A Query Language for NC

Dan Suciu  Val Breazu-Tannen
University of Pennsylvania
Motivation

- \( FO \) and equivalent formalisms accepted as theoretical core for query languages since '70.
- But \( FO \) not expressive enough: transitive closure, parity, etc., are not expressible in \( FO \).
- Add iterative constructs:
  - Fixpoints [Immerman86, Vardi82]:
    \[
    FO + LFP^+ \leq = PTIME
    \]
    \[
    FO + PFP^+ \leq = PSPACE
    \]
    \( \Rightarrow \) fixpoints fit sequential query evaluation.
Motivation (cont'd)

- But what fits parallel evaluation? Here we propose: divide and conquer recursion on sets.
- This is an old construct. Called:
  - *pump* in FAD
    [Bancilhon, Briggs, Khoshafian, Valduriez]
  - *hom* in MACHIAVELLI
    [Ohori, Buneman, Breazu-Tannen]
  - a form of *transducer* in SVP
    [Parker, Simon, Valduriez]
Two Examples of Divide and Conquer Recursion on Sets

Parity

\[
\begin{align*}
\text{parity}(&\emptyset) & \overset{\text{def}}{=} \text{false} \\
\text{parity}(&\{x\}) & \overset{\text{def}}{=} \text{true} \\
\text{parity}(&S_1 \cup S_2) & \overset{\text{def}}{=} \text{parity}(S_1) \ XOR \ \text{parity}(S_2) \\
& \text{when } S_1 \cap S_2 = \emptyset
\end{align*}
\]

E.g. \(\text{parity}\{a, b, c, d, e\}\) can be computed as follows:

Note that there are several ways a set \(S\) can be decomposed during a divide and conquer recursion.
Two Examples of Divide and Conquer Recursion on Sets (cont’d)

Transitive closure

Given a binary relation $R$, compute its transitive closure $tc(R)$ by:

$$tc(R) \overset{\text{def}}{=} \varphi(V)$$

where $V$ and $\varphi$ are defined by:

$$V \overset{\text{def}}{=} \Pi_1(R) \cup \Pi_2(R)$$

$$\left\{\begin{array}{l}
\varphi(\emptyset) \overset{\text{def}}{=} \emptyset \\
\varphi\{x\} \overset{\text{def}}{=} R \\
\varphi(S_1 \cup S_2) \overset{\text{def}}{=} \varphi(S_1) \cup \varphi(S_2) \cup \varphi(S_1) \circ \varphi(S_2)
\end{array}\right.$$ 

when $S_1 \cap S_2 = \emptyset$

Note that, when $\text{card}(S') = k$, then

$$\varphi(S') = R \cup R^2 \cup \ldots \cup R^k$$
Formal definition of Divide and Conquer Recursions

Given $e, f, u$, define $\varphi \overset{\text{def}}{=} dcr(e, f, u)$

\[
\begin{align*}
\varphi(\emptyset) & \overset{\text{def}}{=} e \\
\varphi(\{x\}) & \overset{\text{def}}{=} f(x) \\
\varphi(S_1 \cup S_2) & \overset{\text{def}}{=} u(\varphi(S_1), \varphi(S_2)) \\
\text{when } S_1 \cap S_2 = \emptyset
\end{align*}
\]

Need to check:

\[
\begin{align*}
\quad u(a, u(b, c)) & = u(u(a, b), c) & \text{(associativity)} \\
\quad u(a, b) & = u(b, a) & \text{(commutativity)} \\
\quad u(e, a) & = u(a, e) = a & \text{(identity)}
\end{align*}
\]

Checking the conditions is undecidable, not even r.e.!
The Rest of the Language

- desired: same expressive power as $FO$, at flat types.
- but in a formalism and type system compatible with the presentation of $dcr$ [Breazu-Tannen, Buneman, Wong 92].

The Types

\[
t ::= D \quad \text{the base type}
\]
\[
  \mid \quad \text{bool} \quad \text{booleans}
\]
\[
  \mid t \times t \quad \text{product type}
\]
\[
  \mid \{t\} \quad \text{set type}
\]

Restrictions of the set height

Set height $\leq 1$ correspond to flat relations and scalar values. E.g. $t = D \times \{D \times D\}$.

Arbitrary set heights correspond to complex objects. E.g. $t = \{D \times \{D\}\}$. 
The Rest of the Language (cont’d)

The Language $\mathcal{NRA}$

\[
\begin{array}{c}
\frac{x : t}{x^t : t} & \frac{e_1 : t_1 \quad e_2 : t_2}{(e_1, e_2) : (t_1, t_2)} \\
\frac{e : t_1 \times t_2}{\pi_1(e) : t_1} & \frac{e : t_1 \times t_2}{\pi_2(e) : t_2} \\
\frac{e : t}{\emptyset : \{t\}} & \frac{\{e\} : \{t\}}{e_1 \cup e_2 : \{t\}} \\
\frac{e_1 : D \quad e_2 : D}{e_1 = e_2 : \text{bool}} & \frac{() : \text{unit}}{e : \{t\}} \\
\frac{e : \text{bool} \quad e_1 : t \quad e_2 : t}{\text{empty}(e) : \text{bool}} & \frac{\text{if } e \text{ then } e_1 \text{ else } e_2 : t}{e : t} \\
\frac{\lambda x^s.e : s \to t}{f : s \to t \quad e : s} & \frac{f(e) : t}{f : s \to \{t\}} \\
\frac{f : s \to \{t\}}{\text{ext}(f) : \{s\} \to \{t\}}
\end{array}
\]

