Invariants in SQL

Limsoon Wong

Kent Ridge Digital Labs
Singapore

Joint work with Leonid Libkin.

Plan

- Nested Relational Calculus $\mathcal{NRC}$
- Conservative Extension Property
- Theoretical Reconstruction of SQL
- Finite-Cofiniteness Property
- Bounded Degree Property
- Locality Property
- Infinitary Logics with Aggregate Functions

Nested relational calculus $\mathcal{NRC}$

The complex object types are:

$$s, t ::= \ \mid bool \mid b \mid s \times t \mid \{s\}$$

The expression constructs are:

$$\frac{}{x^s : s}$$

$$\frac{e_1 : s \quad e_2 : t}{(e_1, e_2) : s \times t} \qquad \frac{e : s \times t}{\pi_1\ e : s \quad \pi_2\ e : t}$$

$$\frac{}{true\ \ : bool} \qquad \frac{}{false\ \ : bool}$$

$$\frac{e_1 : bool \quad e_2 : s \quad e_3 : s}{if\ e_1\ then\ e_2\ else\ e_3 : s}$$

$$\frac{}{\{\}^s : \{s\}} \qquad \frac{e : s}{\{e\} : \{s\}} \qquad \frac{e_1 : \{s\} \quad e_2 : \{s\}}{e_1 \bigcup e_2 : \{s\}}$$

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

$$\frac{e : \{s\}}{empty\ e : bool} \qquad \frac{e_1 : s \quad e_2 : s}{e_1 = e_2 : bool}$$

Explanation

- $\pi_1\ e$ stands for the first component of the pair e.

  Eg: $\pi_1(o_1, o_2) = o_1$.

- $\cup\{e_1 \mid x \in e_2\}$ stands for the set obtained by combining the results of applying the function $f(x) = e_1$ to each element of e_2.

  Eg: let $e_2 = \{o_1, \cdots, o_n\}$ and let $f(x) = e_1$; then $\cup\{e_1 \mid x \in e_2\} = f(o_1) \bigcup \cdots \bigcup f(o_n)$.

  Eg: $\cup\{\{x, x+1\} \mid x \in \{1, 2, 3\}\} = \{1, 2, 3, 4\}$.

Examples

- Relational projection. $\Pi_2 : \{s \times t\} \to \{t\}$.

$$\Pi_2(R) \triangleq \cup\{\{\pi_2\, x\} \mid x \in R\}$$

- Selection. $\mathit{select}(p) : \{s\} \to \{s\}$ for $p : s \to \mathit{bool}$.

$$\mathit{select}(p)(R) \triangleq \cup\{\mathit{if}\ p(x)\ \mathit{then}\ \{x\}\ \mathit{else}\ \{\} \mid x \in R\}$$

- Cartesian product. $\mathit{cartprod} : \{s\} \times \{t\} \to \{s \times t\}$.

$$\mathit{cartprod}(R, S) \triangleq \cup\{\cup\{\{(x, y)\} \mid x \in R\} \mid y \in S\}$$

- Relational nesting. $nest_2 : \{s \times t\} \rightarrow \{s \times \{t\}\}$.

  $$nest_2(R) \triangleq \cup\{\{(\pi_1\ x,\ findMatch(x,\ R))\} \mid x \in R\}$$

  where

  $$findMatch(x, R) \triangleq \Pi_2(select(p(x))(R))$$
  $$p(x)(y) \triangleq (\pi_1\ y = \pi_1\ x)$$

  Eg:

  $$nest_2 \begin{array}{||c||}\hline (1,1)\\\hline (1,2)\\\hline (2,1)\\\hline (1,3)\\\hline (2,3)\\\hline\end{array} = \begin{array}{||c||}\hline \left(1, \begin{array}{||c||}\hline 1\\\hline 2\\\hline 3\\\hline\end{array}\right)\\\hline \left(2, \begin{array}{||c||}\hline 1\\\hline 3\\\hline\end{array}\right)\\\hline\end{array}$$

Proposition 1 *$\mathcal{NRC}$ has the same expressive power as Schek&Scholl, Thomas&Fischer, etc.*

Conservative extension

Definition 2 The height $ht(s)$ of type s is the depth of nesting of set brackets in s. The height $ht(e)$ of an expression e is the sup of the heights of types appearing in its typing derivation.

