

ECAI 2012

Workshop proceedings

Spatio-Temporal Dynamics



STeDy 2012
European Conference on Artificial Intelligence
Montpellier, France

August 27 and 28 2012

Organizing and Editorial Committee

Mehul Bhatt
SFB/TR 8 Spatial Cognition
University of Bremen
P.O. Box 330 440, 28334 Bremen, Germany
T +49 (421) 218 64 237
F +49 (421) 218 98 64 237
bhatt@informatik.uni-bremen.de

Hans W. Guesgen
School of Engineering and Advanced Technology
Massey University
Private Bag 11222, Palmerston North 4442, New Zealand
T +64 (6) 356 9099 extn 7364
F +64 (6) 350 2259
h.w.guesgen@massey.ac.nz

Ernest Davis
Dept. of Computer Science
New York University
Room 329, 251 Mercer St.
New York NY 10012
T +1 (212) 998 3123
F +1 (212) 995 4121
davise@cs.nyu.edu

Advisory and Program Committee

Alia Abdelmoty (Cardiff University, UK)
Andrew Lovett (Northwestern University, USA)
Antony Galton (University of Exeter, UK)
Bernhard Nebel (University of Freiburg, DE)
Christian Freksa (University of Bremen, DE)
Christophe Claramunt (Naval Academy Research Institute, FR)
Diedrich Wolter (University of Bremen, DE)
Ernest Davis (New York University, USA)
Esra Erdem (Sabanci University, TR)
Frank Dylla (University of Bremen, Germany)
Gerhard Lakemeyer (Aachen University of Technology, DE)
Hans Guesgen (Massey University, NZ)
Inderjeet Mani (Children's Organization of Southeast Asia)
Jean-Francois Condotta (University of Artois, FR)
John Stell (University of Leeds, UK)
Joohyung Lee (Arizona State University, USA)
Kathleen-Stewart Hornsby (University of Iowa, USA)
Matt Klenk (PARC, USA)
Mehul Bhatt (University of Bremen, DE)
Nico Van de Weghe (Ghent University, BE)
Paulo E. Santos (Technical University FEI, BR)
Philippe Muller (Université Paul Sabatier, FR)
Reinhard Moratz (University of Maine, USA)
Shyamanta Hazarika (Tezpur University, IN)
Stefan Woelfl (University of Freiburg, DE)
Vinay Chaudhri (SRI International, USA)

Organizing Institutions

SFB/TR 8 Spatial Cognition. University of Bremen, and University of Freiburg (GERMANY)
Massey University (NEW ZEALAND)
New York University (USA)

Contents

Preface	
<i>Mehul Bhatt, Hans Guesgen, and Ernest Davis</i>	iv
Invited Talks	
<i>Patrick Doherty</i>	v
<i>Christian Freksa</i>	vi
<i>Gérard Ligozat</i>	vii
Full Papers	
1 Dynamics of a Nearness Relation - First Results <i>Özgür Lütfü Özcep, Rolf Grütter and Ralf Möller</i>	1
2 Socially Compliant Navigation in Crowded Environments <i>Frank Dylla, Martin Coors, and Mehul Bhatt</i>	9
3 Asynchronous Decentralized Algorithm for Space-Time Cooperative Pathfinding <i>Michal Čáp, Peter Novák, Jiří Vokřínek, and Michal Pěchouček</i>	18
4 Now: Between Indexical Thoughts, and Indexical Thought <i>Haythem Ismail</i>	27
5 Knots World: an investigation of actions, change and space <i>Paulo Santos, and Pedro Cabalar</i>	36
6 Reasoning about Relative Relationships in 3D Space for Objects Extracted from Dynamic Image Data <i>Kazuko Takahashi</i>	45
7 On the Neighborhood and Distance Between Qualitative Spatio-temporal Configurations <i>Dominique D'Almeida, Mouny Samy Modeliar, and Nicolas Schwind</i>	53
8 Probabilistic Region Connection Calculus <i>Codruta Girlea, and Eyal Amir</i>	62

Preface

Welcome to the Workshop on Spatio-Temporal Dynamics (STeDy) at the European Conference on Artificial Intelligence 2012 in Montpellier, France. The workshop is a follow-up of the STeDy 2010 workshop organised at the ECAI 2010 conference in Lisbon, Portugal.

Knowledge Representation and Reasoning has been one of the key thrust areas within Artificial Intelligence research. Driven by the motivation for a qualitative approach for the embodiment of commonsense spatial knowledge in intelligent systems, Qualitative Spatial Information Theory has emerged as a discipline within Artificial Intelligence. Located within this discipline are specialisations concerned with the development of formal methods to represent and reason about Space, Time, Actions and Change.

Space, actions and change are inextricably linked: actions and events are a crucial connecting-link between space and spatial change, e.g., spatial configurations typically change as a result of interaction within the environment. Actions and events, both in a predictive as well as an explanatory sense, also constitute the mechanisms by which we establish and nurture commonsense knowledge about the world that we live in: our anticipations of spatial reality conform to our commonsense knowledge of the effects of actions and events in the real world. Similarly, our explanations of the perceived reality too are established on the basis of such a priori established commonsense notions. This view of Space, Actions and Change is general and applicable in a wide-range of application areas: qualitative spatial and temporal reasoning in general and formalising spatial change in particular is increasingly becoming a core issue within many application domains such as Robotics and Computer Vision, Ambient Intelligence and Smart Environments, Spatial / Architecture Design, GIS / Spatial Information Systems, Mobile and Location-based Computing.

The edited volume covers both theory and application-centric research in the area of spatio-temporal dynamics. Thrust is on research that focuses on formalising commonsense spatial knowledge and directs the integration of qualitative spatial reasoning with general approaches for reasoning about action and change. Applications that demonstrate the utility of well-established qualitative spatial and temporal calculi are also covered.

The proceedings of STeDy 2012 would be a contribution primarily to Spatio-Temporal Representation and Reasoning within Qualitative Spatial Information Theory. Additionally, it is envisaged that the results will also offer direct guidance to other AI Practitioners for the application of formal methods in spatio-temporal dynamics in their respective disciplines.

Mehul Bhatt, Hans Guesgen, Ernest Davis
(STeDy 2012 Co-Chairs)

Invited Talk

Leveraging KR Techniques in Autonomous Unmanned Aircraft Systems

In this talk I will describe a number of autonomous unmanned aircraft systems we have developed and used as research platforms during the past decade. A major focus of our research effort has been to push AI technologies into fully deployed unmanned aircraft systems. I will show how we have leveraged different knowledge representation systems and integrated them into our platforms. Particular systems that will be discussed are automated temporal planning systems, execution monitoring systems, stream-based reasoning systems and frameworks for cooperative robotics based on delegation. Use of these techniques will be demonstrated in diverse emergency services applications using both single and multiple platform scenarios. Both simulated and actual missions will be shown.

Patrick Doherty

Professor of Computer Science, Linköping University
SWEDEN

Invited Talk

Spatial Computing for Commonsense Reasoning

My contribution deals with the relation between natural and formal descriptions of commonsense knowledge and cognitive processes on one hand and their implementations in natural and artificial cognitive systems, on the other hand. The expressive power of general languages and formalisms by far exceeds the structural power of physical systems they describe. In particular, natural and formal languages can describe hypothetical situations and processes which cannot exist in the physical world.

In AI, we use formal languages to abstractly characterize knowledge and commonsense reasoning on the meta-level. The strengths of the formalisms for characterizing cognitive systems may turn out to be a weakness for modeling and implementing them on the object level, where structural constraints of the medium control decisions and actions. I will present some of the structural features of cognitive systems that support cognition and I will discuss spatial computing as an object-level approach to implementing cognitive principles of spatial and temporal processing.

Christian Freksa

Professor of Informatics, University of Bremen

GERMANY

Invited Talk

Three decades of Qualitative Spatial and Temporal Reasoning

Allen's 1983 paper on the Interval Algebra introduced constraint propagation techniques to the domain of temporal reasoning. During the following three decades, a significant body of research has been devoted to studying the properties of Allen's formalism, as well as to proposing a host of spatial and temporal formalisms modeled on it.

Once established that the problem of determining consistency for the Interval Algebra is a NP-complete problem, the search for tractable sub-classes of relations in it, and the analogous question for other formalisms, have occupied a central position in the literature. Basically, two approaches have been used for characterizing tractable subsets: a syntactical approach, and a geometrical approach.

As a consequence of the emergence of many related formalisms, the search for a general framework has also become a timely topic. The introduction of new frameworks has resulted in clarifying the nature of the formalisms, their properties, their expressiveness and the nature of their models.

The relationship of these formalisms with the domain of (finite) CSPs has also been exploited, both on a theoretical level (relating complexity properties to properties of the associated clones) and on a practical level (devising efficient methods for solving consistency problems).

Our talk will start with a presentation of Allen's formalism from a contemporary perspective, highlighting the points on which important issues have been considered and solved. It will then present some of the main formalisms which have been developed in recent years, and stress the commonalities and differences they exhibit. A general framework based on the notion of a partition scheme will then be discussed and its use for classifying problems and properties will be illustrated. Various extensions to the basic qualitative calculi, such as hybrid calculi and fuzzy calculi will then be presented. Finally, the talk will conclude on a discussion of currently developed approaches as well as long term perspectives for further research.

Gérard Ligozat

Emeritus Professor, Paris-Sud University
FRANCE

Dynamics of a Nearness Relation—First Results

Özgür L. Özcep¹ and Rolf Grütter² and Ralf Möller³

Abstract. The system of administrative units for a state like Switzerland can be formally described by a totally ordered set of nested partitions where, e.g., the municipalities make up a finer partition than the partition induced by districts. Based on these partitions one can define binary non-symmetric nearness relations between regions in which the second argument determines the granularity (or the scaling level) w.r.t. which the first-argument region is to be considered near or not. The logical properties of such nearness relations, especially w.r.t. their relation to proximity relations, have been worked out in the authors' contribution to ECAI 2012. Referring to these properties, in this paper we extend the investigation to the dynamics of the nearness relation. In particular, we investigate how a change within the total order of partitions (e.g., two municipalities are merged) affects the induced nearness relation.

1 INTRODUCTION

The nearness relation whose dynamics we are going to discuss is defined on the basis of a hierarchy of nested regions which make up a total order of partitions [3], [7], [8]. Typical examples of such total orders of nested partitions are made up of administrative units where the administrative units in a rougher granularity are the sums (unions) of administrative units of the lower level. As an example think of two partitions of Switzerland, where the first partition consists of municipalities and where the second consists of districts. All districts are municipalities or are unions of two or more municipalities.

Every partition provides a granularity or scale w.r.t. which the nearness of two regions are declared; the main idea is to consider one of the arguments (we took the second one) to determine the scaling context that is the level on the ground of which two regions are defined to be near or not. There are different ways to exploit the nested partitions pc (which mathematically is a totally ordered set of partitions and hence termed *partition chain*) for defining nearness relations. We will fix a specific type of nearness relation NR_{pc} induced by a partition chain pc which has some desirable properties.

Having constructed such a nearness relation one can consider its properties in a mathematically abstract way by declaratively specifying properties of a binary relation δ in a formal language like first order logic. In previous work, we described the properties that every right-scaled proximity nearness relation NR_{pc} induced by a partition chain pc has [7], [8]. In this paper, we add some further properties of this type and extend the investigations in two ways: we describe the local dynamics of nearness relations, that is we describe how a change from one region to another (in the second argument) affects

the set of regions considered to be near. For this purpose, we describe properties that directly refer to the given (unchanged) partition chain. Second we investigate the question how the change of the partition chain affects the nearness relation, i.e., we investigate the global dynamics of nearness. More concretely, assume a partition chain pc_1 is changed to a new partition chain pc_2 ; what can we say about the change from the induced nearness relation NR_{pc_1} to the induced nearness relation NR_{pc_2} ? In particular, one can ask what kind of change transitions $pc_1 \rightsquigarrow pc_2$ do not change the nearness relation, $NR_{pc_1} = NR_{pc_2}$, or between what regions (on what level) does a change of the total orderings affect the nearness in between them. Similar problems have been tackled by [4] and especially [11], which considers the global dynamics of tree-like spatial configurations.

The change transition \rightsquigarrow between total orders are not allowed to be arbitrary transitions but some intuitive changes which have corresponding real world counterparts. In particular, the kind of changes that are worth being investigated are the merger of regions, the switch of levels, the additions of partitions etc.

Investigations into this kind of relation are necessary for a formal theory of dynamics of nearness. In particular such a theory provides a formal grounding for optimizations within a cognitive agent that bases its nearness relation on partition chains; rather than recalculating the nearness relation between all regions in case the agent moves around (local change) or a partition chain is updated (global change) it directly uses the knowledge on regions between which the nearness relation is expected to have changed. In this work, we lay the foundations for such a theory; thereby we give preliminary results on the local dynamics and the global dynamics of the nearness relation. Concerning the latter we focus on the effects on the nearness relation resulting from merging two regions in the same partition level.

The paper is structured as follows. In Section 2 we describe the main structure for our nearness relations, the partition chain. A specific nearness relation is defined and illustrated in Sect. 3. The following two sections 4 and 5 describe properties of the nearness relation, the former more abstractly by referring only to properties describable by an abstract binary relation, the latter referring also to the underlying partition chain, thereby providing insights into the local dynamics of the nearness relation. The last section before the conclusion starts the preliminary investigation into dynamic aspects of the nearness relation.

2 NORMAL PARTITION CHAINS

In this section we recapitulate the notion of a partition chain underlying the formal framework of a nearness relation as developed in [8], and specify the special class of normal partition chains, which is the main structure for the nearness relation. Different from [8], in this paper, we abstract from the region connection calculus [9], and hence define partition chains and nearness relations only on the basis

¹ Institute for Software Systems (STS), Hamburg University of Technology, Germany, email: oezguer.oezcep@tu-harburg.de

² Swiss Federal Research Institute WSL, Birmensdorf, Switzerland, email: rolf.gruetter@wsl.ch

³ Institute for Software Systems (STS), Hamburg University of Technology, Germany, email: moeller@tu-harburg.de

of the usual set theoretical notions.

As we want to allow more cells to be on different levels, we have to type the sets. This is needed for modelling situations in the real world where the same spatial region may have two different administrative functions.⁴ Hence we define the following notion of partition:

Definition 1 (partition). *Under a partition of a set X on level $i \in \mathbb{N}$ we understand a family of pairs $(i, a_j)_{j \in J}$ such that $(a_j)_{j \in J}$ is a (set) partition of X , i.e., $X = \bigsqcup_{j \in J} a_j$ where \sqcup indicates a union of disjoint sets and J is a finite index set. A pair $c = (i, a_j)$ is called a cell of level i . Its level i is denoted $l(c)$ and its underlying set a_j (the second argument) is denoted $us(c)$. The usual mathematical notion of a partition will be called set partition.*

Now we look at $n + 1$ different partitions of X that are nested or more formally: totally ordered from 0 to n . This is concretised in the following definition.

Definition 2 (partition chain). *Let be given $n + 1$ different partitions of X where all partitions have only finitely many cells. We call this set of partitions a partition chain pc iff*

1. *all cells $(i + 1, a_j)$ of level $i + 1$ (for $i \in \{0, \dots, n - 1\}$) are unions of i -level cells, i.e., there exist (i, b_k) , $k \in K$, such that $a_j = \bigsqcup_{k \in K} b_k$;⁵*
2. *and the last partition (level n) is made up by (X) .*

According to this assumption, every cell has a unique upper cell. For a cell (i, a_j) (with $1 \leq i \leq n - 1$) let $(i, a_j)^{\uparrow, pc} = (i + 1, a_k)$ be the unique cell of the upper level in this partition chain pc such that $a_j \subseteq a_k$. For the cell of level n set $(n, X)^{\uparrow, pc} = (n, X)$. We call $(i, a_j)^{\uparrow, pc}$ the upper cell of (i, a_j) . If the partition chain is clear from the context, we write $(i, a_j)^{\uparrow}$ for $(i, a_j)^{\uparrow, pc}$.

Between cells (i, a) and (j, b) (perhaps of different partition chains) we define an order \leq by setting $(i, a) \leq (j, b)$ iff $i \leq j$ and $a \subseteq b$.

This definition is too general in order to be used for an interesting nearness notion as it also allows for a configuration where all underlying sets of cells in a partition re-occur in the partition of the next upper level. An example for such an unusual partition is given as follows: let $X = a_1 \sqcup a_2$ and let for $i \in \{0, 1, 2, 3\}$ be given the partition a^i of level i by $((i, a_1), (i, a_2))$; the partition of level 3 shall be $(3, X)$. We exclude such partition chains by defining the notion of a *normal partition chain*, in which it is allowed that a set is the underlying sets of two different levels $i, i + 1$, but only if the underlying set partitions on level i and $i + 1$ are different.

Definition 3 (normal partition chain). *A partition chain is normal iff all set partitions underlying the partitions are pairwise distinct.*

In practical real-world applications the induced partition can pretty safely assumed to be normal as otherwise a distinction between the administrative units would not even be introduced. But, as in the case of normal partition chains, it may be the case that the same region has two different administrative functions.

⁴ Note that in case of the region connection calculus, this typification can be handled directly by allowing non-strict models [10], that is models in which two objects may stand in EQ-relation (same spatial extension) without being identical.

⁵ Perhaps in future work we have to reverse the ordering so that we can also consider infinite ordering of partitions: then we may have at level 0 the roughest partition X and at higher levels more fine-grained partitions ad infinitum.

Due to the fact that the total order is finite and that all partitions are finite one can easily describe all possible normal partition chains. For illustration of the notion of a normal partition chain, we will describe the normal partition chains induced by a given set partition (a) with n cells for different n up to $n = 3$.

Example 1. *In this example we write $i : (a)$ as shorthand for partitions $(i, a_j)_{j \in J}$.*

- *If $n = 1$, then the partition (a) is the partition (X) and we do have only the order of partitions containing (X) .*
- *Let $n = 2$, i.e. let $X = a_1 \sqcup a_2$. We can only have the order of partitions $0 : (a) \leq 1 : (X)$ and $0 : (X)$.*
- *Let $n = 3$, $X = a_1 \sqcup a_2 \sqcup a_3$. We may have*
 - $0 : (a) \leq 1 : (X)$
 - $0 : (a_1, a_2 \cup a_3) \leq 1 : (X)$
 - $0 : (a_1 \cup a_2, a_3) \leq 1 : (X)$
 - $0 : (a_1 \cup a_3, a_2) \leq 1 : (X)$
 - $0 : (a) \leq 1 : (a_1, a_2 \cup a_3) \leq 2 : (X)$
 - $0 : (a) \leq 1 : (a_1 \cup a_2, a_3) \leq 2 : (X)$
 - $0 : (a) \leq 1 : (a_1 \cup a_3, a_2) \leq 2 : (X)$

3 NEARNESS BASED ON PARTITION CHAINS

Having defined the main structure for a hierarchical nearness relation, we are now in a position to define the notions of apriori nearness and (general) nearness w.r.t. a partition chain as follows:

Definition 4 ((apriori) nearness NR). *Let be given a partition chain pc . Cell $c_1 = (i, a_1)$ is apriori near $c_2 = (i, a_2)$, $NR_{pc}^{ap}(c_1, c_2)$ for short, iff there is a cell $(i + 1, b)$ of level $i + 1$ such that, $a_1, a_2 \subseteq b$, i.e., iff the upper cells of c_1, c_2 are the same. An arbitrary set a is near a cell $c_1 = (i, a_1)$ of level i , $NR_{pc}(a, c_1)$ for short, iff there is a cell $c_2 = (i, a_2)$ of the same level of c_1 such that $NR_{pc}^{ap}(c_1, c_2)$ and $a \cap a_2 \neq \emptyset$.*

For an arbitrary set $b \neq \emptyset$ let \tilde{b}^{pc} denote the cell (i, a_j) such that $b \subseteq a_j$ and i is minimal. The integer $i = l_{pc}(b)$ is called the level of b in pc . For arbitrary sets a, b we define nearness by:

$$NR_{pc}(a, b) \text{ iff } NR_{pc}(a, \tilde{b}^{pc})$$

If the partition chain pc is unique in the used context, then we do not mention it in the subscripts.

As a shorthand for $(\tilde{b}^{pc})^{\uparrow, pc}$ we write $b^{\uparrow, pc}$ or even shorter b^{\uparrow} .

Note that we excluded the empty set as a second argument b , as we cannot define $\tilde{\emptyset}$. For the empty set as left-hand argument we get that not $NR(\emptyset, b)$ for all $b \subseteq X$.

The apriori nearness relation underlying the nearness relation of [8] is different from the one defined here. We chose to work with this definition as it has a very simple equivalent form which does not need the detour with apriori nearness. An arbitrary set a is near a cell (i, a_1) if the intersection with the underlying set of the upper cell of (i, a_1) is non-empty.

Proposition 1. *The nearness relation NR can be equivalently described as follows:*

$$NR(a, b) \text{ iff } a \cap us(b^{\uparrow}) \neq \emptyset \quad (1)$$

Proof. If $a \cap us(\tilde{b}^\uparrow) \neq \emptyset$, then there is a cell $c_1 = (i, b_1)$ in the same level as \tilde{b} such that $a \cap b_1 \neq \emptyset$, because a has a nonempty intersection with the upper cell of \tilde{b} and this upper cell is a union of cells of the level of \tilde{b} . But by definition $NR^{ap}(c_1, \tilde{b})$. So $NR(a, b)$. The argument for the other direction works similarly. \square

The corresponding equivalent definition for NR within the framework of RCC (or more general: a region-based framework with a connectedness relation C) would be:

$$NR(a, b) \text{ iff } C(a, us(b^\uparrow)) \quad (2)$$

The following example illustrates the nearness relations.

Example 2. We define a partition chain with four levels as illustrated in Figure 1. Let $X = \{1, \dots, 6\}$, $a_i = \{i\}$ for $1 \leq i \leq 4$ and

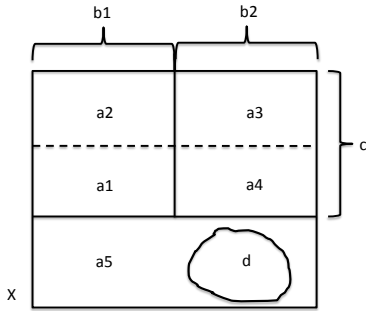


Figure 1. Illustration of configuration in Example 2

$a_5 = \{5, 6\}$. All a_i are sets underlying cells of level 0. The set $d = 6$ is an arbitrary set (region) which does not underly any of these cells. The partitions on the levels 0 to 3 are defined by:

- $\{(0, a_1), (0, a_2), \dots, (0, a_5)\}$
- $\{(1, b_1), (1, b_2), (1, a_5)\}$ where $b_1 = a_1 \cup a_2$ and $b_2 = a_3 \cup a_4$.
- $\{(2, c), (2, a_5)\}$ where $c = b_1 \cup b_2$
- $\{(3, X)\}$.

It can be easily seen that $NR(d, c)$, because $\tilde{c} = (2, c)$, and $c^\uparrow = (3, X)$ and $d \cap X \neq \emptyset$. Similarly $NR(d, (a_1 \cup a_4))$ holds as $a_1 \cup a_2 = (2, c)$. It is not $NR(d, a_1)$ as $\tilde{a}_1 = (0, a_1)$ and $a_1^\uparrow = (1, b_1)$ but $d \cap b_1 = \emptyset$. Similarly one can see that not $NR(d, a_4)$.

4 PROXIMITIES AND NEARNESS

With proofs similar to the ones of [8] one can show that NR fulfills the properties of a *right-scaled proximity relation*. This is detailed out in the following proposition.

Proposition 2. Let X be a set, pc be a partition chain over X and $NR_{pc} = NR$ be a nearness relation as defined by (1). The relation NR fulfills the properties of a right-scale scaled proximity, that is:

1. for all $a, b \subseteq X$: if $NR(a, b)$, then a and b are nonempty;
2. for all $a, b, c \subseteq X$:
 - (a) if $NR(a, b)$ or $NR(a, c)$, then $NR(a, b \cup c)$;
(if a is near one of b or c , then it is near the union of b and c .)

- (b) $NR(a, c)$ or $NR(b, c)$ if and only if $NR(a \cup b, c)$;
(the union of a and b is near c iff one of the sets of the union a or b is near c .)
3. if $a \cap b \neq \emptyset$, then $NR(a, b)$.
(a and b have one element in common, then a is near b (and so also b is near a)).

The main difference of right-scaled proximities to minimal proximity structures in the meaning explicated by [1] is the fact that for right-scaled proximities the other direction in condition 2.(a) is, in general, not fulfilled, i.e., there may be sets a, b, c , such that a is near the union of b and c , formally: $NR(a, (b \cup c))$, but neither is a near b nor is a near c . (Compare Ex. 2, where $NR(d, a_1 \cup a_4)$ but neither $NR(d, a_1)$ nor $NR(d, a_4)$.) This is due to the fact that the union of b and c may belong to a higher level than b and c . So, putting two sets (in the second) argument together may have positive emergent effects—more concretely, the positive emergent effect of switching the level (or scale) from a lower to a higher one.

Note, that this kind of positive emergent effect is also handled by super-additive measures in general measure theory [12]. Classical measures μ have to be additive, i.e., must fulfill the condition that for disjoint events a, b we must have $\mu(a \uplus b) = \mu(a) + \mu(b)$. In generalized measure theory one considers measures that weaken this condition in both directions. μ is called super-additive iff $\mu(a \uplus b) \geq \mu(a) + \mu(b)$. It is called sub-additive iff $\mu(a \uplus b) \leq \mu(a) + \mu(b)$ [12, p.67]. Super-additivity means that the union has synergetic positive effects, sub-additivity means that the union has prohibiting effects.

It is possible to further characterise the case where a region is near a union of regions but not near one of them. Let δ denote a right-scaled proximity relation. Let a, b, c such $b \cap c = \emptyset$ and we have $\delta(a, (b \cup c))$ but not $\delta(a, b)$ and not $\delta(a, c)$. We call (b, c) an *irregular split* of $b \cup c$ w.r.t. a .

Definition 5 (Regularity). A weak right-scaled proximity δ over X is called regular iff for every set $a \subseteq X$ there is at most one irregular split of a set $b \cup c$ w.r.t. a .

Now we can show that NR is a right-scaled proximity that fulfills the regularity condition.

Proposition 3. NR is a regular right-scaled proximity relation.

Proof. Assume $NR(a, b \uplus c)$ and not $NR(a, b)$ and not $NR(a, c)$. As $us(b^\uparrow) \cap a = \emptyset$ and $us(c^\uparrow) \cap a = \emptyset$, we have $us(b^\uparrow) \cup us(c^\uparrow) \subsetneq us((b \uplus c)^\uparrow)$. We must have $us((b^\uparrow)^\uparrow) \neq us(c^\uparrow)$. Now, let $b \uplus c = b' \uplus c'$ where $b' \neq b$ and $c' \neq c$. One of b', c' must have elements of both b and c . W.l.o.g let us assume it is b' . That means that $\tilde{b}' = \tilde{b} \cup c$ and hence $NR(a, b')$. \square

Please note, that this property also holds for a model of the nearness relation NR which is defined in the RCC framework [9] according to (2). In this canonical model regions are defined to be regularly closed sets in the 2-dimension real plane. The crucial point is that the underlying sets b and c of cells that touch each other make up an irregular splitting of $b \cup c$ w.r.t. some region a —where $b \cup c$ stands for the sum operation of regions according to [9]. Now, one could move border points of b to c (or vice versa) in order to get a different irregular splitting $b' \cup c'$ of $b \cup c$ w.r.t. a ; but b' and c' will not be regions anymore. Hence, the uniqueness of irregular splits is conserved, as long as b and c are constrained to be regions.

Another additional feature of the nearness relations NR_{pc} based on normal partition chains pc is that it fulfills the *connecting* property (cf. [1]), i.e., every region is near its complement or vice versa.

Proposition 4. Let be given a normal partition chain pc and a nearness relation $NR = NR_{pc}$ according to the equivalent definition in (1). Then for all $a \subseteq X$ it holds that $NR(a, X \setminus a)$ or $NR(X \setminus a, a)$.

Proof. Let $a \subseteq X$ be an arbitrary non-empty set. We have to show $NR(a, X \setminus a)$ or $NR(X \setminus a, a)$. First assume that a or $X \setminus a$ are not underlying sets of cells, e.g., w.l.o.g. assume a is not an underlying set of a cell. Then $us(\tilde{a})$ overlaps with $X \setminus a$ and we have $NR(X \setminus a, a)$. Now assume that both a and $X \setminus a$ are (underlying sets of) cells. But, because the order is normal, either $a \subsetneq us(a^\uparrow)$ or $X \setminus a \subsetneq us((X \setminus a)^\uparrow)$, hence either $NR(X \setminus a, a)$ or $NR(a, X \setminus a)$. \square

Note, that the proposition does not hold for arbitrary (i.e. non-normal) partitions chains as shown by the following example.

Example 3. Assume $a \neq \emptyset$. We can construct a non-normal partition chain pc , such that in the first three levels one has the same two subsets a_1, a_2 as cells. That is, let $X = a_1 \uplus a_2$ and let for $i \in \{0, 1, 2\}$ be given the partition a^i of level i by $((i, a_1), (i, a_2))$ the partition of level 3 shall be $(3, X)$. Let $NR = NR_{pc}$ be the nearness relation defined by this non-normal partition chain. Then we have $a_1 = X \setminus a_2$ and $a_2 = X \setminus a_1$ but not $NR(a_1, a_2)$ and not $NR(a_2, a_1)$.

In general, the nearness relations NR_{pc} for normal partition chains pc will not fulfill the so called strong axiom (3) for proximity relations δ (cf. [6]).

$$\begin{aligned} \text{If not } \delta(a, b), \text{ there is an } e \subseteq X \text{ s.t.:} \\ \text{not } \delta(a, e) \text{ and not } \delta((X \setminus e), b) \end{aligned} \quad (3)$$

This axiom says that if a is not near b , there is a set e which separates a and b . In particular, if also for all sets a', b' with $a' \cap b' \neq \emptyset$ it holds that $\delta(a', b')$, then the fact that not $\delta(X \setminus e, b)$ entails $b \subseteq e$ (because it must be the case that $(X \setminus e) \cap b = \emptyset$).

We give a simple counterexample to the strong axiom.

Example 4. Take $X = \{1, 2, 3, 4, 5, 6\}$. Consider the following normal chain as illustrated in Fig. 2:

$$0 : \overbrace{\{1, 2\}}^{a_1} \cup \overbrace{\{3, 4\}}^{a_2} \cup \overbrace{\{5, 6\}}^{a_3} \leq 1 : \overbrace{\{1, 2\}}^{b_1} \cup \overbrace{\{3, 4, 5, 6\}}^{b_2} \leq 2 : X$$

Take $a = \{1\}$, $b = a_2 = \{3, 4\}$. Then not $NR(a, b)$. But there is no

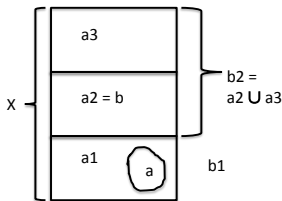


Figure 2. Illustration of configuration in Ex. 4

$e \supseteq b$ such that not $NR(a, e)$ and not $NR(X \setminus e, b)$. The reason is: If $e = b$, then $NR(X \setminus b, b)$ as $(X \setminus b) \cap us(b^\uparrow) \neq \emptyset$. Similarly, if $b \subsetneq e$, then $us(e^\uparrow) = X$ and hence $NR(a, e)$.

5 CELL PROPERTIES AND LOCAL DYNAMICS OF NEARNESS

In the subsections before we gave properties of the nearness relation NR that do refer only to NR but not to the underlying partition chain. As we will consider the effects of changing the arguments in NR and the effects of changing the partition chain on the induced nearness, we investigate in this section properties referring also to the partition chains. Concerning the first point of change, these properties are relevant to what we call the *local dynamics of nearness*. The investigation of local dynamics means—among other things—answering the following question: How does a change of the right argument of NR affect the set of sets considered near it? In particular, for which two regions (or more concretely: sets underlying cells b_1 and b_2) does the change from b_1 to b_2 conserve the nearness relations?

In order to answer (if only partly) this question, we introduce the following equivalence relations on the basis of a relation δ (which will be instantiated by NR) over a set X .

$$a^\bullet = \{b \subseteq X \mid \delta(a, b)\} \quad (4)$$

$$\bullet a = \{b \subseteq X \mid \delta(b, a)\} \quad (5)$$

$$a \sim^\bullet b \quad \text{iff} \quad a^\bullet = b^\bullet \quad (6)$$

$$a \bullet \sim b \quad \text{iff} \quad \bullet a = \bullet b \quad (7)$$

$$a \sim b \quad \text{iff} \quad a \sim^\bullet b \text{ and } a \bullet \sim b \quad (8)$$

As the identity $=$ is an equivalence relation (i.e., it is reflexive, symmetric, and transitive), the definitions immediately entail the fact that $\sim^\bullet, \bullet \sim, \sim$ are equivalence relations, too. Now, if we look at cells (i, a) and (i, b) that are contained in the same upper cell, then these are left-equivalent.

Proposition 5. Let $a, b \subseteq X$ such that $\tilde{a} = (i, a)$, $\tilde{b} = (i, b)$ and $a^\uparrow = b^\uparrow$. Then $a \bullet \sim b$.

Proof. Let $c \subseteq X$ be an arbitrary set. Then, by assumption $NR(c, a)$ iff $NR(c, b)$. \square

Concerning the main question of the local dynamic of nearness this proposition has the following consequence: Changing the perspective from a cell to another cell of the same level with the same upper level does not change the perspective on what regions (as the first argument) are considered to be near. For illustration, consider again Fig. 1 in Example 2. Think of an agent that stays at cell a_1 and has calculated the regions near a_1 . Then the agent moves to cell a_2 , which has the same upper cell b_1 . Then according to Prop. 5 he does not have update the regions near it as the regions near a_2 are exactly those near a_1 . The situation is different if the agent moves from a_1 to a_4 which has a different upper cell than a_1 .

A dual assertion with respect to this lemma is the observation that two disjoint cells are near each other in both directions iff they are cells on the same level with the same upper level cell.

Proposition 6. For all sets a, b with $\tilde{a} = (i, a)$ and $\tilde{b} = (j, b)$ and $a \neq b$ the following equivalence holds: $NR(a, b)$ and $NR(b, a)$ iff $i = j$ and $a^\uparrow = b^\uparrow$.

Proof. The direction from right to left follows from Prop. 5. For the other direction assume $NR(a, b)$ and $NR(b, a)$. Then by definition of NR , the first argument of the conjunct implies $a \cap us(b^\uparrow) \neq \emptyset$. But this means, as a is the underlying set of a cell, that $a \subseteq us(b^\uparrow)$. As $a \cap b = \emptyset$, we can exclude the case that $a = us(b^\uparrow)$; hence, it follows that $us(a^\uparrow) \subsetneq us(b^\uparrow)$ and $i \leq j$. Symmetrically, we can deduce $us(b^\uparrow) \subsetneq us(a^\uparrow)$ and $j \leq i$. In the sum we get $b^\uparrow = a^\uparrow$. \square

Moreover, if a, b are cells of the lowest level and are contained in the same upper level, then they are equivalent.

Proposition 7. Let $a, b \subseteq X$ be such that $\tilde{a} = (0, a)$, $\tilde{b} = (0, b)$ and $a^\uparrow = b^\uparrow$. Then $a \sim b$.

Proof. Because of Prop. 5 we are done with the proof if we can show that $a \sim^\bullet b$. Let $c \subseteq X$ be an arbitrary set. $\text{NR}(a, c)$ iff (by definition) $a \cap \text{us}(c^\uparrow) \neq \emptyset$ iff (as different cells are either disjoint or comparable with respect to \subseteq , and the level of \tilde{c} is greater than or equal to the level of \tilde{a}) $\text{us}(a^\uparrow) \subseteq \text{us}(c^\uparrow)$ iff $\text{us}(b^\uparrow) \subseteq \text{us}(c^\uparrow)$ iff $\text{NR}(b, c)$. \square

Again, concerning the main question of the local dynamic of nearness this proposition has the following consequence: Changing the perspective from a cell to another cell on the lowest level, where both have the same upper level, does not change the set of regions that are considered to be near—and this holds in both cases of changing the first argument or of changing the second argument.

6 MERGING AND GLOBAL DYNAMICS OF NEARNESS

In their study of regional changes of municipalities in Finland, Kauppinen and colleagues [5] found seven kinds of type changes which are as follows:

1. a region is established
2. two or more regions are merged into one
3. a region is split into two or more regions
4. a region's name is changed
5. a region is annexed to a different country
6. a region is annexed from a different country
7. a region is moved to another city or municipality

We are interested in changes that concern changes of cells for partitions in a given partition chain. Hence we adapt a subset of the types of changes to our setting by explicitly formalizing the type of change.

Clearly the most interesting changes are that of merging two regions to a new region and its counterpart, the split of regions into two regions. These types of changes are low frequent-changes (in contrast to the local dynamics case where an agent updates the nearness relations when moving around); e.g., Kauppinen and colleagues [5] recognized 144 merges and 94 splits of municipalities in Finland between 1865 and 2007. But nonetheless, the effects of merges and splits on the nearness relation are worth to be investigated.

Here, we restrict our attention to different forms of merging. We have to explain what it means that two cells (of a partition) are merged, and whether such a merge is possible such that the result is again a (normal) partition chain.

So let pc be a normal partition chain over X having levels 0 to n . We will look at merging two cells on the same level into a new cell; in order to get a first rough picture on the effects of merging, we look at the special case where the cells are members of the next-to-last level $n - 1$. In this case, both cells to be merged have always the same upper cell, namely X . For illustration of the possible merge operations have a look at the partition chain in Fig. 3, which we have arranged such that one can see the tree structure of the the partition chain, with X being its root. The cells labelled with the letter c make up the cells of the next-to-last level 2. The different forms of changes within a partition chain can be seen as different forms of updating a tree.

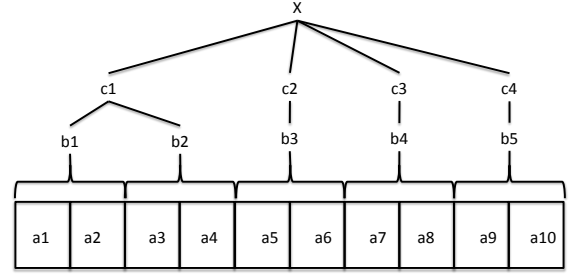


Figure 3. Illustration of example configuration for merge

Merging the cells $(2, c_2)$ and $(2, c_3)$ into a new cell means that the underlying set of the merging result has to have the union of c_2 and c_3 as the underlying set. But there are in principle two ways to conduct this merge that depend on specifying the level of the merge result.

The first option is to modify the next-to-last level, so that the whole number of levels is untouched. In case of the example illustrated in Fig. 3 this would mean that the partition of c -cells is substituted by the new partition of c -cells that consists of the cells $(2, c_1)$, $(2, c_2 \cup c_3)$ and $(2, c_4)$ (see Fig. 4). We term this type of merge *level modifying merge*—*lm merge* for short. If a normal partition chain pc_2 results from another normal partition chain pc_1 by an *lm merge*, then we write $pc_1 \rightsquigarrow^{lm} pc_2$.

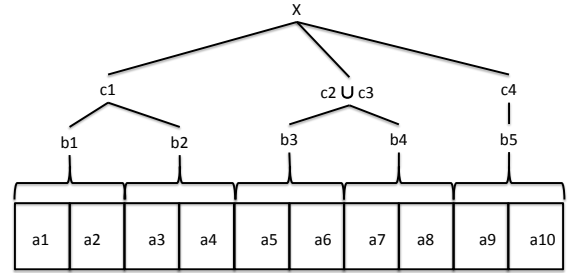


Figure 4. Illustration of merge by modifying

The other option is to make the union of the sets to be part of a new level. Hence, in addition to the original partition made up by $(2, c_1)$, $(2, c_2)$, $(2, c_3)$ and $(2, c_4)$, one adds the partition $(3, c_1)$, $(3, c_2 \cup c_3)$ and $(3, c_4)$ and raises the level of X by one to $(4, X)$ (see Fig. 5). We term this type of change *level adding merge*—*la merge* for short. If a normal partition chain pc_2 results from another normal partition chain pc_1 by a *la merge*, then we write $pc_1 \rightsquigarrow^{la} pc_2$.

In some cases, either form of merge may not be possible without violating the normality condition. For example, if the next-to-last level consists only of two cells $(n - 1, x_1)$ and $(n - 1, x_2)$, then the union of x_1 and x_2 is the whole domain X ; so the merge results in the same set partition (X) on two different levels, which violates the normality condition.

What can we say about the change of the nearness relation induced by level modifying merges on the next-to-last level? First we note

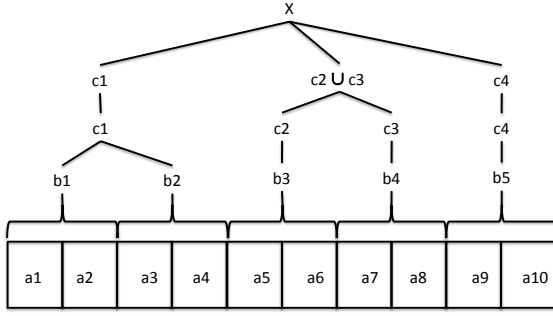


Figure 5. Illustration of merge by adding

that the level of a set in pc_1 is identical to the level in pc_2 if the former is below or equal to $n - 1$. If its level in pc_1 is n , then its level in pc_2 may be n or $n - 1$.

The change of pc_1 into pc_2 affects only the next-to-last partition, e.g., by merging cells $(n - 1, c_1)$ and $(n - 1, c_2)$; hence, the nearness relation is affected only locally. So, if the second argument b has level at most $n - 3$, then one can say that a is near b in pc_2 if and only if it is near in pc_1 .

Proposition 8. *Let pc_1, pc_2 be two normal partition chains over X such that $pc_1 \rightsquigarrow^{lm} pc_2$ w.r.t. cells $(n - 1, c_1)$ and $(n - 1, c_2)$ on the next-to-last level $n - 1$. Then the following assertions hold:*

1. *For all sets $a \subseteq X$ and all sets $b \subseteq X$ with level $l_{pc_2}(b) \leq n - 3$ one has : $NR_{pc_1}(a, b)$ iff $NR_{pc_2}(a, b)$.*
2. *For all sets $a, b \subseteq X$: If $NR_{pc_1}(a, b)$, then $NR_{pc_2}(a, b)$.*

Proof. The assertions can be proved as follows:

1. This assertion follows from the fact, that for all $b \subseteq X$ with level at most $n - 3$ (in pc_2) the upward cells in both pc_1 and pc_2 are identical, $b^{\uparrow, pc_1} = b^{\uparrow, pc_2}$. Hence, by definition of nearness it immediately follows that $NR_{pc_1}(a, b)$ iff $NR_{pc_2}(a, b)$.
2. In order to proof this assertion suppose $NR_{pc_1}(a, b)$, i.e., $a \cap us(b^{\uparrow, pc_1}) \neq \emptyset$. We distinguish different cases depending on the level $l_{pc_1}(b)$ of b in pc_1 .
Assume $l_{pc_1}(b) = n - 2$, then $b^{\uparrow, pc_1} = (n - 1, c)$ for some set c on the level $n - 1$. If $c = c_1$ or $c = c_2$, then $b^{\uparrow, pc_2} = (n - 1, c_1 \cup c_2)$. So from $a \cap us(b^{\uparrow, pc_1}) \neq \emptyset$ one deduces $a \cap us(b^{\uparrow, pc_2}) \neq \emptyset$, i.e. $NR_{pc_2}(a, b)$. If c is an underlying set of another cell on level $n - 1$, then we have $us(b^{\uparrow, pc_2}) = (n - 1, c)$ and hence also $NR_{pc_2}(a, b)$. Now assume that $l_{pc_1}(b) = n - 1$. Then $b^{\uparrow, pc_1} = (n - 1, c)$ for some set c on the partition level $n - 1$. Then we will have $b^{\uparrow, pc_2} = (n - 1, c')$ for $c \subseteq c'$. Hence, $b^{\uparrow, pc_2} = (n, X)$ and so $NR_{pc_2}(a, b)$. Last assume that $l_{pc_1}(b) = n - 1$. In this case, the level of b in pc_2 may be $n - 1$ or n . But in any case, one has $b^{\uparrow, pc_2} = (n, X)$, and therefore $NR_{pc_2}(a, b)$. □

The consequence of this proposition for a cognitive agent using NR as a nearness notion is that it has to update his NR graph only locally when the partition chain is updated by a level modifying change.

Due to the level addition, the situation for la merges is a little bit different. For example, considering our example partition chain illustrated in Fig. 3 one can have $a \subseteq X$ such that $NR_{pc_1}(a, c_2)$ but not

$NR_{pc_2}(a, c_2)$ because, the upper level cell of $(n - 1, c_2)$ in pc_1 is the biggest cell (n, X) , but in pc_2 the upper cell is $(n - 1, c_2 \cup c_3)$. So, choosing, e.g., $a = a_1$ and $b = c_2$ one has $NR_{pc_1}(a, b)$ but not $NR_{pc_2}(a, b)$. But still we can show as above that sets with level below $n - 3$ have the same nearness relations.

Proposition 9. *Let pc_1, pc_2 be two normal partition chains over X such that $pc_1 \rightsquigarrow^{la} pc_2$ w.r.t. cells $(n - 1, c_1)$ and $(n - 1, c_2)$ on the next-to-last level $n - 1$. Then for all sets $a \subseteq X$ and all sets $b \subseteq X$ with level $l_{pc_2}(b) \leq n - 3$ one has : $NR_{pc_1}(a, b)$ iff $NR_{pc_2}(a, b)$.*

7 CONCLUSION

Cognitive agents using a hierarchical nearness relation based on a partition chain have to deal with two aspects of dynamics of nearness, the local dynamics (the cognitive agent changes his position and so has to update the nearness relations) and a global dynamics (the partition chain may change, and hence the induced nearness relation has to be changed.) We have shown that under some circumstances both a local change and a global change affect the nearness relation only w.r.t. a small set of regions; hence, under these circumstances, the nearness relations between few regions have to be updated.

