Dissertation zur Erlangung des Doktorgrades der Naturwissenschaften
— Dr. rer. nat. —

Qualitative Spatial and Temporal Reasoning based on And/Or Linear Programming

An approach to partially grounded qualitative spatial reasoning

Arne Kreutzmann

Bremen, August 2014

Universität Bremen

Die vorliegende Doktorarbeit wurde in der Zeit von Januar 2010 bis August 2014 in der Arbeitsgruppe Cognitive Systemes an der Universität Bremen im Rahmen des SFB/TR-8 Raumkognition angefertigt.

1. Gutachter:	Prof. Christian Freksa, PhD
2. Gutachter:	Prof. Alessandro Saffiotti, PhD

To my little family:
Anne-Christin and Finn

Summary

Acting intelligently in dynamic environments involves anticipating surrounding processes, for example to foresee a dangerous situation or acceptable social behavior. Knowledge about spatial configurations and how they develop over time enables intelligent robots to safely navigate by reasoning about possible actions. The seamless connection of high-level deliberative processes to perception and action selection remains a challenge though. Moreover, an integration should allow the robot to build awareness of these processes as in reality there will be misunderstandings a robot should be able to respond to. My aim is to verify that actions selected by the robot do not violate navigation or safety regulations and thereby endanger the robot or others. Navigation rules specified qualitatively allow an autonomous agent to consistently combine all rules applicable in a context. Within this thesis, I develop a formal, symbolic representation of right-of-way-rules based on a qualitative spatial representation.

This cumulative dissertation consists of 5 peer-reviewed papers and 1 manuscript under review. The contribution of this thesis is an approach to represent navigation patterns based on qualitative spatio-temporal representation and the development of corresponding effective sound reasoning techniques. The approach is based on a spatial logic in the sense of Aiello, Pratt-Hartmann, and van Benthem. This logic has clear spatial and temporal semantics and I demonstrate how it allows various navigation rules and social conventions to be represented.

I demonstrate the applicability of the developed method in three different areas, an autonomous robotic system in an industrial setting, an autonomous sailing boat, and a robot that should act politely by adhering to social conventions. In all three settings, the navigation behavior is specified by logic formulas. Temporal reasoning is performed via model checking. An important aspect is that a logic symbol, such as *turn left*, comprises a family of movement behaviors rather than a single pre-specified movement command. This enables to incorporate the current spatial context, the possible changing kinematics of the robotic system, and so on without changing a single formula. Additionally, I show that the developed approach can be integrated into various robotic software architectures.

Further, an answer to three long standing questions in the field of qualitative spatial reasoning is presented. Using generalized linear programming as a unifying basis for reasoning, one can jointly reason about relations from different qualitative calculi. Also, concrete entities (fixed points, regions fixed in shape and/or position, etc.) can be mixed with free variables. In addition, a realization of qualitative spatial description can be calculated, i.e., a specific instance/example. All three features are important for applications but cannot be handled by other techniques. I advocate the use of And/Or trees to facilitate efficient reasoning and I show the feasibility of my approach. Last but not least, I investigate a fourth question, how to integrate And/Or trees with linear temporal logic, to enable spatio-temporal reasoning.

Zusammenfassung

Eine Vorraussetzung für intelligentes Navigieren in dynamischen Umgebungen ist, dass die dort stattfindenden Prozesse sowie das Verhalten Anderer nicht nur wahrgenommen sondern auch verlässlich vorausgesagt werden können. Dieses ist insbesondere bei der Erkennung potenziell gefährlicher Situationen entscheidend, aber ebenfalls wichtig, wenn ein angemessenes soziales Verhalten eines intelligenten Roboters gefordert ist. Damit ein intelligenter Roboter über seine Navigationsoptionen schlussfolgern kann, benötigt er Kenntnis darüber, wie sich räumliche Konfigurationen über die Zeit verändern.

Ein Ziel dieser Arbeit ist es, Methoden zu entwickeln, die sicherstellen, dass ein Robotersystem diesen Anforderungen genügt und so beispielsweise nicht gegen Sicherheitsbestimmungen oder Verkehrsregeln verstößt. Da Verkehrsregeln qualitativ beschrieben werden, ist es möglich alle Regeln, die in einem gegebenen Kontext Anwendung finden, gleichzeitig zu berücksichtigen. Im Rahmen dieser Arbeit entwickle ich eine formale, symbolische Darstellung von Vorfahrtsregeln auf der Grundlage einer qualitativen räumlichen Darstellung. Eine weitere Herausforderung bei der Implementierung eines solchen Systems, mit der sich die Arbeit auseinandersetzt, besteht in der nahtlosen Verknüpfung von geplantem Navigationsverhalten und den Sensordaten bzw. den Motorsteuerungs-Befehlen. Diese Verknüpfung sollte es dem Roboter ermöglichen durch ein Verständnis der Prozesse auf mögliche Missverhältnisse angemessen zu reagieren.

Diese kumulative Dissertation besteht aus fünf begutachteten Veröffentlichungen sowie einem eingereichten Manuskript. Im Rahmen der Arbeit beschreibe ich Navigationsmuster auf Grundlage von qualitativen räumlich-zeitlichen Darstellungen und entwickele die dafür notwendigen korrekten Schlussfolgerungsmethoden. Mein Ansatz basiert auf einer räumlichen Logik im Sinne von Aiello, Pratt-Hartmann und van Benthem, die eine klare räumliche und zeitliche Semantik aufweist. In dieser Arbeit zeige ich, wie sich verschiedene Vorfahrtsregeln und gesellschaftliche Konventionen mit einer solchen Logik darstellen lassen.

Die Anwendbarkeit der entwickelten Methode demonstriere ich anhand von drei verschiedenen Beispielen: einem autonomen Robotersystem in einem industriellen Umfeld, einem autonomen Segelboot sowie einem Roboter, der sich höflich an gesellschaftliche Konventionen hält. In allen drei Anwendungen wird das Navigationsverhalten durch logische Formeln beschreiben und die notwendige zeitliche Inferenz geschieht auf Basis von Modellprüfungen. Ein wichtiger Aspekt der Modellierung ist, dass ein einzelnes logisches Symbol, wie *links abbiegen*, im Allgemeinen eine unendliche Menge von Aktionen repräsentiert, statt eines einzelnen, numerisch definierten Bewegungsbefehls. Dies ermöglicht es ohne Veränderungen der Formeln Variablen, wie den aktuellen räumlichen Kontext und die sich möglicherweise verändernde

Kinematik des Robotersystems, einzubeziehen. Außerdem zeige ich, dass sich der entwickelte Ansatz in verschiedene Roboter-Software-Architekturen integrieren lässt.

Darüber hinaus präsentiere ich mit der Und/Oder Lineare Programmierung eine Antwort auf drei wichtige Fragen im Bereich des formalen, qualitativen und räumlichen Schließens darstellt. Diese Fragen sind: Erstens, wie lassen sich verschiedene Kalküle gleichzeitig betrachten lassen? Zweitens, wie können (teilweise) spezifizierte Entitäten (Fixpunkte, Regionen in Form oder Position usw. fest) mit freien Entitäten kombiniert werden? Drittens, wie kann eine Realisierung der qualitative räumliche Beschreibung berechnet werden? Alle drei Fragen sind für Anwendungen wichtig, können jedoch mit bisherigen Techniken nicht einheitlich beantwortet werden. Darüber hinaus integrier ich Und/Oder Lineare Programmierung mit Linearer temporaler Logik, um eine vierte Frage zu beantworten: Wie lassen sich die zeitlichen Aspekte von Navigationsmustern darstellen?

Contents

List of Figures

List of Tables

List Of Abbreviations

AACN	Action-Augmented Conceptual Neighborhood
AI	Artificial Intelligence
AIS	Automatic Identification System
ASP	Answer Set Programming
BDI	BeliefDesireIntention software model
BR	Base Relation
CBR	Constraint Based Reasoning
CNF	Conjunctive Normal Form
CN	Conceptual Neighborhood
CNG	Conceptual Neighborhood Graph
CNL	Conceptual Neighborhood Logic
COLREG	International Regulations for Preventing Collisions at Sea
CSP	Constraint-Satisfaction Problem
CTL	Computational Tree Logic
DCNG	Directed Conceptual Neighborhood Graph
DNF	Disjuctive Normal Form
DSS	Dicision Support System
ICA	Inevitable Collision Aarea
ICS	Inevitable Collision State
JEPD	Jointly Exhaustive and Pairwise Disjoint
LP	Linear Programming
LRF	Laser Range Finder
LTL	Linear Temporal Logic
MILP	Mixed-Integer Linear Programming
NBR	Neighborhood-Based Reasoning
PRM	Randomized Roadmap Planner
QR	Qualitative Reasoning
QSR	Qualitative Spatial Reasoning
QSTR	Qualitative Spatial and Temporal Reasoning
SAT	Boolean Satisfiability Problem
SLAM	Simultaneous Localization And Mapping

1 Introduction

This cumulative dissertation provides an answer to the question in artificial intelligence of how to reason with diverse spatio-temporal calculi over partially bounded domains. The dissertation addresses this research problem in five peer-reviewed publications and one manuscript currently under review. The first part of this introductory chapter outlines the motivation for conducting this research as well as the four research questions, this dissertation specifically focusses on. The second part of the introduction describes the working hypothesis, the pursued approach, and the overall contribution of this dissertation to answering the primary research problem: *What is needed to base spatial behavior of a robot on qualitative spatio-temporal descriptions?* An outline and remarks on the form of this thesis close this chapter.

1.1 Motivation

As Bredeweg and Struss (2003) nicely state: “Reasoning about, and solving problems in, the physical world is one of the most fundamental capabilities of human intelligence and a fundamental subject for AI.” While the long-term goal of artificial intelligence (AI) is to recreate human-level intelligence, an intermediate goal is the imitation of human intelligent behavior through artificial means and techniques, especially those based on sound and complete reasoning, which are often referred to as *classical* AI. Currently, considerable effort is made in the AI community to reintegrate classical AI approaches with modern robotic methods. The present dissertation is situated at this intersection.

To draft the aim of this thesis, the following exemplary task is used: *Develop an autonomous robotic assistant that can be used when transporting dangerous materials through a factory.* Obviously, the robot should move safely in the factory but should generally also move as quickly as possible, because the transport of dangerous material poses a risk in itself. In order to increase the average speed of moving within the factory while maintaining a high level of safety, the human workers use traffic rules (right of way rules). Both human and robot should follow the same rules[1] and the rules should not be altered just because a robotic system is introduced into the working environment. These rules are as follows: a) stop at a “stop” sign, b) first-come, first-serve at an “all-way stop” intersection, and c) left yields to right in all other cases. Some materials need to be transported very fast or they may become unsuited for further processing so that an additional rule is given: A vehicle with red flashing lights does not have

[1] Of course, having different rules whether human-human, human-robot, or robot-robot interaction takes place are also interesting. However, for simplicity of the example, the rules should not distingush between human and robot.

to stop at a stop sign and is always allowed to go first, provided the others can definitely notice it. Further, due to safety concerns and legal requirements, the autonomous robotic system to develop has to be verifiably safe.

A prerequisite for solving the task of adhering to such regulations is a suitable representation of the physical space these regulations are situated in. Given that the desired transport vehicle is supposed to operate in an environment shared by humans and robots, the robot has to adhere to the established safety regulations as they are executed by the human workers. Such established regulations are described in natural language and involve human descriptions of space. Consequently, natural language descriptions have to be translated to an unambiguous computer-comprehensible format, which ideally should be verifiable by humans as easily as possible. Knauff, Rauh, and Renz (1997) as well as Klippel and Montello (2007) state that spatial concepts used in *Qualitative Spatial Reasoning* closely resemble human spatial understanding. “The basic idea of qualitative reasoning is that we use low-resolution representations to describe the essence of the state of affairs” (Freksa, 2004), resulting in a *spatial logic* in the sense of Aiello, Pratt-Hartmann, and van Benthem (2007b). Therefore, the methods developed in qualitative spatial reasoning suggest themselves as a starting point for an adequate knowledge representation of space understandable for both, humans and computers.

Besides space, a notion of time is required to be able to describe *first come, first serve* intersections. Representing and reasoning about temporal aspects of rules is therefore the second prerequisite for solving the exemplary task. Furthermore, the temporal aspects of the system have to be verifiable as well, requiring a formal representation of time. Provided, that both, suitable spatial and temporal logics can be found, they still need to be combined into a single formal representation. As Kontchakov et al. (2007) demonstrate in “Spatial logic + temporal logic =?”, this is generally not straightforward. Inevitably, each combination has to be thoroughly analyzed to avoid computational pitfalls.

Before rules can be applied by a robotic system they have to be written down, i.e., to be stated in the formal representation used. Generally the rules are first developed with an abstract 2-dimensional space in mind and often skip over some of the details. Consequently, this step of representing the rules formally over a specific domain generally includes further refinement and interpretation of the rules. Ideally, these refinements result in a set of rules that covers *all* possible situations and is *conflict free*. It is important to note that a rule set might not cover all situations one can think of, but that the situations not covered may not be realizable within a specific spatial setting. In general, we strive for rules that are intuitively comprehensible as well as provable conflict free. These two aims often contradict themselves, especially the more rigid a domain is. However, some conflicts in the rules caused by the abstraction of the domain, could be *solved* by the world itself, i.e., such a conflict might not be realizable.

The above rules, for example, do not state who has the right of way if two vehicles with flashing red lights meet. However, there might only be specific routes, which these vehicles can take and none of these routes actually meet. Therefore, it should not only be possible to check the rules for conflicts, but also to consider the space in which these rules are to be executed and whether the conflict persists there or not. In the given exemplary task it might be possible to handle some exceptional cases by a more complex rule system, but this would require a retraining of the human staff, defying the premise that robotic systems should adapt to the human and not the human to the machine. However, it might be more reasonable to modify the environment, e.g. by introducing oneway streets already familiar to the human workers, than to develop a more complex rule system. Therefore, in order to ensure a verifiably safe translation of rules, methods and tools should support a knowledge engineer during the translation process.

When a complete and conflict-free rule set has been established—for a given spatial setting—the final step is to control a robot based on these rules. This can be accomplished in two ways: In the first case, the robot is under the supervision of an external system, which cancels actions that would violate the rules. The second approach is to apply methods that allow to derive a controller for the robot that is *correct by construction*, i.e. if the rules are correct so is the automatically derived controller. Both of these approaches are currently being researched, for example the former by Täubig et al. (2012) and the latter by Kress-Gazit, Wongpiromsarn, and Topcu (2011). The shortest path or sequence is not necessarily the fastest, due to the required compliance with the rules. For example, the shortest way might force the robot to slow down due to obstructed sight, whereas a long but more open path could be traveled faster, leading to an overall earlier arrival. Consequently, it is desirable that the robot considers the rules already when planning actions or routes, in order to achieve a high performance with regards to task execution.

In summary, to solve the autonomous transportation task, various aspects and their interplay have to be considered in detail and will be introduced below. The first aspect, described in Section 1.1.1, is an effective qualitative representation of space that allows for efficient reasoning. The second aspect, discussed in Section 1.1.2, is the integration of external constraints, such as a floor plan, with qualitative spatial reasoning. In Section 1.1.3 the third aspect, the possible representations of time, is described and the difficulties when combining temporal logic with spatial logic are indicated. The forth aspect, a possible support for knowledge engineers is discussed in Section 1.1.4. Each of these four aspects will be concluded with an open research question that needs to be addressed in order to solve the proposed task.

1.1.1 Qualitative Spatial Reasoning

Qualitative Spatial Reasoning[2] (QSR) is based on the idea that rather than using numerical coordinates, a finite set of spatial relations between objects is used, e.g., the robot is *inside* the loading zone. For a detailed coverage, please refer to the literature, e.g. the overview paper by Ligozat (2011), Renz and Nebel (2007) or a more technical analysis by Dylla et al. (2013b). A further aim was to develop efficient techniques for reasoning based on qualitative relations.

However, spatial relations are first of all a representation. Reasoning, such as inferring new information, requires methods that can manipulate the represented knowledge. For example, in Figure 1.1 three different circles are pictured $A, B,$ and C, where a) A is part of B, and b) B

[2] In some earlier literature it is called *Qualitative Spatial Representation and Reasoning*, but the general consensus is that there can be no reasoning without representation and therefore only *Qualitative Spatial Reasoning* persisted.

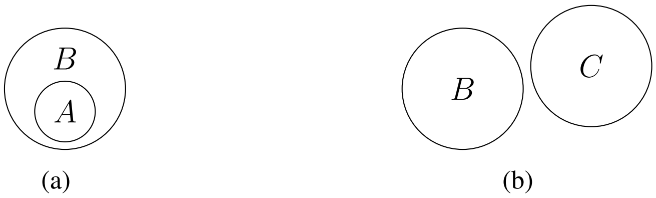
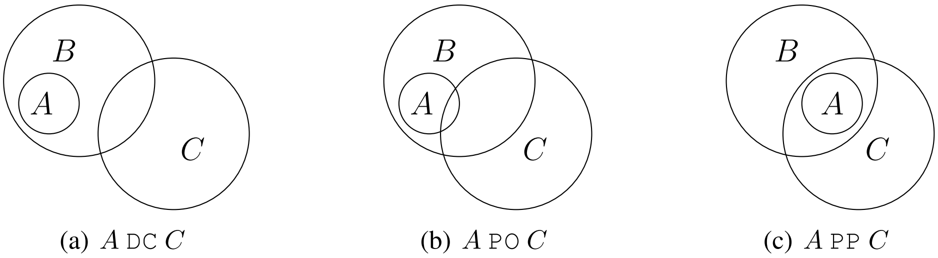


Figure 1.1: Given are two pictorial representation of the relations between a) A and B, and b) B and C. What are the possible relations between C and A?



(a) A DC C (b) A PO C (c) A PP C

Figure 1.2: All possible realizations of the constraint network as given by RCC-5 over discs in 2D: $(A \texttt{ PP } B) \wedge (B \texttt{ PO } C)$. The RCC-5 relations are: *disconnected* (DC), *partial-overlap* (PO), *equal* (EQ), *proper-part* (PP), and *proper-part-inverse* (PPi).

that does not overlap C, what are the possible relations between C and A? Obviously C and A also do not overlap. What if B and C were instead partially overlapping? Almost nothing could be inferred about the relation between C and A. Nevertheless, the relation between C and A can not be arbitrary, as C can not be a proper part of A and they can not be equal. This type of reasoning is called *compositional reasoning*. Figure 1.2 shows all all solutions using the region connection calculus (RCC-5) by Randell, Cui, and Cohn (1992). As a result from compositional reasoning, which can lead to sets of relations rather than individual relations, the *(weak) composition* is defined to operate on sets of relations. *Converse* is another type of reasoning used to obtain the set of relations between B and A when the set of relations between A and B is known. A finite set of relations called *base relations*, together with the two operators (weak) *composition* and *converse* is generally called a *qualitative calculus*.

In qualitative spatial reasoning, the most prominent method used is constraint-based reasoning. For this to be applicable, two restrictions to the base relations are necessary. The base relations should be *jointly exhaustive*, i.e., they should cover the complete domain. *Pairwise disjoint* is the second property that the set of base relations should have, viz. no two base relations describe the same (spatial) configuration of objects. Taken together, this means that each (spatial) configuration is assigned exactly one base relation but obviously not the other way around. Generally these restrictions lead to a relation algebra.

Numerous qualitative calculi have been developed so far, as the essence of a problem or its solution is highly dependent on the task at hand. In the rules of the introductory examples, various spatial and temporal aspects are mentioned. First of all, stopping at a stop sign generally implies a specific area in which the vehicle has to come to a full stop. Requiring (mereo-)topological aspects of space to be represented, for example with the Region Connection Calculus (RCC) by Randell, Cui, and Cohn (1992). Second, we require relative directions to describe that *left yields to right*. An apparent calculus, the left-right-distinguishing calculus $\mathcal{LR}$, developed by Scivos and Nebel (2005), is not expressive enough, as left and right each cover 180°, however, one generally does not yield to others behind oneself. A more expressive calculus is the Oriented Point Relation Algebra[3] ($\mathcal{OPRA}_\mathrm{m}$), developed by Moratz (2006), which has a scalable granularity m and subsumes $\mathcal{LR}$. However, in the running example a representation is required that can jointly express topological relations and relative orientations. Taken together: *how can different calculi be combined?*

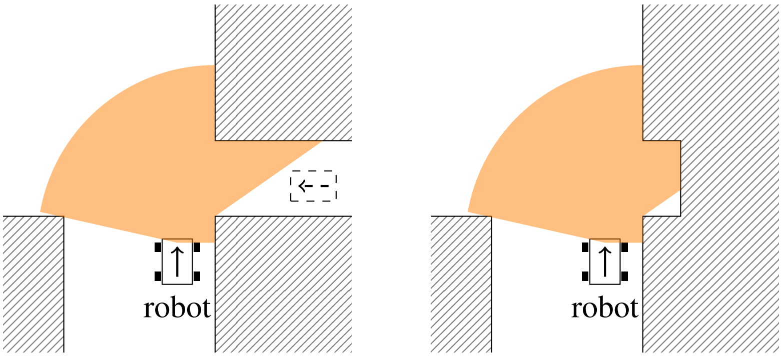
Wölfl and Westphal (2009) define two algebraic approaches to the question of how to combine calculi: loose and tight coupling. The latter essentially is the manual development of a new joint calculus, while the loose coupling is generally too weak for sound and complete reasoning. Consequently, tight coupling is used throughout the literature, as is evident by the plethora of calculi developed. A third way is to translate each qualitative calculus into a common, expressive formalism. Bhatt, Lee, and Schultz (2011) and Wolter (2012) use algebraic geometry to capture a multitude of spatial relations, but due to the inherent computational complexity their approaches are limited to toy problems. As a result of the requirement to simultaneously reason with different calculi and the shortcomings of previous approaches, a first research question is identified:

> *How can qualitative calculi be combined, i.e. how can one jointly reason with knowledge represented in distinct calculi?*

1.1.2 External constraints

In qualitative (spatial) constraint reasoning, a single joint domain is assumed, namely every object can be everywhere in the domain. In “Here, There, but Not Everywhere [...]” Liu and Li (2012) identify the problem that in applications different objects have different restriction on the underlying domain. For example, the robotic transporter is instructed to fetch a pallet upon which some dangerous goods are stored. The location is described by a worker as follows: *The pallet is* in *the central storage area, and* close to *an emergency exit, and it is to the* north *or* north-east *of the loading zone*. Whereas, the location of the pallet is unknown, the central storage area, the locations of the emergency exits, as well as the loading zone *are* known. In the case of the central storage area and the loading zone both might be single, fixed, entries, which Li, Liu, and Wang (2013) call *landmarks*. The emergency exits on the other hand are only *finitely fixed*, as the factory has several emergency exits. Consequently, the emergency exit referred to has a different restricted domain than the pallet. One way to model this request is to state that the domain of the *specific* emergency exit referred to is restricted to the locations

[3] Dylla et al. (2013a) propose to call it Oriented Point *Representation* Algebra instead, as $\mathcal{OPRA}_\mathrm{m}$is not a *relation algebra*.



(a) A vehicle could be hypothesized

(b) Limited space does not allow the hypothesis of vehicle

Figure 1.3: Top-down pictorial representation of a robot and its field-of-view in different situations. *Can another vehicle be hypothesis such that it is not visible to the robot*, but the robot would have to yield to it, i.e., the vehicle to be placed would be in front and to the right of the robot. In the left picture this is possible, but in the right picture the only space that could work is to small.

of all of the emergency exits. Another way is to use a disjunction as to which "emergency exit" is referred to. Constraint languages used in QSR can not express such disjunctions, and henceforth either a more expressive formalism or a different approach to this kind of reasoning is required. A third option is to exhaustively enumerate the possibilities and check wether they are spatially consistent, but this approach generally does not scale well for large domains.

Checking whether a spatial configuration is realizable within a given context, such as a floor plan, can be viewed as applying constraints imposed on individual objects in the domain. The following query is a typical example in safe navigation: given the current position and orientation of the robot, can there exist a vehicle that is occluded and would have the right of way. In Figure 1.3 two very similar scenarios are displayed, but the query can only be answered with *yes* in one of them. Obviously, the current locations, the observed free space, and especially the unobservable space—because it is occluded—need to be considered to answer that query.

Reasoning with such special kinds of restrictions will be called reasoning with *partially grounded* information throughout this thesis. This naming is derived from the denotation of a logical formula that has no free variables, viz. a *grounded formula*. Taken together, the following research question is identified:

> *How can qualitative representations incorporate grounded information, i.e. how can free-ranging and constrained variable domains (singleton, finite, numerical constraints) be mixed?*

While Li, Liu, and Wang (2013) developed a specific answer for the region connection calculus (Randell, Cui, and Cohn, 1992), it is generally still an open question for other calculi.

1.1.3 Representing Time

The basic nature of time is generally thought of as being either linear or branching. Pnueli (1977) developed a theory of linear time, which is accordingly called *Linear Temporal Logic* (LTL). Viewing time as branching has been researched by Clarke and Emerson (1982) and resulted in the *Computational Tree Logic* (CTL). Emerson and Halpern (1986) unified both approaches and developed the superset called CTL*. Further, while CTL*and LTL both are PSPACE-complete, Lichtenstein and Pnueli (1985) showed that LTL scales linearly with the number of (possible) states and that in applications generally the size of the state-space dominates the size of the formula by a large factor. State of the art model checkers such as PRISM (Kwiatkowska, Norman, and Parker, 2011) are capable of efficiently handling LTL and even CTL*.

Regarding the task of developing the autonomous transporter, how should the temporal aspects be represented? Such a representation has to be suitable to describe the temporal ordering in *first come, first served* and should allow for the specification of the robot control, i.e. which actions the robot should take. On the *high-level* side, the situation calculus developed by McCarthy (1963) is one prominent approach and is the basis for the robot control language Golog developed by Levesque et al. (1997). Bhatt, Rahayu, and Sterling (2006) present an extension that includes spatio-temporal constraints resulting in a highly expressive language. A limitation of this approach is that the situation calculus is an undecidable logic and therefore, cannot have a sound reasoning method as required for safety applications.

On the robotic side, LTL rather than CTL has been advocated. Antoniotti and Mishra and Kress-Gazit, Wongpiromsarn, and Topcu (2011) used LTL to specify a controller in a *correct-by-construction* manner. Kloetzer and Belta (2006) used LTL for high-level specifications of (motion) planning. Further, Smith et al. (2010) as well as Lahijanian, Andersson, and Belta (2011) advanced the capabilities of motion planning given LTL specifications. In 2010 Kloetzer and Belta applied these specifications to real robotic systems. In summary, linear temporal logic is a decidable formalism for representing time, that is well established in the robotic community. However, Kontchakov et al. (2007) demonstrate in “Spatial logic + temporal logic =?” that even the combination of a decidable spatial and a decidable temporal logic easily is too expressive and therefore undecidable. Therefore, the following research question arises:

> *How can a spatial logic and linear temporal logic be combined to yield a decidable formalism, that can be applied to various applications?*

1.1.4 Supporting Knowledge Engineers

Assuming that the three previous research questions can be answered, a key part for the exemplary task still needs to be done: translating or modeling the rules. Rules such as those in the example are generally not complete, for example, the situation in which two or more vehicles with flashing red light meet at an intersection is not governed by the rules. A human would most likely default to reasoning about the intention of the rule and might fall back to one of the other rules or find a temporary other agreement. A robot might show some kind of emergent behavior but is likely to do something unintended.

Instead of finding out these unintended emergent behavior through testing, let the computer do what it is good at: *painstakingly* search for violations of the intentions. The intentions of the rules have to be translated into a formula as well, such that it can be checked if it is possible to violate these intentions while following the (current) rules. In the running example, one intention of the rules is to avoid collisions, therefore it has to be checked whether a collision can be achieved, given that everyone acts according to the rules. Generally a full guarantee of safety of humans is not possible if any kind of rule violation on the human part is assumed. For example, if the robot is standing still, a human could still willingly try to harm himself by crashing into the robot.

Knowing that a situation is not covered by the rules is an important step when refining the rule system. However, without knowing which situations cause the violation, the knowledge engineer's task of fixing the rule system is still a difficult one. Given that the methods aim for using high-level logic, the counter example found, will be a logical description itself. This is very helpful, but if the described situation is a complex one, such as involving various entities of different kinds, the logical description might not be an adequate representation for understanding the problem at hand.

In the seminal paper "Why a Diagram is (Sometimes) Worth Ten Thousand Words" by Larkin and Simon (1987), the following is stated:

> ”[...] a diagram can be superior to a verbal description for solving problems:
>
> - Diagrams can group together all information that is used together, thus avoiding large amounts of search for the elements needed to make a problem-solving inference.
> - Diagrams typically use location to group information about a single element, avoiding the need to match symbolic labels.
> - Diagrams automatically support a large number of perceptual inferences, which are extremely easy for humans."

The above observations lead to the assumption, that drawing an example of the situation that violates the intention, can be vastly superior to the pure logic description. Especially, as the rules described so far are mainly of spatial nature. This leads to the following research question:

> *How can a prototypical pictorial representation be derived from a (pure) qualitative description of a scene?*

1.2 Thesis and Contribution

Based on the example introduced in the previous section four research questions have been identified. Answering each of these research questions is essential to solve the following research problem:

> What is needed to base spatial behavior of a robot on qualitative spatio-temporal descriptions?

Thesis

> And/Or enhanced Linear Programming combined with Linear Temporal Logic is an adequate way to model spatial conventions and allows to reason in partially grounded scenarios with mixed qualitative representations.

1.2.1 Approach

The scientific approach to the above stated research problem taken in this thesis is twofold. First, exemplary applications are identified, which would benefit from an answer to the research questions. For each application a specific answer is developed providing necessary insights on individual aspects. Second, based on the requirements identified across all applications, a single unifying answer is developed.

Ideally, in a chosen exemplary application either one research question should manifest strongly, or the application should span (almost) all of the research questions. While the first option provides a deep insight into the respective aspect, the latter, an all-embracing application, provides the potential to focus on the interdependencies of these aspects. Three applications are identified that are of exemplary character. One focusing on the temporal representation, one spans the aspects of temporal reasoning and pictorial representations as well as verifiability, and the third application is an all-encompassing one.

Next, the three applications are shortly introduced and discussed, followed by a brief overview of the connecting methods. The contribution of this thesis concludes this section.

Applications

In the first application a (simple) mobile robot has to infer various processes happening in a warehouse, based on its partial observations. While it does know what to look for, i.e., it knows (all) the processes taking place in the warehouse, it is missing some key information. For example, within a process description a specific zone, such as a buffer area, is mentioned but it is unknown to the robot where it is located. On the one hand, if the robot knew which processes a *good*[4] is currently taking part in, it could infer the locations of the involved zone. On the other hand, if the robot knew the location of the zones, it could identify the possible processes a

[4] In Chapter 2 this is called a *ware*.

good is presently involved in. The research question this application exemplifies is the required interplay between (simple) spatial reasoning and temporal reasoning. The application, its research question, and a solution for this specific aspect is presented in Chapter 2 (Kreutzmann, Colonius, et al., 2011).

Developing an autonomous sailing vessel that obeys the International Regulations for Preventing Collisions at Sea (Colregs), is the second application investigated. The posed task, is to ensure that the sailing vessel exhibits correct behavior, as specified in the natural language regulations. As was demonstrated previously, for example by Dylla, Frommberger, et al. (2007); Dylla (2009), (most) of the involved regulations can be modeled using a single qualitative calculus, namely $\mathcal{OPRA}_{\mathrm{m}}$. Therefore the research question investigated within this application, is how to represent spatial knowledge for *control*. In Chapter 3 Wolter, Dylla, and Kreutzmann present the results of this investigation.

While working on this application, a second aspect became evident. The Colregs govern the behavior of *two* vessels, such as to avoid collision with one another, but they may fail as soon as three or more vessels are directly involved. Consequently, this application contributes the the research question of a decidable formalism by posing the question: *how to detect contradictions or missed cases* within a given spatial rule set. Solutions in the form of lengthly formulas describing such contradictions tend to be hard to imagine or draw, making it even harder to find a solution. As a result, the applications features a third research question: can a *visual representation* of the formula be generated? A solution focusing on these two research questions—contradictions and visual representation—is presented in Chapter 4 (Kreutzmann, Wolter, et al., 2013).

The last application is quite similar to the introductorily example and is of unifying character. In an industrial setting, such as a factory, a mobile robot should operate safely in spaces also occupied by humans. Further, the robotic system should follow the same rules as the humans do to prevent collisions and it should be the robot that adapts to the human workers rather than the other way around. The safe operation of mobile robotic systems in industrial settings, even without other humans involved, is currently still under research, for example by Täubig et al. (2012). Indeed, the approach developed within this thesis can incorporate and extend the results obtained by Täubig et al. (2012) to also include traffic rules. The manuscript presented as Chapter 5 collects our results in researching safe navigation based on qualitative spatio-temporal reasoning.

Methods

The highest possible safety a system can have is not achieved by rigorous testing alone, but requires also proof of correctness. Consequently, the development of a formal method is required that allows to specify the spatio-temporal regulations in a away that can be proven to be conflict free, cover all aspects, and allows a direct application or a *correct by construction* translation.

The method of this work is to develop a formal spatio-temporal representation, that is based on concepts developed in qualitative spatial reasoning. This allows for a straight forward (human) translation from natural language descriptions to such a formal representation. The four key methods that are used throughout the thesis are:

QSR and LTL For the knowledge representation various qualitative spatial calculi are used together with linear temporal logic.

Model Checking Model checking is used on the abstract level as the main reasoning technique.

Oracle Variables At various places *oracle variables* are used, which Morgenstern and Schneider (2011) describe as "[...] may represent 'angelic' nondeterminism that may be resolved in favor to satisfy the specification.".

And/Or LP I developed And/Or Linear Programming for checking spatial consistency of mixed-qualitative and quantitative spatial description. Oracle variables are used to approximate non-linearities.

1.2.2 Contribution of this Thesis

In this thesis I develop an *And/Or Linear Programming* and combine it with Linear Temporal Logic. This combination is an answer to three questions in the qualitative spatial reasoning community, because it

- allows joint reasoning about most known calculi,
- enables reasoning about partially restricted domains, and
- provides a method for calculating a realization of QSR formulas.

It continues and even accelerates the arising trend in QSR to develop alternative reasoning methods beside constraint based reasoning.

Further contributions are:

- identifying the importance and applicability of model checking as method to spatial reasoning,
- the identification of exemplary applications for various research problems in the field of qualitative spatio-temporal reasoning.

1.3 Outline

This cumulative dissertation has the following outline. First the publications resulting from the three applications described in Section 1.2.1 are presented. Starting with the publication about the recognition of spatio-temporal logistic processes (Chapter 2). The next application, namely an autonomous sailing vessel, resulted in two publications: how to control autonomous sailing ships (Chapter 3), and an in-depth analysis about the required high-level modeling and possible support tools (Chapter 4). Chapter 5 is a manuscript under review that contains the third application, namely developing a verifiable safe robotic system. This manuscript has an equal focus on the temporal aspect as well as the spatial reasoning part and demonstrates an application relevant, decidable combination of these two aspects. Nevertheless, resulting from the overall aim of the manuscript, the spatial reasoning part spans only the application relevant calculi. Chapter 6 concludes the application driven part with a publication that demonstrates the expressivity of the developed methods by showing that various social conventions with spatial extent can be modeled. The final publication is of (purely) theoretical nature, it is an in-depth view on the capabilities of the developed And/Or Linear Programming technique (Chapter 7). This thesis is concluded by an overall discussion of the presented approach, and gives an outlook towards possible followup research. As each manuscript has only a limited amount of space, the conclusion also discusses further references to the state of the art.

To establish a coherent presentation, the layout of published manuscripts have been altered. These changes are described in the next paragraph below.

Form of this Thesis

To achieve a coherent layout throughout this *cumulative* thesis, the following change to the (published) manuscripts were performed: The numbers of figures, theorems, corollary, etc. are adapted to include the chapters within this thesis. Also the presentation of algorithms has been unified across all manuscripts, as well as the bibliography and citation style. Each chapter has its own bibliography.

Nothing concerning the content or the wording of the manuscripts has been altered. When referring to or citing something from chapters 2–7, please cite the original published article.

References

Aiello, Marco, Ian E. Pratt-Hartmann, and Johan van Benthem, eds. (2007a). *Handbook of Spatial Logics*. Springer. ISBN: 978-1-4020-5586-7.

Aiello, Marco, Ian E. Pratt-Hartmann, and Johan van Benthem (2007b). “What is Spatial Logic?” In: *Handbook of Spatial Logics*. Ed. by Marco Aiello, Ian E. Pratt-Hartmann, and Johan van Benthem. Springer, pp. 1–11. ISBN: 978-1-4020-5586-7.

Antoniotti, Marco and Bud Mishra (1995). “Discrete event models + temporal logic = supervisory controller: Automatic synthesis of locomotion controllers”. In: *Proceedings of the IEEE Conference on Robotics and Automation (ICRA)*. Vol. 2, pp. 1441–1446.

Bhatt, Mehul, Jae Hee Lee, and Carl Schultz (2011). “CLP(QS): a declarative spatial reasoning framework”. In: *Proc. of Cosit*, pp. 210–230.

Bhatt, Mehul, Wenny Rahayu, and Gerald Sterling (2006). “Qualitative Spatial Reasoning with Topological Relations in the Situation Calculus”. In: *In Proc. of (FLAIRS 06)*. AAAI Press.

Bredeweg, Bert and Peter Struss (2003). “Current topics in qualitative reasoning”. In: *AI Magazine* 24.4, p. 13.

Clarke, Edmund M. and E. Allen Emerson (1982). “Design and Synthesis of Synchronization Skeletons Using Branching-Time Temporal Logic”. In: *Logic of Programs, Workshop*. London, UK, UK: Springer-Verlag, pp. 52–71. ISBN: 3-540-11212-X.

Levesque, Hector J., Raymond Reiter, Yves Lespérance, Fangzhen Lin, and Richard B. Scherl (1997). “GOLOG: A Logic Programming Language for Dynamic Domains”. In: *Journal of Logic Programming* 31, pp. 59–84.

Dylla, Frank (2009). “Qualitative Spatial Reasoning for Navigating Agents - Behavior Formalization with Qualitative Representations”. In: *Behaviour Monitoring and Interpretation - Ambient Assisted Living*. Ed. by Björn Gottfried and Hamid K. Aghajan. Vol. 3. Ambient Intelligence and Smart Environments. IOS Press, pp. 98–128. ISBN: 978-1-60750-459-7.

Dylla, Frank, Lutz Frommberger, Jan Oliver Wallgrün, Diedrich Wolter, Bernhard Nebel, and Stefan Wölfl (2007). “SailAway: Formalizing Navigation Rules”. In: *Proocedings of the AISB07 workshop on Spatial Reasoning and Communication*, pp. 470–474.

Dylla, Frank, Till Mossakowski, Thomas Schneider, and Diedrich Wolter (2013a). “Algebraic Properties of Qualitative Spatio-Temporal Calculi”. In: *Proceedings of Conference On Spatial Information Theory 2013*.

Dylla, Frank, Till Mossakowski, Thomas Schneider, and Diedrich Wolter (2013b). “Algebraic Properties of Qualitative Spatio-Temporal Calculi”. In: *Proceedings of the Conference On Spatial Information Theory (COSIT-13)*. Vol. 8116. LNCS. North Yorkshire, UK: Springer, pp. 516–536.

Emerson, E. Allen and Joseph Y. Halpern (Jan. 1986). ““Sometimes” and “Not Never” Revisited: On Branching Versus Linear Time Temporal Logic”. In: *J. ACM* 33.1, pp. 151–178. ISSN: 0004-5411.

Freksa, Christian (2004). “Spatial Cognition – an AI perspective”. In: *Proc. European Conference on Artificial Intelligence (ECAI’04)*. Ed. by Ramon López de Mántaras and Lorenza Saitta. IOS Press; Amsterdam, pp. 1122–1128.

Klippel, Alexander and Dan Montello (2007). “Linguistic and nonlinguistic turn direction concepts”. In: *Proceedings of Conference of Spatial Information Theory (COSIT)*. Vol. 4736. Springer, pp. 354–372.

Kloetzer, Marius and Calin Belta (2006). “LTL planning for groups of robots”. In: *Proceedings of the IEEE International Conference on Networking, Sensing and Control (ICNSC)*, pp. 578–583.

Kloetzer, Marius and Calin Belta (2010). “Automatic deployment of distributed teams of robots from temporal logic motion specifications”. In: *IEEE Transactions on Robotics* 26.1, pp. 48–61.

Knauff, Markus, Reinhold Rauh, and Jochen Renz (1997). “A cognitive assessment of topological spatial relations: Results from an empirical investigation”. In: *Spatial Information Theory: A theoretical basis for GIS*. Proceedings of COSIT. Springer, pp. 193–206.

Kontchakov, Roman, Agi Kurucz, Frank Wolter, and Michael Zakharyaschev (2007). “Spatial Logic + Temporal Logic = ?” In: *Handbook of Spatial Logics*. Ed. by Marco Aiello, Ian E. Pratt-Hartmann, and Johan van Benthem. Springer, pp. 497–564. ISBN: 978-1-4020-5586-7.

Kress-Gazit, Hadas, Tichakorn Wongpiromsarn, and Ufuk Topcu (Sept. 2011). “Correct, Reactive Robot Control from Abstraction and Temporal Logic Specifications”. In: *Special Issue of the IEEE Robotics and Automation Magazine on Formal Methods for Robotics and Automation* 18.3, pp. 65–74.

Kreutzmann, Arne, Immo Colonius, Lutz Frommberger, Frank Dylla, Christian Freksa, and Diedrich Wolter (Sept. 2011). “On Process Recognition by Logical Inference”. In: *Proc. of the 5th European Conference on Mobile Robots (ECMR)*, pp. 7–12.

Kreutzmann, Arne, Diedrich Wolter, Frank Dylla, and Jae Hee Lee (2013). “Towards Safe Navigation by Formalizing Navigation Rules”. In: *TransNav, the International Journal on Marine Navigation and Safety of Sea Transportation* 7.2, pp. 161–168. ISSN: 2083-6473.

Kwiatkowska, Marta, Gethin Norman, and David Parker (2011). “PRISM 4.0: Verification of Probabilistic Real-time Systems”. In: *Proc. of 23rd International Conference on Computer Aided Verification (CAV’11)*. Ed. by Ganesh Gopalakrishnan and Shaz Qadeer. Vol. 6806. LNCS. Springer, pp. 585–591.

Lahijanian, Morteza, Sean B. Andersson, and Calin Belta (2011). “Temporal Logic Motion Planning and Control With Probabilistic Satisfaction Guarantees”. In: *IEEE Transactions on Robotics* PP.99, pp. 1–14.

Larkin, Jill H. and Herbert A. Simon (1987). “Why a Diagram is (Sometimes) Worth Ten Thousand Words”. In: *Cognitive Science* 11.1, pp. 65–100. ISSN: 1551-6709.

Li, Sanjiang, Weiming Liu, and Shengsheng Wang (2013). “Qualitative constraint satisfaction problems: An extended framework with landmarks”. In: *Artificial Intelligence* 201, pp. 32–58.

Lichtenstein, Orna and Amir Pnueli (1985). “Checking That Finite State Concurrent Programs Satisfy Their Linear Specification.” In: *Proceedings of the 12th ACM SIGACT-SIGPLAN symposium on Principles of programming languages (POPL)*, pp. 97–107.

Ligozat, Gérard (2011). *Qualitative Spatial and Temporal Reasoning*. Wiley. ISBN: 978-1848212527.

Liu, Weiming and Sanjiang Li (2012). “Here, There, but Not Everywhere: An Extended Framework for Qualitative Constraint Satisfaction.” In: *ECAI*, pp. 552–557.

McCarthy, John (1963). *Situations, Actions, and Causal Laws*. Tech. rep. Memo 2. Stanford University: Stanford Artificial Intelligence Project.

Moratz, Reinhard (2006). “Representing Relative Direction as a Binary Relation of Oriented Points”. In: *17th European Conference on AI*. Ed. by Gerhard Brewka, Silvia Coradeschi, Anna Perini, and Paolo Traverso. Italy: IOS Press, pp. 407–411. ISBN: 1-58603-642-4.

Morgenstern, Andreas and Klaus Schneider (2011). “Program Sketching via CTL* Model Checking”. In: *Model Checking Software (SPIN)*. Ed. by Alex Groce and Madanlal Musuvathi. Vol. 6823. LNCS. Snowbird, Utah, USA: Springer, pp. 126–143.

Pnueli, Amir (1977). “The temporal logic of programs”. In: *Proceedings of the 18th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 46–57.

Randell, David A., Zhan Cui, and Anthony G. Cohn (1992). “A Spatial Logic based on Regions and “Connection””. In: *Proc, of KR’92*. Ed. by Bernhard Nebel, Charles Rich, and William Swartout. San Mateo (CA), USA: Morgan Kaufmann, pp. 165–176.

Renz, Jochen and Bernhard Nebel (2007). “Qualitative spatial reasoning using constraint calculi”. In: *Handbook of Spatial Logics*. Ed. by Marco Aiello, Ian E. Pratt-Hartmann, and Johan van Benthem. Springer, pp. 161–215. ISBN: 978-1-4020-5586-7.

Scivos, Alexander and Bernhard Nebel (2005). “The Finest of its Class: The Natural Point-Based Ternary Calculus $\mathcal{LR}$ for Qualitative Spatial Reasoning”. In: *Spatial Cognition IV. Reasoning, Action, Interaction*. Ed. by Christian Freksa, Markus Knauff, Bernd Krieg-Brückner, Bernhard Nebel, and Thomas Barkowsky. Vol. 3343. Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 283–303. ISBN: 978-3-540-25048-7.

Smith, Stephen L., Jana Tůmová, Calin Belta, and Daniela Rus (Oct. 2010). “Optimal path planning under temporal logic constraints”. In: *Proceeding of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Taipeh, Taiwan, pp. 3288–3293.

Täubig, Holger, Udo Frese, Christoph Hertzberg, Christoph Lüth, Stefan Mohr, Elena Vorobev, and Dennis Walter (2012). “Guaranteeing functional safety: design for provability and computer-aided verification”. In: *Autonomous Robots* 32.3, pp. 303–331.

Wölfl, Stefan and Matthias Westphal (2009). “On Combinations of Binary Qualitative Constraint Calculi”. In: *Proc. of IJCAI*, pp. 967–973.

Wolter, Diedrich (2012). “Analyzing Qualitative Spatio-Temporal Calculi using Algebraic Geometry”. In: *Spatial Cognition and Comp.* 12.1, pp. 23–52.

Wolter, Diedrich, Frank Dylla, and Arne Kreutzmann (2011). “Rule-Compliant Navigation With Qualitative Spatial Reasoning”. In: *Proceedings of the 4th Intl. Robotic Sailing Conference*. Springer.

2 Temporal Logic for Process Specification and Recognition

Arne Kreutzmann[1], Immo Colonius[1*], Diedrich Wolter[1], Frank Dylla[1], Lutz Frommberger[1], and Christian Freksa[1]

1 Cognitive Systems Group, University of Bremen, Bremen, Germany

* Corresponding Author

Published in “Intelligent Service Robotics”, 2013, Volume 6, Number 1.

This paper is a significantly extended and improved version of (Kreutzmann et al., 2011) presented at ECMR 2011. We have improved the interpretation of robot observations and we present a new experimental evaluation, based on an enhanced model checker implementation.

Contributions:

The study was conducted jointly by Immo Colonius and me. I researched the theoretical foundations and provided the idea to use LTL and further to use ASP for model checking. Consequently I implemented the model checking parts and did most of the LTL/ASP modeling, which itself was based on Immo's research on warehouse processes. Immo Colonius focused on developing and implementing the necessary extensions to Diedrich Wolter's robotic framework necessary for the conducted experiments and performed most of the experiments. I also contributed to preparing the manuscript.

Acknowledgements:

This paper presents work done in the project R3-[Q-Shape] of the Transregional Collaborative Research Center SFB/TR 8 Spatial Cognition. Financial support by the German Research Foundation (DFG) is gratefully acknowledged. We like to thank U. Frese for his valuable comments and his support in extending the TreeMap-algorithm. We also thank the anonymous reviewers for their helpful comments.

Abstract

Acting intelligently in dynamic environments involves anticipating surrounding processes, for example to foresee a dangerous situation by recognizing a process and inferring respective safety zones. Process recognition is thus key to mastering dynamic environments including surveillance tasks.

In this paper we are concerned with a logic-based approach to process specification, recognition, and interpretation. We demonstrate that linear temporal logic (LTL) provides the formal grounds on which processes can be specified. Recognition can then be approached as a model checking problem. The key feature of this logic-based approach is its seamless integration with logic inference which can sensibly supplement the incomplete observations of the robot. Furthermore, logic allows us to query for process occurrences in a flexible manner and it does not rely on training data. We present a case study with a robotic observer in a warehouse logistics scenario. Our experimental evaluation demonstrates that LTL provides an adequate basis for process recognition.

2.1 Introduction

Mastering dynamic environments is a demanding challenge in autonomous robotics, involving recognition and understanding processes in the environment. Recent advances in simultaneous localization and mapping in dynamic environments build the basis for sophisticated navigation, but *understanding* processes goes even beyond. The ability to recognize and to understand processes allows a robot to interact with its environment in a goal orientated fashion. For example, in processes that involve dangerous situations like the violation of safety zones, process understanding enables a robot to avoid dangerous situations in an anticipatory manner. But first of all, processes need to be represented in a way that fosters process understanding. Moreover, the representation should be seamlessly integrated with other high-level robot control tasks to ease the control flow.

We approach process understanding with linear temporal logic (LTL) (Pnueli, 1977, see Sect. 2.3) which allows us to represent processes as logic formulas in a declarative manner. LTL is a slender knowledge representation language that recently has received increasing attention from the autonomous robotics community. The use of LTL in robotics has been advocated much earlier though (Antoniotti and Mishra, 1995). For example, LTL has been used to specify controllers in a correct-by-construction manner (Kress-Gazit, Wongpiromsarn, and Topcu, 2011). LTL is widely used for motion planning from high-level specifications (e.g. Kloetzer and Belta, 2006; Smith et al., 2010; Lahijanian, Andersson, and Belta, 2011). Kloetzer and Belta (2010) demonstrate the applicability to real robotic systems. Our motivation of using LTL is twofold. Firstly, we want to demonstrate that LTL specifications also provide an adequate basis for process recognition and understanding, supplementing existing approaches to robot control. Secondly, LTL allows a domain expert to describe processes of interest in a way that does not require knowledgeability of robot technology. LTL further provides an excellent basis for flexibly querying the observations of the robot. It is then the task of the robotic system to turn a query into an effective observation and reasoning strategy.

In this paper we focus on spatio-temporal processes, i.e., processes that are characterized by temporal patterns of movements in space. Spatio-temporal aspects are at the core of any process description and so this study achieves a high degree of generality. As scenario for our experimental evaluation we have selected warehouse logistics which is an interesting and relevant domain for studying spatio-temporal processes. In a warehouse, there is a steady flow of goods which are moved through space, establishing *functional zones* that are connected with certain types of storage processes (for example, admission of goods into a warehouse makes use of buffer zones to temporarily store goods). Note that these functional zones are not necessarily known a-priori. Hildebrandt et al. (2010) argue for use of autonomous robots as a minimally invasive means to recognize in-warehouse processes which, in turn provides the knowledge for optimizing the warehouse. The task of the robot is to recognize the storage processes that occur. However, a robot is generally not able to gather all potentially relevant information about a process and therefore needs to infer missing pieces of information, in particular identifying functional zones and their whereabouts.

The first contribution of this paper is to show that LTL offers adequate means for declaratively specifying processes in a way that fosters process recognition from robot observations. We demonstrate how a mobile observer can recognize various processes in a warehouse based on sensor perception backed up by a formal process specification. The second contribution of this work is to show that logic reasoning can be performed with the declarative process specifications and observations, enabling the robot sensibly to supplement missing pieces of information.

This paper is organized as follows. We first point out connections to existing work and we discuss reasons for choosing a logic-based formalism (Section 2.2). In Section 2.3, we briefly introduce LTL and summarize its important features. Thereafter, we describe our formalization of in-warehouse processes (Section 2.4) which consist of a domain axiomatization and an appropriate grounding of logic primitives. Section 2.5 presents our system realization, followed by an experimental evaluation (Section 2.6). We discuss our results (Section 2.7) and conclude with some final remarks (Section 2.8).

2.2 Related Work

Many approaches have been used to tackle process recognition, which can roughly be categorized into learning approaches, probabilistic process descriptions, and logic-based declarative approaches.

Machine learning approaches such as Markov networks (Bennewitz et al., 2005; Liao et al., 2007), Bayesian networks (Yang, 2009), supervised learning (Balcan and Blum, 2010), or inductive logic programming (Dubba et al., 2011) require a training phase before deployment. By contrast, we are particularly interested in mastering contexts in which no training data is available beforehand. Our aim is to enable querying the robot's observations using a flexible formal language for specifying process descriptions. Thus, any process to be recognized could be specified on the fly and does not need to be known beforehand; also queries to the system can be changed flexibly without need of relearning.

Declarative, logic-based formalisms enable us to pose queries flexibly. Utilizing logics in robotics dates back to the first appearances of AI robotic research (recall, for example, seminal work related to Shakey (Nilsson, 1984)). More recently, Mastrogiovanni, Sgorbissa, and Zaccaria (2009) have been using a logic-based approach integrating ontologies to recognize contexts in a ubiquitous robotics setting, which relates to our process recognition task. Mastrogiovanni, Scalmato, et al. (2009) introduced a new formal language to specify these contexts. In their framework, time is represented by a series of discrete time steps such that a formula holds at a given time instant. Computing time than increases exponentially with the number of time steps considered, such that only a limited number of time steps can be maintained. In the approach we present in this paper, we avoid this shortcoming by representing time explicitly on the level of the chosen logic formalism, namely linear temporal logics (LTL). This reduces the complexity and yields linear complexity with respect to the number of time steps as we will show in Section 2.3.2.

Moreover, formalisms based on LTL or its extensions neatly integrate into other LTL-based approaches to robot control such as motion planning or construction of controllers. Kress-Gazit, Wongpiromsarn, and Topcu (2011) propose a method for constructing controllers from an LTL formula and they determine that mastering state explosion is a key challenge, i.e., to develop techniques that avoid generating more states than feasible. This problem arises as LTL formulas are naturally evaluated over infinite time sequences, hence they potentially involve infinitely many states. One practical approach is to employ a *receding horizon* (Wongpiromsarn, Topcu, and Murray, 2010; Wongpiromsarn, Topcu, and Murray, 2009; Kress-Gazit, Wongpiromsarn, and Topcu, 2011, e.g.), which aims to cut off irrelevant future states. In our work we use a similar approach to interpret queries over the finite sequence of observations available to the robot. Ding et al. (2011) propose a method to transform LTL specifications of processes into a control policy for Markov decision process, taking into account probabilities of successful action execution. This is accomplished in a way similar to model checking with a probabilistic temporal logic. Putting this into a more general context, probabilistic extensions of LTL, such as probabilistic temporal logic (Bianco and De Alfaro, 1995), are interesting. However, process detection with probabilistic logics requires the probabilities of process occurrence to be specified beforehand. In settings like ours where no training data is available to determine the required probabilities, probabilistic logics are hence not suitable.

Model checking is widely used in software verification, but it has important applications in robotics, as well. For example, planning can be posed as a model checking task (Cimatti et al., 1997; Edelkamp and Jabbar, 2006; Kloetzer and Belta, 2010, e.g.). Consider ϕ to be the specification of a plan to be fulfilled and M to be the set of all possible states a robot can be in. In this setting, verifying that M models ϕ means that we find a time linear sequence of robot states that meets the requirements of the specified plan. As we will show later, the same principle can be applied to process detection: the set of worlds M is then derived from sensor observations of the robot.

