, since the Horn property of $\neg\hat{\eta}(\mathbf{x}, \mathbf{w}, \mathcal{S})$ is unknown:

$$[U_i(\mathbf{w}) \longleftrightarrow \hat{\eta}(\mathbf{x}, \mathbf{w}, \mathcal{S})] \equiv [U_i(\mathbf{w}) \vee \neg\hat{\eta}(\mathbf{x}, \mathbf{w}, \mathcal{S})] \wedge [\neg U_i(\mathbf{w}) \vee \hat{\eta}(\mathbf{x}, \mathbf{w}, \mathcal{S})]\,. \tag{3.8}$$

3.2 Example: Weighted Matching

Here, we provide an example of how WEIGHTED MATCHING (optimization version) can be expressed. Given a graph $\mathbf{G}$ with weights on the edges, the objective is to *mark* certain edges such that the sum of the weights on the marked edges is maximized, with the condition that no two adjacent edges in $\mathbf{G}$ can be marked. (In the context of this problem, a *Matched edge* is a synonym for a *Marked edge*.) An instance (structure) $\mathbf{A}$ consists of
(a) the universe A (the union of the set of vertices and the set of tuple-weights),
(b) a relation G (the set of edges),
(c) a relation $C(x)$, which defines whether a variable is a vertex or the weight of a tuple,
(d) and a ternary relation $R(x_0, x_1, x_2)$ that decides whether an edge (x_1, x_2) is assigned a weight of x_0.

Let relation $U(v_i, v_j)$ be true if (v_i, v_j) is a matched edge. Obviously it can be a matched edge only if the edge exists in the given graph. This is expressed by ϕ_0 below. If edge (v_i, v_j) is matched and x is a vertex not in $\{v_i, v_j\}$, then an adjacent edge $G(x, v_i)$ (if it exists in the given graph) cannot be matched. This is expressed by ϕ_1. The three other expressions ϕ_2, ϕ_3 and ϕ_4 perform the same task.

$\phi_1 = U(v_i, v_j) \rightarrow G(v_i, v_j)$,

$\tau = (x \neq v_i) \wedge (x \neq v_j) \wedge U(v_i, v_j)$,

$\phi_1 = \tau \wedge G(x, v_i) \rightarrow \neg U(x, v_i), \quad \phi_2 = \tau \wedge G(v_i, x) \rightarrow \neg U(v_i, x)$,

$\phi_3 = \tau \wedge G(x, v_j) \rightarrow \neg U(x, v_j), \quad \phi_4 = \tau \wedge G(v_j, x) \rightarrow \neg U(v_j, x)$.

Let set of weights $B = \{z \in A \mid \exists x \exists y\, U(x, y) \wedge R(z, x, y)\}$ — however, since B is a set, if the same weight is assigned to two or more edges in U, only one of them will be counted towards total edge weights. Thus there is a need to split B into B_i ($1 \leq i \leq m$, $m =$ number of edges in the input) — a weight w in B_i occurs among i edges in U. Hence[6]

$$\begin{aligned} B_1 &= \{z \in A \mid \exists x\, \exists y\, \forall u\, \forall v\; \tau \wedge U(x, y) \wedge R(z, x, y)\}\,, && (3.9)\\ \text{where } \tau &= \{[(u \neq x) \vee (v \neq y)] \wedge U(u, v)\} \rightarrow \neg R(z, u, v). && (3.10) \end{aligned}$$

(Note: In the definition of B_k below, $\exists_{i=1}^{k} x_i$ is a shorthand for $\exists x_1\, \exists x_2 \cdots \exists x_k$.)

[6]Issues such as quantifier complexity and Horn property are irrelevant for the logic expressions in (3.9)-(3.14). Logic expressions are used here for the sole purpose of defining B_k, $1 \leq k \leq m$.

In general, any B_k ($1 \leq k \leq m$) can be expressed as

$$B_k = \left\{ z \in A \mid \exists_{i=1}^{k} x_i \; \exists_{i=1}^{k} y_i \; \forall u \; \forall v \; \bigwedge_{i=1}^{k} U(x_i, y_i) \bigwedge_{i=1}^{k} R(z, x_i, y_i) \wedge \tau \right\}, \tag{3.11}$$

where $\tau = \tau_1 \wedge \tau_2$, and,

$$\tau_1 = \left\{ \bigwedge_{i=1}^{k} [(u,v) \neq (x_i, y_i)] \wedge U(u,v) \right\} \rightarrow \neg R(z,u,v), \tag{3.12}$$

$$\tau_2 = \bigwedge_{i \neq j} (x_i, y_i) \neq (x_j, y_j) \tag{3.13}$$

$$\{(x_i, y_i) \neq (x_j, y_j)\} \equiv \{(x_i \neq x_j) \vee (y_i \neq y_j)\}. \tag{3.14}$$

Expression (3.13) says that there are k distinct edges (x_i, y_i). Expression (3.14) is an explanation of the shorthand notation used in (3.12) and (3.13) — that if two edges are different, then at least one of their endpoints should be different.

