Temporalizing Spatial Calculi: On Generalized Neighborhood Graphs

Marco Ragni and Stefan Wölfl

Institut für Informatik, Albert-Ludwigs-Universität Freiburg,
Georges-Köhler-Allee, 79110 Freiburg, Germany
{ragni, woelfl}@informatik.uni-freiburg.de

Abstract. To reason about geographical objects, it is not only necessary to have more or less complete information about where these objects are located in space, but also how they can change their position, shape, and size over time. In this paper we investigate how calculi discussed in the field of qualitative spatial reasoning (QSR) can be temporalized in order to gain reasoning formalisms that can be used to express spatial configurations and their dynamics. In a first step, we briefly discuss temporalized spatial constraint languages. In particular, we investigate how the notion of continuous change can be expressed in such languages and how continuous change is represented in the so-called conceptual neighborhood graph of the spatial calculus at hand. In a second step, we focus on a special reasoning problem, which occurs quite naturally in the context of temporalized spatial calculi: Given an initial spatial scenario of some physical objects, which scenarios are accessible if the set of all possible paths of these objects is constrained by some further conditions? We show that for many spatial calculi this general problem cannot be dealt with by using the information encoded in the classical neighborhood graphs, as usually discussed in the literature. Rather, we introduce a generalized concept of neighborhood graph, which allows for reasoning about objects in such dynamic settings.

1 Introduction

To reason about geographical objects, it is not only important to have information about where these objects are located in space, but also how they can change their position, shape, and size over time. Some physical objects such as chairs, towers, and stars are usually assumed to be rather robust to changes in shape and size (at least from the point of what we can experience without using scientific instruments). Other objects such as hurricanes, clouds, and balloons may vary their size and shape quite rapidly. Obviously, how physical objects can change such spatial properties depends on the physical quality structure of the respective object and its environment. A crucial notion in this context is the notion of continuous change since it seems common-sense that many property changes occur continuously. Topic of the paper will be to discuss how continuity concepts can be integrated into the formal calculi discussed in the qualitative spatial reasoning literature.

Under the heading of *qualitative spatial reasoning*, many formalisms for representing, and reasoning with, spatial configurations have been discussed in the past two

U. Furbach (Ed.): KI 2005, LNAI 3698, pp. 64–78, 2005.

decades. In recent years also the issue of how to temporalize such spatial formalisms has gained more attention in the literature. Obviously, temporalizations of spatial calculi can be developed by exploiting different research strategies. First, they can be embedded into rich first-order theories by integrating mereotopological and temporal concepts. For example, Muller [16] has proposed a first-order theory of spatio-temporal entities, which is based on the first-order theory of the *region connection calculus* [19]. Second, temporalizations may be discussed in the framework of temporal logics. The combination of RCC8 and linear time temporal logic, for example, has been investigated by Wolter and Zakharyaschev [21] and Gabelaia et al. [10]. Third, spatio-temporal representation languages can be obtained via temporalizing a spatial constraint language (e. g., RCC8) by a suitable temporal constraint language such as Allen's interval calculus. Bennett et al. [3] proposed such a reasoning formalism, which was further investigated by Gerevini and Nebel [13]. From a more philosophical perspective, Galton [11] discussed various facets of continuous change, in particular, how such changes can be consistently described at different levels of granularity and how their qualitative and quantitative descriptions are related to each other.

In this paper we focus on the third research strategy outlined previously. In more detail, we discuss temporal constraint languages, which are enriched by formulae expressing time-dependent spatial constraints. In these languages it is possible to express temporally annotated spatial information as well as their temporal relationships. Then two kinds of reasoning tasks may be distinguished: The *static reasoning problem* is to determine whether such a spatio-temporal description is consistent, i. e., whether there exists a temporal model satisfying each temporal constraint as well as each temporally annotated spatial constraint. The *dynamic reasoning problem* is to determine whether, given a set of *transformation constraints*, there exists a continuous transformation between two spatial configurations such that none of the transformation constraints is violated. For example, the well-known *Towers of Hanoi* puzzle may be cast as such a problem.

A central notion in the context of temporalized spatial constraint networks is that of a *neighborhood graph*. For many qualitative spatial reasoning formalisms researchers have intensively discussed the so-called conceptual neighborhood graph, a concept introduced first by Freksa [9]. The neighborhood graph is usually understood as to describe which relation transitions be possible if the objects are subject to minimal changes. This interpretation of neighborhood graphs is clearly *temporal*, that means, it aims at describing the dynamics of the relations at hand.[1] Interestingly, the neighborhood graph often is not uniquely determined by the underlying background theory of the respective calculus [e. g., 18]. For instance, different neighborhood graphs can be found depending on whether objects are allowed to change their size or whether one allows two objects to be changed at the same moment. This means that, in principle, neighborhood graphs are a suitable means for encoding spatial information about the kind of objects that are described by the qualitative spatial calculus at hand. But in our opinion, the traditional concept of neighborhood graph is too restricted to be really useful for reasoning with spatio-temporal constraints. For this reason we present a gen-

[1] In the literature there is a further research stream, in which neighborhood graphs are discussed in the context of *conceptual* neighborhoodness of relations Knauff [14].

eralized concept of neighborhood graph, which may be considered a first step towards developing a more general theory on dynamic reasoning problems.

The paper is organized as follows: In section 2 we briefly introduce the calculi that will be of interest in the following sections. In section 3 we discuss spatio-temporalized constraint languages and their models. Moreover, we present a precise notion of continuous change that enables us to analytically prove the correctness of neighborhood graphs. In section 4 we explain how generalized neighborhood graphs can be applied in order to solve dynamic reasoning problems. In more detail, we present the generalized neighborhood graphs for the point algebra and for RCC5. Finally, section 5 gives a short summary of our results and a brief outlook on interesting future work.

