A paper to celebrate Professor Peter Buneman's 70th birthday.
Released online, 14 May 2012.

The Dichotomous Intensional Expressive Power of the Nested Relational Calculus with Powerset[*]

Limsoon Wong

National University of Singapore
wongls@comp.nus.edu.sg

Abstract. Most existing studies on the expressive power of query languages have focused on what queries can be expressed and what queries cannot be expressed in a query language. They do not tell us much about whether a query can be implemented efficiently in a query language. Yet, paradoxically, efficiency is of primary concern in computer science. In this paper, the efficiency of queries in $\mathcal{NRC}(\mathit{powerset})$, a nested relational calculus with a powerset operation, is discussed. A dichotomy in the efficiency of these queries on a large general class of structures—which include long chains, deep trees, etc.—is discovered. In particular, it is shown that these queries are either already expressible in the usual nested relational calculus or require at least exponential space. This Dichotomy Theorem, when coupled with the Bounded Degree Property of the usual nested relational calculus proved earlier by Libkin and Wong, becomes a powerful general tool in studying the intensional expressive power of query languages. The Bounded Degree Property makes it easy to prove that a query is inexpressible in the usual nested relational calculus. Then, if the query is expressible in $\mathcal{NRC}(\mathit{powerset})$, subject to the conditions of the Dichotomy Theorem, the query must take at least exponential space.

1 Introduction

Existing research on the power of query languages has focused almost exclusively on the expressive power of query languages. So we have many results of the following kinds:

- Is a specific function expressible in a given query language? For example, Libkin & Wong showed that all usual nested relational calculi and algebras cannot express the transitive closure function in general [11].
- What complexity class do functions expressible in a given query language belong to? For example, Buneman et al. showed that functions expressible in all the usual nested relational calculi and algebras have polynomial complexity [4].

[*] Supported in part by a Singapore National Research Foundation grant NRF-G-CRP-2007-04-082(d), a Singapore Ministry of Education grant MOE2009-T2-2-004, and a Singapore Agency for Science Technology and Research grant SERC-102-101-0030.

- What general properties do functions expressible in a given query language have? For example, Dong et al. [7] showed that all functions on unordered graphs expressible in a nested relational calculus with aggregate functions have the Bounded Degree Property and, thus, cannot transform a simple graph (which has an arbitrarily large but fixed degree) into a complex graph (which has an arbitrary number of distinct degrees).

These results are purely extensional. They basically state that a large class of queries is expressible or representable in a query language. However, they say nothing about the efficiency of such a representation, even though the efficiency aspect is of primary concern for computer science.

A function f that is expressible in a query language can be implemented in many different ways, each corresponding to a different algorithm. These different algorithms—which implement that same function f, as far as input/output is concerned—may have rather different complexity. Moreover, some algorithms for f may not even be expressible in the given query language, though some other algorithm for f is expressible in the given query language. Seldom do we see results that study the power of query languages from this "intensional" perspective. Some of the exceptional papers that are in the spirit of intensional expressive power include:

- The work of Colson [5] which showed that the function which computes the minimum of two integers in unary representation cannot be programmed using primitive recursion in $O(min(m, n))$ complexity.
- The work of Abiteboul and Vianu [2] which proved that the parity query cannot be expressed in PTIME by a generic machine.
- The work of Suciu and Wong [14] which proved that any uniform translation of sequential iteration queries (*sri* queries) into data-parallel iteration queries (*sru* queries) over a nested relational algebra must map some PTIME queries into exponential space ones.
- The work of Suciu and Paredaens [13] which proved that any implementation of the transitive closure query in Abiteboul and Beeri's complex object algebra must use an exponential amount of space.

However, these intensional results tend to be very query specific. Furthermore, the proofs tend to be complex and are not easily "portable" to other queries. So they do not shed sufficient light on the structure of the query languages concerned or the structure of inefficient queries in these query languages that render the cause of the inefficiency clear.

In contrast, the intensional expressive power of $\mathcal{NRC}(powerset)$, a nested relational calculus endowed with a powerset operation, is studied here in a more general non-query-specific setting—I think this is probably the first time that intensional expression power is studied in such a general setting. This calculus, to be presented in Section 2, is equivalent to the complex object algebra of Abiteboul and Beeri [1] which, as mentioned earlier, was shown by Suciu and Paredaens [13] to use exponential space to implement the transitive closure query.

Here, all flat relational queries on a general class of structures that exhibit a "seriously dichotomous" property are considered. Intuitively, a seriously dichotomous structure has two groups of "motifs" that characterize all the elements in the structure. The first group of motifs have small radius and are populated by a small predictable number of elements in the structure, while the second group of motifs have large radius and are populated by an arbitrarily large number of elements in the structure. Graphs with a few long chains or a few deep trees are seriously dichotomous structures. Specifically, the points near the ends of the few long chains satisfy the first group of motifs, while the rest of the chains—being long and thus arbitrarily many—satisfy the second group of motifs. Similarly, the points near the roots of the few deep trees satisfy the first group of motifs, while the rest of the trees—being deep and thus arbitrarily many—satisfy the second group of motifs.

The class of seriously dichotomous structures, however, exclude structures that have arbitrarily large fan-out but shallow depth. For example, structures containing an arbitrary number of short chains or an arbitrary number of short circles are not members of this class. These two types of structures do not possess the second group of motifs which are required to have large radius, as the chains and circles are short. They also do not possess the first group of motifs which are required to be populated by a small number of elements in the structure, as there are arbitrary number of chains and circles.

By virtue of the fact that non-seriously dichotomous structures lack the second group of motifs, all recursive queries on them can be converted to ones that do not need recursion, provided the maximum radius of the group of motifs for them is known in advance. Thus, the class of seriously dichotomous structures are those that *really* require an arbitrarily deep level of recursion or the full power of the powerset operation (if recursion is unavailable) to manipulate. Indeed, this paper proves—in Section 4—that all flat relational queries in $\mathcal{NRC}(\mathit{powerset})$ on seriously dichotomous structures either (i) are already expressible without the powerset operation and, hence, has a PTIME implementation in $\mathcal{NRC}(\mathit{powerset})$; or (ii) are inexpressible without using the powerset operation on a non-trivial amount of data and, hence, can only be implemented in $\mathcal{NRC}(\mathit{powerset})$ using an exponential amount of space.