We gave preliminary results on the local dynamics and on the global dynamics of the partition-chain based nearness relation. The results on global dynamics have to be completed by investigations on merges for levels below the next-to-last level. In this case one will have to differentiate between merging regions with the same upper level cells vs. merging regions with different upper level cells. Additionally one has to define how to propagate the merge effect to the higher levels (as the merger on level i may affect also cells on levels above $i + 1$.) Moreover we plan to define adaptations of the other changes mentioned by [5] to the partition-chain framework and investigate their effects on the change of the nearness relation.

The presented approach considers only partition chains, i.e. a totally ordered set of nested partitions. For more realistic approaches we are going to formally investigate the more general scenario where partitions may not be nested/aligned. This is, e.g., the case when one considers micro functional regions [2] in addition to administrative units.

ACKNOWLEDGEMENTS

We would like to thank the referees for their comments, which helped improve this paper.

REFERENCES

- [1] Ivo Düntsch and Dimitar Vakarelov, ‘Region-based theory of discrete spaces: A proximity approach’, *Annals of Mathematics and Artificial Intelligence*, **49**, 5–14, (2007).
- [2] Rolf Grütter, Iris Helming, Simon Speich, and Abraham Bernstein, ‘Rewriting queries for web searches that use local expressions’, in *Proceedings of the 5th International Symposium on Rule-Based Reasoning, Programming, and Applications (RuleML 2011 – Europe)*, eds., Nick Bassiliades, Guido Governatori, and Adrian Paschke, volume 6826 of *LNCs*, pp. 345–359, (2011).
- [3] Rolf Grütter, Thomas Scharrenbach, and Bettina Waldvogel, ‘Vague spatio-thematic query processing: A qualitative approach to spatial closeness’, *Transactions in GIS*, **14**(2), 97–109, (2010).
- [4] Jixiang Jiang and Michael Worboys, ‘Event-based topology for dynamic planar areal objects’, *Int. J. Geogr. Inf. Sci.*, **23**(1), 33–60, (January 2009).

- [5] Tomi Kauppinen, Jari Väättäin, and Eero Hyvönen, 'Creating and using geospatial ontology time series in a semantic cultural heritage portal', in *Proceedings of the 5th European semantic web conference on The semantic web: research and applications*, ESWC'08, pp. 110–123, Berlin, Heidelberg, (2008). Springer-Verlag.
- [6] S.A. Naimpally and B. D. Warrack, *Proximity Spaces*, number 59 in Cambridge Tracts in Mathematics and Mathematical Physics, Cambridge University Press, 1970.
- [7] Özgür L. Özçep, Rolf Grütter, and Ralf Möller, 'Nearness rules and scaled proximity', Technical report, Institute for Softwaresystems (STS), Hamburg University of Technology, (2012). Available online at <http://www.sts.tu-harburg.de/tech-reports/papers.html>.
- [8] Özgür L. Özçep, Rolf Grütter, and Ralf Möller. Nearness rules and scaled proximity, 2012. Paper accepted for publication in the proceedings of ECAI 2012.
- [9] David A. Randell, Zhan Cui, and Anthony G. Cohn, 'A spatial logic based on regions and connection', in *Proceedings of the 3rd International Conference on Knowledge Representation and Reasoning*, pp. 165–176, (1992).
- [10] J. G. Stell, 'Boolean connection algebras: a new approach to the region-connection calculus', *Artificial Intelligence*, **122**(1-2), 111–136, (2000).
- [11] John G. Stell and Michael F. Worboys, 'Relations between adjacency trees', *Theor. Comput. Sci.*, **412**(34), 4452–4468, (2011).
- [12] Z. Wang and G.J. Klir, *Generalized measure theory*, volume 25 of *IFSR International Series on Systems Science and Engineering*, Springer Verlag, 2008.

Socially Compliant Navigation in Crowded Environments

Frank Dylla and Martin Coors and Mehul Bhatt¹

Abstract. Mobile robot technology develops fast nowadays. Currently, commercial robot systems are restricted to controlled areas like production halls or storage buildings. It is only a question of time until robots are able to operate in arbitrary populated environments. Current efforts to improve social acceptance of robots aim at the hardware, e.g. two legged-robots or human like heads. Although, software deals with collision avoidance and path or task planning, the question how to improve social acceptance of the navigation behavior of such robots is only tackled partially.

For recognizing robots as socially accepted beings reactive collision avoidance is not enough. As humans follow – in most cases unwritten – social norms and regulations in every day life, robots need to follow them as well. In this paper we concentrate on pedestrian navigation. First, we extract a set of social navigation rules and formalize them in a qualitative manner. By simulation we investigate how traveling time is influenced by being polite, i.e. following social norms, to different extents.

1 Introduction

Social acceptance of autonomous robots plays a key role whether they will find their way in everyday life. Up to now most effort is spent on reliability of tasks like navigation, collision avoidance, or manipulation of objects in order to enable robots to perform in arbitrary indoor or outdoor environments. Additionally, robots with a more human like look are developed. Although, up to now autonomous robots are only successfully applied in controlled environments like storage and production buildings it is only a question of time until they can be applied in human populated environments from a technical perspective. With this in mind, quite some work has been spent on linguistic human robot interaction. Nevertheless, only little effort has been spent on social behavior regarding navigation. From our point of view the ability to understand social behavior and performing accordingly, also in navigation, is a key issue to social acceptance of autonomous robots.

In this paper we focus on polite pedestrian behavior and its impact on travel time from a start position to destination. As rules of polite pedestrian behavior are only given in a vague linguistic manner, if at all, we need to collect them and formalize them afterwards (Section 3). For example, a rule of politeness is ‘don’t walk too close to other people’ or ‘don’t cut another one’s walk path directly ahead of that person’. Nevertheless, such linguistic rule descriptions contain qualitative terms like ‘ahead’, ‘turn left’, or ‘evade to’. The research field of Qualitative Spatial and Temporal Reasoning (QSTR) is specifically concerned with the formalization of such qualitative terms in formal calculi (Section 2). Therefore, we apply techniques

from the QSTR community to formalize rules of politeness in pedestrian navigation. Within a simulated environment we implemented several settings to investigate the impact of politeness to travel time (Section 4). As this work is quite new and at its start point, we need to discuss our results (Section 5).

We start with an introduction of QSTR, general motion planning, and existing approaches to model pedestrian behavior.

2 Related Work

The background of our work builds Qualitative Spatial Reasoning (Section 2.1). Specifically, the relative orientation calculus \mathcal{OPRA}_m builds the basis for the formalization of politeness (Section 2.2). Tools to deal with qualitative calculi are given by the toolbox SparQ (Section 2.3). We apply qualitative approaches in order to formalize polite behavior (Section 2.5) which is closely related to motion planning (Section 2.4). In our implementation we employ the pedestrian simulation JWalkerS (Section 2.6).

2.1 Qualitative Spatial and Temporal Reasoning

Qualitative Spatial and Temporal Reasoning (QSTR) is an established field of research dealing with aspects of space in an abstracted, i.e. qualitative, manner rather than numerical values. For example, qualitative abstractions of relative orientation comprise terms like ‘left of’ or ‘right of’ instead of absolute angular values. Kuipers [22] states that ‘Although the world is infinitely complex and our knowledge of the world is limited, i.e. incomplete, biological systems, especially humans, function quite well within this world without understanding it completely’. The basic idea of QSTR is to capture spatial distinctions between objects that make an important qualitative difference but ignore others. These differences are captured by *relations*, which summarize indistinguishable cases into a single symbol. For example, if it is irrelevant for the task at hand whether an object is in 90° or 91° angle to the observer, both may be represented by the relation *left of*. Relations may represent different kinds of spatial domains.

Freksa et al. [13] distinguish between topological (e.g. disconnected, part of, etc.) and positional calculi, which can be subdivided into orientation (e.g. relative: left, right, etc. and absolute: south, north, etc.) and distance calculi (e.g. close, far, etc.). Other aspects are size or shape. A set of such primitive relations, also called *base relations* \mathcal{BR} , over a domain forms a *Qualitative Spatial Calculus*. The relations should fulfill the property of jointly exhaustiveness and pairwise disjointness (JEPD), i.e. for any pair of objects (in the binary case) exactly one relation holds at a specific time point.

Reasoning over \mathcal{BR} is then performed over all possible relations, i.e. the power set $2^{|\mathcal{BR}|}$. A set of base relations, e.g. {left, front}, between two objects reflects a disjunction of the base relations contained, i.e. the objects are either in relation left *or* in relation front.

¹ University of Bremen, Cognitive Systems, Germany, email: {dylla,marcoors,bhatt}@informatik.uni-bremen.de

Reasoning can be classified in two main branches: constraint-based reasoning (CBR) and neighborhood-based reasoning (NBR).

CBR is concerned with the question of satisfiability or global consistency of a static configuration (a set of objects with its pairwise relational constraints), i.e. whether a configuration is physically realizable or not. The most important operation for this is *composition* \circ . If we know object A is north of B and B is north of C, we know A is north of C ($A \text{ N } B \circ (B \text{ N } C) \rightarrow (A \text{ N } C)$). As global consistency is complex to determine very often local consistencies are calculated in order to approximate global consistency. The most prominent method is called *algebraic closure*. For any three objects $(O_1 \text{ r}_{12} O_2) \circ (O_2 \text{ r}_{23} O_3) \cap (O_1 \text{ r}_{13} O_3) \rightarrow (O_1 \text{ r}'_{13} O_3)$ is calculated until a fix point is reached. If any $\text{r}'_{13} = \emptyset$ it is known that the configuration is not consistent. For further details we refer to [7].

NBR is concerned with the dynamics of objects regarding a specific calculus, i.e. which spatial change may occur under the assumption of continuous motion. *Conceptual Neighborhood* (CN) is a general abstraction of change and was introduced by Freksa [12]: *Two spatial relations of a qualitative spatial calculus are conceptually neighbored, if they can be continuously transformed into each other without resulting in a third relation inbetween.* For example, given the three basic relations $\{<, =, >\}$ for points on a line $<$ and $=$ as well as $=$ and $>$ are conceptual neighbors, but $<$ and $>$ are not.

As CN only deals with the general term of 'transformation', i.e. arbitrary motion of the objects involved, the concept was extended by Dylla in [9, 10] to deal with different modes of motion like 'forward/backward', rotation, or sideways motion. With the additional knowledge of the action performed the branching factor in the conceptual neighborhood graph is reduced significantly. Due to this, sophisticated motion planning is possible. It has been shown that on the basis of the action-augmented neighborhood graph, right-of-way regulations in the domain of vessel navigation can be formalized [9]. Simplified, we will use the same approach for modeling the rules of politeness in pedestrian navigation. We come back to this in Section 2.4.

2.2 $OPRA_m$: A Relative Orientation Calculus

The domain of the Oriented Point Relation Algebra ($OPRA_m$) [28, 27] is the set of oriented points (points in the plane with an additional direction parameter). The $OPRA_m$ calculus is very well suited for dealing with objects that have an intrinsic front or move in a particular direction and can be abstracted as points.

The calculus relates two oriented points with respect to their relative orientation towards each other. An oriented point \vec{O} can be described by its Cartesian coordinates $x_O, y_O \in \mathbb{R}$ and a direction $\phi_{\vec{O}} \in [0, 2\pi)$ with respect to a reference direction and thus the domain is $\mathbb{R}^2 \times [0, 2\pi)$. The exact set of base relations distinguished in $OPRA_m$ depends on the granularity parameter $m \in \mathbb{N}$. For each of the two related oriented points, m lines are used to partition the plane into $2m$ planar and $2m$ linear regions. Figure 1 shows the partitions for the cases $m = 2$ (a) and $m = 4$ (b). The orientation of the two points is depicted by the arrows starting at \vec{A} and \vec{B} , respectively. The regions are numbered from 0 to $4m - 1$, region 0 always coincides with the orientation of the point. An $OPRA$ base relation consists of a pair (i, j) where i is the number of the region of \vec{A} which contains \vec{B} , while j is the number of the region of \vec{B} that contains \vec{A} . These relations are usually written as $\vec{A} \text{ }_m\text{ } \angle^j_i \vec{B}$. Thus, the examples in Figure 1 depict the relations $\vec{A} \text{ }_2\text{ } \angle^1_7 \vec{B}$ and $\vec{A} \text{ }_4\text{ } \angle^3_{13} \vec{B}$. Additional base relations called '*same*' relations describe situations in which both oriented points coincide. In these cases, the relation is determined

by the number s of the region of \vec{A} in which the orientation arrow of \vec{B} is positioned (as illustrated in Figure 1(c)). These relations are written as $\vec{A} \text{ }_m\text{ } \angle_s \vec{B}$ ($\vec{A} \text{ }_2\text{ } \angle_1 \vec{B}$ in the example). We denote the set of base relations by \mathcal{BR}_{OPRA_m} . The total number of base relations with respect to granularity m is $|\mathcal{BR}_{OPRA_m}| = (4m)^2 + 4m$.

2.3 SparQ

SparQ is a generic toolbox for representing space and reasoning about space based on qualitative spatial relations. The toolbox makes available several qualitative spatial calculi and general reasoning techniques developed in the QSTR community in a single homogeneous framework. This comprises CBR and NBR. A detailed introduction to SparQ can be found in [34]. SparQ consists of a set of modules that logically structure the different services provided. The services can be accessed via a TCP/IP interface. Among the qualitative reasoning techniques offered we use the interface for mapping the continuous domain to qualitative representations (qualification), i.e. transferring quantitative positions into qualitative $OPRA_4$ relations.

2.4 Motion Planning

In general, motion planning can be categorized in two classes: a) local path planning and collision avoidance and b) global path planning.

In local path planning no map or other representation of the complete environment is given or the environment is not static. This means, motion planning can only be performed on the basis of local and partial information. Therefore, reactive collision avoidance methods belong to this class as well. A drawback of these methods is always that agents may get stuck in local minima, e.g. a dead end. Potential fields are based on attractive and repellent forces of other agents, obstacles and points of interest present in the environment [18]. Virtual Force Fields extends this approach in order to derive more stable action in presence of sensory noise [2]. Vector Field Histogram (VFH) methods are based on local sensory information represented in histograms [3]. The histogram reflects the obstacle density regarding the relative orientation towards the agent. Angles with lowest obstacle densities are considered the best choices for future action. Extensions are VFH+ (smoothing trajectories by a hysteresis function) [1] and VFH* (minimizing computational costs by A* search) [32]. The Dynamic Window Approach (DWA) calculates subsequent collision free states based on possible changes in translational and rotational changes [11, 4]. The Curvature Velocity Method (CVM) refines the DWA by the assumption that the agent moves on a curvature determined by translational and rotational speed. This means, action selection in CVM is performed in cartesian space and in DWA it is performed in velocity space.

In global path planning (GPP) the complete environment and the agent's location needs to be known in advance. Then, GPP is the search for the minimal path in a graph structure. The underlying representation may be grid-based [19] or roadmap-based. In the latter case free or occupied space is represented in a graph structure. Prominent approaches are: visibility graphs [14] or Voronoi graphs [5, 33]. As nodes are set on the boundaries of objects, navigation is problematic. Therefore, Höcker et al. improve navigation by setting nodes with distance to obstacles [17].

Neighborhood-based planning can be applied for motion planning as well. The nodes reflect possible sets of relations between all objects at hand. The conceptual neighborhood structure defines possi-

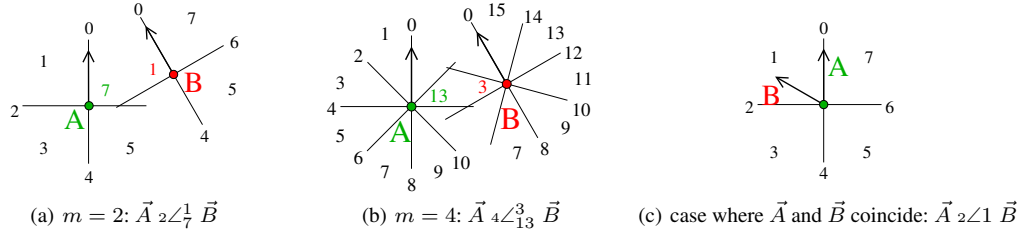


Figure 1. Two oriented points related at different granularities.

ble transitions from the current state to a neighboring state. If the action-augmented CNG is available explicit actions to reach subsequent states are given. Then planning reduces to a graph search from the current state (start state) to the desired goal state. This approach has been shown adequate to formalize collision regulations in the domain of vessel navigation (ColRegs) resulting in so called *rule transition systems*, subgraphs of the complete action-augmented neighborhood graph [9, 10]. The ColRegs are given in natural language and therefore, contain qualitative terms, which naturally map on qualitative relations. The effectiveness of the approach is shown by a simulation.

2.5 Pedestrian behavior

In general, approaches for modeling pedestrian behavior under the consideration of social interaction can be categorized in two classes: 1) with active prediction, and 2) without active prediction.

In the first case, the system, i.e. the agent, continuously adapts its behavior on the basis of assumptions about the future behavior of other pedestrians. Behavior prediction for individuals can be emulated on different levels of detail. In general three levels can be distinguished: individual behavior regarding the motion of a) complete flows of people masses (e.g. in train stations or fairs) [15], b) groups (i.e. an individual accompanies a small group with the same goal) [26], or c) individuals [25].

In the second case, no future behavior of other pedestrians is considered. Therefore, robust methods to model the behavior based on local observations must be applied. The Social-Force Model [17, 16] is an enhancement of the potential fields methods and models individual propulsive and repulsive forces for each object in the environment.

In [20] a model called PEDFLOW [35] is applied. Nevertheless, in the next version they plan to incorporate social forces model. In PEDFLOW each agent has a preferred clearance area that it tries to maintain. If another pedestrian enters this area a conditional behavior is triggered. For example, it may turn to a side or change its speed. Teknomo applies a force-based model which is comparable to the social forces model [31]. Osaragi uses a model based on stress (pedestrian stress, destination stress, etc.) which is again very similar to social forces [29]. Dijkstra [8] splits the environment into grid cells and assigns properties to them (e.g. wall, occupied). An agent moves through the environmental grid and inspects neighboring cells. Its steering algorithm makes use of the attributes of these cells and reacts with respect to the composition of different steering behaviors: path following, collision and obstacle avoidance. In [24] a sophisticated model of space around a human being is presented. They distinguish personal space, activity space, affordance space, territory, and penetrated space. They formalize these spaces in terms of the Region Connection Calculus (RCC8) [23, 6].

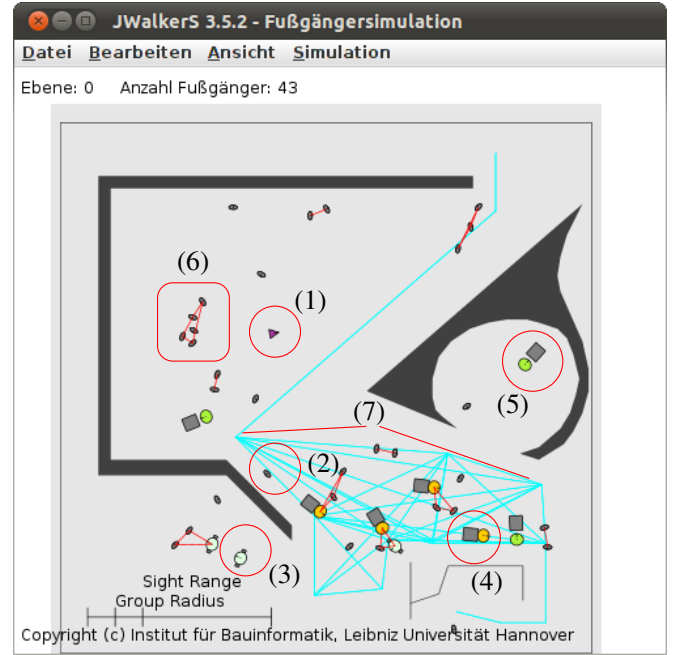


Figure 2. The simulation environment JWalkerS. Visible items include (1) agent, (2) sportive walker, (3) handicapped walker, (4) walker with vehicle in back, (5) walker with vehicle in front, (6) visualized group, (7) underlying path graph

2.6 Pedestrian Simulation: JWalkerS

For our simulation we applied JWalkerS, a pedestrian simulation software developed at the institute of civil engineering informatics at the Leibniz Universität Hannover². It is designed to support the development of crowded environments like train stations, shopping malls, etc. In specific, it helps identifying causes of disturbances in the flow of pedestrians and how they can be prevented. For example, in simulating evacuation scenarios the development of safer buildings is supported. Figure 2 shows the graphical interface of JWalkerS with different types of agents. Navigation of pedestrians is derived by the Social-Force Model (see Section 2.5) together with a roadmap-based global path planning [17]. In Figure 2 the light blue lines denote the global path planning graph.

JWalkerS applies a microscopic model for motion planning. For every agent its behavior is simulated individually, i.e. each agent may have its own behavior and goals. Typical implementations are cellular automata and software agents [21]. In contrast, macroscopic

² Website: <http://www.bauinf.uni-hannover.de/> (in german, 9.7.2012)

models are based on the similarity of pedestrian flows and fluids. The individuals in a pedestrian flow behave just about identical.

The original simulation loop for deriving pedestrian behavior consists of 4 steps: 1) orientate agents based on global path planning, 2) calculate acceleration wrt. social forces affecting the pedestrians, 3) move agents wrt. time elapsed, 4) check simulation time (in order to prevent infinite simulations). We explain necessary modifications to the simulator in Section 4.1.

3 Rules of Politeness in Pedestrian Behavior

3.1 General Classification Aspects

In order to model social pedestrian behavior we need to extract parameters that influence what is considered as *polite behavior*. We categorize the general kinematics or motion capabilities of pedestrians by the following aspects:

- translation: only straight forward motion or also backward motion
- rotation: rotation on the spot (cf. differential drive in robotics) or restricted rotation in combination with translational motion (e.g. cars with Ackermann kinematics).
- angular motion: any motion not aligned to the intrinsic front (e.g. steps to the right or left)

All these aspects have to be considered in connection with their maximum speed. In general, the maximum speed of a young adult can be considered higher than the speed of elderly people. Restrictions to the previous aspects in combination with additional objects which are pushed, pulled or carried have to be considered as well. For example, people carrying a large box or pushing a bike do not move as fast as people without carrying anything. Additionally, their perception may be restricted. Furthermore, individuals may have to be considered as a group which makes different behavior considered as polite. For example, walking through a family is not considered to be polite.

3.2 Polite Behavior of Pedestrians

To our knowledge no list of socially accepted polite behavior exists. The only sources on good social behavior we found are Emily Post³ and specifically for Germany by Knigge⁴.

Hence, we extracted a set of rules of politeness for pedestrians based on these sources and our personal experience. We are aware that this list cannot be complete as, e.g. rules incorporating stable obstacles are not integrated. Additionally, we don't consider rules that are based on non spatial aspects so far, e.g. gender aspects like 'ladies first'. In the pictorial representations of the rules we distinguish the agent to be polite (Figure 3(a)) and its motions (Figure 3(c)) and other pedestrians which may behave arbitrarily (Figure 3(b)) and their corresponding motions (Figure 3(d)).

We classify the regulations into the following five classes with the according rules:

(I) approaching head-on or from behind

- rule 1) pedestrians approaching in head-on both have to evade to the right (Figure 4(a))
- rule 2) pedestrians moving in the same direction have to be overtaken on the left side (Figure 4(b))

³ <http://www.emilypost.com/> (9.7.2012)

⁴ <http://www.umrechnung.org/artikel/knigge-verhaltensregeln-uebersicht/knigge-regeln-gutes-benehmen.htm> (in german, 9.7.2012)

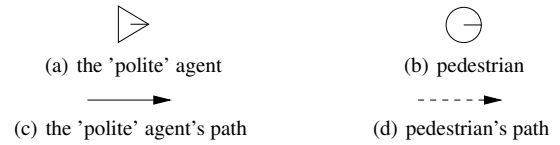


Figure 3. Icons for types of agents and their motion

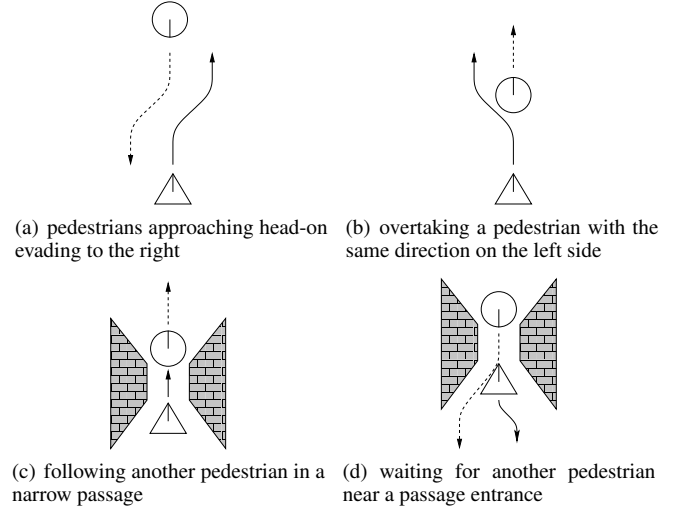


Figure 4. Four example rules for pedestrians.

These rules may have to be adapted regarding cultural background. In Great Britain pedestrians would evade to the left and overtaking should take place on the right side.

(II) crossing situations

- rule 3) other (not necessarily moving) pedestrians on the left or right may be passed in their front with some distance,
- rule 4) may be passed in their back (with smaller distance), or
- rule 5) the agent lets the other pedestrians pass (by stopping). Here the other agent has to move.

(III) bottlenecks or narrow passages

- rule 6) follow other pedestrians in a narrow passage (no overtaking possible) (Figure 4(c))
- rule 7) 'crossing' other pedestrian at narrow passage
- rule 8) waiting for other pedestrian near passage entrance (Figure 4(d))
- rule 9) going back for other head-on pedestrian in passage

(IV) interaction with groups

- rule 10) evading or passing a group on the outside
- rule 11) crossing large groups (if crossing is not possible otherwise)

We are aware that it is questionable whether rule 11 can be considered as 'polite', but we leave it as a point of discussion.

(V) individual constraints

In this class context dependent rules are summarized, e.g.

- rule 12) in general moving on the right of a walkway (or left in GB), or
- rule 13) no running in a library

A subset of these regulations builds the basis for our experiments. We give details on the implementation in the next section.

3.3 Formalization

The rules of politeness are formalized in a Lisp like syntax on the basis of the $OPRA_4$ calculus. We give an example formalization of rule 2 in Figure 5. We note that we neglect an explicit representation of the actual distance between agents. One could argue that for different distances or distance classes different behavior might be adequate. This would result in a classification of space around a pedestrian like in [24]. Within this paper we simplify this by assuming a fixed distance of interaction. In general, different behaviors can be defined taking a second or even more spatial aspects, i.e. calculi, into account.

For the remainder of the paper we will distinguish between the *agent* and other *pedestrians*. The agent is the person which we actively control on the basis of our formalization in the simulation. Other pedestrians cannot be influenced as they are controlled otherwise.

With *types* it is given for which pairs of agents the rule is valid. If a *start-condition* between agents of a valid type arises the rule has to be executed until an *end-condition* is achieved. In between the agent tries to follow the *preferred-transitions* by executing the first action given. Experiments has shown adequate to assume that the other pedestrian is static and only the polite agent is moving. If the action does not result in the expected relation, the current relation is most probably a neighboring one. If the relation is not a preferred transitions, it is most probable in the admissible transitions. Nevertheless, a follow up action is defined. If a relation arises that is not contained in the currently executed rule, the rule execution is aborted. Due to prior testing of the formalizations this happened only very rarely in the final experiments. Nevertheless, in these cases pedestrian and agent are not in direct interaction anymore as they are moving away from each other anyways.

```
:types ( (Agent Walker)(Agent Handicap)
  (Agent FrontVehicle)(Agent BackVehicle) )
:states ("PoliteNavigation")
:default-granularity 4
:start-conditions (15_7 15_9)
:end-conditions (7_1 7_15 9_1 9_15)
:preferred-transitions
( (15_7 -> fl stable -> 14_7)
  (14_7 -> fl stable -> 13_7)
  (13_7 -> fwd stable -> 13_6)
  (13_6 -> fwd stable -> 13_5)
  (13_5 -> fwd stable -> 12_4)
  (12_4 -> fwd stable -> 11_3)
  (11_3 -> fr stable -> 10_2)
  (10_2 -> fr stable -> 11_2)
  (11_2 -> fl stable -> 11_1)
  (11_1 -> fl stable -> 10_1)
  (10_1 -> fwd stable -> 9_1) )
:admissible-transitions
( (15_9 -> fl stable -> 14_8)
  (14_8 -> fwd stable -> 13_7)
  (11_5 -> fr stable -> 12_5)
  (12_5 -> fr stable -> 13_5)
  (15_5 -> fl stable -> 14_5)
  (14_5 -> fl stable -> 13_5)
  (9_3 -> fr stable -> 11_3)
  (9_2 -> fwd stable -> 9_1)
  (1_9 -> fl stable -> 15_9)
  (11_7 -> fr stable -> 13_7)
  (13_3 -> fl stable -> 11_3) )
```

Figure 5. The formalization of rule 2 (overtaking from behind)

4 Implementation and Experiments

4.1 Modifications to JWalkerS

In order to integrate behavior of an external polite agent we had to modify JWalkerS. First, an external interface had to be integrated which sends world model data and receives control commands for the next simulation step. Nevertheless, the agent is represented in the same way as the pedestrians. Furthermore, we had to modify the simulation loop in the following way. In step 1 and 2 orientation and acceleration of the external agent has to be calculated as well, and executed in step 3. Now the world model data has to be sent to the external agent via the interface (step 3-A). This comprises the current simulation step, obstacle positions (as line segments), and positions of individual pedestrians and groups. Directly afterwards JWalkerS expects to receive motion parameters (step 3-B). We chose a non-blocking connection here, because in a real world scenario, other pedestrians won't wait for an artificial pedestrian to compute its behavior. Step 4 remains unmodified. Additionally, we have to check whether the external agent reached its destination (step 5). As an option the simulation loop can be ended automatically in order to ease experiments. For the latter reason, via the interface scenarios can be loaded and started as well.

Groups We implemented a simple group recognition algorithm based on distance. If two pedestrians are closer than a given threshold, they are considered to form a group. We neglect any other aspects which may be important to identify pedestrians belonging to a group. The agent itself never belongs to any group. Furthermore, for each group the convex hull is calculated. We applied the algorithm derived by Graham implemented by Sedgewick und Wayne⁵ [30] Finally, from each convex hull the two agents are extracted which are closest to our agent and sent to it. In our experiments we distinguish individuals, small groups (2 to 6 people, e.g. a family), and large groups (more than 6 people).

Agent Types JWalkerS allows for modelling different types of pedestrians. Besides using the standard pedestrian (normal size, no motion constraints) we added the following types: pedestrians a) with walking disabilities (slower translation and rotation speed; backward or sideways motion is not possible), b) pushing a vehicle, e.g. a push chair (while turning the front remains static; no sideways motion), and c) pulling a vehicle (while turning the back remains static; only very slow backward motion; no sideways motion). The types of pedestrians are displayed differently in JWalkerS (see Figure 2).

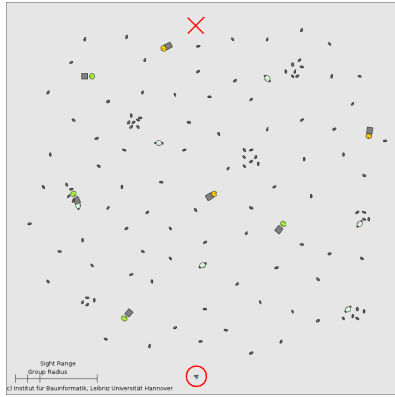
4.2 Scenarios

Overall, we generated six scenarios: 1) *groups* (several groups spread over an obstacle free space), 2) *circles* (in an obstacle free space the destination is surrounded by pedestrians with small distances, i.e. the agent must be impolite in order to reach the destination), 3) *lines* (several groups of people in line cross the agents path in an obstacle free space), 4) *corridor* (a crowded corridor), 5) *station* (a crowded station), and 6) *townsquare* (a crowded town square, i.e. populated by individuals and groups with arbitrary deviation). Here, we only elaborate on *station* (Figure 6(a)) and *townsquare* (Figure 6(b)).

⁵ <http://algs4.cs.princeton.edu/99hull/> (9.7.2012)



(a) Scenario station. The red circle marks the agent, the red cross the goal location.



(b) Scenario townsquare. The red circle and red cross are used analogously.

Figure 6. Sample scenarios as seen in the simulation JWalkerS.

4.3 Settings

In the experiments we apply only a subset of the regulations introduced in Section 3.2. The numbers of the rules we apply are: 1, 2, 3, 4, and 10. Rule 3 and 4 are only partially applicable as the agent and the pedestrians are not able to vary their speed actively. The speed is computed by the social forces affecting it, including the agent. Therefore, in our settings pedestrians are moving all the time (based on social forces) or are not moving at all (see motion index below). Similar, rule 5 is not implemented as the agent itself never stops. As no obstacle recognition is available in JWalkerS rules 6-9 cannot be considered. Additionally, no rules from class (V) are formulated as we only consider spatial aspects for now.

Furthermore, we investigate several *population densities* for each scenario. Density 0 denotes that no other pedestrians than the agent itself is present. From density 1 to density 3 the number of pedestrians in the environment increases. We have not connected these density classes with any independent measure like 'average number of people per square meter'. Therefore, comparison of results between scenarios regarding pedestrian densities is not sound. The *motion index* defines whether pedestrians are moving or standing still. Finally, we introduced the *visibility index*. By this parameter it can be set whether pedestrians perceive the agent or not in order to estimate which role evasion behavior (calculated by the social force model) plays with respect to travel time.

In order to evaluate the influence of politeness wrt. travel time

we introduce five classes of politeness. First, an agent may be completely impolite (\mathcal{H}_0), i.e. he never follows a rule of politeness, or absolutely polite (\mathcal{H}_{100}), i.e. he always follows the rules. Additionally, we define intermediate classes \mathcal{H}_{25} (on average every fourth rule is followed), \mathcal{H}_{50} (on average every second rule), and \mathcal{H}_{75} (on average three out of four rules). A decision to follow a rule is taken randomly when the agent is in a start condition of a rule. Once the decision is made it is not revised until they don't interact anymore. If they meet again, a new decision has to be taken. As it is not known in advance how many decision possibilities will arise during a run and rule following is decided by chance we define intervals around the ideal values. For \mathcal{H}_i ($i \in \{25, 50, 75\}$), the percentage of rules followed must lie in the interval $[i - 10, i + 10]$.

For each scenario we performed 100 runs with varying parameter settings. All tests were run on a IBM-compatible 64-bit Intel processor (3.47GHz) with more than 10GB RAM and the operating system Linux Ubuntu 10.04.4 LTS. JWalkerS was started without the graphical interface.

5 Results and Discussion

5.1 Results

In Figure 7 we present the results of the townsquare scenario. As expected, in general the travel time increases if 1) more rules have to be followed and 2) the more densely crowded the area is regardless whether the agent is visible to the other pedestrians or not. It is very prominent in the diagrams that the variance of absolute travel times is very high for each class of experiments. Looking at several runs revealed that agent behavior varies significantly in different runs. It may be a reason for these significant differences that the simulator calculates the social forces on a time step basis. Due to slight timing changes different forces are calculated such that behavior differs completely compared to previous runs.

If no other pedestrians are present (density $0 = d_0$) the minimal time is achieved (274 simulated seconds in average) for all \mathcal{H}_i as no rules have to be followed. Although a little higher, travel time for impolite agents (\mathcal{H}_0) is just about equal for all other d_i if the agent is visible for the other agents ($< 300s$). The increase is a little higher if the agent is invisible ($< 340s$). In these cases the agent is pushed away more from the ideal path or has to reduce its speed more due to social forces. Due to the impoliteness the agent is able to move closer to the ideal path. For completely polite agents travel time increases up to 400s.

For d_1 time increases only very little in case of \mathcal{H}_{25} ($\approx 20s$). For \mathcal{H}_{50} to \mathcal{H}_{100} the travel times are very similar ($\approx 320s$). As there are only very few decisions the agent is only 14% slower being polite compared to being impolite. For d_2 and d_3 travel time is 30% slower for the impolite agent. Interestingly, for these two densities travel time is nearly equal over all \mathcal{H}_i , although, significantly more pedestrians were set in d_3 .

In general, times are longer in the invisible case, especially in case of higher pedestrian density. For d_3 the invisible agent is 50s slower than the visible agent, for d_1 and d_2 it is only $\approx 20s$. In case of \mathcal{H}_{25} the largest difference by 100s is given. If agents are more polite the difference reduces to only 50s.

In Figure 8 we present the results from the station scenario, which are different compared to the results of the townsquare scenario. Here, in some cases travel time decreases although more rules have to be followed. This is most significantly for d_3 with the agent being invisible. Furthermore, it is interesting that travel time is shorter than

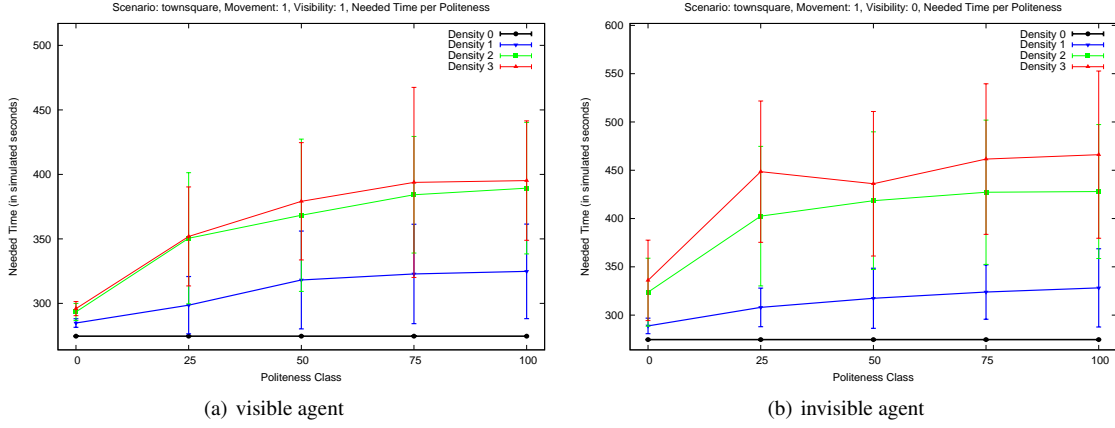


Figure 7. Graphs with results from the townsquare scenario with different visibility configurations

for d_1 and d_2 . We are still looking for a general explanation. We assume that this is connected to the choice of start point and destination of the agent wrt. to the deviation of static obstacles.

Generalized, in both scenarios impolite navigation leads to short travel times and polite behavior to longer ones. But the differences between \mathcal{H}_{25} to \mathcal{H}_{100} are comparatively small regarding the difference between \mathcal{H}_0 and \mathcal{H}_{25} . This leads us to the interpretation: 'be absolutely polite or completely not' as travel time for being 'a little or somewhat polite' is nearly as long as for being completely impolite.

5.2 Discussion

Although the results are promising, there are some drawbacks. So far, the density measure is chosen arbitrarily for each scenario. Therefore, the comparison of the results of different scenarios is not sound. Nevertheless, we don't think that an average measure like 'pedestrians per square meter' is meaningful as well. So far, we haven't found a reasonable approach to compare results between scenarios.

Furthermore, the agent takes the decision whether to follow a rule or not by chance. While humans probably act more strategically, e.g. by cutting through a complete group in an impolite manner in order to avoid future decision points, an agent decides completely independent of the previous decisions. Therefore, an agent may walk into a group by not following a rule, but decides to be polite in all upcoming decisions. This may lead to a very long travel time. A memory function may help at this point to allow for more strategic behavior, e.g. escaping from a group.

We have to note that the politeness rules can be formalized by means of social forces as well. Nevertheless, from our perspective, this model would result in complicated partial and context dependent force functions. This makes testing and maintenance very difficult. Additionally, planning on the basis of such a model is problematic.

6 Conclusion

In this paper we have proposed a set of rules of politeness in pedestrian navigation. We formalized them by means of a qualitative calculus ($OPRA_4$) from the QSTR community. The general message we take from the experiments is: "be absolutely polite or completely not" as travel time for being 'a little polite' is nearly as long as for being completely impolite.

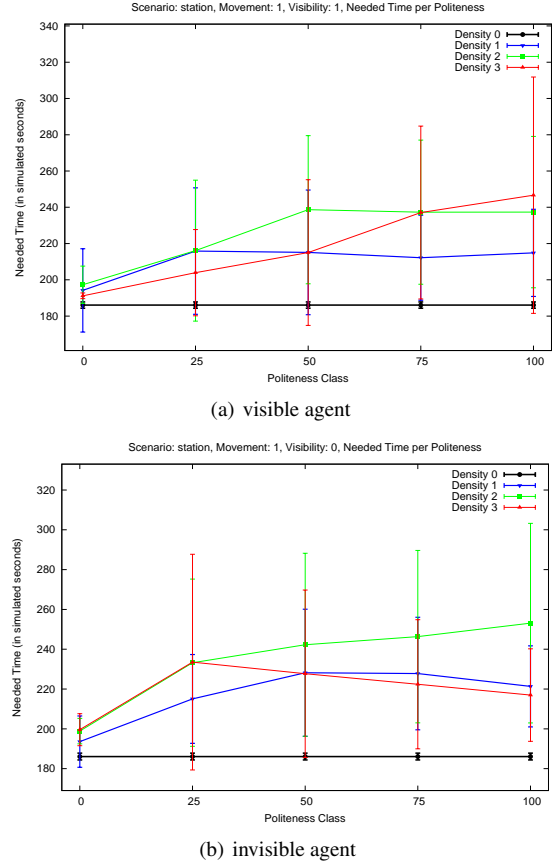


Figure 8. Graphs with results from the station scenario with different visibility configurations

In future, we need to run more experiments with subsets of rules. This may reveal whether some rules of politeness are in general more time consuming than others. Nevertheless, we need to integrate obstacle recognition in order to formulate all rules we listed so far. Furthermore, we currently work on the transfer of our approach to a real robot.

7 Acknowledgements

We like to thank Volker Berkhahn and Wassim Abu Abed for providing JWalkerS and giving very valuable comments during the development. Additionally, we like to thank the anonymous reviewers for their valuable comments. Funding by the German Research Foundation (DFG) SFB/TR8 Spatial Cognition is gratefully acknowledged.