2.3 Linear Temporal Logic (LTL) for Process Detection

In classical propositional logic, formulas are evaluated with respect to a single fixed interpretation called *world*. Thus, a formula is either true or false. In order to acknowledge dynamic environments in which a proposition may hold for some limited time only, temporal logics utilize a set of worlds over which formulas are evaluated. Formulas may be satisfied in some worlds, but not in others. Linear temporal logic is a modal logic that extends propositional logic by a sequential notion of time. A formula ϕ in LTL is defined over a finite set of propositions with the usual logic operators ($\wedge$, $\vee$,$\neg$, $\rightarrow$). The temporal component is established by an accessibility relation R that connects the individual worlds (also called states with LTL) over which formulas are interpreted. In linear temporal logic, the relation R is a discrete linear ordering. We say that a world W is a future world of V if $(V, W) \in R$, i.e., W is reachable from V by R. LTL defines three unary modal operators on the basis of R:

$\circ\phi$ (next) ϕ holds in the following world
$\Box\phi$ (always) ϕ holds in the current world and in all future worlds
$\Diamond\phi$ (eventually) ϕ holds in some future world ($\Diamond\phi \leftrightarrow \neg\Box\neg\phi$)

One important reasoning task in logic is model checking: given a specification ϕ and a model which valuates the logic primitives, does the model satisfy ϕ?

2.3.1 Process Recognition as Model Checking

We describe a process by an LTL formula ϕ. Then, a process is said to be recognized if a model derived from the observations of the robot satisfies ϕ. Thus, model checking matches logic predicates with observations. Usual techniques for LTL model checking are based on translation of the formulas to either ω-automata or Büchi-automata. These automata are then used to process an infinite model. However, in process detection we are involved with finite models. For any given time point one can decide whether a complete process has been observed or not.

2.3.2 Computational Complexity of Model Checking

Sistla and Clarke (1985) show that the model checking problem of LTL is PSPACE-complete, but various fragments have a significantly lower complexity. Bauland et al. (2011) investigated various fragments of LTL and found tractable subsets with time complexity as low as NLOGSPACE-complete. Efficient subclasses either exclude the *eventually* operator or they do not allow the Boolean *and*. However, both operators add important expressiveness to process descriptions. Our formalization of warehouse processes detailed in Section 2.4.6 involves both operators. The resulting subclass of model checking which involves only $\Diamond$, $\wedge$, and $\neg$ is NP-complete (Sistla and Clarke, 1985; Bauland et al., 2011).

In a survey about complexity of temporal logic model checking Schnoebelen (2003) describes the influence of various factors. If the length of formulas is fixed, complexity of model checking is in NLOGSPACE with respect to model size. By contrast, if the model is fixed, then the complexity is in PSPACE with respect to varying length of the formulas. Lichtenstein and Pnueli (1985) show that model checking can be done in $2^{O(|\phi|)}O(|M|)$, i.e., model checking is linear-time with respect to the size of the model. This is an important result since in applications like process detection the model size grows with the amount of observations, but the length of the query formulas is fixed, assuming fixed process descriptions. In other words, the exponential growth with respect to formula length does not apply.

We consider an important variety of model checking. Consider formulas which disjunctively combine sub-formulas which only vary in one atom, i.e., formulas which can be written $\bigvee_{a\in A}\phi(a)$, whereby A denotes a set of atoms. If all $\phi(a)$ are within an NP-complete fragment of LTL model checking, the complete formula is also in NP as one can non-deterministically select a clause from the disjunction in polynomial time and then continue with model checking. Morgenstern and Schneider (2011) pursue a similar idea by introducing *oracle variables* which

"[...] may represent 'angelic' nondeterminism that may be resolved in favor to satisfy the specification."

We can thus apply LTL model checking to formulas that involve an extensional quantifier ranging over a set of atoms without increasing computational complexity further. This observation is important in our context since we are interested in querying for process occurrences and queries naturally involve unknowns. For example, one would rather query whether a good was moved to *some* place within the storage area, rather than querying whether a specific good G was moved to a specific location L. With respect to computational complexity of querying we note that the procedure can be carried out in NP given that the set of atoms to consider is fixed, i.e., our domain is not expanding. Considering our warehouse domain we naturally have a fixed set of locations but a potentially growing set of goods. However, within a limited amount of time the amount of goods visiting a warehouse can be regarded to remain constant.

2.4 Specification and Interpretation of In-Warehouse Processes

In the following, we describe a case study of warehouse logistics in which a mobile robot observes processes in a warehouse. The robot can later be queried by a logistic expert who is involved with improving storage strategies. We use LTL to describe relevant storage processes and their functional components. Both temporal and spatial primitives used in the logic are grounded in the observations of the robot. We conclude this section by a small example.

2.4.1 Scenario

We address the problem of understanding so-called *chaotic* or *random-storage warehouses*, characterized by a lack of predefined spatial structures, that is, there is no fixed assignment of storage locations to specific goods. Thus, storage processes are solely under the responsibility of the warehouse operators and basically are not predictable: goods of the same type may be distributed over various locations and no data base keeps track of these locations. This makes it a hard problem for people aiming to improve the internal storage processes. We are interested in representing the spatio-temporal change that occurs in the warehouse, but we are not interested in tracking individual movements. Therefore, we can assume the environment to be piecewise static.

On a conceptual level, storage processes are defined by a unique pattern (Ten Hompel and Schmidt, 2010): on their way into and out of the warehouse, goods are (temporarily) placed into *functional zones* which serve specific purposes (see Fig. 2.1). All goods arrive in an *entrance zone* (E). From there, they are picked up and temporarily moved to a *buffer zone* (B) before they are finally stored in the *storage zone* (S). This process is called 'admission'. Within the *storage zone* 'redistribution' of goods can occur arbitrarily. When 'taking out' goods, they are

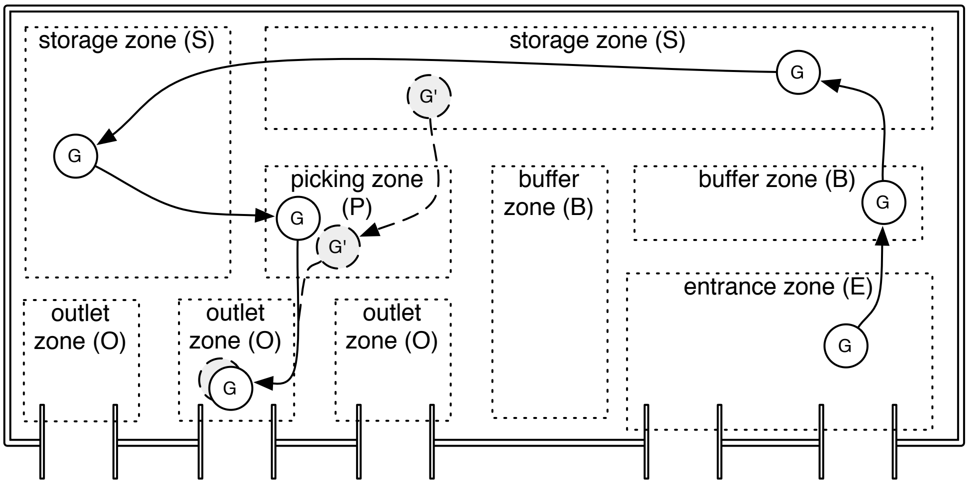


Figure 2.1: A warehouse, its functional zones, and typical movements of different goods (G, G').

first moved from the *storage zone* to the *picking zone* (P) from where they are taken to an *outlet zone* (O), before being moved out of the warehouse.

A mobile robot observing such a warehouse is not able to perceive these zones directly, as they are not marked. For all zones we know that they exist (that is, that such regions are used within the storage operations), but neither their concrete spatial extents nor the number of their occurrences is known. This information solely depends on the dynamic in-warehouse storage processes. The robot can detect and identify goods and it can estimate their position. We thus face the challenge that for detecting concrete storage processes we need to rely on knowledge about functional zones which is not yet available. For example, if a robot perceives a good at three different locations the process interpretation largely depends on the zones of the locations. If all locations are in the storage zone, a redistribution may have occurred, whereas if all locations are in different zones an admission or a take-out process may have occurred.

2.4.2 Formalizing the Warehouse Scenario

In this section we explain the formalization of processes and general background knowledge in terms of spatio-temporal integrity constraints like, for instance, the fact that objects can only be at one location at a time. To this end we need to compose LTL formulas which capture the characteristics of spatio-temporal processes. These formulas serve as axioms and are used to enforce a sensible interpretation of the observations of the robot. To begin with, observations are mapped in a spatio-temporal grounding process to primitives of our logic. Our formalization is based on the following set of primitives:

goods: A set $\mathcal{G} = \{G_1, \ldots, G_n\}$ of goods constitutes the entities that move in space over time and determine the dynamics of the scenario. They are observable by the robot and their position can be estimated.

locations: A location is a property of a good which remains the same when a good is not moved. During spatio-temporal grounding, position estimates are abstracted to a discrete set of locations. For a spatially restricted scenario the set of locations $\mathcal{L} = \{L_1, \ldots, L_m\}$ is finite.

zones: The warehouse scenario involves *functional zones* $\mathcal{Z} = \{E, B, S, P, O\}$ as described in Section 2.4.1. The extent of a zone is defined by the set of locations it contains. Zones are considered to be fixed in our scenario, but their extent is a-priori unknown to the reasoning system.

2.4.3 Atomic Propositions for Spatio-Temporal Processes

Modeling with LTL involves devising a finite set of atomic propositions which capture relevant facts about the state of the world. Atomic propositions can either be determined by interpreting observations of the robot or by logic inference. We utilize the following atomic propositions which we denote in a predicate style for ease of readability, i.e., the atom at(G, L) stands for $|G| \cdot |L|$ atoms, one per combination of good G and location L.

- at$(G, L) \Leftrightarrow$ good G is at a location L.
  This type of atom is data-driven, that is, its value can directly be obtained from sensor observations of the robot. Proposition at(G, L) holds if and only if a good G is known to be at location L. Truth of this proposition can thus change over time if a good is moved.
- in$(L, Z) \Leftrightarrow$ location L is contained in a zone Z.
  As the set of locations is generated at runtime, in(L, Z) also depends on sensor perceptions. The interpretation of in(L, Z) remains constant over time.
- close$(L_1, L_2) \Leftrightarrow$ two locations L_1, L_2 are close to one another.
  We use closeness as a central concept to distinguish different zones. close(L_1, L_2) remains constant over time.

2.4.4 Spatio-Temporal Grounding

In LTL, time is represented as a sequence of independent worlds. A temporal interpretation can thus be achieved by sampling the observations of the robot. To this end, we make use of the perception loop of the robot. During each cycle, sensors are read and localization and mapping are updated. The updated information is then used to determine which of the atomic propositions currently holds. Since our domain does not require us to state that nothing has changed, we can reduce the set of worlds emitted by temporal grounding. A new world is only generated if the interpretation of at least one atomic proposition changes.

One central task of spatio-temporal grounding is the robust interpretation of position estimates $(x, y) \in \mathbb{R}^2$ to discrete locations $L_i \in \mathcal{L}$. Naturally, position estimates are subject to noise and may vary over time even if an object does not move. Additionally, by keeping the size of the set of locations minimal, we can minimize the set of atomic propositions and thereby limit the size of our formulas. This can be accomplished by updating the set of locations at runtime, adding new locations only if necessary. In our implementation we apply a clustering approach that takes uncertainty of estimates into account (see Section 2.5.2). It provides us with a compact and robust interpretation, but other methods would be possible too. A requirement is however that the mapping from positions to locations is stable over time, i.e., if a good G is said to be located at L_i then this interpretation shall not be revised when observations are integrated into the localization procedure. With $\mathcal{L}$ available, at(G, L) can be directly derived for every time step. Furthermore, close(L_1, L_2) is valuated by applying a metric and checking whether the distance between L_1 and L_2 is below a certain threshold (in the experiments in this paper, we use an Euclidean distance of 1 meter). The propositions in(L, Z) can be set if knowledge about zones is available a-priori, otherwise they need to be inferred by reasoning.

2.4.5 Spatio-Temporal Integrity Constraints

Commonsense knowledge about spatio-temporal processes in our domain is captured by the following set of axioms which enforce a sensible interpretation of data available. While processes and related queries can be freely specified, axioms remain the same over any process detection task. Explicating this knowledge in axioms separately allows us to keep the process specification simple and intuitive. We define the following four axioms:

- Locations are fixed, i.e., if two locations are close to one another they are always close to one another.

$$A1_{L_i,L_j} = \text{close}(L_i, L_j) \rightarrow \Box\text{close}(L_i, L_j) \tag{A1}$$

- A good G can only be at one location at a time.

$$A2_G = \Box \bigwedge_{L_i \neq L_j} \neg\big(\text{at}(G, L_i) \wedge \text{at}(G, L_j)\big) \tag{A2}$$

Axioms A1 and A2 describe common-sense spatio-temporal constraints. Their fulfillment is guaranteed by the spatio-temporal grounding process and therefore can be omitted in the reasoning process. The next two axioms need to be addressed explicitly during model checking:

- A location $L \in \mathcal{L}$ always belongs to the same zone and is exactly in one zone $Z \in \mathcal{Z}$. In other words, zones are static and do not overlap.

$$A3_L = \bigvee_{Z \in \mathcal{Z}} \Box\Big(\text{in}(L, Z) \wedge \big(\bigwedge_{Z' \in \mathcal{Z} \setminus \{Z\}} \neg\text{in}(L, Z')\big)\Big) \tag{A3}$$

- Locations in different zones are not close to one another, that is, zones are at least some minimum distance apart. We note that it is still possible that two locations which are not close to one another can belong to the same zone (multiple occurrences of zones).

$$A4_{L_i,L_j} = \Box\big(\text{close}(L_i, L_j) \rightarrow \bigvee_{Z\in\mathcal{Z}} (\text{in}(L_i, Z) \wedge \text{in}(L_j, Z))\big) \tag{A4}$$

In the following we use $\mathcal{A}'$ to refer to axioms (A3)–(A4) that are essential for process recognition. In conduction with all propositions 'close' and 'in' which are static over all worlds (also constituted by (A1) and (A3)) this forms the set

$$\mathcal{B} = \mathcal{A}' \cup \bigcup_{L_i,L_j\in\mathcal{L}} \text{close}(L_i, L_j) \cup \bigcup_{L\in\mathcal{L},Z\in\mathcal{Z}} \text{in}(L, Z) \tag{2.1}$$

that we call the *background knowledge* of the warehouse domain.

In situations where further knowledge about zones is available, the axioms can be modified to accommodate such a-priori knowledge, for example by adding appropriate propositions in(L, Z) to the set of axioms. In our evaluation we make use of such modifications to $\mathcal{B}$ in order to study the effectiveness of inferring zone membership by reasoning.

2.4.6 In-Warehouse Processes

We now formalize in-warehouse processes. In particular, we define *admission*, *take-out*, and *redistribution* of goods. All processes are specified using the following schema:

$$\text{start condition} \wedge \Diamond(\text{next state condition} \wedge \Diamond(...)). \tag{2.2}$$

The schema solely captures the characteristic states of a process. This ensures a robust detection of processes which can vary in the level of detail.

- *Admission* – a good G is delivered to the warehouse's entrance zone E and moved to the storage zone S via the buffer zone B. For all $G \in \mathcal{G}$ and $L_i, L_j, L_k \in \mathcal{L}$:

$$\begin{aligned} &\text{Admission}_{G,L_i,L_j,L_k} = \text{at}(G, L_i) \wedge \text{in}(L_i, E)\wedge \\ &\Diamond\Big(\text{at}(G, L_j) \wedge \text{in}(L_j, B) \wedge \Diamond\big(\text{at}(G, L_k) \wedge \text{in}(L_k, S)\big)\Big) \end{aligned} \tag{2.3}$$

- *Take-out* – a good G is moved from the storage zone S to the outlet zone O via a picking zone P. For all $G \in \mathcal{G}$ and $L_i, L_j, L_k \in \mathcal{L}$:

$$\begin{aligned} &\text{Takeout}_{G,L_i,L_j,L_k} = \text{at}(G, L_i) \wedge \text{in}(L_i, S)\wedge \\ &\Diamond\Big(\text{at}(G, L_j) \wedge \text{in}(L_j, P) \wedge \Diamond\big(\text{at}(G, L_k) \wedge \text{in}(L_k, O)\big)\Big) \end{aligned} \tag{2.4}$$

- *Redistribution* – a good G is moved within the storage zone S. For all $G \in \mathcal{G}$ and $L_i, L_j \in \mathcal{L}$, $i \neq j$:

$$\begin{aligned} &\text{Redistribution}_{G,L_i,L_j} = \text{at}(G, L_i) \wedge \text{in}(L_i, S)\wedge \\ &\Diamond\big(\text{at}(G, L_j) \wedge \text{in}(L_j, S)\big) \end{aligned} \tag{2.5}$$

2.4.7 Inferring Functional Zones

The axioms and the process specifications make use of *functional zones* like entrance or buffer. While some zones may be known beforehand, others are not known and need to be inferred. Zones are characterized by the functional role they take, for example, an outlet zone is the set of locations in which a good can be seen last before it is taken out of the warehouse. Thus, identifying zones is the task of identifying a set of locations close to one another which all take the same functional role in the storage processes observed. This task can be viewed as model checking: do the observations provide a model for a hypothesis that a set of locations takes a specific functional role? During model checking variables in a process description are instantiated with observations. This includes that the locations involved in process descriptions are assigned to zones. In other words, inference of functional zone happens naturally during model checking. For example, if trying to verify that an admission has taken place, at least one location L_e is required to belong to an entrance zone. Axiom (A4) further enforces that all locations close to each other belong to the same zone. Technically speaking, zone membership is ruled by the transitive closure of the *close* relation on locations. This leads to an interpretation that is consistent with all processes detected.

2.4.8 Histories and Complex Process Queries

One piece of good can participate in many processes. Goods enter a warehouse in an admission process, they possibly get redistributed a couple of times before they eventually leave the warehouse in a takeout process. We call the sequence of processes a good participates in the *history* of the good. In our domain histories naturally begin with an admission and end with a takeout. Analogous to process detection, histories can also be detected in an atomic manner, i.e., LTL allows us to pose complete histories as a single query. This can be accomplished by using the same basic schema as shown in Equation (2.2). We give three examples of complex queries to highlight the generality of the LTL-based approach to process detection by model checking:

- Has a good G been moved back and forth?

$$Q_1 = \text{at}(G, L_i) \wedge \Diamond\big(\text{at}(G, L_j) \wedge \Diamond\text{at}(G, L_i)\big) \wedge \overbrace{\neg(\text{at}(G, L_i) \wedge \text{at}(G, L_j))}^{L_i \neq L_j} \tag{2.6}$$

- Has a good G been redistributed k or more times?

$$Q_2 = \phi_{L_{i,1}} \wedge \underbrace{\Diamond(\phi_{L_{i,2}} \wedge \Diamond(\phi_{L_{i,3}} \wedge \Diamond(\cdots)))}_{k \text{ nestings}} \tag{2.7}$$

with

$$\mathrm{Redistribution}_{G,L_k,L_l} = \overbrace{\text{at}(G, L_k) \wedge \text{in}(L_k, S)}^{=\phi_{L_{i,j}}} \wedge \Diamond \overbrace{(\text{at}(G, L_l) \wedge \text{in}(L_l, S))}^{=\phi_{L_{i,j+1}}}$$

- Do two goods G_i, G_j with $G_i \neq G_j$ that have been observed together once remain co-located?

$$\begin{aligned} Q_3 =& \big(at(G_i, L_k) \wedge at(G_j, L_l) \wedge \mathrm{close}(L_k, L_l)\big) \\ \rightarrow \Box\Big(& \overbrace{\bigvee_{\substack{L_m,L_n \\ \mathrm{close}(L_m,L_n)}} \big(\mathrm{at}(G_i, L_m) \wedge \mathrm{at}(G_j, L_n)\big)}^{G_i,G_j \text{ are observed together...}} \\ & \underbrace{\vee \neg \bigvee_{L_o} \big(\mathrm{at}(G_i, L_o) \vee \mathrm{at}(G_j, L_o)\big)}_{\text{...or neither is observed}} \Big) \end{aligned} \tag{2.8}$$

By conjunctively joining process specifications, arbitrarily complex queries can be stated. Prefixing a process in a conjunctive query by the eventual operator ($\Diamond$) states, that the process may happen any time, independent of the other processes. Joint queries are important to enable reasoning across histories. By conjunctively combining several history queries and prefixing them by $\Diamond$ we obtain a single query for the existence of a model that satisfies all processes involved (*joined histories*). In particular, this leads to a jointly compatible interpretation of functional zones. For example, during individual queries in one of the solutions a location might be interpreted to be a buffer area while during querying for a different good it is interpreted as part of an entrance. In the case of jointly querying for both histories, the same location cannot be part of different zones as of axiom (A3). Therefore, we only obtain histories as a result that satisfy a process specification using the same interpretation of location-zone membership. This results in more robust interpretation but comes at the cost of higher computing time due to increased formula size.

2.4.9 Example

A good G enters the warehouse and is stored in the entrance zone E at position L_1 at time t_0. Between t_1 and t_2 the good is moved to a location L_2 and between t_2 and t_3 the good is moved further to L_3. Let us assume that this process is observed as follows: We perceive G to be at L_1 at t_1, at L_2 at t_2 and at L_3 at t_4. Furthermore, all these locations are not close to one another. See Fig. 2.2 for a depiction and the logic interpretations—to ease understanding the worlds are labeled by time points. These observations constitute a model that satisfies (2.3), i.e., the observed process is an admission that starts in world t_1 and ends in world t_4. By inference it follows that location L_2 is contained in the buffer zone and L_3 is contained in a storage zone. Note that detected start and end times differ from the real admission times: while the admission takes place from t_0 to t_3, we detect it from observations t_1 to t_4; this is due to incomplete observation of the world.

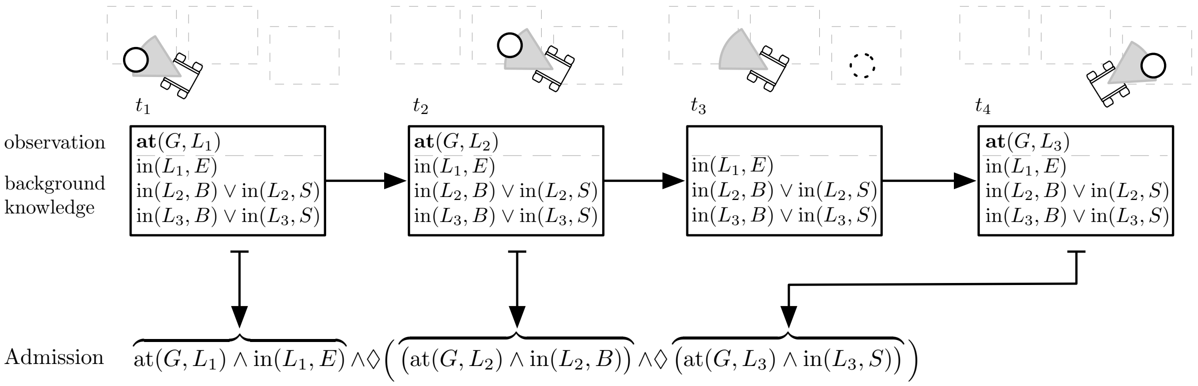

t_1
t_2
t_3
t_4
observation
background knowledge
$\mathbf{at}(G, L_1)$
$\text{in}(L_1, E)$
$\text{in}(L_2, B) \vee \text{in}(L_2, S)$
$\text{in}(L_3, B) \vee \text{in}(L_3, S)$
$\mathbf{at}(G, L_2)$
$\text{in}(L_1, E)$
$\text{in}(L_2, B) \vee \text{in}(L_2, S)$
$\text{in}(L_3, B) \vee \text{in}(L_3, S)$
$\text{in}(L_1, E)$
$\text{in}(L_2, B) \vee \text{in}(L_2, S)$
$\text{in}(L_3, B) \vee \text{in}(L_3, S)$
$\mathbf{at}(G, L_3)$
$\text{in}(L_1, E)$
$\text{in}(L_2, B) \vee \text{in}(L_2, S)$
$\text{in}(L_3, B) \vee \text{in}(L_3, S)$
Admission
$\overbrace{\text{at}(G, L_1) \wedge \text{in}(L_1, E)} \wedge \Diamond \Big(\overbrace{(\text{at}(G, L_2) \wedge \text{in}(L_2, B))} \wedge \Diamond \overbrace{(\text{at}(G, L_3) \wedge \text{in}(L_3, S))} \Big)$

Figure 2.2: Example: Model checking for an admission process of good G (only the relevant assertions for each world $t_{1...4}$ are shown). $\text{in}(L_1, E)$ is background knowledge, also it is known that locations L_2 and L_3 are either part of the buffer zone (B) or the storage zone (S) but not close to one another so that they do not have to belong to the same zone. From this admission refined knowledge about the buffer and storage zones can be inferred: $\text{in}(L_2, B) \wedge \text{in}(L_3, S)$.

2.5 System Realization

Our implementation of logic-based process recognition essentially consists of three parts: perceptions by the robot, their symbolic interpretation and process understanding using the high-level process model. The system architecture is shown in Fig. 2.3. The first part, perception, is integrated with our robot control software and its main objective is to localize the robot and to provide an up-to-date map of the changing warehouse. Essentially, we utilize a feature-based SLAM to map the environment, using tag-based good identification. Perceptions then lifted to the symbolic level. By evaluating the posterior probability of changes in landmark positions at each time step we determine whether a good was moved and can update the map accordingly, keeping track of all re-locations. During symbolic grounding the position estimates obtained from the robot map are also clustered to discrete qualitative locations that are then employed to describe the trajectory of good movements. We refer to the output of the symbolic grounding stage as *qualitative observations*. Process recognition is realized by the symbolic reasoning component that matches qualitative observations against the process descriptions or process queries, supplementing the qualitative observations by inferred knowledge.

2.5.1 Perception—Localizing and Mapping Goods

Localization and mapping of goods is realized as a feature-based SLAM using visual tags attached to both goods and the environment (see Fig. 2.5). We use the ARToolKit software[1] for identifying tags in camera images. This toolkit provides us with a tag identifier and with a 3D

[1] http://artoolkit.sourceforge.net/

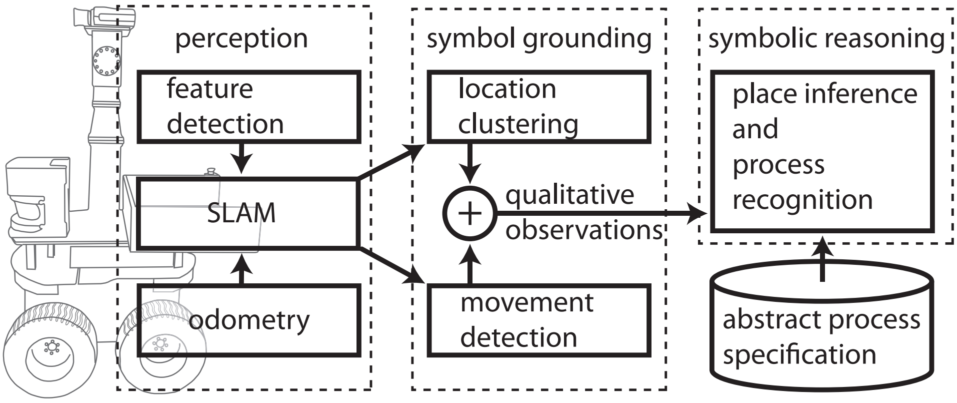


Figure 2.3: System architecture of robotic platform and reasoning.

projection matrix that estimates the tag position and its orientation relative to the camera. We project this transformation to the 2D ground plane in order to obtain a bearing-and-distance estimate that is used with the SLAM system. To this end, we determine a measurement model for our camera. This model is coarsened to mimic RFID-scanners that would be typical in an industrial context. By only working with tags positioned in the same height, a simple projection suffices to calculate the 3D to 2D transformation. ARToolKit also provides a quality of recognition which we use to gate tag recognition, discarding any identifications with less than 80% quality.

Feature-based SLAM is accomplished by a modified version of the TreeMap system (Frese, 2004) for simultaneous localization and mapping in static environments. TreeMap estimates the position of 2D landmarks by a least square approach, assuming a Gaussian noise model for odometry and observations. Originally, TreeMap does not grant access to covariances. We extended TreeMap to provide us with covariances for position estimates of landmarks. This allows movement detection in an uncertainty-sensitive manner by determining the Mahalanobis distance between the position of an observed landmark and its position given by the map, using both covariance of observations according to the measurement model and covariance of the position estimate. A movement is said to be detected if the Mahalanobis distance exceeds a fixed threshold of 1.9. Moved goods are re-entered into the SLAM system using a new identifier. By keeping track of the different identifiers used to refer to a single landmark we can reconstruct its trajectory.

2.5.2 Symbol Grounding—From Perception to Qualitative Observations

Position estimates in the robot map are clustered immediately before the logic-based process detection is invoked in order to obtain a small and robust set of locations which describe positions of goods in LTL. We use a straightforward centroid clustering that iteratively processes position estimates generated by the SLAM system. A new cluster is generated whenever a position does not fit into any cluster already established. For every cluster we generate a location L and valuate the atoms at(G, L) and close(L, L') (distance between locations less than 1m) accordingly. Clusters are limited in size to a circle of 0.25m radius as they represent single qualitative locations only. The iterative clustering method may not yield the most sensible interpretation of locations, but it ensures that the assignment from positions to clusters and thus locations will not be revised if new observations are available. This avoids detection of spurious movements and ensures monotony of the reasoning process (cp. Section 2.4.4). To study the effects of the autonomous clustering method we employ an additional method that uses pre-defined centroids and allows us to test ground truth data for the centroids.

All in all, we obtain a time-discrete sequence of observations, e.g., $t_0 : \{at(a, l_1), at(b, l_2)\}$, $t_1 : \{at(b, l_1)\}$, and so on. To construct the sequence of qualitative observations, repetitive time points are collapsed into a single qualitative state, i.e., we omit all observations which share the same set of $at(\cdot)$ atoms as the preceding observation.

2.5.3 Symbolic Reasoning—Process Understanding with Qualitative Observations

For performing process recognition we utilize the modeling language of *answer set programming* (ASP). Based on its roots in logic-based knowledge representation and non monotonic reasoning, databases, satisfiability testing and logic programming, ASP offers high-performance tools while providing us with a rich yet simple modeling language. The semantics of ASP is based on the *stable model* semantics (Lifschitz, 1996; Lifschitz, 2002).

LTL semantics can easily be achieved with ASP. First, qualitative observation atoms are attributed with the world in which they hold, for example at(G, L) becomes at(W, G, L). Second, the modal operators are realized. The *next* operator is realized as a preposition on worlds, i.e., we add $next(W_i, W_{i+1})$ if W_{i+1} directly follows W_i. *future* is realized as a recursively defined preposition utilizing the next operator. Then, process specifications and queries are rewritten accordingly. Axioms are modeled as constraints and free variables occurring in queries are represented by choice rules in ASP. We use GRINGO for grounding and CLASP as ASP solver(Gebser, Kaufmann, et al., 2007; Gebser, Kaminski, et al., 2011)[2].

In general, one set of observations can be interpreted differently in terms of which histories could have occurred. Consider the example of moving a good from A to B and further to C. This clearly satisfies the model Redistribution$_{G,A,B} \wedge$ Redistribution$_{G,B,C}$, but it also satisfies Redistribution$_{G,A,C}$. The latter interpretation ignores the observation that the good visited location B. In such cases we select the maximal model in the sense of selecting the history that involves the largest number of processes—this can also be performed by the ASP solver. The example is thus interpreted as two redistributions.

[2] as provided at `http://potassco.sourceforge.net`

scenario	#goods (histories)	#processes	duration [m:s]	#observations	largest joined history[1]
A	1	2	5:24	5	0.0s (±0.0)
B	4	8	8:08	70	1.0s (±0.1)
C	4	8	10:42	125	8.5s (±0.8)
D	4	8	11:56	188	41.1s (±4.3)
E	4	8	13:08	192	55.2s (±6.8)
F	4	8	14:35	71	0.8s (±0.5)
G	4	8	15:48	95	3.5s (±0.5)
H	4	10	10:35	197	71.8s (±11.1)
I	4	10	18:03	107	3.5s (±1.6)
J	4	10	18:38	177	37.1s (±6.5)
K	11	24	29:00	326	23.7s (±2.4)
L	12	32	34:06	473	152.7s (±21.7)

[1] Averaged over varying zone knowledge: full, partial, and no previous knowledge. For a definition of *joined history* see Section 2.4.8.

Table 2.1: Scenarios evaluated, their characteristics with respect to problem size, and compute time for the symbolic process recognition

2.6 Experiments and Evaluation

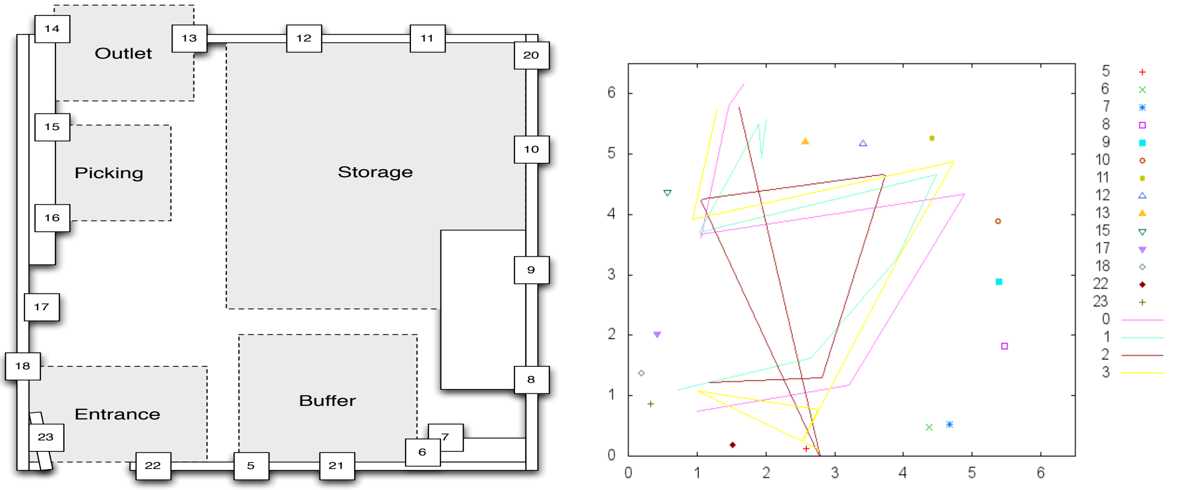
In our experimental setting we simulate warehouse processes in our lab. We measure how many histories, i.e., chains of processes per good, can be identified correctly. The numbers are further detailed to study the ability of the symbolic component to counteract absence of process knowledge. Also, we analyze the computing time of the symbolic reasoning component.

2.6.1 Experimental Setup

Our experimental robot platform consists of an Active Media Pioneer 2-DX (differential drive) controlled by a top-mounted laptop and equipped with a SONY DFW SX900 (approx. 160° FOV) camera that delivers 7.5 frames per second.

We simulate a warehouse that consists of five dedicated zones (entrance, buffer, storage, picking, outlet) as depicted in Fig. 2.1 and 2.4(a). Each good is labeled with a unique visual tag as shown in Fig. 2.5 (rectangular shapes on paper sheets). For tag identification, we rely on the ARToolKit. We distribute 17 tags as static landmarks over the environment in order to ease robot localization.

One run of the experiment consists of a series of movements of goods between the zones while our robot is monitoring the environment. The location of all tags is determined and we record which processes happen to obtain ground truth data for evaluation. For each of the 12



(a) Warehouse layout with positions of static landmarks.

(b) SLAM map for experimental run I consisting of 4 trajectories for goods $(0, 1, 2, 3)$ and the static landmarks $(5..23)$

Figure 2.4: Experimental setup: warehouse in the lab (6.12 m × 7 m).

scenarios performed, the robot was manually driven around the test environment until each landmark has been seen at least once to ensure a robust localization. Then, we steered the robot in random courses, while we moved boxes through the lab, simulating the previously defined logistic processes (Sect. 2.4.6). The duration of a single run was between approx. 5 and 34 minutes in which we moved 1 to 12 goods through the warehouse, resulting in 2 to 32 detectable processes (admission, redistribution, take-out) per run. Details are shown in Table 2.1. Goods were moved between zones while not covered by sensor surveillance to comply with Axiom (A2) in Section 2.4.2. Data gathered by the robot is processed as described above to obtain good trajectories (see Fig 2.4(b) for an example depicting the movement of our goods) that are then interpreted in terms of qualitative observations and passed to the symbolic process recognition to recognize histories of all goods. We say that a history is correct if all detected processes and their temporal order matches with the ground-truth.

2.6.2 Evaluation

For the evaluation we perform 12 experimental process scenarios, resulting in a total of 60 histories, one for each ware. We record all intermediate processing results. Additionally, we record ground truth information. The characteristics of these scenarios with respect to problem size are shown in Table 2.1. In the column ‘# observations’ we give the number of qualitative observations obtained by the automatic clustering. The computing times for the symbolic processing also refer to the fully automatic clustering method. Fig. 2.7 further presents the computing times for queries obtained with respect to the number of qualitative observations as this is the essential factor of the computing time.

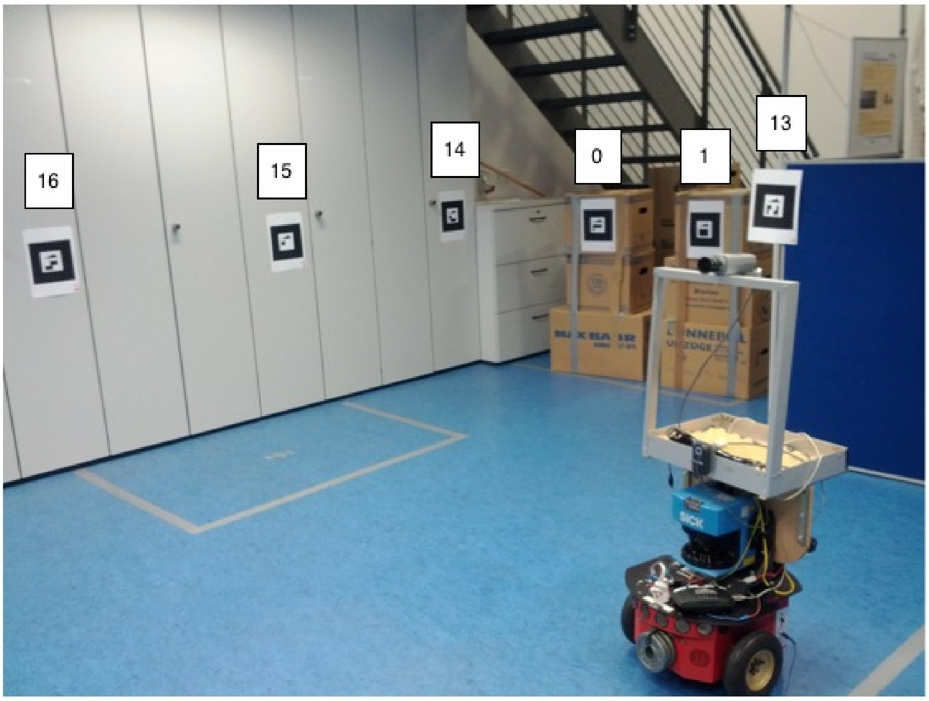
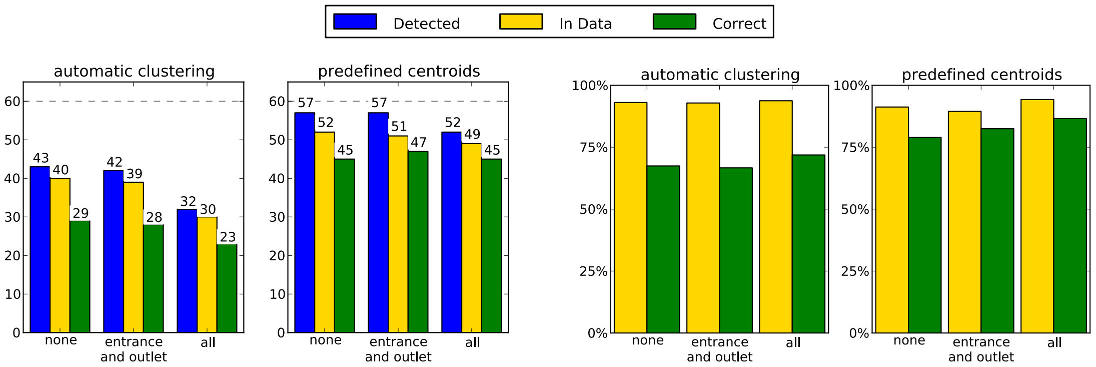


Figure 2.5: Warehouse mockup equipped with AR-Tags and a Pioneer 2-DX (see also Fig. 2.4(a))

We determine the total number of correctly identified histories across all scenarios, breaking up the numbers into the availability of zone knowledge (all location to zone mappings known, only entrance and outlet known, no zone known) and by the spatial grounding method used (automatic clustering vs. pre-defined centroids). Fig. 2.6(a) shows the results obtained and graphically presents the percentage of histories recognized depending on these factors. Furthermore, the plots in Fig. 2.6 also show the relative amount of correct histories verifiable by qualitative observations *in the data*. To obtain this measure we query the qualitative observations using the ground-truth.

2.7 Discussion

We first consider quality of process recognition. Looking at Fig. 2.6, the *in data* bars represent the relative amount of histories correctly verifiable matching the qualitative observations against ground truth. In other words, this bar represents how well the real-world process is captured by the robot observations, their symbolic interpretation, and the overall process description. This bar can thus be considered a *gold standard* for the actual process recognition algorithm. There are several reasons why the gold standard does not reach the 100% mark, for example good histories could not be reconstructed correctly (small movements easily remain unrecognized



(a) Evaluation results

(b) Relative quality of detected histories

Figure 2.6: Results from the experimental evaluation. (a): Showing the number of histories detected, how many are supported by the data, and how many are correctly recognized. (b): Percentage of histories supported by the data vs. correctly detected histories while optimizing for maximal history length; the columns are normalized by detected histories.

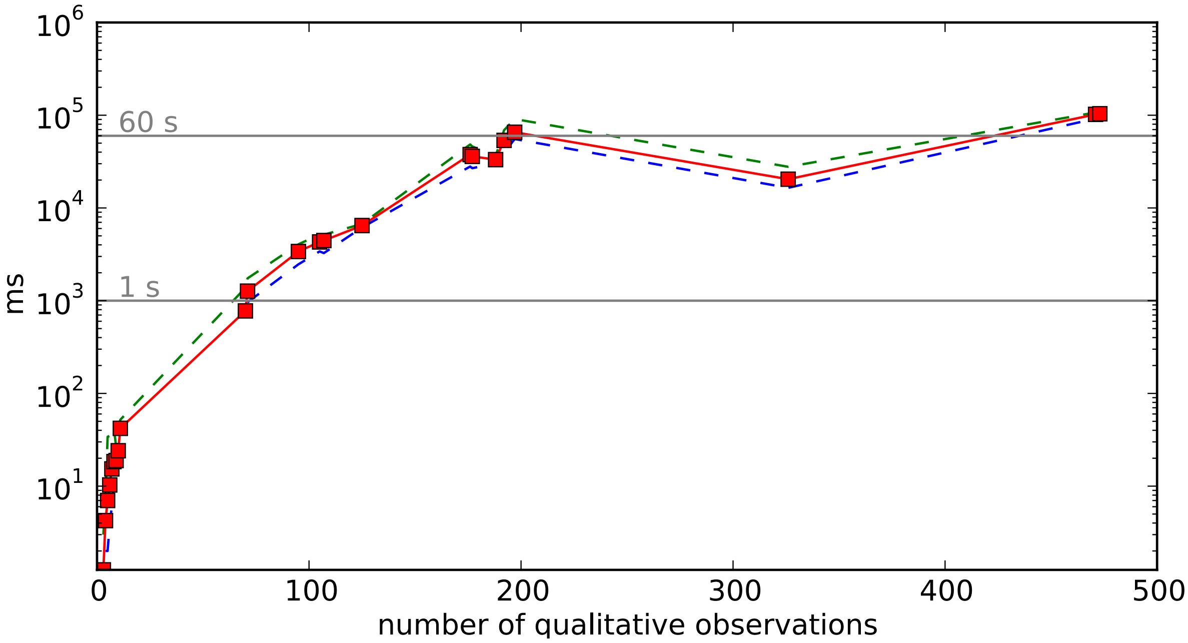


Figure 2.7: Plot of the computing times (in log scale) for symbolic process recognition vs. number of qualitative observations.

by the mapping component), or the robot may have simply overlooked essential information. We obtain around 61%/84% recognition rate for this gold standard, which indicates that the symbolic process descriptions, the qualitative interpretation of sensor data, and the integration with the robotic system provides us with an adequate foundation. The *correct* bars in Fig. 2.6 present the absolute/relative correct recognitions achieved by our process recognition algorithms. Naturally, the performance is less than that of the gold standard and we observe a difference in performance comparing the automatic clustering method against clustering with pre-defined centroids. This difference indicates the importance of a sensible spatial grounding and motivates further research to obtain more sophisticated automated methods. As expected, we observe that with increased background knowledge the relative number of histories matching the ground truth increases while the total number of detectable processes decreases. The reason for this is due to the fact that providing more background information restricts the way the data can be interpreted, leading to fewer interpretations that meet a process description.

In some scenarios when all regions are known it occurred that some locations are not within any zone, thereby violating Axiom (A3) and hence inhibiting recognition at all. This is especially true in the case of the automatic clustering as can be seen by the drop in the case when all regions are known. Overall the increase in background knowledge reduces the amount of false positives while having little impact on the number of correctly detected histories.

The most important observation is however that the relative number of correctly identified histories, i.e., how many from the detected histories are correct, is hardly affected by the amount of background knowledge available about zone membership. In case of pre-defined cluster centers the average relative recognition rate is 83% whereas for the case of automatic clustering the average relative recognition is 69%. Missing zone membership information is compensated for by logic reasoning during model checking which determines the unknown variables sensibly. In other words, the inference process is capable of supplementing all missing zone membership information to the process recognition process. This demonstrates that a logic-based approach is a valuable contribution to process recognition methods.

We note that these results confirm a previous study with respect to the overall conclusion, absolute recognition rates have improved though (cp. Kreutzmann et al. (2011)), in the case of pre-defined clusters from 68% achieved previously to about 76% in this study (for automatic clustering from 42% to 44%). The relative recognition rates, i.e., how many of the detected histories are correct ones, has improved even more: in case of pre-defined clusters from 73% to 83% and in case of automatic clustering from 57% to 69%. This improvement is due to three changes: first, we changed the robot hardware from a Pioneer 3-AT (4 wheel skid steering drive) to a Pioneer 2-AT (differential drive) as slip and drift for the 3-AT robot are very high on the lab floor. Second, we changed the visual tags for landmarks and goods as we have previously been suffering from mixups in tag detection. Last but not least we extended the TreeMap SLAM algorithm[3] to provide uncertainty estimates. By exploiting covariances for position estimates from map and observation we are able to detect movements more robustly which increases the overall map quality too.

[3] as provided at `http://openslam.org/TreeMap.html`

2.8 Conclusion

In this paper we propose an approach to process detection based on a specification of processes as temporal logic formulas in LTL. We demonstrate the applicability of our approach by an evaluation with real sensory data from a mobile robot. In our case study of warehouse logistics, the observations of a robot can be queried for process occurrences using an abstract process description. This allows a domain expert to obtain valuable information.

The evaluation demonstrates usefulness of the LTL-based approach to process description and recognition. With LTL one takes a declarative approach that is accessible to any domain expert as the declarative formulas abstract from the details of underlying algorithms. The performance of the gold standard clearly demonstrate feasibility of the symbolic approach to process specification and recognition, confirming the first claim of this paper. The claim is further supported by the actual recognition rates of the autonomous process recognition. Let us now consider the second claim of this paper, namely that the declarative approach enables logic reasoning to supplement observations of the robot, sensibly filling in missing pieces of information. Indeed, the experimental setting in which no information about zone membership is available a priori resembles a chicken-and-egg problem: on the one hand side, zones need to be known in order to identify processes. On the other hand, the processes need to be known to identify zones. Approaching process recognition as a model checking problem allows us to jointly recognize processes and zones using the well-defined semantics of anwser set programming. Naturally, the less information is available the poorer the recognition rate. Fig. 2.6 however shows that the declarative approach effectively counteracts the loss of information, showing only a small decline despite loss of zone information. This demonstrates a key benefit of a logic-based approach: the seamless integration of inference processes into the robot control architecture. Last but not least, the approach is sufficiently efficient to handle real-world data. Two factors are the essential contributors: the qualitative representation cuts down comprehensive experimental runs to few observations (see Table 2.1) and the ASP solver exhibits a low-degree polynomial growth of computing time.

In a real-world warehouse we expect the robot to only observe a relative small amount of processes occurring, as a robot's perception is limited in range. Nevertheless, an analysis of the overall warehouse processes is still possible if the processes and histories detected are prototypical for the overall warehouse. This requires a high degree of correct recognitions though. As our approach meets this requirement we are confident that the approach will also scale to a large setting.

While the focus of this paper was to present an LTL-based approach to process recognition and understanding, we aim to extend this approach to a comprehensive LTL-based control strategy. An interesting next step is to automatically derive an observation strategy that generates a sensible surveillance behaviour. In particular, we aim to use temporal logic to incorporate so-called search control knowledge and perform high-level planning (Bacchus and Kabanza, 2000), i.e., we shift to *active* process detection in the sense of planning which places to observe in order gather most valuable information.

For some complex queries it would be helpful to address all knowledge gathered during observations, in particular information about goods we have observed before and which are included in the map, but which we are unable to perceive at the very moment. Currently, we take an conservative approach that only explicates knowledge that is certain. However, for such objects we still have a strong *belief* of their existence and position in the warehouse , but this belief can—according to the actual observation—not be validated. A possibility to include reasoning on such beliefs is to use a logic that provides a modal belief operator, such as the logic for BDI agents presented in (Meyer and Veltman, 2007). Another source of information for more complex queries could be provided by an ontology, as shown in (Mastrogiovanni, Sgorbissa, and Zaccaria, 2009).

References

Antoniotti, Marco and Bud Mishra (1995). “Discrete event models + temporal logic = supervisory controller: Automatic synthesis of locomotion controllers”. In: *Proceedings of the IEEE Conference on Robotics and Automation (ICRA)*. Vol. 2, pp. 1441–1446.

Bacchus, Fahiem and Froduald Kabanza (2000). “Using temporal logics to express search control knowledge for planning”. In: *Artificial Intelligence* 116.1-2, pp. 123–191. ISSN: 0004-3702.

Balcan, Maria-Florina and Avrim Blum (2010). “A Discriminative Model for Semi-Supervised Learning”. In: *Journal of the ACM (JACM)* 57.3, pp. 1–46. ISSN: 0004-5411.

Bauland, Michael, Martin Mundhenk, Thomas Schneider, Henning Schnoor, Ilka Schnoor, and Heribert Vollmer (Jan. 2011). “The tractability of model checking for LTL: The good, the bad, and the ugly fragments”. In: *ACM Transactions on Computational Logic* 12.2, 13:1–13:28. ISSN: 1529-3785.

Bennewitz, Maren, Wolfram Burgard, Grzegorz Cielniak, and Sebastian Thrun (2005). “Learning Motion Patterns of People for Compliant Robot Motion”. In: *The International Journal of Robotics Research (IJRR)* 24.1, pp. 39–41.

Bianco, Andrea and Luca De Alfaro (1995). “Model checking of probabilistic and nondeteministic systems”. In: *Foundations of Software Technology and Theoretical Computer Science*. Vol. 1026. LNCS. Springer, pp. 499–513.

Cimatti, Alessandro, Fausto Giunchiglia, Enrico Giunchiglia, and Paolo Traverso (1997). “Planning via Model Checking: A Decision Procedure for *AR*”. In: *European Conference on Planning (ECP)*. Ed. by Sam Steel and Rachid Alami. Vol. 1348. Lecture Notes in Computer Science. Springer, pp. 130–142. ISBN: 3-540-63912-8.

Ding, Xu Chu, Stephen L. Smith, Calin Belta, and Daniela Rus (Apr. 2011). “LTL Control in Uncertain Environments with Probabilistic Satisfaction Guarantees”. In: arXiv:1104.1159v2 [math.OC]. Technical Report accompanying IFAC 2011. eprint: 1104.1159v2.

Dubba, Krishna S. R., Mehul Bhatt, Frank Dylla, Anthony G. Cohn, and David C. Hogg (2011). “Interleaved Inductive-Abductive Reasoning for Learning Event-Based Activity Models”.

In: *Inductive Logic Programming - 21st International Conference, ILP 2011. Windsor Great Park, UK.* Lecture Notes in Computer Science.

Edelkamp, Stefan and Shahid Jabbar (Feb. 2006). "Action Planning for Directed Model Checking of Petri Nets". In: *Electronic Notes in Theoretical Computer Science* 149.2, pp. 3–18. ISSN: 1571-0661.

Frese, Udo (2004). "An $O(logn)$ Algorithm for Simulateneous Localization and Mapping of Mobile Robots in Indoor Environments". PhD thesis. University of Erlangen-Nürnberg.

Gebser, Martin, Roland Kaminski, Arne König, and Torsten Schaub (2011). "Advances in *gringo* Series 3". In: *Proceedings of the Eleventh International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR'11).* Ed. by James P. Delgrande and Wolfgang Faber. Vol. 6645. Lecture Notes in Artificial Intelligence. Springer-Verlag, pp. 345–351.

Gebser, Martin, Benjamin Kaufmann, André Neumann, and Torsten Schaub (2007). "Conflict-Driven Answer Set Solving". In: *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI'07).* Ed. by Manuela M. Veloso. Available at http://www.ijcai.org/papers07/contents.php. AAAI Press/The MIT Press, pp. 386–392.

Hildebrandt, Torsten, Lutz Frommberger, Diedrich Wolter, Christian Zabel, Christian Freksa, and Bernd Scholz-Reiter (June 2010). "Towards Optimization of Manufacturing Systems using Autonomous Robotic Observers". In: *Proceedings of the 7th CIRP International Conference on Intelligent Computation in Manufacturing Engineering (ICME).*

Kloetzer, Marius and Calin Belta (2006). "LTL planning for groups of robots". In: *Proceedings of the IEEE International Conference on Networking, Sensing and Control (ICNSC)*, pp. 578–583.

Kloetzer, Marius and Calin Belta (2010). "Automatic deployment of distributed teams of robots from temporal logic motion specifications". In: *IEEE Transactions on Robotics* 26.1, pp. 48–61.

Kress-Gazit, Hadas, Tichakorn Wongpiromsarn, and Ufuk Topcu (Sept. 2011). "Correct, Reactive Robot Control from Abstraction and Temporal Logic Specifications". In: *Special Issue of the IEEE Robotics and Automation Magazine on Formal Methods for Robotics and Automation* 18.3, pp. 65–74.

Kreutzmann, Arne, Immo Colonius, Lutz Frommberger, Frank Dylla, Christian Freksa, and Diedrich Wolter (Sept. 2011). "On Process Recognition by Logical Inference". In: *Proc. of the 5th European Conference on Mobile Robots (ECMR)*, pp. 7–12.

Lahijanian, Morteza, Sean B. Andersson, and Calin Belta (2011). "Temporal Logic Motion Planning and Control With Probabilistic Satisfaction Guarantees". In: *IEEE Transactions on Robotics* PP.99, pp. 1–14.

Liao, Lin, Donald J. Patterson, Dieter Fox, and Henry Kautz (2007). "Learning and Inferring Transportation Routines". In: *Artificial Intelligence* 171.5–6, pp. 311–331.

Lichtenstein, Orna and Amir Pnueli (1985). "Checking That Finite State Concurrent Programs Satisfy Their Linear Specification." In: *Proceedings of the 12th ACM SIGACT-SIGPLAN symposium on Principles of programming languages (POPL)*, pp. 97–107.

Lifschitz, Vladimir (1996). "Foundations of Logic Programming". In: *Principles of Knowledge Representation*. Ed. by Gerhard Brewka. CSLI Publications, pp. 69–128.