The weight of relation U, $w(U)$, is computed as:

$$w(U) = \left(\sum_{z \in B_1} z \right) + \left(2 \sum_{z \in B_2} z \right) + \cdots + \left(m \sum_{z \in B_m} z \right). \tag{3.15}$$

Finally, Φ is the expression that a solution U should satisfy, and the optimal solution value is obtained by maximizing over all such solutions:

$$\Phi = \forall v_i \; \forall v_j \; \forall x \; [C(v_i) \wedge C(v_j) \wedge C(x)] \rightarrow \bigwedge_{k=0}^{4} \phi_k, \tag{3.16}$$

$$\text{opt}_{Q'}(\mathbf{A}) = \max_{U} \{ w(U) : (\mathbf{A}, U) \models \Phi \}. \tag{3.17}$$

4 Future Research

Expressions similar to those in Section 2.1.1 should be developed for the decision versions of problems in $\mathbf{P_{opt}}$ and $\mathbf{NP_{opt}}$.

A formal proof is needed for the arguments in Sect. 3.1. Furthermore, Sect. 3 studies only maximization problems — research should be carried out for minimization problems as well. Complete problems should be discovered for the respective subclasses. The reductions carried out in this paper are polynomial time; however, logspace reductions are more appropriate, and hence should replace polynomial time reductions.

Since the decision version of the weighted MAXFLOW problem (where arc capacity can be any non-negative integer) is complete for the class $\mathbf{P}$ [8, 10], the optimization version of weighted MAXFLOW is likely to be a complete problem for $\mathbf{P_{opt}}$ — this is yet to be proven.

Acknowledgments

We benefited from discussions with Dov Gabbay of Kings College (London), the theoretical computer science group at the University of Leicester (UK), as well as with J. Radhakrishnan and A. Panconesi at TIFR (Mumbai). We thank the anonymous referee for very helpful comments and suggestions. A preliminary version of the paper was presented at the Algorithms and Complexity in Durham (ACiD 2005) workshop at Durham, UK.

References

[1] G. AUSIELLO, P. CRESCENZI, G. GAMBOSI, V. KANN, A. MARCHETTI-SPACCAMELA, AND M. PROTASI: *Combinatorial Optimization Problems and Their Approximability Properties*. Springer, Germany, 1999. 3

[2] O. BUENO AND P. MANYEM: Polynomial-time maximisation classes: Syntactic hierarchy. *Fundamenta Informaticae*, 84(1):111–133, 2008. 2

[3] H.D. EBBINGHAUS, J. FLUM, AND W. THOMAS: *Mathematical Logic*. Springer, 1994. 3

[4] H.B. ENDERTON: *A Mathematical Introduction to Logic*. Academic Press, 2 edition, 2001. 3, 20

[5] R. FAGIN: Generalized first-order spectra and polynomial-time recognizable sets, 1974. SIAM-AMS Proceedings (no.7). 1, 6

[6] M.R. GAREY AND D.S. JOHNSON: *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, New York, 1979. 15

[7] E. GRÄDEL: The expressive power of second order horn logic. In *STACS 1991: Proceedings of the 8th annual symposium on Theoretical aspects of computer science*, volume 280 of *Lecture Notes in Computer Science*, pp. 466–477. Springer-Verlag, 1991. 1, 2, 6

[8] R. GREENLAW, H. JAMES HOOVER, AND W.L. RUZZO: *Limits to Parallel Computation: P-Completeness Theory*. Oxford University Press, 1995. 15, 22

[9] J. HÅSTAD: Some optimal inapproximability results. In *ACM-STOC 1997: Proceedings of the 29th ACM Symposium on the Theory of Computing*, pp. 1–10, 1997. 15

[10] NEIL IMMERMAN: *Descriptive Complexity*. Springer-Verlag, 1999. 15, 22

[11] S. KHANNA, R. MOTWANI, M. SUDAN, AND U. VAZIRANI: On syntactic versus computational views of approximability. *SIAM Journal of Computing*, 28(1):164–191, 1998. 2

[12] R. KOHLI, R. KRISHNAMURTI, AND P. MIRCHANDANI: The minimum satisfiability problem. *SIAM Journal of Discrete Mathematics*, 7:275–283, 1994. 15

[13] P.G. KOLAITIS AND M.N. THAKUR: Logical definability of np-optimization problems. *Information and Computation*, 115(2):321–353, December 1994. 1, 2, 6, 15, 19

[14] P.G. KOLAITIS AND M.N. THAKUR: Approximation properties of np-minimization problems. *Journal of Computer and System Sciences*, 50:391–411, 1995. 2

[15] A. PANCONESI AND D. RANJAN: Quantifiers and approximation. *Theoretical Computer Science*, 107:145–163, 1993. 2, 4

[16] C.H. PAPADIMITRIOU: *Computational Complexity*. Addison-Wesley, Reading, Massachusetts, 1994. 13, 18

[17] C.H. PAPADIMITRIOU AND M. YANNAKAKIS: Optimization, approximation, and complexity classes. *Journal of Computer and System Sciences*, 43(3):425–440, December 1991. 2, 19

[18] M. ZIMAND: Weighted np-optimization problems: Logical definability and approximation properties. *SIAM Journal of Computing*, 28(1):36–56, 1998. 3, 19

AUTHOR

Prabhu Manyem
School of Information Technology and Mathematical Sciences
University of Ballarat
Mount Helen, VIC 3350, Australia.
Prabhu [dot] Manyem [at] gmail [dot] com