REFERENCES

- [1] J. Borenstein, ed. *VFH+: Reliable Obstacle Avoidance for Fast Mobile Robots*, volume 2, 1998.
- [2] J. Borenstein and Y. Koren, 'Real-time obstacle avoidance for fast mobile robots', *IEEE Transactions on Systems, Man and Cybernetics*, **19**(5), 1179–1187, (1989).
- [3] J. Borenstein and Y. Koren, 'The vector field histogram - fast obstacle avoidance for mobile robots', *IEEE Journal of Robotics and Automation*, **7**, 278–288, (June 1991).
- [4] Oliver Brock and Oussama Khatib, 'High-speed navigation using the global dynamic window approach', *Proc. IEEE Int. Conf. Robotics and Automation*, 341–346, (1999).
- [5] Howie Choset and Joel Burdick, 'Sensor based motion planning: The hierarchical generalized voronoi graph', in *Proceedings of the Workshop on Algorithmic Foundations of Robotics*, Toulouse, Frankreich, (1996).
- [6] A G Cohn, B Bennett, J M Gooday, and N Gotts, 'RCC: A calculus for region based qualitative spatial reasoning', *GeoInformatica*, **1**, 275–316, (1997).
- [7] Anthony G Cohn and Jochen Renz, 'Qualitative spatial representation and reasoning', in *Handbook of Knowledge Representation*, 551–596, Elsevier Science, (2008).
- [8] J. Dijkstra, A.J. Jessurun, B. de Vries, and H.J.P. Timmermans, 'Agent architecture for simulating pedestrians in the built environment', in *Fourth International Workshop on Agents in Traffic and Transportation [ATT2006]*, eds., A.L.C. Bazzan, B. Chaib-draa, F. Kluegl, and S. Ossowski, pp. 8–16, New York, NY, (2006). ACM.
- [9] Frank Dylla, *An Agent Control Perspective on Qualitative Spatial Reasoning*, Akademische Verlagsgesellschaft Aka GmbH, Heidelberg, 2008.
- [10] Frank Dylla, 'Qualitative spatial reasoning for navigating agents', in *Behaviour Monitoring and Interpretation - BMI*, eds., B. Gottfried and H. Aghajan, IOS Press; Amsterdam; <http://www.iospress.nl>, (2009).
- [11] Dieter Fox, Wolfram Burgard, and Sebastian Thrun, 'The dynamic window approach to collision avoidance', *IEEE Robotics & Automation Magazine*, **4**, (1997).
- [12] Christian Freksa, 'Conceptual neighborhood and its role in temporal and spatial reasoning', *Decision Support Systems and Qualitative Reasoning*, 181–187, (1991).
- [13] Christian Freksa and Ralf Röhrig, 'Dimensions of qualitative spatial reasoning', in *Proceedings of the III IMACS International Workshop on Qualitative Reasoning and Decision Technologies - QUARDET'93*, eds., N. Piera Carreté and M. G. Singh, pp. 483–492. CIMNE Barcelona, (1993).
- [14] Hector H. Gonzales-Baños, David Hsu, and Jean-Claude Latombe, 'Chapter 10 - motion planning: Recent developments', in *Autonomous Mobile Robots: Sensing, Control, Decision Making and Applications*, eds., Shuzhi Sam Ge and Frank L. Lewis, Taylor & Francis Group, Boca Raton, Florida, (2006).
- [15] Peter Henry, Christian Vollmer, Brian Ferris, and Dieter Fox, 'Learning to navigate through crowded environments', in *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, (2010).
- [16] Mario Höcker, *Modellierung und Simulation von Fußgängerverkehr*, Shaker Verlag, Aachen, Germany, 2010.
- [17] Mario Höcker, Volker Berkhahn, Angelika Kneidl, Andre Borrmann, and Wolfram Klein, 'Graph-based approaches for simulating pedestrian dynamics in building models', in *8th European Conference on Product & Process Modelling (ECPPM)*, University College Cork, Cork, Ireland, (2010). <http://zuse.ucc.ie/ECPPM/>.
- [18] O. Khatib, 'Real-time obstacle avoidance for manipulators and mobile robots', *1985 IEEE International Conference on Robotics and Automation*, 500–505, (March 1985).
- [19] Jungtae Kim and Daijin Kim, 'Visibility graph path planning using a quad-tree', in *Proceeding of The 9th POSTECH-KYUTECH Joint Workshop On Neuroinformation*, pp. 37–38, Pohang, Korea, (2009).
- [20] Franziska Klügl and Guido Rindsfuser, 'Large-scale agent-based pedestrian simulation', in *MATES*, eds., Paolo Petta, Jörg P. Müller, Matthias Klusch, and Michael P. Georgeff, volume 4687 of *Lecture Notes in Computer Science*, pp. 145–156. Springer, (2007).
- [21] Milen Koychev, *Software-Agenten-Paradigma am Beispiel einer Simulationsumgebung für Indoornavigation im Flughafenkontext*, Master's thesis, Hochschule für Angewandte Wissenschaften Hamburg, Hamburg, 2008.
- [22] Benjamin Kuipers, *Qualitative Reasoning: Modeling and Simulation with Incomplete Knowledge*, The MIT Press, 1994.
- [23] Sanjiang Li and Mingsheng Ying, 'Region connection calculus: Its models and composition table', *Artif. Intell.*, 121–146, (2003).
- [24] Felix Lindner and Carola Eschenbach, 'Towards a formalization of social spaces for socially aware robots', in *COSIT*, eds., Max J. Egenhofer, Nicholas A. Giudice, Reinhard Moratz, and Michael F. Worboys, volume 6899 of *Lecture Notes in Computer Science*, pp. 283–303. Springer, (2011).
- [25] Matthias Luber, Gian Diego Tipaldi, and Kai O. Arras, 'Place-dependent people tracking', *International Journal of Robotics Research*, **30**(3), (March 2011).
- [26] Jörg Müller, Cyrill Stachniss, Kai O. Arras, and Wolfram Burgard, 'Socially inspired motion planning for mobile robots in populated environments', in *Proceedings of the International Conference on Cognitive Systems (CogSys)*, pp. 85–90, (2010).
- [27] Reinhard Moratz, 'Representing relative direction as a binary relation of oriented points', in *ECAI-06*, eds., Gerhard Brewka, Silvia Coradeschi, Anna Perini, and Paolo Traverso, pp. 407–411, Riva del Garda, Italy, (2006). IOS Press.
- [28] Reinhard Moratz, Frank Dylla, and Lutz Frommberger, 'A relative orientation algebra with adjustable granularity', in *Proceedings of the Workshop on Agents in Real-Time and Dynamic Environments (IJCAI 05)*, pp. 61–70, Edinburgh, Scotland, (July 2005).
- [29] Toshihiro Osaragi, 'Modeling of pedestrian behavior and its applications to spatial evaluation', in *AAMAS*, pp. 836–843. IEEE Computer Society, (2004).
- [30] Robert Sedgewick and Kevin Wayne, *Algorithms - Fourth Edition*, Addison-Wesley Professional, 2011.
- [31] Kardi Teknomo, *Microscopic Pedestrian Flow Characteristics: Development of an Image Processing Data Collection and Simulation Model*, Ph.D. dissertation, Tohoku University Japan, 2002.
- [32] Iwan Ulrich and Johann Borenstein, 'Vfh*: Local obstacle avoidance with look-ahead verification', *2000 IEEE International Conference on Robotics and Automation*, 2505–2511, (April 2000).
- [33] Jan Oliver Wallgrün, 'Hierarchical voronoi-based route graph representations for planning, spatial reasoning, and communication', 2004.
- [34] Jan Oliver Wallgrün, Lutz Frommberger, Diedrich Wolter, Frank Dylla, and Christian Freksa, 'A toolbox for qualitative spatial representation and reasoning', in *Spatial Cognition V: Reasoning, Action, Interaction: International Conference Spatial Cognition*, pp. 79–90, Bremen, Germany, (2006).
- [35] Alexandra Willis, Robert Kukla, and Jon Kerridge, 'Developing the behavioural rules for an agent-based model of pedestrian movement', in *Proceedings of the European Transport Conference 445*, pp. 69–80, (2000).

Asynchronous Decentralized Algorithm for Space-Time Cooperative Pathfinding

Michal Čáp¹ and Peter Novák² and Jiří Vokřínek¹ and Michal Pěchouček¹

Abstract. Cooperative pathfinding is a multi-agent path planning problem where a group of vehicles searches for a corresponding set of non-conflicting space-time trajectories. Many of the practical methods for centralized solving of cooperative pathfinding problems are based on the prioritized planning strategy. However, in some domains (e.g., multi-robot teams of unmanned aerial vehicles, autonomous underwater vehicles, or unmanned ground vehicles) a decentralized approach may be more desirable than a centralized one due to communication limitations imposed by the domain and/or privacy concerns.

In this paper we present an asynchronous decentralized variant of prioritized planning ADPP and its interruptible version IADPP. The algorithm exploits the inherent parallelism of distributed systems and allows for a speed up of the computation process. Unlike the synchronized planning approaches, the algorithm allows an agent to react to updates about other agents' paths *immediately* and invoke its local spatio-temporal path planner to find the best trajectory, as response to the other agents' choices. We provide a proof of correctness of the algorithms and experimentally evaluate them on synthetic domains.

1 Introduction

When mobile agents operate in a shared space, one of the essential tasks for them is to prevent collisions among themselves, possibly even to maintain a safe distance from each other. Prominent examples of domains requiring robust collision avoidance techniques are different kinds of autonomous multi-robotic systems, next-generation air traffic management systems, road traffic management systems etc.

A range of methods is being currently employed to realize a safe operation of agents within a shared space. Some of the methods assume a cooperative setting where all the involved agents work together to solve their mutual conflicts, others assume a non-cooperative setting where the agents cannot coordinate their actions, and yet others consider pursuit-evasion adversarial scenarios where a solution is a trajectory that is collision free against the worst-case behaviour of other agents. In this work, we focus on the cooperative pathfinding.

Cooperative path planners are used to plan the routes for a number of agents, taking in consideration the objectives of each agent while avoiding conflicts between the agents' paths. If the agents execute the resulting multi-agent plan precisely, it is guaranteed that the agents will not collide. Centralized solvers in literature are based either on global search or decoupled planning. Global search meth-

ods find optimal solutions, but they do not scale well for higher (over ten) numbers of conflicting agents. One of the most efficient optimal solvers for cooperative pathfinding on grids has been introduced by Standley in 2010 [8].

Decoupled approaches are incomplete, but can be fast enough for real-time applications e.g., in the video-game industry. One of the standard technique employed in gaming is the *Local Repair A** (LCA*) algorithm [7]. In LCA* each agent plans a path independently and tries to follow it to the goal position. If a collision occurs during the path plan execution, the agent replans the remainder of the route from the collision position taking into account the positions in its vicinity occupied by the other agents involved in the collision. Due to its greedy and reactive nature, the method does not perform well in cluttered environments with bottlenecks and can generate cycles, or otherwise aesthetically unpleasant, or inefficient behaviours of agents [6]. To mitigate these problems, Silver [7] introduced *Cooperative A** (CA) a cooperative pathfinding algorithm based on the idea of prioritized planning [3].

In prioritized planing, each agent is assigned a priority and the planning process proceeds sequentially agent after agent in the order of the agents' priorities. The first agent plans its path using a single-agent planner disregarding the positions and objectives of other agents. Each subsequent agent models the paths of the higher-priority agents as moving obstacles and plans its path such that the collisions with the higher-priority agents' paths are avoided. Such an approach has been shown to be effective in practice [4]. The quality of the generated solution is sensitive to the assigned priority ordering, however there is a simple heuristic for choosing an efficient ordering for the prioritized planning [9].

Recently, Velagapudi presented a decentralized prioritized planning technique for large teams of mobile robots [10]. The method is shown to generate the same results as the centralized planner. However, the formulation of the decentralized algorithm is based on the assumption that the robots have a "distributed synchronization mechanism allowing them to wait for all team mates to reach a certain point in algorithm execution" [10] and thus it does not exploit the asynchrony common in distributed systems. Rather the computation proceeds in iterations and the agents wait for each other at the end of each algorithm iteration. As a consequence, the algorithm does proceeds in synchronous rounds, where the length of a round is dictated by the agent performing the longest computation due to either a high workload, or low computational resources available.

After stating the cooperative pathfinding problem and exposing the underlying ideas of the state-of-the-art prioritized planning approaches in Sections 2 and 3, Section 4 presents the main contribution of the paper, the *asynchronous decentralized prioritized planning algorithm* (ADPP). ADPP, is an extension of the synchronized

¹ FEL, Czech Technical University in Prague, Czech Republic, e-mail: {michal.cap, jiri.vokrinek, michal.pechoucek}@agents.fel.cvut.cz

² EEMCS, Delft University of Technology, The Netherlands, e-mail: P.Novak@tudelft.nl

decentralized prioritized planning algorithm (SDPP), which removes the assumptions of synchronous execution of the decentralized algorithm. Besides the generic form of the ADPP algorithm, we also present a locally asynchronous modification of the ADPP algorithm (*interruptible ADPP*, IADPP) enabling interruptible path planner execution so that the individual agents can react to updates received from their peers more swiftly. To prove termination and correctness properties of ADPP and IADPP, we provide a new proof of termination and correctness also for the SDPP algorithm. Our proof is an alternative to the original argument presented in [10]. We implemented and extensively evaluated the discussed algorithms on a number of synthetic scenarios. Section 5 provides both an illustrative theoretical comparison of the SDPP and ADPP approaches, as well as details the experimental evaluation of the introduced algorithms. The experimental validations show that the asynchronous versions of the prioritized planning algorithm offer better runtime performance, as well as improved use of the available computational resources.

2 Cooperative Pathfinding Problem

Consider n agents a_1, \dots, a_n operating in an Euclidean space \mathcal{W} . Each agent a_i is characterized by its starting and goal positions $start_i, dest_i$ respectively. The task is to find a set of space-time trajectories $P = \{p_1, \dots, p_n\}$, such that $p_i : \mathbb{R} \rightarrow \mathcal{W}$ is a mapping from time points to positions in \mathcal{W} , $p_i(0) = start_i$, $p_i(t_i) = dest_i$ and the trajectories are mutually collision free, i.e., $\forall i, j : i \neq j \Rightarrow \neg C(p_i, p_j)$, where $C(p_i, p_j)$ denotes a space-time mutual collision relation between p_i and p_j . Informally, two trajectories collide (are in a conflict) when the trajectories touch, or intersect. That is $p_i[t'] = p_j[t']$ for some timepoint t' . However, more complex collision relations can be considered, such as those considering a minimal separation range between trajectories, etc. $t_i^{dest} = \min\{t_i \mid p_i[t_i] = g_i\}$ denotes the shortest timepoint in which the agent a_i reaches its destination $dest_i$. As a solution quality metric we use the cumulative time spent by agents navigating their trajectories defined as $dur(P) = \sum_{i=1}^n t_i^{dest}$. The cost of solution P is defined as $cost(P) = \frac{dur(P) - dur(P^*)}{dur(P^*)}$, where P^* is the set of best trajectories for each agent if the collisions are ignored.

3 Prioritized Planning

In general, the complexity of complete approaches to multi-agent path planning grows exponentially with the number of agents. Therefore, the complete approaches often do not scale-up well and hence are often not applicable for nontrivial domains with many agents. To plan paths for a high number of agents in a complex environment, one has to resort to one of the incomplete, but fast approaches. A simple method often used in practice is *prioritized* planning [3, 9, 1]. In prioritized planning the agents are assigned a unique priority. In its simplest form, the algorithm proceeds sequentially and agents plan individually from the highest priority agent to the lowest one. The agents consider the trajectories of higher priority agents as constraints (moving obstacles), which they need to avoid. It is straightforward to see that when the algorithm finishes, each agent is assigned a trajectory not colliding with either higher priority agents, since the agent avoided a collision with those, nor with lower priority agents who avoided a conflict with the given trajectory themselves.

The complexity of the generic algorithm grows linearly with the number of agents, which makes the approach applicable for problems involving many agents. Clearly, the algorithm is greedy and incomplete in the sense that agents are satisfied with the first trajectory not

colliding with higher priority agents and if a single agent is unable to find a collision-free path for itself, the overall path finding algorithm fails. The benefit, however, is fast runtime in relatively uncluttered environments, which is often the case in multi-robotic applications. Prioritized planner is also sensitive to the initial prioritization of the agents. Both phenomena are illustrated in Figure 1 that shows a simple scenario with two agents desiring to move from s_1 to d_1 (s_2 to d_2 resp.) in a corridor that is only slightly wider than a single agent. The scenario assumes that both agents have identical maximum speeds.

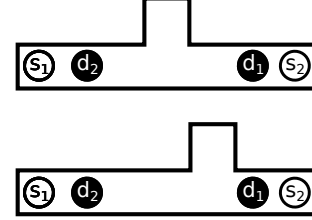


Figure 1: Top: example of a problem to which a prioritized planner will not find a solution. The first agent plans its optimal path first, but such a trajectory is in conflict with all feasible trajectories of the second agent. Bottom: example of a problem to which a prioritized planner will find a solution only if agent 1 has a higher priority than agent 2.

3.1 Computing best response

During prioritized planning, an individual agent searches the shortest path to its destination considering other higher-priority agents as moving obstacles during the planning process. Ideally, the agent should compute the best possible trajectory, a best response to the trajectories of the higher-priority agents. To find such a best response, the agent needs to solve a motion planning problem with dynamic obstacles, which is a significantly more complex task than the motion planning with static obstacles since a new independent time dimension has to be considered during planning. Henceforth, we will denote the single-agent best-response planer process as a function $BEST-PATH_i(start, dest, avoids)$, which returns the selected best trajectory for the agent i , starting in the position $start$, eventually reaching the position $dest$ and at the same time not colliding with any of the trajectories in the set $avoids$. Note, we do not precisely specify what the best trajectory means, the notion can be application-specific for the individual agent. For simplicity, however, in the following we assume the notion of the best path to correlate with time-optimality of trajectories, i.e., the how fast a given agent can navigate along the trajectory given its specific motion dynamic constraints.

3.2 Centralized Algorithm

A collision-free operation of a multi-agent team can be ensured by forcing all agents to communicate their objectives to a centralized planner, which centrally computes a solution and informs the agents about the trajectory they have to follow in order to maintain the conflict-free operation. As a baseline for evaluation of performance of the latter introduced algorithms, we use the *cooperative A* algorithm* [7]. Cooperative A* is a centralized algorithm for cooperative path finding based on prioritized planning employing the well-known A* trajectory planning algorithm on grids [5]. Algorithm 1 lists the pseudocode of the cooperative A* algorithm. We discussed the correctness of this generic algorithm above.

Algorithm 1 Centralized Prioritized Planning (Cooperative A*)

Ensure: After the algorithm finishes, $Path_i$ contains the final computed path for the agent with priority i . If the agent couldn't find a path not colliding with higher priority agents, $Path_i$ stores \emptyset .

```
1: procedure CA( $\langle start_1, dest_1 \rangle, \dots, \langle start_n, dest_n \rangle$ )
2:    $Avoids \leftarrow \emptyset$ 
3:   for  $i \leftarrow 1 \dots n$  do
4:      $Path_i \leftarrow \text{BEST-PATH}_i(start_i, dest_i, Avoids)$ 
5:      $Avoids \leftarrow Avoids \cup \{Path_i\}$ 
6:   end for
7: end procedure

8: function BEST-PATH $_i(start, dest, avoids)$ 
9:   return the best path from  $start$  to  $dest$  not conflicting with
10:    any of the paths in  $avoids$ . Otherwise return  $\emptyset$ .
11: end function
```

3.3 Decentralized Algorithms

A decentralized algorithm for solving cooperative pathfinding problems by means of prioritized planning has been presented in [10]. The algorithm is synchronous in that it contains synchronization points in the program execution through which all agents proceed simultaneously. Due to the synchronous nature of the algorithm, we will refer to this algorithm as *synchronized decentralized prioritized planning* (SDPP). Algorithm 2 lists the pseudocode of SDPP. We slightly adapted the algorithm listing for exposition purposes and comparison with the later introduced algorithms. Note that in the decentralized setting we assume communication to be reliable and the communication channel preserves the order of messages they were sent in. Furthermore, the algorithm assumes that before the start of the algorithm, each agent is assigned a unique priority, an ordinal $I \in 1 \dots N$, where N is the number of agents taking part on the algorithm run (the lowest I means the highest priority). The algorithm is also locally asynchronous and we assume safe (thread-safe) access to global variables (denoted by capitalized identifiers). To simplify the exposure, the thread-barrier locking mechanism is omitted from the pseudocode.

The algorithm proceeds in iterations. In each iteration the agents compute the best path if necessary and subsequently communicate it to the lower priority agents. An agent must recompute its trajectory in the case its current path collides with some trajectories of higher priority agents computed and communicated in the previous iterations. Upon receiving an INFORM message, the agent simply replaces the information about the trajectory of the sender agent in its $Agentview_i$ set. Note, the algorithm is asynchronous, therefore the trajectory planning routine BEST-PATH $_i$ operates on a copy the $Agentview_i$ set.

The algorithm finishes when all the agents cease to communicate and either hold a trajectory, or they were not able to find a collision-free trajectory. We assume that the global termination condition is detected by some concurrently running global state detection algorithm, such as the Chandy and Lamport's snapshot algorithm [2].

The presented SDPP algorithm is correct in that when it finds a solution for all the participating agents, the paths are mutually collision free. However, the algorithm is incomplete in the sense that there are situations when the algorithm fails to find a solution for all the participating agents, even though such a solution exists. In order to facilitate and simplify exposition of the later introduced algorithms, we developed a new alternative proof of the SDPP algorithm, which

Algorithm 2 Synchronized Decentralized Prioritized Planning

\triangleright pseudocode for the agent i \triangleleft

Ensure: After the algorithm finishes, $Path_i$ contains the final computed path. If no solution was found, $Path_i$ stores \emptyset .

```
1: procedure SDPP( $start, dest, nagents, priority$ )
2:    $Start_i \leftarrow start; Dest_i \leftarrow dest$ 
3:    $N \leftarrow nagents; I \leftarrow priority$ 
4:    $Agentview_i \leftarrow \emptyset; Path_i \leftarrow \emptyset$ 
5:   repeat
6:     CHECK-CONSISTENCY-AND-PLAN
7:     wait for all other agents to finish the planning iteration
8:   until global termination detected
9: end procedure

10: procedure CHECK-CONSISTENCY-AND-PLAN
11:   if  $Path_i$  collides with  $Agentview_i$  then
12:      $\triangleright$  Work on a copy of the  $Agentview_i$   $\triangleleft$ 
13:      $Path_i \leftarrow \text{BEST-PATH}_i(Start_i, Dest_i, Agentview_i)$ 
14:     for all  $j \leftarrow I + 1 \dots N$  do
15:       SEND-INFORM-TO- $j(I, Path_i)$ 
16:     end for
17:   end if
18: end procedure

19: message handler RECEIVE-INFORM( $j, path$ )
20:    $Agentview_i \leftarrow (Agentview_i \setminus \langle j, \_ \rangle) \cup \{ \langle j, path \rangle \}$ 
21: end message handler
```

deviates from the original one devised by the authors of SDPP [10].

To see the correctness of the SDPP algorithm we need to show that firstly, the algorithm terminates, and secondly that the resulting paths are mutually collision free.

Proof (SDPP termination):. First of all, we need to show that the algorithm finishes. That is, each agent i eventually stops sending INFORM messages. We proceed by induction on the individual agent priority i .

initial step since there is no agent with priority higher than agent a_1 , the highest priority agent a_1 informs the lower priority agents only once in the first iteration of the algorithm and from then on it remains silent since its path will always be non-colliding with an empty set of paths - there are no higher priority agents to inform this agent about an update of the situation.

induction step Let's assume the following induction hypothesis: "*after the agents with priorities $1 \dots k - 1$ stopped communicating, eventually also the agent with priority k stops sending INFORM messages*". Let's assume this is not the case and there is a situation such that the agent k would end up sending INFORM messages forever. For such to occur, the agent however must have its mailbox continually being filled with INFORM messages so that it's RECEIVE-INFORM handler routine gets invoked infinitely many times. In a consequence the agent would possibly need to recompute its best path and subsequently inform the lower priority agents infinitely often. That however implies existence of a sender for each such a message and hence by necessity there must be at least one agent with priority higher than k which keeps sending INFORM messages forever, which contradicts the induction hypothesis. □

As a consequence of the consecutive silencing of agents from high to lower priorities, it's also relatively straightforward to see that the SDPP algorithm makes at most N iterations before it terminates.

Note, that not necessarily it is the agent with the lowest priority which stops communicating the last. In the case a lower priority agent computes a route which is not in a conflict with a current set of temporary routes of the higher priority agents, nor with any routes they will compute later on, its reactions to receiving INFORM messages will be silent and won't result in further cascade of communication.

Proof (SDPP correctness):. To see that after the algorithm termination the variables $Path_i$ store a set of non-conflicting paths is rather straightforward. Since each agent eventually sends its last INFORM message and cedes to communicate, each agent with priority lower than its own eventually collects all the last INFORM messages from all the higher priority agents, together with their ultimate paths (being either a valid path, or \emptyset). At that moment, all the couples $\langle j, Path_j \rangle$ for all $j > i$ are stored in the set $Agentview_i$ of the agent with priority i . Subsequently the agent eventually invokes the CHECK-CONSISTENCY-AND-PAN routine for the last time and thus either $Path_i$ will end up unchanged, recomputed and again non-conflicting with either of $\langle j, Path_j \rangle$ for all $j < i$, or being invalid (\emptyset). Finally, the agent informs all the agents with priorities lower than i and cedes to communicate. At the moment when the last agent stops communicating, all the $Path_i$ variables are either set correctly, or the algorithm failed to find a solution for some of the participating agents. \square

As we already noted above, the SDPP algorithm is incomplete. To see that, consider a situation in which the agent with the highest priority makes a choice which later on constraints some of the lower priority agents so that they are unable to find a solution. In the case there would be a locally worse choice for the highest priority agent, which however would enable the lower priority agents to find valid solutions, the SDPP algorithm does not facilitate re-consideration of the first choice, nor some backtracking mechanism.

During the algorithm computation, it can however happen that an agent i sets its $Path_i$ to \emptyset and later reconsiders this decision. This happens when among paths of the higher priority agents there are conflicting couples, but those agents did not manage to resolve the collisions yet and at the same time the lower priority agent i is temporarily not able to route around the space occupied by the temporary paths of the higher priority agents.

Note that in the distributed prioritized planning, one can use a simple marking-based termination-detection mechanism. Following the proof of termination, agent i can mark its path *final* if the path of agent priority $i - 1$ in $Agentview_i$ is marked final. The initial path of a_1 is final. When an agent sends his final path to a lower-priority agent, the higher-priority agent can safely terminate its computation. When the final path is generated by the lowest-priority agent, the computation terminated globally.

4 Asynchronous Prioritized Planning

The SDPP algorithm does not fully exploit the parallelism of the distributed system, a drawback stemming from its synchronous nature. The running time of a single iteration of the SDPP algorithm is largely influenced by the speed of the computationally slowest agent of the group. In every iteration, the agents which finished their trajectory planning routine faster, or did not have to re-plan at all sit idle while waiting for the agents with higher workload in that iteration (or simply slower computation), even though they could theoret-

Algorithm 3 Asynchronous Decentralized Prioritized Planning

▷ pseudocode for the agent i ◁

```

1: procedure ADPP( $start, dest, nagents, priority$ )
2:    $Start_i \leftarrow start; Dest_i \leftarrow dest$ 
3:    $N \leftarrow nagents; I \leftarrow priority$ 
4:    $Agentview_i \leftarrow \emptyset; Path_i \leftarrow \emptyset$ 
5:   repeat
6:      $CheckFlag_i \leftarrow false$ 
7:     CHECK-CONSISTENCY-AND-PAN
8:     wait for  $CheckFlag_i$ , or global termination
9:   until global termination detected
10: end procedure

11: message handler RECEIVE-INFORM( $j, path$ )
12:    $Agentview_i \leftarrow (Agentview_i \setminus \langle j, \_ \rangle) \cup \{ \langle j, path \rangle \}$ 
13:    $CheckFlag_i \leftarrow true$ 
14: end message handler

```

ically resolve some of the conflicts they have among themselves in the meantime and thus speed up the overall algorithm run.

To improve the performance of the decentralized cooperative path finding, we propose an *asynchronous decentralized prioritized planning algorithm* (ADPP), an asynchronous variant of SDPP. Algorithm 3 lists the pseudocode of ADPP.

The main deviation from the SDPP listed in Algorithm 2 is the formulation of the waiting condition in the main loop of the algorithm. While each agent of the group waits for all the other to finish in the SDPP algorithm, in the ADPP algorithm, they break their idle upon receiving the next INFORM message or a need to process updated $Agentview_i$, in the case the agent received a number of INFORM messages during the time it was occupied with planning its own trajectory. The arrival of a new INFORM message and thus the need to re-check the consistency of the currently computed path with respect to the new information is indicated by the state of the $CheckFlag_i$ variable.

The proof of correctness of the ADPP algorithm follows exactly the correctness proof of the SDPP algorithm above. Note, in the SDPP proof, the condition that the algorithm proceeds in a synchronized manner was never exploited. The ADPP algorithm terminates for exactly the same reasons as SDPP. Namely, the agent with the highest priority stops communicating right after it computes its path for the first time and in consequence the agents with lower priority consecutively cede to communicate later on as well until the algorithm terminates. The argument for ADPP incompleteness follows the incompleteness argument for SDPP as well.

Interruptible ADPP

The ADPP algorithm exploits the potential speed up with respect to the inter-agent communication. However, while the agent is computing the best path in the current situation, messages keep arriving. In a consequence, it can happen that an individual agent's computation returns from the path planning routine $BEST-PATH_i$ only to find out that large part of the work was invalidated by some later received messages. This reveals a potential further speed-up of the ADPP algorithm by interrupting the path planning upon reception of every INFORM message and re-considering the computation in the light of the newly received message. Algorithm 4 lists a pseudocode of a modified ADPP algorithm which pro-actively interrupts the trajectory planning computation upon receiving every new INFORM message. Alternatively, it is conceivable to exploit algorithms

Algorithm 4 Interruptible Asynchronous Decentralized Prioritized Planning - pseudocode for the agent i

```

1: procedure IADPP( $start, dest, nagents, priority$ )
2:    $Start_i \leftarrow start; Dest_i \leftarrow dest$ 
3:    $N \leftarrow nagents; I \leftarrow priority$ 
4:    $Agentview_i \leftarrow \emptyset; Path_i \leftarrow \emptyset$ 
5:   CHECK-CONSISTENCY-AND-PLAN
6:   wait for global termination
7: end procedure

8: message handler RECEIVE-INFORM( $j, path$ )
9:    $Agentview_i \leftarrow (Agentview_i \setminus \{j, \_ \}) \cup \{j, path\}$ 
10:  asynchronously launch/restart {
    CHECK-CONSISTENCY-AND-PLAN}
11: end message handler

```

for dynamic trajectory planning, which allow topological changes during the planning process.

Note, the main **repeat-until** loop was replaced by simple wait for the algorithm termination. The repeated consistency check (calls of the CHECK-CONSISTENCY-AND-PLAN routine) is secured by its asynchronous invocation from the RECEIVE-INFORM routine. That is, the routine is executed in a newly created computation run (thread) and the call does not wait for its termination, it runs in parallel to the RECEIVE-INFORM routine from then on. In the case there is already a concurrent invocation of the CHECK-CONSISTENCY-AND-PLAN routine running, it is killed and run anew (restarted) with the updated $Agentview_i$ set.

The termination and correctness of the IADPP algorithm stems from the termination and correctness of the ADPP algorithm. The same proof applies, since the IADPP modification was strictly local, not affecting the communication patterns between the participating agents.

5 Evaluation

The motivation for introducing the decentralized algorithm and its asynchronous variants is oriented mainly to the runtime improvements of the algorithm. Clearly, such a potential improvement is greatly influenced by the topology of the problem and the selection of agent priorities. In this section, we first discuss the noticeable features of the presented algorithms and our expectation on their performance. Then, we will present experimental evaluations using super-conflict and randomly generated scenarios.

5.1 Theoretical analysis

As indicated above, the decentralized approaches should benefit from the concurrent execution on a higher number of processors (i.e., equal, or higher than the number of agents). The wall-clock runtime of the algorithms is expected to be lower for decentralized algorithms, but there might exist some problem configurations that yield directly opposite results. In this section we sketch a theoretical analysis of the impact of the parallelism and asynchronicity and show examples to demonstrate the presented ideas.

Let us first discuss the differences between the centralized and decentralized approaches. For simplicity, let the processing time of the best-path search routine be one time unit for each path searched (one path for one agent). Figure 2 illustrates an example of the algorithm execution sequence for three agents, where priorities of the agents

are given from left to right and match the agent indices. The centralized algorithm simply computes the agents' paths sequentially in the order of agents' priorities. The total wall-clock runtime is 3 time units here.

To analyze the algorithm runs in decentralized scenarios, consider a scenario where the agents have non-conflicting trajectories and a superconflict scenario, in which the best trajectories of all the agents collide. In a distributed setting, we assume three parallel processors, i.e. one for each agent. In the case of non-conflicting trajectories the agents should be able to fully utilize the inherent parallelism of the distributed system, so that the wall-clock runtime of the algorithm is only one time unit. However, in the case of superconflict scenario the situation is different. Each lower-priority agent has to recompute his path when a higher-priority agent produces a new solution. Clearly, the parallel execution has no speed-up effect here since the wall-clock runtime stays 3 time units. This example provides an intuition for the bounds of the decentralized algorithm execution time. One would expect that the wall-clock runtime of a decentralized algorithm will be equal or lower than the execution time of the centralized algorithm depending on the scale of coupling between the agents. That is, informally, on the size of a cluster in which agents' trajectories influence each other.

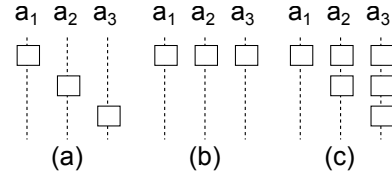


Figure 2: Example of the path search execution sequence for (a) centralized algorithm, (b) decentralized algorithm for non-colliding trajectories problem and (c) decentralized algorithm for mutually-colliding trajectories problem. The boxes represent invocations of best-response planners.

However, the situation changes if we assume non-uniform run-times of the agents' best-response planners. In such a situation, SDPP may suffer from significant synchronization overheads. Figure 3 illustrates the difference between the synchronous and the asynchronous variant of the decentralized approach. In this example ADPP exploits existence of independent conflict clusters and is able to lower the total wall-clock runtime from 5 to 4 time units. Main distinguishing feature of the ADPP algorithm over SDPP is that in ADPP an agent starts resolving conflicts immediately after the agent detects them, while in SDPP the conflicts are resolved in the next iteration of the algorithm. Since the duration of one SDPP iteration is determined by the slowest computing agent, the computational power of faster computing agents may stay unutilized. This example illustrates how can be the wall-clock runtime reduced by the asynchronous algorithm.

The interruptible variant of ADPP strengthens the asynchronous aspect of the ADPP. Figure 4 shows a another example of the decentralized algorithms execution sequence. The total running time is 5 time units for SDPP and ADPP while IADPP is able to shorten the execution to 4 time units.

5.2 Experimental evaluation

We compare the centralized CA, SDPP, ADPP and IADPP on a few variants of superconflict scenario and on a series of randomly generated problem instances. The experiments were performed on Intel

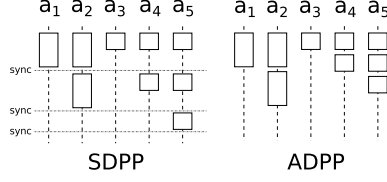


Figure 3: Sequence diagram showing the execution of SDPP resolution process and ADPP resolution process for a scenario with two independent conflict clusters, where agents in $\{a_1, a_2\}$ and $\{a_3, a_4, a_5\}$ need different amount of time to find their best response.

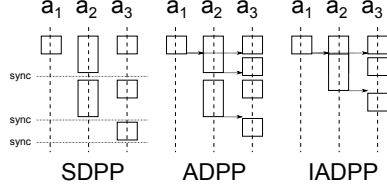


Figure 4: Sequence diagram illustrating how can be wall-clock runtime further reduced by interrupting the best-response planning.

Core 2 Duo @ 2.1 Ghz. The problem instances used have the following common structure. A given number of agents n operate in a shared 20 m x 20 m 2-d square space. The agents generate a space-time trajectory between their start and the destination position using a 4- or 8- connected grid graph. The agents can move on the edges of the graph with the constant speed of 1 m/s or they can wait for 0.5 s on any of the vertices in the graph. The wait “move” can be used repeatedly. The agents are required to maintain the separation distance 0.8 m from all other agents at all times, even after they reached their destination.

The best-response planner used by all the agents is a spatio-temporal A* planner operating over the grid graph, where the heuristic is the time needed to travel the euclidean distance from the current node to the destination node at the maximum speed. All the compared algorithms use the identical best-response planner.

To measure the runtime characteristics of the execution of decentralized algorithms, we emulate the concurrent execution of the algorithms using a discrete-event simulation. The simulation measures the execution time of each message handling and uses the information to simulate the concurrent execution of the decentralized algorithm as if it is executed on n independent computers. In the simulation we assume zero communication delay. The concurrent process execution simulator was implemented using Alite multi-agent simulation toolkit. The complete source code of the experimental environment (including the concurrent process simulator) and the video recordings of the experiments are available at <http://labe.felk.cvut.cz/~mcap/adpp/>.

Superconflict scenarios

We performed a number of experiments on a few variants of a challenging superconflict scenario. In the superconflict scenario, the agents’ start positions are put evenly spaced on a circle and their goal positions are exactly at the opposite side of the circle. Therefore, the agents’ nominal trajectories all cross in the center of the circle. The superconflict scenario is considered a challenging benchmark since each agent participating in one superconflict circle is in conflict with all other agents of that circle. Due to this coupling, the problem cannot be easily split into independent subproblems and solved in parallel. In our implementation, the agents plan their trajectory using a

60x60 8-connected grid graph. We evaluated the algorithms on the following variants of superconflict scenario:

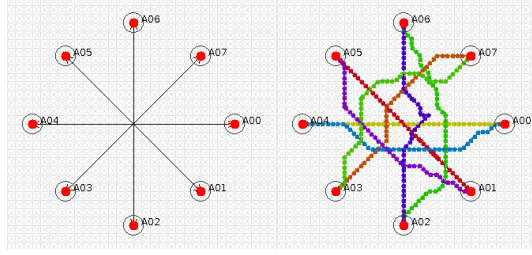
Single superconflict scenario with a 4 meters-wide superconflict of 8 agents placed in the middle of the square space. Agents’ starting configuration and the final trajectories obtained from IADPP are depicted in Figure 5a. Note that A00 is the highest priority agent in all our experiments.

Four homogeneous superconflicts scenario with four independent superconflicts of 8 agents (4 meters wide). This scenario allows the cooperative pathfinding problem to be split into four independent parts and thus the decentralized algorithms have an opportunity to exploit the computational power of more processor (see Figure 5b).

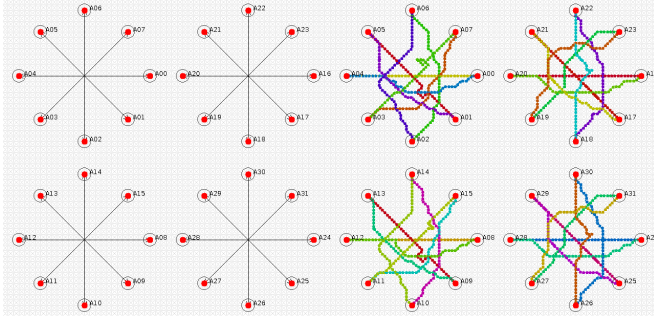
Four heterogeneous superconflicts scenario that combines two superconflicts of four agents (4 meters wide) and two superconflicts of eight agents (only 2 meters wide). The former two have bigger radius than the latter two and thus we expect that the best-response planner invocations in the first group of superconflicts will take on average longer to finish than the planners of the agents from the second group. Such a difference in planning times leads to an inefficient execution of SDPP, since the slowest progressing cluster of conflicts limits the speed at which the other conflict clusters are resolved. The asynchronous algorithm can resolve each of the superconflicts at a different pace and thus we expect ADPP and IADPP to converge faster than SDPP (see Figure 5c).

Spiral superconflict scenario is a superconflict of eight agents, where the distance between an agent’s start position and the center of the superconflict increases with each agent. In our scenario the radius varies between 2 m and 6 m. In result, the higher priority agents often finish planning before the lower priority agents and since all the agents are in mutual conflict, the planning process of the lower priority agents is often invalidated. In both SDPP and ADPP, the planning cannot be interrupted, and the agent will adapt to the new situation only after the currently running planning process finishes. Since the interruptible version of ADPP is designed to mitigate this problem, we expect that it will outperform the other decentralized methods in the scenario (see Figure 5d).

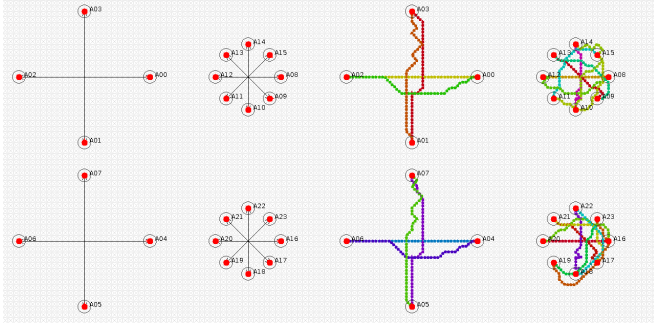
Table 1 shows the wall-clock runtimes of the four evaluated algorithms in the four presented scenarios. For the single superconflict scenario, ADPP and IADPP runtimes are close to CA, but SDPP shows significant synchronization overheads. The second scenario in fact contains four independent instances of the single superconflict as used in the first scenario. The total complexity of this problem is expected to be four times higher than that of the first scenario. The runtime of CA is more than quadrupled, while the runtime of the decentralized algorithms stays almost unchanged, which indicates perfect parallelization of the solution search process. In the heterogeneous variant of the last scenario, the situations looks different. As we can see from CA, the total complexity of the problem is slightly lower than that of the first scenario. Due to the differences in average planning times in the individual superconflicts, the wall-clock runtime in SDPP is dominated by the slowest progressing superconflict. We can see that both ADPP and IADPP can handle the heterogeneity well. The spiral superconflict is a challenging scenario for the non-interruptible asynchronous method. Thus, the ADPP wall-clock runtime is closer to that of SDPP.



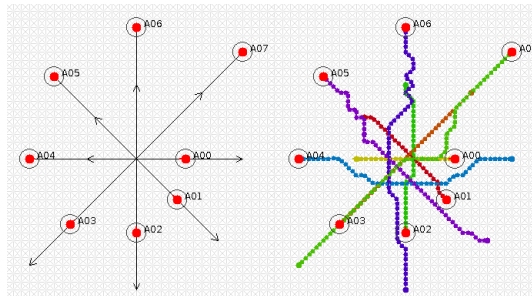
(a) Single superconflict scenario example.



(b) Four homogenous superconflicts scenario example.



(c) Four heterogeneous superconflicts scenario example.



(d) Spiral superconflict scenario example.

Figure 5: Superconflict scenarios example – problem configurations (left) and solutions from IADPP algorithm (right).

	CA	SDPP	ADPP	IADPP
single superconflict	10.30 s	26.24 s	11.91 s	9.50 s
four homogeneous superconflicts	45.81 s	26.97 s	13.86 s	11.62 s
four heterogeneous superconflicts	9.084 s	16.01 s	4.89 s	2.59 s
spiral superconflict	6.15 s	21.02 s	17.64 s	3.77 s

Table 1: Wall-clock runtimes for four versions of superconflict scenario (averaged over 10 runs)

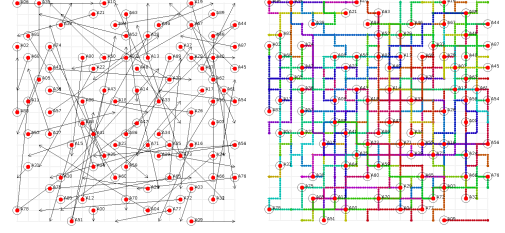


Figure 6: One instance of random scenario with 90 agents. The start and goal position of each agent are depicted on the left, the final solution found is on the right.

Random scenario

We measured the wall-clock runtime, communication complexity and solution quality of the four algorithms CA, SDPP, ADPP and IADPP on a series of problem instances that varied in the number of agents from 30 to 100. The start and goal vertices for each agent in the scenario were selected randomly (see Figure 6). The distance between the start and goal position was taken uniformly from the interval (5, 10) and we further asserted that no two agents share the start node and no two agents share the destination node. The agents plan their trajectory on a 20x20 4-connected grid graph. For each number of agents we ran 10 different random scenarios and averaged the results. When any of the algorithms failed to find a solution to a problem instance, the problem instance was excluded from the experiment.

The wall-clock runtime represents the real-world time a particular algorithm would need to converge to a solution. The wall-clock time for CA is equal to its CPU-time and can be measured directly. The average wall-clock runtime of the three decentralized algorithms on random scenarios with n agents was obtained by running an n concurrent processes simulation of the algorithm execution. The results for the wall-clock runtime experiment are shown in Figure 7a. We can see that all decentralized algorithms can offer a speed-up over the centralized solver. Further, we find that ADPP and IADPP provide comparable wall-clock runtime performance, which is significantly better than the runtime performance of SDPP, especially in dense problem instances with many conflicting agents.

Further, we measured the communication complexity by counting the messages each of the algorithms broadcasts during the execution. The communication complexity of the CA algorithm is computed analytically. We assume that the algorithm is used to coordinate paths in a distributed system in the following way. All the agents are required to communicate their objectives to the central solver. When the central solver finishes the planning, it informs each agent about its new path. Thus, we use $2n$ as the communication complexity of the centralized solver. In Figure 7b we can see that the decentralized algorithms start exceeding the communication complexity of the centralized solution for scenarios with more than 60 agents. Further, we find that IADPP algorithm has lower communication complexity than ADPP. This can be explained by looking at how the two algorithms react to an inform message that invalidates the current running planning effort. In ADPP, the planning is finished, the new plan broadcast and only after that a new planning is started. In IADPP, the planner is restarted quietly, yielding no extra communication.

Figure 7c shows the quality of the generated solutions. The reason why decentralized algorithms return on average slightly worse solutions than the CA algorithm lies in the replanning condition used by the decentralized algorithms. The condition states that an agent should replan his trajectory only if the trajectory is inconsistent with

his agentview. In result, the agent may receive an updated trajectory from a higher-priority agent that allows for improvement in his current trajectory, but since the trajectory may be still consistent, the agent will not exploit such an improvement opportunity.

Finally, Figure 7d shows the failure rates of the individual algorithms as a function of the number of agents in a scenario.

6 Conclusion

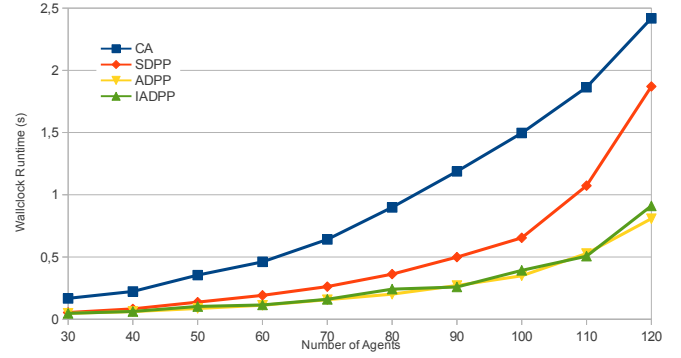
In this paper we introduced an asynchronous decentralized prioritized planning algorithm for space-time cooperative pathfinding problem. Two variants of the algorithm, ADPP and IADPP, were presented. We proved the correctness and termination of both introduced algorithms. The algorithms were compared to both central and decentralized state-of-the-art techniques for prioritized planning. Experimental validation and evaluation showed the benefits and limitations of the discussed algorithms. The experiments show the advantages of asynchronous and interruptible execution of the presented algorithms on a set of superconflict scenarios.

The large scale evaluation on a set of random problem instances documents a significant reduction of average wall-clock runtime of both ADPP and IADPP in comparison to the centralized (approx. 65% time reduction) and the decentralized synchronous algorithm (approx. 45% time reduction). The communication complexity is the worst for ADPP, while IADPP is still better than SDPP, but worse than CA for higher numbers of agents. The average cost of generated solutions is similar for all decentralized algorithms and only approx. 10% worse than CA. The failure ratio of all prioritized methods is comparable. The experimental validation fully supports the expectations on the improvements of the ADPP and IADPP over both CA and SDPP.

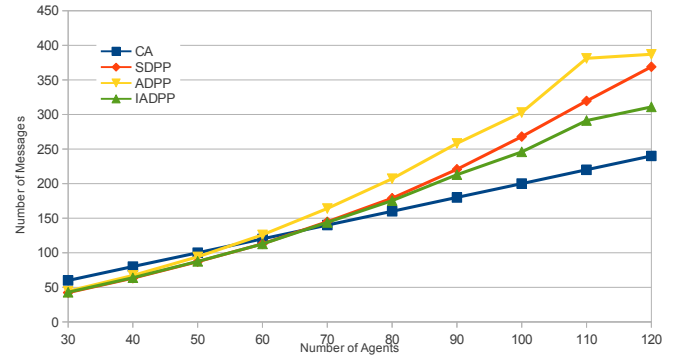
Acknowledgements This work was supported by the Ministry of Education, Youth and Sports of Czech Republic within the grant no. LD12044.