Lifschitz, Vladimir (2002). "Answer Set Programming and Plan Generation". In: *Artificial Intelligence* 138, pp. 39–54.

Mastrogiovanni, Fulvio, Antonello Scalmato, Antonio Sgorbissa, and Renato Zaccaria (Sept. 2009). "On Situation Specification in Context Aware Robotics Applications". In: *Proceedings of the 4th European Conference on Mobile Robots (ECMR)*. Mlini/Dubrovnik, Croatia, pp. 265–270.

Mastrogiovanni, Fulvio, Antonio Sgorbissa, and Renato Zaccaria (2009). "Context assessment strategies for Ubiquitous Robots". In: *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2717–2722.

Meyer, John-Jules and Frank Veltman (2007). "Intelligent Agents and Common Sense Reasoning". In: *Handbook of Modal Logic*. Ed. by Patrick Blackburn, Johan Van Benthem, and Frank Wolter. Vol. 3. Studies in Logic and Practical Reasoning. Elsevier. Chap. 18, pp. 991–1029.

Morgenstern, Andreas and Klaus Schneider (2011). "Program Sketching via CTL* Model Checking". In: *Model Checking Software (SPIN)*. Ed. by Alex Groce and Madanlal Musuvathi. Vol. 6823. LNCS. Snowbird, Utah, USA: Springer, pp. 126–143.

Nilsson, Nils J. (Apr. 1984). *Shakey The Robot*. Tech. rep. 323. 333 Ravenswood Ave., Menlo Park, CA 94025: AI Center, SRI International.

Pnueli, Amir (1977). "The temporal logic of programs". In: *Proceedings of the 18th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 46–57.

Schnoebelen, Philippe (2003). "The Complexity of Temporal Logic Model Checking". In: *Selected Papers from the 4th Workshop on Advances in Modal Logics (AiML'02)*. Ed. by Philippe Balbiani, Nobu-Yuki Suzuki, Frank Wolter, and Michael Zakharyaschev. Toulouse, France: King's College Publication. Chap. 19, pp. 393–436.

Sistla, Aravinda Prasad and Edmund M. Clarke (July 1985). "The complexity of propositional linear temporal logics". In: *Journal of the Association for Computing Machinery* 32.3, pp. 733–749.

Smith, Stephen L., Jana Tůmová, Calin Belta, and Daniela Rus (Oct. 2010). "Optimal path planning under temporal logic constraints". In: *Proceeding of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Taipeh, Taiwan, pp. 3288–3293.

Ten Hompel, Michael and Thorsten Schmidt (2010). "Management of Warehouse Systems". In: *Warehouse Management*. Berlin Heidelberg: Springer. Chap. 2, pp. 13–63.

Wongpiromsarn, Tichakorn, Ufuk Topcu, and Richard M. Murray (Dec. 2009). "Receding horizon temporal logic planning for dynamical systems". In: *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, pp. 5997–6004.

Wongpiromsarn, Tichakorn, Ufuk Topcu, and Richard M. Murray (2010). "Receding horizon control for temporal logic specifications". In: *Proceedings of the 13th ACM international conference on Hybrid systems: computation and control*. HSCC '10. Stockholm, Sweden: ACM, pp. 101–110. ISBN: 978-1-60558-955-8.

Yang, Qiang (July 2009). “Activity Recognition: Linking Low-level Sensors to High Level Intelligence”. In: *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI)*. Pasadena, CA, pp. 20–25.

3 Rule-Compliant Navigation With Qualitative Spatial Reasoning

Diedrich Wolter[1], Frank Dylla[1] and **Arne Kreutzmann**[1]

1 Cognitive Systems Group, University of Bremen, Bremen, Germany

Published in “Proceedings of the 4th International Robotic Sailing Conference”, 2011, Springer.

Contributions:

This paper resulted from intense discussions between the three authors, about how to best integrate my approach with possible control of a robotic sailboat. I introduced the idea of using a random roadmap planner. The study as well as the manuscript preparation was conducted by Diedrich Wolter. The modeling was based on Frank Dylla’s work and he kindly provided it.

Acknowledgements:

This work is supported by the Deutsche Forschungsgemeinschaft (DFG) in context of the transregional collaborative research center SFB/TR 8 *Spatial Cognition*, project R3-[Q-Shape]. Financial support is gratefully acknowledged. We also acknowledge the comments of the anonymous reviewers.

Abstract

We develop a formal, symbolic representation of right-of-way-rules for sea navigation based on a qualitative spatial representation. Navigation rules specified qualitatively allow an autonomous agent consistently to combine all rules applicable in a context. The focus of this paper is to show how the abstract rule specification can be used during path-planning. We propose a randomized-qualitative approach to navigation, integrating the symbolic level with a probabilistic roadmap planner. The resulting navigation system maneuvers under the side constraint of rule compliancy. Evaluating our approach with case studies we demonstrate that qualitative navigation rules contributes to autonomous sailing.

3.1 Introduction

A considerable amount of everyday behavior is not self-determined but subject to regulations. For example, right-of-way regulations govern how to travel public spaces. Action planning for an autonomous agent needs to respect right-of-way regulations. These rules are special in that they have been designed for the general public and are denoted in natural language, using abstract concepts of space. Making these regulations accessible to an artificial agent requires translating them into a formal language that can be understood by the agent and which seamlessly integrates with the agent's navigation process. In order to facilitate correctness and verifiability of the translation, an abstract language is particularly suited if it is able to reflect the concepts originally used in the right-of-way regulations. We use *qualitative representations* to abstract real-wold observations to abstract knowledge about space and time on a conceptual level. Qualitative spatial representations (see Cohn and Renz (2007) for an overview) aim to provide a formal model for human-level commonsense understanding of space and time. Moreover, they enable abstract reasoning processes. Technically, qualitative representations summarize similar real-world states by a discrete, finite set of qualitative categories that give rise to symbolic reasoning.

This paper demonstrates the utility of qualitative reasoning in autonomous sea navigation. In previous work we have studied how purely symbolic reasoning can help to consistently integrate pair-wise rule constraints when multiple agents meet (Dylla et al., 2007). We now focus on the problem of actually controlling a vessel in a rule-compliant manner. The contribution of our work is to show how the official right-of-way rules for vessel navigation (COLREGS: vessels in sight of each other) according to the International Maritime Organization (IMO) can be modeled using qualitative spatial representation. Furthermore, we show how the representation supports rule-compliant action planning for autonomous vessels. This paper is organized as follows. We start by putting our approach in the context of high-level agent control. Section 3.3 introduces qualitative representation and reasoning techniques. Using these techniques, Section 3.4 details our formalization of navigation rules. Section 3.5 explains how we incorporate the qualitative rules into action planning. We give an experimental account of our approach in Section 3.6 and conclude the paper by summarizing the results and discussing further research directions.

3.2 Rule-Compliant Navigation

Rule-compliant navigation starts by formalizing navigation rules in a formal language that can be understood by autonomous agents. To this end, symbolic navigation rules are suitable (Pommerening, Wölfl, and Westphal, 2009), in particular qualitative representations can be incorporated in logic-based agent control (Bhatt and Loke, 2008). Such techniques tackle planning only on the level of abstract actions though. Navigation rules, in particular codes of practice for sailing, can also be captured in Fuzzy representations (Stelzer, Pröll, and John, 2007), but they lack the formal semantics that allow abstract processes to reason about the con-

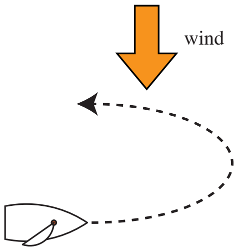


Figure 3.1: Tacking a sailboat

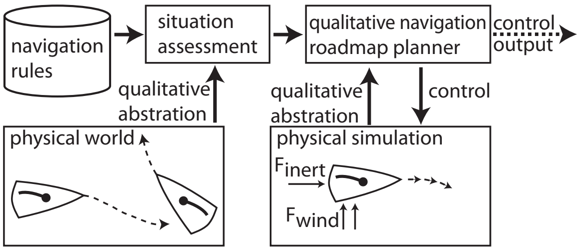


Figure 3.2: Architecture overview

sistency of actions possible with qualitative representations (Dylla et al., 2007). Navigating by qualitative rules requires bridging from abstract spatial relations to concrete control parameters needed by the actuators of the robotic system. Thus, symbolic reasoning needs to be linked to control parameter selection. The example of tacking, a complex turn maneuver in sailing (see Figure 3.1), illustrates the difficulty of this integration. Tacking requires several preconditions to be met in order to perform the maneuver, for example, enough free maneuver space needs to be available. The amount of space required depends on the specific physical context like wind, initial vessel speed, inertia of the vessel, etc. If the initial speed of the vessel is too slow, tacking fails. It appears to be difficult to come up with an abstract definition of tacking that is precise enough to represent exactly those situations in which the maneuver is possible. For example, overestimating the space requirements may inhibit planning to identify situations in which the action can be performed, underestimating it may lead to accidents. Thus, the applicability of symbolic planning can be questioned.

By contrast, we use the symbolic level to formalize navigation rules as sequences of key configurations to pass through, avoiding action definitions. Key configurations are used as intermediate goals in a probabilistic roadmap planner. Probabilistic planning has previously been shown to be applicable in traffic planning. In contrast to the approach by (Smierzchalski and Michalewicz, 2000), we explicitly model collision regulations in a formal language. The combination of qualitative representation with probabilistic roadmap planning has also been suggested in (Westphal et al., 2011), but our approach does not need to generate a plan on an abstract level before invoking the action planner. This way we avoid the aforementioned problem of exactly describing action preconditions. Figure 3.2 presents an overview of our approach. Based on the qualitative assessment of an observation we select the navigation rules applicable to the situation. Then we employ the planner to determine control actions for rudder and sheet rope length that allow the vessel to navigate under the side constraints of rule consistency. Our approach can be regarded as a hybrid navigation system involving mathematical models in the terminology of Statheros et al. (Statheros, Howells, and McDonald-Maier, 2008), which argue for the use of hybrid models in ship collision avoidance.

3.3 Qualitative Spatial Knowledge Representation

Qualitative Spatial Reasoning (QSR)[1] is the subfield of knowledge representation involved with spatial representations that abstract from the details of the physical world. Its reasoning techniques allow predictions about spatial relations, even when precise quantitative information is not available (Cohn and Renz, 2007). Based on qualitative representation and corresponding reasoning methods computers can be enabled to monitor, diagnose, predict, plan, or explain the behavior of physical systems (Kuipers, 1994). In general, two categories of reasoning based on qualitative spatial representations can be distinguished: *constraint-based reasoning* to reason about static configurations and *neighborhood-based reasoning* to reason about how qualitative representations can change over time. In the following we give an intuitive introduction to the basic concepts of QSR—the interested reader is pointed to the literature (Cohn and Renz, 2007; Renz and Nebel, 2007).

A central notion in QSR is that of a *qualitative calculus* which comprises a finite set of *atomic relations* to describe the relationships between entities as well as operations on these relations. Technically, these relations are binary (sometimes ternary) relations between domain level objects. In this paper we only consider binary relations. Relations feature a set-theoretic semantics, i.e., a binary relation r on the domain D is a subset $r \subseteq D \times D$, i.e., a set of ordered pairs (x, y) with $x, y \in D$. The set R of all relations of a qualitative calculus is assumed to be jointly exhaustive and pairwise disjoint. Hence, we have a boolean set algebra with the usual set operations. Elements of this set algebra, i.e., arbitrary disjunctions of atomic relations, are also called *qualitative relations* for short.

Qualitative calculi define two relation operations which allow new facts to be derived from given ones: *conversion* and *composition*. Mathematically, qualitative calculi relate to relation algebras in the sense of Tarski. *Conversion* can be interpreted as shifting perspective from one entity to another. For example, conversion allows us to infer from the fact that Lübeck is *NortEast* of Bremen that it also holds that Bremen is *SouthWest* of Lübeck. The *composition* operation allows knowledge about a common entity to be combined. Only using that Hamburg is *NortEast* of Bremen and Lübeck is *NortEast* of Hamburg, we can still derive with composition that Lübeck is *NorthEast* of Bremen. Since QSR is involved with finite sets of atomic relations only, the composition operation is usually provided in form of look-up tables called *composition tables*. These operations are particularly important for constraint-based qualitative reasoning (Renz and Nebel, 2007) and often enable efficient algorithms (Renz, 2007) to tackle the problem of deciding whether a set of constraints involving qualitative relations is consistent or not.

Conceptual neighborhood extends static qualitative representations by interrelating the discrete set of base relations (Freksa, 1991): *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 in between.* We note, conceptual neighborhood on the

[1] As reasoning is not possible without representation we will not distinguish between them generally in the remainder of this paper. That is, we shall refer to qualitative spatial representation or qualitative spatial reasoning only.

qualitative level corresponds to continuity on the physical level. For example, let us consider the relations *behind*, *same*, and *ahead* to relate the positions of two vessels in a match race. For reasons of simplicity we assume that vessels are only able to move forward with changing speed. In the leftmost configuration shown in Figure 3.3 vessel $\vec{A}$ is *behind* $\vec{B}$. Observing the scene a few minutes later shows that now $\vec{A}$ is *ahead* of $\vec{B}$. Assuming continuous motion it is not possible for $\vec{A}$ to overtake $\vec{B}$ without passing $\vec{B}$ at some time, i.e. being at the *same* level. Therefore, *ahead* and *behind* are not conceptually neighbored, whereas *ahead* and *behind* are both conceptual neighbors of *same*.

In order to apply conceptual neighborhoods for reasoning about actions, it is helpful to label neighborhood transitions with actions that initiate the respective transitions. The resulting structure is called the *action-augmented conceptual neighborhood* (Dylla, 2009).

3.3.1 A Qualitative Calculus of Relative Agent Position

Navigation rules are often formulated in an egocentric frame of reference. For example, the notion "oncoming traffic" refers to traffic traveling in the direction opposite to that of the observer. In order to represent such knowledge we require a qualitative calculus about directional information. We base our formalization on the $\mathcal{OPRA}_m$ (Moratz, 2006) calculus which describes relations between object in the domain of oriented points in the plane, i.e., 2D points equipped with a direction. $\mathcal{OPRA}_m$ relations describe the position of an object B as seen from A and, simultaneously, the position of A as seen from B. Relations are thus pairs of directions, written $\vec{A}\,{}_m\angle_i^j\,\vec{B}$ where i, j denote the directions. The calculus is defined for an arbitrary granularity $m \in \mathbb{N}$, controlling how many directions are to be distinguished. In our work we use $m = 4$ (Figure 3.4). For each of the two related oriented points, m lines are used to partition the plane into $2m$ planar and $2m$ linear regions. Direction 0 is aligned with the orientation of the point. If two positions coincide, so-called *'same' relations* occur. In these cases a number s denotes the direction in which B is oriented, as seen from $\vec{A}$ (Figure 3.4(b)), written as $\vec{A}\,{}_m\angle s\,\vec{B}$.

$\mathcal{OPRA}_4$ provides no atomic front or back region, as it is needed to represent 'head-on', for example. Such region can easily we described as disjunctions of finer-grained $\mathcal{OPRA}_8$ relations. To ease notation we define $\mathcal{OPRA}_4^{\mathrm{x}}$ relations as a shorthand notation for such disjunctions. Geometrically, $\mathcal{OPRA}_4^{\mathrm{x}}$ relations can be obtained as follows: we rotate the segment boarders of $\mathcal{OPRA}_4$ by half of the angular resolution, i.e., by $\frac{1}{2} \cdot \frac{360°}{2m} = 22.5°$, and join the linear regions with the planar ones. In Figure 3.6 we depict an example of relation ${}_4\mathrm{x}_0^0$.

3.4 Formalizing Navigation Rules

The basis for our rule formalization is formed by parts of the *International Regulations for Avoiding Collisions at Sea* (COLREGS, simplified rules 11–18). These rules describe how vessels have to behave if they are in sight of each other in order to avoid collisions. The rules in the COLREGS are given in natural language using abstract spatial terms. Additionally,

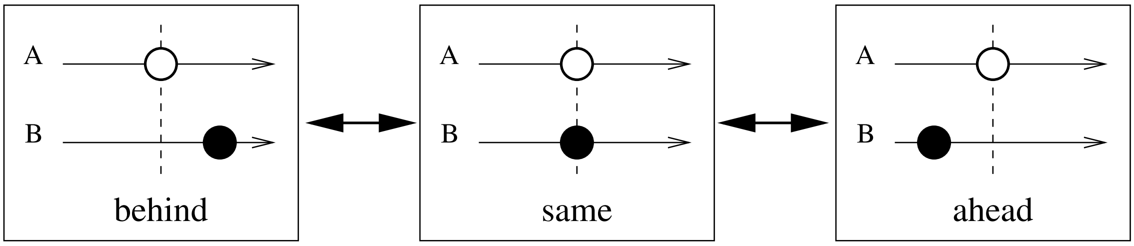
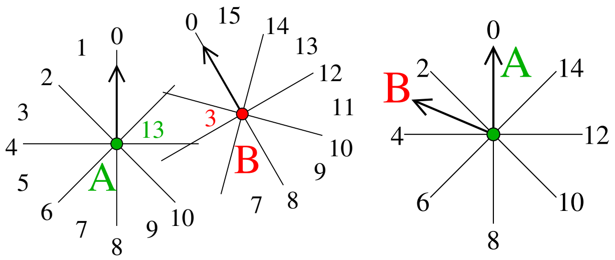


Figure 3.3: Ordering relations arranged their conceptual neighborhood relationship



(a) $\vec{A}\ {}_4\angle^{3}_{13}\ \vec{B}$ (b) $\vec{A}\ {}_4\angle 3\ \vec{B}$

Figure 3.4: Two oriented points related at granularity $m = 4$.

textbooks show pictorial representations in order to give people a more vivid interpretation about conditions and execution of rules. For deriving rule formalizations we follow the approach taken by Dylla et al. (Dylla, 2009; Dylla et al., 2007), which employs $\mathcal{OPRA}_m$ and its conceptual neighborhood structure. In extension, we give here additional qualitative representations and consider sailing vessels as well. Let us consider rule 12 (a) and its pictorial representations (Figure 3.5) in head-on and crossing situation:

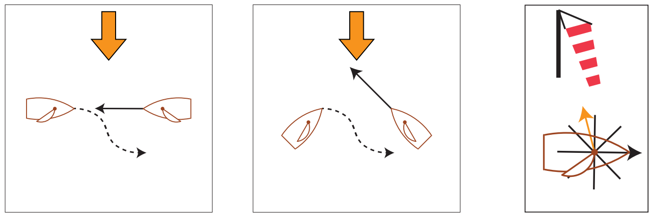
Rule 12:

(a) when two sailing vessels are approaching one another, so as to involve risk of collision, one of them shall keep out of the way of the other as follows:

(i) when each of them has the wind on a different side, the vessel which has the wind on the port side shall keep out of the way of the other;

(ii) ...

In order to define *rule-consistent* or *rule-compliant* behavior we need to operationalize the individual rules. First, we translate and ground the natural language terms in qualitative relations and, second, formalize the rules by means of this representation. Finally, we will define rule-compliant behavior based on these formalizations. By using $\mathcal{OPRA}_m$ relations we abstract from the physically extended objects to oriented points. Wether the extended objects collide or not will be handled in the planning phase.



(a) head-on: $\vec{A}\ {}_4\times^0_0\ \vec{B}$ (b) crossing: $\vec{A}\ {}_4\times^1_7\ \vec{B}$ (c) port-side wind: $\vec{A}\ {}_4\angle 5\ \vec{A}_{\text{wind}}$

Figure 3.5: Two sailing vessels in a head-on and a crossing situation and the advised collision avoidance behavior.

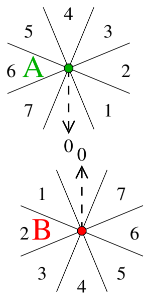


Figure 3.6: An example relation for $\mathcal{OPRA}^{\times}_{4}$: $\vec{A}\ {}_4\times^0_0\ \vec{B}$

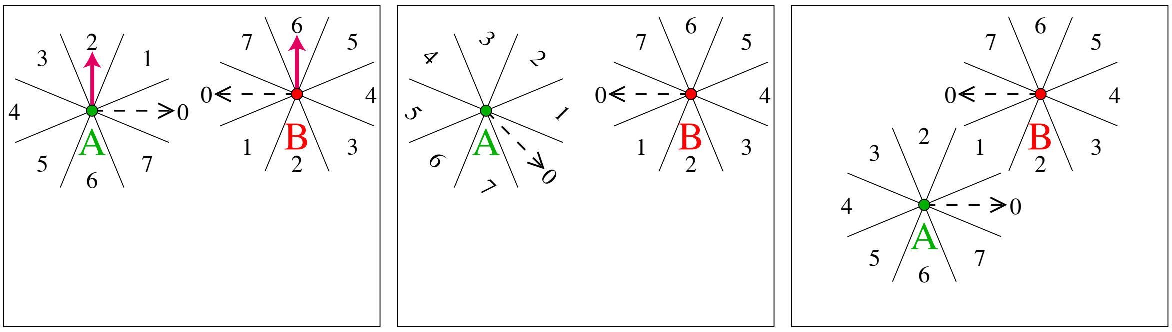
The start configurations for the application of rule 12 (a) is that 'two sailing vessels are approaching one another, so as to involve risk of collision'. For example, this configuration is given if two vessels are meeting on reciprocal or nearly reciprocal courses, which is described by relation ${}_4\times^0_0$.[2]

Furthermore, we need to know from which side of the vessel the wind comes from. We represent the orientation of the wind by an $\mathcal{OPRA}_4$ 'same' relation which is generated from the vessel's heading and the orientation of the wind as seen from the center point of the vessel. In contrast to the representation of the relative orientation of the vessels we need the exact heading. Therefore, we cannot apply $\mathcal{OPRA}^{\times}_{4}$ and must apply the original $\mathcal{OPRA}_4$. Thus, if $\vec{A}\ {}_4\angle i\ \ \vec{A}_{wind}$ with $i \in \{1, .., 7\}$ than the wind is coming from port and with $i \in \{9, .., 15\}$ its coming from starboard. In summary, the conditions of rule 12 (a) can be represented as

$$\vec{A}\ {}_4\times^0_0\ \vec{B} \wedge \vec{A}\ {}_4\angle i\ \ \vec{A}_{wind} \wedge \vec{B}\ {}_4\angle j\ \ \vec{B}_{wind} \tag{3.1}$$

with $i \in \{1, ..., 7\}$ and $j \in \{9, ..., 15\}$. The evasion behavior needs to be modeled next. Vessel $\vec{A}$ is the give-way vessel and $\vec{B}$ the stand-on vessel. Regarding the advised behavior as shown in Figure 3.5(a), $\vec{B}$ must keep its course and $\vec{A}$ must turn starboard in order to avoid a collision. Considering action-augmented conceptual neighborhood this results in a changeover to relation ${}_4\times^0_1$. After the turn it is a reasonable strategy for $\vec{A}$ to move just about straight on, which leads to relation ${}_4\times^1_1$. Going on like this brings us to relation ${}_4\times^2_2$ and ${}_4\times^3_3$ subsequently. At this point the rule could be considered to be successfully performed, but to acknowledge that $\vec{A}$ should return to its original course we also add relation ${}_4\times^4_4$. In summary, we represent rule 12 (a) as the sequence (denoted $\rightarrow$) of formulae (see also Figure 3.7):

[2] We are aware that this representation also includes situations where agents are far apart and no risk of collision is given. For reasons of simplicity we do not to exclude these cases by a refined representation here.



(a)

$$\begin{array}{llll} \text{head-on:} & \vec{A} & {}_4\mathrm{x}_4^0 0 & \vec{B} \\ \wedge & \vec{A} & {}_4\angle 4 & \vec{A}_{\text{wind}} \\ \wedge & \vec{B} & {}_4\angle 12 & \vec{B}_{\text{wind}} \end{array}$$

(b) $\vec{A}\ {}_4\mathrm{x}_1^0\ \vec{B}$

(c) $\vec{A}\ {}_4\mathrm{x}_1^1\ \vec{B}$

Figure 3.7: Iconographic representation of the first steps in formalizing Rule 12 (a)

$$\begin{aligned} & \vec{A}\ {}_4\mathrm{x}_0^0\ \vec{B} \wedge \vec{A}\ {}_4\angle i\ \vec{A}_{\text{wind}} \wedge \vec{B}\ {}_4\angle j\ \vec{B}_{\text{wind}} \\ & \qquad\qquad \text{with } i \in \{1, ..., 7\} \text{ and } j \in \{9, ..., 15\} \\ \rightarrow\ & \vec{A}\ {}_4\mathrm{x}_1^0\ \vec{B} \rightarrow \vec{A}\ {}_4\mathrm{x}_1^1\ \vec{B} \rightarrow \vec{A}\ {}_4\mathrm{x}_2^2\ \vec{B} \rightarrow \vec{A}\ {}_4\mathrm{x}_3^3\ \vec{B} \rightarrow \vec{A}\ {}_4\mathrm{x}_4^4\ \vec{B} \end{aligned} \tag{3.2}$$

Finally, based on such rule descriptions we define *rule-compliant* behavior. We assume rule $R^x = r_0^x \rightarrow ... \rightarrow r_{n_x}^x$ as given with x being the rule number.

Definition 1. *Admissible configuration*: In context of a rule $R^x = r_0^x \rightarrow \ldots \rightarrow r_{n_x}^x$ only configurations r_i^x included in the rule are admissible.

Definition 2. Agent behavior is *rule-compliant* or *admissible* wrt. R^x if:

1. the vessels' initial configuration is r_0^x
2. during rule execution the vessels are only in admissible configurations
3. only changes from r_i^x to $r_{i\pm1}^x$ occur and r_{i-2}^x does not occur after r_i^x
4. the vessels are in configuration $r_{n_x}^x$ at the end
5. no collision occurs

Other formalizations, for example, following shipping routes defined by buoys and light signals can also be formalized using the same approach. Currently, we do not regard disjunctive rules as, for example in overtake situations where one is allowed to pass port or starboard. The extension to include such formalizations is straightforward though.

3.5 Navigation by Qualitative Rules

In our approach qualitative representations serve exclusively for categorizing a configuration and for checking rule-consistency. A randomized planner generates hypotheses of actions to perform, only requiring the forward kinematics of the agent (given by a sailing simulator in our case). Actions generated by the planner are then assessed qualitatively—actions that violate rules are discarded and the most promising actions are further considered. Although a planner may be capable of determining complex action sequences, it is advisable only to exectute the first actions and to re-plan as soon as possible. Continuous re-planning allows the system to respond to unforeseen situations, for example changing winds or unexpected behavior of others.

3.5.1 Probabilistic Roadmap Planner

In our approach action selection is performed by a *probabilistic* or *randomized roadmap planner (PRM)* (Kavraki et al., 1996). This type of planners is particularly helpful for motion planning when no inverse kinematic model is given, only a forward kinematic or simulation is needed. Another feature of interest is its ability to incorporate further constraints, such as scoring solutions by the intermediate locations visited or efficiently re-computing paths in dynamic environments (Belghith, Kabanza, Hartman, and Nikambou, 2006; Belghith, Kabanza, and Hartman, 2010). In a nutshell, a PRM builds a graph of the search space similar to classical AI search techniques. Nodes in the graph represent states of the search space, they are linked by edges that are labeled by the action that allow an agent to get from one state to the next. The objective is to determine a path from the start node to a goal node. During planning, a node is randomly selected and expanded by performing a fixed number of random expansions, i.e., random actions are performed. A heuristic scoring function h is employed to rate the expansion probability of a node and to facilitate goal-directedness.

We represent the dynamic state of the vessel as nodes which are then linked by the rudder and sail actions performed. Also, we record the complete trajectory from the start position to the respective node as well as the total plan duration measured in simulation time. Every node representing a vessel position that is closer than 10m to the desired goal position is considered to be a goal state. We use bold to denote vectors and $\langle\cdot,\cdot\rangle$ for the scalar product. Our heuristic scoring function contoling random node selection is based on the position $\mathbf{p}$ of a vessel, its velocity vector $\mathbf{v}$, and the goal position $\mathbf{g}$.

$$h(n) := \begin{cases} 0 & \text{, the trajectory of } n \text{ is not rule-compliant} \\ h' & \text{, otherwise} \end{cases} \tag{3.3}$$

$$h' = \langle \mathbf{p}-\mathbf{g}, \mathbf{p}-\mathbf{g}\rangle \cdot (1+\max\{0, \langle \mathbf{v}, \mathbf{p}-\mathbf{g}\rangle\}))^2 \tag{3.4}$$

Any node corresponding to a trajectory that is not rule-compliant is assigned a score of zero, i.e., it cannot be selected any more for expansion. This ensures that the planner always determines a rule-compliant plan. For rule-compliant nodes, the scoring combines distance to the goal with a speed component (second term in Equation 3.4). This term serves to differentiate positions that are similarly close to the goal, but in which the vessel is either sailing towards the goal or away from it.

Random node selection first determines the total score $s = \sum_{i=1}^{o} h(N_i)$ of all open nodes $N_1, N_2, \ldots, N_o$ and then samples a uniformly distributed random number r in the interval $[0, s]$, selecting the node N_j with the smallest value of j such that $\sum_{i=1}^{j} h(N_i) > r$. If a node is selected, n random actions are generated and the search graph is expanded.

In order to avoid combinatorial explosion that would occur if continuously expanding nodes, we restrict the size of the set of active nodes which can be further expanded. After a series of node expansions, the set of active nodes is sampled to cut it down to its initial size. Doing so, the memory requirement of the planner is kept constant. We perform the sampling simply by first selecting all k nodes to expand as explained above and then performing node expansion. Nodes that have not been selected are discarded immediately. Although limiting the set of active nodes may discard states that lead to the goal, the step is necessary to obtain a method that can generate a plan using limited computational resources. In our evaluation we analyze different choices for the size of the set of active nodes in order to identify a good balance between the ability of the planner to determine a path and memory requirements.

3.5.2 Physical Simulation

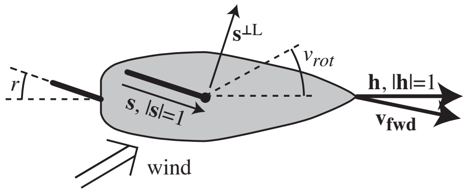
Designing our simulation we aimed to create a mock-up of the sailing experience with cruising yachts (approximately 10m of length, deep single-fin keel, and a single mainsail). As control commands we only consider the position of the rudder and the length of the sheet rope that controls how far the boom is opened. We employ a very efficient but idealized physical simulation to determine the effects of actions and development of the environment. Most essentially, we make the following idealization:

- No waves, no fluid simulation
- Simple wind model, no turbulence, no slip streams
- Control actions are performed in a single simulation step

Although true sailing sport draws its attraction from some of these facets, we believe that they can be neglected in context of navigating in safe operation range. Unfortunately, any realistic simulation involves careful modeling of physical phenomena beyond the scope of our work and it would require considerable computational resources. Since roadmap planners make intense use of the simulation, its efficient implementation is key. We employ the variables shown in Figure 3.8 to describe the dynamic physical state of any vessel.

By $\mathbf{v}^{\perp}$ we refer to the left normal vector of a vector $\mathbf{v}$ and $\mathbf{v}^{\perp \mathrm{L}}$ stands for the lee site normal vector. In order to model the water resistance we decompose friction into friction along its longitudinal and its lateral axis.

$$d(\mathbf{f}, \mathbf{h}, \alpha, \beta) := \alpha \langle \mathbf{f}, \mathbf{h} \rangle \cdot \mathbf{h} + \beta \langle \mathbf{f}, \mathbf{h}^{\perp} \rangle \cdot \mathbf{h}^{\perp} \tag{3.5}$$



symbol	value	description
c_{sail}	50	size of sail in m^2
c_{m}	10000	mass in kg
c_{hull}	0.02	coefficient longitudinal vs. lateral friction of hull
c_{up}	0.5	buoyant force
c_{drift}	0.001	wind drift
c_{fr}	0.005	friction
c_{rot}	100	rotational inertia
c_{r}	5.0	rudder force coefficient

Figure 3.8: Legend of variables and parameters used in the physical simulation

Based on the current local wind vector $\mathbf{w}$ obtained as the difference between global wind and current speed we determine the resulting acceleration, which is then integrated in a constant time-step simulation to update the dynamic vessel parameters.

$$
\begin{aligned}
a_{\vec{f}wd} \;&:=\; \frac{1}{c_{\mathrm{m}}}\Big(\underbrace{d(|\mathbf{v_{fwd}}|\cdot(-\mathbf{v_{fwd}}),\mathbf{h}^{\perp},c_{\mathrm{fr}},c_{\mathrm{fr}}\cdot c_{\mathrm{hull}})}_{\text{friction}} \\
&\qquad + d\big(\underbrace{c_{\mathrm{up}}c_{\mathrm{sail}}|\mathbf{w}|^2\mathbf{s}^{\perp L}}_{\text{buoyant}} + \underbrace{c_{\mathrm{drift}}c_{\mathrm{sail}}|\mathbf{w}|^2}_{\text{drift}},\mathbf{h},c_{\mathrm{fr}}\cdot c_{\mathrm{hull}},c_{\mathrm{fr}}\big)\Big) \\
a_{\mathrm{rot}} \;&:=\; \frac{-1}{c_{\mathrm{m}}}\underbrace{c_{\mathrm{rot}}|v_{\mathrm{rot}}|v_{\mathrm{rot}}}_{\text{friction}} + \underbrace{c_{\mathrm{r}}\langle\mathbf{h},\mathbf{v}_{\mathrm{fwd}}\rangle^2\sin(-r)}_{\text{rudder}} - \underbrace{c_{\mathrm{drift}}c_{\mathrm{sail}}|\mathbf{w}|\langle\mathbf{w},\mathbf{s}^{\perp}\rangle}_{\text{wind rotation}}
\end{aligned}
$$

We have determined all parameters empirically by selecting values that yield reasonable sailing behavior, the values are listed in Figure 3.8. While the model can easily be extended to include effects like currents, changing winds, etc., it is sufficiently complex to give the appearance of sailing as well as to require sophisticated planning techniques. We note that the simulation needs to be realized as a efficient constant time step simulation since the roadmap planner needs to compare actions om a variety of different situations.

3.6 Experimental Evaluation

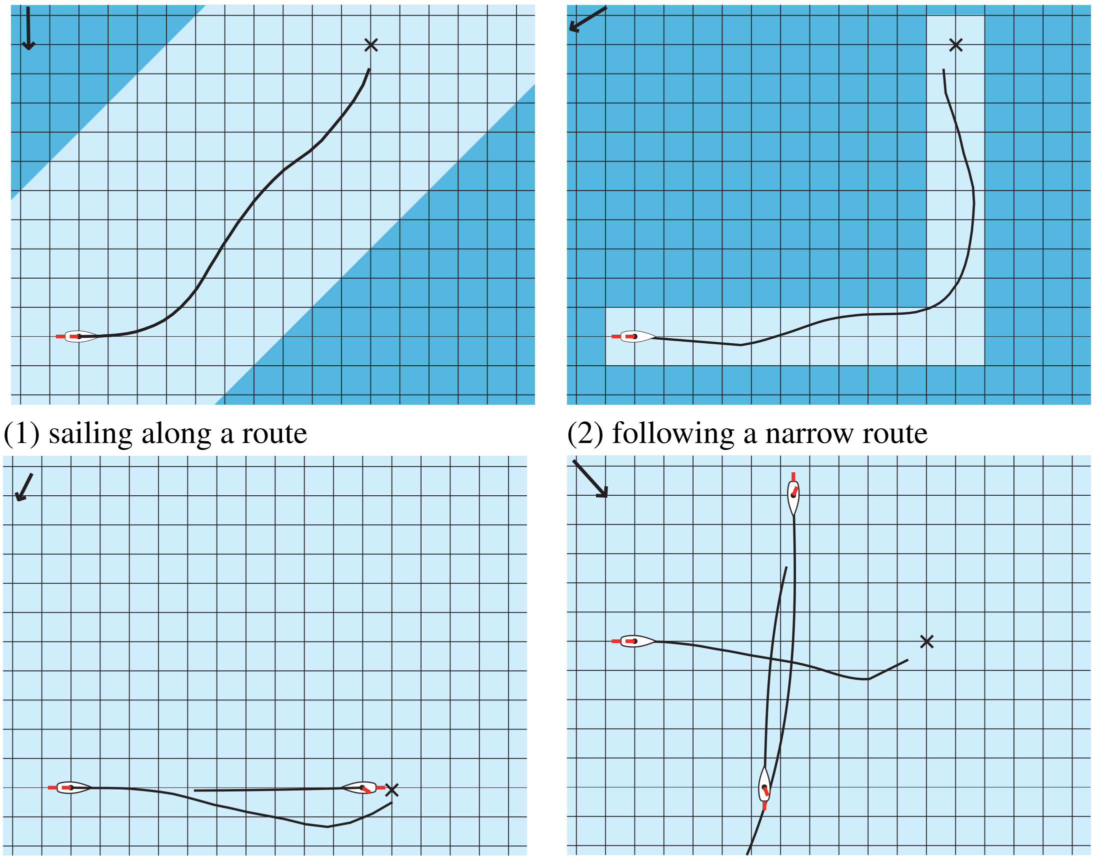
From the great variety of possible planning tasks, we selected some interesting scenarios. For each scenario we randomly instantiate planning tasks by varying the global wind and the speed of additional vessels involved in the scenarios. We keep the courses of additional vessels fixed to gain independency of multi-agent aspects. We have selected the following types of scenarios (see Figure 3.9 for illustration):

1. Sailing along a straight route
   Sail from $(0,0)$ to $(100,100)$ on a 100m wide route with wind from an arbitrary direction.
2. Sailing along a narrow route
   Sail from $(0,0)$ to $(110,100)$ on a bend route restricted in width to 20m with wind from an arbitrary direction.
3. Giving way to an oncoming vessel (rule depicted in Figure 3.5(a))
   Sail from $(0,0)$ to $(100,0)$ with wind from a random northern direction (compass angle between $270°$ and $90°$), avoiding an oncoming vessel.
4. Crossing a frequented channel (rule depicted in Figure 3.5(b))
   Sail from $(0,50)$ to $(100,50)$ with wind from the west (compass angle $225°$ and $315°$), passing behind the stern of two crossing vessels. The challenge is to start sailing slowly (which is not favored by the heuristic) in order to pass behind the other vessels before increasing sailing speed.

The vessel always starts with zero speed and the wind speed is $3ms^{-1}$. For a fixed amount of n active nodes we determine whether the planner is able to determine a solution and we record the execution time of the plan as well as the length of the trajectory computed. We regard planning as successful if the planner can determine corresponding to a trajectory to a position closer than 10m to the goal (the planner stops immediately when a solution is found). By design, any trajectory returned is rule-compliant. For each type of scenario we randomly generate 100 instances and measures the aforementioned criteria for different choices of n. The planner has additionally been equipped with a time-out, trajectories are discarded if they exceed 500 seconds of simulation time or 150 control actions. During node expansion, 100 random actions are generated per node. The results we obtain are presented in Table 3.1. For every scenario we give the average over all successful planning attempts and, in parentheses, the respective standard deviation.

3.6.1 Discussion

Considering the success rate of planning as shown in Table 3.1, it can be seen that the performance varies between up to 90% for the easy scenarios to as low as 22% for the more difficult tasks, in which precise control actions are necessary. The probabilistic nature of the PRM cannot guarantee to identify a plan in all situations. However, as we could not find any systematic failure, the planner can be restarted, eventually finding a solution. Varying the amount of active nodes influences the success rate (as well as influencing the computational requirements). This can be used to balance the need of restarts with the computational demands. For example, considering the simple scenario, it can be seen that that a set of only 20 active



(1) sailing along a route

(2) following a narrow route

(3) giving way to an oncoming vessel

(4) passing a channel

Figure 3.9: Test scenarios for the evaluation including an example trajectory. Vessels are shown at their start positions, goals are marked with a cross. The grid distance is 10m.

nodes gives a success rate of 86% which, by using ten times the amount of active nodes, can only be increased to 93%. Thus, using a small set of active nodes and restarting if necessary provides efficient means for path-planning. The measured standard deviation shows that the overall navigation quality (shortest/quickest route) leaves room for improvement, in particular the high standard deviation in the easy scenario results from some outliers in terms of long detours. However, the performance of our simple metric planning systems already indicates that a randomized-qualitative approach enables rule-compliant navigation. In particular, results for scenarios 3 and 4 show that the influence of pruning away not allowed configurations does not interfere with action planning.

As the planner does not include any pre-defined behavior, e.g., how to sail against wind or how to start sailing a vessel from a complete standstill if facing the wind, such basic sailing maneuvers have to be continuously (re-)discovered by the planner. This lack of expert knowledge can also explain the poor performance in sailing the narrow passage in scenario 2.

scenario	active nodes	success rate	path length [m]	plan duration [s]
1	20	86%	180.6 (±40.5)	64.6 (±31.1)
1	100	90%	152.2 (±16.0)	57.5 (±55.0)
1	200	93%	138.8 (±39.8)	78.0 (±72.0)
2	20	22%	201.5 (±21.9)	70.0 (±16.5)
2	100	48%	223.3 (±23.0)	68.0 (±14.8)
2	200	63%	214.8 (±26.4)	62.1 (±14.8)
3	20	81%	106.3 (±19.0)	38.5 (±8.0)
3	100	84%	105.5 (±9.2)	34.5 (±6.5)
3	200	84%	102.9 (±5.6)	31.6 (±5.5)
4	20	38%	94.2 (±12.4)	109.8 (±74.1)
4	100	66%	95.8 (± 13.9)	58.2 (±33.0)
4	200	79%	97.13 (± 16.0)	60.2 (±39.8)

Table 3.1: Analysis of plans obtained for the test scenarios

While excluding such basic knowledge from the planning step may sound like an artificially created difficulty at first, it significantly hints at the capability of the presented approach when moving closer to a realistic sailing simulation and, ultimately, when applying the method to a real autonomous sailing vessel. With respect to a reasonable code of practice for basic sailing tasks, we believe that the abstract qualitative representation provides solid means to formalize such general rules. Our current approach can be extended to accommodate for such general rules. The key difference between right-of-way rules and rules of good practice is that the latter kind only provides default knowledge that may be violated.

3.7 Conclusion

This paper demonstrates how navigation rules can be formalized with qualitative constraint calculi and how qualitative reasoning can contribute to solving the navigation problem in autonomous robotic sailing. Formalizing spatial knowledge occurring in sea navigation essentially involves representation of directional spatial information, i.e., to describe the positions as seen from specific points of view (egocentric frame of reference). Our approach employs relations from the qualitative constraint calculus $\mathcal{OPRA}_m$, according qualitative reasoning methods allow us to combine information from different frames of reference into a coherent whole. Qualitative directional relations can capture static traffic regulations as imposed by buoys as well as it can capture spatio-temporal movement patterns of, for example, official right-of-way regulations or strategic maneuvers. While qualitative reasoning can be used to determine coarse, qualitative actions that are admissible with respect to the navigation rules, additional means are required to check whether such actions are possible for a specific agent in a specific physical context. In particular the kinematics of sailing vessels largely depend on the current wind, speed, etc. We use probabilistic roadmap planners to determine applicability of actions. The randomized approach to planning is particularly attractive for its ability to cover large search spaces. Furthermore, the approach can easily be integrated with a qualitative rule formalization. In this paper we demonstrate how the integration can be achieved. We also give first results of an integrated randomized-qualitative approach, demonstrating that reasonable control commands can be determined to control an autonomous robotic vessel in a rule-compliant manner.

In future work, we aim to reproduce our results in a sophisticated simulation context, stepping closer to control a real autonomous vessel. We plan to extend the qualitative rule formalization by high-level description of navigation recommendations to improve sailing performance (see Stelzer, Pröll, and John (2007)). While we currently use a simple model to anticipate the actions of other agents, interesting scenarios like regatta racing call for a much more involved handling of multi-agent aspects. We are confident that the qualitative rule formalization provides excellent grounds to tackle such competitive multi-agent navigation problems.

References

Belghith, Khaled, Froduald Kabanza, and Leo Hartman (2010). “Using a randomized path planner to generate 3D task demonstrations of robot operations”. In: *Autonomous and Intelligent Systems (AIS), 2010 International Conference on*, pp. 1–6.

Belghith, Khaled, Froduald Kabanza, Leo Hartman, and Roger Nikambou (May 2006). “Anytime dynamic path-planning with flexible probabilistic roadmaps”. In: *Proceedings IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2372–2377.

Bhatt, Mehul and Seng Loke (2008). “Modelling Dynamic Spatial Systems in the Situation Calculus”. In: *Spatial Cognition and Computation* 8.1, pp. 86–130. ISSN: 1387-5868.

Cohn, Anthony G. and Jochen Renz (2007). “Qualitative Spatial Representation and Reasoning”. In: *Handbook of Knowledge Representation*. Ed. by Frank van Harmelen, Vladimir Lifschitz, and Bruce Porter. Elsevier, pp. 551–596.

Dylla, Frank (2009). “Qualitative Spatial Reasoning for Navigating Agents - Behavior Formalization with Qualitative Representations”. In: *Behaviour Monitoring and Interpretation - Ambient Assisted Living*. Ed. by Björn Gottfried and Hamid K. Aghajan. Vol. 3. Ambient Intelligence and Smart Environments. IOS Press, pp. 98–128. ISBN: 978-1-60750-459-7.

Dylla, Frank, Lutz Frommberger, Jan Oliver Wallgrün, Diedrich Wolter, Bernhard Nebel, and Stefan Wölfl (2007). “SailAway: Formalizing Navigation Rules”. In: *Proocedings of the AISB07 workshop on Spatial Reasoning and Communication*, pp. 470–474.

Freksa, Christian (1991). “Conceptual neighborhood and its role in temporal and spatial reasoning”. In: *Proceedings of the IMACS Workshop on Decision Support Systems and Qualitative Reasoning*. Ed. by Madan G. Singh and Luise Travé-Massuyès. Amsterdam: Elsevier, pp. 181–187.

Kavraki, Lydia E., Petr Švestka, Jean-Claude Latombe, and Mark H. Overmars (Aug. 1996). “Probabilistic roadmaps for path planning in high-dimensional configuration spaces”. In: *Robotics and Automation, IEEE Transactions on* 12.4, pp. 566–580.

Kuipers, Benjamin (1994). *Qualitative Reasoning: Modeling and simulation with incomplete knowledge*. Cambridge, Massachusetts, USA: MIT Press.

Moratz, Reinhard (2006). “Representing Relative Direction as a Binary Relation of Oriented Points”. In: *17th European Conference on AI*. Ed. by Gerhard Brewka, Silvia Coradeschi, Anna Perini, and Paolo Traverso. Italy: IOS Press, pp. 407–411. ISBN: 1-58603-642-4.

Pommerening, Florian, Stefan Wölfl, and Matthias Westphal (2009). “Right-of-Way Rules as Use Case for Integrating GOLOG and Qualitative Reasoning”. In: *KI*. Ed. by Bärbel Mertsching, Marcus Hund, and Muhammad Zaheer Aziz. Vol. 5803. Lecture Notes in Computer Science. Springer, pp. 468–475. ISBN: 978-3-642-04616-2.

Renz, Jochen (2007). “Qualitative Spatial and Temporal Reasoning: Efficient Algorithms for Everyone”. In: *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI-07)*, pp. 526–531.

Renz, Jochen and Bernhard Nebel (2007). “Qualitative spatial reasoning using constraint calculi”. In: *Handbook of Spatial Logics*. Ed. by Marco Aiello, Ian E. Pratt-Hartmann, and Johan van Benthem. Springer, pp. 161–215. ISBN: 978-1-4020-5586-7.

Smierzchalski, Roman and Zbigniew Michalewicz (2000). “Modeling of ship trajectory in collision situations by an evolutionary algorithm”. In: *IEEE Transactions on Evolutionary Computation* 4, pp. 227–241.

Statheros, Thomas, Gareth Howells, and Klaus McDonald-Maier (2008). “Autonomous Ship Collision Avoidance Navigation Concepts, Technologies and Techniques”. In: *Journal of Navigation* 61, pp. 129–142.

Stelzer, Roland, Tobias Pröll, and Robert Ivor John (2007). “Fuzzy Logic Control System for Autonomous Sailboats”. In: *FUZZ-IEEE’07*, pp. 1–6.

Westphal, Matthias, Christian Dornhege, Stefan Wölfl, Marc Gissler, and Bernhard Nebel (2011). “Guiding the Generation of Manipulation Plans by Qualitative Spatial Reasoning”. In: *Spatial Cognition and Computation* 11.1, pp. 75–102.

4 Towards Safe Navigation by Formalizing Naviagtion Rules

Arne Kreutzmann[1], Diedrich Wolter[1], Frank Dylla[1], and Jae Hee Lee[1]

1 Cognitive Systems Group, University of Bremen, Bremen, Germany

Published in “TransNav, the International Journal on Marine Navigation and Safety of Sea Transportation”, 2013, Volume 7, Number 2.

Contributions:

The study was conducted by Diedrich Wolter and me, resulting from my research on bridging the gap between a simple, verifiable spatial-logic and navigation concepts. Frank Dylla provided the necessary expertise in maritime sailing. Jae Hee Lee contributed his implementation of StarVars. I developed the tools and did most of the modeling. The manuscript was jointly prepared by all authors.

Acknowledgements:

This work is carried out in context of the Transregional Collaborative Research Center Spatial Cognition, project R3-[Q-Shape]. Financial support by the Deutsche Forschungsgemeinschaft (DFG) is gratefully acknowledged.

Abstract

One crucial aspect of safe navigation is to obey all navigation regulations applicable, in particular the collision regulations issued by the International Maritime Organization (IMO Colregs). Therefore, decision support systems for navigation need to respect Colregs and this feature should be verifiably correct. We tackle compliancy of navigation regulations from a perspective of software verification. One common approach is to use formal logic, but it requires to bridge a wide gap between navigation concepts and simple logic. We introduce a novel domain specification language based on a spatio-temporal logic that allows us to overcome this gap. We are able to capture complex navigation concepts in an easily comprehensible representation that can directly be utilized by various bridge systems and that allows for software verification.

4.1 Introduction

Navigation regulations such as the official collision regulations of the International Maritime Organization (IMO Colregs) are an essential instrument for safety in navigation. Some situations may require further rules and general recommendations may be implemented to foster sensible navigation behavior, e.g., with respect to fuel effciency. All these regulations need to be obeyed-which can be a very demanding task in complex situations. By augmenting bridge systems such as ECDIS and autopilots to understand navigation regulations we can support crews, reducing the risk of regulation violations. To start with, this requires an implementation of navigation regulations that is known to be correct.

We argue for a declarative, logic-based approach to represent navigation regulations. Logics offer precise semantics for reasoning and they build a common basis for software verification. The use of such formal methods during software development is a common requirement for higher standards of safety-critical software. However, logics are usually based on primitive concepts and it requires overly complex statements to represent everyday concepts such as “oncoming traffic”. Trying to formalize a non-trivial set of navigation regulations with a simple logic inevitably leads to incomprehensible formalizations that are error-prone to align with navigation software, rendering effectiveness of the overall approach questionable.

The contribution of this paper is to show how the opposition of primitive concepts in logic on the one hand and abstract concepts in navigation regulations on the other hand can be overcome. To this end, we develop an abstract logic, a so-called qualitative spatio-temporal logic, which can adequately represent navigation concepts. They allow comprehensible representations specifically suited for navigation problems. Qualitative spatial logics as studied in the field of Artificial Intelligence (AI) are acknowledged for their ability to grasp concepts of human cognition. We thus can connect formal logic to concepts of human cognition, obtaining formalizations with precise logic semantics that can be understood and even adapted by navigators, not only by computer science experts. These formalizations are universal in the sense that the very same representation can be used in a variety of tasks: to display regulation violations in ECDIS, to enforce rule-compliant path-planning in autopilots, and above all to support the software development process by verification.

This paper is organized as follows. We give references to related work, then we present our qualitative spatial logic. We outline how navigation formalizations in this logic can be integrated with various navigation tasks using logic-based software tools. Finally, we show how logic reasoning developed for our logic can be employed in verification and to reveal problems with software or with the regulations themselves. The paper concludes by a discussion and outlook section.

4.2 Background

Sophisticated bridge system can be considered as decision support systems (DSS) as they aim to support crew in navigation decisions. Various computer science techniques have been applied to devise such systems. Smierzchalski and Michalewicz (2000) and Szłapczynski (2010) demonstrate how evolutionary algorithms can be applied for collision free navigation even in case of multi-ship encounter. A related approach has been pursued by Mohamed-Seghir (2012) using a combination of branch-and-bound and genetic algorithms. Both approaches aim to determine a cost-optimal path, but they cannot guarantee to respect official regulations, i.e., it can be illegal and even dangerous to follow a path computed. It is thus necessary to integrate a representation of Colregs in order to obtain decision support that complies to official regulations. As reported by Pietrzykowski and Uriasz (2010), various approaches to represent knowledge contained in navigation regulations like Colregs have been applied. Their approach aims at combining different techniques, but does not handle situations in which multiple vessels are mutually subject to regulations at the same time. By contrast, Banas and Breitsprecher (2011) argue for the use of logic rule-based systems as framework for representing navigation regulations. They claim logic to provide the best means to tackle requirements on a DSS for navigation identified, namely reproducible and verifiable results, integration of informal knowledge, easy update or extension of knowledge, regulation prioritization, and comprehensibility of the representation. Indeed, the use of formal methods based on logic is a common means to foster reliability of safety-critical software like a DSS for navigation. We adopt the motivation of Banas and Breitsprecher to employ logic for formalization. Our primary focus is to adequately capture the complex spatio-temporal concepts involved in navigation regulations. We improve on previous work by devising an advanced logic framework that incorporates sophisticated spatial reasoning. This allows us to better meet the aforementioned criteria for bridge systems, in particular with respect to the safety-critical aspect of verifiability of the software and with respect to comprehensibility of the representation.

Developing a formalization of Colregs one has to face several design criteria which are somewhat competing. Any formalization of Colregs has to bridge the gap from the official regulations denoted in natural language to a verified formal framework on which the system is based. A common approach is to develop a domain language which abstracts from the formal framework and offers concepts and techniques close to the application domain. Of course, the mapping from domain language to the formal system must be transparent and verifiable itself to avoid introducing errors in the translation. To this end, our approach utilizes a formal framework that already incorporates many abstract concepts necessary to represent navigation regulations. This allows us to obtain a transparent mapping from domain language to underlying logic. Moreover, bridge systems can benefit from the underlying formal framework, given the framework provides sophisticated reasoning mechanisms that are capable of tackling navigation tasks. As we demonstrate, reasoning methods of qualitative spatio-temporal logics are well-suited to meet this goal.

4.2.1 Qualitative spatio-temporal logics

Qualitative spatial and temporal reasoning is an established field of research dealing with representation and reasoning about spatio-temporal knowledge in an abstracted, i.e. qualitative, manner. Qualitative approaches are symbolic and symbols serve to represent concepts like “left” rather than using numerical values that measure directions. The aim of qualitative approaches is to capture the important distinctions that make a difference for a task at hand while abstracting from irrelevant details.

Qualitative spatial and temporal reasoning provides different methods of reasoning, most notably methods that can decide whether a given symbolic description of a scene is consistent, i.e., whether it can be realized by a physical configuration. For example, the three temporal statements about events A,B, and C, namely "A occurs before B", "B occurs before C", and "C occurs before A", are not jointly realizable as time evolves linearly. Qualitative reasoning provides techniques to reason about various aspect of space and time (Cohn and Renz, 2007) and specialized reasoning tools are available, e.g., SparQ (Wolter and Wallgrün, 2010).

Recently, qualitative approaches have been studied in conjunction with logics, thus coining the term spatio-temporal logic. These logics are formed by "any formal language interpreted over a class of structures featuring geometrical entities or relations" (Aiello, Pratt-Hartmann, and van Benthem, 2007, Chapter 1). The logic itself is not restricted, i.e., it may be a fragment of first order logic or any higher-order logic. In this paper we are concerned with a combination of a modal logic of linear time with a qualitative approach to representing directional knowledge presented in Section 3.3.

