Bounded Fixpoints for Complex Objects

Dan Suciu*

August 1993

*With thanks to Val Breazu-Tannen and Peter Buneman

Purpose: Desing a *robust* query language which:

- Works on complex objects
- Is in PTIME
- Can express *recursive queries* (at least $DATALOG^{\neg}$)

Plan:

- Introduce the *Nested Relational Algebra*
- Introduce the *Bounded Fixpoint*
- State the main result
- Sketch the proof

The Language

Types:

$$\tau ::= unit \mid b \mid \tau \times \tau \mid \{\tau\}$$

$b \in$ an unspecified set of **base types** ($nat, bool, string$, etc). $unit \stackrel{\text{def}}{=} \{()\}$.

Complex Objects

E.g. $x = \{(a, \{a, c\}), (b, \{\}), (c, \{a, b, c\})\}$
of type $\{char \times \{char\}\}$.

The Nested Relational Algebra

Formalism: from Breazu, Buneman and Wong.

Other names:

The Nested Algebra (Paredaens and Van Gucht)

The Algebra without Powerset (Abiteboul and Beeri), etc.

- Union ($\cup$), empty set ($\emptyset$), cross product ($\times$).

  ...

- $$\frac{}{\eta_\sigma : \sigma \to \{\sigma\}} \qquad \frac{}{\mu_\sigma : \{\{\sigma\}\} \to \{\sigma\}}$$

  $$\frac{f : \sigma \to \tau}{map(f) : \{\sigma\} \to \{\tau\}}$$

  $$\begin{array}{rcl} \eta(x) & \stackrel{\text{def}}{=} & \{x\} \\ \mu(\{x_1, \ldots, x_n\}) & \stackrel{\text{def}}{=} & x_1 \cup \ldots \cup x_n \\ map(f)(\{x_1, \ldots, x_n\}) & \stackrel{\text{def}}{=} & \{f(x_1), \ldots, f(x_n)\} \end{array}$$

- $$\frac{f \in \Sigma}{f : d_f \to c_f}$$ A given set of external functions

Facts About the Nested Relational Algebra

1. All queries are in PTIME
2. It is a conservative extension of the First Order queries. (Paredaens and van Gucht, Wong, Van den Bussche). Hence, it cannot express *recursive queries*, like transitive closure $tc : \{b \times b\} \to \{b \times b\}$.

Problem: Extend NRA such as to preserve (1), and to express recursive queries.

First attempt: Add a fixpoint, as for first order logic:

$$\frac{f : \sigma \times \{\tau\} \to \{\tau\}}{fix(f) : \sigma \to \{\tau\}}$$

$fix(f)(x) \stackrel{\text{def}}{=} \cup_{n \geq 0} y_n$, where:

$$\begin{aligned} y_0 &\stackrel{\text{def}}{=} \emptyset \\ y_{k+1} &\stackrel{\text{def}}{=} y_k \cup f(x, y_k) \end{aligned}$$

(inflationary semantics).

BUT: can express powerset, an exponential time (and space) query !

The Powerset

$$f : \{\sigma\} \times \{\{\sigma\}\} \to \{\{\sigma\}\}$$

$$f(x, Y) \stackrel{\text{def}}{=} \{\emptyset\} \cup map(\eta)(x) \cup map(\cup)(Y \times Y)$$

Then, $fix(f)(x) = powerset(x)$. More, $\mathcal{NRA} + fix =$ *the algebra (with powerset)* of Abiteboul and Beeri.

The Bounded Fixpoint (idea due to Peter Buneman)

$$\frac{f : \sigma \times \{\tau\} \to \{\tau\} \quad g : \sigma \to \{\tau\}}{bfix(f, g) : \sigma \to \{\tau\}}$$

Inflationary Semantics: $bfix(f, g)(x) \stackrel{\text{def}}{=} \cup_{n \geq 0}\, y_n$, where:

$$\begin{aligned} y_0 &\stackrel{\text{def}}{=} \emptyset \\ y_{k+1} &\stackrel{\text{def}}{=} y_k \cup f(x, y_k) \cap g(x) \end{aligned}$$

Partial Semantics: $bfix(f, g)(x) \stackrel{\text{def}}{=} y_n$, where:

$$\begin{aligned} y_0 &\stackrel{\text{def}}{=} \emptyset \\ y_{k+1} &\stackrel{\text{def}}{=} f(x, y_k) \cap g(x) \end{aligned}$$

and $y_{n+1} = y_n$.