References

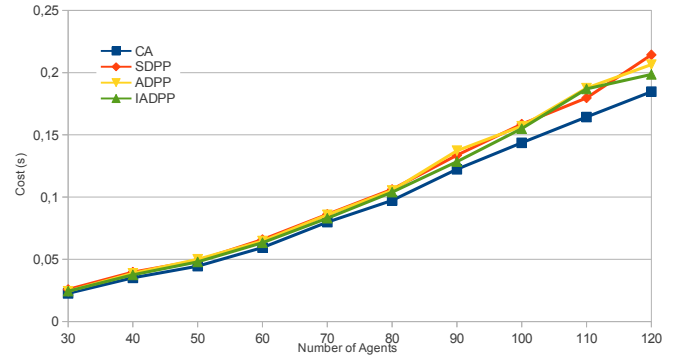
- [1] M. Bennewitz, W. Burgard, and S. Thrun. Finding and optimizing solvable priority schemes for decoupled path planning techniques for teams of mobile robots. *Robotics and Autonomous Systems*, 41(2):89–99, 2002.
- [2] K. Mani Chandy and Leslie Lamport. Distributed snapshots: determining global states of distributed systems. *ACM Trans. Comput. Syst.*, 3(1):63–75, February 1985.
- [3] Michael Erdmann and Tomas Lozano-prez. On multiple moving objects. *Algorithmica*, 2:1419–1424, 1987.
- [4] Carlo Ferrari, Enrico Pagello, Jun Ota, and Tamio Arai. Multirobot motion coordination in space and time. *Robotics and Autonomous Systems*, 25(3-4):219 – 229, 1998.
- [5] N. J. Nilsson P. E. Hart and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems, Science, and Cybernetics*, SSC-4(2):100–107, 1968.
- [6] Dave Pottinger. Implementing coordinated movement. *Game Developer*, 2007.
- [7] David Silver. Cooperative pathfinding. In *AIIDE*, pages 117–122, 2005.
- [8] Trevor Scott Standley. Finding optimal solutions to cooperative pathfinding problems. In Maria Fox and David Poole, editors, *AAAI*. AAAI Press, 2010.
- [9] Jur P. van den Berg and Mark H. Overmars. Prioritized motion planning for multiple robots. In *IROS*, pages 430–435, 2005.
- [10] Prasanna Velagapudi, Katia P. Sycara, and Paul Scerri. Decentralized prioritized planning in large multirobot teams. In *IROS*, pages 4603–4609. IEEE, 2010.



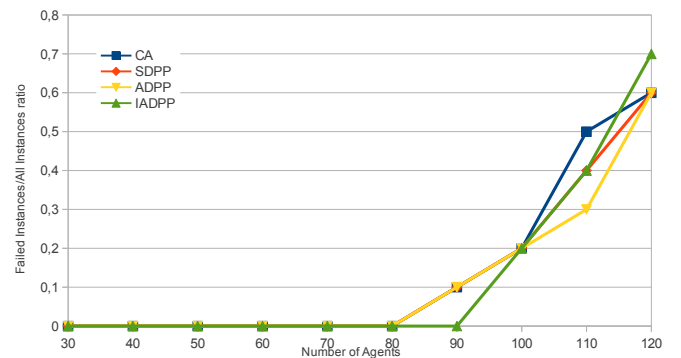
(a) Average wall-clock runtime for n -agent random scenario



(b) Average messages broadcasted



(c) Average cost (prolongation of trajectories)



(d) Failed instances ratio

Figure 7: Results from the random scenario

Now: Between Indexical Thoughts and Indexical Thought

Haythem Ismail¹

Abstract. It has long been argued that rational action necessitates having indexical thoughts. The ability to understand and generate indexical expressions is also a precondition for interacting in natural language. A rational acting agent that interacts in natural language is, thus, required to develop a proper treatment of indexicality. Treatments of indexicality in the literature have, almost always, assumed an indexical language of thought. I shall argue that, in most common cases, all indexical expressions in said language may be reduced to expressions involving only counterparts of “I” and “now”. I dismiss the language-of-thought “I” as being not indexical at all; this leaves us with “now”. First, I review past approaches to representing and reasoning about “now”, and systematically evaluate them against four criteria which I develop and motivate. Second, I push forward a totally different treatment of “now”, based on a grounded, layered agent architecture. In this architecture, the language of thought—at the top, knowledge layer—is a classical, first-order, non-indexical language; the indexical “now” features at a lower, perceptuo-motor layer. It is the reasoning process that cuts across both layers, giving rise to indexical thinking. Thus, we trade *indexical thoughts* for *indexical thought*. The proposed treatment is shown to supersede previous approaches with respect to the proposed criteria.

1 INTRODUCTION

Nothing is perhaps more common to our talk than indexical expressions. Indexicals, such as “I”, “you”, “this”, “here”, and “now” are familiar to native speakers and provide quite efficient means of communication. Their distinguishing feature is that, though unambiguous, their reference is determined and often *only* determined by the non-linguistic context of use. Thus, upon hearing an indexical like “I”, one immediately identifies the speaker as its referent; no knowledge other than knowledge of the context of utterance is required. On the other hand, non-indexical expressions typically mandate knowledge of the linguistic context (for example, with pronouns) or general world knowledge (for example, with proper nouns or descriptive noun phrases) to identify their referents. Yet, indexicals have proved to be tough beasts to the semanticist and the philosopher, as attested by the amount of ink they have drawn over the years [13, 30, 20, 3, 22, 5, for example].

According to Kaplan’s classical treatment of indexicals [13], the interpretation of linguistic expressions happens in two stages (three, if we consider intensionality): given an expression, we, first, consider its *character*, and, second, given the context, we compute the *content* of the expression. Here, the content is the referent of the expression, and the character is a function from contexts to contents. For non-indexical expressions, the character is a constant function. For

indexicals, the character determines the *meaning* of the expression. For example, the characters of “I”, “now”, and “here” map a context to the speaker, the time, and the location of the context, respectively. Kaplan’s theory did not go unchallenged, and several authors have attacked several aspects of it [30, 20, 3, for example]. Although I will have nothing much to say about these debates, I will come back to them later.

The aspect of indexicals that I am concerned with in this short note has to do with the observed relation between rational action and indexicals. In a (rightly) celebrated article [21], John Perry argues that beliefs underlying rational action often involve an essential indexical—one that cannot be replaced by a non-indexical or, by any means, be swept under the carpet.² But what is an indexical thought? And why is it necessary for rational action? The following examples will illustrate the point.

The Case of the Messy Shopper [21]. You are in the supermarket and you notice a trail of sugar on the supermarket floor. You go around, following that trail, trying to find the messy shopper. Suddenly, you realize that *you* are the messy shopper and, hence, you adjust your own sac of sugar.

What is it that you came to believe which motivated you to adjust your sac of sugar?

The Case of the Lost Hiker [21]. You are hiking and you get lost. You have a map, and you reason that if you can get to a certain, marked location on the map, Mt. Tallac, you will be able to get back on track. After some aimless walking, you suddenly realize that you are facing Mt. Tallac. Accordingly, you start following the map to your destination.

What is it that you came to believe which motivated you to start following the map?

The Case of the Fire Alarm [11]. You believe that whenever the fire alarm sounds you should leave the building. Suddenly, you hear the fire alarm and, hence, you leave the building.

What is it that you came to believe which motivated you to leave the building?

According to most authors (particularly, Perry), the answers to the three questions is that it is an indexical thought that triggered the reasoning agent to act. In particular, these are the beliefs “*I* am the messy shopper”, “Mt. Tallac is *here*”, and “the alarm is sounding *now*”, respectively. As Perry suggests, the indexicals in these thoughts are essential since they cannot be replaced by any co-referring terms. Thus, coming to believe that “John Perry is the messy shopper” or “the author of ‘the problem of the essential indexical’ is the messy

¹ Department of Engineering Mathematics, Cairo University and Department of Computer Science, German University in Cairo, e-mail: haythem.ismail@guc.edu.eg

² However, among others, Millikan [17] argues that such beliefs indeed include certain essential ingredients that are, nonetheless, not indexical.

shopper” cannot possibly be what would motivate John Perry to adjust his sac of sugar, since he might not believe that *he* is John Perry or that *he* is the author of ‘the problem of the essential indexical.’ Similarly, coming to believe that “the alarm is sounding at 12:00” at 12:00 does not explain why the agent would leave the building unless it also believes that “now is 12:00.” Ditto for the lost hiker.

In artificial intelligence (AI), since acting agents are our favorite subjects of research, we have to face Perry’s problem head on. Approaches to indexicality in AI, almost unanimously, take Perry’s remarks about “indexical thoughts” pretty faithfully, by building indexicality in the logical language used for representing the reasoning agent’s beliefs [6, 15, 16, 1, 4]. I beg to differ. Although I agree with Perry, and with those authors, that indexicality is essential for rational action, I contend (partly with Millikan in [17]) that indexicality is more suitably built, not in the knowledge representation language, but in the interaction of reasoning and acting. Thus, I propose to trade indexical *thoughts* for indexical *thought*.

Here is my plan. In Section 2, I adopt and motivate the position that for the purpose of rational action and straightforward linguistic interaction, all relevant indexicals are reducible to “I” and “now.” Hence, I argue that, under some reasonable assumptions, “I” is not indexical at all. Thus, a proper treatment of “now” is all that is indeed for a proper treatment of indexicality (for rational action and simple linguistic interaction). In Section 3, I lay down some criteria for evaluating such a treatment and I evaluate previous approaches against them. In Section 4, I present my own proposal, evaluating it with respect to our criteria in Section 5.

2 “I” and “Now”

Kaplan assumes that contexts of utterance satisfy the requirement that the speaker is at the location of the utterance, at the time of the utterance [13]. Hence, he concludes, all utterances of “I am here now” are analytically true. This conclusion came under fire by several authors and for several reasons. For example, certain uses of indexicals in linguistic discourse that are common, but not typical, involve so-called “deferred reference”:

- (1) Condemned prisoner: “I am traditionally allowed to order whatever I like for my last meal” [20, p.20]
- (2) Chess instructor to student: “According to all textbooks, you often get in trouble with that move” [20, p.21]
- (3) Medical pathologist pointing at a spot on his chest: “When a person is shot here, we can usually conclude that it was not suicide” [20, p.29]

In addition, occurrences of indexical expressions in written or recorded messages are also problematic for Kaplan’s theory.

- (4) Message played by an answering machine: “I am not here now” [30]

Had we been concerned with fully interpreting linguistic utterances and generating subtle ones, we will have had to face such examples head on. But since our main concern is the role of indexicality in rational action and straightforward linguistic discourse, we may safely disregard such uses of indexicals. In particular, I assume that knowledge representation and reasoning are carried by and in a logical language of thought. There is no evidence, and there is no reason why we should assume, that expressions in said language of

thought include similar uses of indexicals.³ As pointed out, the above data will be pressing if we are to tackle the tricky issue of translation between the language of thought and natural language.

Hence, I will assume that, in a language of thought, all indexicality may be reduced to expressions involving only “I” and “now”, as illustrated in Table 1.

Expression	Reduction
here	the location of I now
you	the addressee of I now
(demonstrative) this	the object you is pointing at now
today	the day of now
tomorrow	the day after today

Table 1. Indexicals in terms of “I” and “now”

Of course the entries in the above table do not account for all indexical expressions, but they include those which, it is reasonable to assume, play a role in rational action (especially, “you”, “here”, and “this”).

Now, are “I” and “now” really indexical? I think it is beyond doubt that they, or their counterparts in other natural languages, are indeed indexical. However, remember that we are not concerned with natural languages. Rather, our focus is on a language of thought (LOT). I believe that the LOT “now” is indexical, but the LOT “I” is not. To see why, note that there are at least two differences between a LOT and a natural language:

1. a LOT is private to each agent, a natural language is public; and
2. natural language expressions can be uttered, LOT expressions cannot be uttered, they can only be thought.

Alluding to Kaplan, indexicality of LOT is, thus, rooted in sensitivity to a context of thought, not a context of utterance. Hence, the character of LOT “I” is the *thinker* of the context. But, since the LOT is private, this character is a constant function, yielding the same thinker in every context, namely the reasoning agent whose LOT we are considering. Hence, nothing is indexical about LOT “I.” It is just a constant of the LOT, which is psychologically marked for action.⁴

LOT “now”, on the other hand, retains its indexicality. Since thought occurs over time, two numerically distinct contexts of thought necessarily have different times. Thus, the character of LOT “now” is not a constant function and, hence, LOT “now” is indeed indexical.

3 “Now”

3.1 Tense-Marker and Post-It-Note “Now”

We may distinguish two uses of the English “now” that correspond to two functions LOT “now” may serve. These two uses are exemplified by the following sentences.

- (5) Speaker looking outside a window: “It is now raining.”
- (6) A note posted on a door: “I am not in now.”

³ Millikan [18] might disagree though.

⁴ How are we then to explain the behavior of the messy shopper? Wait for Section 5.

In (5), “now” merely serve as a tense marker, indicating that the raining is contemporaneous with the utterance. Let us refer to such uses of “now” as *tense-marker* “now.” In (6), “now” is used to refer, not to the unique time of writing the note, but to any time at which the note is read [22, 5]. Thus, the “now” written on the post-it note changes its reference as times goes by. Let us refer to such uses of “now” as *post-it-note* “now.”⁵

These two uses of “now” serve quite important purposes in a LOT. Tense-marker LOT “now” allows an agent to distinguish the present from the past and the future. There are at least two reasons why this distinction is important. First, if the agent expresses its beliefs in natural language, then a distinction between the present, the past, and the future is required for generating sentences with the correct tense expressing what has happened, is happening, or will happen. Second, and more important, present facts have a distinguished status for an acting agent. For example, for our fire alarm agent from Section 1, if we tell the agent that the alarm sounded yesterday, it merely needs to remember this fact, and maybe derive some inferences from it. However, if we tell it that the fire-alarm is *now* sounding, it also needs to act on this fact and leave the building.

Instead of posting a note on a door, acting agents often use mental post-it notes to remind themselves of important future actions. Again, for our fire alarm agent, forming the attitude that whenever the alarm sounds it should leave the building may be achieved by posting a mental note saying (in mentalese) “if the fire alarm is now sounding, leave the building.” The LOT “now” occurring in this LOT expression is a post-it-note “now”, not a tense-marker “now.”

While it may seem that tense-marker “now” is more common than the exotic post-it-note “now”, it is interesting to note that most approaches to the representation of “now” in AI have been about the latter and not the former. We need to investigate these approaches; but, first, we should consider how we are to evaluate them.

3.2 Criteria for an Adequate Treatment

For a treatment of “now” to be adequate for a rational agent that can potentially interact in simple natural language (cf. [26]), reporting on what it has done and is doing, it has to satisfy at least the following four criteria (listed in no significant order).

- N1.** *The treatment should account for tense-marker “now” and post-it-note “now.”* A treatment which accounts for one and not the other is lacking.
- N2.** *The treatment should account for temporal progression.* As pointed out above, LOT “now” is always changing its referent. A treatment of “now” which does not reflect this intuition is (i) not psychologically adequate and (ii) cannot accommodate an acting agent, since acting does take time.
- N3.** *The treatment should be as computationally tractable as possible.* Of course, this is a matter of degree, but nothing is more mundane than the passage of time, and a treatment that burdens the agent with a relatively heavy computation to catch up with the passage of time is both psychologically and computationally unfavorable.
- N4.** *The treatment should not make unmotivated assumptions.* In general, a treatment should not impose constraints on, e.g., the structure of time, the agent’s beliefs, the agent’s actions, that are

only needed for the treatment to be adequate. Again, this is a matter of degree, but assumptions about the agent or the ontology should be independently motivated as much as possible.

Given the above criteria, let us consider how existing treatments of “now” fair.

3.3 “Now” in the Literature

Treatments of “now” in the literature may be divided into three major classes. First, there are the classical Priorian tense logics [23]. Classical tense logic is essentially a temporally-interpreted bimodal logic. If p is a proposition, “ Pp ” means that “It has been the case that p ” and “ Fp ” means that “It will be the case that p ”. By itself, “ p ” refers to the *current* truth of p . Thus, syntactically, the present is distinguished by having the proposition outside the scope of any tense operators. Semantically, expressions (which may be embedded within tense operators) are interpreted with respect to a particular temporal index representing the present. Other treatments, within the same framework, explicitly introduce a “now” tense operator N to account for the curious property of the English “now” that, in typical uses, it always refers to the time of utterance even when embedded within a nest of tense operators [12].

Although it clearly accounts for tense-marker “now”, the tense logical approach fails to account for post-it-note “now” and for temporal progression; thus violating **N1** and **N2** and, hence, avoiding **N3** and **N4**.

The second approach, usually adopted in reasoning about actions and plans, is to represent the present using an indexical **now** term. The use of indexical terms, in general, was studied in depth by Lespérance and Levesque in [16] with special attention to the case of **now** in [15].⁶ The indexicality of such a term stems from its having a context-dependent interpretation, much in the same spirit of Kaplan’s semantics discussed above. However, unlike the English “now”, whose content depends on the context of utterance (or assertion), the semantics of the indexical **now** depends on the *evaluation* context. In the context of acting and planning, it is the time of executing a particular instance of a plan that includes occurrences of **now** in its specification. Along the lines of [15] (and using the same syntax), the following is a possible representation of a plan to get to other side of the street (probably for a rather despondent agent):

if(At(now, WALKLIGHTON),CROSS,noOp)

This roughly says that, if, at that the time of performing the action, the walk-light is on, then cross the street; otherwise do nothing. What should be noted is that **now** in the above form does not refer to the time of introducing the form into the knowledge base, or to any other *fixed* time for that matter. It is, in a sense, a place-holder for any time at which the plan is performed.

What about temporal progression? Lespérance and Levesque briefly discuss an approach which we will now consider in some detail. The obvious approach to modelling the passage of time within the theory of [16] would be to appropriately edit the knowledge base every time “now” changes in order to preserve the truth of its sentences. Thus, **At(now, RAINING)** should be replaced by something more appropriate once “now” changes. One problem, of course, is that such updates are computationally expensive. To get around the problem, [16, p. 101] suggest that “if all occurrences of ‘now’ are replaced by a new constant and the fact that this new constant is equal

⁵ It should be clear that nothing is special here about the word “now” itself; “now” may be dropped from both (5) and (6) without affecting their propositional contents. Rather, it is the present tense in both sentences that really plays the two indicated roles.

⁶ Other authors have also used the same or a similar approach [25, 1, 4].

to ‘now’ is added, then only this single assertion need be updated as time passes.” This indeed eliminates the problem of expensive belief update and provides a neat logical and computational account of “now”.

I believe that Lespérance and Levesque’s treatment of “now” satisfies **N1** and **N2**. However, I also believe that it does not fair well with respect to **N3** and **N4**.

First, note that, though they do not mention it, Lespérance and Levesque’s treatment of equality will have to be subtler than usual, if their approach to temporal progression is to work effectively. For example, one should block instances of rules like demodulation when the term involved in the conclusion is **now**, since we do not want to express any transient beliefs using **now** itself but, rather, the non-indexical term it is currently equal to. This is a violation of both **N3** and **N4**.

Second, from a cognitive perspective, I find the very idea of *erasing* sentences from an agent’s knowledge base as time passes by far from natural. If such sentences represent beliefs that the agent once held, where do they go, and how come the agent would have no memory of them once time passes? Note that this cannot be explained away as a matter of forgetting, for forgetting is not that selective to always affect beliefs involving “now”, nor is it vigorous enough to take effect with every tick of the clock. The only way to explain this mysterious disappearance of beliefs is by arguing that they exist at a lower level of consciousness with respect to other beliefs. If this were the case, why are such beliefs part of the logical theory (which we take to be representing conscious beliefs of the agent)? This points to a possible violation of **N4**.

The third approach to represent “now” is to do it indirectly, by means of a *Now* predicate, where the expression $Now(i)$ means that the current time is represented by the term i . This is exactly the method adopted in *active logic*, originally known as *step logic* [6]. Temporal individuals are represented by integers, with the usual numerical order implicitly representing chronological order. In active logic, time moves with every inference step. This movement of time is represented both logically and meta-logically. Logically, this is achieved by a special inference rule that essentially replaces $Now(i)$ by $Now(i+1)$. Meta-logically, assertions are associated with the step, i , of inference at which they were asserted.

Though there clearly is an account for tense-marker “now” in active logic, I am not aware of an explicit treatment of post-it-note “now.” **N2** and **N3** are, I believe, observed by active logic. However, it is **N4** which I think is somehow violated. Apparently, the use of integers facilitates the expression of some crucial rules of inference (also the *counting* of reasoning steps [19]) that depend on having a well-defined notion of the *next moment of time*, represented by the integer successor operator. However, such a representation forces a certain degree of rigidity on the kind of knowledge that may be entered into the system. For example, there is no way to assert at step $i+m$ ($m > 1$) that a certain event e_2 occurred between events e_1 and e_3 that happened at times i and $i+1$, respectively. In other words, once “now” moves, there is no way to go back and create arbitrary past temporal locations. This is definitely a big drawback if the system is to be used in interacting with humans, where assertions need not be only about the present.

4 INDEXICAL REASONING WITHOUT INDEXICAL REPRESENTATION

From the foregoing discussion, we may conclude that some way of representing and reasoning about “now” is essential for rational

action. Moreover, it seems fair to conjecture, at least temporarily, that it is *only* essential for rational action. That is, it is action—and not merely armchair reasoning *about* action, but actual *acting*—that mandates a treatment of indexicality; in the absence of action, no indexical is essential. This conjecture cannot be fully defended at this point; suffice it to say that all debates about indexicals are about the role they may or may not play in explaining behavior [21, 17, 31, for example]—no one has ever tried to argue for indexical thoughts in the absence of action.⁷ Even if, in the final analysis, indexicals turn out to have a more prominent role, independent of action (although I cannot imagine how), the treatment of “now” that I shall outline may still be valuable in relating indexicals to action.

But how can we relate indexical reasoning to action without mentioning a framework for relating the two activities: reasoning and acting? The work cited in Section 3, presents logical frameworks, without a mention of how the symbols of the logical language are grounded, nor of how the reasoning activities guide action. These issues are implicitly assumed to be somehow treated and they, no doubt, are. It is my conviction, however, that said treatment may be at least useful to consider as part of the very approach to indexicality. Hence, I will present an approach to reasoning about “now” based on a grounded agent architecture—GLAIR.

4.1 GLAIR

In the sequel, I assume a theory of agents based on the GLAIR agent architecture [7, 26]. GLAIR is a layered architecture consisting of three layers:

1. **The Knowledge Layer (KL):** The layer at which symbolic reasoning takes place. This layer may be implemented in any logic-based system, where anything that we may think or talk about is abstractly represented by a term, including actions and behaviors. (Historically, the KL has been implemented by the SNePS knowledge representation, reasoning, and acting system [27, 28, 29].) This is also the level responsible for interpreting composite action terms, and scheduling them for execution, as a result of reasoning or natural language instructions [26].
2. **The Perceptuo-Motor Level (PML):** The layer at which routines for carrying out primitive acts are located. This layer also includes an elaborate representation of perceivable physical objects, properties, and relations, typically in terms of feature vectors resulting from the processing of sensory input. The representation of an entity at this level is more fine-grained than the symbolic representation at the KL, so that the agent may, for example, be able to perceptually distinguish two physical objects given their PML representations, though it may not be able to discern them by mere KL reasoning. KL terms that represent objects which are also represented at the PML are grounded in the corresponding PML representations, through a relation of alignment. Likewise, KL terms representing primitive actions are aligned with the corresponding PML routines.
3. **The Sensori-Actuator Level (SAL):** The layer controlling the operation of sensors and actuators. I will have nothing much to say about the SAL henceforth.

The treatment of “now” to be presented below is based on the intuition that recognizing the passage of time is more a process of per-

⁷ In Lespérance and Levesque’s argument for indexical “reasoning” [15], it is actually not the “reasoning” per se that is required to be indexical. Rather, it is the representation of a plan that, they claim, mandates indexicality for its correct execution.

ception than one of conscious reasoning; one does not need to reason in order to determine that time has passed, one just *feels* the passage of time. Hence, tense-marker “now” will be accounted for by a careful synchronization of PML and KL activities which, I claim, gives rise to the sense of temporal progression. Post-it note “now”, on the other hand, is accounted for by building temporality in the very process of *practical* reasoning which, unlike armchair reasoning, is not limited to the manipulation of KL terms, but also involves consulting PML structures.

4.2 Language

I will take the KL language to be a first-order, sorted language \mathcal{L} , intended to be the language of thought of the agent. In what follows, we identify a sort s with the set of symbols of sort s . \mathcal{L} -terms are partitioned into eight base syntactic sorts, $\sigma_A, \sigma_G, \sigma_E, \sigma_T, \sigma_S, \sigma_N, \sigma_C$ and σ_O . Intuitively, terms of each sort respectively denote actions, agents, events, times, propositional fluents (or “states”), names, clock readings, and objects. Each denoting symbol of \mathcal{L} belongs to a unique sort from a set Σ of syntactic sorts. The set Σ is the smallest superset of $\sigma = \{\sigma_A, \sigma_G, \sigma_E, \sigma_T, \sigma_S, \sigma_N, \sigma_C, \sigma_O\}$ containing the following sorts.

1. $\times_{i=1}^k \tau_i$, and
2. $(\times_{i=1}^k \tau_i) \longrightarrow \tau$

where $\tau_i, \tau \in \sigma$, for $1 \leq i \leq k$, for every $k \in \mathbb{N}$. Intuitively, the above accounts for the syntactic sorts of k -adic predicate and function symbols, respectively.

The alphabet of \mathcal{L} is made up of Boolean connectives ($\neg, \wedge, \vee, \supset, \equiv$) and quantifiers (\forall, \exists); a set of syncategorematic punctuation symbols; a countably infinite set of variables; a set of domain-dependent constants, function symbols, and predicate symbols; and the two special symbols *When* and *Whenever*.

As usual, terms of \mathcal{L} are defined in the standard way as the closure of the set of constants and variables under combination with function symbols, provided that sort restrictions are observed. Similarly, well-formed formulas (WFFs) are defined as in any sorted first-order language. In addition to terms and WFFs, \mathcal{L} includes another kind of expression—well-formed directives (WFDs). A well-formed directive is an expression of the form *Whenever*(s, a) or *When*(s, a), where $s \in \sigma_S$ and $a \in \sigma_A$. Directives provide the link between reasoning and acting [14] and are akin to Millikan’s pushme-pullya representations [18]. Intuitively, *Whenever*(s, a) means that the agent will attempt to execute the action a whenever it comes to believe that s holds. *When*(s, a) is a once-only variant, where the agent follows the directive only once.

A full, careful exposition of the semantics of the WFD-free fragment of \mathcal{L} is not possible given space limitations and is, fortunately, mostly orthogonal to the issues at stake here. The unifying semantics of [2] more than suffices for the WFD-free fragment, but I briefly sketch important ingredients of \mathcal{L} -semantics for completeness. Expressions are interpreted with respect to an ontologically rich structure:

$$\mathfrak{M} = \langle \mathfrak{D}, < \rangle$$

where $\{\mathfrak{A}, \mathfrak{E}, \mathfrak{S}, \mathfrak{T}, \mathfrak{G}, \mathfrak{N}, \mathfrak{C}, \mathfrak{O}\}$ is a partition of the domain \mathfrak{D} . Intuitively, the parts are non-empty sets of, respectively, actions, events, states, time points, agents, names, clock readings, and objects. $< \subseteq \mathfrak{T}^2 \cup \mathfrak{C}^2$ is an irreflexive order which is partial on \mathfrak{T} and total on \mathfrak{C} . The interpretation function $\llbracket \cdot \rrbracket^{\mathfrak{M}}$ with respect to \mathfrak{M} is such

that if e is a term of some base sort σ , then $\llbracket e \rrbracket^{\mathfrak{M}}$ is in the corresponding part of \mathfrak{D} . For convenience, the superscript \mathfrak{M} will be dropped when there is no ambiguity.

Given the intended use of \mathcal{L} to serve as the language of thought of an acting agent, we constrain \mathcal{L} thus:

1. There is a constant $I \in \sigma_G$ denoting the agent itself (for itself). There is a predicate symbol *Name* $\in \sigma_G \times \sigma_N$, associating names with agents. $\llbracket \text{Name} \rrbracket$ is neither total nor tabular at \mathfrak{G} and \mathfrak{N} : agents do not necessarily have names and, if they do, said names need be neither unique nor exclusive, and names need not be associated with any agents.
2. Similar to the predicate *Name*, there is a function symbol *Clk* $\in \sigma_C \longrightarrow \sigma_S$, where $\llbracket \text{Clk} \rrbracket : \mathfrak{C} \longrightarrow \mathfrak{S}$ is a total, injective function, mapping each clock reading to the state of the clock’s displaying it.
3. Constants of sort σ_T are countably infinite and \mathfrak{T} is (possibly uncountably) infinite. Temporal order is represented by $< \in \sigma_T \times \sigma_T$, where $\llbracket < \rrbracket = <_{\mathfrak{T}^2}$ (the restriction of $<$ to \mathfrak{T}^2).
4. A predicate symbol *HoldsAt* $\in \sigma_S \times \sigma_T$ represents the incidence of states on time points. *HoldsOn* $\in \sigma_S \times \sigma_T \times \sigma_T$ represents *homogeneous* incidence on intervals: $\llbracket \text{HoldsOn} \rrbracket = \{\langle s, t_1, t_2 \rangle \mid \text{for every } t_1 < t < t_2, \langle s, t \rangle \in \llbracket \text{HoldsAt} \rrbracket\}$. *HoldsAt* and *Clk* are synchronized in the following sense: if $\langle \llbracket \text{Clk} \rrbracket(c_1), t_1 \rangle, \langle \llbracket \text{Clk} \rrbracket(c_2), t_2 \rangle \in \llbracket \text{HoldsAt} \rrbracket$, then $c_1 < c_2$ if and only if $t_1 < t_2$.
5. Event terms are constructed by a function symbol *Does* $\in \sigma_G \times \sigma_A \longrightarrow \sigma_E$, where $\llbracket \text{Does} \rrbracket : \mathfrak{G} \times \mathfrak{A} \longrightarrow \mathfrak{E}$ is a bijection. Hence, actions are the only events, and no group actions are considered. A predicate symbol *Occurs* $\in \sigma_E \times \sigma_T \times \sigma_T$ represents event occurrence. Intuitively, $\langle e, t_1, t_2 \rangle \in \llbracket \text{Occurs} \rrbracket$ when event e occurs on the interval (t_1, t_2) .
6. Actions are atomic or composite. Atomic actions are denoted by functional terms formed by domain-dependent symbols and the special action term *NoOp*. Composite actions are, as usual, denoted by functional terms corresponding to imperative programming constructs: *Seq* $\in \sigma_A \times \sigma_A \longrightarrow \sigma_A$, *If* $\in \sigma_S \times \sigma_A \times \sigma_A \longrightarrow \sigma_A$, and *While* $\in \sigma_S \times \sigma_A \times \sigma_A \longrightarrow \sigma_A$. The semantics is given in terms of constraints on action occurrences:
 - For every $a \in \mathfrak{A}$ and $\langle \llbracket \text{Does} \rrbracket(a, \llbracket \text{NoOp} \rrbracket), t_1, t_2 \rangle \in \llbracket \text{Occurs} \rrbracket$, $t_1 = t_2$.
 - For every $a \in \mathfrak{A}$ and $\langle \llbracket \text{Does} \rrbracket(a, \llbracket \text{Seq} \rrbracket(\alpha, \beta)), t_1, t_2 \rangle \in \llbracket \text{Occurs} \rrbracket$, there is $t \in \mathfrak{T}$, $t_1 < t < t_2$ with $\langle \llbracket \text{Does} \rrbracket(a, \alpha), t_1, t \rangle, \langle \llbracket \text{Does} \rrbracket(a, \beta), t, t_2 \rangle \in \llbracket \text{Occurs} \rrbracket$.
 - For every $a \in \mathfrak{A}$ and $\langle \llbracket \text{Does} \rrbracket(a, \llbracket \text{If} \rrbracket(s, \alpha, \beta)), t_1, t_2 \rangle \in \llbracket \text{Occurs} \rrbracket$, if $\langle s, t_1 \rangle \in \llbracket \text{HoldsAt} \rrbracket$, then $\langle \llbracket \text{Does} \rrbracket(a, \alpha), t_1, t_2 \rangle \in \llbracket \text{Occurs} \rrbracket$, else $\langle \llbracket \text{Does} \rrbracket(a, \beta), t_1, t_2 \rangle \in \llbracket \text{Occurs} \rrbracket$.
 - For every $a \in \mathfrak{A}$ and $\langle \llbracket \text{Does} \rrbracket(a, \llbracket \text{While} \rrbracket(s, \alpha)), t_1, t_2 \rangle \in \llbracket \text{Occurs} \rrbracket$, $\langle \llbracket \text{Does} \rrbracket(a, \llbracket \text{If} \rrbracket(s, \llbracket \text{Seq} \rrbracket(\alpha, \llbracket \text{While} \rrbracket(s, \alpha)), \text{NoOp})), t_1, t_2 \rangle \in \llbracket \text{Occurs} \rrbracket$.

For completeness, an \mathcal{L} -theory will include axioms capturing the above constraints on interpretation. I also assume the existence of predicate and function symbols to represent preconditions and effects of actions. However, I do not take the agent to be a planning agent; rather, the agent is provided with \mathcal{L} -representations of recipes of action to achieve desired states. These assumptions are, nonetheless, totally harmless, given the nature of our task.

4.3 The PML

The language \mathcal{L} , comprising the symbolic structures at the KL, is an objective, non-indexical, first-order language. As such, it suffices for reasoning *about* action and time. But as the examples of Section 1 attest, more is needed. To arrive at an adequate treatment of indexicality, we need to now turn to the PML. We may describe the relevant aspects of the PML using the notion of a PML state.

Definition 1 A PML state is a quadruple $\mathbb{P} = \langle \Pi, \gamma, \Sigma, *NOW \rangle$, where

1. Π is a set of PML representations (typically, feature vectors) of perceivable entities (objects, properties, and relations), and behaviors that directly control the SAL.
2. γ , the grounding relation, is set of pairs of \mathcal{L} -terms and members of Π . In particular, γ is functional and left-total on the set of atomic σ_A -terms, mapping each such term a into a routine $\gamma(a)$.
3. Σ is a sequence of σ_A -terms representing acts scheduled for execution.
4. NOW is a PML variable, whose value, at any time, is a σ_T -constant. The $*$ is a de-referencing operator and, hence, $*NOW$ is the σ_T -constant which is the value of NOW in the state.

The σ_T -constant $*NOW$ denotes the current time, for the agent, in a given PML state. This term is distinguished in practical reasoning by being the value of the variable NOW .

4.4 Dynamics

Reasoning, acting, and perception change the state of the agent. Such changes are governed by a set of rules. In the case of a reasoning-only agent, these are logical rules of inference. In the case of a reasoning *and* acting agent, we need to generalize the notion of an inference rule. When interpreted operationally, classical rules of inference transform one *belief* state into another. Hence, the first step in generalizing inference rules is to generalize the notion of a state.

Definition 2 An agent state is a triple $\mathbb{S} = \langle \mathbb{K}, \mathbb{D}, \mathbb{P} \rangle$, where \mathbb{K} is a set of \mathcal{L} -WFFs, \mathbb{D} is a set of \mathcal{L} -WFDs, and \mathbb{P} is a PML state.

A practical inference cannon is a mapping (which is not necessarily functional) from agent states to agent states. A common way of viewing this mapping is as a set of transformation rules on agent states. Such rules sometimes have preconditions and side effects. Rules will be displayed as follows

$$Pre|\langle \mathbb{K}, \mathbb{D}, \mathbb{P} \rangle \longrightarrow \langle \mathbb{K}', \mathbb{D}', \mathbb{P}' \rangle | Eff$$

where, Pre is a set of preconditions and Eff is a set of effects. Preconditions are typically conditions on \mathcal{L} -terms appearing in \mathbb{K} ; effects are exclusively of the form $Initiate(\beta)$, where β is a PML behavior in Π . This indicates that a side-effect of applying the rule is for the agent to start carrying out β . A rule $Pre|\mathbb{S} \longrightarrow \mathbb{S}' | Eff$ is applicable to state \mathbb{S} if its preconditions are satisfied.

Definition 3 Let $r = Pre|\langle \mathbb{K}, \mathbb{D}, \mathbb{P} \rangle \longrightarrow \langle \mathbb{K}', \mathbb{D}', \mathbb{P}' \rangle | Eff$ be a rule with $\mathbb{P} = \langle \Pi, \gamma, \Sigma, *NOW \rangle$ and $\mathbb{P}' = \langle \Pi', \gamma', \Sigma', *NOW' \rangle$.

1. r is said to be an inference rule if $\mathbb{D}' = \mathbb{D}$, $\mathbb{P}' = \mathbb{P}$, and $Eff = \emptyset$. \mathbb{K} is deductively-closed, denoted $\mathbb{K} = \text{Cn}(\mathbb{K})$ if it is closed under the application of all inference rules.
2. r is said to be a decomposition rule if $\mathbb{K}' = \mathbb{K} = \text{Cn}(\mathbb{K})$, $\mathbb{D}' = \mathbb{D}$, $\Pi = \Pi'$, $\gamma = \gamma'$, $*NOW = *NOW'$, $\Sigma \neq \langle \rangle$ and $Eff = \emptyset$.

3. r is an initiation rule if $\mathbb{K} = \text{Cn}(\mathbb{K})$, $\mathbb{D}' = \mathbb{D}$, $\Pi = \Pi'$, $\gamma = \gamma'$, $\Sigma \neq \langle \rangle$, and $Eff \neq \emptyset$.
4. r is a directive rule if $\mathbb{K}' = \mathbb{K} = \text{Cn}(\mathbb{K})$, $\Pi = \Pi'$, $\gamma = \gamma'$, $*NOW = *NOW'$, $\Sigma = \langle \rangle$, and $\mathbb{D} \neq \emptyset$

As is customary, the above rules define a *yielding* relation between agent states.

Definition 4 An agent state \mathbb{S} yields an agent state \mathbb{S}' , denoted $\mathbb{S} \longrightarrow^+ \mathbb{S}'$, if there is sequence of states $\mathbb{S}_1, \dots, \mathbb{S}_n$, such that $\mathbb{S}_1 = \mathbb{S}$, $\mathbb{S}_n = \mathbb{S}'$, and, for every $1 \leq i < n$, there is a rule $Pre|\mathbb{S}_i \longrightarrow \mathbb{S}_{i+1} | Eff$ which is applicable to \mathbb{S}_i . A sequence of rules taking the agent from state \mathbb{S} to state \mathbb{S}' via the intermediate states is called an $(\mathbb{S}, \mathbb{S}')$ -path.

Definition 3 imposes a strict order on the application of rules: all applicable inference rules must be first applied, followed by decomposition and initiation rules, and finally followed by directive rules.

Observation 4.1 Let \mathbb{S} be an agent state.

1. If there is a state \mathbb{S}' and an $(\mathbb{S}, \mathbb{S}')$ -path which is a sequence of inference rules, then no decomposition, initiation, or directive rule is applicable to \mathbb{S} .
2. If there is a state \mathbb{S}' and an $(\mathbb{S}, \mathbb{S}')$ -path which is a sequence of inference, decomposition, and initiation rules, then no directive rule is applicable to \mathbb{S} .

Figure 1 shows the set of rules we consider for our agent. Absent from this figure, and the entire discussion, is any mention of perception. Perception results in changing the agent state when PML routines read-off the values of SAL sensors and interpret them by constructing PML representations. The effect of perception is primarily on Π , γ , \mathbb{K} , $*NOW$, and possibly \mathbb{D} (see [26, 10]).

The rules in Figure 1 embody (at least) three simplifying (but totally inert) assumptions about our agent:

1. Atomic actions are punctual and immediately successful. A more careful approach is, in general, called for. (See [8, 9].)
2. Time moves (i.e., $*NOW$ changes) only when the agent acts. We could have chosen otherwise. For example, following [6], each rule may be defined to change $*NOW$ akin to the initiation rule. Alternatively, we may have rules dedicated to changing $*NOW$, which are applied synchronously with a PML *pacemaker* [26]. Whatever the choice, not much depends on it when it comes to the proposed treatment of indexicality.
3. A solution to the frame problem needs to be incorporated in order to account for which states persist and which do not as a result of applying an initiation rule. Again, this is not the main concern here, and a monotonic solution to the frame problem (e.g., [24]) will suffice for our purposes.

Given these rules, we can prove that, the agent's beliefs can only expand as time unfolds and that, as far as the agent is concerned, time unfolds from the past to the future.

Proposition 4.1 If $\mathbb{S} \longrightarrow^+ \mathbb{S}'$ then

1. $\mathbb{K} \subseteq \mathbb{K}'$ and
2. $*NOW = *NOW'$ or $*NOW \prec *NOW' \in \mathbb{K}'$.

Since we assume a sound and complete set of inference rules, the agent's reasoning will be sound and complete given a correct and complete axiomatization of the domain-independent symbols of \mathcal{L} . However, we also need to verify that the rules guide the agent to correct execution of actions. To this end, we need some terminology.

Inference Rules. Any set of monotonic, first-order rules which is sound and complete for \mathcal{L} .

Decomposition Rules. In what follows, \odot represents sequence concatenation.

1. $\{\mathbb{K} = \text{Cn}(\mathbb{K})\} | \langle \mathbb{K}, \mathbb{D}, \langle \Pi, \gamma, \text{NoOp} \odot \Sigma, *NOW \rangle \rangle \longrightarrow \langle \mathbb{K} \cup \{\text{Occurs}(\text{Does}(\mathbb{I}, \text{NoOp}), *NOW, *NOW)\}, \mathbb{D}, \langle \Pi, \gamma, \Sigma, *NOW \rangle \rangle | \emptyset$
2. $\{\mathbb{K} = \text{Cn}(\mathbb{K})\} | \langle \mathbb{K}, \mathbb{D}, \langle \Pi, \gamma, \text{Seq}(\alpha, \beta) \odot \Sigma, *NOW \rangle \rangle \longrightarrow \langle \mathbb{K}, \mathbb{D}, \langle \Pi, \gamma, \alpha \odot \beta \odot \Sigma, *NOW \rangle \rangle | \emptyset$
3. $\{\mathbb{K} = \text{Cn}(\mathbb{K}), \text{Holds}(s, *NOW) \in \mathbb{K}\} | \langle \mathbb{K}, \mathbb{D}, \langle \Pi, \gamma, \text{If}(s, \alpha, \beta) \odot \Sigma, *NOW \rangle \rangle \longrightarrow \langle \mathbb{K}, \mathbb{D}, \langle \Pi, \gamma, \alpha \odot \Sigma, *NOW \rangle \rangle | \emptyset$
4. $\{\mathbb{K} = \text{Cn}(\mathbb{K}), \text{Holds}(s, *NOW) \notin \mathbb{K}\} | \langle \mathbb{K}, \mathbb{D}, \langle \Pi, \gamma, \text{If}(s, \alpha, \beta) \odot \Sigma, *NOW \rangle \rangle \longrightarrow \langle \mathbb{K}, \mathbb{D}, \langle \Pi, \gamma, \beta \odot \Sigma, *NOW \rangle \rangle | \emptyset$
5. $\{\mathbb{K} = \text{Cn}(\mathbb{K}), \text{Holds}(s, *NOW) \in \mathbb{K}\} | \langle \mathbb{K}, \mathbb{D}, \langle \Pi, \gamma, \text{While}(s, \alpha) \odot \Sigma, *NOW \rangle \rangle \longrightarrow \langle \mathbb{K}, \mathbb{D}, \langle \Pi, \gamma, \alpha \odot \text{While}(s, \alpha) \odot \Sigma, *NOW \rangle \rangle | \emptyset$
6. $\{\mathbb{K} = \text{Cn}(\mathbb{K}), \text{Holds}(s, *NOW) \notin \mathbb{K}\} | \langle \mathbb{K}, \mathbb{D}, \langle \Pi, \gamma, \text{While}(s, \alpha) \odot \Sigma, *NOW \rangle \rangle \longrightarrow \langle \mathbb{K}, \mathbb{D}, \langle \Pi, \gamma, \Sigma, *NOW \rangle \rangle | \emptyset$

Initiation Rule.

$\{\alpha \text{ is atomic}, \mathbb{K} = \text{Cn}(\mathbb{K}), t_2 \text{ and } t_3 \text{ appear nowhere in } \mathbb{K}\} | \langle \mathbb{K}, \mathbb{D}, \langle \Pi, \gamma, \alpha \odot \Sigma, t_1 \rangle \rangle \longrightarrow \langle \mathbb{K} \cup \{t_1 \prec t_2, t_2 \prec t_3, \text{Occurs}(\text{Does}(\mathbb{I}, \alpha), t_2, t_2)\}, \mathbb{D}, \langle \Pi, \gamma, \Sigma, t_3 \rangle \rangle | \{\text{Initiate}(\gamma(\alpha))\}$

Directive Rules.

1. $\{\mathbb{K} = \text{Cn}(\mathbb{K}), \text{Holds}(s, *NOW) \in \mathbb{K}\} | \langle \mathbb{K}, \{\text{Whenever}(s, \alpha)\} \cup \mathbb{D}, \langle \Pi, \gamma, \langle \rangle, *NOW \rangle \rangle \longrightarrow \langle \mathbb{K}, \{\text{Whenever}(s, \alpha)\} \cup \mathbb{D}, \langle \Pi, \gamma, \langle \alpha \rangle, *NOW \rangle \rangle | \emptyset$
2. $\{\mathbb{K} = \text{Cn}(\mathbb{K}), \text{Holds}(s, *NOW) \in \mathbb{K}\} | \langle \mathbb{K}, \{\text{When}(s, \alpha)\} \cup \mathbb{D}, \langle \Pi, \gamma, \langle \rangle, *NOW \rangle \rangle \longrightarrow \langle \mathbb{K}, \mathbb{D}, \langle \Pi, \gamma, \langle \alpha \rangle, *NOW \rangle \rangle | \emptyset$

Figure 1. Rules of Practical Inference

Definition 5 Let \mathbb{S} and \mathbb{S}' be agent states and p be an $(\mathbb{S}, \mathbb{S}')$ -path. If $\langle r_1, \dots, r_n \rangle$ is the (longest) subsequence of p of instances of the initiation rule, then the action trace of p , denoted $\text{tr}(p)$, is the sequence $\langle (t_1, \alpha_1), \dots, (t_n, \alpha_n) \rangle$, where, for $1 \leq i \leq n$,

$$r_i = \text{Pre}|S_i \longrightarrow \langle \mathbb{K} \cup \{t_{i1} \prec t_i, t_i \prec t_{i3}, \text{Occurs}(\text{Does}(\mathbb{I}, \alpha_i), t_i, t_i)\}, \mathbb{D}, \langle \Pi, \gamma, \Sigma, t_{i3} \rangle \rangle | \{\text{Initiate}(\gamma(\alpha_i))\}$$

for some agent state S_i .

