Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2015)

Modular Systems with Preferences

Alireza Ensan and **Eugenia Ternovska**
Simon Fraser University
Canada
{aensan,ter}@sfu.ca

Abstract

We propose a versatile framework for combining knowledge bases in modular systems with preferences. In our formalism, each module (knowledge base) can be specified in a different language. We define the notion of a preference-based modular system that includes a formalization of meta-preferences. We prove that our formalism is robust in the sense that the operations for combining modules preserve the notion of a preference-based modular system. Finally, we formally demonstrate correspondences between our framework and the related preference formalisms of cp-nets and preference-based planning. Our framework allows one to use these preference formalisms (and others) in combination, in the same modular system.

1 Introduction

Combining knowledge bases (KBs) is very important when common sense reasoning is involved. For example, in planning, we may want to combine temporal and spatial reasoning, or reasoning from the point of view of several agents. Here, we focus on *search problems*, i.e., the problems where some input is given, and we are looking for a solution (e.g. a schedule, a trajectory, a business plan) according to a KB or a combination of KBs. Search problems are formalized as the task of *Model Expansion (MX)* [Mitchell and Ternovska, 2005], which is the task of expanding a given structure[1] (that represents an instance of the problem) with interpretations of new relations and functions (that represent solutions) to satisfy a specification in some logic, e.g. first-order logic, Answer Set Programming, etc. For example, consider the problem of constructing a trajectory of a falling ball. The input structure represents the initial conditions, and it is expanded with interpretation of a function (or a predicate) that represents spatial coordinates of the ball over a time interval, to satisfy an axiomatization of the trajectory. Another example is, given initial situation on the input, construct a plan of actions for an agent to satisfy a certain goal, by taking into account action preconditions and effects.

Modular Systems (MS) [Tasharrofi and Ternovska, 2011] is an extension to the MX framework. Each module (and a combination of them) is an MX task. Modules are combined through the operators of composition, union, projection, complementation and feedback. The framework is able to specify multi-component problems where their constituents are characterized in different languages. An algorithm for solving MSs was proposed in [Tasharrofi *et al.*, 2011]. An improvement of the algorithm, in the same paper, uses approximations to reduce the search space. Connections to Satisfiability Modulo Theory and other systems were discussed in [Tasharrofi *et al.*, 2011].

An important aspect of knowledge representation systems is the capability to represent preferences. The literature presents a variety of approaches to formalize preferences, e.g. [Brafman and Domshlak, 2009], [Santhanam *et al.*, 2011], [Delgrande *et al.*, 2003], [Brewka *et al.*, 2010], [Sohrabi *et al.*, 2008], [Boutilier *et al.*, 2004a], [Delgrande *et al.*, 2007], [Wilson, 2004], and [Faber *et al.*, 2013]. Several surveys have appeared in recent years categorizing preference formalisms from various perspectives. For example, in [Baier and McIlraith, 2008], a set of preference formalisms for planning have been introduced. The authors of [Delgrande *et al.*, 2004] classified preference frameworks in non-monotonic reasoning.

Preferences in database systems have been broadly investigated by different researchers such as [Kiessling, 2002], [Borzsony *et al.*, 2001] and [Stefanidis *et al.*, 2011]. A primary well-known preference language in database systems was proposed in [Kiessling, 2002]. In this language, some preference constructors were introduced to express basic preference terms. For example, POS is a constructor that given two n-arity tuples $A = (a_1, ..., a_n)$, $B = (b_1, ..., b_n)$ and a set called POS, A is preferred to B (notation $A >_P B$) with respect to i^{th} attribute (column) in the database table if and only if $A[i] \in POS$ and $B[i] \notin POS$. Logical connectives can be also applied on basic preferences. This language offers operators for combining preferences to construct complex preference terms. Pareto and prioritized accumulation are two operators broadly used in several frameworks. Prioritized accumulation (notation &) gives priority to a preference. Let A and B be tuples of the same relational schema R. A is preferred to B (notation $A >_{P_1 \& P_2} B$) if and only if

[1]A structure, e.g. $\mathcal{A} = (A, R_1^{\mathcal{A}}, ..., R_n^{\mathcal{A}}, f_1^{\mathcal{A}}, ..., f_m^{\mathcal{A}}, c_1^{\mathcal{A}}, ..., c_l^{\mathcal{A}})$ is a domain A together with an interpretation function $\mathcal{I}$ of relations (R) such that $R^{\mathcal{A}} = \mathcal{I}(R) \subseteq A^n$, function symbols (f) where $f^{\mathcal{A}} = \mathcal{I}(f) : A^m \to A$, and constants (c) where $c^{\mathcal{A}} = \mathcal{I}(c) \in A$.

$A >_{P_1} B \vee (A \not>_{P_1} B \wedge B \not>_{P_1} A \wedge A <_{P_2} B)$. Pareto operator combines two preferences such that A is preferred to to B with respect to composition of P_1 and P_2 ($P = P_1 \otimes P_2$) if and only if $(A >_{P_1} B \wedge \neg(B >_{P_2} A) \vee (A >_{P_2} B \wedge \neg B >_{P_1} A)$.

However, the main principles behind combinations of preference formalisms (possibly written in different languages) have not been fully formalized yet. Here are some key challenges of current formalisms: First, they cannot combine preferences in systems consisting of intricate interconnected parts (see feedback connection in Example 1 below). Second, preference terms are written in the same language such as [Ross, 2007] or [Kiessling, 2002] that use first-order logic to express preferences. They are not able to formalize preference composition in heterogeneous data systems. The following example clarifies the complexity of formalizing a modular system with preferences.

Example 1. *A Logistic Service Provider (LSP) is a modular system that can be used by a company such as Oracle. It decides how to pack goods and deliver them. It solves two NP-complete tasks interactively, – Multiple Knapsack (module M_K) and Travelling Salesman Problem (module M_{TSP}). The system takes orders from customers (items $Items(i)$ to deliver, their profits $p(i)$, weights $w(i)$), and the capasity of trucks available $c(t)$, decides how to pack $Pack(i,t)$ items in trucks, and for each truck, solves a TSP problem. The feedback about solvability of TSP is sent back to M_K. Module M_{TSP} takes a candidate solution from M_K, together with the graph of cities and routes with distances, allowable distance limit and destinations for each product. The output of this module is the route, for each truck $Route(t,n,c)$, where t is a truck, n is the number in the sequence, and c is a city. The Knapsack problem is written, in, e.g. Integer Linear Programming (ILP), and TSP in Answer Set Programming (ASP). The modules M_K and M_{TSP} are composed in sequence, with a feedback going from an output of M_{TSP} to an input of M_K. A solution to the compound module, M_{LSP}, to be acceptable, must satisfy both sub-systems. The company may have preferences for packing and delivery of products. E.g. if a fragile item is packed in a truck, it may be preferred to exclude heavy items. Among certain routes with equal costs, some of them may be preferred to others. It is possible that preferences in the Knapsack problem are formalized by cp-nets [Boutilier* et al.*, 2004a] and the TSP's preferences are represented in preference-based Answer Set Programming framework [Brewka* et al.*, 2003]. In Figure 1, $\mathcal{P}_k$ denotes the preferences of the knapsack module and $\mathcal{P}_{TSP}$ denotes the TSP module's preferences. Formalizing this modular problem with preferences is not easy because: 1) the Knapsack and the TSP are axiomatized in different languages, 2) preferences of each module are represented by a different formalism, 3) preference formalisms use different languages than the axiomatizations of the modules themselves, 4) two modules communicate in a complex way that includes a feedback loop from M_{TSP} to M_K.*