Definition 3 *Let $\mathcal{L}_{i,o,k}$ be the class of functions definable by a closed expression $f : s \to t$ of a language $\mathcal{L}$ (such as $\mathcal{NRC}$), where $ht(s) \leq i$, $ht(t) \leq o$, and $ht(f) \leq k$. The language $\mathcal{L}$ has the conservative extension property* at input height i and output height o if $\mathcal{L}_{i,o,k} = \mathcal{L}_{i,o,k+1}$ for any $k \geq \max(i, o)$.

Implication. If a language has the conservative extension property, the height of intermediate data has no influence on the expressive power of the language.

A rewrite system

- $\pi_i(e_1, e_2) \leadsto e_i$
- *if true then* e_2 *else* $e_3 \leadsto e_2$
- *if false then* e_2 *else* $e_3 \leadsto e_3$
- $\cup\{e \mid x \in e_1 \bigcup e_2\} \leadsto \cup\{e \mid x \in e_1\} \bigcup \cup\{e \mid x \in e_2\}$
- $\cup\{e \mid x \in \{\}\} \leadsto \{\}$
- $\cup\{e \mid x \in \{e'\}\} \leadsto e[e'/x]$
- $\cup\{e_1 \mid x \in \cup\{e_2 \mid y \in e_3\}\}$
  $\leadsto \cup\{\cup\{e_1 \mid x \in e_2\} \mid y \in e_3\}$
- $\cup\{e \mid x \in$ *if* e_1 *then* e_2 *else* $e_3\}$
  $\leadsto$ *if* e_1 *then* $\cup\{e \mid x \in e_2\}$ *else* $\cup\{e \mid x \in e_3\}$
- and two other rules dealing with equality and empty tests.

Proposition 4 *The rewrite system is sound.*

Proposition 5 *The rewrite system is strongly normalizing.*

Theorem 6 *Let $e : s$ be an expression of $\mathcal{NRC}$ in normal form. Then $ht(e) \leq \max(\{ht(s)\} \bigcup \{ht(s) \mid s$ is the type of a free variable in $e\})$.*

Corollary 7 *$\mathcal{NRC}$ has the conservative extension property at all input and output.*

Corollary 8 *Every function from flat relation to flat relation expressible in $\mathcal{NRC}$ is expressible in flat relational algebra.*

Theoretical Reconstruction of SQL

Expressions of $\mathcal{NRC}(\mathbb{Q},+,\cdot,-,\div,\Sigma,=,\leq^{\mathbb{Q}})$ are those of $\mathcal{NRC}$ plus the followings:

$$\frac{e_1 : \mathbb{Q} \quad e_2 : \mathbb{Q}}{e_1 + e_2 : \mathbb{Q}} \qquad \frac{e_1 : \mathbb{Q} \quad e_2 : \mathbb{Q}}{e_1 \cdot e_2 : \mathbb{Q}} \qquad \frac{e_1 : \mathbb{Q} \quad e_2 : \mathbb{Q}}{e_1 \div e_2 : \mathbb{Q}}$$

$$\frac{e_1 : \mathbb{Q} \quad e_2 : \mathbb{Q}}{e_1 - e_2 : \mathbb{Q}} \qquad \frac{e_1 : \mathbb{Q} \quad e_2 : \{s\}}{\Sigma\{\!|e_1 \mid x^s \in e_2|\!\} : \mathbb{Q}}$$

$$\frac{e_1 : \mathbb{Q} \quad e_2 : \mathbb{Q}}{e_1 \leq e_2 : \mathit{bool}}$$

Semantics. $\Sigma\{\!|e_1 \mid x \in e_2|\!\} = f(o_1) + \ldots + f(o_n)$, where f is the function $f(x) = e_1$ and $\{o_1, \ldots, o_n\}$ is the set e_2.

Note. $\leq$ is restricted to numbers only.