The proof of this Dichotomy Theorem reveals the exact cause of the blowup and, briefly, it proceeds as follows. $\mathcal{NRC}(\mathit{powerset})$ is known to have the Conservative Extension Property [16, 9], which is described later in Section 3.1. Moreover, the normal form induced by this property does not increase the complexity of the query. Inspecting this normal form, the subexpression containing the first instance of the powerset operation—say *powerset* e—to be executed is analyzed. By the Conservative Extension Property, e is known to be equivalent to a first order formula $\xi(\boldsymbol{x}, \boldsymbol{y})$, where $\boldsymbol{x}$ are free variables corresponding to input that is bound before e is excuted, and $\boldsymbol{y}$ are free variables corresponding to output produced after e finishes execution. There are only three situations that need to be considered:

1. y_j in $\boldsymbol{y}$ is connected to some x_i in $\boldsymbol{x}$; that is, the point that y_j is instantiated with is close to some point that is used to instantiate an x_i. If the query is restricted to structures with a known maximum fan-out, then the number of possible values that y_j can take with respect to each instantiation of x_i can be calculated in advance.
2. y_j has to be instantiated to a point characterized by the first type of motifs in a seriously dichotomous structure. This type of motifs are populated by a small predictable number of elements. So the number of possible values that y_j can take can be calculated in advance.
3. y_j has to be instantiated to a point that is not close to any point x_i and is characterized by the second kind of motifs in a seriously dichotomous structure. As mentioned, this kind of motifs are populated by an arbitraily large number of elements in the structure. By the Locality Property of first order formula [8, 7, 11], which is described later in Section 3.2, y_j must take on an arbitrarily large number of values. Unfortunately, this number cannot be calculated in advance independent of the size of the input relations.

If each y_j in $\boldsymbol{y}$ takes only a predictable number of possible values that can be calculated in advance and independent of the size of the input relations, then the number of tuples—say, $h*$—in the result of evaluating e can be estimated in advance and independent of the input relations. Then this *powerset* e can be replaced by $\mathit{powerset}_{h*}\ e$, where $\mathit{powerset}_{h*}$ is an operation that computes subsets of size up to $h*$. Clearly, $\mathit{powerset}_{h*}$ can be implemented in $\mathcal{NRC}(\mathit{powerset})$ without using the powerset operation. If all the powerset operations can be eliminated in this manner, we get a PTIME implementation of the query in $\mathcal{NRC}(\mathit{powerset})$. On the other hand, if the third situation is encountered, then that *powerset* e cannot be eliminated. It is easy to see that, in a seriously dichotomous structure $\mathcal{A} = \langle A, \boldsymbol{O} \rangle$, there are many more elements that populate the second kind of motifs than the first kind. Thus, the expression e in *powerset* e is guaranteed to produce at least $c * |\boldsymbol{O}|$ number of elements, where c is a fraction close to 1. Consequently, *powerset* e is forced to produce at leasy $2^{c*|\boldsymbol{O}|}$ number of elements, causing the exponential blow up.

2 Nested Relational Calculus with Powerset

Let me first recall the nested relational calculus $\mathcal{NRC}$ from Buneman et al. [4]. The types and expressions in $\mathcal{NRC}$ are given in Figure 1. The type superscripts in the figure are usually omitted because they can be inferred.

The semantics of a type is just a set of complex objects. There are some unspecified base types b and the usual Boolean base type *bool*. An object of type $s_1 \times \cdots \times s_n$ is a tuple whose ith component is an object of type s_i, for $1 \leq i \leq n$. An object of type $\{s\}$ is a finite set whose elements are objects of type s; an object of type $\{s\}$ is called a "relation". Moreover, if $s = b \times \cdots \times b$, then an object of type $\{s\}$ (or s) is called a "flat relation". On the other hand, if s contains some set brackets, then an object of type $\{s\}$ is called a "nested relation". More generally, a type s containing n levels of nested set brackets is

Types in $\mathcal{NRC}$

$$s ::= b \mid \mathit{bool} \mid s_1 \times \cdots \times s_n \mid \{s\}$$

Expressions in $\mathcal{NRC}$

$$\frac{}{c : b} \qquad \frac{}{x^s : s} \qquad \frac{e_1 : s_1 \quad \ldots \quad e_n : s_n}{(e_1, \ldots, e_n) : s_1 \times \cdots \times s_n} \qquad \frac{e : s_1 \times \cdots \times s_n}{\pi_i \ e : s_i} 1 \leq i \leq n$$

$$\frac{}{\{\}^s : \{s\}} \qquad \frac{e : s}{\{e\} : \{s\}} \qquad \frac{e_1 : \{s\} \quad e_2 : \{s\}}{e_1 \cup e_2 : \{s\}} \qquad \frac{e_1 : \{s\} \quad e_2 : \{t\}}{\bigcup\{e_1 \mid x^t \in e_2\} : \{s\}}$$

$$\frac{}{\mathit{true} : \mathit{bool}} \qquad \frac{}{\mathit{false} : \mathit{bool}} \qquad \frac{e_1 : \mathit{bool} \quad e_2 : s \quad e_3 : s}{\mathit{if} \ e_1 \ \mathit{then} \ e_2 \ \mathit{else} \ e_3 : s}$$

$$\frac{e_1 : b \quad e_2 : b}{e_1 = e_2 : \mathit{bool}} \qquad \frac{e : \{b \times \cdots \times b\}}{\mathit{isempty} \ e : \mathit{bool}}$$

Powerset Operator in $\mathcal{NRC}(\mathit{powerset})$

$$\frac{e : \{b \times \cdots \times b\}}{\mathit{powerset} \ e : \{\{b \times \cdots \times b\}\}}$$

Fig. 1. $\mathcal{NRC}$ and its extension $\mathcal{NRC}(\mathit{powerset})$.

said to be of height n; e.g., $b \times b$ has height 0, $\{b \times b\}$ has height 1, and $\{b \times \{b\}\}$ has height 2.

