Polymorphic type inference for the relational algebra

Jan Van den Bussche (Limburg)
Emmanuel Waller (Paris-Sud)

The relational algebra

Operations on tables (relational databases)

A	B
a	b
c	d
e	f

$\cup$

A	B
a	c
c	d

$=$

A	B
a	b
c	d
e	f
a	c

A	B
a	b
c	d
e	f

$-$

A	B
a	c
c	d

$=$

A	B
a	b
e	f

$$\begin{array}{|cc|}\hline A & B \\ \hline a & b \\ \hline\end{array} \times \begin{array}{|cc|}\hline C & D \\ \hline c & d \\ e & f \\ \hline\end{array} = \begin{array}{|cccc|}\hline A & B & C & D \\ \hline a & b & c & d \\ a & b & e & f \\ \hline\end{array}$$

$$\begin{array}{|cc|}\hline A & B \\ \hline a & b \\ d & e \\ \hline\end{array} \bowtie \begin{array}{|cc|}\hline B & C \\ \hline b & c \\ e & f \\ \hline\end{array} = \begin{array}{|ccc|}\hline A & B & C \\ \hline a & b & c \\ d & e & f \\ \hline\end{array}$$

$$\sigma_{A=B} \begin{array}{|ccc|}\hline A & B & C \\ \hline a & a & b \\ c & d & c \\ \hline\end{array} = \begin{array}{|ccc|}\hline A & B & C \\ \hline a & a & b \\ \hline\end{array}$$

$$\pi_{A,C} \begin{array}{|ccc|}\hline A & B & C \\ \hline a & a & b \\ c & d & c \\ \hline\end{array} = \begin{array}{|cc|}\hline A & C \\ \hline a & b \\ c & c \\ \hline\end{array}$$

$$\rho_{C/D} \begin{array}{|ccc|}\hline A & B & C \\ \hline a & a & b \\ c & d & c \\ \hline\end{array} = \begin{array}{|ccc|}\hline A & B & D \\ \hline a & a & b \\ c & d & c \\ \hline\end{array}$$

Constraints on types of operands

Type: set of attributes ("relation schema")

Relations r, r_1, r_2 of types τ, τ_1, τ_2, then:

$r_1 \cup r_2$ or $r_1 - r_2$ only if $\tau_1 = \tau_2$

$r_1 \times r_2$ only if $\tau_1 \cap \tau_2 = \varnothing$

$\sigma_{\theta(A_1,\ldots,A_p)}(r)$ or $\pi_{A_1,\ldots,A_p}(r)$ only if each $A_i \in \tau$

$\rho_{C/D}(r)$ only if $C \in \tau$, $D \notin \tau$

Relational algebra expressions

$$
\begin{array}{rcl}
e & \rightarrow & r \\
 & | & e \cup e \\
 & | & e - e \\
 & | & e \times e \\
 & | & e \bowtie e \\
 & | & \sigma_{\theta(A_1,\ldots,A_p)}(e) \\
 & | & \pi_{A_1,\ldots,A_p}(e) \\
 & | & \rho_{A/B}(e)
\end{array}
$$

$$\sigma_{A<5}(r \bowtie s) \bowtie ((r \times u) - v)$$

Typing rules

$$\frac{\mathcal{T}(r) = \tau}{\mathcal{T} \vdash r : \tau}$$

$$\frac{\mathcal{T} \vdash e_1 : \tau \quad \mathcal{T} \vdash e_2 : \tau}{\mathcal{T} \vdash (e_1 \cup e_2) : \tau}$$

$$\frac{\mathcal{T} \vdash e_1 : \tau_1 \quad \mathcal{T} \vdash e_2 : \tau_2}{\mathcal{T} \vdash (e_1 \bowtie e_2) : \tau_1 \cup \tau_2}$$

$$\frac{\mathcal{T} \vdash e_1 : \tau_1 \quad \mathcal{T} \vdash e_2 : \tau_2 \quad \tau_1 \cap \tau_2 = \varnothing}{\mathcal{T} \vdash (e_1 \times e_2) : \tau_1 \cup \tau_2}$$

$$\frac{\mathcal{T} \vdash e : \tau \quad A_1, \ldots, A_n \in \tau}{\mathcal{T} \vdash \sigma_{\theta(A_1,\ldots,A_n)}(e) : \tau}$$

$$\frac{\mathcal{T} \vdash e : \tau \quad A_1, \ldots, A_n \in \tau}{\mathcal{T} \vdash \pi_{A_1,\ldots,A_n}(e) : \{A_1, \ldots, A_n\}}$$

$$\frac{\mathcal{T} \vdash e : \tau \quad A \in \tau \quad B \notin \tau}{\mathcal{T} \vdash \rho_{A/B}(e) : (\tau - \{A\}) \cup \{B\}}$$

