Measuring Infinite Relations

(Extended Abstract)

Jan Chomicki*
Kansas State University
chomicki@cis.ksu.edu

Gabriel Kuper
ECRC
kuper@ecrc.de

Abstract

We define a new aggregation operator μ_n for constraint databases that makes it possible to measure infinite subsets of the n-dimensional space defined by constraints. We show that it is well defined for real linear arithmetic constraints and integer linear arithmetic constraints together with periodicity constraints. We also show that relational algebra augmented with μ_n is closed in the real case and, under certain restrictions, in the integer case as well.

1 Introduction

Constraint databases contain infinite relations that are finitely representable using constraints [KKR90]. In this context, aggregation operators familiar from relational query languages like SQL need to be re-thought. Some new aggregation operators, e.g., **area**, with quite intuitive meaning have already been proposed for constraint databases [Kup95]. That paper describes a general framework, modelled after [Klu82], for adding aggregate operators to constraint query languages. As long as the representation of the result is not an issue, Klug's tuple calculus and algebra can be adapted to the constraint setting. Once we consider the representation, however, we encounter the problem that the query language is not necessarily *closed*: the result of a query that uses aggregation may fail to be representable using the constraint language of the database.

Example 1.1 [Kup95] If the constraint database contains order constraints, the result of applying **area** may fail to be representable using linear arithmetic constraints. Consider a (generalized) relation

$$R = \{(x, y, z) : x \leq y \wedge x \geq 0 \wedge x \leq z\}.$$

*Research done while visiting ECRC.

For any given z, the corresponding plane fragment is shown in Figure 1. Thus, applying **area** to the first and second attributes of R results in the relation

$$S = \{(w, z) : w = z^2/2\}$$

which is not representable using linear arithmetic constraints.

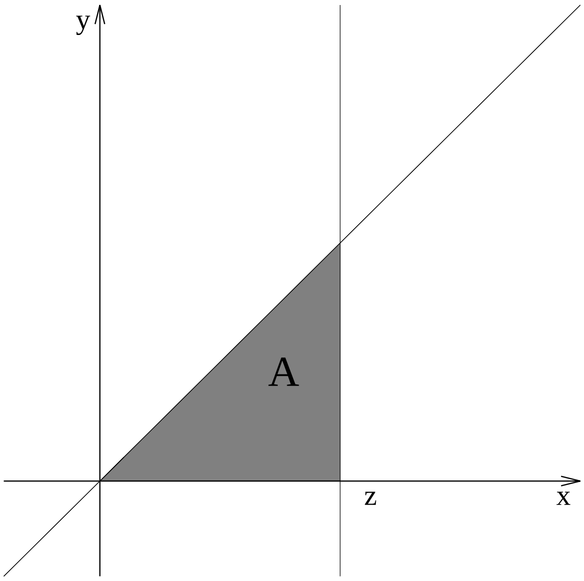

Figure 1: $area(A) = z^2/2$

∎

[Kup95] proposes several approaches to deal with this problem, on the one hand, ways to restrict the query language, and on the other, to restrict the classes of aggregate operators allowed. The second approach was illustrated by the *max* aggregate operator, that (trivially) produces a closed language.

In this paper we show that there are nontrivial aggregate operators, under which the query language of [Kup95] is closed. We propose here a new aggregation operator μ_n for constraint databases. Its meaning is very intuitive: $\mu_n(A)$ represents how much of the n-dimensional space is taken by the set A. In other words, $\mu_n(A)$ is the asymptotic probability that an arbitrary point in the n-dimensional space belongs to the set A. We consider constraint databases with linear arithmetic constraints over reals or integers. In the latter case, we consider also periodicity constraints. Linear arithmetic constraints have been extensively studied in Operations Research [Sch86]. They provided one of the first constraint domains to which the CLP scheme of [JL87] was applied [JMSY92], leading to further theoretical studies [Las90, LM92]. Constraint databases with linear arithmetic constraints were studied in [ACGK94, BJM93, BK95, BLLM95, RS93, Sri93]. Periodicity constraints were used in [KSW90, TCR94] to represent periodic data in temporal databases.

While in the real number case, our results are developed in the standard framework of measure theory [Hal50], the integer case is somewhat novel. In particular, our definition of μ_n makes it possible to state precisely that $\mu_1(Even) = 1/2$ where *Even* is the set of all even integers.

For the classes of constraint databases mentioned above, relational algebra and calculus with μ_n have the closure property in the real case. This answers an open question from [Kup95] about the existence of an aggregation operator which does not violate closure but is not expressible in the underlying constraint language. In the integer case closure is obtained only if the language of periodicity and linear arithmetic constraints is extended with rational equality constraints. The reason for this is clear: $\mu_n(A)$ is a rational number but can be noninteger.

We believe that the tools and techniques developed in the course of this research may prove useful in the study of other asymptotic aggregation operators in constraint databases.