Examples of aggregate functions

- "Count the number of records in R" is

$$count(R) \triangleq \Sigma\{\!|1|\ x \in R|\!\}$$

- "Total the first column of R" is

$$total(R) \triangleq \Sigma\{\!|\pi_1\ x \mid x \in R|\!\}$$

- "Average of the first column in R" is

$$average(R) \triangleq total(R) \div count(R)$$

- "Variance of the first column of R" is

$$variance(R) \triangleq (\Sigma\{\!|sq(\pi_1\ x) \mid x \in R|\!\} -$$

$$(sq(\Sigma\{\!|\pi_1\ x \mid x \in R|\!\}) \div count(R))) \div count(R)$$

  where $sq(y) \triangleq y \cdot y$.

- A totally generic query expressible in SQL but inexpressible in relational calculus:

$$eqcard(R, S) \triangleq \Sigma\{\!|1 \mid x \in R|\!\} = \Sigma\{\!|1 \mid x \in S|\!\}$$

Rewrite rules for summation

- $\Sigma\{\!|e \mid x \in \{\}|\!\} \leadsto 0$

- $\Sigma\{\!|e \mid x \in \{e'\}|\!\} \leadsto e[e'/x]$

- $\Sigma\{\!|e \mid x \in \mathit{if}\ e_1\ \mathit{then}\ e_2\ \mathit{else}\ e_3|\!\} \leadsto$
  $\mathit{if}\ e_1\ \mathit{then}\ \Sigma\{\!|e \mid x \in e_2|\!\}\ \mathit{else}\ \Sigma\{\!|e \mid x \in e_3|\!\}$

- $\Sigma\{\!|e \mid x \in e_1 \cup e_2|\!\} \leadsto$
  $\Sigma\{\!|e \mid x \in e_1|\!\} + \Sigma\{\!|\mathit{if}\ x \in e_1\ \mathit{then}\ 0\ \mathit{else}\ e \mid x \in e_2|\!\}$

- $\Sigma\{\!|e \mid x \in \bigcup\{e_1 \mid y \in e_2\}|\!\} \leadsto$
  $\Sigma\{\!|\Sigma\{\!|(e \div \Sigma\{\!|\Sigma\{\!|\mathit{if}\ x = v\ \mathit{then}\ 1\ \mathit{else}\ 0 \mid v \in e_1|\!\} \mid y \in e_2|\!\}) \mid x \in e_1|\!\} \mid y \in e_2|\!\}$

Conservative Extension Property of SQL

Proposition 9 *The extended rewrite system is sound.*

Proposition 10 *The extended rewrite system is strongly normalizing.*

Theorem 11 *Let $e : s$ be a normal form in our theoretically reconstructed SQL. Then $ht(e) \leq \max(\{ht(s)\} \cup \{ht(t) \mid t$ is the type of a free variable occurring in $e\})$.*

Therefore, $\mathcal{NRC}(\mathbb{Q},+,\cdot,-,\div,\Sigma,=,\leq^{\mathbb{Q}})$ has the conservative extension property.

Finite-Cofiniteness

Theorem 12 *Let $P : \mathbb{Q} \to bool$ be a predicate definable in $\mathcal{NRC}(\mathbb{Q},+,\cdot,-,\div,\Sigma,=,\leq^{\mathbb{Q}})$. Then either P holds for finitely many natural numbers or P fails for finitely many natural numbers.*

Hence a predicate on natural numbers is definable in SQL only if it is finite-cofinite.

It follows that

Corollary 13 *Parity test on natural numbers is not definable in $\mathcal{NRC}(\mathbb{Q},+,\cdot,-,\div,\Sigma,=,\leq^{\mathbb{Q}})$.*

Explanation

Note that $P : \mathbb{Q} \to \mathit{bool}$ has height 0. By the conservative extension property on $\mathcal{NRC}(\mathbb{Q},+,\cdot,-,\div,\Sigma,=,\leq^{\mathbb{Q}})$, any implementation of it in $\mathcal{NRC}(\mathbb{Q},+,\cdot,-,\div,\Sigma,=,\leq^{\mathbb{Q}})$ is equivalent to one that does not use sets. It is easy to see that such an implementation must be equivalent to something like:

$$\bigvee \bigwedge p_i(n) \leq 0$$

where $p_i(n)$ are polynomials in n.

