UNIVERSITÄT AUGSBURG

Preference Constructors for Deeply Personalized Database Queries

Werner Kießling

Report 2004-7 März 2004

INSTITUT FÜR INFORMATIK
D-86135 AUGSBURG

Preference Constructors for Deeply Personalized Database Queries

Werner Kießling

Institute of Computer Science, University of Augsburg
Universitätsstraße 14
D- 86159 Augsburg
Germany
kiessling@informatik.uni-augsburg.de

Abstract

Deep personalization of database queries requires a semantically rich, easy to handle and flexible preference model. Building on preferences as strict partial orders we provide a variety of intuitive and customizable base preference constructors for numerical and categorical data. For complex constructors we introduce the notion of 'substitutable values' (SV-semantics). Preferences with SV-semantics solve major open problems with Pareto and prioritized preferences. Known laws from preference relational algebra remain valid under SV-semantics. These powerful modeling capabilities even contribute to improve efficient preference query evaluation. Moreover, for the first time we point out a semantic-guided way to cope with the infamous flooding effect of query engines. Performing a series of test queries on sample data from an e-procurement application, we provide evidence that the flooding problem comes under control for deeply personalized database queries.

1. Introduction

Personalization of database queries is an increasingly important issue. For instance let's consider e-procurement, which is one of the fastest growing application areas for e-commerce. There are multiple reasons, why today the sales process in e-procurement is still a business with lots of costly human interaction. The misery starts already with the product search. Often B2B customers are forced to manually scroll through huge electronic product catalogs. Frequently commercial search engines simply interpret the customer's search conditions as hard constraints, yielding the embarrassing 'empty result' effect. A failing solution attempt is to interpret the search constraints as 'or'-conditions, causing the 'flooding' effect. Another time-consuming and error-prone approach is parametric search. Offering a plain full-text search is no remedy either, because B2B product search is largely an attribute based search, if e-catalog standards (being mostly XML-based) are in place. Thus state-of-the-art approaches to find products are not enough for the B2B customer ([23]). In fact, a good product search demands a personalized search engine that can handle attribute-rich e-catalog data, that can be personalized to the customer's wishes, roles and situations, and that fully automatically delivers best alternatives when there is no perfect match.

Let's consider the following motivating example from an e-procurement setting.

Example 1 Personalized query composition

Let's assume that an embodied virtual agent called Homer is a notebook reseller. A business woman named Marge contacts him telling her purchase interests:

/1/ "I am interested in notebooks. The CPU speed *must* be at least 2 GHz.

/2/ The order quantity *should* be *around* 40, and *equally important*, the main memory capacity *should* be the *highest* possible."

Homer as a clever salesman maintains a preference repository and thus knows that Marge has long-term preferences, too:

/3/ "Her *favorite* manufacturers are Toshiba and Hewlett Packard, which is *equally important* to her explicit customer preferences."

Naturally, Homer has his own *vendor* preferences:

/4/ "I want to *maximize* my profit margin. But since I am a fair dealer for Marge, all her customer preferences are *more important* to me." ☼

This example emphasizes the need for various components interoperating during the personalization process: ***Personalized query composition*** has to inductively assemble the query from various sources, including explicit customer preferences entered e.g. through the search mask, long-term customer preferences matching the current situation, and current vendor preferences. Long-term preferences should be detectable by automatic *preference mining* algorithms and be managed intelligently in a *preference repository*.

In our example we carefully differentiated between hard constraints ("must") and preferences ("should"). Extending the exact-match query paradigm of database query languages by preferences to specify soft constraints is considered a necessity for successful personalization by many researchers. Research on preferences in databases reaches back quite some time (see e.g. [22], [16], [20]). A powerful framework founded on preferences as strict partial orders has been proposed recently by [13]. This constructor-driven approach was implemented by *Preference SQL* ([19]), its first commercial release being available in 1999, and by *Preference XPath* for XML databases ([18]). Skyline queries ([6]) and numerical ranking ([1]) are special cases of [13]. To date an extensive amount of theoretical results on preferences exists (see e.g. [2], [7]).

The main focus of this paper is concerned with semantic issues of personalized query composition. Since it is well-known that even simple scenarios like above example cannot be modeled by numerical ranking or by skyline queries, more powerful preference frameworks are needed. Moreover, modeling preferences only by raw numbers neglects the importance of an intuitive semantics, which must be a key issue for personalization. Here we extend the constructor-driven approach of strict partial order preferences to pave the way for a paradigm shift towards a more semantic-guided personalization. Let's coin the term "***deep personalization***" for it.

The rest of this paper is organized as follows: We revisit needed concepts in section 2. In section 3 we show how to customize base preference constructors. Section 4 is dedicated to enhancing preferences with more intuitive semantics, called SV-semantics. In section 5 we analyze the impact of this extension on preference query optimization. Section 6 investigates the issue of preference query cardinalities and exhibits striking impacts on the annoying flooding effect. A discussion of related work in section 7 and a summary and outlook in section 8 conclude this work. All proofs are collectively presented in the appendix of this paper.

2. Basic Preference Concepts Revisited

To be self-contained let's revisit needed concepts from [13]. Most preferences appearing in practical database applications can be modeled by *strict partial orders*.

2.1 Preferences

Definition 1 Basic definitions for preferences

Let $A = \{A_1, A_2, \ldots, A_k\}$ be a set of attributes A_j with domains $dom(A_j)$, $1 \le j \le k$. The domain of A is defined as $dom(A) := \times_{A_j \in A}\, dom(A_j)$.

a) A ***preference*** **P** on a set of attributes A is defined as $\mathbf{P = (A, <_P)}$, where $<_P \subseteq dom(A) \times dom(A)$ is a *strict partial order* (i.e. irreflexive and transitive). $x <_P y$ is interpreted as "*I like y better than x*".

b) The ***unordered*** (synonym ***incomparability***) relation $\|_P \subseteq dom(A) \times dom(A)$ is defined as:
$x \|_P y$ iff $\neg(x <_P y) \wedge \neg(y <_P x)$

c) A preference P is a ***chain*** (synonym ***total order***) iff for all $x, y \in dom(A)$, $x \ne y$: $x <_P y \vee y <_P x$

d) A preference P is an ***anti-chain*** iff $<_P = \emptyset$. The anti-chain on an attribute A is denoted as $A^{\leftrightarrow}$.

e) A preference P is a ***weak order***, if negative transitivity holds, i.e. for all $x, y, z \in dom(A)$:
$\neg(x <_P y) \wedge \neg(y <_P z)$ implies $\neg(x <_P z)$

f) The ***maximal*** values of $P = (A, <_P)$ are defined as: $max(P) := \{v \in dom(A) \mid \neg\exists\, w \in dom(A): v <_P w\}$

Note that in general $\|_P$ is reflexive and symmetric, but *not* transitive. If P is a weak order, then $\|_P$ is transitive.

To specify a preference $P = (A, <_P)$ we allow a great flexibility. Any first order predicate on dom(A) can be given for $<_P$, possibly using built-in predicates including equality of values ($=, \ne$) and numeric constraints ($<, \le, >, \ge$). But $<_P$ may also be written in some programming language. For the ease of use we promote a ***constructor-based*** approach, distinguishing between ***base*** preference constructors and ***complex*** preference constructors. Throughout this paper we will use the following notation:

- Creating a base preference constructor:
`base` *bname*(*A, paramlist*) {*definition of* $<_{P\_new}$};
- Defining a base preference P:
P := *bname*(*actual_A, actual_params*);
- Creating a complex preference constructor:
`complex` *Pref*$_1$ *cname Pref*$_2$ {*definition of* $<_{P\_new}$};
- Defining a complex preference P:
P := *actual_Pref*$_1$ *cname actual_ Pref*$_2$;

To illustrate this notation let's repeat some preference constructors presented in [13].

Example 2 Sample use of preference constructors

`base` SCORE(A, f) {$x <_{P\_new} y$ iff $f(x) < f(y)$};
`base` HIGHEST(A) {$x <_{P\_new} y$ iff $x < y$};
`base` AROUND(A, z) {$x <_{P\_new} y$ iff
$abs(x - z) > abs(y - z)$};
`complex` $Pref_1 \otimes Pref_2$ {$(x_1, x_2) <_{P\_new} (y_1, y_2)$ iff
$(x_1 <_{Pref1} y_1 \wedge (x_2 <_{Pref2} y_2 \vee x_2 = y_2)) \vee$
$(x_2 <_{Pref2} y_2 \wedge (x_1 <_{Pref1} y_1 \vee x_1 = y_1))$};

The preferences labeled /2/ in Example 1 can be stated as:

P_1 := AROUND(quantity, 40);

P_2 := HIGHEST(capacity); $P_3 := P_1 \otimes P_2$; ☼

Note that *paramlist* for base constructors is optional, like in HIGHEST. Defining P_1 by AROUND constructs a base preference P_1 = ({quantity}, $<_{P1}$), instantiating the set of attributes A by {quantity} and $<_{P1}$ by $<_{P\_new}$. This single-attribute case is also abbreviated as P_1 = (quantity, $<_{P1}$). Defining P_3 by $\otimes$ constructs a complex preference P_3 = ({quantity, capacity}, $<_{P3}$), instantiating $<_{P3}$ by $<_{P\_new}$ which is defined in terms of $<_{P1}$ and $<_{P2}$.

Definition 2 Preference sub-constructor

C_2 is a *preference sub-constructor* of C_1, if the definition of $<_{C2\text{-new}}$ can be gained from $<_{C1\text{-new}}$ by some specializing constraints.

For instance, HIGHEST is a sub-constructor of SCORE: specialize f(x) := x (see Example 2).

Since sub-constructors are due to specializing constraints, sub-constructor hierarchies are taxonomic. This observation economizes proof efforts: *Properties proved for a constructor are inherited to all its sub-constructors*.

2.2 Preference query languages

In personalized database applications a cooperative query model is needed that supplements the exact-match query of SQL or XPath. Personalized constraints may be hard constraints (in which case the exact-match model is appropriate) or preferences, i.e. *soft constraints*. Whether preferences can be satisfied depends on the current database contents, capturing the status of the real world. Thus a *match-making* between wishes and reality has to be accomplished. To this purpose the ***BMO*** query model *("Best Matches Only")* has been introduced in [13].

Given a schema R(A_1: dom(A_1), …, A_m: dom(A_m)) we consider a preference P = (A, $<_P$), where $A \subseteq \{A_1, \ldots, A_m\}$. For an instance of R let P^R denote the subset preference obtained by restricting P from dom(A) to $\pi_A(R)$, i.e. the currently available A-values in R.

Definition 3 Preference selection, BMO-size

a) ***Preference selection*** **σ[P](R)** is defined as:
$$\sigma[P](R) = \{t \in R \mid t[A] \in \max(P^R)\}$$
b) $t \in \sigma[P](R)$ is a ***perfect match*** iff $t[A] \in \max(P)$.

c) card(σ[P](R)) is called the ***BMO-size*** of σ[P](R).

σ[P](R) retrieves all maximal values from the instance of R. Not all of them are necessarily perfect matches of P. Thus the principle of *query relaxation* is implicit in above definition. Moreover, any non-maximal values of P^R are excluded; hence can be considered as *discarded on the fly*. Thus only best matching tuples are retrieved.

In many personalized e-commerce applications σ[P](R) serves as an *intelligent pre-selection*, which is the basis for further sales negotiations. Therefore it is often important that BMO-sizes come in "handy portions".

Example 3 A preference query and its BMO result

Continuing Example 2 we evaluate σ[P_3](Sales), given this small sales relation:

Sales(quantity, capacity, notebook) =
{(45, 768, 1), (20, 1024, 2), (30, 1024, 3), (45, 512, 4),

Notebook 1 is better than 4, because it has the same quantity but a better capacity; 3 is better than 2, but 1 and 3 are incomparable. Thus we get a result with BMO-size 2:

σ[P_3](Sales) = {(45, 768, 1), (30, 1024, 3)} ☼

Note that σ[P] has been implemented in Preference SQL (as PREFERING-clause) and in Preference XPath. For brevity we will stick to the algebraic notation σ[P].

3. Customization of Base Constructors

In [13] several intuitive preference constructors were presented, which have proven their usefulness in e-commerce ([19]). We will extend this repertoire in several respects.

3.1 The base preference constructor $SCORE_d$

When dealing with numerical scores, it is a common practice to group ranges of scores together; e.g., forming grades at school, or offering differential price discounts in e-procurement, or setting target deadlines for events to happen (e.g. "payment due within two weeks"), etc. Now we show how to model such real-world practice on top of a given preference constructor.

Definition 4 $SCORE_d$

Given a utility function f: dom(A) $\to \mathbb{R}$ and some $d \in \mathbb{R}_0^+$, we define for all v ∈ dom(A):

f_d: dom(A) $\to$ {if d = 0 then $\mathbb{R}$ else $\mathbb{Z}$}, where

$f_d(v)$:= {if d = 0 then f(v) else $\lceil f(v) / d \rceil$}

`base` $SCORE_d$(A, f) {x $<_{P\_new}$ y iff $f_d(x) < f_d(y)$};

Each preference constructed by $SCORE_d$ is constructible by SCORE and vice versa. Thus due to [9] $SCORE_d$ constructs a *weak order*. Choosing d > 0 effects that values with identical f_d-values become unordered:

x $\|_{P\text{-new}}$ y iff $f_d(x) = f_d(y)$

In this way a categorical view on numerical scores is achieved. As we will see later on, certain unordered values can be interpreted as 'substitutable' or 'equally good'.

Now we present several sub-constructors of $SCORE_d$, focusing on preferences P = (A, $<_P$), where A is a *single* attribute with a *numerical* domain, i.e. dom(A) $\subseteq \mathbb{R}$.

3.2 Sub-constructors of $SCORE_d$

Given v, low, up ∈ dom(A) we define the *distance* of v from the closed interval [low, up] as follows:

$\text{dist}[\text{low, up}]: \text{dom}(A) \rightarrow \mathbb{R}_0^+$
$\text{dist}[\text{low, up}](v) := \{\text{if } v \in [\text{low, up}] \text{ then } 0 \text{ else if } v < \text{low then low} - v \text{ else } v - \text{up}\}$

Given $d \in \mathbb{R}_0^+$ we group distances together as follows:

$\text{dist}_d[\text{low, up}]: \text{dom}(A) \rightarrow \{\text{if } d = 0 \text{ then } \mathbb{R}_0^+ \text{ else } \mathbb{N}_0\}$
$\text{dist}_d[\text{low, up}](v) := \{\text{if } d = 0 \text{ then dist}[\text{low, up}](v) \text{ else } \lceil \text{dist}[\text{low, up}](v) / d \rceil\}$

Definition 5 BETWEEN$_d$

`base` BETWEEN$_d$(A, [low, up]) $\{x <_{P\_new} y$ iff $\text{dist}_d[\text{low, up}](y) < \text{dist}_d[\text{low, up}](x)\}$;

For $d > 0$ a BETWEEN$_d$ preference can be envisioned as a one-dimensional *dart board*: Perfect matches hit the interval [low, up] at dist_d being 0, second bests are those with dist_d being 1, and so on. Values with identical dist_d-values become unordered.

Special cases of this construction are obtained by identifying low = up =: z (setting $\text{dist}_d[z, z] =: \text{dist}_d[z]$) and by choosing z as the *finite* infimum or supremum of dom(A).

Definition 6 AROUND$_d$, LOWEST$_d$, HIGHEST$_d$

a) `base` AROUND$_d$(A, z) $\{x <_{P\_new} y$ iff $\text{dist}_d[z](y) < \text{dist}_d[z](x)\}$;
b) `base` LOWEST$_d$(A) $\{x <_{P\_new} y$ iff $\text{dist}_d[\inf_A](y) < \text{dist}_d[\inf_A](x)\}$;
c) `base` HIGHEST$_d$(A) $\{x <_{P\_new} y$ iff $\text{dist}_d[\sup_A](y) < \text{dist}_d[\sup_A](x)\}$;

Example 4 AROUND$_d$, LOWEST$_d$

Let dom(age) = $[6, 20] \subseteq \mathbb{R}$, hence $\inf_A = 6$. Further let d = 2 and R(age) = {7, 8, 11, 13}.

- Let P := AROUND$_2$(age, 10): Since $\text{dist}_2[10](v) = 1$ if $v \in \{8, 11\}$ and $\text{dist}_2[10](v) = 2$ if $v \in \{7, 13\}$, we get $\sigma[P](R) = \{8, 11\}$.
- Let P := LOWEST$_2$(age): Since $\text{dist}_2[6](7) = 1 = \text{dist}_2[6](8)$, $\text{dist}_2[6](11) = 3$ and $\text{dist}_2[6](13) = 4$, we get $\sigma[P](R) = \{7, 8\}$. ☼

It should be emphasized that above parameter d defines a symmetrical distance from the perfect hit. If, however, the application semantics suggests an *unsymmetrical* treatment, d has to be replaced properly by two parameters d_1 and d_2.