The semantics of the expression constructs are described below. The expression c denotes some constants of base type b. The expressions *true*, *false*, and $\mathit{if} \ e_1 \ \mathit{then} \ e_2 \ \mathit{else} \ e_3$ have their usual semantics. The expression $(e_1, \ldots, e_n)$ denotes the tuple whose ith component is the object denoted by e_i, for $1 \leq i \leq n$. The expression $\pi_i \ e$ denotes the ith component of the tuple denoted by e. The expression $\{\}$ denotes the empty set. The expression $\{e\}$ denotes the singleton set containing the object denoted by e. The expression $e_1 \cup e_2$ denotes the union of the sets e_1 and e_2. The expression $\bigcup\{e_1 \mid x \in e_2\}$ denotes the set obtained by first applying the function $f(x) = e_1$ to each object in the set e_2 and then taking their union; that is, $\bigcup\{e_1 \mid x \in e_2\} = f(C_1) \cup \ldots \cup f(C_n)$, where $f(x) = e_1$ and $\{C_1, \ldots, C_n\}$ is the set denoted by e_2.

Note that the $x \in e_2$ part in the $\bigcup\{e_1 \mid x \in e_2\}$ construct is not a membership test. It is an abstraction that introduces the variable x whose scope is the expression e_1. This construct is the sole means in $\mathcal{NRC}$ for iterating over a set. For example, the cartesian product of two sets X and Y can be defined as $\mathit{cartprod}(X, Y) =_{\mathit{df}} \bigcup\{\bigcup\{\{(x, y)\} \mid x \in X\} \mid y \in Y\}$. As a second example, the flattening of a nested set X can be defined as $\mathit{flatten}(X) =_{\mathit{df}} \bigcup\{x \mid x \in X\}$. As a last example, the projection of the first column of a relation X can be defined as $\mathit{\Pi}_1(X) =_{df} \bigcup\{\{\pi_1\ x\} \mid x \in X\}$.

The notation $e[\boldsymbol{R}]$ stands for the an expression e with free variables $\boldsymbol{R}$; however, when it is not important to explicitly list the free variables, it is written simply as e. For a list of objects $\boldsymbol{O}$ that conform to the types of $\boldsymbol{R}$, the notation $e[\boldsymbol{O}/\boldsymbol{R}]$ stands for the expression obtained by substituting $\boldsymbol{O}$ for $\boldsymbol{R}$ in the standard way. The expression $e[\boldsymbol{R}]$ can be thought of as a "query" where $\boldsymbol{R}$ are its input; equivalently, it can be thought of as a function $f(\boldsymbol{R}) = e[\boldsymbol{R}]$. The expression $e[\boldsymbol{R}]$ is said to be a "flat relational query" if each R in $\boldsymbol{R}$ is a flat relation and $e[\boldsymbol{R}] : \{b \times \cdots \times b\}$. Recall that a flat relation can have type $\{b \times \cdots \times b\}$ or type $b \times \cdots \times b$. So, the notation $e[\boldsymbol{R}, \boldsymbol{x}]$ is used here when it is important to explicitly separate the two kinds of variables in a flat relational query. The result below on the expressive power of $\mathcal{NRC}$ is well known.

Proposition 1 (Wong [16]).

1. *$\mathcal{NRC}$ is in PTIME.*
2. *$\mathcal{NRC}$ is equivalent to the classical nested relational algebra.*
3. *$\mathcal{NRC}$, when restricted to flat relational queries, is equivalent to the classical relational algebra.*

As $\mathcal{NRC}$ is not more powerful than the classical relational algebra, recursive queries such as the transitive closure query are inexpressible in $\mathcal{NRC}$. In fact, as shown by Libkin and Wong [11], these queries remain inexpressible even when $\mathcal{NRC}$ is augmented with arithmetics and aggregate functions. One proposal to enable a nested relational calculus or algebra to express complex queries, without resorting to explicit recursion, is to endow the calculus or algebra with a powerset operation. Indeed, this option was proposed by Abiteboul and Beeri [1] and by Suciu and Paredaens [13].

Following in their foot steps, a more powerful nested relational calculus $\mathcal{NRC}(\mathit{powerset})$ is defined here by augmenting $\mathcal{NRC}$ with a powerset operation on flat relations, as shown in Figure 1. Here, *powerset* e produces a set containing all the subsets of the set denoted by e, provided e is a flat relation. By factoring through the equivalence [4] between $\mathcal{NRC}$ and a corresponding nested relational algebra, the result below on the expressive power of $\mathcal{NRC}(\mathit{powerset})$ is readily obtained.

Proposition 2 (Buneman et al. [4]). *$\mathcal{NRC}(\mathit{powerset})$ is equivalent to the complex object algebras of Abiteboul and Beeri and of Suciu and Paredaens.*

Following Suciu and Paredaens [13], a call-by-value operational semantics is defined for $\mathcal{NRC}(\mathit{powerset})$, as shown in Figure 2. In this operational semantics, $e \Downarrow C$ means the closed expression e is evaluated to the object C. The notation $C_1 \cup \cdots \cup C_n$ denotes the set of objects obtained by the union of the sets C_1, $\ldots$, C_n. This evaluation is sound in the sense that, when $e : s$ and $e \Downarrow C$, then C is an object of type s and $e = C$. Thus, each $e : s$ evaluates to a unique C. The notation $e \Downarrow$ is used here to refer to the unique evaluation tree of e.

$$\frac{}{c \Downarrow c} \qquad \frac{e_1 \Downarrow C_1 \quad \ldots \quad e_n \Downarrow C_n}{(e_1, \ldots, e_n) \Downarrow (C_1, \ldots, C_n)} \qquad \frac{e \Downarrow (C_1, \ldots, C_n)}{\pi_i \; e \Downarrow C_i} 1 \le i \le n$$

$$\frac{}{\{\} \Downarrow \{\}} \qquad \frac{e \Downarrow C}{\{e\} \Downarrow \{C\}} \qquad \frac{e_1 \Downarrow C_1 \quad e_2 \Downarrow C_2}{e_1 \cup e_2 \Downarrow C_1 \cup C_2}$$

$$\frac{e_2 \Downarrow \{C_1, \ldots, C_n\} \quad e_1[C_1/x] \Downarrow C'_1 \quad \cdots \quad e_1[C_n/x] \Downarrow C'_n}{\bigcup\{e_1 \mid x \in e_2\} \Downarrow C'_1 \cup \cdots \cup \; C'_n}$$

$$\frac{}{\mathit{true} \Downarrow \mathit{true}} \qquad \frac{}{\mathit{false} \Downarrow \mathit{false}}$$