4.3 Formalizing Navigation Regulations for use in Bridge Systems

The key question in designing an appropriate formalization is what are the individual components that make up a set of navigation regulations? Since we formalize a safety-critical system we must ensure that these components need a clear linkage to the primitives in the underlying logic.

At the core of a regulation we can identify the navigation behavior. Navigation behaviors come in two flavors. Firstly, we have navigation behaviors as instructed behaviors: the regulation defines which actions are allowed to perform. Secondly, we find navigation behaviors setting the context in which a specific regulation is applicable, for example, with respect to the vessels' relative course. While both flavors share many commonalities, there exist decisive differences. One must ensure that a context description can be evaluated at any point in time to allow instructed behaviors to be performed as soon as a regulation is applicable. If, by contrast, the context would be allowed to refer to the future, one could not tell whether one's current situation matches the context. We say that a context is a discernible navigation behavior, i.e., a pattern of actions and events that can be recognized by an observer. Analogously, instructed behaviors are restricted to only talk about future actions. In other words, regulations are of the form "if you approach the port, reduce speed" rather than "if you crashed into a quay wall, you should have reduced speed in first place". Although context and instructed behavior are distinct, we can apply a common framework of representation to both of them.

As second component of regulations we identify a valuation of liability. As soon as a regulation is applicable, its instructed behavior defines which actions are allowed. As applicability of a regulation is subject to change, we introduce the term valuation of liability to indicate whether a navigation behavior is applicable and how it relates to competing regulations. The Colregs regulations have different liability and their liability might change depending on other regulations currently applicable. For example, the regulations state that (Rule 13,d): "Any subsequent alteration of the bearing between the two [overtaking] vessels shall not make the overtaking vessel a crossing vessel within the meaning of these rules or relieve her of the duty of keeping clear of the overtaken vessel until she is finally past and clear." In this example, certain behaviors (being a crossing vessel) are temporarily forbidden while vessels are in the context of overtaking one another. While inhibiting certain behaviors can easily be formalized, a true modeling of rule precedence and conflict resolution is a challenging aspect in its own right and outside the scope of this paper. For time being, we simply say that a valuation may take either the value applicable or not applicable.

In summary, a set of navigation regulations can be formalized as a mapping from the set of navigation behaviors describing the context to a valuation of liability of navigation behaviors that state which behaviors are allowed to take place. Our terminology is close to that of rules in the classical sense of logic in computer science: an antecedence leading to consequence.

Throughout the remainder of this paper we use Colregs Rule 12,a,i (sailing vessels) as a running example to illustrate our approach. Let us start by looking at the example of how Rule 12 can be formalized in our approach shown in Figure 4.1.

As can be seen, we have chosen a simple syntax using parentheses for grouping. The context and instructed behavior part of a regulation are indicated by respective labels. The formalization only explicitly states one case of having "the wind on a different side" which eases readability, as the other case is symmetrical and achieved by swapping the variables. Observe that the formalization utilizes terms like "is approaching" or "keep course" that are very close to the natural language used in Colregs. At this point it is important to note that these terms are logic concepts which need a clear grounding in spatio-temporal knowledge about the world. Assuming a reasonable interpretation of these terms, the formalization can easily be checked against the official Colregs by any domain expert, e.g., trained helmsman or naval expert. Let us now look into the technical details of how these concepts are grounded in the logic and how logic reasoning can be performed.

4.3.1 A spatio-temporal logic for formalizing navigation behaviors

We give a brief introduction of the modal logic underlying our formalization. Since the key focus of this paper is not discussing the logic itself but to demonstrate its applications as well as the domain dependent language established on top of it, we only introduce the logic informally.

For our approach we developed a so-called multi-modal logic. Like any modal logic, this logic is a generalization of propositional logic which is equipped with the concept of different *states*, also called *worlds*. Truth of a formula is evaluated with respect to a specific state. For example, the logic primitive "`sailsSet`" may be true in one state, but false in another. All

official rule (natural language):
When two sailing vessels **are approaching one another**, so as to involve risk of collision, one of them shall keep out of the way of the other as follows:
(i) when each has the **wind on a different side**, the vessel which has the **wind on the port side** shall **keep out of the way** of the other.

formalization (modeling language):

```
(rule12_i
 :context (AND (is_sailing_vessel ?X)
               (is_sailing_vessel ?Y)
               (is_approaching ?X ?Y)
               (is_approaching ?Y ?X)
               (COULD (collide ?X ?Y))
               (wind_on ?X PORT)
               (wind_on ?Y STARBOARD))
 :behavior (AND (give_way ?X) (keep_course ?Y)))
```

Figure 4.1: From Colregs (top) to regulation formalization (bottom). The formalization describes context and required behavior in a declarative manner; `?X` and `?Y` are variables that stand for vessels.

possible states constitute the so-called universe and individual states are connected by specific relations called modals. Typically, a universe is assumed to be given and to be finite (Aiello, Pratt-Hartmann, and van Benthem, 2007). A universe and a set of modals together with the information about which state of the universe makes which logical primitives true form a *model* of a modal logic.

A prominent example for a modal is time: one state may represent the circumstances at a time point ti and the connected state talks about the next mo-ment in time t_{i+1}. As navigation regulations are grounded in time and space we employ two modals (thus we have a multi-modal logic): one modal captures the course of time and another one captures possible spatial changes. The spatial modal will allow us to talk about possible changes of the states and, e.g., to express the possibility of collision as a logic primitive. Technically speaking, we adopt the relation of conceptual neighborhood defined in qualitative spatial reasoning; two states are conceptually neighbored if one state can be continously changed to another (Dylla, 2009). The model for our logic is thus a set of such states along with their temporal ordering, spatial structure, and valuation of all logic primitives. Essentially, our logic is a spatially enhanced generalization of the well-established Linear Temporal Logic (Pnueli, 1977).

For convenience, we write, e.g., `sailsSet(X)`, to denote the logic primitive holding the truth value that corresponds to whether vessel X has sails set or not. Returning to our previous example (Rule 12 i), it can be written in logic notation as follows:

$$\bigvee_{X,Y \in \texttt{Vessels}} \Big[\big(\texttt{Sailboat}(X) \land \texttt{Sailboat}(Y) \tag{4.1a}$$

$$\land\, \texttt{Approaching}(X,Y) \land \texttt{Approaching}(X,Y) \tag{4.1b}$$

$$\land\, \langle cn \rangle \, \big(\texttt{Collsion}(X,Y) \big) \tag{4.1c}$$

$$\land\, \texttt{WindOn}(X, port) \land \texttt{WindOn}(Y, startboard) \big) \tag{4.1d}$$

$$\longrightarrow \big(\texttt{GiveWay}(X) \land \texttt{KeepCourse}(Y) \big) \Big] \tag{4.1e}$$

Note that our logic is already close to the modeling language; so we meet the demand of easy translation from domain language to logic. In this example $\langle cn \rangle$ in line 4.1c stands for a conceptual change which can lead into a state where X and Y collide. The instructed behavior (line 1e in the formula) is written as implication of the preconditions 4.1a–4.1d. Also note that some spatial relations such as `Approaching` used above are in fact independent formulas themselves as we will explain in the following. As a regulation is applicable to all vessels, the simple logic form “context → instructed behavior” needs to be stated explicitly for all logic primitives representing vessels. This is achieved by combining sub-formulas for any choice of X and Y by the logic conjunction “or”. By building a modeling language atop this logic layer we can ensure that all regulation formalizations adhere to this pattern of logic formulas.

4.3.2 A domain language for navigation regulations

In this section we explain how our domain language is build atop the spatio-temporal logic outlined above. We describe how the key notions of context and instructed behavior are expressed and how spatial and temporal knowledge can be represented.

The set of primitive symbols used by the logic is divided, identifying the subset of discernible primitives. Discernible primitives can directly be observed by others (like `sailsSet(X)`, for example) whereas other primitives may not. We employ this distinction such that it can be checked whether a navigation behavior can be recognized by observation: specifications of navigation behaviors allow for recognition if they only involve discernible primitives. The context comprises a set of navigation behaviours. In order to decide whether a context formalization matches a given situation we require the context to only involve discernible behaviours. Moreover, formalization of contexts is restricted to only talk about now, the past, and things possible in future. This can easily be accomplished by restricting the set of modal operators allowed in the formalization. Thus, we inhibit the use of universal-qualified expressions in this part of the formalization. With respect to instructed behavior there is only one requirement: it must not refer to past actions. This is also achieved by disallowing the respective modal operators in the formula. All in all we obtain that all parts of a regulation are logic formulas, each class with a specifically restricted syntax.

In summary, our system translates all rules into the pattern

$$\bigvee_{X,Y} \left[\text{context}\right] \rightarrow \left[\text{instructed behavior}\right] \tag{4.2}$$

as shown in the previous section. The key feature of our approach is its seamless integration with qualitative spatial logics that allows us to define a rich repertoire of spatial relations.

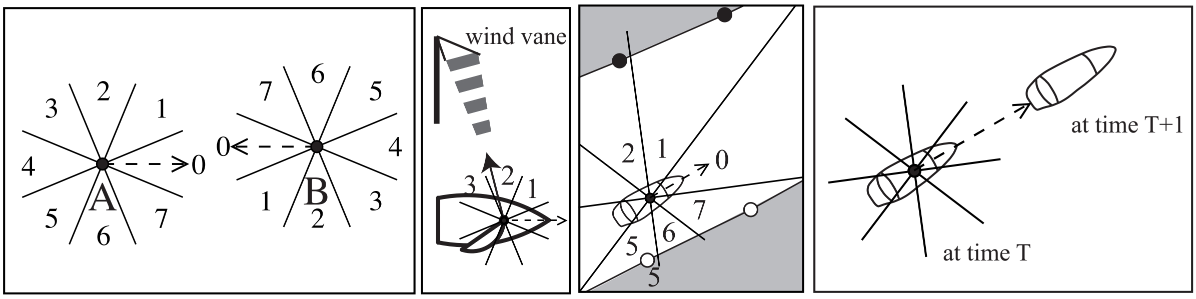
4.3.3 Spatio-temporal primitives

In formalizing Colregs it is essential to formalize the manifold spatio-temporal concepts referenced in the regulations. The key building block of the spatial formalization is a set of qualitative spatial relations that capture directional information as presented in (Wolter, Dylla, and Kreutzmann, 2011). This modeling is a sector-based model presented in (Moratz, 2006) (see Figure 4.2) which allows us to derive most important spatial concepts. Essentially, the model allows directional sectors to be defined that are aligned with respect to position and orientation of an observer. While the number of sectors can be chosen arbitrarily to accommodate for any desired resolution, we restrict the presentation here due to space constraints to showing only the eight-sector variant. In the example shown in Figure 4.2 (A), the position of B is in sector 0 with respect to A and vice versa—A and B are thus oriented to one another. Figure 4.2 (B) shows how the model can be used to describe the wind. The vessel depicted has the wind of port side as the wind comes from sector 2. Analogously, the same model serves to state which is the right side to pass by a buoy, see Figure 4.2 (C). Here, the white area reresents a waterway.

Exhausting the expressivity of a temporal logic we can also exploit these spatial relations to define dynamic navigation behavior. For example, the term “head-on course” can be defined by saying that at one time point two vessels are oriented towards one another (see above), while in the next time point they are still oriented the same way but that both have advanced towards one another. A’s position at time point t$_{n+1}$ is ahead of where A was at time point tn, i.e., A at t$_{n+1}$ is within sector 0 as seen from A at time point t$_n$ —see Figure 4.2 (D) for illustration. It is the modal operators of a temporal logic that grants us the expressivity to relate A’s position between different points in time.

4.3.4 Model checking with spatio-temporal logics

Generally speaking, given a model M and a state w in M and a formula ϕ, the task of model checking in modal logic is to determine whether w along with M satisfies the formula ϕ. Specifically in the context of our spatio-temporal logic, model checking is the task of searching for a sequence of spatio-temporal transitions starting with the input state w of vessels which makes regualtion ϕ true with respect to the model M of the spatio-temporal logic described in Section 3.1. By means of the combination of model checking with methods from qualitative reasoning we are able to reason about whether given input states are critical with respect to safety. The important feature of modal logics is that model checking can be realized efficiently.



(A) (B) (C) (D)

Figure 4.2: Illustration formalizing the spatial concepts underlying Colregs.

In our system we utilize the state of the art model checker PRISM (Kwiatkowska, Norman, and Parker, 2011) which requires us to provide a set of states to check. PRISM either returns that all states satisfy the given formula or it provides us with a counter-example that falsifies the formula. In order to generate all possible states in our spatio-temporal logic, qualitative spatial reasoning is required. For example, consider the statement "`WindOn`(X, port) ∧ `WindOn`(X, starboard)" which is of course not satisfiable. However, from the perspective of a pure modal logic model checker the formula is just the same as "a ∧ b" and thus there is no reason why a and b should not hold at the same time. This is where spatio-temporal reasoning is required to rule out configurations which are spatially or temporally not possible. To this end, we combine our spatio-temporal reasoning system SparQ (Wolter and Wallgrün, 2010) to check all candidates of states for their spatial and temporal consistency. In the following section we show how various practical problems can be supported with the two reasoning tasks on the logic level only: model checking of formulas in our spatio-temporal logic (PRISM) and consistency checking of qualitative spatial configurations (SparQ).

4.4 Reasoning for safe navigation

We now demonstrate how a formalization of navigation regulations serves three major applications in bridge systems: recognition of regulation compliant/violating behavior, regulation compliant planning, and verification of regulation specifications.

4.4.1 Identifying regulation compliancy and regulation violations

Observing the navigation behavior around one's own position, a natural question to ask is: do all other vessels comply with the regulations or is some vessel violating a regulation? In the domain of safe sea navigation a system assisting in the detection of regulation violating behavior of others can be of great importance to a bridge crew. By alerting the bridge crew of such violations appropriate preparations can be made. Assuming that observations of the surrounding navigation behavior are available (e.g., extracted from AIS data or radar), we apply model checking of the Colregs formalization to the observations as follows: From the formalization of Colregs we have the corresponding logic formulas $\phi = \phi_1, \phi_2, ..., \phi_m$ which we combine by logic conjunction "and": $\phi_1 \wedge \phi_2 \wedge ... \wedge \phi_m$. Doing so we obtain a single formula ϕ that represents the complete body of navigation regulations. Observed behavior constitutes the set of states, each snapshot of time defines its own valuation of logic primitives. For example, at time point tn we have that `Approaching`(X, Y), while at the next time point t_{n+1} we already have `TurningAwayFrom`(X, Y) and it holds that `TurningAwayFrom`(X, Y) $\rightarrow \neg$`Approaching`(X, Y). If and only if the set of states obtained from observations provides a model for the logic formula ϕ, the observed behavior is compliant with Colregs. If model checking fails, a counter-example is generated. This counter-example falsifies ϕ and identifies which vessels/actions did not comply with the regulations.

We demonstrated feasibility of this method in a previous study using a different domain, showing that the approach works even on noisy and incomplete sensor data (Kreutzmann et al., 2013). A small scale warehouse was simulated in a laboratory and a robot observed the warehouse, gathering partial observations. The observations were matched against a set of logistic movement patterns—which can be considered as a form of navigation regulations. Based on these partial observations, a matching between real-world observations and abstract model could be established, i.e., compliancy with some logistic rules was verified.

4.4.2 Regulation compliant planning

In complex situations such as crowed waterways like the English Channel, planning routes that are regulation compliant but avoid detours can be a demanding task. An autopilot system could factor in routes of other vessels and inhibit the planning procedure to output routes that are known or likely to violate regulations. This enables the vessel to avoid unnecessary evasive maneuvers and thereby save fuel. Kolendo, Smierzchalski, and Jaworski (2011) have demonstrated that randomized planners are appropriate for computing collision-free routes. In (Wolter, Dylla, and Kreutzmann, 2011) we demonstrated how this approach can be advanced to ensure regulation compliant planning. We extend the state-of-the-art paradigm of randomized roadmap planners to acknowledge navigation regulations as side constraint in planning. This also augments existing work on randomized planning by a verification component.

In essence, the approach to regulation compliant planning is similar to that of recognizing regulation-violating navigation behavior discussed above. While planning takes place, all partial plans considered by the planner are checked for their regulation-compliancy using the very same technique as explained before. Whenever a partial plan is identified to violate some regulation, it is discarded and the planner has to search for an alternative route.

4.4.3 Verification of regulation specifications

When developing safety-critical software all development steps should be verifiable. Moreover, software developers need to be supported to identify problems with their software. To this end, we are currently developing a tool to support the verification of navigation software based on the formalization of navigation regulations. Following the concept of Proof-Carrying Code (Necula, 1997), developers can declare complex assertions in their software. For example, the command to increase engine speed may be guarded by an assertion that no obstacle is in front of the vessel.

Such a tool is particularly valuable when working with Colregs-compliant navigation. Regulations like the Colregs have mostly been developed with respect to defining right of way for just two vessels at a time. Thus, in complex situation a multitude of regulations may be applicable at the same time. Regulations might even contradict themselves or endanger a collision if strictly followed. We can support a knowledge engineer by providing a set of useful verification methods. Given a set of regulations, our tool provides the necessary means to check that regulations are non-overlapping, i.e., there are no situations in which different regulations are contradicting one another. See Figure 4.4 for a screenshot of the tool checking regulations for problems. The tool also allows us to check whether all (critical) situations are covered by some rule, i.e., whether a set of rules is complete. One important feature of the tool is that it does not only identify conflicts of regulations, but it can also generate an exemplary configuration that triggers the conflict. Here we use spatial reasoning to generate a prototypical pictogram that depicts the conflict (see Figure 4.4, top right). The position of the boom indicates direction of the local wind. Conflicting regulations are particularly present if an autonomous navigation systems implements its own regulations for special maneuvers, which might interfere with Colregs. Consider the commonly recommended evading actions in the form of collision avoidance patterns as presented in text books for navigation (e.g. Dreyer, 2012) as shown in Figure 4.3. In the situation depicted in Figure 4.3, the vessel on the right has the conflict of keeping course and give way at the same time. Further the top left vessel should turn port and starboard at the same time. While each collision avoidance pattern on its own is sensible, taking into account the specific requirements such as wind direction, they fail to combine in some situations, e.g., multi-ship encounters. Our tool is able to detect this inconsistency solely by reasoning about the rule definitions as shown in Figure 4.4.

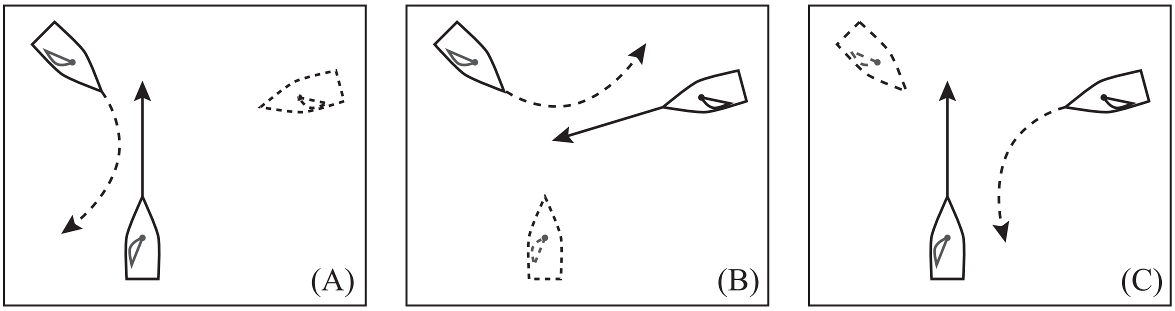
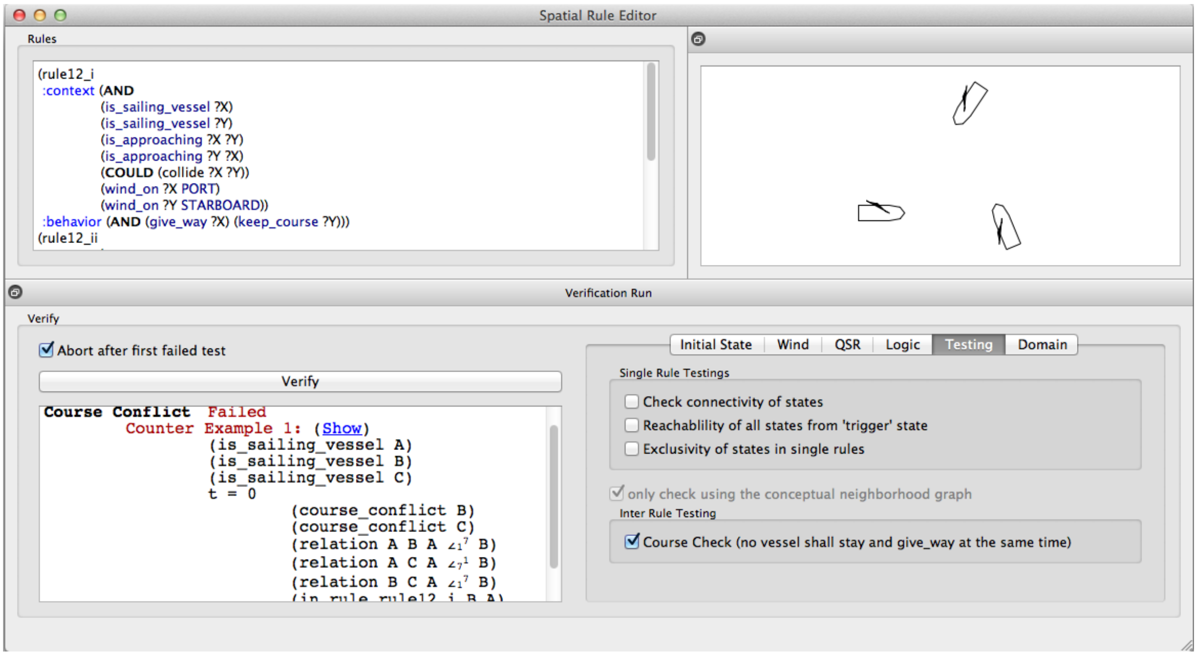


Figure 4.3: Depiction of collsion avoidance patterns (i.e., interpretations of the rules) for pair of vessels as found in text books on navigation. A dashed line indicates that the vessel has to *give way* while a solid line means that the vessel should keep course. In the depicted configuration, these collision avoidance patterns are contradicting one another.



```
(rule12_i
 :context (AND
        (is_sailing_vessel ?X)
        (is_sailing_vessel ?Y)
        (is_approaching ?X ?Y)
        (is_approaching ?Y ?X)
        (COULD (collide ?X ?Y))
        (wind_on ?X PORT)
        (wind_on ?Y STARBOARD))
 :behavior (AND (give_way ?X) (keep_course ?Y)))
(rule12_ii
```

☑ Abort after first failed test

Verify

```
Course Conflict  Failed
        Counter Example 1: (Show)
                (is_sailing_vessel A)
                (is_sailing_vessel B)
                (is_sailing_vessel C)
                t = 0
                        (course_conflict B)
                        (course_conflict C)
                        (relation A B A ∠_1^7 B)
                        (relation A C A ∠_7^1 B)
                        (relation B C A ∠_1^7 B)
                        (in_rule rule12_i B A)
```

Single Rule Testings

□ Check connectivity of states
□ Reachablility of all states from 'trigger' state
□ Exclusivity of states in single rules

☑ only check using the conceptual neighborhood graph

Inter Rule Testing

☑ Course Check (no vessel shall stay and give_way at the same time)

Figure 4.4: Screen shot of the automatically generated visualization of a conflict

4.5 Summary and outlook

Adhering to navigation regulations is an important factor in safe navigation. In order to allow bridge systems to incorporate regulations such as Colregs in a verifiably correct manner, a formalization of navigation regulations is required. In this paper we show that a spatio-temporal logic provides a solid basis for formalizing Colregs. We propose an easy to understand domain language built atop a spatio-temporal logic. Logic reasoning enables automated tools to check formalizations for correctness. This enables software developers for bridge systems to verify their software, meeting high standards of safe software design. Our approach is applicable to a wide range of navigation regulations and can even help to develop new ones (see, for example Kemp, 2007). Thereby, we can provide an answer to a longstanding problem that "there is no possibility of testing new proposals [for Colregs] before they are introduced." (Kemp, 2007). Moreover, a formalization of Colregs in our spatio-temporal logic can also be utilized in navigation systems directly: autopilots can determine routes that are known to comply with Colregs or chart displays can identify violations of Colregs and signal appropriate warnings. We develop a software tool based on the concept of proof-carrying code that enables software developers to verify their software with respect to a formalization of Colregs. In future work, we aim to apply this tool to real navigation software in order to improve safety in navigation technology.

References

Aiello, Marco, Ian E. Pratt-Hartmann, and Johan van Benthem (2007). “What is Spatial Logic?” In: *Handbook of Spatial Logics*. Ed. by Marco Aiello, Ian E. Pratt-Hartmann, and Johan van Benthem. Springer, pp. 1–11. ISBN: 978-1-4020-5586-7.

Banas, Paweł and Marcin Breitsprecher (2011). “Knowledge Base in the Interpretation Process of the Collision Regulations at Sea”. In: *TransNav - International Journal on Marine Navigation and Safety of Sea Transportation* 5.3, pp. 359–364.

Cohn, Anthony G. and Jochen Renz (2007). “Qualitative Spatial Representation and Reasoning”. In: *Handbook of Knowledge Representation*. Ed. by Frank van Harmelen, Vladimir Lifschitz, and Bruce Porter. Elsevier, pp. 551–596.

Dreyer, Rolf (2012). *Sportküstenschifferschein & Sportbootführerschein See*. (in german). Delius Klasing. ISBN: 9783768834902.

Dylla, Frank (2009). “Qualitative Spatial Reasoning for Navigating Agents - Behavior Formalization with Qualitative Representations”. In: *Behaviour Monitoring and Interpretation - Ambient Assisted Living*. Ed. by Björn Gottfried and Hamid K. Aghajan. Vol. 3. Ambient Intelligence and Smart Environments. IOS Press, pp. 98–128. ISBN: 978-1-60750-459-7.

Kemp, John (2007). “The Colregs and the Princess Alice”. In: *TransNav, the International Journal on Marine Navigation and Safety of Sea Transportation* 1.1, pp. 57–61. ISSN: 2083-6473.

Kolendo, Piotr, Roman Smierzchalski, and Bartosz Jaworski (2011). “Experimental Research on Evolutionary Path Planning Algorithm with Fitness Function Scaling for Collision Scenarios”. In: *TransNav, the International Journal on Marine Navigation and Safety of Sea Transportation* 5.4, pp. 489–495. ISSN: 2083-6473.

Kreutzmann, Arne, Immo Colonius, Diedrich Wolter, Frank Dylla, Lutz Frommberger, and Christian Freksa (2013). “Temporal logic for process specification and recognition”. In: *Intelligent Service Robotics* 6.1, pp. 5–18.

Kwiatkowska, Marta, Gethin Norman, and David Parker (2011). “PRISM 4.0: Verification of Probabilistic Real-time Systems”. In: *Proc. of 23rd International Conference on Computer Aided Verification (CAV’11)*. Ed. by Ganesh Gopalakrishnan and Shaz Qadeer. Vol. 6806. LNCS. Springer, pp. 585–591.

Mohamed-Seghir, Mostefa (2012). “The branch-and-bound method and genetic algorithm in avoidance of ships collisions in fuzzy environment”. In: *Polish Maritime Research* 19.S1, pp. 45–49. ISSN: 1233-2585.

Moratz, Reinhard (2006). “Representing Relative Direction as a Binary Relation of Oriented Points”. In: *17th European Conference on AI*. Ed. by Gerhard Brewka, Silvia Coradeschi, Anna Perini, and Paolo Traverso. Italy: IOS Press, pp. 407–411. ISBN: 1-58603-642-4.

Necula, George C. (1997). “Proof-carrying code”. In: *Proceedings of the 24th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*. POPL ’97. Paris, France: ACM, pp. 106–119. ISBN: 0-89791-853-3.

Pietrzykowski, Zbigniew and Janusz Uriasz (2010). “Knowledge Representation in a Ship’s Navigational Decision Support System”. In: *TransNav, the International Journal on Marine Navigation and Safety of Sea Transportation* 4.3, pp. 265–270. ISSN: 2083-6473.

Pnueli, Amir (1977). “The temporal logic of programs”. In: *Proceedings of the 18th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 46–57.

Smierzchalski, Roman and Zbigniew Michalewicz (Sept. 2000). “Modeling of ship trajectory in collision situations by an evolutionary algorithm”. In: *Trans. Evol. Comp* 4.3, pp. 227–241. ISSN: 1089-778X.

Szłapczynski, Rafał (2010). “Evolutionary Sets of Cooperating Trajectories in Multi-Ship Encounter Situations-Use Cases”. In: *TransNav, the International Journal on Marine Navigation and Safety of Sea Transportation* 4.2, pp. 191–196.

Wolter, Diedrich, Frank Dylla, and Arne Kreutzmann (2011). “Rule-Compliant Navigation With Qualitative Spatial Reasoning”. In: *Proceedings of the 4th Intl. Robotic Sailing Conference*. Springer.

Wolter, Diedrich and Jan Oliver Wallgrün (2010). “Qualitative Spatial Reasoning for Applications: New Challenges and the SparQ Toolbox”. In: *Qualitative Spatio-Temporal Representation and Reasoning: Trends and Future Directions*. Ed. by Shyamanta M. Hazarika. Hershey (PA), USA: IGI Global.

5 Conceptual Neighborhood Logic with Partially Grounded Information for Safe Navigation

Arne Kreutzmann[1] and Diedrich Wolter[1,2]

1 Cognitive Systems Group, University of Bremen, Bremen, Germany

2 Smart Environments, University of Bamberg, Bamberg, Germany

Manuscript submitted to the journal *Artificial Intelligence* for the special issue "AI and Robotics"

Submitted:	6th of September, 2013
Revised:	8th of May, 2014
Status:	under review

Contributions:

I conducted the study, developed the theoretical part and implemented the algorithms. Diedrich Wolter did the proof of NP-hardness and extended SparQ where necessary. Preparing the manuscript was a joint effort with both author contributing equally.

Acknowledgements:

We would like to thank the anonymous reviewer for their helpful feedback, and Cyrill Stachniss and Giorgio Grisetti for making their data available. This work is partially funded by the DFG, financial support is gratefully acknowledged.

Abstract

Knowledge about spatial configurations and how they develop over time enables intelligent robots to reason about actions. The seamless connection of high-level deliberative processes to perception and action selection remains a challenge though.

This paper demonstrates how spatial reasoning can be used to tackle the important problem of safe action selection in navigation of robotic systems. Our aim is to verify that actions selected by the robot do not violate navigation or safety regulations and thereby endanger the robot or others. Mastering this requirement is an important step towards creating shared human-robot environments. We combine contributions from the fields of qualitative spatial reasoning and temporal logics. While qualitative spatial representation techniques integrate perception and domain knowledge, sound reasoning provides the means to obtain guarantees of rule-compliant action selection. Our approach advances qualitative spatial representations by linking spatial change to time and by introducing expressivity to capture non-spatial knowledge. We propose Conceptual Neighborhood Logic, a spatial logic in the sense of Aiello, Pratt-Hartmann & van Benthem which is a modal logic based on Freksa's conceptual neighborhoods. Further we show how navigation rules can be incorporated as background knowledge and how a robotic system can exploit this knowledge during path planning as well as during action execution. In case studies we demonstrate how our logic approach generalizes previous approaches to safe navigation in robotics and that it can handle complex navigation rules in industrial settings.

5.1 Introduction

Robotic systems are becoming more common in industrial and home environments, although crucial questions of safety have not been answered yet. How can we make make sure—ideally *guarantee*—that a robot will not endanger or harm others? Safety concerns particularly apply to heavy robotic systems that operate in shared spaces in which they easily could injure humans. Aside from elementary safety components such as reactive collision avoidance behavior, further steps are necessary to achieve safe co-existence of humans and robots.

One step towards safe navigation is provably correct behavior, i.e., to ensure that there are no software bugs with respect to algorithm design and implementation that can lead to undesired navigation behavior. Successful experiments alone are not sufficient to warrant that a robotic system will not malfunction in some environment and in some situation. A possible solution to this problem is to foster formal analysis of motion behavior (Bouraine, Fraichard, and Salhi, 2012; Täubig et al., 2012).

So far, sound and complete analysis of motion behavior is performed manually—a highly involved process that needs to be applied whenever the motion algorithms are adapted. In this paper we describe a flexible and fully automated approach to verifying safety of action selection. We show how AI techniques can provide a sound framework for reasoning about navigation decisions that allows abstract assertions like “give way to other vehicles” to be modeled declaratively and their implications on control parameter selection to be determined automatically.

Another step towards safety in navigation in shared spaces is to respect the traffic regulations implemented. For example, industrial environments often adapt traffic rules from road traffic. Regulations make the behavior of others more predictable, lead to much smoother and efficient flow of traffic, and help to avoid dangerous situations. Adhering to such rules is imperative and no one (and no robot) should be allowed to participate in traffic until suitable knowledge about applicable regulations and their observance can be certified. For a robotics engineer the challenge is to encode regulations which are denoted in natural language, e.g., “left yields to right”. It can be a error-prone endeavor to implement such rules inside the navigation components of the robot and it remains difficult to verify that the robot acts in compliance with these rules.

The contribution of this paper is to show how the problem of safe and goal-directed navigation can be tackled with spatial reasoning and how traffic regulations can be represented declaratively. We present Conceptual Neighborhood Logic (CNL), a flexible logic for specification of navigation behavior capable of representing navigation regulations with cognitive concepts about space, enabling intuitive specifications. By integrating individually successful approaches for representing space, time, and change we obtain a sound *spatial logic* in the sense of Aiello, Pratt-Hartmann, and van Benthem (2007b). We also show how reasoning in this logic can be integrated with robot perception to improve safety of navigation. Given a declarative specification of navigation regulations as input, the technique described in this paper allows limits on control actions to be determined automatically, for example the maximum speed that guarantees clearance for evasive actions. We advance over previous work in robotics, in particular (Bouraine, Fraichard, and Salhi, 2012; Täubig et al., 2012), by describing a new method capable of performing the manual analysis presented in the respective articles in an automated manner. Our reasoning method proves to be fast enough for online monitoring of robotic systems. By specifying motion behaviors using CNL, our technique can also be used to identify dead spots or contradictions in a specification. This yields a significant step towards verifying complex robot motion behaviors.

Our work draws motivation from the field of qualitative spatial and temporal reasoning (QSR) (Cohn and Renz, 2007; Cohn and Hazarika, 2001) which aims at capturing common sense in a symbolic representation. Spatial concepts used in QSR can reflect the cognitive level of human spatial understanding (see e.g., (Klippel and Montello, 2007; Knauff, Rauh, and Renz, 1997)) and thus provide a suitable basis for intuitive specification languages. In order to apply QSR to specifying navigation behavior, various aspects of spatial knowledge need to be considered like topology (e.g., being *inside* a specific region) or directional knowledge (e.g., vehicles approaching from the *left*). These aspects also need to be integrated with temporal knowledge and temporal reasoning. So far, qualitative approaches solely address a specific aspect or domain and they commonly remain isolated in the sense that combinations with other domains or domain-independent knowledge are not considered. We overcome this shortcoming in two steps. First, we formalize qualitative spatial relations from distinct representation languages in an integrative framework that can also incorporate grounded knowledge obtained by perception of the environment. Second, we connect this representation with notions of time and change using a modal logic. This new spatial logic also contributes to declarative approaches to robot programming. In general, the theory of spatial logics remains an open question, nicely reflected by the title of an article by Kontchakov et al. (2007): “Spatial logic + temporal logic = ?”. This paper provides an application-specific answer to that question by showing how different forms of reasoning (spatial, temporal, domain-independent) can be integrated into a single formal framework. Formal, automated reasoning enables the robot to determine goal-directed navigation actions that are also admissible with respect to a set of navigation rules, reaching far beyond capabilities of self-determined collision avoidance. We show how our new logic relates to popular logics used in software verification. In case studies we demonstrate that the approach can be applied to robotic systems and how it generalizes previous approaches.

This paper is organized as follows. We first discuss related approaches to fostering safety of navigation, Section 5.3 then gives an overview of our approach. We then present the main components for spatial reasoning (Section 5.4) and the logic CNL (Section 5.5). Thereafter, we discuss formalization of navigation rules (Section 5.6) and present case studies to showcase the capability of this approach (Section 5.7) before concluding with summary and discussion.

5.2 Related work

Safety in navigation reaches far beyond so-called collision avoidance. As (Bouraine, Fraichard, and Salhi, 2012) notes, from a limited set of experiments one cannot infer that robots, once deployed in large amounts and among humans, will not cause any harm. The authors formulate the ultimate goal of formally proving that a specific control scheme will not lead to collisions. Here, one important concept is that of the *inevitable collision state (ICS)* (Bouraine, Fraichard, and Salhi, 2012) which is any state of the robot from which a collision cannot be prevented. Safe navigation thus means to not enter an ICS. But how can we guarantee that this will not happen?

Generally, such guarantees require certain assumptions about the environment and the robot sensors. Often, motion of other agents (humans or robots) is assumed to be predictable. Not assuming adequate motion behaviors of others under stringent exclusion of collision possibilities would otherwise lead to too conservative driving behavior. Consider for example the case of regular road traffic where one commonly assumes that, traffic on neighboring lanes will not suddenly steer towards one's own lane, aiming to collide. Without an assumption of rule-compliant behavior of others, a guarantee for collision-free driving seems infeasible.

In order to give guarantees about collision-free driving, (Bouraine, Fraichard, and Salhi, 2012; Täubig et al., 2012) define spatio-temporal constraints that describe safety-zones. A formal proof then guarantees absence of collisions, given that all motions are bounded by certain limits (in particular maximum speed). Assumptions about further navigation constraints are not made and henceforth the approach does not handle traffic on multiple lanes as discussed above.

Although we pursue similar aims, the methodology of our approach is different. While the manual proofs given in (Bouraine, Fraichard, and Salhi, 2012; Täubig et al., 2012) would need to be redone for different scenarios, we aim to circumvent the need for manual proofs by employing sound automated reasoning techniques that allow safety constraints to be determined automatically on the basis of a formal specification of navigation regulations. Formal logics have previously been applied to control robots and to reasoning about spatio-temporal constraints. The most prominent approach is the situation calculus (McCarthy, 1963) on which the robot control language Golog (Levesque et al., 1997) is based. Bhatt, Rahayu, and Sterling (2006) presents an extension of this approach to include spatio-temporal constraints. The crucial drawback of this approach is however that it yields an undecidable logic, i.e., we can not have a sound reasoning method and thus no safety guarantees can be obtained. Therefore, we propose a new logic that allows for sound reasoning.

For integrating safety constraints with motion controllers, (Chung et al., 2009) propose to annotate maps with control parameter limits applicable for a specific robot pose. They present a sound way to determine speed limits imposed by visibility constraints. We adopt this idea since it allows safety constraints to be considered already during path planning. However, this interface is not expressive enough to handle complex regulations that involve local context, for example, to indicate that a robot has to stop in order to give way to another vehicle. Such requirements can only be handled by an online approach which has access to the current state and context of the robot. We are thus interested in sound logic reasoning techniques that seamlessly integrate with perception of the robotic system.

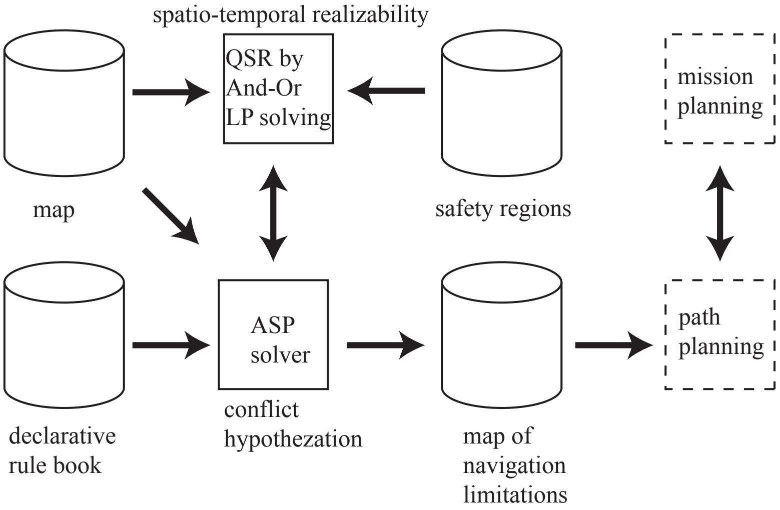


Figure 5.1: System overview for offline application

5.3 Safe navigation by spatio-temporal reasoning

This section gives an overview of how Conceptual Neighborhood Logic (CNL) and its according reasoning techniques can foster safe robot navigation. Before detailing CNL let us just say that the logic is expressive enough to specify inevitable collision states and that it can also represent statements like "after executing action A, the safety-region of the robot will not overlap with the safety-region of another vehicle".

The spatial domain considered in CNL consists of points, lines, and polygons of known shape positioned in Euclidean space $\mathbb{R}^2$ as well as their orientations. The set of orientations is discretized to a finite set by means of an upper approximation, growing safety regions to accommodate for orientation variation. Discretizing orientations is necessary to obtain efficient reasoning by avoiding non-linear relations (Lee, Renz, and Wolter, 2013; Wolter and Lee, 2010). On the temporal side, CNL represents sequences of configurations that are linked by spatio-temporal continuity constraints. With respect to the aforementioned example, the future location of the robot after execution of action A would be determined by the action effects and by the spatio-temporal changes possible in the current state of the robot.

We consider two applications of CNL. First, we consider an offline application in the sense that reasoning is performed once before the robots starts to plan and execute a path. This use case resembles an approach by Chung et al. (2009) where an heatmap-like[1] representation is computed to indicate areas of navigation risks (see Figure 5.13 for a map computed by our system). The aim is to make path-planning aware of areas that require careful, slow driving. In Figure 5.1 we depict an overview of our approach. Interface to the robot is provided by a map of navigation limitations which in our experiments are speed limitations required for

[1] sometimes called costmap

clearance of the braking path. These limitations take into account potential collisions with obstacles registered in the map as well as potential collisions with moving obstacles that are not yet within sensor range. For example, if approaching an intersection with poor visibility, conflicts with vehicles concealed by other obstacles have to be considered too. In order to obtain this map we start with a declarative rule book of navigation rules and a (possibly coarse or simplified) map of the environment. We describe a simple specification language to represent navigation rules which essentially is a restricted family of CNL formulae (Section 5.6.1). The rule book is assumed to include a rule that disallows entering an ICS. Similar to (Chung et al., 2009), map and control commands are discretized. For each control command, a safety polygon needs to defined giving the region that may be visited by the robot in order to carry out the command, taking the discretization of positions and control actions into account. We do not aim to capture kinematics as precise as possible but to approximate them conservatively, i.e., rather overestimating safety regions, never underestimating them. The field of view of the robot's sensor system is specified as a polygon too. We assume that sensors are suitable for safety-critical applications in the sense that they detect all obstacles not occluded within the specified field of view—safety laser rangefinders meet this requirement. For each combination of map position and control action, reasoning is applied to check whether performing the action would entail violation of a rule, e.g., an inevitable overlap of safety regions. Admissible actions are then registered in the map.

In our approach, reasoning is essentially accomplished by Answer Set Programming (ASP) which is a "form of declarative programming oriented towards difficult search problems" (Lifschitz, 2008). A readily available ASP solver searches for a temporal sequence of spatial configurations that witnesses a violation of some navigation rule. At this stage, possible whereabouts of other obstacles (e.g., moving vehicles) have to be explored. On the symbolic level of qualitative representations this can be accomplished by the ASP solver as there are only finitely many qualitative configurations for any finite set of objects. This maximum number of objects to consider is given by the number of objects referred to by the navigation rule. Most rules only involve one object other than the robot. A subsequent step of spatio-temporal reasoning is carried out to check that the symbolic scene descriptions determined by the ASP solver are realizable in the continuous and infinite spatio-temporal domain.

The design of CNL enables spatio-temporal continuity to be handled purely on the symbolic level by ASP (Definition 9, Theorem 2). Two aspects of spatial realizability need to be checked separately though, filtering out scene descriptions that are not realizable in the spatial domain. A scene description may involve relations that are not jointly satisfiable due to their spatial semantics. For example no planar scene can exist that simultaneously requires robot #1 to be ahead of robot #2, robot #2 to be ahead of robot #3 and robot #3 to be ahead of robot #1. This kind of reasoning is a typical application of qualitative spatial reasoning. Second, the scene description needs to fit with the environment, for example, a narrow passage may not allow a robot to pass through. To handle both tests simultaneously we introduce a new method for qualitative spatial reasoning with partially grounded information (Section 5.4).

As a second application of CNL we consider the online application in a monitoring system that ensures admissible action selection. The corresponding architecture shown in Figure 5.2

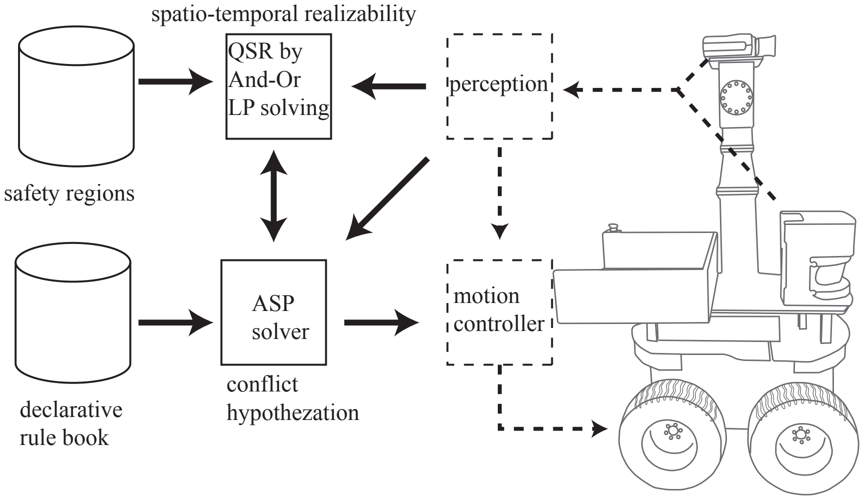


Figure 5.2: System overview for online safety assessment

is very similar to the offline application. Instead of an environment map and possible robot poses, local perception of the robot and the concrete control command suggested by the robot control program are input to our system. Reasoning determines whether the control command in the current context leads to a violation of a navigation rule. If no violation can be identified we say that it is safe to perform the motion command and proceed. It is sensible to design navigation rules that declare an action A to be safe if there exists a certain evasive action E that, if executed directly after A, will bring the robot to a safe state, e.g., a braking action E bringing the robot to a safe stop. In this case, running the monitor system in the control loop of the robot, the evasive action E corresponding to the previous command can safely be issued whenever the motion command currently suggested by the controller is not safe to execute.

5.4 Qualitative spatial reasoning with partially grounded information

Our aim is to develop an integrative framework that can jointly reason about various spatial relations, most importantly topological relations (e.g., being *inside* a region, a safety region *not overlapping* with an obstacle) and orientation information (e.g., a vehicle approaching from the *right*, one's orientation being *aligned* with the direction of an one-way street). Also, we require the framework to handle concrete entities described by numeric values alongside unknown entities described by variables ranging over the spatial domain.

5.4.1 Qualitative spatial representations—bridging cognition to logics

Qualitative spatial representations capture cognitive concepts of space and they can be regarded as link between cognition and formal methods (Freksa, 1991b). Representations based on relations such as *left of* or *north-of* are easily understandable by humans and therefore qualitative approaches are claimed to provide a solid basis for intuitive human-machine interaction or knowledge engineering (Cohn and Renz, 2007). QSR provides a rich pool of representations of space, each modeling distinct aspects of the spatial domain. The relations of a qualitative representation give rise to an algebraic structure and so the term qualitative calculus has been coined. Aside from their aim to resemble intuitive concepts of space, qualitative calculi exhibit the important technical feature of providing us with a finite set of jointly exhaustive and pairwise disjoint relations to represent knowledge about an infinite spatial domain. This enables integration with logics in the sense that a piece of qualitative information like "p is located inside Q" can be represented as atoms and, consequently, interpreted as truth values in the logic since there are only finitely many distinct atoms necessary. We later show how the ability to reason about qualitative spatial relations can be lifted to the logic level.

But how can we reason with qualitative knowledge? Commonly, QSR is approached as constraint-based reasoning in which variables range over a spatial domain (Renz and Nebel, 2007). Constraints considered in QSR are solely the relations defined by the respective calculus. Since spatial domains are typically infinite, special techniques are required to decide satisfiability of constraint satisfaction problems in QSR. Recently, attention has also been put on problems in which variables do not range freely over the spatial domain but are subject to further constraints, e.g., limited to finite subsets or singletons of the domain. Li, Liu, and Wang (2013) study restricted domains for the special case of topological relations, results for other spatial calculi are still lacking. The ability to handle such singletons seems is however a fundamental prerequisite to applying QSR in robot applications: we are not only involved with agents located *somewhere* in space, but we also need to reason about agents we observe and the location of which is known. This ability is what we refer to by reasoning with *partially grounded information*.

A second issue with QSR is that its techniques are limited to reasoning about one single specific aspect of space, i.e., to tackle constraint problems which only involve relations from one calculus. Some specific combinations of useful representations have already been researched and combined calculi developed, e.g., combinations of topological and cardinal direction knowledge (Liu, Li, and Renz, 2009). Existing approaches are however not yet expressive enough for jointly reasoning about agent position, orientation and regions.

Therefore we propose a new framework for performing qualitative spatial reasoning that can jointly handle various qualitative spatial representations and also allows partially grounded information.

5.4.2 QSR with And-Or LP trees

In robot navigation we are involved with spatial information represented by locations in Euclidean space. Since we aim to represent regions like safety regions conservatively by an upper approximation, regions can be represented by compositions of simple polygons which in turn can be described as conjunctions of linear inequalities. This allows us to approach spatial reasoning on basis of linear programming techniques in a similar way as presented in (Lee, Renz, and Wolter, 2013):

Definition 3. Linear programming (LP) is the task of solving a set of linear inequalities $\mathbf{A}\vec{x} \leq \vec{b}$ for $\vec{x}$ with $\mathbf{A} \in \mathbb{R}^{n \cdot m}$, $b \in \mathbb{R}^n$ and $\vec{x} \in \mathbb{R}^m, \vec{x} \geq 0$.[2] LPs can be solved in polynomial time (Schrijver, 1986).

By definition, LP restricts the range of variables to non-negative values only. This restriction can easily be circumvented by modeling $x \in \mathbb{R}$ as $x = x^+ - x^-$ using two fresh variables x^+, x^- subject to non-negative value restriction. Multiple valuations of x^+, x^- then stand for the same value x. Some qualitative relations can easily be posed as an LP, for example the constraint that a point $\vec{x} \in \mathbb{R}^2$ is located inside a triangular area can be written in this form using three inequalities that define the three half-planes that delimit the triangle. Aside from inequalities of the kind "$\leq$", equalities can also be expressed by rewriting $\phi = \beta$ to $\phi \leq \beta$ and $-\phi \leq -\beta$. In practical applications we can also simulate $<$ or $>$ by using an appropriate ϵ value, for example by defining that the interior of a triangle needs to be at least 1mm away from the boundary. In the following we assume that such rewriting is applied.

Yet we are faced with the problem that not all qualitative relations can be posed as LP. For example, in order to state that a location is outside a triangle we require disjunctions: A location is outside if at least one of the inequalities (i.e., half-planes) defining the triangle is not satisfied—but we do not know which. In order to handle such situations we introduce And-Or trees to spatial reasoning with linear inequalities.

Definition 4. An *And-Or LP Problem* is a tree whose inner nodes are labeled either "AND" or "OR" and whose leaves are of the form $\mathbf{A}_l \cdot \vec{x} \leq \vec{b_l}$ with $\mathbf{A}$ being a $n_l \times M$ matrix and M being the same for all nodes. Thus, $\vec{x}$ is an element of $\mathbb{R}^M$. We say that an AND-Or LP problem is satisfiable if there exist $\vec{x^\star} \in \mathbb{R}^M$ such that all nodes labeled with an LP are annotated with "true" if $\mathbf{A}_l \cdot \vec{x^\star} \leq \vec{b_l}$ holds and the tree evaluates to "true" using the usual semantics, i.e., a node evaluates to true

- if it is labeled "AND" and all its children evaluate to true,
- if it is labeled "OR" and one of its children evaluates to true,

and to false otherwise. An example of such tree is shown in Figure 5.6(a).

[2] In mathematical optimization, LPs are commonly extended with a linear function $\vec{w}^T\vec{x}$ that is to be optimized by $\vec{x}$. This is not required in our approach as we are interested in existence of a solution only.

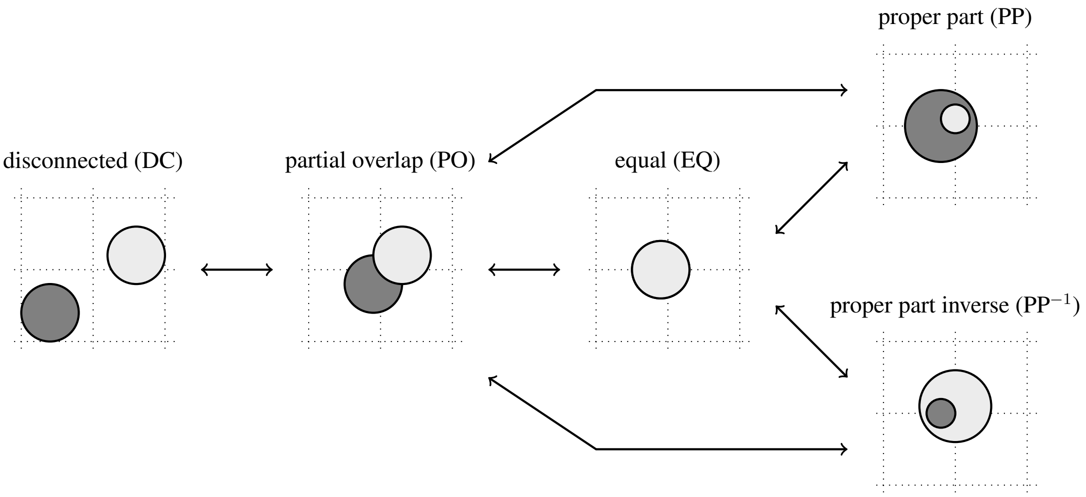


Figure 5.3: Region Connection Calculus (RCC-5) relations and the conceptual neighborhoods of the relations are indicated by the arrows.

Theorem 1. *Deciding satisfiability of an instance of an And-Or LP problem is NP-complete.*

Proof sketch. NP hardness immediately follows by a reduction to SAT. NP membership follows from the observation that we can guess the leaves that will evaluate to true and, by adjoining the LPs attached to these leaves, we obtain a single LP which can be solved in polynomial time. □

We now detail how relations from relevant qualitative calculi can be encoded in And-OR LP trees. In principle, our approach allows all spatial relations to be represented which can either be specified by linear relations (linear ordering relations, for example) or relations that can be described by a finite disjunction of linear relationships. Exploring the expressivity of And-Or LP trees is the scope of this paper and we restrict presentation to relations used in our case studies.

5.4.3 Region Connection Calculus (RCC)

RCC Randell, Cui, and Cohn, 1992 is a popular formalism for representing relationship between regions. In our approach we consider the set of RCC-5 relations which is depicted in Figure 5.3. The exact equality relation EQ is not useful to represent knowledge in context of regions obtained by interpreting noisy sensor data though and will therefore not be used. As a byproduct of specifying RCC-5 relations as And-Or LP trees we also obtain point-to-polygon relations “inside” and “outside” which are themselves useful primitives.

As domain we consider compositions of simple (convex) polygons which are polygons in which no edges share a common point, except for the start and end point of consecutive edges. Section 5.4.3 discusses how the non-convex polygons are handled, but for the reminder of this section we assume the polygons to be convex and their vertices to be ordered counterclockwise.