Now we study ***categorical*** data, not requiring any numerical operations for defining the preference order.

Definition 7 LAYERED$_m$

Let $L = (L_1, \ldots, L_{m+1})$, $m \geq 0$, be an ordered list of sets with the following properties:

- L is a partition of dom(A).
- Exactly m out of these m+1 sets are given as finite enumerations of values from dom(A).
- The remaining set is specified as '*other values*'.

We define a function layer: $\text{dom}(A) \rightarrow \mathbb{N}$ as follows:
for $i \in \{1, \ldots, m + 1\}$, for all $v \in L_i$: layer(v) := i.
`base` LAYERED$_m$(A, L) $\{x <_{P\_new} y$ iff layer(y) < layer(x)$\}$;

LAYERED$_m$ is a sub-constructor of SCORE$_d$, specializing d = 0 and f(v) = layer(v). Relating LAYERED$_m$ to other constructors on categorical attributes as defined in [13] yields:

Proposition 1 Sub-constructors of LAYERED$_m$

a) POS/POS is a sub-constructor of LAYERED$_m$: m = 2, L = (POS$_1$-set, POS$_2$-set, 'other values')
b) POS/NEG is a sub-constructor of LAYERED$_m$: m = 2, L = (POS-set, 'other values', NEG-set)
c) POS is a sub-constructor of LAYERED$_m$: m = 1, L = (POS-set, 'other values')
d) NEG is a sub-constructor of LAYERED$_m$: m = 1, L = ('other values', NEG-set)
e) ANTI-CHAIN is a sub-constructor of LAYERED$_m$: m = 0, L = ('other values')

The categorical view of numerical data by BETWEEN$_d$ for $d > 0$ is reflected by this relationship to LAYERED$_m$: Defining layer(v) := $\text{dist}_d[\text{low, up}](v) + 1$ for all $v \in \text{dom}(A)$, then BETWEEN$_d$ maps values onto $m = \max\{\text{dist}_d[\text{low, up}](\inf_A), \text{dist}_d[\text{low, up}](\sup_A)\}$ layers.

3.3 Extensibility of base preference constructors

Our repertoire can be extended as required by the application semantics. For illustration we present three examples.

Definition 8 SuperSCORE$_d$

Given a utility function f: $\text{dom}(A) \rightarrow \mathbb{R}$ and some $e \in \mathbb{R}_0^+$, we define for all $v \in \text{dom}(A)$:
`base` Super$_e$SCORE(A, f) $\{x <_{P\_new} y$ iff $f(x) < f(y) - e\}$;

Super$_e$SCORE constructs a strict partial order, but no weak order for $e > 0$. SCORE$_0$ is a sub-constructor of Super$_e$SCORE, but not vice versa. Values are unordered if their f-values differ at most by e, i.e. if they are 'close together': $x \parallel_{P\text{-new}} y$ iff $\text{abs}(f(x) - f(y)) \leq e$

These results plus relationships, which can be proved for our numerical base constructors and the EXPLICIT constructor from [13], are visualized in Figure 1.

All sub-constructors presented so far are defined on a single attribute. Now we give an example of a multi-attribute base constructor. (In [7] it was claimed this one cannot be expressed in our framework.)

Example 5 Base constructor on multiple attributes

`base` Cho-Ex10.1($\{A_1, A_2\}$)$\{(x_1', x_2') <_{P\_new} (x_1, x_2)$ iff $x_1 = x_2 \wedge x_1' \neq x_2'\}$

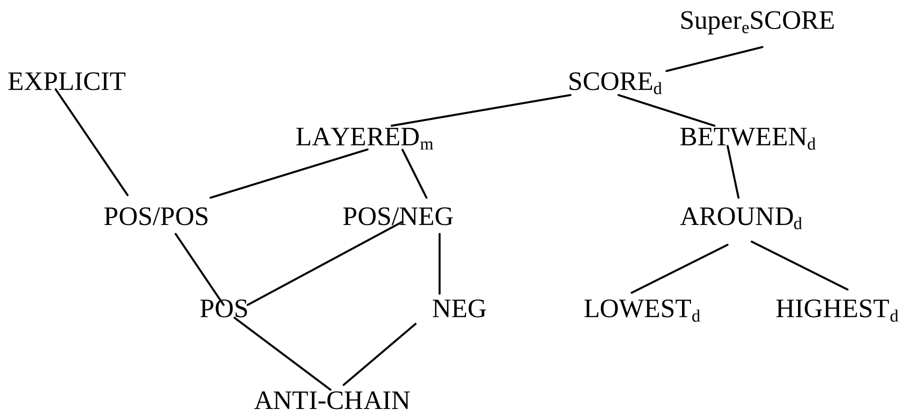


Figure 1: Extensible base preference sub-constructor hierarchy.

Next we demonstrate how to build new *custom-tailored* constructors. As pointed out in [13], the linear sum constructor '$\oplus$' can be viewed as a convenient design method for sophisticated base constructors.

Example 6 Custom-designed base constructor

Assume we need a "MyAROUND" constructor like this:

A perfect match is the value z. All values v, where v < z, are better than all values w, where w > z. Different parameters d_1 and d_2 should be available for v and w, respectively.

Given an attribute A with $\text{dom}(A) \subseteq \mathbb{R}$ and $z \in \text{dom}(A)$, let A_1 and A_2 denote attributes where

- $\text{dom}(A_1) := \{v \in \text{dom}(A) \mid v \le z\}$, hence $\sup_{A1} = z$,
- $\text{dom}(A_2) := \{v \in \text{dom}(A) \mid v > z\}$, hence $\inf_{A2} = z$.

Then $P_1 := \text{HIGHEST}_{d1}(A_1)$ and $P_2 := \text{LOWEST}_{d2}(A_2)$ are preferences. Since $\text{dom}(A_1) \cap \text{dom}(A_2) = \varnothing$, $P := P_1 \oplus P_2$ is a preference with the desired properties.

`base` $\text{MyAROUND}_{d1,\, d2}(A, z)$
$\{\text{HIGHEST}_{d1}(A_1) \oplus \text{LOWEST}_{d2}(A_2)\}$;
(e.g.) $P := \text{MyAROUND}_{5,\,2}(\text{quantity}, 40)$ ☼

Thus the choice of base constructors can be customized as desired to deeply personalize database queries. Now we turn our attention to complex constructors.

4. Preferences with SV-Semantics

A distinctive feature of strict partial orders is that unordered values may exist. Often people have a strong opinion about better-than relationships for a selected choice of options, but without being complete: For some values they don't mind or some values may be equally good for them in a given scenario, etc. Thus the freedom of not having to specify a total order is not a bug, but rather a valuable asset for real world modeling.

4.1 SV-relations

Our interest concentrates on a *debatable* effect caused by unordered values. To this end we will study the Pareto constructor '$\otimes$' as stated in Example 2 more closely.

Example 7 The impact of unordered values

Given $\text{dom}(A_i) = [-10, 10] \subseteq \mathbb{R}$ for $i \in \{1, 2, 3\}$, we consider $P_1 := \text{AROUND}_0(A_1, 0)$, $P_2 := \text{LOWEST}_0(A_2)$, $P_3 := \text{HIGHEST}_0(A_3)$, $P_4 := P_1 \otimes P_2$ and $P_5 := P_3 \otimes P_4$.
For $v = (-5, 3, 4)$ and $w = (5, 1, 8) \in \times\, \text{dom}(A_i)$ we get:

- $-5 \parallel_{P1} 5$, $3 <_{P2} 1$, $4 <_{P3} 8$
- $(-5, 3) \parallel_{P4} (5, 1)$, $v \parallel_{P5} w$

But there are situations where some unordered values should be treated as ***substitutable*** (synonym: ***equally good***): For P_1 we might be indifferent between a mismatch of +5 or –5 from the perfect match 0. If so, then it is reasonable to re-assess the relationship of v with w in P_4. The *intuitive feeling* is that we would rather expect:

$(-5, 3) <_{P4} (5, 1)$

This is because w and v, though not equal, are substitutable wrt. P_1 and w is better than v wrt. P_2. This in turn would change the rating for v and w in P_5: $v <_{P5} w$ ☼

The challenge now becomes to find out, how the definition of '$\otimes$' must be adapted to capture the *intuitive semantics* of substitutability:

a) Values x and y can only be substitutable, if both are unordered.
b) If x is better than z and x can be substituted by y, then y should be better than z as well.
c) Dually, if z is better than x and x can be substituted by y, then z should be better than y as well.
d) Substitutability should be an equivalence relation.

Thus not all unordered values need to be substitutable! E.g., point d) is no consequence from a), since $\parallel_P$ is not transitive in general. This is the very reason, why simply replacing '$x_1 = y_1$' by '$x_1 \parallel_{P1} y_1$' and '$x_2 = y_2$' by '$x_2 \parallel_{P2} y_2$' in the definition of '$\otimes$' is semantically not justified in general. Technically this flaw impacts that would '$\otimes$' violate the strict partial order property ([7]).

Definition 9 SV-relation

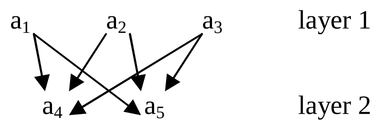
Given $P = (A, <_P)$, $\cong_P$ is called ***substitutable values relation*** (***SV-relation*** for short) iff for all $x, y \in dom(A)$:

a) $x \cong_P y$ implies $x \parallel_P y$
b) $x \cong_P y \wedge \exists z : z <_P x$ implies $z <_P y$
c) $x \cong_P y \wedge \exists z : x <_P z$ implies $y <_P z$
d) $\cong_P$ is reflexive, symmetric and transitive.

Unordered values that are not substitutable are called ***alternative*** values.

Example 8 SV-relations

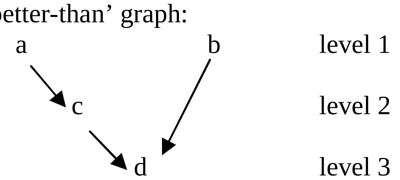
Case study 1: P := POS(A, $\{a_1, a_2, a_3\}$), where dom(A) = $\{a_1, a_2, a_3, a_4, a_5\}$, has this 'better-than' graph (see [13]):



Given $x, y \in dom(A)$, SV-relations are e.g.:

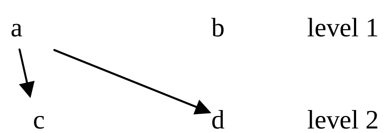
- $x \cong_P y$ iff layer(x) = layer(y) // *regular case*
- $x \cong_P y$ iff $x = y \vee x, y \in \{a_1, a_2\} \vee x, y \in \{a_4, a_5\}$
- $x \cong_P y$ iff $x = y$ // *trivial case*

Case study 2: Consider an EXPLICIT preference P with this 'better-than' graph:



There is only the trivial SV-relation: $x \cong_P y$ iff $x = y$
Because $c <_P a$ and $\neg(c <_P b)$, a and b are *alternatives*. Also, c and b are *alternatives*, since $c <_P a$ but $\neg(a <_P b)$.

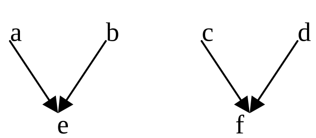
Case study 3: Consider an EXPLICIT preference P with this 'better-than' graph:



- $x \cong_P y$ iff $x = y \vee x, y \in \{c, d\}$

This is a non-trivial SV-relation. Note that negative transitivity is violated here; hence P is no weak order.

Case study 4: Consider P with this 'better-than' graph:



- $x \cong_P y$ iff $x = y \vee x, y \in \{a, b\} \vee x, y \in \{c, d\}$

Negative transitivity is violated, but $\cong_P$ is non-trivial. ☼

Proposition 2 Properties of SV-relations

a) '=' is an SV-relation for each preference P (called **trivial** SV-relation).
b) If P is the anti-chain $A^{\leftrightarrow}$, each partition of dom(A) defines an SV-relation.

In Example 8 we have seen cases, where P is not a $SCORE_d$ preference, but has a non-trivial SV-relation. Also we have seen instances of alternative values, which are not substitutable (not equally good). For $SCORE_d$ it turns out that the full $\parallel_P$-relation is a valid SV-relation.

Proposition 3 Regular SV-relation for $SCORE_d$

Given a $SCORE_d$ preference P, let's define for all $x, y \in dom(A)$: $x \cong_P y$ iff $x \parallel_P y$

a) $\cong_P$ is an SV-relation (called **regular** SV-relation).
b) If P is not a chain, then $\cong_P$ is non-trivial.

Referring back to the discussion on the categorical view on numerical data imposed by $SCORE_d$ for $d > 0$, this result says that *all* values with equal f_d-values may be 'substitutable' or 'equally good'. However, this nice behavior does not hold for other constructors like EXPLICIT or $Super_e$SCORE, which are no weak orders.

4.2 Enriching the preference definition

From now on we will enrich our definition of preferences to accommodate the semantics of SV-relations.

Definition 10 Preferences with SV-semantics

Enriching Definition 1 a) we use the following notation:

- A preference P with an SV-relation $\cong_P$ is denoted as:
  P = (A, $<_P$, $\cong_P$).
- Each base constructor receives one additional parameter for the SV-relation.

Let's explore the impact of SV-relations for *inductive* preference construction. These aspects are important:

- Consider a base preference $P_i = (A_i, <_{Pi}, \cong_{Pi})$. Then $\cong_{Pi}$ ***does not*** affect $<_{Pi}$ itself, but expresses that it is admissible to substitute $\cong_{Pi}$-values for each other.
- A complex constructor C, using P_i in its definition for $<_{P\_new}$, can make use of $\cong_{Pi}$. This ***does*** affect $<_{P\_new}$! Moreover, if an SV-relation $\cong_{P\_new}$ can be determined for P, then C can inductively be used itself as input for complex preferences.

4.3 Complex constructors with SV-semantics

We present in detail, how Pareto and prioritized construction can be enriched by SV-semantics. A Pareto preference combines *equally important* preferences P_1 and P_2, whereas the combination of P_1 with a *less important* P_2 is modeled by a prioritized preference.

Definition 11 Pareto and prioritized constructors

We assume $P_1 = (A_1, <_{P1}, \cong_{P1})$ and $P_2 = (A_2, <_{P2}, \cong_{P2})$.

a) **Pareto constructor '⊗'**

- A_1 and A_2 don't overlap:

  `complex` $P_1 \otimes P_2$ $\{(x_1, x_2) <_{P\_new} (y_1, y_2)$ iff
  $(x_1 <_{P1} y_1 \wedge (x_2 <_{P2} y_2 \vee x_2 \cong_{P2} y_2)) \vee$
  $(x_2 <_{P2} y_2 \wedge (x_1 <_{P1} y_1 \vee x_1 \cong_{P1} y_1))$;
  $(x_1, x_2) \cong_{P\_new} (y_1, y_2)$ iff $x_1 \cong_{P1} y_1 \wedge x_2 \cong_{P2} y_2\}$;
- Otherwise: Identify overlapping attributes above.

b) **Prioritized constructor '&'**

- A_1 and A_2 don't overlap:

  `complex` P_1 & P_2 $\{(x_1, x_2) <_{P\_new} (y_1, y_2)$ iff
  $x_1 <_{P1} y_1 \vee (x_1 \cong_{P1} y_1 \wedge x_2 <_{P2} y_2)$;
  $(x_1, x_2) \cong_{P\_new} (y_1, y_2)$ iff $x_1 \cong_{P1} y_1 \wedge x_2 \cong_{P2} y_2\}$;
- Otherwise: Identify overlapping attributes above.

Note that $\cong_{P\_new}$ is defined using $\cong_{P1}$ and $\cong_{P2}$.

Example 9 Pareto preferences with SV-semantics

Let's revisit Example 7 introducing SV-relations, e.g.:

$P_1 := \text{AROUND}_0(A_1, 0, \cong_{P1})$ where $\cong_{P1}$ is regular
$P_2 := \text{LOWEST}_0(A_2, \text{'='})$, $P_3 := \text{HIGHEST}_0(A_3, \text{'='})$
$P_4 := P_1 \otimes P_2$, $P_5 := P_3 \otimes P_4$

For $v = (-5, 3, 4)$ and $w = (5, 1, 8)$ we now can state:

- Since $\cong_{P1}$ does not change $<_{P1}$ we get as before:
  $-5 \parallel_{P1} 5$, $3 <_{P2} 1$, $4 <_{P3} 8$
- Since $-5 \cong_{P1} 5$ we now get: $(-5, 3) <_{P4} (5, 1)$
- Since $4 <_{P3} 8$, $(-5, 3) <_{P4} (5, 1)$ we now get: $v <_{P5} w$

This is our intuitively expected result, provided that $\cong_{P1}$ holds in the given application situation. ☼

Example 10 Prioritization with SV-semantics

Assuming P_1 := POS(category, {luxury, sport}, '=') and P_2 := POS(color, {red}, '='), let's consider $P := P_1$ & P_2. Given R = {(luxury, black), (sport, green), (sport, red)}, we get the following results:

- Since sport = sport and green $<_{P2}$ red:
  (sport, green) $<_P$ (sport, red)
- Since luxury ≠ sport: (luxury, black) $\parallel_P$ (sport, red)
  (luxury, black) $\parallel_P$ (sport, green)

Thus $\sigma[P](R)$ = {(sport, red), (luxury, black)}.