$$\frac{e_1 \Downarrow \mathit{true} \quad e_2 \Downarrow C}{\mathit{if} \; e_1 \; \mathit{then} \; e_2 \; \mathit{else} \; e_3 \Downarrow C} \qquad \frac{e_1 \Downarrow \mathit{false} \quad e_3 \Downarrow C}{\mathit{if} \; e_1 \; \mathit{then} \; e_2 \; \mathit{else} \; e_3 \Downarrow C}$$

$$\frac{e_1 \Downarrow C_1 \quad e_2 \Downarrow C_2}{e_1 = e_2 \Downarrow \mathit{true}} C_1 = C_2 \qquad \frac{e_1 \Downarrow C_1 \quad e_2 \Downarrow C_2}{e_1 = e_2 \Downarrow \mathit{false}} C_1 \ne C_2$$

$$\frac{e \Downarrow C}{\mathit{isempty} \; e \Downarrow \mathit{true}} C = \{\} \qquad \frac{e \Downarrow C}{\mathit{isempty} \; e \Downarrow \mathit{false}} C \ne \{\}$$

$$\frac{e \Downarrow \{C_1, \ldots, C_n\}}{\mathit{powerset} \; e \Downarrow \{C'_1, \ldots, C'_{2^n}\}}$$
$$\text{where } C'_1, \ldots, C'_{2^n} \text{ are the subsets of } \{C_1, \ldots, C_n\}$$

Fig. 2. A call-by-value operational semantics of $\mathcal{NRC}(\mathit{powerset})$.

The complexity $\mathit{sizeof}(e \Downarrow)$ of an evaluation is normally defined in terms of the size of the evaluation tree. However, for the purpose of this paper, and analogous to Suciu and Paredaens [13], it is sufficient to define it in terms of the size of the largest object in the evaluation tree. That is, $\mathit{sizeof}(e \Downarrow) = \max\{\mathit{sizeof}(C) \mid \text{the object } C \text{ occurs in the evaluation tree } e \Downarrow\}$. The size of an object is defined in some standard way, e.g., the number of symbols needed to write it out.

Suciu and Paredaens [13] showed a deep result that can be restated in $\mathcal{NRC}(\mathit{powerset})$ as follows:

Proposition 3 (Suciu and Paredaens [13]). *Let $e[R]$ be a query that implements the transitive closure of an input flat relation $R : \{b \times b\}$ in $\mathcal{NRC}(\mathit{powerset})$. Let O be a sufficiently long chain of type $\{b \times b\}$. Then $\mathit{sizeof}(e[O/R] \Downarrow)$ is $\Omega(2^{|O|})$. That is, every implementation of transitive closure in $\mathcal{NRC}(\mathit{powerset})$ requires exponential space.*

In this paper, an alternative proof of this result is presented. Moreover, it is generalized here to a dichotomy result on practically all flat relational queries expressible in $\mathcal{NRC}(powerset)$. In particular, practically all flat relational queries expressible in $\mathcal{NRC}(powerset)$ are shown here to be dichotomous in the sense that either they are already expressible in $\mathcal{NRC}$ or they require at least exponential space. Hence, the extra expressive power that the powerset operation buys for $\mathcal{NRC}(powerset)$ comes strictly with an exponential cost.

3 Conservative Extension and Locality Properties

Two main machineries are needed to prove the Dichotomy Theorem. The first is the Conservative Extension Property of $\mathcal{NRC}$ and the system of rewrite rules used for proving this property. The second is the Locality Property of first-order queries.

3.1 Conservative Extension

The Conservative Extension Property and the associated system of rewrite rules were initially described by Wong [16] and, later, generalized by Libkin and Wong [9, 11]. This system of rewrite rules is given in Figure 3.

$$
\begin{aligned}
\bigcup\{e \mid x \in \{\}\} &\mapsto \{\} \\
\bigcup\{e_1 \mid x \in \{e_2\}\} &\mapsto e_1[e_2/x] \\
\bigcup\{e \mid x \in (e_1 \cup e_2)\} &\mapsto \bigcup\{e \mid x \in e_1\} \ \cup\ \bigcup\{e \mid x \in e_2\} \\
\bigcup\{e_1 \mid x \in \bigcup\{e_2 \mid y \in e_3\}\} &\mapsto \bigcup\{\bigcup\{e_1 \mid x \in e_2\} \mid y \in e_3\} \\
\bigcup\{e \mid x \in (\mathit{if}\ e_1\ \mathit{then}\ e_2\ \mathit{else}\ e_3)\} &\mapsto \mathit{if}\ e_1\ \mathit{then}\ \bigcup\{e \mid x \in e_2\}\ \mathit{else}\ \bigcup\{e \mid x \in e_3\} \\
\pi_i(e_1, \ldots, e_2) &\mapsto e_i \\
\pi_i\ (\mathit{if}\ e_1\ \mathit{then}\ e_2\ \mathit{else}\ e_3) &\mapsto \mathit{if}\ e_1\ \mathit{then}\ \pi_i\ e_2\ \mathit{else}\ \pi_i\ e_3 \\
\mathit{if}\ \mathit{true}\ \mathit{then}\ e_2\ \mathit{else}\ e_3 &\mapsto e_2 \\
\mathit{if}\ \mathit{false}\ \mathit{then}\ e_2\ \mathit{else}\ e_3 &\mapsto e_3
\end{aligned}
$$

Fig. 3. A system of rewrite rules for $\mathcal{NRC}(powerset)$.

The following properties of this system of rewrite rules are well known.

Proposition 4 (Conservative Extension [16, 9]).

1. *This system of rewrite rules is sound.*
2. *This system of rewrite rules is strongly normalizing.*
3. *Let e be an expression in $\mathcal{NRC}(powerset)$ that is in normal form with respect to this system of rewrite rules. That is, no rule can be applied to further rewrite e. Let $e'[\boldsymbol{R}] : s$ be a subexpression in e. Suppose $\boldsymbol{R}$ have types whose height is atmost h, and the type s has height h'. Then all the types appearing in the type derivation of $e'[\boldsymbol{R}] : s$ have height atmost $\max(h, h')$, if the powerset operation does not appear in $e'[\boldsymbol{R}]$; or, they have height atmost $\max(h, h', 2)$, if the powerset operation appears in $e'[\boldsymbol{R}]$.*

It is straightforward to show that this system of rewrite rules does not increase the complexity of evaluation.

