Abstract Preference Frameworks — A Unifying Perspective on Separability and Strong Equivalence

Wolfgang Faber
University of Calabria
faber@mat.unical.it

Mirosław Truszczyński
University of Kentucky
mirek@cs.uky.edu

Stefan Woltran
Vienna University of Technology
woltran@dbai.tuwien.ac.at

Abstract

We introduce *abstract preference frameworks* to study general properties common across a variety of preference formalisms. In particular, we study *strong equivalence* in preference formalisms and their *separability*. We identify abstract postulates on preference frameworks, satisfied by most of the currently studied preference formalisms, that lead to characterizations of both properties of interest.

Introduction

The literature on preferences and preference formalisms is vast; the collection of articles edited by Goldsmith and Junker (2008) and the monograph by Kaci (2011) provide good overviews and are excellent sources of references. The main objectives of the area are to design expressive yet intuitive languages to model preferences, and to characterize the notions of optimality they define. Recently, researchers identified another fundamental problem related to preference languages, that of characterizing various notions of *equivalence* of preference theories. A particularly important one is *strong equivalence* (Faber, Truszczyński, and Woltran 2012). Strong equivalence guarantees semantic-preserving replacements of parts of preference theories, the so-called *replacement property*, and is fundamental for understanding preference rewriting and modularity.

Preference formalisms are inherently nonmonotonic, that is, additional preferences can add to the set of optimal outcomes and not only remove from them. Consequently, they figured prominently in the studies of nonmonotonic logics (Delgrande et al. 2004). Faber et al. (2012) observed that as in other nonmonotonic formalisms, the standard notion of equivalence, requiring that two theories have the same preferred outcomes, is too weak to guarantee the replacement property. Building on earlier work on strong equivalence of logic programs with the answer-set semantics (Lifschitz, Pearce, and Valverde 2001), Faber et al. (2012) introduced and studied strong equivalence of preference theories in the language of answer-set optimization (ASO) problems.

Our goal is to identify general principles behind strong equivalence in order to extend the results of Faber et al.

(2012) to other preference formalisms. The main challenge is the vast diversity of preference formalisms. To overcome it, we propose the notion of an *abstract preference framework* and study strong equivalence in that language.

Preference languages have to capture two phenomena: (i) physical and logical *constraints* on the domain of interest that *must* be obeyed — they give rise to *intended* or *feasible outcomes*; and (ii) *preferences* that describe what is desirable (but not absolutely necessary) — they are employed to select out of the intended outcomes the *most preferred* or *desirable* ones. The distinction between the two is often explicit in the language, such that different constructs specify constraints and preferences, and preference theories are *pairs* consisting of a theory in a "pure" constraint formalism and another theory in a "pure" preference one. Following the terminology proposed by Faber et al. (2012), we refer to these two components as *generators* and *selectors*, respectively. Examples of such preference systems include logic programs with optimization statements (Buccafurri, Leone, and Rullo 2000; Simons, Niemelä, and Soininen 2002), ASO problems (Faber, Truszczyński, and Woltran 2012), and formalisms obtained by extending common preference languages such as CP-nets with a constraint language (Boutilier et al. 2004).

However, in other languages such a distinction is less clear. Theories are built of statements that combine constraints with preferences in ways that make it hard to separate their effects. Examples of such languages include prioritized versions of logic programming and default logic (Brewka and Eiter 1999; Delgrande et al. 2004) and logic programs with ordered disjunctions (LPODs) (Brewka, Niemelä, and Syrjänen 2004). In such formalisms, one can define the concepts of a generator and a selector. The problem is how to represent an arbitrary theory as a pair consisting of a generator and a selector, and whether such a "separation" is even possible. The preference framework that we introduce here provides a setting that encompasses *both* types of preference languages and allows us to study the question of "separability" in abstract terms.

The concepts of generator and selector theories suggest several forms of strong equivalence. We say that preference theories x, y are *strongly equivalent* (*strongly generator equivalent*, *strongly selector equivalent*) if for every preference (generator, selector) theory z, the extensions of x and of y with z have the same semantics, that is, the same preferred

(or optimal) outcomes. The main objective when studying strong equivalence is to find characterizations which do not refer to z and can be stated entirely in terms of theories x and y being compared. Such characterizations are known for nonmonotonic logics (Lifschitz, Pearce, and Valverde 2001; Turner 2003) and for some specific preference formalisms (Faber and Konczak 2006; Faber, Tompits, and Woltran 2008; Faber, Truszczyński, and Woltran 2012). By imposing abstract postulates on the semantics of preference frameworks, we obtain the result already anticipated by Faber et al. (2012), namely that for many preference formalisms the characterization of strong equivalence is fully determined by characterizations of the simpler notions of strong generator and strong selector equivalence. Under some additional postulates on preferences we also obtain a natural abstract characterization of strong selector equivalence.

The main contributions of our paper are as follows (for full proofs, see (Faber, Truszczyński, and Woltran 2013)):

- We introduce the notion of an abstract preference framework. Abstract preference frameworks make weak syntactic assumptions and their semantics is specified in terms of intended outcomes and preference preorders.

- We introduce abstract *generator* and *selector* frameworks for describing constraints and preferences independently of each other. We use these notions to define *separated* abstract preference frameworks which, speaking informally, are the products of generator and selector frameworks.

- For general abstract preference frameworks we study the problem of their *separability*, that is representability by means of a separated framework, and provide sufficient conditions to guarantee that property.

- We characterize strong equivalence in separated preference frameworks in terms of characterizations of strong equivalence in generator and selector frameworks. We then lift the results to the separable case. The results apply to *all* preference formalisms which are (separable) instantiations of our framework.

