RELATIONAL ALGEBRA FOR SPATIO-TEMPORAL DATA MANAGEMENT

Jose Ramon Rios Viqueira
rios@aua.gr
Informatics Laboratory
Agricultural University of Athens
Iera Odos 75, GR 11855 Athens, Greece

1 Introduction

An outline is given of a formalism for the definition of a spatio-temporal extension to the relational model. The formalism considers *temporal* and *spatial quanta* and, based on them, defines relevant data types. This way, a series of relational algebra operations can be defined, that are closed and enable the uniform management of either conventional or spatial or spatio-temporal data.

2 Quanta and Data Types of Time

A *generic* data type for *time* is defined as the set $I_n = \{1, 2, \ldots, n\}$, $n > 0$ [2]. The elements of I_n are called *quanta of time* or (*time*) *instants*. Based on this data type, another *generic* data type is defined, PERIOD(I_n), with elements of the form $[p, q] \equiv \{ i \mid i \in I_n, p \le i \le q\}$ that are called *periods* (*of time*). If the elements of I_n are replaced by n consecutive dates, then the respective data types for time are DATE and PERIOD(DATE). In a similar manner, a variety of data types can be defined like TIME and TIMESTAMP, which are supported in SQL.

3 Spatial Quanta and Spatial Data Types

If $I_m = \{1, 2, \ldots, m\}$, $m > 0$, is a subset of the integers, then I_m^2 is finite. Each element of I_m^2 is a 2-d point that can be identified uniquely by an integer (see Figure 1 for m = 15). If $p \equiv (i, j+1)$ is such a point, then $p_N \equiv (i, j+1)$ and $p_E \equiv (i+1, j)$ are two *neighbors* of p. Points p, p_E, $p_{NE} \equiv \{(i+1, j+1)\}$ and p_N are *corner* points. In Figure 1, 193 and 207 are neighbors of 192, whereas 192, 193, 208 and 207 are *corner* points. Based on this terminology, the following *spatial quanta* are defined [4].

Quantum Point: It is any set {p}, where p is a 2-d point (see {192}, {193}, {208} in Figure 1). The set of all the quantum points is denoted by Q_{POINT}.

Quantum Line: It is either a line segment $ql_{p,q}$ with edges two neighbor 2-d points, p and q, or a quantum point p (see $ql_{184,185}$, $ql_{188,203}$ and {184} in Figure 1). Clearly, $ql_{p,q}$ consists of an infinite number of R^2 elements. The set of all the quantum lines is denoted by Q_{LINE}.

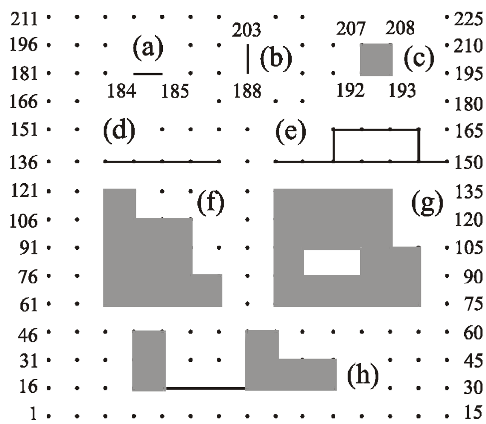
Quantum Surface: It is either the surface of a square $qs_{p,q,r,s}$, where p, q, r and s are corner points or a quantum line (see $qs_{192,193,208,207}$, $ql_{184,185}$ and {184} in Figure 1). Clearly, $qs_{p,q,r,s}$ consists of infinitely many elements of R^2. The set of all these surfaces is denoted by $Q_{SURFACE}$.

Assuming now that the concept of a *connected* set is known, it is defined that a non-empty connected subset $S = \bigcup_i q_i$ of R^2 elements is of a (2-D)

– POINT data type if $q_i \in Q_{POINT}$

– LINE data type if $q_i \in Q_{LINE}$

– SURFACE data type if $q_i \in Q_{SURFACE}$.

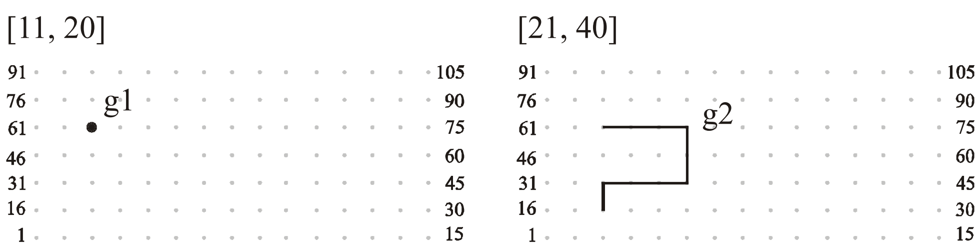


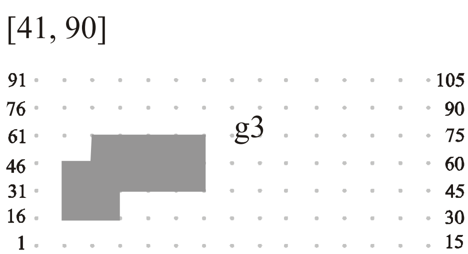
Points
{184}, {185}

Lines
Pure: (a), (b), (d), (e)
Degenerate: {184}, {185}

Surfaces
Pure: (c), (f), (g)
Hybrid: (h)
Degenerated to pure line: (a), (b), (d), (e)
Degenerated to point: {184}, {185}

Figure 1: Spatial quanta and spatial objects.





