A Model-Theoretic View on Preferences in Declarative Specifications of Search Problems

Alireza Ensan[1], Eugenia Ternovska[2], and Heng Liu[3]

[1] Simon Fraser University, Canada
aensan@sfu.ca
[2] Simon Fraser University, Canada
ter@sfu.ca
[3] Simon Fraser University, Canada
liuhengl@sfu.ca

Abstract. Automated decision making often requires solving difficult and primarily NP-hard problems. In many AI applications (e.g., planning, robotics, recommender systems), users can assist decision making by specifying their preferences over some domain of interest. To take preferences into account, we take a model-theoretic approach to both computations and preferences. Computational problems are characterized as model expansion, that is the logical task of expanding an input structure to satisfy a specification. The uniformity of the model-theoretic approach allows us to link preferences and computations by introducing a framework of a prioritized model expansion. The main technical contribution is an analysis of the impact of preferences on the computational complexity of model expansion. We also discuss how prioritized model expansion can be related to other preference-based declarative programming paradigms. Finally, we study an application of our framework for the case where some information about preferences is missing.

Keywords: Preference Modeling, Model Expansion, Model Theory, Descriptive Complexity

1 Introduction

Solving computationally hard problems (e.g., NP-hard) is in the core of many AI tasks. Due to the significant progress in performance of modern solvers, finding solutions of such problems (e.g., planning, travelling salesman, graph colouring, etc.) has become feasible in many applications. Automated reasoning in intelligent systems often generates a multitude of acceptable results. For example, in the context of resource management, consider the prominent problem of Airport Gate Scheduling [14] that, in a nutshell, is the task of assigning flight arrivals and departures to different gates of an airport. The problem can be formalized as a Constraint Satisfaction Problem (CSP). Some variations of the problem are NP-hard and have been encoded as scheduling or clique partitioning that itself can be transformed to graph colouring problem [4].
Model expansion [27] is the logical task of expanding a mathematical structure (a problem instance) to a solution structure that satisfies a formula (problem specification). The

formula can be written in any specification language with a model-theoretic semantics. The authors showed that other declarative frameworks such as satisfiability problem (SAT), CSP, and answer set programming (ASP) can be encoded as model expansion. By distinguishing between problem instances and problem specifications, model expansion provides a robust modelling framework and establishes a connection to Descriptive Complexity [22].

In many industrial applications, a decision maker may be inclined toward a subset of results. The preferred alternatives can be chosen in automated decision making. For instance, in the Airport Gate Scheduling problem, there could be some preferences for scheduling gates. A certain gate may be preferred to be assigned domestic flights rather than international flights. For language-independent solving, it is crucial to connect model expansion to a preference framework.

Since preferences play a key role in AI, a large number of frameworks for handling preferences have been proposed during the last two decades such as [8, 23, 7, 21, 5, 28, 26]. These proposals are often language-dependent because they are added to a host formal language such as ASP [8, 13] or default logic [12].

In this paper, we propose a model-theoretic approach to handle preferences associated to a model expansion. The main motivation of our work is to connect model theory, descriptive complexity, and preference modelling to study computationally hard optimization problems. To the best our knowledge this is the first proposal of this kind in the literature. We aim to construct a language-independent preference framework that can be added to any declarative language (e.g., SAT, CSP, and ASP) which can be encoded as a model expansion. Toward this goal, we introduce the notion of a prioritized model expansion that extends model expansion by adding a set of preferences of a decision maker. Preferences can be expressed as priorities over ground atoms and then generalized to structures.

Using a model-theoretic approach has promising advantages from both modelling and application viewpoints. For the modelling purposes, properties of model theory [10] can be used to identify certain syntactic fragments of a logic for answering tractable queries with preferences. Moreover, given that model expansion underlines all predominant declarative frameworks such as ASP, SAT, and CSP, prioritized model expansion can model the main preference-based declarative frameworks such as [8], [33], etc. From a practical perspective, by viewing preferences as first order structures, we can use model theory to study properties of preferences and deal with practical cases in declarative programs such as when some information is missing (similar to null values in database systems). We do it by introducing the notion of an *incomplete preference term* that prioritizes some domain elements when some values are unknown. For example, in Airport Gate Scheduling, a decision maker states that gate number 2 is not the worst option for domestic flights, while it is not known what the best option is. We address the problem of finding preferred solutions of search problems in the presence of incomplete preferences. We argue that this problem can be viewed as query answering over incomplete databases. It was demonstrated in [20] that, under some conditions, computing certain answers can equivalently be done using a naive evaluation, that is by evaluating queries directly on incomplete relational databases, by considering unknown values as domain elements. The equivalence is due to the preservation theorems [31]. The main advan-

tage is that naive evaluations drop the computational complexity of computing certain answers dramatically.
A model-theoretic view on modular systems with preferences was presented in [17]. The authors argued that their proposal can express preference statements in other formalisms such as CP-nets [7]. They used Codd's relational algebra [11] to combine modules with preferences. The combination was static – the authors did not focus on the model expansion and its computational complexity.
Our main contributions are as follows. First, we propose the notion of prioritized model expansion that is a declarative framework for specifying computational problems with preferences. Second, we discuss that adding preference even in the simplest formulation leads to the rise of computational complexity of Σ_k^P-complete (k^{th} level in the polynomial hierarchy) model expansions. Third, we study the relations of some preference-based frameworks with prioritized model expansion. We apply the computational complexity result of the task to obtain similar results for those frameworks. Fourth, a method to deal with incomplete information about preferences is proposed. Finally, we show that preservation theorems can be used for finding preferred solutions of problems with incomplete preferences.

2 Background

A τ-structure $\mathcal{A} = (A, R_1^{\mathcal{A}}, ..., R_n^{\mathcal{A}})$ is a tuple where $\tau = \{R_1, ..., R_n\}$ is a relational vocabulary that is a set of non-logical relation symbols R_i with associated arity k_i, A is a domain, and for any $R_i \in \tau$, $R_i^{\mathcal{A}}$ is called the interpretation of R_i and $R_i^{\mathcal{A}} \subseteq A^{k_i}$. For a formula ψ in any logic $\mathcal{L}$, $vocab(\psi)$ denotes the set of vocabulary symbols that appear in ψ. A boolean query Q over τ-structures is defined as a mapping from the set of all τ-structures to $\{0{,}1\}$. A boolean query Q can be related to a model checking task such that for a formula φ in logic $\mathcal{L}$ and $vocab(\varphi) = \tau$, $Q_\varphi(\mathcal{A}) = 1$ if and only if $\mathcal{A} \models \varphi$ where $\mathcal{A}$ is a τ-structure.
Model expansion (MX) is the task of expanding a structure to satisfy a formula in any logic $\mathcal{L}$ [27]. Let us fix a finite relational vocabulary τ and a domain Dom.

Definition 1 ***(Model Expansion MX$_{\mathcal{I}_\sigma,\psi}$)***
Input: *formula ψ in logic $\mathcal{L}$, input vocabulary $\sigma \subseteq vocab(\psi)$, and a σ-structure $\mathcal{I}$ over domain Dom,*
Find *a structure $\mathcal{A}$ where $\mathcal{A} \models \psi$ and expands $\mathcal{I}$. (The decision version: is there a structure $\mathcal{A}$ such that $\mathcal{A} \models \psi$ and $\mathcal{A}$ expands $\mathcal{I}$?)*

We call $\mathcal{A}$ an *expansion structure* of MX$_{\mathcal{I}_\sigma,\psi}$. Any expansion structure $\mathcal{A}$ is a τ-structure with domain Dom.
For any logic $\mathcal{L}$, the data complexity of (the decision version of) model expansion (MX) is always in-between model checking (MC) and satisfiability (SAT). For example, for first-order logic, MC is AC$_0$, MX is NP-complete, SAT is undecidable. The combined complexity of model expansion for first-order logic is NEXPTIME. A complexity analysis of the three tasks (MC, MX, SAT) for several logics of interest was performed in [24]. Note that, when the input vocabulary is empty, MX is often called model generation, and if the input vocabulary is equal to the vocabulary of formula ψ, it is equivalent to model checking.