2 Preliminaries

Let us start by briefly sketching the qualitative calculi that will be of interest in the following sections. Readers familiar with these calculi may wish to skip this section.

Constraint Satisfaction Problems. Qualitative reasoning problems are usually cast as constraint satisfaction problems (CSP), i. e., as a problem to determine whether a constraint network (a finite set of constraints) is satisfiable or entailed by another constraint network. Typically, a qualitative constraint network is a finite set of constraints of the form xRy where x and y are variables taking values in a given domain D, and a binary relation R defined on D. For modelling imprecise knowledge, one usually considers sets of relations that are closed with respect to unions. In more detail, given a specific level of granularity chosen describing the domain at hand, one starts by identifing a (finite) set jointly exhaustive and pairwise disjoint sets of *base relations* on the domain. A *composition table* gives information about which constraints xRy are possible if one has complete knowledge about how x and y are related (via base relations) to a third object z. Speaking more algebraically, from a set of base relations (containing the identity relation) and a composition table (satisfying some requirements), one can build up a *relation algebra*, i. e., a set of relations that contains the identy relation and is closed with respect to unions, intersections, converse formation, and composition of relations.

To put these notions in a more precise context, we introduce the following terminology: A qualitative constraint satisfaction problem is defined by a constraint language $\mathscr{L}$ and a class of *(intended) models*. The constraint language usually consists of an infinite set of variables and a finite set of (binary) base relation symbols. A *constraint* is a formula of the form $x\{R_1,\dots,R_n\}y$ (meaning $xR_1y \vee \dots \vee xR_ny$), where x and y are variables and each R_i is a base relation symbol. Finite sets of constraints are referred to as *constraint networks*. A *model* is a first- or higher-order structure $\mathcal{M} = \langle \dots, D, \dots \rangle$ assigning an interpretation $R^{\mathcal{M}} \subseteq D^2$ to each base relation symbol R. A *(variable) assignment in* $\mathcal{M}$ is a function a that assigns an element $a(x) \in D$ to each variable x. Given an assignment a and an element $d \in D$, the function $a(x/d)$ is defined as the function that coincided with a in all variables distinct from x and assigns object d to variable x. A constraint $x\{R_1,\dots,R_n\}y$ is said to be *satisfied* in $\mathcal{M}$ by a (denoted by $\mathcal{M}, a \models x\{R_1,\dots,R_n\}y$) if $(a(x), a(y)) \in R_i^{\mathcal{M}}$ for some $1 \le i \le n$. A *constraint network* C is said to be *satisfiable* in a class of models if there exists a model $\mathcal{M}$ in this class as well as an assignment a in $\mathcal{M}$ such that all constraints in C are satisfied. Furthermore, *a composition table* is a map assigning to each pair of base relations R_i and R_j a set of base relations $R_i \circ R_j = \{R_{k_1}, \dots, R_{k_m}\}$. A composition table is said to be *extensionally correct* for $\mathcal{M}$ if for each pair of base relations R_i and R_j and each assignment a,

$$\mathcal{M}, a \models x\,(R_i \circ R_j)\,y \iff \exists d \in D \text{ s. t. } \mathcal{M}, a(z/d) \models x R_i z \text{ and } \mathcal{M}, a(z/d) \models z R_j y.$$

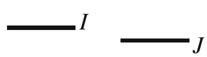
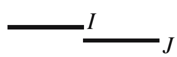
If the base relations defined by $\mathcal{M}$ are (a) jointly exhaustive (i. e., $\bigcup_{1 \le i \le n} R_i^{\mathcal{M}} = D^2$), (b) pairwise disjoint (i. e., $R_i^{\mathcal{M}} \cap R_j^{\mathcal{M}} = \emptyset$ for $i \neq j$), (c) closed with respect to converses (i. e., $R_i^{\mathcal{M}} = (R_j^{\mathcal{M}})^{\smile} := \{ (x,y) : (y,x) \in R_j^{\mathcal{M}} \}$), and (d) have an extensionally correct composition table, then there exists a uniquely determined algebra of binary relations on the domain set D.

Point Algebra. The point algebra (for linear time) may be considered the most simple qualitative calculus. The point algebra (PA) describes the relations between instants of linear flows of time. Hence, this algebra considers the three base relations $<$ ("earlier"), $=$ ("equal"), and $>$ ("later"), as well as unions of them. Point algebras can also be defined for much weaker relational structures such as branching flows of time, partial orders, etc.

Interval Algebra. Given a linear flow of time, an *Allen interval* is a pair of instants $\langle t_1, t_2 \rangle$ with $t_1 < t_2$. By comparing the relative positions of start and endpoints of two intervals, one can identify thirteen jointly exhaustive and pairwise disjoint base relations between intervals, which are known in the literature as the *Allen 13 relations* (cf. Table 1).

RCC5 and RCC8. The most prominent calculi in the domain of spatial qualitative reasoning are the region connection calculi RCC5 and RCC8. These calculi allow for

Table 1. The 13 base relations of Allen's interval algebra

Relation	Converse	Pictorial Representation
I b J	J bi I	
I m J	J mi I	
I o J	J oi I	
I d J	J di I	
I s J	J si I	
I f J	J fi I	
I e J	J e I	

