An Algebra of Layered Complex Preferences

Bernhard Möller   Patrick Roocks

Institut für Informatik, Universität Augsburg

September 18, 2012

Motivation

Our work is based on:

- Preferences for Database queries
- Abstract Relation Algebra

Motivation

Our work is based on:

- Preferences for Database queries
- Abstract Relation Algebra

What are database preferences?

- Strict partial orders expressing user wishes, e.g.
  - "I like x more than y"
- Soft constraints in database queries, e.g.
  - if no tuples with "$X \leq 0$" exist, return those with lowest X
- Used for personalised information systems, e.g.
  - queries are extended by personalised preferences

$\longrightarrow$ Introductory example

Motivation

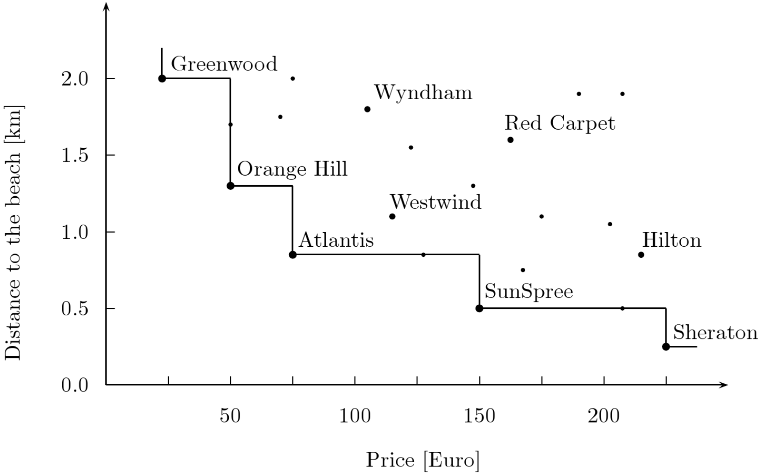


Figure: Skyline of hotels which are cheap *and* near to the beach

Motivation

- Preference relations are irreflexive and transitive (strict orders)
- Some are additionally negatively transitive (strict weak orders)
- Complex preferences (e.g. *“cheap and near to the beach”*)...
  - ... are no weak orders in general!

Motivation

- Preference relations are irreflexive and transitive (strict orders)
- Some are additionally negatively transitive (strict weak orders)
- Complex preferences (e.g. *“cheap and near to the beach”*)...
  - ... are no weak orders in general!

Strict weak orders:

- Induce a total order of equivalence classes
- Useful for constructing complex preferences

Motivation

- Preference relations are irreflexive and transitive (strict orders)
- Some are additionally negatively transitive (strict weak orders)
- Complex preferences (e.g. *"cheap and near to the beach"*)...
  - ... are no weak orders in general!

Strict weak orders:

- Induce a total order of equivalence classes
- Useful for constructing complex preferences

The challenge:

- Transform arbitrary complex preferences to weak orders
  $\rightarrow$ *"Layered Complex Preferences"*
- Show that many properties are preserved

Outline

The basic work was done in our first paper

"An Algebraic Calculus of Database Preferences" (at MPC 2012)

Therein we presented:

- Typed relational algebra to represent preference terms
- Maximal element algebra to formalize preference selections

Outline

The basic work was done in our first paper

"An Algebraic Calculus of Database Preferences" (at MPC 2012)

Therein we presented:

- Typed relational algebra to represent preference terms
- Maximal element algebra to formalize preference selections

The talk is structured as follows:

1. Recapitulation of the basics
2. Extensions of our calculus
3. Transformation: General preferences → Layered preferences
4. Application: The "Pareto-regular" preference

Types

Motivation for typing:

- Handling compositions of preferences on different attributes
  - e.g. “Lower price” and “Lower distance”
- Mathematically, both are ordered sets $(\mathbb{R}, <)$ on the same domain

Types

Motivation for typing:

- Handling compositions of preferences on different attributes
  - e.g. “Lower price” and “Lower distance”
- Mathematically, both are ordered sets $(\mathbb{R}, <)$ on the same domain

We introduce *types* of relations according to their *attribute names*.

