**Journal of Intelligent and Robotic Systems**
preprint of DOI 10.1007/s10846-011-9606-0

# Reasoning with Qualitative Positional Information for Domestic Domains in the Situation Calculus

**Stefan Schiffer · Alexander Ferrein · Gerhard Lakemeyer**

Received: 12 December 2010 / Accepted: 16 May 2011

**Abstract** In this paper, we present a thorough integration of qualitative representations and reasoning for positional information for domestic service robotics domains into our high-level robot control. In domestic settings for service robots like in the RoboCup@Home competitions, complex tasks such as "get the cup from the kitchen and bring it to the living room" or "find me this and that object in the apartment" have to be accomplished. At these competitions the robots may only be instructed by natural language. As humans use qualitative concepts such as "near" or "far", the robot needs to cope with them, too. For our domestic robot, we use the robot programming and plan language Readylog, our variant of Golog. In previous work we extended the action language Golog, which was developed for the high-level control of agents and robots, with fuzzy set-based qualitative concepts. We now extend our framework to positional fuzzy fluents with an associated positional context called *frames*. With that and our underlying reasoning mechanism we can transform qualitative positional information from one context to another to account for changes in context such as the point of view or the scale. We demonstrate how qualitative positional fluents based on a fuzzy set semantics can be deployed in domestic domains and showcase how reasoning with these qualitative notions can seamlessly be applied to a fetch-and-carry task in a RoboCup@Home scenario.

**Keywords** Qualitative Spatial Representation · Reasoning · Golog · Situation Calculus · Fuzzy Sets · Domestic Service Robotics

Part of the work was conducted while A. Ferrein was affiliated with the Robotics and Agents Research Lab at the University of Cape Town, South Africa.

Stefan Schiffer · Alexander Ferrein · Gerhard Lakemeyer
Knowledge Based Systems Group
RWTH Aachen University, Aachen, Germany
E-mail: {schiffer,ferrein,gerhard}@cs.rwth-aachen.de

## 1 Introduction

Suppose you want to instruct your domestic robot with the instruction "Get me my cup from the kitchen table". Besides natural language processing and sophisticated robot control for navigation, localization, object recognition etc., the robot needs to be equipped with a flexible high-level control entity that can figure out from the instruction that the robot needs to go to the kitchen, pick up "my cup" (if there are more cups on the table, which is the right one?), grab it, and bring it to the human instructor. A field dealing with these kinds of problems is the field of cognitive robotics. Classical applications here are delivery tasks, where the robot should deliver a letter or fetch a cup of coffee. The field of cognitive robotics is studying knowledge representation and reasoning problems which are faced by autonomous robots. The central purpose of studying such problems is to design the high-level control for autonomous systems. The high-level control enables the robot to act goal-directed and purposefully towards achieving its mission goals. That means that the robot is not executing a previously programmed fixed sequence of commands, but should figure out how to achieve a certain goal in an intelligent way by itself. See, for instance, [29] for an excellent overview of the field.

In these domains, it becomes obvious that solving such tasks deploying reasoning and knowledge representation is superior to, say, reactive approaches in terms of flexibility and expressiveness. An even more advanced application domain is RoboCup@Home [57,58]. As a distinguished league under the roof of the RoboCup federation the robots have to fulfil complex tasks such as "*Lost&Found*", "*Fetch&Carry*", or "*WhoIsWho*" in a domestic environment. In the first tasks the robot has to remember and to detect objects, which are hidden in an apartment, or has to fetch a cup of coffee from, say, the kitchen and bring it to the sitting room, while in the latter the robot needs to find persons and recognize their faces. The outstanding feature of these applications is that they require integrated solutions for a number of sub-tasks such as safe navigation, localization, object recognition, and high-level control (e.g. reasoning). A particular complication is that the robot may only be instructed by means of natural interaction, e.g. speech or gestures. Human-robot interaction is hence largely based on natural language. For example, in the *Fetch&Carry* task it is allowed to help the robot with hints like "the teddy is near the TV set" or "the cup is on the kitchen table".