- We illustrate the notion of separability by showing that logic programs with ordered disjunction are separable and, therefore, in principle, admit a simpler, separated representation.

Abstract Preference Frameworks

We represent preference theories by elements from some set L. We make no assumptions on the syntax of preference theories. However, we allow two preference theories to be combined into a single one. We denote that theory "conjunction" (or "union") operation by the symbol $\wedge$. We impose on $\wedge$ the properties of commutativity (the order in which the theories to be combined are listed should not matter), associativity (if more than two theories are to be combined, the order in which $\wedge$ is applied should not matter), and idempotence (combining a theory with itself should not change the meaning of the theory). We also assume the existence of a preference theory, denoted by $\top$, that when conjoined with any other preference theory, does not change the meaning of the latter (that is, does not impose any constraints and

does not distinguish between any two outcomes). This is modeled by assuming that $\top$ is the unit element of $\wedge$. In preference formalisms considered in the literature, preference theories are typically represented as sets of "elementary" preference formulas. In such cases, L is the powerset of the set of preference formulas, the union operator plays the role of $\wedge$ and the empty theory that of $\top$. All such cases fall under the scope of our abstract representation. Examples include the penalty and possibilistic logics (Kaci 2011), answer set optimization (Faber, Truszczyński, and Woltran 2012), and general CP-nets (Wilson 2004; Goldsmith et al. 2008).

An algebraic structure $\mathcal{L} = \langle L, \wedge, \top \rangle$ with the properties we enumerated is a *bounded meet semilattice*. Bounded meet semilattices, for simplicity referred to from now on just as semilattices (as we do not consider any other semilattices here), arguably represent the weakest abstract desiderata on the space of preference theories. Semilattices can equivalently be thought of as partially ordered sets that have a greatest element and in which every finite set has a greatest lower bound: the relation $\preceq$ defined by $x \preceq y$ if $x \wedge y = x$ is such a partial order.

Since we do not adopt any syntactic assumptions on preference theories, we also do not make any assumptions about the nature of outcomes. We simply assume that they are elements of some set $\mathcal{I}$ of all possible outcomes. To reflect the fact that preference theories typically encompass both constraints and preferences, we specify the semantics of preference theories by means of two functions. The first one, ι, assigns to each preference theory x its set of feasible outcomes $\iota(x)$. It models the constraints contained in x. However, the constraints show up only implicitly, we see them through the effect they have on the outcomes — $\iota(x)$ consists precisely of those outcomes that satisfy all the constraints. The second function, $\geq$, assigns to x the preorder $\geq_x$ implied by the preferences in x. That preorder specifies the concept of desirability represented by x. If outcomes $\alpha, \beta \in \mathcal{I}$ satisfy $\alpha \geq_x \beta$, then α is *at least as desirable as* β. The preferences in x are, again, implicit, as is the mechanism by which they are combined to yield the preorder $\geq_x$.

These considerations lead us to the following definition of an abstract preference framework.

Definition 1 *An* abstract preference framework *(or just a* preference framework*) is a quadruple* $\sigma = \langle \mathcal{L}, \mathcal{I}, \iota, \geq \rangle$, *where* $\mathcal{L} = \langle L, \wedge, \top \rangle$ *is a semilattice of preference theories;* $\mathcal{I}$ *is the space of* outcomes*;* ι *is a function that assigns to each* $x \in L$ *a set* $\iota(x) \subseteq \mathcal{I}$ *of* feasible *(or* intended*) outcomes for* x*; and* $\geq$ *is a function that assigns to each* $x \in L$ *a preorder* $\geq_x$ *on* $\mathcal{I}$ *(a binary relation that is reflexive and transitive).*

The preorder $\geq_x$ gives rise to its *strict* version $>_x$. For every outcomes $\alpha, \beta \in \mathcal{I}$, we define $\alpha >_x \beta$ if $\alpha \geq_x \beta$ and $\beta \not\geq_x \alpha$, and read $\alpha >_x \beta$ as "α *is strictly more desirable than* β in a preference theory x." For $\succ \in \{>, \geq\}$ and $S \subseteq \mathcal{I}$, we also use $\succ^S$ for the restriction of $\succ$ to S.

Let $\sigma = \langle \mathcal{L}, \mathcal{I}, \iota, \geq \rangle$ be a preference framework. For every $x \in L$, we define the set of *optimal* outcomes for x:

$$\pi(x) = \{\alpha \in \iota(x) \mid \text{for every } \beta \in \iota(x), \beta \not>_x \alpha\}.$$

The function π assigning optimal outcomes to preference theories determines the semantics of σ.

In some situations, we consider optimality wrt a preorder $\geq_x$ but in the context of a set S of outcomes other than $\iota(x)$. In such cases we use the notation

$$\pi_S(x) = \{\alpha \in S \mid \text{for every } \beta \in S, \beta \not\succ_x \alpha\}.$$

Generators and Selectors. Preference theories typically combine constraint and preference components in a non-trivial way. Some preference theories can however be regarded as concerned exclusively with just one of these two aspects. We thus define for a preference framework $\langle \mathcal{L}, \mathcal{I}, \iota, \geq \rangle$ its *generator theories* L^g, and its its *selector theories* L^π as

1. $L^g = \{x \in L \mid \text{for every } y \in L, \geq_{y \wedge x} = \geq_y\}$,

2. $L^\pi = \{x \in L \mid \text{for every } y \in L, \iota(y \wedge x) = \iota(y)\}$.