When no external functions are present ($\Sigma = \emptyset$), at *flat types* (i.e. set height 1), $bfix$ and fix are equivalent.

Example Transitive closure:

$f : \{b \times b\} \times \{b \times b\} \rightarrow \{b \times b\}$,

$$f(x, y) \stackrel{\text{def}}{=} x \cup (x \circ y)$$

Then, $tc(x) = fix_i(f)(x) = fix_p(f)(x)$.

For $bfix$, take $g : \{b \times b\} \rightarrow \{b \times b\}$:

$$g(x) \stackrel{\text{def}}{=} (\Pi_1(x) \cup \Pi_2(x)) \times (\Pi_1(x) \cup \Pi_2(x))$$

Then $tc(x) = bfix_i(f, g)(x) = bfix_p(f, g)(x)$.

Bounding is “harmless” at flat types.

Main Result

Theorem 1 *The following properties hold:*

1. *Even with external functions ($\Sigma \neq \emptyset$), we have:*
   - $\mathcal{NRA}(\Sigma) + bfix_i \subseteq PTIME$
   - $\mathcal{NRA}(\Sigma) + bfix_p \subseteq PSPACE$
2.
   - $\mathcal{NRA} + bfix_i + order = PTIME$
   - $\mathcal{NRA} + bfix_p + order = PSPACE$

   (Does not follow directly from Immerman, Vardi's results).
3.
   - $\mathcal{NRA} + bfix_i$ *is a conservative extension of FO+LFP (First Order Logic with Least Fixpoints), i.e. of* $DATALOG^{\neg}$ *(with inflationary fixpoints).*
   - $\mathcal{NRA} + bfix_p$ *is a conservative extension of FO+ partial fixpoints, i.e. of* $DATALOG^{*,\neg}$*, i.e. of the while-queries.*

Proof of the Conservativity Result

Technique: **index type** I

$$\begin{array}{rcl} left & : & unit \to I \\ right & : & unit \to I \\ pair & : & I \times I \to I \text{ injective} \end{array}$$

1. Translate $\mathcal{NRA}(\Sigma) + bfix$ into $\mathcal{RA}(\Sigma \cup I) + bfix$ (i.e. the **relational algebra** extended with Σ, I, and $bfix$).
    - Translate **types** to **flat types** $\tau \leadsto \pi_\tau$.
    - Translate **functions** $f : \sigma \to \tau$ to $R_f : \pi_\sigma \to \pi_\tau$.
    - Encode **complex objects** $x : \tau$ by **flat relations** (with indexes) $r : \pi_\tau$.
2. When $f : \sigma \to \tau$, show how to eliminate the indexes from $R_f : \pi_\sigma \to \pi_\tau$. Write: $x \sim r$ (a one to many relation)

1. The Translation

Lemma 1 *$\pi = \{s_1\} \times \ldots \times \{s_k\}$ flat type. Then:*

$$[I \Rightarrow \pi] \stackrel{\text{def}}{=} \{I\} \times \{I \times s_1\} \times \ldots \times \{I \times s_k\}$$

can encode all partial finite functions $\psi : I \to \pi$ and, hence, all elements of $\{\pi\}$.

E.g. $\pi = \{s_1\} \times \{s_2\}$:

i_1
i_2
i_3
i_4

i_1	a
i_1	b
i_1	c
i_2	a
i_2	c

i_1	m
i_3	m
i_3	n

encodes the partial function:

$$\begin{array}{rcl}
i_1 & \mapsto & \left(\begin{array}{|c|}\hline a \\ \hline b \\ \hline c \\ \hline\end{array}, \begin{array}{|c|}\hline m \\ \hline\end{array}\right) \\
i_2 & \mapsto & \left(\begin{array}{|c|}\hline a \\ \hline c \\ \hline\end{array}, \emptyset\right) \\
i_3 & \mapsto & \left(\emptyset, \begin{array}{|c|}\hline m \\ \hline n \\ \hline\end{array}\right) \\
i_4 & \mapsto & \emptyset \\
i & \mapsto & \text{undefined, when } i \notin \{i_1, i_2, i_3, i_4\}
\end{array}$$

and, hence, encodes the complex object:

$$\{(\{a,b,c\},\{m\}),(\{a,c\},\emptyset),(\emptyset,\{m,n\}),\emptyset\}$$

Translation of Types, and Encoding of Complex Objects

- Base types: $\pi_b \stackrel{\text{def}}{=} \{b\}$.

  Values of base types: $x \sim \{x\}$.

- Product types: $\pi_{\sigma\times\tau} \stackrel{\text{def}}{=} \pi_\sigma \times \pi_\tau$.