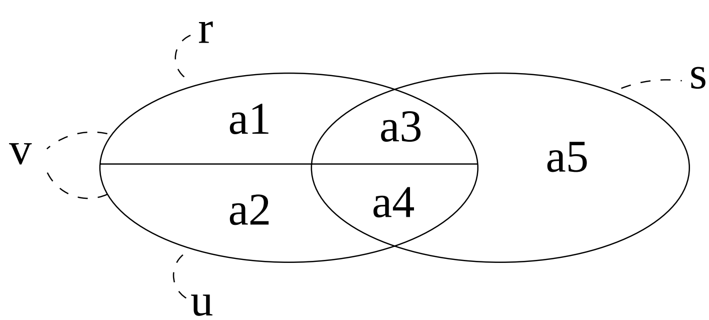
Well-typedness: e is *well-typed* under $\mathcal{T}$ if there exists τ such that $\mathcal{T} \vdash e : \tau$

Constraints on type assignments

$$\sigma_{A<5}(r \bowtie s) \bowtie ((r \times u) - v)$$

$\tau_r \cap \tau_u = \varnothing,\ \tau_r \cup \tau_u = \tau_v$

$(A \in \tau_r \cup \tau_s) \wedge (A \notin \tau_r \cap \tau_u) \wedge (A \in \tau_r \leftrightarrow A \in \tau_r \cup \tau_u)$



$r : a_1 a_3$
$u : a_2 a_4$
$v : a_1 a_2 a_3 a_4$
$s : a_3 a_4 a_5$
$A : \underbrace{(r \vee s) \wedge \neg(r \wedge u) \wedge (v \leftrightarrow (r \vee u))}_{\varphi}$

Polymorphic type inference

Add also description of output type:

$$\begin{array}{lcl} r : a_1 a_3 & & \\ u : a_2 a_4 & \mapsto & a_1 a_2 a_3 a_4 a_5 \\ v : a_1 a_2 a_3 a_4 & & \\ s : a_3 a_4 a_5 & & \\ A : \varphi & & A : \textbf{true} \end{array}$$

"Principal type formula"

⇒ *Polymorphic type inference problem* for the relational algebra:

Input: Relational algebra expression e

Output: Principal type formula for e

We have a complete algorithm, implemented

Some more examples

$$\pi_A(r) - \pi_A((\pi_A(r) \times s) - r)$$

$r : a$	$\mapsto$	$\varnothing$
$s : a$		
$A : r \wedge \neg s$		A : **true**

$$\rho_{A/B}(r) \times s$$

$r : a_1$	$\mapsto$	$a_1 a_2$
$s : a_2$		
$A : r$		$A : s$
$B : \neg r \wedge \neg s$		B : **true**

$$\sigma_{A=B}\pi_{B,C}(r)$$

untypeable!

Unification of Venn diagrams

$$(r \bowtie s) \bowtie ((r \times u) - v)$$

$r \bowtie s$	$\bowtie$	$(r \times u) - v$
$r : a_1a_2$		$r : b_1$
$s : a_2a_3$		$u : b_2$
		$v : b_1b_2$

$r : c_1c_3$
$u : c_2c_4$
$v : c_1c_2c_3c_4$
$s : c_3c_4c_5$

Polynomial algorithm

Setting the attribute constraints

$$(\pi_A(r) \bowtie s) \bowtie \pi_C \sigma_{B<5}(u)$$

$$\pi_A(r) \bowtie s$$

$$\begin{array}{lcl} r : a_1 a_2 & \mapsto & a_2 a_3 \\ s : a_2 a_3 & & \\ A : r & & A : \textbf{true} \\ B : \textbf{true} & & B : s \end{array}$$

Lemma: If a type variable is part of the output type, then it is part of the declaration of a relation variable whose complete declaration is part of the output type.

Discussion

Monadic first-order logic

Exactly to which set-operations is our algorithm applicable?

Complexity is exponential, but polynomial in output size, and implementation runs fast

Pure decision problem (typability) is in NP, probably NP-complete

Polymorphic expressive power

If r of type $\{A, B\}$ and s of type $\{B, C\}$, then:

$$r \bowtie s \quad \equiv \quad \pi_{A,B,C}\sigma_{B=D}(r \times \rho_{B/D}(s))$$

Theorem: No *polymorphic* such simulation is possible

Proof: No expression without $\bowtie$ has the principal type of $r \bowtie s$

Similar theorems for other typical "derived" relational algebra operators ($\ltimes$, $\widehat{\pi}$, ...)

$\Rightarrow$ Polymorphic relational algebra?