Now suppose that 'luxury' and 'sport' shall be substitutable: P_1 := POS(Category, {luxury, sport}, $\cong_{P1}$) where $\cong_{P1}$ is regular. This changes the picture:

- As before: (sport, green) $<_P$ (sport, red)
- Since luxury $\cong_{P1}$ sport we get:
  black $<_{P2}$ red implies (luxury, black) $<_P$ (sport, red)
  black $\parallel_{P2}$ green implies (luxury, black) $\parallel_P$ (sport, green)

Therefore $\sigma[P](R)$ = {(sport, red)}, offering less alternatives than before. ☼

The following main theorem is the result of our semantically well-founded approach.

Theorem 1 Preservation of strict partial order

Given $P_1 = (A_1, <_{P1}, \cong_{P1})$ and $P_2 = (A_2, <_{P2}, \cong_{P2})$, consider $P := P_1 \otimes P_2$ and $P := P_1$ & P_2, respectively. Then $P = (A_1 \cup A_2, <_P, \cong_P)$ is a preference with SV-semantics, i.e.:

a) $<_P$ is a strict partial order on $A_1 \cup A_2$.
b) $\cong_P$ is an SV-relation for $<_P$.

Please refer to Definition 1 for dealing with overlapping A_1 and A_2; $<_P$ and $\cong_P$ are due to Definition 11a, b.

Consequently *inductive* preference construction, being essential for personalized query composition, preserves strict partial order, too! Moreover, obeying to SV-semantics is the most general approach to preserve strict partial order for '⊗' and '&'.

Theorem 2 Further properties of '⊗' and '&'

a) Pareto or prioritized preferences don't possess regular SV-relations in general.
b) Any relaxation of SV-semantics for Pareto or prioritized construction violates strict partial order.

In [13] we reported an interesting correlation between prioritization and grouping for trivial SV-relations. Now we extend this observation to arbitrary SV-relations.

Definition 12 Grouped preference

Given $P = (B, <_P, \cong_P)$ and the anti-chain $A^{\leftrightarrow} = (A, \emptyset, \cong_A)$, $\mathbf{A^{\leftrightarrow}}$ **& P** is called a ***grouped preference***. As a synonym we also write: **P groupby A**

Due to Proposition 2b, $\cong_A$ can be any partition of dom(A). If $\cong_{\mathbf{A}}$ is trivial, then grouping is done wrt equal A-values. Otherwise a sophisticated grouping effect is achieved by building groups wrt *substitutable* A-values.

Grouped preferences appear in skyline queries ([6]). If DIFF attributes are given, then a skyline preference can be characterized as $P := (P_1 \otimes \ldots \otimes P_n)$ groupby DIFF, where each P_i is a LOWEST_0 or HIGHEST_0 preference. Due to Figure 1 and Theorem 1 we can extend the class of skyline preferences without violating strict partial orders.

Definition 13 Skyline preference with SV-semantics

A ***skyline preference*** P is defined as

$$P := (P_1 \otimes \ldots \otimes P_n) \text{ groupby DIFF}$$

where $P_i := \text{HIGHEST}_{di}(A_i, \cong_{Pi})$ or $P_i := \text{LOWEST}_{di}(A_i, \cong_{Pi})$, $i \in \{1, \ldots, n\}$, and $\text{DIFF}^{\leftrightarrow} = (\text{DIFF}, \emptyset, \cong_{\mathbf{DIFF}})$.

A *numerical preference* is a weighted combination of $SCORE_{di}$ preferences $P_1, \ldots, P_n$, applying a combining function F. In the Appendix we show how to adapt the '$rank_F$' constructor to SV-semantics. There it is also shown, how the complex constructors *intersection* ('♦'), *disjoint union* ('+') and *linear sum* ('⊕') presented in [13] can be enriched by SV-semantics.

4.4 Expressiveness Results

The next result suggests that for deep personalization a rich repertoire of complex constructors with SV-semantics should be supported.

Theorem 3 Expressiveness of complex constructors

a) Pareto is no sub-constructor of $rank_F$ and vice versa. (Skyline preferences cannot be expressed by $rank_F$.)
b) Pareto is no sub-constructor of '&' and vice versa.
c) '&' is no sub-constructor of '$rank_F$' and vice versa. (Grouped preferences are not expressible by $rank_F$.)

Thus personalization that solely relies on numerical ranking is of rather limited expressiveness.

Example 11 Deeply personalized query

Let's get back to our motivating Example 1. Personalized preference composition has to inductively construct a complex preference P from the statements labeled /2/ (customer preferences), /3/ (long-term preferences from the repository) and /4/ (vendor preferences). Using '⊗' to model equal importance and '&' for ordered importance we can state: $P := (P_{customer} \otimes P_{repository}) \,\&\, P_{vendor}$

Our sales story leaves open the issues of how to categorize numerical data (i.e. choices of d-parameters) and of substitutability (i.e. choices of SV-relations). This knowledge can be gained in manifold ways, e.g. by interviewing the customer, from personalized long-term knowledge in the preference repository, from defaults, etc. For our sales story we assume this scenario:

- Marge lets Homer know that a deviation up to +3 or –3from the stated quantity don't really worry her.
- From the preference repository it is known that Marge doesn't mind capacity differences up to 256 Mbytes and that Toshiba and HP are equally good manufacturers for her that can be substituted.

Then we can complete the definition of P e.g. as follows:

$P_{customer}$:= $AROUND_3$(quantity, 40, 'regular') ⊗ $HIGHEST_{256}$(capacity, 'regular');
$P_{repository}$:= POS(make, {'Toshiba', 'HP'}, 'regular');
P_{vendor} := $HIGHEST_0$(profit_margin, 'regular');

Using Preference XPath syntax our entire sales scenario, including the hard customer constraint labeled /1/, can be expressed declaratively by one query statement:

```
/Notebook
   [CPU_speed >= 2.0]
  #[(quantity around (3, 40, 'reg') and
     capacity highest(256, 'reg')   and
     make in(('Toshiba','HP'), 'reg'))
     prior_to profit_margin
              highest(0, 'reg')]#
```

Note that hard constraints are scoped by "[…]", preferences by "#[…]#"; Pareto is 'and', prioritization is 'prior to', POS is 'in'. ☼

5. Query Optimization Issues

After we have discussed our semantic intuition about substitutable values let's explore its implications for the optimization of deeply personalized preference queries.

5.1 Preference Algebra

In [13] we have identified many algebraic laws amongst preferences with trivial SV-relations. The subsequent main theorem is the key that these laws carry over to non-trivial SV-relations.

Definition 14 SV-order

Given $P = (A, <_{P1}, \cong_P)$, let $A/\cong_P$ denote the set of equivalences classes of dom(A) over $\cong_P$. For all $X, Y \in A/\cong_P$ we define:

- $X <_{[P]} Y$ iff $\forall x \in X, \forall y \in Y: x <_P y$
- $X \cong_{[P]} Y$ iff $\forall x \in X, \forall y \in Y: x \cong_P y$

Then $[P] = (A/\cong_P, <_{[P]})$ is called ***SV-order***.

Theorem 4 Every SV-order is a strict partial order

Moreover, $\cong_{[P]}$ is the trivial SV-relation, i.e. equality of equivalence classes.

Consequently all preference algebra laws given in [13] hold for any SV-order, characterizing preferences with non-trivial SV-relations. For more details see the Apenix.

5.2 Preference Relational Algebra

Declarative query languages with hard constraints like SQL or XPath can be seamlessly extended by preference selection towards Preference SQL or Preference XPath, supporting the BMO query model. Implementing such preference query languages can be accomplished by loose coupling and query rewriting ([19]). For higher performance tight coupling is required, integrating the preference selection operator directly into the database kernel and extending relational algebra towards a *preference relational algebra*.

To date many transformation laws for preference relational algebra are known ([17], [7], [10]). For illustration, here are two laws, given a preference $P = (A, <_P)$:

	d = 0	d = 5	d = 9	d = 15	d = 25	d = 30	d = 50
$BETWEEN_d$	1	20	20	27	39	56	97
$AROUND_d$	3	3	3	26	41	49	74
$LOWEST_d$	4	6	4	6	9	6	9

Figure 2: Sample BMO-sizes for base preferences with varying parameter d.

- *Push preference over Cartesian product*:
  If $A \subseteq attr(R)$, then $\sigma[P](R \times S) = \sigma[P](R) \times S$
- *Push preference over union*: If $A \subseteq attr(R)$, then
  $\sigma[P](R \cup S) = \sigma[P](\sigma[P](R) \cup \sigma[P](S))$

The proofs for such laws sometimes critically rely on the transitivity of the given preference order. Since according to Theorem 4 strict partial order property is preserved for arbitrary SV-relations, *all such transformation laws carry over to preferences with SV-semantics*.

6. Observations on BMO-sizes

The BMO query model avoids the notorious 'empty result' effect and reduces the 'flooding' effect. But the latter can exhibit two opposite effects from a personalization point of view: The BMO-size is too large, i.e. fewer alternatives would be preferable. Or, the BMO-size is too small, i.e. more alternatives would be preferable.

Our constructor-based approach offers two semantically guided opportunities to influence the BMO-sizes of preferences queries: choosing the d-parameters of base constructors and choosing the SV-relations. We will investigate in more detail both options subsequently.

6.1 BMO-sizes for base preferences

Let's start with a basic property of $SCORE_d$ for varying d. At first guess one might conjecture that $d_1 \leq d_2$ implies $\sigma[SCORE_{d1}(A, f)](R) \subseteq \sigma[SCORE_{d2}(A, f)](R)$, however:

Proposition 4 BMO-sizes of $SCORE_d$ are non-monotonic in d.

Thus the BMO-size cannot be influenced deterministically by d; it also depends on the data distribution. But statistically speaking, it is reasonable to assume the following ***rule of thumb***: *Choosing a larger d tends to increase the BMO-size of $\sigma[SCORE_d](R)$.*

Note that the choice of d only impacts BMO-sizes, if there are *no* perfect matches. The BMO-sizes observed in the next example nicely demonstrate this non-monotonic behavior and support our rule of thumb stated as well.

Example 12 BMO-sizes for varying parameter d

We used a data set taken from a real-life application. The COSIMA[B2B] prototype ([14]), being a sophisticated sales agent for e-procurement portals, works with an XML-based electronic product catalog for storage and transport boxes and waste containers. There are attributes on numerical domains (like length, height, width, weight) and on categorical domains (like color, type of material) as well. This sample catalog comprises about 1000 such sales objects. Using Preference XPath we executed a series of test queries for different settings of d. Characteristic patterns of observed BMO-sizes for base preference queries on numerical domains are given in Figure 2. ☼

6.2 BMO-sizes for complex preferences

Choosing the right SV-relations is an important factor in influencing BMO-sizes for complex constructors, in particular for Pareto and prioritized preferences. The following classification of SV-relations is important.

Definition 15 More liberal than ($\cong_2 \succcurlyeq_P \cong_1$)

Given SV-relations $\cong_1$ and $\cong_2$ for a preference P, $\cong_2$ is ***more liberal than*** $\cong_1$ ($\cong_2 \succcurlyeq_P \cong_1$) if:

$$\cong_2 \succcurlyeq_P \cong_1 \text{ iff } (\forall x, y \in dom(A): x \cong_1 y \text{ implies } x \cong_2 y)$$

If $\cong_2 \succcurlyeq_P \cong_1$, then each substitutable value of $\cong_1$ is also substitutable in $\cong_2$, but $\cong_2$ may have additional ones, which is regarded as a more liberal behavior.

Proposition 5 Properties of $\succcurlyeq_P$

a) $\succcurlyeq_P$ is a non-strict partial order on the set of all SV-relations of a preference P.
b) If P is constructed by $SCORE_d$, then the regular (trivial) SV-relation is the greatest (smallest) element of $\succcurlyeq_P$.

Example 13 Liberality of SV-relations

Given $dom(A) = \{a_1, a_2, a_3, a_4, b_1, b_2, b_3\}$, for $P := POS(A, \{a_1, a_2, a_3, a_4\}, \cong_P)$ we know that:

- $layer(x) = 1$ iff $x \in \{a_1, a_2, a_3, a_4\}$.
- $layer(x) = 2$ iff $x \in \{b_1, b_2, b_3\}$.

Then choices for $\cong_P$ are e.g.:

- $x \cong_1 y$ iff $layer(x) = layer(y)$ // *regular case*
- $x \cong_2 y$ iff $x, y \in \{a_1, a_2\} \vee x, y \in \{a_3, a_4\} \vee x, y \in \{b_1, b_2, b_3\}$
- $x \cong_3 y$ iff $x, y \in \{a_1, a_3\} \vee x, y \in \{a_2, a_4\} \vee x, y \in \{b_1, b_2, b_3\}$
- $x \cong_4 y$ iff $x = y$ // *trivial case*

We get: $\cong_1 \succcurlyeq_P \cong_2$, $\cong_1 \succcurlyeq_P \cong_3$, $\cong_2 \succcurlyeq_P \cong_4$, $\cong_3 \succcurlyeq_P \cong_4$ ☼

Q_α	rel_d = 0 %	rel_d = 5 %	rel_d = 10 %	rel_d =15 %	rel_d = 20 %	rel_d = 30 %
trivial SVs	8	10	10	28	51	116
regular SVs	8	2	2	2	1	1

Q_β	rel_d = 0 %	rel_d = 5 %	rel_d = 10 %	rel_d =15 %	rel_d = 20 %	rel_d = 30 %
trivial SVs	4	4	15	33	51	101
regular SVs	4	4	4	5	12	19

Q_γ	rel_d = 0 %	rel_d = 5 %	rel_d = 10 %	rel_d =15 %	rel_d = 20 %	rel_d = 30 %
trivial SVs	48	48	62	67	72	104
regular SVs	8	8	10	14	14	24

Q_δ	rel_d = 0 %	rel_d = 5 %	rel_d = 10 %	rel_d =15 %	rel_d = 20 %	rel_d = 30 %
trivial SVs	388	476	519	502	545	519
regular SVs	84	44	35	31	23	5

Figure 3: Sample BMO-sizes for complex preferences with varying SV-relations.

The following main theorem supports the common experience that accepting more things as substitutable typically ends up in having fewer alternative choices left.

Theorem 5 Monotonicity of BMO-sizes for ⊗, &

Let $P_1 = (A_1, <_{P1}, \cong_1)$, $P_1{}^* = (A_1, <_{P1}, \cong_{1^*})$, differing only wrt the SV-relation, likewise $P_2 = (A_2, <_{P2}, \cong_2)$, $P_2{}^* = (A_2, <_{P2}, \cong_{2^*})$.

a) $\sigma[P_1 \otimes P_2](R) \subseteq \sigma[P_1{}^* \otimes P_2{}^*](R)$

if $\cong_1 \succcurlyeq_{P1} \cong_{1^*}$ and $\cong_2 \succcurlyeq_{P2} \cong_{2^*}$

b) $\sigma[P_1 \& P_2](R) \subseteq \sigma[P_1{}^* \& P_2](R)$ if $\cong_1 \succcurlyeq_{P1} \cong_{1^*}$

c) $\sigma[P_1 \& P_2](R) \subseteq \sigma[P_1 \otimes P_2](R)$

Theorem 6 Smallest / largest BMO-sizes for ⊗, &

Consider $SCORE_d$ preferences $P_1 = (A_1, <_{P1}, \cong_{P1})$ and $P_2 = (A_2, <_{P2}, \cong_{P2})$. Varying $\cong_{P1}$ and $\cong_{P2}$ we have:

a) *Trivial* $\cong_{P1}$ and $\cong_{P2}$ yield *largest* BMO-sizes for $\sigma[P_1 \otimes P_2](R)$ and $\sigma[P_1 \& P_2](R)$, resp.

b) *Regular* $\cong_{P1}$ and $\cong_{P2}$ yield *smallest* BMO-sizes for $\sigma[P_1 \otimes P_2](R)$ and $\sigma[P_1 \& P_2](R)$, resp.