Thereby we define:

- $\mathcal{A}$: set of attribute names (e.g. set of column names)
- D_A for all $A \in \mathcal{A}$: The *type domain* of the attribute, e.g. $\mathbb{R}, \mathbb{N}$, strings,... (`int, float, varchar,...`)
- A subset $T \subseteq \mathcal{A}$ is a *type* with the *type domain* D_T

Typed semirings

Basic structure:

- Consider an idempotent semiring with choice "$+$" and composition "$\cdot$" with neutral element 1
- Preference relations are general elements therein with choice "$\cup$" and composition ";" with $\emptyset$ and identity relation as neutral elements
- Sets are represented as elements ≤ 1 (algebraically: *tests*)

Typed semirings

Basic structure:

- Consider an idempotent semiring with choice "$+$" and composition "$\cdot$" with neutral element 1
- Preference relations are general elements therein with choice "$\cup$" and composition ";" with $\emptyset$ and identity relation as neutral elements
- Sets are represented as elements ≤ 1 (algebraically: *tests*)

Special elements:

- 0_T: smallest element
- 1_T: identity relation
- $\top_T$: greatest element

Type assertions

$$a :: T^2 \quad \Leftrightarrow_{df} \quad a = 1_T \cdot a \cdot 1_T$$

$$p :: T \quad \Leftrightarrow_{df} \quad p \leq 1_T$$

Type assertions

$$a :: T^2 \quad \Leftrightarrow_{df} \quad a = 1_T \cdot a \cdot 1_T$$

$$p :: T \quad \Leftrightarrow_{df} \quad p \leq 1_T$$

In the concrete relational instances:

$$a :: T^2 \quad \Leftrightarrow \quad a \subseteq D_T \times D_T$$

$$p :: T \quad \Leftrightarrow \quad p \subseteq D_T$$

Type assertions

$$a :: T^2 \quad \Leftrightarrow_{df} \quad a = 1_T \cdot a \cdot 1_T$$

$$p :: T \quad \Leftrightarrow_{df} \quad p \leq 1_T$$

In the concrete relational instances:

$$a :: T^2 \quad \Leftrightarrow \quad a \subseteq D_T \times D_T$$

$$p :: T \quad \Leftrightarrow \quad p \subseteq D_T$$

For $r :: T$ (i.e. $r \leq 1_T$) the *r-induced sub-type of T* is defined as:

$$p :: T[r] \quad \Leftrightarrow \quad p \leq r$$

$$a :: T[r]^2 \quad \Leftrightarrow \quad a \leq r \cdot a \cdot r$$

with $1_{T[r]} =_{df} r$ and $\top_{T[r]} = r \cdot \top_T \cdot r$

Joins

- We introduce the join operator ("$\bowtie$") to represent relational compositions of preferences.

$$a :: T_a^2, b :: T_b^2 \quad \Longrightarrow \quad a \bowtie b :: (T_a \bowtie T_b)^2$$

- Join is required to be associative, commutative, distributes over "$+$", diamond distributes over join, etc.
- In the concrete instances $T_a \bowtie T_b$ is the Cartesian product $D_{T_a} \times D_{T_b}$.

Abstract relation algebra

- We also need the converse and the complement

Definition (Abstract relation algebra)

- Idempotent semiring
- Additional operators: converse $(\ldots)^{-1}$ and complement $\overline{(\ldots)}$

Abstract relation algebra

- We also need the converse and the complement

Definition (Abstract relation algebra)

- Idempotent semiring
- Additional operators: converse $(\ldots)^{-1}$ and complement $\overline{(\ldots)}$
- Axiomatised by the Schröder equivalences and Huntington's axiom:

$$x \cdot y \leq z \Leftrightarrow x^{-1} \cdot \overline{z} \leq \overline{y} \Leftrightarrow \overline{z} \cdot y^{-1} \leq \overline{x}\,, \qquad x = \overline{\overline{x} + y} + \overline{\overline{x} + \overline{y}}\,.$$

Abstract relation algebra

- We also need the converse and the complement

Definition (Abstract relation algebra)

- Idempotent semiring
- Additional operators: converse $(\ldots)^{-1}$ and complement $\overline{(\ldots)}$
- Axiomatised by the Schröder equivalences and Huntington's axiom:

$$x \cdot y \leq z \Leftrightarrow x^{-1} \cdot \overline{z} \leq \overline{y} \Leftrightarrow \overline{z} \cdot y^{-1} \leq \overline{x}\,, \qquad x = \overline{\overline{x} + y} + \overline{\overline{x} + \overline{y}}\,.$$

We additionally stipulate the Tarski rule

$$a \neq 0_a \Rightarrow \top_a \cdot a \cdot \top_a = \top_a\,,$$

where $\top_a = \overline{0_a}$.

