

21st International Joint Conference on Artificial Intelligence

Proceedings

**Workshop on
Spatial and Temporal Reasoning**

Sunday, 12 July 2009
Pasadena, California, USA

*Hans W. Guesgen, Massey University, New Zealand
Mehul Bhatt, Universität Bremen, Germany*

Table of Contents

<i>Extending Temporal Causal Graphs For Diagnosis Problems</i> Lamia Belouaer, Maroua Bouzid, and Malek Mouhoub	1
<i>Extending Fuzzy Qualitative-Metric Constraint Networks for Spatial Reasoning</i> Marco Falda	9
<i>Reasoning about a Temporal Scenario in Natural Language</i> Benjamin Han	19
<i>A Study of Heuristic based Evolutionary Algorithms for Temporal Constraint Problems</i> Bahareh Jafari Jashmi and Malek Mouhoub	27
<i>9-Intersection Calculi for Spatial Reasoning on the Topological Relations between Multi-Domain Objects</i> Yohei Kurata	34
<i>Spatio-Temporal Aspects of the Monitoring of Complex Events</i> Gérard Ligozat, Zygmunt Vetulani, and Jędrzej Osiński	44
<i>H-Complexity Metric for Qualitative Spatial and Temporal Reasoning Applications</i> Carl Schultz, Robert Amor, and Hans W. Guesgen	52
<i>Pragmatic Scenario Inference on Static Spatial Configurations</i> Dan Tappan	61
<i>Interval Algebra Networks with Infinite Intervals</i> André Trudel	70

Preface

Welcome to the Workshop on Spatial and Temporal Reasoning at IJCAI-2009 in Pasadena, California. This workshop continues in the spirit of a series of such activities over the last dozen years spanning related communities of researchers that study representing and reasoning about either space or time or both. In addition, the workshop has encouraged a mix of theory and applied work. While a number of central themes recur, a wide variety of topics typify the workshops.

Various basic representational problems in space (direction, location, proximity, geometry, intersection) and in time (coincidence, order, concurrency, overlap, granularity) attract repeated attention due to their fundamental and difficult nature. Likewise, common reasoning problems thread their way through many papers on space (path finding, orientation, relative position) and time (constraint satisfaction, schedule optimisation, precedence). Beyond that, however, the richness of different ontologies, different applications, and different objectives assures that no small collection of “solutions” will serve to satisfy all needs. The established intractability of many reasoning problems also broadens the search for approximate and partial solutions.

The interchange between spatial research and temporal methods has proved fruitful, particularly in the domains of qualitative reasoning and modelling. It is the continued wish of the organisers that the presentations and interchange in this workshop stimulate cross-fertilisation, new applications of known techniques, and new approaches to well-studied applications. Your attendance at the workshop indicates your interest in finding computerised solutions to representation and reasoning problems that deal with space and time, be they geographic or robotic, dealing with transportation or communication, theoretical or applied. We hope you come away with a richer knowledge, sharing our view that space and time may serve as a unifying theme for many areas, and that we may contribute to some standardisation of terminology, principles, and results which cut through so much research.

Hans W. Guesgen
Mehul Bhatt
(Workshop Co-Chairs)

Extending Temporal Causal Graphs For Diagnosis Problems

Lamia Belouaer and Maroua Bouzid

GREYC (UMR 6072), Université de Caen Basse-Normandie,
Campus Côte de Nacre, boulevard du Marchal Juin
BP 5186 - 14032 Caen CEDEX, FRANCE

Malek Mouhoub

Dept of Computer Science, University of Regina
3737 Wascana Parkway, Regina, SK Canada S4S 0A2

Abstract

We propose a new approach for Temporal Diagnosis Problems. This approach is an extension of Bouzid and Ligeza's method for temporal diagnosis problems. In this latter work, the authors define a Temporal Causal Graph (TCG) where time delays are expressed as temporal instants. We extend the TCG by including two quantitative relations in order to handle temporal intervals. We call this new model ExTCG. Solving a temporal diagnosis problem represented by the ExTCG consists of finding all possible explanations. It is performed using a backtrack search algorithm. In many diagnosis applications, the generation of all possible explanations is not necessary. For this reason, we augment the ExTCG in order to consider the degree of causality between symptoms. We call weighted ExTCG this extended model. Solving it consists of finding the explanation having the highest probability to occur. Through a real world diagnosis application in medicine, we illustrate the weighted ExTCG and its corresponding solving algorithm.

Introduction

The goal of Model Based Diagnosis (MBD) (Brusoni et al. 1998) is finding the cause of the abnormalities observed in the behavior of the studied system. To reach this goal, MBD uses a model of the system to determine which components are responsible for observed abnormalities. We distinguish two families for MBD : diagnosis based on the coherence (Taouf 2005) and abductive diagnosis (Brusoni et al. 1998). The abductive diagnosis consists in finding explanations for given observations by using rules of inference based on the causal dependences of the system. From the beginning it was clear that time is important for MBD (Hamscher, Console, and Kleer 1992). In fact, the assumptions that a system to be diagnosed is static and that all observations are given at the same instant, are restrictive in many domains such as medicine where the duration and the order of the arisen symptoms constitute a determining factor to differentiate two pathologies. However, taking time into account makes the diagnosis more complex both from the conceptual and practical point of view (Hamscher and Davis 1984). For this reason, there are few works in literature handling temporal diagnosis (Kautz 1999). They differ in the

Copyright © 2009, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

expressiveness of the temporal knowledge.

We are interested in the approach for temporal abductive diagnosis presented in (Bouzid and Ligeza 2000). The causal structure is an AND/OR/NOT causal graph¹ representing the various kinds of causal dependencies. The temporal dimension is represented with delays. The authors define a Temporal Causal Graph (TCG) incorporating two types of connections components. The first one is a set of basic logical connectives. The second one is a set of time delay components. A temporal diagnostic problem is defined as a TCG and a set of positive (resp. negative) manifestations observed at given instants. Solving such problems consists of finding the set of initial symptoms with their truth values explaining the set of observations. It is performed using backward search algorithm in TCG by propagating temporal information ensuring coherence. Each step of the algorithm consists in replacing one node by its possible cause. For logical nodes, the transformation rules are exactly the same as in the static case ; no changes over time instants are performed. For time delay nodes, a node n is replaced by its cause (node n') and the time index changes according to the time delay specification. This approach allows to deduce dependencies between causes and effects by taking time into account. However it is not expressive enough for many real life applications as we will see in the following example motivating our work.

Example 1. *An alarm begins to ring after some smoke is detected and continues to ring as long as there is smoke. The alarm is electric and can be cut if there is a power outage.*

We consider the situation depicted in figure 1: smoke is observed and electricity is down.

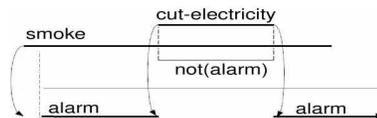


Figure 1: Graphic of the situation.

To ensure a coherent chaining of these events, extra knowledge is needed: the cut of electricity is not persistent.

¹AND/OR/NOT causal graph is a directed acyclic graph

More precisely, when electricity is back on; if smoke still persists the alarm starts ringing again. To represent this situation, we need to clarify the relations between the beginning and the end of this phenomenon:

- the beginning of smoke causes the alarm to ring if electricity is on,
- the beginning of cut of electricity causes the end of the alarm,
- at the end of cut of electricity, the alarm starts ringing again if smoke is still there.

The above information cannot be modelled with the TCG. In TCG, we can modelize the causality and the delay between events, but we cannot specify overlapping between events. Indeed, it is not able to place in time the beginning and the end of this phenomenon. We extend the work in (Bouzid and Ligeza 2000) as follows. First we extend the TCG, by introducing two quantitative relations in order to manipulate intervals. We call extended TCG (ExTCG) this new model. A temporal diagnosis problem is now defined by an ExTCG and a set of observations. To solve it, we propose an algorithm of temporal propagation in the ExTCG, in order to find all possible explanations. The generation of all explanations is not always necessary. For example, for a group of symptoms a doctor is going to prefer one diagnosis over others, by taking into account some criteria, particularly the degree of causality between symptoms. We distinguish three kinds of causality and we give a numeric qualification (a *weight*) for each one. This weight is considered as a probability. It allows us to get explanations having the highest probability to occur. In this way, we define the Weighted ExTCG. To find such explanations, we propose an algorithm of temporal propagation taking into account the weight of causality in the Weighted ExTCG.

The rest of the paper is organized as follows. First, we describe how we extend the TCG. Then, we give the corresponding solving method. After, we define the Weighted ExTCG, and we give the second method of resolution. Then we present an application based on the second approach. Finally, we conclude and discuss the future work.

Extended Temporal Causal Graph

The language of temporal representation

From a topological point of view, we suppose that we are in a frame where time is linear and discrete. The ontology of time considers both the instant and the interval. A symptom s is interpreted:

- on an interval $[t_1, t_2]$: s is true for all instants between t_1 and t_2 (t_1 and t_2 included),
- or on an instant : s is true over t_1 .

Definition 1 (episode). *An episode is defined by a pair (s, i) where s is a symptom. i is an interval or an instant. A symptom represents some phenomenon reflecting an occurrence of a partial characteristic of the system.*

We are interested in the truth value of s over time. An episode can have dates beginning as noted by *start_date*

or of ending as noted by *end_date*. To indicate that a *start_date* or *end_date* are unknown we use -1 . For example if these dates are unknown for s we note as follow: $(s, [-1, -1])$.

The diagnosis is based on the analysis of several observations spaced out in a unpredictable way in time.

Definition 2 (observation). *The observations are given by a set OBS of pairs (o, i) where o is a symptom and i is an interval or an instant.*

The semantics of (o, i) is that o was observed during i (if i is an interval), or at the instant i (if i is an instant).

Relations

Let us note by R the set of relations proposed in our approach. $R = \{(r_t, r_c) \mid r_t \in R_T, r_c \in R_C\}$ where: R_C defines all the causal relations and R_T defines all the temporal relations.

Causal relations Causality means influence leading to occurrence of selected episodes as a result of the appearance of some other ones. There are three different causal relations referring to the “strength” of causality. First one, episode e always causes episode e' when the former occurs, the occurrence of n' is bound to be caused by n . Second one, episode e always causes episode e' always causes episode e' when the former occurs, but there are several possible different episodes causing e' as well. Third one, occurrence of episode e may cause episode e' to occur, however there are cases when e' does not follow e . We assume the second type of causality and we noted by *cause*.

Causal relations between episodes are expressed by arcs in the graph structure. In other words, whenever there is a causal relation between nodes n and n' , there is a directed arc pointing from n to n' .

To express logical relations between episodes, we consider three basic logical functions : AND, OR and NOT as depicted in figure 2.

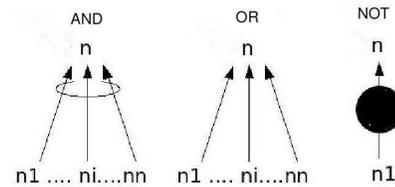


Figure 2: Graphical representations of AND/OR/NOT functions.

If episodes $\{n_1, \dots, n_n\}$ cause n only when occurring together, then all the arcs from $\{n_1, \dots, n_n\}$ to n are joined by an “horizontal” arc and form an AND branching. We call n an AND-node. If there are arcs pointing to n independently from more than one node, then n is referred to as an OR-node. Further, there is an arc labelled NOT from n_1 to n , whenever the negation of n_1 causes n to occur and vice versa.

Temporal relations $R_T = \{r_{qi}(\partial t) \mid r_{qi} \in R_{QL}\}$, where ∂t is a positive integer indicating the delay and R_{QL} the set of two temporal relations defined below and described in figure 3.

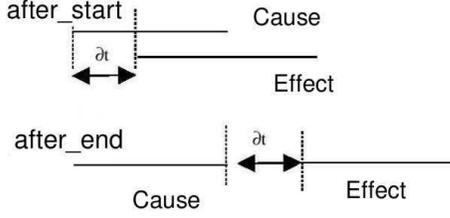


Figure 3: Quantitative relations.

after_end represents that the effect is after the end of the cause. *after_start* represents that the effect is after the beginning of the cause. These relations are transformed into equations and inequalities allowing to refer to an instant or to locate two episodes one to another. In a more precise way; we have e , c_1 and c_2 three episodes:

- If the relation is *after_start* with a delay d_1 between c_1 and e then $start_time(c_1) = start_time(e) - d_1$ and $end_time(c_1) > start_time(e) - d_2$.
- If the relation is *after_end* with a delay d_2 between c_2 and e then $end_time(c_2) = start_time(e) - d_2$ and $start_time(c_2) < start_time(e) - d_1$.

Extension of a TCG

Definition 3. A Temporal Causal Graph (TCG) is a structure $G = (N, F, H)$, where :

- N : set of all symptoms.
- $F = \{(n, f, [n_1, \dots, n_k])\}$ denotes the set of logical connections such that $f \in \{AND, OR, NOT\}$, n_1, \dots, n_k are the input nodes and n the output node.
- $H = \{(n, k, n')\}$ denotes the set of time delay components where n is the input node, k is the time delay and n' is the output node.

The graphic representation of a relation r , $r \in R$, is given by an edge labelled by r_t and r_c (where $r_t \in R_T$ and $r_c \in R_C$). An extended temporal causal graph noted ExTCG is a TCG where nodes are episodes, and edges are the relations defined by the set R .

Definition 4. An Extended Temporal Causal Graph (ExTCG) is a structure $G = (E, F, R)$, where:

- E : set of episodes.
- F : is the same set defined in a TCG.
- R : set of causal and temporal relations between episodes.

Example 2. Figure 4 represents an ExTCG.

Remark 1. It is assumed that there are only “pure” AND, OR, NOT and H connections. If the connection is of a mixed type, it can be split into separate ones by introducing an artificial intermediate node. The intention is to keep a clear

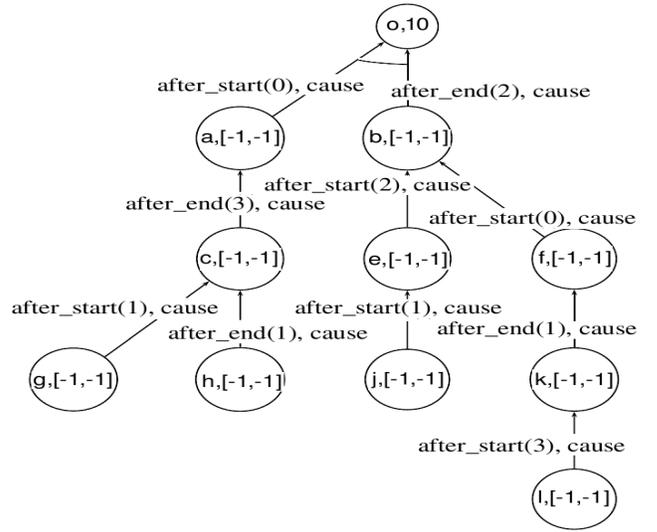


Figure 4: An example of an ExTCG.

distinction between AND, OR, NOT and delay nodes. In our approach, we keep this principle in order to have pure AND, OR and NOT nodes but we allow that a node can be pointed by edges each having a precise temporal link.

Solving Method

Solving a Temporal Diagnostic Problem, means finding all the episodes explaining the given observations and placing them in time.

Definition 5 (Temporal Diagnostic Problem). A Temporal diagnostic problem P is defined by an ExTCG and a set of observations OBS as follows: $\{ExTCG, OBS = ((o_1, i_{o_1}), \dots, (o_n, i_{o_n}))\}$, where ExTCG represent the theoretical domain and OBS the set of observations o_k at the instants i_{o_k} .

Definition 6 (Solution to a Temporal Diagnostic Problem). Let us consider a temporal diagnostic problem $P = \{ExTCG, OBS\}$. A solution S to P is defined by a set of pairs of initial nodes: $\{(e_1, i_{e_1}), \dots, (e_n, i_{e_n})\}$, where i_{e_j} can be an interval or an instant, such that:

$$ExTCG \cup S \models OBS \text{ and } S \text{ is consistent.}$$

To solve a temporal abductif diagnostic problem, we proceed in the following two steps.

Step 1 : Abduction and propagation

In general, abductive reasoning consists in finding the best explanation for a set of data or observations by using inference rules which are interpreted backwards. For the sake of this approach, abduction is considered as a backward search procedure.

This step allows to generate all sets of explanations as well as the equations and inequalities corresponding to temporal information. A solution, noted sol is a couple (*explanation*, $\{equations/inequalities\}$). Let us note by Sol_{solve} , all the solutions sol generated in this step.

Given an observation o , we visit the ExTCG using a depth-first strategy, moving backward from o to non abductibles episodes. In every step of abduction, we replace a node by its possible cause. Every temporal relation is converted into equation and inequality. In a more formal way :

- At abductive step k , o is an AND node, caused by $\{c_1, c_2, \dots, c_n\}$ so : for each c_i , we convert r_t (r_t temporal relation between c_i and o) in equations and inequalities, and o is replaced by $\{c_1, c_2, \dots, c_n\}$.
- At abductive step k , o is an OR node, caused by $\{c_1, c_2, \dots, c_n\}$ so : we select one by one the causes of o . For the number of causes, we duplicate the solution sol for every c_i , so we have one sol_{c_i} by node c_i . We replace o by every cause c_i and we do the same thing. Finally, we add every sol_{c_i} to Sol_{solve} .
- At abductive step k , o is an NOT node, caused by c . The steps are the same as for the link AND. Furthermore, it is necessary to modify the truth value of c ; if the truth value of o is true (resp. false) then we set c to false (resp. true).

If in a step one node is not abductible, it is considered to be explaining o . It will be added to the explanation corresponding to sol . This step generates the set Sol_{solve} .

Example 3. Let P be a temporal diagnosis problem given by $(o, 10)$ and by the ExTCG in figure 4.

At time 10, o is an AND-node caused by the conjunction of a and b (see figure 4). By logical transformation (AND) o is replaced with the conjunction of a and b . Also, we have indication about delays between causes (a and b) and observation (o). By temporal transformation, we have :

- $start_time(o) = 10$.
- $start_time(a) = start_time(o) - 0$,
 $end_time(a) > start_time(o) - 0$.
- $start_time(b) < start_time(o) - 2$,
 $end_time(b) = start_time(o) - 2$.

Now, the set of $OBS = \{a, b\}$. First, we consider a as an observation. We do the transformations by respecting logical and time relations. We do this in recursive way. We consider c the new observation. It is an OR-node. By logical transformation (OR) c is repalced by g or h . So we consider two possible solutions. Time transformation is done as described in the Section 2.2.2. When the current node is not abductible we stop the backward search. If the set of OBS is not empty, we take the next observation in this set. The next one is b , an OR-node. So we consider two possible paths. Finally, this step allows to generate $Sol_{solve} = \{sol_1, sol_2, sol_3, sol_4\}$. Each sol_i has an $explanation_i$ and a ξ_i .

We give here just one of these solutions :

$$sol_1 = \left\{ \begin{array}{l} explanation_1 = (g, [-1, -1]) \wedge (j, [-1, -1]) \\ \xi_1 = \left\{ \begin{array}{l} start_time(o) = 10 \\ start_time(a) = start_time(o) - 0 \\ end_time(a) > start_time(o) - 0 \\ start_time(c) < start_time(a) - 3 \\ end_time(c) = start_time(a) - 3 \\ start_time(g) = start_time(c) - 1 \\ end_time(g) > start_time(c) - 1 \\ start_time(b) < start_time(o) - 2 \\ end_time(b) = start_time(o) - 2 \\ start_time(e) = start_time(b) - 2 \\ end_time(e) > start_time(b) - 2 \\ start_time(j) = start_time(e) - 1 \\ end_time(j) > start_time(e) - 1 \end{array} \right. \end{array} \right.$$

The possible explanations to this problem: $(g, [-1, -1]) \wedge (j, [-1, -1])$, $(g, [-1, -1]) \wedge (l, [-1, -1])$, $(h, [-1, -1]) \wedge (j, [-1, -1])$ or $(h, [-1, -1]) \wedge (l, [-1, -1])$.

Step 2 : Resolution

This step consists in solving equations and inequalities generated in the first step. The intention is to locate the explanation temporally. We consider these equations and inequalities as numeric temporal constraints. For this reason we use the Simple Temporal Problem (STP) (Planken, de Weerd, and van der Krogt 2008).

An STP is defined by the structure (V, C) :

- V : set of variables $\{X_1, \dots, X_n\}$.
- C : set of binary constraints over pairs of these variables. The variables have continuous or discrete domains; each variable represents a time point.

A solution to the STP is an assignment of a real value to each time-point variable such that the differences between each constrained pair of variables fall within the range specified by the constraint. To see how an STP can be used to find the ansewer to our question, we first consider :

- the set of temporal variables $V = \{x_1, \dots, x_{2n}\}$ representing the start and the end of n episodes ($x_i = start_date$ or end_date),
- the domain : \mathbb{N}^+ ,
- and the set of equations and inequalities constitutes C .

Then we define the relation of precedence between two episodes E_1 and E_2 as :

- $start_date(E_2) - start_date(E_1) \in [d, +\infty[$ and,
- $start_date(E_2) - end_date(E_1) \in [d, +\infty[$,

where d is a positif integer indicating the delay. We use STP in order to verify the coherence of the given temporal information and to solve equations/inequalities.

To solve the formed STPs we use the P³C algorithm presented in (Planken, de Weerd, and van der Krogt 2008).

Example 4. By applying P³C to sol_1 we have the following explanation to P : $(g, [5, 7]) \wedge (j, [4, 5])$.

This kind of search called *exhaustive search* returns k solutions, where every solution $sol_i = (explanation_i, stp_i)$. In order to locate all explanation, we must solve every stp_i . For example, for an ExTCG with five OR nodes, we have 40 STPs to solve. In order to reduce this number, we can use different types of causality as described in the following section.

Weighted ExTCG

In literature, there are several types of causality. We assume here three types. For every type of causality we give a numeric qualification (a weight):

- 1 : episode e always causes episode e' , the occurrence of e' is bound to be caused by e ,
- 0.7 : episode e always causes episode e' , but there are several possible different episodes causing e' as well,
- 0.3 : occurrence of episode e may cause episode e' , however there are cases where e' does not follow e .

So, R_C becomes $R_C = \{r_c(p) \text{ with } p = \{1, 0.3, 0.7\}\}$.

A Weighted ExTCG is an ExTCG where each edge is labelled by the temporal relation and the weight of causality of this link. Figure 5 describes the ExTCG in figure 4 when adding causality weights.

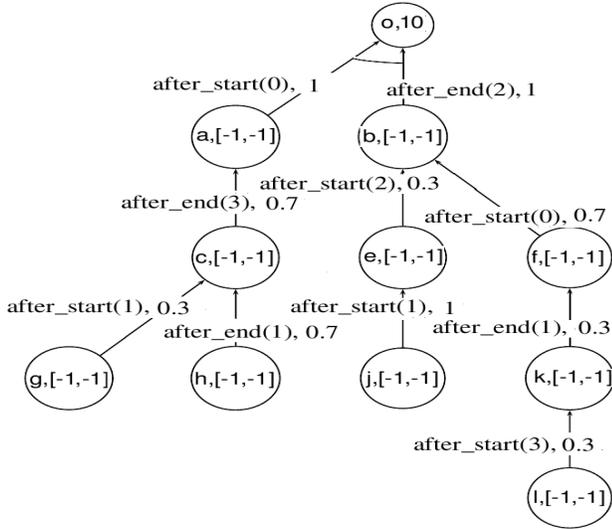


Figure 5: A Weighted ExTCG.

Computation of the weight of causality

We define three functions : $W_{relation}$, W_{level} and $W_{solution}$.

Definition 7 (Weight of a relation). Let e_1 causes e_2 ($e_1, e_2 \in E$). $W_{relation}(e_1 \rightarrow e_2)$ is a function giving the weight of relation between e_1 and e_2 : $W_{relation}(e_1 \rightarrow e_2) = p$.

In a more general way, let us consider e an episode caused by $\{e_1, e_2, \dots, e_f\}$. e is an AND node. If all relations have the same weight p , we note : $W_{relation}(\{e_1, e_2, \dots, e_f\} \rightarrow e) = p$.

A Weighted ExTCG is a graph. An observation is a root. We decompose this graph into levels. The first one, level 0 corresponds to root. The last level is given by the depth of the tree (see figure 6). At level 0, the weight is equal to 1. At level i , the weight is computed with the function $W_{level}(i)$ as follows.

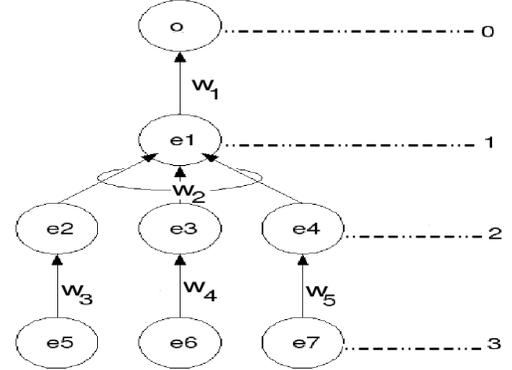


Figure 6: Levels of a weighted ExTCG.

$$\begin{aligned}
 W_{level}(0) &= 1 \\
 W_{level}(1) &= W_{level}(0) * W_{relation}(e_1 \rightarrow o) \\
 W_{level}(2) &= W_{level}(1) * W_{relation}(\{e_2, e_3, e_4\} \rightarrow e_1) \\
 W_{level}(3) &= W_{level}(2) * W_{relation}(e_5 \rightarrow e_2) \\
 &\quad * W_{relation}(e_6 \rightarrow e_3) * W_{relation}(e_7 \rightarrow e_4) \\
 &= W_{level}(2) * \prod_{j=1}^3 W_{relation}(e_j^3 \rightarrow e_j^2)
 \end{aligned}$$

Definition 8 (Weight of level). $W_{level}(i)$ is a function giving the weight of level i :

$$W_{level}(i) = W_{level}(i-1) * \prod_{j=1}^{|\chi^{i-1}|} W_{level}(e_j^i \rightarrow e_j^{i-1})$$

- χ^i set of episodes of level i and $|\chi^i|$ the number of episodes in χ^i
- e_j^i : is the episode j in χ^i

Let e be an OR node in level $(i-1)$, caused by $\{e_1, \dots, e_n\}$. Level i has n values of weight of levels.

A solution sol is given by a weighted ExTCG which is a sub graph of the principal weighted ExTCG. Computing the weight of a solution means multiplying all the weights labelling every edge, of the weighted ExTCG, considered as a solution. In a more formal way, we note $W_{solution}(explanation, stp)$ this product.

The search for explanations uses the weight of causality as heuristics and is performed in three steps. The third step is the resolution of STPs, and is the same as in exhaustive search. We describe here the two first steps of the search with heuristic.

Step 1 : Search for threshold solution and its weight

It consists in backward research by replacing every OR node, having several causes, with a cause having the highest

weight of causality. Formally, for an OR node e caused by $\{c_1, c_2, \dots, c_n\}$ we select a cause c_c as :

$$\max(W_{relation}(c_1 \longrightarrow e), \dots, W_{relation}(c_n \longrightarrow e)).$$

So, we build the first solution. We consider this solution as a threshold solution, noted $\zeta_{solution}$. Its weight noted $\rho_{\zeta_{solution}}$ is the threshold weight.

Example 5. Let us consider the part of Weighted ExTCG given by the figure 7. b is an OR node caused by $\{e, f\}$.

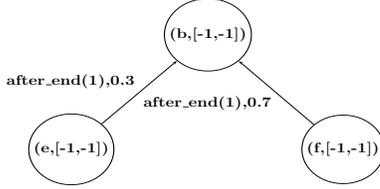


Figure 7: A sub graph of Weighted ExTCG.

We prune the subtree in which e is a root and we select the subtree in which f is a root. More precisely, we have :

$$\max(W_{relation}(f \longrightarrow b), W_{relation}(e \longrightarrow b)) = W_{relation}(f \longrightarrow b)$$

In the case where all relations have the same weight, we select the subgraph with minimal depth. Finally, we determine the threshold solution $\zeta_{solution} = ((h \wedge l), stp_\zeta)$ with a weight $\rho_{\zeta_{solution}} = 0.0308$.

A robust explanation is an explanation been considered many times for one observation. So, it has a high probability to occur. In order to generate this kind of explanation, we use the threshold weight as a minimal weight and we try to find explanations with better weight.

Step 2 : Search for solutions having better weight than the threshold weight

It consists in going through the graph by replacing every node by its causes with respecting the fact that the weight of the current level is superior to the threshold weight. More precisely, at level i , we replace a node e of this level with its causes if and only if: $W_{level}(i) > \rho_{\zeta_{solution}}$. In case this condition is not satisfied, we do not explore the subtree below this node. This allows us to reduce the number of solutions. This step allows to generate the solution having the upper weight in threshold weight. Formally, $Sol_{solve} = \{(exp, stp)_1, \dots, (exp, stp)_m\}$, where $W_{solution}((exp, stp)_i) > \rho_{\zeta_{solution}}$.

Example 6. For the same problem P given in Example 4, we have instead of four solutions to solve, just two solutions. More precisely, this method of search generates : $Sol_{solve} = \{sol_1, sol_2\}$, where : $sol_1 = ((g \wedge j), stp_1)$ with a $W_{solution}((g \wedge j), stp_1) = 0.063$ and $sol_2 = ((h \wedge j), stp_2)$ with a $W_{solution}((h \wedge j), stp_2) = 0.147$.

Introducing the causality as an heuristic allows to reduce the number of STPs to solve. Also, it allows to rerun the explanations having a very strong causality. For example, in

the case of a weighted ExTCG with 5 OR nodes, we have with this method only one STP to solve.

We will now give the complexity of the abduction and propagation steps. The complexity in the worst case of these steps in the exhaustive search and search based on weight of causality is the same. The complexity is $O(nm^k)$, where n is the number of AND nodes, m is the number of OR nodes and k is the depth of the ExTCG.

Case Study

Temporal abductive diagnosis is very important for medicine. In literature several examples exist and the most known one is MYCIN (Shortliffe 1976). We set up an application dedicated to identify the causes of death owed to the bradycardia.

Formalization

We make some official reports² from the results given by the medical examination of Holter (Avilés et al. 2004) and we have deduced the causes of death. We give here several causes ordered by the weight of causality:

- cardiac diseases (sinus node disease, heart attack, . . .),
- not cardiac diseases (intoxication with certain drugs, serious illnesses, . . .),
- and in certain cases, no cause is identified.

Our example concerns patients having a congenital heart disease and as pathology the bradycardia.

$$congenitalheartdisease \wedge bradycardia \rightarrow death$$

By exploiting statistics, this causality is very important since these two pathologies often cause death. So we deduce that the weight of causality is 0.7:

$$\{congenitalheartdisease, \langle afterstart(0), 0.7 \rangle\} \wedge \{bradycardia, \langle afterstrat(1), 0.7 \rangle\} \rightarrow (death, 8)$$

This is how we construct our knowledge base.

Approach based on weight of causality

Figure 8 is an extract of screen shot of our application. It points out the problem of diagnosis to be solved, observation $(death, 8)$ and all the rules given by the weighted ExTCG (the left graph of figure 8).

For this example, the explanation having the upper weight in threshold weight (given by the right graph of figure 8) is: the patient has a ischemic heart disorder and a bradycardia caused by a very important stress during the interval of time $[5, 6]$, where the time unit is one day. The weight of this explanation is 0.2401. In this example, it is the threshold solution that we consider since there are no solutions of better weight.

For ischemic heart disorder the location in time is not really important, because it is a genetic malformation. For this reason, we consider the relation *after_start* with delay 0. Another possible formalization for this relation is to set the delay to 8 since the observation is $(death, 8)$, so, we can say that heart disorder ischemic starts at date 0 to the death.

²We do not take into account a context in diagnosis, for example the age and sex of the patient.

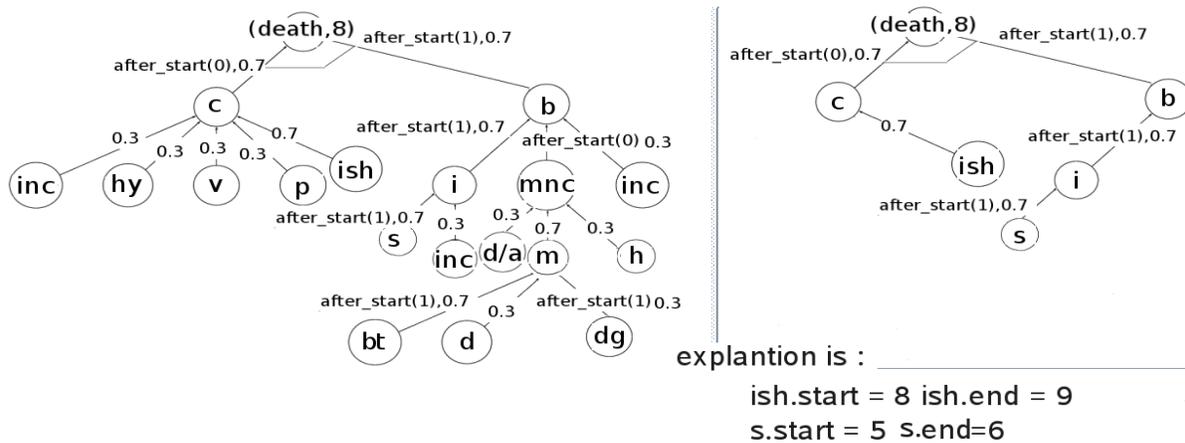


Figure 8: Diagnosis problem and its solution.

Discussion

In this paper we define an extended temporal causal graph to solve diagnosis problems.

Abduction corresponds to a backward search. Initially, the graph is traversed from observations to the initial breakdowns by going up the causal relations in order to determine a candidate explanation. Then a predicative step is carried out in forward search in order to assure the coherence. In order to reject the explanations temporally incoherent, the temporal constraints are propagated in each step of abduction. In the past years, two approaches have been proposed for temporal abduction (Gamper 1996) (Brusoni et al. 1997). These two approaches differ by the type of temporal constraints handled: qualitative for (Gamper 1996) and both qualitative and quantitative for (Brusoni et al. 1997). The temporal constraints used in (Brusoni et al. 1997) are represented by an STP, which makes it possible to benefit from the properties of local propagation (by using LaTer). The two approaches limit the complexity of abduction by assuming that an initial breakdown can have only one single occurrence. (Gamper 1996) made a very restrictive assumption that an effect can have a single cause.

In our approach, we consider that an effect can be caused by several causes. Abduction is made to replace each effect by these temporal causes and to propagate temporal information progressively. Our objective is to describe the qualitative temporal information between the episodes and to be able to transform this description into quantitative information being able to be propagated to ensure the temporal coherence of the explanation. These relations enable us to define the limits of intervals.

In the case where all relations have the same weight, we select the subgraph with a minimal depth. This choice is arbitrary. We suppose that in this kind of situation one will prefer the explanation giving less intermediate causes.

Conclusion and Future Work

We have extended the TCG (Bouzd and Ligeza 2000) by including two qualitative relations in order to manipulate time intervals. In order to take into account the causality between symptoms, we have augmented the extended TCG (ExTCG) into a new structure that we call weighted ExTCG. We have developed search algorithms respectively for solving the ExTCG and the weighted ExTCG. These algorithms consist in a backward search by propagating temporal information. Temporal information are considered as constraints. Thus, we formalize a set of constraints associated to each possible explanation as an STP. To solve the STP we use the P³C algorithm. Finally we showed how to apply the weighted ExTCG to a real life application in medicine.

One possible improvement to this work is to integrate more powerfull models into the weighted ExTCG in order to manipulate the weights expressing preferences. These models can be qualitative such as CP-nets (Boutilier et al. 2004) or quantitative such as c-semiring (BISTARELLI, MONTANARI, and ROSSI 1997).

References

- Avilés, F.; San Román, J.; Frade, J.; Valdés, M.; Sánchez, A.; de la Fuente, L.; Peñarrubia, M.; Fernández, M.; Tejedor, P.; Durán, J.; et al. 2004. Cardiopatía isquémica. Marzo 2004. Número 03-Volumen 57 p. 201-208 Regeneración miocárdica mediante la implantación intracoronaria de células madre en el infarto agudo de miocardio. *Rev Esp Cardiol* 57:201-208.
- BISTARELLI, S.; MONTANARI, U.; and ROSSI, F. 1997. Semiring-Based Constraint Satisfaction and Optimization. *Journal of the ACM* 44(2):201-236.
- Boutilier, C.; Brafman, R.; Domshlak, C.; Hoos, H.; and Poole, D. 2004. CP-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements. *Journal of Artificial Intelligence Research* 21:135-191.

- Bouzid, M., and Ligeza, A. 2000. Temporal Causal Abduction. *Constraints* 5(3):303–319.
- Brusoni, V.; Console, L.; Terenziani, P.; and Dupre, D. 1997. An efficient algorithm for temporal abduction. *Lecture Notes in Computer Science* 1321:195–206.
- Brusoni, V.; Console, L.; Terenziani, P.; and Dupré, D. 1998. A spectrum of definitions for temporal model-based diagnosis. *Artificial Intelligence* 102(1):39–79.
- Gamper, J. 1996. *A temporal reasoning and abstraction framework for model-based diagnosis systems*. RWTH Aachen.
- Hamscher, W., and Davis, R. 1984. Diagnosing circuit with state : an inherently underconstrained problem. 142–147. AAAI 84.
- Hamscher, W.; Console, L.; and Kleer, J. D. 1992. *Readings in Model-Based Diagnosis*.
- Kautz, H. 1999. Temporal reasoning. *MIT Encyclopedia of Cognitive Science* <http://www.research>.
- Planken, L.; de Weerd, M.; and van der Krogt, R. 2008. P³c : A new algorithm for the simple temporal problem. In *ICAPS 2008 : Proceedings of the Eighteenth International Conference on Automated Planning and Scheduling*, 256–263.
- Shortliffe, E. 1976. *Computer Based Medical Consultations: MYCIN*. New York.
- Taouf, S. 2005. *Diagnostic logique des systemes complexes dynamiques dans un contexte multi-agent*. Ph.D. Dissertation, Université Joseph Fourier–Grenoble 1.

Extending Fuzzy Qualitative-Metric Constraint Networks for Spatial Reasoning

Marco Falda

Dept. of Information Engineering

via Gradenigo, 6 – 35131 Padova (Italy)

email: marco.falda@unipd.it

Abstract

The introduction of metric information in Qualitative Spatial Reasoning represents an important expressiveness gain, but in order to be of more practical application, it should provide a way to specify vagueness and uncertainty. In this paper Rectangle Algebra is extended using Fuzzy Sets Theory and an integrated approach for adding metric information is proposed. A set of 25 fuzzy Point-Region relations is defined to provide a link between points and regions, in this way a Spatial Qualitative Algebra (SQA) between points and regions is obtained. Fuzzy metric information is then added to SQA by providing transformation functions that allow passing from qualitative to metric information and vice versa.

The new framework is applied to a small example which shows the reasoning capabilities of the integrated spatial reasoning model.

Keywords: Spatial Reasoning, Integrated models, Fuzzy Algebras

Introduction

Spatial Reasoning plays a central role in many Artificial Intelligence applications such as robot navigation, visual object recognition, intelligent image information systems, query processing in geographic databases. As in the case of other qualitative reasoning formalisms, there are basically two approaches to build a model suitable for spatial reasoning: model physical space and the objects within it or model the relationships between the objects (Frank 1996); a set of qualitative relations may be incomplete and even inconsistent, and the consistent integration of such information relies on the algebraic properties of the qualitative relations. Spatial Reasoning can be formulated using the framework of Constraint Satisfaction Problems (CSPs), for example TCSPs have been defined for reasoning about time (Dechter, Meiri, and Pearl 1991).

Different aspects of space can be considered; here, the more common classification of spatial relations into topological relations and directional relations is taken into account. Also distances are considered, but in the usual context of metrics, not as qualitative relations. Topological spatial relations are those that are invariant under continuous

transformations, such as rotation or scaling. Directional relations are defined between a reference object and a primary object with respect to a fixed frame of reference, usually determined by a predefined entity such as the North Pole. Topological information is commonly represented using extended regions as basic entities, while orientation is based on points. In this paper an algebra dealing with relations between regions, namely the Rectangle Algebra (Balbiani, Condotta, and Fariñas del Cerro 1998) is combined with the algebra of Cardinal Directions (Frank 1996). This allows obtaining a more expressive algebra, which will be called Spatial Qualitative Algebra (SQA); moreover, also metric information about distances has been added.

In Temporal Reasoning the most classical model of integration between qualitative and quantitative constraints was proposed by Meiri (Meiri 1996) who defined an extended Temporal CSP able to deal with both types of information using an unique constraint network. In Spatial Reasoning Condotta (Condotta 2000) proposed to manage these two types of information using distinct CSPs (as made for Temporal Reasoning in (Kautz and Ladkin 1991)). This paper applies the idea of Meiri to spatial constraints, that is an unique constraint network for both qualitative and metric information.

Realistic applications usually contain information pervaded by vagueness and uncertainty. This kind of notions can be dealt in the framework of Fuzzy Constraint Satisfaction Problem (FCSP) (Dubois, Fargier, and Prade 1996) where constraints are satisfied to a degree, rather than satisfied or not satisfied, and the acceptability of a potential solution is a gradual notion. The spatial constraints taken into account are extended in a fuzzy way by associating a preference degree to each basic relation of the qualitative relations and a pyramidal preference distribution to each metric constraint.

In the following section Rectangle Algebra and Cardinal Directions Algebra are extended to the fuzzy case; in the next section the metric spatial constraints are defined. Then, the Point-Region relations are introduced to allow building the Spatial Qualitative Algebra, and the integration of qualitative and metric temporal constraints in a fuzzy framework is presented. Finally algorithmic and complexity issues are considered and a simple application scenario is provided.

Qualitative constraints

The Fuzzy Rectangle Algebra fRA

Balbiani et al. (Balbiani, Condotta, and Fariñas del Cerro 1998) define the Rectangle Algebra (RA) as an extension of the well-known Allen's Interval Algebra (IA) (Allen 1983) to the bidimensional space. The IA models the relative position between any two intervals as a suitable set of thirteen basic (or atomic) relations \mathcal{I} , namely: before, meets, overlaps, starts, during, finishes (b, m, o, s, d, f) together with their inverses (bi, mi, oi, si, di, fi) and the basic relation equal (eq). The domain considered in the Rectangle Algebra is the set of rectangles with sides parallel to the axes of some orthogonal basis in \mathbb{R}^2 , this domain is called REC . A basic relation between two rectangles (atomic RA-relation) is a pair (r_x, r_y) of basic IA-relations: the x-axis relation and the y-axis relation; their set is called \mathcal{A}_{rec} . In this way, there are $13^2 = 169$ possible basic relations between any two given rectangles. If a and b are two rectangles in REC then $a(r_x, r_y)b$ is a basic RA-constraint which is satisfied if and only if the IA-constraints $a_x r_x b_x$ and $a_y r_y b_y$ are satisfied by the projections a_x, b_x and a_y, b_y , respectively.

An RA-constraint $R = \bigcup_i \{(r_{x,i}, r_{y,i})\}$ is satisfiable if and only if there exist two rectangles a and b satisfying one of the basic RA-relations in R . In Rectangle Algebra the usual operations of inversion, intersection and composition are defined. All the operations are performed on pairs of unions of basic relations; recall that the projected basic relations are IA relations, so the operations can be easily defined (Balbiani, Condotta, and Fariñas del Cerro 1998), for example the inverse of relation $R = \bigcup_i \{(r_{x,i}, r_{y,i})\}$ is $R^{-1} = \bigcup_i \{(r_{x,i}^{-1}, r_{y,i}^{-1}) : (r_{x,i}, r_{y,i}) \in R\}$. Composition between atomic IA relations has been defined in (Allen 1983) by means of a transitivity table which has an entry for all the $13^2 = 169$ combinations of atomic relations pairs. It is a true composition (see Table 1 in (Renz and Ligozat 2005)).

An RA-network is a graph $G = (V, E)$ given by a set of variables V which represent rectangles and a set M of RA-constraints between the variables in V . An RA-network \mathcal{N} with variables $V = \{v_1, \dots, v_n\}$ is consistent if and only if there exists a solution given by n rectangles $(a_1, \dots, a_n), a_i \in REC^n$ such that all RA-constraints are satisfied by the assignment $v_i = a_i, i = 1, \dots, n$. The Rectangle Algebra has the same complexity of the IA, as far as the consistency problem of an RA-network is concerned.