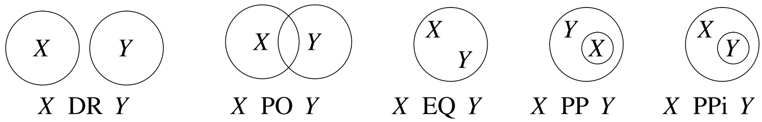


Fig. 1. The RCC5 relations

expressing relations between *regions*, which often are represented as non-void, connected, and regular closed (or regular open) subsets of some topological space. The set of RCC5 base relations consists of the relations DR ("discrete"), PO ("partially overlap"), PP ("proper part"), PPi ("proper part inverse"), and EQ ("equal") (cf. Fig. 1). RCC8 refines these relations by splitting DR into the relations DC ("disconnected") and EC ("externally connected") and by splitting PP (analogously PPi) into the relations TPP ("tangential proper part") and NTPP ("non-tangential proper part"). From the semantical point of view, a *topological model* is a tuple $\mathcal{M} = \langle S, O, \mathrm{Reg}\rangle$, where $\langle S, O\rangle$ is a topological space (O being its set of open sets) and Reg is a non-void set of regular closed subsets of S. Topological models induce RCC5 and RCC8 models in a natural manner: For example, for regions X and Y, the pair (X,Y) is in $\mathrm{DR}^{\mathcal{M}}$ if and only if $X \cap Y = \emptyset$ and $(X,Y) \in \mathrm{NTPP}^{\mathcal{M}}$ if and only if there exists an $U \in O$ such that $X \subsetneq U \subseteq Y$.

The composition table of RCC5 (cf. Table 2) is known to be correct if the relations are interpreted on closed discs in the Euclidean plane. In this case the RCC5 relations coincide with the relations definable in terms of the subset relation. For this reason, RCC5 is sometimes also referred to as *containment algebra*.

Table 2. The composition table of RCC5

	EQ	DR	PO	PP	PPi
EQ	EQ	DR	PO	PP	PPi
DR	DR	EQ, DR, PO, PP, PPi	DR, PO, PP	DR, PO, PP	DR
PO	PO	DR, PO, PPi	EQ, DR, PO, PP, PPi	PO, PP	DR, PO, PPi
PP	PP	DR	DR, PO, PP	PP	EQ, DR, PO, PP, PPi
PPi	PPi	DR, PO, PPi	PO, PPi	EQ, PO, PP, PPi	PPi

3 Temporalizing Spatial Calculi

The general method for temporalizing the language of a given spatial calculus is the following: Let $\langle V_{\mathrm{T}}, \mathcal{R}_{\mathrm{T}}\rangle$ be the language of a temporal calculus $\mathfrak{T}$ and $\langle V_S, \mathcal{R}_S\rangle$ be the language of a spatial calculus $\mathfrak{S}$, that is, V_{T} and V_S are disjoint sets of variables and $\mathcal{R}_S$

and $\mathcal{R}_{\mathrm{T}}$ are the sets of base relation symbols of the respective calculi. In general, the temporal calculus will be the point algebra or the interval algebra for linear time (but, of course, the method is not restricted to these calculi). In what follows, constraints of the temporal calculus, i. e., formulae of the form

$$i\{R_1,\ldots,R_n\}j \qquad (i,j \in V_{\mathrm{T}},\ R_1,\ldots,R_n \in \mathcal{R}_{\mathrm{T}})$$

are referred to as *temporal constraints*. We now enrich the language of $\mathfrak{T}$ by *temporalized spatial constraints*, namely formulae of the form:

$$i : x\{S_1,\ldots,S_n\}y \qquad (i \in V_{\mathrm{T}},\ x,y \in V_{\mathrm{S}},\ S_1,\ldots,S_n \in \mathcal{R}_{\mathrm{S}}).$$

In the sequel, the combined calculus will be referred to as $\mathfrak{T}:\mathfrak{S}$.

How can we define models for this language in terms of respective models for the temporal and spatial languages? To illustrate this, let us start by defining the models of a spatial calculus chosen from the RCC-family (denoted by RCCx), which is temporalized by PA. The key concept for defining such models is that of a temporalized topological model (note that the concept used here presents a modified version of the concept introduced by Wolter and Zakharyaschev [21]):

Definition 1. *A* temporalized topological model *(abbr. by* tt-model*) is a tuple* $\mathcal{M} = \langle T,<,S,O,\mathrm{Reg},\Pi\rangle$*, where* $\langle T,<\rangle$ *is a linear flow of time,* $\langle S,O\rangle$ *is a topological space,* Reg *is a set of* regions *(i. e., non-void, connected, and regular closed subsets of S), and* Π *is a non-void set of* (object) paths $\pi\colon T \longrightarrow \mathrm{Reg}$.

The idea on which the definition is based is the following: We assume that at each instant, an object occupies a specific region in a fixed topological space. Since we are only interested in the path of an object, i. e., in the sequence of regions occupied by the object, it is reasonable to represent objects as functions assigning regions to instants.

Obviously, each temporalized topological model $\mathcal{M}$ induces a (temporal) model for PA (denoted by $\mathcal{M}_{\mathrm{T}}$) and a (spatial) model for RCCx (denoted by $\mathcal{M}_{\mathrm{S}}$). A *PA:RCCx (variable) assignment* in a tt-model is a pair $a = \langle a_{\mathrm{T}},a_{\mathrm{S}}\rangle$, where $a_{\mathrm{T}}\colon V_{\mathrm{T}} \longrightarrow T$ is a function assigning instants to temporal variables and $a_{\mathrm{S}}\colon V_{\mathrm{S}} \times T \longrightarrow \mathrm{Reg}$ is a function assigning a region to each spatial variable at each instant t such that $a_{\mathrm{S,x}}(t) := a_{\mathrm{S}}(t,x)$ defines an object path of $\mathcal{M}$. Note that for each instant t, a_{S} also defines an RCC5 assignment by $a_{\mathrm{S,t}}(x) := a_{\mathrm{S}}(t,x)$. The model relation is then introduced as follows: For temporal PA:RCCx constraints we set

$$\mathcal{M},a \models i\{R_1,\ldots,R_n\}j \iff \mathcal{M}_{\mathrm{T}},a_{\mathrm{T}} \models i\{R_1,\ldots,R_n\}j,$$

and for temporalized spatial constraints we define

$$\mathcal{M},a \models i : x\{S_1,\ldots,S_n\}y \iff \mathcal{M}_{\mathrm{S}},a_{S,t} \models x\{S_1,\ldots,S_n\}y,$$

where $t = a_{\mathrm{T}}(i)$.