  Values of product types: $(x, y) \sim (r, q)$ iff $x \sim r$ and $y \sim q$.

- Set types: $\pi_{\{\sigma\}} \stackrel{\text{def}}{=} [I \Rightarrow \pi_\sigma]$.

  Values of set types: $\{x_1, \ldots, x_n\} \sim r$ iff r encodes some finite, partial function $\psi : I \to \pi_\sigma$, and $\forall k = 1, n$, $\exists i \in I$ s.t. $x_k \sim \psi(i)$:

$$\begin{array}{|c|}\hline x_1 \\ \hline \ldots \\ \hline x_n \\ \hline \end{array} \sim \begin{array}{|c|c|}\hline i_1 & r_{11} \\ \cline{2-2} & \vdots \\ \cline{2-2} & r_{1n_1} \\ \hline \vdots & \vdots \\ \hline i_m & r_{m1} \\ \cline{2-2} & \vdots \\ \cline{2-2} & r_{mn_m} \\ \hline \end{array}$$

Translation of Functions

Lemma 2 *For any $f : \sigma \to \tau$ in $\mathcal{NRA}(\Sigma) + bfix$, there is some $R_f : \pi_\sigma \to \pi_\tau$ in $\mathcal{RA}(\Sigma \cup I) + bfix$, such that:*

- $\forall x, r,\ x \sim r \Rightarrow f(x) \sim R_f(r)$.

The interesting cases are:

Flatten (μ) ($\mu(\{x_1, \ldots, x_m\}) \stackrel{\text{def}}{=} x_1 \cup \ldots \cup x_m$) Take R_μ to be:

$$R_\mu\left(\begin{array}{|c|c|c|}\hline i_1 & i'_{11} & r_{11} \\ & \vdots & \vdots \\ & i'_{1n_1} & r_{1n_1} \\ \hline \vdots & \vdots & \\ \hline i_m & i'_{m1} & r_{m1} \\ & \vdots & \vdots \\ & i'_{mn_m} & r_{mn_m} \\ \hline\end{array}\right) \stackrel{\text{def}}{=} \begin{array}{|c|c|}\hline pair(i_1, i'_{11}) & r_{11} \\ & \vdots \\ pair(i_1, i'_{1n_1}) & r_{1n_1} \\ \hline \vdots & \vdots \\ \hline pair(i_m, i'_{m1}) & r_{m1} \\ & \vdots \\ pair(i_m, i'_{mn_m}) & r_{mn_m} \\ \hline\end{array}$$

Union ($\cup$) Cannot take $R_\cup \stackrel{\text{def}}{=} \cup$. Instead, translate $doubleton : \sigma \times \sigma \to \{\sigma\}$, $doubleton(x, y) \stackrel{\text{def}}{=} \{x, y\}$:

$$R_{doubleton}(r_1, r_2) \stackrel{\text{def}}{=} \begin{array}{|c|c|}\hline left & r_1 \\ \hline right & r_2 \\ \hline\end{array}$$

Then, $x \cup y = \mu(\{x, y\})$.

Map ($(map(f))$**)**

$$\frac{f : \sigma \to \tau}{map(f) : \{\sigma\} \to \{\tau\}}$$

We have $R_f : \pi_\sigma \to \pi_\tau$.

Take $R_{map(f)} : [I \Rightarrow \pi_\sigma] \to [I \Rightarrow \pi_\tau]$,

$$R_{map(f)} \stackrel{\text{def}}{=} [I \Rightarrow R_f]$$

(need induction on R_f).

Bounded Fixpoint ($bfix(f, g)$**)** More complicated than $bfix(R_f, R_g)$, because indexes in R_f and R_g have no connection: "Rename" those in R_f.

2. Elimination of Indexes

$f : \sigma \to \tau$, σ is flat $\Rightarrow$ use the elements of $x \in \sigma$ as indexes themselves. *pair* $\stackrel{\text{def}}{=}$ tuple concatenation.

Need a lot of work to keep the types right.

Conclusion

- Fills in a gap:

First Order Logic (= Relational Algebra)	FO with fixpoints (= $DATALOG^{\neg}$)
Nested Relational Algebra (= Nested Algebra, = Algebra w/o powerset = Strictly Safe Calculus)	?

  We propose: "?" = Nested Relational Algebra with bounded fixpoints.

- "Bounding" works for other kinds of iterations as well.

- Rather powerful proof technique: it is order independent, and suggests a implementation technique (could be used for flattening of *nested parallelism*).