A variety of problems in AI such as the Airport Gate Scheduling problem can be reduced to graph colouring that is a widely discussed NP-hard problem. Graph colouring can be characterized as a first-order model expansion (i.e, the problem specification is in first-order logic) as follows.

Example 1 *Let binary relation E denotes edges relation between vertices and unary relation symbols $R, G,$ and B denote red, green, and blue colours respectively. The formula ψ specifies three-graph colouring*

$$\begin{gathered}\psi = \forall x\ [(R(x) \vee B(x) \vee G(x)) \\ \wedge\neg((R(x) \wedge B(x)) \vee (R(x) \wedge G(x)) \vee (B(x) \wedge G(x)))] \\ \wedge\ \forall x \forall y\ [E(x, y) \supset (\neg(R(x) \wedge R(y)) \\ \wedge\neg(B(x) \wedge B(y)) \wedge \neg(G(x) \wedge G(y)))].\end{gathered}$$

A graph $\mathcal{G} = (V, E)$ is an instance structure with vocabulary $\sigma = \{E\}$ and domain V that is the set of vertices. The model expansion MX$_{\mathcal{G}_E,\psi}$ finds expansion structure $\mathcal{A}$ (i.e., three-colouring of $\mathcal{G}$) that interprets symbols $R, B,$ and G satisfying ψ. Note that $R, B,$ and G are implicitly second-order-$\exists$ quantified.

$$\underbrace{\overbrace{(V; E^{\mathcal{G}}}^{\mathcal{G}},\ R^{\mathcal{A}}, B^{\mathcal{A}}, G^{\mathcal{A}})}_{\mathcal{A}} \models \psi.$$

In order to study the computational aspect of prioritized model expansion, we employ the notion of a Turing machine as the model of computation.

Definition 2 *Oracle $M_{\mathcal{L}}$ is a Turing machine M augmented by an oracle tape that can decide whether some input string x in the oracle tape belongs to a language $\mathcal{L}$.*

Let X be a complexity class.

Notation 1 *P^X is the class of languages (complexity class) that can be computed by a deterministic Turing machine with an oracle in X.*

Notation 2 *co-X is the complexity class of decision problems whose complements are in X.*

The polynomial hierarchy (PH) that is a hierarchy of complexity classes is defined as $P = \Sigma_0^P = \Pi_0^P = \Delta_0^P$, $\Sigma_{k+1}^P = \mathrm{NP}^{\Sigma_k^P}$, $\Delta_{k+1}^P = P^{\Delta_k^P}$, and $\Pi_{k+1}^P = \mathrm{coNP}^{\Sigma_k^P}$ for $k > 0$.

3 Prioritized Model Expansion

In this section, we introduce the notion of prioritized model expansion which is model expansion with a collection of preferences of a decision maker. To model preferences, we define preference expression as an order over a set of ground atoms. We study the computational complexity of solving problems related to prioritized model expansions including *Dominant Structure* (i.e., given two structures, whether one is preferred to another), *Optimal Expansion* (i.e., given a structure, if it is an optimal expansion of a model expansion task), and *Goal-Oriented Optimal Expansion* (i.e., deciding if there is an optimal expansion satisfying a certain goal). At the end of this section, we define preference term as a generalization of preference expressions.

3.1 Preference Expression

Let us fix a domain Dom. Consider k first order variables $x_1, ..., x_k$ over Dom, vocabulary τ, and k-ary $R \in \tau$. A k-ary tuple $\overline{a} = (a_1, ..., a_k)$ is an assignment to an ordered set of variables $\overline{x} = (x_1, ..., x_k)$ such that for $1 \leq i \leq k$, $a_i \in Dom$. We use the symbol $\overline{a}[x_i]$ to denote value a_i. $R(\overline{a})$ is called a ground atom where $\overline{a} \in Dom^k$. A preference expression P is an order on a set of ground atoms.

Definition 3 ***(Preference Expression)*** *A preference expression P is a pair $P = (\mathcal{S}_\tau, \sqsupseteq_P$) where $\mathcal{S}_\tau$ is the set of all ground atoms of vocabulary τ over domain Dom and $\sqsupseteq_P$ is a preorder on S_τ.*

For ground atoms $R(\overline{a})$ and $T(\bar{b})$ where $R, T \in \tau$, k-ary tuple $\overline{a} \in Dom^k$, and k'-ary tuple $\bar{b} \in Dom^{k'}$, $R(\overline{a}) \sqsupseteq_P T(\bar{b})$ is read as $R(\overline{a})$ is preferred to $T(\bar{b})$. Also, $R(\overline{a})$ is called strictly preferred to $T(\bar{b})$ with notation $R(\overline{a}) \sqsupset_P T(\bar{b})$ if $R(\overline{a}) \sqsupseteq_P T(\bar{b})$ is true and $T(\bar{b}) \sqsupseteq_P R(\overline{a})$ does not hold. Also, $R(\overline{a}) \approx_P T(\bar{b})$ if $R(\overline{a}) \sqsupseteq_P T(\bar{b})$ and $T(\bar{b}) \sqsupseteq_P R(\overline{a})$.

Example 2 *Consider a graph $\mathcal{G} = (V, E)$ and three colours R, B, and G as in Example 1. Suppose $P = (\mathcal{S}_\tau, \sqsupseteq_P)$ where domain $V = \{v_1, v_2, ...v_n\}$ is a finite set of nodes, $\tau = \{E, R, G, B\}$, and $R(v_1) \approx_P R(v_2) \sqsupset_P R(v_3)$ states that it is equally preferred to have v_1 and v_2 with red colour and either of which is strictly preferred to v_3 with red colour. Also, $G(v_2) \sqsupset B(v_3)$ denotes that having v_2 with green colour is favoured to blue v_3.*

The relation between ground atoms can be lifted to a preference order among structures. The idea of comparing two sets based on the preference on their members has been widely discussed in different domains such as in database systems [35] and even beyond the realm of theoretical computer science such as in economic studies and decision theories [9]. Inspired by [35, 1], here, we introduce three different methods to construct a relation $\geq_P$ from $\sqsupseteq_P$ among τ-structures with domain Dom as follows.

Definition 4 ***(Preference Relation on Structures)***
Given a preference expression $P = (\mathcal{S}_\tau, \sqsupseteq_P)$, let $\mathcal{A}$ and $\mathcal{B}$ be two τ-structures with domain Dom,

- ***Weak Pareto (WP).*** *$\mathcal{A} \geq_P^{wp} \mathcal{B}$ iff for all $R, S \in \tau$ and for all $\overline{a} \in R^{\mathcal{A}}$ and all $\bar{b} \in S^{\mathcal{B}}$, $R(\overline{a}) \sqsupseteq_P S(\bar{b})$.*
- ***Upper Bound Dominance (UBD).*** *$\mathcal{A} \geq_P^{ubd} \mathcal{B}$ iff for all $S \in \tau$ and for all $\bar{b} \in S^{\mathcal{B}}$, for some $R \in \tau$, there is $\overline{a}$ such that $\overline{a} \in R^{\mathcal{A}}$ and $R(\overline{a}) \sqsupseteq_P S(\bar{b})$.*
- ***Element Dominance (ED)*** *$\mathcal{A} \geq_P^{ed} \mathcal{B}$ iff for some $R, S \in \tau$, there is $\bar{b} \in S^{\mathcal{B}}$ and there is $\overline{a} \in R^{\mathcal{A}}$ such that $R(\overline{a}) \sqsupseteq_P S(\bar{b})$ and there is not $\overline{c}$ for some $T \in \tau$ such that $\overline{c} \in T^{\mathcal{B}}$ and $T(\overline{c}) \sqsupset_P R(\overline{a})$.*

