Proceedings of the 25th International Conference on Computer Applications in Industry and Engineering, CAINE 2012, New Orleans, Louisiana, USA, pp.29-34, 2012.

AN EFFICIENT ALGORITHM FOR GENERATING COMPOSITION TABLES FOR QUALITATIVE SPATIAL RELATIONS

Chaman L. Sabharwal
Computer Science Department
Missouri University of S&T
Rolla, MO – 65409, USA
chaman@mst.edu

Jennifer L. Leopold
Computer Science Department
Missouri University of S&T
Rolla, MO – 65409, USA
leopoldj@mst.edu

Abstract

Qualitative Spatial Reasoning (QSR) is useful for deriving logical inferences when quantitative spatial information is not pertinent or not available. The automated derivation of such logical inferences becomes a binary constraint satisfaction problem; binary relations can be formulated as a region-based qualitative reasoning problem involving disjunctive constraints. It has been proved [1] that composition-based reasoning is complete for deriving inferences in various calculi; hence a composition table is a central part of QSR and consistency-checking. Previously developed algorithms to construct such tables have required manual or, at best, semi-automatic analysis. Herein we present a new divide-and-conquer algorithm to automatically construct a composition table for spatial relations, and apply it to the 3D region connection calculus, VRCC3D+. This validates the hypothesis that consistency-checking, entailment checking, and occlusion can be supported in 3D.

1 Introduction

Mereotopology typically is the basis for qualitative spatial reasoning when knowledge is vague in terms of quantitative specification, or when quantitative reasoning is intractable. The qualities of spatial objects that are taken into consideration for such analyses include: topology (i.e., connectedness), mereology (i.e., parthood), morphology, dimensionality, measurement, and distance.

Compositions are useful for predicting the relation between two objects when we know their relation with a third common object (i.e., given R_1(A, B) and R_2(B, C), we can narrow down the possibilities for the composition, R_3(A, C)). Compositions also are used to reduce the computational effort required to perform consistency-checking. However, the problem of creating a composition table still remains a challenge today [2]. Historically, composition tables have been generated manually for qualitative spatial reasoning [3]. If the number of relations is small, this approach is acceptable; however, manual construction is both inefficient and error-prone when there are a large number of relations. Although several algorithms have been proposed for automated construction, they can produce erroneous results [4], and can be quite complicated to implement [5]. Recently a new algorithm was proposed by [2] that employs a semi-automatic, dynamic table construction method. This method is independent of the number of relations, but still suffers from several shortcomings; the authors [2] admit that it does not guarantee 100% accuracy in the results, because it assumes entailment checking that the input dataset is a prior consistent before creating the composition table.

Herein we propose a new divide-and-conquer technique that is simpler and more accurate than the algorithms cited above. The paper is organized as follows. Section 2 includes a brief background on the mathematics of spatial relations. Section 3 discusses the basic ideas of a composition table for spatial relations, including its construction, complexity, and utility for consistency-checking. Section 4 describes our new approach for constructing the table. Concluding remarks are made in Section 5.

2 Background

2.1 Mathematical Preliminaries

R^3 denotes the three dimensional space endowed with a distance metric. Here the notions of *subset, proper subset, equal sets, empty set (∅), universal complement, union, intersection, and relative complement* are the same as those typically defined in set theory. The notions of *neighborhood*, *open set, closed set, limit point, boundary, interior, exterior,* and *closure* of sets are as in point-set topology.

A set is *connected* if it cannot be represented as the union of disjoint open sets. For any non-empty bounded set A, we use A^i, A^e, and A^b to represent the interior, exterior, and boundary of A, respectively. A is *weakly* connected to B, denoted by $\mathcal{C}(A,B)$, if $\bar{A} \cap \bar{B} \neq \emptyset$; that is, $A^i \cup A^e$ is *weakly* connected. This is different from the mathematical definition of connectedness where $A^i \cup A^e$ is considered disconnected. If A is a subset of B, it is denoted by $\mathcal{P}(A,B)$ (i.e. A is *part* of B); specifically, $A \subseteq B$ or $A \cap B^c = \emptyset$. Note that if we allow ∅ to be a region, contradictions can arise: ∅ is contained in every region, and is disconnected from it.