In the case that a spatial calculus is temporalized with respect to the interval algebra, we need to modify this semantics as follows: An *IA : RCC5 assignment* in a tt-model is a pair $a = \langle a_{\mathrm{T}},a_{\mathrm{S}}\rangle$, where a_{T} assigns an ordered pair $(a_{\mathrm{T}}(i)^-,a_{\mathrm{T}}(i)^+) \in T^2$ with

$a_{\mathrm{T}}(i)^- < a_{\mathrm{T}}(i)^+$ to each interval variable and a_{S} is defined as above. Here the model relation is defined as follows:

$$\mathcal{M}, a \models i : x\{S_1, \ldots, S_n\}y \iff \mathcal{M}_{\mathrm{S}}, a_{\mathrm{S,t}} \models x\{S_1, \ldots, S_n\}y, \text{ for each } t \in T \text{ with } a_{\mathrm{T}}(i)^- < t < a_{\mathrm{T}}(i)^+.$$

Note that we only require that the spatial constraints hold in the interior of the interval. This is necessary since if these spatial constraints need to hold at starting and endpoints of the interval as well, then it would not be possible that a base relation holding between objects X and Y in interval I changes to a different base relation between these objects in any interval met by I. Hence, it would follow that a base relation holding between two object would remain the same all the time, which is apparently unacceptable.

Let us illustrate these notions by some examples: If we temporalize the region connection calculus RCC5 with respect to the point algebra, we can express that two objects X and Y are disjoint at some instant t, but overlap at some later instant t' by the following constraints:

$$t: X \text{ DR } Y,\ t < t',\ t': X \text{ PO } Y.$$

If we temporalize RCC8 with respect to the interval algebra, we obtain the calculus STCC introduced by Gerevini and Nebel [13]. Here we can state constraints such as

$$I\{\mathrm{m}, \mathrm{b}\}J,\ \ I : X \text{ DC } Y,\ \ I : Y \text{ DC } Z,\ \ J : X\{\text{NTPP}, \text{TPP}\}Y,\ \ J : Y \text{ PO } Z,$$

which express that interval I (weakly) precedes interval J, that during I region X is disconnected from region Y and Y is disconnected from region Z, that during J region X is proper part of region Y, and so on.

The semantical definitions presented so far do not impose any restrictions on how objects can change their size, shape, or position. But how can we introduce such conditions on the semantic level? To explain this, let us focus on the condition that objects need to change their position, size, and shape continuously. For the sake of simplicity, we will assume that each region in a tt-model is a homeomorphic image of the n-dimensional closed unit circle E_n (for $n = 2, 3$)—these circles provide typical examples of connected, regular closed subsets. This means that for each region $X \in \text{Reg}$, we have a continuous function $\varepsilon_X : E_n \longrightarrow S$ induced by a fixed homeomorphism between E_n and X. We will be refer to such models as *simple tt-models*.

Definition 2. *Let $\mathcal{M}$ be a simple tt-model. A path $\pi\colon T \longrightarrow \text{Reg}$ of $\mathcal{M}$ is said to be* continuous *if the function*

$$\tau\colon E_n \times T \longrightarrow S, \quad \tau(p,t) := \varepsilon_{\pi(t)}(p)$$

is continuous (in both variables, think of T as equipped with the order topology). A simple tt-model is said to be continuous *if each of its object paths is continuous, and it is* strictly continuous *if Π consists exactly of all continuous object paths possible for the regions of $\mathcal{M}$.*

Apparently, this concept of continuous object paths is closely related to the topological notion of homotopic functions, i. e., continuous transformations between continuous functions. In fact, a continuous object path π defines homotopies between arbitrary regions $\pi(t)$ and $\pi(t')$. The important point is not that $\pi(t)$ and $\pi(t')$ are homotopic (which is trivial since both are homeomorphic to E_n), but that the object path itself defines such a homotopy. For example, let π be an object path from $\mathbb{R}$ into a suitable set of all subsets of $\mathbb{R}^n$ assigning the unit circle to each $t \neq 0$ and the unit cube at $t = 0$. Then obviously this object path cannot be continuous.

Prepared with these notions, we can define a precise notion of the neighborhood graph of a spatial calculus. For this let $\mathcal{M}$ be a simple tt-model. We define the *RCCx neighborhood graph* associated to $\mathcal{M}$ as follows: Let S be an RCCx base relation. The set of *$\mathcal{M}$-neighbors* of S is defined as the smallest set of base relations, $N(S)$, satisfying the following two conditions:

- $S \notin N(S)$;
- For each pair of object paths π, π' and each pair of instants $t_0 < t_1$ of $\mathcal{M}$ with $\pi(t_0) S^{\mathcal{M}} \pi'(t_0)$ and not $\pi(t_1) S^{\mathcal{M}} \pi'(t_1)$, there exists a relation $S' \in N(S)$ and an instant $t_0 < t \leq t_1$ such that $\pi(t) S'^{\mathcal{M}} \pi'(t)$.

Thus the RCCx neighborhood graph w. r. t. $\mathcal{M}$ is defined as the directed graph $G_{\mathrm{RCC}x,\mathcal{M}}$ that has the RCCx base relations as vertices and for each base relation an edge to each of its $\mathcal{M}$-neighbors. A graph G with vertex set $R_{\mathrm{RCC}x}$ is said to be *correct* for *a class of models* if G is the neighborhood graphs w. r. t. all models of that class.