Elements in L^g (L^π, respectively) can be regarded as concerned purely with constraints (preferences, respectively) as they do not affect preferences (constraints, respectively) when conjoined with any other preference theory.

In any preference framework $\langle \mathcal{L}, \mathcal{I}, \iota, \geq \rangle$, since $\top$ is the unit element of $\wedge$, $\top \in L^g$ and $\top \in L^\pi$. Moreover, for every $y, z \in L^g$,

$$\geq_{x \wedge (y \wedge z)} = \geq_{(x \wedge y) \wedge z} = \geq_{x \wedge y} = \geq_x$$

and for every $y, z \in L^\pi$,

$$\iota(x \wedge (y \wedge z)) = \iota((x \wedge y) \wedge z) = \iota(x \wedge y) = \iota(x).$$

Thus, both L^g and L^π are closed under $\wedge$. It follows that $\langle L^g, \wedge, \top \rangle$ and $\langle L^\pi, \wedge, \top \rangle$ are semilattices and, in fact, sub-semilattices of $\langle L, \wedge, \top \rangle$. They can be viewed as "one-dimensional" preference frameworks concerned with generating intended outcomes, and with selecting among all outcomes those optimal wrt preferences. One of the key questions, which we discuss later in the paper, is whether a preference framework is "separable," that is, can be reconstructed from its one-dimensional "generator" and "selector" frameworks (or of some sub-frameworks thereof).

(Non)Monotonicity. A semantics s on preference theories, that is, any function from L to $\mathcal{I}$, for instance, ι or π, is *monotone* if for every $x, z \in L$, $s(x \wedge z) \subseteq s(x)$ (equivalently, if for every $x, y \in L$ such that $x \preceq y$, $s(x) \subseteq s(y)$). In general, (the semantics of) preference formalisms are not monotone as new constraints or preferences (or both) may "promote" non-optimal outcomes to become optimal. New constraints may render outcomes that were more desirable no longer feasible and hence, eliminate them from the optimization process; new preferences may put additional importance on some non-optimal outcomes, pushing them up on the scale of desirability.

If only the preferences behave nonmonotonically, our framework is already sufficient. However, in general, we require one further concept determined by ι. An outcome $\alpha \in \mathcal{I}$ is *potentially intended* for a preference theory $x \in L$ if it is an intended outcome of $x \wedge z$, for some $z \in L$ (becomes intended under some extension of x). We denote the set of all potentially intended outcomes of x by $\mu(x)$:

$$\mu(x) = \bigcup_{y \in L} \iota(x \wedge y).$$

Clearly, for every $x \in L$, $\iota(x) \subseteq \mu(x)$ (since $x \wedge \top = x$). One can also check that if ι is monotone then $\iota(x) = \mu(x)$ (and then does not need not to be explicitly introduced).

Notions of Equivalence. The fact that preference theories are not monotonic has an important consequence when considering two preference theories as equivalent. The first concept that comes to mind is based on the comparison of optimal outcomes: two preference theories x and y are *equivalent*, written $x \equiv y$, if they have the *same* optimal outcomes, that is, if $\pi(x) = \pi(y)$. However, extending x and y with new information z may "promote" to the level of optimality outcomes that (without z in the picture) were below optimal ones, a manifestation of nonmonotonicity of preference formalisms discussed above. Hence, $\pi(x) = \pi(y)$ does not necessarily imply that $\pi(x \wedge z) = \pi(y \wedge z)$. In other words, the equivalence of x and y is too weak to guarantee mutual replaceability of x and y wrt common new information.

Consequently, an alternative concept of equivalence, based directly on the idea of replaceability is of more interest. Preference theories x and y are *strongly equivalent*, written $x \equiv_s y$, if for every preference theory z, $x \wedge z \equiv y \wedge z$, that is, $\pi(x \wedge z) = \pi(y \wedge z)$. We already mentioned the concept of separability of a preference framework into its generator and selector components. These give rise for two further definitions: preference theories x and y are *strongly g-equivalent*, in symbols $x \equiv_s^g y$, if for each $z \in L^g$, $\pi(x \wedge z) = \pi(y \wedge z)$, and *strongly π-equivalent*, in symbols $x \equiv_s^\pi y$, if for each $z \in L^\pi$, $\pi(x \wedge z) = \pi(y \wedge z)$. In this paper we study characterizations of strong equivalence in the abstract setting of preference frameworks, and show that strong equivalence of preference theories can be understood in terms of the "one-dimensional" versions of strong equivalence just introduced.

Separated Preference Frameworks

We start with the case of preference frameworks in which the separation between constraints and preferences is explicit. Most of the current preference formalisms are or can be extended to be in that form. We start by defining "one-dimensional" preference frameworks which are concerned only with constraints, resp. with preferences.

A *generator framework* is a triple $\sigma_g = \langle \mathcal{G}, \mathcal{I}, \iota \rangle$, where $\mathcal{G} = \langle G, \wedge, \top \rangle$ is a semilattice, $\mathcal{I}$ is a set of outcomes and ι is a function assigning to each $g \in G$ a set $\iota(g) \subseteq \mathcal{I}$ of *intended* outcomes for g. Since the concept of a potentially intended outcome in preference frameworks depends only on the function ι, it can also be defined for generator frameworks where, as before we write $\mu(x)$ for the set of all potentially intended outcomes of $x \in G$. For $x, y \in G$, we write (with some abuse of notation) $x \equiv_s y$ to denote that for every $z \in G$, $\iota(x \wedge z) = \iota(y \wedge z)$ and refer to this relation as strong equivalence in σ_g. Strong equivalence implies equivalence wrt potentially intended outcomes.