We assume: For $x :: T^2$ we have also $x^{-1}, \overline{x} :: T^2$

Derived relational operations

- Meet of two elements (intersection)

$$x \sqcap y =_{df} \overline{\overline{x} + \overline{y}}$$

- Relative complement

$$x - y =_{df} x \sqcap \overline{y}$$

- For tests $p, q \leq 1$ these are:

$$p \sqcap q = p \cdot q \,, \qquad p - q = p \cdot \neg q$$

Preferences

Definition ((Layered) preferences)

A relation a is a *preference* if and only if it is irreflexive and transitive, i.e.

1. $a \sqcap 1_a = 0_a$,
2. $a \cdot a \leq a$.

a is a *layered preference* if additionally *negative transitivity* holds:

$$\overline{a} \cdot \overline{a} \leq \overline{a}$$

Preferences

Definition ((Layered) preferences)

A relation a is a *preference* if and only if it is irreflexive and transitive, i.e.

1. $a \sqcap 1_a = 0_a$,
2. $a \cdot a \leq a$.

a is a *layered preference* if additionally *negative transitivity* holds:

$$\overline{a} \cdot \overline{a} \leq \overline{a}$$

Layered preferences induce a “layered structure”, i.e. for $a :: T^2$ with finite D_T there is always a function $f : D_T \to \mathbb{N}$ s.t.

$$t_1 \, a \, t_2 \quad \Leftrightarrow \quad f(t_1) < f(t_2)$$

Complex preferences

The *prioritisation*, also known as *lexicographical order:*

$$a \,\&\, b = a \bowtie \top_b + 1_a \bowtie b$$

This means:

- Better w.r.t. a, and if equal w.r.t. a then better w.r.t. b

But does this meet the user expectation?

Complex preferences

The *prioritisation*, also known as *lexicographical order:*

$$a \,\&\, b = a \bowtie \top_b + 1_a \bowtie b$$

This means:

- Better w.r.t. a, and if equal w.r.t. a then better w.r.t. b

But does this meet the user expectation?

- For a being layered:
  - Incomparable tuples form equivalence classes
  - Instead of "equal w.r.t. a"

    $\longrightarrow$ "equal w.r.t. these equivalence classes"
- Formal basis: SV-Semantics (substitutable values)

Substitutable values

Definition (SV relation)

For $a :: T_a^2$ we call $s_a :: T_a^2$ an *SV relation for a*, if:

1. The relation s_a is an equivalence relation

Substitutable values

Definition (SV relation)

For $a :: T_a^2$ we call $s_a :: T_a^2$ an *SV relation for a*, if:

1. The relation s_a is an equivalence relation
2. s_a is compatible with a:
   1. $s_a \sqcap a = 0_a$,
   2. $s_a \cdot a \leq a$,
   3. $a \cdot s_a \leq a$.

Default SV relation: $s_a = 1_a$.

Substitutable values

Definition (SV relation)

For $a :: T_a^2$ we call $s_a :: T_a^2$ an *SV relation for a*, if:

1. The relation s_a is an equivalence relation
2. s_a is compatible with a:
   1. $s_a \sqcap a = 0_a$,
   2. $s_a \cdot a \leq a$,
   3. $a \cdot s_a \leq a$.

Default SV relation: $s_a = 1_a$.

Lemma

If $a :: T^2$ is a layered preference then $s_a = \overline{a + a^{-1}}$ is an SV relation.

Complex preferences

Definition (Prioritisation and Pareto composition with SV)

For $a :: T_a^2$ and $b :: T_b^2$ with SV relations $s_a :: T_a^2$ and $s_b :: T_b^2$:

- Prioritisation:

$$a \,\&\, b :: (T_a \bowtie T_b)^2$$

$$a \,\&\, b = a \bowtie \top_b \;+\; s_a \bowtie b$$

- Pareto composition:

$$a \otimes b :: (T_a \bowtie T_b)^2$$

$$a \otimes b = a \bowtie (s_b + b) \;+\; (s_a + a) \bowtie b$$

We say that $a \,\&\, b$ or $a \otimes b$ is *SV-preserving* if

$$s_{a\&b} = s_a \bowtie s_b \quad \text{or} \quad s_{a\otimes b} = s_a \bowtie s_b$$

Maximal elements

- The preference selection returns maximal elements!