Mathematically, a set theoretic relation R from a set A to a set B is a subset of the direct product of A and B, e.g.. $R \subseteq A x B$. Thus a relation R on a set $\mathcal{D}$ is a subset of $\mathcal{D}^2$: $R \subseteq \mathcal{D} x \mathcal{D} = \mathcal{D}^2$. With this definition, ∅ is a relation. The set of relations depends on the type of reasoning framework in which it will be utilized. Note that $\varnothing$ is not a spatial relation; it is purely an abstract mathematical relation.

2.2 Related Work in Region Connection Calculi

Much of the foundational research on qualitative spatial reasoning is based on a region connection calculus that describes 2D regions by their possible relations to each other. Spatial relations are constraints in a logical framework. Most notable are the RCC-5 and RCC-8 models [6]. The RCC-5 relations are: *disconnected* (DC), *partial overlap* (PO), *equal* (EQ), *proper part* (PP), and *proper part converse* (PPc). RCC-8 differs from RCC-5 in that it adds a relation for *externally connected* (EC), distinguishes PP as two relations, *tangential proper part* (TPP) and *non-tangential proper part* (NTPP), and separates the converse relation PPc into TPPc and NTPPc.

Utilizing first order logic, the RCC-8 (Randell et al. 1992) framework is defined in terms of a connectedness primitive, $\mathcal{C}(A,B)$. It is independently identified in the context of Geographic Information Systems (GIS) by Egenhofer and Franzosa [7] in terms of 9-intersections that compare the intersections of the interiors, exteriors, and boundaries of two regions.

Let $\mathcal{D}$ be the domain of the relations. The Connection relation $\mathcal{C}$(A,B) for spatial reasoning is defined as:

$$(\forall A \in \mathcal{D})\ \mathcal{C}(A,A)$$

$$(\forall A,B \in \mathcal{D})\ \mathcal{C}(A,B) \Leftrightarrow \mathcal{C}(B,A)$$

The Parthood relation is defined using connectivity $\mathcal{C}$(A,B):

$$\mathcal{P}(A,B) \equiv (\forall x \in \mathcal{D})\ \{\mathcal{C}(x,A) \Rightarrow \mathcal{C}(x,B)\}$$

$$Pc(A,B) \equiv \mathcal{P}(B,A) \equiv (\forall x \in D)\ \{\mathcal{C}(x,B) \Rightarrow \mathcal{C}(x,A)\}$$

$$PP(A,B) \equiv P(A,B) \wedge \neg Pc(A,B)$$

$$PPc(A,B) \equiv Pc(A,B) \wedge \neg P(A,B)$$

Using $\mathcal{P}$ and PP, the Jointly Exhaustive Pairwise Disjoint (JEPD) RCC-8 relations are as follows:

$$PO(A,B) \equiv (\exists C)[P(C,A^i) \wedge P(C,B^i)] \wedge \neg P(A,B) \wedge \neg P(B,A)$$

$$EQ(A,B) \equiv P(A,B) \wedge Pc(A,B)$$

$$EC(A,B) \equiv C(A,B) \wedge (A^i \cap B^i == \varnothing)$$

$$DC(A,B) \equiv \neg C(A,B)$$

$$TPP(A,B) \equiv PP(A,B) \wedge \exists x[EC(x,A) \wedge EC(x,B)]$$

$$TPPc(A,B) \equiv PPc(A,B) \wedge \exists x[EC(x,A) \wedge EC(x,B)]$$

$$NTPP(A,B) \equiv PP(A,B) \wedge \nexists x[EC(x,A) \wedge EC(x,B)]$$

$$NTPPc(A,B) \equiv PPc(A,B) \wedge \nexists x[EC(x,A) \wedge EC(x,B)]$$

See Figure 1 for a graphical description of the RCC-8 JEPD relations. As previously mentioned, these axiomatic relations also can be distinguished in terms of 9-Intersections [6].