Proposition 1 *Let $\mathcal{G} = \langle G, \wedge, \top \rangle$ be a generator framework. For every $x, y \in G$, if $x \equiv_s y$ then $\mu(x) = \mu(y)$.*

A *selector framework* is a triple $\sigma_\pi = \langle \mathcal{P}, \mathcal{I}, \geq \rangle$, where $\mathcal{P} = \langle P, \wedge, \top \rangle$ is a semilattice, $\mathcal{I}$ is a set of outcomes and $\geq$ is a function assigning to each $p \in P$ a preorder $\geq_p$ on

$\mathcal{I}$. For $p, q \in P$, and $S \subseteq \mathcal{I}$ we write $p \equiv_{s,S} q$ if for every $r \in P$, $\pi_S(p \wedge r) = \pi_S(q \wedge r)$ and refer to this relation as strong equivalence in σ_π relative to S.

From a generator framework $\sigma_g = \langle \mathcal{G}, \mathcal{I}, \iota \rangle$ and a selector framework $\sigma_\pi = \langle \mathcal{P}, \mathcal{I}, \geq \rangle$, we can build a preference framework $\sigma_g \times \sigma_\pi = \langle \mathcal{L}, \mathcal{I}, \iota', \geq' \rangle$ by setting:

1. $\mathcal{L} = \mathcal{G} \times \mathcal{P}$, that is, $\mathcal{L}$ is the product of $\mathcal{G}$ and $\mathcal{P}$ ($L = G \times P$; for every $(x, p), (y, q) \in L$, $(x, p) \wedge (y, q) = (x \wedge y, p \wedge q)$; and the top element of $\mathcal{L}$ is $(\top, \top)$)

2. for every $(x, p) \in L$, $\iota'((x, p)) = \iota(x)$ and $\geq'_{(x,p)} = \geq_p$.

We call preference frameworks of that type *separated*. To simplify the notation we use the same symbols for the corresponding concepts coming from different semilattices. Thus, we write ι for ι' and $\geq$ for $\geq'$. We also often write $\iota(x, p)$ for $\iota((x, p))$ and similarly $\mu(x, p)$ for $\mu((x, p))$. Finally, we note that for every $(x, p) \in L$,

$$\mu(x, p) = \bigcup_{(y,q) \in L} \iota((x, p) \wedge (y, q)) = \bigcup_{(y,q) \in L} \iota(x \wedge y, p \wedge q)$$
$$= \bigcup_{y \in G} \iota(x \wedge y) = \mu(x).$$

(Here μ on the left-hand side denotes the "μ"-function for $\mathcal{L}$ and the one on the right-hand side the "μ"-function for $\mathcal{G}$.)

To obtain useful characterizations of strong equivalence in separated frameworks, we impose some additional assumptions on generator and selector frameworks. These assumptions are natural and hold for many specific formalisms.

The filter property. Let $\sigma_g = \langle \mathcal{G}, \mathcal{I}, \iota \rangle$ be a generator framework. An element $y \in G$ is a *filter* if for every $x \in G$, $\iota(x \wedge y) = \iota(x) \cap \mu(y)$. A preference framework σ_g satisfies the *filter* property if for every outcome $\alpha \in \mathcal{I}$, there is a filter y such that $\mu(y) = \{\alpha\}$.

The generator promotion (GP) property. A generator framework $\sigma_g = \langle \mathcal{G}, \mathcal{I}, \iota \rangle$ satisfies the *generator promotion* property if for every $x, y \in G$ and every $\alpha, \beta \in \mu(x) \cap \mu(y)$, there is $z \in G$ such that $\iota(x \wedge z) = \iota(y \wedge z) = \{\alpha, \beta\}$.

All standard constraint satisfaction formalisms, including propositional logic and answer set programming (with choice rules), satisfy both properties.

The first result provides conditions characterizing the property $(x, p) \equiv_s^g (y, q)$. We note that the statement claiming that $\mu(x) = \mu(y)$ is justified (cf. Proposition 1).

Theorem 2 *Let $\sigma_g \times \sigma_\pi$ be a separated preference framework obtained from $\sigma_g = \langle \mathcal{G}, \mathcal{I}, \iota \rangle$ and $\sigma_\pi = \langle \mathcal{P}, \mathcal{I}, \geq \rangle$, and let σ_g satisfy the filter and the GP properties. For every $x, y \in G$ and $p, q \in P$, $(x, p) \equiv_s^g (y, q)$ if and only if $x \equiv_s y$ (in σ_g) and $>_p^\mu = >_q^\mu$, where μ is the common value of $\mu(x)$ and $\mu(y)$.*

This result shows that strong equivalence wrt changing generators in a separated framework can be characterized by the strong equivalence in the generator framework, and a natural condition on the preference orders. An interesting aspect here is that $G \times \{\top\}$ is (in general) a proper subset of L^g. Nevertheless, $\equiv_s^g$, which is defined relative to elements in L^g can be characterized in terms of $\equiv_s$ in σ_g, which is defined relative to elements in G.

Next, we consider strong equivalence wrt changing selectors. As before, we introduce an additional assumption.

The selector promotion (SP) property. For every $\alpha, \beta \in \mathcal{I}$ and for every $p, q \in P$, there is $t \in P$ such that (i) $\alpha >_s \beta$ if and only if $\alpha >_{s \wedge t} \beta$, where $s \in \{p, q\}$; (ii) for every $\gamma \in \mathcal{I} \setminus \{\alpha, \beta\}$, $\gamma \not>_{s \wedge t} \alpha, \beta$, where $s \in \{p, q\}$.