Point inside/outside of a polygon

A point is said to be inside a simple convex polygon iff it is positioned on the left side of all edges. Checking whether a point is positioned left of (or right of) a directed line can be achieved by comparing the dot product between the point and the normal of the edge vector to zero. Outside can be defined analogously using disjunction, requiring that the point lies on the right side of at least one edge. Representing *inside* as And-Or LP tree only a single LP Node is required, whereas *outside* involves an OR-node with (many) LP nodes is required, one for each edge.

Disconnected (DC)

Lemma 1. *If the two simple convex polygons are disconnected, there exists a dividing straight line that is parallel to one of the edges of the polygons. This dividing straight line partitions the space such that exactly one polygon is on each side of the line.*

Proof. This is a direct application of the method of separating axis based on the Minkowski's *hyperplane separation theorem.* □

The previous lemma can be employed directly to obtain an encoding of the relation DC as And-Or LP problem: The root node is an *OR* node with one child for each edge E_i in the two polygons. Each child node is itself a root node of a subtree. This subtree representing that all vertices of one of polygon P with E_i being and edge of P are on the left hand side of E_i, whereas all vertices of the other polygon are on the right hand side of E_i.

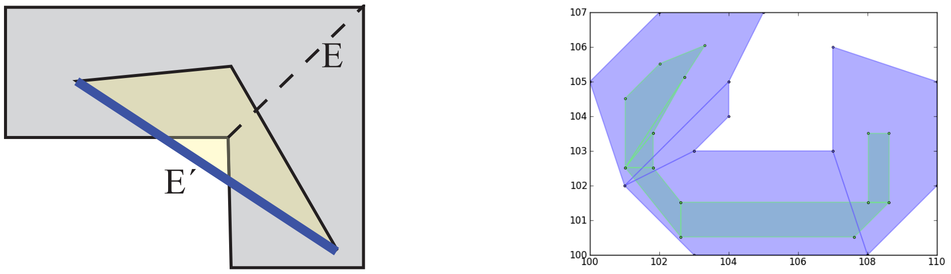
Partial overlap (PO)

In order to define relation PO, we first define the relation *overlap* that is the complement of the RCC relations DC: Two polygons P_1, P_2 overlap if there exists a point p such that p is inside P_1 and P_2. The overlap relation can be easily encoded as a single LP problem according to 5.4.3, using a fresh pair of variables x, y to represent p.

Partial overlap refines overlap in that there also need to exist two points p'_1 and p'_2, such that p'_i is inside P_i and outside of the other polygon. This leads to a graph with a root *and* node that has three children, one representing the subtree of relation overlap, the other two representing existence of vertices that are inside exactly one of the polygons.

Proper part (PP) and proper part inverse (PP^{-1})

A polygon is part of another polygon if all vertices of the inner one are inside of the containing polygon, thus yielding the encoding of an *AND* node with LP node children for each vertex.



(a) The yellow polygon is not a proper part of the grey one since inner edge E introduced by the convex partitioning does not cross edge E'

(b) complex example of proper part

Figure 5.4: Proper part for non-convex polygons.

Handling non-convex polygons

For applying the techniques to non-convex polygons a partition into convex part is applied, any partition scheme can be used, for example with the algorithm by Hertel and Mehlhorn (1983)[3]. Once the partition has been computed, the encodings previously described are extended to take into account that a region can be made up of multiple parts. For example, if a non-convex polygon is said to be inside a convex polygon, then each part of the partition needs to be inside the enclosing polygon, introducing an *AND* node. This generalization is generally straightforward, e.g., in the case of relation DC this means each convex part has to be disconnected to each convex part of the other polygon, see Figure 5.6(b). Only relations PP and PP^{-1} require special attention. In Figure 5.4 a corner case is shown that illustrates that for the proper part relation it is not sufficient to require vertices of the inner polygon (yellow) to be inside another non-convex polygon (grey). In general, specifying proper part for two non-convex polygons also needs to acknowledge effects caused by convex partitioning. We call an edge introduced by the convex partitioning an inner edge. The special case arises if an edge E of the inner polygon overlaps with a sequence of different, adjacent convex parts of the outer polygon. In such cases all inner edges of this sequence need to cross E. This is formalized by stating that one endpoint of the inner edge lies to the left of E and the other to the right of E.

5.4.4 Cardinal directions

We consider the Star calculus (Renz and Mitra, 2004) to represent cardinal direction relations since it is very flexible and an intermediate step towards representing relative orientation knowledge. Star is based on a cone-shaped partition scheme that defines sectors $0, \dots, K$ with a variable granularity parameter K, see Figure 5.5 for a depiction. Star relations are defined by intersection of two half-planes and a constraint $(p\,s\,q)$ with $p = (p_x\,p_y)^T$, $q = (q_x\,q_y)^T$, and $s \in \{0, \dots, K-1\}$ can thus be modeled as LP problem:

[3] For our system implementation we use the version provided by the CGAL library (CGAL, *Computational Geometry Algorithms Library* n.d.).

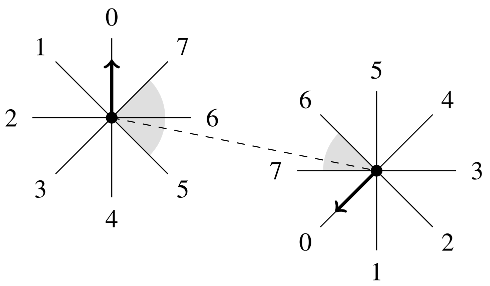


Figure 5.5: Qualitative relations defined by the Star calculus and its extension StarVars that allows relative directions to be represented

$$A_{s,K} := \left(\cos(s\tfrac{2\pi}{K})\,\sin(s\tfrac{2\pi}{K})\right), \quad \begin{pmatrix} A_{s,K} \\ -A_{s+1,K} \end{pmatrix} \begin{pmatrix} q_x - p_x \\ q_y - p_y \end{pmatrix} > (0\,0)^T \tag{5.1}$$

A realization of a consistent set of Star constraints can directly be read off the the solution $\vec{x}^\star$ of the LP.

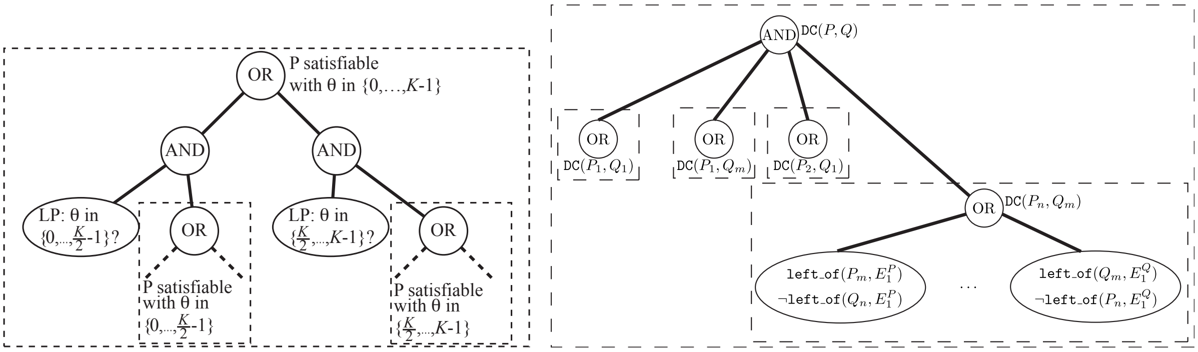
5.4.5 Relative directions

Navigation regulations often involve directional knowledge, for example the relation “left of” occurring in “left yields to right” needs to be interpreted with respect to the current orientation of a vehicle. The StarVars calculus (Lee, Renz, and Wolter, 2013) defines relations capable of representing relative directions such as “left of”. StarVars generalizes Star by augmenting the carnival direction relations of Star with orientation knowledge to represent relative directions. Technically speaking, StarVars introduces orientation variables that offset the globally aligned Star sector indices, aligning the zero direction with vehicle orientation (see Figure 5.5). As given by Lee, Renz, and Wolter, 2013, Lemma 8, satisfiability of a single StarVars relation $s \in \{0, 1, \ldots, K-1\}$ of granularity K between points (x_1, y_1) and (x_1, y_1) with orientations $\Theta_p, \Theta_q \in \{0, K-1\}$ can be written similar to (5.1) as

$$\begin{pmatrix} A_{s+\Theta_p,K} \\ -A_{s+1+\Theta_p,K} \end{pmatrix} \begin{pmatrix} q_x - p_x \\ q_y - p_y \end{pmatrix} > (0\,0)^T \tag{5.2}$$

that is linear in p_x, p_y, q_x, q_y but nonlinear in Θ_p. Since there are finitely many Θ_i a straightforward encoding into an And-Or LP problem would be possible at the cost of introducing one Or-node with K leaves that each represent a distinct choice of Θ_i by replacing Θ_i in (5.2) with the according constant. This approach is however not practical since a reasonable resolution of orientations leads to very large disjunctions.

The search algorithm proposed in (Lee, Renz, and Wolter, 2013) to decide consistency of StarVars constraints gives however rise to a hierarchical encoding scheme. In a divide-and-conquer manner (assuming K to be a power of 2), one first checks whether an orientation Θ could be found in the set$\{0, \ldots \frac{K}{2} - 1\}$ or $\{\frac{K}{2}, \ldots, K-1\}$ using a coarsened problem, then



(a) Recursive encoding scheme of StarVars constraints

(b) Encoding scheme of disconnected (DC) constraints, where the P_i, Q_j are convex parts of the respective polygons

Figure 5.6: Examples of And-Or LP problem encodings, dashed boxes indicate (sub-)trees and the labeling indicates their intended semantics.

recursively refining the respective interval. Deciding the coarsened problem is again an LP in the form of (5.2). The StarVars reasoning strategy can hence be adapted to and And-Or tree in the way illustrated in Figure 5.6(a), reducing tree width from $O(K)$ to $O(\log K)$. Using this encoding, a model that satisfies StarVars constraints can be obtained from a satisfiable And-Or LP tree.

5.4.6 Exploiting implicit Constrains of Partial Solutions

Deciding consistency of an And-Or LP problem can benefit from knowledge about the qualitative reasoning problem encoded in the tree. Similar to the trees illustrated in Figure 5.6(a), we retain some semantics of the qualitative spatial relations encoded to implement a caching mechanism. Nodes may be attributed with spatial constraints, e.g., fixing the range of an orientation Θ of some entity or stating a point-polygon relationship. If an annotated node gets evaluated, the annotated constraint and the according truth value determined are posted on a black board structure. If within the same branch of the And-Or tree another node is encountered that exhibits a label posted on the black board, costly re-evaluation of that node can be avoided, using the value posted on the black board. This cacheing is also exploited as heuristic when branching at Or-nodes occurs. Children that involve fewest nodes with unknown values are examined first.

5.4.7 Exploiting spatial structure to reduce problem size

Choosing the smallest And-Or LP tree among a set of equivalent trees reduces problem size and hence can improve reasoning efficiency. Determining the smallest And-Or LP tree among all equivalent trees would be an NP-complete problem in its own right though. While constructing an And-Or LP tree from a set of spatial constraints we realize two heuristics to reduce the size of the resulting tree.

Pruning redundant constraints

There are two types of redundant constraints that can be pruned away. First, if constraints refer to grounded knowledge, one can simply evaluate their truth values, avoiding the constraints to be encoded as subtrees in the And-Or LP tree. Second, constraints are redundant if they are implied by other constraints.

For example, the formula

$$\phi := \texttt{properPart}(A, B) \land \texttt{overlaps}(C, A) \land \texttt{overlaps}(C, B)$$

can be reduced to

$$\phi' := \texttt{properPart}(A, B) \land \texttt{overlaps}(C, A).$$

since A is part of B and hence $\texttt{overlaps}(C, B)$ follows from $\texttt{overlaps}(C, A)$. This procedure can be automated by constraint propagation, iteratively removing any constraint that follows from relation composition. At this step, composition tables defined for qualitative calculi are exploited that represent all valid consequences $t(A, C)$ that follow from $r(A, B) \land s(B, C)$ for any combination of constraint relation r, s (see Renz and Nebel (2007)).

Precomputing regions

If a sub-formula describes a uniquely identified region, this region can be precomputed, allowing the sub-formulae to be rewritten to refer to the new region. Consider the following example that involves grounded regions A and B (i.e., constants) as well as variable X that stands for an undetermined region:

$$\phi := \texttt{overlaps}(A, B) \land \texttt{disconnected}(A, X) \land \texttt{properPart}(X, B)$$

Obviously, X is required to be a proper part of $B \setminus A$. Since A, B are known we can introduce a new region $D := B \setminus A$ and rewrite ϕ to

$$\phi' := \texttt{properPart}(X, D).$$

5.5 Spatio-temporal reasoning with CNL

We present *Conceptual Neighborhood Logic (CNL)* to represent navigation regulations and to reason about navigation. Aiello, Pratt-Hartmann, and van Benthem (2007b) define *spatial logic* as “any formal language interpreted over a class of structures featuring geometrical entities and relations” in analogy to how temporal logic is defined over a structure of temporal relations.

A spatial logic can be interpreted over a structure inhabiting any class of geometrical spaces, such as topological spaces, projective spaces, or some Euclidean space; Aiello, Pratt-Hartmann, and van Benthem (2007a) provides a comprehensive overview. In navigation it is foremost necessary to represent static (but possibly changing) spatial information like the regions of passable space, safety regions, obstacles, etc. as well as their interrelation, for example the fact that the position of the robot is inside a certain region. We utilize qualitative spatial relations whose semantics can be represented as And-Or LP trees. Navigation regulations are further involved with talking about possible situations and stating which actions should or should not be carried out. Situations can be regarded as patterns that involve unknowns and thus require variables. For representing actions we opt for including a temporal component in our logic in conjunction with a notion of spatial change, the so-called *conceptual neighborhood* (Freksa, 1991a).

Conceptual neighborhoods have been introduced to augment existing qualitative calculi with a notion of change, specifying which relation changeovers on the level of qualitative relations are possible if the underlying model is continuously varied. This leads to the conceptual neighborhood graphs that represent possible changeovers as edges in a graph, see Figure 5.3 for the case of RCC-5. The approach has later been extended to also acknowledge the temporal implications of these transitions (Galton, 1995).

If a relation r_1 can be continuously transformed into a relation r_2, then r_1 is said to dominate r_2 if r_1 has to hold at the (time) point of transition. Galton (2000) calls the quantitative level the *phase space* and the qualitative level the *mode space* and further generalizes the notion of conceptual neighborhood and dominance spaces into the *topological mode spaces*, which we will discuss further in the next section.

5.5.1 Spatial and temporal logic

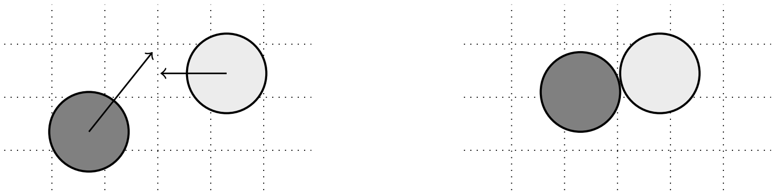
As space and spatial change can be captured by qualitative approaches we are essentially looking for an efficient temporal logic. There are two significant approaches that each provide a different view on time. Pnueli (1977) regard time to be linearly evolving and developed an according logic, the *linear temporal logic* (LTL). By contrast, Clarke and Emerson (1982) regards time to be branching and proposed *computational tree logic* (CTL). Both approaches have been generalized to the logic CTL*which is a superset of both LTL and CTL (Emerson and Halpern, 1986). Formulae in CTL*can be used with state of the art model checkers, e.g., PRISM (Kwiatkowska, Norman, and Parker, 2011) is capable of efficiently handling CTL*and therefore capable of LTL as well as CTL. Kontchakov et al. (2007) show that even the combination of a decidable spatial logic and a decidable temporal logic easily results in too expressive and therefore undecidable formalisms. Consequently, for each combination of logics a fine-tuned analysis is required.

5.5.2 Topological mode space

Topological Mode Spaces introduced by Galton describe possible *qualitative* changes in complex scenarios. This section summarizes the fundamental definitions and the product theorem on which the semantics of the conceptual neighborhood logic is build. For more details, see (Galton, 2000).

Definition 5. Given two qualitative relations q_1 and q_2, such that q_1 holds at t_1 and q_2 holds at t_2 and either q_1 or q_2 hold over the open interval $]t_1, t_2[$, then the following relations between q_1 and q_2 can hold:

$q_1 \vdash q_2$	q_1 pre-dominates q_2,	if q_2 holds over $]t_1, t_2]$;
$q_1 \dashv q_2$	q_2 post-dominates q_1,	if q_1 holds over $[t_1, t_2[$.



(a) t_0: disconnected (DC)

(b) t_1: externally connected (EC)

Figure 5.7: Two closed regions are continously transformed in the quantitative phase space resulting in a change on the qualitative level. During the interval $[t_0, t_1[$ DC holds and because EC holds at t_1, we say that EC post-dominates DC.

Definition 6. A *topological mode space* is a set Q of qualitative states together with two binary relations $\vdash$ and $\dashv$ on Q. A *model* for a topological mode space is specified as a triple $\langle P, \mathcal{C}, \mu \rangle$, where

- $P \subset \mathbb{R}^n$, representing the quantitative phase space;
- $\mathcal{C}$ is a set of continuos functions from $\mathbb{R}$ to P;
- μ is a surjective mapping from P onto Q giving the (qualitative) mode associated with each (quantitative) phase

and for each $f \in \mathcal{C}$ the following is satisfied:

- *No intermingling*: for each pair of real numbers $t < t'$, there is a finite sequence $t = t_0 < \ldots < t_m = t'$ of real numbers, such that for $i = 1, \ldots, m$, if $t_{i-1} < u < v < t_i$ then $\mu(f(u)) = \mu(f(v))$;
- *Dominance*: If $\mu(f(t))$ = q for all $t \in (t_0, t_1)$ then $\mu(f(t_0)) \vdash q$ and $q \dashv \mu(f(t_1))$.

Perturbation abstracts from the topological character of the transition and states that a direct transition from q to q' is possible: $q \rightsquigarrow q' := q \vdash q' \vee q \dashv q'$.

Unlike binary relations of change, application need to talk about changes involving more than two or three entities and might even require combinations of distinct qualitative representations. When combining qualitative representations their respective topological mode spaces need to be combined as well. Galton developed a product theorem showing how a joint topological mode space can be obtained, see Galton, 2000, page 359.

Dominance spaces and conceptual neighborhood

Topological mode spaces are an extension of conceptual neighborhood (Freksa, 1991a) as well as dominance spaces (Galton, 1995). Galton defines a notion of *conceptual neighborhood* similar to the notion defined by Freksa (1991a) $(q \sim q')$:

$$q \sim q' \quad := \quad q \rightsquigarrow q' \vee q' \rightsquigarrow q.$$

One important difference between Galton's and Freksa's notion of *conceputal neighborhood* is that Galton allows it to be reflexive while Freksa defines it as being irreflexive. If the $\rightsquigarrow$ relation is symmetric, a topological mode space can be expressed as a dominance space.

Abstraction properties

Bäckström and Jonsson define in (Bäckström and Jonsson, 2012) various *provable* abstraction properties for transition systems and show how other established abstraction classifications can be stated within their formalism. They define *instance properties* that classify what happens to paths by abstraction. Properties holding upwards can be viewed as a classification of completeness and properties holding downwards as a classification of soundness of the abstraction.

As qualitative spatial reasoning forms a language to describe observable states and transitions, every possible transformation must correspond to a sequence of states (connected by perturbation) in the topological mode space. Of course, perturbation $(q \rightsquigarrow q')$ should not connect qualitative states when no such connection exists in the (quantitative) phase space. Bäckström and Jonsson call the first property *strong upward instance property* $(P_{S\uparrow})$ and the second *trivial downward instance property*. Nevertheless longer sequences in the topological mode space generally do not have a corresponding continuos transformation in the phase space, for example due to limitations of the kinematics of real-world object.

As a result, the abstraction provided by topological mode spaces is complete but generally only for paths of length one sound. Anything that is build on top will carry up to the same properties with respect to the underling quantitative phase space.

5.5.3 Conceptual Neighborhood Logic CNL

Conceptual neighborhood as introduced by Freksa (1991a) is irreflexive and symmetric, consequently the resulting conceptual neighborhood graphs are undirected. By contrast, the connection relation *perturbation* is directed and as a result we define directed conceptual neighborhood graphs.

Definition 7. A *directed conceptual neighborhood graph* $\mathcal{DCNG} = \langle Q, \rightsquigarrow \rangle$ is a directed graph with the set of vertices Q and the set of edges $\rightsquigarrow$. A $\mathcal{DCNG}$ is induced by a topological mode space $\langle Q, \vdash, \dashv \rangle$ and its perturbation relation $\rightsquigarrow$.

In case of a symmetric perturbation relation $\mathcal{DCNG}$s are equivalent to dominance spaces, which are a super set of Freksa's conceptual neighborhood graphs.

Definition 8. Given a $\mathcal{DCNG}$, a *path* $\pi = q_0, q_1, \ldots$ is a sequence of states such that for all $i \geq 0 : q_i \rightsquigarrow q_{i+1}$ holds. We define the following syntax to refer to sub-path $\pi[k \ldots] := q_k, q_{k+1}, \ldots$ and to refer to a single state $\pi[k] := q_k$.

Definition 9 (Conceptual Neighborhood Logic (CNL))**.** Let $\mathcal{DCNG} = \langle Q, \rightsquigarrow \rangle$ be a directed conceptual neighborhood graph of a model $\langle P, \mathcal{C}, \mu \rangle$ for a topological mode space $\langle Q, \vdash, \dashv \rangle$. Further, let a finite set of proposition symbols R be defined by assigning a unique symbol to each (spatial) relation from Q. Three modal operators are defined: $\langle \mathrm{cn} \rangle$ for conceptual neighborhood and, as with linear temporal logic: $\circ$ (next) and U (until).

Given R and the modal operators, the language *conceptual neighborhood logic* (CNL) is defined by formulae of the following type which are called path formula

$$\phi ::= \varphi \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \circ\, \phi \mid \phi_1 U \phi_2,$$

and (sub-)formulae of the following type which are called scenario formulae:

$$\varphi ::= r \mid \top \mid \bot \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \langle cn \rangle\, \varphi.$$

That is, a CNL formula is a path formula, which is either a scenario formula, boolean combination of path formulae, a path formula prefixed by $\circ$ (next), or path formulae connected by the binary modal U (until). Whereas a scenario formula is either a proposition symbol (a spatial relation), a boolean constant, a boolean combination of scenario formulae, or a scenario formula prefixed by $\langle cn \rangle$ (conceptual neighbored).

CNL semantic is defined with respect to a Kripke structure induced by the $\mathcal{DCNG}$. Thus a model $\mathcal{M}$ for CNL is defined as $\mathcal{M} = \langle Q, \{\langle cn \rangle, \circ, U\}, V \rangle$. V is a valuation function that maps a proposition symbol (i.e., spatial constraint) to the subset of Q in which it holds. The notion of a (path) formula ϕ being satisfied in a model $\mathcal{M}$ over a path π is inductively defined, starting with scenario formulae:

$$\begin{array}{lll}
\mathcal{M},\pi \models r & \text{iff} & \pi[0] \in V(r) \\
\mathcal{M},\pi \models \top & \text{iff} & \text{always} \\
\mathcal{M},\pi \models \bot & \text{iff} & \text{never} \\
\mathcal{M},\pi \models \neg\varphi & \text{iff} & \text{not } \mathcal{M},\pi \models \varphi \\
\mathcal{M},\pi \models \varphi_1 \wedge \varphi_2 & \text{iff} & \mathcal{M},\pi \models \varphi_1 \text{ and } \mathcal{M},\pi \models \varphi_2 \\
\mathcal{M},\pi \models \langle cn\rangle\,\varphi & \text{iff} & \text{exists } q' \text{ such that } \pi[0] \rightsquigarrow q' \\
 & & \text{and } \mathcal{M},(\pi' = q') \models \varphi
\end{array}$$

and the semantics of path formulae are then defined as:

$$\begin{array}{lll}
\mathcal{M},\pi \models \neg\phi & \text{iff} & \text{not } \mathcal{M},\pi \models \phi \\
\mathcal{M},\pi \models \phi_1 \wedge \phi_2 & \text{iff} & \mathcal{M},\pi \models \phi_1 \text{ and } \mathcal{M},\pi \models \phi_2 \\
\mathcal{M},\pi \models \circ\phi & \text{iff} & \mathcal{M},\pi[1] \models \phi \\
\mathcal{M},\pi \models \phi_1 U \phi_2 & \text{iff} & \text{exists } i \in \mathbb{N}_0 \text{ such that} \\
 & & \mathcal{M},\pi[i] \models \phi_2 \text{ and} \\
 & & \mathcal{M},\pi[k] \models \phi_1 \text{ for } 0 \leq k < i
\end{array}$$

The other boolean connectives are derived as usual, e.g.:

$$\begin{array}{rcl}
\phi_1 \vee \phi_2 & \Leftrightarrow & \neg\phi_1 \wedge \neg\phi_2 \\
\phi_1 \rightarrow \phi_2 & \Leftrightarrow & \neg\phi_1 \vee \phi_2
\end{array}$$

and for convenience we define five more modal operators:

$$\begin{array}{rcll}
[cn]\varphi & \Leftrightarrow & \neg\,\langle cn\rangle\,\neg\varphi & \\
\diamond\phi & \Leftrightarrow & \top U \phi & \text{(eventually)} \\
\square\phi & \Leftrightarrow & \neg \diamond \neg\phi & \text{(always)} \\
\phi_1 R \phi_2 & \Leftrightarrow & \neg(\neg\phi_2 U \neg\phi_1) & \text{(release)} \\
\phi_1 W \phi_2 & \Leftrightarrow & (\phi_1 U \phi_2) \vee \square\phi_1 & \text{(weak until)}
\end{array}$$

The last two modal operators carry the following intuition: ϕ_2 *is released by* ϕ_1, more precisely ϕ_2 either holds always or until $\phi_1 \wedge \phi_2$ holds. *Weak until* is similar to the normal until but ϕ_2 does not necessarily occur in the future.

5.5.4 Incoporating non-spatial knowledge

CNL as just defined talks about space and spatial change. Often other knowledge and how it changes needs to be represented too. Expressing change in non-spatial systems can change

Algorithm 1: Labeling Algorithm for $\langle cn \rangle \, \varphi$

```
Require: $\varphi$                                  ▷ Collect all states labeled $\varphi$
$T \leftarrow \{q \mid \varphi \in \mathrm{label}(q)\}$
while $T \neq \emptyset$ do
    choose $q \in T$
    $T \leftarrow T \setminus \{q\}$
    for all $q'$ such that $q' \rightsquigarrow q$ do
        $\mathrm{label}(q') \leftarrow \mathrm{label}(q') \cup \{\langle cn \rangle \, \varphi\}$
    end for
end while
```

can be done in with a little trick: map or interpret the non-spatial change as a topological one. For example, a spatial configuration of vessels might only be dangerous if the lighting conditions are bad, such as at night and at least one participant has its lights turned off. Let us say a light is `off` when its luminosity $L = 0$ and `on` if $L > 0$. This can be described as topological mode space of the light: $Q_{\text{light}} = \{\texttt{on}, \texttt{off}\}$, and the two domination relations as $\vdash_{\text{light}} = \{\langle \texttt{on}, \texttt{off} \rangle\}$, $\dashv_{\text{light}} = \{\langle \texttt{off}, \texttt{on} \rangle\}$.

This approach can be applied to any (countably finite) property and thus provides an extended view of topological modes space, which allows CNL to incorporate any non-spatial countably finite properties. The combination is granted by the product theorem for topological mode spaces Galton, 2000, page 359.

5.5.5 Extending the labeling algorithm

In (Blackburn and van Benthem, 2006) a straightforward labeling algorithm is described as a method for model checking. Given a formula ϕ, label every qualitative state in the model with all the sub-formulae of ϕ that are known to be true at that state. This is a bottom-up approach that starts by using the valuation V to label all states where a proposition symbol is true accordingly. Algorithm 1 extends the labeling algorithm by defining a procedure for (sub-)formulae of the following type: $\langle cn \rangle \, \varphi$.

5.5.6 A mapping into CTL*

To employ sophisticated model checkers that are freely available such as PRISM (Kwiatkowska, Norman, and Parker, 2011), we describe a translation from CNL to CTL*.

Theorem 2. *Every CNL formula can be mapped to an equivalent CTL* formula but not vice versa.*

Proof. The following is a mapping of CNL formulae into CTL*:

$$
\begin{aligned}
&\text{CNL} && \text{CTL}^\star \\
&\langle \mathrm{cn} \rangle\, \phi &\mapsto\;& \exists X \phi && \text{(5.3a)} \\
&[\mathrm{cn}]\, \phi &\mapsto\;& \forall X \phi && \text{(5.3b)} \\
&\circ\phi &\mapsto\;& X \phi && \text{(5.3c)} \\
&\phi_1 U \phi_2 &\mapsto\;& \phi_1 U \phi_2 && \text{(5.3d)}
\end{aligned}
$$

However as CNL does not have any quantifiers the following valid CTL* formula can not be expressed in CNL: $\exists \diamond \phi$. □

Corollary 1. *CNL can be expressed in neither LTL nor CTL.*

The corollary easily follows from the observation that CNL formula $\langle \mathrm{cn} \rangle\, \phi$ refers to a conceptually neighbored state with respect to some path π, but that state does not necessarily have to be on the path as would be required for LTL. Since CNL involves a next state operator $\circ$ not included in CTL, a mapping to CTL is not possible.

Since model checking with CTL* is decidable and the topological mode spaces only involves a finite set of states that are filtered by our qualitative spatial reasoning method, we immediately obtain the following corollary.

Corollary 2. *Model checking with CNL is decidable.*

5.6 Specifying safe navigation with CNL

Traffic in public spaces is widely governed by regulations, ranging from sweeping claims like "do not collide" to sophisticated right-of-way regulations. In this section we show how such regulations and general assumptions, e.g., about the braking system or the robot's field of view, can specified in CNL using partially grounded spatial constraints. Specification in CNL are declarative and involve cognitive concepts of space. Sound reasoning can then turn the simple do-not-collide-rule into the emergent behavior of slowing down before blind curves.

5.6.1 Formalizing navigation regulations

We employ a fixed formula pattern for representing typical navigation regulations given as natural language clauses. By constraining formula construction, readability of CNL formulae is eased. A real application would define a dedicated domain language built atop of CNL to support formalization.

Navigation regulations are formalized with respect to the context that triggers applicability of the regulation:

$$\phi_{\texttt{context}} \rightarrow \circ\big(\phi_{\texttt{actions}} \wedge (\phi_{\texttt{flags}} U \phi_{\texttt{end condition}})\big) \tag{5.4}$$

Context may be an indication of actions and optional flags. Flags allow rules to not only respond to actions but to extend over a period of time, e.g., to block certain rules while performing an action. To ensure that the agent behaves deterministically, we require that the context can be evaluated at the time the actions have to be performed. Therefore, we restrict *context* to state formula and do not allow any path quantifiers. Otherwise, context could refer to unknown future events, for example *after two steps in time, the sky is blue again*, which can generally not be evaluated due to its non-deterministic and unforeseeable nature: there might be a future in which the sky will be blue but there might be another week of rain to come.

Action formulae are either formulae indicating what the agents have to accomplish or they describe situations to be achieved next. Flags are conjunctions of atoms that hold until an end condition is achieved. They allow keeping track which regulations are currently activated. By referring to flags that mark activation of a rule within a context clause, a rule can temporally be disabled. For example, if A is overtaking B then a right of way rule that would cause B to yield to A can be disabled during the overtaking event.

5.6.2 Example navigation regulations for ground transportation vehicles

In the case studies of this paper we utilized two navigation rules:

1. the robot may not collide, either with other vehicles or with obstacles

2. unless right of way is granted, the robot needs to be able to come to a full stop in front of other vehicles or obstacle;
   right of way is granted by "left yields to right", i.e., a vehicle approaching from right has the right of way

To demonstrate the expressiveness of the presented approach, we also discuss the more involved regulation of an all-way intersection.

Disallowing collision states

The first rule essentially says that at no time polygons representing two entities (vehicles, robot, or obstacles) may overlap. Therefore the context is

$$\phi^1_{\texttt{context}} := \bigvee_{o \in \texttt{Objects}} \bigvee_{o' \in \texttt{Objects} \setminus \{o\}} \texttt{overlaps}(o, o').$$

In this context there are no admissible actions, as this context is a collision state and hence must not occur.

$$\phi^1_{\texttt{action}} := \bot.$$

Left yields to right

The second rule is modeled in a similar fashion. Based on the characteristics of each entity and each driving command class, a polygon describing an inevitable collision area is selected. Extending this area by a safety margin we end up with a safety area for each driving command. If two safety areas overlap, both vehicles have to brake in order to avoid a collision, except for cases resolved by the right of way regulation. Therefore the context for distinct moving entities m and m' is as follows:

$$\phi^2_{\texttt{context},m,m'} := \texttt{overlaps}(m_{\texttt{safety area}}, m'_{\texttt{safety area}}) \wedge \neg\texttt{leftOf}(m, m')$$

with the action

$$\phi^2_{\texttt{action},m,m'} := \texttt{stop}(m).$$

The rule is instantiated for any pair of vehicles.

First-come-first-serve intersection

At an all-way stop intersection, every vehicle has to stop. The vehicle arriving there fist has the right of way.[4] The following sub-formula defines the *stop-at* behavior, where mobile entity m has to stop at an area A (the stop line):

$$\phi_{\text{stop},m,A} := \texttt{disconnected}(m, A) U \big[\texttt{overlapping}(m_{\text{safety area}}, A) \wedge \texttt{stop}(m)\big].$$

The statement saying entity m' reaches the intersection I prior to m implies m has to wait until m' cleared the intersection area can be written as

$$\begin{aligned} &\overbrace{\neg\texttt{disconnected}(m'_{\text{safety area}}, I) \wedge \texttt{disconnected}(m_{\text{safety area}}, I)}^{m' \text{ comes first}} \\ &\qquad \rightarrow \circ\overbrace{\big(\phi_{\text{stop at},m,I} U \texttt{disconnected}(m'_{\text{safety area}}, I)\big)}^{m \text{ waits until free}}. \end{aligned} \tag{5.5}$$

Just as with *left yields to right* the rule is instantiated for any pair of vehicles. Consequently, in the case that multiple entities are approaching the intersection this formula will also ensure the correct order.

Example 1. A vehicle a reaches the intersection before vehicles b and c, the last to approach the intersection is c. Formula 5.5 entails a model that involves at least three distinct time points $t < t' < t''$. For readability the index *safety area* in $m_{\text{safety area}}$ is omitted in Figure 5.8.

At time point t a has reached the intersection and both b, c have to yield to a, at t' b reaches the intersection and stops. When at time point t'' a leaves the intersection, b continues its travel, while c still has to wait.

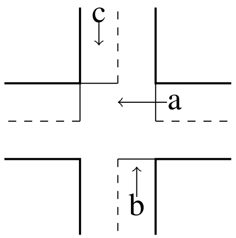
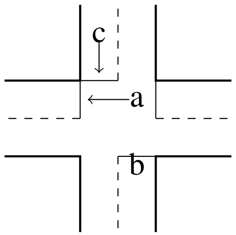
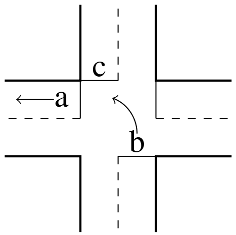
	t	**t′**	**t″**
Scene			
Description	$\neg\texttt{disconnected}(a, I)$ $\texttt{disconnected}(b, I)$ $\texttt{disconnected}(c, I)$	$\neg\texttt{disconnected}(a, I)$ $\neg\texttt{disconnected}(b, I)$ $\texttt{disconnected}(c, I)$	$\texttt{disconnected}(a, I)$ $\neg\texttt{disconnected}(b, I)$ $\neg\texttt{disconnected}(c, I)$
Active Rules	$\phi_{\text{stop},\mathbf{b},I} U \texttt{disconnected}(a, I)$ $\phi_{\text{stop},\mathbf{c},I} U \texttt{disconnected}(a, I)$	$\phi_{\text{stop},\mathbf{b},I} U \texttt{disconnected}(a, I)$ $\phi_{\text{stop},\mathbf{c},I} U \texttt{disconnected}(a, I)$ $\phi_{\text{stop},\mathbf{c},I} U \texttt{disconnected}(b, I)$	$\phi_{\text{stop},\mathbf{c},I} U \texttt{disconnected}(b, I)$

Figure 5.8: Pictorial representation of key time points for the First-Come First-Serve example.

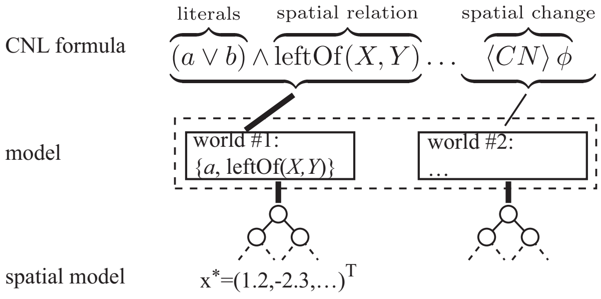


Figure 5.9: Models for CNL formulae, the use of conceptual neighborhood allows separated computations of the spatial model.

5.6.3 Reasoning for safe navigation

If every vehicle moves according to the rules, then we say that a *drive command* is safe if an evasive maneuver can still be executed *directly* after the command was issued any collision can be avoided. Such an evasive maneuver is in the case of passive safety: braking.

$$\Box\phi_{\texttt{rules}} \wedge \phi_{\texttt{drive command}} \wedge \circ\Big(\overbrace{\big(\phi_{\texttt{evasive maneuver}} \wedge [cn]\,\neg\phi_{\texttt{collision}}\big)}^{\texttt{evasive maneuver path is clear}} U \phi_{\texttt{stopped}}\Big) \tag{5.6}$$

By employing answer set programming (ASP) to generate hypothetical situations that would violate the above statement we avoid generating the full topological mode induced by this formalization. The resulting (partial) models are than verified to correspond to paths through the actual topological mode space by means of creating the required partial topological mode space. This approach is more efficient than generating the full topological mode space. Verification of partial models involves verifying that all modes can be realized spatially, given the partial grounding as described in Section 5.4.2.

By determining a realization of the above formula, we obtain a polygon that represents the maximum area covered by the path of a vehicle. This polygon describes the inevitable collision area (ICA) and the polygon used for this paper are shown in Figure 5.10.

5.7 Case studies

In this section we present case studies of using CNL to guarantee safe navigation. The two use cases we consider are, first, to determine a-priori heat/cost maps of motion restrictions for a known floor map. Second, we consider the monitoring task where the system is applied to sensor data.

[4] For simplicity we omit the special case of two participants arriving at the same time which is according to left yields to right.

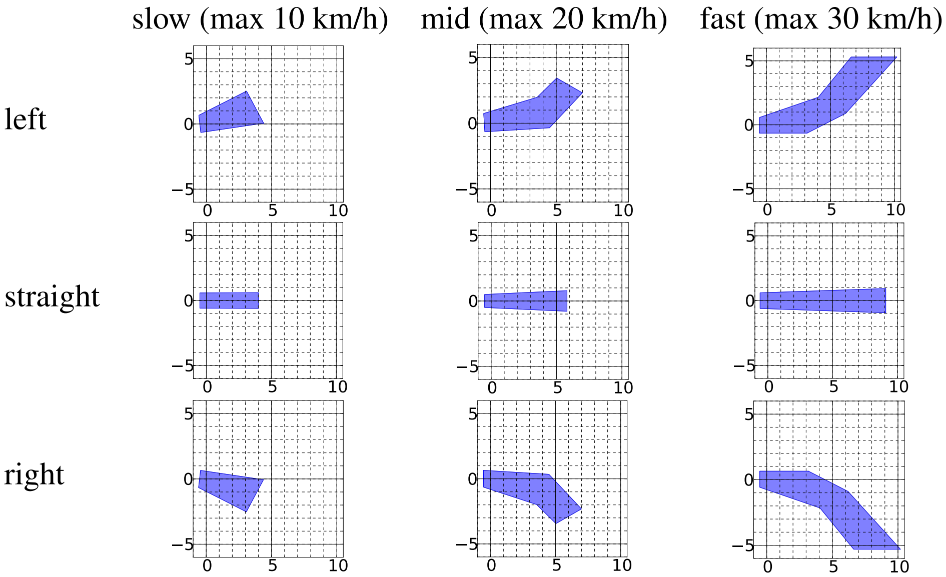


Figure 5.10: Safety polygons (inevitable collision area plus safety margin) for the driving commands used in the examples. This polygons are derived from simple simulation runs.

5.7.1 System implementation

We implemented And-Or LP trees in Python. The resulting LP problems were solved by calling the external LP solver *lp_solve* (Berkelaar, Eikland, and Notebaert, 2010). CNL formula have been manually mapped to ASP programs for the hypothesis generation. Each generated hypothesis is a formula in conjunctive from with only positive terms. These formulas are than reduced as described in Section 5.4.7, utilizing SparQ (Wolter and Wallgrün, 2010) for the pure qualitative spatial reasoning, such as pruning of the RCC formulas. All experiments were run on a Intel Xeon® CPU® X5690 3,47GHz with 148 GB ram, but no more than 1 GB of ram was allocated.

5.7.2 Identifying restrictions to avoid inevitable collision states

In order to avoid dangerous situations robot motion planning needs to anticipate other agents that may suddenly become visible. Robots should thus to slow down at places with poor visibility. As a result, the shortest route determined from a map may not yield the quickest route. A possibility to acknowledge motion limitation (in particular speed limits) is to augment robot maps, Chung et al. (2009) present heat maps that represent speed limitations determined from limited visibility as grey-scale images. These speed limitations are imperative in the sense that violating them may lead to entering an ICS. Using the logic framework CNL provides an automated and sound way of computing these heat maps. In our case study we consider the navigation rules presented in the previous section.

- Do not enter an ICS, i.e., being able to come to a full stop before collision with a (moving) object occurs
- Left yields to right

These rules are compiled into an CNL formula Θ.

For robot motion we consider

- Kinematics of a front-wheel steered vehicle for robot and other vehicles
- Motion actions turning left, right, and driving forward with 3 speed classes each as well as a special slow motion forward/turn action.

The ICAs are determined for each vehicle and each motion action, see Section 5.6.3. For the environment we choose a 4-way intersection given as polygonal obstacle outlines. While polygonal maps offer a more compact representation than grid maps commonly used in robotics, compact polygonal maps can be extracted from grid maps, e.g., Veeck and Burgard (2004).

For brevity of presentation, we only consider heat maps representing speed limitations as opposed to limitations on turning actions. We discretize robot position to 20cm resolution, maximum speed into four classes, and had three different directional movements (forward, left, right); robot heading is discretized into four cardinal directions. The choice of discretization is arbitrary but tailored to easy depiction, any other discretization would be possible.

In order to determine a heat map we apply model checking with the CNL formula Θ and the polygonal map. For each position in the map (x, y), heading of the robot ϕ and candidate for maximum speed v, the model checker tries to construct an ICS, i.e., to supplement the partial grounding given to construct a geometric situation in which a collision will occur. The result of this procedure is shown in Figure 5.11 (a). To the left, the four distinct heat maps for the cardinal directions are shown. It can be observed that the maximum speed is decreased close to obstacles and close the intersection if another vehicle with right of way can be hidden by occlusion. Figure 5.11 (b) shows a magnification of the interesting part of the northbound map in front of the intersection. As can be observed, the robot needs to reduce speed advancing the intersection but it can continue at full speed once it gains a view into the right-handed submission. In Figure 5.11 (c) we present one of the ICS models generated by our system from that intersection. As can be seen the robot position has been grounded close to the intersection, the braking polygon (blue) intersecting with the polygonal area of another vehicle approaching from the left driving a curve to the left. As from the position of the robot this vehicle would not be visible, a slower speed (and smaller braking polygon) is required.

5.7.3 Motion monitoring

We consider an industrial setting in which a robot travels 1) a factory building, 2) exploring an outside environment. The task of motion monitoring is to make sure that the robot does not move too fast to avoid collisions. The assumption underlying the monitoring approach is

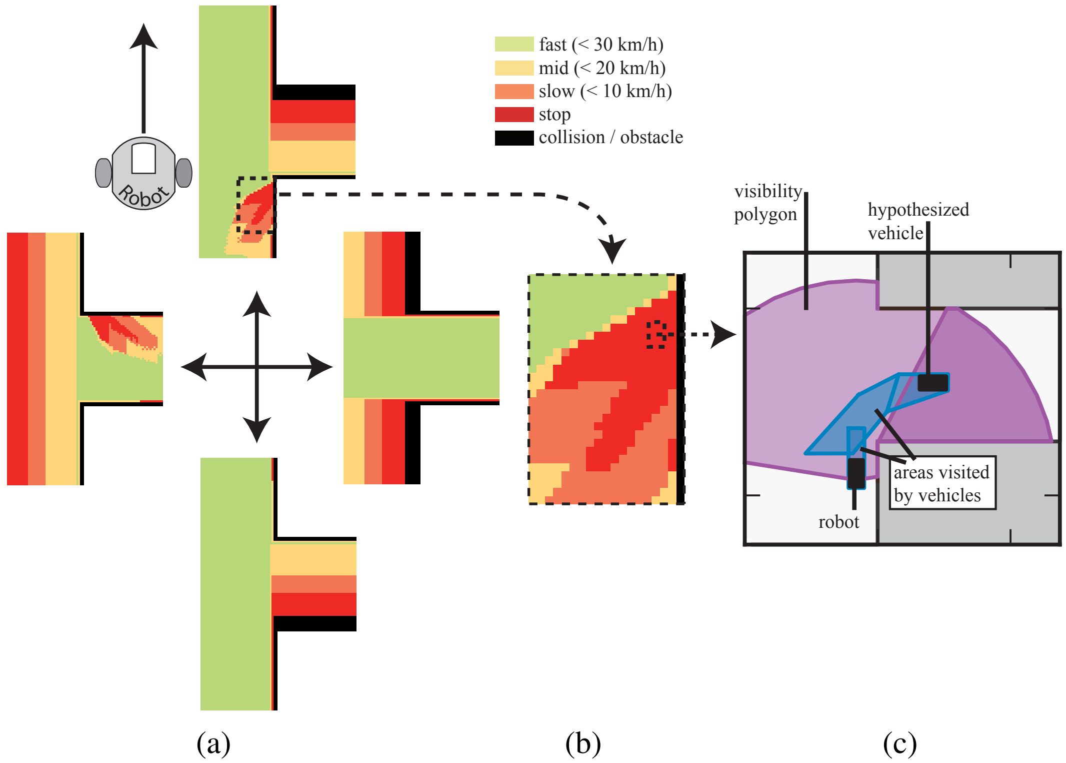


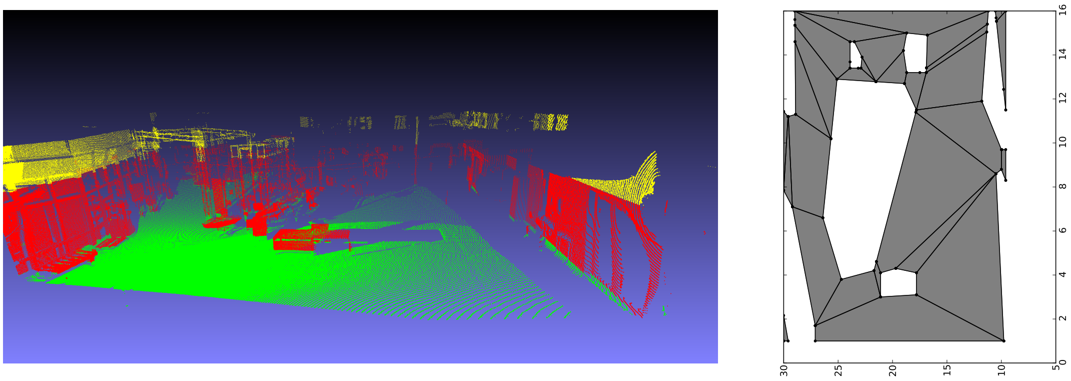
Figure 5.11: Map of speed limitations at a T-intersection Heatmap, considering rule *left yield to right*. (a) Speed limitations differentiated by four cardinal directions. (b) Magnification from map for north direction. (c) Model constructed for too high speed: the vehicle approaching the intersection from the left is hypothesised by the spatial model checking system proposed. Accordingly, a lower speed is required.

that no obstacles are missed by the robot sensors. All robots in the experiments have been equipped with laser range finders suited for safety-critical applications and we directly process data obtained from the range finders.

Navigating inside a factory building

In Figure 5.12 (a) a 3D laser scan is shown, color-coded into traversable ground plane (green, height less than 10cm), obstacles out of reach (yellow, height more than 250cm above ground), and obstacles (red). Obstacles are projected into the plane and discretized to a 10cm grid. Then contour extraction by Pavlidis' method Pavlidis, 1982, Chapter 7, Section 5 is performed. The resulting ground plane and its partition into convex parts is shown in Figure 5.12 (b).

Similar to the previous case study we compute heat maps for five directions of movement. Doing so we also obtain several runs of the reasoning system to consider computation time. The resulting maps are shown in Figure 5.13, a histogram of computation time is shown in



(a)

(b)

Figure 5.12: (a) Point cloud obtained by 3d laser range scanner in industrial setting, color coded as ground floor (green), obstacles (red), and objects above the robot (yellow). (b) Polygonal map and convex partition of the ground plane

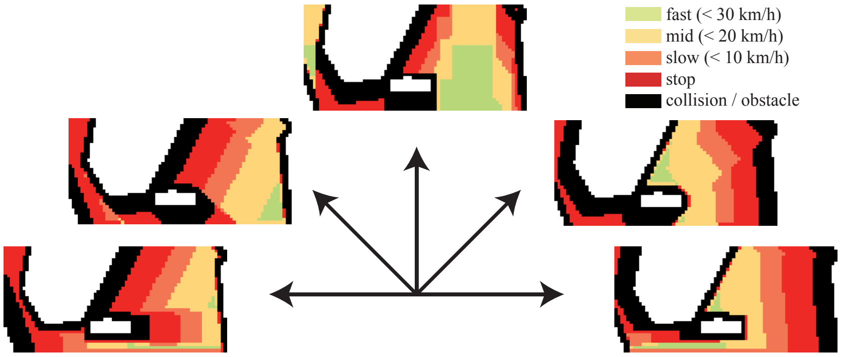


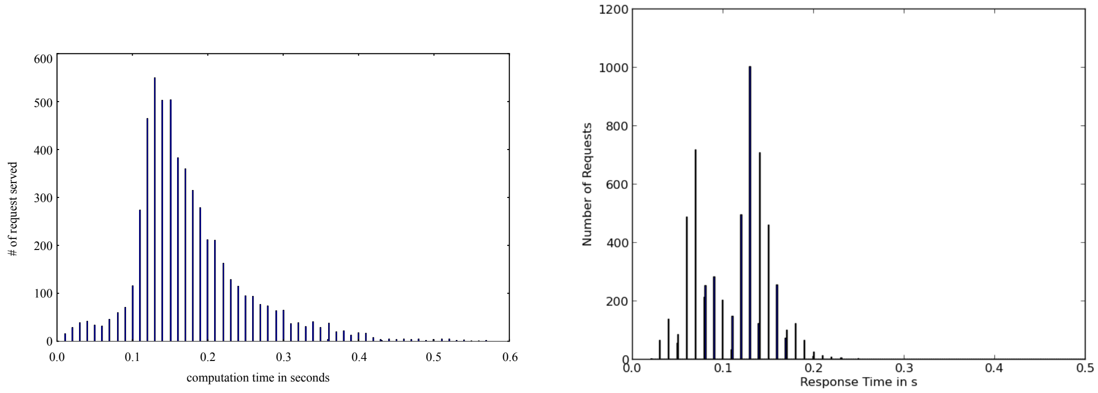
Figure 5.13: Speed limitations for safely travelling in an industrial setting

Figure 5.14(a). The histogram depicts the computation time for testing a single hypothesis, i.e., trying to construct an ICS state from the given robot's position, direction, and speed.

Outdoor exploration

We apply monitoring to a robot exploring an outdoor campus area. Our aim is to study the compute time necessary to handle real-world observations in the same way as with the industrial setting example, i.e., to verify that the robot always maintains clearance for stopping in front of static and moving obstacles. Right-of-way rule "left yields to right" allows the robot to pass by vehicles approaching from the left, assuming they respect the rule and stop.

Evaluation is based on a log file recording the raw sensor data of the robot, the data set *fr_campus* is published by Cyrill Stachniss and Giorgio Grisetti as part of the Robotics Data



(a) Moving inside a Factory Building

(b) Outdoor Exploration

Figure 5.14: Histogram of computation time for single hypotheses

Set Repository (Radish) (Howard and Roy, 2003). A Pioneer2AT robot equipped with one SICK LMS291-S05 laser range finder travels around the campus environment of University of Freiburg, Germany shown in Figure 5.15. The robot travels a distance of 1.750km, collecting a total of 15.654 laser scans.

From the laser scan we extracted lines using *Split-and-Merge* with the following parameters: inlier threshold of 5 cm, minimum number of points 4, maximum range of 50 m. Each line segment is than transformed into a rectangle with width equal to 2×inlier threshold. The line extraction required less than 100 ms.

At each time point we checked if we could move straight forward at maximum speed and recorded the computation time. A histogram of the computation time is shown in Figure 5.14(b), this does not include the 100 ms for preprocessing.

Except for very few outliers ($< 0.01\%$) the computation time was well below 300 ms. Consequently including the preprocessing time a conservative estimate is 400 ms from laser scan to result.

5.7.4 Discussion

With respect to the results obtained it can be observed that CNL-based model checking is able to reproduce results obtained by Chung et al. (2009) (see, e.g. Figure 5 in that article) using a manually derived algorithm. By contrast, the method presented here is based on a declarative input that specifies navigation regulation. This has two major implications. First, we are able to tackle several application domains that involve a different set of navigation rules. Second, adaption of navigation rules is possible online, in particular to change safety regions of a robot in response to robot configurations, for example if a robot carries large goods. Beyond the results previously obtained, the CNL-based approach presented here allows for specification of rules in way expressive enough to handle common traffic rules such as “left yields to right”.

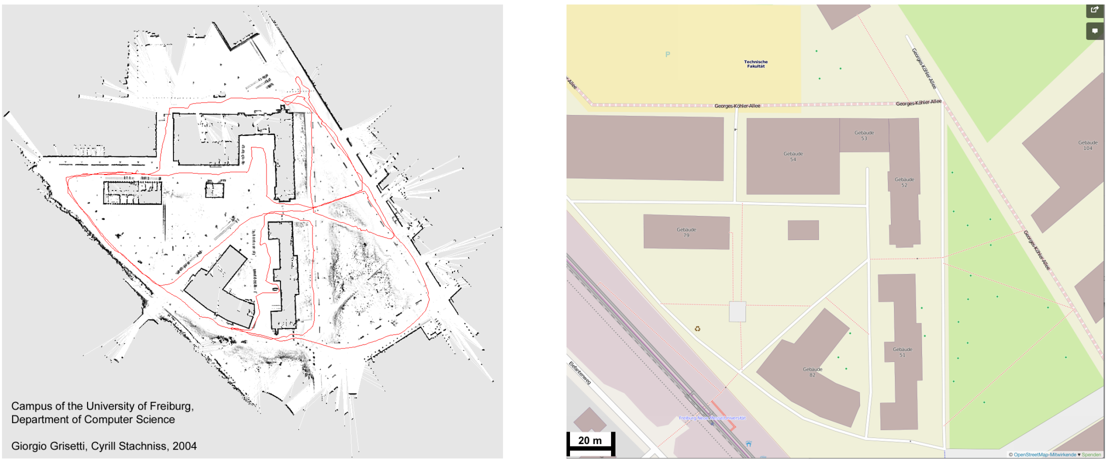

Campus of the University of Freiburg,
Department of Computer Science
Giorgio Grisetti, Cyrill Stachniss, 2004
Technische Fakultät
Georges-Köhler-Allee
20 m

Figure 5.15: Left: A map showing the explored fr_campus environment created with Grisetti's Rao-Blackwellized SLAM system; the image is provided as part of the data set. Right: OpenStreetMap image showing the same area for comparison.