Example 14 BMO-sizes for varying SV-relations

We measured the BMO-sizes of the following queries for different choices of z, low, up, set_1, set_2 and set_3, evaluated against the same test collection as in Example 12:

- $Q_\alpha = \sigma[LOWEST_{d1}(height, \cong_1) \otimes HIGHEST_{d2}(length, \cong_2)](test\_coll)$
- $Q_\beta = \sigma[AROUND_{d1}(height, z, \cong_1) \otimes BETWEEN_{d2}(length, [low, up], \cong_2)](test\_coll)$
- $Q_\gamma = (\sigma[POS(color, set_1, \cong_1) \otimes POS(material, set_2, \cong_2)) \& HIGHEST_{d3}(height, \cong_3)](test\_coll)$
- $Q_\delta = \sigma[POS(color, set_1, \cong_1) \otimes NEG(material, set_2, \cong_2) \otimes AROUND_{d3}(height, \cong_3) \otimes HIGHEST_{d4}(length, \cong_4) \otimes HIGHEST_{d5}(width, \cong_5) \otimes BETWEEN_{d6}(weight, [low, up], \cong_6)](test\_coll)$

In Figure 3 we present selected test runs with characteristic effects concerning the flooding effect. The parameter rel_d indicates an equal proportion of the domain size. E.g., in Q_α d_1 and d_2 are chosen such that $(d_1 * 100) / (sup_{height} - inf_{height}) = (d_2 * 100) / (sup_{length} - inf_{length}) = rel_d$. ☼

The observed drops of BMO-sizes from trivial SV-relations to regular SV-relations are quite striking. In particular, looking at Q_δ in Figure 3 the often heard claim that Pareto queries are inherently prone to flooding seems to be refuted. The transition to equivalence classes in Theorem 4 algebraically explains this phenomenon. Note that a personalized query often has some hard constraints in addition to preferences (cmp. our Example 11), yielding even smaller query results. Thus the *flooding effect* becomes much less of an issue for deeply personalized database queries.

In many application scenarios BMO-results are *intelligent pre-selections*, which have to be refined afterwards. In our e-procurement example the virtual salesman Homer would apply his so-called *presentation preferences* to decide which item to pick first from the BMO-set to start the sales negotiation ([14]). This decision depends on many things, including sales psychology, and may even involve non-transitive arguments like majority voting.

As a synergetic effect, smaller BMO-sizes often coincide with *faster query evaluation*, in particular if the heuristics of ‘push preference’ for preference relational algebra is applied Moreover, deep personalization with a rich repertoire of preference constructors enables the chance to implement specialized efficient evaluation algorithms for each constructor (see e.g. [10]).

7. Related Work

The work of Chomicki ([7]) contains a fine survey on preference research. He investigates preference queries under the BMO model, calling it the 'winnow' operator. His definitions relax the strict partial order semantics of preferences. In particular, Pareto and prioritized preferences are defined using $\|_{P1}$ and $\|_{P2}$ instead of SV-relations, which fails to preserve strict partial order ([7], theorem 4.14). To abandon strict partial orders without need does impact the declarative semantics of preference query languages. The theory of subsumption lattices ([20]) guarantees both the existence of a model theory and of a corresponding fixpoint theory for BMO queries, if strict partial order is maintained. As reported in [7], some transformation laws of preference relational algebra are invalidated, if transitivity is dropped. This negative impact on query optimization is avoided by preferences with SV-semantics. For a survey on the decades-long discussion in decision theory on the sense or nonsense of non-transitive preferences see [8].

The importance of personalization in database queries is also emphasized in [21], proposing a preference model relying on scores and numerical ranking. Its lack of intuitive semantics may become an issue, since people usually don't think in plain numbers. Thus it is difficult to explain what a degree of interest of 0.755 really means. Adding more preference functionality as announced makes much sense, see e.g. [4] describing a personalized application that integrates numerical ranking with linguistic variables and categorical preferences. The smooth integration of personalization and database queries with the use of structured user profiles proposed in [21] has been supported by Preference SQL and Preference XPath for sophisticated applications ([15], [14]). There persistent preference repositories can be queried e.g. by Preference XPath to find best-matching preferences for a given situation ([11]). Efficient preference mining algorithms ([12]) can feed their findings into the repository.

Our deep personalization capabilities offer a novel means to combat the infamous flooding effect. So far in literature it has been criticized that Pareto preferences are impractical, because BMO-sizes get too large for increasing numbers of attributes. Empirical and analytical studies for skyline queries seemingly support that view (see e.g. [6], [5], [3]). But such investigations did not explore the full picture; instead only the worst case being d = 0 and trivial SV-relations has been investigated. Pareto preferences with categorical constructors haven't been considered either. However, we showed that reasonable choices of d and the SV-relations offer a novel, semantically guided way to influence BMO-sizes. Naturally, just like in the conventional relational model there are always queries with large result cardinalities. But with deep personalization the flooding effect can be controlled much better. Moreover, smaller BMO-sizes often coincide with faster query evaluation, in particular if the heuristics of 'push preference' for preference relational algebra is applied. Another approach to address the flooding problem is the Top-k query model, which can be problematic, if the intuitive semantics of the top k objects is unclear. As proved in [7], Top-k and BMO can be combined consistently.

8. Summary and Outlook

Deep personalization of database queries and applications requires a semantically rich, easy to handle and flexible preference model for query composition. The constructor-driven foundation of preferences as strict partial orders serves all these requirements. In this paper we have extended this approach in various crucial ways. First we showed up possibilities how to model a categorical view on numerical data for base constructors. Second, we demonstrated, how to custom-build new base constructors. Third we enriched preferences by SV-semantics, being a novel and semantically well-founded way to influence complex constructors. We could prove that inductive preference construction with SV-semantics (including Pareto, prioritization and numerical ranking) preserves the strict partial order property. All proposed constructor extensions were implemented in Preference XPath, using loose coupling and query rewriting into XPath for rapid prototyping.

Our increased modeling capabilities come with no added performance penalties. We showed that known transformation laws of preference relational algebra remain valid under SV-semantics. Thus increased query performance (over the loosely coupled case) can be expected for tightly coupling the BMO query optimizer with an existing database kernel for exact-match queries.

The BMO query model avoids the embarrassing 'empty result' effect. Concerning the annoying flooding effect we presented novel insights. We could relate lower and upper bounds for BMO-sizes of Pareto and prioritized preference queries to regular and trivial SV-relations. Performing a series of test queries using Preference XPath on real e-catalog data, we reported evidence that BMO-sizes can come up in handy portions. This observation makes us confident that also the flooding problem can be tamed in a semantically guided way.

There are many more research challenges for personalized database applications. For cost-based query optimization, which was not the topic here, analytic methods to estimate BMO-sizes for the full constructor spectrum are essential. For extended modeling explorations on how to integrate imprecision (e.g. fuzzy sets) with preference constructors seems worthwhile. Preferences and user modeling are investigated within the interdisciplinary Bavarian research cooperation FORSIP on "**S**ituated, **I**ndividualized and **P**ersonalized Human-Computer Interaction" (www.forsip.de). E.g., the fully automated sales agent COSIMA^{B2B}, which was recently exhibited at the computer fair SYSTEMS 2003 in Munich, enables a deep personalization of the B2B sales process and automates skills that so far could be performed only by human vendors. We also continue to extend Preference XPath beyond what was presented in this paper. Preference constructors on tree-structured and set-valued XML-objects as well as extensions to deal with ontologies are being added. Of course, upgrading Preference XPath to Preference XQuery will be done concurrently. We will require all of this advanced functionality to build a deeply personalized notification system for future MPEG-7 multimedia libraries.

Acknowledgments:
I would like to express my thanks to U. Güntzer for his valuable comments on a draft of this work. Many thanks also to S. Döring, T. Ehm and B. Hafenrichter who implemented the constructor extensions in Preference XPath.

References

[1] R. Agrawal, E. L. Wimmers: *A Framework for Expressing and Combining Preferences*. Proc. ACM SIGMOD, May 2000, Dallas, pp. 297 - 306.

[2] H. Andreka, M. Ryan, P.-Y. Schobbens: *Operators and Laws for Combining Preference Relations*. J. of Logic and Computation, 12, 1, 2002, pp. 13 – 53.

[3] W.T. Balke, U. Güntzer, J.X. Zheng: *Efficient Distributed Skylining for Web Information Systems*. Proc. Intern. Conf. on Extending Database Technology (EDBT), Heraklion, March 2004.

[4] W.-T. Balke, W. Kießling, C. Unbehend: *Performance and Quality Evaluation of a Personalized Route Planning System*. 18th Brazilian Symp. on Databases (SBBD), pp. 328-340, Manaus, 2003.

[5] J. Bentley, H. Kung, M. Schkolnick, C. Thomson: *On the average number of maxima in a set of vectors and applications*. JACM 25, 536-543 (1978).

[6] S. Borzsonyi, D. Kossmann, K. Stocker: *The Skyline Operator*. Proc. 17th Intern. Conf. On Data Engineering (ICDE), Heidelberg, April 2001.

[7] J. Chomicki: *Preference Formulas in Relational Queries*. ACM Transactions on Database Systems, Vol. 28, No. 4, Dec. 2003, pp. 1 – 39.

[8] P.C. Fishburn: *Nontransitive Preferences in Decision Theory*. Journal of Risk and Uncertainty, Kluwer Academic Publishers, 4:113 – 134, 1991.

[9] P. C. Fishburn: *Preference Structures and their Numerical Representations*. Journal of Theoretical Computer, 217, 1999, pp. 359 – 383.

[10] B. Hafenrichter: *Optimization of Relational Preference Database Queries* (in German). Ph.D. thesis, Univ. of Augsburg, Jan. 2004.

[11] S. Holland: *Preference Mining and Preference Repositories: Design, Algorithms and Personalized Applications*. Ph.D. thesis Univ. Augsburg, ibidem-Verlag Stuttgart, Jan. 2004, ISBN 3-89821-352-8.

[12] S. Holland, M. Ester, W. Kießling: *Preference Mining: A Novel Approach on Mining User Preferences for Personalized Applications.* 7th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD), pp. 204-216, Dubrovnik, Croatia, Sep. 2003.

[13] W. Kießling: *Foundations of Preferences in Database Systems*. Proc. 28th Intern. Conf. on Very Large Databases (VLDB), pp. 311-322, Hong Kong, China, 2002.

[14] W. Kießling, S. Fischer, S. Döring: *COSIMA^{B2B}-Sales Automation for E-Procurement*. Technical Report 2004-2, Institute of Computer Science, Univ. of Augsburg, Jan. 2004. (submitted for publ.)

[15] W. Kießling, S. Fischer, S. Holland, T. Ehm: *Design and Implementation of COSIMA - A Smart and Speaking E-Sales Assistant*. 3rd Intern. Workshop on Advanced Issues of E-Commerce and Web-Based Inform. Systems, pp. 21-30, San Jose, 2001. (demonstrated also at SIGMOD 2001, St. Barbara)

[16] W. Kießling, U. Güntzer: *Database Reasoning - A Deductive Framework for Solving Large and Complex Problems by means of Subsumption*. Proc. 3rd Workshop on Information Systems and Artificial Intell., LNCS 777, pp. 118-138, Hamburg, 1994.

[17] W. Kießling, B. Hafenrichter: *Optimizing Preference Queries for Personalized Web Services*. IASTED Intern. Conf. on Communication, Internet, Information Technology. St. Thomas, USA, Nov. 2002, pp. 461 - 466.

[18] W. Kießling, B. Hafenrichter, S. Fischer, S. Holland: *Preference XPATH: A Query Language for E-Commerce*. Proc. 5th Intern. Conf. Wirtschaftsinformatik, pp. 427-440, Augsburg, Germany, 2001.

[19] W. Kießling and G. Köstler: *Preference SQL - Design, Implementation, Experiences*. Proc. 28th Intern. Conf. on Very Large Data Bases (VLDB), pp. 990-1001, Hong Kong, China, 2002.

[20] G. Köstler, W. Kießling, H. Thöne, U. Güntzer: Fixpoint Iteration with Subsumption in *Deductive Databases*. Journal of Intelligent Information Systems, Vol. 4, pp. 123-148, Boston, USA, 1995.

[21] G. Koutrika, Y. Ioannidis: *Personalization of Queries in Database Systems*. Proc. 20th International Conference on Data Engineering (ICDE), Boston, USA, March 30 - April 2, 2004.

[22] M. Lacroix, P. Lavency: *Preferences: Putting more Knowledge into Queries*. Proc. 13th Intern. Conf. on Very Large Databases (VLDB), 1987.

[23] B. Xiao, E. Aïmeur, J. M. Fernandez: *PCFinder: An Intelligent Product Recommendation Agent for E-Commerce*. 2003 IEEE International Conference on Electronic Commerce (CEC 2003), pages 181-190, Newport Beach, CA, USA, 2003.

9. Appendix

We present the proofs of all propositions and theorems stated before explicitly or implicitly in the text. In addition, we list several laws from preference algebra and give some more illustrating examples.

9.1 Material related to section 3

Lemma 1 Properties of $SCORE_d$

a) $SCORE_d$ and SCORE are *equally expressive*.
b) $SCORE_d$ constructs a *weak order*.
c) $x \,\|_{P\text{-new}}\, y$ iff $f_d(x) = f_d(y)$

Proof:

a) f_d is a valid score function for SCORE. Since f_0 is identical to f, $SCORE_d$ and SCORE are equally expressive. This also implies that $SCORE_d$ constructs a strict partial order.
b) Due to a theorem by Fishburn ([9]), a necessary condition that a strict partial order P can be represented as a SCORE preference is that P is a weak order, which in turn implies that $\|_P$ is transitive. Thus due to a) $SCORE_d$ constructs a weak order.
c) Obvious. qed

Lemma 2 Properties of base constructors on numerical attributes

a) $BETWEEN_d$ is sub-constructor of $SCORE_d$.
   $x \,\|_{P\_new}\, y$ iff $dist_d[low, up](x) = dist_d[low, up](y)$
b) $AROUND_d$ is sub-constructor of $BETWEEN_d$.
   $x \,\|_P\, y$ iff $dist_d[z](x) = dist_d[z](y)$
c) $LOWEST_d$ and $HIGHEST_d$ are sub-constructors of $AROUND_d$.
   $x \,\|_{P\_new}\, y$ iff $dist_d[inf_A](x) = dist_d[inf_A](y)$
   $x \,\|_{P\_new}\, y$ iff $dist_d[sup_A](x) = dist_d[sup_A](y)$

Proof:

a) Specialize **f(a) := – dist[low, up](a)** in $SCORE_d$:

$x <_{P\_new} y$ iff $f_d(x) < f_d(y)$
iff {if d = 0 then – dist[low, up](x) else $\lceil$– dist[low, up](x)/d$\rceil$} <
{if d = 0 then – dist[low, up](y) else $\lceil$– dist[low, up](y)/d$\rceil$}

Since for a > 0: $\lceil -a \rceil = -\lceil a \rceil + 1$, we can continue:

iff {if d = 0 then – dist[low, up](x) else – $\lceil$dist[low, up](x)/d$\rceil$ + 1} <
{if d = 0 then – dist[low, up](y) else – $\lceil$dist[low, up](y)/d$\rceil$ + 1}
iff {if d = 0 then dist[low, up](y) < dist[low, up](x)
else $\lceil$dist[low, up](y)/d$\rceil$ < $\lceil$dist[low, up](x)/d$\rceil$
iff $dist_d[low, up](y) < dist_d[low, up](x)$

b) Specialize **z := low := up** in $BETWEEN_d$.
c) In $AROUND_d$ specialize **z := inf_A** for $LOWEST_d$ and **z := sup_A** for $HIGHEST_d$, respectively. qed

For the *'better-than' graph* of an EXPLICIT preference P a ***level***-function is defined with the following properties:

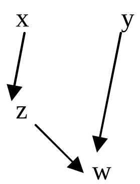
- $x <_P y$ implies level(y) < level(x), but not vice versa
- level(x) = level(y) implies $x \,\|_P\, y$, but not vice versa

It is interesting to note that for $LAYERED_m$ the definitions of a layer and a level coincide. Moreover, for $LAYERED_m$ also the reverse directions of above implications hold:

- $x <_P y$ iff layer(y) < layer(x)
- $x \,\|_P\, y$ iff layer(x) = layer(y)

Lemma 3 **EXPLICIT is no sub-constructor of $SCORE_d$ and vice versa.**

Proof: Consider this 'better-than' graph of a preference P, being EXPLICIT but not $LAYERED_m$:



We have $\neg(x <_P y)$ and $\neg(y <_P z)$, but $x <_P z$. This violates negative transitivity; hence P is no weak order and cannot be specified by $SCORE_d$. qed

Lemma 4 **Properties of $Super_e$SCORE**

a) $Super_e$SCORE constructs a strict partial order.
   $x \,\|_{P\_new}\, y$ iff $abs(f(x) - f(y)) \le e$
b) $Super_e$SCORE constructs no weak order, hence is no sub-constructor of $SCORE_d$.