We note that the SP property is not particularly restrictive. It holds for many preference formalisms, in particular, for the selectors used by ASO problems (Faber, Truszczyński, and Woltran 2012), as well as in preference formalisms based on some forms of utility such as the penalty and possibilistic logics.

Theorem 3 *Let $\sigma_g \times \sigma_\pi$ be a separated preference framework obtained from $\sigma_g = \langle \mathcal{G}, \mathcal{I}, \iota \rangle$ and $\sigma_\pi = \langle \mathcal{P}, \mathcal{I}, \geq \rangle$, and let σ_π satisfy the SP property. For every $x, y \in G$ and $p, q \in P$, $(x, p) \equiv_s^\pi (y, q)$ if and only if $\iota(x) = \iota(y)$ and $p \equiv_{s,\iota} q$ (in σ_π) with ι the common value of $\iota(x)$ and $\iota(y)$.*

This result shows that strong equivalence wrt changing selectors in a separated framework, can be characterized by the strong equivalence in the selector framework relative to a certain set of outcomes (the set of intended outcomes for x and y). As before, it is important to stress a non-trivial aspect of the characterization that comes from the fact that $\{\top\} \times P$ is in general a proper subset of L^π.

Finally, we present results concerning the "combined" case of strong equivalence in a separated framework.

Theorem 4 *Let $\sigma_g \times \sigma_\pi$ be a separated preference framework obtained from $\sigma_g = \langle \mathcal{G}, \mathcal{I}, \iota \rangle$ and $\sigma_\pi = \langle \mathcal{P}, \mathcal{I}, \geq \rangle$, and let σ_g satisfy the filter and the GP properties and σ_π satisfy the SP property. For every $x, y \in G$ and $p, q \in P$, $(x, p) \equiv_s (y, q)$ if and only if $x \equiv_s y$ (in σ_g), and $p \equiv_{s,\mu} q$ (in σ_π), where μ is the common value of $\mu(x)$ and $\mu(y)$.*

This result shows that strong equivalence in a separated framework can be characterized by the strong equivalence of the generator components and the strong equivalence of the selector components relative to the set of potentially intended outcomes for x and y. Potentially intended outcomes are relevant here (and not intended outcomes, as in the case of Theorem 3) because now generators can vary too and that may make *any* potentially intended outcome intended in the extended theory.

Separable Preference Frameworks

A preference framework $\sigma = \langle \mathcal{L}, \mathcal{I}, \iota, \geq \rangle$ gives rise to sub-semilattices $\mathcal{L}^g = \langle L^g, \wedge, \top \rangle$ and $\mathcal{L}^\pi = \langle L^\pi, \wedge, \top \rangle$, and to two related preference frameworks:

1. $\sigma^* = \langle \mathcal{L}^*, \mathcal{I}, \iota, \geq \rangle$, where $L^* = \{x_g \wedge x_\pi \colon x_g \in L^g, \; x_\pi \in L^\pi\}$, and ι and $\geq$ are the restrictions to L^* of the corresponding functions in σ; it is easy to verify that $\mathcal{L}^*$ is a (bounded meet) semilattice and a sub-semilattice of $\mathcal{L}$

2. $\sigma^\times$, the separated framework $\sigma_g \times \sigma_\pi$ obtained from the generator framework $\sigma_g = \langle \mathcal{L}^g, \mathcal{I}, \iota \rangle$ and the selector framework $\sigma_\pi = \langle \mathcal{L}^\pi, \mathcal{I}, \geq \rangle$.

The strong equivalence concepts in $\sigma^\times$ can be expressed in terms of strong equivalence in σ_g and σ_π as discussed in the

previous section. We will show that these characterizations extend to strong equivalence notions in σ^* and σ. The first result relates strong equivalence in preference frameworks $\sigma^\times$ and σ^*. Regarding items (2) and (3) in Theorems 5 and 6 below, we note that for elements in L^*, the relations $\equiv_s^g$ and $\equiv_s^\pi$ in σ and σ^* coincide, as both are defined relative to the same sets of elements, L^g and L^π, respectively.

Theorem 5 *For every $x_g, y_g \in L^g$ and $x_\pi, y_\pi \in L^\pi$:*

1. $x_g \wedge x_\pi \equiv_s y_g \wedge y_\pi$ *(in σ^*) if and only if $(x_g, x_\pi) \equiv_s (y_g, y_\pi)$*
2. $x_g \wedge x_\pi \equiv_s^g y_g \wedge y_\pi$ *if and only if $(x_g, x_\pi) \equiv_s^g (y_g, y_\pi)$*
3. $x_g \wedge x_\pi \equiv_s^\pi y_g \wedge y_\pi$ *if and only if $(x_g, x_\pi) \equiv_s^\pi (y_g, y_\pi)$.*

Theorem 5 implies that to relate strong equivalence in $\sigma^\times$ and σ, it suffices to relate strong equivalence in σ^* and σ. To this end, we need additional concepts and assumptions.

A preference framework $\sigma = \langle L, \mathcal{I}, \iota, \geq \rangle$ is *g-complete*, if there is a homomorphism $(\cdot)^g$ from L to L^g (that is, for every $x \in L$, $x \mapsto x^g \in L^g$) such that for every $x \in L$, $\iota(x) = \iota(x^g)$. A preference framework $\langle \mathcal{L}, \mathcal{I}, \iota, \geq \rangle$ is *π-complete*, if there is a homomorphism $(\cdot)^\pi$ from L to L^π (that is, for every $x \in L$, $x \mapsto x^\pi \in L^\pi$) such that for every $x \in L$, $\geq_x = \geq_{x^\pi}$. A preference framework $\langle \mathcal{L}, \mathcal{I}, \iota, \geq \rangle$ is *separable* if it is both g- and π-complete and if:

1. the homomorphism $(\cdot)^g$ is the identity on L^g
2. the homomorphism $(\cdot)^\pi$ is the identity on L^π
3. for every $x \in L$, $(x^g)^\pi \in L^g$ and $(x^\pi)^g \in L^\pi$.