The degree of expressivity varies in region connection calculi. RCC-5 [6] distinguishes only five relations, whereas RCC-23 [8] extends RCC-8 to 23 relations in order to accommodate concave regions in 2D. RCC-62 [9] is even more expressive than RCC-23; whereas RCC-23 considers a concave region as one whole part, RCC-62 decomposes such a region into an outside, boundary, interior, and inside. The resulting 62 relations are based on a 16-Intersection that compares one object's outside, boundary, interior, and inside with those of another object. However, like RCC-8 and RCC-23, RCC-62 only describes the relationship between regions considering two dimensions.

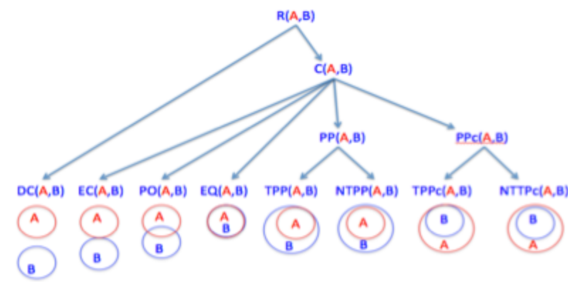

Figure 1. Hierarchical structure in terms of connectivity for RCC-8 relations

In contrast to RCC-8, RCC-23, and RCC-62, the LOS-14 [10] and ROC-20 [11] models qualify the spatial relation between 2D regions in terms of the obscuration that occurs between them. Similarly, VRCC-3D+ [12] supports a full spectrum of spatial relations including occlusion as observed by the viewer; however, unlike the aforementioned RCC models, connectivity in VRCC-3D+ is considered in 3D, not simply 2D. As with RCC-8, the VRCC-3D+ connectivity relations can be distinguished based on an 8-Intersection that compares the intersection of the interior (Int), boundary (Bnd), and/or exterior (Ext) of one object with those of another object (e.g., IntBnd is the intersection of the interior of one object and the boundary of another object); see Table 1, where each intersection is either empty (e) or non-empty (n). These 8-Intersections actually have been shown to be equivalent to 4-Intersections [13].

Table 1. 8-Intersection Characterization of VRCC-3D+ (and RCC-8) Connectivity Relations

	IntInt	IntBnd	IntExt	BndInt	BndBnd	BndExt	ExtInt	ExtBnd
DC	e	e	n	e	e	n	n	n
EC	e	e	n	e	n	n	n	n
PO	n	n	n	n	n	n	n	n
EQ	n	e	e	e	n	e	e	e
TPP	n	e	e	n	n	e	n	n
TPPc	n	n	n	e	n	n	e	e
NTPP	n	e	e	n	e	e	n	n
NTPPc	n	n	n	e	e	n	e	e

The occlusion part of a VRCC-3D+ relation (i.e., the projection of 3D in the 2D plane) is represented syntactically as xObs_y where x is n for discrete, p for partial, e for equal, or c for proper part in 2D; y accounts for a qualitative depth parameter, InFront, with value Y for near, N for farther, and E for equal. For example, cObs_c(A,B) means B is closer to the viewer than A, and the projections A_P and B_P on the view plane are such that

B_P completely (properly) includes A_P. The VRCC-3D+ relations distinguish 12 occlusion relations (nObs, nObs_c, nObs_e , pObs, pObs_c, pObs_e, eObs, eObs_c, eObs_e, cObs, cObs_c,cObs_e) with respect to the viewer position. The 3-Intersections and depth information that characterize the occlusion relations are depicted in Table 2.

Table 2. Intersection and Distance Parameters for Occlusion Relations in VRCC-3D+

	IntInt	IntExt	ExtInt	InFront
nObs	e	n	n	Y
nObs_c	e	n	n	N
nObs_e	e	n	n	E
pObs	n	e	n	Y
	n	n	n	Y
pObs_c	n	n	e	N
	n	n	n	N
pObs_e	n	n	n	E
eObs	n	e	e	Y
eObs_c	n	e	e	N
eObs_e	n	e	e	E
cObs	n	n	e	Y
cObs_c	n	e	n	N
cObs_e	n	n	e	E
	n	e	n	E

When occlusion relations are combined with RCC-8 relations (computed in 3D), the resulting VRCC-3D+ model yields 37 valid relations. Only certain types of occlusion are possible for each RCC-8 relation, as indicated by entries of * in Table 3; if R represents the RCC-8 relation in the left hand column and xObs_y represents the obscuration relation in the top row, then the entries in the table are interpreted as: R_xObs_y is a valid VRCC-3D+ relation (e.g., DC_nObs, TPPc_cObs_e, etc.).