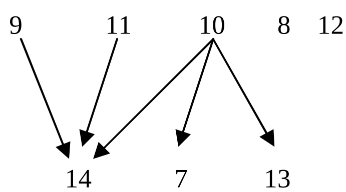
Proof:

a) Irreflexivity: $x <_P x$ iff $f(x) < f(x) - e$ iff $0 < -e$ iff false
   Transitivity: $x <_P y \wedge y <_P z$ iff $f(x) < f(y) - e \wedge f(y) < f(z) - e$
   implies $f(x) < f(z) - e - e < f(z) - e$ iff $x <_P z$

b) Let's study the $Super_e$AROUND sub-constructor of $Super_e$SCORE, which is defined by specializing $f(v) := -dist_z(v)$, where $dist_z(v) := abs(v - z)$.

   Assume dom(A) = [6, 20] $\subseteq \mathbb{R}$, z = 10 and R(A) = {7, 8, 9, 10, 11, 12, 13, 14}.
   Choosing e = 2 we get:
   $dist_{10}(v) = 0$, if $v = 10$
   $dist_{10}(v) = 1$, if $v \in \{9, 11\}$
   $dist_{10}(v) = 2$, if $v \in \{8, 12\}$
   $dist_{10}(v) = 3$, if $v \in \{7, 13\}$
   $dist_{10}(v) = 4$, if $v = 14$

   Thus the 'better-than' graph of P := $Super_2$AROUND(A, 10) for R(A) looks as follows:



   Obviously this is *no weak order*; hence it does not represent a $SCORE_d$ preference. Qed

9.2 Material related to section 4

Proposition 2 (see section 4.1) **Properties of SV-relations**

a) '=' is an SV-relation for each preference P (called **trivial** SV-relation).
b) If P is the anti-chain $A^{\leftrightarrow}$, each partition of dom(A) defines an SV-relation.

Proof:

a) '=' trivially satisfies Definition 9.
b) Each partition of dom(A) for an anti-chain $A^{\leftrightarrow}$ satisfies Definition 9a, b and c, because $<_{A\leftrightarrow} = \emptyset$. qed

Proposition 3 (see section 4.1) **Regular SV-relation for SCORE_d**

Given a SCORE_d preference P, let's define for all x, y $\in$ dom(A): $x \cong_P y$ iff $x \parallel_P y$

a) $\cong_P$ is an SV-relation (called **regular** SV-relation).

b) If P is not a chain, then $\cong_P$ is non-trivial.

Proof:

a)

- By definition: $x \cong_P y$ implies $x \parallel_P y$
- $x \cong_P y \wedge z <_P x$
  iff $x \parallel_P y \wedge z <_P x$
  iff {if d = 0 then f(x) = f(y) else $\lceil f(x) / d \rceil = \lceil f(y) / d \rceil$} $\wedge$
  {if d = 0 then f(z) < f(x) else $\lceil f(z) / d \rceil < \lceil f(x) / d \rceil$}
  implies {if d = 0 then f(z) < f(y) else $\lceil f(z) / d \rceil < \lceil f(y) / d \rceil$} iff $z <_P y$
- $x \cong_P y \wedge x <_P z$
  iff $x \parallel_P y \wedge x <_P z$
  iff {if d = 0 then f(x) = f(y) else $\lceil f(x) / d \rceil = \lceil f(y) / d \rceil$} $\wedge$
  {if d = 0 then f(x) < f(z) else $\lceil f(x) / d \rceil < \lceil f(z) / d \rceil$ }
  implies {if d = 0 then f(y) < f(z) else $\lceil f(y) / d \rceil < \lceil f(z) / d \rceil$} iff $y <_P z$
- $\parallel_P$ is reflexive and symmetric in general.
  Since SCORE_d constructs a weak order, $\parallel_P$ is transitive due to [9], hence $\cong_P$ is transitive.

b) If P is not a chain, then there are v, w $\in$ dom(A), $v \neq w$: $v \parallel_P w$, hence $v \cong_P w$. qed

We give two useful lemmas concerning the ***dual*** preference constructor ('∂') and the intersection of SV-relations.

Theorem 1 (see section 4.2) **Preservation of strict partial order for Pareto and prioritized construction**

Given $P_1 = (A_1, <_{P1}, \cong_{P1})$ and $P_2 = (A_2, <_{P2}, \cong_{P2})$, consider $P := P_1 \otimes P_2$ and $P := P_1 \,\&\, P_2$, respectively.
Then $P = (A_1 \cup A_2, <_P, \cong_P)$ is a preference with SV-semantics, i.e.:

a) $<_P$ is a strict partial order on $A_1 \cup A_2$.

b) $\cong_P$ is an SV-relation for $<_P$.

Proof:

We start with $P := P_1 \otimes P_2$:

a) $<_P$ is a strict partial order:

- *Irreflexivity:* $x <_P x$ iff (false $\wedge$ (false $\vee$ true)) $\vee$ (false $\wedge$ (false $\vee$ true)) iff false
- *Transitivity:* For abbreviation we define:
  $F_1 \equiv$ '$x_1 <_{P1} y_1$', $F_2 \equiv$ '$x_2 <_{P2} y_2$', $F_3 \equiv$ '$x_2 \cong_{P2} y_2$', $F_4 \equiv$ '$x_1 \cong_{P1} y_1$',
  $F_5 \equiv$ '$y_1 <_{P1} z_1$', $F_6 \equiv$ '$y_2 <_{P2} z_2$', $F_7 \equiv$ '$y_2 \cong_{P2} z_2$', $F_8 \equiv$ '$y_1 \cong_{P1} z_1$',
  $F_9 \equiv$ '$x_1 <_{P1} z_1$', $F_{10} \equiv$ '$x_2 <_{P2} z_2$', $F_{11} \equiv$ '$x_2 \cong_{P2} z_2$', $F_{12} \equiv$ '$x_1 \cong_{P1} z_1$'

  Due to Definition 9 for SV-relations we can state: // ***
  - Because $\cong_{P1}$ and $\cong_{P2}$ did *not* change $<_{P1}$ and $<_{P2}$: $F_1 \wedge F_5$ implies F_9, $F_2 \wedge F_6$ implies F_{10}
  - Because SV-relations are *transitive*: $F_4 \wedge F_8$ implies F_{12}, $F_3 \wedge F_7$ implies F_{11}
  - Because of properties of SV-relations: $F_1 \wedge F_8$ implies F_9, $F_2 \wedge F_7$ implies F_{10},
    $F_3 \wedge F_6$ implies F_{10}, $F_4 \wedge F_5$ implies F_9

  Then we get: $x <_P y \wedge y <_{P2} z$

  iff $[(F_1 \wedge (F_2 \vee F_3)) \vee (F_2 \wedge (F_1 \vee F_4))] \wedge [(F_5 \wedge (F_6 \vee F_7)) \vee (F_6 \wedge (F_5 \vee F_8))]$

  iff $[(F_1 \wedge F_2) \vee (F_1 \wedge F_3) \vee (F_2 \wedge F_4)] \wedge [(F_5 \wedge F_6) \vee (F_5 \wedge F_7) \vee (F_6 \wedge F_8)]$

  iff $(F_1 \wedge F_2 \wedge F_5 \wedge F_6) \vee (F_1 \wedge F_2 \wedge F_5 \wedge F_7) \vee (F_1 \wedge F_2 \wedge F_6 \wedge F_8) \vee$
  $(F_1 \wedge F_3 \wedge F_5 \wedge F_6) \vee (F_1 \wedge F_3 \wedge F_5 \wedge F_7) \vee (F_1 \wedge F_3 \wedge F_6 \wedge F_8) \vee$
  $(F_2 \wedge F_4 \wedge F_5 \wedge F_6) \vee (F_2 \wedge F_4 \wedge F_5 \wedge F_7) \vee (F_2 \wedge F_4 \wedge F_6 \wedge F_8)$

  According to // *** we can now continue:

  implies $(F_9 \wedge F_{10}) \vee (F_9 \wedge F_{10}) \vee (F_9 \wedge F_{10}) \vee$
  $(F_9 \wedge F_{10}) \vee (F_9 \wedge F_{11}) \vee (F_9 \wedge F_{10}) \vee$
  $(F_9 \wedge F_{10}) \vee (F_9 \wedge F_{10}) \vee (F_{10} \wedge F_{12})$

iff $((F_9 \wedge F_{10}) \vee (F_9 \wedge F_{11})) \vee ((F_9 \wedge F_{10}) \vee (F_{10} \wedge F_{12}))$
iff $(F_9 \wedge (F_{10} \vee F_{11})) \vee (F_{10} \wedge (F_9 \vee F_{12}))$ iff $x <_P z$ qed

b) $\cong_P$ is an SV-relation:
- $x \cong_P y$ iff $x_1 \cong_{P1} y_1 \wedge x_2 \cong_{P2} y_2$
  implies $x_1 \|_{P1} y_1 \wedge x_2 \|_{P2} y_2$
  iff $\neg(x_1 <_{P1} y_1) \wedge \neg(y_1 <_{P1} x_1) \wedge \neg(x_2 <_{P2} y_2) \wedge \neg(y_2 <_{P2} x_2)$ implies $x \|_P y$
- [Please note that the given proof consistently renames variables x, y, z in Definition 9b).]
  $x <_P y \wedge y \cong_P z$
  iff $(F_1 \wedge (F_2 \vee F_3)) \vee (F_2 \wedge (F_1 \vee F_4)) \wedge (F_8 \wedge F_7)$
  iff $((F_1 \wedge (F_2 \vee F_3)) \wedge F_8 \wedge F_7) \vee ((F_2 \wedge (F_1 \vee F_4)) \wedge F_8 \wedge F_7)$
  implies $((F_9 \wedge (F_2 \vee F_3)) \wedge F_7) \vee ((F_{10} \wedge (F_1 \vee F_4)) \wedge F_8)$
  implies $((F_9 \wedge (F_{10} \vee F_{11})) \vee ((F_{10} \wedge (F_9 \vee F_{12}))$ iff $x <_P z$
- [Please note that the given proof consistently renames variables x, y, z in Definition 9c).]
  $x <_P y \wedge x \cong_P z$
  iff $(F_1 \wedge (F_2 \vee F_3)) \vee (F_2 \wedge (F_1 \vee F_4)) \wedge (F_{12} \wedge F_{11})$
  iff $((F_1 \wedge (F_2 \vee F_3)) \wedge F_{11} \wedge F_{12}) \vee ((F_2 \wedge (F_1 \vee F_4)) \wedge F_{11} \wedge F_{12})$
  implies $((z_1 <_{P1} y_1 \wedge (F_2 \vee F_3)) \wedge F_{11}) \vee ((z_2 <_{P2} y_2 \wedge (F_1 \vee F_4)) \wedge F_{12})$
  implies $((z_1 <_{P1} y_1 \wedge (z_2 <_{P2} y_2 \vee z_2 \cong_{P2} y_2)) \vee$
  $((z_2 <_{P2} y_2 \wedge (z_1 <_{P1} y_1 \vee z_1 \cong_{P1} y_1))$ iff $z <_P y$
- $\cong_P$ is reflexive, symmetric and transitive, since it is the intersection of two equivalence relations $\cong_{P1}$ and $\cong_{P2}$.
  qed

Now let's turn to $P := P_1 \& P_2$:

a) $<_P$ is a strict partial order:
- *Irreflexivity:* $x <_P x$ iff false $\vee$ (true $\wedge$ false) iff false
- *Transitivity:* Given the same abbreviations and implications [***] as stated in the proof for Pareto preferences above we can conclude:
  $x <_P y \wedge y <_P z$
  iff $(F_1 \vee (F_4 \wedge F_2)) \wedge (F_5 \vee (F_8 \wedge F_6))$
  iff $(F_1 \wedge F_5) \vee (F_1 \wedge F_6 \wedge F_8) \vee (F_2 \wedge F_4 \wedge F_5) \vee (F_2 \wedge F_4 \wedge F_6 \wedge F_8)$
  implies $F_9 \vee (F_6 \wedge F_9) \vee (F_2 \wedge F_9) \vee (F_{10} \wedge F_{12})$
  iff $F_9 \vee (F_9 \wedge (F_2 \vee F_6)) \vee (F_{10} \wedge F_{12})$ iff $F_9 \vee (F_{10} \wedge F_{12})$ iff $x <_P z$

b) $\cong_P$ is an SV-relation:
- $x \cong_P y$ implies $x \|_P y$: Clear (see Pareto preferences).
- [Please note that the given proof consistently renames variables x, y, z in Definition 9b).]
  $x <_P y \wedge y \cong_P z$
  iff $(F_1 \vee (F_4 \wedge F_2)) \wedge (F_8 \wedge F_7)$
  iff $(F_1 \wedge F_8 \wedge F_7) \vee (F_4 \wedge F_2 \wedge F_8 \wedge F_7)$
  implies $(F_9 \wedge F_7) \vee (F_4 \wedge F_{10} \wedge F_8)$
  implies $(F_9 \wedge F_7) \vee (F_{12} \wedge F_{10})$ implies $F_9 \vee (F_{12} \wedge F_{10})$ iff $x <_P z$
- [Please note that the given proof consistently renames variables x, y, z in Definition 9c).]
  $x <_P y \wedge x \cong_P z$
  iff $(F_1 \vee (F_4 \wedge F_2)) \wedge (F_{12} \wedge F_{11})$
  iff $(F_1 \wedge F_{12} \wedge F_{11}) \vee (F_4 \wedge F_2 \wedge F_{12} \wedge F_{11})$
  implies $(z_1 <_{P1} y_1 \wedge F_{11}) \vee (z_1 \cong_{P1} y_1 \wedge F_2 \wedge F_{11})$
  implies $(z_1 <_{P1} y_1 \wedge F_{11}) \vee (z_1 \cong_{P1} y_1 \wedge z_2 <_{P2} y_2)$
  implies $z_1 <_{P1} y_1 \vee (z_1 \cong_{P1} y_1 \wedge z_2 <_{P2} y_2)$ iff $z <_P y$
- $\cong_P$ is reflexive, symmetric and transitive: Clear (see Pareto preferences). qed

Theorem 2 (see section 4.2) Further properties of '⊗' and '&' with SV-semantics

a) Pareto or prioritized preferences don't possess regular SV-relations in general.
b) Any relaxation of SV-semantics for Pareto or prioritized construction violates strict partial order.

Proof:

a) Given two weak order preferences P_1 and P_2, then due to **[7]**, Proposition 4.15, P := $P_1 \otimes P_2$ is not a weak order. The same holds, if $P_1 = (A_1, <_{P1}, \cong_{P1})$ and $P_2 = (A_2, <_{P2}, \cong_{P2})$ are weak orders with SV-semantics. As can be seen from the case studies in Example 8, it's straightforward to find alternative values, prohibiting the existence of a regular SV-relation.
b) All SV-properties are required to establish the proof of Theorem 1. An example, which relaxes SV-properties and hence violates strict partial order, can be found in the proof of **[7]**, theorem 4.14. qed

A numerical preference is a weighted combination of preferences $P_1, \dots P_n$.

Definition 16 Numerical preference constructor with SV-semantics ('rank$_F$')

For $1 \le i \le n$ let $P_i := \mathrm{SCORE}_{di}(A_i, f_i, \cong_{Pi})$ and $f_{di}(x_i) := \{\text{if } d_i = 0 \text{ then } f_i(x_i) \text{ else } \lceil f_i(x_i) / d_i \rceil\}$.
Further we assume an n-ary *combining function* F: $\mathbb{R} \times \dots \times \mathbb{R} \to \mathbb{R}$:

$$\texttt{complex}\ \mathrm{rank}_F(P_1, \dots, P_n)$$
$$\{(x_1, \dots, x_n) <_{P\_new} (y_1, \dots y_n) \text{ iff } F(f_{d1}(x_1), \dots, f_{dn}(x_n)) < F(f_{d1}(y_1), \dots, f_{dn}(y_n));$$
$$(x_1, \dots, x_n) \cong_{P\_new} (y_1, \dots y_n) \text{ iff } x_1 \cong_{P1} y_1 \wedge \dots \wedge x_n \cong_{Pn} y_n\};$$

Lemma 5 Preservation of strict partial order for rank$_F$

Given $P_i := \mathrm{SCORE}_{di}(A_i, f_i, \cong_{Pi})$, $1 \le i \le n$, for $P := \mathrm{rank}_F(P_1, \dots, P_n)$ the following holds:

a) $<_P$ is a strict partial order on $A_1 \cup \dots \cup A_n$.
b) $\cong_P$ is an SV-relation for $<_P$.