Finite-cofiniteness then follows straightforwardly from the fact that polynomials have finite number of roots.

A more advanced "finite-cofiniteness" result.

Definition 14 A binary relation $O : \{b \times b\}$ is a **k-multi-cycle** if it is nonempty and it has the form

$$\left\{ \begin{array}{cccc} (o_1^1, o_2^1), (o_2^1, o_3^1), & \ldots, & (o_{h-1}^1, o_h^1), (o_h^1, o_1^1), \\ \vdots & & \vdots \\ (o_1^m, o_2^m), (o_2^m, o_3^m), & \ldots, & (o_{h-1}^m, o_h^m), (o_h^m, o_1^m) \end{array} \right\}$$

where $h \geq k$ and o_i^j are all distinct. That is, it is a graph containing $m \geq 1$ unconnected cycles of equal length $h \geq k$.

Theorem 15 *Let* $P : \{b \times b\} \to bool$ *be a predicate definable in* $\mathcal{NRC}(\mathbb{Q}, +, \cdot, -, \div, \Sigma, =, \leq^{\mathbb{Q}})$*. Then there is* k *such that either* P *holds for all* k*-multi-cycles or* P *fails for all* k*-multi-cycles.*

Therefore $\mathcal{NRC}(\mathbb{Q}, +, \cdot, -, \div, \Sigma, =, \leq^{\mathbb{Q}})$ is not able to distinguish one multi-cycle from another as long as the multi-cycles are long enough. Thus, solving several long-standing conjectures on SQL:

Corollary 16 *The parity test on cardinality of unordered graphs is undefinable in* $\mathcal{NRC}(\mathbb{Q}, +, \cdot, -, \div, \Sigma, =, \leq^{\mathbb{Q}})$*.*

Corollary 17 *Transitive closure on unordered graphs is not definable in* $\mathcal{NRC}(\mathbb{Q}, +, \cdot, -, \div, \Sigma, =, \leq^{\mathbb{Q}})$*.*

Explanation

Normal forms induced by the conservative extension theorem have a special trait:

$$\Sigma \left\{ \left| \ \ldots\ \Sigma \left\{ \left| \begin{array}{l} \mathit{if}\ P_1 \\ \mathit{then}\ f_1 \\ \vdots \\ \mathit{else\ if}\ P_h \\ \mathit{then}\ f_h \\ \mathit{else}\ f_{h+1} \end{array} \right| \ x_1 \in R \ \right\| \right\} \ldots \ \right| \ x_n \in R \ \right\| \right\}$$

Assume that the probability, in terms of the number of edges in R, of P_i being true and $P_{j<i}$ being false is p_i. Then the above expression is equivalent to the polynomial

$$N^n \cdot (p_1 \cdot f_1 + \ldots + p_{h+1} \cdot f_{h+1})$$

with N being the number of edges in R.

Bounded Degree Property

Let b be an infinite base type having only the equality test.

Definition 18 A language $\mathcal{L}$ is said to have the **bounded degree property** if for any $f : \{b \times b\} \rightarrow \{b \times b\}$ definable in $\mathcal{L}$ and for any number k there exists a number c such that $card(deg(f(G))) \leq c$ for any graph G satisfying $deg(G) \subseteq \{0, 1, \ldots, k\}$.

That is, $\mathcal{L}$ cannot transform an unordered graph of small degree to a graph having a large number of distinct degrees.

Proposition 19 *Relational calculus or $\mathcal{FO}(=)$ has the bounded degree property.*

Motivation I

The bounded degree property reveals a fundamental structure of the query language. Thus it let us provide *simple* and *uniform* proofs of many negative results on the expressive power of the query language.

For example, it is not difficult to show that first-order logic cannot express queries such as transitive closure *in general*. However, it is often hard to show that first-order logic cannot express these queries *in special cases* such as transitive closure of a chain or a binary tree.

Bounded degree property gives us a tool to deal with inexpressibility uniformly.

For instance, the number of degrees in the transitive closure of a chain or a tree must clearly depend on the depth of the chains and trees and hence cannot be bounded by f(k). Then we can conclude trivially that any language having the bounded degree property would not be able to express transitive closure in the general case as well as in special cases such as chains and trees.