Lemma 3. *The neighborhood graph of RCC5 (cf. Fig. 2) is correct for each class of strictly continuous tt-models that instantiate all RCC5 base relations.*

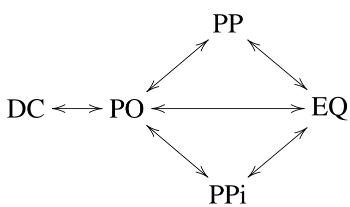


Fig. 2. The neighborhood graph of RCC5

Given a neighborhood graph G, we define the *neighborhood distance* between spatial relations as follows: For base relations B and B', $\Delta_G(B, B')$ is defined as the length of the shortest path in G between B and B'. For arbitrary relations S and S' we set

$$\Delta_G(S, S') = \min_{B \in S, B' \in S'} \Delta_G(B, B').$$

Obviously, $\Delta_G(S, S') = 0$ if and only if S intersects with S', and $\Delta_G(S, S') = 1$ if and only if S and S' are disjoint, but contain base relations B and B' respectively such that B' is a neighbor of B in G.

Finally, let us turn to the question whether continuity is expressible in the temporalized spatial constraint language presented here. The quick answer is that continuity is not expressible by formulae, but is expressible via *rules* in the language IA:RCCx. To see this, suppose that we have a constraint set, which contains the temporalized spatial constraint $I : X\{\mathrm{DC}, \mathrm{PP}, \mathrm{EQ}\}Y$. This constraint is satisfied by an assignment in a *continuous* tt-model if and only if either $I : X\ \mathrm{DC}\ Y$ or $I : X\{\mathrm{PP}, \mathrm{EQ}\}Y$ is satisfied by that assignment. To show this, let t_0 be an instant such that $I^- < t_0 < I^+$ and $X\ \mathrm{DC}\ Y$ is false at t_0, and let t_1 be an arbitrary instant in the interior of I. Then we obtain that at t_1 $X\{\mathrm{DC}, \mathrm{PP}, \mathrm{EQ}\}Y$ is true. Without restriction we may assume that $t_1 < t_0$. Now if $X\{\mathrm{DC}\}Y$ holds at t_1, there must be an instant $t_1 < t \leq t_0$ such that $X\{\mathrm{PO}\}Y$ is true at t. But this cannot be since at t one of the constraints $X\ \mathrm{DC}\ Y$, $X\ \mathrm{PP}\ Y$, or $X\ \mathrm{EQ}\ Y$ must be true.

In fact, continuity rules could be applied in tableau algorithms as well as in natural deduction systems. But this goes beyond the scope of this paper.

4 Generalized Neighborhood Graphs

In the previous section we presented a precise notion of continuous change in spatial settings, which can be described in terms of the RCC relations. In this section we will deal with the question whether there exists a continuous transformation from an initial spatial scenario to a final spatial scenario, even when the set of all possible transformations is restricted by some further constraints (so-called *transformation constraints*).

In the following, we will argue within concrete models (i. e., within the reals as flow of time and a fixed topological model such as as the Euclidean plane or the three-dimensional Euclidean space). Then the reasoning problems we are now concerned with have the following form: Let σ_s and σ_f be two spatial scenarios, each describing the same set of objects $X_1, \dots X_n$, i. e., σ_s and σ_f are sets of "interpreted" constraints where between each pair of objects a spatial base relation holds. Furthermore, let Σ be a set of constraints in which at most variables for $X_1, \dots, X_n$ occur. Now the question is whether these objects can be continuously transformed from the first into the second scenario so that none of the constraints in Σ is violated. In terms of Definition 1, we may reformulate this as follows: Are there continuous paths for the objects $X_1, \dots, X_n$ so that the constraints of σ_s and σ_f hold at the starting and the endpoint, respectively, and the constraints of Σ hold everywhere in the *closed* interval defined by starting and endpoint.

It is clear that this problem can also be expressed in terms of the temporalized spatial constraint language presented in the previous section since a problem instance $\langle \Sigma, \sigma_s, \sigma_f \rangle$ is satisfiable in a fixed topological model if and only if the constraint set

$$I_s\,\mathrm{m}\,J,\ \ J\,\mathrm{m}\,I_f,\ \ I_s : \sigma_s \cup \Sigma,\ \ J : \Sigma,\ \ I_f : \sigma_f \cup \Sigma$$

is satisfiable in a suitably chosen strictly continuous tt-model based on the reals and the topological model at hand (here I_s, I_f, and J denote intervals: I_s is an interval in which the start scenario holds, I_f an interval in which the final scenario is true, and J is the interval in which the transformation occurs).

Transformation problems $\langle \Sigma, \sigma_s, \sigma_f \rangle$ can easily be solved by using the information encoded in the classical neighborhood graph of the spatial calculus at hand (e. g., Fig. 2) *if* transformations of at most two objects are considered. But, in general, this method already fails for more than two objects. As an examples consider the scenario $\{X \text{ EQ } Y, Y \text{ EQ } Z, X \text{ EQ } Z\}$. By only applying the information encoded in the classical neighborhood graph, we cannot conclude that every change of the first constraint X EQ Y results in a change of at least one of the other two constraints [cf. 13].

The main idea to solve transformation problems is the following: Try to find a partitioning of the transformation interval into subintervals such that in each of these subintervals only a minimal number of objects has to be transformed, but the sequence of subintervals describes for all objects a continuous transformation from the start into the final scenario. This means that a transformation problem is satisfiable if there exists a sequence of satisfiable transformation problems $\langle\Sigma, \sigma_s^1, \sigma_f^1\rangle, \ldots, \langle\Sigma, \sigma_s^m, \sigma_f^m\rangle$ with $\sigma_s^1 = \sigma_s$, $\sigma_f^m = \sigma_f$, and $\sigma_f^k = \sigma_s^{k+1}$. This means, there is a chance to solve large and complicated transformation problems by solving transformation problems for a restricted number of objects.