Saturated RA-relations are those which are obtained though the Cartesian product of two Interval Algebra relations, as an example, relation $\{(b, d), (b, b), (d, b), (d, d)\} \Leftrightarrow (b, d) \times (b, d)$ is saturated, while $\{(b, b), (d, b)\}$ is not.

IA relations are somewhat similar to mono-dimensional Region Connection Calculus (RCC) relations over regular regions (Randell, Cui, and Cohn 1992), and to give to the spatial constraints a more intuitive meaning, in this paper a sort of "orientation" in RCC relations has been introduced for relations DC, EC, O and TPP ; in this way the analogy is clearer, and 13 atomic relations \mathcal{R} corresponding to the 13 Allen's atomic relations \mathcal{I} can be devised, as shown in Table 1 and Figure 1.

Definition 1. *the set \mathcal{R} is the set of the atomic relations*

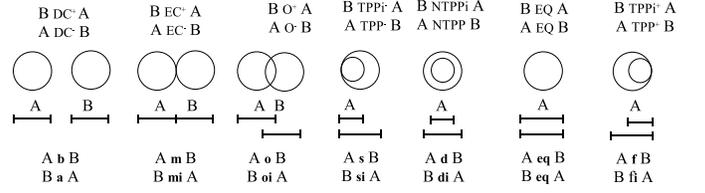


Figure 1: similarities between IA and extended RCC.

$\{DC^-, DC^+, EC^-, EC^+, O^-, O^+, TPP^-, TPP^+, TPPi^-, TPPi^+, NTPP, NTPPi, EQ\}$.

Table 1: correspondences between IA and \mathcal{R} relations.

\mathcal{I}	\mathcal{R}	\mathcal{I}	\mathcal{R}
		<i>inverses</i>	
b	DC^-	a	DC^+
m	EC^-	mi	EC^+
o	O^-	oi	O^+
s	TPP^-	si	$TPPi^-$
d	$NTPP$	di	$NTPPi$
eq	EQ	eq	EQ
f	TPP^+	fi	$TPPi^+$

If \mathcal{R} relations are combined over two orthogonal axes, relations that satisfy RCC relations are obtained; Figure 2 depicts all $13^2 = 169$ combinations, for example

$$A(eq, di)B \Leftrightarrow A(TPPi)B$$

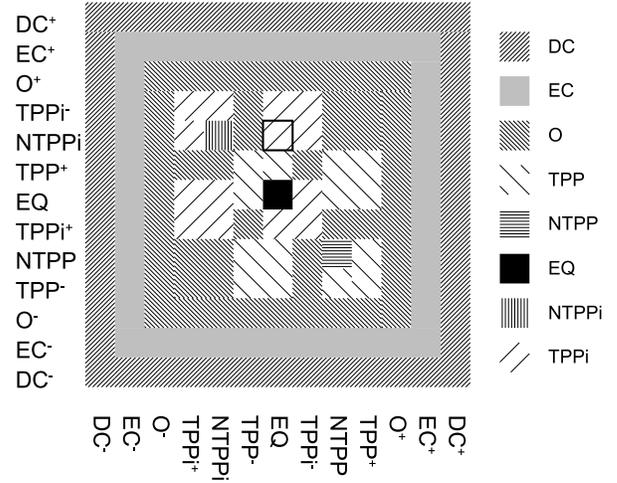


Figure 2: correspondence between RA and RCC; the box represents the relation $A(eq, di)B \Leftrightarrow A(TPPi)B$.

Flexibility and uncertainty can be introduced in Rectangle Algebra in a way similar to that proposed in (Güsgen, Hertzberg, and Philpott 1994; Badaloni and Giacomini 2006). The definition of RA-constraints is relaxed by assigning to every atomic relation $r_i \in \mathcal{R}$ a degree $\alpha_i \in [0, 1]$,

which tells the preference degree of the corresponding assignment among the others; in this way a fuzzy Rectangle Algebra fRA can be defined.

Definition 2. Let a and b be two rectangles in REC , then a fRA constraint is defined as

$$R = \bigcup_i \{(r_{x,i}, r_{y,i})[\alpha_i]\}$$

where $r_{z,i}, z \in \{x, y\}, i = \{1, \dots, 13\}$ are \mathcal{R} relations and $\alpha_i \in [0, 1]$ are the preference degrees of $r_{z,i}$. Each disjunct $(r_{x,i}, r_{y,i})[\alpha_i]$ is an atomic fuzzy RA relation.

As usual, when the preference degree is zero the corresponding \mathcal{R} relations are not specified, when 1 it is omitted. Using preference degrees the expressiveness of classical RA-relations can be increased, for example it is possible to denote the fact that *Portugal is West w.r.t Spain but partially also South and South-West* (see Figure 3).

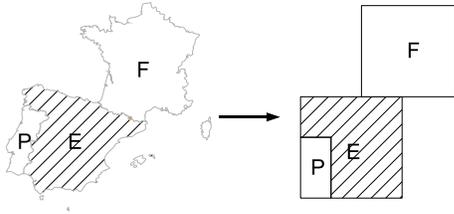


Figure 3: “Portugal is almost West w.r.t. Spain, but also partially South and partially South-West of it”.

Example 1. the position of Portugal (P) w.r.t. Spain (E) can be expressed using the fRA constraint

$$P\{(EC^-, EQ), (EQ, EC^-)[0.2], (EC^-, EC^-)[0.5]\}E$$

The preference degree of the first pair (EC^-, EQ) is 1 and it has been omitted; the other two pairs have preference degrees less than 1 but greater than zero, so their preference degrees have been written. The remaining combinations have not been specified at all, since their preference degrees are zero. Notice that, according to the Fuzzy Set Theory, the preference degrees have not to sum up to 1 as in Probability Theory.

Saturated fRA relations will be written as $a\langle R_x, R_y \rangle b$, where R_x and R_y are fuzzy \mathcal{R} relations, that is \mathcal{R} relations with an associated preference degree; as said before for the classical RA, they can be viewed as a Cartesian product of R_x and R_y atomic relations. In the case of fRA relations each component will have a preference degree given by the minimum between the pair elements, since both projections must be satisfied. The usual conventions for preference degrees zero and 1 hold.

Definition 3. a saturated fRA relation can be written as

$$a\langle R_x, R_y \rangle b$$

where $R_x = (r_{x,1}[\alpha_{x,1}], \dots, r_{x,13}[\alpha_{x,13}])$, $R_y = (r_{y,1}[\alpha_{y,1}], \dots, r_{y,13}[\alpha_{y,13}])$, $r_{z,i}, z \in \{x, y\}, i \in \{1, \dots, 13\}$ are the 13 basic \mathcal{R} relations and $\alpha_{z,i} \in [0, 1]$

are the preference degrees of $r_{z,i}$. It is equivalent to the explicit relation

$$\begin{aligned} & a\{(r_{x,1}, r_{y,1})[\min(\alpha_{x,1}, \alpha_{y,1})], \\ & \dots, (r_{x,1}, r_{y,13})[\min(\alpha_{x,1}, \alpha_{y,13})], \\ & \dots, (r_{x,13}, r_{y,13})[\min(\alpha_{x,13}, \alpha_{y,13})]\}b \end{aligned}$$

The notation $aR_x^2b = a\langle R_x, R_x \rangle b$ will be also used if the two components of the pair are the same.

Example 2. an example of a saturated fRA constraint is

$$\langle (DC^-[0.7], NTPP[0.5]), (DC^-, NTPP[0.3]) \rangle$$

and its graphic representation is shown in Figure 4; it expresses the fact that none of the implied spatial relations $(DC^-, DC^-)[0.7]$, $(DC^-, NTPP)[0.3]$, $(NTPP, DC^-)[0.5]$, $(NTPP, NTPP)[0.3]$ is fully plausible, since this fact would require a preference degree of 1, but the first is the most preferred while the last two are equally the least preferred.

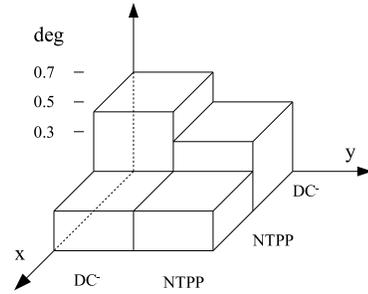


Figure 4: representation of the saturated fRA relation $\langle (DC^-[0.7], NTPP[0.5]), (DC^-, NTPP[0.3]) \rangle$.

With respect to classical RA now preference degrees have to be taken into account, therefore intersection and union have to combine them, and they will be called conjunctive and disjunctive combination respectively. The operations between fRA constraints are defined as follows:

Definition 4. given a fRA relation $R = \bigcup_i \{(r_{x,i}, r_{y,i})[\alpha_i]\}$, the inverse relation R^{-1} is defined, according to Table 1, as

$$R^{-1} = \bigcup_i \{(r_{x,i}^{-1}, r_{y,i}^{-1})[\alpha_i] : (r_{x,i}, r_{y,i})[\alpha_i] \in R\}$$

Example 3. if $R = ((DC^-, EQ)[0.3])$ then is $R^{-1} = ((DC^+, EQ)[0.3])$.

Definition 5. given two fRA relations $R = \bigcup_i \{(r_{x,i}, r_{y,i})[\alpha_i]\}$ and $S = \bigcup_j \{(s_{x,j}, s_{y,j})[\beta_j]\}$ the disjunctive combination between R and S is defined as

$$\begin{aligned} R \oplus S = \bigcup_i \{ & (r_{x,i}, r_{y,i})[\gamma_i] : (r_{x,i}, r_{y,i})[\alpha_i] \in R \wedge \\ & (s_{x,j}, s_{y,j})[\beta_j] \in S \wedge r_{x,i} = s_{x,j} \wedge \\ & r_{y,i} = s_{y,j}, \gamma_i = \max(\alpha_i, \beta_j)\} \end{aligned}$$

Example 4. the disjunctive combination of the *fRA* relations

$$R = \{(DC^-, EQ)[0.3], (NTPPi, EQ)[0.7]\} \text{ and } S = \{(DC^-, EQ)[0.5], (NTPPi, DC^+)[0.7]\} \text{ is}$$

$$T = \{(DC^-, EQ)[0.5], (NTPPi, EQ)[0.7], (NTPPi, DC^+)[0.7]\}$$

Definition 6. given two *fRA* relations $R = \bigcup_i \{(r_{x,i}, r_{y,i})[\alpha_i]\}$ and $S = \bigcup_j \{(s_{x,j}, s_{y,j})[\beta_j]\}$ the conjunctive combination between R and S is defined as

$$R \otimes S = \bigcup_i \{(r_{x,i}, r_{y,i})[\gamma_i] : (r_{x,i}, r_{y,i})[\alpha_i] \in R \wedge (s_{x,j}, s_{y,j})[\beta_j] \in S \wedge r_{x,i} = s_{x,j} \wedge r_{y,i} = s_{y,j}, \gamma_i = \min(\alpha_i, \beta_j)\}$$

Example 5. the conjunctive combination of the *fRA* relations

$$R = \{(DC^-, EQ)[0.3], (NTPPi, EQ)[0.7]\} \text{ and } S = \{(DC^-, EQ)[0.5], (NTPPi, DC^+)[0.7]\} \text{ is}$$

$$T = \{(DC^-, EQ)[0.3]\}$$

Definition 7. given two *fRA* relations $R = \bigcup_i \{(r_{x,i}, r_{y,i})[\alpha_i]\}$ and $S = \bigcup_j \{(s_{x,j}, s_{y,j})[\beta_j]\}$ the composition between R and S is defined as

$$R \circ S = \bigoplus \{(t_{x,i}, t_{y,i})[\min(\alpha_h, \beta_k)]\} \\ h, k : (r_{x,h}, r_{y,h}) \circ (s_{x,k}, s_{y,k}) = (t_{x,i}, t_{y,i})\}$$

composition between atomic relations is performed as in the case of classical Rectangle Algebra taking into account the correspondences between \mathcal{I} and \mathcal{R} .

Example 6. the composition of the *fRA* relations $R = \{(DC^-, EQ)[0.3], (NTPPi, EQ)[0.7]\}$ and $S = \{(DC^-, EQ)[0.5], (NTPPi, DC^+)[0.7]\}$ is

$$T = \{(DC^-, EQ)[0.5], (DC^-, DC^+)[0.3], (NTPPi, EQ)[0.5], (O^-, EQ)[0.5], (EC^-, EQ)[0.5], (TPPi^+, EQ)[0.5], (NTPPi, DC^+)[0.7]\}$$

The *fRA* is an algebra, that is a set of relations closed under certain operations. It is easy to see that inversion is closed, since every atomic relation in *fRA* has an inverse. Also combinations (conjunctive and disjunctive) give relations belonging to *fRA*, in fact the resulting relations are formed by atoms in \mathcal{R} coming from both or either operands. In composition the disjunctive composition of relations coming from the classical composition of atomic relations is used, while preference degrees are computed by means of max and min functions. *Stableness* (Balbiani, Condotta, and Fariñas del Cerro 1998) is guaranteed by saturated relations (also in the fuzzy case), therefore in the following just these will be considered.

The Fuzzy Cardinal Directions Algebra

When qualitative spatial positions between two points have to be described, a natural way is to model them using cardinal directions. Frank (Frank 1992) suggested methods for describing the cardinal direction of a point with respect to a reference point in a geographic space, i.e., directions are in the form of “North”, “East”, “South”, and “West” depending on the granularity. He distinguishes between two different methods for determining the different sectors corresponding to the single directions: the cone-based method and the projection-based method. The projection-based system consists of nine acceptance areas, one for each of the directions plus a neutral zone $EQ: \mathcal{C} = \{E, NE, N, NW, W, SW, S, SE, EQ\}$. The projection-based approach describes these relations in terms of the Point Algebra (PA) (Vilain, Kautz, and van Beek 1989) by specifying a point algebraic relation for each of the two axes separately. This provides the projection-based approach with a formal semantics and allows defining the Cardinal Directions Algebra, or CDA. For this reason the second method has been preferred here.

Flexibility and uncertainty can be introduced also in CDA by assigning to every atomic relation $r_i \in \mathcal{C}$ a degree $\alpha_i \in [0, 1]$, which tells the preference degree of the corresponding assignment among the others, obtaining in this way a fuzzy Cardinal Directions Algebra, or *fCDA*

Definition 8. Let a and b be two points, then a *fCDA* constraint is defined as

$$R = \bigcup_i \{(r_{x,i}, r_{y,i})[\alpha_i]\}$$

where $r_{z,i}, z \in \{x, y\}, i = \{1, 2, 3\}$ are basic Point Algebra relations $\{<, >, =\}$ and $\alpha_i \in [0, 1]$ are the preference degrees of $r_{z,i}$. Each disjunct $(r_{x,i}, r_{y,i})[\alpha_i]$ is an atomic *fCDA* relation.

As in the case of fuzzy *RA* relations, saturated *CDA* relations can be defined and the notation is similar to that for fuzzy *RA*.

Definition 9. a saturated *fCDA* relation can be written as

$$a \langle R_x, R_y \rangle b$$

where $R_x = (<_x [\alpha_{x,<}], =_x [\alpha_{x,=}], >_x [\alpha_{x,>}])$ and $R_y = (<_y [\alpha_{y,<}], =_y [\alpha_{y,=}], >_y [\alpha_{y,>}])$ and $\alpha_{z,i} \in [0, 1]$ are the preference degrees of $r_{z,i}$. It is equivalent to the explicit relation

$$a \{ (<_x, <_y) [\min(\alpha_{x,<}, \alpha_{y,<})], \dots, (<_x, >_y) [\min(\alpha_{x,<}, \alpha_{y,>})], \dots, (r_{x,>}, r_{y,>}) [\min(\alpha_{x,>}, \alpha_{y,>})] \} b$$

The notation $aR_x^2b = a \langle R_x, R_x \rangle b$ will be also used if the two components of the pair are the same.

Due to the limited number of basic relations involved, each pair of relations can be interpreted in a more natural way, as shown in Table 2 and Figure 5. For example $a \langle (> [0.7], =) \rangle b \Leftrightarrow a \{ (>, =) [0.7], (=, =) [0.9] \} b$ becomes $a \{ E [0.7], EQ [0.9] \} b$.

Table 2: interpretation of atomic $fCDA$ relations.

Pair	interpretation	Pair	interpretation
(=, =)	EQ	(<, =)	W
(>, =)	E	(<, <)	SW
(>, >)	NE	(=, <)	S
(=, >)	N	(>, <)	SE
(<, >)	NW		

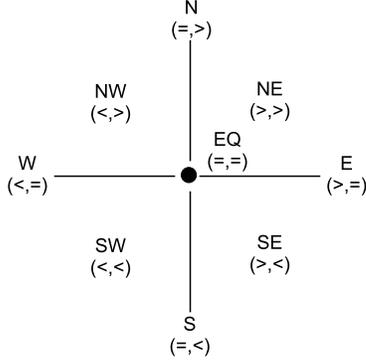


Figure 5: Point-point spatial relations.

The operations on $fCDA$ constraints are defined in a way analogous to what done for fuzzy RA, with the only difference that now the atomic relations belong to PA^2 and no more to \mathcal{R}^2 .

Fuzzy Spatial Metric constraints

In (Condotta 2000) metric spatial knowledge is represented by means of two constraint networks $\langle V, C \rangle$, one for each coordinate, which limit the possible distances between the variables in V . In this paper the metric constraints still limit the distances between the variables, but there is an unique constraint network for both coordinates. The variables therefore take values on \mathbb{R}^2 . Moreover, a fuzzy relation $R_P(C_{ij}) : \mathbb{R} \times \mathbb{R} \rightarrow [0, 1]$ is associated to each constraint C_{ij} between variables v_i and v_j in V . $[R_P(C_{ij})](d_x, d_y)$ indicates to what extent an assignment $(v_j|_x - v_i|_x, v_j|_y - v_i|_y) = (d_x, d_y)$ satisfies the constraint C_{ij} .

Normalized trapezoidal possibility distributions usually adopted in Fuzzy Temporal Networks (Marín et al. 1997; Godo and Vila 2001) are extended here to two orthogonal dimensions. They will be called pyramidal distributions.

Definition 10. a pyramidal distribution is a pair of normalized¹ trapezoidal distributions plus an associated preference degree: $\langle T_x, T_y \rangle[\alpha]$, $\alpha \in [0, 1]$

The trapezoidal possibility distributions proposed in (Badaloni, Falda, and Giacomini 2004) is adopted; each T_z is described by a 4-tuple of values, each describing four characteristic points of the two orthogonal trapezoids in x and y .

¹in the Fuzzy Set Theory a normalized possibility distribution is a distribution which contains at least a preference degree equal to 1, that is fully plausible.

Definition 11. a well-formed trapezoid T is a 4-ple $\langle a, b, c, d \rangle$ where $a, b \in \mathbb{R} \cup \{-\infty\}$, $c, d \in \mathbb{R} \cup \{+\infty\}$, \triangleleft is either (or [and \triangleright is either) or]. A trapezoidal distribution T is allowed if and only if it satisfies the following conditions:

- $a \leq b \leq c \leq d$
- if $a = -\infty$ then $b = -\infty \wedge \triangleleft$ is (
- if $a < b$ then \triangleleft is (
- if $a = d$ then \triangleleft is [$\wedge \triangleright$ is]
- if $d = +\infty$ then $c = +\infty \wedge \triangleright$ is)
- if $c < d$ then \triangleright is)

Definition 12. the set of well-formed pyramidal distributions is denoted by \mathcal{P} .

A metric constraint C_{ij} , is a disjunction of pyramidal distributions:

Definition 13. A metric constraint C_{ij} is a set of pyramidal distributions

$$C_{ij} = \{P_1 \cdots P_n\}$$

where $P_k = \langle T_{x,k}, T_{y,k} \rangle[\alpha_k]$.

The semantics of a constraint C_{ij} is identified by the possibility distribution

$$[R_P(C_{ij})](x, y) = \max_{k=1 \cdots n} [R_P(P_k)](x, y)$$

corresponding to the disjunction of pyramidal distributions $R_P(P_k) : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ and

$$[R_{P_k}(C_{ij})](x, y) = \min\{[R_{T_{x,k}}(C_{ij})](x), [R_{T_{y,k}}(C_{ij})](y)\}$$

where

$$[R_{T_{z,k}}(C_{ij})](z) = \begin{cases} 0 & \text{if } z < a_{z,k} \vee \\ & (z = a_{z,k} \wedge \triangleleft \text{ is } () \vee \\ & (z = d_{z,k} \wedge \triangleright \text{ is })) \vee \\ & z > d_{z,k} \\ \alpha_{z,k} \left(\frac{z - a_{z,k}}{b_{z,k} - a_{z,k}} \right) & \text{if } a_{z,k} < z < b_{z,k} \\ \alpha_{z,k} \left(\frac{z - d_{z,k}}{c_{z,k} - d_{z,k}} \right) & \text{if } c_{z,k} < z < d_{z,k} \\ \alpha_{z,k} & \text{otherwise} \end{cases}$$

Example 7. as an example of metric constraint, on the right of Figure 6 a region with an undefined boundary is represented and, on the left, a possible corresponding fuzzy constraint. The fuzzy constraint could be expressed, in relative coordinates, as

$$\{\langle (0, 5, 10, 15), (0, 3, 8, 11) \rangle[0.7], \langle (7, 10, 13, 16), (5, 8, 11, 14) \rangle[1.0]\}$$

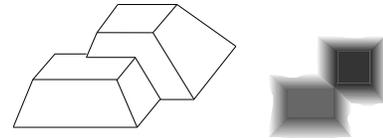


Figure 6: Example of spatial metric constraint.

The height of the pyramidal distribution is not necessarily normalized to 1, and this allows reasoning about

preferences, truth of imprecise events, priorities and so on (Dubois, Fargier, and Prade 1996). For instance, the user can set the possibility degrees according to his own preferences using non-normalized distributions to indicate partial inconsistency of constraints coming from unreliable information sources.

Operations between metric constraints

All operations between pyramidal possibility distributions involve pairs of trapezoids and are applied independently on the orthogonal projections. The usual operations are provided:

Definition 14. given a metric constraint $C_{ij} = \{P_1, \dots, P_m\}$ between variables v_i and v_j , each disjunct of the inverse constraint C_{ij}^{-1} is defined as

$$P_k^{-1} = \langle \langle x - d_{x,k}, -c_{x,k}, -b_{x,k}, -a_{x,k} \rangle x, \langle y - d_{y,k}, -c_{y,k}, -b_{y,k}, -a_{y,k} \rangle y \rangle [\alpha_k]$$

Definition 15. given two metric constraints $C_{ij} = \{P_1, \dots, P_m\}$ between variables v_i and v_j and $C'_{jw} = \{P'_1, \dots, P'_n\}$ between variables v_j and v_w , the constraint $C_{ij} \circ C'_{jw} = \bigcup_h P''_h$ is such that for any two disjuncts $P_k = \langle T_{x,k}, T_{y,k} \rangle [\alpha_k] \in C_{ij}$ and $P'_l = \langle T'_{x,l}, T'_{y,l} \rangle [\alpha_l] \in C'_{jw}$

$$P''_h = \langle T_{x,k} \circ T'_{x,l}, T_{y,k} \circ T'_{y,l} \rangle [\min\{\alpha_k, \alpha_l\}]$$

where composition between trapezoidal distributions is defined as in (Badaloni, Falda, and Giacomini 2004).

The disjunctive and the conjunctive combinations correspond to the usual set-theoretic operations and can be obtained by reasoning about the orthogonal projections, which are both trapezoids.

Definition 16. given two metric constraints $C_{ij} = \{P_1, \dots, P_m\}$ and $C'_{ij} = \{P'_1, \dots, P'_n\}$ between variables v_i and v_j , the constraint $C_{ij} \oplus C'_{ij} = \bigcup_h P''_h$ is such that for any two disjuncts $P_k = \langle T_{x,k}, T_{y,k} \rangle [\alpha_k] \in C_{ij}$ and $P'_l = \langle T'_{x,l}, T'_{y,l} \rangle [\alpha_l] \in C'_{ij}$

$$P''_h = \langle T_{x,k} \oplus T'_{x,l}, T_{y,k} \oplus T'_{y,l} \rangle [\max\{\alpha_k, \alpha_l\}]$$

Definition 17. given two metric constraints $C_{ij} = \{P_1, \dots, P_m\}$ and $C'_{ij} = \{P'_1, \dots, P'_n\}$ between variables v_i and v_j , the constraint $C_{ij} \otimes C'_{ij} = \bigcup_h P''_h$ is such that for any two disjuncts $P_k = \langle T_{x,k}, T_{y,k} \rangle [\alpha_k] \in C_{ij}$ and $P'_l = \langle T'_{x,l}, T'_{y,l} \rangle [\alpha_l] \in C'_{ij}$

$$P''_h = \langle T_{x,k} \otimes T'_{x,l}, T_{y,k} \otimes T'_{y,l} \rangle [\min\{\alpha_k, \alpha_l\}]$$

Finally, the composition operations is defined as:

Definition 18. given two metric constraints $C_{ij} = \{P_1, \dots, P_m\}$ and $C'_{ij} = \{P'_1, \dots, P'_n\}$ between variables v_i and v_j , the constraint $C_{ij} \circ C'_{ij} = \bigcup_h P''_h$ is such that for any two disjuncts $P_k = \langle T_{x,k}, T_{y,k} \rangle [\alpha_k] \in C_{ij}$ and $P'_l = \langle T'_{x,l}, T'_{y,l} \rangle [\alpha_l] \in C'_{ij}$

$$P''_h = \langle T_{x,k} \circ T'_{x,l}, T_{y,k} \circ T'_{y,l} \rangle [\min\{\alpha_k, \alpha_l\}]$$

Qualitative and metric constraints

In (Condotta 2000) Condotta proposed to build two constraint networks, one for qualitative constraints and the other for metric constraints. In this paper the idea used by Meiri to integrate temporal constraints (Meiri 1996) is adopted: a single network for both types of constraints.

Relations between Points and Regions

The first step to integrate metric and qualitative information is to define an algebra that includes all the combinations that can occur between a point and a (rectangular) region. There is therefore the need to find relations that link points with regions. An intuitive way to do this is to extend in two dimensions the Point-Interval relations for Temporal Reasoning. Being 5 the atomic mono-dimensional relations between a point and an interval, in the spatial case there will be $2^5 = 25$ atomic relations; they are represented in Figure 7. In this paper the mono-dimensional relations coming from the projections of a spatial relation on an orthogonal axis will be named in a different way w.r.t. Meiri's Point-Interval relations:

Definition 19. the set of atomic Point Region relations is defined on the set $\mathcal{PR} = \{E^-, T^-, I, T^+, E^+\}$.

Flexibility and uncertainty can be introduced in PR relations by assigning to every atomic relation $r_i \in \mathcal{PR}$ a degree $\alpha_i \in [0, 1]$, which tells the preference degree of the corresponding assignment among the others, and obtaining in this way a set of fuzzy Point-Region relations, or *fPR*.

Definition 20. Let a be a point and $b \in REC$, then a *fPR* constraint is defined as

$$R = \bigcup_i \{(r_{z,i}, r_{y,i})[\alpha_i]\}$$

where $r_{z,i}, z \in \{x, y\}, i = \{1, 2, 3\}$ are \mathcal{PR} relations and $\alpha_i \in [0, 1]$ are the preference degrees of $r_{z,i}$. Each disjunct $(r_{x,i}, r_{y,i})[\alpha_i]$ is an atomic *fPR* relation.

Saturated *fPR* relations can be defined as follows.

Definition 21. a saturated *fPR* relation can be written as

$$a \langle R_x, R_y \rangle b$$

where $R_x = (r_{x,1}[\alpha_{x,1}], \dots, r_{x,5}[\alpha_{x,5}])$, $R_y = (r_{y,1}[\alpha_{y,1}], \dots, r_{y,5}[\alpha_{y,5}])$, $r_{z,i}, z \in \{x, y\}, i = 1, \dots, 5$ are the 5 atomic Point-Region relations in \mathcal{PR} , and $\alpha_{z,i}$ is the preference degree of $r_{z,i}$. It is equivalent to the explicit relation

$$\begin{aligned} & a \{ (r_{x,1}, r_{y,1})[\min(\alpha_{x,1}, \alpha_{y,1})], \\ & \dots, (r_{x,1}, r_{y,5})[\min(\alpha_{x,1}, \alpha_{y,5})], \\ & \dots, (r_{x,5}, r_{y,5})[\min(\alpha_{x,5}, \alpha_{y,5})] \} b \end{aligned}$$

The notation $aR_x^2b = a \langle R_x, R_x \rangle b$ will be used if the two components of the pair are the same.

Also *fPR* relations can be interpreted in a more natural way; besides the standard names of the (combined) cardinal directions, 16 additional relations have been added; they have been named as in Table 3 and Figure 8.

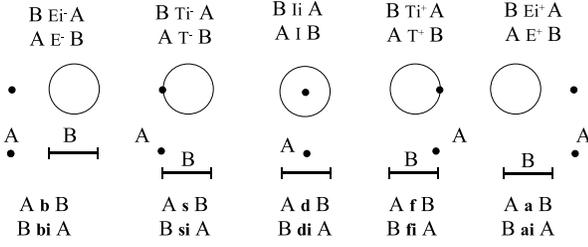


Figure 7: Point-Region spatial relations.

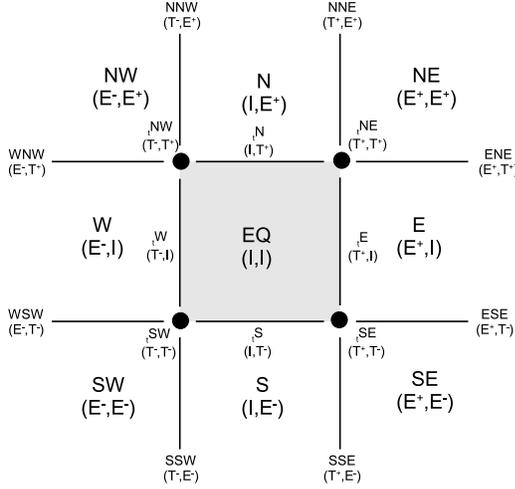


Figure 8: Point-Region spatial relations.

The Fuzzy Spatial Qualitative Algebra ($fSQA$)

Once the fuzzy Point-Region relations have been defined, an algebra that encloses all the fuzzy relations between Points and Regions can be defined; it will be called Fuzzy Spatial Qualitative Algebra or $fSQA$.

Definition 22. the Fuzzy Spatial Qualitative Algebra $fSQA$ is given by

$$fRA \cup fCDA \cup fPR$$

where fRA is the fuzzy Rectangle Algebra, $fCDA$ the fuzzy Cardinal Directions Algebra and fPR is the fuzzy Point Region set.

The $fSQA$ algebra is closed under the inversion, intersection and composition operations; inversion and intersection for \mathcal{A}_{rec} and $fCDA$ relations concern operands coming from the same algebra, and these have already been defined before. As far as fPR relations are involved, inverse fPR relations are denoted by adding a suffix “i” to the corresponding relations in \mathcal{PR} (for example, given a region a and a point b if $a \text{ }_tNE b$ then $b \text{ }_tNE_i a$), while disjunctive and conjunctive combination operations are defined as follows.

Definition 23. given two fPR relations $R = \bigcup_i \{(r_{x,i}, r_{y,i})[\alpha_i]\}$ and $S = \bigcup_j \{(s_{x,j}, s_{y,j})[\beta_j]\}$

Table 3: relations for fPR interpretation.

Pair	relation	Pair	relation
(I, I)	EQ	(T^-, E^-)	SSW
(E^+, I)	E	(T^+, E^-)	SSE
(E^+, E^+)	NE	(E^+, T^-)	ESE
(I, E^+)	N	(E^+, T^+)	ENE
(E^-, E^+)	NW	(T^+, I)	${}_tE$
(E^-, I)	W	(T^+, T^+)	${}_tNE$
(E^-, E^-)	SW	(I, T^+)	${}_tN$
(I, E^-)	S	(T^-, T^+)	${}_tNW$
(E^+, E^-)	SE	(T^-, I)	${}_tW$
(T^+, E^+)	NNE	(T^-, T^-)	${}_tSW$
(T^-, E^+)	NNW	(I, T^-)	${}_tS$
(E^-, T^+)	WNW	(T^+, T^-)	${}_tSE$
(E^-, T^-)	WSW		

the disjunctive combination between R and S is defined as

$$R \oplus S = \bigcup_i \{(r_{x,i}, r_{y,i})[\gamma_i] : (r_{x,i}, r_{y,i})[\alpha_i] \in R \wedge (s_{x,j}, s_{y,j})[\beta_j] \in S \wedge r_{x,i} = s_{x,j} \wedge r_{y,i} = s_{y,j}, \gamma = \max(\alpha, \beta)\}$$

Definition 24. given two fPR relations $R = \bigcup_i \{(r_{x,i}, r_{y,i})[\alpha_i]\}$ and $S = \bigcup_j \{(s_{x,j}, s_{y,j})[\beta_j]\}$ the conjunctive combination between R and S is defined as

$$R \otimes S = \bigcup_i \{(r_{x,i}, r_{y,i})[\gamma_i] : (r_{x,i}, r_{y,i})[\alpha_i] \in R \wedge (s_{x,j}, s_{y,j})[\beta_j] \in S \wedge r_{x,i} = s_{x,j} \wedge r_{y,i} = s_{y,j}, \gamma = \min(\alpha, \beta)\}$$

The composition operation may involve operands belonging to different algebras, and therefore it is defined in terms of a combined composition table which takes into account all possible combinations between a point and a region; preference degrees are again obtained by means of a “max-min” weighting. Table 4 shows all these combinations; the symbol “ \emptyset ” denotes illegal combinations.

Table 4: transitivity table of SQA

	CDA	PR	RP	RA
CDA	T_{CDA}	T_1	\emptyset	\emptyset
PR	\emptyset	\emptyset	T_2	T_4
RP	$(T_1)^T$	T_3	\emptyset	\emptyset
RA	\emptyset	\emptyset	$(T_4)^T$	T_{RA}

Table T_{CDA} is the transitivity table of CDA Algebra (Frank 1996). T_{RA} is the transitivity table of the Rectangle Algebra (Balbiani, Condotta, and Fariñas del Cerro 1998), which can be replaced by a double look-up in the IA transitivity table (Allen 1983), one for each of the two orthogonal components of an RA relation.

The remaining tables T_1, \dots, T_4 are analogous to those proposed by Meiri (Meiri 1996) considering the correspon-

dences between temporal and spatial relations discussed before (see Figures 1 and 7) but have not been reported here for space limits.

Definition 25. given two $fSQA$ relations $R = \bigcup_i \{(r_{x,i}, r_{y,i})[\alpha_i]\}$ and $S = \bigcup_j \{(s_{x,j}, s_{y,j})[\beta_j]\}$ the composition between R and S is defined as

$$R \circ S = \bigoplus (t_{x,i}, t_{y,i})[\min(\alpha_h, \beta_k)]$$

$$h, k : (r_{x,h}, r_{y,h}) \circ (s_{x,k}, s_{y,k}) = (t_{x,i}, t_{y,i})$$

where the composition between atomic relations $(r_{x,j}, r_{y,j})$ and $(s_{x,k}, s_{y,k})$ is given by Table 4.

Transformation functions

Having defined the possibility distributions of the metric constraints as a combination of two trapezoidal distributions along two orthogonal axes and having said that (fuzzy) CDA relations are formed by PA relations along orthogonal axes, the transformation functions introduced originally by Meiri (Meiri 1996) and extended then by (Badaloni, Falda, and Giacomini 2004) in order to be applied to trapezoidal distributions can be easily defined.

More specifically a (qualitative) $fCDA$ constraint can be transformed in a metric constraint by applying the $QUAN^{fuz}$ function to both its components (which are PA^{fuz} relations)

Definition 26. given a $fCDA$ constraint $R = \bigcup_i \{(r_{x,i}, r_{y,i})[\alpha_i]\}$ the function $fQUAN2(R) : fCDA \rightarrow \mathcal{P}$ is defined as

$$\langle \bigoplus_i \{QUAN^{fuz}(r_{x,i})\}, \bigoplus_i \{QUAN^{fuz}(r_{y,i})\} \rangle [\alpha_i]$$

Example 8. The $fCDA$ constraint $R = \{(<, <)[0.5], (<, =)[0.3]\}$ becomes a pyramidal distribution $fQUAN2(R) = \{((0, 0, +\infty, +\infty), (0, 0, +\infty, +\infty))[0.5], ((0, 0, +\infty, +\infty), [0, 0, 0, 0])[0.3]\}$.

On the other hand, if $C = \bigcup_i \langle T_{i,x}, T_{i,y} \rangle [\alpha_i]$ is a metric constraint then it can be transformed in a qualitative constraint by applying the $QUAL^{fuz}$ function to both components $T_{i,x}$ and $T_{i,y}$ of each disjunct (which are trapezoids).

Definition 27. the function $fQUAL2(R) : \mathcal{P} \rightarrow fSQA$ is defined as

$$fQUAL2(R) = \langle \bigoplus_i \{QUAL^{fuz}(T_{i,x}[\alpha_i])\}, \bigoplus_i \{QUAL^{fuz}(T_{i,y}[\alpha_i])\} \rangle$$

The resulting qualitative constraint is saturated because metric constraints are defined as a Cartesian product of two orthogonal trapezoidal distributions.

Example 9. The metric constraint $C = \{((-5, 5, 10, 15), (0, 3, 6, 9))[0.5]\}$ becomes a saturated $fCDA$ constraint $fQUAL2(C) = \{(< [0.5], = [0.25], > [0.25]), (< [0.5])\}$.

Notice that the explicit representation of different kinds of extremes, open, closed or unbounded, is essential to define such transformation functions, since they require generalized trapezoids.

By means of the concepts introduced above, spatial networks whose variables can represent both points and rectangular regions, and whose edges are accordingly labelled by qualitative and quantitative fuzzy spatial constraints can be modelled.

In particular, as in (Badaloni, Falda, and Giacomini 2004), Point-Point metric constraints are maintained in a numerical form as long as possible, while Region-Region and Point-Region constraints are necessarily qualitative, that is they are modelled as fRA and fPR relations, respectively.

On the basis of these considerations it is possible to define the operations involving all kinds of constraints introduced so far. Since metric constraints can be defined only between points, the definitions of the operations between qualitative and metric constraints can be limited, without loss of generality, to the following cases:

Definition 28. given a metric constraint C_{ij} and a qualitative constraint C'_{ij} between variables v_i and v_j their disjunctive combination is

$$C_{ij} \oplus C'_{ij} = C_{ij} \otimes fQUAN2(C'_{ij})$$

Definition 29. given a metric constraint C_{ij} and a qualitative constraint C'_{ij} between variables v_i and v_j their conjunctive combination is

$$C_{ij} \otimes C'_{ij} = C_{ij} \otimes fQUAN2(C'_{ij})$$

Definition 30. given a metric constraint C_{ij} between variables v_i and v_j and a qualitative constraint $C'_{jk} \notin fPR$ between variables v_j and v_k their metric composition is

$$C_{ij} \circ C'_{jk} = C_{ij} \circ fQUAN2(C'_{jk})$$

Definition 31. given a metric constraint C_{ij} between variables v_i and v_j and a qualitative constraint $C'_{jk} \in fPR$ between variables v_j and v_k their qualitative composition is

$$C_{ij} \circ C'_{jk} = fQUAL2(C_{ij}) \circ C'_{jk}$$

In this last case since only $fCDA$ relations can be transformed in metric constraints the operation must be performed in a qualitative way. By means of these operations, an integrated qualitative-metric Fuzzy Spatial Constraint Network $\mathcal{N} = (V, E)$ can be defined: V is a set of points and regions and E is a set of qualitative and metric fuzzy spatial constraints between them.

Reasoning about space

Algorithms and complexity

Given an qualitative-metric Fuzzy Spatial Constraint Network (FSCN), the most interesting reasoning tasks are finding an optimal solution, determining the degree of consistency and finding the minimal network; the latter is the most

difficult, because it requires to find all solutions; in this work only the first two have been considered. Since the problem of determining the consistency of a classical RA network is \mathcal{NP} -Complete (Balbiani, Condotta, and Fariñas del Cerro 1999) and RA is a particular instance of a FSCN where preferences degrees are all zero or 1 and variables are regions, the consistency of a FSCN is, in general, a \mathcal{NP} -Complete problem.

An algorithm that can be applied to find consistency is Path-Consistency (PC), which is polynomial. This method can be applied to any relations algebra provided that it is closed under the operations of inversion, intersection and composition, and that the composition is not *weak*. The composition in *fSQA* is a true composition: RA has a true composition, CDA and PR can be viewed as subsets of RA where one or more ontological entities are collapsed to a point. Unfortunately, however, in the general case Path-Consistency is a sound but not complete method for finding consistency, therefore the solution space has to be searched in a systematic way using backtracking. It is possible to prune the search space exploiting heuristic techniques guided by preference degrees, obtaining in this way a Branch & Bound algorithm; for example all instantiations that give preference degrees lower than the current partial solution can be discarded, and PC method can be applied at each step of the Branch & Bound algorithm to exclude inconsistent partial instantiations. In (Balbiani, Condotta, and Fariñas del Cerro 1999) a tractable subclass of RA is defined and in (Condotta 2000) the same subclass augmented with STP metric constraints is proved to be still polynomial: if enough expressive they could be the basis for finding a tractable subclass of the new fuzzy framework proposed here.

Application example

In order to show the expressiveness of the integrated system a small scenario is presented. In a touristic town, whose map is in Figure 9, a new hotel has to be built and there are some constraints on the possible locations. There are two allowed areas (marked with the number 7), the first, more attractive, near the sea, the second at the end of the valley that closes the town.

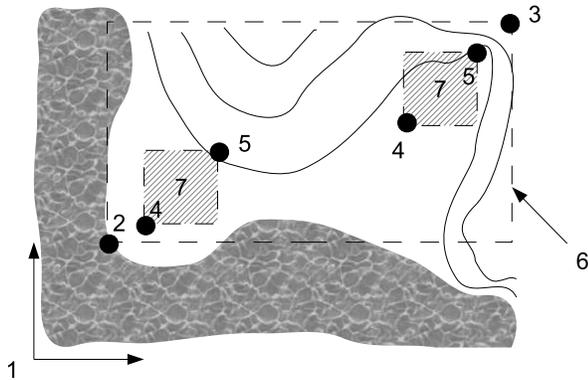


Figure 9: map.

Starting from information given, six significant coordinates plus an origin can be identified and therefore the problem can be modelled in a graph with seven vertices:

1. the origin for the relative coordinates;
2. the lower left limit of the town;
3. the upper right limit of the town w.r.t. point 2;
4. the lower left limit of the hotel area;
5. the upper right limit of the hotel area w.r.t point 4;
6. the town;
7. the hotel area.

The constraints that compose the scenario can be derived from the map of the scenario and from deductions coming from background knowledge as, for example, the first one:

1. “the hotel must be inside the town” (implicit):

$$6\langle\langle NTPP, NTPPi, O^+, O^-, TPP^-, TPPi^-, TPP^+, TPPi^+, EQ \rangle^2 \rangle 7$$

2. “the lower left coordinates of the town w.r.t. point 1 are”:

$$1\{\{(2.5, 3, 3.5, 4), (2.5, 3, 3.5, 4)\}\} 2$$

3. “the upper right coordinates of the town w.r.t. point 2 are”:

$$2\{\{(6.5, 7, 7.5, 8), (9.5, 10, 10.5, 11)\}\} 3$$

4. “the lower left coordinates of the two hotel areas w.r.t. point 1 are”:

$$1\{\{(3.25, 3.5, 3.5, 3.75), (1.75, 2, 2, 2.25)\}, \langle\langle 6.25, 6.5, 6.5, 6.75 \rangle, (10.75, 10, 10, 10.25) \rangle\} 4$$

5. “the upper right coordinates of the hotel area w.r.t. point 4 are”:

$$4\{\{[2, 2, 2, 2], [2, 2, 2, 2]\}\} 5$$

To express the fact that “the (hotel) area near the sea is more attractive”: the preference degree for the second disjunct of the constraint between points 1 and 4 has been lowered.

The resulting constraint graph is depicted in Figure 10.