Contributions

We propose model-theoretic foundations for combining KBs with preferences in modular systems. On the logic level, each

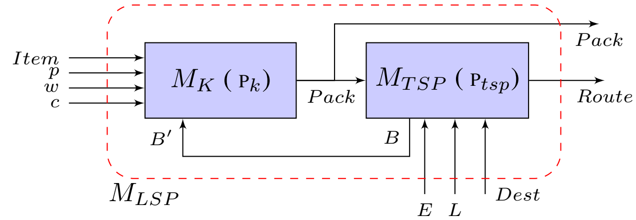

Figure 1: Logistics Service Provider $(M_K \rhd M_{TSP})[B = B']$.

module is represented by a KB in some logic $\mathcal{L}^2$, and its preferences (and meta-preferences) are represented by (strict) partial orders on partial structures in a preference formalism named $\mathcal{P}$-$\mathcal{MS}$. Different logics and preference formalisms can be used for modules in the same system. Operations for combining modules are generalized to combine preferences of each module. We prove that our formalism is *robust* in the sense that the operations for combining modules preserve the notion of a preference-based modular system. Our formalism is consistent with (and extends) the model-theoretic semantics of modular systems [Tasharrofi and Ternovska, 2011]. In model theoretic semantics, an MX task is viewed as a set of structures, where each input is expanded with solutions. We also prove that our formalism represent cp-nets and preference-based planning. Thus we can combine them in one modular system.

Novelty

With our formalism, *each module can be formalized in a different framework*. To our knowledge, this is the first multi-language preference formalism. This generality is achieved through the model-theoretic semantics of modular systems. Another novelty is the ability of handling preferences in complexly structured systems. For instance, in Example 1, there is a complex combination of Knapsack and TSP problems (feedback from TSP to Knapsack). In contrast, these complex systems were not representable in previous work. E.g., in [Kiessling, 2002].

2 Preliminaries

A *vocabulary* (denoted, e.g. $\tau, \sigma, \varepsilon$) is a set of non-logical (predicate and function) symbols. A τ-structure is a domain (a set), and interpretation of vocabulary symbols in τ.

In [Tasharrofi and Ternovska, 2011], the authors formalize combinatorial search problems as the task of *model expansion (MX)*, the logical task of expanding a given structure with new relations. Formally, the user axiomatizes the problem in some logic $\mathcal{L}$. This axiomatization relates an instance of the problem (a *finite structure*), and its solutions (*expansions* of that structure with new relations or functions). Logic $\mathcal{L}$ corresponds to a specification/modelling language. It could be an ASP program, or a specification in a language from the CP community, or even a Java program, as long as model-theoretic semantics can be provided.

[2]Any logic with model-theoretic semantics can be used, including logic programs.

Definition 1 (Model Expansion task). *Given: a formula ϕ in logic $\mathcal{L}$ with a vocabulary $\sigma \cup \varepsilon$, such that $\sigma \cap \varepsilon = \emptyset$ and σ-structure $\mathcal{A}$. Find: structure $\mathcal{B}$ that expands $\mathcal{A}$ to $\sigma \cup \varepsilon$ and satisfies ϕ. We call σ* instance *and ε* expansion *vocabularies.*

Definition 2 (Module). *A* module *is a set (class) of $\sigma \cup \varepsilon$-structures, where $\sigma \cap \varepsilon = \emptyset$.*

A module can be given by any decision procedure, be a set of models of a KB, be given by an inductive definition, a C or an ASP program, or by an agent making decisions. Modules of [Tasharrofi *et al.*, 2011] were introduced as model expansion tasks. The view of Definition 2 is equivalent. In modular systems, information propagation happens through vocabulary symbols that are equal. Modules are combined using the following algebraic operations. Projection ($\pi_\nu(M)$) hides some vocabulary of a module. Composition ($M_1 \rhd M_2$) connects outputs of M_1 to inputs of M_2. Union ($M_1 \cup M_2$) models choice. Complementation ($\overline{M}$) does "the opposite" of what M does. Feedback ($M[R{=}S]$) connects output S of M to its input R and was inspired by feedbacks in logical circuits. Intuitively, the operations correspond to conjunction, disjunction, negation and existential quantifier. Feedback represents fixpoints (not necessarily minimal) of modules viewed as operators. One can introduce other operations, e.g. least fixpoints or combinations of the operations above.

We now define the syntax of the algebra of modular systems. Following [Järvisalo *et al.*, 2009], we say modules M_1 and M_2 are *composable* if $\varepsilon_{M_1} \cap \varepsilon_{M_2} = \emptyset$ (no output interference). Module M_2 is *independent* from M_1 if $\sigma_{M_2} \cap \varepsilon_{M_1} = \emptyset$ (no cyclic dependencies). A module is *primitive* if the only sub-module (algebraic sub-formula) of it is itself. *Well-formed modular systems* $\mathrm{MS}(\sigma, \varepsilon)$, with instance ($\sigma$) and expansion ($\varepsilon$) vocabularies, are defined recursively.

- If M is a primitive module with instance (input) vocabulary σ and expansion (output) vocabulary ε, then $M \in \mathrm{MS}(\sigma, \varepsilon)$.
- If $M \in \mathrm{MS}(\sigma, \varepsilon)$, $\tau \subseteq \sigma \cup \varepsilon$, then $\pi_\tau(M) \in \mathrm{MS}(\sigma \cap \tau, \varepsilon \cap \tau)$.
- If $M \in \mathrm{MS}(\sigma, \varepsilon)$, $M' \in \mathrm{MS}(\sigma', \varepsilon')$, M is composable with and independent from M', then $(M \rhd M') \in \mathrm{MS}(\sigma \cup (\sigma' \setminus \varepsilon), \varepsilon \cup \varepsilon')$.
- If $M \in \mathrm{MS}(\sigma, \varepsilon)$, $M' \in \mathrm{MS}(\sigma', \varepsilon')$, and they are independent, then $(M \cup M') \in \mathrm{MS}(\sigma \cup \sigma', \varepsilon \cup \varepsilon')$.
- If $M \in \mathrm{MS}(\sigma, \varepsilon)$, $R \in \sigma$, $S \in \varepsilon$, and R and S are of the same type and arity, then $M[R{=}S] \in \mathrm{MS}(\sigma \setminus \{R\}, \varepsilon \cup \{R\})$.
- If $M \in \mathrm{MS}(\sigma, \varepsilon)$ then $\overline{M} \in \mathrm{MS}(\sigma, \varepsilon)$.