Proposition 5. *Let $e[\boldsymbol{R}] \mapsto e'[\boldsymbol{R}]$. Let $\boldsymbol{O}$ be a list of objects conforming to the types of $\boldsymbol{R}$. Then sizeof$(e[\boldsymbol{O}/\boldsymbol{R}] \Downarrow) \geq$ sizeof$(e'[\boldsymbol{O}/\boldsymbol{R}] \Downarrow)$.*

3.2 Locality

The second main machinery needed to prove the dichotomy result is the Locality Property. Let me first introduce the notions of "τ structure", "Gaifman graph", "r-sphere", and "r-neighbourhood", before explaining what the Locality Property is.

A signature τ is a list of symbols $\boldsymbol{R}$, where $\boldsymbol{R}$ is to be regarded as input for a query. The signature τ_m is obtained by extending the signature τ with m new constant symbols. For the purpose of this paper, each R_i in $\boldsymbol{R}$ has type of the form $\{b \times \cdots \times b\}$. A τ structure $\mathcal{A} = \langle A, \boldsymbol{O} \rangle$ has a universe A (which is a finite nonempty set of objects of type b) and a list of objects $\boldsymbol{O}$ (where each object O_i in $\boldsymbol{O}$ is the interpretation of the corresponding R_i and, thus, having the type of R_i). Also, all elements of $\boldsymbol{O}$ are in the universe A. The class of τ structures is denoted by STRUCT$[\tau]$. The symbol $\simeq$ is used to denote isomorphism of τ structures.

Given a τ structure $\mathcal{A} = \langle A, \boldsymbol{O} \rangle$, its Gaifman graph $\mathcal{G}(\mathcal{A})$ is defined as a graph such that its edges are precisely those pairs (a, b) where there is a tuple $t_i \in O_i$, for some O_i in $\boldsymbol{O}$, such that both a and b are in t_i. The distance $d^{\mathcal{A}}(a, b)$ is defined as the length of the shortest path from a to b in $\mathcal{G}(\mathcal{A})$. Given a tuple $\boldsymbol{a} = (a_1, \ldots, a_m)$ of objects in A, and some $r \geq 0$, the r-sphere of $\boldsymbol{a}$ is defined as $S_r^{\mathcal{A}}(\boldsymbol{a}) = \bigcup_{1 \leq i \leq m} S_r^{\mathcal{A}}(a_i)$, where $S_r^{\mathcal{A}}(a_i) = \{b \in A \mid d^{\mathcal{A}}(a_i, b) \leq r\}$. Also, the r-neighbourhood of $\boldsymbol{a}$ is defined as the τ_m structure $N_r^{\mathcal{A}}(\boldsymbol{a}) = \langle S_r^{\mathcal{A}}(\boldsymbol{a}), \boldsymbol{O}|_{S_r^{\mathcal{A}}(\boldsymbol{a})}, a_1, \ldots, a_m \rangle$. That is, $N_r^{\mathcal{A}}(\boldsymbol{a})$ is obtained by restricting $\mathcal{A}$ to the universe $S_r^{\mathcal{A}}(\boldsymbol{a})$ and adding some extra constants that are the elements of $\boldsymbol{a}$.

Gaifman [8] showed that first-order queries exhibit a kind of locality property in the sense that the result of these queries can be determined by considering "small neighbourhoods" of its input. It follows easily from the work of Gaifman and Part 3 of Proposition 1 that flat relational queries in $\mathcal{NRC}$ has this kind of locality property.

Proposition 6 (Locality [8, 7]). *Every flat relational query $e[\boldsymbol{R}]$ in $\mathcal{NRC}$ has the Locality Property. That is, there is a finite natural number r such that, for every $\mathcal{A} = \langle A, \boldsymbol{O} \rangle \in STRUCT[\boldsymbol{R}]$, for every two m-ary vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ of elements of A, it is the case that $N_r^{\mathcal{A}}(\boldsymbol{a}) \simeq N_r^{\mathcal{A}}(\boldsymbol{b})$ implies $\boldsymbol{a} \in e[\boldsymbol{O}/\boldsymbol{R}]$ if and only if $\boldsymbol{b} \in e[\boldsymbol{O}/\boldsymbol{R}]$.*

In short, for every flat relational query expressible in $\mathcal{NRC}$ there is some number r such that, for every pair $(\boldsymbol{a}, \boldsymbol{b})$, so long as a and b have neighbourhoods that are isomorphic up to radius r, they must either be both in the result of the query or both not in the result of the query. The smallest such number r is called the "locality index" of the query.

An equivalence class $\boldsymbol{a} \approx_r^{\mathcal{A}} \boldsymbol{b}$ is induced by $N_r^{\mathcal{A}}(\boldsymbol{a}) \simeq N_r^{\mathcal{A}}(\boldsymbol{b})$. Such an equivalence class is called a neighbourhood type here. If a restriction is imposed so that $\mathcal{G}(\mathcal{A})$ has degree atmost k, then the number of neighbourhood types is finite. Thus, under this restriction, for any flat relational query $e[\boldsymbol{R}]$ in $\mathcal{NRC}$, its result is completely characterized by a finite number of neighbourhood types. Each neighbourhood type $N_r^{\mathcal{A}}(\boldsymbol{a})$ can be thought of as a "diagram" showing how objects in this neighbourhood type are "connected" to each other and to the fixed reference objects (i.e., $\boldsymbol{a}$); c.f. the "neighbourhood formula" of Dong et al. [7].

4 Complexity of Queries on Dichotomous Structures

Given a signature τ. A "motif" of radius r is a first order formula $\psi(y)$ with a single free variable y such that $\psi(y)$ has locality index r on all τ structures. A τ structure $\mathcal{A}$ is said to be "bounded" by a motif $\psi(y)$ at a threshold g if $|\{a \in A \mid \mathcal{A} \models \psi(a)\}| \leq r * g$, where r is the radius of $\psi(y)$. That is, there are atmost $r * g$ elements in the universe of $\mathcal{A}$ that make $\psi(y)$ true. A class $\mathcal{C}$ of τ structures is said to be "bounded" by a motif $\psi(y)$ at a threshold g if that motif $\psi(y)$ bounds all structures in $\mathcal{C}$ at the threshold g. On the other hand, $\mathcal{C}$ is said to be "unbounded" by $\psi(y)$ if there is a g such that, for every $g' > g$, there is some $\mathcal{A} \in \mathcal{C}$ that is not bounded by $\psi(y)$ at threshold g'.