We now define a function $(\cdot)^* : L \to L$ (that is, for every $x \in L$, $x \mapsto x^* \in L$) by $x^* = x^g \wedge x^\pi$.

Theorem 6 *Let $\sigma = \langle \mathcal{L}, \mathcal{I}, \iota, \geq \rangle$ be a separable preference framework. For every $x, y \in L$:*

1. $x \equiv_s y$ *(in σ) if and only if $x^* \equiv_s y^*$ (in σ^*)*
2. $x \equiv_s^g y$ *(in σ) if and only if $x^* \equiv_s^g y^*$*
3. $x \equiv_s^\pi y$ *(in σ) if and only if $x^* \equiv_s^\pi y^*$.*

Putting Theorems 5 and 6 together with earlier characterizations of strong equivalence in *separated* frameworks, we obtain the following result of strong equivalence in *separable* frameworks.

Theorem 7 *Let σ be a separable preference framework, such that the generator framework σ_g (of $\sigma^\times$) satisfies the filter and the GP properties and the selector framework σ_π (of $\sigma^\times$) satisfies the SP property. Then*

1. $x \equiv_s^g y$ *if and only if $x^g \equiv_s y^g$ (in σ_g) and $>_{x^\pi}^\mu = >_{y^\pi}^\mu$, where μ is the common value of $\mu(x)$ and $\mu(y)$*
2. $x \equiv_s^\pi y$ *if and only if $\iota(x) = \iota(y)$ and $x^\pi \equiv_{s,\iota} y^\pi$ (in σ_π), where ι is the common value of $\iota(x)$ and $\iota(y)$*
3. $x \equiv_s y$ *if and only if $x^g \equiv_s y^g$ (in σ_g) and $x^\pi \equiv_{s,\mu} y^\pi$ (in σ_π), where μ is the common value of $\mu(x)$ and $\mu(y)$.*

Characterizing Strong Equivalence in Selector Frameworks

Strong equivalence in formalisms for specifying constraints is well understood. In most cases, such formalisms satisfy a strong monotonicity property that for every theories x

and y, $\iota(x \wedge y) = \iota(x) \cap \iota(y)$. It is the case for propositional logic and for all standard constraint satisfaction languages. That property implies that for every theories x and y, if $\iota(x) = \iota(y)$ then for every theory z, $\iota(x \wedge z) = \iota(y \wedge z)$. That, in turn, implies that strong and classical equivalence coincide. The situation is different in the formalisms in which the strong monotonicity property above does not hold. Answer-set programming is a prominent example of that class of constraint systems. Strong equivalence in answer set programming has been thoroughly investigated and characterizations of its basic form and numerous variants are well known (Lifschitz, Pearce, and Valverde 2001; Turner 2003). These results can be used to replace the condition "$x^g \equiv_s y^g$ (in σ_g)" in Theorem 7 by more specific ones (depending on a constraint formalism used).

We will now show that under some additional postulates on selector frameworks we can provide an elegant characterization of strong equivalence in selector frameworks, too.

The uniformity property. A selector framework $\sigma_\pi = \langle \mathcal{P}, \mathcal{I}, \geq \rangle$ satisfies the *uniformity* property if for every $p, q, s \in \mathcal{P}$, $\geq_p = \geq_q$ implies $\geq_{p \wedge s} = \geq_{q \wedge s}$.

The conservative promotion (CP) property. For all outcomes $\alpha, \beta \in \mathcal{I}$ and all preferences $p, q \in \mathcal{P}$ such that $\alpha \geq_p \beta$ and $\alpha \not\geq_q \beta$ there is a preference $u \in \mathcal{P}$ such that $\alpha >_{p \wedge u} \beta$ and $\alpha \not\geq_{q \wedge u} \beta$.

Both postulates hold for standard preference formalisms (for instance, the penalty and the possibilistic logics), and ASO problems (Faber, Truszczyński, and Woltran 2012).

Theorem 8 *Let $\sigma_\pi = \langle \mathcal{P}, \mathcal{I}, \geq \rangle$ be a selector framework satisfying the uniformity, CP and SP properties. For every $X \subseteq \mathcal{I}$ and for every $p, q \in P$, $p \equiv_{s,X} q$ if and only if $\geq_p^X = \geq_q^X$.*

This theorem can be used to replace the conditions "$p \equiv_{s,X} q$" and "$x^\pi \equiv_{s,X} y^\pi$" in the earlier results with $\geq_p^X = \geq_q^X$ and $\geq_{x^\pi}^X = \geq_{y^\pi}^X$, respectively (for the appropriate value of X).

Discussion

Our results on separated frameworks are broadly applicable, as they rely only on the weakest assumptions on the structure of generator and selector theories. For instance, our results on separated frameworks apply to ASO problems and yield characterizations for the class of "non-ranked" problems obtained by Faber et al. (2012; Corollary 5 and Corollary 14).