Definition (Maximal elements)

For $a :: T^2$ and a set $p :: T$ we define

$$a \rhd p =_{df} p - \langle a \rangle\, p \,.$$

- $\langle a \rangle\, p$ consists of all elements having an a-successor.

Maximal elements

- The preference selection returns maximal elements!

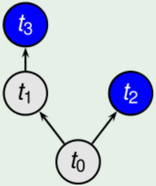
Definition (Maximal elements)

For $a :: T^2$ and a set $p :: T$ we define

$$a \rhd p =_{df} p - \langle a \rangle p \,.$$

- $\langle a \rangle p$ consists of all elements having an a-successor.

Example



- Let $p = t_0 + \ldots + t_3$
- $\langle a \rangle p = t_0 + t_1$
- $a \rhd p = t_2 + t_3$

A first example

Example

Consider the following dataset r and preference a:

Model	**Fuel**	**Power**	**Color**
BMW 5	11.4	230	silver
Mercedes E	12.1	275	black
Audi 6	12.7	225	red

$$a = \underbrace{(\mathrm{LOWEST}(\mathit{fuel}) \otimes \mathrm{HIGHEST}(\mathit{power}))}_{b} \,\&\, \underbrace{\mathrm{POS}(\mathit{color}, \{\mathrm{black}\})}_{c}$$

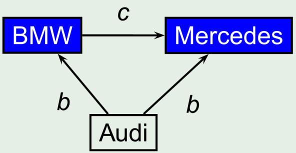
A first example

Example

Consider the following dataset r and preference a:

Model	Fuel	Power	Color
BMW 5	11.4	230	silver
Mercedes E	12.1	275	black
Audi 6	12.7	225	red

$$a = \underbrace{(\text{LOWEST}(\mathit{fuel}) \otimes \text{HIGHEST}(\mathit{power}))}_{b} \,\&\, \underbrace{\text{POS}(\mathit{color}, \{\text{black}\})}_{c}$$



- $s_b = 1_b$: $\quad a \triangleright r = (\text{BMW}) + (\text{Mercedes})$

A first example

Example

Consider the following dataset r and preference a:

Model	**Fuel**	**Power**	**Color**
BMW 5	11.4	230	silver
Mercedes E	12.1	275	black
Audi 6	12.7	225	red

$$a = \underbrace{(\mathrm{LOWEST}(\mathit{fuel}) \otimes \mathrm{HIGHEST}(\mathit{power}))}_{b} \,\&\, \underbrace{\mathrm{POS}(\mathit{color}, \{\mathrm{black}\})}_{c})$$

$$\begin{tikzcd}
\boxed{\mathrm{BMW}} \arrow[rr, "c"] & & \boxed{\mathrm{Mercedes}} \\
& \boxed{\mathrm{Audi}} \arrow[ul, "b"] \arrow[ur, "b"'] &
\end{tikzcd}$$

- $s_b = 1_b$: $\quad a \triangleright r = (\mathrm{BMW}) + (\mathrm{Mercedes})$
- Assume $s_b = \overline{b + b^{-1}}$

$\Rightarrow$ $a \triangleright r = (\mathrm{Mercedes})$

Pareto: Not a weak order

Example

- Let $a :: A^2, b :: B^2$ with $D_A = D_B = \{0, 1, 2\}$ be the $<$-order on $\mathbb{N}$
- Consider the incomparability relation $s_{\text{inc}} =_{df} \overline{(a \otimes b) + (a \otimes b)^{-1}}$.

$$\begin{tikzcd}
(2,0) \arrow[dr, leftrightarrow, "s_{\text{inc}}"] & \\
(2,1) \arrow[u, "a \otimes b"] \arrow[r, leftrightarrow, "s_{\text{inc}}"'] & (1,2)
\end{tikzcd}$$

$\Rightarrow$ s_{inc} is not transitive

$\Rightarrow$ It is no equivalence relation, hence no SV relation

Transforming general preferences to weak orders

- $a \otimes b$ is in general not layered!
- Can we construct a layered preference from it?

Transforming general preferences to weak orders

- $a \otimes b$ is in general not layered!
- Can we construct a layered preference from it?

The strategy: For a dataset r and a preference a we calculate:

- The maxima set: $q_0 = a \rhd r$
- The remainder: $r_1 = r - q_0$
- The maxima therein: $q_1 = a \rhd r_1$, ...