Motivation II

The de-facto database query language is founded upon the relational calculus. However, it differs from the latter in that it has arithmetics, aggregate functions, and group-by.

Arithmetics and aggregate functions give SQL extra power over the relational calculus. Thus, many results on the relational calculus cannot be transferred to SQL.

Bounded degree property was known to hold for the relational calculus. By the result of this talk, it holds for SQL. Then any result obtain from general principles based on the bounded degree property, can be transferred from relational calculus to SQL.

Topological Parameters

Definition 20 Let G denote a graph and x_1, ..., x_m denote edges in G. A **neighbourhood** of x_1, ..., x_m is a *satisfiable* formula $M(G, x_1, \ldots, x_m)$ of the form

$$\begin{aligned}\exists y_1 \ldots y_n \in G.\ & \Psi(x_1, \ldots, x_m, y_1, \ldots, y_n) \wedge \\ & \Delta(G, x_1, \ldots, x_m, y_1, \ldots, y_n) \wedge \\ & \Phi(x_1, \ldots, x_m, y_1, \ldots, y_n) \wedge \\ & \forall z \in G. \Theta(x_1, \ldots, x_m, y_1, \ldots, y_n, z)\end{aligned}$$

- $\Psi(x_1, \ldots, x_m, y_1, \ldots, y_n)$ is quantifier-free formula that specifies the connections between x_1, ..., x_m, y_1, ..., y_n such that they form several connected components and that each of these components involves some $x_{1 \leq i \leq m}$;
- $\Delta(G, x_1, \ldots, x_m, y_1, \ldots, y_n)$ is a formula that specifies the in- and out-degrees of the endpoints of x_1, ..., x_m, y_1, ..., y_n in G;
- $\Phi(x_1, \ldots, x_m, y_1, \ldots, y_n)$ is a formula that says y_1, ..., y_n are distinct edges and that y_1, ..., y_n are distinct from x_1, ..., x_m;
- $\Theta(x_1, \ldots, x_m, y_1, \ldots, y_n, z)$ is a formula that says z has an endpoint different all the endpoints of x_1, ..., x_m, y_1, ..., y_n whenever z is not equal to any of these edges.

We work with neighbourhoods that are "complete" up to a specified radius in the following sense.

Definition 21 A neighbourhood $M(G, x_1, \ldots, x_m)$ of G is said to have **radius** r if the following two conditions hold:

- All nodes in the $\Psi(x_1, \ldots, x_m, y_1, \ldots, y_n)$ part of $M(G, x_1, \ldots, x_m)$ are at most r edges away from both end-points of some $x_{1 \le i \le m}$. (We do not require the nodes to be close to the same x_i.)
- All nodes in the $\Psi(x_1, \ldots, x_m, y_1, \ldots, y_n)$ part of $M(G, x_1, \ldots, x_m)$ that are less than r edges away from any endpoints of $x_{1 \le i \le m}$ must have as many in-coming and out-going edges in $\Psi(x_1, \ldots, x_m, y_1, \ldots, y_n)$ as specified by the $\Delta(G, x_1, \ldots, x_m, y_1, \ldots, y_n)$ part of $M(G, x_1, \ldots, x_m)$.

Now we define the topological parameters of a graph. These are defined in terms of the graph and do not refer to any particularly edges of the graph. Note that they can be expressed in SQL.

Definition 22 A **topological parameter** of G with respect to a neighbourhood formula $M(G, x)$ having radius r is the number of x in G satisfying $M(G, x)$. Thus every topological parameter of a graph is just a count of the number of subgraphs having the associated topology.

Definition 23 A **topological polynomial** $Q(G)$ is a polynomial defined in terms of topological parameters of G.

Definition 24 A **topological predicate** $N(G)$ is a Boolean combination of polynomial (in)equations defined in terms of topological parameters of G.

A Normal Form for SQL

We need a key lemma before we establish our normal form for SQL.