To build a one-to-one correspondence between our preference model and other preference frameworks in the literature, e.g., [35], [34], [8], etc, we use one of preference semantics Weak Pareto, Upper Bound Dominance, or Element Dominance. Defining these different semantics contributes to the flexibility and generality of our proposal in

which a variety of optimization problems can be encoded. We discuss this generality in more details in section 4.
The strict version of $\geq_P^y$ where $y \in \{$wp, ubd, ed$\}$ is defined as $\mathcal{A} >_P^y \mathcal{B}$ if $\mathcal{A} \geq_P^y \mathcal{B}$ and $\mathcal{B} \geq_P^y \mathcal{A}$ does not hold. $\mathcal{A}$ is dominant to $\mathcal{B}$ based on semantics y where $y \in \{$wp, ubd, ed$\}$ when $\mathcal{A} >_P^y \mathcal{B}$. Also, we say $\mathcal{A}$ is dominant to $\mathcal{B}$ with notation $\mathcal{A} >_P \mathcal{B}$ when there is $y \in \{$ wp, ubd, ed$\}$ such that $\mathcal{A} >_P^y \mathcal{B}$.
The problem of deciding if $\mathcal{A}$ is dominant is called *Dominant Structure* problem that is characterized as follows.

Definition 5 ***(Dominant Structure)***
Input: *a preference expression $P = (\mathcal{S}_\tau, \sqsupseteq_P)$ and τ-structures $\mathcal{A}$ and $\mathcal{B}$ with domain Dom,* ***Question:*** *is $\mathcal{A} >_P \mathcal{B}$?*

Proposition 1 *Solving Dominant Structure problem is polynomial in the size of Dom.*

Proof. As stated by Definition 4, at most, we compare all tuples in $R^{\mathcal{A}}$ and $R^{\mathcal{B}}$ for all $R \in \tau$. The total possible number of k-ary tuples is $|Dom|^k$ where k is the maximum arity of predicate symbols in τ. Therefore, $\mathcal{O}(|Dom|^{2k})$ comparisons are required for each $R \in \tau$. Thus, deciding if $\mathcal{A} >_P \mathcal{B}$ is in $m \cdot \mathcal{O}(|Dom|^{2k})$ (polynomial in the size of Dom) where m is the number of elements in τ.

We remark that the vocabulary τ is fixed and our discussion on computational complexity is focused on data complexity.

3.2 Prioritized Model Expansion

A prioritized model expansion (PMX) is the problem of finding the most preferred expansion structures with respect to preferences.

Definition 6 ***(Prioritized Model Expansion)***
Input: *a formula ψ, input vocabulary $\sigma \subseteq vocab(\psi)$, a σ-structure $\mathcal{I}$, and a preference expression $P = (\mathcal{S}_\tau, \sqsupseteq_P)$,* ***Find*** *structure $\mathcal{A}$ such that $\mathcal{A}$ is an expansion structure of $MX_{\mathcal{I}_\sigma,\psi}$ and there is not an expansion structure $\mathcal{B}$ such that $\mathcal{B} >_P \mathcal{A}$.*

Notation 3 *The problem of prioritized model expansion is denoted by $\Pi_{\mathcal{I}_\sigma,\psi} = (MX_{\mathcal{I}_\sigma,\psi}, P)$ where $MX_{\mathcal{I}_\sigma,\psi}$ is a model expansion and $P = (\mathcal{S}_\tau, \sqsupseteq_P)$ is a preference expression.*

Definition 7 *Any solution to a problem of prioritized model expansion $\Pi_{\mathcal{I}_\sigma,\psi}$ is called an optimal expansion of $\Pi_{\mathcal{I}_\sigma,\psi}$.*

In the rest of this subsection, we study some decision problems that can be associated to a prioritized model expansion. The Optimal Expansion problem asks whether a given structure is an optimal expansion of a prioritized model expansion.

Definition 8 ***(Optimal Expansion)***
Input: *a τ-structure $\mathcal{A}$ and a prioritized model expansion $\Pi_{\mathcal{I}_\sigma,\psi} = (MX_{\mathcal{I}_\sigma,\psi}, P)$ where $\tau = vocab(\psi)$ and $P = (\mathcal{S}_\tau, \sqsupseteq_P)$ is a preference expression.*
Question: *Is $\mathcal{A}$ an optimal expansion of $\Pi_{\mathcal{I}_\sigma,\psi}$?*

Proposition 2 *Let model checking of ψ (given a structure $\mathcal{A}$, decide if $\mathcal{A} \models \psi$) in $MX_{\mathcal{I}_\sigma,\psi}$ be in the complexity class Y. Solving the problem of Optimal Expansion is in co-NP^Y.*

Proof. The complementary problem is deciding if there is an expansion structure $\mathcal{B}$ such that $\mathcal{B} >_P \mathcal{A}$. The complementary problem can be solved by a non-deterministic polynomial Turing machine guessing $\mathcal{B}$ with access to an oracle in Y that decides if $\mathcal{B}$ is an expansion of $\text{MX}_{\mathcal{I}_\sigma,\psi}$ (that includes checking if $\mathcal{B}$ expands $\mathcal{I}$ in polynomial time and if $\mathcal{B} \models \psi$ in the complexity Y) and based on Proposition 1, in polynomial time checks if $\mathcal{B} >_P \mathcal{A}$. Thus, the complementary problem is in NP^Y and the original problem is in co-NP^Y.

One of the common tasks in many AI applications is to determine if a certain goal is achieved by solutions of a problem, e.g., in planning [3]. In the context of prioritized model expansion, it can be translated into whether there is an optimal expansion that satisfies a formula. The problem is formulated as follows.

Definition 9 ***(Goal-Oriented Optimal Expansion)***
Input: *a prioritized model expansion $\Pi_{\mathcal{I}_\sigma,\psi} = (MX_{\mathcal{I}_\sigma,\psi}, P)$ where $\tau = vocab(\psi)$ and $P = (\mathcal{S}_\tau, \sqsupset_P)$ is a preference expression, and a formula ϕ such that ϕ is of the form $R_1(\overline{a}_1) \wedge ... \wedge R_l(\overline{a}_l)$ where $R_i \in \tau$ and $R_i(\overline{a}_i)$ is a ground atom for $1 \leq i \leq l$.*
Question: *Is there an optimal expansion $\mathcal{A}$ of $\Pi_{\mathcal{I}_\sigma,\psi}$ such that $\mathcal{A} \models \phi$?*

Proposition 3 *Let solving the Optimal Expansion problem of $\Pi_{\mathcal{I}_\sigma,\psi} = (MX_{\mathcal{I}_\sigma,\psi}, P)$ be in the complexity class X. Solving the problem of Goal-Oriented Optimal Expansion is in NP^X.*

Proof. First, we non-deterministically guess a τ-structure $\mathcal{A}$ and in polynomial time check if $\overline{a}_i \in R_i^{\mathcal{A}}$, for $1 \leq i \leq l$, that can be done by means of a non-deterministic polynomial Turing machine. Second, we check whether our guess is an optimal expansion that is in the complexity class X by the assumption. Thus, the problem can be solved by a non-deterministic polynomial Turing machine using an oracle in X. Hence, the problem is in NP^X.