$\Rightarrow$ This yields a layered preference by construction

Transforming general preferences to weak orders

- $a \otimes b$ is in general not layered!
- Can we construct a layered preference from it?

The strategy: For a dataset r and a preference a we calculate:

- The maxima set: $q_0 = a \rhd r$
- The remainder: $r_1 = r - q_0$
- The maxima therein: $q_1 = a \rhd r_1$, ...

$\Rightarrow$ This yields a layered preference by construction

Definition (Layer-i Elements)

For $i = 0, 1, 2, \ldots$ we define the tests q_i and r_i:

$$q_i =_{df} a \rhd r_i \quad \text{where} \quad r_i =_{df} r - \sum_{j=0}^{i-1} q_j \ .$$

By convention, the empty sum is 0_a.

Visualisation

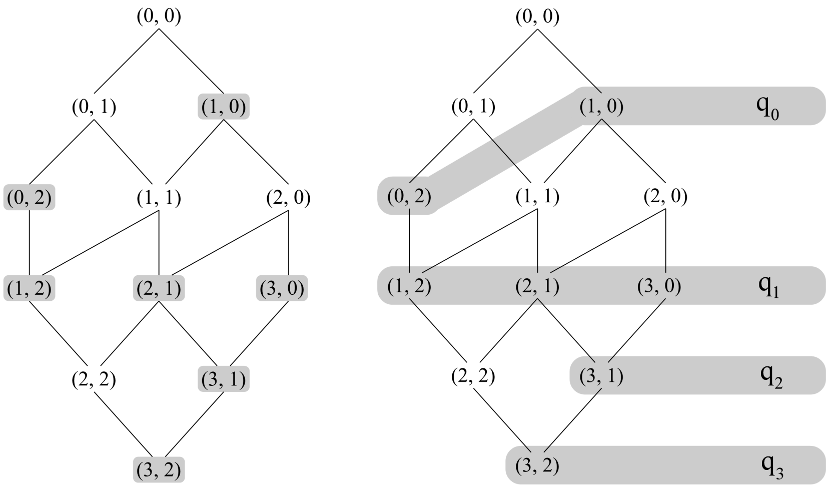


Figure: Visualisation for a Pareto preference on $[0, 3] \times [0, 2]$ (Preisinger09)

Properties of Iterated Maxima

- The q_i are calculated recursively:

$$q_i =_{df} a \rhd r_i \text{ where } r_i =_{df} r - \sum_{j=0}^{i-1} q_j \ .$$

- Is there a non-recursive formula for the q_i?

Properties of Iterated Maxima

- The q_i are calculated recursively:

$$q_i =_{df} a \rhd r_i \text{ where } r_i =_{df} r - \sum_{j=0}^{i-1} q_j \ .$$

- Is there a non-recursive formula for the q_i?

Lemma (Closed formula for layer-i elements)

For $i \in \mathbb{N}$ we have:

1. $(ra)^{i+1} \leq (ra)^i$
2. $\langle (ra)^{i+1} \rangle r \leq \langle (ra)^i \rangle r$,
3. $r_i = \langle (ra)^i \rangle r$.

Lemma

1. *Let r be finite. Then the calculation of the r_i becomes stationary, i.e.*

$$\exists N \in \mathbb{N} \text{ with } N = \max\{k \in \mathbb{N} \mid r_k \neq 0_a\}$$

2. *The q_i form a partition:*
    - *The q_i cover r, i.e., $\sum_{i=0}^{N} q_i = r$.*
    - *The q_i are pairwise disjoint, i.e., for $i \neq j$ we have $q_i \cdot q_j = 0_a$.*

Induced layered preference

Definition (Induced layered preference)

Let a be a preference and r a basic set, q_i and N as before. We define:

$$b_{ij} = q_i \cdot \top_a \cdot q_j \text{ for } i, j \in [0, N]$$

and the *induced layered preference* $m(a, r) :: T_a[r]^2$

$$m(a, r) =_{df} \sum_{i>j} b_{ij}$$

$T_a[r]$ is a sub-type of T_a with identity r and greatest element $r \cdot \top_a \cdot r$.