A modular system is given by an algebraic formula, with input-output vocabulary specified for each primitive module. *Subsystems* correspond to sub-formulas and are modules themselves.

Model-theoretic semantics associates, with each modular system, a set of structures. Each such structure is called a *model* of the modular system. The semantics does not put any finiteness restriction on the domains. Thus, the framework works for modules with infinite structures. We assume that the domains of all structures are included in a (potentially infinite) universal domain U.

Definition 3 (Models of a Modular System). *Let $M \in \mathrm{MS}(\sigma, \varepsilon)$ be a modular system and $\mathcal{B}$ be a $(\sigma \cup \varepsilon)$-structure. We construct the set M^{mt} of* models of module *M by structural induction on the structure of a module.*

Primitive Module: *$\mathcal{B}$ is a model of M if $\mathcal{B} \in M$.*

Projection: *$\mathcal{B}$ is a model of $M := \pi_{(\sigma \cup \varepsilon)}(M')$ (with $M' \in \mathrm{MS}(\sigma', \varepsilon')$) if a $(\sigma' \cup \varepsilon')$-structure $\mathcal{B}'$ exists such that $\mathcal{B}'$ is a model of M' and $\mathcal{B}'$ expands $\mathcal{B}$.*

Composition: *$\mathcal{B}$ is a model of $M := M_1 \rhd M_2$ (with $M_1 \in \mathrm{MS}(\sigma_1, \varepsilon_1)$ and $M_2 \in \mathrm{MS}(\sigma_2, \varepsilon_2)$) if $\mathcal{B}|_{(\sigma_1 \cup \varepsilon_1)}$ is a model of M_1 and $\mathcal{B}|_{(\sigma_2 \cup \varepsilon_2)}$ is a model of M_2.*

Feedback: *$\mathcal{B}$ is a model of $M := M'[R{=}S]$ (with $M' \in \mathrm{MS}(\sigma', \varepsilon')$) if $R^{\mathcal{B}} = S^{\mathcal{B}}$ and $\mathcal{B}$ is model of M'.*

To save space, we skip union and complementation.

E.g. the Knapsack-TSP system in Example 1 is formalized as $M_{LSP} = [M_K \rhd M_{TSP}]_{B=B'}$.

Partial structures allow interpretation of some vocabulary symbols to be partially specified. The algorithm for solving modular systems [Tasharrofi *et al.*, 2011] constructs expansions incrementally, by adding information to partial structures.

Definition 4. *$\mathcal{B}$ is a τ_p-partial structure over vocabulary τ if: (1) $\tau_p \subseteq \tau$, (2) $\mathcal{B}$ gives a total interpretation to symbols in $\tau \setminus \tau_p$, and (3) for each $n-ary$ symbol R in τ_p, $\mathcal{B}$ interprets R using two sets R^+ and R^- such that $R^+ \cap R^- = \emptyset$, and $R^+ \cup R^- \neq [dom(\mathcal{B})]^n$.*

Definition 5. *For two partial structures $\mathcal{B}$ and $\mathcal{B}'$ over the same vocabulary and domain, we say that $\mathcal{B}$* extends *$\mathcal{B}'$ if all undefined symbols in $\mathcal{B}$ are also undefined in $\mathcal{B}'$.*

Notation 1. *Let $V = \{a_1, a_2, ..., a_n\}$ be a set of vocabulary symbols. Let $\mathcal{A}$ be a partial structure that interprets a subset $X \subseteq V$ such that V-X is undefined. Each $a_i \in X$ can be interpreted as false, represented by a_i^-, or as true, represented by a_i or a_i^+. Suppose Y is a set of the form $\{a_i^{\mathcal{A}} | a_i \in X\}$ where $a_i^{\mathcal{A}} = a_i^+$ or a_i^- is interpretation of a_i by $\mathcal{A}$. We assume that set Y is representation of partial structure $\mathcal{A}$.*

3 $\mathcal{P}$-$\mathcal{MS}$: Preference-based Modular Systems

In this section we introduce Preference-based Modular Systems ($\mathcal{P}$-$\mathcal{MS}$). To have a formalism compatible with model theoretic semantics of modular systems, we define preference statements based on the concept of structures. However, using structures to model preferences is not always practical. Formally speaking, some interpreted symbols may be preferred to others, and there could not be enough information to decide about the rest. Unlike structures, partial structures interpret a subset of vocabulary symbols, while interpretation of other symbols is unknown. The idea of partial structures originates from the notion of three-valued logic that a truth value of a statement can be true, false, or unknown [Kleene, 1952]. In our formalism, a preference statement can be represented by a partial order over a set of partial structures when certain conditions hold. First, we explain the meaning of strict partial order.

Definition 6. *A strict partial order $\mathcal{O}$ over a set $\mathcal{S}$ is a pair $\mathcal{O} := (\mathcal{S}, \prec)$ such that $\prec$ is a binary relation over elements of $\mathcal{S}$ that is anti-reflexive, asymmetric and transitive.*

Now, we define one preference statement for a primitive module (single model expansion task).

Definition 7. *Let M be a primitive module and $vocab(M)=\tau$. A τ_o-preference (or simply called preference) $\mathcal{P}=(\mathcal{O},\Gamma)$ in M is a pair where $\mathcal{O}=(\mathcal{S},\prec)$ is a strict partial order over $\mathcal{S}$ that is the set of all τ_o-partial structures in M where $\tau_o\subseteq\tau$. As well, $\Gamma=\{\mathcal{C}_1,\mathcal{C}_2,...,\mathcal{C}_m\}$ is a set of τ_{p_i}-partial structures, $1\leq i\leq m$, in M that $\tau_{p_i}\subseteq\tau$.*

In practical domains, preferences are usually represented by conditional statements. In the above definition, we utilize a set of partial structures Γ to express the premises of a preference statement, and $\mathcal{O}$ represents the conclusion. Once the preference has been defined, a preferred structure is introduced as follows:

Definition 8. *Let M be a primitive module, and $\mathcal{B}$, $\mathcal{B}'$ be two structures in M. Given preference $\mathcal{P}=(\mathcal{O},\Gamma)$ in M, let Δ be a set of all structures in M that extend at least one member of Γ. We say structure $\mathcal{B}$ is* preferred *to $\mathcal{B}'$ with respect to $\mathcal{P}$ (denoted by $\mathcal{B}\succ_{\mathcal{P}}\mathcal{B}'$) if 1) $\mathcal{B},\mathcal{B}'\in\Delta$, 2) there are partial structures $\mathcal{B}_i$ and $\mathcal{B}_j$ over $vocab(M)$ that can be extended to structures in M such that $\mathcal{B}_i\succ\mathcal{B}_j$, and $\mathcal{B}$ is an extension of $\mathcal{B}_i$, where $\mathcal{B}'$ extends $\mathcal{B}_j$, and 3) there are no partial structures $\mathcal{B}_k$ and $\mathcal{B}_m$ such that $\mathcal{B}$ and $\mathcal{B}'$ extend them respectively and $\mathcal{B}_m\succ\mathcal{B}_k$.*

This definition states that when a part of $\mathcal{B}$ is preferred to $\mathcal{B}'$, if a condition specified by Γ is satisfied by both structures, we can conclude that $\mathcal{B}$ is preferred to $\mathcal{B}'$. It makes no difference how the rest of the vocabulary is interpreted because it is irrelevant to $\mathcal{P}$.