Our results on separable frameworks are also of interest. While preference frameworks that are not explicitly separated are not common, some formalisms of that kind were indeed proposed and studied. A prominent example is the formalism of *logic programs with ordered disjunction* or LPODs (Brewka, Niemelä, and Syrjänen 2004) stemming from qualitative choice logic (Brewka, Benferhat, and Le Berre 2002). LPODs can be cast as a preference framework and, moreover, this framework can be shown to be separable. We will now state these results formally for a slight generalization of LPODs that allows for the use of so-called *choice* rules (Simons 1999), which are widely used in answer set programming.

An LPOD (Brewka, Niemelä, and Syrjänen 2004) is a finite set of rules. These rules can be *ordered disjunction rules* (*od rules*, for short) of the form

$$p_1 \times \cdots \times p_k \leftarrow p_{k+1}, \ldots, p_m, not\ p_{m+1}, \ldots, not\ p_n \quad (1)$$

or *choice rules* of the form

$$\{p_1, \ldots, p_k\} \leftarrow p_{k+1}, \ldots, p_m, not\ p_{m+1}, \ldots, not\ p_n \quad (2)$$

where (in both cases) $1 \leq k \leq m \leq n$, and each p_i ($1 \leq i \leq n$) is an *atom* from a fixed, suitably large universe $\mathcal{U}$. We denote the positive body atoms of a rule r of form (1) or (2) by $body^+(r) = \{p_{k+1}, \ldots, p_m\}$, the negative body atoms as $body^-(r) = \{p_{m+1}, \ldots, p_n\}$, body atoms $body(r) = body^+(r) \cup body^-(r)$, head atoms $head(r) = \{p_1, \ldots, p_k\}$, and for od rules the i-th head atom as $head_i(r) = p_i$ and $head_{<i}(r) = \{p_1, \ldots, p_{i-1}\}$ ($1 \leq i \leq k$).

An interpretation $I \subseteq \mathcal{U}$ satisfies an od rule r ($I \models r$) if $I \cap head(r) \neq \emptyset$ whenever $body^+(r) \subseteq I$, and $I \cap body^-(r) = \emptyset$. A choice rule is satisfied by all interpretations. A model of a program P is an interpretation M that satisfies all of its rules.

The *reduct* r^I, as defined by Faber, Tompits, and Woltran (2008), of an od rule r with respect to an interpretation I is $\{head_i(r) \leftarrow body^+(r)\}$ if $I \cap (body^-(r) \cup head_{<i}(r)) = \emptyset$ and $head_i(r) \in I$ ($i \leq k$); $r^I = \{head_k(r) \leftarrow body^+(r)\}$ if $I \cap (body^-(r) \cup head(r)) = \emptyset$ (where $head_k(r)$ is the last head atom); and $r^I = \emptyset$ otherwise. The *reduct* r^I (Simons 1999) of a choice rule r of the form (2) with respect to an interpretation I is $\{p \leftarrow body^+(r) \mid p \in head(r) \cap I\}$ if $I \cap body^-(r) = \emptyset$; and $r^I = \emptyset$ otherwise.

The reduct P^I of an LPOD P is $\bigcup_{r \in P} r^I$. An interpretation I is an *answer set* if I is the subset minimal model of P^I. The set of all answer sets of P is denoted by $AS(P)$.

An od rule r contributes to degree j in interpretation I ($v_I(r) = j$) if $body^+(r) \subseteq I$, $body^-(r) \cap I = \emptyset$, $head_j(r) \in I$, and $head_{<j}(r) \cap I = \emptyset$ ($j \leq k$). If $body^+(r) \not\subseteq I$ or $body^-(r) \cap I \neq \emptyset$, the rule is irrelevant and we set $v_I(r) = 1$. If $I \not\models r$ ($body^+(r) \subseteq I$, $body^-(r) \cap I = \emptyset$, and $head(r) \cap I = \emptyset$), it is also irrelevant and $v_I(r) = 1$. Choice rules contribute to degree 1 in all interpretations. We denote by $P_I[j]$ the set of rules in an LPOD P that contribute to degree j in interpretation I.

Given two interpretations I, J of an LPOD P, $I >^c_P J$ if there is a k such that $|P_I[k]| > |P_J[k]|$, and for all $j < k$, $|P_I[j]| = |P_J[j]|$; $I \geq^c_P J$ if $I >^c_P J$ or $|P_I[k]| = |P_J[k]|$ for all k, $k \geq 1$. An interpretation I is a *(card)-preferred answer set* of an LPOD P if $I \in AS(P)$ and there is no $J \in AS(P)$ such that $J >^c_P I$.

Since od rules participate both in "generating" answer sets and in specifying the preference, LPODs do not have an explicit separated nature. However, LPODs are separable. To show that, we first note that LPODs give rise to a semilattice $\mathcal{L}_{od} = \langle L_{od}, \cup, \emptyset \rangle$, where L_{od} is the set of all LPODs over $\mathcal{U}$, $\cup$ plays the role of $\wedge$ and $\emptyset$ the role of $\top$, and $\sigma_{od} = \langle \mathcal{L}_{od}, 2^{\mathcal{U}}, AS, \geq^c \rangle$ forms the preference framework of LPODs under cardinality-based preferences. The notion of preferred outcomes applied to σ_{od} matches exactly the definition of (card)-preferred answer sets.

To characterize the sub-semilattices L^g_{od} and L^π_{od}, we introduce two more definitions.

An LPOD rule is *purely generating* if it is a choice rule; an od rule such that $body^+(r) \cap body^-(r) \neq \emptyset$; an od rule such that $|head(r)| = 1$; or an od rule such that $head_1(r) \in body^+(r)$.

An LPOD rule is *purely selecting* if it is a choice or od rule such that $body^+(r) \cap body^-(r) \neq \emptyset$; a choice rule such that $head(r) \subseteq body(r)$; or an od rule such that $head_i(r) \in body^+(r)$ and $head_{<i}(r) \subseteq body^-(r)$.