With respect to computation time we can observe that our prototypical implementation can verify a motion command within typically less than 0.4 seconds. As motion commands are often issued every 0.1 seconds, it would require a more sophisticated implementation to meet real-time constraints. Several options for improving the implementation however present themselves, ranging from technical aspects (e.g., changing from interpreted Python to compiled code) to algorithmic ones (e.g., parallelized model checking). In the light of these options we are confident that the proposed system is suitable for real-time robotic application.

5.7.5 Further applications of CNL

CNL can also be applied in other tasks related to safe navigation. Kreutzmann et al. (2011) show how process recognition in robotics can be approached by model checking. They represent processes similar to how navigation is specified here, but utilizing LTL and not considering complex spatial constraints. This approach, applied to safe navigation, can be used to check whether other vehicles adhere to navigation rules and, if not, enforce larger safety zones.

Regulation-compliant planning

Heat maps only provide a coarse view on safety-related constraints. A tighter coupling between the safety component and path planning can be achieved by bringing the monitoring system into the loop of the planning module. For example, connecting a randomized roadmap planner with the monitoring approach described here allows a regulation-compliant plans to be determined. The monitoring component is used to reject any steps in the planner that do not agree with the regulations. This extends and earlier approach that is based on hand-crafted spatial constraints (Wolter, Dylla, and Kreutzmann, 2011).

CNL based reasoning aims at determining models for a hypothetical ICS within the specifications of the navigation capabilities. Such model can serve as a counter-example to prove a programmer wrong who designed the motion controller. These examples can thus help the programmer to reveal problems in the either a too conservative safety specification or too risky motion control.

5.8 Summary and conclusion

This paper tackles safe navigation from an AI perspective of knowledge representation and symbolic reasoning. We develop a sound reasoning mechanism for a logic representation of navigation rules and safety requirements that provides an automated answer to the question whether an action can safely be executed. Two assumptions are necessary in order to commit to safety guarantees. First, robot perception needs to be reliable, for example by using certified sensors. Second, rule compliancy of other traffic participants needs to be assumed to some degree as otherwise hardly any action could be regarded safe at all (consider fast moving vehicles that would actually aim to collide with the robot).

The first contribution of this paper is a new approach to qualitative spatial reasoning using And-Or LP trees. The domain of this representation consists of points, lines, and polygons of known shape in the Euclidean plane as well as a finite set of orientations. The approach allows all spatial relations to be represented that can be described as finite disjunctions of linear relations. This includes topology relations between polygons as well as directional knowledge. A key feature of And-Or LP trees is however their seamless integration with grounded knowledge in the sense that some variables in the abstract representation correspond to specific real-world entities described by numerical values. The advantage of employing qualitative spatial representations is twofold. On the one hand, they can resemble cognitive spatial concepts (Klippel and Montello, 2007; Knauff, Rauh, and Renz, 1997) and thus lead to an intuitive knowledge representation language. On the other hand, qualitative spatial representations form a simple symbolic language that can be integrated with logics in a way that grants decidable decision procedures.

As second contribution we propose the conceptual neighborhood logic CNL, a new modal logic that incorporates qualitative spatial representations, qualitative notion of change, and a temporal logic in a joint framework. By giving a translation that maps CNL formulae to CTL*, sophisticated tools already available for CTL*can also be applied to CNL. This also positions CNL in context of logic approaches to software verification. From an AI perspective, we contribute to answering the question of how qualitative representations can be effectively combined with general logics.

In a case study we give formalizations of navigation rules previously considered in robotics. By using CNL-based reasoning results of existing work can be reproduced in a fully automated procedure. CNL is expressive enough to represent the spatial and temporal knowledge involved in typical navigation regulations for road traffic. The declarative and fully automated system allows the approach to be easily adapted to various and possibly changing requirements, even at runtime.

We consider two ways of exploiting CNL with a robotic system. First, speed limitations can be computed a priori given a map of the environment. Second, commands issued by a motion controller can be verified to be compliant with navigation rules. Runtime analysis demonstrates that our approach meets requirements of online applications in the control loop of a robotic system.

With respect to computation time, the limiting factor in formal reasoning is the amount of spatio-temporal configurations required to explore in order to verify that no rule will be violated. The qualitative representation underlying CNL partitions the infinite variability of spatio-temporal configurations into a finite set of conceptually equivalent situations. This abstraction is crucial to achieve decidability and efficiency in reasoning, but identifying the most suitable level of abstraction remains an open question. Too coarse relations cannot represent navigation rules adequately, whereas too fine relations impede online application. In our approach we opt for an upper approximation of safety zones that ensures that no unsafe action can be considered safe, but this may lead to over-cautious decisions. The most suitable representation clearly depends on application context and requires extensive empirical evaluation. Independent of the spatial representations chosen, a second impact factor on runtime is the necessary size of a model to witness a rule violation. Although the model size is a fixed constant for any set of navigation rules, it grows exponentially with the amount of time steps and entities involved. Model checking in CNL has the same worst time complexity class as that of CTL*, namely PSPACE. Rules in road traffic seldom involve more than two objects, though.

From the perspective of robotics, CNL-based reasoning significantly improves techniques for realizing safe navigation by guaranteeing rule compliancy—an important step towards shared human-robot environments. Since regulation specifications denoted in CNL involve cognitive concepts of space, they are verifiable by persons that are knowledgeable of the regulations, not only to experienced robot engineers.

In future work, we will focus on tightening the integration of reasoning steps to obtain more efficient algorithms. One particular aim is to achieve better real-time capabilities. We will further research means for debugging support on basis of CNL annotation of navigation algorithms, for example by generating pictorial counter-examples of the spatial situation that illustrates how an assumption made in the code can be violated.

References

Aiello, Marco, Ian E. Pratt-Hartmann, and Johan van Benthem, eds. (2007a). *Handbook of Spatial Logics*. Springer. ISBN: 978-1-4020-5586-7.

Aiello, Marco, Ian E. Pratt-Hartmann, and Johan van Benthem (2007b). "What is Spatial Logic?" In: *Handbook of Spatial Logics*. Ed. by Marco Aiello, Ian E. Pratt-Hartmann, and Johan van Benthem. Springer, pp. 1–11. ISBN: 978-1-4020-5586-7.

Bäckström, Christer and Peter Jonsson (2012). "Abstracting Abstraction in Search with Applications to Planning". In: *Proc. of KR 2012*. Ed. by Gerhard Brewka, Thomas Eiter, and Sheila A. McIlraith. AAAI Press. ISBN: 978-1-57735-560-1.

Berkelaar, Michel, Kjell Eikland, and Peter Notebaert (May 2010). *lp_solve 5.5, Open source (Mixed-Integer) Linear Programming system*. Software. Available at `http://lpsolve.sourceforge.net/5.5/`. Last accessed April, 2013.

Bhatt, Mehul, Wenny Rahayu, and Gerald Sterling (2006). "Qualitative Spatial Reasoning with Topological Relations in the Situation Calculus". In: *In Proc. of (FLAIRS 06)*. AAAI Press.

Blackburn, Patrick and Johan van Benthem (2006). "Modal logic: a Semantic Perspective". In: *Handbook of Modal Logic*. Ed. by Patrick Blackburn, Johan van Benthem, and Frank Wolter. Elsevier, pp. 1–82. ISBN: 0-444-51690-5.

Bouraine, Sara, Thierry Fraichard, and Hassen Salhi (2012). "Provably safe navigation for mobile robots with limited field-of-views in dynamic environments". In: *Autonomous Robots* 32.3, pp. 267–283.

CGAL, *Computational Geometry Algorithms Library* (n.d.). Available at `http://www.cgal.org`.

Chung, Woojin, Seokgyu Kim, Minki Choi, Jaesik Choi, Hoyeon Kim, Chang-bae Moon, and Jae-Bok Song (2009). "Safe Navigation of a Mobile Robot Considering Visibility of Environment". In: *IEEE Transactions on Industrial Electronics* 56.10, pp. 3941–3950.

Clarke, Edmund M. and E. Allen Emerson (1982). "Design and Synthesis of Synchronization Skeletons Using Branching-Time Temporal Logic". In: *Logic of Programs, Workshop*. London, UK, UK: Springer-Verlag, pp. 52–71. ISBN: 3-540-11212-X.

Levesque, Hector J., Raymond Reiter, Yves Lespérance, Fangzhen Lin, and Richard B. Scherl (1997). "GOLOG: A Logic Programming Language for Dynamic Domains". In: *Journal of Logic Programming* 31, pp. 59–84.

Cohn, Anthony G. and Shyamanta M. Hazarika (2001). "Qualitative Spatial Representation and Reasoning: An Overview". In: *Fundamenta Informaticae* 46.1-2, pp. 1–29.

Cohn, Anthony G. and Jochen Renz (2007). "Qualitative Spatial Representation and Reasoning". In: *Handbook of Knowledge Representation*. Ed. by Frank van Harmelen, Vladimir Lifschitz, and Bruce Porter. Elsevier, pp. 551–596.

Emerson, E. Allen and Joseph Y. Halpern (Jan. 1986). ""Sometimes" and "Not Never" Revisited: On Branching Versus Linear Time Temporal Logic". In: *J. ACM* 33.1, pp. 151–178. ISSN: 0004-5411.

Freksa, Christian (1991a). "Conceptual neighborhood and its role in temporal and spatial reasoning". In: *Proceedings of the IMACS Workshop on Decision Support Systems and Qualitative Reasoning*. Ed. by Madan G. Singh and Luise Travé-Massuyès. Amsterdam: Elsevier, pp. 181–187.

Freksa, Christian (1991b). "Qualitative Spatial Reasoning". In: *Cognitive and Linguistic Aspects of Geographic Space*. Ed. by David M. Mark and Andrew U. Frank. Dordrecht: Kluwer, pp. 361–372.

Galton, Antony (1995). "Towards a Qualitative Theory of Movement". In: *Proc. of COSIT '95*, pp. 377–396.

Galton, Antony (2000). *Qualitative Spatial Change*. Spatial Information Systems. Oxford University Press. ISBN: 978-0-198-23397-8.

Hertel, Stefan and Kurt Mehlhorn (1983). "Fast triangulation of simple polygons". In: *In Proc. 4th Internat. Conf. Found. Comput. Theory*. Vol. 158. Lecture Notes Comput. Sci. Springer-Verlag, pp. 207–218.

Howard, Andrew and Nicholas Roy (2003). *The Robotics Data Set Repository (Radish)*. Available at `http://radish.sourceforge.net/`.

Klippel, Alexander and Dan Montello (2007). "Linguistic and nonlinguistic turn direction concepts". In: *Proceedings of Conference of Spatial Information Theory (COSIT)*. Vol. 4736. Springer, pp. 354–372.

Knauff, Markus, Reinhold Rauh, and Jochen Renz (1997). "A cognitive assessment of topological spatial relations: Results from an empirical investigation". In: *Spatial Information Theory: A theoretical basis for GIS*. Proceedings of COSIT. Springer, pp. 193–206.

Kontchakov, Roman, Agi Kurucz, Frank Wolter, and Michael Zakharyaschev (2007). "Spatial Logic + Temporal Logic = ?" In: *Handbook of Spatial Logics*. Ed. by Marco Aiello, Ian E. Pratt-Hartmann, and Johan van Benthem. Springer, pp. 497–564. ISBN: 978-1-4020-5586-7.

Kreutzmann, Arne, Immo Colonius, Lutz Frommberger, Frank Dylla, Christian Freksa, and Diedrich Wolter (Sept. 2011). "On Process Recognition by Logical Inference". In: *Proc. of the 5th European Conference on Mobile Robots (ECMR)*, pp. 7–12.

Kwiatkowska, Marta, Gethin Norman, and David Parker (2011). "PRISM 4.0: Verification of Probabilistic Real-time Systems". In: *Proc. of 23rd International Conference on Computer Aided Verification (CAV'11)*. Ed. by Ganesh Gopalakrishnan and Shaz Qadeer. Vol. 6806. LNCS. Springer, pp. 585–591.

Lee, Jae Hee, Jochen Renz, and Diedrich Wolter (2013). "StarVars—Effective Reasoning about Relative Directions". In: *IJCAI*. Ed. by Francesca Rossi. IJCAI.

Li, Sanjiang, Weiming Liu, and Shengsheng Wang (2013). "Qualitative constraint satisfaction problems: An extended framework with landmarks". In: *Artificial Intelligence* 201, pp. 32–58.

Lifschitz, Vladimir (2008). "What Is Answer Set Programming?" In: *AAAI*. Ed. by Dieter Fox and Carla P. Gomes. AAAI Press, pp. 1594–1597. ISBN: 978-1-57735-368-3.

Liu, Weiming, Sanjiang Li, and Jochen Renz (2009). "Combining RCC-8 with Qualitative Direction Calculi: Algorithms and Complexity". In: *Proceedings of the Twenty-first International Joint Conference on Artificial Intelligence (IJCAI'09)*. Ed. by Craig Boutilier. AAAI Press, pp. 854–859.

McCarthy, John (1963). *Situations, Actions, and Causal Laws*. Tech. rep. Memo 2. Stanford University: Stanford Artificial Intelligence Project.

Pavlidis, Theo (1982). *Algorithms for Graphics and Image Processing*. Rockville, Maryland, USA: Computer Science Press.

Pnueli, Amir (1977). "The temporal logic of programs". In: *Proceedings of the 18th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 46–57.

Randell, David A., Zhan Cui, and Anthony G. Cohn (1992). “A Spatial Logic based on Regions and “Connection””. In: *Proc, of KR’92*. Ed. by Bernhard Nebel, Charles Rich, and William Swartout. San Mateo (CA), USA: Morgan Kaufmann, pp. 165–176.

Renz, Jochen and D. Mitra (2004). “Qualitative Direction Calculi with Arbitrary Granularity”. In: *Proc. of PRICAI-04*. Vol. 3157. LNAI. Springer, pp. 65–74.

Renz, Jochen and Bernhard Nebel (2007). “Qualitative spatial reasoning using constraint calculi”. In: *Handbook of Spatial Logics*. Ed. by Marco Aiello, Ian E. Pratt-Hartmann, and Johan van Benthem. Springer, pp. 161–215. ISBN: 978-1-4020-5586-7.

Schrijver, Alexander (1986). *Theory of linear and integer programming*. New York, NY, USA: John Wiley & Sons, Inc. ISBN: 0-471-90854-1.

Täubig, Holger, Udo Frese, Christoph Hertzberg, Christoph Lüth, Stefan Mohr, Elena Vorobev, and Dennis Walter (2012). “Guaranteeing functional safety: design for provability and computer-aided verification”. In: *Autonomous Robots* 32.3, pp. 303–331.

Veeck, Michael and Wolfram Burgard (2004). “Learning Polyline Maps from Range Scan Data Acquired with Mobile Robots”. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

Wolter, Diedrich, Frank Dylla, and Arne Kreutzmann (2011). “Rule-Compliant Navigation With Qualitative Spatial Reasoning”. In: *Proceedings of the 4th Intl. Robotic Sailing Conference*. Springer.

Wolter, Diedrich and Jae Hee Lee (2010). “Qualitative Reasoning with Directional Relations”. In: *Artif. Intell.* 174.18, pp. 1498–1507.

Wolter, Diedrich and Jan Oliver Wallgrün (2010). “Qualitative Spatial Reasoning for Applications: New Challenges and the SparQ Toolbox”. In: *Qualitative Spatio-Temporal Representation and Reasoning: Trends and Future Directions*. Ed. by Shyamanta M. Hazarika. Hershey (PA), USA: IGI Global.

6 A Qualitative Representation of Social Conventions for Application in Robotics

Frank Dylla[1], Diedrich Wolter[2], and **Arne Kreutzmann**[1]

1 Cognitive Systems Group, University of Bremen, Bremen, Germany

2 Smart Environments, University of Bamberg, Bamberg, Germany

Published in“2014 AAAI Spring Symposium - Qualitative Representations for Robots”, 2014, AAAI Publications, 2014 AAAI Spring Symposium Series.

Contributions:

Based on my work on multi-calculi reasoning and how to overcome the shortcomings of single calculi reasoning in QSR, Frank Dylla conducted this study. Frank Dylla did the literature research on social conventions as well as their modeling in QLTL. QLTL was jointly developed by me and Diedrich Wolter. Further, I did the few required implementation tasks.

Acknowledgements:

Financial support by the Deutsche Forschungsgemeinschaft in the Transregional Collaborative Research Center SFB/TR 8 Spatial Cognition project R3-[Q-Shape] is gratefully acknowledged.

Abstract

Acceptance of everyday robots will largely depend on their ability of social interaction. Patterns of socially acceptable behavior have been characterized in social science by means of abstract concepts of space and time. This makes integration with robot navigation a challenging problem. Moreover, an integration should allow the robot to build awareness of these patterns as in reality there will be misunderstandings a robot should be able to respond to. The contribution of this paper is a representation of navigation patterns that is based on qualitative representations of space. We present a logic with clear semantics for specification of navigation patterns and show how it allows several conventions of social interactions to be represented. We also outline integration with robot navigation and give examples with real world situations.

6.1 Introduction

Robots becoming part of our everyday life is a vision that is widely shared among researchers in the robotics community. Aside from technical challenges in design and development of such robots, we also need to acknowledge that such a system would be part of our human society. Acceptance of robots is henceforth also depending on their social acceptance. How does the robot interact with humans?

Social interaction goes beyond goal-directed human-robot interaction, it also comprises casual or emergent interaction, for example, how a robot traverses a populated area. Shall it take advantage of its advanced motion and path-planning capabilities to rush through even the most crowded places?

We are motivated by the assumption that acceptability of robots will depend on acknowledging established patterns of social interaction. Lindner and Eschenbach (2011)claim that people would feel offended if a robot cuts path in front of them. Other patterns may be more imperative. For example, humans recognize and acknowledge that others have queued up in a waiting area and they line up as well, not trying to overtake and press forward if space permits. Programmers realizing a robot's navigation component need thus to include patterns of socially acceptable behavior. In recent years several robots have been presented that are capable of interacting with individuals or groups of people, putting forward the prominent example of the museum tour guide robot RHINO (Burgard et al., 1999). Systems were designed to pass people (Pacchierotti, Christensen, and Jensfelt, 2005) or approaching groups of people (Althaus et al., 2004) in an adequate manner. In (Shi et al., 2008) the change of velocity near other pedestrians is considered, whereas in the Companion framework more general joint human-robot navigation is considered (Kirby, 2010).

Taking a closer look at these existing social robot systems, they all have in common a behavior that is directly based on quantitative parameters tailored to specific tasks, i.e., they are not *aware* of their spatial context but their actions are hardwired to the control system. By contrast, social behavior as described in the social sciences involves abstract concepts of space and time that vary across social status, culture, etc. It requires observation and interpretation to align oneself with the current context. Not until awareness of conventions is tackled, a robot will be able to recognize that it is interfering with human activity unintentionally. Viewed from the perspective of intelligent agent design, abstract declarative representations of social conventions could offer appropriate means for such reasoning. Linking such abstract representations to robot control is a difficult endeavor though. With our work we aim to bridge the gap between abstract descriptions of spatio-temporal patterns and robot navigation techniques. The contribution of this paper is to describe how patterns of socially acceptable behavior can be represented using qualitative concepts of space and time. The representation obtained can then provide the basis on which a robot can achieve awareness of social context.

We first give a system overview of how the approach integrates with a robotic software architecture. Section 6.3 reviews the concept of spaces (proxemics) and navigation conventions which are assumed to be socially acceptable. Section 6.4 shows how spatial knowledge present in the navigation conventions can be captured with qualitative representations. Using these spatial primitives we construct the new logic QLTL (Linear Temporal Logic with Qualitative Spatial Primitives), discuss required reasoning techniques and their realization. Section 6.5 presents a modeling of the socially acceptable navigation behaviors presented earlier and how they are processed in Section 6.6. Finally, Section 6.7 outlines handling of real-world situations on a robotic platform.

6.2 Approach

On the technical level, we develop a new representation language that combines the temporal logic LTL (Pnueli, 1977) with qualitative spatial calculi (see Section 6.4). Based on our representation language, we formalize several conventions of socially acceptable navigation. We discuss how the new logic can easily be integrated with existing navigation components to ensure socially acceptable navigation behavior.

For conceiving system integration we adapt the approach of a layered architecture as used successfully in many robotic systems, e.g. for museum tour guide robots (Burgard et al., 1999; Thrun et al., 1999). In the most general interpretation it consists of three layers (cp. Figure 6.1): 1) the low-level hardware control layer for any kind of sensor or actuator (e.g. wheels or camera), 2) the intermediate navigation and (pre-)processing layer (e.g. path planning and feature detection) and 3) the high-level reasoning and learning layer (e.g. Golog as in case of (Burgard et al., 1999)). The main organizational criteria is that on the lowest level data is completely quantitative and on the highest level abstract symbolic representations are predominant.

We adapt this kind of architecture. Our robot setup involves a laser range finder (LRF) for obstacle detection and a Kinect camera for people detection. On the intermediate layer we consider the standard navigation modules. Socially aware navigation mainly affects the local path planning ('local pp' Figure 6.1), but for example due to unavailability of socially acceptable paths, different routes on the global level may need to be considered. In this work, we focus on the interaction with local path-planning only.

Within the navigation layer we propose adding a new module, which we call *social (behavior) module*, consisting of a declarative rule base of social conventions, means to keep track of rules applicable in the current situation, and interaction to the path-planning module. The declarative nature of the knowledge eases integration with the high-level controller to foster awareness of the social context.

6.3 Social Conventions in Human Navigation

Human navigation is spatial interaction of individuals based on social conventions (Kirby, 2010). Hall described this interaction on the basis of spaces around individuals called *proxemics* or *social spaces* (Hall, 1966; Hall et al., 1968).

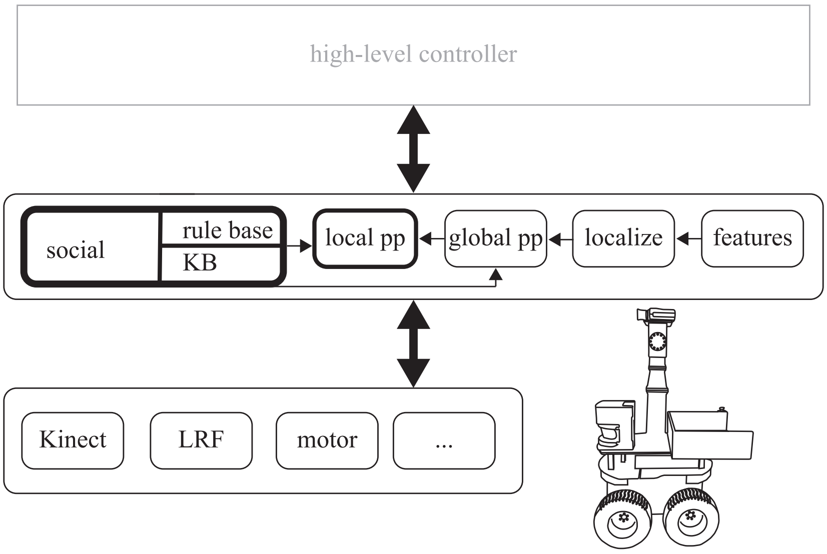


Figure 6.1: System overview, showing where social conventions could be integrated into a layered architecture.

6.3.1 Social Spaces

Hall distinguishes four different kinds of spaces:

intimate space touching, hugging, close conversation,

personal space interaction with well known people,

social space interaction with people, and

public space people in this region are ignored in general.

Additionally, for each region Hall defines a *close* and *far* sub-region to distinguish social interaction on a finer level of granularity. The nested structure of the spaces is depicted in Figure 6.2. Hall does not limit public space in its extend and the 7.6m relates to the *close public space*. Based on his work further investigations were carried out, e.g. asymmetries of personal spaces (Ashton and Shaw, 1980), intercultural differences of spaces (Burgess, 1983), or on which side people may pass in case walking paths are intersecting (Bitgood and Dukes, 2006). Furthermore, C. D. Frith and U. Frith (2006) considered how these findings of social conventions help to predict other people's behavior. Nevertheless, not much work has been presented on what people exactly do and why they behave like that. That is, the connection between qualitative descriptions and actual behavior remains unclear.

In addition to Hall's view other kinds of spaces which influence social behavior can be distinguished (Lindner and Eschenbach, 2011). *Affordance Spaces* are based on possible actions an agent could perform in that space (see e.g. (Galton, 2010), e.g. a park affords to play football in. In contrast *activity spaces* are defined on the basis of which actions are actually performed in them, for example a space a football game is actually played (see e.g. Ostermann

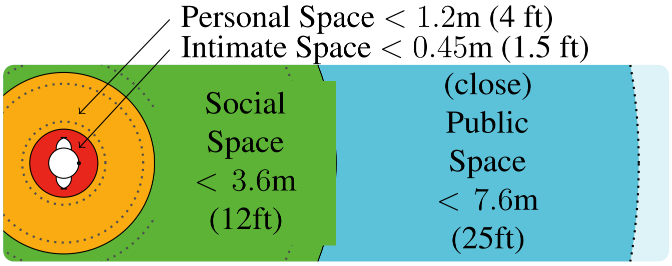

Personal Space < 1.2m (4 ft)
Intimate Space < 0.45m (1.5 ft)
Social Space < 3.6m (12ft)
(close) Public Space < 7.6m (25ft)

Figure 6.2: Social Spaces as described by Hall (1966)

and Timpf (2007); Kendon (1990)). *Territory Spaces* are regions which are claimed by a single or group of agents. In general, boundaries or extent of territories are marked with markers that are perceivable by potential intruders (e.g. Lawson (2001)). The last space to mention is *Penetrated Space* describing spaces which are influenced by other activities, for example, the space one's voice is perceivable (auditory) or the olfactory penetrated space of a barbecue. Although, the discrimination of all kinds of spaces is important, we restrict ourselves in this work to social spaces as defined by Hall for reasons of simplicity of the examples. In general, our approach is capable to deal with all kinds.

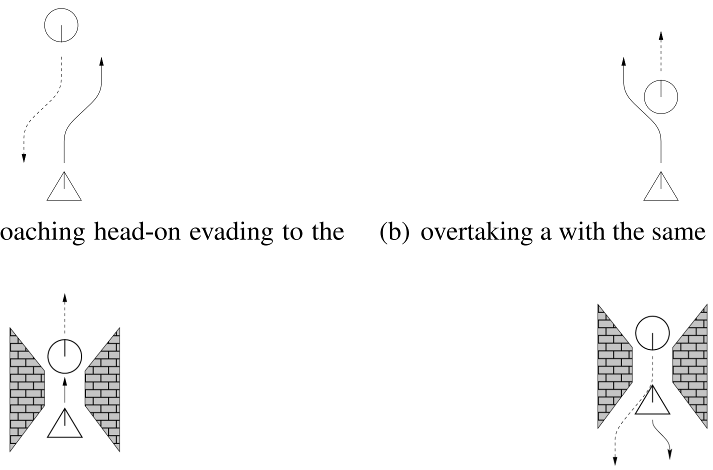
A first formalization of social spaces for robotic systems has been presented in citetLindner:2011:SocialSpaces, purely based on mereo-topological knowledge that is axiomatized within the highly expressive Situation Calculus. In contrast to this approach, we make use of qualitative concepts to link robotic perception with logic primitives. Also, we opt for a more restricted logic that nicely integrates with robotic architectures.

6.3.2 Classification of Social Conventions

In (Dylla, Coors, and Bhatt, 2012) a classification of social navigation conventions is presented. It comprises five (not necessarily disjunctive) categories mainly discriminated on the basis of spatial configurations and the number of people involved. The authors don't claim this taxonomy to be complete as, e.g., non-spatial discriminators like gender and cooperative behaviors like 'ladies first' are not considered.

The five categories and some of the conventions are:

1. approaching head-on or from behind
   a) pedestrians approaching in head-on both have to evade to the right (Figure 6.3(a))
   b) pedestrians moving in the same direction have to be overtaken on the left side (Figure 6.3(b))
2. crossing situations
   a) other (not necessarily moving) pedestrians on the left or right may be passed in their front with some distance,



(a) pedestrians approaching head-on evading to the right

(b) overtaking a with the same direction on the left

(c) following another pedestrian in a narrow passage

(d) yield for another pedestrian near a passage entrance

Figure 6.3: Four examplary conventions of pedestrian navigation. The triangle depicts the robot and the circle some other agent.

   b) may be passed in their back (with smaller distance), or
   c) the agent lets the other pedestrians pass (by stopping). Here the other agent has to move.

3. bottlenecks or narrow passages
   a) follow other pedestrians in a narrow passage (no overtaking possible) (Figure 6.3(c))
   b) ’crossing’ other pedestrian at narrow passage
   c) yield to others near passage entrance (Figure 6.3(d))
4. interaction with groups
   a) evading or passing a group on the outside
   b) crossing large groups (if passing is inappropriate or not possible at all)
5. individual constraints , i.e. conventions very dependent on the context an agent is in, e.g.
   a) in general moving on the right of a walkway, or
   b) no running in a library

These conventions may have to be adapted regarding cultural background. For example, in Great Britain pedestrians would evade to the left and overtaking should take place on the right side. Although not complete these kinds of conventions must be formalized in order to enable robots to be aware of these conventions and behave accordingly.

6.4 Representation of Coarse Knowledge

Social conventions are given in natural language and thus in a vague, coarse, and imprecise manner. By means of qualitative descriptions one can focus on distinctions between objects that make an important and relevant difference with respect to a given task (Kuipers, 1994). The field of *Qualitative Reasoning* (QR) is concerned with capturing such distinctions about objects in the real world, also considered as commonsense knowledge, with a limited set of symbols, i.e. without numerical values (Cohn and Hazarika, 2001). These distinctions are captured by *relations*, which summarize indistinguishable cases into a single symbol. For example, in most cases it is sufficient to refer to the color 'red' as it is not of importance whether it is slightly lighter or darker than some prototypical red. *Qualitative Spatio-Temporal Representation and Reasoning (QSTR)* is a subfield of QR, where the underlying physical structure of the domain can be exploited for performing well defined reasoning. In general, topological (e.g. part of) and positional relations can be distinguished (Freksa and Röhrig, 1993). Positional relations can be subdivided into orientation (relative: e.g. left, right; and absolute: e.g. south, north) and distance calculi (e.g. close, far). A set of relations together with operations on them is called a qualitative calculus.

Qualitative calculi are based on partition schemes of the underlying domain. The set of all possible relations between two spatial entities (or three in case of ternary calculi) is categorized into a set of atomic relationships called *base relations* ($\mathcal{BR}$) which either represent themselves meaningful relations for the task at hand or which allow these relations to be obtained by means of union of base relations. Since relations are ordinary sets, set-theoretic operations are applicable to qualitative relations. Algebraically, qualitative calculi are related to relation algebras in the sense of Tarski (Dylla, Mossakowski, et al., 2013). For the purpose of this paper it is sufficient that a qualitative calculus allows us to model binary (or ternary) relations between spatial entities using unions of base relations. Since base relations are defined by a partition scheme, they are naturally disjoint and thus conjunction would be useless. The most widely considered knowledge representation for qualitative calculi is constraint-based. Given a set of variables $X = \{x_1, \ldots, x_n\}$ and a set of base relations $\mathcal{BR} = \{b_1, \ldots, b_m\}$, a knowledge base consists of constraints $(x_i\ \{b_{i_1}, \ldots, b_{i_k}\}\ x_j)$ which say that spatial entities x_i and x_j are in relation $b_{i_1} \cup \ldots \cup b_{i_k}$. Qualitative spatial reasoning then provides us with (calculi-specific) algorithms to decide whether a constraint-satisfaction problem (CSP) consisting of such constraints is satisfiable or not (Renz and Nebel, 2007). The test of satisfiability also allows new constraints that follow from a given set of constraints to be inferred, similar to how resolution of logic formulas allows for deduction. An implementation of the methods described above is available via the qualitative spatial reasoning toolbox SparQ (Wolter and Wallgrün, 2010)[1].

In our formalizations we apply the topological Region Connection Calculus (*RCC-8*) and the Oriented Point Reasoning Algebra ($\mathcal{OPRA}_m$) (Moratz, 2006).

[1] available from `http://www.sfbtr8.uni-bremen.de/project/r3/sparq/`

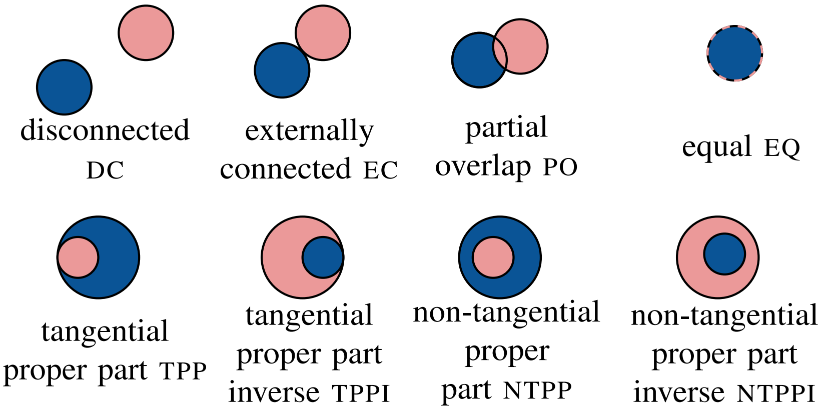

disconnected DC
externally connected EC
partial overlap PO
equal EQ
tangential proper part TPP
tangential proper part inverse TPPI
non-tangential proper part NTPP
non-tangential proper part inverse NTPPI

Figure 6.4: The eight base relations of *RCC-8*.

RCC-8: The Region Connection Calculus

Topological distinctions are inherently qualitative in nature and they also represent one of the most general and cognitively adequate ways for the representation of spatial information (Renz, Rauh, and Knauff, 2000). Based on this inherent qualitative nature different qualitative calculi were developed, among them the Region Connection Calculus (*RCC-8*) (Cohn, Bennett, et al., 1997) which is based on a binary *connection* relation $C(a, b)$ denoting that region a is connected to region b. Exploiting the connectivity of regions eight base relations are defined (see Figure 6.4).

$\mathcal{OPRA}_m$: A Relative Orientation Calculus

In $\mathcal{OPRA}_m$ relative orientation is partitioned into $2m$ equidistant angular cones and their separating lines with an intrinsic orientation. Within this work we apply granularity $m = 4$. For simplicity, these are numbered from 0 (intrinsic orientation) to 15. The relation is formed by a tuple of the relative orientation i of object B wrt. object A and vice versa (j), usually written as $A\,{}_4\angle_i^j\,B$. In Figure 6.5 (left) relation $A\,{}_4\angle_{13}^5\,B$ is depicted. This directly relates to a linguistic expression like ”B is ahead to the right of A moving in the same direction”. If point positions coincide relations are only determined by the segment number s of A the orientation of B falls in, e.g. $A\,{}_4\angle 3\,B$ in Figure 6.5 (right). We abbreviate complex relations by ${}_m\angle_{i_1-i_n}^{j_1-j_m} = \bigvee_{i=i_1}^{i_n} \bigvee_{j=j_1}^{j_m} {}_m\angle_i^j$ with $i, j \in \mathbb{Z}_{4m}$, $*$ abbreviates $0 - 4m$.

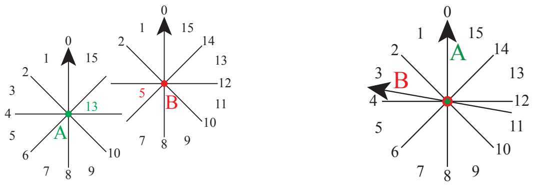


Figure 6.5: $\mathcal{OPRA}_4$ relations $\vec{A}\,{}_4\angle_{13}^5\,\vec{B}$ and $\vec{A}\,{}_4\angle 3\,\vec{B}$.

6.5 Formalization of Social Conventions

We have argued that a representation of socially adequate behavior is essentially based on qualitative concepts of spatial configurations. While qualitative constraint networks provide the means to represent a *configuration*, the representation of a *convention* also involves inscribed behavior. Conventions inscribe behavior within a certain context, typically triggered by an event or associated with a process (e.g., keep left when standing on an escalator). Among the different options to represent behavior, we choose to apply a temporal logic that links spatial configuration knowledge (snapshots) to temporal sequence. This approach has several nice implications:

- Employing a temporal logic to connect qualitative representations of snapshots allows static as well as dynamic spatial knowledge to be represented with the same vocabulary of spatial concepts.
- Like any standard logic, it allows non-spatial and non-temporal knowledge to be represented aside the spatio-temporal knowledge using propositional symbols.
- Last but not least, temporal logic has been integrated with motion planning and robot control (Kress-Gazit and Pappas, 2010; Ding et al., 2011; Kloetzer and Belta, 2010) and recognition of processes (Kreutzmann et al., 2013).

Of course, this approach has limitations. A technical limitation are missing quantifiers, thus the approach requires a predetermined maximum number of distinct objects to consider. A potential knowledge modeling limitation can be seen in binary truth evaluation of classic logics, e.g., if handling graduated, fine-grained concepts.

6.5.1 QLTL: Linear Temporal Logic with Qualitative Spatial Primitives

For representing social conventions we require temporal sequence knowledge, i.e. which situation occurs before/after another. This can be achieved using a lean logic like Linear Temporal Logic (LTL) (Pnueli, 1977). LTL is a modal logic that extends propositional logic with modal operators which connect statements about snapshots (called *worlds* in modal logic) using dedicated modal operators. In order to integrate this logic with spatial primitives, we choose to encode qualitative spatial relations in terms of propositional symbols and we extend the semantics to also include a spatial semantic.

Since conventions also depend on the types of objects involved in a certain configuration (person, obstacle, robot, etc.), we also introduce sorts to retain category information in logic formulae. This leads to the following syntax definition for our logic QLTL. The ingredients are:

- a set of spatial symbols $\mathcal{S}$. Let s be a number of sorts, then $S_i = \{s_{i,1}, s_{i,2}, \ldots\}, i = 1, \ldots, s$ are sets of spatial symbols and $\mathcal{S} := \bigcup_{i=1,\ldots,s} S_i$

- $\mathcal{R} = \{r_1, \ldots, r_n\}$ is a set of qualitative relation symbols
- $F = \{f_1, \ldots, f_l\}$ be a set of function symbols
- $G = \{g_1, g_2, \ldots\}$ is a set of propositional symbols for representing general, non-spatial knowledge
- The set of propositions P is defined as $P := G \cup \{r(s,t) | r \in \mathcal{R}, s, t \in (\mathcal{S} \cup \{f(s_i) | f \in F, s_i \in \mathcal{S}\})\}$.

The idea underlying this definition is to use natural notation of qualitative relations so that it is possible to represent spatial knowledge by a single propositional symbol. Propositions are either describing non-spatial facts (G) or some spatial relation r between two objects s and t which can either be sorts or some sort dependent aspect $f(s_i)$. For example, $\textsc{ntpp}(h, \mathrm{sec}(r))$ that a human h is standing in the security range sec of the robot r.

Formulae in QLTL are then defined recursively:

- p is a formula for every $p \in P$
- If ϕ is a formula, so is $\neg\phi$
- If ϕ, ψ are formulae, so is $\phi \otimes \psi$ with $\otimes \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$
- If ϕ is a formula, so is $M\phi$ with $M \in \{\circ, \Box, \diamond\}$
- If ϕ, ψ are formulae, so is $\phi\, N\, \psi$ with $N \in \{U, R\}$

The semantics of QLTL are similar to LTL, i.e., an interpretation establishes an ordered sequence of worlds. Within each single world, all propositional symbols are mapped to truth values *true* and *false*, inducing the interpretation of formulae composed with logic conjuncts $(\wedge, \vee, \ldots)$. For convenience of notation QLTL includes function symbols for relating the individual regions established by an agent, e.g., social space, personal space, etc. The semantics of a function f is a mapping $\mathcal{S} \rightarrow \mathcal{S}$ of spatial symbols. In other words, functions are used as shorthand notations for the respective spatial symbols. In QLTL we further require interpretations within all worlds to be *spatially consistent*, i.e., sensor interpretations must not be contradictory. This defines the *spatial semantics* of QLTL. Within one world, the interpretation of all (spatial) propositions $r(s_1, s_2)$ with $s_i \in \mathcal{S}$ or $s_i = f_j(s)$ for some $f_j \in F, s \in \mathcal{S}$ induces a qualitative constraint network with variables $\mathcal{S}$ and according constraints $r_i(x, y)$ where x, y are either the spatial symbols s_1, s_2 or the symbols obtained by application of $f_j(s_i)$. The spatial semantics of a relation r is defined by the respective qualitative calculus. Functions F are also assigned with a respective spatial semantic, e.g., mapping the position of a human to its estimated personal space. We say that an interpretation is spatially consistent if, first, all induced constraint networks are consistent and, second, if all mappings in F respect their spatial semantics, e.g., $\mathrm{personal}(h)$ is the region that determines the personal space of h as defined by the function $\mathrm{personal}$.

The semantics of modal operations connect distinct worlds within a sequence, identical to the sematics of LTL:

$\circ\phi$ **(next)** ϕ holds in the following world

$\Box\phi$ **(always)** ϕ holds in the current and in all future worlds

$\diamond\phi$ **(eventually)** ϕ holds in a future world, $(\diamond\phi \leftrightarrow \neg\Box\neg\phi)$

$\phi\, U\, \psi$ **(until)** ϕ holds at least until eventually ψ holds, but they dont have to hold at the same time

$\phi\, R\, \psi$ **(release)** ψ holds until and including the world in which ϕ first becomes true

In this work, we define the set of qualitative relation symbols Q to be the union of $\mathcal{OPRA}_4$ and *RCC-8* relations. This allows us to represent the social conventions. Since we will obtain an interpretation from the perception of the robot, we can assume it to be spatially consistent.

6.5.2 Conventions as QLTL formulae

We introduce a convenient notation for representing conventions as QLTL formulae. Conventions considered in this work essentially come in an “if-then-until” flavor in the sense that observing a certain event or process triggers (start) a sequence of required configurations (effects) to achieve a behavior in accordance with the regulation until an end state or a break criteria is reached. The end criteria is reached if everybody behaves as expected while the break criteria prevents the system from getting stuck in a rule if something unexpected happens, e.g. a person involved does not behave as expected by turning around and moving away. Break criteria might be a timeout or if involved persons leave the public space of the agent. For reasons of space we do not consider break criteria in further detail. Conventions can easily be represented as QLTL formula if we pursue a declarative description of regulation-compliant behavior using the pattern

$$\phi_{\text{start}} \rightarrow \circ\big(\phi_{\text{effect}}\, U\, (\phi_{\text{end}} \vee \phi_{\text{break}})\big) \tag{6.1}$$

We note that QLTL is expressive enough to allow conventions to be represented which are not effective. If, for example, the sub-formula specifying the trigger condition would refer to a future situation, than it may not be possible to decide whether the trigger condition is satisfied. Roughly saying, we want to exclude conventions of the kind “if this will turn out to be wrong, don’t do it”. Deciding effectiveness of a convention is beyond the scope of this paper—we assume that the conventions are stated in a way that allows trigger conditions to be detected at the time they apply.

We point out an inconvenience of directly using modal logic for knowledge representation, namely its lack of variables. The pattern introduced in (6.1) can involve propositional symbols only. As a consequence, for a convention that says how to avoid an obstacle, we require separate formulae, one for each individual obstacle. To this end, we introduce a shorthand notation for conventions that supports variables. In the following we write $x_1 : s_1, \ldots, x_n : s_n \ : \ \phi$ with variables $x_1, \ldots, x_n$ of respective sorts $s_1, \ldots, s_n$, meaning $\bigwedge_{x_i \in s_i, i=1,\ldots,n} \phi'$ with ϕ' obtained by substitution of x_i for the respective spatial or propositional symbol.

6.5.3 QLTL Representation of Social Conventions

We now exemplarily formalize social conventions from Class 1 and 2 (see Section 6.3.2) in QLTL. Throughout this section we use r to denote the robot ($r : \mathrm{robot}$), and h for humans ($h : \mathrm{human}$). In the following we use x and y to denote objects of any of these sorts.[2] To refer to the social spaces constituted by a human h we use the functional notation $\mathrm{intimate}(h)$, $\mathrm{personal}(h)$, $\mathrm{social}(h)$, $\mathrm{public}(h)$ respectively. On the syntactical level these functions denote special symbols for referring to the regions, whereas on the semantic level the specific region must be interpreted based on the physical object specified by h. In order to improve readability of formal conventions we define some macro relations to abbreviate complex relations. First, describing that two agents are in a head on situation: $\textsc{head\_on}(x,y) := x\,{}_4\angle_{15-1}^{15-1}\,y$ or are moving in the same direction: $\textsc{same\_dir}(x,y) := x\,{}_4\angle_{7-9}^{15-1}\,y \lor x\,{}_4\angle_{15-1}^{7-9}\,y$. Next we define x being on the left or right side of y: $\textsc{on\_left}(x,y) := y\,{}_4\angle_{2-6}^{*}\,x$, $\textsc{on\_right}(x,y) := y\,{}_4\angle_{11-13}^{*}\,x$ and x being in front or behind y: $\textsc{in\_front}(x,y) := y\,{}_4\angle_{15-1}^{*}\,x$, $\textsc{behind}(x,y) := y\,{}_4\angle_{5-11}^{*}\,x$. Finally, by $\textsc{overlap}$ we define that there is a partial or complete overlap: $\textsc{overlap}(x,y) := \textsc{po}(x,y) \lor \textsc{tpp}(x,y) \lor \textsc{ntpp}(x,y) \lor \textsc{tppi}(x,y) \lor \textsc{ntppi}(x,y)$. Using these primitives we can formalize convention 1a as follows:

$$\begin{aligned}
\phi^{1a}_{\text{start}} &:= \textsc{overlap}(r, \mathrm{social}(h)) \land \textsc{head\_on}(r,h) \\
\phi^{1a}_{\text{effect}} &:= \neg\textsc{po}(r, \mathrm{personal}(h)) \land \\
&\ \diamond\big(\textsc{on\_left}(h,r)\ R\ \textsc{behind}(h,r)\big) \\
\phi^{1a}_{\text{end}} &:= \textsc{behind}(h,r)
\end{aligned} \tag{6.2}$$

This means, if the robot enters or is in the social space of another agent h and they are head on, the robot must not move into the personal space of h. Furthermore, h has to be on the left of r, thus r needs to turn right until r has passed h, i.e., r is behind h. Convention 1b can be modeled similar except that they are in $\textsc{same\_dir}(r,h)$ instead of $\textsc{head\_on}(r,h)$ and r has to overtake on the left ($\textsc{on\_right}(h,r)$).

All conventions of class 2 are covered by the following:

$$\begin{aligned}
\phi^{2}_{\text{start}} &:= \textsc{overlap}(r, \mathrm{social}(h)) \land \\
&\ (\textsc{on\_left}(h,r) \lor \textsc{on\_right}(h,r)) \\
\phi^{2a}_{\text{effect}} &:= \neg\textsc{po}(r, \mathrm{personal}(h)) \land \diamond\big(\textsc{in\_front}(r,h)\big) \\
\phi^{2b}_{\text{effect}} &:= \neg\textsc{po}(r, \mathrm{intimate}(h)) \land \diamond\big(\textsc{behind}(r,h)\big) \\
\phi^{2c}_{\text{effect}} &:= \texttt{stop}(r)\ U\ \textsc{behind}(r,h) \\
\phi^{2}_{\text{effect}} &:= \phi^{2a}_{\text{effect}} \lor \phi^{2b}_{\text{effect}} \lor \phi^{2c}_{\text{effect}} \\
\phi^{2}_{\text{end}} &:= \textsc{behind}(h,r)
\end{aligned} \tag{6.3}$$

[2] Remark: these symbols need different interpretations regarding the relation they are used for. For *RCC-8* they need to be interpreted as regions, e.g. the space covered by an object ($\mathrm{obj}(x)$), and for $\mathcal{OPRA}_4$ as oriented point ($\mathrm{opos}(x)$). For brevity we omit this distinction in the presented formalizations.

If the robot enters the social space of h on the left or right, r has three options. Either he passes h in the front with not entering the personal space, pass behind h with a smaller distance, i.e. it is allowed to enter personal but not the intimate space, or he can stop until the human has passed.

For brevity we only sketch the main aspects to consider formalizing the three remaining classes. In case of a narrow passage (class 3) we need to represent an overlap of obstacles with a space of the agent to interact with, e.g. a wall to the left in the social space of h: $\textsc{overlap}(w, \mathrm{personal}(h)) \wedge \textsc{on\_left}(w, h)$. For dealing with conventions regarding groups (class 4) we need to redefine the spaces with respect to the individuals involved. One approach is to define the group space as the union of all individual spaces, e.g. $\mathrm{social}(g) := \cup_{h \in g}\mathrm{social}(h)$. Dealing with class 5 is straightforward. If the robot is in a specific context, e.g. a library l ($\phi_{\text{start}} = \textsc{overlap}(r, l)$), he has to adapt his behavior, e.g., switch to quiet mode ($\phi_{\text{effect}} = \texttt{q\_mode}(r)$), until he is not in the context anymore ($\phi_{\text{end}} = \neg\phi_{\text{start}}$).

6.6 Processing QLTL Conventions

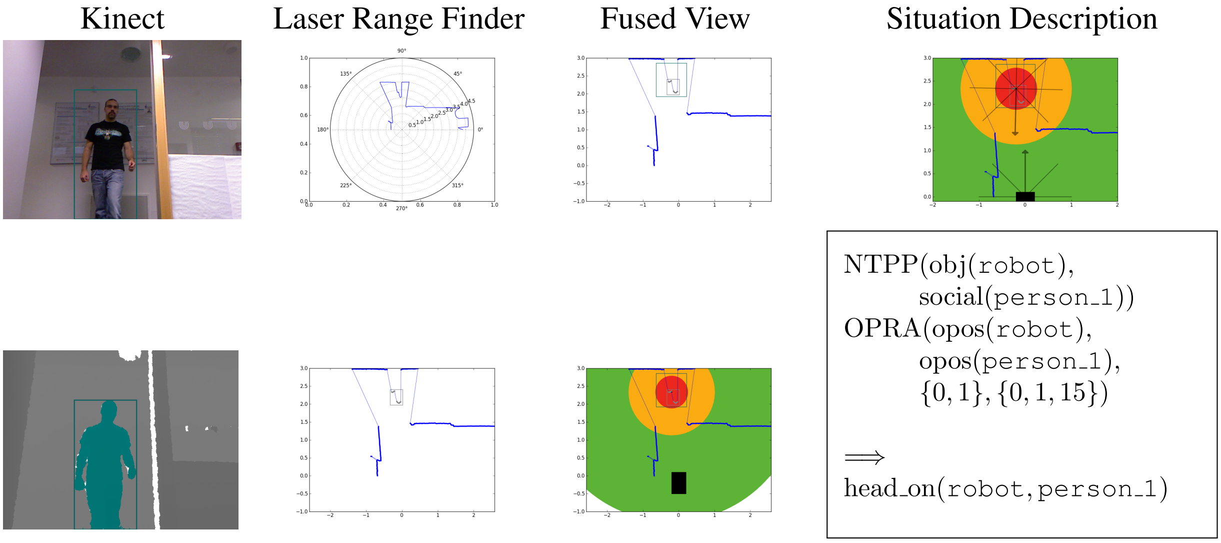
Given a set of QLTL formulae representing the social conventions known to the robot, the two tasks are to (1) identify that the preconditions of a pattern are met and (2) to determine actions which are admissible with respect to the pattern. Although the two tasks seem different at first sight, they both can be tackled with the method of testing convention applicability.

6.6.1 Detecting Applicability

Testing convention applicability requires perception of the robot to be matched against the precondition of a convention ϕ_{start}. The convention is applicable if the observations allow for a model of ϕ_{start}. We are thus confronted with the task of model checking where observations provide (partial) knowledge and the task is to decide whether this partial model can be extended to a spatially consistent interpretation that makes ϕ_{start} come true:

$$\text{observations} \models \phi_{\text{start}} \Leftrightarrow \phi_{\text{effect}}\ U\ \phi_{\text{end}} \text{ becomes active} \tag{6.4}$$

As discussed in (Kreutzmann et al., 2013) this task of model checking can be accomplished by first assigning propositional symbols to truth values according to the robot's observation and then applying an ASP solver (answer set programming) to search a model for a given precondition formula. In this work we are however involved with a more complex logic that additionally involves object sorts. During model checking one needs to ensure that a model also respects the correct association of object sorts, i.e., humans in the formula can only be matched to humans, obstacles to obstacles, etc. A straightforward yet sufficient solution can be realized in ASP-based model checking. ASP supports relational knowledge that allows sort knowledge like "$\mathrm{person}(x)$" to be denoted. Doing so for convention formulae as well as for observation ensures correct association.



$\mathrm{NTPP}(\mathrm{obj}(\texttt{robot}),$
$\quad \mathrm{social}(\texttt{person\_1}))$
$\mathrm{OPRA}(\mathrm{opos}(\texttt{robot}),$
$\quad \mathrm{opos}(\texttt{person\_1}),$
$\quad \{0,1\},\{0,1,15\})$

$\Longrightarrow$
$\mathrm{head\_on}(\texttt{robot},\texttt{person\_1})$

Figure 6.6: From the sensor readings to the knowledge base. Using a kinect and a laser range finder a human is detected and the positions and extends of his social spaces are determined, resulting in the knowledge base on the lower right. Due to uncertainty of the sensor readings, the $\mathcal{OPRA}_4$ relation of $robot\ {}_4\angle_1^1\ person$ is coarsened to $robot\ {}_4\angle_{0-1}^{15-1}\ person$ and returned by our system as $\mathrm{OPRA}(\mathrm{opos}(\texttt{robot}),\mathrm{opos}(\texttt{person\_1})\{0,1\},\{0,1,15\})$.

However, we are only interested in spatially consistent interpretations of the propositional symbols and filter out all spatially inconsistent models, as provided by SparQ (Wolter and Wallgrün, 2010). Filtering out spatially inconsistent models in a subsequent step to model checking can lead to large amounts of models that need to be rejected.

Therefore, we propose to introduce an intermediate step that responds to partial observations. Prior to grounding propositional symbols with observations, QSR is applied for constraint propagation in order to make implicit knowledge explicit. For example, we might observe two facts: two persons are standing in front, one to the left looking to the right and one to right looking left. Here, constraint propagation in $\mathcal{OPRA}_m$ would reveal the relation between both persons would be looking at one another. Enriching observations prior to model checking can thus help recognizing whether a convention's precondition is met. We are aware that due to noisy sensor data qualitative relations might be computed which do not map reality exactly, e.g., jitter at relation transition. For reasons of space we neglect a detailed consideration here and refer to work like (Dubba, Cohn, and Hogg, 2010) where sensor data is interpreted robustly by a pre-computation step.

6.6.2 Planning Admissible Actions

As mentioned earlier we reduce action planning to checking convention admissibility. The idea, as discussed in (Wolter, Dylla, and Kreutzmann, 2011), is to link the applicability check to a planner. Whenever the planner generates or extends a partial plan, these plans are checked for admissibility, discarding them if they do not match the convention. This kind of integration is easy with many planners like randomized road map planners or lattice-based planners. In order to check that a plan satisfies the inscribed effects of convention $\phi_{\text{start}} \rightarrow \circ\big(\phi_{\text{effect}}\ U\ \phi_{\text{end}}\big)$, we simply need to model-check $\phi_{\text{effect}}\ U\ \phi_{\text{end}}$ with the plan generated by the planner, treating the intermediate configurations of the plan like observations when checking convention applicability. Essentially, this step is the same as process recognition with LTL formulae as described in (Kreutzmann et al., 2013).

6.7 Examplary Case Study

In this section we outline a robot implementation of our approach using a SICK LMS200 laser range finder and a Microsoft Kinect RGBD camera mounted to an Active Media Pioneer 3-AT mobile robot. The Kinect sensors allows recognition of humans and estimating their position and orientation. The orientation estimate in our implementation is not very crisp, but as can be seen in Figure 6.6 our approach can handle such uncertainty. The laser scanner is used to detect obstacles and determine the free space. Currently, only local reactive behavior is employed.