Example 2. *In Example 1, consider that safety of delivering items is an important preference for the company. So, it is preferred to avoid packing heavy and light items together to reduce the risk of damage to the light items. Let $\mathcal{P}_{safe}=(\mathcal{O}_{safe},\Gamma_{safe})$ be the safety preference where $\mathcal{O}_{safe}=(\mathcal{S},\prec)$ is a partial order over $\mathcal{S}$ that is the set of all items. Relation "$\prec$" is defined as $\{pack^-(i)\prec pack(i)|w(i)\leq W\}$; it means that for each item i that is lighter than a constant weight W, it is preferred to not put i in the pack. According to Notation 1, $pack^-(i)$ is representation of a partial structure that interprets ground atom $pack(i)$ as false. The premise of the conditional statement is formalized by $\Gamma_{safe}=\{\mathcal{C}_1,\mathcal{C}_2,...,\mathcal{C}_m\}$ where $\mathcal{C}_i=\{item(i),w(i)\}$ such that $w(i)\geq W'$. This states when there is an item with weight not less than W', it is preferred to not include items lighter than W in the pack.*

Definition 9. *For two structures $\mathcal{B},\mathcal{B}'\in M$, if a) neither $\mathcal{B}\succ_{\mathcal{P}}\mathcal{B}'$ nor $\mathcal{B}'\succ_{\mathcal{P}}\mathcal{B}$, b) for any $\mathcal{B}''\in M$, if $\mathcal{B}''\succ_{\mathcal{P}}\mathcal{B}$ then $\mathcal{B}''\succ_{\mathcal{P}}\mathcal{B}'$, and c) if $\mathcal{B}\succ_{\mathcal{P}}\mathcal{B}''$ then $\mathcal{B}'\succ_{\mathcal{P}}\mathcal{B}''$, they are called equally preferred with respect to $\mathcal{P}$ and are represented by $\mathcal{B}\approx_{\mathcal{P}}\mathcal{B}'$. If one of the conditions (b) or (c) do not hold, then, $\mathcal{B}$ and $\mathcal{B}'$ are incomparable and are represented by $\mathcal{B}\sim_{\mathcal{P}}\mathcal{B}'$. Also, $\mathcal{B}\succeq_{\mathcal{P}}\mathcal{B}'$ means that $\mathcal{B}\succ_{\mathcal{P}}\mathcal{B}'$ or $\mathcal{B}\approx_{\mathcal{P}}\mathcal{B}'$.*

Considering that a module is defined as a set of structures, we can conclude the following.

Proposition 1. *Given a preference $\mathcal{P}=(\mathcal{O},\Gamma)$ in module M, the pair $(M,\prec_{\mathcal{P}})$ is a strict partial order, $\approx_{\mathcal{P}}$ is an equivalence relation over structures of M, and $\succeq_{\mathcal{P}}$ is a transitive and reflexive binary relation over structures of M.*

Meta-Preferences

In practice, each module may have more than one preference. Some of them may be preferred to others. The question then arises how a preferred structure is defined in this case. The notion of meta-preference addresses this question.

Definition 10. *Given a module M and a set of preferences $\Pi=\{\mathcal{P}_1,\mathcal{P}_2,...,\mathcal{P}_n\}$, let $\Omega_i:=\{\mathcal{P}_j\in\Pi|(\mathcal{P}_i\succ\mathcal{P}_j)\ \vee\ (\mathcal{P}_j\succ\mathcal{P}_i)\}$ be a subset of Π such that its elements have order relation with $\mathcal{P}_i$. Assume $\mathcal{O}_{MP}=(\Pi,\prec)$ is a strict partial order over elements of Π. Binary relation $\succ_{\mathcal{MP}^*}$ over structures of M is defined as:*

$\mathcal{B}\succ_{\mathcal{MP}^*}\mathcal{B}'$ if there is a preference $\mathcal{P}_i\in\Pi$ such that $\mathcal{B}\succ_{\mathcal{P}_i}\mathcal{B}'$ and

- *there does not exist $\mathcal{P}_j\in\Omega_i$ that $\mathcal{P}_j\succ\mathcal{P}_i$ with respect to $\mathcal{O}_{MP}$ and $\mathcal{B}'\succ_{\mathcal{P}_j}\mathcal{B}$, and*
- *there is no a preference $\mathcal{P}_k\in\Pi\backslash\Omega_i$ that $\mathcal{B}'\succ_{\mathcal{P}_k}\mathcal{B}$.*

Meta-preference $\mathcal{MP}$ is characterized as $\mathcal{MP}:=\mathcal{O}_{MP}$. We say structure $\mathcal{B}$ is preferred *to $\mathcal{B}'$ with respect to binary relation $\succ_{\mathcal{MP}}\subseteq M\times M$ (notation $\mathcal{B}\succ_{\mathcal{MP}}\mathcal{B}'$) whenever if $\exists\mathcal{B}''\in M;\mathcal{B}'\succ_{\mathcal{MP}^*}\mathcal{B}''$, then $\mathcal{B}''\not\succ_{\mathcal{MP}^*}\mathcal{B}$.*

This definition states that structure $\mathcal{B}$ is preferred to $\mathcal{B}'$ with respect to $\mathcal{MP}$ if we can find a preference such as $\mathcal{P}_i$ that $\mathcal{B}\succ_{\mathcal{P}_i}\mathcal{B}'$ and there is no preference that makes $\mathcal{B}'$ preferred to $\mathcal{B}$. If there is a preference $\mathcal{P}_j$ such that $\mathcal{B}'\succ_{\mathcal{P}_j}\mathcal{B}$ then $\mathcal{B}$ is not preferred to $\mathcal{B}'$ with respect to the meta-preference unless $\mathcal{P}_i$ is preferred to $\mathcal{P}_j$. Also, the definition prevents conflicts may happen between a mix of preferences, though it does not guarantee transitivity of $\succ_{\mathcal{MP}}$. If $\mathcal{B}$ is preferred to $\mathcal{B}'$ with respect to $\mathcal{MP}$, and if $\mathcal{B}'$ is preferred to $\mathcal{B}''$, then $\mathcal{B}''$ cannot be preferred to $\mathcal{B}$ with respect to $\mathcal{MP}$.

