The Θ-Join as a Join with Θ

Jules Desharnais and Bernhard Möller

RAMiCS 2020

Context

- preference queries in data bases, where users may express that they like certain results better than others
- here: an algebraic calculus for (sets) of tuples, queries and constructors for preference relations
- reasons: more compact than predicate logic and suitable for automatic verification of optimisation rules
- started in 2011
- several publications since (with Markus Endres and Patrick Roocks, a.o. RAMiCS 2012, Dissertation by Patrick 2016)
- but a satisfactory treatment of Θ-joins was still missing

Preliminaries

Definition

- *type* T: set of attributes, e.g., `Name, Age, Salary`
- T-tuple: function t from attributes $A \in T$ to *domains* D_A
- $D_T = \prod\limits_{A \in T} D_A$: set of all T-tuples
- $t :: T \Leftrightarrow_{df} t \in D_T$ and $P :: T \Leftrightarrow_{df} P \subseteq D_T$
- *projection* $\pi_{T'}(t)$ of t to T' is $t|_{T'}$
- $t_i :: T_i \ (i = 1, 2)$ *matching* iff $\pi_{T_1 \cap T_2}(t_1) = \pi_{T_1 \cap T_2}(t_2)$
- in this case $t_1 \bowtie t_2 \ =_{df} t_1 \cup t_2$
- if $T_1 \cap T_2 = \emptyset$ then the t_i trivially are matching

(database) tables: sets of tuples with equal types

Definition *(inner) join* of tables $P_i :: T_i$ $(i = 1, 2)$:

$$P_1 \bowtie P_2 =_{df} \{t_1 \bowtie t_2 \,|\, t_i \in P_i\}$$

set of all combinations of matching P_i-tuples

Lemma

- $\bowtie$ is associative, commutative and distributes over $\cup$
- hence also isotone w.r.t. $\subseteq$
- has unit $\{\emptyset\}$

we take $\bowtie$ as our central operator, rather than $\times$ and σ

The Θ-Join

- assume tables $P :: T_P, Q :: T_Q$ with $T_P \cap T_Q = \emptyset$
- consider attributes $A \in T_P, B \in T_Q$ and a binary relation $\Theta \subseteq D_A \times D_B$, such as $=, \leq, >$
- then Θ provides "glue" between the type-disjoint P and Q
- want to model the classical database query expression

$$\text{"}P \bowtie_{\Theta(P.A,Q.B)} Q\text{"}$$

- result should be the table $\{t \in P \bowtie Q \,|\, t(A)\,\Theta\,t(B)\}$
- central idea of our approach: consider Θ mathematically as another table, of type $\{A, B\}$
- then the above expression algebraically simply reads

$$P \bowtie \Theta \bowtie Q$$

- this explains the title of our paper

Example

P_1:

Name1	Age1
A	50
B	55
C	60

$<$:

Age1	Age2
50	55
50	60
55	60

P_2:

Age2	Name2
50	E
55	F
55	G

$P_1 \bowtie P_2 = P_1 \times P_2$:

Name1	Age1	Age2	Name2
A	50	50	E
A	50	55	F
A	50	55	G
B	55	50	E
B	55	55	F
B	55	55	G
C	60	50	E
C	60	55	F
C	60	55	G

The Θ-Join

P_1:

Name1	Age1
A	50
B	55
C	60

$P_1 \bowtie <$:

Name1	Age1	Age2
A	50	55
A	50	60
B	55	60

$<$:

Age1	Age2
50	55
50	60
55	60

$< \bowtie P_2$:

Age1	Age2	Name2
50	55	F
50	55	G

P_2:

Age2	Name2
50	E
55	F
55	G

$P_1 \bowtie < \bowtie P_2$:

Name1	Age1	Age2	Name2
A	50	55	F
A	50	55	G

□

two standard optimisation rules:

Theorem

1. if $Q :: L \subseteq T_P$ then $\pi_L(P \bowtie Q) = \pi_L(P) \bowtie Q$
2. ["Push projection over Θ-join"] for $T_P \cap T_Q = \emptyset$ and $\Theta :: L$ for some $L \subseteq T_P \cup T_Q$

$$\pi_L(P \bowtie \Theta \bowtie Q) = \pi_{L_P}(P) \bowtie \Theta \bowtie \pi_{L_Q}(Q)$$

   where $L_P =_{df} T_P \cap L$ and $L_Q =_{df} T_Q \cap L$

Proof of Part 2:

$$
\begin{array}{ll}
 & \pi_L(P \bowtie \Theta \bowtie Q) \\
= & \quad \{\!\![\text{ associativity and commutativity of } \bowtie \text{ }]\!\!\} \\
 & \pi_L(P \bowtie Q \bowtie \Theta) \\
= & \quad \{\!\![\text{ Part 1 }]\!\!\} \\
 & \pi_L(P \bowtie Q) \bowtie \Theta \\
= & \quad \{\!\![\text{ assumption } T_P \cap T_Q = \emptyset \text{ }]\!\!\} \\
 & \pi_{L_P}(P) \bowtie \pi_{L_Q}(Q) \bowtie \Theta \\
= & \quad \{\!\![\text{ associativity and commutativity of } \bowtie \text{ }]\!\!\} \\
 & \pi_{L_P}(P) \bowtie \Theta \bowtie \pi_{L_Q}(Q)
\end{array}
$$

□

Selection as Join

- representation of Θ-join as join with Θ has proved useful
- we treat selection $\sigma_C(P)$ for table P and condition C analogously
- *condition*: predicate on tuples
- simply represented as subset $C :: D_L$ for some type L
- conjunction and disjunction of $C, C' :: L$ represented by $C \bowtie C'$ and $C \cup C'$
- for $P :: T$ and $C :: D_L$ with $L \subseteq T$ now just set

$$\sigma_C(P) =_{df} P \bowtie C$$

Lemma assume again $P :: T$

1. selections commute, i.e., $\sigma_C(\sigma_{C'}(P)) = \sigma_{C'}(\sigma_C(P))$
2. selections can be combined, i.e., $\sigma_C(\sigma_{C'}(P)) = \sigma_{C \bowtie C'}(P)$
3. if C depends only on attributes from $L \subseteq T$, i.e., $C \subseteq D_L$, then $\pi_L(\sigma_C(P)) = \sigma_C(\pi_L(P))$

Proof

1. immediate from associativity/commutativity of $\bowtie$
2. ditto
3. by definitions, previous theorem and definitions again:

$$\pi_L(\sigma_C(P)) = \pi_L(P \bowtie C) = \pi_L(P) \bowtie C = \sigma_C(\pi_L(P))$$

$\square$

The Join of Binary Relations

now we turn to *preference relations*, which are (homogeneous) binary strict-order relations between tuples

"better" values at the right

Example we want to model the sentence "I prefer cars that are green and, equally important, have low fuel consumption" □

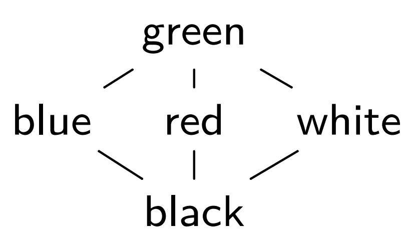
this needs a new combination operator on binary relations R_1, R_2

Definition *join* $R_1 \bowtie R_2$ given by

$$t\,(R_1 \bowtie R_2)\,u \quad \Leftrightarrow_{df} \quad \pi_{T_1}(t)\,R_1\,\pi_{T_1}(u) \;\wedge\; \pi_{T_2}(t)\,R_2\,\pi_{T_2}(u)$$

Example we treat the above-mentioned simple database of cars

- attribute set: $\{\texttt{Col}, \texttt{Fuel}\}$ with
  - $D_{\texttt{Col}} = \{\text{black}, \text{blue}, \text{green}, \text{red}, \text{white}\}$
  - $D_{\texttt{Fuel}} = \{4.0, 4.1, \ldots, 9.9, 10.0\}$.
- $R_{\texttt{Fuel}} =_{df} >$ and $R_{\texttt{Col}}$ given by the Hasse diagram



- a user with preference $R_{\texttt{Col}}$ does not like black at all, likes green best and otherwise is indifferent about blue, red, white
- hence $s\,(R_{\texttt{Col}} \bowtie R_{\texttt{Fuel}})\,t$ iff the colour of t is closer to green and the fuel value of t less than the corresponding values of s $\square$

now for some laws

Definition R_1, R_2 *weakly matching* if for all matching $x_i \in \ulcorner R_i$ there are matching y_i with $x_i \, R_i \, y_i$