In the example shown in Figure 6.6 a person is about to enter an office, which the robot is trying to leave. This triggers Convention 1a, but it cannot be executed since there exists no path in the free space, that would not cross the persons *personal space*. Thus the robot will stop until the world changes such that the convention is not applicable any more, i.e., the start condition is not met any more.

6.8 Conclusions and Future Work

The ability to respect social conventions is key for public acceptance of shared human-robot environments. Ultimately, the robot requires the ability to reflect on the social conventions, e.g., enabling it to decide on applicability of certain conventions. A promising approach towards such awareness is to pursue a declarative, abstract representation of conventions that supports abstract deliberation as well as integration with navigation components of a robotic system. By proposing the qualitative spatio-temporal logic QLTL we indicate how such abstract representation can be constructed. Qualitative primitives in the representation provide the important link between robot navigation and abstract logic. By adjoining qualitative spatial reasoning techniques from the SparQ toolbox with answer set programming (ASP) we obtain effective means to reason about QLTL formulae, in particular to recognize applicability of conventions. In future work we aim to exploit the flexibility of declarative convention specification in order to allow the robot to adapt to situations dynamically, in particular to handle necessary convention violations adequately.

References

Althaus, Philipp, Hiroshi Ishiguro, Takayuki Kanda, Takahiro Miyashita, and Henrik I. Christensen (2004). “Navigation for Human-Robot Interaction Tasks”. In: *IEEE International Conference on Robotics and Automation*, pp. 1894–1900.

Ashton, Nancy L. and Marvin E. Shaw (1980). “Empirical investigations of a reconceptualized personal space”. In: *Bulletin of the Psychonomic Society* 15.5, pp. 309–312.

Bitgood, Stephen and Stephany Dukes (2006). “Not Another Step! Economy of Movement and Pedestrian Choice Point Behavior in Shopping Malls”. In: *Environment and Behavior* 38.3, pp. 394–405. ISSN: 0013-9165.

Burgard, Wolfram, Armin B. Cremers, Dieter Fox, Dirk Hähnel, Gerhard Lakemeyer, Dirk Schulz, Walter Steiner, and Sebastian Thrun (1999). “Experiences with an interactive museum tour-guide robot”. In: *Artificial Intelligence* 114.1-2, pp. 3–55. ISSN: 0004-3702.

Burgess, J. Wesley (1983). “Interpersonal spacing behavior between surrounding nearest neighbors reflects both familiarity and environmental density”. In: *Ethology and Sociobiology* 4.1, pp. 11–17.

Cohn, Anthony G., Brandon Bennett, John M. Gooday, and Nicholas Mark Gotts (1997). “RCC: A calculus for Region based Qualitative Spatial Reasoning”. In: *GeoInformatica* 1, pp. 275–316.

Cohn, Anthony G. and Shyamanta M. Hazarika (2001). “Qualitative Spatial Representation and Reasoning: An Overview”. In: *Fundamenta Informaticae* 46.1-2, pp. 1–29.

Ding, Xu Chu, Stephen L. Smith, Calin Belta, and Daniela Rus (Apr. 2011). “LTL Control in Uncertain Environments with Probabilistic Satisfaction Guarantees”. In: arXiv:1104.1159v2 [math.OC]. Technical Report accompanying IFAC 2011. eprint: `1104.1159v2`.

Dubba, Krishna S. R., Anthony G. Cohn, and David C. Hogg (2010). “Event Model Learning from Complex Videos using ILP”. In: *19th European Conference on AI*. Ed. by Helder Coelho, Rudi Studer, and Michael Wooldridge. Vol. 215. IOS Press, pp. 93–98. ISBN: 978-1-60750-605-8.

Dylla, Frank, Martin Coors, and Mehul Bhatt (2012). “Socially Compliant Navigation in Crowded Environments”. In: *Spatial and Temporal Dynamics Workshop at ECAI 2012*.

Dylla, Frank, Till Mossakowski, Thomas Schneider, and Diedrich Wolter (2013). “Algebraic Properties of Qualitative Spatio-Temporal Calculi”. In: *Proceedings of the Conference On Spatial Information Theory (COSIT-13)*. Vol. 8116. LNCS. North Yorkshire, UK: Springer, pp. 516–536.

Freksa, Christian and Ralf Röhrig (1993). “Dimensions of Qualitative Spatial Reasoning”. In: *Proceedings of the III IMACS International Workshop on Qualitative Reasoning and Decision Technologies – QUARDET’93*. Ed. by N. Piera Carreté and Madan G. Singh. CIMNE Barcelona, pp. 483–492.

Frith, Chris D. and Uta Frith (2006). “How we predict what other people are going to do”. In: *Brain Research* 1079.1, pp. 36–46.

Galton, Antony (2010). “The Formalities of Affordance”. In: *Spatial and Temporal Dynamics Workshop at ECAI 2010*, pp. 1–6.

Hall, Edward T. (1966). *The Hidden Dimension, Man's Use of Space in Public and Private*. The Bodley Head Ltd.

Hall, Edward T. et al. (1968). "Proxemics [and Comments and Replies]". English. In: *Current Anthropology* 9.2/3, pages. ISSN: 00113204.

Kendon, Adam (1990). *Conducting Interaction: Patterns of Behavior in Focused Encounters*. Studies in interactional sociolinguistics. Cambridge University Press. ISBN: 9780521380676.

Kirby, Rachel (2010). "Social Robot Navigation". PhD thesis. Pittsburgh, PA: Robotics Institute, Carnegie Mellon University.

Kloetzer, Marius and Calin Belta (2010). "Automatic deployment of distributed teams of robots from temporal logic motion specifications". In: *IEEE Transactions on Robotics* 26.1, pp. 48–61.

Kress-Gazit, Hadas and George J. Pappas (2010). "Automatic Synthesis of Robot Controllers for Tasks with Locative Prepositions". In: *IEEE International Conference on Robotics and Automation*, pp. 3215–3220.

Kreutzmann, Arne, Immo Colonius, Diedrich Wolter, Frank Dylla, Lutz Frommberger, and Christian Freksa (2013). "Temporal logic for process specification and recognition". In: *Intelligent Service Robotics* 6.1, pp. 5–18.

Kuipers, Benjamin (1994). *Qualitative Reasoning: Modeling and simulation with incomplete knowledge*. Cambridge, Massachusetts, USA: MIT Press.

Lawson, Bryan (2001). *Language of Space*. Taylor & Francis. ISBN: 9780750652469.

Lindner, Felix and Carola Eschenbach (2011). "Towards a Formalization of Social Spaces for Socially Aware Robots". In: *COSIT*. Ed. by Max Egenhofer, Nicholas Giudice, Reinhard Moratz, and Michael Worboys. Vol. 6899. LNCS. Springer, pp. 283–303. ISBN: 978-3-642-23195-7.

Moratz, Reinhard (2006). "Representing Relative Direction as a Binary Relation of Oriented Points". In: *17th European Conference on AI*. Ed. by Gerhard Brewka, Silvia Coradeschi, Anna Perini, and Paolo Traverso. Italy: IOS Press, pp. 407–411. ISBN: 1-58603-642-4.

Ostermann, Frank and Sabine Timpf (2007). "Modelling space appropriation in public parks". In: *Proceedings of the 10th AGILE International Conference on Geographic Information Science*.

Pacchierotti, Elena, Henrik I. Christensen, and Patric Jensfelt (2005). "Embodied Social Interaction for Service Robots in Hallway Environments". In: *FSR*. Ed. by Peter I. Corke and Salah Sukkarieh. Vol. 25. Springer Tracts in Advanced Robotics. Springer, pp. 293–304. ISBN: 978-3-540-33452-1.

Pnueli, Amir (1977). "The temporal logic of programs". In: *Proceedings of the 18th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 46–57.

Renz, Jochen and Bernhard Nebel (2007). "Qualitative Spatial Reasoning Using Constraint Calculi". In: *Handbook of Spatial Logics*. Ed. by Marco Aiello, Ian E. Pratt-Hartmann, and Johan van Benthem. Springer, pp. 161–215. ISBN: 978-1-4020-5586-7.

Renz, Jochen, Reinhold Rauh, and Markus Knauff (2000). "Towards Cognitive Adequacy of Topological Spatial Relations". In: *Proc. of Spatial Cognition II*. Springer, pp. 184–197. ISBN: 3-540-67584-1.

Shi, Dongqing, Emmanuel G. Jr. Collins, Arturo Donate, Xiuwen Liu, Brian Goldiez, and Damion D. Dunlap (2008). “Human-aware robot motion planning with velocity constraints”. In: *CTS*. Ed. by William K. McQuay and Waleed W. Smari. IEEE, pp. 490–497.

Thrun, Sebastian et al. (1999). “MINERVA: A Second-Generation Museum Tour-Guide Robot”. In: *ICRA*. IEEE Robotics and Automation Society, pp. 1999–2005.

Wolter, Diedrich, Frank Dylla, and Arne Kreutzmann (2011). “Rule-Compliant Navigation With Qualitative Spatial Reasoning”. In: *Proceedings of the 4th Intl. Robotic Sailing Conference*. Springer.

Wolter, Diedrich and Jan Oliver Wallgrün (2010). “Qualitative Spatial Reasoning for Applications: New Challenges and the SparQ Toolbox”. In: *Qualitative Spatio-Temporal Representation and Reasoning: Trends and Future Directions*. Ed. by Shyamanta M. Hazarika. Hershey (PA), USA: IGI Global.

7 Qualitative Spatial and Temporal Reasoning with AND/OR Linear Programming

Arne Kreutzmann[1] and Diedrich Wolter[2]

1 Cognitive Systems Group, University of Bremen, Bremen, Germany

2 Smart Environments, University of Bamberg, Bamberg, Germany

Presented at
21st Europen Conference on Artificial Intelligence, 2014
and manuscript published in
Frontiers in Artificial Intelligence and Applications. Volume 263: ECAI 2014

Contributions:

I conducted the study, developed the theoretical part and implemented the algorithms. The manuscript was jointly written with larger parts mainly written by Diedrich Wolter.

Acknowledgements:

This work is partially funded by the DFG (SFB/TR-8, R3-[QShape]), financial support is gratefully acknowledged.

Abstract

This paper explores the use of generalized linear programming techniques to tackle two long-standing problems in qualitative spatio-temporal reasoning: Using LP as a unifying basis for reasoning, one can jointly reason about relations from different qualitative calculi. Also, concrete entities (fixed points, regions fixed in shape and/or position, etc.) can be mixed with free variables. Both features are important for applications but cannot be handled by existing techniques. In this paper we discuss properties of encoding constraint problems involving spatial and temporal relations. We advocate the use of AND/OR graphs to facilitate efficient reasoning and we show feasibility of our approach.

7.1 Introduction

Qualitative spatial and temporal reasoning (QSTR) is involved with knowledge representations that explicate relational knowledge between (spatial or temporal) entities (Ligozat, 2011; Renz and Nebel, 2007). QSTR has several important application areas both inside and outside AI. Over the past two decades of research, a rich repertoire of specialized representations has been proposed (see Dylla et al. (2013) for recent summary). Aside from the development of individually successful representations, called *qualitative calculi*, there are two penetrating and long-standing research questions that apply to all representations.

- How can qualitative calculi be combined, i.e., how can one jointly reason with knowledge represented in distinct calculi?
- How can qualitative representations incorporate grounded information, i.e., how can free-ranging and constrained variable domains (singleton, finite, numerical constraints) be mixed?

For the first question, two algebraic approaches have been considered, the loose and the tight coupling of spatial calculi (Wölfl and Westphal, 2009). While the loose coupling is too weak to obtain sound and complete reasoning, the tight coupling essentially means to manually develop a combined calculus. Combining individual approaches by translation into a common, expressive formalism would provide an answer to the question. However, formalisms expressive enough to capture a multitude of spatial and temporal relations such as algebraic geometry (e.g., see Bhatt, J. H. Lee, and Schultz (2011); Wolter (2012)) lead to infeasible complexity which limits applicability to toy problems.

The second question addresses needs of practical applications in which it is common that some objects to be reasoned about are already identified with concrete entities. This question has recently received attention (Li, Liu, and Wang, 2013), revealing the specific answer for the region connection calculus (RCC) (Randell, Cui, and Cohn, 1992). For other calculi, this question remains open.

In this paper we are concerned with developing a unified framework for QSTR that provides a solution to both questions and which is applicable to a wide range of qualitative calculi. To this end, we further explore the use linear programming (LP). LP is interesting since it can capture several calculi in an efficient framework, either exactly or by tight approximations. While LP techniques have already been used in QSTR for selected tasks (e.g., J. H. Lee, Renz, and Wolter (2013); Ligozat (2011); Jonsson and Bäckström (1998)), potentials of LP frameworks have not yet been explored thoroughly. We propose a basic language Q_{basic} for QSTR and describe how selected qualitative calculi can be encoded in it. For reasoning with Q_{basic}, translations into LP frameworks are performed. Comparing mixed integer linear programming (MILP) and AND/OR graphs combined with LP, we advocate the latter since it allows sophisticated optimizations that foster efficient reasoning. To further motivate our aims, let us outline a problem from the field of safety in autonomous mobile systems.

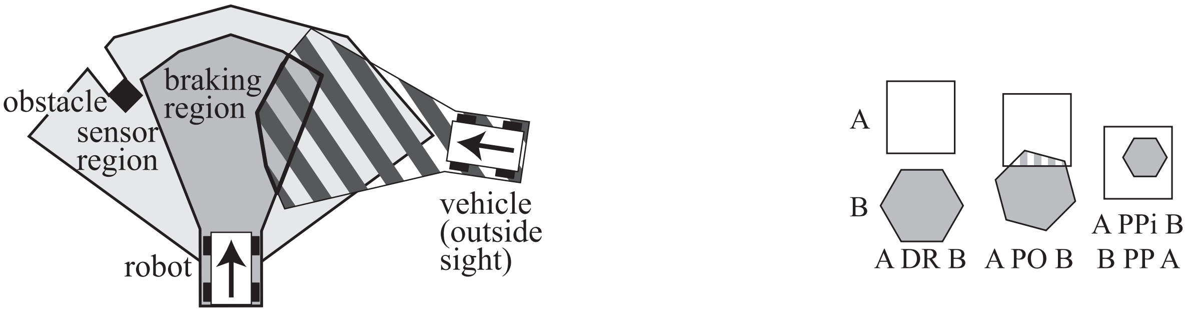


Figure 7.1: **Left:** regions in safe navigation, overlapping braking regions are dangerous. **Right:** RCC-5 topological relations discrete (DR), partial overlap (PO) and proper part (inverse) (PP, PPi); equality (EQ) not shown

7.1.1 Motivating Problem

Täubig et al. (2012) present in "Guaranteeing functional safety: design for provability and computer-aided verification" a supervisory method for an autonomous vehicle to ensure that the vehicle does not issue commands which could (potentially) lead to a collision with a static obstacle. The particular contribution is a formal method for which certification according to IEC 61508 was achieved.

From a QSTR perspective, safe navigation could have been formalized using RCC relations. Considering the primitives illustrated in Fig. 7.1, we call $\mathrm{free\_space\_sensed}$ the region within sensor range that is free of obstacles. Using $\mathfrak{r}$ as reference to the position of the robot, an intuitive formalization could start as follows:

$$\phi_{\text{safe}} = (\mathrm{braking\_region}(\mathfrak{r})\ \texttt{pp}\ \mathrm{sensor\_region}(\mathfrak{r})) \tag{7.1a}$$

The specification would also identify potentially dangerous locations (denoted $\mathfrak{h}$), i.e., positions of obstacles within the braking region but outside sensor range, e.g., due to occlusion. Using $\mathrm{reg}()$ to refer to the region occupied by an obstacle, we obtain

$$\begin{aligned}\phi_{\text{dangerous}} = &((\mathrm{reg}(\mathfrak{h})\ \ \texttt{PP}\ \ \mathrm{braking\_region}(\mathfrak{r})\\ &\vee (\mathrm{reg}(\mathfrak{h})\ \ \texttt{PO}\ \ \mathrm{braking\_region}(\mathfrak{r})))\\ &\wedge (\mathrm{reg}(\mathfrak{h})\ \ \texttt{DR}\ \ \mathrm{sensor\_region}(\mathfrak{r}))\end{aligned} \tag{7.1b}$$

The above formulae essentially describe safety of navigation as considered in (Täubig et al., 2012), they are valid for both static and dynamic obstacles. Extending the specification to consider a moving object $\mathfrak{m}$, its respective braking region needs to be considered too:

$$\psi_{\text{dangerous}} = (\mathrm{braking\_region}(\mathfrak{r})\ \texttt{PO}\ \mathrm{braking\_region}(\mathfrak{m})) \tag{7.2}$$

Observe that $\mathrm{braking\_region}(\mathfrak{m})$ may either refer to a concrete region if $\mathfrak{m}$ is observed, but it may also be unknown if $\mathfrak{m}$ is positioned outside sensor range, i.e., $(\mathrm{sensor\_region}(\mathfrak{r})\ \texttt{DR}\ \mathrm{reg}(\mathfrak{m}))$.

A next step in a formalization could involve traffic rules such "left shall yield to right", saying that the robot has to let vehicles pass which approach from the right, but in turn the robot is allowed to pass by a vehicle approaching from the left:

$$\begin{aligned}\psi'_{\text{dangerous}} = &(\text{braking\_region}(\mathfrak{r})\ \texttt{PO}\ \text{braking\_region}(\mathfrak{m}))\\ &\wedge (\text{sensor\_region}(\mathfrak{r})\ \texttt{DR}\ \text{reg}(\mathfrak{m}))\\ &\wedge (\mathfrak{r}\ \texttt{right}\ \mathfrak{m}),\end{aligned} \tag{7.3}$$

As can be seen, the example of safe navigation from the literature can be represented with qualitative relations and easily be advanced beyond (Täubig et al., 2012) by considering moving obstacles. However, in order to decide whether an issued driving command is safe, we require means to handle partially grounded information such as the polygonal braking area alongside unknown regions such as the breaking area of a hidden object $\mathfrak{m}$. For considering traffic rules, qualitative representations for region topology (e.g., RCC) and directional knowledge (e.g., OPRA (Mossakowski and Moratz, 2012)) would need to be mixed. As we will see, the techniques proposed in this paper provide a solution to both problems.

7.2 Qualitative Spatial and Temporal Reasoning

We briefly introduce key concepts from the field of QSTR necessary in our context. For more detailed coverage we kindly refer to the literature, e.g., Ligozat (2011); Renz and Nebel (2007); Dylla et al. (2013).

In QSTR, one is involved with representations that are based on finite sets of relations called *base relations* which partition a spatial or temporal domain into meaningful parts. Technically speaking, the set of base relations is jointly exhaustive and pairwise disjoint (JEPD). Due to the set-theoretic semantics of relations, any set of base relations $\mathcal{B}$ induces a Boolean set algebra of qualitative relations $\mathcal{R}_{\mathcal{B}} = \bigcup_{R \in 2^{\mathcal{B}}} \left(\bigcup_{r \in R} r\right)$. The Boolean set algebra, in conjunction with relation operation converse $\smile$: $\mathcal{R} \to \mathcal{R}$, $r^{\smile} = \{(x,y)|(y,x) \in r\}$ and weak composition $\diamond : \mathcal{R} \times \mathcal{R} \to \mathcal{R}$, $r \diamond s = \{(\bigcap_q |(r \circ s) \subseteq q,\, q \in \mathcal{R}\}$ constitutes the algebraic structure of the representation which is also called a *qualitative calculus* (Dylla et al., 2013).

These qualitative relations serve as constraint language to represent constraints like $(X\ \texttt{DR}\ Y)$, or $(X\ (\texttt{DR} \cup \texttt{PO})\ Z)$ whereby DR is a base relation in RCC-5 (Randell, Cui, and Cohn, 1992) and $(\texttt{DR} \cup \texttt{PO})$ is a respective qualitative relation (see Fig. 7.1.). Constraint-based reasoning is the single most important form of QSR and it is considered as a decision problem.

Definition 10 (QCSP)**.** Given a constraint satisfaction problem (CSP) with variables X ranging over domain D that involves only binary constraints that are qualitative relations in one calculus over domain D, i.e., $c_{i,j} \in \mathcal{R}_{\mathcal{B}}$ for some set of base relations $\mathcal{B}$ over D. The problem QCSP is then to decide whether there exists a valuation of X with values from D such that all constraints are satisfied.

Since D is typically infinite, special techniques are necessary that allow QCSP to be solved efficiently for various qualitative calculi. The complexity of QCSP is usually NP-complete, while reasoning with base relations only may be in P. There exist however calculi that involve directional relations such as `right` from the motivating example that are inherently NP-hard and, assuming P≠NP, require exponential time algorithms (Wolter and J. H. Lee, 2010).

7.3 Approaches to Unifying QSTR

With respect to capturing semantics of QSTR, expressive and hence computationally very hard languages are commonly used. For example, algebraic geometry provides a suitable basis to represent many qualitative calculi, but reasoning is only feasible for toy problems (Wolter, 2012). In order to obtain an efficient unified approach to reasoning, few approaches have been proposed so far.

A decomposition of the algebraic structure of calculi has been proposed in (Hué, Westphal, and Wölfl, 2012) that allows QCSP instances to be encoded as SAT instances. However, the method is limited to calculi in which composition-based reasoning can be used to decide consistency (see (Renz and Nebel, 2007)) which, e.g., excludes RCC in the domain of polygons (Kontchakov et al., 2010) or calculi involving directional relations.

Linear programming has previously been considered to tackle selected, isolated problems in QSTR. J. H. Lee, Renz, and Wolter (2013) describe a reasoning method for directional relations that employs an LP solver to check consistency of STAR (Renz and Mitra, 2004) QCSPs and to compute a realization. In temporal reasoning, LP has previously been considered as a backbone to unifying temporal reasoning, since temporal relations are largely based on linear inequalities. Jonsson and Bäckström (1998) describe an approach based on disjunctive linear relations that is similar to ours. In order to extend their idea to spatial relations, we introduce *oracles* that allows us to cope with the higher expressiveness of spatial relations. This requires a new approach to reasoning.

7.4 A unifying language for QSTR

We now introduce the new language Q_{basic}. The motivation of this language is to separate the translation from QSTR into a common language from translation into a specific LP framework in order to allow different LP backends to be used without the need of re-encoding all spatial calculi. Moreover, Q_{basic} explicates some nice features we obtain as side effects but which are helpful on their own, most notably the propositional closure of qualitative constraints that is not expressible in standard QSTR, e.g., in Q_{basic} we can express $((x\,\alpha\,y) \wedge (y\,\beta\,z)) \vee (x\,\gamma\,y)$.

The primitives of the new language Q_{basic} are systems of inequalities that may contain non-linear elements. When the non-linear elements are externally grounded, the resulting system of inequalities becomes linear. By restricting the domains of the non-linear elements to finite sets we obtain a flexible discretization scheme that easily outperforms any fixed discretization of a spatial or temporal domain. For example, we can choose a finite set of 360 angular 2D directions of lines $\{(\sin(\frac{k}{180}\pi), \cos(\frac{k}{180}\pi))|k = 0, 1, \ldots 359\}$ when reasoning about lines in the plane, while realizing these directions on a discrete coordinate would require a grid that grows with the number of lines to be positioned.

Definition 11. We call $S^n = \langle \mathcal{O}, G \rangle$ a *system of finite disjunctive linear inequalites over* $\mathbb{R}^n$ *with oracle values* $\mathcal{O}$, where $\mathcal{O}$ is a finite set and G is a mapping $G : \mathcal{O} \to \langle \mathbb{R}^{m_G \times n}, \mathbb{R}^{m_G} \rangle$. We say $s = \langle x, o \rangle \in \langle \mathbb{R}^n, \mathcal{O} \rangle$ is a solution of S^n iff $G(o) = \langle A_o, b_o \rangle$ and $A_o \cdot x \leq b_o$, using the component-wise interpretation of $\leq$ used in LP, i.e., $(x_1, \dots, x_n) \leq (y_1, \dots, y_n)$ iff $x_i \leq y_i$ for all $i = 1, \dots, n$.

Definition 12 (Q_{basic})**.** We call $\langle \mathbb{R}^n, \mathcal{O} \rangle$ the *domain* and $\mathcal{S} = \{S_1^n, \dots\}$ the set of *symbols*, whereby any symbol S_i^n is a system of finite disjunctive linear inequalities sharing the same oracle $\mathcal{O}$ as defined above. A choice of D and $\mathcal{S}$ is called the signature of our language. Given a signature, we define a Q_{basic} formula ϕ as follows:

$$\phi =_{\text{def}} S_i^n \mid \top \mid \bot \mid \neg\phi \mid \phi \wedge \psi.$$

Given $x \in D$ and a $o \in \mathcal{O}$, we inductively define the notion of a formula ϕ being satisfied in $\langle x, o \rangle$ as follows:

$$x, o \models S^n \quad \text{iff } \langle x, o \rangle \text{ is a solution of } S^n \tag{7.4}$$

$$x, o \models \top \quad \text{always} \tag{7.5}$$

$$x, o \models \bot \quad \text{never} \tag{7.6}$$

$$x, o \models \neg\phi \quad \text{iff } \langle x, o \rangle \text{ is } \textit{not} \text{ a solution of } S^n \tag{7.7}$$

$$x, o \models \phi \wedge \psi \quad \text{iff } x, o \models \phi \text{ and } x, o \models \psi \tag{7.8}$$

The other Boolean connectives are defined as usual.

Corollary 3. *Deciding satisfiability of a* Q_{basic} *formula is NP-complete*

7.5 Encoding QCSP in Q_{basic}

This section provides an overview of how QCSP instances for several calculi can be encoded in Q_{basic}. We show how qualitative relations can be represented as systems of finite disjunctive linear inequalites. Due to space constraints, definitions of the individual calculi are omitted. Refer to (Ligozat, 2011; Dylla et al., 2013) for definitions and further references.

7.5.1 Temporal Calculi

As pointed out in Jonsson and Bäckström (1998), temporal relations can be described by linear inequalities. Strictness in the sense $x < y$ can be resolved by introducing a fixed $\varepsilon > 0$ and rewriting to $x + \varepsilon \leq y$ since the qualitative temporal relations considered do not rely on absolute values.

7.5.2 Direction Calculi

Given a vector $\vec{v} \in \mathbb{R}^2$, we call $\vec{v}^\perp$ its left normal obtained by $90°$ counter-clockwise rotation.

Given two (variable) points $p, q \in \mathbb{R}^2$ and a fixed orientation expressed as a vector $\vec{v} \in \mathbb{R}^2$, we define the following constraints by translation to Q_{basic}:

$$\begin{array}{llll} p\ \texttt{left}_{\vec{v}}\ q & =_{\text{def}} \vec{q}^T \cdot \vec{v}^\perp - \vec{p}^T \cdot v^\perp \leq 0 & (q \text{ left of } p) \\ p\ \texttt{right}_{\vec{v}}\ q & =_{\text{def}} \vec{p}^T \cdot \vec{v}^\perp - \vec{q}^T \cdot v^\perp \leq 0 & (q \text{ right of } p) \\ p\ \texttt{front}_{\vec{v}}\ q & =_{\text{def}} \quad \vec{p}^T \cdot \vec{v} - \vec{q}^T \cdot v \leq 0 & (q \text{ in front of } p) \\ p\ \texttt{back}_{\vec{v}}\ q & =_{\text{def}} \quad \vec{q}^T \cdot \vec{v} - \vec{p}^T \cdot v \leq 0 & (q \text{ behind } p) \end{array} \tag{7.9}$$

The relations $\texttt{left}_{\vec{v}}$, $\texttt{right}_{\vec{v}}$, $\texttt{front}_{\vec{v}}$, $\texttt{back}_{\vec{v}}$ are not *pairwise disjoint* (they overlap in one quadrant) but they are *jointly exhaustive*.

Theorem 3. *Let ϕ be a propositional formula with atoms of the kind $(\mathbf{x}\ \texttt{R}\ \mathbf{y})$, where $\texttt{R}$ is a relation as defined above.*

Let $\operatorname{var}(\phi)$ *denote the number of (distinct) variables in ϕ and let* $\operatorname{rel}(\phi)$ *denote the number of (distinct) relations in ϕ, then ϕ can be translated into a Q_{basic} formula with signature $D = \mathbb{R}^{2\operatorname{var}(\phi)}$, $|(S)| = \operatorname{rel}(\phi)$, and $\mathcal{O} = \emptyset$.*

Proof. Let $I : V \to \{1, \ldots, n\}$ be a bijective mapping between the variables and corresponding dimension in $\mathbb{R}^{2\operatorname{var}(\phi)}$. We define

$$H_i =_{\text{def}} \begin{pmatrix} \underbrace{0 \ldots 0}_{2\cdot(I(i)-1)} & 1 & 0 & 0\ldots \\ 0 \ldots 0 & 0 & 1 & 0\ldots \end{pmatrix}, \quad H_{i,j} =_{\text{def}} \begin{pmatrix} H_i \\ H_j \end{pmatrix}.$$

In the given formula ϕ, replace all atoms $(x_i\ R_{\vec{v}}\ x_j)$ by $S_k = \langle \{\}, \langle H_{i,j}^T A_{R_{\vec{v}}} H_{i,j}, 0 \rangle \rangle$, where $A_{R_{\vec{v}}}$ is the corresponding matrix to represent inequality as given by Eq. 7.9. This yields a Q_{basic} formula with the signature, $D = \mathbb{R}^{2\operatorname{var}(\phi)}$, $\mathcal{O} = \{\}$, and $\mathcal{S}$ as the set comprising all S_k defined above.

□

Consider two arbitrarily fixed vectors $\vec{s}$ and $\vec{t}$ such that the counter-clockwise angle between $\vec{s}$ and $\vec{t}$ does not exceed $180°$. A (variable) point q with respect to a (variable) point p is said to be inside the sector spanned by $\vec{s}$ and $\vec{t}$, iff:

$$(p\ \texttt{left}_{\vec{s}}\ q) \wedge (p\ \texttt{right}_{\vec{t}}\ q) \tag{7.10}$$

All cardinal direction calculi considered in the literature are either based on half-plane or sector membership, whereby half-plane normals and sectors are globally aligned to one of finitely many directions. This makes mapping QCSP instances to Q_{basic} with any of these calculi straightforward using either Eq. 7.10 or $\texttt{front}_{\vec{n}}$ where $\vec{n}$ denotes the respective half-plane normal. No oracle needs to be introduced. Since all these calculi are scale-invariant like temporal calculi, the same approach of introducing ε can be applied to represent truly $\texttt{left}_{\vec{v}}$, $\texttt{right}_{\vec{v}}$, etc. Applicability to the most important cardinal direction calculi is shown in Tab. 7.1.

Theorem 4. *StarVars (J. H. Lee, Renz, and Wolter, 2013) can be represented by Q_{basic}.*

Proof. StarVars, like Star (Renz and Mitra, 2004), employs sector-shaped spatial relations. The sectors in StarVars are rotated by an undetermined angle $\frac{2i}{2^N}\pi,\ i = 0, \dots 2^N - 1$ for a fixed N. Choosing these angles as oracles, the construction of the Q_{basic} formula follows directly from (J. H. Lee, Renz, and Wolter, 2013) which also employs an LP algorithm to decide consistency. □

Theorem 5. *$\mathcal{OPRA}$ can be mapped to Q_{basic} if the domain of directions is restricted to a finite set.*

Proof. Interpreted over finite domain of directions, $\mathcal{OPRA}$ relations can be represented as two conjuncts of StarVars relations (J. H. Lee, Renz, and Wolter, 2013). □

7.5.3 Region Connection Calculus

In this work we only consider planar regions in form of simple, i.e, not self-intersecting polygons. We start with convex polygons since the mappings can then be generalized to non-convex polygons by considering a convex partitioning and disjunctively adjoining the linear programs.

First note that the relation saying that a point is located inside a simple convex polygon positioned at an unknown origin can be represented by a LP. This is due to the point-in-polygon test being based on half-plane membership tests which are linear inequalities and stay linear if the whole polygon is translated by unknown x, y. For convex polygons, point-outside-polygon can also be modeled by disjunctively adjoining the negated clauses of the point-in-polygon test.

Corollary 4. *If two simple convex polygons do not share a common point, then there exists a line parallel to one edge which separates the space between both polygons.*

This fact grants a mapping for the RCC relation *discrete* saying that regions do not share a common interior part. For simple convex polygons, we disjunctively choose one edge as the dividing line.

Let two simple convex polygons P and Q be defined by vertices $v_1^P, \dots, v_k^P$ and $v_1^Q, \dots, v_m^Q$ in counter-clockwise orientation. We write e_i^P to refer to edge $v_i^P, v_{(i+1) \mod k}^P$ and d_i^P to refer to direction $v_{(i+1) \mod k}^P - v_i^P$ and obtain:

$$\begin{aligned}(P\ \mathtt{dr}_{\text{conv}}\ Q) =_{\text{def}} & \bigvee_{\vec{e}_i^P} \bigwedge_{v_j^Q} (v_i^P\ \mathtt{right}_{\vec{d}_i^P}\ v_j^Q) \\ \vee & \bigvee_{\vec{e}_i^Q} \bigwedge_{v_j^P} (v_i^Q\ \mathtt{right}_{\vec{d}_i^Q}\ v_j^P)\end{aligned} \tag{7.11}$$

Analogously, $\mathtt{dc}_{\text{conv}}$ can be defined, except that touching points need to be excluded by using $\neg(v_i^P\ \mathtt{left}\ v_j^Q)$ instead of $(v_i^P\ \mathtt{right}\ v_j^Q)$.

Given P as above we can express that point x lies on the edge e_i^P, i.e., between v_i^P and v_{i+1}^P, including both vertices.

$$\begin{aligned}(e_i^P \, \mathtt{cont} \; x) =_{\text{def}} (v_i^P \; \text{left}_{(\vec{v}_{i+1}^P - \vec{v}_i^P)} \; x) \wedge (v_i^P \; \text{right}_{(\vec{v}_{i+1}^P - \vec{v}_i^P)} \; x) \\ \wedge \, (v_i^P \; \text{front}_{(\vec{v}_{i+1}^P - \vec{v}_i^P)} \; x) \wedge (v_{i+1}^P \; \text{back}_{(\vec{v}_{i+1}^P - \vec{v}_i^P)} \; x), \quad (7.12)\end{aligned}$$

External connection can be mapped to Q_{basic} as follows:

$$\begin{aligned}(P \; \mathtt{t}_{\text{conv}} \; Q) &=_{\text{def}} \bigvee_{e_i^P} \Big[\bigwedge_{v_j^Q} (v_i^P \; \mathtt{right}_{(\vec{v}_{i+1}^P - \vec{v}_i^P)} \; v_j^Q) \\ &\qquad \wedge \bigvee_{v_j^Q} (e_i^P \; \mathtt{cont} \; v_j^Q) \Big] \\ (P \; \mathtt{ec}_{\text{conv}} \; Q) &=_{\text{def}} (P \; \mathtt{t}_{\text{conv}} \; Q) \vee (Q \; \mathtt{t}_{\text{conv}} \; P) \qquad (7.13)\end{aligned}$$

Theorem 6. *RCC-5 and RCC-8 (Randell, Cui, and Cohn, 1992) can be mapped to Q_{basic} for the domain of simple (i.e., not self-intersecting) polygons in 2D space that involve at most N vertices each.*

Proof. We need to show how the relations of RCC-8 can be stated in Q_{basic}, RCC-5 relations can then be obtained by disjunctive combinations, e.g., $(P \, \mathtt{DR}_{\text{RCC-5}} \, Q) = (P \, \mathtt{DC}_{\text{RCC-8}} \, Q) \vee (P \, \mathtt{EC}_{\text{RCC-8}} \, Q)$. The vertex limit N is required to obtain finite formulae. For RCC-8, the following mapping can be employed:

$$\begin{aligned}(P \; \mathtt{dc} \; Q) &=_{\text{def}} \bigwedge_{P^C \in C_P} \bigwedge_{Q^C \in C_Q} (P^C \; \mathtt{dc}_{\text{conv}} \; Q^C) \qquad (7.14) \\ (P \; \mathtt{ec} \; Q) &=_{\text{def}} \bigvee_{P^C \in C_P} \bigvee_{Q^C \in C_Q} (P^C \; \mathtt{ec}_{\text{conv}} \; Q^C) \\ &\qquad \wedge \bigwedge_{P^C \in C_P} \bigwedge_{Q^C \in C_Q} (P^C \; \mathtt{dr}_{\text{conv}} \; Q^C) \qquad (7.15)\end{aligned}$$

Given three fresh variables τ_1, τ_2, τ_3 denoting points:

$$\begin{aligned}(P \; \mathtt{po} \; Q) =_{\text{def}} &\big((\tau_1 \; \mathtt{inside} \; P) \wedge (\tau_1 \; \mathtt{inside} \; Q)\big) \\ \wedge &\big((\tau_2 \; \mathtt{inside} \; P) \wedge \neg(\tau_2 \; \mathtt{inside} \; Q)\big) \\ \wedge &\big(\neg(\tau_3 \; \mathtt{inside} \; P) \wedge (\tau_3 \; \mathtt{inside} \; Q)\big) \qquad (7.16)\end{aligned}$$

For containment it is not sufficient that all vertices of one polygon P are inside another polygon Q, see Fig. 7.2. Let I_Q denote edges introduced by the convex partitioning. If an edge E of P

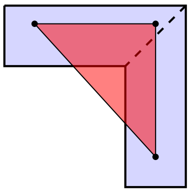


Figure 7.2: Convex region (red) partially overlapping a non-convex region (blue) although all vertices of the red region are inside the blue region.

overlaps with a sequence of adjacent convex parts of Q, all I_Q's of this sequence need to cross, i.e, one endpoint of I_Q lies left of and the other right of E. In the following, this is denoted by the formula $(P \otimes Q)$.

$$(P \,\texttt{pp}\, Q) =_{\text{def}} \bigwedge_{v_j^P} (v_j^P \texttt{ inside } Q) \wedge (P \otimes Q)$$

$$\begin{aligned}(P \,\texttt{tpp}\, Q) =_{\text{def}} & (P \,\texttt{pp}\, Q) \\ & \wedge \Big[\bigvee_{e_i^P} \bigvee_{v_j^Q} (e_i^P \texttt{ contains } v_j^Q) \\ & \vee \bigvee_{e_i^Q} \bigvee_{v_j^P} (e_i^Q \texttt{ contains } v_j^P) \Big] \end{aligned} \tag{7.17}$$

$$(P \,\texttt{ntpp}\, Q) =_{\text{def}} (P \,\texttt{pp}\, Q) \wedge \bigwedge_{v_j^Q} \neg (v_j^Q \texttt{ inside } P) \tag{7.18}$$

Due to space constraints we omit converse relations `ntppi`, `tppi` and equality `eq`, as well as $(P \otimes Q)$. □

7.6 Using Spatial Reasoning to Reduce Formula Size

Key to making reasoning in Q_{basic} efficient is reducing formula size. Aside from rewriting and simplification, we also apply classic QSTR reasoning methods to prune away implicit sub-formulae. The process of simplification can be interwoven with how QCSP instances are translated into Q_{basic} formulae to avoid uneccesarily generating systems of finite disjunctive linear programs.

Removing Redundant Information In case of partially grounded information, we first check whether constraint relations are declared between two grounded entities. Then, we check if the relation holds and replace it accordingly by $\top$ or $\bot$.

Given a set of constraints over a single qualitative calculus, we can apply composition-based constraint propagation to identify redundant constraints, e.g., in the set $\{(A \texttt{ dc } B), (C \texttt{ ntpp } B), (A \texttt{ dc } C)\}$ the constraint $(A \texttt{ dc } C)$ is redundant since it is implied by the other: A must be disconnected form C since A is already disconnected from a container of C. Unfortunately, determining the minimal set of constraints is NP-complete (Gottlob, 2011), so we only perform a greedy search.

Avoiding disjunctions There are several ways of encoding a spatial relation in Q_{basic}. To avoid disjunctions, we consider alternative mappings stored in a table and choose the option that introduces the fewest disjunctions. For example, instead of encoding $\neg(A \texttt{ dc } B)$ at the cost of several disjunctions as explained further below, it can simply we rewritten by saying there exist a common point τ, either truly inside or at their border: $(\tau \texttt{ inside } A) \wedge (\tau \texttt{ inside } B)$. Since spatial calculi comprise a jointly exhaustive set of relations, negation can sometimes be rewritten with less disjunctions by considering the mapping of complementary relations.

7.7 Deciding Q_{basic} and Computing Realizations

In this section we introduce two translations of Q_{basic} to LP frameworks, namely mixed integer linear programming (MILP) and AND/OR graphs of LPs. While existing MILP solvers provide all functionality for deciding consistency of a Q_{basic} formula encoded as a mixed integer linear program, we give an incremental method for solving formulas encoded as AND/OR graphs of LPs.

Definition 13. Given an finite set $\mathcal{O}$ and a system of finite disjunctive linear inequalites $S = \langle \mathcal{O}, G \rangle$ we say for a $o \in \mathcal{O}$

$$[o]_S =_{\text{def}} \{o' \in \mathcal{O} \mid G(o') = G(o)\}$$

is the induced congruent set of o with respect to S. In other words, $[o]_S$ collects all oracle variables that lead to the same linear program.

In order to decide satisfiability of a Q_{basic} formula and to obtain realizations for satisfiable formulae, we first perform normalization. First, we rewrite Boolean operators to only have $\vee, \wedge$ and we remove $\top, \bot$ by absorption, e.g., $\phi \vee \bot \mapsto \phi$. Second, negation is moved inward such that we only have negated atoms $\neg S^n_k$. This negated atom can be replaced by a positive one at the cost of introducing disjunctions which select an inequality from S^n_k that is violated. We can thus assume to be given a Q_{basic} formula without negation.

7.7.1 Mapping Q_{basic} to MILP

We base our translation from Q_{basic} to mixed-integer linear programming upon the fundamental work of Balas (1985) on disjunctive linear programming. Further we draw inspirations from S.

Algorithm 2: Translate (normalized) Q_{basic} formula ϕ to Mixed-Integer Linear Program L, with $x, v_* \in \mathbb{R}^n, y_* \in \{0,1\}$

```
L ← empty mixed integer linear program
for all $S \in \phi$ do
    $\mathcal{O}_S \leftarrow \{[o]_S \mid o \in \mathcal{O}\}$
    for all $[o]_S \in \mathcal{O}_S$ do
        chose a $o \in [o]_S$
        $\langle A, b \rangle \leftarrow G_S(o)$
        $L \leftarrow L \cup Av_{S,[o]_S} \leq by_{S,[o]_S}$                    ▷ Add inequalities
        $L \leftarrow L \cup [y_{S,[o]_S} \leq y_S]$                           ▷ Add relation implication
        $L \leftarrow L \cup [y_{S,[o]_S} \leq \sum_{o \in [o]_S} y_o]$        ▷ oracle implication
    end for
                                                               ▷ Aggregate Disjunction Constrains
    $L \leftarrow L \cup [x = \sum_{[o]_S \in \mathcal{O}_S} v_{S,[o]_S}]$
    $L \leftarrow L \cup [0 \leq v_{S,[o]_S} \leq y_{S,[o]_S} U$ for all $[o]_S \in \mathcal{O}_S]$
    $L \leftarrow L \cup [\sum_{[o]_S \in \mathcal{O}_S} y_{S,[o]_S} = 1]$
end for
$\psi \leftarrow$ replace each $S$ in $\phi$ with $y_S$
$\psi_{\text{CNF}} \leftarrow$ conjunctive normal form of $\psi$
for all disjunctive clauses $Q$ in $\psi_{\text{CNF}}$ do
    $L \leftarrow L \cup [\sum_{y_S \in Q} y_S \geq 1]$                       ▷ All $y_S$ are not negated
end for
```

Lee and Grossmann (2000), who describe a method for approximating non-linear disjunctions, which requires upper bounds u_i on all variables.

The general approach for a given disjunction over k sets of linear inequalities ($A_i x \leq b_i$) is that x is disaggregated into $x = v_1 + ... + v_k$ and for each set of linear inequalities a variable $y_i \in \{0, 1\}$ is defined. Then, $A_i v_i \leq b_i y_i$ constitutes the program, replacing the original set of linear inequalities. Choosing $y_i = 0$ effectively disables the inequality and $y_i = 1$ enables it. A further inequality $v_i \leq y_i u_i$ is added, forcing v_i to zero if the inequality is disabled.

In our case, we have a disjunction for each $S \in \phi$ over the oracle values. The only thing left is to ensure that at least one of the disjunctions is active if the corresponding y_S is: $\sum y_i \geq y_S$.

Alg. 2 shows the complete procedure in algorithmic form. If the resulting MILP has a solution, that solution is also a realization of the Q_{basic} formula[1]. Which oracle value was used, can also be read of from the MILP solution. If no solution was found, the Q_{basic} formula is not realizable.

7.7.2 Incremental Expansion of Linear Programms

Considering the parse tree of a Q_{basic} formula, we can regard the formula as AND/OR graph whose leaves are systems of finite disjunctive linear inequalites. In order to compute a solution we perform a depth-first search with backtracking as shown in Alg. 3. The starting parameters

[1] With a lot of extra variables y_i which however can easily be filtered out

calculus	encoding properties
Allen's interval relations	✓
Block Algebra	✓
Cardinal Direction Calculus	✓
Dipole Calculus	discretized 2D directions
INDU	✓
LR calculus	discretized 2D directions
$\mathcal{OPRA}$	discretized 2D directions
Point algebra	✓
Positional point calculi	discretized 2D directions
Qualitative Trajectory Calculi	via encoding to $\mathcal{OPRA}$
Region Cardinal Dir. Calc.	N-vertex polygons or polyhedra only
RCC	N-vertex polygons or polyhedra only
STAR	✓
StarVars	✓

Table 7.1: Encoding properties of qualitative calculi in Q_{basic}

are the original AND/OR tree T, the partial grounding LP encoded in LP[2], and the set of oracle values $\mathcal{O}$. A solution found at a node is propagated upwards, accumulating the (pure) linear programs (line 29). The algorithm either returns a realization, the corresponding LP, and the oracle values or $\emptyset, \emptyset, \emptyset$ to signal unsatisfiability.

7.8 Practical Analysis

We evaluate the performance of the strategies MILP and incremental expansion experimentally. Since our method actually computes a realization for any consistent QCSP instance, comparison with algorithms that merely check for consistency but cannot compute a realization are not adequate. Additionally, both strategies are compared against the results published for StarVars reasoning algorithm(J. H. Lee, Renz, and Wolter, 2013) that also computes a realization. This comparison is particularly interesting since a StarVars requires a large number of oracle values to be introduced and its parameters allow controlling problem size n (number of entities, $O(n^2)$ constraints) and required oracle values ($|\mathcal{O}| = m \cdot n$) independently. For each combination of n and m we randomly generate 100 QCSP instances, using base relations as constraints.

We implemented the translation described in Alg. 1, 2 in Python. For MILP and LP solving, we rely on lp_solve (Berkelaar, Eikland, and Notebaert, 2010). Tab. 7.2 gives compute times measured on an Intel Core i7 @3.4GHz with 16 GB RAM. The results in the column StarVars are those as reported in J. H. Lee, Renz, and Wolter (2013) using a different, slower machine, and are thus not comparable as such, but sufficient for a qualitative comparison.

[2] If not applicable an empty LP is provided

Algorithm 3: Incremental Expansion

```
procedure REALIZETREE($T, LP, \mathcal{O}$)
    if $T_{\text{root}}$ is conjunction then                          ▷ And-Node
        $C \leftarrow$ select one child of $T$
        while $\mathcal{O} \neq \emptyset$ do
            $S, LP', \mathcal{O}' \leftarrow$ REALIZETREE($C, LP, \mathcal{O}$)
            $\mathcal{O} \leftarrow \mathcal{O} \setminus \mathcal{O}'$
            if $T$ has other children then
                $S, LP', \mathcal{O}'' \leftarrow$ REALIZETREE($T \setminus C, LP', \mathcal{O}'$)
            end if
            if $S \neq \emptyset$ then
                return $S, LP', \mathcal{O}''$
            end if
        end while
        return $\emptyset, \emptyset, \emptyset$
    else if $T_{\text{root}}$ is disjunction then                     ▷ Or-Node
        for all children $C$ of $T$ do
            $S, LP', \mathcal{O}' \leftarrow$ REALIZETREE($C, LP, \mathcal{O}$)
            if $S \neq \emptyset$ then
                return $S, LP', \mathcal{O}'$
            end if
        end for
        return $\emptyset, \emptyset, \emptyset$
    else                                                        ▷ Symbol/Relation
        for $T$ induced congruent sets $\mathcal{O}' \subset \mathcal{O}$ do
            $o \leftarrow$ select from $\mathcal{O}'$
            $LP' \leftarrow G_p(o)$
            $S \leftarrow$ SOLVE($LP \cup LP'$)
            if $S \neq \emptyset$ then
                return $S, LP \cup LP', \mathcal{O}'$
            end if
        end for
        return $\emptyset, \emptyset, \emptyset$
    end if
end procedure
```

n	m	StarVars (Lee et al. 2013)	MILP	IncExpand
4	4	0.64 ±0.39	0.02 ±0.01	0.13 ±0.00
4	8	1.15 ±0.71	0.14 ±0.04	0.20 ±0.01
4	16	2.01 ±1.13	1.08 ±0.30	0.34 ±0.01
4	32	2.63 ±1.61	8.26 ±2.67	0.62 ±0.02
5	4	1.06 ±0.58	0.06 ±0.01	0.19 ±0.01
5	8	1.66 ±1.26	0.44 ±0.10	0.30 ±0.01
5	16	2.56 ±2.14	4.72 ±1.15	0.50 ±0.02
5	32	4.35 ±3.87	34.11 ±9.91	0.92 ±0.02
6	4	2.55 ±0.00	0.14 ±0.03	0.27 ±0.01
6	8	3.16 ±1.64	1.31 ±0.27	0.42 ±0.02
6	16	4.27 ±3.48	12.96 ±2.91	0.69 ±0.03
6	32	6.10 ±5.88	109.46 ±24.15	1.25 ±0.04
7	4	6.83 ±0.10	0.28 ±0.05	0.37 ±0.01
7	8	7.55 ±0.10	3.23 ±0.55	0.55 ±0.02
7	16	8.30 ±1.82	36.21 ±8.03	0.91 ±0.03
7	32	8.76 ±2.68	310.05 ±73.63	1.61 ±0.02

Table 7.2: Compute time in seconds with standard deviation for 100 random scenarios for n entities with m distinct orientations

Discussion of the Results

Let us first consider compute times for MILP shown in Tab. 7.2. The time increases with problem size and, more significantly, with respect to m. This likely results from the translation into MILP since unfolding disjunctions leads to exponential problem size. The steep scaling wrt. m also leads to longer compute times than reported for the handcrafted StarVars algorithm on a slower machine.

For problems with few disjunctions (e.g., $m = 4$), MILP can outperform incremental expansion.

For most of the configurations tested, incremental expansion shows superior performance though. This is due to the algorithm exploiting the structure of the formula, something that gets lost in the translation to MILP. In comparison to the results obtained for the original StarVars algorithm handcrafted for these constraints, we observe a similar scaling with respect to increasing m.

In summary, incremental expansion provides a practical method for reasoning with Q_{basic} formulae.

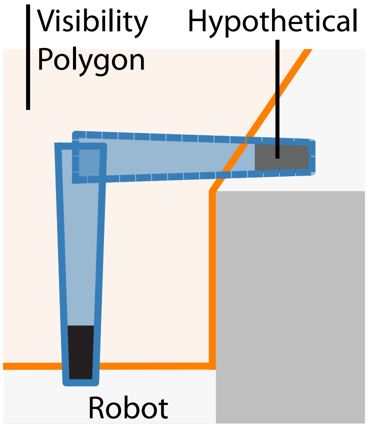


Figure 7.3: Realization computed by our algorithm when provided with shape of breaking regions, outline of the obstacle (grey box), and $\psi'_{\text{dangerous}}$ from Section 7.1.1

7.9 Summary and Conclusion

This paper outlines a practically relevant answer to two longstanding questions in qualitative spatial and temporal reasoning. By encoding spatial and temporal relations into an LP framework, we are able to represent the important domains of points, lines, and polygons. We show how relations from various qualitative calculi can be expressed in our framework, including directional knowledge. This allows distinct qualitative representations to be combined and jointly to be reasoned about. Doing so, we advance earlier work in temporal reasoning by Jonsson and Bäckström (1998). The algorithm of incremental expansion for solving AND/OR LP problems is however more efficient than using disjunctive linear relations like in their work, since incremental expansion avoids exponential blow up of disjunctions occurring with disjunctive linear relations or MILP. While this paper proposes the unifying language Q_{basic} that can be tackled with LP techniques, identifying the most efficient reasoning algorithms is subject to further investigations.

References

Balas, Egon (1985). "Disjunctive programming and a hierarchy of relaxations for discrete optimization problems". In: *SIAM J ALGEBRA DISCR* 6.3, pp. 466–486.

Berkelaar, Michel, Kjell Eikland, and Peter Notebaert (May 2010). *lp_solve 5.5, Open source (Mixed-Integer) Linear Programming system*. Software. Available at `http://lpsolve.sourceforge.net/5.5/`. Last accessed April, 2013.

Bhatt, Mehul, Jae Hee Lee, and Carl Schultz (2011). "CLP(QS): a declarative spatial reasoning framework". In: *Proc. of Cosit*, pp. 210–230.

Dylla, Frank, Till Mossakowski, Thomas Schneider, and Diedrich Wolter (2013). "Algebraic Properties of Qualitative Spatio-Temporal Calculi". In: *Proceedings of the Conference On Spatial Information Theory (COSIT-13)*. Vol. 8116. LNCS. North Yorkshire, UK: Springer, pp. 516–536.

Gottlob, Georg (2011). "On Minimal Constraint Networks". In: *Principles and Practice of Constraint Programming, CP 2011*. Ed. by Jimmy Lee. Vol. 6876. LNCS. Springer Berlin Heidelberg, pp. 325–339. ISBN: 978-3-642-23785-0.

Hué, Julien, Matthias Westphal, and Stefan Wölfl (2012). "An Automatic Decomposition Method for Qualitative Spatial and Temporal Reasoning". In: *Proc. ICTAI-2012*. University Freiburg, SFB/TR 8, pp. 588–595.

Jonsson, Peter and Christer Bäckström (1998). "A unifying approach to temporal constraint reasoning". In: *Artif. Intell.* 102.1, pp. 143–155.

Kontchakov, Roman, Ian E. Pratt-Hartmann, Frank Wolter, and Michael Zakharyaschev (2010). "Spatial Logics with Connectedness Predicates". In: *LOG METH COMPUT SCI*.

Lee, Jae Hee, Jochen Renz, and Diedrich Wolter (2013). "StarVars—Effective Reasoning about Relative Directions". In: *IJCAI*. Ed. by Francesca Rossi. IJCAI.

Lee, Sangbum and Ignacio E. Grossmann (2000). "New algorithms for nonlinear generalized disjunctive programming". In: *Computers and Chemical Engineering Journal* 24.9, pp. 2125–2141.

Li, Sanjiang, Weiming Liu, and Shengsheng Wang (2013). "Qualitative constraint satisfaction problems: An extended framework with landmarks". In: *Artificial Intelligence* 201, pp. 32–58.

Ligozat, Gérard (2011). *Qualitative Spatial and Temporal Reasoning*. Wiley. ISBN: 978-1848212527.

Mossakowski, Till and Reinhard Moratz (2012). "Qualitative Reasoning about Relative Direction of Oriented Points". In: *Artif. Intell.* 180–181, pp. 34–45.

Randell, David A., Zhan Cui, and Anthony G. Cohn (1992). "A Spatial Logic based on Regions and "Connection"". In: *Proc, of KR'92*. Ed. by Bernhard Nebel, Charles Rich, and William Swartout. San Mateo (CA), USA: Morgan Kaufmann, pp. 165–176.

Renz, Jochen and D. Mitra (2004). "Qualitative Direction Calculi with Arbitrary Granularity". In: *Proc. of PRICAI-04*. Vol. 3157. LNAI. Springer, pp. 65–74.