Goal-Oriented Optimal Expansion problem can be generalized to finding an optimal expansion that satisfies a formula ϕ in a logic $\mathcal{L}^*$. In this case, the complexity of model checking in logic $\mathcal{L}^*$ (i.e., given a structure $\mathcal{A}$ if $\mathcal{A} \models \phi$?) taken into account. However, for the sake of simplicity, in this paper, we consider the goal ϕ as a conjunction of ground atoms and it can be verified in polynomial time if a structure $\mathcal{A}$ satisfies ϕ.
It was discussed in [27] that any boolean query computable in NP can be expressed as a first order model expansion $\text{MX}_{\mathcal{I}_\sigma,\psi}$ where ψ is a first order formula. Based on Fagin's theorem [18], NP is the class of boolean queries expressible in existential second order logic ($\exists$SO). This shows that a first order MX and existential second order logic have the same expressive power. Similarly, the polynomial hierarchy is the set of boolean queries expressible in second order logic and any query computable in Σ_k^P can be encoded as $\text{MX}_{\mathcal{I}_\sigma,\psi}$ where ψ is a formula of the form $Q_1, ..., Q_{k-1}\psi^*$ such that for $1 \leq i \leq k$, Q_i is a second order quantifier where $Q_i = \forall$ for k is odd and $Q = \exists$ otherwise, and

ψ^* is a first order formula. Therefore, when a decision version of a model expansion $\text{MX}_{\mathcal{I}_\sigma,\psi}$ is in Σ_k^P, model checking of ψ is in Π_{k-1}^P and based on Proposition 2, solving Optimal Expansion $(\text{MX}_{\mathcal{I}_\sigma,\psi}, P)$ is in Σ_k^P. The following result shows the impact of introducing preferences on Goal-Oriented Optimal Expansion for Σ_k^P-complete model expansions.

Theorem 1 *Let the decision version of a model expansion be Σ_k^P-complete. The problem of Goal-Oriented Expansion Structure is Σ_{k+1}^P-complete.*

Proof. The membership to Σ_{k+1}^P follows from the results of Proposition 2, Proposition 3, and properties of model expansion. Since the model expansion is in Σ_k^P, the complexity of model checking of ψ is in Π_{k-1}^P. Therefore, based on proposition 2, the complexity of Optimal Expansion problem is in co-NP$^{\Sigma_{k-1}^P}$ that is equal to Π_k^P. Also, based on Proposition 3, Goal-Oriented Optimal Expansion problem is in NP$^{\Pi_k^P}$ or NP$^{\Sigma_k^P}$ that is equal to Σ_{k+1}^P. For the proof of hardness, we consider the following steps.

First, we show that the problem of deciding the existence of minimal solutions of an abductive logic program [15] satisfying a goal can be reduced to Goal-Oriented Optimal Expansion similar to preferred logic programs [33]. An abductive logic program is defined as $ALP = \langle H, M, \mathcal{P}\rangle$ over a set A of propositional atoms where $\mathcal{P}$ is a logic program, $H \subseteq A$ is called the hypothesis and $M \subseteq A \cup \{\neg a | a \in A\}$ is the manifestation. A solution to ALP is a set $N \subseteq H$ such that there is a stable model S of $\mathcal{P} \cup N$ and $M \subseteq S$. A solution N is called (H) minimal if there is not a solution N' such that $N' \subset N$. For a given hypothesis $h \in H$, deciding if there is a minimal solution N such that $h \in N$ is Σ_2^P-complete.

Second, consider an abductive logic program $ALP = \langle H, M, \mathcal{P}\rangle$. Let us define a logic program $\mathcal{P}'$ as a set of rules of the form $r : R(\overline{a}) \leftarrow \neg S(\overline{b})$ for any $R(\overline{a}) \sqsupseteq_P S(\overline{b})$ such that $S(\overline{b}) \in H$. Rule r indicates the less preferred ground atom (i.e., $S(\overline{b})$) belongs to the set of hypothesis atoms H. Define $\mathcal{P}^* = \mathcal{P} \cup \mathcal{P}'$. The existence of a stable model of a logic program is NP-complete and it can be translated into the decision version of a model expansion as it was shown in [27]. The problem of finding if there is a H minimal solution of ALP can be reduced to deciding whether there is an optimal expansion in $(\text{MX}_{\mathcal{I},\psi}, P)$ where $\mathcal{P}^*$ is translated into $\text{MX}_{\mathcal{I},\psi}$ based on [27]. Assume X_1 and X_2 are two stable models of $\mathcal{P}^*$. If X_1 is preferred to X_2 with respect to one of preference semantics in Definition 4, there is $R(\overline{a}) \in X_1$ and $S(\overline{b}) \in X_2$ such that $R(\overline{a}) \sqsupseteq_P S(\overline{b})$. So, we have $X_1 \cap H \subseteq X_2 \cap H$ and therefore, any preferred answer set is H minimal. Hence, finding a minimal solution of $\langle H, M, \mathcal{P}\rangle$ is equivalent to finding an optimal expansion of $\Pi_{\mathcal{I}_\sigma,\psi}$ that satisfies a goal M. Thus, *Goal-Oriented Optimal Expansion* for a NP-complete MX is Σ_2^P-complete.

Third, for model expansions in the higher levels of the polynomial hierarchy, consider the following. Σ_k^P-complete problems can be encoded as a combined logic program [6]. $\Pi = (\mathcal{P}_g, \mathcal{P}_t)$ is called a combined logic program where $\mathcal{P}_g$ and $\mathcal{P}_t$ are logic programs over a set of propositional variables G and T respectively. M is a model of Π if it is a stable model of $\mathcal{P}_g$ and there is not a stable model N of $\mathcal{P}_t$ such that $M \cap G = N \cap T$. The decision version of this problem is Σ_2^P-complete. Recursively, the existence of a model of a combined program in depth 2 defined as $\mathit{\Pi}_2 = (\mathcal{P}_{g_2}, (\mathcal{P}_{g_1}, \mathcal{P}_t))$ is $\mathit{\Sigma}_3^P$-complete and similarly, in depth k, the existence of a model of $(\mathcal{P}_{g_{k-1}}, \mathit{\Pi}_{k-2})$ is $\mathit{\Sigma}_k^P$-complete. We introduce abductive combined program as $\mathcal{C} = \langle H, M, \mathit{\Pi} \rangle$ where $\mathit{\Pi} = (\mathcal{P}_g, \mathcal{P}_t)$ is a combined logic program. W is a solution of $\mathcal{C}$ if there is a model S of $(\mathcal{P}_g \cup W, \mathcal{P}_t)$ such that $M \subseteq S$. W is minimal if there is not a solution W' such that $W' \subset W$.

Lemma 1 *The problem of deciding if* $\mathcal{C} = \langle H, M, \mathit{\Pi}_k \rangle$ *for a given* $h \in H$ *has a minimal solution containing* h *is* $\mathit{\Sigma}_{k+1}^P$*-complete.*

Proof: The proof includes a translation from a quantified boolean formula (QBF) to $\mathcal{C}$ for $k = 2$ and then by induction on k for $k > 2$, the result follows. Let φ be a boolean formula in CNF and $X = \{x_1, ..., x_m\}$, $W = \{w_1, ..., w_m\}$, $X' = \{x'_1, ..., x'_m\}$, $Y = \{y_1, ..., y_n\}$, and $Z = \{z_1, ..., z_l\}$ be a set of boolean variables in φ. Let t, h, and f be also boolean variables. Consider $\mathcal{P}_g$ as a set of rules of the form $\{t \leftarrow x_i, x'_i\}$, $\{w_i \leftarrow x_i\}$, $\{w_i \leftarrow x'_i\}$, $\{t \leftarrow y_1, ...y_n, h\}$, and$\{f \leftarrow l_1, ..., l_r\}$ where $\neg(l_1 \wedge, ..., \wedge l_r) \in \varphi$ similar to [15]. For $X \cup X' \subseteq H$, a H-minimal solution of $\langle H, \{t\} \cup W, \mathcal{P}_g \rangle$ does not contain f and it has either x_i or x'_i. On the other hand, similar to [6], assume $\mathcal{P}_t$ determines the truth value of a set of boolean variables Z. Also, for each clause $C \in \varphi$, suppose that $\mathcal{P}_t$ includes a set of rules of the form $t \leftarrow \neg C$ and $f \leftarrow \neg f, t$ that means t must not be in any stable model of $\mathcal{P}_t$. This implies that the validity of $\exists X \forall Y \exists Z \varphi$ is equivalent to the existence of a H-minimal solution of $\mathcal{C}$ that contains h. So, for $k = 2$, the existence of a minimal solution to an abductive combined logic program containing an atom h is $\mathit{\Sigma}_3^P$-complete.