Lemma 25 *Let* $e(G, x_1, \ldots, x_m) : \mathbb{Q}$ *be an expression of SQL of the form*

$$\Sigma \left\{ \left| \begin{array}{l} \mathit{if}\ M(G, x, x_1, \ldots, x_m) \wedge\ N(G) \\ \mathit{then}\ Q(G) \\ \mathit{else}\ 0 \end{array} \right| x \in G \right| \right\}$$

where $M(G, x, x_1, \ldots, x_m)$ *is a neighbourhood of* x, x_1, ..., x_m *in* G *having radius* r, $N(G)$ *is a topological predicate, and* $Q(G)$ *is a topological polynomial. Let* G *denote any graph of degree at most* k *and* x_1, ..., x_m *be restricted to edges in* G. *Suppose* $M'(G, x_1, \ldots, x_m)$ *is a neighbourhood of* x_1, ..., x_m *having radius* $r' > 2 \cdot r$ *that is consistent with* $M(G, x, x_1, \ldots, x_m)$. *Then there is a topological polynomial* $Q'(G)$ *such that* $e(G, x_1, \ldots, x_m)$ *is equivalent to* $Q'(G) \cdot Q(G)$ *whenever* $M'(G, x_1, \ldots, x_m)$ *and* $N(G)$ *hold.*

Applying the lemma in an inside-out fashion to expressions, we obtain

Proposition 26 *Let $G : \{b \times b\}$ denote a graph of degree at most k. Let $e(G) : s$ be an expression of SQL. Then $e(G)$ is equivalent to an expression of the form*

$$\begin{array}{l} \textit{if } N_1(G) \textit{ then } e_1(G) \\ \vdots \\ \textit{else if } N_d(G) \textit{ then } e_d(G) \\ \textit{else } e_{d+1}(G) \end{array}$$

where each $N_i(G)$ is a topological predicate and each $e_i(G)$ is in $\mathcal{FO}(=)$.

Since $\mathcal{FO}(=)$ has the bounded degree property, there is a bound on the number of distinct degrees in each $e_i(G)$. The bound on the number of distinct degrees of $e(G)$ is just the maximum of these bounds. So

Theorem 27 *SQL has the bounded degree property.*

Locality

Given two structures A and B, we write $(A,\vec{a}) \leftrightarrows_d (B,\vec{b})$ if there exists a bijection $f : A \to B$ such that $ntp_d^A(\vec{a}c) = ntp_d^B(\vec{b}f(c))$ for every $c \in A$. And we write $\vec{a} \approx_d^{A,B} \vec{b}$ if there is an isomorphism $N_d^A(\vec{a}) \to N_d^B(\vec{b})$ that maps $\vec{a}$ to $\vec{b}$.

Definition 28 A formula $\varphi(\vec{x})$ is **Gaifman local** if there exists $r \geq 0$ such that for any finite structure A and $\vec{a}$, $\vec{b}$ over A, $\vec{a} \approx_d^{A,A} \vec{b}$ implies $A^* \models \varphi(\vec{a})$ iff $B^* \models \varphi(\vec{b})$.

Definition 29 A formula $\varphi(\vec{x})$ is **Hanf local** if there exists $d \geq 0$ such that for all finite structures A and B, $(A,\vec{a}) \leftrightarrows_d (B,\vec{b})$ implies $A^* \models \varphi(\vec{a})$ iff $B^* \models \varphi(\vec{b})$.

Theorem 30
- *Hanf locality implies Gaifman locality.*
- *Gaifman locality implies bounded degree property.*

Infinitary Logics with Aggregate Functions

Assume we have two signatures on $\mathbb{Q}$: Ω of functions and predicates, and Θ of aggregate functions.

The terms and formula of the two-sorted logic $\mathbb{L}_{\text{aggr}}(\Omega, \Theta)$, over $\mathbb{Q}$ and an unordered type b, are those usual ones in a logic, plus

- infinitary conjunction
- infinitary disjunction
- counting terms (with $\vec{y}$ non-numerical)

$$\#\vec{y}.\varphi(\vec{x}, \vec{y})$$

- aggregate terms (with $\vec{y}$ non-numerical, $\mathcal{F} \in \Theta$)

$$\mathsf{Aggr}_{\mathcal{F}}\vec{y}.(\phi(\vec{x}, \vec{y}), t(\vec{x}, \vec{y}))$$