Example 3. *In Example 1, assume that the company has more than one preference. If an expensive item is selected for delivery, it is not secure to have another precious item in the pack that is specified by $\mathcal{P}_{security}$. Assume we have a meta-preference $\mathcal{MP}$ such that $\Pi_K=\{\mathcal{P}_{safe},\mathcal{P}_{security}\}$ and $\mathcal{MP}=\{\mathcal{P}_{safe}\prec\mathcal{P}_{security}\}$. To have a preferred packing for the Knapsack module, when there is a heavy and expensive item in the pack, it is preferred to not include another heavy item, but it is fine to have two expensive items in the pack.*

Preference-based Modular Systems

Up to now, we defined a preference $\mathcal{P}$ in a single primitive module. In what follows, we study how a preference in a modular system is modelled when preferences of its components are given.

Definition 11. *Let $M=M_1\rhd M_2$ be a modular system, $vocab(M_1)=\tau_1$, and $vocab(M_2)=\tau_2$. Given $\mathcal{P}_1=(\mathcal{O}_1,\Gamma_1)$ in M_1 and $\mathcal{P}_2=(\mathcal{O}_2,\Gamma_2)$ in M_2, for $\mathcal{B},\mathcal{B}'\in M$, $\mathcal{B}$ is preferred to $\mathcal{B}'$ with respect to $\mathcal{P}_1$ and $\mathcal{P}_2$, and is represented by $\mathcal{B}\succ_{\mathcal{P}_1\rhd\mathcal{P}_2}\mathcal{B}'$ when $\mathcal{B}|_{\tau_1}\succ_{\mathcal{P}_1}\mathcal{B}'|_{\tau_1}$ and $\mathcal{B}|_{\tau_2}\succ_{\mathcal{P}_2}\mathcal{B}'|_{\tau_2}$, where $\mathcal{B}|_{\tau_i}$ is restriction of $\mathcal{B}$ to τ_i.*

Informally, $\mathcal{B}$ is preferred to $\mathcal{B}'$ with respect to $\mathcal{P}=\mathcal{P}_1\rhd\mathcal{P}_2$, if $\mathcal{B}$ is preferred to $\mathcal{B}'$ with respect to $\mathcal{P}_1$ when they are restricted to the vocabulary of M_1 and with respect to $\mathcal{P}_2$ when they are restricted to the vocabulary of M_2.

Example 4. *In Example 1, for module M_{tsp}, suppose that if cities c_1,c_2,c_3,c_4 are in the set of destinations, there is* *a path from c_1 to c_4 through c_2 that is preferred to the path from c_1 to c_4 through c_3. This can be formalized by preference $\mathcal{P}_{tsp}=(\mathcal{O}_{tsp},\Gamma_{tsp})$ where $\mathcal{O}_{tsp}=(\mathcal{S}_{tsp},\prec)$ is a partial order over $\mathcal{S}_{tsp}$ that is the set of all possible routes. For a positive integer k and truck t,*

$\{Route(k,c_1,t), Route(k+1,c_2,t), Route(k+2,c_4,t)\succ Route(k,c_1,t), Route(k+1,c_3,t), Route(k+2,c_4,t)\}$ *and* $\Gamma_{tsp}=\{Dest(1,c_1), Dest(2,c_2), Dest(3,c_3), Dest(4,c_4)\}$.

A preferred plan of packing and delivery with respect to $\mathcal{P}_{safe}\rhd\mathcal{P}_{tsp}$ is the one where heavy and light items are not in the same pack and if the truck is supposed to visit cities c_1,c_2,c_3,c_4, then taking road (c_1,c_2) is preferred to (c_1,c_3).

We now extend this notion to meta-preferences.

Definition 12. *Let $M=M_1\rhd M_2$ and let's assume that $vocab(M_1)=\tau_1$ and $vocab(M_2)=\tau_2$. Assume that Π_1 is a set of preferences in M_1 and $\mathcal{MP}_1$ is a meta-preference over Π_1. Similarly, Π_2 and $\mathcal{MP}_2$ are a set of preferences and a meta-preference respectively in M_2. For $\mathcal{B},\mathcal{B}'\in M$, $\mathcal{B}$ is preferred to $\mathcal{B}'$ with respect to $\mathcal{MP}_1$ and $\mathcal{MP}_2$, and is represented by $\mathcal{B}\succ_{\mathcal{MP}_1\rhd\mathcal{MP}_2}\mathcal{B}'$ when $\mathcal{B}|_{\tau_1}\succ_{\mathcal{MP}_1}\mathcal{B}'|_{\tau_1}$ and $\mathcal{B}|_{\tau_2}\succ_{\mathcal{MP}_2}\mathcal{B}'|_{\tau_2}$.*

We proceed to the union operator.

Definition 13. *Let $M=M_1\cup M_2$ be a modular system. Suppose $vocab(M_1)=\tau_1$ and $vocab(M_2)=\tau_2$. Assume $\mathcal{P}_1$ and $\mathcal{P}_2$ are preferences in M_1 and M_2 respectively. For $\mathcal{B},\mathcal{B}'\in M$, if $\mathcal{B}|_{\tau_1}\succ_{\mathcal{P}_1}\mathcal{B}'|_{\tau_1}$ or $\mathcal{B}|_{\tau_2}\succ_{\mathcal{P}_2}\mathcal{B}'|_{\tau_2}$ then $\mathcal{B}$ is preferred to $\mathcal{B}'$ with respect to $\mathcal{P}_1\cup\mathcal{P}_2$ and is denoted by $\mathcal{B}\succ_{\mathcal{P}_1\cup\mathcal{P}_2}\mathcal{B}'$.*

For meta-preferences we have:

Definition 14. *Let $M=M_1\cup M_2$ be a modular system. Suppose $vocab(M_1)=\tau_1$ and $vocab(M_2)=\tau_2$. Let Π_1 be a set of preferences in M_1 and $\mathcal{MP}_1$ is a meta-preference over Π_1, and let Π_2 be a set of preferences in M_2 and $\mathcal{MP}_2$ a meta-preference over Π_2. For $\mathcal{B},\mathcal{B}'\in M$, $\mathcal{B}$ is preferred to $\mathcal{B}'$ with respect to $\mathcal{MP}_1$ and $\mathcal{MP}_2$ (notation $\mathcal{B}\succ_{\mathcal{MP}_1\cup\mathcal{MP}_2}\mathcal{B}'$) when $\mathcal{B}|_{\tau_1}\succ_{\mathcal{MP}_1}\mathcal{B}'|_{\tau_1}$ or $\mathcal{B}|_{\tau_2}\succ_{\mathcal{MP}_2}\mathcal{B}'|_{\tau_2}$.*

Let us comment briefly on the feedback operator. Let M be a $\sigma\cup\varepsilon$ modular system, $R\in\sigma$, and $S\in\varepsilon$. If R and S are two vocabulary symbols of the same type and arity, then $M[R=S]$ is a $(\sigma\setminus\{R\})\cup(\varepsilon\cup\{R\})$ modular system. The feedback operator does not change the vocabulary of a module. Hence, definition of a preference remains unchanged. When $\mathcal{B}\succ_{\mathcal{P}}\mathcal{B}'$ holds in M, if $\mathcal{B}$ and $\mathcal{B}'$ are also structures of M', we conclude that $\mathcal{B}$ is preferred to $\mathcal{B}'$ in M'.