Induced layered preference

Definition (Induced layered preference)

Let a be a preference and r a basic set, q_i and N as before. We define:

$$b_{ij} = q_i \cdot \top_a \cdot q_j \text{ for } i, j \in [0, N]$$

and the *induced layered preference* $m(a, r) :: T_a[r]^2$

$$m(a, r) =_{df} \sum_{i>j} b_{ij}$$

$T_a[r]$ is a sub-type of T_a with identity r and greatest element $r \cdot \top_a \cdot r$.

A corresponding SV relation $s_{m(a,r)}$ is defined as

$$s_{m(a,r)} =_{df} \sum_{i} b_{ii} .$$

Well-definedness and useful properties

Lemma (Well-definedness)

1. *The relation $m(a, r)$ from the previous definition is a layered preference.*
2. $s_{m(a,r)}$ *is an SV relation for* $m(a, r)$.

Well-definedness and useful properties

Lemma (Well-definedness)

1. *The relation $m(a,r)$ from the previous definition is a layered preference.*
2. *$s_{m(a,r)}$ is an SV relation for $m(a,r)$.*

Lemma (Useful properties)

- *The original preference is still contained in $m(a,r)$:*

$$r \cdot a \cdot r \leq m(a,r)$$

- *The induced SV relation is part of the incomparability relation:*

$$s_{m(a,r)} \leq r \cdot \overline{(a + a^{-1})} \cdot r$$

Proof of the well-definedness Lemma

Proof (Well-definedness).

- Strict order property of $m(a, r)$ is quite clear
- We show negative transitivity of $m(a, r)$:

$$\left(\overline{m(a,r)}\right)^2 = \left(\sum_{i \leq j} b_{ij}\right) \cdot \left(\sum_{k \leq l} b_{kl}\right) = \sum_{i \leq j \leq l} b_{ij} \cdot b_{jl} \leq \sum_{i \leq l} b_{il} = \overline{m(a,r)}$$

Proof of the well-definedness Lemma

Proof (Well-definedness).

- Strict order property of $m(a, r)$ is quite clear
- We show negative transitivity of $m(a, r)$:

$$\left(\overline{m(a,r)}\right)^2 = \left(\sum_{i \le j} b_{ij}\right) \cdot \left(\sum_{k \le l} b_{kl}\right) = \sum_{i \le j \le l} b_{ij} \cdot b_{jl} \le \sum_{i \le l} b_{il} = \overline{m(a,r)}$$

- We show that $s_{m(a,r)}$ is the incomparability relation of $m(a, r)$:

$$\overline{m(a,r) + m(a,r)^{-1}} = \overline{\sum_{i>j} b_{ij} + \sum_{i<j} b_{ij}} = \overline{\sum_{i \ne j} b_{ij}} = \sum_{i} b_{ii} = s_{m(a,r)}$$

- This shows that $s_{m(a,r)}$ is an SV relation (by a previous lemma)

□

Application: Pareto-regular preference

- We apply $m(...)$ to the Pareto preference
- This yields a weak order
- *"regular"*: SV relation is the incomparability relation

Application: Pareto-regular preference

- We apply $m(\ldots)$ to the Pareto preference
- This yields a weak order
- *"regular"*: SV relation is the incomparability relation

Definition (Pareto-regular preference)

Let $a :: T_a^2$, $b :: T_b^2$ and $r :: T_a \bowtie T_b$.

$$a \otimes_{\text{reg}} b :: (T_a \bowtie T_b)^2$$
$$a \otimes_{\text{reg}} b = m(a \otimes b, r)$$
$$s_{a \otimes_{\text{reg}} b} = s_{m(a \otimes b, r)} \qquad \left(= \overline{(a \otimes_{\text{reg}} b) + (a \otimes_{\text{reg}} b)^{-1}} \right)$$

The difference in practice

Example

Consider again the following dataset r and preference a:

Model	**Fuel**	**Power**	**Color**
BMW 5	11.4	230	silver
Mercedes E	12.1	275	black
Audi 6	12.7	225	red

$$a = \underbrace{(\text{LOWEST}(\mathit{fuel}) \otimes_{\text{reg}} \text{HIGHEST}(\mathit{power}))}_{b} \,\&\, \underbrace{\text{POS}(\mathit{color}, \{\text{black}\})}_{c}$$

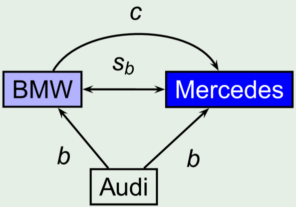
The difference in practice

Example

Consider again the following dataset *r* and preference *a*:

Model	Fuel	Power	Color
BMW 5	11.4	230	silver
Mercedes E	12.1	275	black
Audi 6	12.7	225	red

$$a = \underbrace{\left(\mathrm{LOWEST}(fuel) \otimes_{\mathrm{reg}} \mathrm{HIGHEST}(power)\right)}_{b} \,\&\, \underbrace{\mathrm{POS}(color, \{\mathrm{black}\})}_{c}$$



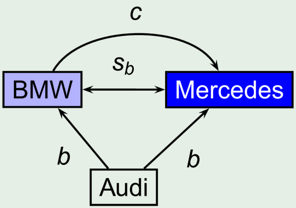
The difference in practice

Example

Consider again the following dataset r and preference a:

Model	**Fuel**	**Power**	**Color**
BMW 5	11.4	230	silver
Mercedes E	12.1	275	black
Audi 6	12.7	225	red

$$a = \underbrace{(\text{LOWEST}(\mathit{fuel}) \otimes_{\text{reg}} \text{HIGHEST}(\mathit{power}))}_{b} \,\&\, \underbrace{\text{POS}(\mathit{color}, \{\text{black}\})}_{c}$$



$\Rightarrow$ (Mercedes) and (BMW) are equivalent according to s_b

$\Rightarrow$ Preference c decides for (Mercedes)

$\Rightarrow$ $a \rhd r$ = (Mercedes)

Aspects of implementation

- $\otimes_{\text{reg}}$ is **not** associative!

Aspects of implementation

- $\otimes_{\text{reg}}$ is **not** associative!
- We decided to implement a *regularised prioritisation* $\&_{\text{reg}}$:

$$a \,\&_{\text{reg}}\, b \;=_{df}\; m(a, r) \,\&\, b$$

- Thus we have

$$(a \otimes b) \,\&_{\text{reg}}\, c = (a \otimes_{\text{reg}} b) \,\&\, c$$

Aspects of implementation

- $\otimes_{\text{reg}}$ is **not** associative!
- We decided to implement a *regularised prioritisation* $\&_{\text{reg}}$:

$$a \,\&_{\text{reg}}\, b =_{df} m(a, r) \,\&\, b$$

- Thus we have

$$(a \otimes b) \,\&_{\text{reg}}\, c = (a \otimes_{\text{reg}} b) \,\&\, c$$

- For the calculation of maxima:

$$(a \,\&_{\text{reg}}\, b) \triangleright r = b \triangleright a \triangleright r$$

Aspects of implementation

- $\otimes_{\text{reg}}$ is **not** associative!
- We decided to implement a *regularised prioritisation* $\&_{\text{reg}}$:

$$a \,\&_{\text{reg}}\, b =_{df} m(a, r) \,\&\, b$$

- Thus we have

$$(a \otimes b) \,\&_{\text{reg}}\, c = (a \otimes_{\text{reg}} b) \,\&\, c$$

- For the calculation of maxima:

$$(a \,\&_{\text{reg}}\, b) \triangleright r = b \triangleright a \triangleright r$$

- Note that for MAX-Queries (i.e. $(\ldots) \triangleright (\ldots)$) only q_0 is relevant
- For TOP-k querys the situation is more complex!

Conclusion and Outlook

What was done in this paper:

- Extended our calculus to preferences with SV-Semantics
- Introduced the Pareto-regular preference
- Point-free proofs for useful properties of it

Conclusion and Outlook

What was done in this paper:

- Extended our calculus to preferences with SV-Semantics
- Introduced the Pareto-regular preference
- Point-free proofs for useful properties of it

This work is part of a larger project:

- An advanced formalisation of “preference algebra”
- A toolbox for constructing preference evaluation algorithms
- A comprehensive algebraic description of “preference algebra”

Conclusion and Outlook

What was done in this paper:

- Extended our calculus to preferences with SV-Semantics
- Introduced the Pareto-regular preference
- Point-free proofs for useful properties of it

This work is part of a larger project:

- An advanced formalisation of "preference algebra"
- A toolbox for constructing preference evaluation algorithms
- A comprehensive algebraic description of "preference algebra"

The next steps:

- Formalising projections, e.g. $(a \bowtie b)|_{T_a} = a$
- Applying the calculus at a larger scale using machine assistance