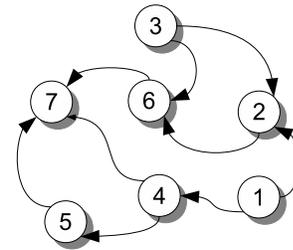


Figure 10: graph for the example.

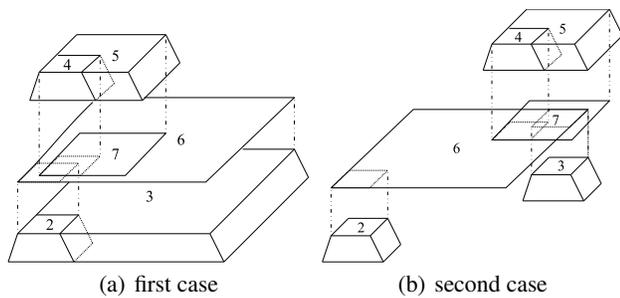


Figure 11: solutions.

Solving the problem

The solutions of the FSCN problem modelled so far can be obtained applying a Branch & Bound algorithm, as said in the previous subsection. A first solution is represented in Figure 11a; this solution tells that the hotel can be built near the sea.

Suppose now that the following additional condition holds:

“The hotel cannot be built near the sea because of environmental restrictions”

In this case the first choice in the constraint between vertices 1 and 4 must be deleted. The new solution has now a degree of consistency 0.7, due to the degree of the metric relation between nodes 1 and 4 that has been chosen. The hotel in this case can still be built, but only near the end of the valley, as depicted in Figure 11; the consistency degree of the solution is lowered with respect to the previous solution.

Conclusions

A general constraints satisfaction framework for spatial reasoning able to manage fuzzy spatial constraints involving qualitative points, rectangular qualitative regions and metric points has been presented.

Rectangle Algebra and Cardinal Direction Algebra have been extended with the Fuzzy Sets Theory and a new set of 25 Point-Region relations has been defined in order to build an integrated Spatial Qualitative Algebra (SQA) which involves points and regions. Metric spatial constraints can be imposed between points and are modelled using fuzzy possibility distributions, in particular pyramidal distributions. Metric and qualitative constraints are managed within a single constraint network and are transformed one into another when needed; two transformation functions have been provided for this purpose.

The study of *ad-hoc* algorithms for the fundamental reasoning tasks such as determining the consistency of the network and finding the optimal solutions have been just sketched, and there is room for enhancing them. Another interesting research direction would be the introduction of line segments or complex regions in the SQA.

References

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

Badaloni, S., and Giacomini, M. 2006. The algebra IA^{fuz} : a framework for qualitative fuzzy temporal reasoning. *Artificial Intelligence* 170(10):872–908.

Badaloni, S.; Falda, M.; and Giacomini, M. 2004. Integrating quantitative and qualitative constraints in fuzzy temporal networks. *AI Communications* 17(4):187–200.

Balbani, P.; Condotta, J. F.; and Fariñas del Cerro, L. 1998. A model for reasoning about bidimensional temporal relations. In *Proc. of KR’98*, 124–130.

Balbani, P.; Condotta, J. F.; and Fariñas del Cerro, L. 1999. A new tractable subclass of the rectangle algebra. In *Proceedings of IJCAI’99*, 442–447.

Condotta, J. F. 2000. The augmented interval and rectangle networks. In *Proc. of KR 2000*, 571–579.

Dechter, R.; Meiri, I.; and Pearl, J. 1991. Temporal constraint networks. *Artificial Intelligence* 49:61–95.

Dubois, D.; Fargier, H.; and Prade, H. 1996. Possibility theory in constraint satisfaction problems: Handling priority, preference and uncertainty. *Applied Intelligence* 6:287–309.

Frank, A. 1992. Qualitative spatial reasoning about distance and directions in geographic space. *Journal of Visual Languages and Computing* 3:343–373.

Frank, A. 1996. Qualitative spatial reasoning: Cardinal directions as an example. *Int. Journal of Geographical Information Science* 10(3):269–290.

Godo, L., and Vila, L. 2001. Possibilistic temporal reasoning based on fuzzy temporal constraints. In *Proc. of IJCAI95*, 1916–1922.

Güsgen, H. W.; Hertzberg, J.; and Philpott, A. 1994. Towards implementing fuzzy Allen relations. In *Proc. of ECAI-94 workshop on Spatial and Temporal Reasoning*, 49–55.

Kautz, H., and Ladkin, P. 1991. Integrating metric and qualitative temporal reasoning. In *Proc. of the AAI-91*, 241–246.

Marín, R.; Cárdenas, M. A.; Balsa, M.; and Sanchez, J. L. 1997. Obtaining solutions in fuzzy constraint network. *International Journal of Approx. Reasoning* 16:261–288.

Meiri, I. 1996. Combining qualitative and quantitative constraints in temporal reasoning. *Artificial Intelligence* 87:343–385.

Randell, D.; Cui, Z.; and Cohn, A. 1992. A spatial logic based on regions and connections. In *3rd International Conference on Knowledge Representation and Reasoning*, 165–176. Morgan Kaufmann.

Renz, J., and Ligozat, G. 2005. Weak composition for qualitative spatial and temporal reasoning. In *CP 2005*, 534–548.

Vilain, M.; Kautz, H.; and van Beek, P. 1989. Constraint propagation algorithms for temporal reasoning: a revised report. In D. S. Weld, J. d. K., ed., *Readings in Qualitative Reasoning about Physical Systems*, 373–381. San Mateo, CA: Morgan Kaufmann.

Reasoning about a Temporal Scenario in Natural Language

Benjamin Han

IBM Watson Research Center
1101 Kitchawan Road, Yorktown Heights, NY 10598, U.S.A.
dbhan@us.ibm.com

Abstract

Linguistically the temporal information of an event is often introduced in an incremental and incomplete fashion, and understanding a complete temporal scenario requires both a flexible event-level representation and a global model capable of capturing the interactions among them. In this paper we describe a method of constructing a Dependency Simple Temporal Problem with Mixed Granularities (DGSTP) from a set of event-level representations. The constraint network can then be solved to obtain a set of possible times for the events and to discover implicit temporal relationships among them.

Introduction

The capability to deduce the temporal location of an event described linguistically can benefit many real-world applications such as question answering, text summarization and intelligence analysis. Like many phenomena in natural language, however, temporal information about events are usually given throughout a discourse in a piecemeal fashion, often incomplete. For example, consider the following sentence in a news story published on Aug 16, 2006:

*Karr admitted to being involved in the death of the **6-year-old** beauty pageant winner.*

Lacking any additional information, one might assume that the death had occurred in the same year as the publication and be tempted to conclude that the victim was born in year 2000. But if the reader is given another sentence from the same story:

*Authorities are examining John Mark Karr's writings for clues that might link him to the death of JonBenet Ramsey **10 years ago**.*

It is then possible to conclude that the victim was born in 1990, assuming that the two *death* events described are identical. These observations can be summarized using the following formulae:

$$t_1 := \{t_0 + |6_{\text{year}}|\} \quad (1)$$

$$t_2 := \{\{2006_{\text{year}}, \text{aug}, 16_{\text{day}}\} - |10_{\text{year}}|, = t_1\} \quad (2)$$

Copyright © 2009, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Each formula encodes the *local* temporal information of a death event: (1) defines temporal variable t_1 to be a time point 6 years after the birth of the victim (t_0), and (2) defines variable t_2 to be a point 10 years before the publication date. The assumption that the two events are identical is then explicated as a conjunctive constraint in the second half of (2) ($= t_1$) - this assumption can come from an automatic event coreference system, or simply come from a human analyst performing a what-if experiment.

In the above we have captured the *global* information of a temporal scenario in a set of time formulae, and all there is left is a way to systematically solve for the variables. In this paper we will describe a method for solving such formulae by constructing a temporal constraint problem called *Dependency Simple Temporal Problem with Mixed Granularities* (DGSTP), where temporal relations among events are encoded as temporal constraints among the temporal variables. The resulting constraint network can then be solved to determine its consistency, to obtain a set of possible times for the variables, and to discover other valid temporal relationships among the variables.

In the next section we will first make a brief introduction on our event-level temporal representation called TCNL. We will then describe in the following three sections a progression of three classes of temporal constraint problems: Temporal Constraint Satisfaction Problems (and in particular its subclass STP), GSTP as an extension to STP with mixed granularities, and the further extension DGSTP. In particular the core solution procedures are described in the section *Modeling Temporal Scenarios*, and the methods for translating TCNL formulae into a DGSTP, our tool for modeling a temporal scenario, are discussed in the section *From Formulae to Dependency GSTP*. Finally we conclude the paper and suggest future work in the final section.

Event-Level Representation

Temporal information of an event can be conveyed linguistically via verbal tenses, temporal expressions (“10 years ago”), prepositional words (“before” and “during”) and aspectual relations (“admitted to being involved” where the admission happened after the involvement). We encode this information using an arithmetic-like formalism called *Time Calculus for Natural Language* (TCNL), where temporal information is viewed as constraints to the possible times an

event can take place. In the recent years TCNL has been successfully applied to the task of normalizing temporal expressions found in emails (Han, Gates, and Levin 2006b), newswire and web texts (Florian et al. 2007). In this section we will provide a concise review of TCNL, but readers are recommended to refer to (Han, Gates, and Levin 2006a) and (Han 2008) for a more detailed description.

Calendar Models

The foundation of TCNL is a constraint-based calendar model providing a repertoire of temporal concepts for writing time formulae. There are two kinds of concepts: temporal *units* (e.g., month) and temporal *values* (e.g., feb), and each unit can take on a set of fully ordered values. The entire calendar model is therefore a constraint satisfaction problem (CSP) (Dechter 2003), with each temporal unit acting as a variable, and the modeling task involves designing constraints among a set of units (e.g., February in a non-leap year cannot have 29 days).

Temporal units in the calendar model are ordered by two relations: the *measurement* relation and the *periodicity* relation. Unit u_i is measured by u_j , written as $u_j \leq u_i$, if every value of u_i can be mapped to a set of consecutive values of u_j on a time line; e.g., month is measured by day. A unit u_j is periodic in u_i , written as $u_j \succ u_i$, if u_j is measured by u_i and iterating through the values of u_j does not advance the value of u_i ;¹ e.g., day (days of a month) is periodic in month but is not periodic in week, because iterating through all possible values of days of month surely advances the time from one week to another. These two relations play a crucial role in defining the concepts of granularity and the anchoring status of a time entity.

TCNL Formulae

Built on top of the constraint-based calendar model is a way of representing temporal semantics via *formulae*. Every TCNL formula is of one of the three possible types: *coordinates* (C), *quantities* (Q) and *enumerations* (E).² A coordinate represents a time point and is essentially a set of assignments to the temporal units of a calendar model; e.g., {fri, 13_{day}} represents the under-specified expression “Friday the 13th”.³ A quantity represents a certain number of temporal units or coordinates; e.g., |2_{day}| means “2 days” and |2_{fri}| represents “two Fridays”. Finally an enumeration represents a set of coordinates such as intervals ({wed}: {fri}) for “Wednesday to Friday” and discontinuous sets ({wed}, {fri}) for “Wednesday and Friday”. The basic idea behind a TCNL formula is to translate whatever is said in an expression into a constraint satisfaction problem in the hope of inferring more information.

¹This is a simplified definition – the complete definition involves concepts of *periods* and the *immediate* measurement relation (Han 2008)

²We use {}, |·| and [·] respectively to mark the formulae of these types.

³Subscript dow (day-of-week) is dropped for fri since there is no ambiguity.

Associated with a formula f is its *granularity*: it is the set of minimal units (under the measurement relation) appearing in f :

$$g(f) = \min(\{u|u \in f\}) \quad (3)$$

For examples $g(\{2006_{\text{year}}, \text{aug}\}) = \{\text{month}\}$ (because $\text{month} \leq \text{year}$) and $g(|2_{\text{day}}|) = \{\text{day}\}$. We also say $g(f_1) \leq g(f_2)$ if for every unit $u_i \in g(f_1)$ we can find $u_j \in g(f_2)$ such that $u_i \leq u_j$. Granularity of a coordinate can also be used to check the “anchoring” status of a coordinate. Intuitively $\{2007_{\text{year}}, \text{may}\}$ (“May 2007”) is *anchored* in the sense that it can be identified as a unique interval on a timeline, but {may} is not. This distinction is defined as follows: a coordinate c is anchored if for every $u_1 \in g(c)$ there exists a path $\langle u_n, \dots, u_1 \rangle$ such that $\pi_{u_i}(c)$ is defined, where $u_i \succ u_{i+1}$ for $i = 1 \dots (n - 1)$ and u_n is a maximal unit under the measurement relation. E.g., $c = \{2007_{\text{year}}, \text{may}\}$ is anchored because $\text{month} \in g(c)$, $\pi_{\text{month}}(c) = \text{may}$ and $\pi_{\text{year}}(c) = 2007$, $\text{month} \succ \text{year}$ and year is a maximal unit under the measurement relation.

Operators and Relations

The representational power of TCNL mostly comes from its set of infix operators and relations (see Table 1 and 2). Each of them has a set of type requirements stipulating the types of its operands; e.g. in $\{ _ + |1_{\text{day}}| \}$ (“the next day”) the left operand ‘_’ (a temporal variable representing the temporal focus) must be of type C or E, the right operand $|1_{\text{day}}|$ must be of type Q, and the entire term is of type C. Each operator also ensures the result is at the granularity of one of its operands. The operators ‘+’/‘-’ implement a granularity-sensitive arithmetic: the granularity of the left operand (op_l) will first be converted to that of the right operand (op_r) before the addition/subtraction, therefore the result is at the granularity $g(op_r)$; e.g., $\{\{2006_{\text{year}}, \text{feb}, 1_{\text{day}}\} + |2_{\text{month}}|\}$ is evaluated to $\{2006_{\text{year}}, \text{apr}\}$, with the information at day granularity eliminated. The selection operator ‘@’ picks time points from op_r based on the constraints given by op_l , therefore the result granularity is $g(op_l)$. Finally the merging operator ‘&’ merges the non-conflicting constraints from op_l to op_r (result granularity is $g(op_l)$) and the proximity operator picks the nearest time point around op_l that satisfies the constraints given in op_r (result granularity is $g(op_r)$).

Temporal Variables

Temporal variables in TCNL serve two purposes: they are used to represent contextual information and to encode interactions among formulae. There are two kinds of variables in TCNL: the pre-defined variables speech time ‘now’ (of type C) and temporal focus ‘_’ (of type C or E), and user variables (can be of type C or E). Formulae making references to only the pre-defined variables are always easy to evaluate – we just need to substitute the variables with their denotations and evaluate away⁴. On the other hand, formulae using user variables are not always straightforward; e.g.,

⁴The denotation of ‘_’ needs to be determined by an external module.

Operator	Semantics	Type requirements	Result granularity	Example
+ and -	Arithmetic	$(\mathbb{C} \mid \mathbb{E}) \times \mathbb{Q} \rightarrow \mathbb{C}$	$g(op_r)$	$\{ _ + 1_{\text{day}} \}$ (“the next day”)
@	Selection	$\mathbb{Q} \times (\mathbb{C} \mid \mathbb{E}) \rightarrow \mathbb{C}$ $(\mathbb{C} \mid \mathbb{E}) \times (\mathbb{C} \mid \mathbb{E}) \rightarrow \mathbb{E}$	$g(op_l)$	$\{ 2_{\text{sun}} @ \{ \text{now} + 0_{\text{may}} \} \}$ (“the 2nd Sunday this May”) $\{ 9_{\text{hour}} @ \{ \{ \text{now} + 0_{\text{wed}} \} : \{ \text{fri} \} \} \}$ (“9am on this Wednesday, Thursday, and Friday”)
&	Merging	$\mathbb{C} \times \mathbb{C} \rightarrow \mathbb{C}$	$g(op_l)$	$\{ \text{now} \& \{ \text{now} + 1_{\text{year}} \} \}$ (“this time next year”)
\wedge	Proximity	$(\mathbb{C} \mid \mathbb{E}) \times \mathbb{C} \rightarrow \mathbb{C}$	$g(op_r)$	$\{ \{ 1_{\text{mon}} @ \{ \text{now} + 0_{\text{sep}} \} \} \wedge \{ \text{weekend} \} \}$ (“this Labor Day weekend”)

Table 1: Operators in TCNL; op_l/op_r is the left/right operand.

Relations	Semantics	Type requirements	Examples
<, <=, =, >=, >	before, before or equal-to, equal-to, after or equal-to, and after	$\mathbb{C} \times \mathbb{C}$	$\{ < \{ 2006_{\text{year}}, \text{may} \} \}$ (“sometime before May 2006”)
b , s , d , de , f , di	before, starting, during, during/equal, finishing, and after; de :=(s or d or f)	$\mathbb{C} \times \mathbb{E}$	$\{ \mathbf{d} \{ \{ \text{now} + 0_{\text{day}} \} : 3_{\text{day}} \} \}$ (“sometime during the 3-day period starting from today”)
b , s , f , bi	before, starting at, finishing at, and after	$\mathbb{E} \times \mathbb{C}$	$\{ \mathbf{s} \text{ now} \}$ (“from now on”)
b , m , o , s , d , f , =, fi , di , si , oi , mi , bi	See (Allen 1984).	$\mathbb{E} \times \mathbb{E}$	

Table 2: Selected relations in TCNL.

$\{ t_0 + |6_{\text{year}}| \}$ is *resolvable* only when t_0 is defined with a resolvable formula $t_0 := \{ 1990_{\text{year}} \}$ (or we say when t_0 is resolvable). For an unresolvable formula the process of *variableization* kicks in to automatically introduce a variable representing the formula (e.g., $t_1 := \{ t_0 + |6_{\text{year}}| \}$), and the constraints between this variable and the others in the formula can then be extracted for constraint solving (described later).

Temporal Constraint Satisfaction

As motivated in *Introduction*, to fully understand a temporal scenario it is often insufficient to consider events individually. Instead we will capture the temporal relations among events by way of constructing a variation of Temporal Constraint Satisfaction Problems (TCSP). This section introduces the basic concepts behind TCSP and its subclass STP.

A TCSP is a particular kind of constraint satisfaction problems (Dechter, Meiri, and Pearl 1991): it contains a set of temporal variables $\{ t_1, \dots, t_n \}$ with continuous domains and a set of unary/binary constraints. A binary constraint between variable t_i and t_j must be formulated as a disjunction of allowed *time differences* between the variables: $(a_1 \leq t_j - t_i \leq b_1) \vee \dots \vee (a_k \leq t_j - t_i \leq b_k)$ (also written as a set of disjunctive intervals $\{ [a_1, b_1], \dots, [a_k, b_k] \}$ and is said to have a *scope* $\{ t_i, t_j \}$), and a unary constraint on t_i is encoded as a binary constraint between t_i and t_0 , which is an artificially introduced variable with a singleton domain $\{ 0 \}$. A tuple (a_1, \dots, a_n) is a solution to a TCSP if the assignment $(t_1 = a_1, \dots, t_n = a_n)$ violates no constraint, and a TCSP is consistent if there exists at least one solution to the problem.

Solving a TCSP is a NP-hard problem. However if no disjunction is allowed in any constraint, solving the problem - a Simple Temporal Problem (STP) - takes only polynomial

time. This is done by converting an STP to its corresponding “distance graph”: for a constraint $[a_k, b_k]$ from variable t_i to t_j , we add an edge (t_i, t_j) of distance b_k and an edge (t_j, t_i) of distance $-a_k$ to the graph. We can then run Floyd-Warshall’s all-pairs-shortest-paths algorithm on the distance graph to derive the minimal but equivalent STP (takes $O(n^3)$ time): a constraint from t_i to t_j is $[a'_k, b'_k]$ when the shortest distance from t_i to t_j (or t_j to t_i) is b'_k (or $-a'_k$). The STP is consistent if no negative cycle is detected, and a backtrack-free search can be used to assemble a solution.

Despite its no-disjunction-allowed restriction, STPs are attractive in its simplicity and efficiency. We shall therefore focus on STPs in the rest of the work, with a note that our approach is generally extensible to include the use of disjunctions.

Modeling Temporal Scenarios

STPs have many deficiencies for our purpose due to their disconnect from natural language. For one they do not accommodate temporal constraints expressed in mixed granularities (e.g., $[10, 20]_{\text{day}}$ and $[5, 10]_{\text{month}}$). Another problem is their use of the artificial “origin of time” (t_0) to transform unary constraints into binary ones: this approach is not applicable to many unary constraints often encountered in natural language, such as “variable t_2 must be a *Tuesday*.”

In this section we describe GSTP as an extension to STP that allows mixed granularities. Our formulation is based on (Bettini, Wang, and Jajodia 2002) but specifically designed to work with our event-level representation TCNL.

Formulating the GSTP

In our version of GSTP each variable can have a set of unary constraints specified in the form of a (usually unanchored)

coordinate, and the domain of the variable contains all possible *anchored* coordinates satisfying the constraints (e.g., the domain of variable t_2 in Fig. 1 contains all possible Tuesdays). Each binary constraint tc in a GSTP is colored by a temporal unit u , and we overload the granularity function in (3) to give $g(tc) = \{u\}$. The granularity of a variable t , on the other hand, is determined by its unary constraints uc (a coordinate) and TC , the set of binary constraints whose scopes include t :

$$g(t) = \text{glb}(\left(\bigcup_{tc \in TC} g(tc)\right) \cup g(uc)) \quad (4)$$

$\text{glb}(\cdot)$ is a function returning the greatest lower bound unit of the input set under the measurement relation. For example in Fig. 1 we have $g(t_3) = \text{glb}(\text{month}, \text{day}) = \{\text{day}\}$ (recall $\text{day} \leq \text{month}$).

In an STP a *qualitative* constraint such as $t_i \leq t_j$ can be represented by converting it into its *quantitative* counterpart $[0, \infty]$. In a GSTP this can be done similarly with special care taken to infer appropriate granularity for the resulting constraints: for a qualitative constraint tc whose scope is $\{t_i, t_j\}$, we add a quantitative constraint of granularity $\{u_k\}$ for every $u_k \in \min(g(t_i) \cup g(t_j))$. E.g., if $g(t_i) = \{\text{day}\}$ and $g(t_j) = \{\text{hour}\}$, we can convert $t_i \leq t_j$ into $[0, \infty]\text{hour}$, or $t_i < t_j$ into $[1, \infty]\text{hour}$. It can be easily shown that this conversion will ultimately lower every variable weakly connected by qualitative constraints into a common granularity (because of (4)), but it will not alter the granularity of any of the other variables. This allows us to use the following one-pass procedure for inferring granularity in a GSTP with both quantitative and qualitative constraints:

Procedure 1 (Granularity inference).

1. For every variable t compute $g(t)$ according to (4) if possible; if not (because t has no unary constraint and participates no quantitative constraint), assign a default granularity to $g(t)$.
2. For a set of variables T weakly connected by qualitative constraints, assign $g(t_j) = \text{glb}(\bigcup_{t_i \in T} g(t_i))$ for all $t_j \in T$.

Constraint Propagation

Using a coordinate as the implicit domain of a variable has two consequences: the domain is no longer contiguous, and the propagation processes of the unary and the binary constraints are now separate. The first consequence also implies that disjunctions are back to the GSTP, thus breaking the *decomposability* that enables a backtrack-free search for solutions (Dechter, Meiri, and Pearl 1991). Assuming a GSTP with only quantitative constraints (i.e., all qualitative constraints are already quantified), we use an approximate method similar to the one described in (Bettini, Jajodia, and Wang 2000) to propagate the binary constraints (pictorially depicted in Fig. 1):

Procedure 2 (Approximate constraint propagation).

1. We first decouple a GSTP into single-granularity STP_{u_i} where $\{u_i\}$ is a granularity.
2. We then run the all-pairs-shortest-paths algorithm on each STP_{u_i} to derive its minimized counterpart. If any negative cycle is detected, stop the algorithm and report the

inconsistency. E.g., in Fig. 1 the constraint $[30, 60]\text{day}$ in STP_{day} is produced by this step.

3. For every STP_{u_i} we convert its constraints into granularity $\{u_j\}$ and add them into STP_{u_j} , meaning intersecting a preexisting constraint between the same pair of variables with the new one. If any constraint is refined as a result, we go back to the previous step. E.g., in Fig. 1 the constraint from t_2 to t_3 in $\text{STP}_{\text{month}}$ was $[-\infty, \infty]$ before this step, and is refined to $[1, 2]\text{month}$ afterwards. We will come back to the granularity conversion of a constraint later.
4. Finally we conjoin the constraints of all possible granularities and produce a propagated GSTP.

Procedure 2 is an approximation because the granularity conversion done in the step 3, although guaranteeing no loss of solutions, could unduly enlarge the set of solutions. The advantage of this procedure is its polynomial time complexity: if we have m different granularities and n variables, the overall runtime is $O(mn^2I(m+n))$ where I is the number of the iterations. For GSTP with single granularity this reduces to $O(n^3)$, otherwise the loop can run at maximum mn^2w iterations (w is the maximum width of any constraint after the first iteration). Although in practice the procedure seldom runs longer than a few iterations, when quantifying a qualitative constraint we replace ∞ with a large number to avoid this potentially infinite runtime.

Granularity Conversion

The step 3 of Procedure 2 converts a constraint $[a, b]u_1$ into $[a', b']u_2$ where u_1 and u_2 are two different temporal units and $a \leq b, b > 0$. The conversion must satisfy one criterion: if an assignment satisfies $[a, b]u_1$, the same assignment must also satisfy $[a', b']u_2$ (i.e., no loss of solutions). E.g., in Fig. 1 the constraint $[0, 1]\text{month}$ is a valid conversion of $[10, 20]\text{day}$ since all possible assignments of t_1 and t_2 with difference between 10 to 20 days must also fall within a 0- to 1-month window. It is therefore clear that we prefer a “tighter” conversion since the target constraint can be made arbitrarily lenient to let in more assignments.

In general this conversion task can be very difficult because temporal granules do not always have fixed sizes. As an example, we could argue that in Fig. 1 $[1, 2]\text{month}$ is not the tightest conversion possible for $[20, 40]\text{day}$ from variable t_2 to t_3 , since the difference between March 1, 2006 and April 9, 2006 is clearly less than two months. Our approach is only a result of compromise: it is a constant-time operation and it requires much simpler engineering in terms of calendar modeling.

We first define two utility functions $\text{minsize}(u_i, u_j)$ and $\text{maxsize}(u_i, u_j)$ where u_i and u_j are two temporal units and $u_j \leq u_i$: the functions return the min/max number of consecutive granules of u_j that can overlap on a timeline with a granule of u_i . E.g., $\text{minsize}(\text{month}, \text{day}) = 28$ and $\text{maxsize}(\text{month}, \text{day}) = 31$. The conversion is then given

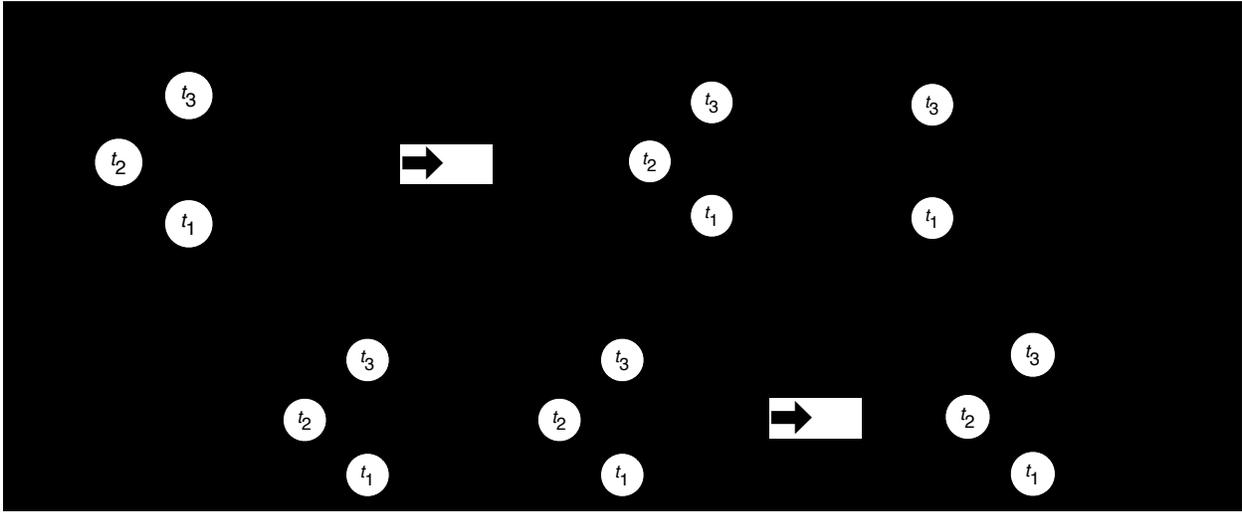


Figure 1: Approximate method for constraint propagation on a GSTP.

as follows ($u = \text{glb}(u_1, u_2)$):

$$b' = \left\lfloor \frac{(b+1) \cdot \text{maxsize}(u_1, u) - 1}{\text{minsize}(u_2, u)} \right\rfloor$$

$$a' = \left\lceil \frac{(a-1) \cdot \text{minsize}(u_1, u) + 1}{\text{maxsize}(u_2, u)} \right\rceil \quad (\text{if } a > 0)$$

Note when $a \leq 0$ we can simply swap the two bounds and treat $-a$ as an upper bound.

Assembling Solutions

After constraint propagation is done on a GSTP and no inconsistency is reported, we still need to check if any solution exists to confirm its consistency. For certain applications it is also desirable to find some or all solutions of a GSTP. For these purposes we use a simple backtracking method for assembling solutions of a propagated GSTP:

Procedure 3 (Backtracking search for solutions of a propagated GSTP).

1. For each variable t_i compute its initial domain.
2. From a list of unassigned variables pick t_i .
3. Pick the next possible anchored coordinate from the domain of t_i at its inferred granularity; if it is not possible, backtrack to the previously assigned variable as the new t_i and re-run this step; if no previously assigned variable is available, stop.
4. For each possible granularity and variable $t_j \neq t_i$, update the domain of t_j based on the constraint from t_i to t_j in that granularity. If the domain of t_j should become empty, return to 3. Note that when updating domains we use the TCNL operator '+' and invoke granularity conversion on coordinates if necessary. Continuing the example given in Fig. 1, assuming t_1 is already assigned with $\{2006_{\text{year}}, \text{feb}, 1_{\text{day}}\}$ and we want to update t_3 using the constraint $[2, 2]_{\text{month}}$, we first compute $\{t_1 + [2_{\text{month}}]\} = \{2006_{\text{year}}, \text{apr}\}$. We then convert the granularity of the

result to $\text{g}(t_3) = \{\text{day}\}$ and derive the new bounds $\{2006_{\text{year}}, \text{apr}, 1_{\text{day}}\}$ and $\{2006_{\text{year}}, \text{apr}, 30_{\text{day}}\}$.

5. t_i is now assigned; if there is no unassigned variable left, report all possible assignments as solutions, then backtrack to the previously assigned variable as the new t_i and return to 3. Otherwise return to 2.

Note that the ordering of the unassigned variables in step 2 can greatly affect the performance of the procedure. One useful ordering is to pick a more constrained variable with a smaller domain earlier in the search process. For example, in Fig. 1 we use the ordering $t_1 \rightarrow t_2 \rightarrow t_3$: after assigning the only possible coordinate $\{2006_{\text{year}}, \text{feb}, 1_{\text{day}}\}$ to t_1 we update the domain of t_2 to contain only $\{2006_{\text{year}}, \text{feb}, 14_{\text{day}}\}$ and $\{2006_{\text{year}}, \text{feb}, 21_{\text{day}}\}$ and the domain of t_3 to contain only $\{2006_{\text{year}}, \text{apr}, 1_{\text{day}}\}$ and $\{2006_{\text{year}}, \text{apr}, 2_{\text{day}}\}$. Later iterations will eliminate $\{2006_{\text{year}}, \text{feb}, 14_{\text{day}}\}$ from the domain of t_2 and give us two solutions in total.

From Formulae to Dependency GSTP

We are now left with the final task of translating a set of TCNL formulae into a GSTP. Naturally this translation needs to deal with the various syntactic and semantic devices provided by TCNL. An immediate complication is that several of the TCNL operators – such as the proximity operator '@' – have semantics not expressible in the form of a time-difference constraint. We will propose an extension *Dependency GSTP* (DGSTP) to address this problem.

At a higher level, since our GSTP extension only allows variables with coordinate domains to be present, while TCNL allows variables to be of type \mathbb{C} or \mathbb{E} , we need to re-interpret a constraint to eliminate any possible variable of type \mathbb{E} in its scope. A corollary is that we need to infer variable types first - this is our next topic below.

Variables and Their Types

Variables in a TCNL formula can be of type \mathbb{C} or \mathbb{E} . If a variable is defined explicitly, it must have the same type as its definition (e.g., in $t := \{t_0 + |6_{\text{year}}|\}$ we have $\text{type}(t) = \mathbb{C}$). Otherwise its type can be inferred from the contexts as follows. For each context the variable is in, if it is used as an operand to an operator/relation, the context allows the types compatible with the requirements of the operator/relation (see Table 1 and 2), otherwise we assume the context allows both types. After considering all of the contexts, the variable is assigned the type that is compatible to all: if both \mathbb{C} and \mathbb{E} are compatible, \mathbb{C} is picked, but if no compatible type can be found, a type mismatch is detected and no more processing is attempted. E.g., in $\{\{15_{\text{hour}}\} @ t\}$ we have $\text{type}(t) = \mathbb{C}$, but in formulae $\{t, 15_{\text{hour}}\}$ and $[d t]$ we assign $\text{type}(t) = \mathbb{E}$.

A slight complication for determining $\text{type}(t)$ arises when variable t is involved in relation ‘=’ (both a $\mathbb{C} \times \mathbb{C}$ and an $\mathbb{E} \times \mathbb{E}$ relation): if we have $t = t'$ and $\text{type}(t) = \mathbb{E}$, we will assign $\text{type}(t') = \mathbb{E}$ as well. This “type propagation” can be easily done over the closure of the ‘=’ relation.

Having decided types for variables, for every variable t of type \mathbb{E} we then create two *bound variables* t^-/t^+ (of type \mathbb{C}) to represent its lower/upper bound, and we also add a constraint $t^- \leq t^+$ to relate the two. Our goal later in the section *Constraint Re-interpretation* will be replacing all occurrences of t (of type \mathbb{E}) in constraints with its bound variables and re-interpreting the constraints.

Translating Coordinates

A coordinate formula in TCNL can pack a lot of information. Among the terms that can appear inside a coordinate are temporal values (e.g., $\{\text{feb}\}$), embedded coordinates (e.g., $\{\{\text{feb}\}, 1_{\text{day}}\}$), terms with operators (e.g., $\{\text{now} + |6_{\text{year}}|\}$) and relations (e.g., $\{< \{2006_{\text{year}}, \text{may}\}\}$). An unresolved variable can appear almost at any place where a coordinate/enumeration is expected. Below we will discuss each of the possibilities.

Translating a term with operator $+$ or $-$ is straightforward: if a term $t_j + q$ (or $t_j - q$) appears in the formula represented by t_i and $\text{g}(q) = \{u\}$, we add a constraint $[q^-, q^+]u$ from variable t_j to t_i (or from t_i to t_j , respectively), where q^-/q^+ is the lower/upper bound of q . E.g., for $t_1 := \{t_0 + |<= |6_{\text{year}}|\}$ we add a constraint $[0, 6]_{\text{year}}$ from t_0 to t_1 .

We call the terms containing the other operators ($@$, $\&$ and \wedge) *dependency terms* (or d-terms for short) since there are “dependencies” among the involved variables that cannot be made explicit by a time-difference constraint. As a simplification TCNL only allows resolvable formulae to be used as the left operand for the operators ‘@’ and ‘&’ and the right operand for the operator ‘ \wedge ’. From these terms we can still add useful time-difference constraints based on the semantics of the operators, thus making it possible for Procedure 2 to narrow down the domains of the involved variables. We can also *inversely* infer the possible values of an unassigned variable if some of the other variables in these terms are assigned. Consider the formula $t := \{\{2_{\text{day}}\} @ [t_1 : t_2]\}$ (the second day between t_1 and t_2 , inclusive): obviously the constraints $t_1 \leq t \leq t_2$ and $t_1 \leq t_2$

must be true, and if we know t_2 is May 2, 2006 then t_1 must be on or before May 1, 2006 and t must be on or before May 2, 2006. We will defer the discussion on the inverse inference to the section *Solving DGSTP*.

Fig. 2 shows the three allowed d-terms and their accompanied time-difference constraints: q and c are a quantity and coordinate constant respectively, and each d-term is represented by a new variable d . For all of the operators the d-terms are obviously equal to t at the respective result granularity. For the merging operator ‘&’, we constrain t' to be equal to the result of the operator at the granularity $\text{lub}(\text{g}(c) \cup \text{g}(t'))$ ($\text{lub}(\cdot)$ returns the least lower bound unit of the input set under the measurement relation) based on the intended use case of the operator; e.g., in $t := \{\{\text{now} + |0_{\text{day}}|\} \& t'\}$ for “this day in that week”, we constrain the d-term with t' via constraint $[0, 0]_{\text{week}}$ ($\text{lub}(\{\text{day}\} \cup \{\text{week}\})$). For the proximity operator \wedge we specify a “search window” by introducing the constraint $[-w, w]_{u_w}$ between the d-term and t' , because the output of such a term cannot be constrained otherwise; e.g., $\{t' \wedge \{\text{fri}, 13_{\text{day}}\}\}$ (the closest Friday the 13th relative to t') can be earlier or later than t' depending on what t' is resolved to.

Translating a term with a $\mathbb{C} \times \mathbb{C}$ relation (Table 2) is straightforward since we can always quantify qualitative constraints using Procedure 1. Translating terms with $\mathbb{C} \times \mathbb{E}$ relations is also easy since we can reduce them into a conjunctive set of $\mathbb{C} \times \mathbb{C}$ relations; e.g., $t := [d [t_1 : t_2]]$ is equivalent to $t := \{> t_1, < t_2\}$.

For terms of sole variables such as $t := \{t', \dots\}$, they have different semantics compared to $t := \{= t', \dots\}$: the former uses t' to build up t while the latter declares both variables to be equivalent. They also contribute different constraints: the former gives a quantitative constraint $[0, 0]_{u_i}$ for every $u_i \in \text{g}(t')$, but the latter gives a qualitative constraint $t = t'$, which will result in a granularity propagation that brings both $\text{g}(t)$ and $\text{g}(t')$ to a common granularity according to Procedure 1.

Translating Enumerations

Given an enumeration formula $[\dots, t', \dots]$ where t' is an unresolved term, from above we know a variable t of type \mathbb{E} will be introduced via the variabilization process together with two bound variables t^-/t^+ . If $\text{type}(t') = \mathbb{C}$, we can account for the term by introducing constraints $t^- \leq t'$ and $t' \leq t^+$; otherwise we produce constraints $t^- \leq t'^-$ and $t'^+ \leq t^+$ if $\text{type}(t') = \mathbb{E}$.

For terms that use the selection operator $@$, we will introduce constraints relating the bound variables of the host formula and those of the right operand of the operator, i.e., for $t := [c @ t', \dots]$ where c is a coordinate constant and $\text{type}(t') = \mathbb{E}$, constraints $t^- \leq t'^-$ and $t'^+ \leq t^+$ will be introduced.

Finally, terms using any relation involving type \mathbb{E} in Table 2 can easily be made to use a conjunctive set of $\mathbb{C} \times \mathbb{C}$ relations; e.g., $t_1 := [b t_2]$ (t_2 is of type \mathbb{E}) is equivalent to $t_1^+ := \{< t_2\}$.

Constraint Re-interpretation

When following the instructions given above to translate TCNL formulae into constraints, we need to make sure ev-

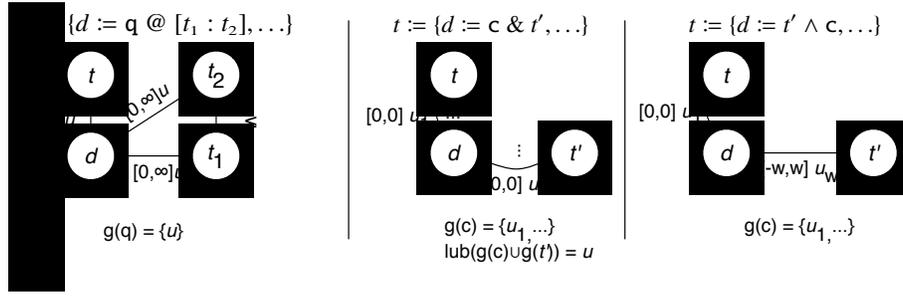


Figure 2: Converting dependency terms into constraints; $\text{lub}(\cdot)$ returns the least upper bound unit of the input set under the measurement relation.

ery variable appearing in the scope of a constraint is of type \mathbf{C} , since the final DGSTP only allows variables of type \mathbf{C} . E.g., we should translate $t := \{< t'\}$ into a constraint $t^+ < t'^-$ (instead of $t < t'$) if both variables are of type \mathbf{E} . In general, when adding a constraint with scope $\{t, t'\}$, we should *re-interpret* the constraint based on $\text{type}(t)$ and $\text{type}(t')$ so no enumeration variable can slip into the scope. We list below the re-interpretations needed when $\text{type}(t) = \text{type}(t') = \mathbf{E}$ (a and b are integers and u is a temporal unit):

$$t < t' \rightarrow t^+ < t'^- \quad (5)$$

$$t \leq t' \rightarrow t^- \leq t'^- \text{ and } t^+ \leq t'^+ \quad (6)$$

$$t = t' \rightarrow t^- = t'^- \text{ and } t^+ = t'^+ \quad (7)$$

$$t - t' \in [a, b]u \rightarrow t^- - t'^+ \in [a, b]u \quad (8)$$

Note that (5) is essentially the $\mathbf{E} \times \mathbf{E}$ relation \mathbf{b} , and (6) is equivalent to the disjunction $(\mathbf{b} ; \mathbf{m} ; \mathbf{o} ; \mathbf{s} ; \mathbf{f} ; =)$. Also, (8) in effect disallows overlapping between t and t' , and with (7) they will force $t^- = t'^- = t'^+ = t^+$ when $a = b = 0$. This might seem strange, but note that (8) is added only when a quantitative constraint is introduced between t and t' , which intuitively asserts that the two enumerations should never overlap.

If $\text{type}(t) = \mathbf{E}$ but $\text{type}(t') = \mathbf{C}$, and t' is not a bound variable, the re-interpretations are

$$t < t' \rightarrow t^+ < t'$$

$$t \leq t' \rightarrow t^+ \leq t'$$

$$t > t' \rightarrow t' < t^-$$

$$t \geq t' \rightarrow t' \leq t^-$$

$$t - t' \in [a, b]u \rightarrow t^- - t' \in [a, b]u$$

Note that $t = t'$ can never occur because of the type propagation described earlier. If the types of both variables are the same but t' is a bound variable, the re-interpretations include the above plus

$$t = t' \rightarrow t^- = t' \quad (\text{if } t' \text{ is a starting bound}) \quad (9)$$

$$t = t' \rightarrow t^+ = t' \quad (\text{if } t' \text{ is an ending bound}) \quad (10)$$

Note that (9) is equivalent to $(t' \mathbf{s} t)$ and (10) is equivalent to $(t' \mathbf{f} t)$.

Solving DGSTP

Solving a DGSTP is almost identical to solving a GSTP: we first run Procedure 2 to narrow down the domains of the

variables, we then run a revised backtracking search based on Procedure 3 to assemble the solutions. This new search method uses both propagated constraints and *d-terms* to update variable domains: if a variable is assigned in a d-term, we can inversely infer the possible values for the other. Here we will only describe how the inverse inference procedure works for the major cases in the d-terms of operator $@$ and \wedge .