The domestic real world features two important aspects: space and humans. What is more, humans tend to use qualitative concepts and notions like near or far to refer to positions in Euclidean space. The robot needs to be able to interpret these concepts to cope with them. When reasoning techniques are deployed to come up with a problem solution for these domestic tasks, also such mechanisms need to be able to cope with those qualitative concepts. But even as logic-based reasoning approaches make inherently use of qualitative concepts, the rest of the complex robot architecture does not. Hence, one needs to bridge the gap between the qualitative high-level control and the quantitative robot control system. Another property when dealing with such qualitative notions is that they are dependent on the context in which they were expressed. For a human, a distance"far" w.r.t. the living room has a smaller scale than "far" with respect to the garden. This means that a robot dealing with these concepts also needs to know the context in which a qualitative instruction is given.

In this paper, we present a unified reasoning framework for qualitative positional information. We present models for qualitative notions of space, i.e. qualitative distance and orientation based on fuzzy sets, and integrate them into the situation calculus [33], a framework for reasoning about actions and change. We show how reasoning can be done with these qualitative notions in the situation calculus. Further, we formalize the concept of positional contexts, so-called frames. Each positional information needs to be related to a particular frame. We present a unified way to transform qualitative positional information into different frames. Finally, we present a high-level controller for RoboCup@Home's fetch-and-carry task in READYLOG [17]. READYLOG is a robot programming and plan language whose formal semantics is based on the situation calculus and which allows for reasoning and programming a robot under real-time constraints. It was deployed for cognitive robotics tasks ranging from intelligent soccer to domestic service robots (e.g. [17, 28, 48]). With this high-level controller, we show how the theoretical concepts of qualitative positional information and frames can be seamlessly integrated into and deployed for a robot controller for domestic service robots. The contributions are in detail:

- We represent qualitative categories as fuzzy sets and formalize them in the situation calculus. The appealing feature is that fuzzy sets allow for categories to overlap, that is, it can be represented that a quantitative value belongs to several qualitative categories at the same time.
- We introduce so-called fuzzy fluents, extensions of functional fluents, that can take qualitative categories as function values. This allows for representing that $dist(A, B) = \mathsf{far}$, i.e. that $A$ is far away from B. We further formalize membership to a qualitative category and a defuzzifier function that allows to re-quantify a qualitative category.
- We introduce positional fuzzy fluents and the notion of frames which allow to relate positional fuzzy fluents to a positional context. With that we can transform positional information from one context into another and we can, for example, account for that "far" in the living room is related to a larger scale than "far" in the bathroom (assuming the bathroom is smaller).
- Finally, we present a READYLOG controller for the fetch-and-carry task and show in an extended example how the qualitative concepts are used to solve this service robotics task.

The rest of the paper is organized as follows. In Section 2, we discuss related work to our approach mainly focusing on qualitative-spatial representations and qualitative-spatial reasoning systems, work from the field of fuzzy logic related to qualitative representations and their applications in robotics. Section 3 introduces the situation calculus, our high-level robot programming and plan language READYLOG, and gives a brief introduction to fuzzy sets. In Section 4 we introduce our qualitative reasoning framework, formalizing fuzzy fluents, qualitative distance and orientation. We show how reasoning with qualitative categories works by giving a simple example from a one-dimensional robot world. Section 5 extends our framework with positional fuzzy fluents and the notion of the positional context (frames). We axiomatize the domestic robot domain in the situation calculus and present a READYLOG controller for the fetch-and-carry task as known from RoboCup@Home competitions. Although we focus on the theoretical foundations for qualitative spatial representations and reasoning in the situation calculus in this paper, the controller indicates how this work can be turned into a real-world application. We conclude with a discussion and an outlook to future work in Section 6.

## 2 Related Work

In this section we review related work on both, qualitative spatial representations and reasoning as well as representations based on fuzzy logic. Furthermore, we briefly discuss combinations of the above and their application to robotics.

### 2.1 Qualitative Spatial Representation and Reasoning

Cohn and Hazarika give an overview of major qualitative spatial representation and reasoning techniques in [8]. They survey the main aspects of qualitative representations and they also consider methods for qualitative reasoning. Their survey covers ontological aspects and topological approaches as well as methods on distance, orientation, and shape.