The following result immediately follows.

Proposition 4.2 If p is an $(\mathbb{S}, \mathbb{S}')$ -path, then $\text{tr}(p)$ is unique.

Proof. This follows since \mathbb{K}' carries the history of which atomic actions were performed when, given the monotonicity of inference and the WFF $\text{Occurs}(\text{Does}(\mathbb{I}, \alpha_i), t_i, t_i)$ added by the initiation rule. \square

Definition 6 Let \mathfrak{M} be an \mathcal{L} -structure and let p be an $(\mathbb{S}, \mathbb{S}')$ -path. \mathfrak{M} is p -faithful if, for every (t_i, α_i) in $\text{tr}(p)$, $\mathfrak{M} \models \text{Occurs}(\text{Does}(\mathbb{I}, \alpha_i), t_i, t_i)$.

For every, $(\mathbb{S}, \mathbb{S}')$ -path, there is a special class of faithful structures.

Observation 4.2 Let p be an $(\mathbb{S}, \mathbb{S}')$ -path, where $\mathbb{S}' = \langle \mathbb{K}', \mathbb{D}', \mathbb{P}' \rangle$. If \mathfrak{M} is an \mathcal{L} -structure such that $\mathfrak{M} \models \mathbb{K}'$, then \mathfrak{M} is p -faithful.

Proof. This follows from Proposition 4.2 and the first clause of Proposition 4.1. \square

Hence, we can now prove that our rules, not only guide our agent to sound reasoning, but also to correct action.

Theorem 1 Let $\mathbb{S} = \langle \mathbb{K}, \{\text{Whenever}(s, \alpha)\} \cup \mathbb{D}, \mathbb{P} \rangle$ ($\mathbb{S} = \langle \mathbb{K}, \{\text{When}(s, \alpha)\} \cup \mathbb{D}, \mathbb{P} \rangle$) be an agent state to which the first (respectively, second) directive rule is applicable. Then there is a state \mathbb{S}' such that $\mathbb{S} \longrightarrow^+ \mathbb{S}'$ and, for every structure \mathfrak{M} , if $\mathfrak{M} \models \mathbb{K}'$, then $\mathfrak{M} \models \text{Occurs}(\text{Does}(\mathbb{I}, \alpha), *NOW, *NOW)$.

Proof. The proof starts by noting that \mathfrak{M} is a model of \mathbb{K} (Proposition 4.1) and is faithful to any $(\mathbb{S}, \mathbb{S}')$ -path (Observation 4.2). We proceed by induction on the structure of α , given the rules of Figure 1 and the semantics of composite actions (Section 4.2). \square

5 EVALUATION

To evaluate the proposed treatment of “now”, we rate it against the four criteria of Section 3.2.

N1. An account of post-it-note “now” is embodied in the directive rules of Figure 1. Tense-marker “now” is accounted for since the present is always distinguished as being denoted by $*NOW$. The rules of Figure 1 provide the link between tense-marker “now” and action.

N2. Temporal progression is accounted for by the initiation rule.

N3. No special heavy computation is mandated by the account of “now”; $*NOW$ changes seamlessly at the PML, leaving the knowledge base intact as time goes by.

N4. No special assumptions about the ontology nor about the agent are made by the proposed treatment. In particular, unlike active logic [6], no assumptions about the structure of time are made; and, unlike Lespérance and Levesque’s treatment, no awkward account of equality nor strange belief updates are required.

To get a feel of the system in action, we consider the case of the messy shopper from Section 1. Consider an agent state \mathbb{S} satisfying the following.

- $\text{Name}(\mathbb{I}, \text{Perry}) \in \mathbb{K} = \text{Cn}(\mathbb{K})$.
- $\text{Holds}(\text{Clk}(12 : 00), t_1) \in \mathbb{K}$.
- $\text{Holds}(\text{Messy}(\mathbb{I}), t_1) \notin \mathbb{K}$.
- $\text{Whenever}(\text{Messy}(\mathbb{I}), \text{FixIt}) \in \mathbb{D}$.
- $\Sigma = \langle \rangle$.
- $*NOW = t_1$.

In this state, the first directive rule is not applicable since the agent does not believe that he is now messy. Now, consider another state \mathbb{S}' which is identical to \mathbb{S} except that

- $\mathbb{K}' = \mathbb{K} \cup \{\text{Holds}(\text{Messy}(c1), t_2), \text{Author}(c1, \text{PEI}), t_1 \prec t_2, \text{Holds}(\text{Clk}(12 : 01), t_2)\}$.

- $*NOW' = t_2$.

Here $Author(c_1, PEI)$ indicates that $\llbracket c_1 \rrbracket$ is the author of ‘the problem of the essential indexical’. Again, the directive rule is not applicable since the agent does not believe that *he* is now messy. This may be fixed if the agent comes to believe that *he* is the indicated author. Assuming that the messiness of the author persists, the agent reaches a state S'' which is identical to S' except that

- $\mathbb{K}'' = \mathbb{K}' \cup \{I = c_1, \text{Holds}(Messy(c_1), t_3), \text{Holds}(Messy(I), t_3), t_2 \prec t_3, \text{Holds}(\text{Clk}(12 : 02), t_3)\}$.
- $*NOW'' = t_3$

In this state, the first directive is applicable, resulting in the agent’s fixing the mess he is causing. Note that in all cases, knowledge of clock time is totally irrelevant to action. It would have been, however, had the agent adopted the directive that it should fix the mess anyway once it becomes 12:05, for instance.

6 CONCLUSION

Though indexicality is indeed essential for rational action, a language of thought with indexical expressions is not. The indexical effect may be achieved through rules of practical reasoning. I have outlined a treatment of indexicality within the framework of a grounded layered agent architecture. The top layer comprises a classical non-indexical language of thought. Indexicality features in the interaction between the top layer and a lower perceptuo-motor layer which grounds action terms and the feel for temporal progression. The proposed treatment appears to be adequate at least as far as it seamlessly provides a motivated account for temporal progression and for the functions of “now” as a tense marker and a placeholder for future times in mental notes of action.

REFERENCES

- [1] Alessandro Artale and Enrico Franconi, ‘A temporal description logic for reasoning about actions and plans’, *Journal of Artificial Intelligence Research*, **9**, 463–506, (1998).
- [2] Brandon Bennett and Antony Galton, ‘A unifying semantics for time and events’, *Artificial Intelligence*, **153**(1–2), 13–48, (2004).
- [3] David Braun, ‘Demonstratives and their linguistic meaning’, *Noûs*, **30**, 145–173, (1996).
- [4] Jürgen Dix, Sarit Kraus, and V.S. Subrahmanian, ‘Temporal agent programs’, *Artificial Intelligence*, **127**(1), 87–135, (2001).
- [5] Dylan Dodd and Paula Sweeney, ‘Indexicals and utterance production’, *Philosophical Studies*, **150**, 331–348, (2010).
- [6] Jennifer J. Elgot-Drapkin and Donald Perlis, ‘Reasoning situated in time I: Basic concepts’, *Journal of Experimental and Theoretical Artificial Intelligence*, **2**, 75–98, (1990).
- [7] Henry Hexmoor and Stuart C. Shapiro, ‘Integrating skill and knowledge in expert agents’, in *Expertise in Context*, eds., P. J. Feltovich, K. M. Ford, and R. R. Hoffman, 383–404, AAAI Press/MIT Press, Menlo Park, CA / Cambridge, MA, (1997).
- [8] Haythem O. Ismail, *Reasoning and Acting in Time*, Ph.D. dissertation, University at Buffalo, The State University of New York, 2001.
- [9] Haythem O. Ismail, ‘On the syntax and semantics of effect axioms’, in *Formal Ontology in Information Systems: Proceedings of the Fifth International Conference (FOIS-08)*, pp. 223–236. IOS Press, (2008).
- [10] Haythem O. Ismail and Nasr S. Kasrin, ‘High-level perception as focused belief revision’, in *Proceedings of the 19th European Conference on Artificial Intelligence (ECAI 2010)*. IOS Press, (2010).
- [11] Haythem O. Ismail and Stuart C. Shapiro, ‘Two problems with reasoning and acting in time’, in *Principles of Knowledge Representation and Reasoning: Proceedings of the Seventh International Conference (KR 2000)*, eds., Anthony Cohn, Fausto Giunchiglia, and Bart Selman, pp. 355–365, San Francisco, CA, (2000). Morgan Kaufmann.
- [12] Hans Kamp, ‘Formal properties of ‘now’’, *Theoria*, **37**, 227–273, (1971).
- [13] David Kaplan, ‘On the logic of demonstratives’, *Journal of Philosophical Logic*, **8**, 81–98, (1979).
- [14] Deepak Kumar and Stuart C. Shapiro, ‘Acting in service of inference (and vice versa)’, in *Proceedings of the Seventh Florida Artificial Intelligence Research Symposium*, ed., Douglas D. Dankel, II, pp. 207–211, St. Petersburg, FL, (May 1994). The Florida AI Research Society.
- [15] Yves Lespérance and Hector Levesque, ‘An argument for indexical representations in temporal reasoning’, in *Proceedings of the Tenth Biennial Conference of the Canadian Society for Computational Studies of Intelligence*, ed., Renee Elio, pp. 271–277, Banff, Canada, (1994).
- [16] Yves Lespérance and Hector Levesque, ‘Indexical knowledge and robot action— a logical account’, *Artificial Intelligence*, **73**(1–2), 69–115, (1995).
- [17] Ruth Garrett Millikan, ‘The myth of the essential indexical’, *Noûs*, **24**(5), 723–734, (1990).
- [18] Ruth Garrett Millikan, ‘Pushmi-pullyu representations’, *Philosophical Perspectives*, **9**, 185–200, (1995).
- [19] Madhura Nirkhe, Sarit Kraus, Michael Miller, and Donald Perlis, ‘How to (plan to) meet a deadline between now and then’, *Journal of Logic and Computation*, **7**(1), 109–156, (1997).
- [20] Geoffrey Nunberg, ‘Indexicality and deixis’, *Linguistics and Philosophy*, **16**(1), 1–43, (1993).
- [21] John Perry, ‘The problem of the essential indexical’, *Noûs*, **13**(1), 3–21, (1979).
- [22] Stefano Predelli, ‘I Am Not Here Now’, *Analysis*, **58**(2), 107–115, (1998).
- [23] Arthur Prior, *Past, Present and Future*, Clarendon Press, Oxford, 1967.
- [24] Ray Reiter, ‘The frame problem in the situation calculus: A simple solution (sometimes) and a completeness result for goal regression’, in *Artificial Intelligence and Mathematical Theory of Computation: Papers in Honor of John McCarthy*, ed., Vladimir Lifschitz, 359–380, Academic Press, San Diego, CA, (1991).
- [25] A. Schmiedel, ‘A temporal terminological logic’, in *Proceedings of the 9th National Conference on Artificial Intelligence*, pp. 640–645, San Mateo, CA, (1990). Morgan Kaufmann.
- [26] Stuart C. Shapiro and Haythem O. Ismail, ‘Anchoring in a grounded layered architecture with integrated reasoning’, *Robotics and Autonomous Systems*, **43**(2–3), 97–108, (2003).
- [27] Stuart C. Shapiro and William J. Rapaport, ‘SNePS considered as a fully intensional propositional semantic network’, in *The Knowledge Frontier*, eds., N. Cercone and G. McCalla, 263–315, Springer-Verlag, New York, (1987).
- [28] Stuart C. Shapiro and William J. Rapaport, ‘The SNePS family’, *Computers and mathematics with applications*, **23**(2–5), 243–275, (1992).
- [29] Stuart C. Shapiro and the SNePS Implementation Group, *SNePS 2.7.1 User’s Manual*, Department of Computer Science and Engineering, State University of New York at Buffalo, 2010.
- [30] Alan Sidelle, ‘The answering machine paradox’, *Canadian Journal of Philosophy*, **21**, 525–539, (1991).
- [31] Evan C. Tiffany, ‘What is essential about indexicals?’, *Philosophical Studies*, **100**(1), 35–50, (2000).

Knots World: an investigation of actions, change and space

Paulo E. Santos¹ and Pedro Cabalar²

Abstract.

This work investigates, from a Knowledge Representation perspective, the spatial knowledge of a domain composed of non-trivial objects such as *strings* and *holed objects*. To this aim, we consider the formalisation of puzzle-like examples as the starting point for the development of Knowledge Representation systems, since these domains offer a small number of objects while keeping enough complexity for a challenging reasoning problem. The present paper concentrates on the definition of “loops” as loops (or loop-like regions) that can be formed by a flexible string. We focus on identifying the representational problems introduced by this feature, and sketch potential solutions for future development. For instance, we start with a discussion about the *sorites* vagueness that arises when deciding at which point a loop begins forming a hole, proposing next a possible solution based on Supervaluation Semantics. This discussion is followed by considerations over the formalisation of Reidemeister moves, in order to use them as actions in a domain containing a string. Finally, we discuss a tentative solution of a spatial puzzle, whose solution is dependent on the execution of actions over loops built with a string.

1 Introduction

Understanding the reasoning processes involved in spatial knowledge is one of the key issues in the investigation of cognition, as space not only shapes our actions in the commonsense world, but also serves as the scenario in which our everyday experiences take place. Research in *Qualitative Spatial Reasoning* (QSR) [28, 14] attempts the logical formalisation of spatial knowledge based on primitive relations defined over elementary spatial entities. For instance, QSR theories include a mereotopological theory based on the connectivity between spatial regions [17], the definition of occlusion and parallax [19, 18], spatial vagueness [7, 10], the abductive assimilation of sensor data [23, 24, 22], as well as the definition of qualitative theories about distance [12, 11], boundaries [15], shapes [27, 6] and so forth [8].

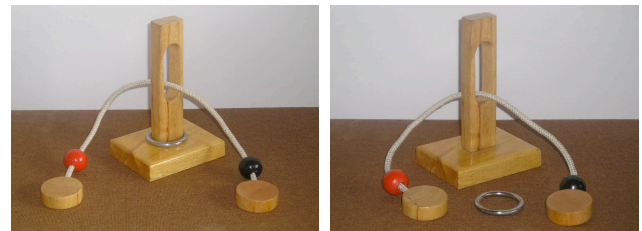
This work investigates, from a QSR perspective, the spatial knowledge of a domain composed of non-trivial objects such as a *string* and *holed objects*. To this aim, we follow the ideas presented in [2] where the formalisation of puzzle-like examples is assumed as the starting point for the development of Knowledge Representation (KR) systems, since these domains offer a small number of objects while keeping enough complexity for a challenging problem from the KR point of view. The present paper concentrates on the definition of

“loops” as loops that can occur in a flexible string. This idea was intentionally left aside in the earlier representations of a family of puzzles [3, 4, 26], but is an essential elaboration that has to be taken into account for the solution of a larger class of problems.

In this paper, we outline and identify the main representational problems derived from the introduction of loops, and sketch some possible formal solutions whose complete development is still under study.

2 Previous Work

Previous work [3, 4, 26] has been concentrated on the formalisation and automated solution of the Fisherman’s Folly puzzle (shown in Figure 1), whose goal is to release a ring from an entanglement of objects (maintaining the object’s physical integrity). The elements of the Fisherman’s Folly puzzle are a holed post (*Post*) fixed to a wooden base (*Base*), a string (*Str*), a ring (*Ring*), a pair of spheres (*Sphere1*, *Sphere2*) and a pair of disks (*Disk1*, *Disk2*). The spheres can be moved along the string, whereas the disks are fixed at each string endpoint. The string passes through the post’s hole in a way that one sphere and one disk remain on each side of the post. It is worth pointing out that the spheres are larger than the post’s hole, therefore the string cannot be separated from the post without cutting either the post, or the string, or destroying one of the spheres. The disks and the ring, in contrast, can pass through the post’s hole.



(a) Initial

(b) Goal: the ring must be free

Figure 1. A spatial puzzle: the Fisherman’s Folly.

In the initial state (shown in Figure 1(a)) the post is in the middle of the ring, which in its turn is supported on the post’s base. The goal of this puzzle is to find a sequence of (non-destructive) transformations that, when applied on the domain objects, frees the ring from the other objects, regardless their final configuration. Figure 1(b) shows one possible goal state.

¹ Universidade da FEI, São Paulo, Brazil. E-mail: psantos@fei.edu.br

² Department of Computer Science, University of Corunna, Spain. E-mail: cabalar@udc.es

A crucial observation is that the puzzle actually deals with four holes: the post hole, the ring hole and the two sphere holes. Note that in a natural language description we would probably identify holes with their host objects, saying that “the string passes through the sphere” (hole) or that “the post passes through the ring” (hole). Furthermore, we would talk about “sliding the ring up the post,” rather than “moving the post down through the ring hole”. Therefore, in the formalisation presented below, “holes” will be identified with their host objects.

A simple planning system capable of finding a solution to the Fisherman’s Folly puzzle was presented in [3], where the states of the puzzle were represented as lists containing the sections of a long object between hole crossings. Based on the hole ontology from [5], in [26], a mereotopological representation of the domain objects was presented. The work in [4] developed a representation of the puzzle actions in a Situation Calculus [21] framework developed in Quantified Equilibrium Logic [16], where we were interested in a solution that was tolerant to elaborations, showing its applicability to other similar puzzles. In none of the previous formalisations, however, the notion of loop was taken into account. This is due to the fact that the object “loop” was not relevant to the solution of the Fisherman’s Folly puzzle or its close relatives. The question that naturally follows from this is whether these previous solutions are tolerant to this elaboration. In other words, would the representation and reasoning system defined in [4] support a puzzle whose solution depends on the manipulation of loops?

The present paper discusses this issue and presents a tentative formalisation of a spatial puzzle whose solution requires the manipulation of loops: the “easy-does-it” puzzle, shown in Figure 2.



Figure 2. Easy does it

The remainder of this paper is organised as follows: in the next section a discussion of the vagueness inherent to the concept of “string loop” is introduced, along with a possible solution to it by means of supervaluation semantics [25, 2]. Section 4 describes the main data structure we use to represent these spatial puzzles (the *chain* structure) and presents the formal solution to the Fisherman’s Folly puzzle [3, 4, 26] as an example. A possible representation of knots is described in Section 5 extending the previous definition of chains. Section 6 describes a tentative formal representation of the solution for a spatial puzzle that has “string loops” as domain objects. Finally, Section 7 concludes this paper.

3 Loops and strings dichotomy

The definition of a “string loop” suffers from the sorites vagueness. I.e., there is an indeterminacy surrounding the limits of applicability of the concept “loop” [13], as depicted in Figure 3. In this paper we

use the supervaluation semantics to solve this issue, following the guidelines presented in [25, 2], as summarised below.

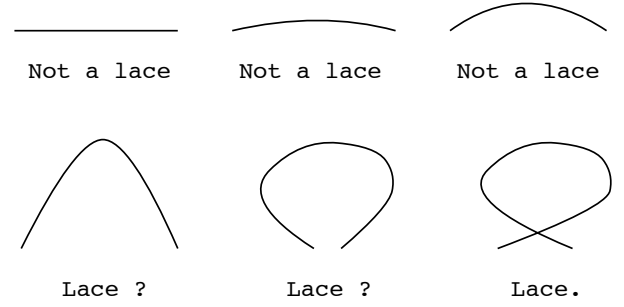


Figure 3. When a string makes a loop (hole).

3.1 Supervaluation semantics

A formal definition of a string loop depends on a particular standpoint. In this work each standpoint is formalised as a particular set of values underlying the formal definitions. In fact, distinct sets of values for the thresholds may result in distinct (sometimes incompatible) definitions of the context objects. We assume the supervaluation semantics [9] to provide a formal treatment for standpoints on feature definition.

Supervaluation semantics views a vague language as a set of distinct precise versions of itself. Each of these versions is called a *precisification* of the language. Formally, each precisification p is identified with a particular interpretation I_p of the language. A *supervaluation model* is defined as a set of precisifications. Therefore, given a supervaluation model Υ we can talk about propositions that are unequivocally true (i.e. are true in *every* interpretation $I_p \in \Upsilon$) and propositions that are *in some sense* true (i.e. are true in *some* interpretation $I_p \in \Upsilon$).

In this work the definition of loop may vary according to the multiple meanings that this concept might assume, or multiple values that the thresholds can assume. For instance, the intermediate case in Figure 3 could (or could not) be considered as either a loop or not a loop in the context of two distinct precisifications. In fact, the apparatus of supervaluation semantics allows for logical relationships between vague concepts to be represented by quantifying over the (possibly infinite) space of precisifications [2].

Using supervaluation semantics we can solve the sorites paradox related to the definition of a loop by assuming a *degree of crossing*, which is a scale running from the *zero crossing* (i.e. a straight string) to a *one crossing* (i.e. a fully crossed string). This scale is shown in Figure 4.

3.2 Viewpoint dependency

Another issue related to the ontology of loops and strings is that the definition of a loop may be viewpoint dependent. Take for instance a coil, if it is observed at an orthogonal angle from its main axis, it does not form a loop, but it is perceived as a hole (or loop) if the observation is made on its main axis. Categorically there is no hole in a coil, as the curve that defines it is open. However, we can use a coil as a hole and apply actions as if it were in fact a closed curve.

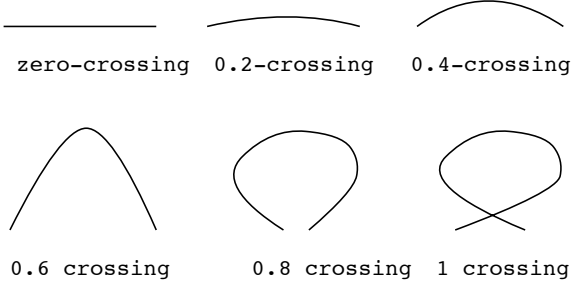


Figure 4. When a string makes a loop (hole).

The solution to the viewpoint dependency in the definition of a loop can also take advantage of the *degree of crossing* idea, since a distinct degree can be assigned to a loop according to the viewpoint where it is being observed. A close curve can, then, be defined as a particular loop observed as a 1-degree loop from every viewpoint.

Besides, the same loop may be a hole or not depending on other factors like the flexibility/rigidity of the long object forming the loop, the objects involved in the actions, etc. For instance, take the loop with 0.8-crossing in Figure 4 and suppose that the string is actually rigid. Then if we have to manipulate a ball with a diameter greater than the open part of the loop, the loop can just be seen as a hole for all purposes. The same happens if the string is flexible but, for instance, its both ends are respectively stuck to a pair of fixed points.

As a first approach to the loop-problem, in this paper we adopt a conservative standpoint by assuming that only closed curves are holes. In this case, for instance, only the last curve in Figure 3 will be considered as a loop in the present work. This also implies that we are only going to consider closed curves (i.e. loops that are independent from the viewpoint) as loops. However, we expect that the further inclusion of such vague definitions during the problem solving procedure will allow for more flexible inferences. This would facilitate a type of “dual” reasoning, whereby in a *string loop* may be considered either as a loop or not in the same proof step. Developing this idea further, however, is a task for future investigations.

4 Spatial puzzles as chains

The simple solution for the Fisherman’s Folly puzzle presented in [3] relies on distinguishing the puzzle’s objects into three sorts: *holes* (which includes the post hole, the ring hole and the holes through the spheres), long objects (that includes the string and the post), and regular objects (including all the remainder objects). For each hole h , its faces are distinguished: h^- and h^+ ; and for each long object l its tips l^- and l^+ are defined.

For helping the reader to figure out a puzzle state, we use schematic representations like the one in Figure 5, which shows the initial state. Arrows correspond to segments of long objects, defined between pairs of hole crossings, or between a hole crossing and a tip. These arrows point in the direction from tip l^- to tip l^+ of a same long object l . Ellipses represent holes and boxes are linked regular objects. The positive face of a hole implicitly corresponds to the “visible” side of the ellipse.

Central to this simple solution is the definition of a list data struc-

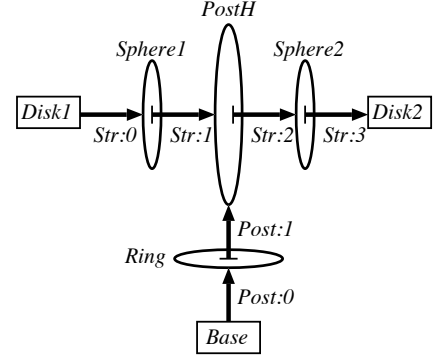


Figure 5. Schematic representation of the initial state.

ture named $chain(X)$. This data structure represents the sequence of all hole crossings on a long object X , when traversing X from its negative tip to its positive one. For instance, the state shown in Figure 5 is represented by the following two chains: $chain(Post) = [Ring^+]$ and $chain(Str) = [Sphere1^+, PostH^+, Sphere2^+]$. The former represents that the long object $Post$ crosses the ring hole and the latter states that the string crosses the hole on the sphere 1, the post hole and the hole on the sphere 2, respectively. Note that, for brevity, only the outgoing hole faces are shown, following the direction negative to positive tip.

An action *pass* was defined to represent the movements of puzzle objects. The effects of *pass* either add or delete a hole crossing from the *chain* on which it is applied.

Using these definitions, a solution to the Fisherman’s Folly puzzle can be represented by the sequence of chains shown on Figure 6, whereby each state is identified by its sequence number plus the pair of lists $chain(Post)$ and $chain(Str)$ in this order. Note that State 5 has actually reached the goal since, at this point, the ring hole *Ring* does not occur in any list, i.e., it is not crossed by any long object.

state	$chain(Post)$	$chain(Str)$
s_0	$[Ring^+]$	$[Sphere1^+, PostH^+, Sphere2^+]$
s_1	$[Ring^+]$	$[Sphere1^+, PostH^+, Sphere2^+, PostH^-]$
s_2	$[\]$	$[Sphere1^+, Ring^-, PostH^+, Ring^+, Ring^-, Sphere2^+, Ring^+, PostH^-]$
s_3	$[\]$	$[Sphere1^+, Ring^-, PostH^+, Sphere2^+, PostH^-, Ring^+]$
s_4	$[\]$	$[Sphere1^+, PostH^+, Ring^-, Sphere2^+, Ring^+, PostH^-]$
s_5	$[\]$	$[Sphere1^+, PostH^+, Sphere2^+, PostH^-]$

Figure 6. A formal solution for the Fisherman’s puzzle and its graphical representation.

An analogous representation will be used in this paper to formalise the solution steps of the Easy-does-it puzzle at a similar level of abstraction³. Next section introduces some concepts in knot theory that

³ We are also exploring the possibility of directly using Reidemeister’s moves from knot-theory, as they provide a more fundamental and fine-grained description of the movements. However, the result involves many more steps and a considerably higher search space for planning.

are relevant to the development of this work.

5 String crossings and knots

Similarly to the idea of chain described above, [29] proposes a representation of the various shapes a string assumes by collecting the points where the string crosses itself, scanning it from one of its terminals to the other. This representation, called *p-data projection* (Figure 7), is based on a 2D projection of a 3D knot and facilitates an algebraic treatment of Reidemeister moves.

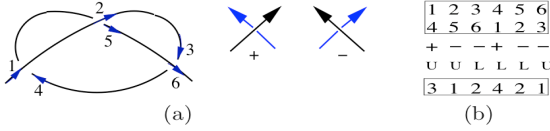


Figure 7. A knot and its p-data representation.

The p-data representation of a string is constructed as follows. We first consider the direction of the string as the direction in which it is being swept. From one terminal of the string to the other, each point where the string crosses itself receives an ID number (starting from the number 1). Each ID appears twice in the p-data structure, since a crossing is noted two times when scanning a string. This amounts to the first two lines of the structure shown in Figure 7. At each crossing we have to annotate also a *sign* of the crossing and whether the string crosses over or under itself (respectively shown in the third and fourth lines in Figure 7). The latter is called the vertical position of the crossing and is determined by verifying the position of the string at each crossing found. The former can be obtained from the sign of the expression: $(\vec{l}_{upper} \times \vec{l}_{lower}) \cdot \vec{e}_z$, where \vec{l}_{upper} and \vec{l}_{lower} are, respectively, the directions of the upper and lower parts of the string at the crossing and \vec{e}_z is the normal vector of the projection plane. The sign and the vertical position of each crossing is summarised in the fifth line shown in Figure 7, where the symbol 1 represents an upper crossing whose sign is $-$; 2 represents a lower $-$; 3 an upper $+$; and 4 a lower $+$ crossing.

The p-data representation captures the essential characteristics of crossings of a string on itself. This representation, however, assumes that there are only single crossings of the string; i.e., the string cannot cross itself more than one time at each point.

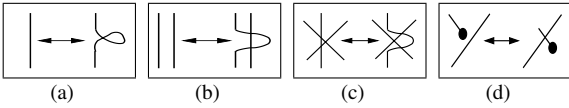


Figure 8. The Reidemeister moves and the cross move.

Traditionally, knot theory assumes three basic actions on knots, called Reidemeister moves [20]. Figure 8 shows the three Reidemeister moves (Figures 8(a), 8(b) and 8(c)) and the cross move (Figure 8(d)), introduced in [29] for handling open strings. These actions that can be described as follows: Reidemeister move I (Figure 8(a)) adds or deletes a simple twist in the string; Reidemeister move II (Figure 8(b)) allows the inclusion (or exclusion) of two crossings in

the string; Reidemeister move III (Figure 8(c)) slides a strand of the string from one side of a crossing to the other; the cross move (Figure 8(d)) is defined on simple open curves and adds or removes a string crossing by sliding an open end of it over a continuous part of the string.

Using the chain notation, summed up by the *upper/lower* description of string crossings introduced within the p-data structure, the Reidemeister moves (Figure 8(a)) can be described as follows, assuming *U* and *L* as representing the *Upper* and *Lower* crossings, as in the p-data structure described above.

- Move 1 (Figure 8(a)): change the chain from state \square to the state $[Cross^U, Cross^L]$;
- Move 2 (Figure 8(b)): there are two strings (S_1 and S_2) and two crossings 1 and 2:
 - initial state: $chain(S_1) = \square$ and $chain(S_2) = \square$
 - state after move: $chain(S_1) = [C_{1,S_2}^U, C_{2,S_2}^U]$ and $chain(S_2) = [C_{1,S_1}^L, C_{2,S_2}^L]$
- Move 3 (Figure 8(c)): there are three strings (S_1 , S_2 and S_3) and two pairs of crossings (1 and 2) and (2 and 3):
 - initial state: $chain(S_1) = [C_{1,S_2}^U, C_{2,S_3}^U]$, $chain(S_2) = [C_{1,S_1}^L, C_{3,S_3}^L]$ and $chain(S_3) = [C_{2,S_1}^U, C_{3,S_2}^L]$
 - state after move: $chain(S_1) = [C_{2,S_3}^L, C_{4,S_1}^L]$, $chain(S_2) = [C_{3,S_3}^L, C_{3,S_1}^L]$ and $chain(S_3) = [C_{4,S_1}^U, C_{3,S_2}^L]$
- Cross move (Figure 8(d)): in this move there are two distinct strings S_1 and S_2 , and the result of the action is as follows:
 - initial state: $chain(S_1) = \square$ and $chain(S_2) = \square$;
 - state after move: $chain(S_1) = [C_{1,S_2}^L]$, $chain(S_2) = [C_{1,S_1}^U]$.

With the chain notation and the Reidemeister moves we can describe a tentative solution to Easy-does-it puzzle, as presented in the next section.

6 Easy-does-it puzzle



Figure 9. Easy does it

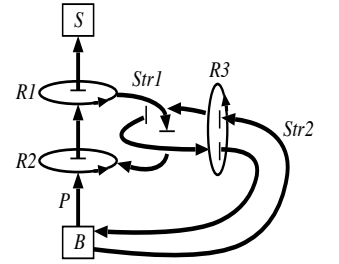


Figure 10. Diagram

Figure 9 shows one possible initial state of the Easy-does-it puzzle. This domain is composed of three rings: two at the tips of a string, and crossing the post (denoted by R_1 and R_2) and a third ring (R_3), which we call *main* ring (since this is the one that should be released from the system of objects). R_1 and R_2 are “locked in” the post, since they cannot pass through the sphere fixed at the post’s tip. The domain also has two strings Str_1 and Str_2 that are entangled with each other, a post P , a base B and six holes: the main ring hole (

R_3), the holes on the two rings attached to String 1 (R_1 and R_2), the hole (loop) made by string 1 (Str_1), the loop at the tip of string 2 (Str_2 , and the loop defined by the set of objects [string 1, string 2, base, ring 2 and post], which we call CH , “combined hole”. This domain also has three chains: $chain(P)$, for the post; $chain(Str_1)$, for string 1 (the one with the two rings attached); and, $chain(Str_2)$ that represents the long string.

The initial state shown in Figure 9 can be depicted using diagrams as shown in Figure 10.

In this work loops (loops) in the string form holes that are represented with two faces + and -, as done previously for holes in rigid objects [3, 4, 26]. The only distinction here is that loops can appear and disappear and they can change their sizes. When it is convenient, we are going to denote loops on a particular string with the function $l(s, n)$, where s is a string and n represents the ordinal (beginning with 0) of the first segment of s involved in the loop, scanning the string from its negative to its positive tips (note that many loops can be done/undone on a single string). For now we are not taking into account the origin of loops, only that they define an empty space (bounded by a string) through which an object can pass.

6.1 Loop detection

Chains of crossings reveal the set of loops formed by a string. Whenever we repeat a link to a same object or a crossing through a same object⁴ (regardless the possible crossing direction), we form a loop in a string. In this way, for instance, in the chain for the string 2 in Figure 10, two loops can be detected:

$$chain(str_2) = \underbrace{[B, R_3^+, R_3^-, B]}_{l(S,0)} \underbrace{[R_1^+, R_1^-, R_2^+, R_2^-]}_{l(S,1)}$$

and the same happens in, say, the more complex chain example:

$$chain(S) = \underbrace{[B, R_1^+, R_2^+, R_3^-, R_4^+, R_2^-]}_{l(S,0)} \underbrace{[R_1^-, R_2^+, R_3^-, R_4^+, R_2^-]}_{l(S,2)}$$

As said before, we denote the loops using the position in $chain(S)$ (the first position is 0) of the origin of the first segment forming the loop. In this case, we have a loop formed by the pair of links to object B , that close both tips of the string. This loop is denoted by $l(S, 0)$ because the first B is at position 0. There is no ambiguity, since the closing part of the loop will be the next occurrence of B in the list, from left to right (in this case, position 6, which is the last one). The second loop is formed by the two crossings of ring R_2 , in this case, in opposite directions. This loop is denoted $l(S, 2)$ pointing out that the origin of the loop is at position 2 (crossing R_2^+). We can deduce that the loop's end will be the next position to the right in which we find a crossing through object R_2 – in the example, position 5.

The same object can form several loops in the string. For instance:

$$chain(S) = \underbrace{[R_1^+, R_2^-, R_3^+, R_1^-]}_{l(S,0)} \underbrace{[R_4^-, R_1^+]}_{l(S,3)}$$

⁴ If we represent the string tips S^- , S^+ at both ends of the chain, these are the only exceptions. However, if we make them to coincide in the space, that is, we add the assumption that $S^- = S^+$ represent the same object, then they form again a loop.

We can use Allen's interval algebra [1] to classify the possible relations about a pair of loops x and y in a same string. However, not all relations in Allen's algebra are possible here, since two different loops cannot share the same left (resp. right) end. As a result, relations “starts,” “finishes” and their inverses are not allowed. This leaves the remaining 9 possibilities: “before,” “meets,” “overlaps,” and “during,” together with their respective inverses plus relation “equal.”

The next section presents a chain description of one possible solution for the Easy-does-it puzzle.

6.2 Solution Steps

The solution presented in this section is one possible solution to the puzzle considered. We do not claim that this is the optimal solution, or that the formalisation below solves all the representational issues related to the Easy-does-it, but that it is an initial formal description on top of which interesting points can be discussed regarding the representation and reasoning about strings and loops. More importantly, the sequence of states in terms of chains of crossings may help us to explore the general effect of the actions involved in terms of loop creation and removal, something we only outline here, leaving its complete formalisation for future study.

Initial state

The chain description of Easy-does-it initial state, shown in Figure 10, is presented as follows:

$$\begin{aligned} chain(P) &= [B, R_2^+, R_1^+, S] \\ chain(Str_1) &= [R_1, l(Str_2, 1)^-, R_2] \\ chain(Str_2) &= [B, R_3^-, l(Str_1, 0)^-, R_3^+, B] \end{aligned}$$

We can recognize two loops in $chain(Str_2)$:

- One beginning in object B at position 0 and returning to B in position 4 in the chain. This loop is formed by the string segment $Str_2 : 0$, the ring R_3 and the string segment $Str_2 : 3$ plus the base B . We denote this loop by $l(Str_2, 0)$ because it begins with (the origin of) segment 0.
- One loop limited by object R_3 , since the chain goes through R_3^- at position 1 and come back through R_3^+ at position 3. This loop is formed by segments $Str_2 : 1$ and $Str_2 : 2$ so we will denote it as $l(Str_2, 1)$.

First transition

The first step of the solution involves passing the segment $Str_2 : 1$ towards R_1^+ (action $pass(Str_2 : 1, R_1^+)$). The result of this action is shown in Figure 11, and the resulting chain description follows.

$$\begin{aligned} chain(P) &= [B, R_2^+, R_1^+, S] \\ chain(Str_1) &= [R_1, l(Str_2, 1)^-, R_2] \\ chain(Str_2) &= [B, R_3^-, R_1^+, R_1^-, R_3^+, B] \end{aligned}$$

In this process we have created a third loop in Str_2 that can be easily seen as the pair $\langle R_1^+, R_1^- \rangle$ in $chain(Str_2)$. This new loop can be denoted as $l(Str_2, 2)$, since it begins with a crossing at position 2 (R_1^+) in $chain(Str_2)$. On the other hand, $l(Str_2, 1)$ is delimited now by R_3 , segment $Str_2 : 1$, R_1 and segment $Str_2 : 3$.

Note that the crossing through $l(Str_1, 0)^-$ has disappeared from $chain(Str_2)$.

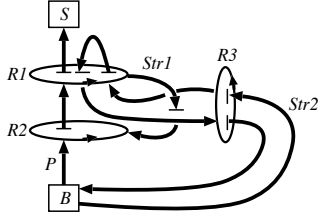


Figure 11. State S_1 .

Second transition

The second action is to pass the sphere S towards $l(Str_2, 2)^+$, resulting in the following state (Figure 12):

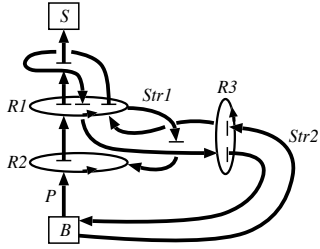


Figure 12. State S_2 .

$$\begin{aligned} \text{chain}(P) &= [B, R_2^+, R_1^+, l(Str_2, 2)^+, S] \\ \text{chain}(Str_1) &= [R_1, l(Str_2, 1)^-, R_2] \\ \text{chain}(Str_2) &= [B, R_3^-, R_1^+, R_1^-, R_3^+, B] \end{aligned}$$

Third transition

Then the segment $Str_2 : 2$ should be passed towards R_1^- (Figure 13):

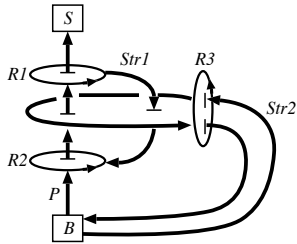


Figure 13. State S_3 .

$$\begin{aligned} \text{chain}(P) &= [B, R_2^+, l(Str_2, 1)^+, R_1^+, S] \\ \text{chain}(Str_1) &= [R_1, l(Str_2, 1)^-, R_2] \\ \text{chain}(Str_2) &= [B, R_3^-, R_3^+, B] \end{aligned}$$

Notice that this movement has collapsed the segments $Str_2 : 1$, $Str_2 : 2$ and $Str_2 : 3$ into a single $Str_2 : 1$. Similarly, loops $l(Str_2, 1)$ and $l(Str_2, 2)$ have become a single loop $l(Str_2, 1)$ again, as we had in the initial state.

Fourth transition

At this point, we could begin passing R_1 down $l(Str_2, 1)^-$. This should leave a loop in Str_1 as depicted in Figure 14.

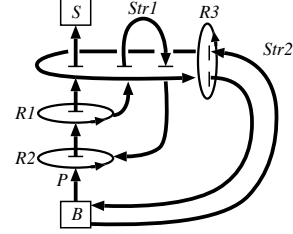


Figure 14. State S_4 .

After that, we would pull from segment $Str_1 : 1$ (the one forming the new loop) down to $l(Str_2, 1)^-$ (Figure 15).

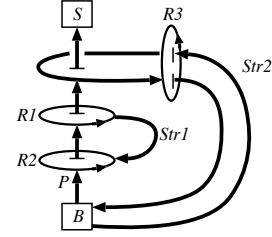


Figure 15. State S_5 .

However, this state (S_5), reached in two steps from S_3 , could be directly reached in one step if we consider a single action $\text{pass}(Str_1 : 0, l(Str_2, 2)^-)$, that is, when being at S_3 , pass the first segment of the string Str_1 down to $l(Str_2, 2)^-$. As ring R_1 is linked to this segment, we get that the ring also crosses through $l(Str_2, 2)^-$. The chain description of Figure 15 is presented as follows:

$$\begin{aligned} \text{chain}(P) &= [B, R_2^+, R_1^+, l(Str_2, 1)^+, S] \\ \text{chain}(Str_1) &= [R_1, R_2] \\ \text{chain}(Str_2) &= [B, R_3^-, R_3^+, B] \end{aligned}$$

Fifth transition

The next action is to move the loop in Str_2 upwards above the sphere. Formally, this means passing the sphere S down to $l(Str_2, 1)^-$, that is $\text{pass}(S, l(Str_2, 1)^-)$, as shown below.

The chain description of Figure 16 is described below.

$$\begin{aligned} \text{chain}(P) &= [B, R_2^+, R_1^+, S] \\ \text{chain}(Str_1) &= [R_1, R_2] \\ \text{chain}(Str_2) &= [B, R_3^-, R_3^+, B] \end{aligned}$$

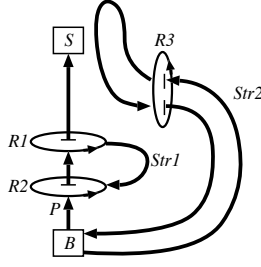


Figure 16. State S_6 .

Sixth transition

Finally, the loop in Str_2 , formed by R_3 , can be removed. Formally, pull segment $Str_2 : 1$ towards R_3^+ , i.e., $pass(Str_2 : 1, R_3^+)$, resulting in the following chain (that describe the state represented in Figure 17).

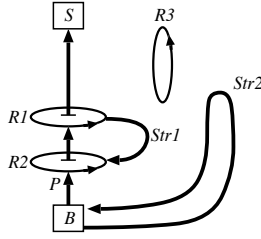


Figure 17. State S_7 .

$$\begin{aligned} chain(P) &= [B, R_2^+, R_1^+, S] \\ chain(Str_1) &= [R_1, R_2] \\ chain(Str_2) &= [B, B] \end{aligned}$$

Note that the main ring R is not present in any of the last chain descriptions. Therefore, R is free from the set of entangled objects, solving the puzzle.

6.3 Loop creation and removal

In transition 1 and 6 (and also 4, if we consider an intermediate step) we have the creation or unwinding of loops. Note that we can undo all the steps in reverse order from the goal situation to the initial one. Each action has an analogous reverse movement. Transition 1 creates a loop when done forward, but we can also consider the loop movement when done backwards. Transition 6 removes the loop when done forward and creates it if done backwards.

Let us start by considering the idea of loop creation. Rather than passing an object to a hole side, we consider an action $pick(x : i, p)$ meaning that we pick some arbitrary point in segment $x : i$ pulling from it towards the hole side p . Figure 18 shows the result of this movement. In principle, this movement can always be executed regardless the origin and target of segment $x : i$. Notice that the action will always create a new loop. Thus, for instance, if in the resulting situation depicted in Figure 18 (shown below) we perform action

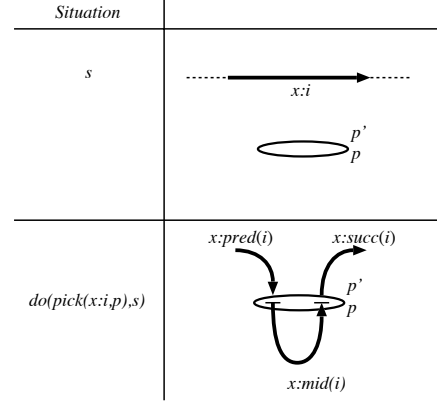


Figure 18. Creating a loop by picking a point inside segment $x : i$.

$pick(x : mid(i), p')$, that is, we pull back to the original hole side p' , then we will not obtain the initial situation s but the one shown in Figure 19 instead.

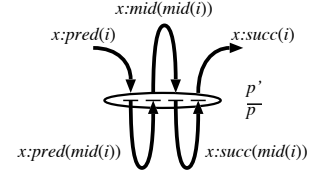


Figure 19. Picking back a point in a loop creates more loops.

The second movement related to loops we shall consider in the future is the action of *removing a loop*. This movement can only be done on a segment that forms a loop.