Consider the d-term $d := \{[n_x]@ [t_1 : t_2]\}$ where n is an integer constant and x is a unit or a coordinate. If only d is assigned, we can infer that $\{-(n+1)_x|@ \{<= d\}\} < t_1 \leq \{-[n_x]@ \{<= d\}\}$ and $t_2 \geq d$; e.g., if d is Sunday, Jan 21, 2007 in $d := \{[2_{\text{sun}}]@ [t_1 : t_2]\}$, we should have Jan 7, 2007 $< t_1 \leq$ Jan 14, 2007 and $t_2 \geq$ Jan 21, 2007. If only t_1 is known, we can easily compute $d = \{[n_x]@ \{>= t_1\}\}$ and $t_2 \geq d$. If only t_2 is known, then we should have $d = \{-[1_x]@ \{<= t_2\}\}$ and $\{-(n+1)_x|@ \{<= d\}\} < t_1 \leq \{-[n_x]@ \{<= d\}\}$.

Consider the d-term $d := \{t \wedge c\}$ where c is a coordinate constant. If only d is known, we can find $c_1 = \{-[2_c]@ \{<= d\}\}$ and $c_2 = \{[2_c]@ \{>= d\}\}$, and compute δ_1 and δ_2 as the distance between c_1 and d and between d and c_2 , respectively. We can then infer that $\{d - \lfloor \delta_1 / 2 \rfloor_u\} \leq t \leq \{d + \lfloor \delta_2 / 2 \rfloor_u\}$ where u is the greatest lower bound of $g(c)$. E.g., if $d := \{t \wedge \{\text{sun}\}\}$ and d is Jan 21, 2007, we should have Jan 18, 2007 $\leq t \leq$ Jan 24, 2007.

We now present the revised search procedure:

Procedure 4 (Backtracking search for DGSTP). Replace the step 4 in Procedure 3 with the following:

- 4' For each possible granularity and variable $t_j \neq t_i$, update the domain of t_j based on the constraint from t_i to t_j in that granularity. Additionally, if t_i is a variable appearing in a d-term, update the domains of the other variables in the d-term using the inverse inference procedure. If any of the updated domains should become empty, return to 3.

Conclusion and Future Work

In this paper we have described a method for modeling a temporal scenario via Simple Temporal Problems with Mixed Granularities (GSTP). More specifically, we capture event-level semantics using a compact representation Time Calculus for Natural Language (TCNL) and construct a Dependency GSTP (DGSTP) from a set of TCNL formulae by: (1) creating variables for unresolved formulae and inferring the variable types; (2) translating the formulae into

constraints whose scopes contain only coordinate variables; and (3) inferring variable granularities and quantifying qualitative constraints accordingly (Procedure 1). We can then propagate the constraints of the resulting DGSTP to narrow down the variable domains (Procedure 2), and search for the solutions by using a backtracking search method (Procedure 4). Finally, qualitative relations can be discovered by inspecting the propagated domains of the relevant variables.

There are at least three parameters in our method that are open for tuning. In the granularity inference procedure (Procedure 1) a default granularity is assigned to a variable if its granularity cannot be inferred from its context. In such cases if we know the typical durations of the events associated to the variable (such information is learned in (Feng, Mulkar, and Hobbs 2006)), we could assign a more sensible granularity to it. A second parameter is the large number we use to replace ∞ when quantifying a qualitative constraint (to avoid the theoretical infinite run-time of Procedure 2). Again we might be able to set this number based on the granularities or the other contextual information of the involved variables. Similarly, contextual information might also be useful in setting the width of the search window imposed in the d -term of the proximity operator \wedge (Fig. 2). In summary, these questions can only be answered from an empirical study using real-world data.

References

- Allen, J. F. 1984. Towards a General Theory of Action and Time. *Artificial Intelligence* 23:123–154.
- Bettini, C.; Jajodia, S.; and Wang, S. X. 2000. *Time Granularities in Database, Data Mining, and Temporal Reasoning*. Berlin: Springer-Verlag.
- Bettini, C.; Wang, X.; and Jajodia, S. 2002. Solving multi-granularity temporal constraint networks. *Artificial Intelligence* 140:107–152.
- Dechter, R.; Meiri, I.; and Pearl, J. 1991. Temporal constraint networks. *Artificial Intelligence* 49:61–95.
- Dechter, R. 2003. *Constraint Processing*. Morgan Kaufmann.
- Feng, P.; Mulkar, R.; and Hobbs, J. R. 2006. Learning Event Durations from Event Descriptions. In *Proceedings of the 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL)*, 393–400.
- Florian, R.; Han, B.; Luo, X.; Kambhatla, N.; and Zitouni, I. 2007. IBM ACE’07 System Description. In *Proceedings of NIST 2007 Automatic Content Extraction Evaluation*.
- Han, B.; Gates, D.; and Levin, L. 2006a. From Language to Time: A Temporal Expression Anchorer. In *Proceedings of the 13th International Symposium on Temporal Representation and Reasoning (TIME 2006)*.
- Han, B.; Gates, D.; and Levin, L. 2006b. Understanding Temporal Expressions in Emails. In *Proceedings of Human Language Technology conference - North American chapter of the Association for Computational Linguistics annual meeting (HLT-NAACL 2006)*.
- Han, B. 2008. A constraint-based modeling of calendars.

A Study of Heuristic based Evolutionary Algorithms for Temporal Constraint Problems

Bahareh Jafari Jashmi and Malek Mouhoub

Computer Science Department, University of Regina
Regina, SK, Canada S4S 0A2
{jafarijb,mouhoubm}@cs.uregina.ca

Abstract

In this paper we discuss the applicability of evolutionary algorithms enhanced by heuristics and adaptive fitness computation for solving the Temporal Constraint Satisfaction Problem (TCSP). This latter problem is an extension of the well known CSP, through our TemPro model, in order to handle numeric and symbolic temporal information. We test the evolutionary algorithms on randomly generated TCSPs and analyze and compare the performance of the algorithms tested, based on different measures. The results show that heuristics do not promise better performance for solving TCSPs. The basic genetic algorithm (GA) and Microgenetic Iterative Descendant (MGID) are the most effective ones. We also noticed that MGID is more efficient than basic GA for easier problems.

Introduction

The Temporal Constraint Satisfaction Problem (TCSP) based on our model TemPro (Mouhoub, Charpillat, and Haton 1998; Mouhoub 2004b) is a framework used for representing and answering queries when solving numeric and symbolic temporal constraint problems. A TCSP is a particular case of a CSP where variables are temporal events defined on a set of numeric intervals and the constraints are disjunctions of Allen primitives (Allen 1983) and represent the possible temporal relations between events. The domain of events (set of intervals) can be seen as unary quantitative constraints while the disjunctive relations between the events are the binary qualitative constraints. TCSPs are applicable to a variety of scheduling (Poesio and Brachman 1991) and planning problems including manufacturing and natural language processing (Song and Cohen 1988), temporal databases and instruction optimization for compilers information.

Most of the methods that are being used for solving temporal CSPs are systematic and deterministic search algorithms in which the search space is explored in a systematic way sometimes with getting help of some heuristics. However, there have been few works for solving temporal CSPs in nonsystematic ways (Mouhoub 2004b; 2004a; Thornton et al. 2004; 2002; Beaumont et al. 2001) unlike general CSPs for which there have been many attempts in

this area. Evolutionary Algorithms (EAs) are good examples of nonsystematic searches. Evolutionary algorithms for solving CSPs are usually fallen into two categories: EAs with adaptive fitness functions and heuristic based EAs.

In this paper, we apply some well known heuristics for solving CSPs, to the TCSP case, and compare them together and with the basic genetic algorithm (GA). These heuristics try to guide the search by changing the mutation or crossover operator or by changing the way they compute the fitness throughout the search process. Although the use of heuristics can guide the search toward a better solution, it may also bias the search toward local optima which is not desirable. For this reason, in most of the heuristics we use in this paper, there is also a uniform random mechanism that helps exploring different parts of search space. From the experiments we conducted and reported in this paper, these algorithms do not necessarily perform better than basic GA. From what we conclude, basic GA and Microgenetic Iterative Descent Algorithm (MGID) seem to be the best algorithms in terms of effectivity (success rate) and efficiency (measured by the average number of conflict checks).

The paper is organized as follows. Section 2 does an overview over Tempro model. Section 3 describes the algorithms enhanced with heuristics to solve CSP problems. The algorithms with adaptive fitness are presented in section 4. Section 5 reports the experimental study we conducted on random TCSPs. Finally, the conclusion and possible future work are presented in section 6.

Tempro model

TemPro (Mouhoub 2004b) transforms a temporal problem under qualitative and quantitative constraints into a binary CSP where constraints are disjunctions of Allen primitives (see Table 1 for the definition of the Allen primitives) and variables, representing temporal events, are defined on domains of time intervals. Each event domain (called also temporal window) contains the Set of Possible Occurrences (SOPO) of numeric intervals the corresponding event can take. The SOPO is the numeric constraint of the event. It is expressed by the fourfold: [earliest_start, latest_end, duration, step] where: earliest_start is the earliest start time of the event, latest_end is the latest end time of the event, duration is the duration of the event and step is the discretization step corresponding to the number of time units between

Relation	Symbol	Inverse	Meaning
X Before Y	B	Bi	$\underline{\quad X \quad} \quad \underline{\quad Y \quad}$
X Equals Y	E	E	$\underline{\quad X \quad}$ $\underline{\quad Y \quad}$
X Meets Y	M	Mi	$\underline{\quad X \quad} \quad \underline{\quad Y \quad}$
X Overlaps Y	O	Oi	$\underline{\quad X \quad} \quad \underline{\quad Y \quad}$
X During Y	D	Di	$\underline{\quad X \quad} \quad \underline{\quad Y \quad}$
X Starts Y	S	Si	$\underline{\quad X \quad} \quad \underline{\quad Y \quad}$
X Finishes Y	F	Fi	$\underline{\quad Y \quad} \quad \underline{\quad X \quad}$

Table 1: Allen primitives.

the start time of two adjacent intervals. For some applications, the consistency of the problem depends on the discretization step. In this particular case, if the solution is not found, the user can decrease the value of the step and run again the solving algorithm. Decreasing the discretization step will however increase the complexity of the problem. Indeed, the total number of combinations (potential solutions) of a TCSP is D^N where N is the number of variables and D their domain size. D is computed as follows: $D = \text{Max}_{1 \leq i \leq N} (\frac{\text{sup}_i - \text{inf}_i - d_i}{s_i})$ where sup_i , inf_i , d_i and s_i are respectively the latest end time, earliest start time, duration and step of a given event Evt_i . As we can easily see, decreasing the value of s_i will increase the domain size D which increases the total number of possibilities of the search space. Note that begintime, endtime, duration and step can be constant values or variables taking values from a discrete and finite domain. We can also use constraints, in the form of equations or inequalities, in order to restrict the values these variables can take (Mouhoub 2004a).

For applying an evolutionary algorithm to a problem, an appropriate presentation for the potential solution should be depicted. The potential solution in our case is a complete assignment of time intervals to all events. Usually each individual is an array of genes. Since events are the variables of the problem, each gene in an individual represents an event and the value of the gene would be a time interval allotted to that event. Each interval is shown as (StartTime, EndTime), in which StartTime is the actual start time of the event and EndTime is the actual end time which is the summation of StartTime and duration of the event. StartTime of each event should be selected based on the unary constraints it is involved in. Hence, if we consider the Temporo model, domain of StartTime of event i would be $\text{earliest_start} + (X * \text{step})$ where X ranges in the interval of $[0, (\text{latest_end} - \text{earliest_start} - \text{duration}) / \text{step}]$. Now that the individuals are defined as an array, genetic operators like crossover can be applied on them the same as they are applied to other array representations. For instance, in the case of the mutation, a random possible interval for the event will be assigned to the gene representing that event.

Let us illustrate the representation of a TCSP with evolutionary algorithms through the following example.

Example 1

1. John, Mary and Wendy separately rode to the soccer game.
2. It takes John 30 minutes, Mary 20 minutes and Wendy 50 minutes to get to the soccer game.
3. John either started or arrived just as Mary started.
4. John left home between 7:00 and 7:10.
5. Mary arrived at work between 7:55 and 8:00.
6. Wendy left home between 7:00 and 7:10.
7. John's trip overlapped the soccer game.
8. Mary's trip took place during the game or else the game took place during her trip.
9. The soccer game starts at 7:30 and lasts 105 minutes.
10. John either started or arrived just as Wendy started.
11. Mary and Wendy arrived together but started at different times.

Figure 1 shows a graph corresponding to the TCSP of the above problem. A possible individual is presented here with the value of fitness equal to 3. Examples of mutation and crossover operators are respectively presented in figures 2 and 3.

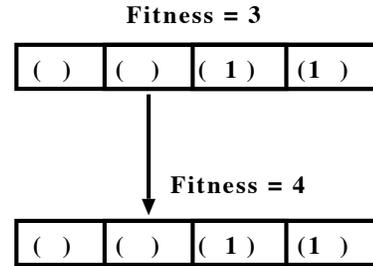


Figure 2: Mutation Operator.

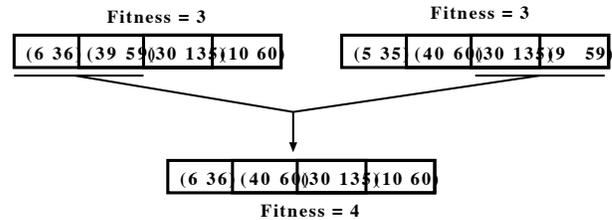


Figure 3: Crossover Operator.

Heuristic based GAs

Evolutionary algorithms can be extended by some heuristics for solving particular types of problems. Some of

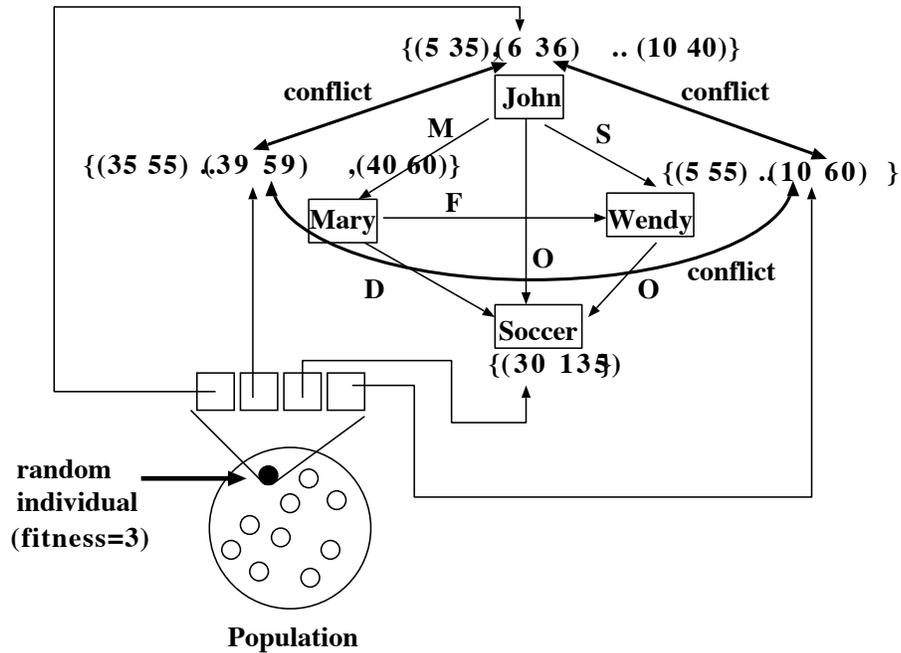


Figure 1: GA representation of a TCSP.

the known heuristics developed for the purpose of solving CSP problems are used in Heuristic Genetic algorithms proposed by Eiben et al (Eiben, RauC, and Ruttkay 1993; Eiben, Raue, and Ruttkay 1994), Knowledge Based Fitness and Genetic Operators (ARC-GA)(M.-C 1998). In this section we use some of the successful algorithms for random binary CSPs according to (Craenen and Eiben 2003) to solve Temporal CSP problems.

HGA

In (Eiben, RauC, and Ruttkay 1993) and (Eiben, Raue, and Ruttkay 1994), Eiben et al. propose to incorporate existing CSP heuristics into genetic operators. Two heuristic based genetic operators are specified: an asexual operator that transforms one individual into a new one and a multiparent operator that generates one offspring using two or more parents.(Craenen, A.E.Eiben, and E.Marchiori 2000) In the next two subsections we discuss both heuristic based evolutionary algorithms.

Asexual heuristic based genetic operator

The asexual heuristic based genetic operator selects a number of variables in a given individual, and then selects new values for these variables. We consider the operator that changes up to one fourth of the variables, selects the variables that are involved in the largest number of violated constraints, and selects the values for these variables which maximize the number of constraints they are involved in that become satisfied. In our implementation, for this purpose all the possible values for that gene are computed. We name a

GA applying this operator with a random mutation HGA1 (Craenen, A.E.Eiben, and E.Marchiori 2000).

Multiparent heuristic crossover

The basic mechanism of this crossover operator is scanning: for each position, the values of the variables of the parents in that position are used to determine the value of the variable in that position in the child. The selection of the value is done using the heuristic employed in the asexual operator. The difference with the asexual heuristic operator is that the heuristic does not evaluate all possible values but only those of the variables in the parents. The basic mechanism of multi-parent operators is scanning. The idea behind scanning is to examine (put a marker on) the positions of the parents consecutively and choose one of the values on the marked positions for the child. Choosing from the marked genes can be done problem independently. For instance choosing the one that occurs the most or by a random mechanism either uniform or biased by the fitness of the parents. Scanning can be enriched by problem dependent heuristics relying on extra information, e.g. edge length for routing problems. A particular way of adjusting scanning to special types of problems is to base it on special marker update mechanisms, e.g. shifting the markers to the first value that does not yet occur in the child and thus obtaining an operator applicable for permutation based problems, such as routing or scheduling.(Eiben, Raue, and Ruttkay 1994) For this problem we scanned different positions in the parents in two ways. One way is by giving the child the value that is repeated in that special position in most parents. The other way is to scan each position in the parents based on the num-

```

Online weight update mechanism
Set initial weights of constraints
while no termination do
  for the next N generation do
    let GA go with current weights
  end for
  Redefine weights of constraints and re evaluate the fitness
  function for all individuals
end while

```

Figure 4: Online weight adaption mechanism (SAW).

ber of violated constraints they are involved in and choose the value of the one involving in least constraints to give to the child. After applying both of these approaches we found the second one more effective and used that for our tests. We name a GA applying this operator with a random mutation HGA2. We call a GA which combines both heuristics HGA3.

Evolutionary algorithms with adaptive fitness

In this section, we describe some methods that guide the search by changing the way they compute fitness during the search so that they evaluate the individuals based on some special characteristics they may have. Two of these successful methods are depicted in this section.

Stepwise Adaption of Weights(SAW)

The SAW mechanism has been introduced by Eiben (Eiben and Van Der Hauw 1998) and van der Hauw (van der Hauw 1996) as an improved version of the weight adaptation mechanism of Eiben et al. The basic idea behind the SAW-ing mechanism is that constraints that are not satisfied or variables causing constraint violations after a certain number of steps must be hard, thus must be given a high weight (penalty). These high weights probably motivate the algorithm to solve these constraints. Thus, finding a way for setting the constraints weights so that they can show the hardness of constraints properly seems to be essential. One idea can be initializing the weights first and incrementing the weight of the constraints that are not satisfied in the best individual of the population after a certain number of generations. Using this algorithm can be done in two modes: offline and online mode (Eiben and van Hemert 1999). In offline mode, weights of constraints are set first through running an evolutionary algorithm for a limited number of generations and then the algorithm is run on the problem with the modified weights. In the online mode, modifying the constraints weights is done while the algorithm is running on the problem. Figure 4 shows the pseudo code for a SAW-ing EA in online mode. To compute the fitness for an individual in a SAWing GA, the weights of the unsolved constraints are simply added together.

In (Craenen and Eiben 2003) there have been studies for testing different heuristics EAs for solving random binary CSPs. The best performing EA in terms of effectivity (suc-

cess rate) and efficiency (measured by the average number of conflict checks) was the SAW algorithm.

Microgenetic Iterative Descent Algorithm (MGID)

Studies in (Eiben et al. 1998) shows that the Microgenetic Iterative Descent Algorithm (MGID) of Dozier et al (Dozier, Bowen, and Bahler 1994) gives the best results for binary CSP problems. MGID applies heuristics to the reproduction mechanism and to the fitness function in order to direct the search towards better individuals. Microgenetic algorithms have fairly small populations. MGID initializes the population by 8 individuals. It incorporates a steady state reproduction system and a roulette wheel selection which is not generational. Individuals are ranked based on their fitness and each individual would have a probability to get selected for being a parent and reproducing an offspring. Each offspring is created by mutating a special gene of the parent which is called pivot gene and then it replaces its parent in the population. The mechanism for selecting pivot gene considers the genes that involve in more constraints and are more probable to improve. The form of inheritance is designed in a way that allows a line of successors to try to minimize the number of constraints violations for a gene that is the current pivot and also allows successors to move on to another pivot if the current pivot can no longer be optimised. The fitness of a chromosome is determined by the sum of the number of constraints violated by each gene for all genes and the sum of the weights of breakouts that each individual has. A breakout consists of two parts: 1) a pair of values for two genes that violates a constraint; 2) a weight associated to that pair that shows how much this pair is bad(nogood tuple) . The set of breakouts is empty at first, but during the execution new breakouts are added to the set and also the weights of the existing breakouts may increase. The mechanism of adding the breakout tuples is that after a number of generations which is based on the number of genes in an individual and the number of values of each of these genes, if no progress would be seen in the best individual of the population the pairs of values in the best individual that violate the constraints would be considered as breakouts and be added to the breakout set. Once the breakouts are created or the weights of the preexisting breakouts are incremented, the fitness of each structure within the population is updated.

Experimentation

The purpose of these experiments is to compare different heuristic based evolutionary algorithms or EAs with adaptive fitness in terms of effectivity and efficiency. Percentage of all cases in which a solution is found or success rate and the average of the fitness value of the best individual found can be a good measure to show effectivity. Also the time it takes to reach a solution (in seconds) and number of fitness calculations is a good measure to show efficiency. For running the algorithms we used a SUN SPARC Ultra 5 station. All the procedures are coded in C/C++. TCSPs are randomly generated using the model RB proposed in (Xu and Li 2000). This model is a revision of the standard Model B (Gent et al. 1998;

Smith and Dyer 1996), has exact phase transition and the ability to generate asymptotically hard instances. Following the model RB, we generate each TCSP instance in two steps as shown below and using the parameters n , p , α and r where :

- n is the number of events,
 - p ($0 < p < 1$) is the constraint tightness which can be measured, as shown in (Sabin and Freuder 1994), as the fraction of all possible pairs of intervals from the domain of two events that are not allowed by the constraint,
 - and r and α ($0 < \alpha < 1$) are two positive constants.
1. Select with repetition $m \ln n$ random constraints. Each random constraint is formed by selecting without repetition 2 of n events.
 2. For each constraint we uniformly select without repetition pd^2 incompatible pairs of intervals from the domains of the pair of events involved by the constraint. $d = n^\alpha$ is the domain size of each event.

In heuristic based algorithms, other than specified heuristics we also used a roulette wheel selection operator, random mutation and replacement operator. With replacement operator we are able to introduce new random individuals to the population so that we can explore the search space more properly and we do that by replacing the worst individuals in the population with new random ones. In table 2 the operator used in each algorithm is shown.

The asexual operator is not a crossover operator because it only operates on one individual. However, in HGA1 we used the asexual operator instead of the crossover operator. In Figure 5 a pseudo code of the basic genetic algorithm used in our implementations is shown. Table 2 shows the operators used in different algorithms. For crossover we used one point crossover for all algorithms which cuts the parent individuals in a random point and swaps the data beyond that point between two parents. The resulting individuals would be children of the two parents. For selection, we use roulette wheel selection in all algorithms. In roulette wheel selection each individual has a probability for getting selected which depends on its fitness. The better the fitness of an individual the more it is likely to be selected. By replacing different operators in a GA based on the above table, other heuristic based GAs can be reached. For example, using asexual operator instead of crossover operator in a basic GA gives us the HGA1 algorithm. The main features of these three versions of HGA are the same as the ones implemented in (Craenen and Eiben 2003).

All heuristic based GAs and Sawing GA were run for the same set of parameters, with the population size of 300, mutation rate of 1, crossover rate of 0.7 and replacement rate of 0.2. Mutation rate and crossover rate are the probabilities with which these operators can be applied to the individuals and replacement rate shows the ratio of the individuals in population that will be replaced by new individuals. The

Initialize the population with random individuals

Repeat

1. Select best-ranking individuals to reproduce
2. Create a crossed population by applying crossover operator on the selected individuals
3. Create a mutated population by applying mutation on crossed population
4. Replace the worst individuals of the mutated population with new random individuals

Until termination

Figure 5: Pseudo code of the used GA.

fitness for heuristic based and basic GA is computed by the number of unsolved constraints for each individual.

Table 3 shows the success rate, the time to reach the solution and the average of the best fitness each algorithm could reach in a specified time for problems with different tightness. The success rate shows the percentage of runs in which a solution could be found. Each problem has 80 variables, was run for 20 times and was given a maximum time of 70 seconds.

As we can see in the table, sawing GA wins from other algorithms with regard to success rate. Clearly constraint weighting approach used in Sawing GA was successful in finding the group of hardest constraints. HGA2 and HGA3 are not successful in solving the problems even the easiest ones. Multiparent heuristic fails to solve the problem while the performance of asexual genetic algorithm is not very different from basic GA. We can conclude that heuristic based GAs do nothing better than basic GA and in case of HGA2 and HGA3 where we use Multiparty crossover it actually reduces the performance. The reason for this might be that the scanning mechanism was not appropriate for this particular problem.

Figure 6 shows the best fitness each algorithm could reach after a specified amount of time (70 second) for an unsolvable temporal CSP problem with tightness of 0.64. As we can see, basic GA wins from other methods.

Figure 7 shows the average number of fitness computations to reach the solution for the easiest tightness and for the algorithms which could reach the solution. As we can see, Microgenetic algorithm has the least number of computations. This is because the population size for this algorithm is very small. Also, it does not require evaluating all the individuals of the population in each generation as the other algorithms do.

	GA	HGA1	HGA2	HGA3
Selection operator	Roulette Wheel	Roulette Wheel	Roulette Wheel	Roulette Wheel
Mutation operator	Random Mutation	Random Mutation	Random Mutation	Asexual operator
Crossover operator	One Point Crossover	Asexual operator	MultiParent Crossover	MultiParent Crossover
Replacement operator	Worst Chromosomes	Worst Chromosomes	Worst Chromosomes	Worst Chromosomes
	Replacement	Replacement	Replacement	Replacement

Table 2: Genetic operators.

	0.24			0.16			0.05		
	SR	Time	Bfit	SR	Time	Bfit	SR	Time	Bfit
GA	0.2	50	1	0.9	18	0.1	1	3	0
HGA1	0.1	61	2	0.9	21	0.1	1	5	0
HGA2	0	-	19	0	-	8.15	0.05	65	1.6
HGA3	0	-	19	0	-	8.2	0	-	2.05
Sawing GA	0.7	45	0.4	1	1	0	1	5	0
MGID	0	-	4.4	1	4	0	1	1	0

Table 3: Success rate and average fitness for different heuristic based algorithms.

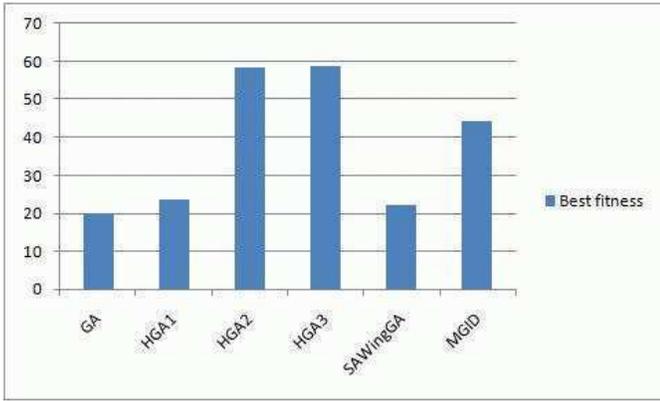


Figure 6: The best fitness each of the algorithms could reach in an specified amount of time.

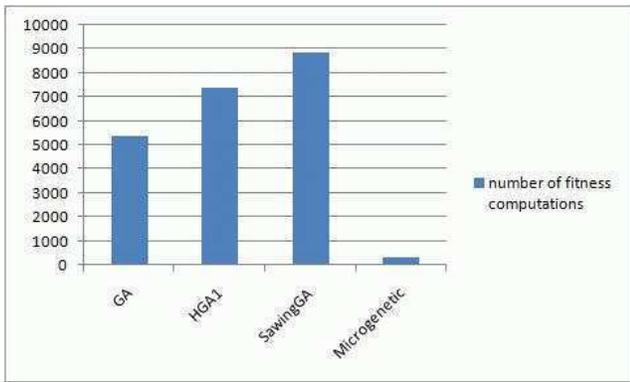


Figure 7: The number of fitness computation that algorithms do till they reach the solution.

Conclusion

In this paper we tested some evolutionary algorithms that use heuristics for solving Temporal CSP problems and compared their results. We used Temporo model for modeling Temporal CSP problems and generated our problems randomly. Overall, the following points can be concluded from the outcomes of the experiments: Adding heuristics to a genetic algorithm does not necessarily help this latter to perform better. It may cause more unnecessary computations which results in slowing the pace to reach the solution or may lead the genetic algorithm to get stuck in local optimums. Failing genetic algorithms enhanced with asexual operator and multiparent operator validates this argument. Sawing GA is the best with regards to success rate, with being able to spot harder constraint and focusing the search on them it was even able to solve the hardest solvable instance with a good success rate. In easier instances Sawing GA and MGID seem to have the same performance and both were better than basic GA. Our work can give some suggestions for future research. First of all, it seems essential not to rely hard on heuristics which bias search and limit the exploration of search space. This suggests a good balance between randomness and guidance. Also finding some possible heuristics specified to Temporal CSPs can be another step to complete this work.

References

- Allen, J. 1983. Maintaining knowledge about temporal intervals. *CACM* 26(11):832–843.
- Beaumont, M.; Sattar, A.; Maher, M.; and Thornton, J. 2001. Solving overconstrained temporal reasoning problems. In *14th Australian Joint Conference on Artificial Intelligence: Advances in Artificial Intelligence*.
- Craenen, B.; A.E.Eiben; and E.Marchiori. 2000. Solving constraint satisfaction problems with heuristic based evo-

- lutionary algorithms. In *Proceedings of the 2000 congress on evolutionary computation*, 1569–1575.
- Craenen, B., and Eiben, A. 2003. Comparing evolutionary algorithms on binary constraint satisfaction problems. *IEEE Transactions on Evolutionary Computation* 7(5):424–444.
- Dozier, G.; Bowen, J.; and Bahler, D. 1994. Solving small and large constraint satisfaction problems using a heuristicbased microgenetic algorithm. In *Proceedings of the First IEEE Conference on Evolutionary Computation*, 306–311.
- Eiben, A. E., and Van Der Hauw, J. K. 1998. Adaptive penalties for evolutionary graph coloring. *Lecture Notes in Computer Science* 1363:95–??
- Eiben, G., and van Hemert, J. 1999. SAW-ing EAs: Adapting the fitness function for solving constrained problems. In Corne, D.; Dorigo, M.; and Glover, F., eds., *New Ideas in Optimization*. London: McGraw-Hill. 389–402.
- Eiben, A.; van Hemert, J.; Marchiori, E.; and A.G.Steenbeek. 1998. Solving binary constraint satisfaction problems using evolutionary algorithms with an adaptive fitness function. In *Proceedings of the 5th Conference on Parallel Problem Solving from Nature*.
- Eiben, A.; RauC, P.-E.; and Ruttkay, Z. 1993. A network based adaptive evolutionary algorithm for constraint satisfaction problems.
- Eiben, A.; Raue, P.-E.; and Ruttkay, Z. 1994. Solving constraint satisfaction problems using genetic algorithms. In *Proc. 1st IEEE Conf. Evolutionary Computation*, 542547.
- Gent, I.; MacIntyre, E.; Prosser, P.; Smith, B.; and Walsh, T. 1998. Random constraint satisfaction: Flaws and structure.
- M.-C, R. R. 1998. A network based adaptive evolutionary algorithm for constraint satisfaction problems.
- Mouhoub, M.; Charpillet, F.; and Haton, J. 1998. Experimental Analysis of Numeric and Symbolic Constraint Satisfaction Techniques for Temporal Reasoning. *Constraints: An International Journal* 2:151–164, Kluwer Academic Publishers.
- Mouhoub, M. 2004a. Handling temporal constraints in a dynamic environment. *AI Communications* 17(4):201–211.
- Mouhoub, M. 2004b. Reasoning with numeric and symbolic time information. *Artificial Intelligence Review* 21:25–56.
- Poesio, M., and Brachman, R. J. 1991. Metric constraints for maintaining appointments: Dates and repeated activities. In *National Conference on Artificial Intelligence*, 253–259.
- Sabin, D., and Freuder, E. C. 1994. Contradicting conventional wisdom in constraint satisfaction. In *Proceedings of the Eleventh European Conference on Artificial Intelligence*, 125–129. Amsterdam, The Netherlands: John Wiley and Sons.
- Smith, B., and Dyer, M. 1996. Locating the phase transition in binary constraint satisfaction problems. *Artificial Intelligence* 81:155–181.
- Song, F., and Cohen, R. 1988. The interpretation of temporal relations in narrative. In *Proceedings of the 7th National Conference of the American Association for Artificial Intelligence (AAAI-88)*, 745750.
- Thornton, J.; Beaumont, M.; Sattar, A.; and Maher, M. 2002. Applying local search to temporal reasoning.
- Thornton, J.; Beaumont, M.; Sattar, A.; and Maher, M. 2004. A Local Search Approach to Modelling and Solving Interval Algebra Problems. *Journal of Logic and Computation* 14(1):93–112.
- van der Hauw, J. 1996. Evaluating and improving steady state evolutionary algorithms on constraint satisfaction problems.
- Xu, K., and Li, W. 2000. Exact Phase Transitions in Random Constraint Satisfaction Problems. *Journal of Artificial Intelligence Research* 12:93–103.

9-Intersection Calculi for Spatial Reasoning on the Topological Relations between Multi-Domain Objects

Yohei Kurata

SFB/TR8 Spatial Cognition, Universität Bremen
Postfach 330 440, 28334 Bremen, Germany
ykurata@informatik.uni-bremen.de

Abstract

Most spatial calculi target spatial relations between single-type objects, whereas there are also a number of spatial models that distinguish spatial relations between objects in different domains. How to equip such cross-domain spatial models with reasoning capability is left as a research question. As a first step, this paper develops a series of qualitative spatial calculi based on the 9-intersection. The 9-intersection distinguishes topological relations between various objects (points, lines, regions, bodies, etc.). We formulate two sorts of calculi: *homogeneous 9-intersection calculi* target the topological relations between single-type objects, while *heterogeneous 9-intersection calculi* can deal with multiple sets of topological relations between various combinations of objects. As the foundation of these calculi, composition tables and lists of converse relations are developed for various sets of topological relations in \mathbb{R}^1 and \mathbb{R}^2 . For heterogeneous 9-intersection calculi, the sets of base relations, composition tables, and list of converse relations are integrated, such that the algebraic framework of ordinary single-domain spatial calculi can be reused. Finally, the use of the new calculi is demonstrated.

Keywords: qualitative spatial calculi, topological relations, cross-domain spatial models, 9-intersection, composition tables

1. Introduction

Qualitative spatial calculi provide reasoning capability for the models of spatial relations. With these calculi, for instance, incompletely-observed spatial arrangements of objects can be disambiguated with regard to a specific set of spatial relations. Interestingly, most of existing qualitative spatial calculi target spatial relations between single-sort objects. For instance, Allen's interval algebra [1], Region Connection Calculus [2], Cardinal Direction Calculus [3], and Double Cross Calculus [4] feature a set of relations between two intervals, two regions, two points, and three points, respectively. Such *single-domain spatial calculi* fit nicely into an algebraic framework (*relation algebra* or its family), since their operations are closed under the single set of spatial relations. On the other hand, spatial database communities have developed a number of spatial models that distinguish spatial relations between objects in differ-

ent domains. For instance, the 9-intersection [5] can distinguish the topological relations between a line and a region, in addition to the relations between two regions or those between two lines. In [6], the target of cardinal direction relations is extended to arbitrary combination of objects (points, lines, and regions). In [7], spatial arrangements of a path and a landmark are modeled as relations between a directed line and a region using a Double-Cross-like frame of spatial reference. How to equip such cross-domain spatial models with reasoning capability is left as a research question. For instance, imagine a space where point-like objects, line-like objects, and region-like objects coexist. Given partial knowledge about their arrangement, can we disambiguate it? If this is possible by computation, the applicability of qualitative spatial calculi will be expanded considerably.

As a first step, this paper develops a series of qualitative spatial calculi based on the 9-intersection [5]. These *9-intersection calculi* consist of *homogeneous 9-intersection calculi*, which target a set of topological relations between single-type objects (e.g., line-line relations), and *heterogeneous 9-intersection calculi*, which can deal with multiple sets of topological relations between various combinations of objects (e.g., mixture of line-line, line-region, and region-region relations). We show that the algebraic framework of ordinary single-domain spatial calculi can be reused for the heterogeneous 9-intersection calculi and, consequently, we can use existing reasoning tools of spatial calculi, such as SparQ [8] and GQR [9], to conduct reasoning in the heterogeneous 9-intersection calculi. We expect that a similar approach achieves reasoning capability in other cross-domain spatial models as well.

A secondary but important challenge of this paper is to develop composition tables for various combinations of topological relations. The composition table of two topological region-region relations in \mathbb{R}^2 is reported in [10], but the tables for other combinations are not fully developed yet. We therefore develop these composition tables systematically with a small number of composition rules.

The remainder of this paper is organized as follow: Sections 2 and 3 summarize major concepts of qualitative spatial calculi and the 9-intersection, respectively. Section 4

develops the lists of converse relations and composition tables for various topological relations. Section 5 develops the 9-intersection calculi based on these lists and composition tables. Section 6 demonstrates the use of these calculi for qualitative spatial reasoning. Finally, Section 7 concludes with a discussion of future problems.

2. Qualitative Spatial Calculi

Qualitative spatial calculi (and their lower dimensional counterparts, qualitative temporal calculi) have been studied extensively in AI communities [11, 12]. In a broad sense, qualitative spatial calculi are the calculi formed by a set of spatial relations and operations on these relations. Typically, binary spatial calculi are equipped with two operations, *conversion* (*converse*) and *composition*, in addition to set-theoretic operations. By conversion we can derive the relation between A and B from the relation between B and A , while composition enable us to derive possible relations between A and B from the relation between A and C and that between C and B . Ternary spatial calculi also have counterparts of these operations [13]. This paper focuses on binary spatial calculi, since topological relations are binary relations.

Normally each binary spatial calculus targets a jointly exclusive and pairwise disjoint set of spatial relations that may hold between two arbitrary objects in a spatial object domain \mathbf{D} (points, regions, etc.), including an identity relation. These spatial relations are called *base relations* and as a set they are denoted \mathcal{B} .

In order to process incomplete knowledge about spatial relations, the set of all base relations that may hold between a pair of objects is treated as a unit of computation, called (*general*) *relation*. For instance, if the topological relations between two regions A and B are known to be *disjoint*_{RR} or *meet*_{RR}, the relation between A and B is represented as $\{\textit{disjoint}_{RR}, \textit{meet}_{RR}\}$. If nothing is known about the possible spatial relations between A and B , the relation between A and B is represented by the set of all base relations in \mathcal{B} , which is called the *universal relation* and denoted \mathcal{U} .

The set of all relations (essentially \mathcal{B} 's power-set $2^{\mathcal{B}}$) is denoted \mathcal{R} . The converse \cup and the composition \circ on \mathcal{R} are defined based on those on \mathcal{B} as equations 1-2. The set \mathcal{R} , together with its converse and composition operations closed under \mathcal{R} , gives rise to an algebra. Normally, a binary spatial calculus forms a *non-associative algebra* (or even its stronger version, a *relation algebra* or a *semi-associative algebra*, depending on its associativity [12]). Actually, from an algebraic point of view, Ligozat and Renz [12] defined a qualitative binary spatial calculus as a tuple of a non-associative algebra and its weak representation.

$$\forall R \in \mathcal{R} \quad R^{\cup} = \bigcup_{r \in R} r^{\cup} \quad (1)$$

$$\forall R_1, R_2 \in \mathcal{R} \quad R_1; R_2 = \bigcup_{r_i \in R_1, r_j \in R_2} (r_i; r_j) \quad (2)$$

The merit of such an algebraic treatment is that we can computationally disambiguate the relations between many objects by algebraic computation without paying attention to actual geometry of the objects. This problem corresponds to a constraint satisfaction problem (CSP). The CSP's key question is consistency checking, i.e., to identify the presence or absence of the variables that satisfy the given constraints. In spatial calculi, the variables and constraints correspond to spatial objects and their relations, respectively. Through checking algebraic closeness of every scenario, we can detect invalid combinations of spatial relations that cannot hold between the objects (or the absence of such combinations). By filtering them out, we can derive the *candidates* for the possible combinations of spatial relations between the objects (although at this level we cannot guarantee that all of these candidates are geometrically realizable). There are already some effective tools to support such constraint-based reasoning on user-defined spatial/temporal (e.g., SparQ [8] and GQR [9]).

3. The 9-Intersection

The 9-intersection [5] is a model of binary topological relations based on point-set topology [14]. This model has been studied extensively in spatial database communities, primarily because it applies to various combinations of objects systematically. In this model, the relations between two objects are distinguished by certain properties of intersections between their *topological parts* (*interior*, *boundary*, and *exterior*). The interior, boundary, and exterior of a spatial object X , denoted X° , ∂X , and X^{-} , are defined as the union of all open sets contained in X , the difference between X 's *closure* (i.e., the intersection of all closed point sets that contain X) and X° , and the complement of X 's closure, respectively. The *9-intersection matrix* in equation 3 concisely represents the 3×3 parts' intersections between two objects A and B .

$$M(A, B) = \begin{pmatrix} A^{\circ} \cap B^{\circ} & A^{\circ} \cap \partial B & A^{\circ} \cap B^{-} \\ \partial A \cap B^{\circ} & \partial A \cap \partial B & \partial A \cap B^{-} \\ A^{-} \cap B^{\circ} & A^{-} \cap \partial B & A^{-} \cap B^{-} \end{pmatrix} \quad (3)$$

In the most basic approach, topological relations are distinguished by the presence or absence of these 3×3 intersections. Thus, we consider *two-valued 9-intersection matrix* whose element are either empty (\emptyset) or non-empty ($\neg \emptyset$). By the patterns of the two-valued 9-intersection matrix, for instance, we can distinguish two point-point relations, three point-line relations, and eight line-line relations in a 1D Euclidian space \mathbb{R}^1 (figure 1). Note that by definition a point does not have an interior and the line's boundary refers to the set of its two endpoints.

The set of topological relations distinguished by the patterns of two-valued 9-intersection is denoted $\mathcal{J}_{\mathbf{D}_1 \mathbf{D}_2 \mathcal{S}}$,

where \mathbf{D}_1 and \mathbf{D}_2 are the domains of two objects (\mathbf{P} : points, \mathbf{L} : simple lines, \mathbf{R} : simple regions, and \mathbf{B} : simple bodies) and \mathcal{S} is the space. For instance, $\mathcal{T}_{\mathbf{LL}-\mathbb{R}^1}$ refers to the set of topological line-line relations in \mathbb{R}^1 (figure 1d). Table 1 summarizes the numbers of topological relations distinguished by the patterns of the two-valued 9-intersection matrix.

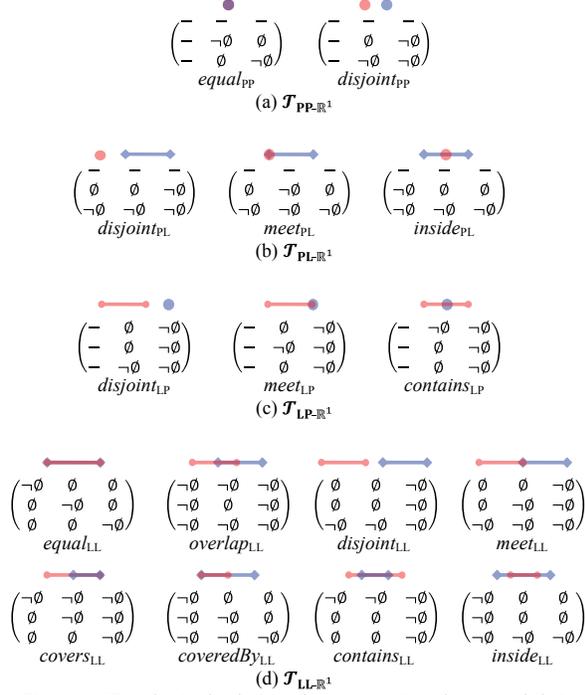


Figure 1. Topological relations between points, lines, and their combinations in \mathbb{R}^1 distinguished by the patterns of the two-valued 9-intersection.