Definition 15. *Let's assume $M'=M[R=S]$ and $\mathcal{P}=(\mathcal{O},\Gamma)$ are preferences in M. For $\mathcal{B},\mathcal{B}'\in M$, whenever $R^{\mathcal{B}}=S^{\mathcal{B}}$ and $R^{\mathcal{B}'}=S^{\mathcal{B}'}$, if $\mathcal{B}\succ_{\mathcal{P}}\mathcal{B}'$, then for $\mathcal{B}_f,\mathcal{B}'_f\in M'$, $\mathcal{B}_f\succ_{\mathcal{P}}\mathcal{B}'_f$ holds where $\mathcal{B}_f=\mathcal{B}$ and $\mathcal{B}'_f=\mathcal{B}'$.*

This definition says that if two structures $\mathcal{B}$ and $\mathcal{B}'$ are in M, and $\mathcal{B}$ is preferred to $\mathcal{B}'$ with respect to $\mathcal{P}$ then $\mathcal{B}$ remains preferred to $\mathcal{B}'$ in module M' that is module M with feedback.

The following definition introduces meta-preferences in a module with feedback operator.

Definition 16. *Assume $M'=M[R=S]$, Π is a set of preferences in M, and $\mathcal{MP}$ is a meta-preference over Π. Assume that $\mathcal{B},\mathcal{B}'\in M$, and $\mathcal{B},\mathcal{B}'\in M'$. If $\mathcal{B}\succ_{\mathcal{MP}}\mathcal{B}'$ in M, then $\mathcal{B}\succ_{\mathcal{MP}}\mathcal{B}'$ in M'.*

In the model expansion task, in a general sense, there are vocabulary symbols (notation ε_h) that are hidden from outer observers while they are interpreted by the structures of the module. By considering the fact that projection operator hides some visible vocabulary symbols of the module, we present the following definition.

Definition 17. *Let us assume $M'=\pi_{\tau'}(M)$, where $vocab(M)=\tau$ and $vocab(M')=\tau'$ (τ and τ' are visible vocabularies). For $\mathcal{B}_\pi,\mathcal{B}'_\pi\in M'$, if there are structures $\mathcal{B}$ and $\mathcal{B}'$ in M such that $\mathcal{B}|_{\tau'}=\mathcal{B}_\pi$ and $\mathcal{B}'|_{\tau'}=\mathcal{B}'_\pi$ and $\mathcal{B}\succ_{\mathcal{P}}\mathcal{B}'$, then we say $\mathcal{B}_\pi\succ_{\mathcal{P}}\mathcal{B}'_\pi$ on the condition that for all vocabulary symbols $\mathcal{R}\in(\tau\setminus\tau')\subseteq\varepsilon'_h$ the following holds: $\mathcal{R}^{\mathcal{B}}=\mathcal{R}^{\mathcal{B}}_\pi$ and $\mathcal{R}^{\mathcal{B}'}=\mathcal{R}^{\mathcal{B}'}_\pi$.*

Intuitively, given two structures $\mathcal{B}_\pi$ and $\mathcal{B}'_\pi$ in M', if we can find two structures $\mathcal{B}$ and $\mathcal{B}'$ in M such that they expand $\mathcal{B}_\pi$ and $\mathcal{B}'_\pi$, if $\mathcal{B}$ is preferred to $\mathcal{B}'$ with respect to $\mathcal{MP}$, we can conclude that $\mathcal{B}_\pi$ is also preferred to $\mathcal{B}'_\pi$. We proceed to meta-preferences.

Definition 18. *Let $M'=\pi_{\tau'}(M)$, $vocab(M)=\tau$, and $vocab(M')=\tau'$ (τ and τ' are visible vocabularies). Assume $\mathcal{B}_\pi,\mathcal{B}'_\pi\in M'$, if there are structures $\mathcal{B}$ and $\mathcal{B}'$in M such that $\mathcal{B}|_{\tau'}=\mathcal{B}_\pi$ and $\mathcal{B}'|_{\tau'}=\mathcal{B}'_\pi$ and $\mathcal{B}\succ_{\mathcal{MP}}\mathcal{B}'$, then we say $\mathcal{B}_\pi\succ_{\mathcal{MP}}\mathcal{B}'_\pi$ on the condition that for all vocabulary symbols $\mathcal{R}\in(\tau\setminus\tau')\subseteq\varepsilon'_h$ the following holds: $\mathcal{R}^{\mathcal{B}}=\mathcal{R}^{\mathcal{B}_\pi}$ and $\mathcal{R}^{\mathcal{B}'}=\mathcal{R}^{\mathcal{B}'_\pi}$.*

A preferred modular system $\mathcal{P}$-$\mathcal{MS}$ is a modular system with a partial order over its preferences.

Definition 19. *A modular system MS with a set of preferences Π is a* preferred modular system*, notation $\mathcal{P}$-$\mathcal{MS}$, if it is specified by a pair $(\prec_{\mathcal{MP}},MS)$ where $\mathcal{MP}$ is a meta-preference in MS.*

The following statement shows the robustness of our notions and is proven by structural induction.

Theorem 1. *Assume for some n, a modular system MS is obtained from $M_1,M_2,...,M_n$, where M_is, $1\le i\le n$ are modular systems, by using operations in modular systems including composition, union, feedback, and projection. For all $1\le i\le n$, if M_i is $\mathcal{P}$-$\mathcal{MS}$ then M is also $\mathcal{P}$-$\mathcal{MS}$.*

4 Relation with Two Preference Formalisms

We now describe two preference formalisms and show how they can be related to our formalism.

CP-Nets

Ceteris pairbus (cp) network is a graphical representation of conditional preferences with reasoning capability [Boutilier *et al.*, 2004b]. The idea underlying cp-nets is to compare different assignments to a set of variables as some of these variables are conditionally dependent on each other. Each node represents an attribute (variable) connected to its parents through directed edges. A preference over domain values of a variable is dependent on all of its parents value. The dependency is shown by a *Conditional Preference Table* (CPT) represented as an annotation for each node. There exists an induced graph derived from each cp-net that shows ordering relation between a subset of outcomes. Each node in the induced graph represents an outcome and each directed edge exhibits ordering relation between nodes. An outcome o_1 is preferred to o_2 if in the induced graph, there is a path from o_1 to o_2. An induced graph comprises all information about preferences over outcomes that can be derived from a cp-net.