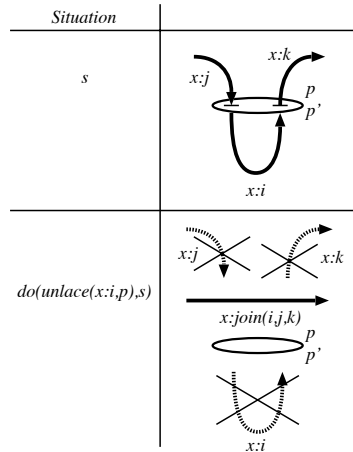


Figure 20. Removing the loop.

7 Conclusion

In this paper we discuss the challenging problem of formally describing a particular characteristic of flexible objects such as strings: their capability of making loops that can be used (and reasoned about) as holes in spatial reasoning processes. We outlined an initial formalisation based on previous investigations on an automated solution for spatial puzzles. There is, however, still a long way to go before deploying this initial formalisation in a real application setting. For instance, a complete description of the set of relevant actions and their effects on the chains of crossings is still under development. We are also studying the correctness of the effects of these actions in terms of Reidemeister's moves. Besides, immediately related to this, we still did not assume the possibility of knots in our domain. Although essential for solving most real world problems (such as tying a shoelace, operating a sailboat or executing sutures), the assumption of knots greatly increases the complexity automated problem solving with strings. We believe that the solution resides on a proper use of heuristics to drive the process of knot tying, but we just ignore how to do it as yet.

Acknowledgements

Paulo E. Santos acknowledges financial support from FAPESP grant 2012/04089-3, and CNPq "bolsa de produtividade em pesquisa" grant 303331/2011-9.

Pedro Cabalar was partially supported by Spanish MEC project TIN2009-14562-C05-04.

REFERENCES

- [1] James F. Allen, 'Maintaining knowledge about temporal intervals', *Communications of the ACM*, **26**(11), 832–843, (November 1983).
- [2] B. Bennett, 'Application of supervaluation semantics to vaguely defined spatial concepts', in *Spatial Information Theory: Foundations of Geographic Information Science; Proceedings of COSIT'01*, ed., D.R. Montello, volume 2205 of *LNCS*, pp. 108–123, Morro Bay, (2001). Springer.
- [3] P. Cabalar and P. Santos, 'Strings and holes: an exercise on spatial reasoning', in *Proc. of SBIA-IBERAMIA*, volume 4140 of *LNAI*, pp. 419–429, (2006).
- [4] Pedro Cabalar and Paulo E. Santos, 'Formalising the fisherman's folly puzzle', *Artif. Intell.*, **175**(1), 346–377, (January 2011).
- [5] R. Casati and A. C. Varzi, *Parts and Places*, MIT press, 1999.
- [6] E. Clementini and P. Di Felice, 'A global framework for qualitative shape description', *GeoInformatica*, **1**(1), 11–27, (1997).
- [7] A. G. Cohn, B. Bennett, J. Gooday, and N.M. Gotts, 'Representing and reasoning with qualitative spatial relations about regions', in *Spatial and Temporal Reasoning*, ed., Oliviero Stock, 97 – 134, Kluwer Academic Publishers, (1997).
- [8] A. G. Cohn and S. M. Hazarika, 'Qualitative spatial representation and reasoning: An overview', *Fundamenta Informaticae*, **46**(1-2), 1–29, (2001).
- [9] K. Fine, 'Vagueness, truth and logic', *Synthese*, **30**, 263–300, (1975).
- [10] H.W. Guesgen, 'From the egg-yolk to the scrambled-egg theory', in *Proc. FLAIRS*, pp. 476–480, (2002).
- [11] H.W. Guesgen, 'Reasoning about distance based on fuzzy sets', *Applied Intelligence (Special Issue on Spatial and Temporal Reasoning)*, (2002).
- [12] D. Hernández, E. Clementini, and P. di Felice, 'Qualitative distances', in *LNAI*, eds., W. Kuhn and A. Frank, 45–58, Springer-Verlag, (1995).
- [13] Dominic Hyde, 'Sorites paradox', in *The Stanford Encyclopedia of Philosophy*, ed., Edward N. Zalta, winter 2011 edn., (2011).
- [14] G. Ligozat, D. Mitra, and J-F Condotta, 'Spatial and temporal reasoning: beyond Allen's calculus', *AI Communications archive*, **17**(4), 223–233, (2004).
- [15] R. C. Meathrel and A. P. Galton, 'A hierarchy of boundary-based shape descriptors', in *Proc. of IJCAI*, pp. 1359–1364, (2001).
- [16] D. Pearce and A. Valverde, 'Quantified equilibrium logic and the first order logic of here-and-there', Technical Report MA-06-02, University of Málaga, Spain, (2006).
- [17] D. Randell, Z. Cui, and A. Cohn, 'A spatial logic based on regions and connection', in *Proc. of KR*, pp. 165–176, Cambridge, U.S., (1992).
- [18] D. Randell and M. Witkowski, 'Building large composition tables via axiomatic theories', in *Proc. of KR*, pp. 26–35, Toulouse, France, (2002).
- [19] D. Randell, M. Witkowski, and M. Shanahan, 'From images to bodies: Modeling and exploiting spatial occlusion and motion parallax', in *Proc. of IJCAI*, pp. 57–63, Seattle, U.S., (2001).
- [20] K. Reidemeister, *Knot Theory*, BCS Associates, 1983.
- [21] R. Reiter, *Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems*, MIT Press, U.S., 2002.
- [22] P. Santos, 'Reasoning about depth and motion from an observer's viewpoint', *Spatial Cognition and Computation*, **7**(2), 133–178, (2007).
- [23] P. Santos and M. Shanahan, 'Hypothesising object relations from image transitions', in *Proc. of ECAI*, ed., Frank van Harmelen, pp. 292–296, Lyon, France, (2002).
- [24] P. Santos and M. Shanahan, 'A logic-based algorithm for image sequence interpretation and anchoring', in *Proc. of IJCAI*, pp. 1408–1410, Acapulco, Mexico, (2003).
- [25] Paulo Santos, Brandon Bennett, and Georgios Sakellariou, 'Supervaluation semantics for an inland water feature ontology', in *Proceedings of the 19th international joint conference on Artificial intelligence, IJCAI'05*, pp. 564–569, San Francisco, CA, USA, (2005). Morgan Kaufmann Publishers Inc.
- [26] Paulo E. Santos and Pedro Cabalar, 'The space within fisherman's folly: Playing with a puzzle in mereotopology', *Spatial Cognition & Computation*, **8**(1-2), 47–64, (2008).
- [27] C. Schlieder, 'Qualitative shape representation', in *Geographic Objects with Indeterminate Boundaries*, eds., P. A. Burrough and A. U. Frank, 123–140, Taylor & Francis Inc., (1996).
- [28] *Spatial and Temporal Reasoning*, ed., Oliviero Stock, Kluwer Academic Publishers, 1997.
- [29] J. Takamatsu, T. Morita, K. Ogawara, H. Kimura, and K. Ikeuchi, 'Representation for knot-tying tasks', *IEEE Transactions on Robotics and Automation*, **22**(1), 65 – 78, (2006).

Reasoning about Relative Relationships in 3D Space for Objects Extracted from Dynamic Image Data

Kazuko TAKAHASHI
Kwansei Gakuin University, Japan,
email: ktaka@kwansei.ac.jp

Abstract. We describe a method of deriving relative relationships in three-dimensional space for objects extracted from video data. The method exploits information about occlusion within a framework of qualitative spatio-temporal reasoning. In particular, we are concerned with correctly deriving movements that are not entirely observable, such as an object moving through an opaque tube. Using a pair of instantaneous static images taken from two different directions, we determine the location in three-dimensional space as far as possible, then use dynamic image data from around that instant to fill in the missing pieces. Moreover, we present an environment that shows qualitative change of relative relationships. This method can be applied to the automatic extraction of events from video data.

Keywords: qualitative spatio-temporal reasoning, RCC, occlusion, viewpoint, environment

1 INTRODUCTION

Recent progress in the performance of computers provides opportunities for handling spatial data such as images or video data. Image and video data are not only frequently uploaded on the Web, but also appear as data captured by robot cameras, and require real-time analysis. We may arrange these in order, provide them with tags or keywords, or predict the behaviors of objects captured in them. In view of these possibilities, an efficient method is required for analyzing these data at an abstract level and recognizing what has happened.

This paper describes a method of deriving relative relationships in three-dimensional (3D) space between objects extracted from video data, within a Qualitative Spatio-Temporal Reasoning framework. Qualitative Spatial Reasoning (QSR) is a method that treats figures or images qualitatively, by extracting the information necessary for a user's purpose [19, 4, 12]. A system that incorporates dynamics is also called Qualitative Spatio-Temporal Reasoning (QSTR). QSTR is related to earlier research on qualitative simulation [8] aiming at the qualitative treatment of physical changes, in the sense that both handle discrete data. Cui et al. applied this idea to spatial reasoning [5]. This research was followed by a number of works on qualitative simulation [3, 10, 20]. However, most of these efforts dealt with two-dimensional (2D) objects. Some researchers have investigated the movement of solid objects that are completely filled, but few studies have focused on 3D space. When we think of objects in 3D space, visibility becomes an inevitable consideration. When an object is in the shadow of another object, it is invisible to an observer. In particular, QSTR frameworks have not discussed or formalized events in which an object passes through the inner part of

another opaque object, which is a characteristic issue in 3D space. In this paper, we discuss the construction of a qualitative 3D model that includes such cases. Several researchers have used the concept of occlusion to represent the degree of visibility [16, 1, 17]. Occlusion and visibility are treated in a 2D plane [7] that is a projection of 3D space. In these works, the main objectives are axiomatization and development of a model of visibility. The authors pay little attention to mechanical reasoning algorithms, and none discuss how an occlusive relation is determined. They assume the transparency of objects; that is, they assume that a hidden part "exists but is invisible," and their reasoning is based on this assumption. However, when one object appears in an image, we do not actually know if another occluded object is also present.

In one well-known technique, a 3D model is constructed by combining multiple image data taken from different viewpoints. While numerical data such as coordinates are used in general image processing, we utilize the relationships between rectangles extracted from video data, which are closures of objects. This is a practical approach, because most image processing tools extract objects in this way. The QSTR approach is also advantageous because of its relatively small demands on memory and workspace.

In general, we cannot determine the relative relationships in 3D space between objects from one instantaneous image, because the image may include a blind spot, depending on the viewpoint and the shape of the object.

As one solution to this problem, we can predict a hidden part from continuous video data taken from a unique viewpoint [6]. For example, consider the image of 3D objects taken from a certain viewpoint shown in Figure 1. The image alone is insufficient for determining whether a part of A is hidden by B, or A is on top of B. However, if the sequence of images shown in Figure 2 is provided, we can reason that only a part of A is shown in Figure 1, and it is highly probable that A moves behind B. In contrast, if the time sequence continues to show the configuration of Figure 1, then we reason that it is highly probable that A is on top of B. However, suppose that B is a hollow tube. In this case, even if a sequence of images is given, we cannot judge whether A moves behind B, or A passes through B, if the video data is taken from a single viewpoint.

Another solution is to project 3D objects onto a 2D plane from a specific viewpoint, and then derive the positional relationship in 3D from multiple projections. Multiple image data are required to eliminate a blind spot, but it is impossible to choose such a viewpoint that completely eliminates the blind spot for some types of objects. For example, assume that a tiny ball is moving around a big ball. In this case, it is impossible to find a stable viewpoint from which the

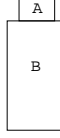


Figure 1. Objects A and B at an instant

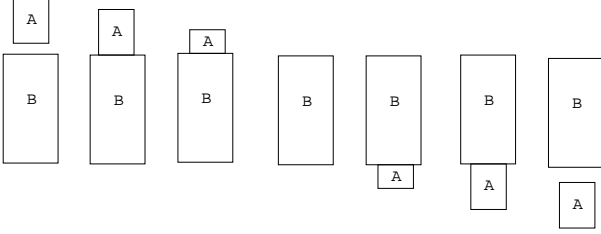


Figure 2. Objects A and B in a time sequence

tiny ball can be observed at any instant. Therefore, to create a 3D model from 2D data, we need a sequence of image data taken from multiple viewpoints. The greater the number of viewpoints is, the more reliable is the reasoning.

In this paper, we describe the reasoning about relative relationships of objects based on data about rectangles extracted from videos taken from two different viewpoints. In particular, for a hollow tubular object, we show a method for identifying an event in which another object passes through it. We also determine whether the tube has a cap from its relative relation with another moving object. Moreover, we demonstrate that we can reason the direction of movement from a sequence of image data taken from a unique viewpoint, and present an envisionment.

This paper is organized as follows. In Section 2, we describe RCC, which is the QSR framework on which our work is based. In Section 3, we discuss our qualitative model. In Section 4, we explain the technique for constructing a 3D model from 2D data. In Section 5, we demonstrate the reasoning of relations in 3D space. In Section 6, we compare our work with related research. Finally, in Section 7, we present our conclusions.

2 Region Connection Calculus

Region Connection Calculus (RCC) is one of the representatives of QSR frameworks [15], lots of QSR systems based on. We also construct a model based on RCC. In RCC, spatial data are represented as relative positional relationships of regions. In RCC8, which is the most popular among several RCC systems, only the connections of regions are considered, and other information is ignored.

Figure 3 shows the eight primitives in RCC8: DC (disconnected), EC (external connection), PO (partial overlap), EQ (equal), TPP (tangential proper part), NTPP (non-tangential proper part); TPPi and NTPPi are inverse relations of TPP and NTPP, respectively. These primitives are jointly exhaustive and pairwise disjoint (JEPD). In this figure, only states connected by an edge can transit directly from one another. For example, if a pair of regions is in the PO relation, then

it does not change to the state DC without passing through the state EC.

RCC itself is free from the concept of visibility.

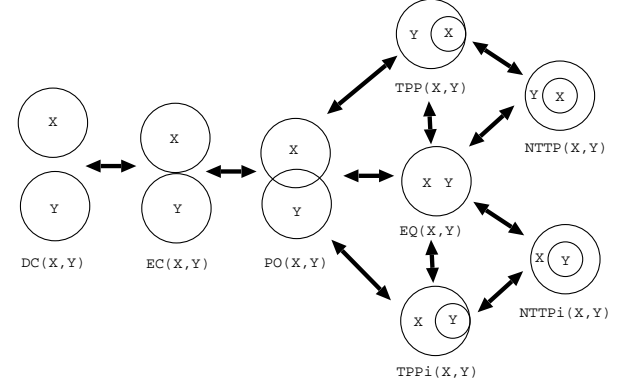


Figure 3. Fundamental relationships of RCC8

3 DESCRIPTION OF THE QUALITATIVE MODEL

3.1 Target objects

Our 3D target objects are classified into two types: *solid objects* and *tube objects*. A solid object is one that is completely filled. A tube object is one that is hollow, allowing another object to pass through or be wholly or a partly contained within it. We assume that an object has no holes and no dents, and that the boundary lines of objects are straight. An object can move, but its inherent shape or size is stable. We also assume that no object splits, is united with another object, is newly created, or becomes extinct. An object that does not move is called a *static object*, and an object that moves is called a *dynamic object*.

The minimum convex polyhedron containing an object is called *the closure* of the object. The closure contains the object itself as its boundary. If an object is twisted or is a tube, then the object and its closure do not coincide. For such an object, we regard the closure as a region occupied by the object. Therefore, when an object X is entirely inside a tube object Y, their RCC8 relationship is NTPP(X,Y) or TPP(X,Y) (Figure 4).

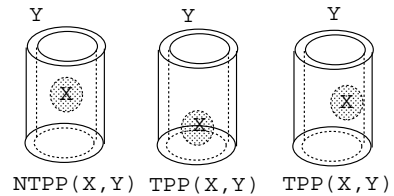


Figure 4. TPP and NTPP relations w.r.t. a tube object

We introduce the concepts of body and image for an object. Given an object x , $bd(x, t)$ denotes its body at an instant t , and refers to

the region x actually occupies in 3D space at that instant, while $im(x, v, t)$ denotes its image from a viewpoint v at an instant t , and refers to its projection in the designated direction.

We assume that each $im(x, v, t)$ is a rectangle, because the object data extracted from the image data is rectangular in shape. When the images of two objects are externally connected by a point or a line, their relation is EC in both cases, and when one is a tangential proper part via a point or a line, their relation is TPP in both cases. Moreover, when an object x is partially hidden by another object y , and its visible part in the image is concave, the relation of $im(x, v, t)$ and $im(y, v, t)$ is not EC, but PO. This fact results in the indeterminacy of the question of which object is in the foreground with regard to a viewpoint if their relation is PO (Figure 5).



Figure 5. Indeterminacy of the layering of objects in a PO relation

Because we do not assume the transparency of objects, and we have no information about the hidden part, RCC is not suitable for representing invisibility. Therefore, we introduce new predicates to represent a situation in which only one object is observed. $Z(x, y)$ indicates that only y is observed, and $Zi(x, y)$ indicates that only x is observed.

To simplify the problem, as a first step, we discuss the relation between two objects x and y that satisfy the following conditions.

- x is a dynamic solid object that is smaller than y .
- y is a static object whose shape is either an n -prism, a pole, or an L-shaped column (Figure 6).
- The relative size of the objects in each image from any viewpoint is stable at every instant.

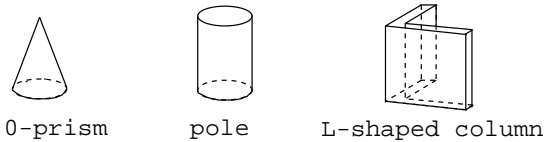


Figure 6. Examples of objects admitted as y

The purpose of the second condition is to reduce the number of blind spots. Note that a ball, for example, cannot be allowed. This constraint ensures that when an object is not observed in an image taken from some viewpoint, this does not mean that the object is located in a blind spot, but that it is located in back of another object.

We denote by \mathbf{R}_2 and \mathbf{R}_3 the sets of relationships of images in the 2D plane and bodies in 3D space, respectively. Let \mathcal{R}_2 be $\{DC, EC, PO, TPP, NTPP, Z\}$. These relations are jointly exhaustive and pairwise disjoint. For $\mathbf{R}_2 \in \mathcal{R}_2$, we write $\mathbf{R}_2(x, y, v, t)$ instead of $\mathbf{R}_2(im(x, v, t), im(y, v, t))$. Let \mathcal{R}_3 be $\{DC, EC, PO, TPP, NTPP\}$. These relations are jointly exhaustive and pairwise disjoint. For $\mathbf{R}_3 \in \mathcal{R}_3$, we write $\mathbf{R}_3(x, y, t)$ instead of $\mathbf{R}_3(bd(x, t), bd(y, t))$. Moreover, we can omit t when the meaning is clear.

Due to the above constraints, $EQ, TPPi, NTPPi$, and Zi never appear. The following axiom reflects this property.

[Axiom1] $\forall v(\exists t(TPP(x, y, v, t) \vee NTPP(x, y, v, t)) \longrightarrow \forall t'(\neg TPPi(x, y, v, t') \wedge \neg NTPPi(x, y, v, t') \wedge \neg EQ(x, y, v, t') \wedge \neg Zi(x, y, v, t')))$

3.2 Viewpoint

For a space constituted by the above x and y , two viewpoints v_u and v_s are specified as follows: v_u points in the direction of y 's base. v_s points in the direction of y 's side. v_u and v_s are called the *upper viewpoint* and *side viewpoint*, respectively.

We introduce the predicates *fore* and *back* indicating which object is nearer from a specified viewpoint.

$fore(x, y, v, t)$: x is in the foreground of y , namely, x is nearer to v than y at the instant t .

$back(x, y, v, t)$: x is in the background of y , namely, x is farther from v than y at the instant t .

Formally, they are defined as follows. Let $dist(p, q)$ denote the distance between the points p and q in 3D space, and let p_x and p_y denote points in $bd(x, t)$ and $bd(y, t)$, respectively. Then

$$fore(x, y, v, t) =_{def} \forall p_x \forall p_y. (dist(p_x, v) \leq dist(p_y, v))$$

$$back(x, y, v, t) =_{def} \forall p_x \forall p_y. (dist(p_x, v) \geq dist(p_y, v))$$

The following axiom indicates the continuity of a transition of relative positional relations. It specifies that if the foreground/background relation of two objects is changed, then there exists an instant in which a change of foreground and background occurs.

[Axiom2] $\forall v((back(x, y, v, t_1) \wedge fore(x, y, v, t_2)) \longrightarrow \exists t((t_1 \leq t \leq t_2) \wedge PO(x, y, v, t)))$

4 MODELING AND REASONING

4.1 Construction of a qualitative 3D model

We represent the relations of rectangles extracted from an image as \mathbf{R}_2 relations. Then we construct a qualitative 3D model from this set of relations via the following process.

First, we derive the \mathbf{R}_3 relation of the bodies of two objects from a single image based on a single viewpoint. The relation is uniquely determined in some cases. If it is not determined, then we derive it from a pair of images taken at the same instant. If we still cannot determine the relation, we check the dynamic change from a specific viewpoint.

(1) Derivation from a single image based on a single viewpoint

If two objects are observed to be disconnected from a certain viewpoint, then they are disconnected in 3D space.

[Rule 1] $DC(x, y, v_s) \vee DC(x, y, v_u) \longrightarrow DC(x, y)$

(2) Derivation from a pair of images taken at the same instant

If two objects are observed to be externally connected from a side viewpoint, and not disconnected from an upper viewpoint, then they are externally connected in 3D space.

[Rule 2] $EC(x, y, v_s) \wedge \neg DC(x, y, v_u) \longrightarrow \mathbf{EC}(x, y)$

If two objects are observed to be externally connected from a side viewpoint, and partially overlapped, or only one object is observed from an upper viewpoint, then they are externally connected in 3D space.

[Rule 3] $EC(x, y, v_s) \wedge (PO(x, y, v_u) \vee Z(x, y, v_u)) \longrightarrow \mathbf{EC}(x, y)$

If only one object is observed from a side viewpoint, and the other is observed to be a tangential proper part from an upper viewpoint, then it is a tangential proper part of the other object in 3D space.

[Rule 4] $(Z(x, y, v_s) \wedge TPP(x, y, v_u)) \longrightarrow \mathbf{TPP}(x, y)$

(3) Focusing relations in 3D space

In the following cases, we cannot determine a unique relation in 3D space, but can narrow the result to two possible relations.

If two objects are observed to be externally connected from an upper viewpoint, and one is a (non-)tangentially proper part from a side viewpoint, then they are externally connected or partially overlapped in 3D space.

[Rule 5] $EC(x, y, v_s) \wedge (TPP(x, y, v_u) \vee NTPP(x, y, v_u) \vee Z(x, y, v_u)) \longrightarrow \mathbf{EC}(x, y) \vee \mathbf{PO}(x, y)$

If only one object is observed from a side viewpoint, and only one object is observed, or the other is a non-tangentially proper part from an upper viewpoint, then it is a (non-)tangentially proper part in 3D space. In this case, we can refine our judgment no further without assuming the transparency of y .

[Rule 6] $(Z(x, y, v_s) \wedge (NTPP(x, y, v_u) \vee Z(x, y, v_u))) \longrightarrow \mathbf{TPP}(x, y) \vee \mathbf{NTPP}(x, y)$

Table 1 lists the rules used to derive a relation in 3D space from 2D data. In this table, “-” means “impossible”.

Table 1. Deriving relations in 3D space from 2D data

$v_s \setminus v_u$	DC	EC	PO	TPP	NTPP	Z
DC	DC	DC	DC	DC	DC	DC
EC	DC	EC	EC	EC	EC	EC/PO
PO	DC	EC	-	-	-	-
TPP	DC	EC	-	-	-	-
NTPP	DC	EC	-	-	-	-
Z	DC	EC	-	TPP	TPP/NTPP	TPP/NTPP

If y is known to be a solid object, either **DC** or **EC** holds in 3D space. Therefore, Rule 5 is replaced by the following Rule 5', where *solid*(y) indicates that y is a solid object.

[Rule 5'] $solid(y) \wedge EC(x, y, v_s) \longrightarrow \mathbf{EC}(x, y)$

If y is not known to be a solid object, we can narrow the result by using data from a time t' which is different from t . This is the situation in which x moves between t and t' and part of x is included in y at time t (Figure 7).

(4) Derivation from a dynamic change of a specific viewpoint

[Rule 7] $EC(x, y, v_s, t) \wedge (TPP(x, y, v_u, t) \vee NTPP(x, y, v_u, t) \vee Z(x, y, v_u, t)) \wedge \exists t' (t' \neq t) \wedge TPP(im(x, v_s, t), im(x, v_s, t')) \longrightarrow \mathbf{PO}(x, y, t)$

(5) More intelligent derivation

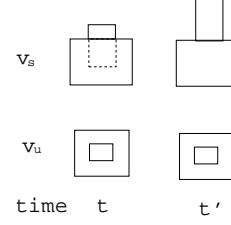


Figure 7. A case in which it is determined that PO holds at an instant t

The example shown in Figure 8(a)(b) is a more complex case.

In the image taken at time t , the three objects x_1 , x_2 and y are observed, and have the following relationships:

$EC(x_1, y, v_s, t) \wedge EC(x_2, y, v_s, t) \wedge DC(x_1, x_2, v_s, t) \wedge TPP(x_1, y, v_u, t) \wedge Z(x_2, y, v_u, t) \wedge Z(x_2, x_1, v_u, t)$

Note that x_2 is not hidden by y in the image taken from the upper viewpoint, but the relation is represented as $Z(x_2, y, v_u, t)$.

In the image taken at time t' , the following relationship holds:

$EC(x, y, v_s, t') \wedge TPP(x, y, v_u, t')$

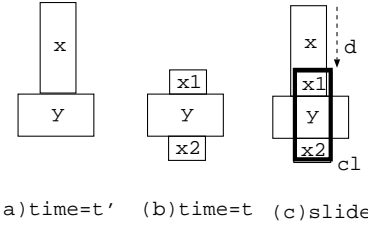


Figure 8. Objects separately observed from v_s

When such 2D data are given, we cannot determine whether both x_1 and x_2 are parts of the same object x , or are totally different objects. According to Rule 7, **PO**(x, y, t) holds, and x_1 is considered to be a part of x . However, we are uncertain about x_2 ; this is due to the fact that although the number of objects is stable, one of them may be invisible (Figure 9). Therefore, the case shown in Figure 9 is also possible.

We define a new function *slide* to solve this problem.

$slide(im(x, v, t), d)$ is the function that returns the data obtained by transferring the image of x from v at the instant t according to the specified vector d (Figure 8(c)). Let cl be the closure of $im(x_1, v_s, t)$ and $im(x_2, v_s, t)$, which is shown in the boldface frame of Figure 8(c). If $slide(im(x, v_s, t), d)$ coincides with cl , then x_2 is considered to be a part of x .

Therefore, we have the following rule:

[Rule 8] $EC(x_1, y, v_s, t) \wedge EC(x_2, y, v_s, t) \wedge \exists t' (t' \neq t) \wedge DC(x, y, v, t') \wedge TPP(im(x_1, v_s, t), im(x, v_s, t')) \wedge \exists d EQ(slide(im(x, v_s, t'), d), cl) \longrightarrow \mathbf{PO}(x, y, t)$

If this condition does not hold, then x_1 and x_2 are different objects.

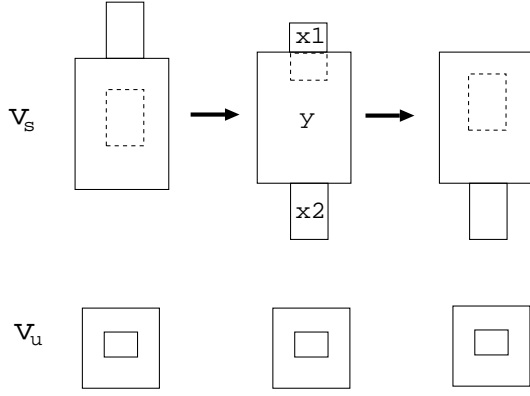


Figure 9. A case of different objects

4.2 Determining a foreground/background relationship

In addition to RCC relations in 3D space, we can determine which object is located in the foreground from a specific viewpoint.

Given a tube object y , $has_cap(y, v)$ indicates that it has a cap in the v direction, and $has_bot(y, v)$ indicates that it has a cap in the inverse direction to v .

We specify the rules for deriving the foreground and background relation of two objects with regard to a given viewpoint.

If only y is observed from the side viewpoint, then x is in the background of y .

[Rule 9] $Z(x, y, v_s) \rightarrow back(x, y, v_s, t)$

If x is a (non-)tangentially proper part, then x is in the foreground of y .

[Rule 10] $(TPP(x, y, v_s) \vee NTPP(x, y, v_s)) \rightarrow fore(x, y, v_s, t)$

If only y is observed from the upper viewpoint, then x is in the background of y .

[Rule 11] $Z(x, y, v_u) \rightarrow back(x, y, v_u, t)$

If x is a (non-)tangentially proper part and y has a cap, then x is in the foreground of y ; otherwise, it is not determined.

[Rule 12] $(TPP(x, y, v_u) \vee NTPP(x, y, v_u)) \wedge has_cap(y, v_u) \rightarrow fore(x, y, v_u, t)$

When $PO(x, y, v, t)$ holds, we cannot determine the foreground and background relation of two objects at t from the given data. In this case, we can derive the relation by using data from a subsequent or previous instant (Figure 10).

[Rule 13] $\exists t' ((t' \neq t) \wedge Z(x, y, v, t') \wedge PO(x, y, v, t)) \rightarrow back(x, y, v, t') \wedge back(x, y, v, t)$

[Rule 14] $\exists t' ((t' \neq t) \wedge TPP(x, y, v, t') \wedge PO(x, y, v, t)) \rightarrow fore(x, y, v, t') \wedge fore(x, y, v, t)$

4.3 Determining the direction of motion

So far, we have assumed that relative size of images of objects is invariant from both viewpoints at any time. If an object moves toward a viewpoint or away from it, this assumption does not hold. In this subsection, we allow the relative size of objects to vary, and discuss the derivation of the direction of motion from the transition of R_2 . By comparing the sizes at instants t and t' , the directions of motion

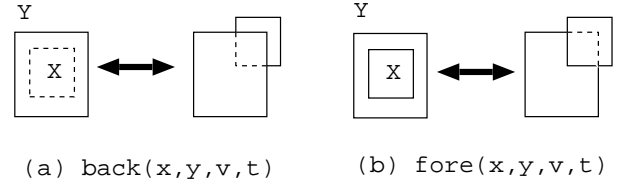


Figure 10. Determining the foreground/background

from t to t' are classified into three types: *sameDistMove*, *goFar*, and *comeNear*.

If x moves while maintaining the same distance from a viewpoint v , then the size of $im(x, v, t)$ is invariant. This type of movement is referred to as *sameDistMove*.

[Rule 15] $\exists d \exists t' ((t \neq t') \wedge EQ(slide(im(x, v, t'), d), im(x, v, t))) \rightarrow sameDistMove(x, t, t')$

If x moves toward a viewpoint or away from it, then the size of $im(x, v, t)$ varies. This type of movement is either *goFar* (departing from v) or *comeNear* (approaching v), depending on the situation.

[Rule 16] $\exists t' ((t < t') \wedge (TPP(im(x, v, t), im(x, v, t')) \vee NTPP(im(x, v, t), im(x, v, t')))) \rightarrow goFar(x, t, t')$

[Rule 17] $\exists t' ((t > t') \wedge (TPP(im(x, v, t), im(x, v, t')) \vee NTPP(im(x, v, t), im(x, v, t')))) \rightarrow comeNear(x, t, t')$

5 REASONING FOR A 3D MODEL

We describe the reasoning for inferring relations in 3D space.

5.1 Reasoning for a tube object

We determine whether a tube object has a cap or a bottom from the transition of relations in 3D space.

[Rule 18] $\exists t' ((t > t') \wedge EC(x, y, t) \wedge PO(x, y, t')) \rightarrow \neg has_cap(y, v_u)$

[Rule 19] $\exists t' ((t < t') \wedge EC(x, y, t) \wedge PO(x, y, t')) \rightarrow \neg has_bot(y, v_u)$

Moreover, if only one object is observable from the upper viewpoint, then we can infer that y has either a cap or a bottom.

[Rule 20] $Z(x, y, v_u) \rightarrow has_cap(y, v_u) \vee has_bot(y, v_u)$

5.2 Event retrieval

The occurrence of an event can be retrieved from a sequence of 3D relations $R_3(x, y, t_0), \dots, R_3(x, y, t_n)$ ($n \geq 1$), where t_{i+1} is the next time instant after t_i , denoted by $next(t_i)$, for each i ($0 \leq i < n - 1$).

We present the definitions of several events.

[Def 1] x enters y .

$enter(x, y) =_{def} EC(x, y, t) \wedge PO(x, y, next(t)) \wedge TPP(x, y, next(next(t)))$

[Def 2] x exits y .

$exit(x, y) =_{def} TPP(x, y, t) \wedge PO(x, y, next(t)) \wedge EC(x, y, next(next(t)))$

[Def 3] x passes through y if x enters y and successively exits y .

$pathThrough(x, y) =_{def} EC(x, y, t) \wedge PO(x, y, next(t)) \wedge$

$$\text{TPP}(x, y, \text{next}(\text{next}(t))) \wedge \text{PO}(x, y, \text{next}(\text{next}(\text{next}(t)))) \wedge \text{EC}(x, y, \text{next}(\text{next}(\text{next}(\text{next}(t)))))$$

[Def4] x fits inside y if the positional relationship is unchanged after entering.

$$\text{fitIn}(x, y) =_{\text{def}} \text{EC}(x, y, t) \wedge \text{PO}(x, y, \text{next}(t)) \wedge \text{TPP}(x, y, \text{next}(\text{next}(t))) \wedge \text{PO}(x, y, \text{next}(\text{next}(\text{next}(t)))) \wedge \text{PO}(x, y, \text{next}(\text{next}(\text{next}(\text{next}(t)))))$$

5.3 Environment

The transition of \mathbf{R}_3 relations in 3D space follows Figure 3. However, the transition of \mathbf{R}_2 differs from the usual transitions, because we do not assume the transparency of objects.

Figure 11 and Figure 12 show the transitions of \mathbf{R}_2 from the viewpoint v , assuming that the relative size of objects from v is unvarying and varying, respectively.

Figure 11 shows the case in which $im(x, v, t)$ is always smaller than $im(y, v, t)$.

In Figure 12, transition is possible if x moves while maintaining the same distance from the viewpoint v , or changes its distance from v . For example, transition from PO to TPP is possible if the motion of x is *sameDistMove*, or x moves away from v and approaches y .

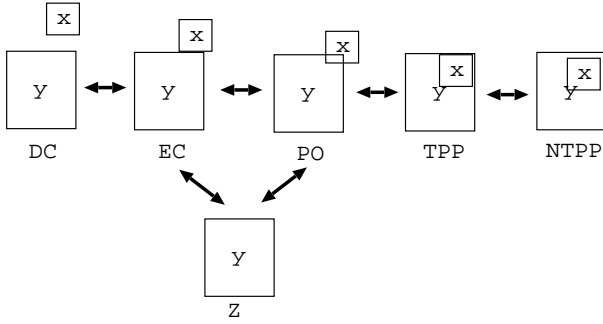


Figure 11. Environment: the relative size of the objects is unvarying

Assume that the size of a moving object x is bigger than the entrance of a static tube object y . In this case, x never enters y . Therefore either DC or EC holds in 3D space. As for the 2D plane, TPPi/NTPPi can hold instead of TPP/NTPP. Therefore, the transition graph for this case is obtained by replacing these parts.

6 RELATED WORK

The reasoning used to derive the relationships of moving solid objects has been well studied, both qualitatively and quantitatively. In these works, the foreground and background relations between objects are determined mostly from the complement of the continuous data. However, these works do not consider tube type objects or objects with containers. Galton formalized various types of objects, including containers, and discussed their properties [9]. However, he did not discuss the use of practical video data. In contrast, our objective is the automatic extraction of data and formalization of the method.

A number of works have focused on qualitative simulation. Bennett et al. explored the expressive power of region-based geometry

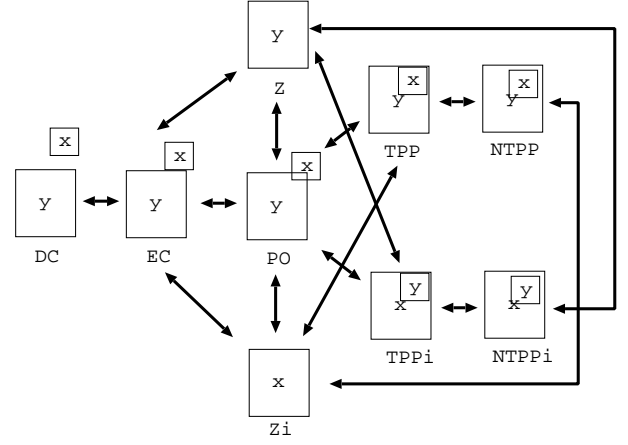


Figure 12. Environment: the relative size of the objects is varying

[2]. They formalized various types of movements in a qualitative manner. Hazarika et al. formalized the abduction of a motion history from local surveys [10]. Weghe et al. presented a trajectory-based theory to handle qualitative changes between moving objects [20]. Boxer et al. demonstrated how general physical behaviors can be learned from a sequence of qualitative representations with direction and velocity, using Bayesian networks [3]. Almost all of these works involve only 2D motions, and issues that arise when formalizing 3D motions, such as the occlusion problem or the tube-passage problem are not discussed.

Randell et al. presented an interesting work on occlusion [16]. They proposed ROC20, a refined RCC8 system that describes relative relationships of objects, including tube type objects, from a single viewpoint. They discussed the relationships between changes of viewpoint and changes of relationship. However, their work is based on the idea that the reasoner knows the locations of the target objects a priori, and investigates the relationships in the 2D plane for the corresponding 3D data. Their objective is to formally represent a situation using QSR. Unlike their procedure, we begin with rectangles extracted from the video data, from which we cannot know the state of the background of an object, with the objective of determining 3D relative positional relations between objects.

Santos et al. formalized abduction from a sequence of snapshots [18, 6]. They proposed Depth Profile Calculus (DPC) and Dynamic Depth Profile Calculus (DDPC). They introduced the relation *coalescent*, which represents occlusion, and modified RCC to fit the representative image data. They discussed the predication of invisible parts. Their basic idea is similar to ours, but their approach is different. They use an image obtained from a single viewpoint, and do not consider the tube-passage problem. They represent image data using the three elements of distance, size, and depth, and retrieve events in 3D space from temporal sequences of these data. In contrast, we use image data obtained from two different viewpoints in a single instant, as well as video data from around that instant, and derive RCC relations in 3D space from these. Moreover, we handle a tube object by using image data from a pair of viewpoints.

Fogliaroni et al. investigated the relationship between viewpoints and blind spots, and demonstrated the reasoning in a QSR framework. They applied their technique to localization and navigation [7].

Sridhar et al. presented a framework for unsupervised learning of event classes from video data, aimed at practical application. In their approach, convex closures of multiple objects are extracted from video data, and their relations are represented qualitatively. The learning of event classes is processed based on a probabilistic model [13]. They also proposed a more efficient method for handling noisy data [14]. They regard video data as a projection of 3D objects onto a 2D plane. In contrast, we treat 3D objects as 3D entities, rather than as projections.

In the research areas of the visual language or image processing, researchers have proposed methods that facilitate RCC over 3D [11, 1, 17]. These are based on projection onto the xyz-axes, and also assume sufficient information about the location in 3D space. However the formalization in these works is not sufficient, and they do not refer to the environment.

7 CONCLUSION

We have described a method of deriving relative relationships in three dimensional space for objects extracted from video data in a QSTR framework. We use image data obtained from two different view-points at a single instant, as well as video data from around that instant.

The proposed method offers the following advantages.

1. The relative relationships of objects in 3D space can be derived without assuming the transparency of objects.
2. Tube type objects and events related to such objects can be handled.
3. The method can be applied to the automatic extraction of events from video data.

As this is a first step toward automated qualitative recognition of relations between objects in 3D space from dynamic image data, we imposed several restrictions on the objects to be handled. In the future, we will generalize this method, investigate the properties more deeply, and perform experiments using actual video data.

ACKNOWLEDGEMENTS

This research is supported by KAKENHI222500141.

REFERENCES

- [1] J. Albath, J. L. Leopold, C. L. Sabharwal, and K. Perry, 'Efficient reasoning with rcc-3d', in *Proceedings of Knowledge Science, Engineering and Management 2010 (KSEM2010)*, number 6291 in LNAI, pp. 470–481. Springer, (2010).
- [2] B. Bennett, A. G. Cohn, P. Torrini, and S. M. Hazarika, 'Describing rigid body motions in a qualitative theory of spatial regions', in *Proceedings of AAAI-2000*, pp. 503–509, (2000).
- [3] P. A. Boxer, 'Learning naive physics by visual observation: Using qualitative spatial representations and probabilistic reasoning', *International Journal of Computational Intelligence and Applications*, **1**(3), 273–285, (2001).
- [4] A. G. Cohn and S. M. Hazarika, 'Qualitative spatial representation and reasoning: An overview', *Fundamental Informaticae*, **46**(1-1), 1–29, (2001).
- [5] Z. Cui, A. G. Cohn, and D. A. Randell, 'Qualitative simulation based on a logical formalism of space and time', in *Proceedings of AAAI-92*, pp. 679–684, (1992).
- [6] M. V. dos Santos, R. C. de Brito, H. H. Park, and P. Santos, 'Logic-based interpretation of geometrically observable changes occurring in dynamic scenes', *Applied Intelligence*, **31**, 161–179, (2009).
- [7] P. Fogliaroni, J. Wallgrün, E. Clementini, F. Tarquini, and D. Wolter, 'A qualitative approach to localization and navigation based on visibility information', in *COSIT09*, pp. 312–329, (2009).
- [8] K. D. Forbus, 'Qualitative process theory', *Artificial Intelligence*, **24**(1-3), 85–168, (1984).
- [9] A. Galton, 'The formalities of affordance', in *Proceedings of ECAI2010 Workshop on Spatio-Temporal Dynamics*, pp. 1–6, (2010).
- [10] S. M. Hazarika and A. G. Cohn, 'Abducting qualitative spatio-temporal histories from partial observations', in *Proceedings of the Eighth International Conference on Principles of Knowledge Representation and Reasoning (KR02)*, pp. 14–25, (2002).
- [11] C. Li, J. Chao, C. Yin, and L. Ma, 'Qualitative spatial representation and reasoning in 3d space', in *Proceedings of 2009 International Conferences on Intelligent Computation Technology and Automation (ICICTA)*, pp. 653–657. IEEE, (2009).
- [12] G. Ligozat, *Qualitative Spatial and Temporal Reasoning*, Wiley, 2011.
- [13] S. Muralikshna, A. G. Cohn, and D. C. Hogg, 'Unsupervised learning of event classes from video', in *Proceedings of the 10th International Conference on Artificial Intelligence (AAAI10)*. AAAI, (2010).
- [14] S. Muralikshna, A. G. Cohn, and D. C. Hogg, 'From video to rcc8: exploiting a distance based semantics to stabilise the interpretation of mereotopological relations', in *Proceedings of the 10th International Conference on Spatial Information Theory (COSIT11)*. Springer, (2011).
- [15] D. Randell, Z. Cui, and A. Cohn, 'A spatial logic based on regions and connection', in *Proceedings of the Third International Conference on Principles of Knowledge Representation and Reasoning (KR92)*, pp. 165–176. Morgan Kaufmann Publishers, (1992).
- [16] D. Randell, M. Winkowski, and M. Shanahan, 'From images to bodies: Modelling and exploiting spatial occlusion and motion parallel', in *Proceedings of the 17th international joint conference on Artificial intelligence (IJCAI01)*, pp. 57–63, (2001).
- [17] C. Sabharwal, J. L. Leopold, and N. Elie, 'A more expressive 3d region connection calculus', in *Proc. of the 2011 International Workshop on Visual Languages and Computing*, pp. 307–311, (2011).
- [18] P. E. Santos, 'Reasoning about depth and motion from an observer's viewpoint', *Spatial Cognition and Computation*, **7**(2), 133–178, (2007).
- [19] O. Stock, *Spatial and Temporal Reasoning*, Kluwer Academic Press, 1997.
- [20] N. van de Weghem, A. G. Cohn, G. de Tré, and P. de Maeyer, 'A qualitative trajectory calculus as a basis for representing moving objects in geographical information systems', *Control and Cybernetics*, **35**(1), 97–119, (2006).

On the Neighborhood and Distance Between Qualitative Spatio-temporal Configurations

Dominique D’Almeida¹ and Mouny Samy Modeliar² and Nicolas Schwind³

Abstract.

Representing and reasoning about space and time are important issues for many AI applications. In the past three decades numerous qualitative formalisms have been pointed out to this purpose. Most of these formalisms consider qualitative constraint networks (QCNs) as a way to represent the relative positions about a finite set of spatial or temporal entities. QCNs are expressive structures to reason with incomplete information, but they do not include any reasoning tool about its dynamics. Then in recent years the temporalization of some formalisms has gained more attention through the notion of conceptual neighborhood between the relations of these formalisms. However, very few works addressed the problem of determining how far are two qualitative scenarios from each other, though it constitutes a central task in the context of belief change (e.g., for merging QCNs coming from several sources.) This paper is intended to fill the gap. We first review the notion of conceptual neighborhood between relations. We consider a general definition that is based on a concept of transformation law for the spatial or temporal entities. We then give a definition of neighborhood and distance between qualitative scenarios. In contrast to previous distances that are highly dependent of the structural aspects of the scenarios, we adopt here an approach that arises from a domain-based notion of neighborhood between scenarios.

1 Introduction

Representing and reasoning about spatial or temporal entities are important tasks in many AI applications [2, 14, 12]. Adhering to a qualitative calculus is necessary when the available information about a set of spatial or temporal entities is expressed in terms of non-numerical relationships between these entities (e.g., when information comes primarily from natural language sentences.) Many such formalisms have been pointed out over the past 30 years. On the spatial side different aspects of representation can be handled with topological relations (e.g., the RCC8 formalism [21]) or those based on a precedence relation when orientation is required (e.g., cardinal directions formalism [19].) On the temporal side let us mention the well-known Allen’s interval algebra [1] used to represent relations between temporal intervals over the rationals. Then, numerous Allen-like calculi [4, 3] and combinations of qualitative formalisms [10] have also been proposed. Most of these formalisms consider qualitative constraint networks (QCNs) as a way to represent information

about a finite set of spatial or temporal entities and their relative positions.