The plan of the paper is as follows. In section 2 we define the basic notions of constraint databases and the constraint languages used in this paper. In section 3 we define the aggregation operator μ_n and discuss its fundamental properties. We study real linear arithmetic constraints in section 4 and integer linear arithmetic constraints together with periodicity constraints in section 5. In section 6 we discuss closure and in section 7 - related and further work.

2 Basic notions

For a full account of constraint databases, see [KKR90, GS94]. The basic definitions are as follows:

Definition 2.1 Let Φ be a class of constraints.

1. A *generalized n-tuple* (over variables $x_1, \ldots, x_n$) is a finite conjunction $\varphi_1 \wedge \cdots \wedge \varphi_N$, where each $\varphi_i, 1 \leq i \leq N$, is a constraint in Φ. Furthermore, the variables in each φ_i are all free and among $x_1, \ldots, x_n$.
2. A *generalized relation of arity n* is a finite set $r = \{\psi_1, \ldots, \psi_M\}$, where each $\psi_i, 1 \leq i \leq M$ is a generalized n-tuple over the same variables $x_1, \ldots, x_n$.
3. The *formula corresponding to a generalized relation* r is the disjunction $\psi_1 \vee \cdots \vee \psi_M$. We use ϕ_r to denote the quantifier-free formula corresponding to the relation r.
4. A *generalized database* is a finite set of generalized relations.□

In database theory, a n-ary relation r is a finite set of n-tuples (or points in a n-dimensional space) and a database is a finite set of relations. However, the relational calculus and algebra can be developed without the finiteness assumption for relations. We will use the term *unrestricted relation* for finite or infinite sets of points in a n-dimensional space. In order to be able to do something useful with such unrestricted relations, we need a finite representation that we can manipulate. This is exactly what the generalized tuples provide.

Definition 2.2 Let Φ be a class of constraints interpreted over domain D, r a generalized relation of arity n with constraints in Φ, and let $\phi_r = \phi_r(x_1, \ldots, x_n)$ be the formula corresponding to r with free variables $x_1, \ldots, x_n$. The generalized relation r represents the unrestricted n-ary relation which consists of all $(a_1, \ldots, a_n)$ in D^n such that $\phi_r(a_1, \ldots, a_n)$ is true. A generalized database represents the finite set of unrestricted relations that are represented by its generalized relations. □

A query ϕ on a constraint database is a first-order formula, whose predicates are constraints or generalized relation symbols. The semantics of a query, are, intuitively, the mapping from unrestricted relations to unrestricted relations defined by this formula. Such a query is well-defined if the result of substituting a generalized relation for each occurrence of the corresponding relation symbol, is equivalent to some generalized relation over the same constraint domain. A query algebra can also be defined, using natural extensions of the normal relational algebra [KG94, PVdBVG94]. (The extension to aggregation will be described later in this paper.)

In this paper, we consider the following constraint languages:

- linear arithmetic constraints of the form

$$a_1x_1 + \cdots + a_mx_m\theta a_0$$

with rational coefficients $a_0, \ldots, a_m$ and $\theta \in \{<, \leq\}$.

- periodicity constraints of the form

$$x \equiv_k y + c$$

  where c and k are nonnegative integers, $k \geq 2$ and $c < k$. Such a constraint means that $x - y = m \cdot k + c$ for some integer m. (A *simple* constraint is of the form $x \equiv_k c$.)

- equality constraints of the form $x = c$ or $x \neq c$ where c is rational.

We consider two basic n-dimensional spaces: R^n (reals) and Z^n (integers), whose elements are vectors of length n. While linear arithmetic constraints are applicable both to reals and integers, periodicity constraints apply only to the latter.

In the context of generalized relations we can assume WLOG that the periodicity constraints are simple. Take any generalized tuple $x \equiv_k y \wedge \alpha$. It can be replaced by an equivalent set of tuples:

$$\begin{array}{l} x \equiv_k 0 \wedge y \equiv_k 0 \wedge \alpha. \\ x \equiv_k 1 \wedge y \equiv_k 1 \wedge \alpha. \\ \ldots \\ x \equiv_k (k-1) \wedge y \equiv_k (k-1) \wedge \alpha. \end{array}$$

A constraint of the form $x \equiv_k y + c$ is handled in a similar way.

Definition 2.3 A set of simple periodicity constraints C is *minimal* if for every variable x, C contains only one constraint of the form $x \equiv_k c$.

Lemma 2.1 *For every satisfiable finite set of periodicity constraints C there exists an equivalent minimal set of periodicity constraints which can be effectively constructed.*

Proof: By the Chinese Remainder theorem. ∎

We will assume that the set of periodicity constraints in every generalized tuple is minimal. (If this set is not satisfiable, then the tuple can be ignored.)