An overview of constraint-based techniques for qualitative spatial reasoning along with a discussion of their computational properties is given in [45]. One of the most well-known works on qualitative spatial reasoning is the region connection calculus (RCC) introduced by Randell, Cui, and Cohn in [43]. The fundamental approach bases on extended spatial entities, that is regions, and the relations, namely connections, between them. Although the RCC provides a powerful method to describe and reason about spatial structures, especially for topological structures, we opt against using this calculus or any of its derivatives for several reasons. Firstly, we do not intend to use regions for representing spatial objects in our target domain, at least not yet. Secondly, we think that we do not have to rely on the generality provided by the RCC. From our point of view, the description of spatial settings needed for our specific application in domestic environments can be achieved more easily. We are looking for a mechanism that allows for qualitative representation of positional information rather than topological relations in the first place.

A well-known approach to qualitative representations of positional information was proposed by Freksa and Zimmermann in [22]. It bases on directional orientation information. The approach is motivated by considerations on how spatial information is available to humans and to animals: directly through their perception. Thus, cognitive considerations about the knowledge acquisition process build the basis here. Qualitative orientation information in two-dimensional space is given by the relation between a vector from start point $A$ to an end point $B$ and a point $C$. The vector represents the orientation of a possible movement. Different positions of the point $C$ can be described in relation to a line through $A$ and $B$, and further in relation to additional lines through $A$ and $B$ orthogonal to the line from $A$ to $B$. Reasoning with these relations is possible through four operations. The approach presented in [21, 22] is quite intuitive, not only because it is based on human cognition. Any relation is based on a vector between two points which, however, cannot be taken for granted in the context of our work, since spatial settings in a domestic environment do not always involve a movement, but they may also describe static situations. Even if we consider the intrinsic orientation of an object to construct such a vector, not all the relations are meaningful if the vector does not have a length. Further, we aim for unified representation for distance and orientation. This is also why we do not consider $\mathcal{OPRA}_m$ [36] or variants thereof despite the favourable property of adjustable granularity.

Another approach to qualitative orientation relation was introduced by Hernandez [25]. Instead of using geometric models which are very precise, Hernandez establishes a cognitive model of space. He argues that cognitive spatial concepts are qualitative in nature and preciseness is normally not needed in cognitive models. Although the representation might correspond to many 'real' situations, it avoids falsifying effects of an exact geometric approach which are likely due to the common limited acuity of perception. Hernandez states that the direct modelling of qualitative statements allows for a more efficient way to handle partial and uncertain spatial information. The orientation relation is meant to represent where objects are placed relatively to each other. The framework proposes to model qualitative orientation with angular intervals. The number of intervals, that is, the number of distinctions is determined by a *level of granularity*. An orientation relation states where a *primary object* is located in relation to a *reference object*. Further, there is a *reference frame* which determines the direction in which the primary object is located in relation to the reference object. Hernandez presents methods for changing the reference frame as well as methods for composing relations.

A basic method for qualitative distances was discussed in 1995 by Hernandez, Clementini, and Felici in [26]. Three elements are needed to establish a distance relation, namely a *primary object*, a *reference object*, and a *frame of reference*. The distance $\mathrm{dist}(A, B)$ between the reference object $A$ and the primary object $B$ is expressed as one out of a set of distance relations. These relations are formed by partitioning the plane into regions. These partitions represent the distinctions being made. The context in which the distinctions are made is represented by the frame of reference. It accounts for contextual information such as type and scale of the distance relations as well as for the *distance system*, which contains the distance relations and a set of structure relations describing how the distance relations relate to each other. Among their representation of distances through geometric intervals, they describe basic inference mechanisms such as composition and switching between different frames of reference.

Two years later, in 1997, Clementini, Felici, and Hernandez presented a unified framework for qualitative representation of positional information in two-dimensional space in [7]. In order to represent positional information they combine the distance relation and the orientation relation mentioned above. Again, they also introduce basic inference and reasoning mechanisms. Our representation of positional information is largely based on the approach presented in [7] and we detail it in Section 4.2.

A combination of an approach to qualitative spatial reasoning with reasoning about actions and change, namely the Situation Calculus, is approached in [14]. While the general idea of integrating the spatial reasoning with the reasoning about actions and change is very appealing and we support the use of the situation calculus as an adequate mechanism for reasoning in agent control, we argue that the underlying concept of spatial neighbourhood based on the dipole calculus does not quite match our needs. There have been further approaches to integrate qualitative spatial reasoning techniques with agent control in the Situation Calculus such as [42] and [13]. Nevertheless, we think that our integration of fuzzy set-based qualitative positional information in the situation calculus is worthwhile pursuing, especially for the domestic service robotics scenario targeted here.