For such restricted problems it is interesting to precompute a generalized neighborhood graph, which encodes possible transformations for a fixed number of objects. For this we represent possible scenarios for n objects $X_1, \ldots, X_n$ as $\binom{n}{2}$-tuples of spatial base relations $(S_{ij})_{1\le i<j\le n}$ where S_{ij} is the spatial base relation that holds between X_i and X_j in that scenario. Note that not each such tuple is a consistent representation of a spatial configuration, but when it does, we will refer to it as an *n-scenario*. Note that, for a spatial calculus with k relations, usually only a subset of all $k^{\binom{n}{2}}$ many tuples represents a scenario. For RCC5, for example, only 54 from $5^3 = 125$ possible triples are scenarios of three objects.

Definition 4. *The* (n,l)-neighborhood graph *(for a fixed spatial model of some spatial calculus) is the directed graph* $\mathcal{G} = (V, E)$ *defined by the following data: V is the set of all n-scenarios and E contains a directed edge from an n-scenario s to a distinct n-scenario* s' *if and only if s can be continuously transformed to* s' *by changing at most l objects.*

In the case of RCC5, both the $(2,1)$- and the $(2,2)$-neighborhood graph coincide with the classical neighborhood graph presented in Fig. 2.[2] But as previously argued, a necessary condition for solving transformation problems is to solve for each triple of objects X_i, X_j, X_k, the transformation problem restricted to these three objects. Hence in what follows we focus on the $(3,1)$-neighborhood graph. In this graph an edge from one vertex to another edge represents that exactly one object is subject to a continuous transformation, while all others are considered fixed (i. e., their object paths are (locally) constant functions). If, for instance, the first object changes its position with respect to the second object, then a scenario (r_1, r_2, r_3) can be connected to (r_1', r_2', r_3') only if (in the classical neighborhood graph) $\Delta(r_1, r_1') = 1$, $\Delta(r_2, r_2') = 0$, and $\Delta(r_3, r_3') \le 1$.

The $(3,1)$-neighborhood has the nice feature that it can be computed easily by extensively using the information encoded in the classical neighborhood graph and in the composition table of the spatial calculus at hand. More precisely, the algorithm *GNG* presented in Fig. 3 takes as input a list of base relations (denoted by *rel_list*[]) and an array representing the composition table (*compTable*[*i*][*j*] refers to the set of base re-

[2] Note that the $(2,1)$- and the $(2,2)$-neighborhood are not necessarily identical [cf. 18].

lations obtained by composing relations i and j). The function $neighbor(i)$ assigns to each base relation its set of neighbors w. r. t. the classical neighborhood graph.

The algorithm *GNG* works as follows: First, it generates a possible scenario (i, j, k), checks if it is consistent. If so, it calls a function that generates a list of all continuous successors of the scenario (i, j, k). Finally, this list is returned. In more detail, the Boolean function $isConsistent()$ checks for a triple (i, j, k), whether k is contained in $compTable[i][j]$, in other words, whether relation k can hold between objects X and Y if there exists an object Z such that $X\,i\,Z$ and $Z\,j\,Y$ are true. The function $Succ()$ generates for a triple (i, j, k) all consistent successors into which the scenario can be transformed. This is done in the following way: For the input (i, j, k) the algorithm successively generates a neighbor relation for each relation of the triple. If, for instance, l is a neighbor relation of i, the algorithm checks if (l, j, k) is consistent. If so, the relation is added to the list of possible successors. This models the qualitative change of object X in relation to some fixed objects Y and Z. By changing object X, its qualitative relation to Z can also be affected. Since in this case object Y and Z are considered fixed, the relation j cannot change. The same is analogously done for the second and third relation of the triple.

Proposition 5. *The* $(3,1)$*-neighborhood graph computed by the algorithm GNG is (semantically) correct, if the algorithm is applied to a correct* $(2,1)$*-neighborhood graph and a correct composition table.*

By applying this proposition to Lemma 2 we obtain that for RCC5 the (3,1)-graph computed by GNG is correct for each class of strictly continuous tt-models that instantiate all base relations. This graph (a subgraph of it is depicted in Fig. 5) has 54 vertices and 291 edges.[3] We also applied this algorithm for computing the $(3,1)$-neighborhood graphs of the point algebra (PA) thought of as a spatial calculus. In this case the graph consists of 13 vertices and 24 edges (cf. Fig. 4).

To put things a little bit further, we can define a refined consistency concept for transformation problems.

Definition 6. *A transformation problem* $\langle \Sigma, \sigma_s, \sigma_f \rangle$ *is said to be* (n,l)-consistent *if for each subscenario of* σ_s *consisting of* n *objects* $X_1, \dots, X_n$*, there exists a path in the* (n,l)*-neighborhood graph to the corresponding subscenario of* σ_f *(the subscenario for* $X_1, \dots, X_n$*) such that no constraint of* Σ *is violated.*

This consistency concept can be useful, when impossible transformations are to be identified. Since a problem instance with m objects is satisfiable only if it is (n,l)-consistent for all $n,l \leq m$, we can apply the (n,l)-neighborhood graph in order to find impossible transformations. To illustrate this, let us discuss the following example for the point algebra: Consider the PA-scenarios $\sigma_s = \{a < b, b < c, a < c\}$ and $\sigma_f = \{a < b, b > c, a < c\}$. Can σ_s be transformed into σ_f if we forbid that $b = c$, i. e., $b\{<,>\}c \in \Sigma$? Certainly not, because a lookup in the $(3,1)$-neighborhood graph shows that there is no path between the corresponding vertices. This can of course also be used