3 Aggregation

Let $V_n(X)$ be the n-dimensional volume of a set X. For R^n, $V_n(X)$ is the Lebesgue measure $l_n(X)$ of X. For Z^n, $V_n(X) = |X|$ is the cardinality of X. Let C_k stand for the closed n-dimensional cube of side k, centered around the origin $(0, \ldots, 0)$. We use the same notation for a cube in R^n and Z^n.

Definition 3.1 Let A be a subset of the n-dimensional space. The aggregation operator μ_n is defined as

$$\mu_n(A) = \lim_{k \to \infty} \frac{V_n(A \cap C_k)}{V_n(C_k)}.$$

Example 3.1 If A is a half-n-space, then $\mu_n(A) = \frac{1}{2}$. Consider Z^1 now. If $Even$ is the set of all even integers, then $\mu_1(Even) = \frac{1}{2}$. If $PosEven$ is the set of all even integers greater than zero, then $\mu_1(PosEven) = \frac{1}{4}$. ∎

The operators μ_n are useful mainly for making inferences about *weighted averages*.

Example 3.2 Assume the measurements of some quantity, e.g., luminosity, over a number of regions are given as a relation R of the form (v_i, c_i) where v_i is a value of this quantity and c_i a constraint formula defining the corresponding region (the regions are infinite and unbounded if, for example, we are dealing with the measurements of a space telescope). Then $R < 1, \mu_3 >$ (in the notation of [Klu82] it means "*group on the first attribute of R, apply the aggregation operator μ_3 to the remaining three*") returns a set of pairs (v_i, m_i) where m_i is the portion of the 3-dimensional space where the measured quantity assumes the value v_i. This relation may in turn be used to determine the weighted average of the measured quantity. ∎

Examples of applying μ_2 are also presented in Figures 2 and 3.

The following properties hold:

Proposition 3.1 *If $\mu_n(A)$ is defined, then*

$$0 \leq \mu_n(A) \leq 1.$$

Proposition 3.2 $\mu_n(A) = 0$ *if the projection of A onto any line is contained in some interval.*

Proposition 3.3 *If $\mu_n(A)$ and $\mu_n(B)$ are defined, then $\mu_n(A \cup B)$ and $\mu_n(A \cap B)$ are defined and*

$$\mu_n(A \cup B) = \mu_n(A) + \mu_n(B) - \mu_n(A \cap B).$$

Proposition 3.4 *μ_n is invariant under the translations of the n-dimensional space.*

Proof: Consider a translation by a vector v with a largest component t. Then the cube of size $n - t$ centered around v is contained in the cube of size n centered around the origin, which in turn is contained in the cube of size $n + t$ centered around v. This implies the result. ∎

4 The n-dimensional space R^n

We study here first the case of a subset A of R^n which is defined by a generalized tuple (a conjunction of constraints). Since the constraints are linear arithmetic constraints, the set A is a convex polyhedron [Sch86]. Then we deal with sets defined by generalized relations, which are finite unions of convex polyhedrons.

Theorem 4.1 *If $A \subseteq R^n$ is defined by a generalized tuple with linear arithmetic constraints, then $\mu_n(A)$ is well defined and rational.*

Proof: Let $A_k = A \cap C_k$. We have to establish that $V_n(A_k)$ is well defined. The set A_k is a convex polytope defined by the linear constraints of A and those of C_k (of the form $x_i \leq k/2$ or $x_i \geq -k/2$). Take $k > k_0$ such that the interior of C_{k_0} contains all the intersection points of the facets of A. Then the vertices of A_k can be classified into *internal* (corresponding to some of the intersection points of the facets of A) and *external* (lying on the surface of C_k).

Consider now a triangulation T_k of the polytope A_k into simplices $S_1^k, \ldots S_m^k$. Clearly

$$V_n(A_k) = \sum_{i=1}^{m} V_n(S_i^k).$$

A simplex S_i^k is the convex hull of $n+1$ points $v_{i,0}^k, \ldots, v_{i,n}^k$. The volume of S_i^k is defined as follows [GK94]

$$V_n(S_i^k) = \frac{1}{n!} |det(v_{i,1}^k - v_{i,0}^k, \ldots, v_{i,n}^k - v_{i,0}^k)|$$

(here $|c|$ denotes the absolute value of c). Now some of the vertices of S_i^k are internal to A_k and their coordinates do not depend on k. The remaining ones are external and their cooordinates can depend linearly on k. Thus $V_n(S_i^k)$ is a polynomial function of k and the degree of this polynomial is at most n. Consequently, $V_n(A_k)$ is a polynomial in k with degree at most n.

It is easy to see that such a construction can be done for any k. Moreover, for every $k > k_0$ the triangulation T_k can be picked in essentially the same way: only the coordinates of simplex vertices corresponding to the external vertices of the polytope will change depending on k. But this dependence is already captured in the definition of the volume of every simplex.

Now $V_n(C_k) = k^n$ and thus

$$\mu_n(A) = \lim_{k \to \infty} \frac{V_n(A_k)}{V_n(C_k)} = a_n$$

where a_n is the degree n coefficient in the polynomial $V_n(A_k)$.