Table 3. Valid VRCC-3D+ Relations

	nObs	nObs_c	nObs_e	pObs	pObs_c	pObs_e	eObs	eObs_c	eObs_e	cObs	cObs_c	cObs_e
DC	*	*	*	*	*		*	*		*	*	
EC	*	*	*	*	*		*	*		*	*	
PO				*	*	*	*	*		*	*	*
EQ									*			
TPP								*	*		*	*
TPPc							*		*	*		
NTPP											*	*
NTPPc										*		

3 Composition of Spatial Relations

Here we briefly discuss the composition-based reasoning technique for spatial reasoning. For sets A, B, C, let R_1 be a relation from A to B, and R_2 be a relation from B to C. The composition of relations R_1 and R_2 is a relation which is a subset of AxC. It is denoted by $R_1 o R_2$, $R_1 o R_2 \subseteq AxC$, and is defined by $R_1 o R_2 = \{(a,c) \in AxC: \exists\, b \in B \ni (a,b) \in R_1 \wedge (b,c) \in R_2\}$. This guarantees the existence of $b \in B$, which is a necessary and sufficient condition for the composition.

In spatial reasoning, the instantiation of b is sufficient, but not necessary, in the definition of composition. The notion of weak composition is defined as follows. Let $\mathcal{R} = \{R_1, R_2, \ldots, R_n\}$ be a finite set of Jointly Exhaustive and Pairwise Disjoint (JEPD) binary relations on a domain $\mathcal{D}$. The *weak composition* is defined as o_w: $\mathcal{R}$ x $\mathcal{R} \rightarrow 2^{\mathcal{R}}$ meaning for given A, B, C $\in \mathcal{D}$, with $R_i(A,B) \wedge R_j(B,C)$, $\exists\, k \in \{1,2,\ldots,n\}$ $R_i o_w R_j(A,C) = R_k(A,C)$ so that $R_i(A,B) \wedge R_j(B,C) \wedge R_k(A,C)$ is true. The problem then is to determine whether there is an instantiation A, B, C $\ni R_i(A,B) \wedge R_j(B,C) \wedge R_k(A,C)$.

Therefore, weak composition is not a single base relation, but a general disjunctive relation, (i.e., a disjunction of a set of relations). This is a disjunctive relation whose state is determined on an instantiation of a base relation in $\mathcal{R}$. That is, for specific A, B, C and a pair of relations R_i and R_j in $\mathcal{R}$, for which $R_i(A,B)$, $R_j(B,C)$ hold, then $R_i o_w R_j(A,C) \in \{R_1(A,C), R_2(A,C), \ldots, R_n(A,C)\}$. Hence, $R_i o_w R_j$ is the disjunction, $R_i o_w R_j = R_1 \vee R_2 \vee \ldots \vee R_n$. More precisely, $R_i o_w R_j$ is the disjunction of the *smallest subset* of $\{R_1, R_2, \ldots, R_n\}$ for which the identity $R_i o_w R_j = R_1 \vee R_2 \vee \ldots \vee R_n$ holds. Mathematically speaking, the weak composition can be expressed more precisely as $R_i o_w R_j$ is the disjunction of base relations in the set $\{R: R \in \mathcal{R}, R \cap (R_i o R_j) \neq \emptyset\}$.

3.1 Composition Table

Often a significant amount of knowledge can be deduced from a small set of binary relations. The inferences can be obtained by using composition-based reasoning. If there are n relations, the composition table will have n^2 entries. Each table entry is represented by more general relations, specifically the disjunction of base relations. Each of these entries is the disjunction of n relations in the worst case, requiring n consistency checks. Thus, the table construction is of order $O(n^3)$.