Lemma

1. relations with disjoint domain (and hence codomain) types are weakly matching
2. domain is subdistributive: $\ulcorner(R_1 \bowtie R_2) \subseteq \ulcorner R_1 \bowtie \ulcorner R_2$
   for weakly matching R_i this becomes an equality
3. inclusional interchange law
   $$(R_1 \bowtie R_2) \,;\, (S_1 \bowtie S_2) \subseteq (R_1 \,;\, S_1) \bowtie (R_2 \,;\, S_2)$$
   for *strongly matching* R_i (see paper) this becomes an equality

matching relaxes the disjointness condition of earlier papers

Inverse Image and Maximal Elements

Definition *inverse image* of table P under relation R

$$|R\rangle P =_{df} \{t \mid \exists u \in P : t\, R\, u\} = \ulcorner(R \,;\, P)$$

this (forward) diamond operator cooperates well with join

Lemma inclusional interchange law

$$|R_1 \bowtie R_2\rangle(P_1 \bowtie P_2) \subseteq |R_1\rangle P_1 \bowtie |R_2\rangle P_2$$

for weakly matching $R_i \,;\, P_i$ this becomes an equality preference queries should return the maximal elements w.r.t. the employed preference relation

- set of *R-maximal* tuples within table P:

$$R \triangleright P \;=_{df}\; P - |R\rangle P$$

- these are the P-tuples without an R-successor in P, i.e., without a properly better tuple in P
- two laws: $\emptyset \triangleright P = P$ $\quad\quad$ $\mathsf{T} \triangleright P = \emptyset$

join and maximum interact as follows

Lemma for tables P, Q and relations R, S such that $R\,;P$ and $S\,;Q$ are weakly matching,

$$(R \bowtie S) \rhd (P \bowtie Q) = (R \rhd P) \bowtie Q \;\cup\; P \bowtie (S \rhd Q)$$

Corollary for tables P, Q and relation R such that $R\,;P$ and $\mathsf{T}_Q\,;Q$ are weakly matching, Then ("push preference over join")

$$(R \bowtie \mathsf{T}_Q) \rhd (P \bowtie Q) = (R \rhd P) \bowtie Q$$

term $R \bowtie \mathsf{T}_Q$ expresses that the user does not care about the attributes in T_Q and is only interested in maxima w.r.t. the T_P part how to guarantee weak matching?

Definition call table P *joinable* with table Q iff for every $p \in P$ there is a $q \in Q$ such that p and q match.

this means that every tuple in P has a join partner in Q

Lemma if P is joinable with Q and R is a relation then $R\,;P$ and $\mathsf{T}_Q\,;Q$ are weakly matching what to do if a theta join is involved?

fortunately, by our treatment of Θ-joins, nothing extra!

Corollary if P and $\Theta \bowtie Q$ are joinable then

$$(R \bowtie \mathsf{T}_{\Theta \bowtie Q}) \triangleright (P \bowtie \Theta \bowtie Q) = (R \triangleright P) \bowtie \Theta \bowtie Q$$

without joinability the property need not hold, see paper

How are Proofs Actually Done?

- define relations *split* $\prec$ and its converse *glue* $\succ$ by

$$t \prec (t_1, t_2) \;\Leftrightarrow_{df}\; (t_1, t_2) \succ t \;\Leftrightarrow_{df}\; t_1 = \pi_{T_1}(t) \wedge t_2 = \pi_{T_2}(t)$$

- *parallel product* (tensor product) of relations R_i given by

$$(t_1, t_2)\,(R_1 \times R_2)\,(u_1, u_2) \Leftrightarrow_{df} t_1\,R_1\,u_1 \wedge t_2\,R_2\,u_2$$

- then the join of the R_i can be expressed point-free as

$$R_1 \bowtie R_2 \;=_{df} \prec ; (R_1 \times R_2) ; \succ$$

- now everything works smoothly

Outlook

- treatment here largely in the concrete algebra of binary relations
- in the mentioned papers this had been abstracted to the more general framework of *join algebras* based on modal semirings
- that has to be carried over to the case of Θ-joins