Proof:

a) Obvious, since SCORE_{di} is equally expressive as SCORE and $\cong_{P1}, \dots, \cong_{Pn}$ are not utilized for $<_P$.
b) $\cong_P$ is an SV-relation:
   - $x \cong_P y$ iff $x_1 \cong_{P1} y_1 \wedge \dots \wedge x_n \cong_{Pn} y_n$
     implies $x_1 \|_{P1} y_1 \wedge \dots \wedge x_n \|_{Pn} y_n$
     iff $f_{d1}(x_1) = f_{d1}(y_1) \wedge \dots \wedge f_{dn}(x_n) = f_{dn}(y_n)$
     iff $F(f_{d1}(x_1), \dots, f_{dn}(x_n)) = F(f_{d1}(y_1), \dots, f_{dn}(y_n))$ iff $x \|_P y$
   - $z <_P x \wedge x \cong_P y$
     iff $F(f_{d1}(z_1), \dots, f_{dn}(z_n)) < F(f_{d1}(x_1), \dots, f_{dn}(x_n)) \wedge x_1 \cong_{P1} y_1 \wedge \dots \wedge x_n \cong_{Pn} y_n$
     implies $F(f_{d1}(z_1), \dots, f_{dn}(z_n)) < F(f_{d1}(x_1), \dots, f_{dn}(x_n)) \wedge f_{d1}(x_1) = f_{d1}(y_1) \wedge \dots \wedge f_{dn}(x_n) = f_{dn}(y_n)$
     iff $F(f_{d1}(z_1), \dots, f_{dn}(z_n)) < F(f_{d1}(y_1), \dots, f_{dn}(y_n))$ iff $z <_P y$
   - $x <_P z \wedge x \cong_P y$
     iff $F(f_{d1}(x_1), \dots, f_{dn}(x_n)) < F(f_{d1}(z_1), \dots, f_{dn}(z_n)) \wedge x_1 \cong_{P1} y_1 \wedge \dots \wedge x_n \cong_{Pn} y_n$
     implies $F(f_{d1}(x_1), \dots, f_{dn}(x_n)) < F(f_{d1}(z_1), \dots, f_{dn}(z_n)) \wedge f_{d1}(x_1) = f_{d1}(y_1) \wedge \dots \wedge f_{dn}(x_n) = f_{dn}(y_n)$
     iff $F(f_{d1}(y_1), \dots, f_{dn}(y_n)) < F(f_{d1}(z_1), \dots, f_{dn}(z_n))$ iff $y <_P z$
   - $\cong_P$ is reflexive, symmetric and transitive: Obvious. qed

From Definition 16 it is obvious that $<_P$ *cannot* benefit from SV-information. However, since $\cong_P$ is inherited from $\cong_{P1}, \dots, \cong_{Pn}$ it has an impact on BMO sizes, if P itself is part of a more complex preference.

Example 15 Impact of SV-relations in rank$_F$

Consider $P := \mathrm{rank}_F(P_1, P_2)$, a preference P_3 and $P_4 := P \,\&\, P_3$. Then $<_{P4}$ is defined as follows:

$$(x_1, x_2, x_3) <_{P4} (y_1, y_2, y_3) \text{ iff } (x_1, x_2) <_P (y_1, y_2) \vee ((x_1, x_2) \cong_P (y_1, y_2) \wedge x_3 <_{P3} y_3)$$
$$\text{iff } F(f_{d1}(x_1), f_{d2}(x_2)) < F(f_{d1}(y_1), f_{d2}(y_2)) \vee (x_1 \boldsymbol{\cong_{P1}} y_1 \wedge x_2 \boldsymbol{\cong_{P2}} y_2 \wedge x_3 <_{P3} y_3)$$ ☼

Lemma 6 Properties of rank$_F$

a) Let $P := \mathrm{rank}_F(P_1, \dots, P_n)$: $\cong_{Pi}$ is regular $(1 \le i \le n)$ iff $\cong_P$ is regular
b) rank_F constructs a weak order.

Proof:

a) $\cong_{Pi}$ is regular $(1 \le i \le n)$

iff $(\forall x_i, y_i \in dom(A_i)$: $x_i \cong_{Pi} y_i$ iff $x_i \,\|_{Pi}\, y_i$, $1 \le i \le n)$

iff $(\forall x = (x_1, \ldots, x_n), y = (y_1, \ldots, y_n) \in dom(A_1) \times \ldots \times dom(A_n)$:

$x_1 \cong_{P1} y_1 \wedge \ldots \wedge x_n \cong_{Pn} y_n$ iff $x_1 \,\|_{P1}\, y_1 \wedge \ldots \wedge x_n \,\|_{Pn}\, y_n)$

iff $(\forall x = (x_1, \ldots, x_n), y = (y_1, \ldots, y_n) \in dom(A_1) \times \ldots \times dom(A_n)$: $x \cong_P y$ iff $x \,\|_P\, y)$

iff $\cong_P$ is regular

b) $P := rank_F(P_1, \ldots, P_n)$ is a weak order, because it can be characterized as a SCORE(A, f, $\cong_P$) preference as follows:

- $A := A_1 \cup \ldots \cup A_n$
- f: $dom(A) \rightarrow \mathbb{R}$, $f((x_1, \ldots, x_n)) := F(f_{d1}(x_1), \ldots, f_{dn}(x_n))$ qed

Note that Definition 16 can now be considered to cover also the case that instead of a $SCORE_{di}$ preference also a numerical preference can be supplied.

The intersection preference constructor '♦' assembles a preference P from two preferences P_1 and P_2 that act on the same attribute set A.

Definition 17 Intersection preference constructor with SV-semantics ('♦')

We assume $P_1 = (A, <_{P1}, \cong_{P1})$ and $P_2 = (A, <_{P2}, \cong_{P2})$.

`complex` P_1 ♦ P_2 $\{x <_{P\_new} y$ iff $x <_{P1} y \wedge x <_{P2} y$; $x \cong_{P\_new} y$ iff $x \cong_{P1} y \wedge x \cong_{P2} y\}$;

Lemma 7 Intersection '♦' is a preference sub-constructor of Pareto '⊗'

Proof:

Given $P := P_1 \otimes P_2$ where $P_1 = (A, <_{P1}, \cong_{P1})$ and $P_2 = (A, <_{P2}, \cong_{P2})$, applying the definition of '⊗' for the case of identical attributes we get for all $x, y \in dom(A)$:

$x <_P y$ iff $(x <_{P1} y \wedge (x <_{P2} y \vee x \cong_{P1} y)) \vee$
$(x <_{P2} y \wedge (x <_{P1} y \vee x \cong_{P2} y))$
iff $(x <_{P1} y \wedge x <_{P2} y) \vee (x <_{P1} y \wedge x \cong_{P1} y)) \vee$
$(x <_{P2} y \wedge x <_{P1} y) \vee (x <_{P2} y \wedge x \cong_{P2} y))$ // property of SV-relations
iff $(x <_{P1} y \wedge x <_{P2} y) \vee$ false$) \vee (x <_{P2} y \wedge x <_{P1} y) \vee$ false$)$
iff $(x <_{P1} y \wedge x <_{P2} y) \vee (x <_{P2} y \wedge x <_{P1} y)$ iff $x <_{P1} y \wedge x <_{P2} y$

$x \cong_P y$ iff $x \cong_{P1} y \wedge x \cong_{P2} y$ qed

Let $range(<_P) := \{x \in dom(A) \mid \exists y \in dom(A): (x, y) \in <_P$ or $(y, x) \in <_P\}$. Two preferences $P_1 = (A_1, <_{P1})$ and $P_2 = (A_2, <_{P2})$ are ***disjoint*** preferences, if $range(<_{P1}) \cap range(<_{P2}) = \emptyset$.

The disjoint union preference constructor '+' assembles a preference P from two *disjoint* preferences P_1 and P_2 that act on the same attribute set A.

Definition 18 Disjoint union preference constructor with SV-semantics ('+')

We assume disjoint $P_1 = (A, <_{P1}, \cong_{P1})$ and $P_2 = (A, <_{P2}, \cong_{P2})$.

`complex` $P_1 + P_2$ $\{x <_{P\_new} y$ iff $x <_{P1} y \vee x <_{P2} y$; $x \cong_{P\_new} y$ iff $x \cong_{P1} y \wedge x \cong_{P2} y\}$;

Lemma 8 Preservation of strict partial order for '+'

Given disjoint $P_1 = (A, <_{P1}, \cong_{P1})$ and $P_2 = (A, <_{P2}, \cong_{P2})$, for $P := P_1 + P_2$ the following holds:

a) $<_P$ is a strict partial order on A.

b) $\cong_P$ is an SV-relation for $<_P$.

Proof:

a) Obvious, since $\cong_{P1}$ and $\cong_{P2}$ are not utilized for $<_P$.

b) $\cong_P$ is an SV-relation:

- $x \cong_P y$ iff $x \cong_{P1} y \wedge x \cong_{P2} y$
  implies $x \parallel_{P1} y \wedge x \parallel_{P2} y$
  iff $\neg(x <_{P1} y) \wedge \neg(y <_{P1} x) \wedge \neg(x <_{P2} y) \wedge \neg(y <_{P2} x)$
  iff $(\neg(x <_{P1} y) \wedge \neg(x <_{P2} y)) \wedge (\neg(y <_{P1} x) \wedge \neg(y <_{P2} x))$
  iff $\neg(x <_{P1} y \vee x <_{P2} y) \wedge \neg(y <_{P1} x \vee y <_{P2} x)$
  iff $\neg(x <_P y) \wedge \neg(y <_P x)$ iff $x \parallel_P y$
- $z <_P x \wedge x \cong_P y$ iff $(z <_{P1} x \vee z <_{P2} x) \wedge x \cong_{P1} y \wedge x \cong_{P2} y$
  iff $(z <_{P1} x \wedge x \cong_{P1} y \wedge x \cong_{P2} y) \vee (z <_{P2} x \wedge x \cong_{P1} y \wedge x \cong_{P2} y)$
  implies $z <_{P1} y \vee z <_{P2} y$ iff $z <_P y$
- $x <_P z \wedge x \cong_P y$ iff $(x <_{P1} z \vee x <_{P2} z) \wedge x \cong_{P1} y \wedge x \cong_{P2} y$
  iff $(x <_{P1} z \wedge x \cong_{P1} y \wedge x \cong_{P2} y) \vee (x <_{P2} z \wedge x \cong_{P1} y \wedge x \cong_{P2} y)$
  implies $y <_{P1} z \vee y <_{P2} z$ iff $y <_P z$
- $\cong_P$ is reflexive, symmetric and transitive: Obvious. qed

The linear sum constructor '⊕' assembles a preference P from $P_1 = (A_1, <_{P1}, \cong_{P1})$ and $P_2 = (A_2, <_{P2}, \cong_{P2})$, assuming that $dom(A_1) \cap dom(A_2) = \emptyset$; hence P_1 and P_2 being disjoint. Assuming that $dom(A_1)$ and $dom(A_2)$ are union-compatible we define a new attribute A, where $dom(A) = dom(A_1) \cup dom(A_2)$.

Definition 19 Linear sum preference constructor with SV-semantics ('⊕')

We assume $P_1 = (A_1, <_{P1}, \cong_{P1})$ and $P_2 = (A_2, <_{P2}, \cong_{P2})$ where $dom(A_1) \cap dom(A_2) = \emptyset$.

`complex` $P_1 \oplus P_2$
$\{x <_{P\_new} y$ iff $x <_{P1} y \vee x <_{P2} y \vee (x \in dom(A_2) \wedge y \in dom(A_1))$;
$x \cong_{P\_new} y$ iff $x \cong_{P1} y \vee x \cong_{P2} y\}$;

Note that $<_{P\_new}$ of linear sum cannot exploit $\cong_{P1}$ or $\cong_{P2}$.

Lemma 9 Preservation of strict partial order for '⊕'

Given $P_1 = (A_1, <_{P1}, \cong_{P1})$ and $P_2 = (A_2, <_{P2}, \cong_{P2})$, where $dom(A_1) \cap dom(A_2) = \emptyset$ and $dom(A) = dom(A_1) \cup dom(A_2)$, then for $P := P_1 \oplus P_2$ the following holds:

a) $<_P$ is a strict partial order on A.
b) $\cong_P$ is an SV-relation for $<_P$.

Proof:

a) Obvious, since $\cong_{P1}$ and $\cong_{P2}$ are not utilized for $<_P$.
b) $\cong_P$ is an SV-relation:

- $x \cong_P y$ iff $x \cong_{P1} y \vee x \cong_{P2} y$ implies $x \parallel_{P1} y \vee x \parallel_{P2} y$
  From the definition of $<_P$ and observing that its right-hand side is a disjoint disjunction, we see that $x \parallel_{P1} y$ implies $x \parallel_P y$ and $x \parallel_{P2} y$ implies $x \parallel_P y$, implying that $x \parallel_P y$.

- $z <_P x \wedge x \cong_P y$ iff $(z <_{P1} x \vee z <_{P2} x \vee (z \in dom(A_2) \wedge x \in dom(A_1))) \wedge x \cong_P y$
  iff $((z <_{P1} x \vee z <_{P2} x) \wedge x \cong_P y) \vee (z \in dom(A_2) \wedge x \in dom(A_1))) \wedge x \cong_P y$
  iff $((z <_{P1} x \vee z <_{P2} x) \wedge x \cong_P y) \vee$ false
  iff $(z <_{P1} x \vee z <_{P2} x) \wedge x \cong_P y$
  iff $(z <_{P1} x \wedge x \cong_P y) \vee (z <_{P2} x \wedge x \cong_P y)$
  iff $(z <_{P1} x \wedge (z \cong_{P1} x \vee z \cong_{P2} x)) \vee (z <_{P2} x \wedge (z \cong_{P1} x \vee z \cong_{P2} x))$
  implies $z <_{P1} y \vee z <_{P2} y$
  Since $\neg(z \in dom(A_2) \wedge y \in dom(A_1))$ holds, we can continue:
  iff $z <_{P1} y \vee z <_{P2} y \vee (z \in dom(A_2) \wedge y \in dom(A_1))$ iff $z <_P y$

- $x <_P z \wedge x \cong_P y$ iff $(x <_{P1} z \vee x <_{P2} z \vee (x \in dom(A_2) \wedge z \in dom(A_1))) \wedge x \cong_P y$
  iff $((x <_{P1} z \vee x <_{P2} z) \wedge x \cong_P y) \vee (x \in dom(A_2) \wedge z \in dom(A_1))) \wedge x \cong_P y$
  iff $((x <_{P1} z \vee x <_{P2} z) \wedge x \cong_P y) \vee$ false

iff $(x <_{P1} z \;\vee\; x <_{P2} z) \;\wedge\; x \cong_P y$

iff $(x <_{P1} z \;\wedge\; x \cong_P y) \;\vee\; (x <_{P2} z \;\wedge\; x \cong_P y)$

iff $(x <_{P1} z \;\wedge\; (x \cong_{P1} y \;\vee\; x \cong_{P2} y)) \;\vee\; (x <_{P2} z \;\wedge\; (x \cong_{P1} y \;\vee\; x \cong_{P2} y))$

implies $y <_{P1} z \;\vee\; y <_{P2} z$

Since $\neg(z \in dom(A_1) \;\wedge\; y \in dom(A_2))$ holds, we can continue:

iff $y <_{P1} z \vee y <_{P2} z \vee (y \in dom(A_2) \wedge z \in dom(A_1))$ iff $y <_P z$

- Reflexivity and symmetry of $\cong_P$ is clear.
  $\cong_P$ is transitive: $x \cong_P y \;\wedge\; y \cong_P z$ iff $(x \cong_{P1} y \;\vee\; x \cong_{P2} y) \;\wedge\; (y \cong_{P1} z \;\vee\; y \cong_{P2} z)$

  iff $(x \cong_{P1} y \;\wedge\; y \cong_{P1} z) \;\vee\; (x \cong_{P1} z \;\vee\; y \cong_{P2} z) \;\vee$
  $(x \cong_{P2} y \;\wedge\; y \cong_{P1} z) \;\vee\; (x \cong_{P2} z \;\vee\; y \cong_{P2} z)$

  Since $x \cong_P y$ implies $\neg(x \in dom(A_2) \;\wedge\; y \in dom(A_1)) \wedge \neg(y \in dom(A_2) \;\wedge\; x \in dom(A_1))$ we can continue:

  iff $(x \cong_{P1} y \;\wedge\; y \cong_{P1} z) \;\vee$ false $\vee$ false $\vee\; (x \cong_{P2} z \;\wedge\; y \cong_{P2} z)$

  iff $(x \cong_{P1} y \;\wedge\; y \cong_{P1} z) \;\vee\; (x \cong_{P2} z \;\wedge\; y \cong_{P2} z)$

  implies $x \cong_{P1} z \;\vee\; x \cong_{P2} z)$ iff $x \cong_P z$ qed

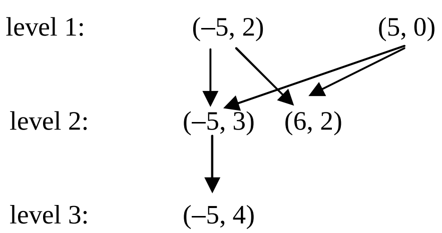
Theorem 3 (see section 4.4) Expressiveness of complex constructors with SV-semantics

a) Pareto is no sub-constructor of $rank_F$ and vice versa. (Skyline preferences cannot be expressed by $rank_F$.)
b) Pareto is no sub-constructor of '&' and vice versa.
c) '&' is no sub-constructor of '$rank_F$' and vice versa. (Grouped preferences are not expressible by $rank_F$.)