The rank of formula and terms is defined as

- Variables or constants, the rank is 0
- $\mathsf{rk}(P(t_1, \ldots, t_n)) = \max \mathsf{rk}(t_i)$
- $\mathsf{rk}(f(t_1, \ldots, t_n)) = \max \mathsf{rk}(t_i)$
- $\mathsf{rk}(\bigvee \phi_i)) = \sup \mathsf{rk}(\phi_i)$
- $\mathsf{rk}(\neg\phi) = \mathsf{rk}(\phi)$
- Quantification over type b, $\mathsf{rk}(\exists x \phi) = \mathsf{rk}(\phi) + 1$
- Quantification over $\mathbb{Q}$, $\mathsf{rk}(\exists q \phi) = \mathsf{rk}(\phi)$
- $\mathsf{rk}(\#\vec{y}.\phi) = \mathsf{rk}(\phi) + |\vec{y}|$
- $\mathsf{rk}(\mathsf{Aggr}_{\mathcal{F}}\vec{y}.(\phi, t)) = \max(\mathsf{rk}(\phi), \mathsf{rk}(t)) + |\vec{y}|$

Definition 31 *The formula and terms of $\mathcal{L}_{\text{aggr}}(\Omega, \Theta)$ are those of $\mathbb{L}_{\text{aggr}}(\Omega, \Theta)$ restricted to finite rank. If there is no restriction on the signature, we write $\mathcal{L}_{\text{aggr}}(\mathsf{All}, \mathsf{All})$.*

Let $\mathcal{NRL}^{\text{aggr}}(\Omega, \Theta)$ denote $\mathcal{NRC}(\mathbb{Q},+,\cdot,-,\div,\Sigma,=,\leq^{\mathbb{Q}})$ augmented with functions, predicates, and aggregate functions from Ω and Θ.

Let $\mathcal{RL}^{\text{aggr}}(\Omega, \Theta)$ denote $\mathcal{NRL}^{\text{aggr}}(\Omega, \Theta)$ restricted to expressions of height 1, ie. the relational queries.

Proposition 32 $\mathcal{NRL}^{\text{aggr}}(\mathsf{All}, \mathsf{All})$ *has the conservative extension property. Thus, any function from flat relations to flat relations definable in* $\mathcal{NRL}^{\text{aggr}}(\mathsf{All}, \mathsf{All})$ *is definable in* $\mathcal{RL}^{\text{aggr}}(\mathsf{All}, \mathsf{All})$.

Theorem 33 *For any closed expression* $e : \{b \times \cdots \times b\}$ *in* $\mathcal{RL}^{\text{aggr}}(\mathsf{All}, \mathsf{All})$ *over a schema* σ*, there is a formula* $\varphi(\vec{x})$ *in* $\mathcal{L}_{\text{aggr}}(\mathsf{All}, \mathsf{All})$ *such that for any* σ*-structure* A*,* $e(A) = \{\vec{a} \mid A \models \varphi(\vec{a})\}$.

Definition 34 *The logic* $\mathcal{L}_{\mathrm{C}}$ *is defined to be* $\mathcal{L}_{\mathrm{aggr}}(\emptyset, \emptyset)$*; that is, aggregate terms are not allowed.*

Proposition 35

- *For every formula in* $\mathcal{L}_{\mathrm{aggr}}(\mathsf{All}, \mathsf{All})$*, there is an equivalent one of the same rank in* $\mathcal{L}_{\mathrm{C}}$*.*
- *For every formula in* $\mathcal{L}_{\mathrm{C}}$*, there is an equivalent one of the same rank in* $\mathcal{L}_{\mathrm{aggr}}(\mathsf{All}, \mathsf{All})$*.*

Theorem 36 *All formula without numeric free variables in* $\mathcal{L}_{\mathrm{C}}$ *are Hanf local and therefore Gaifman local. Thus, so are those in* $\mathcal{L}_{\mathrm{aggr}}(\mathsf{All}, \mathsf{All})$*.*

Corollary 37 *All relational queries in* $\mathcal{NRL}^{\mathrm{aggr}}(\Omega, \Theta)$ *are Hanf local and Gaifman local.*