Definition 1. *A class $\mathcal{C}$ of τ structures is said to be "dichotomous" at radius r and threshold g if and only if (i) $\mathcal{C}$ is bounded at threshold g by some motifs of radius up to r, and (ii) $\mathcal{C}$ is unbounded by all other motifs (as well as at least one motif) of radius r. Moreover, $\mathcal{C}$ is said to be "seriously dichotomous" at threshold g if there is some r such that, $\mathcal{C}$ is dichotomous at threshold g and every radius $r' > r$. Seriously dichotomous structures include long chains, long circles, deep trees, etc.*

I am now ready to sketch a proof of the Dichotomy Theorem for such general classes of structures. Given a flat relational structure $\mathcal{A} = \langle A, \boldsymbol{O} \rangle$, the size of the structure is defined as $|\boldsymbol{O}| = \sum_{O_i \in \boldsymbol{O}} \mathit{sizeof}(O_i)$.

Theorem 1 (Dichotomy). *Let $e[\boldsymbol{R}] : \{b \times \cdots \times b\}$ be a flat relational query in $\mathcal{NRC}(powerset)$ that is intended for the class $\mathcal{C}$ of seriously dichotomous structures whose Gaifman graph has degree atmost k. Then either $e[\boldsymbol{R}]$ is expressible in $\mathcal{NRC}$; or, there is a structure $\mathcal{A} = \langle A, \boldsymbol{O} \rangle \in \mathcal{C}$ such that $\mathit{sizeof}(e[\boldsymbol{O}/\boldsymbol{R}] \Downarrow)$ is $\Omega(2^{|\boldsymbol{O}|})$.*

Proof. By Proposition 5, the system of rewrite rules in Figure 3 does not increase complexity. By Proposition 4, it preserves semantics and is strongly normalizing. Thus it can be assumed without loss of generality that $e[\boldsymbol{R}]$ is an expression in normal form with respect to this system of rewrite rules.

If the powerset operation does not appear in $e[\boldsymbol{R}]$, then the theorem trivially holds. So, let it contain some occurrences of the powerset operation. Let *powerset* $e'[\boldsymbol{R}, \boldsymbol{x}]$ be the occurrence of the powerset operation that corresponds to the earliest instance of the powerset operation to be evaluated when $e[\boldsymbol{R}]$ is evaluated.

Since the $\bigcup\{e_1 \mid x \in e_2\}$ construct is the only way to introduce a new variable in $\mathcal{NRC}(powerset)$, each new free variable x_i in $\boldsymbol{x}$ must have been introduced in an enclosing expression of the form $\bigcup\{\cdots powerset\ e'[\boldsymbol{R}, \boldsymbol{x}] \cdots \mid x_i \in E\}$. As the entire expression $e[\boldsymbol{R}]$ is in normal form, and $e'[\boldsymbol{R}, \boldsymbol{x}]$ is the earliest instance of the powerset operation to be evaluated, E must be one of the R_i in $\boldsymbol{R}$, which is a flat relation. Consequently, x_i must have height 0 and has a type of the form $b \times \cdots \times b$. Furthermore, as $e'[\boldsymbol{R}, \boldsymbol{x}]$ is an input to a powerset operation, its type must have the form $\{b \times \cdots \times b\}$. Thus $e'[\boldsymbol{R}, \boldsymbol{x}]$ is a flat relational query in $\mathcal{NRC}$.

In fact, by the Conservative Extension Property (Proposition 4), all the types that appear in the typing derivation of $e'[\boldsymbol{R}, \boldsymbol{x}]$ have height atmost 1 (i.e., must be flat). By Proposition 1, $e'[\boldsymbol{R}, \boldsymbol{x}]$ is equivalent to a first-order formula $\varphi(\boldsymbol{x}, \boldsymbol{y})$ such that, for every τ_m structure $\mathcal{A} = \langle A, \boldsymbol{O}, \boldsymbol{o}\rangle$ and objects $\boldsymbol{o}'$ of the appropriate types, it is the case that $\boldsymbol{o}' \in e'[\boldsymbol{O}/\boldsymbol{R}, \boldsymbol{o}/\boldsymbol{x}]$ if and only if $\mathcal{A} \models \varphi(\boldsymbol{o}, \boldsymbol{o}')$.

I am now almost ready to use the Locality Property, except for the variables $\boldsymbol{x}$. To deal with this inconvenience, we inspect the original expression $e[\boldsymbol{R}]$, in an outside-in manner until we reach the expression $e'[\boldsymbol{R}, \boldsymbol{x}]$, to extract all the conditions that must hold on $\boldsymbol{x}$ before $e'[\boldsymbol{R}, \boldsymbol{x}]$ gets evaluated. You will have to trust me that the conjunction of these conditions can be expressed as a first order formula $\psi(\boldsymbol{x})$.

It follows by Proposition 6 that $\psi(\boldsymbol{x}) \wedge \varphi(\boldsymbol{x}, \boldsymbol{y})$ enjoys the Locality Property. Let r be its locality index. Since I am only considering structures $\mathcal{A} = \langle A, \boldsymbol{O}\rangle$ whose Gaifman graph has degree atmost k, there is a finite number of neighbourhood types $N_r^{\mathcal{A}}(\boldsymbol{o}, \boldsymbol{o}')$ that make $\psi(\boldsymbol{x}) \wedge \varphi(\boldsymbol{x}, \boldsymbol{y})$ true.