In order to deal with incomplete or inaccurate information, Freksa [9] introduced the notion of *conceptual neighborhood* between the relations of the interval algebra. This allows to reason with different degrees of granularity about the relative positions of intervals, that is, to improve the power of the inference process by abstracting the basic relations of the formalism. For the same purpose, Gooday and Cohn [11] took an inspiration from Freksa’s work and adapted the notion of conceptual neighborhood between the relations of the RCC8 formalism. The notion of conceptual neighborhood has also been studied by Egenhofer [8] who identified the *topological deformations* of the relations of RCC8 in order to study the dynamics of a system, or more precisely, the different steps that are encountered during an evolution of a configuration of two specific entities. From the context of belief change, Condotta et al. [6, 5] recently addressed the problem of merging conflicting QCNs coming from several sources. They proposed several families of merging operators that associate a multiset of QCNs with a consistent set of spatio-temporal information that represents the input QCNs in a global way. The key ingredient of these operators is a *distance* between the QCNs that is induced from a notion of conceptual neighborhood between relations.

The notion of conceptual neighborhood over a set of relations describes which transitions are possible if the underlying entities are subject to a small change ruled by a given *transformation law*. The definition of a transformation law is not trivial, indeed it is not induced by the theory of the underlying qualitative formalism. Different conceptual neighborhoods can be found depending on whether the entities are allowed to change their size or their absolute position, that is, how the corresponding transformation is ruled. For instance, Freksa [9] identifies three neighborhood graphs for the interval algebra that corresponds to three different transformation laws of the intervals onto the rationals (e.g., their expansion, reduction, shift, etc.) Moreover, usually more than two entities are involved and we are faced to the following more sophisticated problem: which spatial or temporal scenarios (i.e., *atomic QCNs*) are directly accessible from a given initial scenario when the possible transformations of the underlying entities are ruled by a given condition? Providing a solution to this problem boils down to extend the notion of conceptual neighborhood between relations to the one between scenarios. It allows us to reason about the dynamics of spatial or temporal configurations and also constitutes a base for defining a distance between scenarios. In [6, 5] a notion of distance between scenarios has first been proposed, but this distance is computed in a piecewise fashion as an aggregation of local distances between the constraints of these scenarios. Thus, it highly depends on the structure of the scenarios instead of the qualitative configurations that they represent.

In this paper, we propose a more “semantical” definition of conceptual neighborhood between scenarios, that is, independent from

¹ CRIL CNRS UMR8188 - Université d’Artois, Rue Jean Souvraz F-62307 Lens France, email: dalmeida@cril.fr

² CRIL CNRS UMR8188 - Université d’Artois, Rue Jean Souvraz F-62307 Lens France, email: samymodeliar@cril.fr

³ National Institute of Informatics, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan, email: schwind@nii.ac.jp

their syntactical structure and thus more representative. For this purpose, we first propose a formal definition of a transformation law, that is, how spatial or temporal entities are allowed to be changed through their domain. Then, we propose a definition of conceptual neighborhood between relations that refines those proposed in the literature; in particular, this conceptual neighborhood is *directed*, this means that when a relation is a neighbor of an other relation, the converse is not necessarily true. Lastly, we extend the notion of neighborhood between relations to the one between scenarios. Based on this latter notion, we propose a new definition of distance between scenarios.

The rest of the paper is organised as follows. We start with some formal preliminaries about qualitative formalisms and qualitative constraint networks in Section 2. In Section 3 we point out the formal notion of a transformation law, that is illustrated with a running example by adapting Freska’s work into our framework. We also derive from it the notion of conceptual neighborhood between relations and some (directed) neighborhood graphs for the interval algebra. In Section 4 we define the neighborhood between consistent scenarios, and the notion of distance between scenarios. We discuss about computational issues in Section 5 before concluding in Section 6.

2 Preliminaries

This section introduces necessary notions of qualitative algebras and definitions around qualitative constraint networks. A qualitative formalism considers a finite set B of binary relations over a non-empty set D , called *basic relations*. The elements of D constitute the universe of all considered spatial or temporal entities. Each basic relation $b \in B$ represents a specific relationship between two elements of D . The set B is required to be a *partition scheme* [20], i.e., it satisfies the following properties: (i) B forms a partition of $D \times D$, namely any pair of $D \times D$ satisfies one and only one basic relation of B ; (ii) the identity relation on D , denoted eq , belongs to B ; lastly, (iii) if b is a basic relation of B , then its inverse, denoted b^{-1} , also belongs to B .

For illustration we consider a well-known qualitative formalism introduced by Allen, called the interval algebra [1]. This formalism considers a set B_{int} of thirteen basic relations defined on the domain of non-punctual (durative) intervals over the rational numbers: $D_{int} = \{(x^-, x^+) \in \mathbb{Q} \times \mathbb{Q} : x^- < x^+\}$. An interval typically represents a temporal entity. The basic relations of $B_{int} = \{eq, p, pi, m, mi, o, oi, s, si, d, di, f, fi\}$ are depicted on Table 1. Each one of them represents a particular relationship between two intervals. For example, the relation $m = \{((x^-, x^+), (y^-, y^+)) \in D_{int} \times D_{int} : x^+ = y^-\}$ represents the case where the upper bound of the first interval and the lower bound of the second one coincide.

The set 2^B , the set of all subsets of B , forms the set of all *relations*. 2^B , together with the usual set-theoretic operators union (\cup), intersection (\cap), complementation ($\bar{}$), and weak composition (\diamond) (see [22] for details) is called a *qualitative formalism*.

Pieces of information about the relative positions of a set of spatial or temporal entities can be represented by means of qualitative constraint networks (QCNs for short.) Formally, a QCN (on 2^B) is defined as a pair (V, C) where $V = \{v_1, \dots, v_n\}$ is a finite set of variables representing the entities, and C is a mapping which associates with each pair of variables (v_i, v_j) a relation $N[i, j]$ of 2^B such that $N[i, i] = \{eq\}$ and $N[i, j] = N[j, i]^{-1}$ for every pair of variables $v_i, v_j \in V$. C is also called the set of *constraints* of N .

Given a QCN $N = (V, C)$, an *instantiation* of N over $V' \subseteq V$ is a mapping α from V' to D ; a *solution* of N is an instantiation of N over V such that for every pair $(v_i, v_j) \in V \times V$, $(\alpha(v_i), \alpha(v_j))$ satisfies $N[i, j]$, i.e., there exists a basic relation $b \in N[i, j]$ such that $(\alpha(v_i), \alpha(v_j)) \in b$ for every $v_i, v_j \in V$; N is *consistent* if it

Relation	Illustration	Inverse relation
X precedes Y p		Y is preceded by X pi
X meet Y m		Y is met by X mi
X overlaps Y o		Y is overlapped by X oi
X starts Y s		Y is started by X si
X is contained in Y d		Y contains X di
X finishes Y f		Y is finished by X fi
X equals Y eq		Y equals X eq

Table 1. The thirteen basic relations of the interval algebra

admits a solution; a *sub-network* N' of N is a QCN (V, C') such that $N'[i, j] \subseteq N[i, j]$, for every pair of variables v_i, v_j ; a *scenario* σ is a QCN such that each constraint is defined by a singleton relation of 2^B , i.e., a relation containing exactly one basic relation. Let σ be a scenario, then abusing notations, when it is clear from the context $\sigma[i, j]$ will also denote the basic relation specifying the constraint between the two variables v_i and v_j . A scenario σ of N is a sub-network of N . In the rest of the paper, we will denote by QCN_V^B the set of all possible consistent scenarios on 2^B and V . From now on, since we only consider scenarios that are consistent, the term “consistent” will be omitted.

Let us illustrate some of these definitions by an example:

Example 1 An interview is scheduled during a time T . It is divided in two parts, a talk E and a discussion phase D . The talk itself is finished by a session Q for specific questions. The corresponding QCN $N = (V, C)$, defined on the interval algebra $2^{B_{int}}$ is represented in Figure 1. We have $V = \{E, T, D, Q\}$ and C is determined by the constraints described below⁴. For instance, the talk E precedes or meets the discussion phase D . Figure 2 illustrates a scenario σ_1 of N and a solution α_1 of σ_1 .

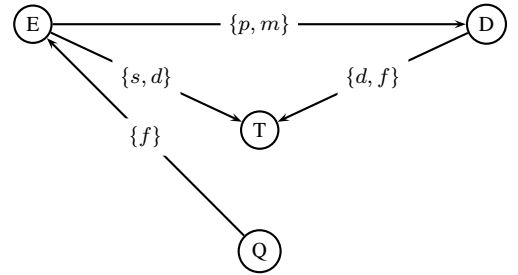


Figure 1. N , a QCN defined on $2^{B_{int}}$

The consistency problem for QCNs, i.e., the problem that consists in deciding whether a given QCN is consistent or not, is undecidable in the general case [13]. Nevertheless, we assume in this paper that this decision problem is decidable in polynomial time when its

⁴ In order to alleviate the figures, for each pair of variables (v_i, v_j) , we do not represent the constraint $N[i, j]$ when $N[i, j] = B$, when $N[j, i]$ is represented or when $i = j$.

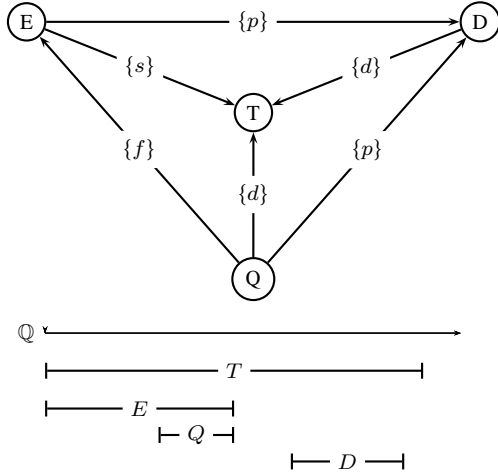


Figure 2. σ_1 , a scenario of N and α_1 , a solution of σ_1 .

input is restricted to the set of all scenarios; this is the case for all the qualitative formalisms of interest proposed in the literature. As a consequence, the consistency problem for QCNs belongs to the complexity class NP. In the general case, it is an NP-complete problem.

3 Transformation Laws and Neighborhood Between Relations

A notion of conceptual neighborhood can be defined between the basic relations of a partition scheme B and is induced by an underlying *transformation law* of the entities through their domain. A transformation law describes every possible way for a spatial or temporal entity to “move” or to be “transformed” through its domain. Different transformation laws can be defined given a qualitative formalism. First, it depends on the structure of the domain D . For example, when D is the set of all closed subsets of a topological space (e.g., a line or a closed region of the euclidean plane), Kurata considers in [18] a transformation law that describes all possible smooth distortions of the elements of D without modifying their topological structure. As an other example, when D is a real affine space of finite dimension, a transformation law does not correspond to a distortion of an element but to its motion in the space. Moreover, among the same domain a transformation law can be defined in different ways. For instance, when D is a real affine space of finite dimension, a transformation law may correspond to all possible continuous motions of a punctual entity onto the space, or it may restrict these motions to those parallel to the axis defining the space.

To illustrate the idea of transformation law, let us consider again the interval algebra. Freksa [9] identifies three transformation laws. The first transformation law describes a continuous shift of one of the two bounds of any interval to the left or to the right. The second transformation law considers a continuous and simultaneous shift of the two bounds of any interval into the same direction. The third transformation law identified by Freksa considers the reduction or the expansion of any interval, i.e., a continuous and simultaneous shift of its two bounds into opposite directions.

We shall now propose a general and formal definition of a transformation law on a domain D . From now on, D is a fixed domain.

Definition 1 (decreasing filtration) A decreasing filtration on D is a sequence of subsets of D that is decreasing for the set inclusion.

If \mathcal{F} is a decreasing filtration on D , then there is an induced totally ordered set I such that $\mathcal{F} = \{D_i \subseteq D \mid i \in I\}$. Then, \mathcal{F} is also denoted $(D_i)_{i \in I}$.

Definition 2 (transformation on D) A transformation on D , denoted θ , is a set $\{\mathcal{F}_x \mid x \in D\}$ of decreasing filtrations on D . θ is also denoted $(\mathcal{F}_x)_{x \in D}$.

The intuition behind the definition of a transformation on D is explained as follows. For every element $x \in D$, every element $D_i \in \mathcal{F}_x$ represents the “neighbor area” of this entity, that is, the set of its possible next “moves”. As \mathcal{F}_x is a decreasing filtration, the further D_i lies in \mathcal{F}_x w.r.t. the induced ordering, the finer is the set of possible moves for the entity. For instance, this allows us to describe continuous transformations of spatial or temporal entities in a natural way, without assuming D to be a topological space (see Example 2 below.) To avoid heavy notations, for every $x \in D$ we will denote $\mathcal{F}_x = (D_i)_{i \in I_x}$ when it is clear from the context.

We now define the notion of transformation law on D .

Definition 3 (transformation law on D) A transformation law on D , denoted Θ , is a set of transformations on D .

Let us illustrate this last definition by the third transformation law on D_{int} proposed by Freksa [9], denoted Θ_3 , that describes the reduction/expansion of any interval.

Example 2 Let x be any interval of D_{int} . $|x|$ denotes the length of x . We define a x -expansion as the filtration $\mathcal{F}_x = (D_t)_{t \in]0, +\infty[}$ with $D_t = \{x' \in D_{int} \mid x'^- < x^-, x'^+ > x^+, |x'| < |x| + \frac{1}{t}\}$. Similarly, we define a x -reduction as the filtration $\mathcal{F}_x = (D_t)_{t \in]0, |x|]}$ with $D_t = \{x' \in D_{int} \mid x'^- > x^-, x'^+ < x^+, t < |x'|\}$. Then, an D_{int} -expansion (respectively, a D_{int} -reduction) is a transformation on D_{int} that is a set of x -expansions (respectively, x -reductions), for every $x \in D_{int}$. The third transformation law Θ_3 on D_{int} described by Freksa is then the set of all possible D_{int} -expansions and D_{int} -reductions.

Let us stress the point that the definitions of D_{int} -expansion and D_{int} -reduction introduced in the example above are specific to the case of intervals of D_{int} .

Given a transformation θ on D , one can decide whether two binary relations over D are considered as being *neighbors*. Intuitively, a binary relation $R' \subseteq D \times D$ is a neighbor of another binary relation $R \subseteq D \times D$ w.r.t. a transformation θ on D if every element $(x, y) \in R$ can directly be switched to an element of R' by “applying” the transformation θ on x .

Definition 4 (neighborhood between binary relations) Let $\theta = (\mathcal{F}_x)_{x \in D}$ be a transformation on D , R and R' two binary relations over $D \times D$. R' is said to be a neighbor of R w.r.t. θ , denoted $R \xrightarrow{\theta} R'$ if for every $(x, y) \in R$, there is a $i \in I_x$ such that for every $x' \in D_i$, $(x', y) \in R'$.

Now, let B be a partition scheme on D . Given a specific transformation θ on D , the notion of neighborhood between binary relations induces a particular relation on the set B , called *neighborhood relation* on B . Yet one can notice from Definition 4 that for every binary relation R, R', R'' over $D \times D$, if $R \xrightarrow{\theta} R'$ and $R \xrightarrow{\theta} R''$, then R' and R'' have a non-empty intersection. Thus, since the basic relations of B are jointly exhaustive, it is easy to see that every basic relation of B has at most one neighbor over B . Then, such a neighborhood relation

on B is always functional⁵. The following definition states that every transformation θ on D induces a neighborhood function on B . From now on, B is considered to be a fixed partition scheme on D .

Definition 5 (neighborhood function on B) Let θ be a transformation on D . The neighborhood function on B induced by θ , denoted τ_θ^B , is the function from B to B such that for every $a, b \in B$, $\tau_\theta^B(a) = b$ if and only if $a \xrightarrow{\theta} b$.

A neighborhood function τ_θ^B can be viewed as a graph, called *neighborhood graph on B induced by θ* and denoted $\mathcal{G}(\tau_\theta^B)$, where each vertex represents a basic relation of B . In such a graph, there is an edge from a vertex n_a (corresponding to a basic relation $a \in B$) to a vertex n_b (corresponding to a basic relation $b \in B$) if and only if $\tau_\theta^B(a) = b$.

Example 3 Let θ be any D_{int} -expansion, θ' be any D_{int} -reduction (cf. Example 2.) Figures 3 and 4 depict respectively the neighborhood graphs $\mathcal{G}(\tau_\theta^{B_{int}})$ and $\mathcal{G}(\tau_{\theta'}^{B_{int}})$. We have for instance $\tau_\theta^{B_{int}}(eq) = di$ (cf. Figure 5), since for any couple of intervals $x, y \in D_{int}$ such that $(x, y) \in eq$, any D_{int} -expansion of x to an interval x' directly leads to $(x', y) \in di$. Likewise, we have $\tau_{\theta'}^{B_{int}}(si) = oi$ (cf. Figure 6), since for any couple of intervals $x, y \in D_{int}$ such that $(x, y) \in si$, any D_{int} -reduction of x to an interval x' directly leads to $(x', y) \in oi$.

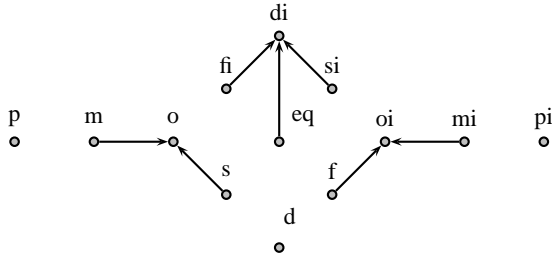


Figure 3. The neighborhood graph $\mathcal{G}(\tau_\theta^{B_{int}})$, where θ is any D_{int} -expansion.

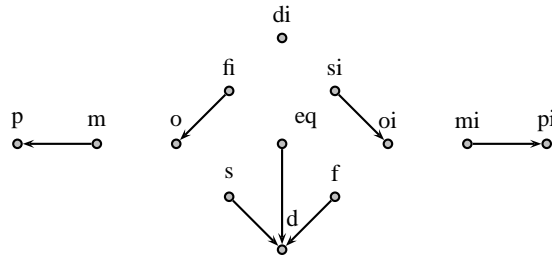


Figure 4. The neighborhood graph $\mathcal{G}(\tau_{\theta'}^{B_{int}})$, where θ' is any D_{int} -reduction.

Obviously enough, any transformation law Θ on D induces a (finite) set of neighborhood functions on B , so-called *neighborhood law on B* :

Definition 6 (neighborhood law on B) Let Θ be a transformation law on D . The neighborhood law on B induced by Θ , denoted T_Θ^B ,

⁵ A binary relation R over $F \times F$ is said to be *functional* if $(\forall f, f', f'' \in F, \text{ if } (f, f') \in R \text{ and } (f, f'') \in R \text{ then } f' = f'')$.

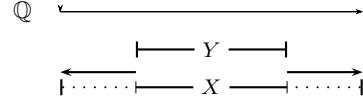


Figure 5. di is a neighbor of eq w.r.t. any D_{int} -expansion θ .

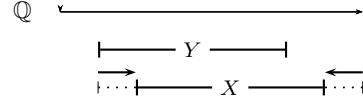


Figure 6. si is a neighbor of oi w.r.t. any D_{int} -reduction θ' .

is the set of all neighborhood functions on B induced by the transformations θ on D , with $\theta \in \Theta$. Formally, $T_\Theta^B = \{\tau_\theta^B \mid \theta \in \Theta\}$.

Example 4 Let θ be any D_{int} -expansion, θ' be any D_{int} -reduction. The neighborhood law on B_{int} induced by the transformation law Θ_3 is the set $T_{\Theta_3}^{B_{int}} = \{\tau_\theta^{B_{int}}, \tau_{\theta'}^{B_{int}}\}$, where θ is any D_{int} -expansion and θ' is any D_{int} -reduction.

One shall compare our general notion of neighborhood law on B with the one proposed by Freksa in [9]. On the first hand, Freksa represents a transformation law on B_{int} through a single neighborhood graph on B_{int} that corresponds to a symmetric neighborhood relation. On the other hand, from our definition, a transformation law on B induces several neighborhood functions on B , where each one of them describes a specific kind of transformation on D ; moreover, a neighborhood function on B is not necessarily a symmetric neighborhood relation, as it is the case in Freksa's proposal. For instance, one can see from Figure 5 that we have $\tau_\theta^{B_{int}}(eq) = di$ and $\tau_\theta^{B_{int}}(di) \neq eq$. Indeed, for any couple of intervals $x, y \in D_{int}$ such that $(x, y) \in di$, one can never directly get an interval x' that satisfies $(x', y) \in eq$ by a D_{int} -expansion of x (this is due to the "smoothness" of its expansion.) However, one can get back the corresponding neighborhood graph given by Freksa by considering the symmetric closure of the union of the neighborhood functions $\tau_\theta^{B_{int}}, \tau_{\theta'}^{B_{int}}$ forming the neighborhood law $T_{\Theta_3}^{B_{int}}$.

4 Neighborhood and Distance Between Scenarios

The main ingredient of belief change operators in qualitative spatial or temporal reasoning is a *distance* between QCNs, or more precisely a distance between scenarios. Such a notion of distance is itself an interesting tool when one needs to evaluate how far is a scenario from an other one, e.g., in the case of a schedule change. As a case study, let us go back to Example 1 that describe the schedule of an interview, i.e., consider again the scenario σ_1 depicted in Figure 2. Suppose now that we would like to optimize the total time (S), i.e., one would like to leave more time to the discussion phase (D) but still keep a scenario that is "close" to the original one. This "ideal" scenario σ_2 is depicted in Figure 7. We also suppose that the only reasonable way to modify a time interval is to extend it or reduce it, that is, the neighborhood law $T_{\Theta_3} = \{\tau_\theta^{B_{int}}, \tau_{\theta'}^{B_{int}}\}$ on D_{int} is considered. Then, we are asked the following question: "In which measure the initial scenario σ has been changed to obtain the ideal scenario σ' ?" This calls for the definition of a distance between scenarios.

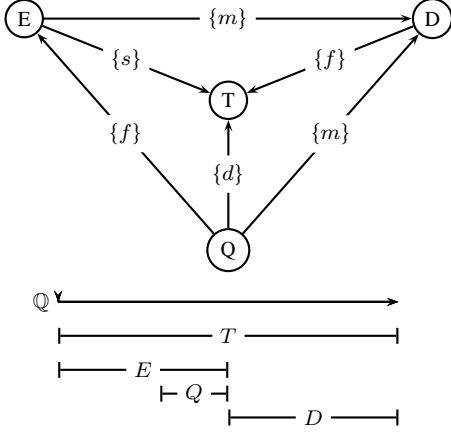


Figure 7. σ_2 , a scenario of N and α_2 , a solution of σ_2 .

In [6, 5] Condotta et al. proposed a notion of distance between scenarios, that we call here a *syntactical distance*. A syntactical distance, denoted $d_{d_B, f}$, depends on an aggregation function f ⁶ and a pseudo-distance d_B^* , called *basic distance*, that is, a pseudo-distance over the set of basic relations B . Then for every pair of scenarios $\sigma, \sigma' \in QCN_V^B$, the syntactical distance between σ and σ' is defined by $d_{d_B, f}(\sigma, \sigma') = f\{d_B(\sigma[i, j], \sigma'[i, j]) \mid v_i, v_j \in V, i < j\}$. Here, the basic distance d_B typically exploits the notion of conceptual neighborhood between basic relations on B [6, 5]. For example, one can consider the basic distance $d_{B_{int}}^*$ such that for every pair of scenarios σ, σ' , $d_{B_{int}}^*(\sigma, \sigma')$ is the length of the shortest chain in the graph that corresponds to the symmetric closure of the union of the neighborhood functions $\tau_{\theta}^{B_{int}}, \tau_{\theta'}^{B_{int}}$ forming the neighborhood law $T_{\Theta_3}^{B_{int}}$ (cf. Figures 3 and 4, see [6, 5] for more details about this basic distance.) When such a basic distance d_B is chosen, the aggregation function f is used to aggregate the distances computed at the constraint level between the two scenarios, and thus get a distance between them. The choice of the aggregation function f depends on the context. For example, for $f = \sum$ [6], the distances d_B on the constraints are simply summed up.

Using a syntactical distance can lead to counter-intuitive results. Consider again the scenarios σ_1 and σ_2 depicted in Figures 2 and 7. Intuitively, σ_1 should be a neighbor of σ_2 w.r.t. the transformation law Θ_3 (cf. example 2) since from any solution of σ_2 , it is sufficient to reduce the interval represented by the variable D to directly get a solution of σ_1 . However, these two scenarios differ on several pairs of variables $((D, S), (D, T))$ and $((D, Q))$. Hence, one can verify in that example that $d_{d_{B_{int}}^*, \sum}(\sigma, \sigma') = 3$. The main problem here is that the transformation of a single entity within a solution of a scenario often leads to a modification of several constraints of this scenario. Indeed, its computation is divided in two steps. First, a "local" distance is computed in a piecewise fashion, independently for each pair of variables. Then, the values computed in the first step are aggregated. Therefore, this distance depends more on the structure of the QCNs than on the qualitative configurations that they represent. Stated otherwise, that distance is more syntactical than semantical.

We propose here a different approach to evaluate how far is a scenario from an other one; our idea exploits an underlying notion of neighborhood between scenarios, that is, a natural extension from the notion of neighborhood between binary relations to the one between sets of instantiations from D . Doing so, the definition of neighborhood between scenarios does not depend on their structure anymore, but on the sets of solutions that they represent. Deciding whether two scenarios σ, σ' defined on 2^B and V are neighbors depends on two parameters: a transformation law θ on D and a variable $v_k \in V$; v_k corresponds to the variable representing the entity that can be modified w.r.t. the transformation law θ on D . Intuitively, for a given transformation θ on D and a variable $v_k \in V$, a scenario σ' is a neighbor of a scenario σ w.r.t. θ and v_k if from a solution α of σ , one can modify the entity represented by v_k through a transformation ruled by θ and thus directly build a solution α' of σ' . Formally:

Definition 7 (neighborhood between scenarios) Let σ, σ' be two scenarios of QCN_V^B , $\theta = \{\mathcal{F}_x \mid x \in D\}$ be a transformation on D and $v_k \in V$. σ' is said to be a neighbor of σ w.r.t. θ and v_k , denoted $\sigma \xrightarrow{\theta, v_k} \sigma'$ if for every solution α of σ , there is a $i \in I_{\alpha(v_k)}$ such that for every $x' \in D_i$, the instantiation $\alpha_{x'}$ defined for every $v_i \in V$ as $\alpha_{x'}(v_i) = x'$ if $v_i = v_k$ and $\alpha_{x'}(v_i) = \alpha(v_i)$ otherwise, is a solution of σ' . Given a transformation law Θ , we say that σ' is a neighbor of σ w.r.t. Θ if there is a transformation law $\theta \in \Theta$ and a variable $v_k \in V$ such that σ' is a neighbor of σ w.r.t. θ and v_k .

As an example, with respect to Definition 7, the scenario σ_1 (cf. Figure 2) is a neighbor of the scenario σ_2 (cf. Figure 7) w.r.t. a D_{int} -reduction and the variable D .

As to the case of the notion of neighborhood between basic relation (cf. Definition 4), given a specific transformation θ on D and a variable v_k , the notion of neighborhood between scenarios induces a particular relation on the set of all scenarios of QCN_V^B , and since the basic relations of B are jointly exhaustive, every scenario of QCN_V^B has at most one neighbor, so this relation of neighborhood between scenarios is functional.

Definition 8 (neighborhood function on QCN_V^B) Let θ be a transformation on D , v_k be a variable. The neighborhood function on QCN_V^B induced by θ and v_k , denoted $\tau_{\theta, v_k}^{QCN_V^B}$, is the function from B to B such that for every $\sigma, \sigma' \in QCN_V^B$, $\tau_{\theta, v_k}^{QCN_V^B}(\sigma) = \sigma'$ if and only if $\sigma \xrightarrow{\theta, v_k} \sigma'$.

Now, given a transformation law on D , a notion of distance over the set of all scenarios of QCN_V^B can naturally be defined:

Definition 9 (semantical distance between scenarios) Let σ, σ' two scenarios of QCN_V^B , Θ be a transformation law on D . The semantical distance between σ and σ' w.r.t. Θ , denoted $d_{\Theta}(\sigma, \sigma')$ is the length of the shortest chain $(\sigma_1, \dots, \sigma_m)$ such that $\sigma_1 = \sigma$, $\sigma_m = \sigma'$, and for every $i \in \{1, \dots, m-1\}$, there is a transformation $\theta \in \Theta$ and a variable v_k such that $\tau_{\theta, v_k}^{QCN_V^B}(\sigma_i) = \sigma_{i+1}$ or $\tau_{\theta, v_k}^{QCN_V^B}(\sigma_{i+1}) = \sigma_i$.

5 Computational Complexity and Experimentations

Definition 7 formalises the notion of neighborhood between scenarios, but it does not provide any algorithmic method to decide whether a given scenario is a neighbor of an other given scenario. Proposition 1 below fills the gap in the case of transformation laws that induce

⁶ An aggregation function is a mapping that associates a vector of non-negative real numbers with a non-negative real number.

⁷ A pseudo-distance d_E on E is a mapping from $E \times E$ to \mathbb{R}^+ that satisfies $\forall e_1, e_2 \in E, d_E(e_1, e_2) = 0$ iff $e_1 = e_2$ and $d_E(e_1, e_2) = d_E(e_2, e_1)$.

left-total⁸ neighborhood functions. We first introduce a preliminary definition:

Definition 10 (componentwise neighborhood between scenarios)

Let σ, σ' be two scenarios of QCN_V^B , θ be a transformation on D and $v_k \in V$. σ' is said to be a componentwise neighbor of σ w.r.t. v_k and θ if for every variable $v_i, v_j \in V \setminus \{v_k\}$, we have $\sigma[i, j] = \sigma'[i, j]$ and $\tau_\theta^B(\sigma[k, i]) = \sigma'[k, i]$. Given a transformation law Θ , we say that σ' is a componentwise neighbor of σ w.r.t. Θ if there is a transformation $\theta \in \Theta$ and a variable $v_k \in V$ such that σ' is a componentwise neighbor of σ w.r.t. θ and v_k .

As an example, it is easy to check that with respect to Definition 10, the scenario σ_1 (cf. Figure 2) is a componentwise neighbor of the scenario σ_2 (cf. Figure 7) w.r.t. a D_{int} -reduction and the variable D . Indeed, let us denote $\sigma_1 = (V, C_1)$ and $\sigma_2 = (V, C_2)$ and let θ be a D_{int} -reduction (see Figure 5); then we have $\tau_\theta^{B_{int}}(C_2(D, E)) = \tau_\theta^{B_{int}}(mi) = pi = C_1(D, E)$, $\tau_\theta^{B_{int}}(C_2(D, T)) = \tau_\theta^{B_{int}}(f) = d = C_1(D, T)$ and $\tau_\theta^{B_{int}}(C_2(D, Q)) = \tau_\theta^{B_{int}}(mi) = pi = C_1(D, Q)$; moreover, all other constraints remained unchanged for σ_1 and σ_2 .

Proposition 1 Let Θ be a transformation law and σ, σ' be two scenarios of QCN_V^B . If $\forall \theta \in \Theta$, τ_θ^B is left-total, then σ' is a neighbor of σ w.r.t. Θ if and only if σ' is a componentwise neighbor of σ w.r.t. Θ .

Let us stress that in Proposition 1, in order to get the characterization of neighborhood between scenarios in terms of componentwise neighborhood between scenarios, the neighborhood function is required to be left-total. Indeed, we may have a relation with no neighbor w.r.t. a transformation θ and at the same time we may find a scenario that is a neighbor of an other scenario w.r.t. θ , as shown in the following example:

Example 5 Let us consider the domain $D = \mathbb{Q}^+$, the basic relations $b_1 = \{(0, q) \mid q \in \mathbb{Q}^{+*}\}$, $b_2 = b_1^{-1}$, $b_3 = \{(q, q') \mid q, q' \in \mathbb{Q}^{+*} \text{ and } q < q'\}$, $b_4 = b_3^{-1}$, $eq = \{(q, q) \mid q \in \mathbb{Q}^+\}$ and the filtration given by $\mathcal{F}_0 = \{\{0\}\}$ and for all $q \in \mathbb{Q}^{+*}$, $\mathcal{F}_q = ([q + \frac{1}{r}, +\infty[\mid q \in \mathbb{Q}^{+*}, r \in \mathbb{N}^+]$. The scenario $X \xrightarrow{eq} Y \xrightarrow{b_3} Z$ is a neighbor of $X \xrightarrow{b_3} Y \xrightarrow{b_3} Z$ w.r.t. Y and the transformation $(\mathcal{F}_q)_{q \in \mathbb{Q}^+}$, however eq is not a neighbor of b_3 for the transformation $(\mathcal{F}_q)_{q \in \mathbb{Q}^+}$ since $\mathcal{F}_0 = \{\{0\}\}$.

In the rest of this section, we implicitly restrict ourselves to neighborhood functions that are left-total and similarly, to transformation laws that induce left-total neighborhood functions.

One can easily see from Definition 10 that deciding whether a scenario is a componentwise neighbor of an other scenario w.r.t. a given transformation law can be done in polynomial time. Thus, as a direct consequence of Proposition 1, deciding whether a scenario is a neighbor of an other scenario w.r.t. a given transformation law can also be done in polynomial time.

Any syntactical distance $d_{d_B, f}$ can be computed in polynomial time, given that f is computed in polynomial time (this is the case for most "usual" aggregation functions.) We are interested in the computational complexity of our new semantical distance. For this purpose, we now consider the following DISTANCE-QCN decision problem: given two scenarios σ, σ' of QCN_V^B , a neighborhood law T_Θ^B induced by a transformation law Θ on D , and a positive integer k , is the semantical distance between σ and σ' w.r.t. Θ equal or less than k ?

⁸ A binary relation R over $F \times F$ is said to be left-total if $(\forall f \in F, \exists f' \in F, (f, f') \in R)$.

Proposition 2 DISTANCE-QCN is NP-hard.

We now propose an algorithmic approach to determine the semantical distance between two scenarios. In Algorithm 1, the function `dsem` computes the semantical distance between two scenarios σ, σ' w.r.t. a given transformation law Θ . `dsem` exploits the function `dk` that decides if the semantical distance between σ and σ' is equal or less than an integer k . In the general case, the number of chains $(\sigma_1, \dots, \sigma_m)$ with $\sigma = \sigma_1$ and $\sigma' = \sigma_m$ is exponential in the number of the considered variables. Thus, we consider a depth-first search algorithm directed by an heuristic criterion with a lookahead strategy.

Algorithm 1: Semantical distance between σ and σ' w.r.t. Θ

```

input: two scenarios  $\sigma = (V, C), \sigma' = (V, C')$ 
input: a transformation law  $\Theta$ 
output: the minimal distance between  $\sigma$  and  $\sigma'$ 
1 Function dsem( $\sigma, \sigma', \Theta$ );
2 begin
3    $k \leftarrow 0$ ;
4   while not (dk( $\sigma, \sigma', \Theta, k$ )) do  $k++$ ;
5   return  $k$ ;
6 Function dk( $\sigma, \sigma', \Theta, k$ );
7 begin
8   if ( $\sigma = \sigma'$ ) then return true;
9   if ( $k = 0$ ) then return false;
10  queue  $\leftarrow$  scenarioPriorityQueue( $\sigma, \Theta$ );
11  while not (isEmpty(queue)) do
12     $\sigma_t \leftarrow$  pop(queue);
13    if dk( $\sigma_t, \sigma', \Theta, k - 1$ ) then return true;
14  return false

```

The efficiency of these functions is essentially due to the function `scenarioPriorityQueue` which aims to schedule the scenarios in $\tau_{\theta, v}^{QCN_V^B}(\sigma)$, with $v \in V$ and $\theta \in \Theta$, in order to quickly reach the goal. The priority is given by an heuristic criterion computed according to the scenarios σ_t and σ' , and the transformations. In our experiments, we define this heuristic as the syntactical distance $d_{d_B, \Sigma}(\sigma, \sigma') = \Sigma\{d_B(\sigma[i, j], \sigma'[i, j], \Theta) \mid v_i, v_j \in V, i < j\}$ where $d_B(\sigma[i, j], \sigma'[i, j])$ is the length of the shortest chain in the graph that corresponds to the symmetric closure of the union of the neighborhood functions forming the neighborhood law T_Θ^B . This heuristic criterion is easy to compute, and it can be used to prune the search tree. Given two neighbor scenarios σ, σ_t and a goal scenario σ' , $|d_{d_B, \Sigma}(\sigma, \sigma') - d_{d_B, \Sigma}(\sigma_t, \sigma')| \leq (n - 1)$. Consequently, if $d_{d_B, \Sigma}(\sigma, \sigma') > (n - 1) \times k$, then σ' can not be reached in k steps.

Some experiments were conducted in order to evaluate our theoretical results. In these experiments, we focused on finding the semantical distance between randomly generated scenarios with 4 variables defined on the interval algebra w.r.t. the transformation law $T_{\Theta_3}^{B_{int}}$, allowing reductions and expansions of the intervals. We considered a set of 1310 evaluations in order to highlight the CPU time to successfully find the semantical distance between two scenarios. The evaluations were done on an Intel Xeon 3GHz processor and 2GB RAM and were limited to 900s for each run. Our application reached the minimal distance by 1007 times over the 1310 runs. In the following results, we only consider those successful evaluations.

Figure 8 illustrates the number of successful evaluations according to the CPU time in seconds. Each point at coordinate (x, y) stands for "y successful runs were produced under x seconds". Thus, half of the semantical distances were computed under 15s and 3 runs over 4 were successfully done under 100s.

An other interest in our experiments is to point out a relation between our semantical distance and the syntactical distance $d_{d_B, \Sigma}$. The graph in Figure 9 shows the proportion between the semantical distance d_Θ and the syntactical distance $d_{d_B, \Sigma}$. There is a circle at (x, y)

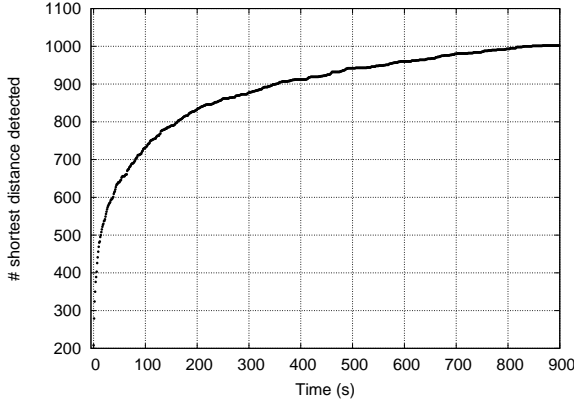


Figure 8. Time versus shortest distance

if we have $d_{\Theta}(\sigma, \sigma') = y$ and $d_{\Sigma, d_B}(\sigma, \sigma') = x$, for at least one instance of the experiments. The size of the circle is proportional to the number of evaluations satisfying this relation. The line through the graph is the average semantical distance according to the syntactical distance. It seems that it could be possible for the semantical distance d_{Θ} to rely on the syntactical distance $d_{d_B, \Sigma}$ in order to bound d_{Θ} and compute the semantical distance more efficiently.

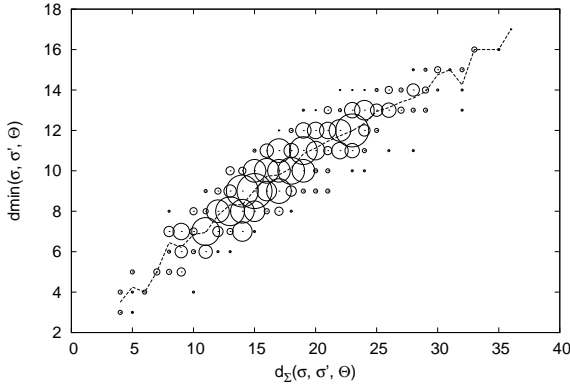


Figure 9. $d_{\Theta}(\sigma, \sigma')$ versus $d_{d_B, \Sigma}(\sigma, \sigma')$

6 Conclusion

A distance between scenarios is the main feature of belief change operators in qualitative spatial and temporal reasoning. In this paper, we first presented a general definition of the notion of *transformation law* for spatial or temporal entities that can be applied to any domain D . This transformation law has led us to define the notion of (directed) neighborhood between basic relations of a qualitative formalism; each one of these neighborhood relations naturally induces a directed conceptual neighborhood graph over the basic relations of this formalism. As an illustration, we imported into our framework a specific transformation law initially proposed by Freksa for the interval algebra. From this last notion, we proposed the definition of conceptual neighborhood between scenarios, and we provided a characterization of this neighborhood that can be checked in polynomial

time. We have derived from it a definition of distance between scenarios, a key ingredient of belief change operators based on QCNs, called *semantical distance* and which is more natural than the *syntactical distances* proposed in the literature. We investigated the computational complexity of the semantical distance between scenarios, which is an *NP-hard* problem in the general case. We have proposed a depth-first search algorithm for this problem and some experiments have been conducted.

As a perspective, it would be interesting to provide a characterization of the neighborhood functions, i.e., give a necessary and sufficient condition to decide whether a function over the set of basic relations B is a neighborhood function. Future works will also include an adaptation of the notion of semantical distance introduced here into other logical formalisms. For example, in the framework of propositional logic, numerous belief change operators are distance-based ones [15] and typically use the Hamming distance [7] between propositional interpretations. Such a distance assumes that the propositional variables are independent. However, when we are faced to integrity constraints represented by a propositional formula (e.g., in the case of IC merging operators [17, 15]) that encodes the physical rules of a system, the Hamming distance is an inappropriate choice [16]. Consequently, following our approach that consists in defining a distance that is “domain-based”, a future work will be to study new distances (in the case of propositional logic as a starting point), more relevant than Hamming distance and that take into account these integrity constraints. Lastly, an ongoing work is to exploit syntactical distances in order to bound the semantical one and make the computation of the semantical distance between scenarios more efficient.

REFERENCES

- [1] J. F. Allen, ‘An interval-based representation of temporal knowledge’, in *Proc. of the 7th International Joint Conference on Artificial Intelligence (IJCAI’81)*, pp. 221–226, (1981).
- [2] J-F. Allen, ‘Towards a general theory of action and time’, *Artificial Intelligence*, **23**(2), 123–154, (1984).
- [3] P. Balbiani and J-F. Condotta, ‘Spatial reasoning about points in a multidimensional setting’, *Applied Intelligence*, **17**(3), 221–238, (2002).
- [4] P. Balbiani and A. Osmani, ‘A model for reasoning about topological relations between cyclic intervals’, in *Proc. of the 7th International Conference on Principles of Knowledge Representation and Reasoning (KR’00)*, pp. 378–385, (2000).
- [5] J-F. Condotta, S. Kaci, P. Marquis, and N. Schwind, ‘A syntactical approach to qualitative constraint networks merging’, in *Proc. of the 17th International Conference on Logic for Programming, Artificial Intelligence and Reasoning (LPAR’10)*, pp. 233–247, Yogyakarta, Indonesia, (2010).
- [6] J-F. Condotta, S. Kaci, and N. Schwind, ‘A Framework for Merging Qualitative Constraints Networks’, in *Proc. of the 21st International FLAIRS Conference (FLAIRS’08)*, pp. 586–591, Miami, FL, USA, (2008).
- [7] M. Dalal, ‘Investigations into a theory of knowledge base revision: preliminary report’, in *Proc. of the 7th National Conference on Artificial Intelligence (AAAI’88)*, pp. 475–479, Seattle, WA, USA, (1988).
- [8] M. J. Egenhofer and K. K. Al-Taha, ‘Reasoning about gradual changes of topological relationships’, in *Spatio-Temporal Reasoning*, pp. 196–219, (1992).
- [9] C. Freksa, ‘Temporal reasoning based on semi-intervals’, *Artificial Intelligence*, **54**(1), 199–227, (1992).
- [10] A. Gerevini and J. Renz, ‘Combining topological and size information for spatial reasoning’, *Artificial Intelligence*, **137**, 1–42, (2002).
- [11] J. M. Gooday and A. G. Cohn, ‘Conceptual neighbourhoods in temporal and spatial reasoning’, in *Proc. of the ECAI Spatial and Temporal Reasoning Workshop*, (1994).
- [12] I. Hamlet and J. Hunter, ‘A representation of time for medical expert systems’, in *Proc. of the European Conference on Artificial Intelligence in Medicine (AIME’87)*, pp. 112–119, (1987).
- [13] R. Hirsch, ‘A finite relation algebra with undecidable network satisfaction problem’, *Logic Journal of the IGPL*, **7**(4), 547–554, (1999).

- [14] H. A. Kautz, *A formal theory of plan recognition*, Ph.D. dissertation, Rochester, NY, USA, 1987.
- [15] S. Konieczny, J. Lang, and P. Marquis, ‘DA² merging operators’, *Artificial Intelligence*, **157**(1-2), 49–79, (2004).
- [16] S. Konieczny, P. Marquis, and N. Schwind, ‘Belief base rationalization for propositional merging’, in *Proc. of the 22nd International Joint Conference on Artificial Intelligence (IJCAI’11)*, pp. 951–956, Barcelona, Spain, (2011).
- [17] S. Konieczny and R. Pino Pérez, ‘Merging information under constraints: a logical framework’, *Journal of Logic and Computation*, **12**(5), 773–808, (2002).
- [18] Y. Kurata, ‘Semi-automated derivation of conceptual neighborhood graphs of topological relations’, in *Proc. of the 9th Conference on Spatial Information Theory (COSIT’09)*, pp. 124–140, Aber Wrach, France, (2009).
- [19] G. Ligozat, ‘Reasoning about cardinal directions’, *Journal of Visual Languages and Computing*, **9**(1), 23–44, (1998).
- [20] G. Ligozat and J. Renz, ‘What is a qualitative calculus? a general framework’, in *Proc. of the 8th Pacific Rim International Conferences on Artificial Intelligence (PRICAI’04)*, pp. 53–64, Auckland, New Zealand, (2004).
- [21] D-A. Randell, Z. Cui, and A. Cohn, ‘A spatial logic based on regions and connection’, in *Proc. of the 3rd International Conference on Principles of Knowledge Representation and Reasoning (KR’91)*, 165–176, (1992).
- [22] J. Renz and G. Ligozat, ‘Weak composition for qualitative spatial and temporal reasoning’, in *Proc. of the 11th International Conference on Principles and Practice of Constraint Programming (CP’05)*, pp. 534–548, (2005).