From the syntactic point of view, $\mathcal{P}$-$\mathcal{MS}$ is able to capture the notion of attributes in cp-nets. Each attribute can be viewed as an interpreted predicate symbol in the context of $\mathcal{P}$-$\mathcal{MS}$. Therefore, an outcome in a cp-net can be represented by a structure that interprets vocabulary symbols. The relation between cp-nets and $\mathcal{P}$-$\mathcal{MS}$ in this way implies that the space of all outcomes in a cp-net can be modelled by a set of structures interpreting vocabulary symbols in $\mathcal{P}$-$\mathcal{MS}$. A preference statement visualized by a cp-net over a set of variables $V=\{V_1, ..., V_n\}$ is an ordering over domain values of a variable that may or may not be dependent on some other variables, and a preference in $\mathcal{P}$-$\mathcal{MS}$ is defined as $\mathcal{P}=(\mathcal{O}, \Gamma)$ where $\mathcal{O}$ is a partial order given a set of partial structures Γ. In a sense, a partial structure in $\mathcal{P}$-$\mathcal{MS}$ is a combination of some interpreted vocabulary symbols. Thus, a partial structure can stipulate a value assigned to an attribute. Orderings over partial structures in our formalism are in fact orderings over attribute values in cp-nets when partial structures in $\mathcal{O}$ are assumed to interpret only one vocabulary symbol. Transforming the condition part of the preference statement in a cp-net is straightforward. Order relation holds for partial structures which extend Γ. Therefore, parents of each cp-net attribute can be represented by Γ.

In order to establish the correspondence between the semantic of cp-nets and $\mathcal{P}$-$\mathcal{MS}$, first we explain the concept of *flip-over* in cp-nets. In an induced graph derived from a cp-net, each outcome node has one attribute value preferred to its child's while other attributes are assumed to be fixed. Therefore, by moving from a node to its children one attribute value is changed that is called a *flip-over*. A path in an induced graph is a chain of *flip-overs* between two outcomes. Hence, an outcome is preferred to another when single or multiple *flip-over(s)* exist between them. Now, we show how a *flip-over* can be represented in $\mathcal{P}$-$\mathcal{MS}$. Consider two structures $\mathcal{B}$ and $\mathcal{B}'$; if $\mathcal{B} \succeq_{\mathcal{MP}} \mathcal{B}'$ ($\succeq_{\mathcal{MP}}$ means that $\succ_{\mathcal{MP}}$ or $\approx_{\mathcal{MP}}$ that is an equivalence relation), we have enough information to know that $\mathcal{B}$ is preferred to $\mathcal{B}'$ at least at one vocabulary symbol interpretation or they are equally preferred. The concept of a single *flip-over* can be specified by $\succeq_{\mathcal{MP}}$ when $\mathcal{O}_{MP}=\emptyset$ (there is no meta-preference in cp-nets). In this case, $\succeq_{\mathcal{MP}}$ has the transitivity property and a chain of *flip-overs* can be modelled by $\mathcal{P}$-$\mathcal{MS}$ as well. If $\mathcal{O}_{MP}$ is not empty, $\mathcal{MP}$ represents the notion of *relative importance* (meta-preference) in TCP-net [Brafman *et al.*, 2006] that is an extension of cp-nets to model meta-preferences. This reasoning leads us to the following theorem, relating cp-nets and the $\mathcal{P}$-$\mathcal{MS}$ formalism.

Theorem 2. *Let $\mathcal{G}$ be a cp-net and $\mathcal{MP}$ be the representation of $\mathcal{G}$ in the context of $\mathcal{P}$-$\mathcal{MS}$. If an outcome o_1 is preferred to outcome o_2 in the induced graph of $\mathcal{G}$, then, for o_1 and o_2 that are transformed into the $\mathcal{P}$-$\mathcal{MS}$, we have $o_1 \succeq_{\mathcal{MP}} o_2$.*

Preference-based Planning

In what follows, we show how $\mathcal{P}$-$\mathcal{MS}$ is able to assert preference statements expressed in $\mathcal{PP}$ [Son and Pontelli, 2004] that is a preference language for planning problems. While we do not discuss the full details of $\mathcal{PP}$ here, we recall the main definitions found in [Son and Pontelli, 2004]. Given a set of fluent symbols $\mathcal{F}$ and a set of actions $\mathcal{A}$, a state is defined as a subset of $\mathcal{F}$. A planing problem is a triple $\langle D, I, G\rangle$ where D indicates pre-conditions and effects of actions, I is the initial state, and G stands for the goal state. A solution to a planing problem, that is called a plan, is a chain of actions and states $I, a_1, ...a_n, G$ that starts from I and ends to G. A *basic desire* ϕ is identified as one of the following: 1) a certain action occurs in the plan denoted by $\phi \equiv occ(a)$, 2) a set of certain fluents are satisfied that is denoted by $\phi \equiv (f_i \wedge ... \wedge f_{i+n})$, 3) any combination of *basic desires* by using classical logic connectives (e.g. $\wedge$, $\vee$, and $\neg$) or temporal connective stemmed from temporal logic such as Next(ϕ_1), Until(ϕ_1, ϕ_2), Always(ϕ), and Eventually(ϕ).

[Son and Pontelli, 2004] state that a planning problem $\langle D, I, G\rangle$ can be reduced to an Answer Set Programming (ASP) problem $\Pi(D, I, G)$ such that for a feasible plan p_M there is an answer set M in program Π. In the context of answer set programming, a formula ϕ is satisfied in M if it is a subset of vocabulary symbols that M makes true. For two plans p_1 and p_2, we say p_1 is preferred to p_2 with respect to a *basic desire* ϕ if ϕ is satisfied in p_1 but not in p_2. In the context of ASP, if M_1 and M_2 are two answer sets of p_1 and p_2 respectively, M_1 satisfies ϕ but M_2 does not.

To express *basic desires* in $\mathcal{P}$-$\mathcal{MS}$, it suffices to show that answer sets can be translated to structures in the context of modular systems. Consider a vocabulary $\{a_1, ..., a_n\}$ and an answer set $M=\{a_1, ..., a_k\}$ $(k \leq n)$. As it can be observed from the notion of answer sets, M can be viewed as a structure that interprets each atom a_i, $i \leq k$, as true and for all a_j, $k<j \leq n$, as false. Having the same argument, a *basic desire* ϕ is a partial structure in modular systems such that a subset of atoms in M is true. As a result, a planning problem $\langle D, I, G\rangle$ with preferences can be translated to answer sets and then to modular systems. Assume that structure $\mathcal{B}$ represents plan p_1 , structure $\mathcal{B}'$ is translation of p_2, and formula ϕ is translated to partial structure $\mathcal{B}_\phi$. A plan p_1 is preferred to p_2 with respect to ϕ when $\mathcal{B}$ is preferred to $\mathcal{B}'$ with respect to $\mathcal{B}_\phi$. This completely coincides with our definition of preferences in modular systems. The following result follows from what we discussed.