Table 1. Numbers of topological relations distinguished by the patterns of the two-valued 9-intersection matrix [15].

	\mathbb{R}^1	\mathbb{R}^2	\mathbb{R}^3	\mathbb{S}^1	\mathbb{S}^2
Point-Point	2	2	2	2	2
Point-Line / Line-Point	3	3	3	3	3
Point-Region / Region-Point	-	3	3	-	3
Point-Body / Body-Point	-	-	3	-	-
Line-Line	8	33	33	11	33
Line-Region / Region-Line	-	19	31	-	19
Line-Body / Body-Line	-	-	19	-	-
Region-Region	-	8	43	-	11
Region-Body / Body-Region	-	-	19	-	-
Body-Body	-	-	8	-	-

4. Conversion and Composition

Conversion and composition are fundamental operations of qualitative spatial calculi. This section develops these two operations for a variety of topological relations.

The converse of a relation r in $\mathcal{T}_{\mathbf{D}_1\mathbf{D}_2-\mathcal{S}}$ is a relation in $\mathcal{T}_{\mathbf{D}_2\mathbf{D}_1-\mathcal{S}}$. For instance, the converse of $contains_{\mathbf{RP}}$ in $\mathcal{T}_{\mathbf{RP}-\mathbb{R}^2}$ (region-point relations) is $inside_{\mathbf{PR}}$ in $\mathcal{T}_{\mathbf{PR}-\mathbb{R}^2}$ (point-region relations). We can derive the converse of a relation r in $\mathcal{T}_{\mathbf{D}_1\mathbf{D}_2-\mathcal{S}}$ simply by transposing r 's 9-intersection matrix and finding the same pattern from the two-valued 9-intersection matrices that represent the relations in $\mathcal{T}_{\mathbf{D}_2\mathbf{D}_1-\mathcal{S}}$. By repeating this process for every relation in $\mathcal{T}_{\mathbf{D}_1\mathbf{D}_2-\mathcal{S}}$, we can obtain the *converse list* of $\mathcal{T}_{\mathbf{D}_1\mathbf{D}_2-\mathcal{S}}$, denoted $\mathcal{T}\text{-CL}_{\mathbf{D}_1\mathbf{D}_2-\mathcal{S}}$, which shows the mapping from $\mathcal{T}_{\mathbf{D}_1\mathbf{D}_2-\mathcal{S}}$ to $\mathcal{T}_{\mathbf{D}_2\mathbf{D}_1-\mathcal{S}}$ by conversion (e.g., table 2).

Table 2. Converse list of topological point-line relations in \mathbb{R}^1 ($\mathcal{T}\text{-CL}_{\mathbf{PL}-\mathbb{R}^1}$).

r	$disjoint_{\mathbf{PL}}$	$meet_{\mathbf{PL}}$	$inside_{\mathbf{PL}}$
r^{\cup}	$disjoint_{\mathbf{LP}}$	$meet_{\mathbf{LP}}$	$contains_{\mathbf{LP}}$

The composition of a relation r_1 in $\mathcal{T}_{\mathbf{D}_1\mathbf{D}_2-\mathcal{S}}$ and a relation r_2 in $\mathcal{T}_{\mathbf{D}_2\mathbf{D}_3-\mathcal{S}}$ is a subset of $\mathcal{T}_{\mathbf{D}_1\mathbf{D}_3-\mathcal{S}}$. The *composition table* of two topological relation sets $\mathcal{T}_{\mathbf{D}_1\mathbf{D}_2-\mathcal{S}}$ and $\mathcal{T}_{\mathbf{D}_2\mathbf{D}_3-\mathcal{S}}$, denoted $\mathcal{T}\text{-CT}_{\mathbf{D}_1\mathbf{D}_2\mathbf{D}_3-\mathcal{S}}$, shows the mapping from $\mathcal{T}_{\mathbf{D}_1\mathbf{D}_2-\mathcal{S}} \times \mathcal{T}_{\mathbf{D}_2\mathbf{D}_3-\mathcal{S}}$ to $2^{\mathcal{T}_{\mathbf{D}_1\mathbf{D}_3-\mathcal{S}}}$ (the power-set of $\mathcal{T}_{\mathbf{D}_1\mathbf{D}_3-\mathcal{S}}$) by composition. In this study, we develop composition tables for the combination of topological relations between simple objects in \mathbb{R}^1 (i.e., $\mathcal{T}\text{-CT}_{\mathbf{PPP}-\mathbb{R}^1}$, $\mathcal{T}\text{-CT}_{\mathbf{PPL}-\mathbb{R}^1}$, $\mathcal{T}\text{-CT}_{\mathbf{PLP}-\mathbb{R}^1}$, ..., $\mathcal{T}\text{-CT}_{\mathbf{LLL}-\mathbb{R}^1}$) and for those in \mathbb{R}^2 (i.e., $\mathcal{T}\text{-CT}_{\mathbf{PPP}-\mathbb{R}^2}$, ..., $\mathcal{T}\text{-CT}_{\mathbf{RRR}-\mathbb{R}^2}$).

Given three objects A , B , and C ($A \in \mathbf{D}_A$, $B \in \mathbf{D}_B$, $C \in \mathbf{D}_C$), the following set-theoretic constraints, originally introduced in [16] for deriving $\mathcal{T}\text{-CT}_{\mathbf{RRR}-\mathbb{R}^2}$, always hold for the composition of the topological relation between A and B and that between B and C :

- A 's topological part p_A and C 's topological part p_C do not intersect if B has a topological part p_B that includes p_A but does not intersect with p_C or that includes p_C but does not intersect with p_A ; and
- p_A and p_C intersect if B has a topological part p_B that intersects with p_A and is included in p_C or that intersects with p_C and is included in p_A .

By filtering all relations in $\mathcal{T}_{\mathbf{D}_A\mathbf{D}_B-\mathcal{S}}$ with these constraints, we obtain the *candidates* for the composition of the topological relation between A and B and that between B and C . Each candidate is examined if they have geometric interpretations. Then, the set of valid candidates are approved as the composition of the topological relation between A and B and that between B and C . By repeating this process for every relation pair in $\mathcal{T}_{\mathbf{D}_A\mathbf{D}_B-\mathcal{S}} \times \mathcal{T}_{\mathbf{D}_B\mathbf{D}_C-\mathcal{S}}$, we can develop the composition table of $\mathcal{T}_{\mathbf{D}_A\mathbf{D}_B-\mathcal{S}}$ and $\mathcal{T}_{\mathbf{D}_B\mathbf{D}_C-\mathcal{S}}$

(i.e., $\mathcal{T}\text{-CT}_{\mathbf{D}_A \mathbf{D}_B \mathbf{D}_C \mathcal{S}}$). For instance, table 3 shows the composition table of line-line relations and line-point relations in \mathbb{R}^1 (i.e., $\mathcal{T}\text{-CT}_{\mathbf{LLP}\text{-}\mathbb{R}^1}$) derived by this method.

Table 3. Composition table of topological line-line relations and topological line-point relations in \mathbb{R}^1 ($\mathcal{T}\text{-CT}_{\mathbf{LLP}\text{-}\mathbb{R}^1}$).

	$disjoint_{LP}$	$meet_{LP}$	$contains_{LP}$
$equal_{LL}$	$disjoint_{LP}$	$meet_{LP}$	$contains_{LP}$
$disjoint_{LL}$	\mathbf{U}_{LP}	$disjoint_{LP}$	$disjoint_{LP}$
$meet_{LL}$	\mathbf{U}_{LP}	$disjoint_{LP} / meet_{LP}$	$disjoint_{LP}$
$overlaps_{LL}$	\mathbf{U}_{LP}	\mathbf{U}_{LP}	\mathbf{U}_{LP}
$covers_{LL}$	$disjoint_{LP}$	$disjoint_{LP} / meet_{LP}$	\mathbf{U}_{LP}
$coveredBy_{LL}$	\mathbf{U}_{LP}	$meet_{LP} / contains_{LP}$	$contains_{LP}$
$contains_{LL}$	$disjoint_{LP}$	$disjoint_{LP}$	\mathbf{U}_{LP}

According to our investigation, the previous two constraints are sufficient when developing the *most* composition tables ($\mathcal{T}\text{-CT}_{\mathbf{PPP}\text{-}\mathbb{R}^1}$, ..., $\mathcal{T}\text{-CT}_{\mathbf{LLL}\text{-}\mathbb{R}^1}$ and $\mathcal{T}\text{-CT}_{\mathbf{PPP}\text{-}\mathbb{R}^2}$, ..., $\mathcal{T}\text{-CT}_{\mathbf{RRR}\text{-}\mathbb{R}^2}$), but not $\mathcal{T}\text{-CT}_{\mathbf{LLL}\text{-}\mathbb{R}^2}$ —in this case, the derived composition candidates may have no geometric interpretation. For instance, imagine that there are three simple lines A , B , and C , where A contains B and B crosses C (figure 2). Obviously, A cannot contain C . The constraints in equation 4, however, do not exclude the composition candidate where A contains C .



Figure 2. The arrangements of three lines A , B , and C , from which we can conclude that A cannot contain C .

In general, when A contains/covers B , A cannot contain/covers C if:

- C directly links B 's interior and exterior (figure 3);
- B directly links C 's interior and exterior; or
- A covers B and B is inside of C (figure 4).

We can tell from the given relations that the first two situations occur whenever the relation between B and C belong to the topological line-line relations not realizable in \mathbb{R}^1 (i.e., the line-line relations other than $equal_{LL}$, $disjoint_{LL}$, $meet_{LL}$, $overlaps_{LL}$, $covers_{LL}$, $contains_{LL}$, and $inside_{LL}$). Thus, the previous condition is simplified as follows:

- If $contains_{LL}(A, B)$ and the relation between B and C is neither $equal_{LL}$, $disjoint_{LL}$, $meet_{LL}$, $overlaps_{LL}$, $covers_{LL}$, $contains_{LL}$, nor $inside_{LL}$, then $A^- \cap C^+ = \neg\emptyset$;
- If $covers_{LL}(A, B)$ and the relation between B and C is neither $equal_{LL}$, $disjoint_{LL}$, $meet_{LL}$, $overlaps_{LL}$, $covers_{LL}$, nor $contains_{LL}$, then $A^- \cap C^+ = \neg\emptyset$;

Similarly, the following two constraints hold:

- If $inside_{LL}(B, C)$ and the relation between A and B is neither $equal_{LL}$, $disjoint_{LL}$, $meet_{LL}$, $overlaps_{LL}$, $covers_{LL}$, $contains_{LL}$, nor $inside_{LL}$, then $A^+ \cap C^- = \neg\emptyset$; and
- If $coveredBy_{LL}(B, C)$ and the relation between A and B is neither $equal_{LL}$, $disjoint_{LL}$, $meet_{LL}$, $overlaps_{LL}$, $covers_{LL}$, nor $contains_{LL}$, then $A^+ \cap C^- = \neg\emptyset$.

By adding these four constraints, we can successfully derive the composition candidates that have geometric interpretations and, accordingly, we can develop the composition table accordingly, we can develop the composition table $\mathcal{T}\text{-CT}_{\mathbf{LLL}\text{-}\mathbb{R}^2}$.

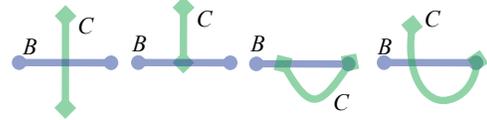


Figure 3. Examples of arrangements where a line C directly connects the interior and exterior of a line B .

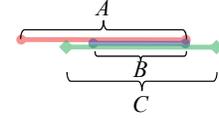


Figure 4. If a line A covers a line B and B is inside of a line C , then A cannot contain/covers C .

5.9-Intersection Calculi

First, to conduct reasoning on topological relations between single-type objects, we introduce *homogeneous 9-intersection calculi*. These calculi are defined for each object domain and each space. The homogeneous 9-intersection calculus for the object domain \mathbf{D} and the space \mathcal{S} , denoted $\text{Homo9IC}_{\mathbf{D}\text{-}\mathcal{S}}$, is formulated based on the following elements:

- a set of base relations $\mathcal{T}_{\mathbf{DD}\text{-}\mathcal{S}}$;
- a converse list $\mathcal{T}\text{-CL}_{\mathbf{DD}\text{-}\mathcal{S}}$; and
- a composition table $\mathcal{T}\text{-CT}_{\mathbf{DDD}\text{-}\mathcal{S}}$.

These elements satisfy the requirements of ordinary qualitative spatial calculi (non-associative algebra); that is,

- $\mathcal{T}_{\mathbf{DD}\text{-}\mathcal{S}}$ is jointly exclusive, pairwise disjoint, and contains an identity element;
- the converse operation on $\mathcal{T}_{\mathbf{DD}\text{-}\mathcal{S}}$ is closed under $\mathcal{T}_{\mathbf{DD}\text{-}\mathcal{S}}$ (i.e., $\forall r \in \mathcal{T}_{\mathbf{DD}\text{-}\mathcal{S}} r^\cup \in \mathcal{T}_{\mathbf{DD}\text{-}\mathcal{S}}$); and
- the composition operation on $\mathcal{T}_{\mathbf{DD}\text{-}\mathcal{S}}$ is closed under $\mathcal{T}_{\mathbf{DD}\text{-}\mathcal{S}}$ (i.e., $\forall r_1, r_2 \in \mathcal{T}_{\mathbf{DD}\text{-}\mathcal{S}} r_1; r_2 \subseteq \mathcal{T}_{\mathbf{DD}\text{-}\mathcal{S}}$).

Consequently, spatial reasoning can be conducted in an algebraic framework.

Next, to conduct reasoning on topological relations between various combinations of objects, *heterogeneous 9-intersection calculi* are introduced. These calculi are developed for each space. The heterogeneous 9-intersection

calculus for the space \mathcal{S} , denoted $\text{Het9IC}_{\mathcal{S}}$, have the ability to deal with all sorts of simple objects (points, simple lines, simple regions, and simple bodies) that \mathcal{S} can contain. Naturally, if \mathcal{S} is a d -dimension space, $\text{Het9IC}_{\mathcal{S}}$ covers:

- $d+1$ object domains $\{\mathbf{D}_0, \dots, \mathbf{D}_d\}$;
- $(d+1)^2$ sets of topological relations $\{\mathcal{T}_{\mathbf{D}_i \mathbf{D}_j \mathcal{S}}\}_{i,j \in \{0, \dots, d\}}$;
- $(d+1)^2$ converse lists $\{\mathcal{J}\text{-CL}_{\mathbf{D}_i \mathbf{D}_j \mathcal{S}}\}_{i,j \in \{0, \dots, d\}}$; and
- $(d+1)^3$ composition tables $\{\mathcal{J}\text{-CT}_{\mathbf{D}_i \mathbf{D}_j \mathbf{D}_k \mathcal{S}}\}_{i,j,k \in \{0, \dots, d\}}$

For instance, $\text{Het9IC}_{\mathbb{R}^1}$ covers:

- two object domains: \mathbf{P} and \mathbf{L} ;
- four sets of topological relations: $\mathcal{T}_{\mathbf{PP}\mathbb{R}^1}$, $\mathcal{T}_{\mathbf{PL}\mathbb{R}^1}$, $\mathcal{T}_{\mathbf{LP}\mathbb{R}^1}$, and $\mathcal{T}_{\mathbf{LL}\mathbb{R}^1}$;
- four converse lists: $\mathcal{J}\text{-CL}_{\mathbf{PP}\mathbb{R}^1}$, $\mathcal{J}\text{-CL}_{\mathbf{PL}\mathbb{R}^1}$, $\mathcal{J}\text{-CL}_{\mathbf{LP}\mathbb{R}^1}$, and $\mathcal{J}\text{-CL}_{\mathbf{LL}\mathbb{R}^1}$; and
- eight composition tables: $\mathcal{J}\text{-CT}_{\mathbf{PPP}\mathbb{R}^1}$, $\mathcal{J}\text{-CT}_{\mathbf{PPL}\mathbb{R}^1}$, $\mathcal{J}\text{-CT}_{\mathbf{PLP}\mathbb{R}^1}$, $\mathcal{J}\text{-CT}_{\mathbf{PLL}\mathbb{R}^1}$, \dots , $\mathcal{J}\text{-CT}_{\mathbf{LLL}\mathbb{R}^1}$.

These elements are integrated as follows. First, the *generalized object domain* \mathbf{D}^* and the generalized base relations \mathcal{B}^* are defined as follows:

- $\mathbf{D}^* = \bigcup_{i \in \{1, \dots, d\}} \mathbf{D}_i$
- $\mathcal{B}^* = \left(\bigcup_{i,j \in \{1, \dots, d\}} \mathcal{B}_{\mathbf{D}_i \mathbf{D}_j} \right) \cup \{equal\}$

Basically \mathcal{B}^* refers to the relations between two arbitrary objects in \mathbf{D}^* , but it also contains an identity relation (*equal*). The presence of an identity relation is a requirement of the calculi's algebraic framework (non-associative algebra). This identity element is different from domain-level identity elements $equal_{\mathbf{D}_i \mathbf{D}_i}$. In $\text{Het9IC}_{\mathbb{R}^1}$, for instance, \mathcal{B}^* contains *equal*, *equal_{PP}*, and *equal_{LL}*. We did not integrate these identity elements to prevent senseless compositions. For instance, *equal_{PP}*; *disjoint_{LL}* must be empty since the composition of a point-point relation and a line-line relation is impossible. However, *equal*; *disjoint_{LL}* = $\{disjoint_{LL}\}$ by definition and, accordingly, it is not appropriate to substitute *equal_{PP}* by *equal*. Thus, *equal* is considered a purely abstract relation with no geometric interpretation (i.e., $\forall A, B \in \mathbf{D}^* (A, B) \notin equal$). Then, we can consider \mathcal{B}^* a jointly exhaustive and pairwise disjoint set of base relations, which is also a requirement of the calculi's algebraic framework.

Next, we integrate the relevant converse lists and composition tables. The integrated converse list $\mathcal{J}\text{-CL}_{\mathbf{D}^* \mathbf{D}^* \mathcal{S}}$ is derived by concatenating the relevant converse lists $\{\mathcal{J}\text{-CL}_{\mathbf{D}_i \mathbf{D}_j \mathcal{S}}\}$ and adding an item "*equal^o* = *equal*." For instance, table 4 shows $\mathcal{J}\text{-CL}_{\mathbf{D}^* \mathbf{D}^* \mathbb{R}^1}$, which is derived from $\mathcal{J}\text{-CL}_{\mathbf{PP}\mathbb{R}^1}$, \dots , $\mathcal{J}\text{-CL}_{\mathbf{LL}\mathbb{R}^1}$. Similarly, the integrated composition table $\mathcal{J}\text{-CT}_{\mathbf{D}^* \mathbf{D}^* \mathcal{S}}$ is derived by adjoining the relevant composition tables $\{\mathcal{J}\text{-CT}_{\mathbf{D}_i \mathbf{D}_j \mathbf{D}_k \mathcal{S}}\}$ and adding one row and one column about *equal*-related composition. For

instance, table 5 shows $\mathcal{J}\text{-CT}_{\mathbf{D}^* \mathbf{D}^* \mathbb{R}^1}$, which is derived from $\mathcal{J}\text{-CT}_{\mathbf{PP}\mathbb{R}^1}$, \dots , $\mathcal{J}\text{-CT}_{\mathbf{LLL}\mathbb{R}^1}$.

Table 4. Integrated converse list $\mathcal{J}\text{-CL}_{\mathbf{D}^* \mathbf{D}^* \mathbb{R}^1}$ (the highlighted part corresponds to $\mathcal{J}\text{-CL}_{\mathbf{PL}\mathbb{R}^1}$ in Table 2).

r	r^{\cup}	r	r^{\cup}
<i>equal</i>	<i>equal</i>	<i>equal_{LL}</i>	<i>equal_{LL}</i>
<i>equal_{PP}</i>	<i>equal_{PP}</i>	<i>disjoint_{LL}</i>	<i>disjoint_{LL}</i>
<i>disjoint_{PP}</i>	<i>disjoint_{PP}</i>	<i>meet_{LL}</i>	<i>meet_{LL}</i>
<i>disjoint_{PL}</i>	<i>disjoint_{LP}</i>	<i>overlaps_{LL}</i>	<i>overlaps_{LL}</i>
<i>meet_{PL}</i>	<i>meet_{LP}</i>	<i>covers_{LL}</i>	<i>coveredBy_{LL}</i>
<i>inside_{PL}</i>	<i>contains_{LP}</i>	<i>coveredBy_{LL}</i>	<i>covers_{LL}</i>
<i>disjoint_{LP}</i>	<i>disjoint_{PL}</i>	<i>contains_{LL}</i>	<i>inside_{LL}</i>
<i>meet_{LP}</i>	<i>meet_{PL}</i>	<i>inside_{LL}</i>	<i>contains_{LL}</i>
<i>contains_{LP}</i>	<i>inside_{PL}</i>		

Now we have:

- an integrated set of base relations \mathcal{B}^* , which is jointly exclusive, pairwise disjoint, and contains an identity relation;
- an integrated converse list $\mathcal{J}\mathcal{R}\text{-CL}_{\mathbf{D}^* \mathbf{D}^* \mathbb{R}^1}$, which is closed under \mathcal{B}^* ; and
- an integrated composition table $\mathcal{J}\text{-CT}_{\mathbf{D}^* \mathbf{D}^* \mathbb{R}^1}$, which is closed under \mathcal{B}^* .

Consequently, it is expected that spatial reasoning on the topological relations between arbitrary objects in \mathbf{D}^* can be conducted in an algebraic framework, just like we can do in ordinary single-domain spatial calculi. This will be demonstrated in Section 6.

Simple Assessment of 9-Intersection Calculi

Based on the converse lists and composition tables developed in Section 4, we developed seven basic calculi: $\text{Homo9IC}_{\mathbf{P}\mathbb{R}^1}$, $\text{Homo9IC}_{\mathbf{L}\mathbb{R}^1}$, $\text{Homo9IC}_{\mathbf{P}\mathbb{R}^2}$, $\text{Homo9IC}_{\mathbf{L}\mathbb{R}^2}$, $\text{Homo9IC}_{\mathbf{R}\mathbb{R}^2}$, $\text{Het9IC}_{\mathbb{R}^1}$, and $\text{Het9IC}_{\mathbb{R}^2}$. We conducted simple assessment of these calculi.

First, for the composition table of each calculus, we calculated the *crispness* and the *ratio of unique compositions* (table 5). These two measures are used in spatial database studies for assessing the effectiveness of composition tables [17, 18]. We found that $\text{Het9IC}_{\mathbb{R}^1}$ and $\text{Het9IC}_{\mathbb{R}^2}$ marked high crispness, but this result is not so meaningful because the integration of composition tables yields the increase of relations not contained in each composition and increases the crispness. We also found that $\text{Homo9IC}_{\mathbf{L}\mathbb{R}^2}$'s ratio of unique compositions was very low. This is because in many compositions the presence or absence of intersection between two lines' interiors cannot be determined. $\text{Het9IC}_{\mathbb{R}^2}$'s ratio of unique compositions was also low, because its composition table has many empty cells that correspond to impossible compositions.

Next, we examined the associativity of the compositions in each calculus (i.e., whether $(A; B); C = A; (B; C)$ holds or not) (table 6). We found that $\text{Homo9IC}_{\mathbb{R}^2}$ is not associative, and accordingly $\text{Het9IC}_{\mathbb{R}^2}$ as well. Alternatively, the compositions in these two calculi satisfy semi-associativity (i.e., $(A; \mathbf{U}); \mathbf{U} = A; (\mathbf{U}; \mathbf{U})$ holds). On the other hand, other five calculi are all associative. One example of non-associativity in $\text{Homo9IC}_{\mathbb{R}^2}$ is that $\text{covers}_{LL}; (\text{covers}_{LL}; \text{cross}_{LL}) \ni \text{covers}_{LL}$ (equations 4-6), but $(\text{covers}_{LL}; \text{covers}_{LL}); \text{cross}_{LL} \not\ni \text{covers}_{LL}$ (equations 7-10). This conflict arises from the ambiguity of the pattern $\begin{pmatrix} \neg\emptyset & \neg\emptyset & \neg\emptyset \\ \neg\emptyset & \emptyset & \neg\emptyset \\ \neg\emptyset & \neg\emptyset & \neg\emptyset \end{pmatrix}$. In equation 4, this pattern is interpreted as *diverge&cross&divergedBy*_{LL} relation (see *B* and *D* in figure 5a), whereas in equation 6 this pattern is interpreted as *overlap*_{LL} relation (see *B* and *D* in figure 5b). Unfortunately, the ambiguity of this pattern is an intrinsic problem of the 9-intersection.

$$\text{covers}_{LL}; \text{cross}_{LL} \ni \begin{pmatrix} \neg\emptyset & \neg\emptyset & \neg\emptyset \\ \neg\emptyset & \emptyset & \neg\emptyset \\ \neg\emptyset & \neg\emptyset & \neg\emptyset \end{pmatrix}_{LL} \quad (4)$$

$$\text{covers}_{LL}; \begin{pmatrix} \neg\emptyset & \neg\emptyset & \neg\emptyset \\ \neg\emptyset & \emptyset & \neg\emptyset \\ \neg\emptyset & \neg\emptyset & \neg\emptyset \end{pmatrix}_{LL} \ni \text{covers}_{LL} \quad (5)$$

$$\therefore \text{covers}_{LL}; (\text{covers}_{LL}; \text{cross}_{LL}) \ni \text{covers}_{LL} \quad (6)$$

$$\begin{aligned} & (\text{covers}_{LL}; \text{covers}_{LL}); \text{cross}_{LL} \\ & = (\text{covers}_{LL} \vee \text{contains}_{LL}); \text{cross}_{LL} \end{aligned} \quad (7)$$

$$= \text{covers}_{LL}; \text{cross}_{LL} \vee \text{contains}_{LL}; \text{cross}_{LL} \quad (8)$$

$$\text{contains}_{LL}; \text{cross}_{LL} \not\ni \text{covers}_{LL} \quad (9)$$

$$\therefore (\text{covers}_{LL}; \text{covers}_{LL}); \text{cross}_{LL} \not\ni \text{covers}_{LL} \quad (10)$$

Table 6. Properties of 9-intersection calculi for \mathbb{R}^1 and \mathbb{R}^2 .

	$\text{Homo9IC}_{\mathbb{P}\text{-}\mathbb{R}^1}$	$\text{Homo9IC}_{\mathbb{L}\text{-}\mathbb{R}^1}$	$\text{Homo9IC}_{\mathbb{P}\text{-}\mathbb{R}^2}$	$\text{Homo9IC}_{\mathbb{L}\text{-}\mathbb{R}^2}$	$\text{Homo9IC}_{\mathbb{R}\text{-}\mathbb{R}^2}$	$\text{Het9IC}_{\mathbb{R}^1}$	$\text{Het9IC}_{\mathbb{R}^2}$
Number of relations	2	8	2	33	8	17	94
Crispness of compositions	.375	.623	.375	.645	.623	.869	.910
Ratio of unique composition	.750	.422	.750	.080	.422	.553	.167
Associativity	√	√	√		√	√	
Semi-Associativity	√	√	√	√	√	√	√

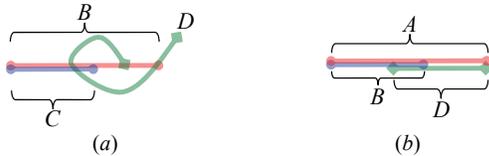


Figure 5. Illustrations of (a) equation 5 and (b) equation 6.

6. Examples

This section demonstrates the application of the proposed 9-intersection calculi for qualitative spatial reasoning. We start from $\text{Homo9IC}_{\mathbb{L}\text{-}\mathbb{R}^2}$ as the representative of homogeneous 9-intersection calculi.

In the Boston metropolitan area, there are four interstate highways; I-90, I-93, I-95, and I-495. Their actual network is like figure 6a, but here we consider a simplified network in figure 6b. Table 7 lists the topological relations between the highways in the simplified network. Imagine that we drive two of these four highways and observe their connections to other highways. For instance, figure 7a/b illustrates the knowledge obtained from the drive on I-90 and I-95/I-93. Based on such knowledge, what can we say about the relations between the remaining two highways? With $\text{Homo9IC}_{\mathbb{L}\text{-}\mathbb{R}^2}$, we can derive possible relations between unvisited highways from partial knowledge about the highway network.

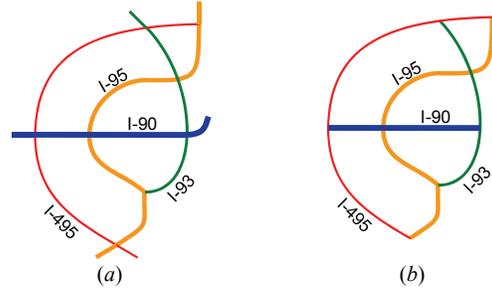


Figure 6. (a) Network of four interstate highways in the Boston metropolitan area and (b) its simplified version for experiment.

Table 7. Topological relations between pairs of highways in the network of figure 6b.

	I-90	I-93	I-95	I-495
I-90	–	<i>diverges</i> _{LL}	<i>cross</i> _{LL}	<i>diverges</i> _{LL}
I-93	<i>divergedBy</i> _{LL}	–	<i>diverges & cross</i> _{LL}	<i>diverges</i> _{LL}
I-95	<i>cross</i> _{LL}	<i>divergedBy & cross</i> _{LL}	–	<i>meet-at-both-ends</i> _{LL}
I-495	<i>divergedBy</i> _{LL}	<i>divergedBy</i> _{LL}	<i>meet-at-both-ends</i> _{LL}	–

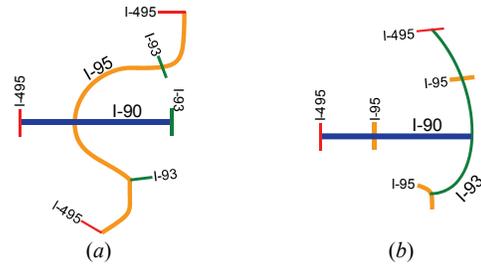


Figure 7. Partial knowledge about the highway network, obtained through the drive on (a) I-90 and I-95 and (b) I-90 and I-93.

For actual computation, we put the data of $\text{Homo9IC}_{\mathbb{L}\text{-}\mathbb{R}^2}$ ($\mathcal{T}_{\mathbb{L}\text{-}\mathbb{R}^2}, \mathcal{T}\text{-CL}_{\mathbb{L}\text{-}\mathbb{R}^2}$, and $\mathcal{T}\text{-CT}_{\mathbb{L}\text{-}\mathbb{R}^2}$) into SparQ [8] and calculate all consistent scenarios under the constraint network that follows table 7 but replaces the relations between unvisited highways by $\mathbf{U}_{\mathbb{L}\text{-}\mathbb{L}}$. The computation result is shown in table 8. For each pair of unvisited highways we obtained 4 to 28 possible relations. Each solution successfully contained the actual relation in the network of figure 6b.

Table 8. Possible relations between pairs of unvisited highways derived as algebraically-consistent scenarios.

Unknown relation	Derived solution (dv: diverges, dB: divergedBy)
(195, I495)	All but $\text{equal}_{\mathbb{L}\text{-}\mathbb{L}}$, $\text{covers}_{\mathbb{L}\text{-}\mathbb{L}}$, $\text{coveredBy}_{\mathbb{L}\text{-}\mathbb{L}}$, $\text{contains}_{\mathbb{L}\text{-}\mathbb{L}}$, $\text{inside}_{\mathbb{L}\text{-}\mathbb{L}}$
(193, I495)	$\text{disjoint}_{\mathbb{L}\text{-}\mathbb{L}}$, $\text{cross}_{\mathbb{L}\text{-}\mathbb{L}}$, $\text{diverges}_{\mathbb{L}\text{-}\mathbb{L}}$, $\text{diverges}\&\text{cross}_{\mathbb{L}\text{-}\mathbb{L}}$
(193, I95)	$\text{disjoint}_{\mathbb{L}\text{-}\mathbb{L}}$, $\text{cross}_{\mathbb{L}\text{-}\mathbb{L}}$, $\text{diverges}_{\mathbb{L}\text{-}\mathbb{L}}$, $\text{diverges}\&\text{cross}_{\mathbb{L}\text{-}\mathbb{L}}$
(190, I495)	$\text{disjoint}_{\mathbb{L}\text{-}\mathbb{L}}$, $\text{cross}_{\mathbb{L}\text{-}\mathbb{L}}$, $\text{diverges}_{\mathbb{L}\text{-}\mathbb{L}}$, $\text{diverges}\&\text{cross}_{\mathbb{L}\text{-}\mathbb{L}}$
(190, I95)	$\text{disjoint}_{\mathbb{L}\text{-}\mathbb{L}}$, $\text{cross}_{\mathbb{L}\text{-}\mathbb{L}}$, $\text{diverges}_{\mathbb{L}\text{-}\mathbb{L}}$, $\text{diverges}\&\text{cross}_{\mathbb{L}\text{-}\mathbb{L}}$
(190, I93)	$\text{disjoint}_{\mathbb{L}\text{-}\mathbb{L}}$, $\text{covers}_{\mathbb{L}\text{-}\mathbb{L}}$, $\text{coveredBy}_{\mathbb{L}\text{-}\mathbb{L}}$, $\text{cross}_{\mathbb{L}\text{-}\mathbb{L}}$, $\text{meet}_{\mathbb{L}\text{-}\mathbb{L}}$, $\text{meet}\&\text{cross}_{\mathbb{L}\text{-}\mathbb{L}}$, $\text{diverges}_{\mathbb{L}\text{-}\mathbb{L}}$, $\text{dv}\&\text{cross}_{\mathbb{L}\text{-}\mathbb{L}}$, $\text{divergedBy}_{\mathbb{L}\text{-}\mathbb{L}}$, $\text{dB}\&\text{cross}_{\mathbb{L}\text{-}\mathbb{L}}$, $\text{dv}\&\text{meet}_{\mathbb{L}\text{-}\mathbb{L}}$, $\text{dv}\&\text{cross}\&\text{meet}_{\mathbb{L}\text{-}\mathbb{L}}$, $\text{dB}\&\text{meet}_{\mathbb{L}\text{-}\mathbb{L}}$, $\text{dB}\&\text{cross}\&\text{meet}_{\mathbb{L}\text{-}\mathbb{L}}$, $\text{dv}\&\text{dB}_{\mathbb{L}\text{-}\mathbb{L}}$, $\text{dv}\&\text{cross}\&\text{dB}_{\mathbb{L}\text{-}\mathbb{L}}$, $\text{dv}\&\text{dB}\&\text{meet}_{\mathbb{L}\text{-}\mathbb{L}}$, $\text{dv}\&\text{cross}\&\text{dB}\&\text{meet}_{\mathbb{L}\text{-}\mathbb{L}}$

For the scenario in figure 7a, we obtained four possible relations between I-93 and I-495— $\text{disjoint}_{\mathbb{L}\text{-}\mathbb{L}}$, $\text{cross}_{\mathbb{L}\text{-}\mathbb{L}}$, $\text{diverges}_{\mathbb{L}\text{-}\mathbb{L}}$, and $\text{diverges}\&\text{cross}_{\mathbb{L}\text{-}\mathbb{L}}$. This solution look reasonable, since we can say from the given knowledge that (i) both endpoints of I-495 do not intersect with I-93 and (ii) one endpoint of I-93 do not intersect with I-495. On the other hand, for the scenario in figure 7b, we obtained as many as 28 possible relations between I-95 and I-495, because from the given knowledge we can only say that I-95 is neither contained nor covered by I-495 and vice versa.

Next, we enrich the previous map by adding two areas—Boston city and the urban area (covering the Boston city). Their spatial arrangement is illustrated in figure 8. Imagine that we have driven three of the four highways and obtained the knowledge about how the three highways connect to other highways and two areas. Based on such knowledge, what can we say about the relation between the remaining highway and two districts? For instance, in figure 9a/b, how I-95/I-93 goes with regard to two districts? To solve this problem, we use $\text{Het9IC}_{\mathbb{R}^2}$, since it concerns the following heterogeneous sets of topological relations:

- topological line-line relations between the visited highways;
- topological line-region between the visited highways and the two areas; and
- topological region-region relations between the two areas— $\text{coveredBy}_{\mathbb{R}\text{-}\mathbb{R}}(\text{Boston}, \text{Urban})$.

In addition, we also use the following optional information to obtain finer solutions:

- topological point-region relations between the highway junctions and the two areas; and
- topological line-point relations between the visited highways and the highway junctions.

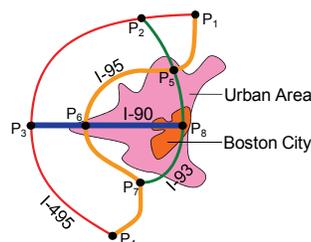


Figure 8. Spatial arrangement of highways and two districts in the Boston metropolitan area.

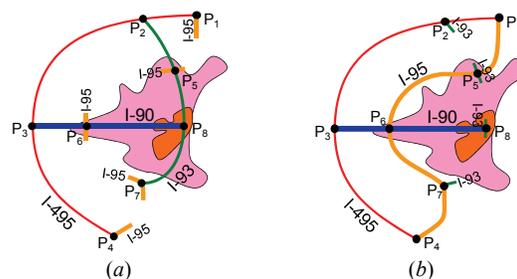


Figure 9. Partial knowledge about the highway network, obtained through the drive on highways except (a) I-95 and (b) I-93.

For actual computation, we put the data of $\text{Het9IC}_{\mathbb{R}^2}$ into SparQ and calculated all consistent scenarios under the constraints described above. We obtained 1 to 16 possible relations for each scenario (table 9). Each solution successfully contains the actual relations in the highway-district arrangement of figure 7a.

For the scenario in figure 9a, we obtained a solution in which I-95 goes through the urban area, and either goes through, touches, or avoids the Boston city. This solution looks reasonable, as we know that (i) both endpoints of I-95 are out of the two districts and (ii) I-95 passes through the urban area. The solutions for I-90 and I-495 also look reasonable. The solution for I-93, however, looks strange. Even though both endpoints of I-93 are located out of the urban area (figure 9b), the derived solution does not filter out such unrealizable relations as $\text{goInto}_{\mathbb{L}\text{-}\mathbb{R}}(\text{I-93}, \text{Urban})$. This is because the current reasoning process does not use the commonsense knowledge that the line's boundary consists of two endpoints, but regard it simply as a point set¹

¹ On the other hand, in figure 9a, the possibility of $\text{goInto}_{\mathbb{L}\text{-}\mathbb{R}}(\text{I-95}, \text{Urban})$ is successfully excluded, because the data already tells that I-95's boundary is completely contained in I-495's interior.

This indicates that we can still improve the reasoning making use of the structural information of spatial objects.

Table 9. Possible relations between an unvisited highway and two districts, derived as algebraically-consistent scenarios.

Unknown relations	Derived solution (gT_{LR} : <i>goThrough</i> _{LR} , $gIBE_{LR}$: <i>goIntoThenBackToEdge</i> _{LR})
[(I-495, Urban), (I-495, Boston)]	$[gT_{LR}, gT_{LR}], [gT_{LR}, touch_{LR}], [gT_{LR}, disjoint_{LR}], [touch_{LR}, touch_{LR}], [touch_{LR}, disjoint_{LR}], [disjoint_{LR}, disjoint_{LR}]$
[(I-95, Urban), (I-95, Boston)]	$[gT_{LR}, gT_{LR}], [gT_{LR}, touch_{LR}], [gT_{LR}, disjoint_{LR}]$
[(I-93, Urban), (I-93, Boston)]	$[gT_{LR}, disjoint_{LR}], [goInto_{LR}, goInto_{LR}], [goInto_{LR}, gT_{LR}], [goInto_{LR}, gIBE_{LR}], [gIBE_{LR}, gT_{LR}], [gIBE_{LR}, gIBE_{LR}]$
[(I-90, Urban), (I-90, Boston)]	$[goInto, goInto]$

7. Conclusions

This paper developed a series of qualitative spatial calculi based on the 9-intersection [5]. These calculi can be used for qualitative spatial reasoning on topological relations between various combinations of objects. Unlike many other calculi, the heterogeneous 9-intersection calculi are concerned with situations where multiple sorts of objects coexist in the same space. However, by integrating sets of base relations, composition tables, and converse lists, such heterogeneity is no longer an obstacle to conduct spatial reasoning in an algebraic framework.

In this work, we featured the 9-intersection because of its popularity in spatial database studies. However, the recent extension of the 9-intersection, called *the 9⁺-intersection* [15, 19], serves as a more flexible framework for modeling topological relations. For instance, the 9⁺-intersection can distinguish the topological relations between two directed lines in \mathbb{R}^1 , which can be mapped to temporal relations between two intervals [15]. Thus, if we extend the 9⁺-intersection into qualitative spatial calculi, the resulting *9⁺-intersection calculi* will cover temporal calculi as well (e.g., Allen’s interval calculus [1]). In addition, the 9⁺-intersection calculi will support topological relations between a directed line and a point/line/region in \mathbb{R}^2 and, accordingly, it will be useful for integrating knowledge about path-landmark arrangements collected by mobile agents. We are planning to develop such 9⁺-intersection calculi and provide a library of both the 9- and 9⁺-intersection calculi on SparQ [8] and CASL [20] for public use.

Another interesting future topic is to develop qualitative spatial calculi that feature non-topological relations (say, cardinal direction relations, relative orientations, distance relations, etc.) between multi-domain objects. We expect that we can use similar integration techniques for the development of such heterogeneous spatial calculi.

Acknowledgments

This work is supported by DFG (Deutsche Forschungsgemeinschaft) through the Collaborative Research Center SFB/TR 8 Spatial Cognition—Strategic Project “Spatial Calculi for Heterogeneous Objects.”