Because the coefficients in the linear constraints defining A and C_k are rational, so are the coordinates of the vertices of A_k and the coefficient a_n. ∎

Computing $\mu_n(A)$ for A defined by a conjunction of linear arithmetic constraints is thus no harder than computing the volume of a convex polytope given as a conjunction of linear arithmetic constraints. This is known to be polynomial for a fixed dimension n and in general exponential in n [GK94].

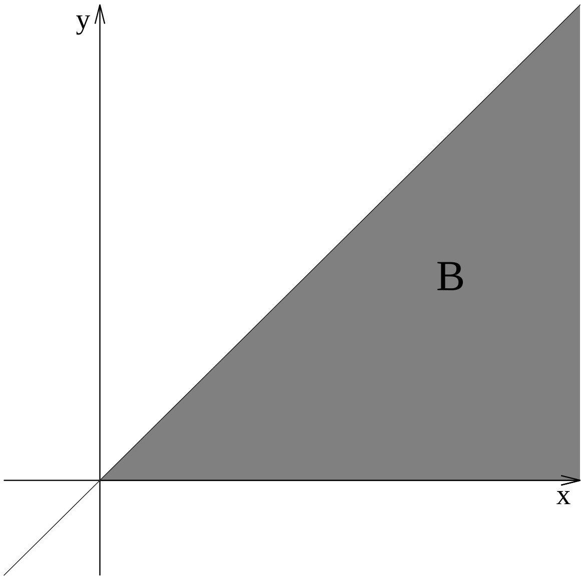

Figure 2: $\mu_2(B) = 1/8$

Theorem 4.2 *If $A \subseteq R^n$ is defined by a generalized relation, then $\mu_n(A)$ is well defined and rational.*

Proof: As A is now a finite union of polyhedrons, $\mu_n(A)$ is easily defined using the Principle of Inclusion-Exclusion. ∎

The Principle of Inclusion-Exclusion requires the computation of $\mu_n(A_B)$ for every $B = \{t_1, \ldots, t_m\} \subseteq A$ and A_B defined by $t_1 \wedge \cdots \wedge t_m$. Thus the data complexity of computing $\mu_n(A)$ for A being a union of sets defined by conjunctions of linear arithmetic constraints is exponential.

We prove now one more property of linear arithmetic constraints which will be useful later.

Proposition 4.3 *Let A be defined by a set of linear arithmetic constraints (and consequently by a set of hyperplanes). Let A' be defined by the same hyperplanes as A, except for one which is translated by a fixed vector. The $\mu_n(A') = \mu_n(A)$.*

Proof: WLOG assume $A \subseteq A'$. Then $\mu_n(A' - A) = 0$ by Proposition 3.2 and the thesis follows. ∎

The above property is illustrated in Figures 2 and 3. Notice that the only difference between B and C is in the shift of one bounding hyperplane (line). We have

$$\mu_2(B) = \mu_2(C).$$

On the other hand, **area** does not have the above property.

Notice also that Proposition 4.3 is different from Proposition 3.4, as we are not considering here a translation of the entire space.

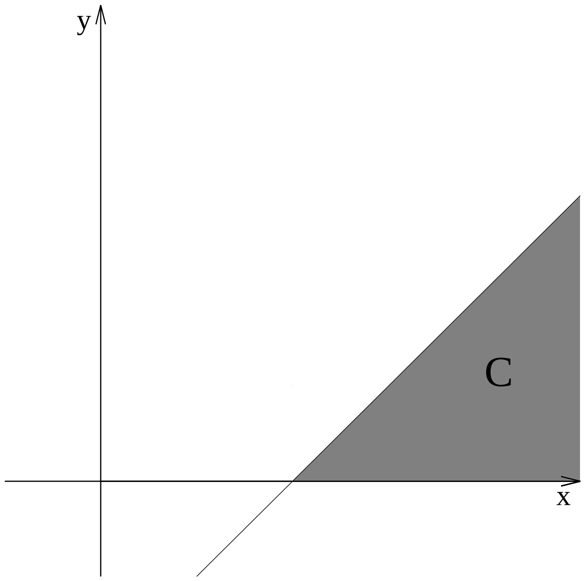

Figure 3: $\mu_2(C) = 1/8$

5 The n-dimensional space Z^n

We study here first the case of a subset A of Z^n which is defined by a generalized tuple which is a conjunction of linear arithmetic constraints. Then we consider periodicity constraints separately and in combination with linear arithmetic constraints. The case of A defined by a generalized relation follows, like in the preceding section.

The reason for considering linear arithmetic together with periodicity constraints is as follows. To obtain closure for relational calculus queries over constraint databases with linear arithmetic constraints over integers, one needs to be able to eliminate quantifiers from formulas with such constraints. This is only possible if periodicity constraints are used. The appropriate quantifier elimination procedure is presented, e.g., in [Wil76].