[3] A representation of the full $(3,1)$-neighborhood graph for RCC5 is available to the public at ftp://ftp.informatik.uni-freiburg.de/documents/papers/ki/ragni-woelfl-nghood.pdf.

```
def GNG ( rel_list[], compTable[][])
for i, j, k in rel_list[]:
      if  isConsistent (i, j, k) :
          Succ(i, j, k)
      else :
          output "scenario (i, j, k) is inconsistent";
     output "all successors of (i, j, k)": Succ(i, j, k);

def function  isConsistent (i, j, k)
   if  k ∈ compTable[i][j]:
       return true ;
   else return  false ;

def function Succ(i, j, k)
 succArray[];
 for   l ∈ {i, j, k}:
    if 1 = i :
       for m ∈ neighbor(i):
             if  isConsistent (m, j, k) && (m, j, k) ∉ succArray[]:
                 succArray[] = succArray[] ∪ (m, j, k);
             for n ∈ neighbor(k):
                   if  isConsistent (m, j, n) && (m, j, n) ∉ succArray[]:
                       succArray[] = succArray[] ∪ (m, j, n);
    if 1 = j :
       for m ∈ neighbor(j):
             if  isConsistent (i, m, k) && (i, m, k) ∉ succArray[]:
                 succArray[] = succArray[] ∪ (i, m, k);
             for n ∈ neighbor(i):
                   if  isConsistent (n, m, k) && (n, m, k) ∉ succArray[]:
                       succArray[] = succArray[] ∪ (n, m, k);
     if 1 = k:
       for m ∈ neighbor(k):
             if  isConsistent (i, j, m) && (i, j, m) ∉ succArray[]:
                 succArray[] = succArray[] ∪ (i, j, m);
             for n ∈ neighbor(j):
                   if  isConsistent (n, j, m) && (n, j, m) ∉ succArray[]:
                       succArray[] = succArray[] ∪ (n, j, k);
 output succArray[]
```

Fig. 3. The algorithm GNG computes the $(3,1)$-neighborhood graph from a set of base relations, a composition table, and a classical neighborhood graph

in more complex formal calculi. For a given transformation problem, check first if for each pair of objects X and Y with $X\, r_1\, Y \in \sigma_s$ and $X\, r_2\, Y \in \sigma_f$, there is a path from r_1 to r_2 in the classical neighborhood graph. Then for each triple of objects X, Y, and Z with $X\, r_1\, Y, Y\, r_2\, Z, X\, r_3\, Z \in \sigma_s$ and $X\, r'_1\, Y, Y\, r'_2\, Z, X\, r'_3\, Z \in \sigma_f$, check if there is a path in the $(3,1)$-, $(3,2)$-, or $(3,3)$-neighborhood graph from (r_1, r_2, r_3) to (r'_1, r'_2, r'_3) so that during that none of the constraints in Σ is violated, and so on.

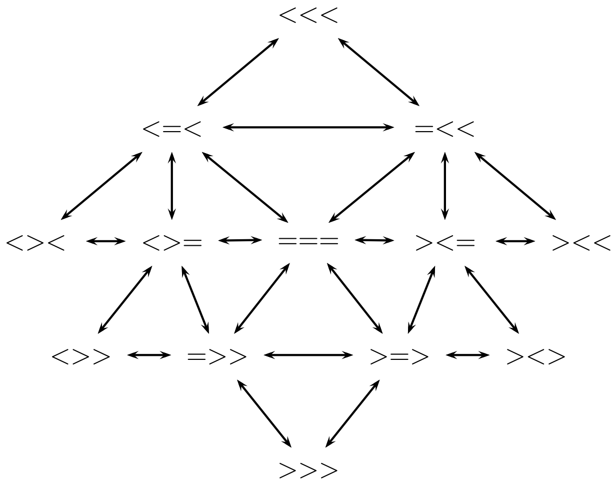


Fig. 4. The generalized neighborhood graph of the point algebra. The relation triple $r_1\,r_2\,r_3$ in the node encode a scenario for the constraints $X\,r_1\,Y$, $Y\,r_2\,Z$, and $X\,r_3\,Z$.

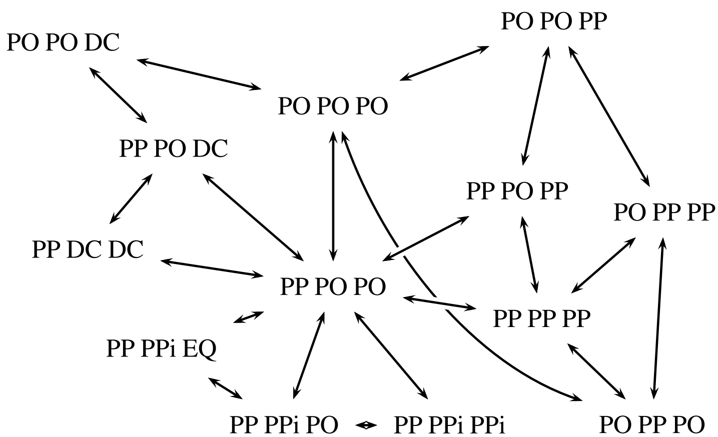


Fig. 5. A subgraph of the $(3,1)$-neighborhood graph for RCC5

5 Summary and Outlook

We started from the question how spatial constraint calculi can be temporalized. In this context, we presented a precise notion of continuous change that seems to be conceptually adequate for temporalized topological calculi. Such a precise concept is necessary for a well-founded semantics of temporalized calculi dealing with continuous transformations. Moreover, it can be used to analytically prove the correctness of neighborhood graphs.