References

- Allen, J.: Maintaining Knowledge about Temporal Intervals. *Communications of the ACM* 26(11), 832-843 (1983)
- Randell, D., Cui, Z., Cohn, A.: A Spatial Logic Based on Regions and Connection. In: Nebel, B., Rich, C., Swarout, W. (eds.): 3rd International Conference on Knowledge Representation and Reasoning, pp. 165-176. Morgan Kaufmann, San Francisco, CA, USA (1992)
- Frank, A.: Qualitative Spatial Reasoning about Cardinal Directions. In: Mark, D., White, D. (eds.): *AutoCarto 10. ACSM/ASPRS* (1991)
- Freksa, C.: Using Orientation Information for Qualitative Spatial Reasoning. In: Frank, A., Campari, I., Formentini, U. (eds.): *International Conference GIS - From Space to Territory: Theories and Methods of Spatio-Temporal Reasoning in Geographic Space, Lecture Notes in Computer Science*, vol. 639, pp. 162-178. Springer, Berlin/Heidelberg, Germany (1992)
- Egenhofer, M., Herring, J.: Categorizing Binary Topological Relationships between Regions, Lines and Points in Geographic Databases. In: Egenhofer, M., Herring, J., Smith, T., Park, K. (eds.): *NCGIA Technical Reports 91-7*. National Center for Geographic Information and Analysis, Santa Barbara, CA, USA (1991)
- Goyal, R., Egenhofer, M.: Consistent Queries over Cardinal Directions across Different Levels of Detail. In: Tjoa, A.M., Wagner, R., Al-Zobaidie, A. (eds.): *11th International Workshop on Database and Expert Systems Applications*, pp. 876-880 (2000)
- Kurata, Y., Shi, H.: RfDL: Models for Capturing Directional and Topological Characteristics of Route-Landmark Arrangements. In: Van de Weghe, N., Billen, R., Kuijpers, B., Bogaert, P. (eds.): *International Workshop on Moving Object* (2008)
- Wallgrün, J.O., Frommberger, L., Wolter, D., Dylla, F., Freksa, C.: Qualitative Spatial Representation and Reasoning in the SparQ-Toolbox. In: Barkowsky, T., Knauff, M., Ligozat, G., Montello, D. (eds.): *Spatial Cognition V, Lecture Notes in Computer Science*, vol. 4387, pp. 39-58. Springer, Berlin/Heidelberg, Germany (2006)
- Gantner, Z., Westphal, M., Wölfl, S.: GQR - a Fast Reasoner for Binary Qualitative Constraint Calculi. In: *AAAI’08 Workshop on Spatial and Temporal Reasoning*. AAAI Press (2008)
- Egenhofer, M., Sharma, J.: Assessing the Consistency of Complete and Incomplete Topological Information. *Geographical Systems* 1(1), 47-68 (1993)
- Cohn, A., Hazarika, S.: Qualitative Spatial Representation and Reasoning: An Overview. *Fundamenta Informaticae* 46(1-2), 1-29 (2001)
- Ligozat, G., Renz, J.: What Is a Qualitative Calculus? A General Framework. In: Zhang, C., Guesgen, H., Yeap, W. (eds.): *PRICAI 2004, Lecture Notes in Artificial Intelligence*,

- vol. 3157, pp. 53-64. Springer, Berlin/Heidelberg, Germany (2004)
13. Zimmermann, K., Freksa, C.: Qualitative Spatial Reasoning Using Orientation, Distance, and Path Knowledge. *Applied Intelligence* 6(1), 49-58 (1996)
 14. Alexandroff, P.: *Elementary Concepts of Topology*. Dover Publications, Mineola, NY, USA (1961)
 15. Kurata, Y.: The 9^+ -Intersection: A Universal Framework for Modeling Topological Relations. In: Cova, T., Miller, H., Beard, K., Frank, A., Goodchild, M. (eds.): *GIScience 2008*, Lecture Notes in Computer Science, vol. 5266, pp. 181-198. Springer, Berlin/Heidelberg, Germany (2008)
 16. Egenhofer, M.: Deriving the Composition of Binary Topological Relations. *Journal of Visual Languages and Computing* 5(2), 133-149 (1994)
 17. Egenhofer, M.: Spherical Topological Relations. *Journal on Data Semantics III*, Lecture Notes in Computer Science, vol. 3534, pp. 25-49, Springer, Berlin/Heidelberg, Germany (2005)
 18. Vasardani, M., Egenhofer, M.: Single-Holed Regions: Their Relations and Inferences. In: Cova, T., Miller, H., Beard, K., Frank, A., Goodchild, M. (eds.): *GIScience 2008*, Lecture Notes in Computer Science, vol. 5266, pp. 337-353. Springer, Berlin/Heidelberg, Germany (2008)
 19. Kurata, Y., Egenhofer, M.: The 9^+ -Intersection for Topological Relations between a Directed Line Segment and a Region. In: Gottfried, B. (ed.): *First International Workshop on Behavioral Monitoring and Interpretation, TZI-Bericht*, vol. 42, pp. 62-76. Technologie-Zentrum Informatik, Universität Bremen, Germany (2007)
 20. Astesiano, E., Bidoit, M., Kirchner, H., Krieg-Brückner, B., Mosses, P., Sannella, D., Tarlecki, A.: CASL: The Common Algebraic Specification Language. *Theoretical Computer Science* 286(2), 153-196 (2002)

Spatio-temporal aspects of the monitoring of complex events

G rard Ligozat ^{†‡}
ligozat@limsi.fr

Zygmunt Vetulani [‡]
vetulani@amu.edu.pl

J drzej Osiński [‡]
josinski@amu.edu.pl

[†] LIMSI, Paris-Sud University, France
[‡] Adam Mickiewicz University, Poznań, Poland

Abstract

This paper discusses the problem of representing and reasoning about the spatio-temporal aspects of events in the context of the monitoring of complex, multi-agent events, using natural language input. The proposals discussed in the paper are used for the development of a system providing assistance to the security staff during a large scale event involving a large number of participants. A subsystem deals with the SMS messages sent by the security staff (observers) and processes the information they convey. The system provides the security personnel (analysts) with visualisation facilities and suggestions for actions, and keeps track of the information exchanged for further use. The paper presents arguments in favor of the formalism called XRCDC (an extension of the Region Cardinal Direction Calculus) designed to represent events and spatio-temporal reasoning about them.

Keywords spatio-temporal reasoning, computer understanding of natural language

1. Introduction

The general context of the research

Enhancing the processing of information in real-life situations where the decision making depends on the quality and adequateness of the information acquired by the deciding person or body is an important problem. Typical cases are situations involving the presence of large concentrations of people sharing strong emotions and feelings, and participating in events spanning extended periods of time. In such situations, an early detection and diagnosis may be decisive for taking adequate preventive actions.

In those situations, the organizers and/or specific services (e.g. the police) are responsible for setting up an appropriate *security structure*. Typically, such a structure involves at least three categories of agents: observers, analysts (or decision makers), and operating forces.

The role of human observers is to keep close attention to what is happening around them and to inform the analysts at a central crisis management center (CMC) of any unusual or menacing event they may have observed. Observers communicate with analysts using appropriate communication channels. The limitations in capacity of those

channels, as well as the structure of the deciding body, may result in incapacitating bottlenecks in case of critical situations (such as a large number of simultaneously incoming messages from different observers). The possible existence of contradictory pieces of information, and the impossibility of interpreting all messages in real time may lead to serious (sometimes critical) misinterpretations of the incoming messages. Therefore, a partial or total automatization of the information processing task can be considered to constitute a very desirable goal.

In this paper, we focus on the particular case of the monitoring of sports events, such as large public soccer matches, where potentially dangerous events can occur at hardly predictable moments. Because of the particular circumstances (a noisy and hostile environment) a privileged way of communication is provided by the use of SMS messages. This requires the development of a specific subsystem using natural language short messages as input. The task of the system consists in spying on the flow of natural language information from the observers to the central management center (and in some cases in interacting with the users) and in interpreting and processing information (including doing consistency checking and visualisation). The overall structure of the system¹ is illustrated in Fig. 1.

The main focus of this paper is on the methods used by the system for processing the spatial and temporal information. The central idea is to process the information conveyed by the messages in terms of events. Those events are located at various locations, and they occur at definite moments of the general event. Hence representing the spatial and temporal aspects of the information to be processed is one of the central functionalities of the monitoring system.

The general considerations of the paper are illustrated by the study case of a soccer match for which the system is being implemented. The interest to consider this study case as an illustration of a much more general situation is clear considering its complex but transparent information structure (see section 2).

¹The system presented here (POLINT-112-SMS) is an extension of the original POLINT system, a question-answering system developed for the Polish language (Vetulani 1995). Notice however that the considerations in this paper are meant to be language-independent.

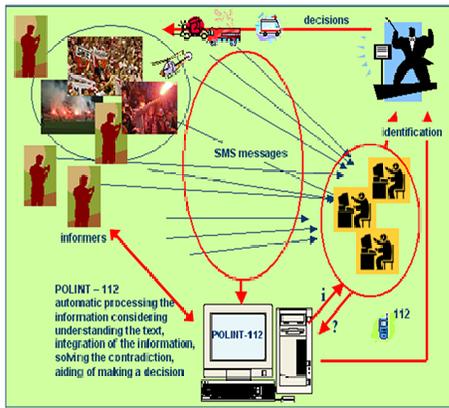


Figure 1: The POLINT-112-SMS system (Vetulani et al. 2008)

The spatio-temporal component

As already mentioned, the main function of the system is to provide information to the CMC and to assist the decision makers in their decisions about security matters. These decisions are based upon the knowledge gathered by the informers and on the inferences and conclusions that may be derived from it. As a consequence, concerning the specific aspect of temporal and spatial knowledge, the system is supposed to allow to represent the information gathered from the sources and to perform necessary reasonings.

Reasoning

The reasoning activity may involve various tasks, such as deciding the right time for a specific action, identifying the components of a complex event and characterizing its temporal structure, deciding that events (in a sequence of events) may be causally related (e.g. an explosion followed by a movement of the crowd) or not (the same events in the reverse order). A specific reasoning activity is also required by the component of the system that provides the user with visualisation tools (to be discussed more in detail below).

Main components of the representation

The representation formalism is based upon a set of primitive notions:

- events, which are spatio-temporal entities with a temporal span (an interval) and a spatial extent (a region);
- relations between events, especially qualitative relations, including purely temporal relations (such as precedence) and purely spatial relations (such as inclusion or overlapping);
- anchoring relations, which provide anchors from the events to a referential 3D space (spatial anchoring) and a time line (temporal anchoring);
- individuals, which may take part in the events with a specific role (witness, actor, victim, etc.);
- groups of individuals.

Events, which are the central notion in this framework, are complex entities. From the point of view of the spatio-temporal model, their temporal span and spatial extent are primary properties used in reasoning. For instance, two events such as : X insulting Y and : Y striking X will be candidates for a causal interpretation if the temporal span of the first event immediately precedes that of the second, and the spatial extents of both events are in a relation of proximity. This of course will require that the two individuals referred to as X and Y are correctly identified as participants in those events with suitable roles.

Further requirements

The notion of an individual participating in a given event is of a complex nature: different individuals may participate in an event (for instance a war) during various periods which do not necessarily coincide with the temporal span of the event itself. The system has to provide suitable ways of dealing with this fact.

One function of the system is to suggest preventive actions to be taken. Subsequently, decisions about taking those actions can be made. The system, and in particular its visualisation component, should therefore allow the possibility of representing the consequence of a decision which has not yet been made (such as a special color or mode of appearance on the screen).

A further step toward the representation of non-existing events is to provide the user with tools for hypothetical reasoning (for instance, based on not yet implemented decisions).

Partial or fuzzy knowledge about events also has to be accommodated. As will be discussed below, many qualitative formalisms allow the representation of partial knowledge about relations between events, and we use this facility in the system. Suitable ways of visually representing fuzzy or partially determined events have to be determined.

2. The spatio-temporal component: general principles

The input is represented in a conceptual space (Gärdenfors 2004) of events which possesses a spatial and a temporal dimension.

Since the point of view of various informers may give various kinds of biases to the pieces of knowledge, the system has the capability of representing various views of a situation: the view of the system, and the views of various informers. According to the view considered, various degrees of reliability can be assigned according to the informer.

The static structure of a stadium

A stadium as a static element has a well-defined structure with hierarchical aspects: there is the playing field (but see below) and there are tribunes which are subdivided into sectors, ranks, exits, etc. Most of the action, from the point of view of security, is likely to happen outside the playing field (excepting the rare possibility that groups of supporters or spectators enter the playing field). So the main locations

will refer to regions in the part of the stadium where spectators and supporters are present.

The *a priori* knowledge

The situation of a soccer game involves a great deal of *a priori* knowledge about what a game is and what has to be expected from the supporters (and players) in a “normal”, undisturbed game, in order to detect the abnormal, potentially dangerous situations as soon as possible.

As a process, a soccer game has a well-defined structure, which implies consequences for the temporal and spatial structures: the teams are supposed to play for well-defined durations at well-defined locations, one of the main distinctions being between the two periods. Each event has to be inserted in this pre-existing frame, and part of the reasoning and decisions to be taken depend on the particular temporal environment of this event.

The system of representation of knowledge has to integrate all these elements. Various aspects have to be considered and represented using adequate formalisms.

3. Active map (AM) and initial events (IE)

Our proposal is that the processing of spatio-temporal information in the system is mediated through the maintenance of an animated visualisation component which we call the active map (AM).

The active map allows the representation of events based on a fixed background representing (in the case of information concerning the stadium) the stadium itself. On this fixed background gradually appear representations of events as the information accumulates due to the processing of the messages. Each new message either generates a new element of representation, or adds information to a pre-existing one. In the first case, this will trigger the generation of an initial event (IE) with a set of features associated to it.

As an illustrative example, assume that the system has to process the following message: *Some supporters in sector 4 are throwing rocks at the security personnel.*

This message triggers the creation of an initial event with this text as an associated text, and the following set of features:

- source (the observer)
- actors (acting agents: supporters; patients: security personnel)
- location (sector 4)
- temporal span (time during which the event happened)
- time of reception (time of reception of the message)
- action (type of action: throw; instruments: rocks)
- aspectual type (on-going event)

4. Introducing new events in the active map

The system receives a sequence of messages. For each new message, either a new Initial Event is created, and then if co-references are detected to some previous initial event the decision has to be made to graft the new information onto this pre-existing event, hence adding new information to it.

As an example of grafting, let us consider the following message by the same informers: *Some among them are wielding baseball bats.* First a new IE is created :

- source (the informer)
- actors (agents: subset of supporters; patients: none)
- location (sector 4)
- temporal span (coincides with that of the previous event)
- time of reception of the message
- action (type of action: wield; instruments: baseball bats)
- aspectual type (on-going event)

Here the system detects co-references:

- the same source;
- actors: the agents are subset of the original set of agents, there are no patients;
- the same location
- the same aspectual type of action

So, the system could decide to add new features (for example : presence of potential weapons) to the representation of the first IE in the active map.

5. Using the active map

The overall goal of the spatio-temporal component is to assist decision making from the part of the security personnel.

Events are present in the active map in various forms. In order to visualize the kind of tool the active map constitutes, one might think of the traditional maps used by the army. In the General Headquarters of World War II, officers used pins, small flags or any other gadgets indicating the locations and moves of the various troops in order to evaluate the situation and to take appropriate decisions. The AM is meant as a sophisticated version of such traditional maps. The idea of building active maps is closely related to previous work on the visualisation of military campaigns described in the literature (Ligozat, Nowak, and Schmitt 2007). Here, the active map is based on a static background representing the stadium. Events are represented by schematic images, but also can have sounds, colored lights, blinking lights associated to them. These representations can be interrogated *e.g.* clicking on them would provide the original texts which resulted in the generation of the initial events (still to be implemented). The active map also provides zooming facilities which allow the user to “take a closer look” at local situations if necessary.

6. Requirements for the spatio-temporal components

As already mentioned above, the languages used to represent the temporal and the spatial structures of the events have to allow the representation of qualitative or indeterminate information. Such formalisms as those derived from Allens calculus (Allen 1983), such as the rectangle calculus or the region cardinal direction calculus have this property.

The choice of a suitable language of representation depends on making decisions about the parameters of the temporal and spatial representation which imply answering the following questions:

1. Ontological questions: of what type are the objects to be represented? Can they be abstracted as points, lines, regions with geometrical shapes, connected regions, to cite only a few possible choices?
2. Nature of the information: is it quantitative (the kind of information a robot receives from its sensors) or predominantly qualitative (as is mainly the case with the information carried by natural language)?
3. Nature of the surrounding space: can we make use of global systems of reference (like north, south, for instance) or do we have to be content with local frames of reference (typically, the frame of reference represented by some moving person or object)?
4. What is the dimension of the surrounding space? Can we reason in 2D, in an augmented version of 2D (for instance with a finite number of 2D levels), or do we need a full 3D space?
5. Nature of the kind of spatial relations we want to represent. If one looks at the state of the art in the domain of qualitative spatial reasoning, three main types of relations have been predominantly studied: topological information, that is relations such as containment, partial overlap, having adjacent boundaries, or disjointedness; directional information, with respect to some frame of reference; and qualitative distance information (near, far, very far).
6. Is time continuous or discrete? In the representation of linguistic data, continuity is usually assumed (since it is a property of language that it is able to “open up” any event and re-consider a previously punctual situation and present it in a second consideration as an extended one). It has to be remarked, however, that many formalisms can accommodate both a continuous and a discrete interpretation. This is in particular the case of Allens calculus.
7. Questions linked to the way of anchoring the abstract model to the actual situation: for instance, a match involves absolute temporal landmarks (beginning of the game, end of the game, half-period, additional time) as well as occasional, or contingent temporal landmarks (important events of the game such as goal, penalty, and so on).

7. Choosing a formalism

Since the input to the system is of a linguistic nature (written or spoken natural language), it is to be expected that most of the spatial information to be processed will be of a qualitative nature; even such expressions as *about 50 meters from me* may be interpreted in a qualitative, imprecise sense.

The entities to be represented are individuals, groups, or regions. One can argue that an individual can be seen as (small) region in which this person can be located at a given time. As for the temporal dimension, Allens calculus is

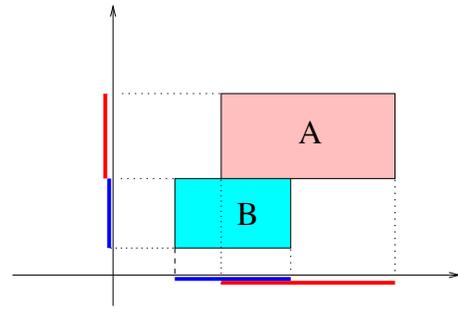


Figure 2: The rectangle calculus

based on intervals, but time points can also be easily accommodated, using for instance the point-and-interval calculus.

The environment, be it inside the stadium or around it, is very clearly structured by the presence of the stadium and its various subdivisions, as well as by the surrounding parts of the city. It seems reasonable, for this reason, to use a global frame of reference, most of the events being presumably located with respect to the frame of reference offered by the environment.

Because of the presence of this fixed frame of reference, formalisms using directional information may be given a preference. Such are the rectangle calculus, a 2-dimensional extension of Allens calculus (first proposed by Guesgen in (Güsgen 1989)), and the region cardinal direction calculi.

8. The rectangle calculus and the region cardinal direction calculus

The rectangle calculus

The rectangle calculus considers objects which are rectangles whose sides are parallel to the axes. Given such a rectangle as a reference, the qualitative position of any other rectangle can be described using the projections on the axes of coordinates, which are intervals (Fig. 2). Hence those positions are described by a pair of Allens relations. For instance, in Fig. 2, rectangle *A* is in relation (oi, mi) with respect to rectangle *B*, since the horizontal projection of *A* is overlapped by that of *B* (Allens *oi* relation) while the vertical projection of *A* is met by that of *B* (Allens relation *mi*).

One obtains in this way a formalism whose composition table, which is the main tool for propagating knowledge, is basically Allens table. The formal properties of the calculus have been extensively studied by Balbiani, Condotta, and Fariñas del Cerro (Balbiani, Condotta, and Fariñas del Cerro 1998; Balbiani, Condotta, and Fariñas del Cerro 1999).

The rectangle calculus can be easily extended to a calculus about *rectangles, points and lines*, with the restriction that the lines have to be parallel to the axes. For objects having other shapes, the simplest method consists in replacing them by their minimal bounding rectangle. For instance, in the case of Fig. 3, the two regions *A* and *B* have two minimal bounding rectangles $mbr(A)$ and $mbr(B)$, and their relative position can be encoded as a rectangle relation.

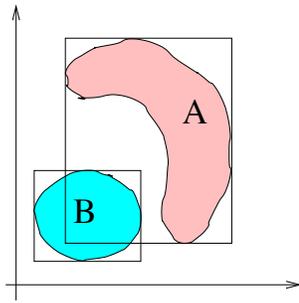


Figure 3: Two objects and their minimal bounding rectangles

The drawback is that much information can be lost in this way. For instance, disjoint objects may have overlapping rectangles. This is illustrated in the same figure, where two objects A and B which are disjoint have overlapping bounding rectangles.

The Region Cardinal Direction Calculus

The basic cardinal calculus deals with points in 2D space and the eight basic cardinal directions N, S, E, W, NE, SE, NW, SW, augmented by the identity relation eq. Actually, the basic cardinal direction calculus is a 2D version of the time-point calculus, in the same way as the rectangle calculus is a 2D version of Allens calculus. It has also been studied extensively and its formal properties are well known (Frank 1992; Ligozat 1998).

In order to deal more precisely with arbitrary extended regions in 2D space, an extension of the cardinal direction calculus, called the RCDC (region cardinal direction calculus) has been proposed by Goyal and Egenhofer (Goyal and Egenhofer 2001) as well as by Skiadopoulos and Koubarakis (Skiadopoulos and Koubarakis 2001; 2004).

The starting point of the representation consists in considering the nine regions (called tiles) defined by the minimal bounding rectangle of a reference object B (Fig. 4). Eight of them, labelled N, S, E, W, NE, SE, NW, SW, are infinite regions. The minimal bounding rectangle itself, labelled O,

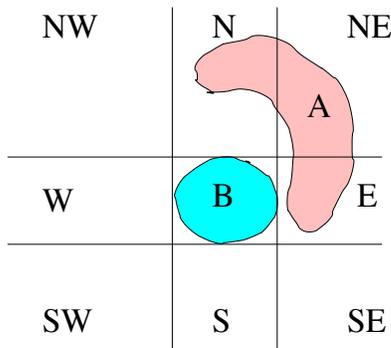


Figure 4: The region cardinal direction calculus

constitutes the ninth tile. In this way, the position of any region A can be described by listing the set of tiles which intersect the interior of A . Alternatively, it can be described by a boolean array $dir(A, B)$ of length 9 listing whether A has a non-empty intersection (denoted by 1) or an empty intersection (denoted by 0) with each of the nine tiles NW, N, NE, W, O, E, SW, S, SE, in that order. For instance, in Fig. 4, the relation of A with respect to B is (N:NE:E), or, using the boolean array $dir(A, B) = [0, 1, 1, 0, 0, 1, 0, 0, 0]$.

The RCDC possesses very interesting properties: it deals with (connected) regions in the plane and, although still based on the same basic relations, allows a finer expression of the relative positions of more general regions.

Because of its extended base, the RCDC also implicitly encodes topological information (as does the rectangle calculus). For instance, referring back to Fig. 4, the fact that A and B do not intersect is a consequence of the stronger fact that A does not intersect the tile containing B , and this fact is encoded in the representation (O does not belong to (N:NE:E)). Moreover, the RCDC may be considered as a refinement of the rectangle calculus. Recent results (Zhang et al. 2008) show that basically it allows to consider a spatial configuration as “pixelized” by the objects present in the environment, so that those “pixels” can be used to characterize a “silhouette” of each object which is finer than its bounding rectangle. This means also in particular that, if we decide to replace the objects by their minimal bounding rectangles, we get in substance the rectangle calculus.

9. Representing spatio-temporal information

We use the RCDC for the representation of spatial relations, and we add to it a temporal dimension, which is also integrated to the framework of the RCDC, yielding a calculus called the XRCDC, the Extended Region Cardinal Direction Calculus. An advantage of this choice is that space and time are treated in the same way and that the corresponding formalisms have been studied in depth. In particular, their theoretical complexity has been determined, and a whole range of algorithms has been developed to solve the basic reasoning problems. An important point is that using the XRCDC formalism allows us to process information more precisely than using the rectangle calculus. Let us analyse the following example: *The fight started during the meeting, to the south of and touching the place of the meeting. After the meeting finished, the fight had also extended to the meeting place.* Fig. 5 (a) and Fig. 5 (b) show a graphical interpretation of the scene in projection on one of the spatio-temporal planes (the T-S plane) using the XRCDC formalism and the rectangle calculus respectively.

It is clear that using techniques based on the extension of Allens algebra may cause distortions and even errors, which is a critical shortcoming for a public security application. For example, interpreting Fig. 5 (b) we may conclude that the fight and the meeting were localized in the same place for some period of time, which is not true (*cf.* Fig. 5 (a)) and may lead to inappropriate decisions.

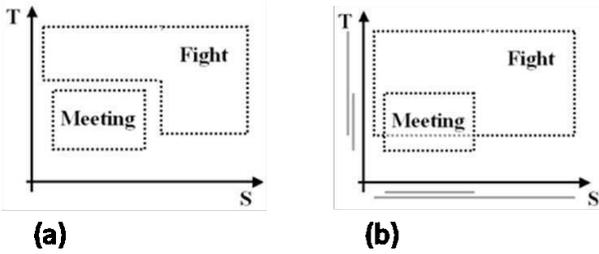


Figure 5: Comparison of two techniques: (a) the XCDC, (b) the rectangle calculus

An example of the use of the XRCDC

The following example shows how the XRCDC formalism, as described in (Osinski 2009), is applied within the POLINT-112-SMS system.

Let us consider three objects located in the analysed area (part of a stadium) represented in the active map (Fig. 6). The information about these object is expressed in natural language by the following English sentences: *Sector A is to the north with respect to sector B and touches it. Sector A is to the east with respect to the soccer ground and touches it. Sector B is to the east of the soccer ground and touches it.* We also assume that the soccer ground was chosen to be the central point of the map.

Now suppose that a new text message is sent by an informer to the security system: *The meeting took place in front of sector A (between A and the soccer ground), close to sector A. The fight started just after the meeting, to the south with respect to the meeting place, and close to this place.* In the first step of the processing the phrase “in front of” is interpreted as the relative direction relation *r*-in-front, the sentence “to the south of” as the absolute direction relation *r*-south, the phrase “just after” as the time relation *r*-after and time distance *r*-right, and finally the “close to” and “touching” as the distance relation *r*-close and *r*-touch respectively. The objects are represented by their unique identifiers, *id*-fight, *id*-meeting and *id*-sectorA. Then the second step is the generation of a *ts*-relation structure. We denote by $\text{dir}[D1,D2](X,Y)$ the direction-relation matrix for the projection on the $(D1,D2)$ plane representing the localization of *X* with respect to *Y* (e.g. for Fig. 4, $\text{dir}(B,A)=[0,1,1,0,0,1,0,0,0]$). The qualitative parameters *s*-dis and *t*-dis represent qualitative spatial and temporal dis-

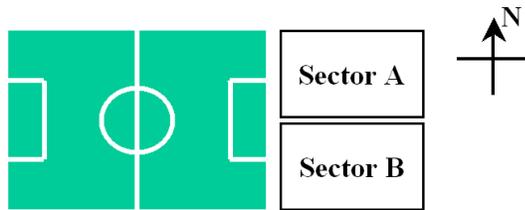


Figure 6: The analysed area in the projection on the soccer ground plane

tances.

Other elements of the *ts*-relation structure (in particular the direction-relation matrix) are generated using predefined conversion rules. In the case of the *r*-in-front-of relation it is necessary to indicate the absolute direction relation between sector A and the soccer ground which is the central point of the map. This allows us to infer the absolute direction relations between the meeting place and sector A from the relative ones. In the case where information is missing (or is not precise enough) we have to take into account all possible situations. In the example analysed here, conversion rules describe both *r*-south and *r*-after relations as independent and fill all the tiles in direction-relation matrices. Finally the two following *ts*-relation structures are added to the system knowledge base: *ts*-relation(*id*-meeting, *id*-sectorA, [0,0,0,1,0,0,0,0,0], [0,0,0,0,0,1,0,0,0], [0,1,0,0,1,0,0,1,0], [0,1,0,0,1,0,0,1,0], [1,0,0,1,0,0,1,0,0], [0,0,1,0,0,1,0,0,1], 3, 0), and *ts*-relation(*id*-fight, *id*-meeting, [0,0,0,0,0,0,0,1,0], [0,1,0,0,0,0,0,0,0], [1,0,0,0,0,0,0,0,0], [0,0,0,0,0,0,0,0,1], [0,1,0,0,0,0,0,0,0], [0,0,0,0,0,0,0,1,0], 3, 2). It is worth noticing that before any new relation is added to the knowledge base a consistency check is performed in order to check whether or not this relation is consistent with the set of relations already stored in this knowledge base. If the new relation is incompatible, the old relation will be removed as we assume that in a dynamic environment a new information (if it comes from a reliable informer) is always to be considered as more truthful than the old one. (This is a strong assumption which may have to be modified according to the context.)

Now suppose the new text message comes from another informer: *I am in sector B. Where was the meeting?* Leaving aside the new information to be added into the system knowledge (actual position of the informer), we can interpret this message as a question about the space relation between the two considered entities: the fight and sector B. In this situation the *get-space-relation* algorithm is applied. In the first step it looks for *ts*-relation(*id*-meeting, *id*-sectorB,) or *ts*-relation(*id*-sectorB, *id*-meeting,) structures. There this search fails as there is no information of this type in the system knowledge base. However this knowledge base contains the following two structures: *ts*-relation(*id*-sectorA, *id*-sectorB, [0,1,0,0,0,0,0,0,0], [0,0,0,0,0,0,0,1,0], [0,0,1,0,0,1,0,0,1], [1,0,0,1,0,0,1,0,0], [0,1,0,0,1,0,0,1,0], [0,1,0,0,1,0,0,1,0], 2, 0) and *ts*-relation(*id*-meeting, *id*-sectorA, [0,0,0,1,0,0,0,0,0], [0,0,0,0,0,1,0,0,0], [0,1,0,0,1,0,0,1,0], [0,1,0,0,1,0,0,1,0], [1,0,0,1,0,0,1,0,0], [0,0,1,0,0,1,0,0,1], 3, 0).

These structures can be used in the second step of the *get-space-relation* algorithm performing the composition of cardinal direction relations. Composition is realized for both $\text{dir}[N,E]$ matrices in the *ts*-relation structure definition. As a result we get the following direction relation matrices: $\text{dir}[N,E](\text{id-meeting, id-sectorB}) = [1, 1, 0, 0, 0, 0, 0, 0, 0]$, $\text{dir}[N,E](\text{id-sectorB, id-meeting}) = [0, 0, 0, 0, 0, 0, 0, 1, 1]$. Now, using the appropriate conversion rule, these matrices can be translated into the absolute direction relation *r*-north-west. The distance relations are composed also by analysing the $\text{dir}[N,E](\text{id-meeting, id-sectorB})$ matrix. As a result we

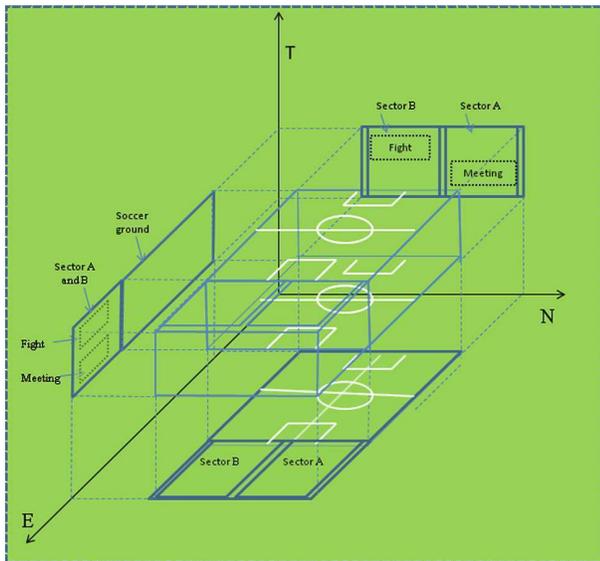


Figure 7: The graphical interpretation of the knowledge about the events collected in the system consists of (a) the projection on the north-east plane, (b) the projection on the time-north plane, (c) the projection on the time-east plane

get the relation *r*-far. The message received by the informer would correspond to the following sentence: *The fight took place to the north-west and far from sector B.*

Similarly we can ask for the time relation between the fight and the meeting. In the example discussed the answer may be found in the first step of searching as the appropriate *ts*-relation structure is already in the database. However the relation can be also computed using the composing algorithm if necessary. In such a situation the composition is realized for the $\text{dir}[T,N]$ and $\text{dir}[T,W]$ matrices in the *ts*-relation structure.

Sometimes adding information about the ensuing event may be not enough for the visualisation module. In such a situation the visualisation module must request the exact (absolute) location of the event. Receiving an answer (from other system modules or from a human operator) exempts it from reasoning. From the point of view of visualisation the precise localization of events is necessary, otherwise visualisation could display a distorted view of the situation. In case of a lack of precise information it is necessary to adjust the image from the visualisation module to the precision level the system has (e.g. we can say that *X* is in the stadium if we do not know in which sector it is exactly). Solutions of this type also protect us from the situation where the *get-space*-relation algorithm would not be able to find the specific relation. The system executes the *find-location* algorithm which uses a predefined ordered list of objects and relations. They determine the sequence in which objects should be searched (first look for seats, then entrances, communication routes, sectors, groups, people, and if all this fails choose the whole stadium as a reference point) and which relations should be chosen (prefer the *r*-in relation and small distances). In

the example analysed, the reference point would be sector A with the direction relation *r*-west and distance relation *r*-close.

Analysing concurrently time and space relations can also help us to discover deeper dependencies between objects. For example, as we know that the fight was located very close to the place of the meeting and started just after it, we may assume that the fight was initiated by people attending the meeting.

10. Conclusions

We have discussed the questions raised by the development of a spatio-temporal module for representing and reasoning about the events involving the participation of a large number of people (such as an international soccer match in presence of an emotional crowd) in the context of the construction of a system assisting the security personnel. We have examined various requirements of such a construction, and proposed a number of solutions. In particular, we have designed and implemented a visualisation facility, the active map, which involves the use of the XRCDC formalism, a temporal extension of Region Cardinal Direction Calculus.

References

- Allen, J. F. 1983. Maintaining knowledge about temporal intervals. *Comm. of the ACM* 26(11):832–843.
- Balbiani, P.; Condotta, J.-F.; and Fariñas del Cerro, L. 1998. A model for reasoning about bidimensional temporal relations. In Cohn, A. G.; Schubert, L.; and Shapiro, S. C., eds., *Proceedings of the Sixth International Conference on Principles of Knowledge Representation and Reasoning (KR'98)*, 124–130. Morgan Kaufmann.
- Balbiani, P.; Condotta, J. F.; and Fariñas del Cerro, L. 1999. A new tractable subclass of the rectangle algebra. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence*, 442–447. Morgan Kaufmann Publishers.
- Frank, A. U. 1992. Qualitative spatial reasoning about distances and directions in geographic space. *J. of Visual Languages and Computing* 3:343–371.
- Gärdenfors, P. 2004. *Conceptual Spaces: The Geometry of Thought*. Cambridge/London: Bradford/MIT Press.
- Goyal, R. K., and Egenhofer, M. J. 2001. Similarity of cardinal directions. In Jensen et al. (2001), 36–58.
- Güsgen, H. 1989. Spatial reasoning based on Allen's temporal logic. Technical Report TR-89-049, ICSI, Berkeley, Berkeley, CA.
- Jensen, C. S.; Schneider, M.; Seeger, B.; and Tsotras, V. J., eds. 2001. *Advances in Spatial and Temporal Databases, 7th International Symposium, SSTD 2001, Redondo Beach, CA, USA, July 12-15, 2001, Proceedings*, volume 2121 of *Lecture Notes in Computer Science*. Springer.
- Ligozat, G.; Nowak, J.; and Schmitt, D. 2007. From language to pictorial representations. In *Proceedings of the Language and Technology Conference (L&TC'07)*, Poznań, Poland, September 5-7, 2007. AAAI Press.

- Ligozat, G. 1998. Reasoning about cardinal directions. *J. Vis. Lang. Comput.* 9(1):23–44.
- Osinski, J. 2009. Extending the Cardinal Direction Calculus to a Temporal Direction. In *Proceedings of the FLAIRS-2009 Conference*.
- Skiadopoulos, S., and Koubarakis, M. 2001. Composing cardinal direction relations. In Jensen et al. (2001), 299–320.
- Skiadopoulos, S., and Koubarakis, M. 2004. Composing cardinal direction relations. *Artif. Intell.* 152(2):143–171.
- Vetulani, Z.; Marciniak, J.; Konieczka, P.; and Walkowska, J. 2008. An SMS-based system architecture (logical model) to support management of information exchange in emergency situations. POLINT-112-SMS PROJECT. In Shi, Z.; Mercier-Laurent, E.; and Leake, D., eds., *Proc. of Intelligent Information Processing IV (IIP IV)*, 240–253. Springer.
- Vetulani, Z. 1995. POLINT - system automatycznej interpretacji pytań w języku polskim i jego realizacja w PROLOGU (POLINT - a system for the automatic interpretation of Polish questions and its implementation in PROLOG). In Pogonowski, J., ed., *Euphonia i Logos, Księga pamiątkowa ofiarowana Profesorowi Marii Steffen-Batogowej i Profesorowi Tadeuszowi Batogowi*, 583–598. Wyd. Naukowe UAM, Poznań.
- Zhang, X.; Liu, W.; Li, S.; and Ying, M. 2008. Reasoning with cardinal directions: An efficient algorithm. In Fox, D., and Gomes, C. P., eds., *AAAI*, 387–392. AAAI Press.

H-Complexity Metric for Qualitative Spatial and Temporal Reasoning Applications

Carl Schultz, Robert Amor

Department of Computer Science, The University of Auckland
Private Bag 92019, Auckland, New Zealand
csch050@aucklanduni.ac.nz, trebor@cs.auckland.ac.nz

Hans Guesgen

School of Engineering and Advanced Technology, Massey University
Private Bag 11222, Palmerston North, New Zealand
h.w.guesgen@massey.ac.nz

Abstract

Commonsense reasoning, in particular qualitative spatial and temporal reasoning (QSTR), provides flexible and intuitive methods for reasoning about vague and uncertain information including temporal duration and ordering, and spatial orientation, topology and distance. Despite significant theoretical advances in QSTR, there is a distinct absence of applications that employ these methods. The central problem is a lack of application-level standards and metrics that developers can use to measure the effectiveness of their QSTR applications. To address this we present a fundamental metric called *H*-complexity that quantifies the expressiveness of QSTR systems according to the number of distinct scenario classes that can be encoded. In this paper we show that *H*-complexity can be employed in a range of powerful and practical ways that support QSTR application development. To illustrate this, we present two examples: calculating test coverage for validation, and quantifying the reduction in expressiveness due to constraints. We thereby demonstrate that *H*-complexity is a useful tool for determining whether a QSTR system meets the needs of a specific application.

Introduction

Commonsense reasoning aims to address the limitations of purely numerical systems, by providing coarser and more intuitive knowledge representation and reasoning techniques (Kuipers, 1994). The subdiscipline of qualitative spatial and temporal reasoning (QSTR) aims to formalise our intuitive understanding of everyday physical relationships such as size, orientation, topology, and distance, and temporal relationships such as ordering, coincidence, and duration (Cohn and Renz, 2007). A seminal example of QSTR is Allen's interval calculus (Allen, 1983) which defines a set of thirteen atomic

relations between time intervals, and an algorithm for reasoning about networks of temporal relations, e.g. (where \bullet is a composition operator)

t_1 before $t_2 \bullet t_2$ contains $t_3 = t_1$ before t_3

t_1 overlaps $t_2 \bullet t_2$ during $t_3 =$

t_1 (overlaps, or during, or starts) t_3

Despite significant theoretical advances, there is a distinct absence of applications that make use of these techniques (Dylla et al., 2006). The central problem is the significant lack of support for adapting and integrating existing QSTR methods with other domain specific qualitative spatial and temporal models. Specifically, there are no methods for assessing the quality of a QSTR system or comparing QSTR systems to determine which one is the most effective for solving a given problem.

Most of the existing research on QSTR analysis has focused on proving correctness of composition (Wölfel et al., 2007), characterising reasoning complexity (Vilain et al., 1989) and determining tractable subsets of well known formalisms (Nebel and Bürckert, 1995). While it is important to know when a problem is NP complete, reasoning complexity cannot be used by the developer to determine whether the particular QSTR system addresses the task at hand, how a change to the system will impact its effectiveness, or how QSTR application validation can be performed to ensure that the application is fit for purpose.

To address this we present a fundamental metric called *H*-complexity that quantifies the expressiveness of a given QSTR system, in terms of the number of distinct scenario classes that can be encoded by the system. The developer can use expressiveness to guide application design in a range of powerful ways, for example:

- (i) comparing different QSTR systems to help determine whether one is more suitable than the other;
- (ii) calculating test coverage during the validation phase of development to quantify confidence in the application being fit for purpose;

(iii) quantifying the reduction in expressiveness due to constraints to assist in developing a test plan.

In this paper we derive a very simple equation for calculating H -complexity, and we consider two applications addressing points (ii) and (iii) above, where we derive very simple equations and a reference table for improved runtime efficiency. We thereby demonstrate that our complexity metric is a flexible and effective tool for supporting the development of QSTR applications.

The remainder of the paper is structured as follows. In the following section we review the basic theory of QSTR systems. We then derive H -complexity in two steps. Firstly we define the concept of homogeneous sets as the fundamental folding of objects into equivalence classes permitted by the QSTR language. We then use homogeneous sets to quantify QSTR expressiveness by identifying a one-to-one relationship between accessible, unique subsets and scenario classes. In the succeeding section we derive methods for calculating homogeneous sets in relations of any arity. We then present two methods for employing H -complexity: calculating test coverage for validation, and quantifying the reduction in expressiveness due to constraints. In the final section we present the conclusions of this paper.

Foundations of QSTR

Informally, QSTR applications model, infer, and check the consistency of object relations in a scenario. We will define QSTR applications in terms of model theory (Marker, 2002; Hodges, 1997) and then define the roles of QSTR application designs and users.

We use the notation \uparrow to represent the exponent operator, $x\uparrow y = x^y$. In model theoretic terms, a language L (or *vocabulary*, or *signature*) is a finite set of relation symbols \mathbf{R} and arities a_R for each $R \in \mathbf{R}$. A model \mathbf{M} of language L (or L -*structure*, or *interpretation*) consists of a *universe* U (or *domain*, or *underlying set*) and for each relation symbol $R \in \mathbf{R}$ there is a set $R_M \subseteq U \uparrow a_R$. That is, \mathbf{M} provides a concrete interpretation of the symbols in L based on the underlying set U . Finally, a *scenario* (or *configuration*, or *substructure*) is a model \mathbf{V} that can be *embedded* into \mathbf{M} , that is, an injective homomorphism $f: V \rightarrow U$ exists such that, for each $R \in \mathbf{R}$ with arity a ,

$$\forall v_1, \dots, v_a \in V \cdot (v_1, \dots, v_a) \in R_V \leftrightarrow (f(v_1), \dots, f(v_a)) \in R_M.$$

A QSTR application has a language L that specifies the set of relation symbols that the designer has deemed relevant to the task at hand. The model \mathbf{M} of a QSTR application is the interpretation of the relations, implemented using *constraints* between the relations (what objects must, or must not, exist in different combinations of relations). For each relation type $R \in \mathbf{R}$ with arity a_R , and for each tuple of arity a_R , the relation either *holds*, *does not hold*, or is *not applicable* for that tuple. Thus, for each relation symbol $R \in \mathbf{R}$ in the language, a QSTR application model \mathbf{M} requires three sets, R_M^+ (holds), R_M^- (does not hold) and R_M^\sim (not applicable), with the axiom

$$\text{Axiom 1. } \forall R \in \mathbf{R} \cdot U \uparrow a_R = R_M^+ \Delta R_M^- \Delta R_M^\sim,$$

where Δ is symmetric difference (the set theoretic equivalent of mutual exclusion). For brevity we will omit the \mathbf{M} and simply write R^+ .

The application *designer* is responsible for selecting an appropriate set of relation symbols and encoding an appropriate set of constraints. A QSTR application *user* can then construct scenarios by specifying a model \mathbf{V} and reasoning is used to determine whether the scenario is valid with respect to the model \mathbf{M} (by proving that \mathbf{V} can be embedded in \mathbf{M}). Table 1 relates these roles to the formal semantics in model theory and QSTR applications.

Often parts of the user's scenario are indefinite or unknown, and reasoning with the application constraints is used to help resolve this ambiguity. For each relation $R \in \mathbf{R}$, the user can place tuples (of objects from V) with arity a_R in a fourth *indefinite* set, $R_M^?$ that is mutually exclusive with the three corresponding *definite* sets. This is a shorthand for specifying a set of models V_1, \dots, V_n each representing a *possible* scenario.

An example of a scenario is

$$\begin{aligned} V &= \{\text{kitchen, lounge, study}\} \\ \text{adjacent}^+ &= \{(\text{lounge, study}), (\text{lounge, kitchen})\} \\ \text{adjacent}^? &= \{(\text{lounge, lounge}), (\text{study, lounge}), \dots\} \\ \text{adjacent}^- &= \{\}. \end{aligned}$$

The *adjacent* relation can be defined as symmetric using the constraint $\{(x,y) \mid (y,x) \in \text{adjacent}^+\} \subset \text{adjacent}^+$. The *LHS* of the constraint as evaluated in the scenario is $\{(\text{study, lounge}), (\text{kitchen, lounge})\}$. The *RHS* as evaluated in the scenario does not contain these tuples as required by the proper subset relation, and so reasoning moves the offending tuples out of $\text{adjacent}^?$ and into adjacent^+ thus satisfying symmetry.