Example 5.1 Consider the formula $\exists y.\ x = y+y$. The variable y can be eliminated to yield the formula $x \equiv_2 0$. ∎

5.1 Linear arithmetic constraints

Theorem 5.1 *If $A \subseteq Z^n$ is defined by a generalized tuple with linear arithmetic constraints, then $\mu_n(A)$ is well-defined and rational.*

Proof: To distinguish between the sets in Z^n and R^n, we add appropriate superscripts: C_k^Z (resp. C_k^R) will denote a cube of side k in Z^n (resp. R^n), $A = A^Z$ the subset of Z^n defined by a generalized tuple $c_1 \wedge \cdots \wedge c_m$, and A^R the subset of R^n defined by the same tuple.

We need to show that

$$\mu_n(A^Z) = \lim_{k\to\infty} \frac{|A^Z \cap C_k^Z|}{|C_k^Z|} = \lim_{k\to\infty} \frac{l_n(A^R \cap C_k^R)}{l_n(C_k^R)}$$

where $||$ is the cardinality in Z^n and $l_n()$ is the Lebesgue measure in R^n. Let $A' \subseteq R^n$ be defined by a generalized tuple $c'_1 \wedge \cdots \wedge c'_m$ where c'_i defines the hyperplane H_i shifted with respect to the hyperplane H'_i defined by c_i by a vector $\delta' = (\delta'_1, \ldots, \delta'_n)$. The vector δ' has components equal to 1 or -1 depending on which side of H_i lies the interior of A. Intuitively, H'_i should be shifted "towards" the interior of A. A'' is defined similarly but the shift should be "away" from A. It is clear that

$$\frac{l_n(A' \cap C_k^R)}{l_n(C_k^R)} \le \frac{|A^Z \cap C_k^Z|}{|C_k^Z|} \le \frac{l_n(A'' \cap C_k^R)}{l_n(C_k^R)}.$$

But

$$\lim_{k\to\infty} \frac{l_n(A' \cap C_k^R)}{l_n(C_k^R)} = \lim_{k\to\infty} \frac{l_n(A'' \cap C_k^R)}{l_n(C_k^R)}$$

by Proposition 4.3. ∎

Note that the correspondence between the measures in R^n and Z^n does not have to hold for an arbitrary constraint language. For example, A could be defined by the formula $integer(x_1) \wedge \cdots \wedge integer(x_n)$. In Z^n, $\mu_n(A^Z) = 1$. In R^n, $\mu_n(A^R) = 0$. But A can not be defined using linear arithmetic constraints.

5.2 Periodicity constraints

Theorem 5.2 *For every minimal and satisfiable set of simple periodicity constraints $C = \{c_1, \ldots, c_m\}$, the subset A of Z^n defined by C has the property that*

$$V_n(A) = \prod_{i=1}^{m} V_n(A_i)$$

where A_i is the set defined by c_i.

Proof: Each periodicity constraint has the form $x_j \equiv_{k_i} c'_i$. By Prop. 3.4, we may assume that each $c'_i = 0$, and, by minimality and by introducing redundant constraints of the form $x_j \equiv_1 0$ when needed, we may assume that we have exactly one constraint of the form $x_i \equiv_{k_i} 0$, for each i.

If we take a n-dimensional cube in R^n with side k a multiple of $k_1 \cdot \cdots \cdot k_n$, we can then see that

$$\frac{l_n(A \cap C_k)}{l_n(C_k)} = \frac{1}{k_1 \cdots k_n}$$

and

$$\frac{l_n(A_i \cap C_k)}{l_n(C_k)} = \frac{1}{k_i}$$

Bounding the measure from above and below, it is straightforward to show that

$$V_n(A) = \frac{1}{k_1 \cdots k_n} = \prod_{i=1}^{m} V_n(A_i)$$

∎

5.3 Linear and periodicity constraints

Lemma 5.3 *Let $c_1, \ldots, c_m$ are periodicity or linear arithmetic constraints, and $c_0 = (x_i \equiv_q c)$ be a periodicity constraint. Let $\{i_1, \ldots, i_r\} \subseteq \{1, \ldots, n\} - \{i\}$, $a_1, \ldots, a_r \in Z$, and $k' = \lfloor k/2 \rfloor$. Denote by $\mathcal{A}_{a_1,\ldots,a_r}$ the class of formulas of the form*

$$c_1 \wedge \cdots \wedge c_m \wedge x_{i_1} = a_1 \wedge \ldots \wedge x_{i_r} = a_r \wedge \bigwedge_{j=1}^{n} (-k' \leq x_j \leq k')$$

and by $\mathcal{B}_{a_1,\ldots,a_r}$ the class of formulas of the form

$$c_0 \wedge c_1 \wedge \cdots \wedge c_m \wedge x_{i_1} = a_1 \wedge \ldots \wedge x_{i_r} = a_r \wedge \bigwedge_{j=1}^{n} (-k' \leq x_j \leq k')$$

for some integer $a_1, \ldots, a_r$. Consider now formulas $\alpha \in \mathcal{A}_{a_1,\ldots,a_r}$ and $\beta \in \mathcal{B}_{a_1,\ldots,a_r}$. Let X and Y be the subsets of Z^n defined by α and β, respectively. Then if Y is nonempty

$$\lim_{k \to \infty} \frac{|X|}{|Y|} = q.$$

Proof: Reverse induction on r.