In a second step we considered so-called transformation problems, i. e., problems of the kind whether some spatial configuration can be continuously transformed into another configuration, even if these transformations are constrained by further conditions. Solving such problems may be especially interesting, for instance, if we want to plan how objects have to be moved in space in order to reach a specific goal state.

The classical neighborhood graphs discussed in the literature only represent possible continuous transformations of at most two objects. We proposed a concept that eliminates this limitation. These generalized neighborhood graphs may also be considered an appropriate tool for solving transformation problems, because they mirror the notion of k-consistency known from static reasoning problems. This idea is reflected in our definition of (n,l)-consistency.

Future work will be concerned with the following questions: What is the exact relationship between (n,l)-consistency and satisfiability of transformation problems? Are there tractable classes of these problems? How can the notion of (n,l)-consistency be used to identify such classes? Finally, how can temporalized QSR be used to solve spatial planning scenarios?

Acknowledgments

This work was partially supported by the Deutsche Forschungsgemeinschaft (DFG) as part of the Transregional Collaborative Research Center SFB/TR 8 Spatial Cognition. We would like to thank Bernhard Nebel for helpful discussions. We also gratefully acknowledge the suggestions of three anonymous reviewers, who helped improving the paper.

References

[1] J. F. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, 1983.

[2] B. Bennett. Space, time, matter and things. In *FOIS*, pages 105–116, 2001.

[3] B. Bennett, A. G. Cohn, F. Wolter, and M. Zakharyaschev. Multi-dimensional modal logic as a framework for spatio-temporal reasoning. *Applied Intelligence*, 17(3):239–251, 2002.

[4] B. Bennett, A. Isli, and A. G. Cohn. When does a composition table provide a complete and tractable proof procedure for a relational constraint language? In *Proceedings of the IJCAI97 Workshop on Spatial and Temporal Reasoning*, Nagoya, Japan, 1997.

[5] A. G. Cohn. Qualitative spatial representation and reasoning techniques. In G. Brewka, C. Habel, and B. Nebel, editors, *KI-97: Advances in Artificial Intelligence*. Springer, 1997.

[6] E. Davis. Continuous shape transformation and metrics on regions. *Fundamenta Informaticae*, 46(1-2):31–54, 2001.

[7] M. J. Egenhofer and K. K. Al-Taha. Reasoning about gradual changes of topological relationships. In A. U. Frank, I. Campari, and U. Formentini, editors, *Spatio-Temporal Reasoning*, Lecture Notes in Computer Science 639, pages 196–219. Springer, 1992.

[8] M. Erwig and M. Schneider. Spatio-temporal predicates. *IEEE Transactions on Knowledge and Data Engineering*, 14(4):881–901, 2002.

[9] C. Freksa. Conceptual neighborhood and its role in temporal and spatial reasoning. In *Decision Support Systems and Qualitative Reasoning*, pages 181–187. North-Holland, 1991.

[10] D. Gabelaia, R. Kontchakov, A. Kurucz, F. Wolter, and M. Zakharyaschev. Combining spatial and temporal logics: Expressiveness vs. complexity. To appear in Journal of Artificial Intelligence Research, 2005.

[11] A. Galton. *Qualitative Spatial Change*. Oxford University Press, 2000.

[12] A. Galton. A generalized topological view of motion in discrete space. *Theoretical Compututer Science*, 305(1-3):111–134, 2003.

[13] A. Gerevini and B. Nebel. Qualitative spatio-temporal reasoning with RCC-8 and Allen's interval calculus: Computational complexity. In *Proceedings of the 15th European Conference on Artificial Intelligence (ECAI-02)*, pages 312–316. IOS Press, 2002.

[14] M. Knauff. The cognitive adequacy of allen's interval calculus for qualitative spatial representation and reasoning. *Spatial Cognition and Computation*, 1:261–290, 1999.

[15] P. Muller. A qualitative theory of motion based on spatio-temporal primitives. In A. G. Cohn, L. K. Schubert, and S. C. Shapiro, editors, *Proceedings of the Sixth International Conference on Principles of Knowledge Representation and Reasoning (KR'98), Trento, Italy, June 2-5, 1998*, pages 131–143. Morgan Kaufmann, 1998.

[16] P. Muller. Topological spatio-temporal reasoning and representation. *Computational Intelligence*, 18(3):420–450, 2002.

[17] B. Nebel and H.-J. Bürckert. Reasoning about temporal relations: A maximal tractable subclass of Allen's interval algebra. Technical Report RR-93-11, Deutsches Forschungszentrum für Künstliche Intelligenz GmbH, Kaiserslautern, Germany, 1993.

[18] M. Ragni and S. Wölfl. Branching Allen: Reasoning with intervals in branching time. In C. Freksa, M. Knauff, B. Krieg-Brückner, B. Nebel, and T. Barkowsky, editors, *Spatial Cognition*, Lecture Notes in Computer Science 3343, pages 323–343. Springer, 2004.

[19] D. A. Randell, Z. Cui, and A. G. Cohn. A spatial logic based on regions and connection. In B. Nebel, W. Swartout, and C. Rich, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the 3rd International Conference (KR-92)*, pages 165–176. Morgan Kaufmann, 1992.

[20] M. B. Vilain, H. A. Kautz, and P. G. van Beek. Contraint propagation algorithms for temporal reasoning: A revised report. In D. S. Weld and J. de Kleer, editors, *Readings in Qualitative Reasoning about Physical Systems*, pages 373–381. Morgan Kaufmann, 1989.

[21] F. Wolter and M. Zakharyaschev. Spatio-temporal representation and reasoning based on RCC-8. In A. Cohn, F. Giunchiglia, and B. Selman, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the 7th International Conference (KR2000)*. Morgan Kaufmann, 2000.