Proof:

a) We start with showing that $rank_F$ is no sub-constructor of Pareto:
   - From Lemma 12a it follows that if $(x_1, x_2) \in max(P_1 \,\&\, P_2)$ or $(x_1, x_2) \in max(P_2 \,\&\, P_1)$, then $(x_1, x_2) \in max(P_1 \otimes P_2)$, too. Thus if we find P_1, P_2 and a combining function F such that (x_1, x_2) is maximal in $P_1 \,\&\, P_2$ or in $P_2 \,\&\, P_1$, but not in $rank_F(P_1, P_2)$, the proof is achieved:
     Let $P_1 := LOWEST_0(A_1, \text{'='})$, $P_2 := LOWEST_0(A_2, \text{'='})$, $P := rank_F(P_1, P_2)$ where $F(f_1(x_1), f_2(x_2)) := -x_1 - x_2$ and consider $R = \{(6, 4), (6, 1), (6, 8), (2, 7), (2, 9), (3, 2)\} \subseteq dom(A_1) \times dom(A_2)$. Then $\sigma[P](R) = \{(3, 2)\}$, however $\sigma[P_1 \,\&\, P_2](R) = \{(2, 7)\}$ and $\sigma[P_2 \,\&\, P_1](R) = \{(6, 1)\}$.

   Now let's prove the reverse:
   - Given $P_1 := SCORE_d(A_1, f_1, \cong_{P1})$ and $P_2 := SCORE_d(A_2, f_2, \cong_{P2})$, we prove that there exists $P := P_1 \otimes P_2$ such that P cannot be constructed by $rank_F(P_1, P_2)$:
     We study an example with $AROUND_0$, being a sub-constructor of $SCORE_d$. Let $P_1 := AROUND_0(A_1, 0, \cong_{P1})$, $P_2 := AROUND_0(A_2, 1, \cong_{P2})$, where both $\cong_{P1}$ and $\cong_{P2}$ are regular, and $P := P_1 \otimes P_2$. Considering $R = \{(-5, 2), (5, 0), (-5, 3), (-5, 4), (6, 2)\} \subseteq dom(A_1) \times dom(A_2)$, then the 'better-than' graph of P restricted to R looks as follows:



// remark: $(-5, 2) \cong_P (5, 0)$

// remark: $(-5, 3) \,\|_P\, (6, 2)$, but $\neg\, (-5, 3) \cong_P (6, 2)$

     Now observe that $\neg((-5, 3) <_P (6, 2))$ and $\neg((6, 2) <_P ((-5, 4))$, but $(-5, 3) <_P (-5, 4)$, violating negative transitivity, hence not representing a weak order. But due to Lemma 6b $rank_F$ constructs a weak order for any F.

b) Lemma 12a and Lemma 7 reveal that '$\otimes$' and '&' are recursively related in a very subtle way. Thus '&' cannot be a sub-constructor of '$\otimes$' and vice versa.

c) We start with showing that '&' is no sub-constructor of $rank_F$:

- Assume $P_2 := \text{LOWEST}_0(B, \text{'='})$ and consider $P := P_2$ groupby $A \equiv A^{\leftrightarrow}$ & P_2, where $A^{\leftrightarrow} = (A, \varnothing, \text{'='})$. Considering $R = \{(6, 4), (6, 3), (6, 8), (2, 7), (2, 9), (3, 2)\} \subseteq \text{dom}(A) \times \text{dom}(B)$, the 'better-than' graph of P restricted to R looks as follows:

$$\begin{tikzcd}[row sep=large]
\text{level 1:} & (6, 3) \arrow[d] & (2, 7) \arrow[d] & (3, 2) \\
\text{level 2:} & (6, 4) \arrow[d] & (2, 9) & \\
\text{level 3:} & (6, 8) & &
\end{tikzcd}$$

  Now observe that $\neg((6, 3) <_P (2, 7))$ and $\neg((2, 7) <_P ((6, 4))$, but $(6, 3) <_P (6, 4)$ violating negative transitivity, hence not representing a weak order. But due to Lemma 6b rank_F constructs a weak order for any F.

Now let's prove the reverse:

- From Lemma 12b it follows that if (x_1, x_2) is maximal in P_1 & P_2, then x_1 is maximal in P_1. Thus if we find P_1, P_2 and a combining function F such that (x_1, x_2) is maximal in $\text{rank}_F(P_1, P_2)$ but x_1 is not maximal in P_1, then the proof is achieved.
  Let $P_1 := \text{LOWEST}_0(A_1, \text{'='})$, $P_2 := \text{LOWEST}_0(A_2, \text{'='})$ and $P := \text{rank}_F(P_1, P_2)$, where $F(f_1(x_1), f_2(x_2)) := -x_1 - x_2$. Note that since $\cong_{P1}$ is trivial, Lemma 12b applies. Now consider $R = \{(6, 4), (6, 1), (6, 8), (2, 7), (2, 9), (3, 2)\} \subseteq \text{dom}(A_1) \times \text{dom}(A_2)$: Then $\sigma[P](R) = \{(3, 2)\}$. However, we have $3 \notin \sigma[P_1](R) = \{2\}$.

That grouped preferences and skyline preferences cannot be expressed by rank_F follows from Theorem 3a, c. The latter observation was also made by [7] for skylines without the DIFF operator and without the extension to SV-semantics. qed

9.3 Material related to section 5

Theorem 4 (see section 5.1) Every SV-order is strict partial order

Moreover, $[P] = (A/\cong_P, <_{[P]})$ is a strict partial order, where $\cong_{[P]}$ is the trivial SV-relation, i.e. equality of equivalence classes.

Proof:

- $<_{[P]}$ is *well-defined*:
  Consider $x <_P y$ for some $x \in X$, $y \in Y$. Then for each $x' \in X$ by Definition 9 b) $x' <_P y$ holds. Likewise, for each $y' \in Y$ by Definition 9 c) $x <_P y'$ holds. Thus the definition of $X <_{[P]} Y$ is independent from the chosen representative for X and Y.
- $<_{[P]}$ is *irreflexive*:
  $X <_{[P]} X$ iff $(\forall\, x \in X, \forall\, y \in X: x <_P y)$ implies $(\forall x \in X: x <_P x)$ iff false
- $<_{[P]}$ is *transitive*:
  $X <_{[P]} Y \wedge Y <_{[P]} Z$ iff $(\forall\, x \in X, \forall\, y \in Y: x <_P y) \wedge (\forall\, y \in Y, \forall\, z \in Z: y <_P z)$
  Considering a fixed, but arbitrary $y_0 \in Y$ we can continue:
  implies $(\forall\, x \in X: x <_P y_0 \wedge \forall\, z \in Z: y_0 <_P z)$
  By transitivity of $<_P$ we get:
  implies $(\forall\, x \in X, \forall\, z \in Z: x <_P z)$ iff $X <_{[P]} Z$
- $\cong_{[P]}$ is *trivial*:
  For $X \neq Y$ by the definition we get:
  $X \cong_{[P]} Y$ iff $\forall x \in X, \forall\, y \in Y: x \cong_P y$ implies (Proposition 2a) $\forall x \in X, \forall\, y \in Y$: false iff false
  Thus $\cong_{[P]}$ is the equality of equivalence classes for $A/\cong_P$, hence it represents the trivial SV-relation on $A/\cong_P$. qed

In [13] we have identified many algebraic laws amongst preferences with trivial SV-relations. Now we extend them towards arbitrary SV-relations. To this end we extend this notion of equivalence of preferences in [13], being an order isomorphism for $<_P$, to include an order-isomorphism for SV-relations.

Definition 20 Equivalence of preferences ($P_1 \equiv P_2$)

$P_1 = (A, <_{P1}, \cong_{P1})$ and $P_2 = (A, <_{P2}, \cong_{P2})$ are **equivalent ($P_1 \equiv P_2$)**, if for all x, y $\in$ dom(A):

- $x_1 <_{P1} y_1$ iff $x <_{P2} y$
- $x_1 \cong_{P1} y_1$ iff $x \cong_{P2} y$

Lemma 10 Commutativity and associativity

a) Pareto ('$\otimes$') is commutative and associative
b) Prioritization ('&') is associative, but not commutative.
c) Intersection ('♦') is commutative and associative.
d) Disjoint union ('+') is commutative and associative.
e) Linear sum ('$\oplus$') is associative, but not commutative.

Proof: For illustration we give the proof for the associativity of '&':

Given $P_1 = (A_1, <_{P1}, \cong_{P1})$, $P_2 = (A_2, <_{P2}, \cong_{P2})$ and $P_3 = (A_3, <_{P3}, \cong_{P3})$, let $P := (P_1 \& P_2) \& P_3$ and $P^* := P_1 \& (P_2 \& P_3)$.
For all $x = (x_1, x_2, x_3)$, $y = (y_1, y_2, y_3) \in dom(A_1) \times dom(A_2) \times dom(A_3)$ we get:

$$
\begin{aligned}
x <_P y \quad & \text{iff} \quad (x_1, x_2) <_{P1\&P2} (y_1, y_2) \vee (\mathbf{(x_1, x_2) \cong_{P1\&P2} (y_1, y_2)} \wedge x_3 <_{P3} y_3) \\
& \text{iff} \quad x_1 <_{P1} y_1 \vee (x_1 \cong_{P1} y_1 \wedge x_2 <_{P2} y_2) \vee (\mathbf{x_1 \cong_{P1} y_1 \wedge x_2 \cong_{P2} y_2} \wedge x_3 <_{P3} y_3) \\
& \text{iff} \quad x_1 <_{P1} y_1 \vee (x_1 \cong_{P1} y_1 \wedge (x_2 <_{P2} y_2 \vee (x_2 \cong_{P2} y_2 \wedge x_3 <_{P3} y_3)) \\
& \text{iff} \quad x_1 <_{P1} y_1 \vee (x_1 \cong_{P1} y_1 \wedge (x_2, x_3) <_{P2\&P3} (y_2, y_3)) \quad \text{iff} \quad x <_{P^*} y
\end{aligned}
$$

Associativity of $\cong_P$ is obvious, since: $\cong_{P1\&P2}$ iff $x_1 \cong_{P1} y_1 \wedge x_2 \cong_{P2} y_2$ qed

Lemma 11 Distributivity

a) '$\otimes$' does not distribute over '&' and vice versa.
b) '&' distributes over '+': $P := P_1 \& (P_2 + P_3) \equiv (P_1 \& P_2) + (P_1 \& P_3)$

Proof:

a) It's easy to find counterexamples.
b) The $<_P$-part is covered by [7], theorem 4.7; the $\cong_P$-part is straightforward. qed

Lemma 12 Intuitive interpretations of '$\otimes$' and '&'

a) $P_1 \otimes P_2 \equiv (P_1 \& P_2) ♦ (P_2 \& P_1)$ (*non-discrimination theorem* for Pareto)
b) If $P_1 = (A, <_{P1}, \cong_{P1})$ and $P_2 = (A, <_{P2}, \cong_{P2})$, where $\cong_{P1}$ and $\cong_{P2}$ are the same or $\cong_{P1}$ is trivial, then:
$P_1 \& P_2 \equiv P_1$ (*discrimination theorem* for prioritization)

Proof:

a) Let $P := P_1 \otimes P_2$ and $P^* := (P_1 \& P_2) ♦ (P_2 \& P_1)$. For all $x = (x_1, x_2)$, $y = (y_1, y_2) \in dom(A_1) \times dom(A_2)$ after having done some Boolean algebra transformations we can state:
$x <_{P^*} y$ iff $x <_P y \vee ((x_1 \cong_{P1} y_1 \wedge x_1 <_{P1} y_1) \wedge (x_2 \cong_{P2} y_2 \wedge x_2 <_{P2} y_2))$
Due to the SV-properties in Definition 9 we can continue:
iff $x <_P y \vee$ (false $\wedge$ false) iff $x <_P y$
$x \cong_{P^*} y$ iff $x \cong_{P1\&P2} y \wedge x \cong_{P2\&P1} y$ iff $x_1 \cong_{P1} y_1 \wedge x_2 \cong_{P2} y_2 \wedge x_2 \cong_{P2} y_2 \wedge x_1 \cong_{P1} y_1$
iff $x_1 \cong_{P1} y_1 \wedge x_2 \cong_{P2} y_2$ iff $x \cong_P y$

b) Note that the case of overlapping attribute sets in Definition 11b applies:
For all x, y $\in$ dom(A), $x \neq y$, we get: $x <_{P1\&P2} y$ iff $x <_{P1} y \vee (x \cong_{P1} y \wedge x <_{P2} y)$ [*]
In case of $P_1 = (A, <_{P1}, \cong_A)$, $P_2 = (A, <_{P2}, \cong_A)$ we can continue [*]:
iff $x <_{P1} y \vee (x \cong_A y \wedge x <_{P2} y)$ iff $x <_{P1} y \vee$ false iff $x <_{P1} y$
In case of $P_1 = (A, <_{P1})$, $P_2 = (A, <_{P2}, \cong_{P2})$ we can continue [*]:
iff $x <_{P1} y \vee (x = y \wedge x <_{P2} y)$ iff $x <_{P1} y \vee$ false iff $x <_{P1} y$

For x = y irreflexivity yields: $x <_{P1\&P2} y$ iff false iff $x <_{P1} y$

$x \cong_{P1\&P2} y$ iff $x \cong_{P1} y \wedge x \cong_{P2} y$ [**]

In case of $P_1 = (A, <_{P1}, \cong_A)$, $P_2 = (A, <_{P2}, \cong_A)$ we can continue [**]:

iff $x \cong_A y \wedge x \cong_A y$ iff $x \cong_A y$

In case of $P_1 = (A, <_{P1})$, $P_2 = (A, <_{P2}, \cong_{P2})$ we can continue [**]:

iff $x = y \wedge x \cong_{P2} y$ iff $x = y$ qed

This theorem is an algebraic support for the intuitive interpretation that:

a) Pareto construction $P_1 \otimes P_2$ treats both preferences as *equally important*.
b) Prioritized construction P_1 & P_2 treats P_1 as *more important* than P_2. Under the given conditions the winner P_1 even "takes it all".

Example 16 Prioritization: The winner takes it all … sometimes

We study P_1 := POS(A, {1, 2, 3}, $\cong_{P1}$) and P_2 := POS(A, {1, 2}, $\cong_{P2}$), where dom(A) = {1, …, 6}.
Now let R(A) = {2, 3, 4} $\subseteq$ dom(A) and consider P := P_1 & P_2 .

- *Case 1*: $x \cong_{P1} y$ iff $x \cong_{P2} y$ iff $x, y \in \{1, 2\} \vee x, y \in \{3\} \vee x, y \in \{4, 5, 6\}$
  Our proposition applies, yielding that σ[P](R) = {2, 3}, which are all of P_1's favorites in R. Note that here 2 and 3 are not substitutable.

- *Case 2*: Let $x \cong_{P1} y$ iff $x, y \in \{1, 2, 3\} \vee x, y \in \{4, 5, 6\}$, // $\cong_{P1}$ is regular
  let $x \cong_{P2} y$ iff $x, y \in \{1, 2\} \vee x, y \in \{3, 4, 5, 6\}$. // $\cong_{P2}$ is regular
  Lemma 12b does not apply here, hence: σ[P](R) = {2}. Note that now 2 and 3 are substitutable for P_1. Therefore only that one favored by P_2 turns up in σ[P](R). ☼

As a new law we present a sort of idempotent behavior, nicely demonstrating once more the intuitive nature of our framework and its technical flexibility, allowing complex preference construction on *overlapping* sets of attributes.