Base case: $r = n - 1$. Then α defines a subset of a line. The linear arithmetic constraints in α yield an interval. If α contains some periodicity constraints of the form $x_i \equiv_{q'} c$, then this interval will contain some regularly spaced "holes". A simple counting argument establishes the thesis.

Inductive case: assume the thesis for r, prove for $r - 1$. Let X (resp. Y) be defined by a formula α from $\mathcal{A}_{a_1,\ldots,a_{r-1}}$ (resp. a formula β from $\mathcal{B}_{a_1,\ldots,a_{r-1}}$). Now $\alpha \wedge x_{i_r} = a_r$ is a formula of $\mathcal{A}_{a_1,\ldots,a_r}$ and defines a set X_{a_r}, and $\beta \wedge x_{i_r} = a_r$ is a formula of $\mathcal{B}_{a_1,\ldots,a_r}$ and defines a set Y_{a_r}. We have

$$|X| = \sum_{a_r=-k'}^{k'} |X_{a_r}|$$

and

$$|Y| = \sum_{a_r=-k'}^{k'} |Y_{a_r}|.$$

We use now the inductive assumption about X_{a_r} and Y_{a_r} to establish the thesis. ∎

Theorem 5.4 *Let $A \subseteq Z^n$ be defined by a generalized tuple $c_1 \wedge \cdots \wedge c_m \wedge d_1 \wedge \cdots \wedge d_l$ where $c_1, \ldots, c_m$ are linear arithmetic constraints and $d_1, \ldots, d_l$ are periodicity constraints. Let A' (resp. A'') be the subsets of Z^n defined by the set of constraints $\{c_1, \ldots, c_m\}$ (respectively $\{d_1, \ldots, d_l\}$). Then*

$$\mu_n(A) = \mu_n(A') \cdot \mu_n(A'').$$

Proof: We use Lemma 5.3 repeatedly, putting $r = 0$. ∎

Finally, as for R^n, we can show the following useful proposition:

Proposition 5.5 *Let A be defined by a set of periodicity and linear arithmetic constraints Let A' be defined by the same constraints as A, except that one of the hyperplanes (which is defined only by linear constraints) is translated by a fixed vector. Then $\mu_n(A') = \mu_n(A)$.*

Proof: Identical to Proposition 4.3. ∎

6 Closure

Closure is an essential requirement in the context of query languages. It means that the result of every query can be represented in the language of the database. We consider here closure of queries formulated in relational algebra augmented with the μ_n aggregation operator. The approach is patterned after that of Klug [Klu82].

Klug's approach is as follows. The algebra is augmented by operators of the form $e\langle X, f\rangle$, where e is an algebraic expression, X a subset (in his paper all attributes except one) of the attributes of e, and f an aggregate operator. The intuitive meaning of such an operator is "group the rows by the attributes in X, apply the f operator to the remaining column of all such groups, and output a new table, where each column consists of the attributes in X and the corresponding result of f". His calculus is rather more complicated. Due to the need to compute intermediate tables, in order to apply aggregate operators, a flat, one-step approach, corresponding to the relational calculus does not work. Instead, a 3-part inductive definition is required: for details see the original paper.

As shown in [Kup95], Klug's relational calculus and algebra with aggregation can be extended to constraint databases with minor modifications, at least as far as the underlying semantics on unrestricted relations is concerned. The main differences are the extension of the aggregate operators to functions on relations of arity greater than one, and some measurability requirements to enable us to define aggregates on infinite sets of points. For the purpose of this paper, it is sufficient to assume that the algebra is extended by the following operator:

Definition 6.1 Let e be a relational expression over the set of attributes U, $X \subseteq U$ such that $|U - X| = n$, and I an instance. Then

$$e\langle X, \mu_n\rangle(I) = \{(t[X], y) : t \in e(I) \wedge y = \mu_n(e^X(I,t))\}$$

where

$$e^X(I,t) = \{t'[U - X] : t' \in e(I) \wedge t'[X] = t[X]\}.$$

Our goal is to show closure. This means that given an instance of a generalized relation, the result of a query using aggregation can be expressed in the same constraint language. It suffices to show that, whenever I is an instance of a generalized relation, and X is a subset of the attributes of X, then the result $R\langle X, \mu_n\rangle$ on I is also an instance of a generalized relation over the same constraint language.

We consider first the case of real linear arithmetic constraints.