Each such neighbourhood type can be described by a first order formula $\xi(\boldsymbol{x}, \boldsymbol{y})$. An x_i in $\boldsymbol{x}$ and a y_j in $\boldsymbol{y}$ is said to be "connected" if there are $R_0(t_0)$, ..., $R_h(t_h)$ and variables z_1, ..., z_{h-1} such that the pair (x_i, z_1) appears in t_0, the pair (z_1, z_2) appears in t_1, ..., and the pair (z_{h-1}, y_j) appears in t_h, and $\xi(\boldsymbol{x}, \boldsymbol{y}) \vdash \exists \boldsymbol{t}. R_0(t_0) \wedge \cdots \wedge R_h(t_h)$, where $\boldsymbol{t}$ is the collection of variables in t_0, ..., t_h, excluding x_i and y_j. It is easy to see that, if x_i and y_j are connected, then the corresponding o_i in $\boldsymbol{o}$ and o'_j in $\boldsymbol{o}'$ satisfy $o'_j \in S_h^{\mathcal{A}}(o_i)$ and $o_i \in S_h^{\mathcal{A}}(o'_j)$.

The analysis above is repeated for all neighbourhood types that make $\psi(\boldsymbol{x}) \wedge \varphi(\boldsymbol{x}, \boldsymbol{y})$ true. If y_j is connected to some x_i (not necessarily the same one) in each qualifying neighbourhood type, let h_{j*} be the largest of the h's found. Recall that the Gaifman graph is restricted to degree atmost k. This means that given any instantiation $\boldsymbol{o}$ for $\boldsymbol{x}$, there can be atmost h_{j*}^k distinct instantiations for y_j that make $\psi(\boldsymbol{x}) \wedge \varphi(\boldsymbol{x}, \boldsymbol{y})$ true.

Assuming each y_j in $\boldsymbol{y}$ is connected to some x_i (not necessarily the same one) in each qualifing neighbourhood type. Then for any instantiation $\boldsymbol{o}$ for $\boldsymbol{x}$, there can be atmost $h* = \prod_j h_{j*}^k$ distinct instantiations $\boldsymbol{o}'$ for $\boldsymbol{y}$ that make $\psi(\boldsymbol{x}) \wedge \varphi(\boldsymbol{x}, \boldsymbol{y})$ true. Notice that $h*$ is independent of the cardinality of relations used for instantiating $\boldsymbol{R}$. Recall that $\boldsymbol{o}$ are values that the free variables $\boldsymbol{x}$ take in an evaluation of *powerset* $e'[\boldsymbol{R}, \boldsymbol{x}]$. This means that for each instantiation of $\boldsymbol{x}$,

$e'[\boldsymbol{R},\boldsymbol{x}]$ evaluates to a set whose cardinality is atmost $h*$. Then the expression *powerset* $e'[\boldsymbol{R},\boldsymbol{x}]$ can be replaced by another expression $\mathit{powerset}_{h*}$ $e'[\boldsymbol{R},\boldsymbol{x}]$, which is an $\mathcal{NRC}$ expression that produces subsets of size atmost $h*$. Thus, in this case, the powerset operation can be eliminated. The entire process above is repeated as many times as necessary. At the end, if all occurrences of the powerset operation are eliminated, then the original query $e[\boldsymbol{R}]$ is expressible in $\mathcal{NRC}$.

On the other hand, if there is an occurrence of the powerset operation that cannot be eliminated as described above, then there must be some y_j in $\boldsymbol{y}$ that is not connected to any x_i in some qualifying neighbourhood type. Recall that the original query is intended for a class $\mathcal{C}$ of seriously dichotomous structures at some threshold g. Thus, there are motifs $\psi(y)$ at radius $\leq r$ that bounds $\mathcal{C}$, and all other motifs (and at least one motif) $\psi'(y_j)$ at radius r that does not bound $\mathcal{C}$. This puts us in two different scenarios.

The first scenario is when $\xi(\boldsymbol{x},\boldsymbol{y}) \vdash \psi(y_j)$ holds for one of the bounding motifs $\psi(y_j)$. Then there are atmost $r * g$ number of values for y_j that make $\psi(y_j)$ true. Consequently, by the Locality Property (Proposition 6), there are atmost $r * g$ number of values for y_j that make $\xi(\boldsymbol{x},\boldsymbol{y})$ true. In this case, when we are setting h_j*, we can let the h for this neighbourhood type to be $r * g$. The elimination of the powerset operation can then be performed as described earlier.

The second scenario is when $\xi(\boldsymbol{x},\boldsymbol{y}) \vdash \psi(y_j)$ does not hold for all the bounding motifs $\psi(y_j)$. Then, by definition of seriously dichotomous structures, there is a motif $\psi'(y_j)$ of radius r that does not bound $\mathcal{C}$ and $\xi(\boldsymbol{x},\boldsymbol{y}) \vdash \psi'(y_j)$ holds (** - more about this statement later when non-seriously dichotomous structures are discussed **). So a structure $\mathcal{A} = \langle A, \boldsymbol{O}\rangle \in \mathcal{C}$ can be chosen so that $\psi'(y_j)$ holds for $\Omega(|\boldsymbol{O}|)$ number of values for y_j. Then, by the Locality Property (Proposition 6), the corresponding $e'[\boldsymbol{R},\boldsymbol{x}]$ must produce a set whose cardinality is $\Omega(|\boldsymbol{O}|)$. Consequently, *powerset* $e'[\boldsymbol{R},\boldsymbol{x}]$ must produce $\Omega(2^{|\boldsymbol{O}|})$ subsets. Thus, in this case, *sizeof*$(e[\boldsymbol{O}/\boldsymbol{R}] \Downarrow)$ is $\Omega(2^{|\boldsymbol{O}|})$, proving the theorem. □

Therefore, for any query in $\mathcal{NRC}(\mathit{powerset})$ on seriously dichotomous structures, either it is already expressible in $\mathcal{NRC}$ (and hence in PTIME) or all of its implementations in $\mathcal{NRC}(\mathit{powerset})$ need exponential space. Since the class of structures containing a single long chain is seriously dichotomous, and the transitive closure of single long chain is inexpressible in $\mathcal{NRC}$ [7], it follows immediately as a corollary of the Dichotomy Theorem above that all implementations of transitive closure in $\mathcal{NRC}(\mathit{powerset})$ must use at least exponent space, as proven earlier by Suciu and Paredaens [13] in a less general brute-force manner.

How about queries on structures that are not seriously dichotomous? By definition, a class of seriously dichotomous structures must be unbounded by at least one motif of radius r' for every large r'. However, for each class $\mathcal{C}$ of structures that is not seriously dichomotous, there is a number r such that for all motifs $\psi(y)$ with radius $r' \geq r$, it is the case that $|\{a \in A \mid \mathcal{A} \models \psi(a)\}| = 0$ for every $\mathcal{A} \in \mathcal{C}$. That is, there is no motif with radius $r' \geq r$ that unbounds $\mathcal{C}$.