■

Finally, according to the previous steps and Lemma 1, finding a minimal solution of an abductive combined logic program in level k can be translated into a *Goal-Oriented Optimal Expansion* where the model expansion is $\mathit{\Sigma}_k^P$-complete and hence the result follows.

Theorem 1 presents an important consequence of adding preferences to a $\mathit{\Sigma}_k^P$-complete model expansion. For the problem of deciding if there is an expansion that satisfies a goal ϕ, adding preferences leads to a jump in the polynomial hierarchy. So, the preference relation between expansion structures derived from a preference expression can not be translated into axiomatization ψ in polynomial time unless P=NP or the polynomial hierarchy collapses.

Example 3 *Consider the problem of graph colouring that was described as model expansion in Example 1. Let* $\mathcal{G} = (V, E)$ *be the input graph where* $V = \{v_1, v_2, v_3, v_4, v_5\}$ *and* $E^{\mathcal{G}} = \{(v_1, v_2), (v_1, v_3), (v_2, v_3), (v_2, v_4), (v_4, v_5), (v_3, v_5)\}$*. Assume that we prefer red colour for* v_1*. Also,* v_4 *with red colour is favoured to* v_5 *with red colour and blue* v_2 *is preferred to green* v_2*. These preference statements can be encoded by a preference expression* P *such that* $R(v_1) \sqsupset_P B(v_1)$ *and* $R(v_1) \sqsupset_P G(v_1)$*. Also,* $R(v_4) \sqsupset_P R(v_5)$ *and* $B(v_2) \sqsupset_P G(v_2)$*. The prioritized model expansion* $\mathit{\Pi}_{\mathcal{G},\psi} = (MX_{\mathcal{G},\psi}, P)$ *where* $MX_{\mathcal{G},\psi}$ *is the characterization of three-colouring for input graph* $\mathcal{G}$ *and* P *is the preference expression. The input graph* $\mathcal{G}$ *has 18 possible three-colourings. Among these solutions,* $\mathcal{A}$ *is an optimal expansion of* $\mathcal{G}$ *(based on Element Dominance semantics) where* $R^{\mathcal{A}} = \{v_1, v_4\}$*,* $B^{\mathcal{A}} = \{v_2, v_5\}$*, and* $G^{\mathcal{A}} = \{v_3\}$*.*

3.3 Preference Term

In this subsection, we extend the notion of a preference expression by introducing preference term that is an order on the domain of a first order variable. Preference terms can be related to the concept of preferences in relational database systems. However, unlike databases [23], where the goal is to find the most preferred records in a table, here we aim to find preferred expansion structures. A preference relation among tuples (and therefore among ground atoms that is specified as a preference expression) is derived from a set of preference terms. The usefulness of defining preferences over domains of variables is to deal with incomplete information about preferences that will be discussed in details in Section 5. A preference term p is defined as follows.

Definition 10 ***(Preference Term)***
A preference term p is a pair $p = (A, \succeq)$ where A is a set and $\succeq$ is a preorder over A.

For $a, b \in A$, $a \succeq b$ means that a is preferred to b. Also, $a \sim b$ if $a \succeq b$ and $b \succeq a$. Moreover, $a \succ b$ (a is strictly preferred to b) when $a \succeq b$ and $b \succeq a$ does not hold.
Consider first order variables $x_1, ..., x_k$. The symbol $dom(x)$ denotes the domain of x. Assume we are given k preference terms $p_{x_1} = (dom(x_1), \succeq_1), ..., p_{x_k} = (dom(x_k), \succeq_k$). A preference relation $\gg$ over k-ary tuples $\overline{a}$ and $\overline{b}$ is constructed from $p_{x_1}, ..., p_{x_k}$ as follows.

Definition 11 ***(Preference Relation on Tuples)***
Given a set of preference terms $\mathfrak{P} = \{p_{x_1}, ..., p_{x_k}\}$ and two k-ary tuples $\overline{a}$ and $\overline{b}$, we say that $\overline{a}$ is preferred to $\overline{b}$ with respect to $\mathfrak{P}$, notation $\overline{a} \gg_{\mathfrak{P}} \overline{b}$, iff for all $p_i \in \mathfrak{P}$, $\overline{a}[x_i] \succeq_i \overline{b}[x_i]$.

A preference expression P can be constructed from a set of preference terms $\mathfrak{P}$ such that for a predicate symbol R, $R(\overline{a}) \sqsupseteq_P R(\overline{b})$ if and only if $\overline{a} \gg_{\mathfrak{P}} \overline{b}$. Prioritized model expansion then is extended to $\Pi_{\mathcal{I}_\sigma}, \psi = (\mathrm{MX}_{\mathcal{I}_\sigma, \psi}, \mathfrak{P})$ where $\geq_{\mathfrak{P}}$ is derived from a set of preference terms and the semantics of optimal expansion is exactly as before.

4 Prioritized Model Expansion and Declarative Specifications of Optimization Problems

In this section we study some examples of declarative specification of optimization problems that can be encoded as a prioritized model expansion and the associated computational complexity can be determined by the result of Theorem 1.
First, consider a prioritized logic program (PLP) [33] as one of the first examples of logic programming with preferences. A PLP is a pair (Pr, Φ) where Pr is a standard logic program with stable model semantics and Φ is a set of preference relations among atoms of the form of $a \succeq b$ that means a is preferred to b. The transitive closure of Φ is denoted by Φ_c. The reflexive transitive binary relation $\sqsupseteq$ among answer sets of Pr is defined as: $X_1 \sqsupseteq X_2$ if there exist $a \in X_1 - X_2$ and $b \in X_2 - X_1$ where $(a \succeq b) \in \Phi_c$ and there is not $d \in X_1 - X_2$ such that $(b \succ d) \in \Phi_c$. X is called a preferred answer set if there is not an answer set Y such that $Y \sqsupset X$. The following result states that any PLP can be encoded as a prioritized model expansion and presents the computational complexity of corresponding prioritized model expansion.

Theorem 2 *A PLP $\Gamma = (Pr, \Phi)$ can be expressed by a prioritized model expansion $\Pi_{\mathcal{I},\psi} = (MX_{\mathcal{I},\psi}, P)$ where $\mathcal{I}$ represents Pr, ψ characterizes the stable model semantics, and P is a preference expression representing Φ_c such that $\mathcal{A}$ is an optimal expansion of $\Pi_{\mathcal{I},\psi}$ if and only if there is a preferred answer set M in Γ that can be constructed in polynomial time from $\mathcal{A}$. Deciding the existence of an optimal expansion of $\Pi_{\mathcal{I},\psi}$ that satisfies a goal ϕ and a preferred answer set of PLP satisfying ϕ are Σ_2^P-complete.*

Proof. Φ can be viewed as a preference expression in PMX and a program Pr as a first order model expansion. According to [27], there is a correspondence between expansion structures of MX$_{\mathcal{I}}, \psi$ and stable models of Pr such that any stable model of Pr can be one-to-one mapped in polynomial time to an answer set of Pr and vice versa. Optimal expansions of $\Pi_{\mathcal{I},\psi}$ are computed based on Element Dominance semantics that is the same as the notion of preferred answer sets in PLP. Also, since the computational complexity of brave reasoning (i.e., the existence of an answer set that satisfies a conjunction of atoms) is NP-complete, based on Theorem 1, deciding if there is a preferred answer sets of Γ that satisfies ϕ is Σ_2^P-complete.