LAKES

Name	Time	Shape
Morpheas	[11, 20]	g1
Morpheas	[21, 40]	g2
Morpheas	[41, 90]	g3

Figure2: Representation of spatio-temporal data.

Given that $Q_{POINT} \subset Q_{LINE} \subset Q_{SURFACE}$, it follows that POINT $\subset$ LINE $\subset$ SURFACE. Examples of elements of the above data types are given in Figure 1. A point or line or surface element is called a *geo* or *spatial object*.

4 Modeling of Spatio-Temporal Data

Based on the above data types, Figure 2 illustrates the evolution of a spatial object, Morpheas, with respect to time. As can be seen on the relevant plots, during the periods [11, 20], [21, 40] and [41, 90], Morpheas was just a spring, a river and an actual lake, respectively. Relation LAKES, in the same figure, is used to record this evolution. The domain of attribute Shape is SURFACE and each of g1, g2 and g3 is a shorthand of the description of the geometry of Morpheas. In this model therefore, a *map* matches the *geometric interpretation* of the content of a relation that contains spatial data.

In [2] it has been shown that *period* is a special case of a more generic data type, *interval*. Functions and predicates for such data have been defined. A set of relational algebra operations has also been defined. It includes the well-known primitive operations, *Union, Except, Project, Cartesian Product* and *Select*. It also includes *Unfold*, *Fold* and some derived operations, that can be used, amongst others, for the management of *temporal* data. The application of these operations to *spatial* data can be found in [3]. Some of the operations that are supported in [3] match those of *Spatial Union*, *Spatial Except, Intersect*, *Overlay*, etc that either have been defined by other researchers [1, 5, 6] or are supported by commercial Geographic Information Systems. Hence, in the sequel only the application of these operations to *spatio-temporal* data is illustrated.

Operations *Unfold* and *Fold*: Consider a table R(**A**, T, G), where **A** is a (possibly empty) list of attributes and T, G are two attributes of some *time*, *geo* data type, respectively. It is then defined that

Unfold[G, T] ≡ *Unfold*[T](*Unfold*[G](R))
Fold[G, T] ≡ *Fold*[T](*Fold*[G](R)).

As an example, consider the plane in figure 1 and assume that R = {(**a**, [1, 2], $l_{2,3}$), (**a**, [2, 3], $l_{3,18}$)}. Then relation U = *Unfold*[G, T](R) consists of the tuples {(**a**, 1, {2}), (**a**, 1, {3}), (**a**, 1, $l_{2,3}$), (**a**, 2, {2}), (**a**, 2, {3}), (**a**, 2, $l_{2,3}$), (**a**, 2, {18}), (**a**, 2, $l_{3,18}$), (**a**, 3, {3}), (**a**, 3, {18}), (**a**, 3, $l_{3,18}$)}. Assume now that *Fold* is applied to U, i.e. F = *Fold*[G, T](U). Then F consists of the tuples {(**a**, [1, 1], $l_{2,3}$), (**a**, [2, 2], $l_{2,3}l_{3,18}$), (**a**, [3, 3], $l_{3,18}$)}. The way F was obtained reveals that it contains the *evolution* of the geometry of the spatial object **a** respect to time. Note that operations *Fold* and *Unfold* are more general, in that they can be applied to any subset of attributes of a relation, not only to those of some *time* or *space* type. Based on *Fold* and *Unfold*, the functionality of some *spatio-temporal* operations is illustrated below [3].

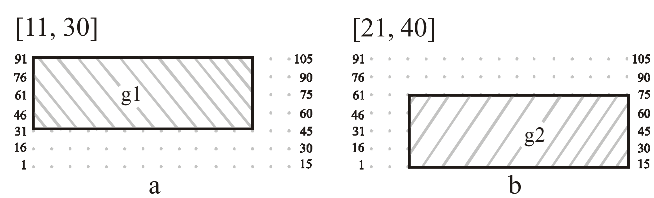


Figure 3: Geometric interpretations and life spans of the spatial objects in two distinct relations.