Model Theory	QSTR Application Domain	Actor
language L	specification of useful qualitative relations	QSTR application designer
model \mathbf{M} based on L	constraints that determine the interaction between the relations	
model \mathbf{V} based on L embedded in \mathbf{M}	Using a QSTR application to represent and reason about objects.	QSTR application user

Table 1. Comparing the domains of model theory, QSTR applications, and the roles of QSTR application designers and users.

Homogeneous and Definable sets

In model theory (Marker, 2002), a set X is *definable* in model \mathbf{M} if there is some formula ϕ such that $X = \{(v_1, \dots, v_n) \in U^n \mid \mathbf{M} \models \phi(v_1, \dots, v_n)\}$ (where entails \models means that the formula is true in \mathbf{M}). Alternatively, if no formula exists that can separate two objects, then the objects are considered equivalent, and we say that they are in the same *homogeneous* set. Let $H = \{h_1, \dots, h_n\}$ be a set

of homogeneous sets, where each $h_i \subseteq U$. By definition, h_1, \dots, h_n partition U , that is, they are mutually exclusive and jointly exhaustive. A homogeneous set h is evaluated in scenario s , $s(h) \subseteq V$.

All possible queries are equivalent to some union of homogeneous sets. We define a query q to be a set of homogeneous sets $q = \{h_x, \dots, h_y\}$, and we say a query q is executed in scenario s , $s(q) = s(h_x) \cup \dots \cup s(h_y)$.

Constraints

The model of a QSTR application is defined by constraints between the qualitative relations. If a scenario does not satisfy a constraint then reasoning attempts to move the offending tuples out of indefinite sets ($R^?$) and into one of the definite sets (R^+ , R^- , or R^0). If offending tuples are not indefinite then reasoning has identified a contradiction and the scenario is inconsistent.

Let constraint $c = X \delta Y$, where $\delta \in \{\subseteq, \supseteq, \subsetneq, \supsetneq, =, \dots\}$ is a set comparison and X, Y are set expressions that are evaluated in scenarios. Set expressions are either sets, or the result of set operations.

We now define the complete set of possible comparisons from which δ can be selected. Two sets LHS, RHS either share some objects, or they do not. The degree of overlap can be used to define all possible set relationships. Let $LHS = h_w \cup \dots \cup h_x$ and $RHS = h_y \cup \dots \cup h_z$, let $q_{LHS} = \{h_x, \dots, h_y\}$ and $q_{RHS} = \{h_x, \dots, h_y\}$ (queries for LHS and RHS respectively). Let

$$\begin{aligned} q_L &= q_{LHS} \setminus q_{RHS} && (H \text{ sets exclusively in } LHS) \\ q_R &= q_{RHS} \setminus q_{LHS} && (H \text{ sets exclusively in } RHS) \\ q_{LR} &= q_{LHS} \cap q_{RHS} && (H \text{ sets in both } LHS \text{ and } RHS) \end{aligned}$$

For any pair of sets LHS and RHS , the queries q_L, q_R , and q_{LR} will evaluate to be either empty or non-empty, giving $2^3 = 8$ different basic set comparisons, as illustrated in Table 2. In each table entry, the outer circle represents the set of all scenarios, and three inner circles each represent a subset of scenarios where a condition holds. The top left circle specifies those scenarios where H sets in the query q_L are empty, $s(q_L) = \emptyset$ and so on. For example, the constraint $X = Y$ is satisfied in exactly those scenarios where both q_L (elements exclusively in X) and q_R (elements exclusively in Y) evaluate to empty and q_{LR} (elements shared by both X and Y) is not empty. The comparison δ can consist of any disjunction of these basic set comparisons (e.g. \subseteq is $\vee \subsetneq$), thus there are $2^8 - 1 = 255$ possible comparisons.

Basic Combinatorics

Finally, we provide some basic equations from combinatorics that will be used in the paper. The size of the powerset of set X is $2^{|X|}$. The number of subsets that contain elements from either set X or set Y is $2^{|X \cup Y|}$, the number of subsets that contain all elements from X and not Y is $2^{|X \setminus Y|}$, and the number of subsets that contain all elements from X and neither Y nor Z is $2^{|X \setminus (Y \cup Z)|}$.

δ	scenarios (shaded=valid scenarios)	δ	scenarios (shaded=valid scenarios)
=		$\bar{\cap}$	
\subsetneq		$= \emptyset$	
\supsetneq		$\subsetneq \emptyset$	
\neq		$\supsetneq \emptyset$	

Table 2. Eight basic relationships between sets derived from the overlap between H sets. Outer circles represent the set of all scenarios, and three inner circles each represent a subset of scenarios where a condition holds. Shaded regions represent scenarios that satisfy the set relationship.

Equivalently, we use bit-string encodings, where the number of unique combinations that can be represented by a bit-string of length n is 2^n . The number of combinations of two bit-strings of length n_1 and n_2 is $2^{n_1+n_2}$. If m bits are fixed in a bit-string of length n then the number of unique combinations that contain the m fixed bits is 2^{n-m} .

Table 3 gives some equations for calculating the number of scenarios that satisfy conditions required by constraints. For example, in each table entry the top left circle specifies those scenarios where H sets in the query q_L are empty, $s(q_L) = \emptyset$, and the number of such scenarios is $2^{|H| \setminus q_L} = 2^{|H| - |q_L|}$. To understand this, consider a bit-string of length $n = |H|$, where $m = |q_L|$ bits are fixed to '0' (representing that $s(q_L) = \emptyset$); the number of unique combinations that this bit-string allows is $2^{n-m} = 2^{|H| - |q_L|}$. Scenarios where H sets in both q_L and q_R are empty, $s(q_L) = \emptyset \wedge s(q_R) = \emptyset$ is $2^{|H| \setminus (q_L \cup q_R)} = 2^{|H| - |q_L| - |q_R|}$ (recall that, by definition, q_L and q_R are mutually exclusive). A bit-string of length $n = |H|$, where $m = |q_L \cup q_R| = |q_L| + |q_R|$, bits are fixed to '0' admits $2^{n-m} = 2^{|H| - |q_L| - |q_R|}$ unique combinations.

formula	scenarios (shaded=valid scenarios)
$2^{ H - qL }$ symmetric with $2^{ H - qR }$ $2^{ H - qLR }$	
$2^{ H - qL \cup qR }$ symmetric with $2^{ H - qL \cup qLR }$ $2^{ H - qLR \cup qR }$	
$2^{ H - qL \cup qR \cup qLR }$	

Table 3. Combinatorial formulae for calculating the number of scenario classes that satisfy particular conditions.

H-Complexity

In this section we present the central theory of *H*-complexity. *H*-complexity is a measure of the number of different scenario classes that can be encoded by a given QSTR language. It identifies the most extreme partitioning of objects that the language allows, so that all other practical partitioning schemes will consist of some subset of the distinctions made in *H*-complexity.

Homogeneous Sets

If no possible set theoretic query exists that can separate two objects, then the objects are considered equivalent. This inherent limitation fundamentally folds objects into homogeneous sets, or *H* sets, providing the foundation for a measure of expressiveness or complexity. *H* sets represent the maximum refinement permitted by the QSTR language, i.e. the point where no further distinctions are possible. Thus, *H*-complexity = $|H|$.

Assume for simplicity that all relations have an arity of 1 (i.e. they represent qualitative properties such as *round* or *large*). If there are n relations, and each relation is represented by four sets, then there are 4^n different combinations of these relations, i.e.

1. $R_1^+ \cap R_2^+ \cap \dots \cap R_n^+$,
2. $R_1^- \cap R_2^+ \cap \dots \cap R_n^+$,
- ...
- 4^n . $R_1^- \cap R_2^- \cap \dots \cap R_n^-$.

In general,

$$|H| = \prod_{R \in \mathbf{R}} |H_R| \quad (1)$$

where $|H_R| = |A_R|$ for unary relations, A_R is the selection of relation sets used by R (e.g. + and -), and \prod is the sequence product operator. We calculate *H* sets for relations with higher arities in a later section.

For *H* sets to truly represent the maximum refinement possible, they must be jointly exhaustive and pairwise disjoint (JEPD) so that every object in a scenario will appear in exactly one *H* set. This property is critical; if it did not hold then further refinements could be achieved by taking *H* set intersections and differences. The *H* sets from Equation 1 are pairwise disjoint because, for any relation R_i , the relation's four sets are mutually exclusive, and any two *H* sets always differ by at least one R_i set. They are also jointly exhaustive because, for any relation R_i , each of its four jointly exhaustive sets is covered by some *H* set.

Query Complexity

A query is used to access a subset of objects in a scenario. Query complexity is the maximum number of unique non-empty subsets that can be accessed by some query. We will now show that all accessible, unique subsets of objects in a scenario can be represented as the union of some combination of *H* sets.

H sets are indivisible and mutually exclusive, and so the query that returns the smallest non-empty subset of objects contained in an *H* set is the set expression of the *H* set itself, e.g. $R_1^+ \cap R_2^+ \cap \dots \cap R_n^+$. The smallest subset containing objects from two different *H* sets h_1, h_2 is the union of those two *H* set expressions $h_1 \cup h_2$. It follows that any accessible subset of objects must be the union of some combination of *H* sets. Thus query complexity is the number of different combinations of *H* sets, $2^{|H|}$.

Scenario Complexity

If two objects in a scenario can not be separated by a query, then the objects are considered equivalent and indistinguishable, that is, the objects are in the same *H* set. If the only difference between two scenarios is the number of indistinguishable objects in each non-empty *H* set, then the scenarios are considered equivalent. This follows the intuitive understanding that qualitative models, unlike metric systems, do not deal with numerical quantities. Thus, a scenario equivalence class is defined by the combination of *H* sets that are empty and non-empty, $2^{|H|}$.

There is an interesting parallel between query and scenario complexity. The number of unique, accessible, non-empty subsets of a QSTR application is equal to the number of distinct scenario classes that can be expressed.

An alternative interpretation is that a scenario is defined by the combination of queries that return non-empty results, i.e. if query q_i returns \emptyset in scenario s_i and some non-empty result in s_2 , then the scenarios are logically distinct.

Calculating Homogeneous Sets

In this section we derive the equation for $|H_R|$ admitted by a relation R of arity a_R , and then derive $|H|$ for a QSTR system. Once relations have an arity greater than 1 there are an infinite number of potential H sets, because a binary relation constitutes a total order. Thus, Equation 1 cannot be used to compare QSTR systems with binary or higher relations. We proceed in our analysis by representing a scenario of binary relation R_i as a graph, where objects represent vertices and directed edges represent tuples as illustrated in Figure 1.

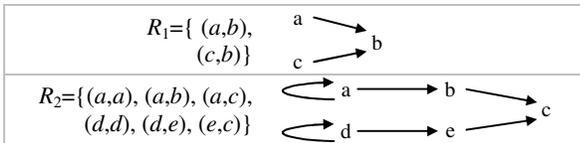


Figure 1. Two graphs representing binary relations R_1 and R_2 .

A set theoretic query describes the structure of a graph and specifies the vertex to be selected with v bound variables,

$$\exists x_1 \dots \exists x_v \cdot x_1 \neq x_2 \wedge \dots \wedge x_1 \neq x_v \wedge \dots \wedge x_{v-1} \neq x_v.$$

For brevity, we will omit explicitly stating these quantifications and conditions for all further queries, and for simplicity we do not allow the universal quantifier \forall . For example, the query $\{x_2 | (x_1, x_2) \in R_1 \wedge (x_3, x_2) \in R_1\}$ will access b from the graph of R_1 in Figure 1.

Calculating $|H_R|$ with Arbitrary Arity

While there are an infinite number of potential graphs and unique accessible subsets, homogeneous sets still exist that contain indistinguishable objects. For example, regarding the graph of R_1 , no query exists that can separate objects a and c (without directly referring to those objects),

$$\begin{aligned} \{a, c\} &= \{x_1 | (x_1, x_2) \in R_1 \wedge (x_3, x_2) \in R_1\} \\ &= \{x_3 | (x_1, x_2) \in R_1 \wedge (x_3, x_2) \in R_1\}, \end{aligned}$$

and the graph of R_2 has three H sets, accessed by the query

$$\{x_i | (x_1, x_1) \in R_2 \wedge (x_1, x_2) \in R_2 \wedge (x_2, x_3) \in R_2 \wedge (x_4, x_4) \in R_2 \wedge (x_4, x_5) \in R_2 \wedge (x_5, x_3) \in R_2\},$$

namely $\{a, d\}$ when $i=1$ or 4 , $\{b, e\}$ when $i=2$ or 5 and $\{c\}$ when $i=3$. Thus, homogeneous sets correspond to graph symmetries or automorphisms. Given a graph of a scenario, the number of H sets is the number of vertices minus the number of automorphisms.

In order to make $|H_R|$ a function of QSTR systems rather than scenarios (particular graphs), the query language must be restricted. If the restricted language only recognises a finite number of graphs, it will admit a finite number of H

sets. It is then possible to quantify the complexity of a relation independent of a particular scenario, and measure the relative difference in expressiveness between two QSTR systems. Two basic restrictions are to limit the number of variables (vertices) and the number of tuples (edges). We will consider the former case. Previously, queries have referred to variables x_i where i can be any positive integer. The strongest restriction on the number of variables is $i \leq 1$, affording only one query, $\{x_1 | (x_1, x_1) \in R_i\}$. If $i \leq 2$ then the allowable tuples are (x_1, x_1) , (x_1, x_2) , (x_2, x_1) , and (x_2, x_2) . If v is the number of variables allowed in a query, and a_R is the arity of relation R (i.e. the size of the tuples) then for each query,

$$\text{number of tuples} = v \uparrow a_R$$

All binary tuples of n objects in relation R_i appear in exactly one of the R_i sets (holds etc.), and thus we focus on those queries that contain all $v \uparrow a_R$ tuples permitted by the language. We refer to these as atomic queries. The difference between two atomic queries is the combination of R_i sets in which the tuples must appear. For example, Table 4 gives an extract of the atomic queries where $v=2$ and $a_R=2$. Some graphs are automorphic, e.g. in row 1, and some graphs are isomorphic, e.g. rows 2 and 3. Initially there are $16 \text{ graphs} \times 2 \text{ variables} = 32$ different executable atomic queries. 4 graphs (8 queries) have pairs of automorphic variables making $8/2$ queries redundant, and the remaining 12 graphs (24 queries) can be put into symmetric pairs, making an additional $24/2$ queries redundant, leaving $32 - 4 - 12 = 16$ unique atomic queries. An easy way to arrive at the number of unique atomic queries is to allow every graph, but restrict the selection to the first variable x_1 (this can be viewed as fixing the variable and rotating the edges of the graph to cover symmetries). Thus,

$$\text{number of unique atomic queries} = |A_R|^{\text{number of tuples}}$$

where $\text{number of tuples} = v \uparrow a_R$. Finally, to calculate $|H_R|$ we must determine the smallest JEPD queries that contain the atomic queries. Atomic queries are not JEPD when their corresponding graphs are overlapping induced subgraphs of the full scenario graph. For example, consider the scenario graph in Figure 2.

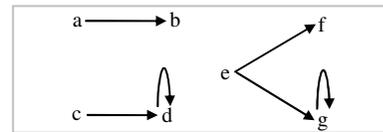


Figure 2. Graph consisting of three disconnected subgraphs. Subgraphs $\{a, b\}$ and $\{c, d\}$ correspond to atomic queries, where $\{e, f, g\}$ contains both atomic queries as induced subgraphs.

If $v=2$ then two atomic queries are:

$$\begin{aligned} \{x_1 | (x_1, x_1) \in R^- \wedge (x_1, x_2) \in R^+ \wedge (x_2, x_1) \in R^- \wedge (x_2, x_2) \in R^-\} &= \{a, e\}, \\ \{x_1 | (x_1, x_1) \in R^- \wedge (x_1, x_2) \in R^+ \wedge (x_2, x_1) \in R^- \wedge (x_2, x_2) \in R^+\} &= \{c, e\}. \end{aligned}$$

The atomic queries are not JEPD, as vertex e appears in both results. However, if we take all combinations of atomic queries by intersection and difference, we can produce a JEPD collection of H sets, hence

$$|H_R| = 2^{\text{number of unique atomic queries}} - 1$$

where the *number of graphs* = $|A_R|^{\text{number of tuples}}$. To summarise,

$$|H_R| = (2 \uparrow |A_R| \uparrow v \uparrow a_R) - 1. \quad (2)$$

Calculating $|H|$ with Arbitrary Arity

The problem is now calculating $|H|$, the total number of H sets across all relations when those relations can take any arity. Previously, we only referred to one relation within a query. Given v bound variables, queries will now take the form,

$$\{x_1 \mid \text{query } R_1, \text{ query } R_2, \dots, \text{ query } R_n\}$$

where *query* R_i is one of the unique atomic queries for relation R_i . The number of queries permitted in this form is

$$|\text{atomic } R_1 \text{ queries}| \times \dots \times |\text{atomic } R_n \text{ queries}|.$$

We have shown that the number of atomic queries for R_i is $|A_{R_i}| \uparrow v \uparrow a_{R_i}$, thus

$$\begin{aligned} |H| &= 2 \uparrow (|\text{atomic } R_1 \text{ queries}| \\ &\quad \times \dots \times \\ &\quad |\text{atomic } R_n \text{ queries}|) \\ &= 2 \uparrow ((|A_{R_1}| \uparrow v \uparrow a_{R_1}) \times \dots \times (|A_{R_n}| \uparrow v \uparrow a_{R_n})) \\ &= 2 \uparrow \prod_{R \in \mathbf{R}} (|A_R| \uparrow v \uparrow a_R). \end{aligned} \quad (3)$$

Using this formulation we can identify some basic properties of expressiveness. In general, H -complexity is a function of the number of atomic queries that the QSTR language allows. Restricting the number of variables in a query to v makes complexity a function of the number of relation states, $|A_R|$, the number of variables v , and the relation arity. Moreover, Equation 3 specifies the relative influence that each component has on complexity; relation arity has the most influence, followed by the number of variables having exponentially less influence, and finally arity having exponentially less influence again. These properties help to inform the developer about how changes to each component affects expressiveness, and the relative difference in expressiveness between two QSTR systems.

Applying H -Complexity

In this section we present two ways that H -complexity can be employed to assist QSTR application development.

Test coverage

Application test space is defined according to system inputs and outputs, and the system structure such as decisions and control paths. Covering the entire, often

infinite test space is clearly impractical and thus software engineers employ methods that isolate key subsets such as boundary checking, equivalence class partitioning, and cause-effect graphs (Burnstein, 2003). Test coverage is a standard metric in software engineering used to guide testing. H -complexity can be used to quantify a QSTR application's potential test space for calculating test coverage, as it specifies the degree of refinement permitted by the language. Consider the following application-specific constraint for determining apparent room colour temperature (Schultz et al., 2009):

"If a room has at least one warm light, and does not have lights of any other temperature, then the room has a warm colour temperature"

This constraint may then be integrated into an existing QSTR system that reasons about spatial arrangements of light sources and surfaces. One formal encoding of this constraint might be

$$\begin{aligned} \{x \mid (\exists y. \text{light}^+(y) \wedge \text{in}^+(y,x) \wedge \text{warm}^+(y)) \\ \wedge (\exists y'. \text{light}^+(y') \wedge \text{in}^+(y',x) \wedge \text{warm}^-(y'))\} \subseteq \text{warm}^+. \end{aligned} \quad (3)$$

Based on the structure of the constraint, the developer may decide to test the class of scenarios where the LHS is not empty $\exists x \cdot \exists y \wedge \exists y'$, and where the LHS is empty $\forall x \cdot \exists y \wedge \exists y'$. Based on the relations in the constraint, the developer may decide to test all combinations of conditions for both y and y' ; noting that not all conditions are independent (e.g. if there exists y such that $y \in \text{warm}^+$ then there must also exist y' such that $y' \in \text{warm}^+$), this provides a test set size of 2^8 . The issue is that the developer needs to know how many tests must be executed before achieving some level of confidence that the application is fit for purpose. Three test coverage metrics are

1. proportion of H sets tested, $|H_T| / |H|$
2. proportion of relevant H sets tested, $|H_T \cap H_\Delta| / |H_\Delta|$
3. degree of tested combinations for particular clusters of H sets (e.g. some subset $H' \subset H$ tested up to all triples)

where the H_T is the set of tested homogeneous sets, and relevant homogeneous sets H_Δ are defined according to the application (e.g. using probability distributions over inputs, and weighting critical inputs that must be handled correctly). If H_T is not precisely known, then it can be roughly approximated if the tradeoff between the number of tested H sets and the degree of combination testing is known²:

- a) exhaustive combination testing, $\log_2(|T|) \approx |H_T|$,
- b) k combinations, $\lfloor (k! \cdot |T|)^{1/k} + k/2 \rfloor \approx |H_T|$, and
- c) up to k combinations, $\lfloor (k! \cdot |T|)^{1/k} \rfloor \approx |H_T|$.

Continuing with the case study, given $A = \{+, -\}$, $v = 2$ and $a = 2$, the *in* relation yields $|H_{in}| = 2^{16} - 1$, and together *light* and *warm* yield $|H_{warm}| \cdot |H_{light}| = 2^2$. Thus, the potential test space is intractable, i.e. applying Equation 2 gives

² Formulae are derived by rearranging the standard combination formula, ${}_n C_k = n! / (k!(n-k)!)$ where $n = |T|$ and identifying the dominate terms.

$2\uparrow(2^{18}-2^2)$ containing $\sim 2.6 \cdot 10^5$ types of distinction. However, this space includes an exhaustive enumeration of tests that violate basic properties. We can therefore focus the potential test space, with respect to the rule at hand, by incorporating the following restrictions:

- ∞ *in* is neither reflexive nor symmetric
- ∞ no object is ever *in* a *light*

Rather than exhaustively testing these basic constraints in every rule, the developer may exercise those properties in isolation (e.g. testing when in^+ erroneously contains a pair of symmetric tuples) and then assume they hold when testing other rules. The first constraint allows three queries,

$$\begin{aligned} q_1 &= \{x \mid (x,y) \in in^+ \wedge (y,x) \in in^-\}, \\ q_2 &= \{x \mid (x,y) \in in^- \wedge (y,x) \in in^+\}, \text{ and} \\ q_3 &= \{x \mid (x,y) \in in^- \wedge (y,x) \in in^-\}, \end{aligned}$$

giving $|H_{in}|=2^3-1$. The second constraint states that for all scenarios, $q_2 \cap light^+ = \emptyset$, hence H sets that contain this query can be removed. 2^{3-1} sets in H_{in} intersect with q_2 giving $|H_{in} \text{ with } q_2| \cdot |H_{light^+}| \cdot |H_{warm}| = 2^2 \cdot 1 \cdot 2 = 8$. Therefore, the relevant number of homogeneous sets is $|H_{\Delta}| = |H_{in}| \cdot |H_{light^+}| \cdot |H_{warm}| - 8 = (2^3-1) \cdot 2 \cdot 2 - 8 = 20$. Recall that in our example above, the developer has chosen exhaustive combination testing of selected relevant components $|T|=2^8$, giving $|H_T| \approx \log_2(|T|) = 8$. Test coverage results are

1. $|H_T|/|H| = 8/(2.6 \cdot 10^5) \approx 0\%$
2. $|H_T \cap H_{\Delta}| / |H_{\Delta}| = 8/20 = 40\%$
3. all combinations $k=1 \dots |H_T|$ of the cluster H_T are tested, and no other clusters are tested at all.

Firstly note that a few key restrictions (particularly the number of tuples in a query) rapidly focus the test space. Secondly it is clear that the majority of the relevant test space is completely untested. The developer can now identify the untested distinctions and decide whether further tests are required.

Constraints and Expressiveness

As we observed in the previous section, a designer can isolate scenario classes by encoding constraints. This is an effective method for improving testing efficiency by isolating and independently testing particular model fragments (to avoid combinatorial explosion). H -complexity can be used to measure the number of scenario classes that satisfy a collection of constraints, thus allowing a designer to quantify the reduction in the test suite.

Consider a simple system with two unary relations, R_1 and R_2 , $A=\{+,-\}$, and a constraint $R_1^+ \subseteq R_2^+$. From Equation 1 we have $|H|=4$ giving 2^4 unique scenario classes, but not all of them will satisfy the constraint. To determine the set of valid scenario classes, the constraint is translated into a union of H sets.

Let $h_1=(R_1^+ \cap R_2^+)$, $h_2=(R_1^+ \cap R_2^-)$, $h_3=(R_1^- \cap R_2^+)$, $h_4=(R_1^- \cap R_2^-)$;
 $R_1^+ = h_1 \cup h_2$
 $R_2^+ = h_1 \cup h_3$

Therefore,

$$\begin{aligned} R_1^+ \subseteq R_2^+ &\equiv h_1 \cup h_2 \subseteq h_1 \cup h_3 \\ &\equiv h_2 \subseteq h_3 \quad \dots \text{remove } h_1 \text{ from both sides} \end{aligned}$$

So a scenario is only valid if $h_2 \subseteq h_3$, but by definition homogeneous sets are mutually exclusive, $h_2 \cap h_3 = \emptyset$. It follows that the only valid scenarios are those where $h_2 = \emptyset$. Given n constraints of the form $LHS_i \subseteq RHS_i$, where $LHS_i = h_{i_w} \cup \dots \cup h_{i_x}$ and $RHS_i = h_{i_y} \cup \dots \cup h_{i_z}$, and let $q_{L_i} = \{h_{i_w}, \dots, h_{i_x}\} / \{h_{i_y}, \dots, h_{i_z}\}$ (H sets exclusively in LHS). The only valid scenarios are those where every LHS_i is empty, $s(q_i) = \emptyset$, thus (refer to the previous section ‘‘Basic Combinatorics’’)

$$\begin{aligned} |\text{valid scenarios}| &= 2\uparrow(|H| - |q_{L_1} \cup \dots \cup q_{L_n}|). \\ \% \text{ of valid scenarios} &= |\text{valid scenarios}| / |\text{all scenarios}| \\ &= 2\uparrow(|H| - |q_{L_1} \cup \dots \cup q_{L_n}|) / 2\uparrow|H| \\ &= 2\uparrow(-|q_{L_1} \cup \dots \cup q_{L_n}|). \end{aligned}$$

We will derive a function Θ_T for calculating valid scenarios for any type of constraint. Let reference Table 4 accept a constraint and return the appropriate formula for calculating valid scenarios. Note that each table entry takes the form

$$a_0 2^{-e_0} + \dots + a_n 2^{-e_n}$$

where each term $j=0 \dots n$ has a sign $a_j \in \{1, -1\}$ and an integer exponent e_j . Let *term map* θ_T be a map that collects the sign and size of the exponent from each term in this series. That is, θ_T takes a constraint c_i and returns a list $\{(a_0, e_0), \dots, (a_n, e_n)\}$ such that for each tuple (a_j, e_j) there exists some term $\dots + a_j 2^{e_j} + \dots$ in the Table 4 entry for c_i .

We can now recursively define the function Θ_T that accurately calculates the term information from n constraints, for $i=2 \dots n$,

$$\Theta_T(c_1) = \{(a, f(X)) \mid (a, X) \in \theta_T(c_1)\} \quad (3.1)$$

$$\Theta_T(c_i) = \Theta_T(c_{i-1}) \times \theta_T(c_i) \quad (3.2)$$

such that, given $(a_x, X) \in \theta_T(c_{i-1})$ and $(a_y, Y) \in \theta_T(c_i)$, $(a_x, X) \times (a_y, Y) = (a_x a_y, g(X, f(Y)))$. The function $f(X)$ determines the information about the exponent that is used, and $g(X, Y)$ determines how the exponent information of different terms should be combined. We evaluate the terms using the formula (where f^{-1} is the inverse of f)

$$\% \text{ valid scenarios} = \sum a_j 2^{\uparrow - |f^{-1}(e_j)|}.$$

For example, let two constraints be

$$\begin{aligned} c_1 &= h_1 \cup h_2 \cup h_3 = h_3 \cup h_4 \cup h_5 \\ c_2 &= h_6 \cup h_7 \subseteq h_1 \cup h_7 \cup h_8. \end{aligned}$$

$$\begin{aligned} q_{1L} &= \{h_1, h_2\}, \quad q_{1R} = \{h_4, h_5\}, \quad q_{1LR} = \{h_3\} \\ q_{2L} &= \{h_6\}, \quad q_{2R} = \{h_1, h_8\}, \quad q_{2LR} = \{h_7\} \end{aligned}$$

$$\Theta_T(c_1) = \left\{ \begin{array}{l} (1, \{h_1, h_2, h_4, h_5\}), \\ (-1, \{h_1, h_2, h_3, h_4, h_5\}) \end{array} \right\}$$

$$\theta_T(c_2) = \{ \begin{array}{l} (1, \{h_6\}), \\ (-1, \{h_1, h_6, h_8\}) \\ (-1, \{h_6, h_7\}) \\ (1, \{h_1, h_6, h_7, h_8\}), \quad \dots \text{for } \subset \\ \\ (1, \{h_1, h_6, h_8\}), \\ (-1, \{h_1, h_6, h_7, h_8\}) \quad \dots \text{for } = \\ \end{array} \}$$

Let $f(X)=X$ and $g(X, Y)=X \cup Y$ (see Table 5).

$$\begin{aligned} \theta_T(c_1) &= \theta_T(c_1) \\ \theta_T(c_2) &= \theta_T(c_2) \times \theta_T(c_1) \\ &= \{(1, \{h_6\}) \times (1, \{h_1, h_2, h_4, h_5\}), \\ &\quad (1, \{h_6\}) \times (-1, \{h_1, h_2, h_3, h_4, h_5\}), \\ &\quad \dots, \\ &\quad (-1, \{h_1, h_6, h_7, h_8\}) \times \\ &\quad \quad (-1, \{h_1, h_2, h_3, h_4, h_5\}), \\ &= \{(1, \{h_1, h_2, h_4, h_5, h_6\}), \\ &\quad (-1, \{h_1, h_2, h_3, h_4, h_5, h_6\}), \\ &\quad \dots, \\ &\quad (1, \{h_1, h_2, h_3, h_4, h_5, h_6, h_7, h_8\}) \}. \end{aligned}$$

set comparison	scenarios (shaded=valid scenarios)	formula
\subset symmetric with \supset $\bar{\cap}$		$\begin{aligned} &2^{- qL } \\ &- 2^{- qL \cup qR } \\ &- 2^{- qL \cup qLR } \\ &+ 2^{- qL \cup qR \cup qLR } \end{aligned}$
$=$ symmetric with $\supset \emptyset$ $\subset \emptyset$		$\begin{aligned} &2^{- qL \cup qR } \\ &- 2^{- qL \cup qR \cup qLR } \end{aligned}$
$= \emptyset$		$2^{- qL \cup qR \cup qLR }$
$\bar{\cap} \neq$		$\begin{aligned} &1 \\ &- 2^{- qL } \\ &- 2^{- qR } \\ &- 2^{- qLR } \\ &+ 2^{- qL \cup qR } \\ &+ 2^{- qL \cup qLR } \\ &+ 2^{- qR \cup qLR } \\ &- 2^{- qL \cup qR \cup qLR } \end{aligned}$

Table 4. Formulae for calculating the number of scenario classes that satisfy particular set comparisons used in constraints.

To avoid actually identifying and testing H sets (which is an extremely resource intensive process due to their enormous quantity) we can calculate bounds by assuming maximum and minimum overlap between the constraint queries. Lower and upper bounds are calculated by changing the definition of the function f , summarised in Table 5.

result	$f(X) =$	$g(X, Y) =$	error tolerance (ϵ)
exact	X	$X \cup Y$	$ g(X, Y) \leq \alpha$
lower bound	$ X $	$X + Y$	$g(X, Y) \leq \alpha$
upper bound		$\max(X, Y)$	

Table 5. Alternative functions used in equations 3.1 and 3.2 for calculating exact, lower and upper bounds of the number of scenario classes that satisfy a set of constraints.

The lower bound of % valid scenarios occurs when no constraints share any H sets, and is calculated by summing exponents $g(X, Y) = X + Y$. The upper bound occurs when every query is an improper subset of some query, and is calculated by taking the maximum exponent, $g(X, Y) = \max(X, Y)$.

Finally, we can vastly improve performance by pruning a term once its exponent has exceeded some threshold, α . As the exponents grow, the size of the term quickly becomes negligible, i.e. $\lim_{e \rightarrow \infty} 2^{-e} = 0$. The threshold α is a function of the required error tolerance ϵ and the number of terms $m = \sum_{i=0 \dots n} |\theta_T(c_i)|$,

$$\alpha = \lceil \log_2 (m / \epsilon) \rceil,$$

derived as follows. The sum of all removed terms (where the magnitude of the exponent exceeds some threshold α) must be less than or equal to the given error tolerance,

$$\begin{aligned} \epsilon &\geq m 2^{-\alpha} \\ \epsilon / m &\geq 2^{-\alpha} \\ \epsilon \cdot 2^\alpha / m &\geq 1 \\ 2^\alpha &\geq m / \epsilon \\ \alpha &\geq \log_2 (m / \epsilon) \end{aligned}$$

For example, an application has 50 constraints and we want the error to be within 0.01%. The number of terms m depends on the constraint relations, but for this example on average let each constraint have 6 terms, giving $m = 50 \times 6 = 300$ terms. Therefore,

$$\begin{aligned} \alpha &= \lceil \log_2 (300 / 0.01) \rceil \\ &= \lceil \log_2 (30000) \rceil \\ \alpha &= 15 \end{aligned}$$

Conclusions

In this paper we presented the H -complexity metric for analysing QSTR systems, with the aim of supporting the design and evaluation of QSTR applications. H -complexity measures the expressiveness of a QSTR system according to the number of distinct scenario classes that can be encoded by the system. Specifically, we

derived H -complexity by firstly defining the concept of homogeneous sets as the fundamental folding of objects into equivalence classes permitted by the QSTR language. We then used homogeneous sets to quantify complexity by identifying a one-to-one relationship between accessible, unique subsets and scenario classes. We have shown that H -complexity of a relation can be quantified independently of a particular scenario by applying restrictions to the query language. This enables a developer to measure the relative difference in expressiveness between two QSTR systems that contain relations that admit an infinite number of potential homogeneous sets. Furthermore, we have shown that relation arity and the number of query variables significantly dominate expressiveness. This is useful in determining the relative difference in expressiveness between two QSTR systems, which can be used by a developer to guide QSTR application design. Finally, we presented two examples which illustrate how H -complexity can be employed to support QSTR application development, firstly, to calculate the potential test space for determining test coverage, and secondly to quantify the reduction in expressiveness due to constraints. These examples demonstrate that our complexity metric is a versatile and effective tool for supporting the development of QSTR applications.

Acknowledgements

This work has been funded by the Bright Future Top Achiever Doctoral Scholarship (Tertiary Education Commission, New Zealand).

References

- Allen, J. F. *Maintaining knowledge about temporal intervals*. Communications of the ACM: 26, 11, pp. 832-843: ACM Press, 1983.
- Burnstein I. *Practical software testing: a process oriented approach*. Springer, New York, 2003.
- Cohn, A. G., and Renz, J. *Qualitative spatial reasoning*. In van Harmelen F, Lifschitz V, Porter B (eds): Handbook of Knowledge Representation: Elsevier Science, 2007.
- Dylla, F., Frommberger, L., Wallgrün, J. O., and Wolter, D. *SparQ: A Toolbox for Qualitative Spatial Representation and Reasoning*. In Proc. Workshop on Qualitative Constraint Calculi: Application and Integration, 2006.
- Hodges, W. *A Shorter Model Theory*. Cambridge University Press, 1997.
- Kuipers, B. *Qualitative reasoning*. MIT Press, 1994.

Ligozat, G., Mitra, D., and Condotta, J. *Spatial and temporal reasoning: beyond Allen's calculus*. AI Communications 17, 4, pp. 223–233, 2004.

Marker, D. *Model Theory: An Introduction*. Springer-Verlag New York, 2002.

Nebel, B., Bürckert, H.J. *Reasoning about temporal relations: a maximal tractable subclass of Allen's Interval Algebra*, Journal of the ACM, 42, 1, pp.43–66, 1995.

Schultz, C., Amor, R., Lobb, B., Guesgen, H., *Qualitative design support for engineering and architecture*. Advanced Engineering Informatics, Elsevier, 23, 1, pp. 68-80, 2009.

Vilain, M., Kautz, H., van Beek, P. *Constraint propagation algorithms for temporal reasoning: a revised report*, Readings in Qualitative Reasoning about Physical Systems, Morgan Kaufmann, San Francisco, CA, USA, 1989.

Wölfl, S., Mossakowski, T., and Schröder, L. *Qualitative constraint calculi: Heterogeneous verification of composition tables*. In Proc. FLAIRS-07, pp. 665-670. AAAI Press, 2007.

Pragmatic Scenario Inference on Static Spatial Configurations

Dan Tappan

Department of Computer Science, Idaho State University
921 S. 8th Ave., Stop 8060
Pocatello, ID 83209-8060
tappdan@isu.edu

Abstract

The work takes static spatial configurations defined by quantitative, graphical data like the positions and orientations of nature-related objects and infers basic, high-level, pragmatic meaning of the scenario from a small set of semantic actions; e.g., the wolves are *chasing* the sheep. It uses an inheritance-based knowledge base to define contextually appropriate, case-based roles, and geometric constraint satisfaction to recognize spatial dependencies. This successful pilot study elicits semantic features of interest for follow-on investigation. It uses a quantitative survey methodology to compare the performance of the system against human subjects based on the standard information-retrieval measures of precision and recall.

Introduction

A static spatial configuration, such as in Figure 1, contains low-level, quantitative knowledge about its objects and their positions and orientations. These details are sufficient to render the image, but they do not directly provide any insight into the higher-level composition of the scene, namely what the objects might be doing individually and collectively, and why. For example, the wolves are arguably chasing the sheep.

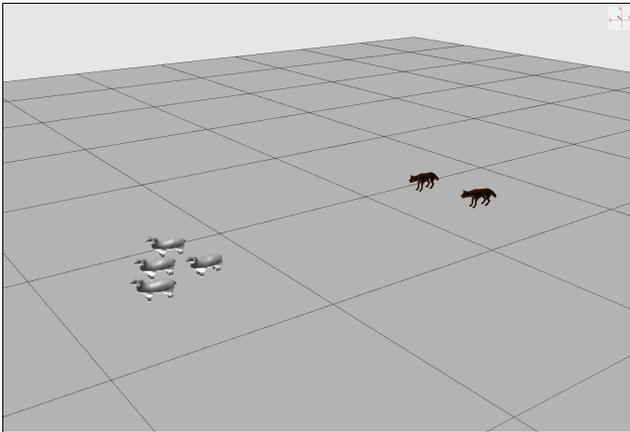


Figure 1: Sample Configuration

The goal of this work is to infer from a small set of spatially relevant semantic actions the superficial pragmatics characterizing a variety of simple predator-prey scenarios with wolves, sheep, and several supporting objects. It uses an inheritance-based knowledge base of concepts, attributes, and declarative rules to define the contextually appropriate spatial interpretation of many contexts. The underlying reasoning mechanism is non-deductive geometric constraint satisfaction.

This paper addresses a pilot study to determine the feasibility of follow-on work and to discover general semantic features it could investigate. It was formally evaluated through a quantitative survey methodology that compared the computer performance against human performance. Specifically, it used the standard measures of precision and recall from the field in information retrieval.

Related Work

This work extends the base system by Tappan (2004a, 2004b, 2008b), which generates and renders configurations from natural-language descriptions, and Tappan (2008a), which infers spatial relations from existing configurations. The approach to inferring spatial knowledge loosely draws upon other work by Neumann (1989), Walter et al. (1998), Koller et al. (1992), and Tsotsos (1985) for scene interpretation. Tversky (2000) covers in comprehensive detail many of the spatial issues that complicate the problem. Several works (Herskovits 1986; Claus et al. 1988; and Olivier and Tsujii 1994), in particular, form the basis for defining and interpreting spatial frames of reference. Most early approaches to spatial analysis adopted purely geometric solutions and did not take advantage of spatial knowledge relevant to the objects (Xu 2002; Yamada 1993). More recent work, especially in Geographic Information Systems, attempts to account for such contextual information (Peters and Shrobe 2003; Davis 1990; Egenhofer and Franzosa 1991; Frank 1992; Frank 1996; Hernández et al. 1995; Randell et al. 1992). This work follows the latter approach. Additional inspiration derives from recent work in spatial-intent recognition, case-based plan recognition, and recognition of natural scene categories (Kiefer and Schlieder 2007; Cheng and Thawonmas 2004; Lazebnik et al. 2006).

Methodology

For space reasons, the description of the system and the methodology of the study running on it are intertwined.

Spatial Configurations

A spatial configuration consists of objects in a static, two-dimensional, tabletop zoo environment. The system currently supports over 120 non-articulated objects, mostly animals and plants, selected because they exhibit great variety in their spatial characteristics and interpretations. The static aspect eliminates the effects of movement, time dependencies, and the frame problem, among others, which are indeed relevant to this work, but beyond its scope (Adorni 1984; Sowa 1991; Srihari 1994; Coyne and Sproat 2001).

The underlying representation of a configuration is a simple semantic network, which is particularly suited to this task for three reasons (Sowa 1991). First, its primary components, nodes and directional arcs, map directly to the objects, properties, and relations in a configuration. For example, Figure 2 is a semantic network that describes a wolf looking north at a sheep a little northeast of it. Second, as a straightforward computational data structure, standard graph algorithms can operate on it natively. Third, as a well-studied and commonly used formalism for artificial intelligence, it facilitates transferring knowledge representations to and from other applications (Russell and Norvig 1995; Sowa 2000).

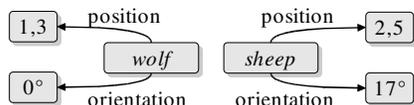


Figure 2: Semantic Network

The semantic networks derive from two sources. The first is manual specification of where the objects are and are facing. This approach is necessary to guarantee adequate coverage of particular, nuanced scenarios to test, but it is tedious for large data sets. The second source is automated scenes derived from rudimentary natural-language descriptions. It is described in detail in Tappan (2004a). The basis of its input is small, descriptive statements, such as:¹

There are two wolves and four sheep. The wolves are south of the sheep, near each other, facing the sheep, and midrange from the sheep. The sheep are near each other and facing away from the wolves.

Figure 1 renders one possible interpretation. Any number can be generated stochastically, which greatly reduces the amount of time needed to create tests.

¹ Paraphrased somewhat for brevity and easier reading. The parser does not actually support plural nouns, irregular plurals, or comma-delimited clauses.

Annotation

A total of 20 manual and automated configurations were tagged by humans to indicate their plausible spatial interpretations; e.g., *wolves chasing sheep*. Open-ended interpretation is not the goal, so the set of tags is limited to the following perceived actions. More than one is possible per configuration.

Unary actions involve only one active object type, such as wolves. Other object types may be present, but they play a passive role. Each tag (in monospace font) is characterized in English here; the actual formalism of their definition will be covered shortly. The only unary actions currently supported are simple, collective sheep behavior:

migrate	sheep grouped and oriented similarly
graze	sheep grouped and oriented dissimilarly
drink	sheep positioned around (passive) water object like pond, lake, or pool

Binary actions involve two active object types. The first group reflects tactical enclosure interpretations:

flank	sheep grouped; 3+ wolves at base and to either side of group, likely facing it
surround	sheep grouped; 3+ wolves around perimeter of group, likely facing it

The next group reflects linear attack interpretations:

conceal	(passive) view-blocking object, like tree or rock, between wolf and sheep; wolf facing sheep, near and likely at edge of view-blocking object
stalk	sheep facing away from wolf; wolf facing sheep, far from sheep
chase	sheep facing away from wolf; wolf facing sheep, close to sheep

The final group reflects situational awareness:

aware	at least one sheep facing wolf
unaware	no sheep facing wolf
anticipate	all sheep facing wolf

There is also an `unknown` tag for scenarios that cannot be assessed as any of the above.

Tagging used a straightforward survey methodology: 9 computer-science undergraduates each annotated the 20 configurations with any combination of these tags. The images were available online, in color, from three consistent, fixed vantage points. For this pilot study, the surveys were not anonymous.

Manual Scenario Extraction

This initial tagging serves as manual training data to extract common spatial semantic features between similar configurations. To be effective, there must be reasonable

agreement between taggers on the interpretations. A formal statistical measure like Kappa correlation is commonly used to measure intercoder reliability (Fleiss 1971). However, for simplicity, and to align with parallel work (in progress) that tries to weight the various choices, this work calculates a straightforward percentage based on the number of participants who selected a tag. This consensus-based approach stipulates that a tag with agreement below an empirically determined threshold of 65% is discarded as too ambiguous and therefore probably not computable.