Lemma 13 "Idempotency" for prioritized and Pareto preference construction

a) $P_1 \& P_2 \equiv P_1 \& (P_1 \otimes P_2)$
b) $P_1 \otimes P_2 \equiv (P_1 \otimes P_2) \& P_1$

Proof: We assume preferences $P_1 = (A_1, <_{P1}, \cong_{P1})$ and $P_2 = (A_2, <_{P2}, \cong_{P2})$.

a) Let P := P_1 & ($P_1 \otimes P_2$). Then P is defined of the set of attributes $A = A_1 \cup (A_1 \cup A_2) = A_1 \cup A_2$. This means that in Definition 11b the case of overlapping attribute sets applies.

$$\begin{aligned}
\mathbf{(x_1, x_2) <_P (y_1, y_2)} \quad &\text{iff} \quad x_1 <_{P1} y_1 \vee (x_1 \cong_{P1} y_1 \wedge (x_1, x_2) <_{P1\otimes P2} (y_1, y_2)) \\
&\text{iff} \quad x_1 <_{P1} y_1 \vee (x_1 \cong_{P1} y_1 \wedge ((x_1 <_{P1} y_1 \wedge (x_2 \cong_{P2} y_2 \vee x_2 <_{P2} y_2)) \vee \\
&\qquad (x_2 <_{P2} y_2 \wedge (x_1 \cong_{P1} y_1 \vee x_1 <_{P1} y_1))))
\end{aligned}$$

Due to our SV-properties we can continue:

$$\begin{aligned}
&\text{iff} \quad x_1 <_{P1} y_1 \vee (\text{false} \vee (x_1 \cong_{P1} y_1 \wedge x_2 <_{P2} y_2 \wedge (x_1 \cong_{P1} y_1 \vee x_1 <_{P1} y_1))) \\
&\text{iff} \quad x_1 <_{P1} y_1 \vee (x_1 \cong_{P1} y_1 \wedge x_2 <_{P2} y_2 \wedge (x_1 \cong_{P1} y_1 \vee x_1 <_{P1} y_1))
\end{aligned}$$

Again, due to our SV-properties we can continue:

$$\begin{aligned}
&\text{iff} \quad x_1 <_{P1} y_1 \vee (x_1 \cong_{P1} y_1 \wedge x_2 <_{P2} y_2 \wedge x_1 \cong_{P1} y_1) \vee \text{false} \\
&\text{iff} \quad x_1 <_{P1} y_1 \vee (x_1 \cong_{P1} y_1 \wedge x_2 <_{P2} y_2) \quad \text{iff} \quad \mathbf{(x_1, x_2) <_{P1\&P2} (y_1, y_2)}
\end{aligned}$$

b) Let P := ($P_1 \otimes P_2$) & P_1. Then P is defined of the set of attributes $A = (A_1 \cup A_2) \cup A_2 = A_1 \cup A_2$. This means that the case of overlapping attribute sets applies again.

$$\begin{aligned}
\mathbf{(x_1, x_2) <_P (y_1, y_2)} \quad &\text{iff} \quad (x_1, x_2) <_{P1\otimes P2} (y_1, y_2) \vee ((x_1, x_2) \cong_{P1\otimes P2} (y_1, y_2) \wedge x_1 <_{P1} y_1) \\
&\text{iff} \quad (x_1, x_2) <_{P1\otimes P2} (y_1, y_2) \vee (x_1 \cong_{P1} y_1 \wedge x_2 \cong_{P2} y_2 \wedge x_1 <_{P1} y_1)
\end{aligned}$$

Due to our SV-properties we can continue:

$$\text{iff} \quad (x_1, x_2) <_{P1\otimes P2} (y_1, y_2) \vee \text{false} \quad \text{iff} \quad \mathbf{(x_1, x_2) <_{P1\otimes P2} (y_1, y_2)}$$ qed

An intuitive reading of these algebraic properties is the following:

Repeating a preference in a less important argument does not change anything.

(A German saying would express this as “*Getretener Quark wird breit nicht stark.*”)

Now we list a series of further useful laws, given under Proposition 3 in [13].

Lemma 14 Further useful preference algebra laws

- Laws for dual preference construction
  - a) $(P^{\partial})^{\partial} \equiv P$
  - b) $(P_1 \oplus P_2)^{\partial} \equiv P_2^{\partial} \oplus P_1^{\partial}$
  - c) $POS(A, V, \cong_A)^{\partial} \equiv NEG(A, V, \cong_A)$, $NEG(A, V, \cong_A)^{\partial} \equiv POS(A, V, \cong_A)$
  - d) $HIGHEST_d(A, \cong_A) \equiv LOWEST_d(A, \cong_A)^{\partial}$
- Laws for intersection preference construction
  - e) $P \blacklozenge P \equiv P$
  - f) $P \blacklozenge P^{\delta} \equiv A^{\leftrightarrow}$ for the anti-chain $A^{\leftrightarrow} = (A, \varnothing, \cong_P)$
  - g) If $P = (A, <_P, \cong_P)$, then $P \blacklozenge A^{\leftrightarrow} \equiv A^{\leftrightarrow}$ for the anti-chain $A^{\leftrightarrow} = (A, \varnothing, \cong_P)$.
- Laws for Pareto preference construction
  - h) $P \otimes P \equiv P$
  - i) If $P = (A, <_P, \cong_P)$, then $P \otimes A^{\leftrightarrow} \equiv A^{\leftrightarrow}$ for the anti-chain $A^{\leftrightarrow} = (A, \varnothing, \cong_P)$
  - j) $P \otimes P^{\partial} \equiv A^{\leftrightarrow}$ for the anti-chain $A^{\leftrightarrow} = (A, \varnothing, \cong_P)$
- Laws for prioritized preference construction
  - k) If P_1 and P_2 are chains, then P_1 & P_2 is a chain.
  - l) If $P = (A, <_P, \cong_P)$, then $P \,\&\, A^{\leftrightarrow} \equiv P$ for the anti-chain $A^{\leftrightarrow} = (A, \varnothing, \cong_P)$.
  - m) $P \,\&\, P^{\partial} \equiv P$
  - n) If $P = (A, <_P, \cong_P)$, then P groupby $A \equiv A^{\leftrightarrow}$ for $A^{\leftrightarrow} = (A, \varnothing)$ or $A^{\leftrightarrow} = (A, \varnothing, \cong_P)$.

Proof: We present only those parts not covered already by [13].

a) It can be proved that $\cong_P$ is SV-relation for P iff $\cong_P$ is SV-relation for P^{∂}. Therefore we have:

$\cong_{(P\partial)\partial}$ is SV-relation for $(P^{\partial})^{\partial}$ iff $\cong_{(P\partial)\partial}$ is SV-relation for P^{∂} iff $\cong_{(P\partial)\partial}$ is SV-relation for P

b) $x \cong_{(P1 \oplus P2)\partial} y$ iff $x \cong_{P1 \oplus P2} y$ iff $x \cong_{P1} y \vee x \cong_{P2} y$ iff $x \cong_{P1\partial} y \vee x \cong_{P2\partial} y$ iff $x \cong_{P1\partial \oplus P2\partial} y$

c) Given $P_1 := POS(A, V, \cong_A)$ and $P_2 := NEG(A, V, \cong_A)$, then from [13] we know that:

$P_1 \equiv V^{\leftrightarrow} \oplus \text{other-set}^{\leftrightarrow}$ and $P_2 \equiv \text{other-set}^{\leftrightarrow} \oplus V^{\leftrightarrow}$

Thus: $P_1^{\partial} \equiv (V^{\leftrightarrow} \oplus \text{other-set}^{\leftrightarrow})^{\partial} \equiv (\text{other-set}^{\leftrightarrow})^{\partial} \oplus (V^{\leftrightarrow})^{\partial} \equiv \text{other-set}^{\leftrightarrow} \oplus V^{\leftrightarrow} \equiv P_2$

$P_2^{\partial} \equiv (P_1^{\partial})^{\partial} \equiv P_1$

d) Let $P_1 := LOWEST_d(A, \cong_A)$ and $P_2 := HIGHEST_d(A, \cong_A)$.

$x <_{P1\partial} y$ iff $y <_{P1} x$ iff {if $d = 0$ then $x < y$ else $\lceil dist_{infA}(x) / d \rceil < \lceil dist_{infA}(y) / d \rceil$}

Applying the duality principle for partially ordered sets we can continue:

iff {if $d = 0$ then $x < y$ else $\lceil dist_{supA}(y) / d \rceil < \lceil dist_{supA}(x) / d \rceil$} iff $x <_{P2} y$

Since $\cong_{P1}$ and $\cong_{P2}$ are identical by assumption we are done.

e) $x \cong_{P\blacklozenge P} y$ iff $x \cong_P y \wedge x \cong_P y$ iff $x \cong_P y$

f) $x \cong_{P\blacklozenge P\delta} y$ iff $x \cong_P y \wedge x \cong_{P\delta} y$ iff $x \cong_P y \wedge x \cong_P y$ iff $x \cong_P y$

g) $x \cong_{P\blacklozenge A\leftrightarrow} y$ iff $x \cong_P y \wedge x \cong_P y$ iff $x \cong_P y$

Case h), i), and j) are direct corollaries from Lemma 7:

h) $P \otimes P \equiv P \blacklozenge P \equiv P$

i) $P \otimes A^{\leftrightarrow} \equiv P \blacklozenge A^{\leftrightarrow} \equiv A^{\leftrightarrow}$

j) $P \otimes P^{\partial} \equiv P \blacklozenge P^{\partial} \equiv A^{\leftrightarrow}$

k) If P_1 and P_2 are chains, then $\cong_{P1}$ and $\cong_{P2}$ are restricted to the trivial SV-relation ‘=’.

l) Immediate from Lemma 12b.

m) $\cong_P$ is identical to $\cong_{P\partial}$, hence Lemma 12b applies.

n) Immediate from Lemma 12b, since ‘P groupby A’ is a synonym for $A^{\leftrightarrow}$ & P. qed

9.5 Material related to section 6

Proposition 4 (see section 6.1) BMO-sizes of $SCORE_d$ are non-monotonic in d.

Proof: Assuming $d_2 > d_1 > 0$, this effect is due to the following property:

- $\mathbf{x <_{Pd1} y}$ iff $\lceil f(x)/d_1 \rceil < \lceil f(y)/d_1 \rceil$ implies $f(x)/d_1 < f(y)/d_1$ iff $f(x)/d_2 < f(y)/d_2$
  implies $\lceil f(x)/d_2 \rceil \le \lceil f(y)/d_2 \rceil$ iff $\mathbf{x <_{Pd2} y \vee x \parallel_{Pd2} y}$
- Assuming in addition that $f(x) < f(y)$:
  $\mathbf{x \parallel_{Pd1} y}$ iff $\lceil f(x)/d_1 \rceil = \lceil f(y)/d_1 \rceil$
  implies $0 \le (f(y) - f(x))/ d_1 \le 1$ implies $0 \le (f(y) - f(x))/ d_2 \le 1$
  implies $\lceil f(x)/d_2 \rceil \le \lceil f(y)/d_2 \rceil$ iff $\mathbf{x <_{Pd2} y \vee x \parallel_{Pd2} y}$

As a net effect we have a ***non-monotonic*** behavior: If value x is not in the BMO-set for d_1, it may get into it for some $d_2 > d_1$. On the other hand, if x is in the BMO-set for d_1, then there is no guarantee that x stays in it for $d_2 > d_1$. qed

Example 17 BMO-sizes for varying family parameter d

We study $P_d := SCORE_d(A, f)$ for a given relation $R(A) = \{a_1, a_2, a_3\}$, where $f(a_1) = 2.5$, $f(a_2) = 3.2$ and $f(a_3) = 3.5$. Let's define BMO-size(d) := card($\sigma[P_d](R)$).

- $d_1 = 1.0$: $\lceil f(a_1) / d_1 \rceil = \lceil 2.5/1.0 \rceil = \lceil 2.5 \rceil = 3$,
  $\lceil f(a_2) / d_1 \rceil = \lceil 3.2/1.0 \rceil = \lceil 3.2 \rceil = 4$,
  $\lceil f(a_3) / d_1 \rceil = \lceil 3.5/1.0 \rceil = \lceil 3.5 \rceil = 4$, yielding BMO-size($d_1$) = 2
- $d_2 = 1.7$: $\lceil f(a_1) / d_2 \rceil = \lceil 2.5/1.7 \rceil = \lceil 1.59 \rceil = 2$,
  $\lceil f(a_2) / d_2 \rceil = \lceil 3.2/1.7 \rceil = \lceil 1.88 \rceil = 2$,
  $\lceil f(a_3) / d_2 \rceil = \lceil 3.5/1.7 \rceil = \lceil 2.06 \rceil = 3$, yielding BMO-size($d_2$) = 1
- $d_3 = 2.0$: $\lceil f(a_1) / d_3 \rceil = \lceil 2.5/2.0 \rceil = \lceil 1.25 \rceil = 2$,
  $\lceil f(a_2) / d_3 \rceil = \lceil 3.2/2.0 \rceil = \lceil 1.60 \rceil = 2$,
  $\lceil f(a_3) / d_3 \rceil = \lceil 3.5/2.0 \rceil = \lceil 1.75 \rceil = 2$, yielding BMO-size($d_3$) = 3

Thus $d_1 \le d_2$ does not imply that BMO-size(d_1) ≤ BMO-size(d_2). ☼

Proposition 6 Proposition 5 (see section 6.2) Properties of $\succcurlyeq_P$

a) $\succcurlyeq_P$ is a non-strict partial order on the set of all SV-relations of a preference P.
b) If P is constructed by $SCORE_d$, then the regular (trivial) SV-relation is the greatest (smallest) element of $\succcurlyeq_P$.

Proof:
a) Immediate, since logical implication is reflexive, transitive and asymmetric:
   $\cong_P \succcurlyeq_P \cong_P$
   $\cong_3 \succcurlyeq_P \cong_2 \wedge \cong_2 \succcurlyeq_P \cong_1$ implies $\cong_3 \succcurlyeq_P \cong_1$
   $\cong_1 \succcurlyeq_P \cong_2 \wedge \cong_2 \succcurlyeq_P \cong_1$ implies $\cong_1 \equiv \cong_2$
b) For a regular SV-relation of $SCORE_d$ (cmp. Proposition 3) all unordered values are substitutable. On the other hand, for every SV-relation substitutable values must be unordered (Definition 9a). For the trivial SV-relation no two different values are substitutable. qed

Theorem 5 (see section 6.2) Monotonicity of BMO-sizes for ⊗ and &

Consider $P_1 = (A_1, <_{P1}, \cong_1)$, $P_1^* = (A_1, <_{P1}, \cong_{1*})$, differing only wrt the SV-relation, and similarly $P_2 = (A_2, <_{P2}, \cong_2)$, $P_2^* = (A_2, <_{P2}, \cong_{2*})$.

a) $\sigma[P_1 \otimes P_2](R) \subseteq \sigma[P_1^* \otimes P_2^*](R)$ if $\cong_1 \succcurlyeq_{P1} \cong_{1*}$ and $\cong_2 \succcurlyeq_{P2} \cong_{2*}$
b) $\sigma[P_1 \,\&\, P_2](R) \subseteq \sigma[P_1^* \,\&\, P_2](R)$ if $\cong_1 \succcurlyeq_{P1} \cong_{1*}$
c) $\sigma[P_1 \,\&\, P_2](R) \subseteq \sigma[P_1 \otimes P_2](R)$

Proof:

a) If $\cong_1 \succcurlyeq_{P1} \cong_{1*}$ and $\cong_2 \succcurlyeq_{P2} \cong_{2*}$, then from Definition 11a it is clear that:

for all x, y $\in$ dom($A_1 \cup A_2$): $x <_{P1* \otimes P2*} y$ implies $x <_{P1 \otimes P2} y$

Then according to [7], Theorem 5.5, the proof is immediate.

b) If $\cong_1 \succcurlyeq_{P1} \cong_{1*}$, then from Definition 11b it is clear that:

for all x, y $\in$ dom($A_1 \cup A_2$): $x <_{P1* \otimes P2} y$ implies $x <_{P1 \otimes P2} y$

Again according to [7], Theorem 5.5, the proof is immediate.

c) Direct corollary from Lemma 12a:

$(x_1, x_2) <_{P1 \otimes P2\_new} (y_1, y_2)$ iff $(x_1, x_2) <_{P1 \,\&\, P2\_new} (y_1, y_2) \wedge (x_1, x_2) <_{P2 \,\&\, P1\_new} (y_1, y_2)$

implies $(x_1, x_2) <_{P1 \,\&\, P2\_new} (y_1, y_2)$ qed

Theorem 6 (see section 6.2) Smallest / largest BMO-sizes for $\otimes$ and &

Consider SCORE$_d$ preferences $P_1 = (A_1, <_{P1}, \cong_{P1})$ and $P_2 = (A_2, <_{P2}, \cong_{P2})$. Varying $\cong_{P1}$ and $\cong_{P2}$ we have:

a) $\cong_{P1}$ and $\cong_{P2}$ being trivial yield *largest* BMO-sizes for $\sigma[P_1 \otimes P_2](R)$ and $\sigma[P_1 \,\&\, P_2](R)$, resp.

b) $\cong_{P1}$ and $\cong_{P2}$ being regular yield *smallest* BMO-sizes for $\sigma[P_1 \otimes P_2](R)$ and $\sigma[P_1 \,\&\, P_2](R)$, resp.

Proof: Direct corollary from Theorem 5 and Proposition 5b. qed