Theorem 3. *Let p_1 and p_2 be two feasible plans of a planning problem $\langle D, I, G\rangle$ that can be translated to ASP program $\Pi(D, I, G)$. Let M_{p_1} and M_{p_2} be ASP translation of p_1 and p_2 respectively. Suppose that M_{p_1} is translated to structure $\mathcal{B}$ and M_{p_2} to structure $\mathcal{B}'$ in the context of $\mathcal{P}$-$\mathcal{MS}$. Given a basic desire ϕ, if p_1 is preferred to p_2 with respect to ϕ in language $\mathcal{PP}$, then $\mathcal{B} \succ_{\mathcal{MP}_\phi} \mathcal{B}'$ in $\mathcal{P}$-$\mathcal{MS}$ where $\mathcal{MP}_\phi$ is translation of ϕ into $\mathcal{P}$-$\mathcal{MS}$.*

5 Conclusion and Future Work

We proposed an abstract framework for unifying preference languages in modular systems. We introduced the notion of preference-based modular systems ($\mathcal{P}$-$\mathcal{MS}$). We demonstrated that a system obtained through combination of some $\mathcal{P}$-$\mathcal{MS}$ is also a $\mathcal{P}$-$\mathcal{MS}$. We studied how preferences expressed in other languages (two languages as examples) can be translated to our framework. Examples included two common preference languages: cp-nets and planning with preferences ($\mathcal{PP}$). Our future work will address expressivity and computational issues of the framework. We will continue our study of practical aspects of our framework in AI applications, in particular, Business Processes that have complex modular structures and different users may communicate through different formal languages.

References

[Baier and McIlraith, 2008] Jorge A. Baier and Sheila A. McIlraith. Planning with preferences. *AI Magazine*, 29(4):25–36, 2008.

[Borzsony *et al.*, 2001] Stephan Borzsony, Donald Kossmann, and Konrad Stocker. The skyline operator. In *Data Engineering, 2001. Proceedings. 17th International Conference on IEEE*, pages 421–430. IEEE, 2001.

[Boutilier *et al.*, 2004a] Craig Boutilier, Ronen Brafman, Carmel Domshlak, Holger Hoos, and David Poole. Cp-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements. *J. Artif. Intell. Res.(JAIR)*, 21:135–191, 2004.

[Boutilier *et al.*, 2004b] Craig Boutilier, Ronen Brafman, Carmel Domshlak, Holger H. Hoos, and David Poole. Preference-based constrained optimization with cp-nets. *Computational Intelligence*, 20(2):137–157, 2004.

[Brafman and Domshlak, 2009] Ronen Brafman and Carmel Domshlak. Preference handling-an introductory tutorial. *AI magazine*, 30(1):58, 2009.

[Brafman *et al.*, 2006] Ronen I. Brafman, Carmel Domshlak, and Solomon Eyal Shimony. On graphical modeling of preference and importance. *J. Artif. Intell. Res. (JAIR)*, 25:389–424, 2006.

[Brewka *et al.*, 2003] Gerhard Brewka, Ilkka Niemela, and Mirosaw Truszczynski. Answer set optimization. In *IJCAI*, volume 3, pages 867–872, 2003.

[Brewka *et al.*, 2010] Gerhard Brewka, Miroslaw Truszczynski, and Stefan Woltran. Representing preferences among sets. In *AAAI*, 2010.

[Delgrande *et al.*, 2003] James P. Delgrande, Torsten Schaub, and Hans Tompits. A framework for compiling preferences in logic programs. *TPLP*, 3(2):129–187, 2003.

[Delgrande *et al.*, 2004] James Delgrande, Torsten Schaub, Hans Tompits, and Kewen Wang. A classification and survey of preference handling approaches in nonmonotonic reasoning. *Computational Intelligence*, 20(2):308–334, 2004.

[Delgrande *et al.*, 2007] James P. Delgrande, Torsten Schaub, and Hans Tompits. A general framework for expressing preferences in causal reasoning and planning. *J. Log. Comput.*, 17(5):871–907, 2007.

[Faber *et al.*, 2013] Wolfgang Faber, Mirosław Truszczyński, and Stefan Woltran. Abstract preference frameworksa unifying perspective on separability and strong equivalence. In *Twenty-Seventh AAAI Conference on Artificial Intelligence*, 2013.

[Järvisalo *et al.*, 2009] Matti Järvisalo, Emilia Oikarinen, Tomi Janhunen, and Ilkka Niemelä. A module-based framework for multi-language constraint modeling. In *Proceedings of the 10th International Conference on Logic Programming and Non-monotonic Reasoning (LPNMR'09)*, volume 5753 of *Lecture Notes in Computer Science (LNCS)*, pages 155–168. Springer-Verlag, 2009.

[Kiessling, 2002] Werner Kiessling. Foundations of preferences in database systems. In *Proceedings of the 28th international conference on Very Large Data Bases*, pages 311–322. VLDB Endowment, 2002.

[Kleene, 1952] Stephen Cole Kleene. *Introduction to metamathematics*. North-Holland Publishing Company, 1952.

[Mitchell and Ternovska, 2005] David G. Mitchell and Eugenia Ternovska. A framework for representing and solving NP search problems. In *Proceedings, The Twentieth National Conference on Artificial Intelligence and the Seventeenth Innovative Applications of Artificial Intelligence Conference, Pittsburgh, Pennsylvania, USA*, pages 430–435, 2005.

[Ross, 2007] Kenneth Ross. On the adequacy of partial orders for preference composition. In *DBRank Workshop*, 2007.

[Santhanam *et al.*, 2011] Ganesh Ram Santhanam, Samik Basu, and Vasant Honavar. Representing and reasoning with qualitative preferences for compositional systems. *Journal of Artificial Intelligence Research*, 42(1):211–274, 2011.

[Sohrabi *et al.*, 2008] Shirin Sohrabi, Jorge A Baier, and Sheila A McIlraith. Htn planning with preferences. In *Twenty-First International Joint Conference on Artificial Intelligence*, 2008.

[Son and Pontelli, 2004] Tran Cao Son and Enrico Pontelli. Planning with preferences using logic programming. In *Logic Programming and Nonmonotonic Reasoning*, pages 247–260. Springer, 2004.

[Stefanidis *et al.*, 2011] Kostas Stefanidis, Georgia Koutrika, and Evaggelia Pitoura. A survey on representation composition and application of preferences in database systems. *ACM Transactions on Database Systems (TODS)*, 36:19–28, 2011.

[Tasharrofi and Ternovska, 2011] Shahab Tasharrofi and Eugenia Ternovska. A semantic account for modularity in multi-language modelling of search problems. In *Frontiers of Combining Systems, 8th International Symposium, FroCoS 2011, Germany*, pages 259–274, 2011.

[Tasharrofi *et al.*, 2011] Shahab Tasharrofi, Xiongnan Wu, and Eugenia Ternovska. Solving modular model expansion tasks. *arXiv preprint arXiv:1109.0583*, 2011.

[Wilson, 2004] Nic Wilson. Consistency and constrained optimisation for conditional preferences. In *Proceedings of the 16th Eureopean Conference on Artificial Intelligence, ECAI'2004, Valencia, Spain*, pages 888–894, 2004.