Second, an ASO program [8] is a pair (P_g, R) where P_g is a generating logic program and R is a set of rules of the form $r: \quad C_1 > ... > C_k \leftarrow a_1, ..., a_n, not\ b_1, ..., not\ b_m$. In each rule, a_i and b_i are literals. Also, C_i is a combination of atoms integrated through conjunction, disjunction, default negation (not) and strong negation ($\neg$) that must appear only before atoms. $C_i > C_j \leftarrow body$ means that if $body$ is satisfied, C_i is preferred to C_j. Given a set of l rules $r_1, ..., r_l$, each answer set M of P_g is associated with a satisfaction vector $d(M) = \langle d_1(M), ..., d_l(M)\rangle$ where $d_i(M)$ is called satisfaction degree of M in r_i. Satisfaction degree denotes the minimum j of C_js in r_i that are satisfied by M. Preorder preference relation $\leq$ is defined over satisfaction degrees of a rule r_k and two answer sets M_1 and M_2 as $d_k(M_1) \geq d_k(M_2)$ when q is the minimum i for all C_is in M_1 and s is the minimum j for all C_js in M_2 and $q < s$. Let M_1 and M_2 be two answer sets of P_g. M_1 is preferred to M_2 with respect to R (notation $M_1 \succeq M_2$) if for all $i \leq l$, $d_i(M_1) \geq d_i(M_2)$. The relation between ASO and prioritized model expansion is formulated as follows.

Theorem 3 *Let $ASO = (P_g, R)$ be an ASO program where P_g is disjunctive program and R is a set of preference rules. There is a prioritized model expansion $\Pi_{\mathcal{I},\psi} = (MX_{\mathcal{I},\psi}, P)$ where ψ specifies the stable model semantics and $\mathcal{I}$ represents P_g such that each optimal expansion $\mathcal{A}$ of $\Pi_{\mathcal{I},\psi}$ can be mapped in polynomial time to a preferred answer set of ASO. The problem of deciding if there is an optimal expansion of $\Pi_{\mathcal{I},\psi}$ that satisfies a goal ϕ or there is a solution of ASO satisfying ϕ is Σ_3^P-complete.*

Proof. An ASO program can be translated into a PLP (Pr^*, Φ) where Pr^* is the union of P_g and a set of rules r_1^* and r_2^* where for each rule $r \in R$ of the form $C_1 > C_2 \leftarrow$ body(r), we have $r_1^* : n_1 \leftarrow C_1, \text{body}(r)$ and $r_2^* : n_2 \leftarrow C_2, \text{body}(r)$ and $n_2 \succeq n_1$ is in Φ. Therefore, based on Theorem 3, ASO can be converted into a prioritized model expansion. Since the existence of a an answer set that satisfies a set of ground atoms is Σ_2^P-complete (i.e., brave reasoning in disjunctive logic programs with stable model semantics), deciding the existence of a solution of ASO that satisfies a goal formula ϕ is Σ_3^P-complete.

Finally, our proposal can be also connected to a variety of other preference frameworks such as CP-nets [7] that model conditional preferences. A CP-net can be translated into an ASO (see [8]) and hence based on Theorem 4 can be encoded as a prioritized model expansion. Also, in the context of planning, preferences about temporal properties of plans can be converted into a logic program with preferences [34] and thus to a prioritized model expansion (based on the result of Theorem 3).

5 Incomplete Preferences

In this section, we focus on handling incomplete preference statements associated to search problems that are common in many practical cases. For example, assume that for buying a car, the customer believes that the red colour is not the best option and it is not the last choice either. In this example, some information is not known, that is to determine what colour is better or worse than red. Dealing with missing information is a broad field of study in database systems and declarative programming frameworks [19, 29]. The information incompleteness might be caused by updating databases [16], exchanging information [25], incomplete knowledge of the user about entire domain of interest, etc. The notion of an incomplete preference term is defined as follows.

Definition 12 ***(Incomplete Preference Term)***
An incomplete preference term p_x^{inc} is a pair $p_x^{inc} = (dom(x) \cup \bot, \succeq)$ where $\succeq$ is a preorder on $dom(x) \cup \bot$ and $\bot = \{\bot_1, ..., \bot_m\}$ is a set of nulls.

An element $\bot_j \in \bot$ represents a missing (or unknown) domain element in p_x^{inc} and $e \succ_i \bot_j$ means that value e assigned to x_i is preferred to something that is unknown or e is not the worst option to choose for x_i. Likewise, $\bot_j \succ_i e$ means that e is not the best value of x_i.
We define valuation $v(p_x^{\text{inc}})$ as a function $v : dom(x) \cup \bot \rightarrow dom(x)$ that assigns elements in $dom(x)$ to members of $\bot$ appearing in p_x^{inc} such that $v(e) = e$ for any $e \in dom(x)$ and $v(\bot_i) \in dom(x)$ for any $\bot_i \in \bot$.
A preference term $p_x = (dom(x), \succeq)$ can be viewed as a first order structure with vocabulary $\succeq$ and domain $dom(x)$. An incomplete preference term corresponds to the well known notion of incomplete databases in which some fields value are null [29]. The completion of an incomplete preference term under closed world semantics is the set of all possible preference terms that are constructed by assignment of domain elements to nulls.

Definition 13 *The completion of an incomplete preference term p_x^{inc} under closed world semantics, notation $[\![p_x^{inc}]\!]$, is defined as a set of preference terms as follows:*
$[\![p_x^{inc}]\!] = \{v(p_x^{inc}) | v \text{ is a valuation } \}$.

Example 4 *Assume vocabulary $\{car\}$ with variables model, colour, and fuel that specifies properties of a car. A car dealership has a set of cars that is equivalent to an interpretation of car. Let {Ford, Toyota, Honda} be the domain of model and {red, black, white, gray} be the domain of colour. Also, fuel can be gas, diesel, or electric. A set of preference terms are defined as follows. $p_m = (dom(m), \succeq_m)$ where m denotes the model and Honda $\succeq_m$ Toyota $\succeq_m$ Ford, and $p_c = (dom(c), \succeq_c)$ is an incomplete* *preference term over possible colours of a car where $\perp_1 \succeq_c$ red that means that red is not the best colour. Also, $p_f = (dom(f), \succeq_f)$ where gas $\succeq_f$ diesel and gas $\succeq_f$ electric indicating the favorite cars are those using gas instead of electric power or diesel. For a valuation v where $v(\perp_1)$=black, a black Honda with gas fuel is favoured to an electric red Ford.*

Model expansion task can be coupled with incomplete preference terms as follows.

Definition 14 *Incomplete prioritized model expansion is denoted by a pair $\Pi^{inc}_{\mathcal{I}_\sigma,\psi} = (MX_{\mathcal{I}_\sigma,\psi}, \mathfrak{P}^{inc})$ where $MX_{\mathcal{I}_\sigma,\psi}$ is a model expansion and $\mathfrak{P}^{inc} = \{p^{inc}_{x_1}, ..., p^{inc}_{x_k}\}$ is a set of incomplete preference terms.*

The completion of an incomplete prioritized model expansion is defined as follows.