Appendix: proofs

Proposition 1

Proof:

\Rightarrow Assume that σ' is a neighbor of σ w.r.t. Θ . Then, there is a solution α of σ , a transformation $\theta \in \Theta$ with $\theta = (\mathcal{F}_x)_{x \in D}$, a variable $v_{k_0} \in V$ and $i \in I_{\alpha(v_{k_0})}$ such that $\forall x' \in D_i, \alpha_{x'}$ is a solution of σ' , where for every $v_k \in V, v_k \neq v_{k_0}, \alpha_{x'}(v_k) = \alpha(v_k)$ if $v_k \neq v_{k_0}$ and x' otherwise. Hence, for every $v_k \in V, v_k \neq v_{k_0}$, we have $(\alpha(v_{k_0}), \alpha(v_k)) \in \sigma[k_0, k]$ and for every $x' \in D_i, (x', \alpha(v_k)) \in \sigma'[k_0, k]$. Moreover, since τ_{θ}^B is left-total, there is a $i_1 \in I_{\alpha(v_{k_0})}$ such that for every $x' \in D_{i_1}, (x', \alpha(v_k)) \in \tau_{\theta}^B(\sigma[k_0, k])$. Hence, for every $x' \in D_{\max(i, i_1)}, (x', \alpha(v_k)) \in \sigma'[k_0, k] \cap \tau_{\theta}^B(\sigma[k_0, k])$. Since the basic relations of B are jointly exclusive and pairwise disjoint, $\sigma'[k_0, k] = \tau_{\theta}^B(\sigma[k_0, k])$.

\Leftarrow Assume that σ' is a componentwise neighbor of σ w.r.t. Θ . Then, there is a variable $v_{k_0} \in V$ and a transformation $\theta \in \Theta$ such that for every $v_k, v_p \in V, v_k, v_p \neq v_{k_0}, \sigma'[k, p] = \sigma[k, p]$ and $\sigma'[k_0, k] = \tau_{\theta}^B(\sigma[k_0, k])$. Let α be a solution of σ . We have $(\alpha(v_{k_0}), \alpha(v_k)) \in \sigma[k_0, k]$ and then there is a $i \in I_{\alpha(v_{k_0})}$ such that for every $x' \in D_i, (x', \alpha(v_k)) \in \sigma'[k_0, k]$. For every $x' \in D_i$, let $\alpha_{x'}$ defined for all $v_k \in V$ as $\alpha'(v_k) = \alpha(v_k)$ if $v_k \neq v_{k_0}$ and x' otherwise. We have for every $v_k, v_p \in V, v_k, v_p \neq v_{k_0}, \sigma'[k, p] = \sigma[k, p]$. Thus $(\alpha_{x'}(v_k), \alpha_{x'}(v_p)) = (\alpha(v_k), \alpha(v_p)) \in \sigma[k, p] = \sigma'[k, p]$. For every $v_k \in V, v_k \neq v_{k_0}$, we have $(\alpha_{x'}(v_{k_0}), \alpha_{x'}(v_k)) = (x', \alpha(v_k)) \in \sigma'[k_0, k]$. Hence for every $x \in D_i, \alpha_{x'}$ is solution of σ' . ■

Proposition 2

Proof: We consider the following polynomial reduction from the well-known NP-hard problem VERTEX COVER: given a graph $G = (X, A)$ and a positive integer k , is there a vertex cover of size k or less for G , i.e., a subset $X' \subseteq X$ with $|X'| \leq k$ such that for each edge $(u, v) \in A$, at least one of u and v belongs to X' ? Consider the RCC8 formalism [21] for which the domain D_{RCC8} is the set of all the closed regions of a topological space and the partition scheme B_{RCC8} is a set of 8 topological basic relations. B_{RCC8}

contains two particular basic relations, denoted EC (i.e., the relation “externally connected”), and DC (i.e., the relation “disconnected”). Consider the transformation law $\Theta = \{\theta\}$ such that θ is the transformation on D_{RCC8} that describes the “shrinking” of every region of the space. In this case, the induced neighborhood function on B_{RCC8} is left-total, and in particular we consider we have $EC \xrightarrow{\theta} DC$.

Now, let $G = (X, A)$ be a graph with $X = \{x_1, \dots, x_n\}$ and k be a positive integer. We associate with every node $x_i \in X$ a variable $f(x_i) = v_i$ and denote $\{v_1, \dots, v_n\} = V$. Let $\sigma = (V, C)$ be the scenario defined for every $i, j \in \{1, \dots, n\}$ as $\sigma[i, j] = EC$ if $(x_i, x_j) \in A, DC$ otherwise. Let $\sigma' = (V, C')$ be the scenario of $QCN_V^{B_{RCC8}}$ defined for every $i, j \in \{1, \dots, n\}$ as $\sigma'[i, j] = DC$. We show now that there is a cover of size k or less for G if and only if $d_{\Theta}(\sigma, \sigma') \leq k$.

- Assume first that there is a vertex cover X' of size $k' \leq k$ for G . Then let us rename $X' = \{x'_1, \dots, x'_{k'}\}$ and $V' = \{v'_1, \dots, v'_{k'}\}$ such that $\forall x'_i \in X', f(x'_i) = v'_i$. Now, define the chain $(\sigma^1, \dots, \sigma^{k'})$ of scenarios such that $\sigma^1 = \sigma$ and for every $p \in \{1, \dots, k' - 1\}$, σ^{p+1} is defined $\forall i, j \in \{1, \dots, k'\}$ as $\sigma^{p+1}[i, j] = DC$ if $i, j \neq p$, $\sigma^p[i, j]$ otherwise. First, it is well know that any scenario σ^* of $QCN_V^{B_{RCC8}}$ such that $\forall v_i, v_j \in V, \sigma^*[i, j] \in \{EC, DC\}$ is consistent; thus, $\forall p \in \{1, \dots, k'\}$, σ^p is consistent. Then we can check that for every $p \in \{1, \dots, k' - 1\}$, σ^{p+1} is a componentwise neighbor of σ^p w.r.t. v_p and θ ; moreover, we obtain that $\sigma^{k'} = \sigma'$. This means that $d_{\Theta}(\sigma, \sigma') \leq k$.

- Assume now that there is no vertex cover of size k' for G , with $k' \leq k$. This means that for any subset $X' \subseteq X$ with $|X'| \leq k$ there is $x_i, x_j \in X \setminus X'$ such that $(x_i, x_j) \in A$. Then it is easy to see that for any chain $(\sigma^1, \dots, \sigma^{k'})$ of consistent scenarios of $QCN_V^{B_{RCC8}}$ such that $k' \leq k, \sigma^1 = \sigma$ and such that for every $p \in \{1, \dots, k' - 1\}$, σ^{p+1} is a componentwise neighbor of σ^p w.r.t. v_p and θ , there is $v_i, v_j \in \{1, \dots, n\}$ such that $\sigma^{k'}[i, j] = EC$, i.e., $\sigma^{k'} \neq \sigma'$. This means that $d_{\Theta}(\sigma, \sigma') > k$. ■

Probabilistic Region Connection Calculus

Codruta Girlea and Eyal Amir¹

Abstract. We present a novel probabilistic model and specification language for spatial relations. Qualitative spatial logics such as RCC are used for representation and reasoning about physical entities. Our probabilistic RCC semantics enables a more expressive representation of spatial relations. We observe that reasoning in this new framework can be hard. We address this difficulty by using a factored representation based on Markov Random Fields.

We formally present the syntax and semantics of a probabilistic RCC. We then use Markov Random Fields to represent our models compactly. Using this representation, we show a basic algorithm for answering queries about the probability of a relation to hold between two entities. Finally, we illustrate the effectiveness of the new approach experimentally over a small set of examples.

1 Introduction

We provide a logic for representing and reasoning about spatial elements, in the presence of uncertainty. Our framework combines a high-level approach based on qualitative spatial reasoning, that avoids the pitfalls and complexities of pixel-level reasoning, with a probabilistic semantics, able to deal with and quantify uncertainty.

Reasoning about space at the pixel level requires too complex computations and does not capture higher-level properties of objects. As a solution, higher-level qualitative calculi have been introduced, such as Region Connection Calculus, or RCC [7]; however, in such calculi there are no shades of gray in representing uncertainty. We take the flexible, high-level approach of qualitative spatial reasoning, RCC-8 in particular, and define probabilistic models.

Using our probabilistic spatial calculus, we are able to answer more accurately questions about the relations between regions: in classic RCC, uncertainty with respect to the base relation that holds between two regions means that some base relations are possible. There is no cue as to which of these relation is more likely. In the worst case, the entire base relation is possible. However, generally, in real world situations, some relations might be more probable than others; using our probabilistic calculus, one can find the probabilities for all the base relations between the two regions and then get, rather than a set of relations, the most probable base relation.

An example of an application for our calculus is recreating a spatial landscape, consisting of all spatial relations that hold between all entities, from a natural language description. The landscape description can be analysed to extract an initial set of spatial relations as the first, incomplete, landscape, and then the most likely complete image can be recreated using inference in probabilistic RCC. The techniques used here could be extended to other spatial formalisms, that are able to capture other meaningful relations between entities. Reconstructing a spatial landscape from text can be useful to answering

deeper understanding queries regarding the text. This kind of queries can nowadays be answered in the context of natural language processing by means of textual entailment [9]. Here, either one uses only lexical cues, which can only lead to a shallow understanding of the text, or one learns to infer deeper, semantic relations implied by the text by training on large corpora of annotated textual entailment pairs. In the latter case, much effort is spent on annotating a corpus and feature engineering. By using qualitative spatial reasoning, one only needs to spend effort in extracting the obvious spatial relations from the text, whereas the deeper understanding queries can be answered by reasoning in the underlying spatial logic.

The paper is structured as follows: first, we present some background notions on RCC. Then, we describe the syntax, semantics and inference for our calculus. Next, we present the MRF-based representation and inference. We then show the results on some examples. Finally, we give an overview of related work and conclude.

2 Background

Qualitative Spatial Reasoning [3] is a term used for any relational reasoning technique for which the objects are spatial entities.

Region Connection Calculus (RCC), introduced by Randell, Cui and Cohn in 1992 [7], is a qualitative spatial calculus used to reason about the relations between regions. The distinction between base relations is made based on either connectedness or the mereological 'part of' relation. The two definitions are equivalent, as the two relations can be defined by means of each other. Given the possible distinctions and additional information considered (e.g., whether the region borders are taken into account or not), the space of possible relations is broken into a set of jointly exhaustive and pairwise disjoint, or JEPD, base relations.

For RCC-8, the base relations are:

- disconnected (DC) - the regions are not connected, i.e. they share no common parts;
- externally connected (EC) - the regions are connected, but their interiors are not;
- partially overlap (PO) - the regions' interiors are connected, but there are regions that are part of either one but not the other
- tangential proper part and its inverse (TPP, TPPI) - one region is a part of the other, but that region is not part of the other's interior (equivalently anything that connects to the inner region also connects to the outer region);
- non-tangential proper part and its inverse (NTPP, NTPPI) - one region is a part of the other one's interior;
- equivalent (EQ) - each region is part of the other.

RCC-8 can be formalized as a relation algebra in the sense of Tarski based on the set algebra over 2^B , where B is $\{DC, EC, PO, EQ, TPP, NTPP, TPPI, NTPPI\}$ (the set of base relations).

¹ University of Illinois at Urbana-Champaign, USA, email: girlea2@illinois.edu

To complete the relation algebra, each relation has a converse (TPP is the converse of TPPI, NTPP is the converse of NTPP, the rest of the base relations are each its own converse), and the composition table for base relations is as shown by Wölfl et al. [12]. EQ is the composition identity.

For RCC-5, the border information is not considered, so EC and PO are coalesced into O (overlap), TPP and NTPP are collapsed into PP (proper part), and analogously for their converse relations. Consequently B is $\{DC, O, EQ, PP, PPI\}$ and the relation algebra is changed accordingly.

3 Probabilistic RCC

Let us consider the following image description:

John's office is on the second floor of the building. Andy's office is across the corridor, right next to the service room. There's a tree right beside Andy's office window. Andy is standing by the window. John realized a couple of minutes ago he needed something from the service room, and thought he'd pass by Andy's office on his way there to exchange a few words.

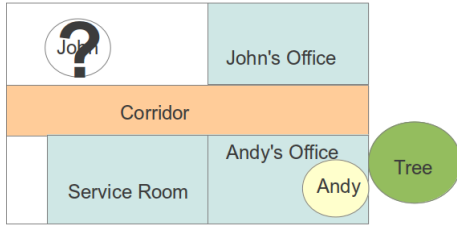


Figure 1. Example of an image

When one reads this description, one builds a mental abstract image (e.g. Figure 1), consisting of spatial relations between entities, and based on this particular image, one can answer questions on what the most likely relative positions of the entities in this world are. Each of these abstract images is similar to a probabilistic RCC model.

In general, in the problem we are trying to solve, we are given a set of regions in a topology, a set of region names or region constants, and a set of spatial constraints on them expressed as a formula. We want to be able to answer queries regarding the probability of certain relations to hold between certain pairs of regions.

3.1 Syntax

In general, the signature of probabilistic RCC is a first-order logic signature of a particular form, containing: a set of constants \mathcal{C} (the region names); and a set of arity 2 relations \mathcal{B} (the base relations).

For RCC-8, $\mathcal{B} = \{DC, EC, PO, EQ, TPP, NTPP, TPPI, NTPPI\}$ and for RCC-5, $\mathcal{B} = \{DC, PO, EQ, PP, PPI\}$.

Two probabilistic RCC-8 (RCC-5) signatures may differ from each other on their set of constants. This leads to the following definition of an RCC-8 signature:

Definition 1 A probabilistic RCC signature is a set of region constants \mathcal{C} .

In the story described above, the signature contains the constants: Andy, John, the corridor, the service room, the tree, and the offices.

Henceforth we will refer to RCC-8 only; the results can easily be applied to RCC-5.

A basic sentence encodes the set of constraints for the problem; this is just a ground FOL sentence. Our example can be encoded as the basic sentence:

$$\begin{aligned} \phi = & TPP(OfficeJohn, Floor) \wedge TPP(Corridor, Floor) \wedge \\ & TPP(OfficeJohn, Floor) \wedge TPP(Corridor, Floor) \wedge \\ & EC(OfficeJohn, Corridor) \wedge \\ & EC(OfficeAndy, Corridor) \wedge \\ & DC(OfficeJohn, OfficeAndy) \wedge \\ & EC(ServiceRoom, Corridor) \wedge \\ & EC(Tree, Floor) \wedge EC(OfficeAndy, ServiceRoom) \wedge \\ & EC(Tree, OfficeAndy) \wedge TPP(John, Floor) \wedge \\ & TPP(Andy, OfficeAndy) \wedge \\ & TPP(OfficeJohn, Floor) \wedge \\ & TPP(OfficeAndy, Floor) \wedge \\ & TPP(ServiceRoom, Floor) \wedge \\ & DC(ServiceRoom, OfficeJohn) \end{aligned} \quad (1)$$

Definition 2 The basic sentences of probabilistic RCC-8 are defined inductively as follows:

- atoms are of the form $r(a, b)$, where $a, b \in \mathcal{C}$ and $r \in \mathcal{B}$;
- if ϕ and ψ are basic RCC-8 sentences, then $\phi \vee \psi$ and $\phi \wedge \psi$ are also basic RCC-8 sentences;
- if ϕ is a basic RCC-8 sentence, then $\neg\phi$ is also a basic RCC-8 sentence

In our example, a query is on the probability of the 'part-of' relation between John and each of the rooms. In general, the queries we want to be able to answer are about the probability of a relation to hold between two regions. This relation may be either a base relation ('externally connected') or a general relation (a disjunction of base relations). In this case, 'part-of' is the disjunction of 'proper part', 'tangential proper part' and 'non-tangential proper part' relations.

One property of PRCC sentences, that stems from JEPD-ness, namely the fact that the negation of a literal can be rewritten as a positive disjunction, is the following:

Property 1 Any basic sentence of probabilistic RCC-8 can be written as a positive sentence

Next, we define query-type sentences. These are the sentences that express probabilities of relations and, as the name implies, will be used to answer queries. A conditional query-type sentence expresses the probability of a relation given a basic type sentence: this is the kind of sentence that generally encodes a full problem. The semantics of these sentences is defined using the semantics of non-conditional query-type sentences.

In the following, α is the probability we are looking for: $p_\alpha(\bigvee_{r \in \mathcal{B}_q} r(a, b))$ has the intuitive meaning that the probability that $r(a, b)$ holds is α .

Definition 3 If $0 \leq \alpha \leq 1$, $a, b \in \mathcal{C}$, $\mathcal{B}_q \subset \mathcal{B}$ and ϕ is a basic sentence, then:

- $p_\alpha(\bigvee_{r \in \mathcal{B}_q} r(a, b))$ is a non-conditional query-type sentence or a query-type atom;
- $p_\alpha(\bigvee_{r \in \mathcal{B}_q} r(a, b) \mid \phi)$ is a conditional query-type sentence.

Definition 4 A probabilistic RCC-8 sentence is either a basic sentence or a query-type sentence.

3.2 Semantics

In our example, a model is any spatial configuration and assignment of names to elements in the spatial configuration, i.e. which room is Andy's office, that satisfies the set of constraints given.

In general, a model of a PRCC signature will specify the topology, a subset of this topology (the 'working' regions), the set of interpretations of region constants in the 'working' region set and a probability distribution on these interpretations.

Let T be a topology on some universe U and let $X \in \mathcal{R}$ be a closed set in T . In the following, let $Int(X)$ be the interior of X and $\Gamma(X) = X - Int(X)$ be the border of X .

Definition 5 Given an RCC-8 signature \mathcal{C} , a model M of the signature is a structure of the form $M = (U, T, \mathcal{R}, W, P)$, where:

- U is a (possibly infinite) universe of points;
 - T is a topology on U ; the closed regular sets in T are called regions;
 - $\mathcal{R} \subset T$ is a finite set of regions;
 - $W = \{(U_w, w) \mid w : \mathcal{C} \uplus \mathcal{B} \rightarrow U_w \uplus (U_w \times U_w)\}$ is a set of possible worlds, where for each possible world w :
 - $U_w = \mathcal{R}$ is the world universe;
 - $w|_{\mathcal{C}} : \mathcal{C} \rightarrow U_w$ is an interpretation of constant symbols as regions;
 - $w|_{\mathcal{B}} : \mathcal{B} \rightarrow U_w \times U_w$ is an interpretation of base relation symbols
- and the interpretation of base relation symbols $w|_{\mathcal{B}}$ is such that:
- $\forall X, Y \in U_w, w(DC)(X, Y) \text{ iff } X \cap Y = \emptyset$;
 - $\forall X, Y \in U_w, w(EC)(X, Y) \text{ iff } Int(X) \cap Int(Y) = \emptyset \text{ and } X \cap Y \neq \emptyset$;
 - $\forall X, Y \in U_w, w(PO)(X, Y) \text{ iff } Int(X) \cap Int(Y) \neq \emptyset \text{ and } X \not\subseteq Y \text{ and } Y \not\subseteq X$;
 - $\forall X, Y \in U_w, w(EQ)(X, Y) \text{ iff } X = Y$;
 - $\forall X, Y \in U_w, w(TPP)(X, Y) \text{ iff } X \subsetneq Y \text{ and } X \not\subseteq Int(Y)$;
 - $\forall X, Y \in U_w, w(TPPI)(X, Y) \text{ iff } w(TPP)(Y, X)$;
 - $\forall X, Y \in U_w, w(NTPP)(X, Y) \text{ iff } X \subseteq Int(Y)$;
 - $\forall X, Y \in U_w, w(NTPPI)(X, Y) \text{ iff } w(NTPP)(Y, X)$.
- $P : W \rightarrow [0, 1]$ (with $\sum_{w \in W} P(w) = 1$) is a probability distribution over the set of interpretations.

These properties of interpretation functions also ensure that the set $w(\mathcal{B})$ forms a partition over $U_w \times U_w$, or in other words the relations in $w(\mathcal{B})$ are jointly exhaustive and pairwise disjoint (JEPD).

In the rest of the paper, we will assume the topological space of the model fixed. The interpretation of base relations in this space will be the same for all models so we will omit both the topology and the interpretation of relations from the definition of a model as implied. Moreover, for all models we will have the set of interpretations to be the entire set of functions from \mathcal{C} to \mathcal{R} , so W will be completely defined by \mathcal{R} and can thus be omitted as well (restrictions to a subset of this \mathbb{I} can be made by forcing the probability of the missing interpretation functions to 0).

For basic sentences, sentence satisfaction is defined for every possible world, inductively on the structure of the sentence, as in any fragment of FOL. A sentence is satisfied if it is satisfied in every world that has a non-zero probability.

Definition 6 Given model $M = (\mathcal{R}, W, P)$, the satisfaction of a basic formula in a possible world $w \in W$ is defined inductively as:

- $w \models r(a, b) \text{ iff } (w(a), w(b)) \in w(r)$;
- $w \models \phi \wedge \psi \text{ iff } w \models \phi \text{ and } w \models \psi$;
- $w \models \neg \phi \text{ iff } w \not\models \phi$;
- $w \models \phi \vee \psi \text{ iff } w \models \neg(\neg \phi \wedge \neg \psi)$;

We say a model $M = (\mathcal{R}, W, P)$ satisfies a basic formula ϕ and write $M \models \phi \text{ iff } w \models \phi \text{ for all } w \in W \text{ with } P(w) > 0$.

Next, we will show how to answer queries, given a model and a set of constraints. The intuition is that, when we are presented with a new piece of information about the world, we constrain our model of the world so as to discard all interpretations that are not consistent with the new piece of information. The model we end up with is what we will call the restriction of a model via a basic-type sentence. Restricting the model via a sentence lowers to 0 the probabilities of all the interpretations that do not satisfy the sentence, and scales the other probabilities such that they still sum to 1.

Definition 7 Let ϕ be a basic formula and $M = (\mathcal{R}, W, P)$ a probabilistic RCC-8 model; then we can define the restriction of M via ϕ as $M|_{\phi} = (\mathcal{R}, W, P|_{\phi})$, where:

- $P|_{\phi}(w) = P(w) \cdot \frac{1}{Z(\phi)}$ if $w \models \phi$;
- $P|_{\phi}(w) = 0$ if $w \not\models \phi$

and $Z(\phi) = \sum_{w \models \phi} P(w)$ is the normalization constant.

Thus $M|_{\phi}$ is intuitively the largest submodel of M that satisfies ϕ .

In order to answer the query given a set of constraints, we restrict the model in order for it to satisfy the set of constraints, and then we sum the probabilities of the interpretations that satisfy the query. So, the satisfaction of a query-type sentence by a model M is defined as follows:

Definition 8 Given model $M = (\mathcal{R}, W, P)$, basic sentence ϕ , $a, b \in \mathcal{C}$, $\mathcal{B}_q \subset \mathcal{B}$ and $0 \leq \alpha \leq 1$, the satisfaction of query-type sentence $p_{\alpha}(\bigvee_{r \in \mathcal{B}_q} r(a, b) \mid \phi)$ is defined as:

- $M \models p_{\alpha}(\bigvee_{r \in \mathcal{B}_q} r(a, b)) \text{ iff } \sum_{w \models \bigvee_{r \in \mathcal{B}_q} r(a, b)} P(w) = \alpha$;
- $M \models p_{\alpha}(\bigvee_{r \in \mathcal{B}_q} r(a, b) \mid \phi) \text{ iff } M|_{\phi} \models p_{\alpha}(\bigvee_{r \in \mathcal{B}_q} r(a, b))$.

It is worth noting that we are really not interested in what exactly the interpretations of constant symbols in a possible world look like, but in their relative position. So we can restrict our attention to equivalence classes of possible worlds, under the equivalence relation \simeq given by the set of base RCC relations that hold in these worlds:

$$\begin{aligned} w_1 \simeq w_2 \quad \text{iff} \quad & \text{for each pair } a, b \in \mathcal{C} \text{ and for each } r \in \mathcal{B} \\ & w_1 \models r(a, b) \Leftrightarrow w_2 \models r(a, b) \end{aligned} \quad (2)$$

This will be particularly useful when introducing the factored representation.

3.3 Inference in Probabilistic RCC

Using definitions 6 and 8, we can derive the following alternative condition for the satisfaction of conditional query-type sentences - $M \models p_{\alpha}(\bigvee_{r \in \mathcal{B}_q} r(a, b) \mid \phi)$:

$$\alpha = \frac{\sum_{w \models \phi \text{ and } w \models \bigvee_{r \in \mathcal{B}_q} r(a, b)} P(w)}{\sum_{w \models \phi} P(w)} \quad (3)$$

It is straightforward to implement an algorithm that finds α using this formula. If N is the size of ϕ , R is the number of regions and C is the number of constant symbols in the signature, this algorithm would require $O(R^{C+1})$ space and $O(N \cdot R^C)$ time.

Notice that this algorithm requires us to know P , the probability distribution over possible worlds. If we don't know it, the proper probability distribution to use is the one with the maximum entropy, according to the principle of maximum entropy. The set of possible worlds being a discrete and finite domain, the maximum entropy probability distribution is the uniform probability distribution.

Using this observation and the equation (3) derived in the beginning of the previous section, we can compute α in $p_\alpha(\bigvee_{r \in \mathcal{B}_q} r(a, b) \mid \phi)$ as:

$$\alpha = \frac{|\{w \in W \mid w \models \phi \text{ and } w \models \bigvee_{r \in \mathcal{B}_q} r(a, b)\}|}{|\{w \in W \mid w \models \phi\}|} \quad (4)$$

4 Factored Representation of PRCC

Given a signature \mathcal{C} and model $M = (\mathcal{R}, W, P)$, for each pair of distinct constant symbols $a, b \in \mathcal{C}$, let $X_B^{a,b}$ be the random variable encoding the base relation that holds between the regions named by a and b . Then, the probability distribution P over possible worlds induces a joint probability P_B distribution over $\{X_B^{a,b}\}_{a,b \in \mathcal{C}, a \neq b}$:

$$P_B(X_B^{p_1} = r_1, \dots, X_B^{p_N} = r_N) = \sum_{w \models \bigwedge_{1 \leq i \leq N} r_i(p_i)} P(w) \quad (5)$$

where $N = \binom{\mathcal{C}}{2}$, $\{p_1, \dots, p_N\} = \{\{a, b\} \in \mathcal{C} \mid a \neq b\}$ and $r_i \in \mathcal{B}$ for $1 \leq i \leq N$.

If we consider the model consisting of equivalence classes of possible worlds, we can recover the probability distribution over these equivalence classes from the joint probability P_B , as:

$$P([w]_{r_1(p_1), \dots, r_N(p_N)}) = P_B(X_B^{p_1} = r_1, \dots, X_B^{p_N} = r_N) \quad (6)$$

where $[w]_{r_1(p_1), \dots, r_N(p_N)} = \{w \in W \mid w \models r_1(p_1) \wedge \dots \wedge r_N(p_N)\}$.

Therefore, reasoning in probabilistic RCC can be reduced to reasoning with such joint probability distributions:

Theorem 1 *Given model $M = (\mathcal{R}, W, P)$, basic sentence ϕ , expressed as a conjunction of atoms, $a, b \in \mathcal{C}$, $r \in \mathcal{B}$ and $0 \leq \alpha \leq 1$, the satisfaction of query-type sentence $p_\alpha(r(a, b) \mid \phi)$ can be computed as follows:*

- $M \models p_\alpha(\bigvee_{r \in \mathcal{B}_q} r(a, b))$ iff $P_B(X_B^{a,b} = r) = \alpha$;
- $M \models p_\alpha(\bigvee_{r \in \mathcal{B}_q} r(a, b) \mid \phi)$ iff $P_B(X_B^{a,b} = r \mid \phi) = \alpha$;

Furthermore, since for every $X_B^{a,b}$, the events $X_B^{a,b} = r$ and $X_B^{a,b} = r'$ are disjoint for every $r \neq r' \in \mathcal{B}$:

Corollary 1 *Given model $M = (\mathcal{R}, W, P)$, basic sentence ϕ , expressed as a conjunction of atoms, $a, b \in \mathcal{C}$, $\mathcal{B}_q \subset \mathcal{B}$ and $0 \leq \alpha \leq 1$, the satisfaction of query-type sentence $p_\alpha(\bigvee_{r \in \mathcal{B}_q} r(a, b) \mid \phi)$ can be computed as follows:*

- $M \models p_\alpha(\bigvee_{r \in \mathcal{B}_q} r(a, b))$ iff $\sum_{r \in \mathcal{B}_q} P_B(X_B^{a,b} = r) = \alpha$;
- $M \models p_\alpha(\bigvee_{r \in \mathcal{B}_q} r(a, b) \mid \phi)$ iff $\sum_{r \in \mathcal{B}_q} P_B(X_B^{a,b} = r \mid \phi) = \alpha$.

4.1 Markov Random Fields

A Markov random field (MRF) is a compact representation of a joint probability distribution by means of an undirected graph describing conditional independence. More specifically, given a joint probability distribution P over random variables X_1, X_2, \dots, X_M , an MRF consists of the following:

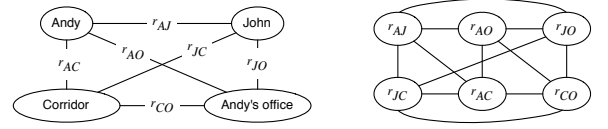


Figure 2. Compact representation of a model

- an undirected graph with vertices X_1, X_2, \dots, X_M , such that:
 - any two non-adjacent random variables are conditionally independent given all the others (*pairwise Markov property*)
 - a random variable is conditionally independent of all the others given its neighbours (*local Markov property*)
 - any two sets of random variables are conditionally independent given a third set that separates any path between the two (*global Markov property*)
- a set of factors ϕ_k associated to the cliques of the graph (over which k ranges). The joint probability distribution is then:

$$P(X_1, X_2, \dots, X_M) = \frac{1}{Z} \prod_k \phi_k(\bar{X}_k) \quad (7)$$

where \bar{X}_k is the set of variables in clique k and Z is a normalization factor called the *partition function*

Going back to PRCC, we can represent the joint probability distribution P_B in a compact way, as an MRF, using the observation that the base relation that holds between two regions named a and b is independent of any other relation that holds in the world, given the relations that hold between region named a and any other region, and the relations that hold between any other region and region named b , and these relations' duals:

$$I(X_B^{a,b}, X_B^{c_i, c_j} \mid X_B^{a,c} \cup X_B^{b,c} \cup X_B^{c,a} \cup X_B^{c,b}) \quad (8)$$

where $X_B^{a,c} = \{X_B^{a,c} \mid c \in \mathcal{C}\}$ and likewise $X_B^{c,a} = \{X_B^{c,a} \mid c \in \mathcal{C}\}$. For example, if we know where John is with respect to Andy, all the rooms on the current floor, and the current floor, and all the spatial relations that hold between the tree and Andy, the floor and all the rooms on the floor, then we don't need to know what spatial relation holds between the service room and the corridor in order to find the relation that holds between John and the tree.

This observation does not hold for all PRCC models, and we will only be able to use this compact representation for those models that do have this property. Intuitively, this is the case if we don't have any prior knowledge of the space of regions \mathcal{R} , and indeed in this case, using the naïve algorithm described in the previous section is infeasible. So we will require, in the rest of the section, that the following property holds for the models for which we give the compact representation:

Property 2 *Let $M = (\mathcal{R}, W, P)$ be a PRCC model. We say that M is unstructured if:*

- P is the uniform probability distribution, and
- for every sentence ϕ (on the same signature), if ϕ is satisfiable in RCC-8, then there is $w \in W$ such that $w \models \phi$

The MRF representation for unstructured models is illustrated on a simple example in Figure 2, for the case where we restrict our attention to the subsignature consisting of only Andy, John, the corridor

and Andy's office. Notice that if (8) holds, then we only have edges between nodes that share a symbol:

Lemma 1 *If Property 2 holds for a model M , then in the MRF representation of M , there is an edge between nodes X_B^p and X_B^q if pairs p, q share a constant symbol, i.e. $|p \cap q| = 1$.*

This lemma leads to the following theorem:

Theorem 2 *Let \mathcal{C} be a probabilistic RCC signature and let $P_B(X_B^{p_1} = r_1, \dots, X_B^{p_N} = r_N)$ be the probability distribution over the base relations that hold between the interpretations of every two constant regions, given an unstructured model M . Then, in the MRF representation of P_B , the largest clique has size $C - 1$.*

Intuitively, every node $X_B^{a,b}$ ($a \neq b \in \mathcal{C}$) in the MRF representation is connected to two cliques of size $C - 1$: one containing all the pairs that share symbol a , and one containing all the pairs that share symbol b . Other cliques that appear in the MRF are triangles representing the relations that hold between any three regions. The interactions represented by those latter cliques stem from the constraints imposed by RCC relation composition.

Let $\bar{X}_B^{a,?} = (X_B^{a,b})_{b \neq a, b \in \mathcal{C}}$ be the tuple containing the nodes in the clique sharing symbol a , and let $X_B^{a,b,c} = X_B^{a,b}, X_B^{b,c}, X_B^{c,a}$. For an unstructured model, one can have any combination of base relations between a region and all the other regions, i.e., we can assume $\phi(\bar{X}_B^{a,?})$ a constant and therefore ignore it in the factorization. Therefore the probability distribution can be written as:

$$P_B(X_B^{p_1} = r_1, \dots, X_B^{p_N} = r_N) = \frac{1}{Z_B} \prod \phi_{a,b,c}(X_B^{a,b,c}) \quad (9)$$

4.2 Inference in the Factored Models

In the following we will assume we know the factors $\phi_{a,b,c}(X_B^{a,b}, X_B^{b,c}, X_B^{c,a})$ in the joint probability distribution. We can infer the probability α of $\forall r \in \mathcal{B}_q r(a, b)$ as the sum of probabilities of each $r(a, b)$, given an evidence $\phi = r_1(a_1, b_1) \wedge \dots \wedge r_k(a_k, b_k)$:

$$\begin{aligned} \alpha &= \sum_{r \in \mathcal{B}_q} P(r(a, b) \mid \phi) \\ &= \frac{\sum_{r \in \mathcal{B}_q} P(r(a, b), r_1(a_1, b_1), \dots, r_k(a_k, b_k))}{\sum_{r \in \mathcal{B}} P(r(a, b), r_1(a_1, b_1), \dots, r_k(a_k, b_k))} \quad (10) \end{aligned}$$

using any inference method in the corresponding MRF.

If we further assume $\phi_{a,b,c}(X_B^{a,b}, X_B^{b,c}, X_B^{c,a}) = w_{a,b,c} f_{a,b,c}(X_B^{a,b}, X_B^{b,c}, X_B^{c,a})$, where the value of the feature $f_{a,b,c}$ is 1 if the configuration specified by the relations between a , b and c is possible and 0 otherwise, we can use any MRF learning algorithm to infer the set of weights $\{w_{a,b,c}\}_{a,b,c}$. We intend to explore this direction in the future, for the current work we assume that all the factors are known, or all the weights are 1.

Note that, although all the cliques have size at most $C - 1$, variable elimination can lead to factors of greater size. Since every node links two cliques of size $C - 1$, eliminating the first variable produces a clique of size $2(C - 2)$. Eliminating further variables increases by at least $C - 2$ the size of the largest clique, therefore in the course of running the algorithm, the largest clique may reach size $O(N \cdot (C - 2))$. Since $N = \binom{C}{2}$, we get the following theorem:

Theorem 3 *Variable Elimination for the factored model of PRCC has a time complexity of $O(2^{C^3})$.*

Approximate inference methods such as loopy belief propagation or sampling are beyond the scope of this paper.

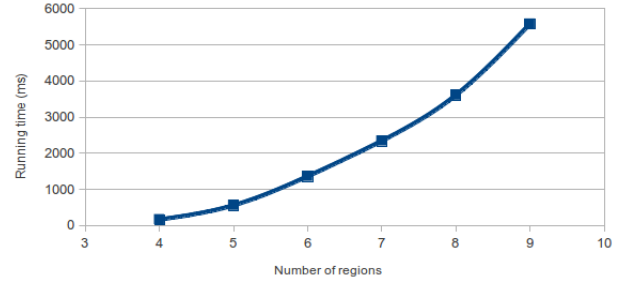


Figure 3. The running time of answering a query as a function of the number of region symbols in the signature

5 Results on Some Examples

We implemented our framework, using the MRF representation for the joint probability distribution. We used variable elimination as the inference algorithm. We experimented with answering queries on our running example (Figure 1) with slight modifications. Figure 3 shows the running time as a function of the number of regions, and Figure 4 shows the dependence on the number of constraints.

We got that John is most likely to be in his office, with probability 0.226, and that Andy is standing by the tree with probability 0.38.

We also investigated a simple story, where John is in his office, and Andy enters the office (Andy partially overlaps both the office and the corridor). In this case, Andy meets John (the disjunction of base relations EC and PO) with probability 0.435.

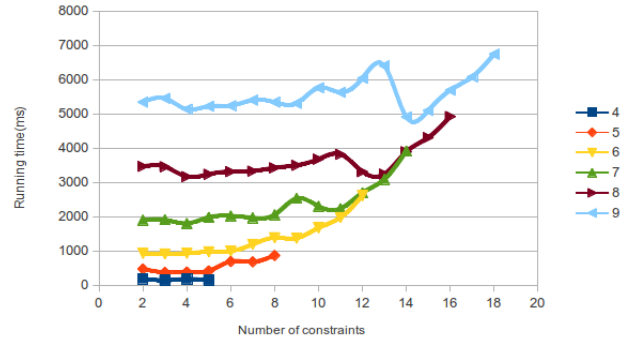


Figure 4. The running time of answering a query as a function of the number of atoms in the evidence, for different numbers of region constants

6 Related Work

Another way to do probabilistic reasoning in RCC is to use the language of Markov Logic Networks [8]. This amounts to representing PRCC as an MLN built from an axiomatization of classic RCC, such as the original axiomatization [7]. All constraints imposed by the axiomatization are considered hard, therefore the sentences in the MLN will have infinite weight. What we do here is to give probabilistic RCC an individuality of its own, with its own well-defined syntax and semantics. Furthermore, we encode the PRCC models directly as Markov Random Fields, taking advantage of the particular independence assumptions that stem from the spatial domain.

We will further discuss other related approaches to represent and reason about uncertainty in Region Connection Calculus.

Cohn et al. [2] address the problem of reasoning in the presence of vague topological information, more specifically in the case where the regions have vague boundaries. They introduce the 'egg-yolk' representation, where each region is divided into its crisp, certain subregion (the 'yolk') and a surrounding vague part (the 'white'). The intuition is that the actual region lies anywhere within the borders of the 'white' and necessarily covers the 'yolk'. In this work there is no quantification for the degree of uncertainty.

Schockaert et al. [11] [10] also deal with vague regions and add quantifiable uncertainty. Rather than work in a probabilistic setting, as in our approach, or divide each region, as in the previous approach, they develop a framework based on fuzzy logic. They take the 'connected' relation to mean the degree to which regions are connected, not a crisp truth value as in the classical RCC. With this, they redefine the entire set of base relations of RCC and subsequently the RCC framework. In contrast, we keep the classic logical framework of RCC and give it a probabilistic semantics.

In order to deal with uncertainty regarding regions, Bittner et al. [1] represent approximate regions by relating them to a frame of reference consisting of a set of unit regions. The definition of approximation makes qualitative distinctions based on the coverage of those unit regions. They then define an approximate region as a set of regions with the same approximation. With this definition, they rewrite the RCC framework to work with approximate regions.

All of these lines of work look at dealing with or quantifying vagueness rather than quantifying the likelihood of relations.

Probabilistic logic programming, or PLP [6], resembles our work mainly in the way they define the satisfaction of probabilistic sentences. One major difference is that, in PLP, each probabilistic formula (representing the probability of a conditional event) is assigned a probability *interval* - we are reasoning over single probabilities, not probability intervals. Another important difference is that any sentence in PLP is a probabilistic sentence; in our case, only the queries are probabilistic, whereas the knowledge base consists only of sentences expressed in classic RCC.

A maximum entropy semantics has also been defined for PLP [13]; that definition is based on the notion of degree of satisfaction. Since we do not use probabilistic sentences in our knowledge base, our maximum entropy model is much more simple.

Kontchakov et al. [5] have proved the sensitivity to the underlying topological space of the complexity of reasoning in a superset of RCC-8, enriched with a unary connectedness predicate and Boolean functions over regions. They also prove NP-completeness of satisfiability for the calculus enriched with connectedness only, as well as EXPTIME-hardness of the full superset. Further results [4] prove reasoning in the 2D Euclidean space RE-hard for the case when Boolean functions are allowed over regions. We do not make assumptions on the underlying topological space, and we do not talk about Boolean functions over regions.

7 Conclusions and Further Work

We showed the syntax and semantics of probabilistic RCC-8. We showed how to represent the models of this calculus compactly, by using Markov Random Fields to model the joint probability distribution over spatial relations. We then used this framework to answer queries regarding the probabilities of relations between regions, given a set of constraints, on a small set of examples.

One problem we don't address is how to handle disjunctive evidence. One way to look at this, is that, writing the evidence in disjunctive normal form, every clause serves as evidence for a possible

abstract image of the world. We would then want to combine the probabilities of base relations that result from each of these possible images. One could take an optimistic approach and use the maximum of these probabilities, for every query, but this does not accurately represent the probability distribution encoded by the model. We will explore ways to look at disjunctive evidence in future work.

As another line of future work, we want to explore more efficient algorithms for the compact representation, possibly sampling, and experiment with real world examples, including learning the weights from a larger description of the world. We would also like to explore the idea of allowing quantification over PRCC sentences and using types of regions to derive meaningful and more general representations. We believe the approach holds promise for recreating a spatial scene from a natural language description, so another line of future work would be exploring this possibility.

ACKNOWLEDGEMENTS

This work was supported by NSF IIS grants 09-17123, 09-68552 and grant NSF EAR 09-43627 EA.

REFERENCES

- [1] Thomas Bittner and John G. Stell, 'Approximate qualitative spatial reasoning', *Spatial Cognition and Computation*, **2**, 435–466, (May 2001).
- [2] A.G. Cohn and N.M. Gotts, 'The 'egg-yolk' representation of regions with indeterminate boundaries', pp. 171–187. Francis Taylor, (1995).
- [3] Anthony G. Cohn, 'Qualitative spatial representation and reasoning techniques', in *Proceedings of the 21st Annual German Conference on Artificial Intelligence: Advances in Artificial Intelligence*, pp. 1–30, London, UK, (1997). Springer-Verlag.
- [4] Roman Kontchakov, Yavor Nenov, Ian Pratt-Hartmann, and Michael Zakharyashev, 'On the decidability of connectedness constraints in 2d and 3d euclidean spaces', *CoRR*, **abs/1104.0219**, (2011).
- [5] Roman Kontchakov, Ian Pratt-Hartmann, and Michael Zakharyashev, 'Interpreting topological logics over euclidean spaces.', in *KR*, eds., Fangzhen Lin, Ulrike Sattler, and Mirosław Truszczyński. AAAI Press, (2010).
- [6] Thomas Lukasiewicz, 'Probabilistic logic programming', in *In Proc. of the 13th European Conf. on Artificial Intelligence (ECAI-98)*, pp. 388–392. J. Wiley & Sons, (1998).
- [7] David A. Randell, Zhan Cui, and Anthony Cohn, 'A Spatial Logic Based on Regions and Connection', in *KR'92. Principles of Knowledge Representation and Reasoning: Proceedings of the Third International Conference*, eds., Bernhard Nebel, Charles Rich, and William Swartout, 165–176, Morgan Kaufmann, San Mateo, California, (1992).
- [8] Matthew Richardson and Pedro Domingos, 'Markov logic networks', *Mach. Learn.*, **62**, 107–136, (February 2006).
- [9] M. Sammons, V. Vydiswaran, and D. Roth, 'Ask not what textual entailment can do for you...', in *ACL*, Uppsala, Sweden, (7 2010). Association for Computational Linguistics.
- [10] Steven Schockaert, Martine De Cock, Chris Cornelis, and Etienne E. Kerre, 'Fuzzy region connection calculus: Representing vague topological information', *International Journal of Approximate Reasoning*, **48**(1), 314 – 331, (2008). Special Section: Perception Based Data Mining and Decision Support Systems.
- [11] Steven Schockaert and Martine De Cock, 'Reasoning about vague topological information', in *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management, CIKM '07*, pp. 593–602, New York, NY, USA, (2007). ACM.
- [12] Stefan Wöflf, Till Mossakowski, and Lutz Schröder, 'Qualitative constraint calculi: Heterogeneous verification of composition tables.', in *FLAIRS Conference*, eds., David Wilson and Geoff Sutcliffe, pp. 665–. AAAI Press, (2007).
- [13] Anbu Yue, Weiru Liu, and Anthony Hunter, 'Measuring the ignorance and degree of satisfaction for answering queries in imprecise probabilistic logic programs', in *Proceedings of the 2nd international conference on Scalable Uncertainty Management, SUM '08*, pp. 386–400, Berlin, Heidelberg, (2008). Springer-Verlag.

Proceedings

Spatio-Temporal Dynamics (STeDy 2012). Editors: Mehul Bhatt, Hans Guesgen, and Ernest Davis. Workshop Proceedings of the European Conference on Artificial Intelligence (ECAI 2012), Montpellier, France.

Also published online as part of Report Series of the Transregional Collaborative Research Center SFB/TR 8 Spatial Cognition, Universität Bremen / Universität Freiburg. SFB/TR 8 Reports, Bremen, Germany.