Explanations

- $\cup$ is union
- $\text{ext}(f)(\{x_1, \ldots, x_n\}) \overset{\text{def}}{=} f(x_1) \cup \ldots \cup f(x_n)$
The Rest of the Language (cont’d)

Connection with other languages

- At flat types: \( \mathcal{NRA}^1 \) has the same expressive power as \( FO \).

\[
\mathcal{NRA}^1 = FO
\]

- At all types (complex object types): \( \mathcal{NRA} \) has the same expressive power as other formalisms for tractable language for complex objects (Abiteboul and Beeri’s algebra without powerset, Scheck and Scholl’s \( NF^2 \), Thomas and Fischer’s algebra, Paredaens and Van Gucht’s nested algebra).

The language for \( NC \) at flat types: \( \mathcal{NRA}^1(dcr) \)
Review: The classes $NC, AC^k$

**Definition 1**

- Let $k \geq 0$. Then $AC^k$ = the functions computable on a CRCW PRAM:
  - in time $O(\log^k n)$, and
  - with polynomially many processors

[Stockmeyer and Vishkin 84]

- $NC = \cup_{k \geq 0} AC^k$.

**Comments**

- $AC^0 \subset AC^1 \subset \ldots \subset NC \subset PTIME$

- Just as PTIME is considered as the class of sequential tractable functions, $NC$ is considered the class of parallel tractable functions.
Main Result

**Theorem 1**
- \( \mathcal{NRA}^1(dcr, \leq) = NC \)
- \( \mathcal{NRA}^1(dcr^{(k)}, \leq) = AC^k, \forall k \geq 1 \)

(where \( dcr^{(k)} \) means that \( dcr \) may be nested at most \( k \) levels).

In contrast:

**Proposition 2** *(This is related to [Immelman, Patnaik, Stemple 91]*)
- \( \mathcal{NRA}^1(sri, \leq) = \mathcal{NRA}^1(sri^{(1)}, \leq) = PTIME \)

where, for given \( e, i \), the structural recursion on the insert presentation \( \varphi = sri(e, i) \) is defined by:

\[
\begin{align*}
\varphi(\emptyset) & \overset{\text{def}}{=} e \\
\varphi(\{x\} \cup S) & \overset{\text{def}}{=} i(x, \varphi(S))
\end{align*}
\]

(Need to check conditions similar to those for \( sr\).)
$NC$ v.s. $PTIME$

- The difference between $NC$ and $PTIME$ reduces to the difference between two ways of recurring over sets.
- [Breazu-Tannen & Subrahmanian 91] give a translation of $dcr$ into $sri$, implying that $\mathcal{NRA}^1(dcr) \subseteq \mathcal{NRA}^1(sri)$. But it is unlikely that $dcr$ can simulate efficiently $sri$, because it is unlikely that $NC = PTIME$. 
Comments on the Main Result

- Without order, $\mathcal{NRA}^1(dcr) \subseteq NC$. This holds even in the presence of external functions in $NC$, e.g. $\mathcal{NRA}^1(int, +, -, *, div; dcr) \subseteq NC$. But we need order to capture the whole class $NC$ with $dcr$ [Immerman, Patnaik, Stemple 91].

- $dcr$ allows iteration of a function $\log n$ times, i.e. $g(x) \overset{\text{def}}{=} f^{(\log n)}(x)$, for $n = \text{card}(S)$.

- We express $f^{(\log^2 n)}(x)$ with two levels of $dcr$ nesting.

- $sri$ allows iteration of a function $n$ times, i.e. $g(x) \overset{\text{def}}{=} f^{(n)}(x)$, for $n = \text{card}(S)$.

- Can get $f^{(n^2)}(x)$ still with one nesting level by taking $S \times S$. 
Complex objects

dcr, sri can both compute powerset over complex objects. To capture NC and PTIME, we define “bounded” versions of dcr and sri, which are tractable.

Given $e, f, u$ and a set $B$, define $\varphi$ by bounded divide and conquer recursion [Buneman]:

$$
\begin{align*}
\varphi(\emptyset) & \text{ def } e \cap B \\
\varphi(\{x\}) & \text{ def } f(x) \cap B \\
\varphi(S_1 \cup S_2) & \text{ def } u(\varphi(S_1), \varphi(S_2)) \cap B \\
\text{when } S_1 \cap S_2 & = \emptyset
\end{align*}
$$

Need to check conditions.

Similar, define bounded structural recursion on the insert presentation.

**Theorem 3**

- $\mathcal{NRA}(bdcr, \leq) = NC$
- $\mathcal{NRA}(bsri, \leq) = PTIME$
Conclusions and open problems

• $NRA^1(dcr)$ and $NRA(bdcr)$ are not r.e, hence the results here are not so nice as e.g. 
$(FO + LFP + \leq) = PTIME$.

However, there are r.e. subsets of $NRA^1(dcr)$ and $NRA(bdcr)$ which capture $NC$ over ordered databases.

• Difficult open problem: is there a r.e. language $L$ expressing exactly the $NC$ queries over all databases?

• Recall [Abiteboul and Vianu 91]:

$$FO + LFP \neq FO + PFP \iff PTIME \neq PSPACE$$

Open problem:

$$NRA^1(dcr) \neq NRA^1(sri) \iff NC \neq PTIME$$