Renz, Jochen and Bernhard Nebel (2007). “Qualitative spatial reasoning using constraint calculi”. In: *Handbook of Spatial Logics*. Ed. by Marco Aiello, Ian E. Pratt-Hartmann, and Johan van Benthem. Springer, pp. 161–215. ISBN: 978-1-4020-5586-7.

Täubig, Holger, Udo Frese, Christoph Hertzberg, Christoph Lüth, Stefan Mohr, Elena Vorobev, and Dennis Walter (2012). “Guaranteeing functional safety: design for provability and computer-aided verification”. In: *Autonomous Robots* 32.3, pp. 303–331.

Wölfl, Stefan and Matthias Westphal (2009). “On Combinations of Binary Qualitative Constraint Calculi”. In: *Proc. of IJCAI*, pp. 967–973.

Wolter, Diedrich (2012). “Analyzing Qualitative Spatio-Temporal Calculi using Algebraic Geometry”. In: *Spatial Cognition and Comp.* 12.1, pp. 23–52.

Wolter, Diedrich and Jae Hee Lee (2010). “Qualitative Reasoning with Directional Relations”. In: *Artif. Intell.* 174.18, pp. 1498–1507.

8 Conclusion

This chapter concludes my cumulative dissertation with a brief summary, a critical assessment of the approach and in particular the assumptions made. As every end is the start of something new, this chapter also includes an outlook towards possible future work, and closing with some final thoughts.

8.1 Summary

Four research questions were presented in detail in the introduction. In the following I will summarize my answers to each of these four research questions. To answer these research questions, I have developed And/Or Linear Programming (And/Or LP; Chapter 7) and introduced two new logics: Conceptual Neighborhood Logic (CNL; Chapter 5) and QLTL (Chapter 6).

How can qualitative calculi be combined, i.e. how can one jointly reason with knowledge represented in distinct calculi?

And/Or Linear Programming (And/Or LP; Chapter 7) allows to map formulas of different (spatial) aspects into a single unified language. The basic assumption underlying And/Or Linear Programming is that non-linearities can be approximated by a set of systems of linear equalities and inequalities. If the set is finite, then all of these systems could (theoretically) be checked, one by one. However, the size of the set of systems of linear equalities and inequalities renders such an exhaustive search impractical. Consequently, a heuristic should be used that either directs towards a satisfiable set, or that can exclude subsets without checking all of its members.

The current heuristic used in And/Or LP is based on pruning the search space by coarsening qualitative relations (Chapter 5 and Chapter 7). Each coarsening adheres to the following property: If no solution can be found for a coarsened system of linear inequalities, then for the original system no solution can be found either. Therefore, the search space can be pruned, whenever no solution can be found. In case a solution is found, the relations are split into finer granular relations, again resulting in a set of systems of linear equalities and inequalities. This deepening is repeated until either no solution is found or the desired level of granularity is reached, in which case the solution is returned.

In the current implementation a depth-first approach is used, which fixes single relations one after the other, using the above described heuristic. In case no intermediate solution is found, backtracking is used.

How can qualitative representations incorporate grounded information, i.e. how can free-ranging and constrained variable domains (singleton, finite, numeric constraints) be mixed?

An assumption in qualitative spatial reasoning is that all variables range over the same domain. However, in applications it is often required to restrict the domain of specific variables, such as restricting a solution to fit into a given floor plan. Such a floor plan can be represented as a polygon fixed in a (local) coordinate system. The free space can than be described by disjunctive sets of linear inequalities, representing a convex partition of the polygon. Robotic applications generally describe—or at least approximate—shapes with polygons. In contrast to a ground floor, other entities represented by a polygons are not necessarily fixed at a single location. To express that an entity is limited to a finite set of locations, the locations can be described as sets of equalities. As systems of equalities and systems of inequalities both are fundamental building blocks of And/Or LP, Therefore, they can be added to the And/Or LP tree resulting from a formula with no change of the reasoning method.

In Chapter 5 and Chapter 7 we demonstrate the encoding and incorporation of such kind of background knowledge. Further, we also describe, that a single spatial region, such as a ”breaking area”, can also have different shapes. While the breaking area depend on the current speed (category) of the vehicles, the formula, that no two breaking areas are allowed to overlap, holds regardlessly. This eases the burden on the modeling and ensures the transferability of the modeled knowledge. In summary, And/Or LP can directly reason with specific instances and qualitative descriptions in one unified manner.

How can a prototypical pictorial representation be derived from a (pure) qualitative description of a scene?

One result presented in Chapter 4 is the generation of high-level *counter examples*. Such counter-examples are hard to interpret when only presented as a logical formula. Provided that a qualitative description is realizable, then by calculating a solution to the And/Or LP tree all variables have a real valued assignment. Consequently, as every part is fully specified, i.e. has a fixed position, a known orientation, and so on, we can draw a pictorial representation of the qualitative description.

How can a spatial logic and Linear Temporal Logic be combined to yield a decidable formalism, that can be applied to various applications?

Regarding time, two aspects are prevailing, first, how to apply a temporal formula to control a robotic system, and second, how to think about time during modeling. Time can be viewed as branching (CTL,), emphasizing on what could be, whereas time viewed as linear (LTL, Pnueli, 1977) stresses what should or even has to happen. While CTL has an overall polynomial computational complexity, LTL has an overall computational complexity that lies in P-SPACE. However, if the formula size is small compared to a large state space—as is typical in applications—the formula size can be viewed as fixed, resulting in a computational complexity in P with respect to the state space size. Further, LTL can be applied to individual histories or runs without the full state space, whereas CTL cannot. Throughout the presented applications time is regarded as linear, as the applications in this thesis are process detection in Chapter 2, controlling a sailboat in Chapter 3, motion planning in Chapter 5, and acting according to social conventions in Chapter 6.

Linear Temporal Logic (LTL, Pnueli, 1977) requires a structure, from which the individual runs can be generated. I choose to use Topological Mode Spaces (Galton, 2000) rather than *pure* Conceptual Neighborhoods (Freksa, 1991). Conceptual Neighborhood is defined on the basis of relations, whereas a product of Topological Mode Spaces can be computed and consequently enables reasoning with scenarios (Section 5.5.2 on page 94 and for more details Galton, 2000, page 359). Conceptual Neighborhood Logic (CNL; Chapter 5) and QLTL (Chapter 6) use the accessibility relation induced by the perturbation relation of the Topological Mode Spaces. Conceptual Neighborhood is accessible in CNL through its own modal operator. Using conceptual neighborhood, a spatial configuration can be described indirectly, e.g., to describe a spatial configuration that is conceptually neighbored to another configuration in which a collision occurs. However, we found that in most applications this is not needed; therefore we introduced QLTL. QLTL allows the direct usage of LTL model checkers without any need for preprocessing. Further, QLTL has a simpler syntax and its name clearly shows the combination it stems from, namely $\mathrm{Q}_{\mathrm{basic}}$ (Chapter 7) and Linear Temporal Logic.

In summary, I developed And/Or LP and two new logics: CNL and QLTL. Both logics are based on LTL and And/Or Linear Programming. Further, CNL also allows to include *conceptual neighborhood* when describing a qualitative situation. The applicability of QLTL and CNL are demonstrated as high-level spatial rule systems for control and supervision of robotic systems (Chapter 5 and Chapter 6).

8.2 Discussion

In this discussion I will reflect on four aspects touching my work: 1) modeling aspects, 2) theoretical as well as 3) application considerations, and 4) the transferability of my results.

8.2.1 Modeling Aspects

> "We think in generalities, but we live in details."
>
> — Alfred N. Whitehead

Humans and computers differ in nature, especially in their sensory experience (e.g., human eye vs. laser range finder) and reasoning methods (large scale neural networks vs. binary arithmetics). Consequently, in the sense of constructive epistemology (Piaget, 1967), robots and humans have different models about the world. For effective communication between humans and computers to take place, a common language is required. As in all forms of communication[1] misunderstandings happen due to different assumed background or because the same proposition is interpreted differently. It is therefore important to look at the details when communicating about space.

In the following, I will examine the challenges that arise when translating human concepts to formal semantics, whether modeling of behavior should be viewed as sequences of configurations or as sequences of actions, and finally how representations are a bridge between high-level descriptions and low-level control.

From human concepts in language to formal semantics

To establish navigation conventions to be followed, such conventions have to be communicated to each participant. For human communication *language* is fundamental. Consequently, language and its use received substantial research also from the perspective of artificial intelligence. Nevertheless, even seemingly simple statements are still not comprehensible with state-of-the-art natural language processing methods, as is evident by the *Winograd Challange* proposed by Levesque, Davis, and Morgenstern (2011). One such challenging statements is the following:

> The book does not fit into the box, because *it* is too *large*.
> What is too *large*?

The question cannot be answered purely by syntactical or statistical means, as one can just replace *large* with *small* and thereby change what *it* refers to. Indeed, answering the question calls for a spatial understanding of *fit into* as well as an ontological understanding, as something can also be *too small to fit*, for example: *this bolt is too small to fit into this nut*.

Bateman et al. (2010) therefore propose a two-level architecture to language understanding. The first layer uses a *linguistic ontology* and represents the “pure semantics”, whereas the second layer than uses the context and background knowledge to entail more information and to provide an interpretation. Moreover, Bateman et al. provide first steps towards an ontology of space for natural language processing based on qualitative spatial reasoning, which closely resemble human understanding of space (Knauff, Rauh, and Renz, 1997; Klippel and Montello, 2007). Tenbrink and Ragni (2012) extend upon the work of Bateman et al. while focusing on relational reasoning and route instructions. They identify systematic patterns and advocate the use of spontaneous rather than artificial spatial relations, which have been predominantly used in previous studies. However, such spontaneous spatial relations are beyond state-of-the-art natural language processing, as even with carefully selected artificial spatial categories, ambiguities arise in human-generated spatial descriptions.

To deal with this problem, the first question to solve is, what distinctions—when describing topological spatial configurations—do humans usually make? Klippel, Wallgrün, et al. (2013)

[1] The only exception being possibly the language of math.

analyze this question for configurations of polygonal regions, especially when regions overlap or contain one another. They find that usually the relations provided by the RCC-5 variant are all that is needed. Klippel, Wallgrün, et al. check further, whether language or cultural difference influence to which degree someone uses a more detailed description. Therefore they conducted their study with Chinese-, Korean-, and English-speaking participants. While some individuals—especially Korean[2]—did use finer granular descriptions[3], a language induced difference in the used granularity of topological relations cannot be found.

However, for directional reasoning, a cultural difference can be found. Haun et al. (2011) compared the performance of various spatial rotation tasks between Dutch and Namibian elementary school children. The tasks are selected to be more easily solvable by either egocentric or geocentric representation. $\neq$Akhoe Hai$||$om is the language spoken by the Namibian children and in this language geocentric representations are used, whereas the Dutch elementary school children predominately use egocentric descriptions. Haun et al. (2011) argue that the differences found in the performance of the school children are of preferential nature and not based on absolute capabilities. However, a (meta-)neuroimaging study by Galati et al. (2010) shows that different cortical regions are active for representing spatial locations of objects as either egocentric or geocentric. To summarize, when designing a domain-specific language for spatial rule systems, it is important to also consider the language and cultural background, so that none of these key concepts are missing. Consequently, my approach supports all of these key concepts.

Regardless of language and culture, misunderstandings occur due to different background knowledge, different interpretations of the same propositional symbol, or due to the ambiguity inherent in natural language. This is especially true in human-computer interaction, as the computer has a crisp understanding of the qualitative representations, whereas a human situationally broadens her interpretation. Even though different interpretations of the same facts are important to be able to explain stock-market trades (Halpern and Kets, 2012) and creativity (Atchley, Keeney, and Burgess, 1999), they should be avoided in case of perilous situations.

> “[...] much of human cognition is domain-specific.”
>
> — Hirschfeld and Gelman (1994, page 3)

Resolving ambiguities and misunderstandings requires some form of reasoning and generally involves the (current) context and domain. Klippel, Yang, et al. (2012) show, that human-perceived similarities of spatial relations is dependent on the domain. Consequently, when resolving misunderstandings or contradicting information, cognitive inspired distance measures should be developed. Such distance measures are sometimes not well represented by conceptual neighborhood alone, e.g. there are more people that confound left and right than there are people confounding front and right. However, the distance induced by conceptual neighborhood for front and right is less than the distance used for left and right . Context and domain have therefore to be taken into account. QLTL is designed to allow for as much context, i.e.

[2] The Korean language offers very fine granular distinctions for overlapping relations.

[3] some even going beyond RCC-8

constraints on the variables, and as much domain-dependent knowledge as possible without sacrificing decidability (c.f. Section 8.2.2). As QLTL is a multi-sorted logic different coarsening operations can therefore be directly implemented based on the sorts of the arguments.

Currently, translations from natural language to QLTL formulas are done by hand. Kordjamshidi et al. (2011) argue, that the translation from natural language to QSR could generally be learned, even though they only demonstrate the learning of a translation from natural language to RCC-8.

Besides representing spatial natural language interpretation using qualitative spatial reasoning, other approaches have been considered as well, such as spatial description clauses. Spatial description clauses are composed of the *subject*, an *action*, a *landmark*, and a *spatial relation*, where any of the fields can also be specified only indirectly. For example, Kollar et al. (2010) translate natural language route instruction to spatial description clauses. From these clauses, a probabilistic graphical model is computed as basis for the actual route instruction of a robotic system. Free form route instructions might use clues that the robot cannot identify, e.g. "pass the *computers*". Therefore, co-occurrences are computed with objects the robot can identify, e.g. monitors, which are than used instead. Tellex et al. (2011) translate (spoken) natural language commands given to a robotic system for navigation and manipulation instructions to spatial description clauses as well. Again, based on these a probabilistic graph model called generalized grounding graph is established and used to identify confusing parts of a command. These confusions are than provided to a dialog system, which asks the operator to disambiguate. Another possibility to represent spatial relations are semantic fields (O'Keefe, 2003) which bear some similarities to probability density functions. Fasola and Mataric (2013) extend these semantic fields to allow for *dynamic* spatial relations and demonstrate that these can adequately be used as a basis for understanding natural language instructions to a service robot. Both, semantic fields and spatial description clauses, are used with probabilistic reasoning methods as opposed to qualitative spatial reasoning, which employes constraint- and logic-based reasoning.

As the application I have in mind is that of *provable* safety, sound and complete logic formalism should be used wherever possible. Consequently, I choose to use qualitative spatial reasoning as the representation for natural language descriptions of spatial configurations. No matter which representation is chosen, the aforementioned difficulties of misunderstandings and ambiguity are inherent in natural language and have always to be considered when translating or verifying a translation from natural language to a computer comprehensible representation.

One last aspect is: can qualitative spatial representations be represented spatially? We show that And/Or Linear Programming can also be used to fully ground a qualitative description and draw a pictorial representation of the qualitative description. The intention of these pictorial representations are of diagrammatic nature, i.e. they should be depicted in a clear way and avoid any ambiguities. However, as the numeric solver is based on the simplex algorithm, objects are "pushed to the boundary" of the relations and consequently, the picture does contain some ambiguity when *just looked at*. Gottfried (2012) would therefore argue that the pictorial representations are sketches and not diagrams. However, an artificial epsilon could be added to all inequalities to "push the objects inwards", removing the possible ambiguity and turn the pictorial representation into a diagrammatic representation.

In summary, correlating human concepts to formal semantics is a multi-layered research problem, even if restricted to space alone. Qualitative spatial representations are situated at the layer of *pure semantics*. However, humans use the same concepts when describing specific spatial situations that they use to describe abstract configurations. In this dissertation, I focused on reasoning in specific instances rather than on abstract configurations, as is usually done in QSR.

Configuration-oriented modeling vs action-based modeling

Usually a notion of action is involved when describing (spatial) behavior, such as *she turned left*, rather than configurational descriptions, *he first faced towards north and than towards west*. However, what does *turn left* mean? Turning on the spot or moving along a left-curving arc? Further, how much to turn, 5° or 77°? If a (continuous) set of possible actions is identified with *turn left*, which specific action should then be chosen? To be able to reason or even plan with these actions the pre- and post-conditions of the action have to be specified. Given that different robotic platforms have different (movement) capabilities, such definition are specific to each robotic system. As a result, the actions used in high-level deliberative systems are rather complex behaviors in themselves, such as, *move through the door*, than low-level driving commands. However, each such complex behavior requires its own low-level controller.

An action is intended to result in a change of the world, e.g. displacing the robot itself. To be able to *sense* whether the action was executed successfully, the change has to lead to a different symbolic world description. Dylla (2008) introduces action-augmented conceptual neighborhood, extending conceptual neighborhood relations by annotating which actions can induce such (qualitative) change. However, Dylla points out several difficulties due to the abstraction of the qualitative relations 1) more than one action could be selected to induce the same change, 2) executing a single action might have more than one possible outcome[4], and 3) the execution of an action might not change the qualitative relations at all. Consequently, combining actions and qualitative descriptions can lead to non-deterministic post-conditions and therefore require rather complex planning mechanisms.

Nevertheless, Dornhege et al. (2009) first generate a sequence of qualitative relations a robotic manipulator should pass through. Second, a probabilistic roadmap planner is used to generate the actual movement commands. However this planner is confined to the space described by the sequence of qualitative relations. In summary, Dornhege et al. use the probabilistic planner to explore the high-dimensional control space with the guidance of a sequence of qualitative configurations situated in physical space.

Extending upon this idea, we describe the collision avoidance behavior of an autonomous sailing vessel using sequences of qualitative descriptions (see Chapter 3). In contrast to Dornhege et al. (2009), who generate complete qualitative descriptions, our sequences are only partial descriptions of the world and therefore can be applied to different situations and

[4] For example, if a small circle is expanding inside a large circle, a transition from the RCC-8 relation *non-tangential-proper-part* to either *tangential-proper-part* or to *equal* can occur, depending on the location of the inner circle.

a combination of such sequences is also possible. However, all intermediate steps and the possible deviations were modeled originally. We later relaxed this requirement to only the key points in this sequence, allowing more flexibility for the low-level planning modules. For example, instead of requiring that the robot has to turn right in a head-on situation, we simply require that both participants pass each other on their left side.

Another important property of modeling behavior as sequences of (partial) key configurations is that the local context can be integrated without requiring the knowledge engineer to foresee all special cases. Such a special case is the head-on situation in which only one participant can make an evasive maneuver, as the movement of the other is restricted, e.g. by a wall. The action-based approach would either fail or would require this special case to be modeled. Whereas the configuration-oriented modeling approach would handle such a case quite fine.

Most navigation rule systems are developed as if they only involve two participant. In case more than two participants are involved the rules have to be combined. However, combining or interleaving actions can be difficult—especially the combination—as the possible outcome of combined actions might not be specified. Rules requiring certain configurations, can be combined, as the system can compute a qualitative description that combines all required partial configurations. If no such combination can be found, than the rules cannot be jointly executed, something that my approach can detect during the modeling process.

Overall, modeling only the key configurations in a sequence of relations, allows more situational context to be integrated, reduces the modeling requirements, and eases the combined execution of different rules. Consequently, I chose to follow this paradigm throughout my dissertation.

Transition from high-level descriptions to low-level control

> “Science is what we understand well enough to explain to a computer. Art is everything else we do.” —Donald Knuth[5]

Galindo and Saffiotti (2013) demonstrate that complex applications require a multitude of reasoning capabilities. For example, Rost, Hotz, and Von Riegen (2012) combine various methods such as RCC-8, CDC, OWL, Prolog, and complex-event processing. However the theoretical foundations for such combinations remain unclear. Bhatt (2010) integrated qualitative spatial reasoning into the situation calculus for commonsense reasoning. He introduces $\mathcal{C}$-Consistency, which requires that the relations are JEPD, that transitivity, symmetry and asymmetry of relations are known, that algebraic closure decides constancy and that the interdependencies between the used calculi can be axiomatized. Nevertheless, $\mathcal{C}$-Consistency only enforces physically plausible but not necessarily physically realizable spatial descriptions.

Suchan (2011) developed in his diploma thesis ExpCog, a framework for cognitive robotics based on Bhatt’s work. ExpCog is also built around fully axiomatized actions and suffers from the same difficulties as mentioned in the previous section. Further, hardly any combination of qualitative spatial calculi can have their interdependencies axiomatized as required for the

[5] in the forword for “A = B” by Petkovšek, Wilf, and Zeilberger (1996)

$\mathcal{C}$-Consistency. Additionally, consistency of relative direction calculi cannot be decided by algebraic closure (see Wolter and J. H. Lee, 2010). As a result, Bhatt, J. H. Lee, and Schultz (2011) develop a new underlying integration. In contrast to the previous approach of directly using the Situation Calculus, Prolog is used, combined with geometrical reasoning based on multivariate polynomial inequalities. While reasoning about relative directions is therefore possible within this new framework, the spatial reasoning part alone has already a double exponential complexity.

Whether actions or configurations are planned, they still have to be executed, i.e., motor driving commands have to be issued. One way to calculate motor commands is to use randomized planning approaches based on simulations, such as probabilistic roadmap planners. As Galindo and Saffiotti (2013) demonstrate, robotic platforms are quite diverse in their capabilities, but if a common language is found, knowledge can be exchanged. Telling a robotic system meticulously how to perform a complex task always requires specific knowledge about the capabilities of the robotic system. However, describing a desired goal can be expressed independent of the robotic system. Therefore, instead of telling a robot how to clean up an environment, Galindo and Saffiotti (2013) provided the robotic system with the knowledge what a clean environment looks like. The robot is then assigned with identifying normative violations and with finding a plan to recover from such violations, i.e. cleaning up. Galindo and Saffiotti (2013) cite: “The European project RobotEarth goes one step further and uses ontologies not only to allow a robot to perform new inferences, but also to enable meaningful communication among heterogeneous robots [Waibel et al., 2011].”

As simple waypoint navigation can be regarded as a basic service in robotic frameworks, spatial rule systems specified in QLTL or CNL can be directly used during planning, as shown in Chapter 5. I follow this paradigm shift in my dissertation, from direct control of actions to robot-independent representations of how the environment should be.

8.2.2 Theoretical Considerations

In Section 8.2.1 I present arguments for symbolic spatial reasoning from a modeling point of view. Further, I argue for a configuration-based rather than an action-based modeling approach. In the present section I discuss these questions from a computational point of view.

Numeric vs. symbolic reasoning

From a computer science perspective, one question that arises is: *What can be expressed and reasoned about on the symbolic level and what needs to be done numerically*? Not only when considering low-level control as in Section 8.2.1, but also when considering context, such as floor plans, the need to include non-symbolic constraints becomes evident. Such non-symbolic constraints can be numeric or more generally restrictions to the domain as a whole or on a per variable scope.

One well researched qualitative spatial calculus is the region connection calculus (RCC) by Randell, Cui, and Cohn (1992) and its variants. Deciding consistency of RCC-8 over general topological spaces is NP-hard, but, restricting the domain can lead to an undecidable formalism. Haarslev, Lutz, and Möller (1998) combine RCC over quantitative fixed polygons with description logic to allow for spatioterminological reasoning. To achieve a decidable reasoning method, Haarslev, Lutz, and Möller make severe restrictions on the language, resulting in difficulties when modeling knowledge or queries. Kontchakov, Nenov, et al. (2011) investigated the computational complexity of RCC over specific domains: "[...] the logic with the interior-connectedness predicate (and without contact) is undecidable over polygons or regular closed sets in $\mathbb{R}^2$, ExpTime-complete over polyhedra in $\mathbb{R}^3$, and NP-complete over regular closed sets in $\mathbb{R}^3$". S. Li and Ying (2004) observe that digital recordings implicitly use a discrete representation of space. Therefore, S. Li and Ying introduced the *Generalized Region Connection Calculus* which is based on RCC and Galton's theory for discrete spaces. Whether it is decidable and if so, into which complexity class it falls, is still unclear.

Generally approximating solutions to qualitative spatial constraint networks has been researched as well, for example by J. Li and S. Li (2013). Using conceptual neighborhood graphs as distant measure, one of the methods presented by J. Li and S. Li is the relaxation of relations during algebraic closure instead of backtracking. They present a total of four different methods for general approximative reasoning about qualitative constraint networks, showing that approximative-reasoning techniques have to be chosen with respect to the intended application.

Reasoning with relative directions, such as left and right, also falls into an unfavorable complexity class. J. H. Lee (2014) proved that reasoning with relative directions is $\exists\mathbb{R}$-complete[6]. As a result, J. H. Lee developed StarVars, a qualitative calculus that can approximate other relative direction calculi. The fundamental idea behind StarVars, using finitely many disjunctions of linear programs, inspired my development of And/Or LP.

All these challenges only concern reasoning using a single calculus. Combining different calculi to reason about various aspects of space simultaneously is another challenge altogether. Wölfl and Westphal (2009) define two types of algebraic couplings: loose and tight integration of calculi. While the loose coupling can result in formalisms too weak to carry out complete and sound reasoning, the tight integration is basically the development of a new calculus.

Instead of combining the calculi on a symbolic level, they can be translated into a *common language*, which is the approach taken in this dissertation. I chose to use a language based on linear equalities and inequalities, and show how different calculi can be translated into this common language. As a result, I can jointly reason with different calculi, as they are translated into a common *language*.

And/Or LP uses a numeric solving method and has consequently also some constraining requirements. As Kontchakov, Nenov, et al. (2011) have shown RCC is undecidable over the set of all possible polygons in $\mathbb{R}^2$, I require that all polygonal shapes are known beforehand. Further, based on the findings by J. H. Lee (2014), I restrict the possible orientations of the polygons to a finite set. However, no restrictions are enforced on the positions of the polygons. In Chapter 5 we prove that reasoning with And/Or LP is in NP.[7] Moreover, using a numeric

[6] $\text{NP} \subset \exists\mathbb{R} \subset \text{PSPACE}$

[7] Based on the results of J. H. Lee (2014) we conjecture that And/Or LP also falls into $\exists\mathbb{R}$

approach allows to base reasoning on the physical sensor readings of a robot.

In summary, while reasoning on the symbolic level can become undecidable, some tasks are well suited for the symbolic level, e.g. computing a minimal qualitative constraint network or sub-graph matching. Numeric reasoning is decidable in the (approximative) domain, however solving speed can suffer from redundant constraints. As we show in Chapter 5 we start with a symbolic description and use various symbolic methods to optimize this description before translating it to the numeric representation, thereby using the best of both approaches.

QLTL vs Action Logics

In Section 8.2.1 the argument whether to model with actions or with spatial configurations to describe a spatial rule system is discussed from the modeling point of view. Here I will discuss this question from a rather theoretical reasoning point of view. In the following I will discuss the difficulties of both approaches, direct combination of QSR and temporal logics and using action logics. For the latter I will use the Situation Calculus based Golog (Levesque, Reiter, et al., 1997) to exemplify the difficulties.

Balbiani and Condotta (2002) show that propositional linear temporal logic based on qualitative spatial or temporal reasoning is PSAPCE-complete. However, two limitations exist: Only a single calculus at a time can be used and such a calculus has to have the property that "Consistent networks of atomic constraints are globally consistent" (Balbiani and Condotta, 2002). Not all spatial calculi exhibit this property, for example, the calculus to reason about line segments[8] developed by Moratz, Renz, and Wolter (2000). As Kontchakov, Kurucz, et al. (2007) show a more general combination easily leads to an undecidable formalisms.

In contrast, change can also be viewed purely qualitatively. Freksa (1991) proposes the notion of Conceptual Neighborhood to describe possible change. This approach is refined by Galton (2000) into Topological Mode Spaces. Westphal et al. (2013) extend upon this idea and introduce a new *qualitative change constraint* (transition constraint) to allow a unified constraint satisfaction problem approach. However, complexity goes to NP-complete even if static descriptions are tractable. A different approach, and the one taken in this dissertation, is to introduce modal operators instead of transition constraints to capture the meaning of qualitative change (see Chapter 5).

Action-based high-level control of robotic systems have been extensively researched. Golog (Levesque, Reiter, et al., 1997) is one well known high-level deductive reasoning robotic control framework that is based on the Situation Calculus developed by McCarthy (1963). However, the situation calculus in its general form is undecidable, even with respect to model checking. Consequently, variants have been proposed to reduce the full situation calculus in such a manner as to become decidable.

J. Lee and Palla (2012) showed how the situation calculus with finite action horizon and finite recursion can be embedded into the stable model semantic. Further, they provide a modeling of this variant of the situation calculus within Answer Set Programming. De Giacomo, Lespérance,

[8] DRA_{24} (dipole relation algebra)

and Patrizi (2012) restrict the situation calculus in a similar way, by bounding the number of fluents that can simultaneously be true. They propose, that some fluents are modeled as fading fluents, i.e., forgetting fluents. Additionally, they prove the decidability of this variant but do not state into which complexity class such an approach falls. Cabalar and Santos (2011) developed a similar variant based on first-order Equilibrium Logic, which is a generalization of the stable model semantics. To achieve decidability of model checking for the bounded situation calculus, De Giacomo, Lespérance, Patrizi, and Vassos (2014), enforce two restrictions: first, they restrict to a single agent, and second, they forbid quantifiers across situations. The latter restriction leads to a rather loose coupling of time/actions and situations.

Baader and Zarrieß (2013) approach the verification of Golog programs by a different route: instead of restricting the situation calculus they replace it altogether by a description logic of actions. In order to obtain a finite, semantics-preserving abstraction of the infinite transition system induced by a program, Baader and Zarrieß introduce a new notion of dynamic type. While this approach leads to a decidable formalism, it is only a subset of the original Golog and it requires deterministic actions.

In summary, to achieve decidability, it seems critical to have a rather loose coupling of time/actions and space. As a consequence I integrate temporal and spatial reasoning loosely, as discussed in Section 8.2.1 and Chapter 5, resulting an a *decidable* formalism that is expressive enough for various applications.

8.2.3 Application Considerations

Concerning the question of whether rule compliance should be handled by integration or by supervision, I argue that planning should be able to take the rules into account, even to such a degree as to exploit them. In contrast, a second system should be in-place to catch errors made during planning. This second system checks whether the actions about to be executed are in correspondence with the rules, based on the current context. However, emergency behavior generally falls outside of such safety rules and could even contradict them.

Outside the rules: emergency behavior

So far, two basic assumptions have been made with regard to safety. First, all participants follow the rules, and second, *hitting the breaks* is the primary/exclusive form of emergency behavior. In contrast, what happens if the rule conformity assumption is violated? Assuming a rule system that covers all cases and is contradiction-free, what happens if a participant does not follow these rules? An example is a pedestrian that suddenly crosses at an intersection despite a red light, clearly violating the traffic rule. Such violations require special emergency behavior. Generally, only *hitting the breaks* might not be the most desired behavior in such cases, but an evasive maneuver might be. In the example of the pedestrian, if the car could avoid the pedestrian by moving into the opposite lane, should it?

What kind of evasive maneuver is taken highly depends on the context, such as the surrounding traffic. In general, evasive actions violate the rule system and therefore a supervision system would not allow these evasive maneuvers. If the robot is to automatically decide about the appropriate actions to take, it not only has to be *aware* of the rules but to *understand* the *intentions* of the rules. However, how should an algorithm decide whether to crash into a pedestrian or an oncoming bicycle? Such ethic questions are beyond the capabilities of artificial intelligence and even go beyond human ethical decision making.

Nevertheless, this approach can be used to define a reasonable set of special situations and check how the robotic system would act. Further, the approach I developed can check whether the allowed emergency behaviors cover all specified cases.

8.2.4 Applying And/Or LP reasoning beyond QLTL

In this section I will present the universality of And/Or Linear Programming by presenting various direct applications.

Meiri (1996) demonstrates how to solve constraint satisfaction problems over mixed qualitative and quantitative constraints in a temporal domain. The qualitative constraints are based on Allen's Interval Algebra and on the Point Algebra. Whereas the quantitative constraints are described as differences of two (end-)points, that have to lie in a set of disjoint intervals. Consequently, this approach requires the modeling of disjunctions but otherwise can be mapped directly to linear programming. Therefore, And/Or LP can be used to model this temporal approach as well.

So far, And/Or LP has been used in the 2D euclidian space, but as the basic primitives are hyperplanes it can be applied to higher dimensions as well. Pacheco, Escrig, and Toledo (2002) introduced a double cross extension into the three dimensional space. This extension can be directly represented by half-planes. Given a finite set of 3D orientations, the 3D double cross calculus can be represented using And/Or LP.

Cabalar and Santos (2011) formalize a physical puzzle using high-level concepts such as *something can pass through something else*. And/Or LP can be used to approximate whether two entities are capable of performing such actions based on the geometry of the involved pieces.

Overall And/Or LP can be used wherever generalized disjunctive linear programming is used, such as in operational research, as both methods have the same expressivity but use different heuristics (see Chapter 7).

8.3 Outlook

In this section, I will describe possible future directions this line of research enables.

Ghosh et al. (2012) research how to compute *solution graphs* for And/Or directed acyclic graphs instead of just single answers. By allowing a directed acyclic graph representation of a formula instead of the currently used tree structure, common sub-formulas would only have to be checked once. Further, a solution graph provides the possibility of further restrictions without requiring a full re-computation, as it is currently the case.

In complex cases, where a formula includes lots of disjunctions, excluding a disjunction early can be very beneficial. The current And/Or LP approach is capable of adding coarsened relations to improve the pruning of the search space. However, why stop at the level of spatial relations, and why not also coarsening (sub-)formulas?

Göbelbecker et al. argue in "Coming Up With Good Excuses: What to do When no Plan Can be Found" (2010) that providing an excuse or explanation can be very beneficial. Coming up with such a good explanation generally requires an understanding, why no plan could be found. This is quite similar to *fault detection*. Gertler (1998) and Chiang, Braatz, and Russell (2001) define the notion of *fault detection, isolation and recovery*, which is also present in the case of data integration with conflicting data. Exploiting And/Or LP to reason from a coarsened consistent scenario and refine until an inconsistency is found, would enable the search of a minimal coarsened qualitative description that is still realizable. Therefore, using such a coarsened consistent qualitative description and compare it to the inconsistent description would improve the fault identification.

Wallgrün (2012) uses mixed integer linear programming to compute the minimal translation needed for polygons so that they correspond to a set of (consistent) constraints. The minimization criterion is based on the Minkowski Sums. However, Wallgrün does not consider (finite) rotations when minimizing the required transformation to achieve the desired qualitative spatial description. And/Or LP could be extended to also consider the Minkowski Sums when computing a solution. Further, And/Or LP would also allow to vary the orientation of the polygonal regions, resulting in possibly smaller overall changes.

Throughout this dissertation model checking is realized by translation to Answer Set Programming which in turn gets translated into a SAT problem. Rintanen (2012) exploited the conflict-driven approach of SAT solvers by selecting a heuristic suitable for general planning. For planning problems this approach provides a huge benefit. On the one hand, it would be interesting to see, whether such a heuristic could also be found for spatial problems such as those used throughout this dissertation. Development of a heuristic for spatial problems would require extensive evaluation and comparison to different heuristics. Gebser et al. (2013) add a *heuristic language* to Answer Set Programming, thereby allowing to easily change the heuristic without having to touch the solver. On the other hand, it would be interesting to integrate And/Or LP and SAT solving directly, i.e. call And/Or LP on partial solutions. If such partial solutions cannot be realized, than it can never be extended to a solution that is realizable, Therefore, unrealizable partial solutions should be exploitable during the conflict-driven approach of SAT solvers.

8.4 Final Thoughts

In this dissertation I showed that by restricting non-linearities to a finite set of linear approximations a middle ground between high-level expressivity and low-level numeric reasoning can be established. The two developed logics, CNL and QLTL, as well as And/Or Linear Programming contribute to the paradigm shift towards loosely-coupled modules that are individually sound and complete. Nevertheless, the combination might not yield a sound and complete reasoning system (Kontchakov, Kurucz, et al., 2007). In the future it will be interesting to see, whether the flaws of such systems are negligible or if they render the system too unstable. Further, would flaws possibly ease the engagement with robotic systems, as having flaws and quirks is quite human after all.

References

Atchley, Ruth Ann, Maureen Keeney, and Curt Burgess (1999). "Cerebral Hemispheric Mechanisms Linking Ambiguous Word Meaning Retrieval and Creativity". In: *Brain and Cognition* 40.3, pp. 479–499. ISSN: 0278-2626.

Baader, Franz and Benjamin Zarrieß (2013). "Verification of Golog programs over description logic actions". In: *Frontiers of Combining Systems*. Springer, pp. 181–196.

Balbiani, Philippe and Jean-François Condotta (2002). "Computational complexity of propositional linear temporal logics based on qualitative spatial or temporal reasoning". In: *Frontiers of Combining Systems*. Springer, pp. 162–176.

Bateman, John A, Joana Hois, Robert Ross, and Thora Tenbrink (2010). "A linguistic ontology of space for natural language processing". In: *Artificial Intelligence* 174.14, pp. 1027–1071.

Bhatt, Mehul (2010). "Commonsense Inference in Dynamic Spatial Systems: Epistemological Requirements." In: *FLAIRS Conference*.

Bhatt, Mehul, Jae Hee Lee, and Carl Schultz (2011). "CLP(QS): a declarative spatial reasoning framework". In: *Proc. of Cosit*, pp. 210–230.

Cabalar, Pedro and Paulo E Santos (2011). "Formalising the Fisherman's Folly puzzle". In: *Artificial Intelligence* 175.1, pp. 346–377.

Chiang, Leo H, Richard D Braatz, and Evan L Russell (2001). *Fault detection and diagnosis in industrial systems*. Springer.

Levesque, Hector J., Raymond Reiter, Yves Lespérance, Fangzhen Lin, and Richard B. Scherl (1997). "GOLOG: A Logic Programming Language for Dynamic Domains". In: *Journal of Logic Programming* 31, pp. 59–84.

De Giacomo, Giuseppe, Yves Lespérance, and Fabio Patrizi (2012). "Bounded Situation Calculus Action Theories and Decidable Verification." In: *KR*.

De Giacomo, Giuseppe, Yves Lespérance, Fabio Patrizi, and Stavros Vassos (2014). "LTL Verification of Online Executions with Sensing in Bounded Situation Calculus". In: Frontiers in Artificial Intelligence and Applications.

Dornhege, Christian, Marc Gissler, Matthias Teschner, and Bernhard Nebel (2009). "Integrating symbolic and geometric planning for mobile manipulation". In: *Safety, Security & Rescue Robotics (SSRR), 2009 IEEE International Workshop on*. IEEE, pp. 1–6.

Dylla, Frank (Aug. 2008). *An Agent Control Perspective on Qualitative Spatial Reasoning*. Bd. 320. Vol. 174. Dissertationen zur Künstlichen Intelligenz. AKA Verlag. ISBN: 978-3-89838-320-2.

ECAI 2014 - 21st European Conference on Artificial Intelligence, Prague, Czech Republic, August 18-22, 2014, Proceedings (2014). Frontiers in Artificial Intelligence and Applications. IOS Press.

Fasola, Juan and Maja J Mataric (2013). “Using semantic fields to model dynamic spatial relations in a robot architecture for natural language instruction of service robots”. In: *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, pp. 143–150.

Freksa, Christian (1991). “Conceptual neighborhood and its role in temporal and spatial reasoning”. In: *Proceedings of the IMACS Workshop on Decision Support Systems and Qualitative Reasoning*. Ed. by Madan G. Singh and Luise Travé-Massuyès. Amsterdam: Elsevier, pp. 181–187.

Galati, Gaspare, Gina Pelle, Alain Berthoz, and Giorgia Committeri (2010). “Multiple reference frames used by the human brain for spatial perception and memory”. In: *Experimental brain research* 206.2, pp. 109–120.

Galindo, Cipriano and Alessandro Saffiotti (2013). “Inferring robot goals from violations of semantic knowledge”. In: *Robotics and Autonomous Systems* 61.10. Selected Papers from the 5th European Conference on Mobile Robots (ECMR 2011), pp. 1131–1143. ISSN: 0921-8890.

Galton, Antony (2000). *Qualitative Spatial Change*. Spatial Information Systems. Oxford University Press. ISBN: 978-0-198-23397-8.

Gebser, Martin, Benjamin Kaufmann, Javier Romero, Ramón Otero, Torsten Schaub, and Philipp Wanko (2013). “Domain-Specific Heuristics in Answer Set Programming.” In: *AAAI*.

Gertler, Janos (1998). *Fault detection and diagnosis in engineering systems*. CRC press.

Ghosh, Priyankar, Amit Sharma, Partha Pratim Chakrabarti, and Pallab Dasgupta (2012). “Algorithms for generating ordered solutions for explicit AND/OR structures”. In: *Journal of Artificial Intelligence Research* 44.1, pp. 275–333.

Göbelbecker, Moritz, Thomas Keller, Patrick Eyerich, Michael Brenner, and Bernhard Nebel (2010). “Coming Up With Good Excuses: What to do When no Plan Can be Found”. In: *ICAPS*, pp. 81–88.

Gottfried, Björn (2012). “Using space to represent data: diagrammatic reasoning”. In: *Cognitive processing*, pp. 1–3.

Haarslev, Volker, Carsten Lutz, and Ralf Möller (1998). “Foundations of spatioterminological reasoning with description logics”. In: *KR* 98, pp. 112–123.

Halpern, Joseph Y and Willemien Kets (2012). “Ambiguous Language and Differences in Beliefs.” In: *KR*.

Haun, Daniel, Christian J Rapold, Gabriele Janzen, and Stephen C Levinson (2011). “Plasticity of human spatial cognition: Spatial language and cognition covary across cultures”. In: *Cognition* 119.1, pp. 70–80.

Hirschfeld, Lawrence A and Susan A Gelman (1994). “Toward a topography of mind: An introduction to domain specificity”. In: *Mapping the mind: Domain specificity in cognition and culture*, pp. 3–35.

Klippel, Alexander and Dan Montello (2007). “Linguistic and nonlinguistic turn direction concepts”. In: *Proceedings of Conference of Spatial Information Theory (COSIT)*. Vol. 4736. Springer, pp. 354–372.

Klippel, Alexander, Jan Oliver Wallgrün, Jinlong Yang, Jennifer S Mason, Eun-Kyeong Kim, and David M Mark (2013). “Fundamental Cognitive Concepts of Space (and Time): Using Cross-Linguistic, Crowdsourced Data to Cognitively Calibrate Modes of Overlap”. In: *Spatial Information Theory*. Springer, pp. 377–396.

Klippel, Alexander, Jinlong Yang, Jan Oliver Wallgrün, Frank Dylla, and Rui Li (2012). “Assessing similarities of qualitative spatio-temporal relations”. In: *Spatial Cognition VIII*. Springer, pp. 242–261.

Knauff, Markus, Reinhold Rauh, and Jochen Renz (1997). “A cognitive assessment of topological spatial relations: Results from an empirical investigation”. In: *Spatial Information Theory: A theoretical basis for GIS*. Proceedings of COSIT. Springer, pp. 193–206.

Kollar, Thomas, Stefanie Tellex, Deb Roy, and Nicholas Roy (2010). “Toward understanding natural language directions”. In: *Human-Robot Interaction (HRI), 2010 5th ACM/IEEE International Conference on*. IEEE, pp. 259–266.

Kontchakov, Roman, Agi Kurucz, Frank Wolter, and Michael Zakharyaschev (2007). “Spatial Logic + Temporal Logic = ?” In: *Handbook of Spatial Logics*. Ed. by Marco Aiello, Ian E. Pratt-Hartmann, and Johan van Benthem. Springer, pp. 497–564. ISBN: 978-1-4020-5586-7.

Kontchakov, Roman, Yavor Nenov, Ian Pratt-Hartmann, and Michael Zakharyaschev (2011). “On the decidability of connectedness constraints in 2D and 3D Euclidean spaces”. In: *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*. Vol. 22. 1, p. 957.

Kordjamshidi, Parisa, Joana Hois, Martijn van Otterlo, and Marie-Francine Moens (2011). “Machine learning for interpretation of spatial natural language in terms of qsr”. In: *International Conference on Spatial Information Theory (COSIT’11). Extended abstract.* technical report, pp. 1–5.

Lee, Jae Hee (2014). “The Complexity of Reasoning with Relative Directions.” In: *ECAI 2014 - 21st European Conference on Artificial Intelligence, Prague, Czech Republic, August 18-22, 2014, Proceedings*. Frontiers in Artificial Intelligence and Applications. IOS Press.

Lee, Joohyung and Ravi Palla (2012). “Reformulating the situation calculus and the event calculus in the general theory of stable models and in answer set programming”. In: *Journal of Artificial Intelligence Research* 43.1, pp. 571–620.

Levesque, Hector J., Ernest Davis, and Leora Morgenstern (2011). “The Winograd schema challenge.” In: *AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning*.

Li, J.J. and Sanjiang Li (Nov. 2013). “On Finding Approximate Solutions of Qualitative Constraint Networks”. In: *Tools with Artificial Intelligence (ICTAI), 2013 IEEE 25th International Conference on*, pp. 30–37.

Li, Sanjiang and Mingsheng Ying (2004). “Generalized region connection calculus”. In: *Artificial Intelligence* 160.1, pp. 1–34.

McCarthy, John (1963). *Situations, Actions, and Causal Laws*. Tech. rep. Memo 2. Stanford University: Stanford Artificial Intelligence Project.

Meiri, Itay (1996). “Combining qualitative and quantitative constraints in temporal reasoning”. In: *Artificial Intelligence* 87.1, pp. 343–385.

Moratz, Reinhard, Jochen Renz, and Diedrich Wolter (2000). “Qualitative Spatial Reasoning about Line Segments”. In: *ECAI 2000. Proceedings of the 14th European Conference on Artifical Intelligence*. IOS Press, pp. 234–238.

O’Keefe, John (2003). *Vector grammar, places, and the functional role of the spatial prepositions in English*. Oxford University Press.

Pacheco, Julio, Mª Teresa Escrig, and Francisco Toledo (2002). “Qualitative spatial reasoning on three-dimensional orientation point objects”. In: *Proccedings of the QR2002. 16th International WorkShop on Qualitative Reasoning. Editors: Nuria Agell and.*

Petkovšek, Marko, Herbert S. Wilf, and Doron Zeilberger (1996). *A = B*. Wellesley, MA: A.K. Peters, Ltd., p. 212. ISBN: 1568810636.

Piaget, Jean (1967). *Logique et connaissance scientifique*. Vol. 22. Encyclopedie de la Pleiade. Paris: Gallimard Paris.

Pnueli, Amir (1977). “The temporal logic of programs”. In: *Proceedings of the 18th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 46–57.

Randell, David A., Zhan Cui, and Anthony G. Cohn (1992). “A Spatial Logic based on Regions and “Connection””. In: *Proc, of KR’92*. Ed. by Bernhard Nebel, Charles Rich, and William Swartout. San Mateo (CA), USA: Morgan Kaufmann, pp. 165–176.

Rintanen, Jussi (2012). “Planning as satisfiability: Heuristics”. In: *Artificial Intelligence* 193, pp. 45–86.

Rost, Pascal, Lothar Hotz, and Stephanie Von Riegen (2012). “Supporting Mobile Robot’s Tasks through Qualitative Spatial Reasoning.” In: *ICINCO (2)*. Citeseer, pp. 394–399.

Suchan, Jakob (2011). *High-level Spatial Control and Planning for Cognitive Robotics: Experiments with the ExpCog Framework*. Diploma Thesis. Germany.

Tellex, Stefanie, Thomas Kollar, Steven Dickerson, Matthew R Walter, Ashis Gopal Banerjee, Seth J Teller, and Nicholas Roy (2011). “Understanding Natural Language Commands for Robotic Navigation and Mobile Manipulation.” In: *AAAI*.

Tenbrink, Thora and Marco Ragni (2012). “Linguistic principles for spatial relational reasoning”. In: *Spatial Cognition VIII*. Springer, pp. 279–298.

Waibel, M. et al. (June 2011). “RoboEarth”. In: *Robotics Automation Magazine, IEEE* 18.2, pp. 69–82. ISSN: 1070-9932.

Wallgrün, Jan Oliver (2012). “Exploiting Qualitative Spatial Reasoning for Topological Adjustment of Spatial Data”. In: *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*. SIGSPATIAL ’12. Redondo Beach, California: ACM, pp. 229–238. ISBN: 978-1-4503-1691-0.

Westphal, Matthias, Julien Hué, Stefan Wölfl, and Bernhard Nebel (2013). “Transition constraints: a study on the computational complexity of qualitative change”. In: *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*. AAAI Press, pp. 1169–1175.

Wölfl, Stefan and Matthias Westphal (2009). “On Combinations of Binary Qualitative Constraint Calculi”. In: *Proc. of IJCAI*, pp. 967–973.

Wolter, Diedrich and Jae Hee Lee (2010). “Qualitative Reasoning with Directional Relations”. In: *Artif. Intell.* 174.18, pp. 1498–1507.

Acknowledgements

A dissertation is a journey that one doesn't travel alone and therefore I now want to thank all those that helped me along the way. First of all, thank you Christian for letting me be part of your group and providing me with this fantastic opportunity to pursue my own interests. I'm especially glad, that you let me go detours and depart form the straight and narrow, as most of my findings have their roots in these sideways.

Thank you, Diedrich, for getting me back on track and for asking all the nasty little questions, for all the long and sometimes heated discussions. I really enjoyed those discussions with you. Your "good idea, write it down briefly and put it on the pile: AFTER THE DISSERTATION" was always the right comment to keep me focused, even though I sometimes didn't want to hear it. A big thank you for all your support and the friendship we share.

At this point I want to thank Christian and Professor Alessandro Saffiotti for reviewing my dissertation.

A huge thank you goes to the R3-Projekt Members, Frank, Immo, Jae, Lutz, and Diedrich for all the interesting discussions, the nice Dart rounds we played, and of course for helping me ironing out all my little spelling mistakes.

Bernd Neumann, I would like to thank you for my start into the world of science. I learned a lot during my time in Hamburg and jumped the first few hurdles. From my time in Hamburg, I would also like to thank you, Hans Meine, for introducing me to the wonderful programming language call Python and for keeping my interest in computer visions high.

A big thank you go to all my friends, especially to Philipp Krohn, Immo Colonius and Sandra Budde. You helped me to not forget of all the wonders and activities outside of academia. Thank you all for all your support, may it be technical or moral.

My parents deserve one of the biggest thank you's, as they provided the support I needed to become the person I am today. I always could count on your support and you helped us a lot, when our little family had to overcome some health related difficulties. Vielen Danke fr eure liebevolle Untersttzung!

Anne, my dear wife, thank you for supporting me as much as possible, be it morally or with time to spend on my dissertation. Thank you for share my joy of board games and miniature painting. But most of all, I thank you for all your love and kindness.

Thank you, Finn Thore Kreutzmann, my dear son, for reminding me of the important things in life and for sharing your Legos with me.

Curriculum vitae

First Name	Arne
Surname	Kreutzmann
Date of birth	29.12.1981
Place of birth	Hannover
Nationality	German

Education

2010 – 2014	**Doctoral Researcher** in the project R3-[Q-Shape] of the Transregional Collaborative Research Center SFB/ TR 8 Spatial Cognition, as part of the Cognitive Systems Group, University of Bremen, Germany
2008 – 2009	**Researcher** in the Cognitive Systems Group, University of Hamburg, Germany
2007	**Diploma thesis**: "Development of a method for classifing temporal varying, dynamic objects"[9] at the Leibniz University of Hannover, Germany
2002 – 2007	**Diploma** in Mathematics with a major in comupter science[10] at the Leibniz University of Hannover, Germany
2001 – 2002	**Civilian service** at Kirchengemeinde Alt-Garbsen, Garbsen, Germany
1998 – 1999	**US High school diploma** at the Grand Meadow Public School, Minnesota, USA
1994 – 2001	**Abitur**[11] at the Bismarkschule Hannover (Gymnasium), Germany

[9] "Enticklung eines Verfahrens zur Klassifikation zeitvarianter, dynaischer Objekte"

[10] Mathematik mit Studienrichtung Informatik

[11] Diploma from german secondary school qualifying for university admission

Journal Publications

Kreutzmann, Arne, Immo Colonius, Diedrich Wolter, Frank Dylla, Lutz Frommberger, and Christian Freksa (2013). “Temporal logic for process specification and recognition”. In: *Intelligent Service Robotics* 6.1, pp. 5–18.

Kreutzmann, Arne, Diedrich Wolter, Frank Dylla, and Jae Hee Lee (2013). “Towards Safe Navigation by Formalizing Navigation Rules”. In: *TransNav, the International Journal on Marine Navigation and Safety of Sea Transportation* 7.2, pp. 161–168. ISSN: 2083-6473.

Submitted

Kreutzmann, Arne and Diedrich Wolter (n.d.). “Conceptual Neighborhood Logic with Partially Grounded Information for Safe Navigation”. In: *Artificial Intelligence*. UNDER REVIEW.

Presented at Conferences

Kreutzmann, Arne, Kasim Terzć, and Bernd Neumann (2009). “Context-aware classification for incremental scene interpretation”. In: *UCVP ’09: Proceedings of the Workshop on Use of Context in Vision Processing*. New York, NY, USA: ACM. ISBN: 978-1-60558-691-5.

Dylla, Frank and Arne Kreutzmann (2010). “Agent Control by Adaptive Neighborhoods”. In: *Proceedings of the ECAI 2010 Workshop on Spatio-Temporal Dynamics*.

Kreutzmann, Arne, Immo Colonius, Lutz Frommberger, Frank Dylla, Christian Freksa, and Diedrich Wolter (Sept. 2011). “On Process Recognition by Logical Inference”. In: *Proc. of the 5th European Conference on Mobile Robots (ECMR)*, pp. 7–12.

Kreutzmann, Arne and Diedrich Wolter (2011). “Physical Puzzles—Challenging Spatio-Temporal Configuration Problems”. In: *Proceedings of IJCAI-2011 Workshop Benchmarks and Applications of Spatial Reasoning*. Ed. by Jochen Renz, Anthony G. Cohn, and Stefan Wölfl.

Wolter, Diedrich, Frank Dylla, and Arne Kreutzmann (2011). “Rule-Compliant Navigation With Qualitative Spatial Reasoning”. In: *Proceedings of the 4th Intl. Robotic Sailing Conference*. Springer.

Dylla, Frank, Arne Kreutzmann, and Diedrich Wolter (2014). “A Qualitative Representation of Social Conventions for Application in Robotics”. In: *2014 AAAI Spring Symposium - Qualitative Representations for Robots*. Ed. by Nick Hawes. Papers from the 2014 AAAI Spring Symposium SS-14-06. Palo Alto, California, USA. ISBN: 978-1-57735-646-2.

Kreutzmann, Arne and Diedrich Wolter (2014). “Qualitative Spatial and Temporal Reasoning with AND/OR Linear Programming”. In: *ECAI 2014 - 21st European Conference on Artificial Intelligence, Prague, Czech Republic, August 18-22, 2014, Proceedings*. Frontiers in Artificial Intelligence and Applications. IOS Press.

Van de Ven, Jasper, Arne Kreutzmann, Sascha Schrader, and Frank Dylla (2014). “Multi-PN-Learning for Tracking Applications”. In: *The 13th International Conference on Control, Automation, Robotics and Vision*.

Other Publications

Hartz, Johannes, Patrick Koopmann, Arne Kreutzmann, and Kasim Terzć (Nov. 2010). *eTRIMS Scene Interpretation Datasets*. Tech. rep. FBI-HH-M-345/10. Department of Informatics, Universität Hamburg.

Erklärung der selbstständigen Erarbeitung

Erklärung gemäß § 6 Abs. 5 der Promotionsordnung
der Universität Bremen vom 14.03.2007
für die mathematischen, natur- und ingenieurwissenschaftlichen
Fachbereiche

Hiermit versichere ich, dass ich die vorliegende Dissertation mit dem Title “Qualitative Spatial and Temporal Reasoning based on And/Or Linear Programming”

1. ohne unerlaubte fremde Hilfe angefertigt habe
2. keine anderen als die von mir angegebenen Quellen und Hilfsmittel benutzt habe und
3. die den benutzten Werken wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Bremen, den 16. Dezember 2014

____
(Arne Kreutzmann)