3.2 Composition Table: Construction

Historically, composition has been used for automated reasoning in a variety of applications. Allen [14] used it for temporal reasoning; [6, 7] used it for efficient spatial reasoning. For a large set of relations of size n, it is difficult to create such a table manually (which could require consideration of 2^n relations).

In general, for composition $R_i o_w R_j$, we would check $R_i(A,B) \wedge R_j(B,C) \wedge R_k(A,C)$ for n values of k for consistency. In practice, for composition $NTPPo_wNTTP$ in RCC-8, only one consistency check is required instead of eight; for composition $TPPo_wTPP$, only two consistency checks are necessary instead of eight. There is a large amount of duplicated effort in the $O(n^3)$ construction of the composition table. The RCC-8 composition shown in Table 4 can be referenced from [3]. In this table the left column relations are for object pair (A,B), the top row corresponds to object pair (B,C), and the table entries correspond to object pair (A,C).

Table 4. Composition Table of RCC-8 Relations

		R(B,C)							
	o	**DC**	**EC**	**PO**	**TPP**	**NTPP**	**TPPc**	**NTPPc**	**EQ**
R(A,B	**DC**	*	DR,PO,PP	DR,PO,PP	DR,PO,PP	DR,PO,PP	DC	DC	DC
	EC	DR,PO,PPc	DR,PO,TPP,TPPc,EQ	DR,PO,PP	EC,PO,PP	PO,PP	DR	DC	EC
	PO	DR,PO,PPc	DR,PO,PPc	*	PO,PP	PO,PP	DR,PO,PPc	DR,PO,PPc	PO
	TPP	DC	DR	DR,PO,PP	PP	NTPP	DR,PO,PPc,EQ	DR,PO,PPc	TPP
	NTPP	DC	DC	DR,PO,PP	NTPP	NTPP	DR,PO,PP	*	NTPP
	TPPc	DR,PO,PPc	EC,PO,PPc	PO,PPc	PO,TPP,TPPc,EQ	PO,PP	PPc	NTPPc	TPPc
	NTPPc	DR,PO,PPc	PO,PPc	PO,PPc	PO,PPc	PO,PP,PPc,EQ	NTPPc	NTPPc	NTPPc
	EQ	DC	EC	PO	TPP	NTPP	TPPc	NTPPc	EQ

There are several methods for creating composition tables. One way is to manually create a static table [3]. A recently presented method based on machine learning techniques [2] is semi-automatic, but imposes some very severe restrictions on the construction of the table. Unfortunately, the results are not guaranteed to be 100% accurate, and the method assumes that the input dataset is a prior consistent to be able to construct the composition table. The primary advantage of the method is that the dynamic table is local to the problem context, and consequently more efficient to use. A static table can be too complex if it must cover all cases for completeness, and its use may impede the overall efficiency of the reasoner application.

3.3 Composition Table: Complexity

Each composition relation is associated with three objects, say, A, B, C, which can be represented as the vertices of a triangle, and the relations can be associated with the edges of the triangle (i.e., $R_1(A, B)$, $R_2(B, C)$, and $R_3(A, C)$). The complexity of computing the composition table amounts to counting the number of triangles for all such compositions. Since a relation may be symmetric or asymmetric, the edges will be undirected (bidirectional) or directed. If a triangle is rotated or inverted, an equivalent configuration results. Let (A, B, C) be a triangle with edge direction represented by arrows along the edges AB, BC, CA. We use u (undirected) for symmetric, and v, w for asymmetric (undirected) edges denoted by v, w. The possible triangles may have (0) no directed edges, (u,u,u,) (1) one directed edge, (w,u,u), (2) two directed edges, (u,v,w), (u,w,v),(u,v,v), or (3) three directed edges. (v,v,v,),(w,v,v), as depicted in Figure 3.