Analysis of the discards suggests that disagreement is due primarily to two factors. One is the lack of articulation in the objects. For example, the states of sitting, standing, walking, running, and even sleeping all appear the same, but they can have profoundly different effects on the overall interpretation. The other is the lack of temporal cues in a single snapshot of a dynamic scene. For example, a wolf facing away from a sheep could be walking away, or it could be merely turning around.

The goal of scenario extraction is to characterize the kinds of details that contribute to different interpretations. They are informally organized into three categories.

Constraint Satisfaction The underlying reasoning formalism, to be discussed shortly, uses geometric constraint satisfaction. Some features of interest map directly to it:

- *visibility*: can the wolf see the sheep, based on field of view, range, and visual acuity (possible degradation over range)?
- *accessibility*: can the wolf get to the sheep it sees?
- *boundary conditions*: when do states apply and not apply, and what kind of transition is there between the two? For example, is going from not visible to visible abrupt or smooth?
- *scale and range*: behavior may be based on size or scope; e.g., wolves far from sheep may be more cautious than those close to them, so as not to alert the sheep.

Behavioral Roles High-level interpretation of objects in concert requires an understanding of their case-based roles (Turner 1998; Cheng and Thawonmas 2004):

- *disabler*: an object that hinders an interpretation; e.g., an uncrossable stream between the wolf and the sheep.
- *enabler*: an object that helps an interpretation; e.g., a bridge over the stream, or a tree for concealment.
- *neutral*: an object that either party can use, but it favors neither; e.g., a wall for hiding.
- *inert*: an object that plays no discernible role; e.g., clouds.
- *outward action*: what an object can and cannot do to other objects; e.g., a wolf can attack sheep but cannot attack more than one simultaneously.

- *inward action*: what an object can and cannot have done to it by other objects; e.g., sheep can be attacked by more than one wolf simultaneously, but not by another sheep.

Superficial Planning There is not enough information to do substantial planning currently, but some elements seem promising:

- *necessary and sufficient conditions*: how objects initiate the trajectory for a chain of events; e.g., a wolf shows up, then the sheep flee. They do not flee without the wolf. Thus, to tag a configuration without a wolf as *chase* makes no sense.
- *utility*: what the objects value; e.g., a wolf “wins” by killing sheep, and sheep “win” by not being killed by wolves.
- *Individual vs. collective outcomes*: how interpretations differ according to scope; e.g., killing one sheep is bad for the individual, but it may allow others to escape, which is good for the collective.

Knowledge-Base Augmentation

This informal characterization of semantic features is not adequate for an automated computational approach. The algorithm needs to be able to infer substantial unstated details about objects from commonsense background knowledge. A knowledge base provides this support.

Existing Knowledge Base The system that this work extends generates a set of plausible images from a restricted class of English sentences describing zoo-related scenes. It employs an inheritance-based, declarative knowledge base of over 120 physical concepts, each of which either inherits its attributes and rules for spatial interpretation from its ancestors, or it defines/overrides them itself. Figure 3 is a highly simplified abstraction, which Tappan (2004a) formally defines in detail.

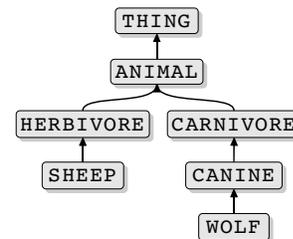


Figure 3: Simplified Knowledge Base

An attribute defines whether a concept exhibits a particular spatial behavior; e.g., whether a concept has a canonical front, which generally corresponds to its having a face or eyes. As objects and concepts are not articulated, any head is always fixed in line with the orientation of the body. This simplification eliminates the need to determine

the configuration of body parts; e.g., the body of the dog is oriented north, but it is looking east.

A rule specifies when a particular spatial relation, like *near*, applies from one object to another. It uses a formalism of geometric fields that describe a collection of cells in a two-dimensional, top-view, polar projection centered around the object (Yamada et al. 1992; Yamada 1993; Gapp 1994; Olivier and Tsujii 1994; Freska 1992). Experimentation suggests that 32 sectors and 100 rings similar to Figure 4 are sufficient for the current domain of concepts and relations. Each cell defines a small subregion of the projection that can be conditionally inspected for the presence of other objects.

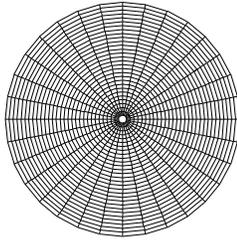


Figure 4: Available Fields

Any combination of selected cells among the 3,200 available is valid, but in practice, only variations of two types define all spatial relations: *wedges* for position and orientation relations, and *rings* for distance relations. Figures 5a and 5b show respective examples of the relations *front-of* and *far-from* for object *c*, which is facing the direction of the arrows.

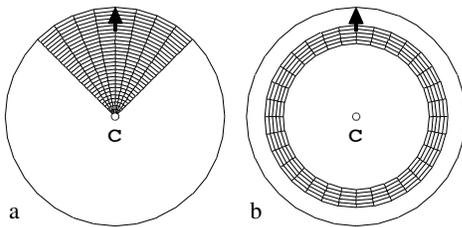


Figure 5: Sample Wedge and Ring Fields

Each concept in the knowledge base has access to the attributes and rules for its spatial interpretation. These rules define the 42 distance, orientation, and position relations in Tables 1 through 3, respectively. For space reasons, Tables 2 and 3 omit 35 other relations prefixed with *direct*, which specifies a narrower interpretation with the same general meaning; e.g., *direct-front-of* would fan out less to the sides. The interpretation of appropriateness depends on certain *ad hoc* generalities of the concept. For example, the relation *near* is closer (in absolute terms) for a mouse than it is for an elephant due to their differences in magnitude (Hernández 1994; Olivier and Tsujii 1994; Stevens and Coupe 1978).

Many relations in Table 3 have both local and global forms, which respectively apply in the intrinsic (or object-centered) and deictic (or viewer-centered) frames of reference (Herskovits 1986). For example, the intrinsic relationship *in front of the dog* specifies a region outward from the dog's face, but the deictic relationship *in front of the tree* specifies a region outward from the tree to the position of the viewer, which is not stated.

inside	midrange-from
outside	far-from
adjacent-to	at-fringe-of
near	

Table 1: Distance Relations

facing	facing-west
facing-away-from	facing-northeast
facing-north	facing-northwest
facing-south	facing-southeast
facing-east	facing-southwest

Table 2: Primary Orientation Relations

local-front-of	global-front-right-of
local-back-of	global-back-left-of
local-left-of	global-back-right-of
local-right-of	between
local-front-left-of	north-of
local-front-right-of	south-of
local-back-left-of	east-of
local-back-right-of	west-of
global-front-of	northeast-of
global-back-of	northwest-of
global-left-of	southeast-of
global-right-of	southwest-of
global-front-left-of	

Table 3: Primary Position Relations

Each concept has access to its contextually applicable rules that map fields to relations. For example, this (slightly abridged) rule returns the set of all objects that have an object (instance) of this concept in their near field:

```
(RELATION near
 (FIELD-MUST-CONTAIN ?b.field-near ?self))
```

This rule returns the set of all objects that are in the front field of this object, if it has a canonical front:

```
(RELATION facing
 (TRUE ?self.has-canonical-front
 (FIELD-MUST-CONTAIN ?self.field-front ?b)))
```

And this rule,

```
(RELATION in-back-of
 (OR
 (TRUE ?b.has-canonical-front
 (FIELD-MUST-CONTAIN ?b.field-back ?self))
 (FALSE ?b.has-canonical-front
 (FIELD-MUST-CONTAIN ?b.field-north ?self))))
```

returns the set of all objects subject to the following criteria:

- the other object has a canonical front (e.g., WOLF) and this object is in its `back` field; or,
- the other object does not have a canonical front (e.g., TREE) and this object is in its `north` field.

These conditional cases account for the deictic and extrinsic frames of reference, respectively (Tappan 2004b; Herskovits 1986). The latter extends the intrinsic frame by fixing the position of the viewer; e.g., *in front of the tree (as seen from the north)*.

The final element of this stage combines the explicitly stated knowledge from the semantic network with the implicitly inferred background knowledge from the knowledge base. Figure 6 depicts a simplified example of this process: objects *wolf* and *tree* link to concepts SHEEP and WOLF, respectively. Thus, *wolf* has access to the rules about itself and, through inheritance, also to its ancestor concepts CANINE, CARNIVORE, ANIMAL, and THING. The same process holds for *sheep*. It is important to note the distinction between an *object*, which is a unique instance in the configuration, and a *concept*, which is a shared set of attributes and rules that all instances of it must have in common. For clarity, this distinction is rendered typographically through italics and capitalized monospace font, respectively.

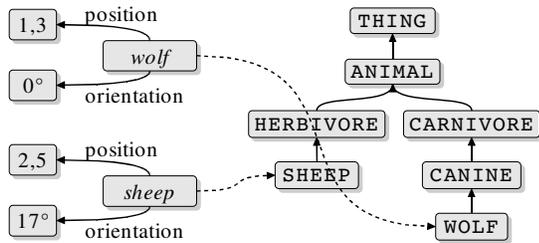


Figure 6: Semantic Network Linked to Knowledge Base

Extended Knowledge Base The knowledge base in the base version of this system targets how one object relates to another on an individual, one-to-one basis. The scenarios to be classified in this work are on a collective basis, which requires one-to-many, many-to-one, and many-to-many relationships. For some semantic features, this extension requires merely adding additional spatial relations. For example, this rule,

```
(RELATION migrating-with
(AND
  (IS-CONCEPT ?self collective-animal)
  (IS-CONCEPT ?b collective-animal)
  (FIELD-MUST-CONTAIN
    (RANGE ?b field-adjacent field-midrange) ?self)
  (FIELD-MUST-CONTAIN
    (RANGE ?self field-adjacent field-midrange) ?b)
  (SIMILARITY ?b.azimuth ?self.azimuth 0.7)))
```

returns the set of all objects subject to the following criteria:

- this object and the other object are both descendants of COLLECTIVE-ANIMAL in the knowledge base; and,
- this object and the other object are in any range field from `adjacent` to `midrange` of each other; and,
- this object and the other object are facing generally in the same direction.

Defining collective concepts is also straightforward: through the existing restricted multiple inheritance (no conflicts allowed), SHEEP and WOLF in Figures 3 and 6 now additionally inherit from the new COLLECTIVE-ANIMAL, which maintains this new, shared migrating-with rule.

Not all the manually identified semantic features would be so straightforward to incorporate, of course. For the 12 tags in this study, however, extending the knowledge base is relatively easy.

Automatic Classification

The knowledge base provides the contextually appropriate, computable background knowledge to identify which of its relations apply between which objects in a configuration. This geometric inference process is documented thoroughly in Tappan (2004b, 2008a). It generates a substantial number of inferences, which correspond to unstated spatial dependencies. For example, Figure 7 shows experimental results from Tappan (2008a) for related work, where this number ranged from 27 inferences for 3 objects to 602 for 10 objects.

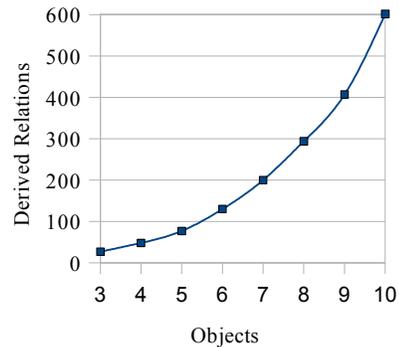


Figure 7: Inferred Relations

Defining relations for pragmatic spatial features is an iterative, experimental process. For each change, the original 20 tagged configurations were run against the updated knowledge base to determine the effectiveness at inferring any of the tags. If these results unsatisfactorily deviated from expectation, as defined in the next section, the knowledge base was manually tweaked, and the process was repeated. This approach constitutes supervised learning in machine-learning terms (Harter and

Hert 1997). The goal is to tailor the knowledge base by hand to perform well on data it has already seen, referred to as the *training set*.

Experiments

Distilling the essence of semantic features through manual training is admittedly subjective, arbitrary, and *ad hoc*. The true test of effectiveness is actually in how well the approach performs on data it has *never* seen, referred to as the *test set*. To this end, an additional 20 configurations were generated as described earlier.

These new configurations were combined with the original ones and randomly shuffled. The original participants then tagged this set as described earlier. The time between the original and subsequent tagging was three weeks to control for any residual familiarity with the originals.

Results and Discussion

For both the training and test sets, performance was evaluated according to the agreement between the results from the human taggers and the computational approach. For this approach to be effective, variation in the human performance must be considered because not every human tagged the same configurations the same way. Thus, if humans cannot determine a consistent answer, it might be unfair to expect a computer to do so.

Controlling for human inconsistency was a two-dimensional process. Lateral agreement, which was already discussed for the training set, is defined as how closely the tags for each configuration within either set agree among all the participants.

Longitudinal agreement is defined as how consistent each participant was between the original and subsequent tagging of the same configurations. This measure was intended to verify that the participants—students who knew there are no true right or wrong answers—took the task seriously and gave consistent answers. It also moderately controlled for possible survey fatigue, where participants grow tired of answering questions and put less effort into later ones (Porter et al. 2004). The system does not use this measure, but it appears to be helpful in informally interpreting the salience of the results.

The tags produced by the computational approach and the human participants can agree or disagree in four ways, as indicated in Table 4.

Type	Computer	Human
true positive (TP)	present	present
true negative (TN)	absent	absent
false positive (FP)	present	absent
false negative (FN)	absent	present

Table 4: Possible Tag Agreements

The ultimate performance measure is based on the standard approach for information retrieval (Harter and Hert 1997):

- *precision* is the accuracy or relevance of the classifications; i.e., the probability that a configuration will be classified with a correct tag. It is defined as the number of true positives divided by the sum of the true and false positives.
- *recall* is the completeness or coverage of the classifications; i.e., the probability that all the relevant configurations will be found given a tag. It is defined as the number of true positives divided by the sum of the true positives and false negatives.

Training Set

As Table 5 shows, overall agreement on the 20 configurations in the training set is perfect: 100% precision and 100% recall for all tags. These results are not surprising, however, because the knowledge base was painstakingly tailored to match these configurations precisely. This decision may have actually overfitted the data and degraded the results of the subsequent test set (Tetko et al. 1995).

Tag	TP	TN	FP	FN	Precision	Recall
migrate	3	17	0	0	1.0	1.0
graze	4	16	0	0	1.0	1.0
drink	2	18	0	0	1.0	1.0
flank	3	17	0	0	1.0	1.0
surround	5	15	0	0	1.0	1.0
conceal	6	14	0	0	1.0	1.0
stalk	7	13	0	0	1.0	1.0
chase	13	7	0	0	1.0	1.0
aware	11	9	0	0	1.0	1.0
unaware	5	15	0	0	1.0	1.0
anticipate	2	18	0	0	1.0	1.0
unknown	0	20	0	0	n/a	n/a
sum	61	179	0	0		
average					1.0	1.0

Table 5: Training Results

Test Set

The test set evaluates the computational performance on the unseen configurations. The previously seen configurations were removed because there is no point in retesting them, and they would skew the results in the positive direction. As Table 6 shows, overall agreement on the 20 remaining new configurations in the test set is respectable: 70% precision and 70% recall. Although these results are not statistically significant given the small sample size, they do suggest that this proof-of-concept work has promise.

Tag	TP	TN	FP	FN	Precision	Recall
migrate	4	13	2	1	0.7	0.8
graze	4	14	1	1	0.8	0.8
drink	3	16	0	1	1.0	0.8
flank	4	11	2	3	0.7	0.6
surround	4	9	2	5	0.7	0.4
conceal	3	12	3	2	0.5	0.6
stalk	3	11	2	4	0.6	0.4
chase	9	9	1	1	0.9	0.9
aware	10	8	1	1	0.9	0.9
unaware	11	5	2	2	0.8	0.8
anticipate	3	4	8	5	0.3	0.4
unknown	0	20	0	0	n/a	n/a
<i>sum</i>	58	132	24	26		
<i>average</i>					0.7	0.7

Table 6: Test Results

Discussion

Even for this small tag set, many independent and dependent relationships are apparent. Some tags are disjoint. For example, *graze* and *migrate* are not compatible given the small number of sheep. For a larger number, it is conceivable that one subgroup could be grazing while the other is migrating, but the configurations were not set up this way. Other tags may have overlaps. For example, *graze* and *drink* are compatible, as are *flank* and *surround*. Others may exhibit logical entailment. For example, *stalk* implies that the wolves are *aware* of the sheep, but the sheep are *unaware* of the wolves. Some have real-world semantic inconsistencies. For example, *chase* implies that each group should be aware of the other, but *aware* requires at least one sheep to be facing the wolves. In practice, fleeing sheep will not behave this way. Feature rules based on orientation appear to be more effective than those based on range, possibly because it is difficult to specify how distance affects their applicability. Finally, follow-up analysis with the participants suggests that false positives appear to be more plausible than false negatives, perhaps because it may be easier to justify the possible presence of a semantic action than to claim its complete absence.

Future Work

This pilot study is intended to direct a more detailed study. A number of extensions are under consideration:

- Incorporating a stronger statistical definition of agreement.
- Weighting the computational results to reflect the human results so multiple interpretations can be ranked by preference.
- Including some degree of articulation in the objects.

- Supporting more than one configuration snapshot to provide some degree of temporal progression in a scenario.
- Adding more breadth and depth to the objects and relations under study.
- Considering deeper plan extraction, namely strategic, tactical, and operational elements, for a top-down decomposition from *what* the objects are doing to *how* they are doing it (Azarewicz et al. 1989).
- Running simulations to determine empirical values for some of the *ad hoc* choices in the knowledge base (Tappan 2008c).
- Performing sensitivity analysis to determine how certain properties transition from one value to another.
- Allowing the participants to control the vantage points dynamically. The rendering engine allows complete control over the perspective, but this functionality is not available online. It may facilitate additional interpretations.

Conclusion

This pilot study considered the feasibility of adding higher-level, collective pragmatic analysis of objects to the existing lower-level, individual analysis in the base system. It showed respectable results within a tightly confined environment. These results are not statistically significant due to the small sample size, but they are promising. A full study would undoubtedly uncover many more implicit semantic and pragmatic dependencies.

References

- Adorni, G., Di Manzo, M., and Giunchiglia, F. 1984. Natural Language Driven Image Generation. In Proceedings of COLING-84, 495-500. Stanford, CA.
- Azarewicz, J., Fala, G., and Heithecker, C. 1989. Template-based multi-agent plan recognition for tactical situation assessment. In Proceedings of Fifth Conference on Artificial Intelligence Applications, 247-254.
- Cheng, D. and Thawonmas, R. 2004. Case-based plan recognition for real-time strategy games. In Proceedings of 5th Game-On International Conference, 36-40.
- Claus, B.; Eyferth, L.; Gips, C.; Hörnig, R.; Schmid, U.; Wiebrock, S.; and Wysotzki, F. 1988. Reference Frames for Spatial Inference in Text Understanding. In Freksa, C., Habel, C., and Wender, K., eds., *Spatial Cognition—An interdisciplinary approach to representing and processing spatial knowledge*, 214-226.
- Coyne, B. and Sproat, R. 2001. WordsEye: An Automatic Text-to-Scene Conversion System. In Proceedings of SIGGRAPH-01, 487-496, Los Angeles, CA.

- Davis, E. 1990. *Representations of Commonsense Knowledge*. San Mateo, CA: Morgan Kaufmann.
- Egenhofer, M. and Franzosa, R. 1991. Point-Set Topological Spatial Relations. *International Journal of Geographical Information Systems* 5(2): 161-174.
- Fleiss, J. 1971. Measuring nominal scale agreement among many raters. *Psychological Bulletin* 76(5): 378-382.
- Frank, A. 1992. Qualitative Reasoning about Distances and Directions in Geographic Space. *Journal of Visual Languages and Computing* 3(4): 343-371.
- Frank, A. 1996. Qualitative Spatial Reasoning: Cardinal Directions as an Example. *International Journal of Geographical Information Systems* 10(3): 269-290.
- Freska, C. 1992. Using Orientation Information for Qualitative Spatial Reasoning. In Frank, A., Campari, I., and Formentini, U., eds., *Theories and Methods of Spatio-Temporal Reasoning in Geographic Space*, LNCS 639, Springer-Verlag, Berlin.
- Gapp, K. 1994. Basic Meanings of Spatial Relations: Computation and Evaluation in 3D Space. In Proceedings of AAAI-94, 1393-1398, Seattle, WA.
- Harter, S. and Hert, C. 1997. Evaluation of information retrieval systems: approaches, issues, and methods. *Annual Review of Information Science and Technology* 32: 3-94.
- Herskovits, A. 1986. *Language and Spatial Cognition: An interdisciplinary Study of the Prepositions in English*. Cambridge, UK: Cambridge University Press.
- Hernández, D. 1994. *Qualitative Representation of Spatial Knowledge*. Berlin: Springer-Verlag.
- Hernández, D.; Clementini, E.; and Di Felice, P. 1995. Qualitative Distances. In Proceedings of Third European Conference on Spatial Information Theory, 45-58, Semmering, Austria.
- Kiefer, P. and Schlieder, C. 2007. Exploring Context-Sensitivity in Spatial Intention Recognition. In Proceedings of 1st Workshop on Behavior Monitoring and Interpretation, 102-116.
- Koller, H.; Heinze, N.; and Nagel, H.-H. 1992. Algorithmic Characterization of Vehicle Trajectories from Image Sequences by Motion Verbs. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 90-95, Maui, Hawaii.
- Lazebnik, S.; Schmid, C.; and Ponce, J. 2006. Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition.
- Neumann, B. 1989. Natural Language Description of Time-Varying Scenes. In Waltz, D., ed., *Semantic Structures: Advances in Natural Language Processing*, 167-207. Hillsdale, NJ: Lawrence Erlbaum.
- Olivier, P., and Tsujii, J. 1994. A computational view of the cognitive semantics of spatial prepositions. In Proceedings of 32nd Annual Meeting of the Association for Computational Linguistics, Las Cruces, NM.
- Peters, S. and Shrobe, H. 2003. Using Semantic Networks for Knowledge Representation in an Intelligent Environment. In Proceedings of 1st Annual IEEE International Conference on Pervasive Computing and Communications, Ft. Worth, TX.
- Porter, S., Whitcomb, M., and Weitzer, W. 2004. Multiple surveys of students and survey fatigue. *New Directions for Institutional Research* 121: 63-73.
- Randell, D.; Cui, Z.; and Cohn, A. 1992. A Spatial Logic based on Regions and Connection. In Proceedings of 3rd International Conference on Knowledge Representation and Reasoning, 165-176, San Mateo, CA.
- Russell, S. and Norvig, P. 1995. *Artificial Intelligence: A Modern Approach*. Upper Saddle River, NJ: Prentice Hall.
- Sowa, J. 2000. *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. Pacific Grove, CA: Brooks/Cole.
- Sowa, J., ed. 1991. *Principles of Semantic Networks: Explorations in the Representation of Knowledge by Computers*. New York, NY: Academic Press.
- Srihari, R. 1994. Computational Models for Integrating Linguistic and Visual Information: A Survey. *Artificial Intelligence Review* 8: 349-369.
- Stevens, A. and Coupe, P. 1978. Distortions in Judged Spatial Relations. *Cognitive Psychology* 13: 422-437.
- Tappan, D. 2004a. Knowledge-Based Spatial Constraint Satisfaction. In Proceedings of Florida Artificial Intelligence Research Society International Conference, Special Track on Spatio-Temporal Reasoning, Miami Beach, FL.
- Tappan, D. 2004b. Knowledge-Based Spatial Reasoning for Automated Scene Generation from Text Descriptions. Ph.D. diss., New Mexico State Univ., Las Cruces.
- Tappan, D. 2008a. Augmentation of Explicit Spatial Configurations by Knowledge-Based Inference on Geometric Fields. In Proceedings of 2nd International Conference on Knowledge Generation, Communication and Management, Orlando, FL.
- Tappan, D. 2008b. Knowledge-Based Spatial Reasoning for Scene Generation from Text Descriptions. In Proceedings of Association for the Advancement of Artificial Intelligence, Chicago, IL.
- Tappan, D. 2008c. Monte Carlo Simulation for Plausible Interpretation of Natural-Language Spatial Descriptions. In Proceedings of WorldComp International Conference on Artificial Intelligence, Las Vegas, NV.

- Tetko, I., Livingstone, D., and Luik, A. 1995. Neural network studies: Comparison of Overfitting and Overtraining. *Journal of Chemical Information and Computer Sciences* 35: 826-833.
- Tsotsos, J. 1985. Knowledge Organization and its Role in Representation and Interpretation for Time-Varying Data: the ALVEN System. *Computational Intelligence* 1: 16-32.
- Turner, R. 1998. Context-Mediated Behavior for Intelligent Agents. *International Journal of Human-Computer Studies* 48(3): 307-330.
- Tversky, B. 2000. Levels and structure of spatial knowledge. In *Cognitive Mapping: Past, present and future*, Kitchin, R. and Freundshuh, S., eds. London and New York: Routledge.
- Walter, I.; Lockemann, P.; and Nagel, H-H. 1988. Database Support for Knowledge-Based Image Evaluation. In *Proceedings of 13th Conference on Very Large Databases*, 3-11, Brighton, UK.
- Xu, K.; Stewart, J.; and Fiume, E. 2002. Constraint-Based Automatic Placement for Scene Composition. In *Proceedings of Human-Computer Interaction and Computer Graphics*, 25-34, Calgary, Canada.
- Yamada, A. 1993. *Studies on Spatial Description Understanding Based on Geometric Constraints Satisfaction*. Ph.D. diss., Univ. of Kyoto, Japan.
- Yamada, A., Yamamoto, T., Ikeda, H., Nishida, T., and Doshita, S. 1992. Reconstructing Spatial Image from Natural Language Texts. In *Proceedings of COLING-92*, 1279-1283, Grenoble, France.

Interval Algebra Networks with Infinite Intervals

André Trudel

Jodrey School of Computer Science
Acadia University
Wolfville, Nova Scotia, B4P 2R6, Canada

Abstract

Interval algebra networks are traditionally defined over finite intervals. In this paper, we relax this restriction by allowing one or more of the intervals involved to be infinite. We then show how algorithms developed for solving interval algebra networks with finite intervals can be used, with minor modifications, in the infinite case.

Introduction

Allen (1984) defines a temporal reasoning approach based on intervals and the 13 possible binary relations between them. The relations are before (b), meets (m), overlaps (o), during (d), starts (s), finishes (f), and equals (=) (see Table 1). Each relation has an inverse. The inverse symbol for b is b_i and similarly for the others: m_i , o_i , d_i , s_i , and f_i . The inverse of equals is equals.

A relation between two intervals is restricted to a disjunction of the basic relations, which is represented as a set. For example, $(A \text{ m } B) \vee (A \text{ o } B)$ is written as $A \{m,o\} B$. The relation between two intervals is allowed to be any subset of $I = \{b, b_i, m, m_i, o, o_i, d, d_i, s, s_i, f, f_i, =\}$ including I itself.

An IA (interval algebra) network is a graph where each node represents an interval. Directed edges in the network are labeled with subsets of I. By convention, edges labeled with I are not shown. An IA network is consistent (or satisfiable) if each interval in the network can be mapped to a real interval such that all the constraints on the edges hold (i.e., one disjunct on each edge is true).

A scenario of an IA network is a singleton labeling of the network (i.e., each edge only has one of its original labels). A consistent scenario is a scenario where each constraint on each edge is true. An IA network is consistent if and only if it has a consistent scenario.

Intervals in Allen's interval algebra are finite and convex. In this paper, the finiteness condition is relaxed. In addition to finite intervals, intervals that are half infinite towards negative infinity, half infinite towards positive infinity, and infinite in both directions are allowed.

Efficient algorithms and implementations have been developed for solving IA networks (van Beek and Manchak 1996). These algorithms were specifically developed for finite intervals. In this paper, we show how these algorithms can be used, with minor modifications, to solve IA networks which contain non-finite intervals.

In the next section, we catalogue the different possible relationships among finite and non-finite intervals. Then, we show how to solve IA networks that possibly contain non-finite intervals. Note that we retain the convex property for all the intervals, including the non-finite ones.

Table 1: Possible relationships between two finite intervals

Relation	Symbol	Example
X before Y	b	
X meets Y	m	
X overlaps Y	o	
X starts Y	s	
X during Y	d	
X finishes Y	f	
X equals Y	=	

Possible relationships involving non- finite intervals

The following graphical notation is used for intervals:

- ∞ Finite length interval (i.e., *finite*):
- ∞ Fixed endpoint on the left and infinite on the right (i.e., *right-infinite*):
- ∞ Fixed endpoint on the right and infinite on the left (i.e., *left-infinite*):
- ∞ Infinite in both directions (i.e., *infinite*):

The possible relationships between two intervals when one or both are non-finite are not obvious. They are catalogued in this section. Table 2 shows the possible relations between a finite and right-infinite interval. Each entry in Table 2 also has an inverse. For example, in the first row finite interval X is before (b) right-infinite interval Y. It is also the case that Y is after (bi) X. All the possible cases are shown in Table 2 - Table 10.

Table 2: Possible relationships between a finite and a right-infinite interval

Relation	Symbol	Example
X before Y	b	
X meets Y	m	
X overlaps Y	o	
X starts Y	s	
X during Y	d	

Table 3: Possible relationships between a finite and a left-infinite interval

Relation	Symbol	Example
X after Y	bi	
X met by Y	mi	
X overlapped by Y	oi	
X finishes Y	f	
X during Y	d	

Table 4: Possible relationships between a finite and an infinite interval

Relation	Symbol	Example
X during Y	d	

Table 5: Possible relationships between two right-infinite intervals

Relation	Symbol	Example
X finished by Y	fi	
X equals Y	=	
X finishes Y	f	

Table 6: Possible relationships between a right-infinite and a left-infinite interval

Relation	Symbol	Example
X overlaps Y	o	
X meets Y	m	
X before Y	b	

Table 7: Possible relationships between a right-infinite and an infinite interval

Relation	Symbol	Example
X finishes Y	f	

Table 8: Possible relationships between two left-infinite intervals

Relation	Symbol	Example
X started by Y	si	
X equals Y	=	
X starts Y	s	

Table 9: Possible relationships between a left-infinite and an infinite interval

Relation	Symbol	Example
X starts Y	s	

Table 10: Possible relationships between two infinite intervals

Relation	Symbol	Example
X equals Y	=	

A summary of all the possible relations appears in Table 11. The entry in row *r* and column *c* is all the possible relations between an interval of type *r* and *c*. For example, if *X* is left-infinite and *Y* is right-infinite, the only possibilities are $X \{o, m, b\} Y$. It is interesting to note that each entry in Table 11 also appears as entries in Allen's (1983) full composition table.

Table 11: Summary

	—	→	←	↔
—	b, bi, m, mi, o, oi, d, di, s, si, f, fi, =	b, m, o, s, d	bi, mi, oi, f, d	d
→	bi, mi, oi, si, di	fi, =, f	oi, mi, bi	f
←	b, m, o, fi, di	o, m, b	si, =, s	s
↔	di	fi	si	=

Important property: Let *X* and *Y* be intervals of type finite, right-infinite, left-infinite, or infinite. Note that *X* and *Y* may or may not be of the same type, and if they are of the same type they may not be equal. Assume there is a valid relation *r* which holds between *X* and *Y* (i.e., $X \{r\} Y$). Now map both *X* and *Y* to real intervals. If *X* and/or *Y* extend towards negative infinity, chop off these intervals at a very small negative value called LE. If *X* and *Y* contain any finite endpoints, they are all larger than LE. Do the same on the right hand side and chop all intervals going towards positive infinity at a large positive point RE. Both *X* and *Y* are now finite. One can verify manually that it is the case that regardless of the original intervals and relation chosen, it is still the case that $X \{r\} Y$. An example of the construction is shown in Figure 1. *X* is right-infinite, *Y* is infinite, and $X \{f\} Y$. Both intervals are made finite by chopping the infinite ends. *X*'s left endpoint is not modified. It remains the case that $X \{f\} Y$ for the finite versions of the intervals.

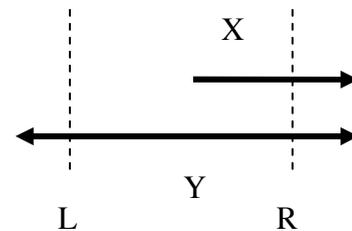


Figure 1: $X \{f\} Y$

Algorithm

A simple IA network is shown in Figure 2. Assume the intervals are of the following type:

- ∞ A: ---
- ∞ B: ←←
- ∞ C: →→
- ∞ D: ↔↔

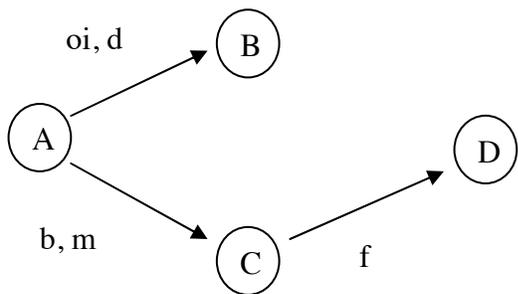


Figure 2: IA network

If the network in Figure 2 is used as input to standard IA network finite interval software, we could generate the solution shown in Figure 3. This solution is correct if all the intervals are finite. It is incorrect in our case. It is impossible for the interval B to be before (b) the infinite interval D. The problem is not with the software, but with the input given to the software.

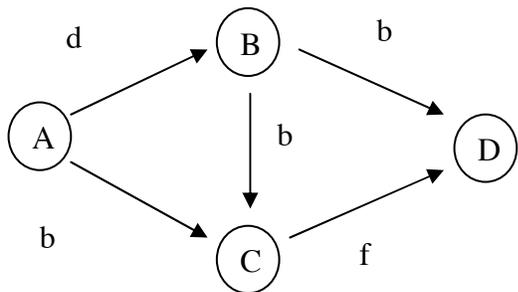


Figure 3: Consistent finite interval scenario

Figure 2 has three hidden edges:
AD, BC, and BD

which are each assumed by the software to have a label of I. This is causing the problem. The label I on each of these edges is incorrect because not all the labels within I are possible. For example, B cannot be before (b) the infinite interval D. To rectify this situation, instead of labeling missing edges with I as is the standard practice, we must

label missing edges with the appropriate entry from Table 11. The correctly labeled network from Figure 2 is shown in Figure 4. Note that the label “oi,mi,bi” in the center of the figure belongs to the edge CB. This label is the entry in row:



and column:



in Table 11.

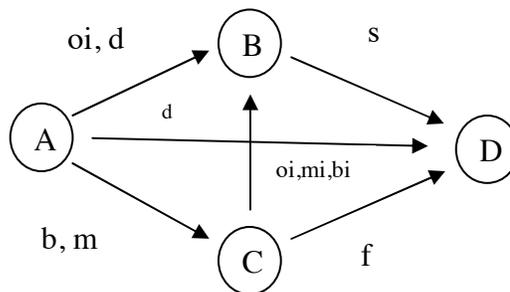


Figure 4: IA network with missing edges properly labeled

If we now input the network in Figure 4 to standard IA network finite interval software, we could generate the correct scenario shown in Figure 5. The scenario in Figure 5 represents the relative arrangement of intervals shown in Figure 6.

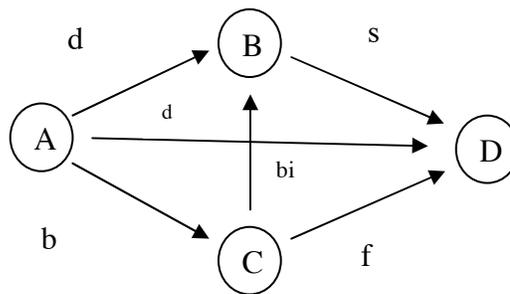


Figure 5: Correct non- finite interval scenario

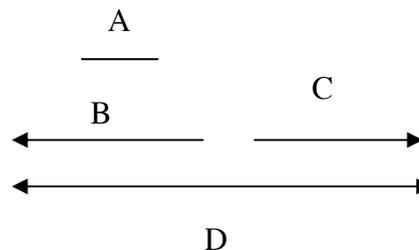


Figure 6: Solution involving non- finite intervals

Proposed Algorithm: Given an IA network with non-finite intervals. First add missing edges labeled with the appropriate entry from Table 11. Then, apply finite interval IA network solution software to the network. The software will either report that the original network is inconsistent, or return a consistent scenario for it. The algorithm is simple, and shown to be correct in the next section.

Correctness

Let NF-IAN be an IA network with one or more non-finite intervals. If NF-IAN has any missing edges, add them to NF-IAN along with the appropriate label from Table 11. Let another IA network F-IAN have the same nodes, edges, and labels as NF-IAN. The only difference between the two is that all the intervals in F-IAN are finite. The following theorem proves the correctness of the proposed algorithm:

Theorem: S is a consistent scenario of NF-IAN if and only if it is a consistent scenario of F-IAN.

Proof: Assume S is a consistent scenario of NF-IAN. We can map all the intervals in S to intervals over the real numbers such that each relation in S holds. As we did in Section 2, we chop off all the left-infinite and infinite intervals on the left hand side at some arbitrary number LE which is smaller in value than any of the finite endpoints which may occur in any of the other intervals. We do the same with right-infinite and infinite intervals on the right hand side. They all get chopped at the same point RE . As observed in Section 2, this chopping does not affect the individual relations between pairs of intervals. All the edge labels in S will still hold and all the intervals are finite. We therefore have a consistent scenario for F-IAN. For example, assume NF-IAN is the network in Figure 4, F-IAN is the same network where all the intervals are finite, and S is the scenario represented by Figure 6. The non-finite intervals of S are chopped in Figure 7, and this is a scenario for F-IAN.

Now consider the case where S is a consistent scenario of F-IAN. Map all the intervals in S to finite intervals over the real numbers such that each relation in S holds. Let LI be the set of left-infinite and infinite intervals in NF-IAN. If LI is non-empty, the following will be the case:

1. The finite intervals in S corresponding to the intervals in LI will all have the same

left endpoint. No other intervals will have this left endpoint. This follows from the possible relations in Table 11 involving left-infinite and infinite intervals.

2. It also follows from Table 11 that no interval endpoint will be to the left of the left endpoints of the intervals in S which correspond to the intervals in LI .

Based on the above properties, we can shift the left endpoints of the intervals in S which correspond to the intervals in LI any arbitrary distance towards negative infinity without violating any relations in S . Push these endpoints all the way to negative infinity. Now perform the same operations in the opposite direction with the right endpoints of the intervals which correspond to the right-infinite and infinite intervals, if there are any. None of the relations in S have been violated. S has been transformed into a consistent scenario for NF-IAN. For example, let F-IAN be the network in Figure 4 and S be the scenario represented in Figure 8. Intervals B and D can be extended on the left towards infinity and, C and D can be extended on the right towards infinity without violating any of the relations in S . This is now a scenario for the NF-IAN version of Figure 4.

Therefore, S is a consistent scenario of NF-IAN if and only if it is a consistent scenario of F-IAN. Q.E.D.

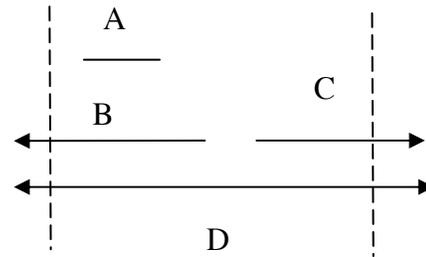


Figure 7: Chopping the non-finite intervals

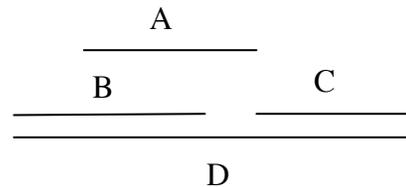


Figure 8: Finite scenario S

Previous work

The concern with infinity is not limited to AI in computer science. For example, the IEEE standard 754-1985 for binary floating-point arithmetic has special values assigned to positive and negative infinity. The default in IEEE arithmetic is to round overflowed numbers to infinity (Goldberg 1991).

I am not aware of any previously published algorithm for solving IA networks containing non-finite intervals.

The problem of representing non-finite intervals was not addressed in this paper. Hobbs (2002), and in a later collaboration with Pan (2004) present a succinct and elegant first order axiomatization of Allen's relations for non-finite intervals. Although they make minimal ontological commitments, the axiomatization contains instants, intervals can have endpoints, and relies on the fact that a left-infinite interval has no left endpoint (similarly for right-infinite and infinite intervals). This seems to imply a point based model of time. Note that a model does not necessarily need to contain an infinite number of points to represent the non-finite intervals.

Another elegant and formal axiomatization of Allen's intervals for non-finite intervals appears in (Cukierman and Delgrande 2004). This axiomatization includes predicates to distinguish between the various types of non-finite intervals. This capability is missing in the logics presented in (Hobbs 2002) and (Hobbs and Pan 2004), where the user must extend the logic.

Bouzid and Ladkin (2002) define temporal intervals as the union of convex finite intervals. They also define operations over these sets. One operation is union. Since their underlying structure is the rationals, they require this infinite interval in their system since the union of a set and its complement is the set of all the rationals. Positive and negative infinity are represented by adding two points at infinity to the set of rationals. They do not consider the possible relationships between infinite sets.

Infinitely periodic temporal data has been studied in the temporal database area. Baudinet et. al. (1991) and Kabanza et. al. (1995) consider infinite sequences of finite intervals over the integers. Note that the individual intervals are finite; not infinite.

Another paper from the temporal database area is by Koubarakis (1994) which uses the rationals as the underlying temporal structure. He does not consider infinitely periodic temporal data. But, since he is using an underlying dense temporal structure, he considers the fact that temporal information can be true at infinitely many points over a finite interval to be infinite temporal information. He does not allow non-finite intervals.

Conclusions

Allen's interval algebra was originally defined and axiomatized for finite temporal intervals. Subsequently, efficient algorithms were implemented for solving finite interval IA networks.

This paper shows that, unless the problem domain is inherently finite, there is no reason to restrict temporal intervals to be finite. Allen's relations have been axiomatized for non-finite intervals (e.g., (Cukierman and Delgrande 2004) and (Hobbs and Pan 2004)). Existing finite interval software can be used to solve non-finite interval AI networks. Care must be taken to properly label missing IA network edges with the proper entry in Table 11.

References

- Allen, J.F. 1983. Maintaining knowledge about temporal intervals, *Communications of the ACM*, 26, p. 832-843.
- Allen, J.F. 1984. Towards a general model of action and time, *Artificial Intelligence*, 23(2), p. 123-154.
- Baudinet, M., Niezette, M., and Wolper, P. 1991. On the representation of infinite temporal data and queries, 10th ACM Symposium on Principles of Database Systems, Denver.
- Bouzid, M., and Ladkin, P. 2002. Simple reasoning with time-dependent propositions, *Logic Journal of the IGPL*, Vol. 10, No. 4, pp. 379-399.
- Cukierman, D.R., and Delgrande, J.P., 2004. A theory for convex interval relations including unbounded intervals, *Proceedings of the 17th International FLAIRS Conference*, Miami Beach, Florida.
- Goldberg, D. 1991. What every computer scientist should know about floating-point arithmetic, *ACM Computing Surveys*, Vol. 23, No. 1, pp. 5-48.
- Hobbs, J.R., 2002. A DAML Ontology of Time, Retrieved on April 8 2009 from: <http://www.cs.rochester.edu/~ferguson/daml/daml-time-nov2002.txt>
- Hobbs, J.R. and Pan, F. 2004. An ontology of time for the semantic Web, *ACM Transactions on Asian Language Information Processing*, Vol. 3, No. 1, March, p. 66-85.
- Kabanza, F., Stevenne, J-M., and Wolper, P., 1995. Handling infinite temporal data, *Journal of Computer and System Sciences*, Vol. 51, No. 1, pp. 3-17.

Koubarakis, M., 1994. Database models for infinite and indefinite temporal information, *Information Systems*, Vol. 19, No. 2., pp. 141-173.

van Beek, P. and Manchak, D.W. 1996. The design and experimental analysis of algorithms for temporal reasoning, *Journal of Artificial Intelligence Research*, 4, p. 1-18.