Theorem 6.1 *Relational algebra augmented with μ_n over generalized databases with real linear arithmetic constraints is closed.*

Proof: Relational algebra without μ_n over this class of generalized databases is closed (this follows from [KKR90]). Assume I is an instance of a relation R represented as a finite generalized relation $r = \{t_1, \ldots, t_m\}$ and the query is $R\langle X, \mu_n\rangle$. For every generalized tuple t_i denote by t_i^X its projection on X which is also a generalized tuple.[1]

Define a *cell* to be a formula of the following form

$$\gamma_1 \wedge \cdots \wedge \gamma_m$$

where every γ_i is t_i^X or its negation. Let C be the set of all cells. Clearly, for every cell $c \in C$ the disjunction τ_c of all formulas t_i corresponding to those t_i^X's that appear positively in the cell describes the $(U - X)$-components of tuples that agree on the attributes in X.

Let $\mathbf{x}$ be an assignment of values to the variables of X. Denote by $A_{c,\mathbf{x}}$ the set of (n-dimensional) points described by τ_c with this assignment of values to X. This is a finite union of convex polyhedra. Now, for any other assignment $\mathbf{y}$ of values to X in the same cell c, $A_{c,\mathbf{y}}$ is described by the same constraints as $A_{c,\mathbf{x}}$ with the exception of the equality constraints corresponding to the variables in X. This is the same as saying that the hyperplane described by these constraints is translated by a vector, and hence $\mu_n(A_{c,\mathbf{x}}) = \mu_n(A_{c,\mathbf{y}})$, by Proposition 4.3.

Denote by M_n^c this value. Then $R\langle X, \mu_n\rangle(I)$ can be represented as a set of generalized tuples $\{t_c \mid c \in C\}$. Each tuple t_c is a conjunction of the cell c and the formula $y = M_n^c$, where y is a new variable. The tuple t_c is clearly a conjunction of linear arithmetic constraints. ∎

We cannot show a similar result for linear arithmetic constraints over the integers with periodicity constraints. The reason for this is simple: μ_n returns a rational, possibly non-integer, number.

[1] This is true for linear constraints, as considered here, but not in general. For polynomial constraints, for example, the projection would be a union of generalized tuples. Another example appears below.

This is the only reason for the lack of closure, and we can therefore use the approach, suggested in [Kup95] of using a many-sorted constraint language to restrict the ways in which aggregation operators can be used. In our case, we will have a 2-sorted language. Sort τ_1 consists of linear arithmetic constraints over the integers with periodicity constraints. Sort τ_2 consists of rational equality constraints. We require that μ_n be applied only to sets of points whose components are of sort τ_1, and that the result be of sort τ_2.

Theorem 6.2 *Under the restrictions described above, relational algebra augmented with μ_n over generalized databases with linear arithmetic constraints over the integers and periodicity constraints is closed.*

Proof: To show that the relational algebra without μ_n is closed, we have to extend the proof in [KG94, PVdBVG94] to sorted languages, with periodicity constraints. As usual, the hardest part is showing closure under projection. This can be shown using the quantifier elimination procedure of [Wil76].

To show closure with aggregation, we use similar techniques to those used to prove Theorem 6.1.

As in that proof, assume that I is an instance of a relation R represented as a finite generalized relation $r = \{t_1, \ldots, t_m\}$ and that the query is $R\langle X, \mu_n\rangle$. For every generalized tuple t_i denote by t_i^X its projection on X. This time, however, the projection can be a set of tuples, rather than a single tuple.

Denote, therefore, the projection t_i^X as the set of tuples $\{t_{i,j}^X\}$ ($j = 1, \ldots, j_i$). Define a cell, similar to before, as a conjunction

$$\gamma_{1,1} \wedge \cdots \wedge \gamma_{m,j_m}$$

where each $\gamma_{i,j}$ is $t_{i,j}^X$ or its negation.

The proof now proceeds in the same way as Theorem 6.1. The key point to observe is that the result depends on the invariance of μ_n under translation of hyperplanes, and this was shown (Proposition 5.5) to hold for the combination of periodicity and linear arithmetic constraints. ∎

7 Related and further work

A notion identical to μ_1 in one-dimensional space of non-negative integers was used in [TW89] to analyze cryptographic protocols. Asymptotic probabilities in discrete spaces are also used for the formulation of $0-1$ laws [Com89]. However, the elements of those spaces are first-order structures, not points of the n-dimensional space.

For further work, other classes of constraints, e.g., polynomial arithmetic constraints may be considered. It is not clear whether μ_n is well defined, and whether the relational algebra is closed in such a case. Also, other asymptotically-defined aggregation operators may be studied. As long as they satisfy the assumptions of Proposition 4.3, closure will be guaranteed (in the context of linear arithmetic constraints).

Acknowledgments

We are grateful to Christopher Hillman and Damian Niwinski for the correspondence on volume computation.