Proposition 9 *For an LPOD P, $P \in L^g_{od}$ iff each rule in P is purely generating, and $P \in L^\pi_{od}$ iff each rule r in P is purely selecting.*

The main result now states that σ_{od} is separable. To prove it, one needs to show that σ_{od} is *g-complete* and π-*complete* by providing appropriate homomorphisms $(\cdot)^g$ and $(\cdot)^\pi$ and show that they satisfy the conditions (1)–(3) of the definition of separability (we omit details due to space constraints).

Theorem 10 *The preference framework σ_{od} is separable.*

The last two results suggest that the formalism of LPODs can be presented in a simplified form as a separated one based on combinations of a generator from L^g_{od} and a selector from L^π_{od}. Combined with the earlier results, they also provide characterizations of strong equivalence for LPODs.

Conclusions

We introduced abstract preference frameworks as a unifying language to study fundamental aspects of preferences that cut across many specific formalisms. We showed the effectiveness of the abstract setting by using it to study the problems of strong equivalence and separability.

Our paper suggests several open problems. First, our conditions for separability are quite restrictive as they require that L^g and L^π be used as generators and selectors, respectively. These sets were defined to contain every element that could possibly be regarded as a generator or a selector, respectively. They may, however, contain redundant elements. Using "non-redundant" sub-semilattices of L^g and L^π may lead to simpler and more natural "separations." Next, many specific preference formalisms explicitly "rank" preferences according to their importance. It is therefore of interest to find an abstract account for ranks in preference frameworks and extend our results to that broader setting. Finally, a comprehensive study of specific preference formalisms from the perspective of abstract preference frameworks (we alluded to some results here and considered in more detail just one, LPODs) is also left for the future.

Acknowledgments

References

Boutilier, C.; Brafman, R.; Domshlak, C.; Hoos, H.; and Poole, D. 2004. Preference-based constrained optimization with CP-nets. *Computational Intelligence* 20:137–157.

Brewka, G., and Eiter, T. 1999. Preferred answer sets for extended logic programs. *Artificial Intelligence* 109(1-2):297–356.

Brewka, G.; Benferhat, S.; and Le Berre, D. 2002. Qualitative choice logic. In Fensel, D.; Giunchiglia, F.; McGuinness, D. L.; and Williams, M.-A., eds., *Proceedings of the 8th International Conference on Principles of Knowledge Representation and Reasoning (KR 2002)*, 158–169. Morgan Kaufmann.

Brewka, G.; Niemelä, I.; and Syrjänen, T. 2004. Logic programs with ordered disjunctions. *Computational Intelligence* 20(2):335–357.

Buccafurri, F.; Leone, N.; and Rullo, P. 2000. Enhancing Disjunctive Datalog by Constraints. *IEEE Transactions on Knowledge and Data Engineering* 12(5):845–860.

Delgrande, J. P.; Schaub, T.; Tompits, H.; and Wang, K. 2004. A classification and survey of preference handling approaches in nonmonotonic reasoning. *Computational Intelligence* 20(2):308–334.

Faber, W., and Konczak, K. 2006. Strong order equivalence. *Annals of Mathematics and Artificial Intelligence* 47(1–2):43–78.

Faber, W.; Tompits, H.; and Woltran, S. 2008. Notions of strong equivalence for logic programs with ordered disjunction. In Brewka, G., and Lang, J., eds., *Proceedings of the 11th International Conference on Principles of Knowledge Representation and Reasoning (KR 2008)*, 433–443. AAAI Press.

Faber, W.; Truszczyński, M.; and Woltran, S. 2012. Strong equivalence of qualitative optimization problems. In Brewka, G.; Eiter, T.; and McIlraith, S. A., eds., *Proceedings of the 13th International Conference onm Principles of Knowledge Representation and Reasoning (KR 2012)*, 188–198. AAAI Press.

Faber, W.; Truszczyński, M.; and Woltran, S. 2013. Abstract preference frameworks — a unifying perspective on separability and strong equivalence. Technical Report DBAI-TR-2013-81, Institute of Information Systems 184/2, Vienna University of Technology, Austria.

Goldsmith, J., and Junker, U., eds. 2008. *Special Issue on Preferences*, volume 29(4) of *AI Magazine*.

Goldsmith, J.; Lang, J.; Truszczyński, M.; and Wilson, N. 2008. The computational complexity of dominance and consistency in CP-nets. *Journal of Artificial Intelligence Research* 33:403–432.

Kaci, S. 2011. *Working with Preferences*. Springer.

Lifschitz, V.; Pearce, D.; and Valverde, A. 2001. Strongly equivalent logic programs. *ACM Transactions on Computational Logic* 2(4):526–541.

Simons, P.; Niemelä, I.; and Soininen, T. 2002. Extending and implementing the stable model semantics. *Artificial Intelligence* 138:181–234.

Simons, P. 1999. Extending the stable model semantics with more expressive rules. In Gelfond, M.; Leone, N.; and Pfeifer, G., eds., *Proceedings of LPNMR'99*, volume 1730 of *Lecture Notes in AI*, 305–316. Springer Verlag.

Turner, H. 2003. Strong equivalence made easy: nested expressions and weight constraints. *Theory and Practice of Logic Programming* 3:609–622.

Wilson, N. 2004. Extending CP-nets with stronger conditional preference statements. In *Proceedings of the 19th National Conference on Artificial Intelligence (AAAI 2004)*, 735–741. AAAI Press.