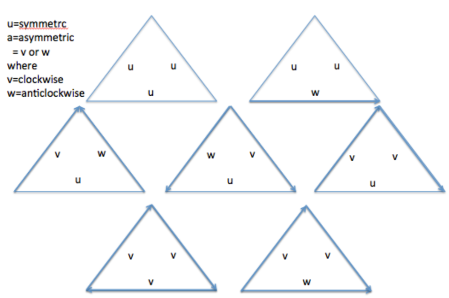

Figure 3. Combinations of Composition Relations

If there are s = #u symmetric, a=#v + #w asymmetric relations, then n = s+2a. If T is the total number of triangles, after calculating the possible instantiations, we have [1]

$$T = (1/6)(s^3+3s^2+2s)+s^2a+s(2a^2+a)+(1/3)(4a^3+2a)$$

Since s= n-2a

$$T = (1/6)(n^3+3n^2+2n) - n\,a$$

Therefore, the number of distinct triangles reflects about one sixth of the brute force approach total n^3. Utilizing these triangle criteria in the composition table construction reduces the computation effort considerably, as shown in Table 5. The percentage of computational effort by this way is calculated as $((1/6)(n^3+3n^2+2n) - n\,a)/n^3 *100$.

Table 5. The last column of the table indicates that the cost of calculations is less than 25%.

System	s	a	n^3	T	%
RCC-5	3	1	125	30	24
RCC-7	3	2	343	70	20
RCC-8	4	2	512	104	20
IC-13	1	6	2197	377	17
LOS-14	2	6	2744	476	17
VRCC-3D	5	6	4913	867	18
ROC-20	6	7	8000	1400	18
RCC-23	7	8	12167	2116	17
VRCC-3D+	9	14	50653	8621	17
RCC-62	10	26	238328	40052	17

The table shows that the computation effort is at most 25%, a saving as at least 75% of computation over the brute force methods.

4 Divide-and-Conquer Algorithm for Composition Table Construction

In VRCC-3D+ there are 37 relations. Clearly, it is not practical to create the VRCC-3D+ composition table manually (i.e., each entry is a disjunction of at most 37 relations). This also would be problematic for other mathematical models used for QSR (e.g., RCC-62, which contains 62 relations). Here we present an automated construction approach that utilizes a divide-and-conquer strategy. We first factor out the common attribute in the relations, then create smaller composition tables, eventually integrating them into one overall table.

We start with a simple example to illustrate this. The RCC-8 composition table is an 8x8 table, where each entry is the disjunction of at most eight relations. Alternately, it can be thought of as a (smaller) 4x4 table, where each entry is a 2x2 table whose entries are the disjunction of at most 4 relations. This requires decomposing the relations into sub-categories (or equivalence classes) before creating a full scale composition table. We also may take advantage of the intrinsic properties of the relations to reduce the size of the table. For example, it may be observed that composition of EQ with any relation is the relation itself (e.g., $X o_w EQ = EQ o_w X = X$ for any relation X). This reduces the computation of an 8x8 table to a 7x7 table. Also, once $PP o_w PP$ is computed, we do not need to compute the $PPc o_w PPc$ table; since composition is not commutative, $PP o_w PPC$ is not the same as $PPc o_w PP$.

These entries need not be computed; its entries are simply the converses of the entries in the table for PPo_wPP.

If we let DR = {DC, EC}, PP={TPP, NTPP}, PPc = {TPPc, NTPPc}, O = {PO, EQ}, we get four relation sub-categories, {DR, O, PP, PPc}. We can easily create a 4x4 table whose entries are 2x2 sub-tables. If we take advantage of the intrinsic property of EQ, O can be replaced with simply singleton {PO}. After partitioning, it becomes {DR, PO, PP, PPc}. This will yield a 4x4 table whose entries correspond to sub-tables of size 2x2, 2x1, 1x2, and 1x1. This smaller composition table is shown in Table 6.