Definition 15 *The completion of $\Pi^{inc}_{\mathcal{I}_\sigma,\psi} = (MX_{\mathcal{I}_\sigma,\psi}, \mathfrak{P}^{inc})$ under closed world semantics (denoted by $[\![\Pi^{inc}_{\mathcal{I}_\sigma,\psi}]\!]$) is a set of prioritized model expansions defined as $[\![\Pi^{inc}_{\mathcal{I}_\sigma,\psi}]\!] = (MX_{\mathcal{I}_\sigma,\psi}, [\![\mathfrak{P}^{inc}]\!])$ where $[\![\mathfrak{P}^{inc}]\!] = \{[\![p^{inc}_{x_1}]\!], ..., [\![p^{inc}_{x_k}]\!]\}$.*

Structure $\mathcal{A}$ is called an optimal expansion of $\Pi^{\text{inc}}_{\mathcal{I}_\sigma,\psi} = (\text{MX}_{\mathcal{I}_\sigma,\psi}, \mathfrak{P}^{\text{inc}})$ if $\mathcal{A}$ is an optimal expansion of all $\Pi' \in [\![\Pi^{\text{inc}}_{\mathcal{I}_\sigma,\psi}]\!]$. Determining if a given structure is an optimal expansion of an incomplete prioritized model expansion is defined as follows.

Definition 16 *(**Optimal Expansion of Incomplete PMX) Input:** a τ-structure $\mathcal{A}$ with domain Dom and an incomplete prioritized model expansion $\Pi^{inc}_{\mathcal{I}_\sigma,\psi} = (MX_{\mathcal{I}_\sigma,\psi}, \mathfrak{P}^{inc})$ where $\tau = vocab(\psi)$ and $\mathfrak{P}^{inc} = \{p^{inc}_{x_1}, ..., p^{inc}_{x_k}\}$ is a set of incomplete preference terms,*
***Question:** Is $\mathcal{A}$ an optimal expansion of $\Pi^{inc}_{\mathcal{I}_\sigma,\psi}$?*

Recall that one of the advantages of using model-theoretic semantics is to utilize model theory [10]. Here we employ some results of model theory including preservation theorems to study the computational complexity of Optimal Expansion of incomplete prioritized model expansion. Preservation theorems show the relations between syntactic and semantic restrictions of first order formulas [30]. Results are not limited to first order logic and have been extended to first order logic with fixed point [2], Datalog [32], etc. We aim to show that the problem of dominant structure in the presence of incomplete preferences can be reduced to naive evaluation of a formula over an incomplete structure and by using preservation properties of first order sentences and results in [20] to solve the problem of dominant structure with incomplete preferences in polynomial in the size of Dom. Before proceeding, we remind the formal definition of homomorphism and preservation of a formula. Assume $\mathcal{A}$ and $\mathcal{B}$ are two τ-structures. Let $dom(\mathcal{A})$ and $dom(\mathcal{B})$ be the domain of $\mathcal{A}$ and $\mathcal{B}$. A function $h : dom(\mathcal{A}) \rightarrow dom(\mathcal{B})$ such that for each $R \in \tau$ and for all $(a_1, ...a_n)$ $(a_i \in dom(\mathcal{A})$ for $1 \leq i \leq n)$, if $(a_1, ...a_n) \in R^{\mathcal{A}}$, then $(h(a_1), ..., h(a_n) \in R^{\mathcal{B}})$, is called a homomorphism from $\mathcal{A}$ to $\mathcal{B}$. Let ϕ be a first-order sentence (all variables are bounded by a quantifier). We say that ϕ is preserved under homomorphism if for any structure $\mathcal{A}$ such that $\mathcal{A} \models \phi$, then for all structures $\mathcal{B}$ that there is a homomorphism h from $\mathcal{A}$ to $\mathcal{B}$, $\mathcal{B} \models \phi$.
Let $\perp = \{\perp_1, ..., \perp_n\}$ be a set of constants (nulls) and α be a vocabulary of symbols. An α-structure $\mathcal{C}$ with domain $Dom \cup \perp$ is called an incomplete structure. Each element of $\perp$ is a null value that represents a missing domain value. Given a first order sentence ϕ

and a structure $\mathcal{C}$, the model checking task (query evaluation) is to decide whether $\mathcal{C} \models \phi$. Naive model checking, according to [29], treats null values as domain elements. It was discussed in [20] that evaluating queries directly on incomplete relational databases is equivalent to standard evaluation of queries if some syntactic restrictions are imposed to the query language.

Theorem 4 *Let model checking of ψ in $MX_{\mathcal{I}_\sigma,\psi}$ be in the complexity class C. Solving the problem of Optimal Expansion of Incomplete prioritized model expansion is in co-NP^{C}.*

Proof. In order to prove Theorem 4, it suffices to show that for two τ-structures $\mathcal{A}$ and $\mathcal{B}$ with domain Dom, $\mathcal{A} \geq_{\mathfrak{P}^{\text{inc}}} \mathcal{B}$ can be decided in polynomial time in the size of Dom and the rest is only to follow the steps taken in the proof of Proposition 2. To this end, we take the following steps. First, we construct an incomplete structure called preference structure as follows. Let $p^{\text{inc}}_{x_1} = (dom(x_1) \cup \bot, \succeq_1)$ be an incomplete preference term. We construct the following incomplete α-structure $\mathcal{C}$ with domain $Dom' = Dom \cup \bot$ as follows. Consider vocabulary $\alpha = \{p_1, S_{\mathcal{A}}, S_{\mathcal{B}}\}$ and $p^{\mathcal{C}}_1 = \{(a_1, a_2)|a_1, a_2 \in Dom'$ and $a_1 \succeq_1 a_2\}$. Also, $S_{\mathcal{A}} = R^{\mathcal{A}}$. Define $S^1_{\mathcal{B}} = R^{\mathcal{B}}$ and $S^2_{\mathcal{B}} = \{\overline{a}|\exists\overline{b} \in S^1_{\mathcal{B}}; \overline{a}[x_2, ..., x_n] = \overline{b}[x_2, ..., x_n] \wedge \overline{a}[x_1] \in \bot\}$. Let define $S_{\mathcal{B}} = S^1_{\mathcal{B}} \cup S^2_{\mathcal{B}}$. Predicate p_1 represents incomplete preferences among values of variable x_1, $S_{\mathcal{A}}$ is the set of all tuples in $\mathcal{R}^{\mathcal{A}}$, and $S^{\mathcal{B}}$ is the set of all tuples in $R^{\mathcal{B}}$ plus all tuples in $\mathcal{B}$ whose value of x_1 is replaced by a null.

Second, by slightly abuse of notation, valuation $v(\mathcal{C})$ means valuation of any null value in $\mathcal{C}$ similar to the valuation of a preference term. Valuation v is a strong onto homomorphism (v is a surjective function) from $\mathcal{C}$ to $\mathcal{C}'$ where $dom(\mathcal{C}) = Dom \cup \bot$ and $dom(\mathcal{C}') = Dom$. The symbol $[\![\mathcal{C}]\!]$ denotes the set of all $\mathcal{C}'$ where there is a valuation v (strong onto homomorphism) such that $v(\mathcal{C}) = \mathcal{C}'$. Any $\mathcal{C}' \in [\![\mathcal{C}]\!]$ is called a completion of $\mathcal{C}$ under closed world semantics.

Third, according to Definition 4, Definition 10, and Definition 11, the task of deciding if $\mathcal{B}$ is preferred to $\mathcal{A}$ can be reduced to evaluation of whether $\mathcal{C} \models \forall x_1 \forall \overline{x}(S_{\mathcal{A}}(x_1, \overline{x})) \rightarrow \exists y_1 \overline{y}(S_{\mathcal{B}})(y_1, \overline{y}) \wedge p_1(y_1, x_1))$ where $\overline{x} = (x_2, ...x_n)$ and $\overline{y} = (y_2, ..., y_n)$. It was discussed in [20] that a positive first order formula φ (with no negation) with a guard of the form of $\forall \overline{x}(R(\overline{x}) \rightarrow \varphi)$, where R is a relation, is preserved under strong onto homomorphism. ϕ is a positive formula with a guard and completion of $\mathcal{C}$ under closed world semantics is equivalent to a set of structures to which there is a strong onto homomorphism from $\mathcal{C}$. Thus, instead of evaluating all completions of $\mathcal{C}$, we can naively evaluate ϕ. Since naive evaluation is in polynomial time in the size of Dom, evaluation of ϕ over all completions of $\mathcal{C}$ is polynomial in the size of Dom. Hence, deciding if $\mathcal{B} \geq_{p^{\text{inc}}_{x_1}} \mathcal{A}$ is polynomial in the size of Dom.