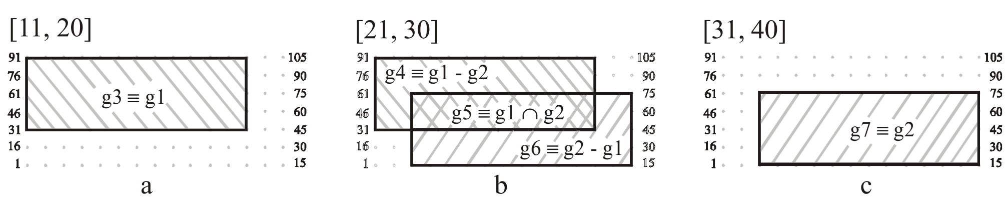


Figure 4: Geometric interpretation and life span of the spatial data obtained by the application of various spatio-temporal operations.

Operations *Quantum Union, Quantum Except* and *Quantum Intersect*: They return, respectively, the evolution of *Spatial Union*, *Spatial Except* and *Spatial Intersect* of the spatial objects, stored in two union-compatible relations, with respect to time. As an example, assume that relations R_1(T, G) and R_2(T, G) consist of the spatial objects whose geometric interpretation and life spans are shown in figure 3(a) and figure 3(b), respectively. Then, QU = R_1 *QUnion*[G, T] R_2 (QE = R_1 *QExcept*[G, T] R_2, QI = R_1 *QIntersect*[G, T] R_2) consists of the tuples {([11, 20], g3), ([21, 30], g4 ∪ g5 ∪ g6), ([31, 40], g7)} ({([11, 20], g3), ([21, 30], g4)}, {([21, 30], g5)}). The geometric interpretation of the spatial data in these relations during various periods of time is shown in figure 4.

Operation *Full Overlay*: It returns the evolution of the *Full Overlay* of the spatial objects stored in two relations with respect to time. As an example, consider R_1(A, T, G) = {(a, [11, 30], g1)} and R_2(B, T, G) = {(b, [21, 40], g2)} where the geometric interpretation of g1 and g2 is shown in figures 3(a) and 3(b), respectively. Then FO = R_1 *FOverlay*[G, T] R_2 consists of the tuples {(a, null, [11, 20], g3), (a, null, [21, 30], g4), (a, b, [21, 30], g5), (null, b, [21, 30], g6), (null, b, [31, 40], g7)}. The geometric interpretation of the spatial data of this relation during various periods of time is shown in figure 4.

Some more operations have also been defined in [3]. Three of them are the *Pair-Wise Union*, *Pair-Wise Except* and *Pair-Wise Intersect*. Three more are the *Inner, Left, and Right Overlay* operations. Note that the definition of all the Overlay operations incorporates the respective *Join* operation of SQL.

5 Conclusions

The advantages of the proposed model can be summarized as follows: All the algebraic operations are closed and, in conjunction with [3], they can be applied uniformly for the management of either spatio-temporal or spatial or temporal or conventional data. It has been identified, in particular, that certain operations, originally defined solely for the management of spatial data, are also of practical interest for the handling of temporal or conventional data. Hence, the algebra is not many-sorted and enables the uniform treatment of any of the above types of data. Regarding the case of spatial data, it has been identified that a *map* matches the geometric representation of a relation that contains such data. The model is also close to human intuition. By definition, for example, a line or a surface consists of an infinite number of 2-d points, a line is treated as a degenerate surface and a point is treated as either a degenerate line or as a degenerate surface. Due to this, it is estimated that the model is also user-friendly. Finally, it is very general. This is not only because it can be applied to the previously mentioned types of data. It can also handle relations with many attributes of some *time* data type [3] and investigation results have shown that such relations may also contain n-d spatial objects. Relevant research concerns implementation issues and the definition of an SQL extension.

References

[1] R. H. Güting, M. Schneider, *Realm-Based Spatial Data Types: The Rose Algebra*, Very Large Data Bases Journal 4, 100-143, 1995.

[2] N. A. Lorentzos, Y. G. Mitsopoulos, *SQL Extension for Interval Data*, IEEE Transactions on Knowledge and Data Engineering 9(3), 480-499, 1997.

[3] N. A. Lorentzos, J. R. Rios Viqueira, N. Tryfona, *Quantum-Based Spatial Data Model*, Internal Rept. in preparation, Informatics Laboratory, AUA, 2000.

[4] N. A. Lorentzos, N. Tryfona, J. R. Rios Viqueira, *Relational Algebra for Spatial Data Management*, Int. Workshop Integrated Spatial Databases: Digital Images and GIS, Portland, Maine, USA, 192-208, 1999.

[5] J. W. van Roessel, *An Integrated Point-Attribute Model for Four Types of Areal Gis Features*, Proc. 6th Int. Symposium on Spatial Data Handling 1 (SDH´94), T. C. Waugh, R. G. Healey (eds.), Edinburg, Scotland, UK, 127-144, 1994.

[6] M. Scholl, A. Voisard, *Thematic Map Modeling, Design and Implementation of Large Spatial Databases*, Proc. SSD′89, in Lecture Notes in Computer Science 409, Springer-Verlag, Berlin, 167-190, 1989.