For example, the class of structures comprising sets of multiple short circles of length atmost l is not seriously dichomotous, because there is no motif of radius greater than l that unbounds these structures.

Intuitively, flat relational queries in $\mathcal{NRC}(powerset)$ on these non-seriously dichotomous structures should not require the use of the powerset operation. Since every class of structures must be either seriously dichotomous or not seriously dichomotous, I think the following claim is very likely true:

Claim. Let $e[R] : \{b \times b\}$ be a query that is expressible in $\mathcal{NRC}(powerset)$, where its input $R : \{b \times b\}$ is restricted to graphs of degree atmost k. Then $e[R]$ has a PTIME implementation in $\mathcal{NRC}(powerset)$ if and only if it is already expressible in $\mathcal{NRC}$.

To settle this claim requires every flat relational query in $\mathcal{NRC}(powerset)$ on non-seriously dichotomous structures to be transformed to a query in $\mathcal{NRC}$, or to be shown to apply the powerset operation only on very small sets of intermediate data. The proof of the Dichotomy Theorem breaks down on non-seriously dichotomous structures at the point marked (**), in the last paragraph of the proof. At that point, the proof requires the existence of a motif $\psi'(y_j)$ of radius r that (i) matches the r-neighbourhood around y_j, i.e., $\xi(\boldsymbol{x}, \boldsymbol{y}) \vdash \psi'(y_j)$, and (ii) has no bound on the number of values that y_j can be instantiated with to make $\psi'(y_j)$ true. In seriously dichotomous structures, the existence of such motifs is guaranteed. In non-seriously dichotomous structures, beyond a certain radius, such motifs do not exist.

5 Remarks

It was with Peter Buneman and Val Tannen that I first defined $\mathcal{NRC}$ in 1992 [3], two decades ago! It was also Peter and Val who first posed me the Conservative Extension Property of $\mathcal{NRC}$ as an open question, which I solved in 1993 [15] by the analysing the normal forms of the system of rewrite rules presented earlier in this paper.

I first saw in 1994 Dan Suciu and Jan Paredaens' proof [12] that all implementations of the transitive closure query in $\mathcal{NRC}(powerset)$ are necessarily inefficient. This was my first encounter with the intensional aspect of expressive power. It intrigued me greatly and I soon co-authored, in 1995, a paper [14] with Dan comparing the efficiency of the algorithms that can be implemented by different forms of structural recursion.

I first learned in 1994 [10] the Locality Property of first-order query languages from Leonid Libkin. It took me three further years to fully appreciate this powerful property and to exploit it to prove, in 1997 [6], with Leonid and Guozhu Dong, the Bounded Degree Property of query languages with aggregate function.

Leonid, Dan, and I were students in the same group led by Peter, Val, and Susan Davidson. While they have continued working in the database theory area, I have more or less left the field to explore challenges in computational biology since the late 1990s. After this ten-year break, I am delighted to briefly re-visit the field and contribute to this Festschrift to Peter. I am pleasantly surprised that I am able to chain together the series of our major past results ($\mathcal{NRC}$, normal forms of my favourite rewrite system, the Conservative Extension Property, and the Locality Property) that Peter had a big role in nurturing, to solve a problem that Peter also had a big role in keeping my continued fascination with it. I hope you have enjoyed reading the paper as much as I have enjoyed working on it.

References

1. Serge Abiteboul and Catriel Beeri. The power of languages for the manipulation of complex values. *The VLDB Journal*, 4(4):727–794, 1995.
2. Serge Abiteboul and Victor Vianu. Generic computation and its complexity. In *Proceedings of 23rd ACM Symposium on the Theory of Computing*, pages 209–219, 1991.
3. Val Breazu-Tannen, Peter Buneman, and Limsoon Wong. Naturally embedded query languages. In *Proceedings of 4th International Conference on Database Theory, Berlin, Germany, October, 1992*, pages 140–154, 1992.
4. Peter Buneman, Shamim Naqvi, Val Tannen, and Limsoon Wong. Principles of programming with complex objects and collection types. *Theoretical Computer Science*, 149(1):3–48, 1995.
5. Loic Colson. About primitive recursive algorithms. *Theoretical Computer Science*, 83:57–69, 1991.
6. Guozhu Dong, Leonid Libkin, and Limsoon Wong. Local properties of query languages. In *Proceedings of 6th International Conference on Database Theory*, pages 140–154, 1997.
7. Guozhu Dong, Leonid Libkin, and Limsoon Wong. Local properties of query languages. *Theoretical Computer Science*, 239:277–308, 2000.
8. Haim Gaifman. On local and non-local properties. In *Proceedings of the Herbrand Symposium, Logic Colloquium '81*, pages 105–135, 1982.
9. Leonid Libkin and Limsoon Wong. Conservativity of nested relational calculi with internal generic functions. *Information Processing Letters*, 49(6):273–280, 1994.
10. Leonid Libkin and Limsoon Wong. New techniques for studying set languages, bag languages, and aggregate functions. In *Proceedings of 13th ACM Symposium on Principles of Database Systems*, pages 155–166, 1994.
11. Leonid Libkin and Limsoon Wong. Query languages for bags and aggregate functions. *Journal of Computer and System Sciences*, 55(2):241–272, 1997.
12. Dan Suciu and Jan Paredaens. Any algorithm in the complex object algebra needs exponential space to compute transitive closure. In *Proceedings of 13th ACM Symposium on Principles of Database Systems*, pages 201–209, 1994.
13. Dan Suciu and Jan Paredaens. The complexity of the evaluation of complex algebra expressions. *Journal of Computer and Systems Sciences*, 55(2):322–343, 1997.
14. Dan Suciu and Limsoon Wong. On two forms of structural recursion. In *Proceedings of 5th International Conference on Database Theory*, pages 111–124, 1995.
15. Limsoon Wong. Normal forms and conservative properties for query languages over collection types. In *Proceedings of 12th ACM Symposium on Principles of Database Systems*, pages 26–36, 1993.
16. Limsoon Wong. Normal forms and conservative extension properties for query languages over collection types. *Journal of Computer and System Sciences*, 52(3):495–505, 1996.