References

[ACGK94] F. Afrati, S. Cosmadakis, S. Grumbach, and G. Kuper. Linear vs. Polynomial Constraints in Database Query Languages. In A. Borning, editor, *PPCP'94, Second International Workshop on Principles and Practice of Constraint Programming*, pages 181–192. Springer-Verlag, LNCS 874, 1994.

[BJM93] A. Brodsky, J. Jaffar, and M.J. Maher. Towards Practical Constraint Databases. In *International Conference on Very Large Data Bases*, Dublin, Ireland, 1993.

[BK95] A. Brodsky and Y. Kornatzky. The LyriC Language: Constraining Objects. In *ACM SIGMOD International Conference on Management of Data*, San Jose, California, 1995.

[BLLM95] A. Brodsky, C. Lassez, J-L. Lassez, and M.J. Maher. Separability of Polyhedra for Optimal Filtering of Spatial and Constraint Data. In *ACM Symposium on Principles of Database Systems*, San Jose, California, 1995.

[Com89] K.J. Compton. $0-1$ Laws in Logic and Combinatorics. In I. Rival, editor, *Algorithms and Order*, pages 353–383. Kluwer, 1989.

[GK94] P. Gritzmann and V. Klee. On the Complexity of Some Basic Problems in Computational Convexity: II. Volume and Mixed Volumes. Technical Report TR-94-31, DIMACS, Rutgers University, New Brunswick, NJ, 1994.

[GS94] S. Grumbach and J. Su. Finitely Representable Databases. In *ACM Symposium on Principles of Database Systems*, pages 289–300, Minneapolis, Minnesota, May 1994.

[Hal50] P. Halmos. *Measure Theory*. Van Nostrand Reinhold, 1950.

[JL87] J. Jaffar and J-L. Lassez. Constraint Logic Programming. In *ACM Symposium on Principles of Programming Languages*, 1987.

[JMSY92] J. Jaffar, S. Michaylov, P.J. Stuckey, and R.H.C. Yap. The CLP(R) Language and System. *ACM Transactions on Programming Languages and Systems*, 14(3):339–395, July 1992.

[KG94] P.C. Kanellakis and D. Q. Goldin. Constraint Programming and Database Query Languages. In *Conference on Theoretical Aspects of Computer Software*. Springer-Verlag, 1994.

[KKR90] P.C. Kanellakis, G.M. Kuper, and P.Z. Revesz. Constraint Query Languages. In *ACM Symposium on Principles of Database Systems*, pages 299–313, Nashville, Tennessee, April 1990. To appear in Journal of Computer and System Sciences.

[Klu82] A. Klug. Equivalence of Relational Algebra and Relational Calculus Query Languages Having Aggregate Functions. *Journal of the ACM*, 29(3):699–717, 1982.

[KSW90] F. Kabanza, J-M. Stevenne, and P. Wolper. Handling Infinite Temporal Data. In *ACM Symposium on Principles of Database Systems*, pages 392–403, Nashville, Tennessee, April 1990. To appear in Journal of Computer and System Sciences.

[Kup95] G. Kuper. Aggregation in Constraint Databases. In *PPCP'93, First International Workshop on Principles and Practice of Constraint Programming*. MIT Press, 1995.

[Las90] J-L. Lassez. Querying Constraints. In *ACM Symposium on Principles of Database Systems*, pages 288–298, Nashville, Tennessee, April 1990.

[LM92] J-L. Lassez and K. McAloon. A Canonical Form for Generalized Linear Constraints. *Journal of Symbolic Computation*, 13:1–24, 1992.

[PVdBVG94] J. Paredaens, J. Van den Bussche, and D. Van Gucht. Towards a Theory of Spatial Database Queries. In *ACM Symposium on Principles of Database Systems*, pages 279–288, Minneapolis, Minnesota, 1994.

[RS93] R. Ramakrishnan and D. Srivastava. Pushing Constraint Selections. *Journal of Logic Programming*, 16(3&4):361–414, 1993.

[Sch86] A. Schrijver. *Theory of Linear and Integer Programming*. J. Wiley and Sons, 1986.

[Sri93] D. Srivastava. Subsumption and Indexing in Constraint Query Languages with Linear Arithmetic Constraints. *Annals of Mathematics and Artificial Intelligence*, 1993.

[TCR94] D. Toman, J. Chomicki, and D.S. Rogers. Datalog with Integer Periodicity Constraints. In *International Logic Programming Symposium*, Ithaca, New York, November 1994. MIT Press.

[TW89] M-J. Toussaint and P. Wolper. Reasoning about Cryptographic Protocols. In J. Feigenbaum and M. Merritt, editors, *Distributed Computing and Cryptography*, volume 2 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 245–262. American Mathematical Society, 1989.

[Wil76] H.P. Williams. Fourier-Motzkin Elimination Extension to Integer Programming Problems. *Journal of Combinatorial Theory A*, 21:118–123, 1976.