Table 6. Reduced Composition Table for RCC-8

o	DR	PO	PP	PPc	EQ
DR	*	DR,PO,PP	DR,PO,PP	DR	DR
PO	DR,PO,PPc	*	PO,PP	DR,PO,PPc	PO
PP	DR	DR,PO,PP	PP	*	PP
PPc	DR,PO,PPc	PO,PPc	PO,PP,PPc,EQ	PPc	PPc
EQ	DR	PO	PP	PPc	EQ

We illustrate how this "reduced" composition table can be used to construct the full table with two examples. First, consider the composition PPo_wPP=PP where PP represents {TPP, NTPP}. PPo_wPP can be expanded into a 2x2 composition sub-table by creating the composition of four pairs whose entries come from {TPP, NTPP}. Table 7 shows that the PPo_wPP is a representative of the 2x2 table with entries from {TPP, NTPP}.

Table 7. Composition Sub-Table for PPo_wPP

		PP	
		TPP	NTPP
PP	TPP	PP	NTPP
	NTPP	NTPP	NTPP

As a second example, consider DRo_wPPc= DR, where DR represents the set {DC, EC} and PPc is the set {TPPc, NTPPc}. DRo_wPPc can be expanded into a 2x2 composition sub-table by creating the compositions of four pairs whose entries come from {DC, EC}. The composition of four pairs can be easily computed manually and integrated with DRo_wPPc. Table 8 shows that the DCo_wPPc is in fact a 2x2 table with entries from{DC, EC}

Table 8. Composition Sub-Table for DRo_wPPc

		PPc	
		TPPc	NTPPc
DR	DC	DC	DC
	EC	DR	DC

Similarly other composition sub-tables for RCC-8 can be computed and integrated to form the full 8x8 table. The utility of this strategy lies in handling cases where there are many relations, resulting in a large table. The ingenuity lies in partitioning the large set of relations into smaller subsets, and grouping them.

Now we apply this strategy to VRCC-3D+, which defines 37 relations. As was discussed in Section 2, each VRCC-3D+ relation consists of one of 8 connectivity relations and one of 12 obscuration relations. We separately create the composition tables of 8 relations (previously shown in Table 4) and 12 relations (not shown). The two tables are then integrated to form the composite table covering all 37 relations.

The integration of the composition tables can be performed using the following algorithm:

```
Compute composition table CT_RCC8;
Compute composition table CT_Obs12;
For each pair R1_Obs1 and R2_Obs2 do:
  For each R in CT_RCC8[R1, R2] do:
    Let ObsR be the set of obscuration relations
      applicable to R (see Table 3);
    For each x_Obs_y in
      CT_Obs12[Obs1, Obs2] ∩ ObsR do:
        Add  R_xObs_y to CT[R1_Obs1, R2_Obs2];
    EndFor
  EndFor
EndFor.
```

Due to space limitations, it not possible to display the VRCC-3D+ composition table in its entirety. Tables 9, 10, and 11 show one entry from the 8x8 composition table of VRCC-3D+ connectivity relations, the 12x12 composition table of VRCC-3D+ obscuration relations, and the 37x37 full composition table for VRCC-3D+, respectively.

Table 9. One Entry from Composition Table for Connectivity Relations

	DC
EC	DC,EC,PO,TPPc,NTPPc

Table 10. One Entry from Composition Table for VRCC-3D+ Occlusion Relations

	nObs
pObs	nObs, pObs, cObs

Table 11. One Entry from Complete VRCC-3D+ Composition Table

	DC_nObs
EC_pObs	DC_cObs, DC_nObs, DC_pObs, EC_cObs, EC_nObs, EC_pObs, NTPPc_cObs, PO_cObs, PO_pObs, TPPc_cObs

4.1 Composition Table and Consistency Checking

It is obvious in Temporal Interval Algebra that for any three times t_1, t_2, and t_3, if $t_1<t_2$, $t_2<t_3$ then the relation $<$ (e.g. $t_3< t_1$) is not consistent because "<" is irreflexive.