For a set of incomplete preference terms $\mathfrak{P}^{\text{inc}} = \{p^{\text{inc}}_{x_1}, ..., p^{\text{inc}}_{x_k}\}$, by induction on the number of incomplete preference terms, deciding $\mathcal{B} \geq_{\mathfrak{P}^{\text{inc}}} \mathcal{A}$ is polynomial in the size of Dom. The immediate result similar to Proposition 1 is that for a model expansion MX$_{\mathcal{I},\psi}$ with model checking of ψ in the complexity C, deciding if a given structure is a preferred expansion based on a set of incomplete preference terms is in co-NP$^{\text{C}}$. The following result immediately follows from Theorem 4.

Corollary 1 *Let solving the decision version of model expansion $MX_{\mathcal{I}_\sigma,\psi}$ be Σ^p_k-complete. Solving the problem of Goal-Oriented Optimal Expansion with incomplete prioritized model expansion is Σ^p_{k+1}-complete.*

6 Conclusion

We proposed a novel language-independent preference framework and connected it to model expansion for characterizing preference-based computational decision and search problems. We demonstrated that adding preferences raises the computational complexity of deciding the existence of an expansion structure satisfying a goal. Additionally, we introduced a new method to model incomplete preferences and used model theory results, namely preservation theorems, to find preferred expansions more efficiently. In future work, we will devise an algorithm that solves prioritized model expansion using generic solvers empowered by propagators. The solver would provide symbolic explanations for rejecting and accepting models, and would follow a preferred computation path to prune the search space.

References

[1] Amgoud, L., Vesic, S.: Repairing preference-based argumentation frameworks. In: IJCAI. pp. 665–670. Citeseer (2009)

[2] Atserias, A., Dawar, A., Kolaitis, P.G.: On preservation under homomorphisms and unions of conjunctive queries. Journal of the ACM (JACM) **53**(2), 208–237 (2006)

[3] Baier, J.A., McIlraith, S.A.: Planning with preferences. AI Magazine **29**(4), 25–37 (2008)

[4] Bhasker, J., Samad, T.: The clique-partitioning problem. Computers & Mathematics with Applications **22**(6), 1–11 (1991)

[5] Bienvenu, M., Fritz, C., McIlraith, S.A.: Specifying and computing preferred plans. Artificial Intelligence **175**(7-8), 1308–1345 (2011)

[6] Bogaerts, B., Janhunen, T., Tasharrofi, S.: Stable-unstable semantics: beyond np with normal logic programs. Theory and Practice of Logic Programming **16**(5-6), 570–586 (2016)

[7] Boutilier, C., Brafman, R.I., Domshlak, C., Hoos, H.H., Poole, D.: Cp-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements. J. Artif. Intell. Res.(JAIR) **21**, 135–191 (2004)

[8] Brewka, G., Niemelä, I., Truszczynski, M.: Answer set optimization. In: IJCAI. vol. 3, pp. 867–872 (2003)

[9] Censor, Y.: Pareto optimality in multiobjective problems. Applied Mathematics and Optimization **4**(1), 41–59 (1977)

[10] Chang, C.C., Keisler, H.J.: Model theory, vol. 73. Elsevier (1990)

[11] Codd, E.F.: Extending the database relational model to capture more meaning. ACM Transactions on Database Systems (TODS) **4**(4), 397–434 (1979)

[12] Delgrande, J., Schaub, T.: Expressing preferences in default logic. Artificial Intelligence **123**(1), 41–87 (2000)

[13] Delgrande, J., Schaub, T., Tompits, H.: Logic programs with compiled preferences. arXiv preprint cs/0003028 (2000)

[14] Dorndorf, U., Drexl, A., Nikulin, Y., Pesch, E.: Flight gate scheduling: State-of-the-art and recent developments. Omega **35**(3), 326–334 (2007)

[15] Eiter, T., Gottlob, G., Leone, N.: Abduction from logic programs: Semantics and complexity. Theoretical computer science **189**(1-2), 129–177 (1997)

[16] Elmasri, R., Navathe, S.: Fundamentals of database systems. Pearson (2015)

[17] Ensan, A., Ternovska, E.: Modular systems with preferences. In: IJCAI. pp. 2940–2947 (2015)

[18] Fagin, R.: Generalized first-order spectra and polynomial-time recognizable sets (1974)

[19] Gelfond, M.: Logic programming and reasoning with incomplete information. Annals of mathematics and artificial intelligence **12**(1), 89–116 (1994)

[20] Gheerbrant, A., Libkin, L., Sirangelo, C.: Naive evaluation of queries over incomplete databases. ACM Transactions on Database Systems (TODS) **39**(4), 31 (2014)

[21] Gonzales, C., Perny, P., Dubus, J.P.: Decision making with multiple objectives using gai networks. Artificial Intelligence **175**(7-8), 1153–1179 (2011)

[22] Immerman, N.: Descriptive complexity. Graduate texts in computer science, Springer (1999). https://doi.org/10.1007/978-1-4612-0539-5, `http://dx.doi.org/10.1007/978-1-4612-0539-5`

[23] Kießling, W.: Foundations of preferences in database systems. In: Proceedings of the 28th international conference on Very Large Data Bases. pp. 311–322. VLDB Endowment (2002)

[24] Kolokolova, A., Liu, Y., Mitchell, D., Ternovska, E.: On the complexity of model expansion. In: Proc., 17th Int'l Conf. LPAR. pp. 447–458 (2010)

[25] Libkin, L.: Data exchange and incomplete information. In: Proceedings of the twenty-fifth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems. pp. 60–69. ACM (2006)

[26] Mindolin, D., Chomicki, J.: Contracting preference relations for database applications. Artificial Intelligence **175**(7-8), 1092–1121 (2011)

[27] Mitchell, D.G., Ternovska, E.: A framework for representing and solving np search problems. In: AAAI. pp. 430–435 (2005)

[28] Pini, M.S., Rossi, F., Venable, K.B., Walsh, T.: Incompleteness and incomparability in preference aggregation: Complexity results. Artificial Intelligence **175**(7-8), 1272–1289 (2011)

[29] Reiter, R.: Towards a logical reconstruction of relational database theory. In: On conceptual modelling, pp. 191–238. Springer (1984)

[30] Rosen, E., Weinstein, S.: Preservation theorems in finite model theory. In: Logic and computational complexity. pp. 480–502. Springer (1995)

[31] Rossman, B.: Existential positive types and preservation under homomorphisms. In: Logic in Computer Science, 2005. LICS 2005. Proceedings. 20th Annual IEEE Symposium on. pp. 467–476. IEEE (2005)

[32] Rudolph, S., Thomazo, M.: Expressivity of datalog variants–completing the picture. In: 25th IICAI (2016)

[33] Sakama, C., Inoue, K.: Prioritized logic programming and its application to commonsense reasoning. Artificial Intelligence **123**(1), 185–222 (2000)

[34] Son, T.C., Pontelli, E.: Planning with preferences using logic programming. In: Logic Programming and Nonmonotonic Reasoning, pp. 247–260. Springer (2004)

[35] Staworko, S., Chomicki, J., Marcinkowski, J.: Prioritized repairing and consistent query answering in relational databases. Annals of Mathematics and Artificial Intelligence **64**(2-3), 209–246 (2012)