In our spatial domain, for objects A, B, and C, if PP(A,B), PP(B,C), then the relation PP (i.e., PP(C,A)) is inconsistent because PP is irreflexive. In general, the more information there is about the objects being observed, the greater is the chance of inconsistency. If we are uncertain about a relation from $\mathcal{R}$ between A and B and if know the relation of A and B to a third object C, we can reduce uncertainty by using the composition table. Formally, if R_{ij} is a possible set of relations between A_i and A_j, it can be strengthened by replacing

$$R_{ij} \text{ with } R_{ij} \cap (R_{ik} o_w R_{kj})$$

for all object instantiations A_k. If the final R_{ij} is empty, the relation is *inconsistent*; otherwise, it is consistent. If the relation between two objects is considered as a constraint, then the logical constraint satisfaction problem is formulated as consistency checking, or *path consistency.*

The composition table is useful for checking the correctness of all logical inferences. Consistency is closely related to uncertainty. For some applications it is useful to pre-determine the consistency of data facts, *entailment checking*; in others it is desired to check the correctness of logical inferences, relation edge consistency or *path consistency*. Hence a composition table can be used as a lookup table for consistency checking.

5 Conclusion

Herein we have presented a new algorithm to construct a composition table for qualitative spatial reasoning. We have implemented this new algorithm in Prolog to construct a composition table for VRCC-3D+, a 3D enhancement of the well-known RCC-8 2D calculus. This work will further support our research in QSR consistency-checking, entailment checking, and occlusion detection, and hopefully will prove to be beneficial for others working in the field of automated reasoning.

6 References

[1] Brandon Bennet, *Some Observations and Puzzles about Composing Spatial and Temporal Relations* 11th European Conference on Artificial Intelligence, ECAI, pp.1-8, 1994.

[2] Weiming Liu and Sanjiang Li *On A Semi-Automatic Method for Generating Composition Tables* IJCAI-2011 Workshop 27, pp.51-58, 2011.

[3] M. J. Egenhofer, *Deriving the Composition of Binary Topological Relations* Journal of Visual Languages and Computing, 5(2), 133-149, 1994.

[4] R. Moratz, *Representing relative direction as a binary relation of oriented points,* ECAI2006, pp.407-411, IOS Press, 2006.

[5] T. Mossakowski and R. Moratz, *Qualitative reasoning about relative direction on adjustable levels of granularity,*Corr abs/1011.0098, 2010.

[6] D.A. Randell, Z. Cui, A.G. Cohn, *A Spatial Logic Based on Regions and Connection* KR92:165–176, 1992.

[7] M. J. Egenhofer, R. Franzosa *Point-Set topological Relations,* International Journal of Geographical Information Systems 5(2), pp.161-174, 1991.

[8] A.G. Cohn, B. Bennet, J. Dooday, and N.M. Gotts, *Qualitative Spatial Representation and Reasoning with the Region Connection Calculus* GeoInformatica 1(1), pp 1-44, 1997.

[9] J.O. Yang, Q. Fu, and D. Liu, *A Model for Representing Topological Relations Between Simple Concave Regions. ICCS 2007, Part I, LNCS 4487*, Y. Shi et al., Editors, pp. 160-167, 2007.

[10] Sanjiang J. Li, Minsheng.S. Ying, *Region Connection Calculus: Its models and composition table* Artificial Intelligence, 145, No. 1-2, April, pp. 121-146, 2003.

[11] D.A. Randell, M. Witkowski, M. Shannahan, *From Images to Bodies: Modeling and Exploiting Spatial Occlusion and Motion Parallax* Proceedings of the 17th International Joint Conference on Artificial Intelligence, vol 1. (San Francisco, California), IJCAI 2001, pp. 57-66.

[12] C. Sabharwal, J. Leopold, and N. Eloe *A More Expressive 3D Region Connection Calculus* In The 2011 International Workshop on Visual Languages and Computing (in conjunction with the 17th International Conference on Distributed Multimedia Systems (DMS'11)), pp. 307-311. KSI, Aug. 2011.

[13] C. Sabharwal and J. Leopold. *Reducing 9-Intersection to 4-Intersection for Identifying Relations in Region Connection Calculus.* The 24th International Conference on Computer Applications in Industry and Engineering (CAINE 2011), (Nov. 2011): pp. 118–123.

[14] J. Allen, *Maintaining Knowledge about Temporal Intervals, Communications of the ACM 26(11); 361-372, 1983.*