An early approach that tried to combine metric and topological representations using fuzzy notions was presented in [34]. Similarly to our basic idea of retaining the connection between qualitative concepts and metric representations, they use associated frames of reference and ranges for positions. These are first used to build a fuzzy map on which later route planning can be done. The fuzzy notions used, according to the authors, may or may not be related to fuzzy logic. We, instead, try to achieve a formal integration of the fuzzy set semantics as the basis of our integration of qualitative spatial information. Furthermore, we are more interested in representations of general positional information than on route planning.

The concept of *frames* that we introduce in Section 5.2 to capture the context of a positional reference is not new. It has been found to develop in children already [41]. In schema theory [1] the similar concept of a schema is used to structure modelling of functional units such as perception or motion. It has even been attempted to replicate the construction of meaning by means of learning structure, e.g. from grammatical constructions [10]. In this article, we rely on frames to capture the contextual information needed to be able to relate positional information given with different contexts to one another. Therefore we restrict ourselves to frames in a positional sense and we try to keep it as simple as possible.

### 2.2 Fuzzy Logic

The underlying concept in fuzzy logic is that of *linguistic variables* as introduced by Zadeh in [56]. These linguistic terms can be used for representation as well as for control. According to [11], "fuzzy sets and possibility theory offer a unified framework for taking into account the gradual or flexible nature of many predicates, requirements, and the representation of incomplete information". It is suited to represent human-like taxonomies in pattern classification, or lexical imprecision of natural language.

In an earlier work, Sugeno and Yasukawa [52] discussed the adequacy of fuzzy logic-based approaches for qualitative modelling. In their work they describe the process of generating a qualitative model based on fuzzy representations from a set of sample input-output data describing the system behaviour. Based on heuristics they identify the input data which influence the output. Then, they approximate the number of fuzzy rules needed to describe the output data. The result is a fuzzy controller, i.e. a set of fuzzy rules, which describes the mapping from the given input values to the output values. Later, Tikk et al. [53] improved the original idea and proposed several algorithms for building trapezoid approximations of membership functions and for rule base reduction. Here, fuzzy logic is used for approximating a given system in a qualitative way.

In [5], Bolloju uses qualitative variables based on fuzzy sets to facilitate decision making. While the approach of Bolloju is similar to ours, his approach is very limited in the way decisions can be made. Similar lines as to use fuzzy qualitative variables are followed in [12,39]. In our approach however, we can make use of the full expressiveness of the situation calculus for taking decisions. A closely related approach is [54]. They propose a fuzzy test-score semantics for soft constraints. They propose a function *fholds* which evaluates the result of combinations of soft constraints in allusion to the *holds* predicate known from action logics. Although they state to use the situation calculus, besides the function *fholds* they do not give a full axiomatization.

As an example for the large body of work dealing with fuzzy control and robots, we like to mention [32, 47]. Soffiotti [47] shows how fuzzy controllers can be used to design robust behaviour-producing modules, and even how high-level reasoning and low-level execution can be integrated on a mobile robot. He attributes the success of fuzzy logic in control to "its ability to represent both the symbolical and the numerical aspects of reasoning. Fuzzy logic can be embedded in a full logical formalism, endowed with a symbolic reasoning mechanism; but it is also capable of representing and processing numerical data." Liu et al. [32] describe a robot kinematics in a qualitative fashion making use of fuzzy descriptions. They use fuzzy qualitative trigonometric functions to describe the movements of a PUMA robot manipulator. This fuzzy qualitative description is, according to the authors, very helpful for calibration procedures in terms of measuring accuracy or repeatability. They further stress that the fuzzy qualitative predicates provide the connection between the numerical data and interval symbols, which are then used for building up the behaviour vocabulary from which in turn the motion control of the robot can be derived. Many other successful examples for fuzzy control applications are given in [27]. Good overviews of the fields are also given in [11, 35, 40, 55].

### 2.3 Fuzzy Approaches to Spatial Representations and Applications in Robotics

Schockaert et al. [51] approach a generalization of the region connection calculus (RCC) to allow for representations and reasoning in terms of fuzzy relations between vague regions using fuzzy set theory. Similarly, there have been approaches [2] to integrate the RCC into the situation calculus. Although the same reasons as given earlier hold for why we do not use the RCC, both these ideas are appealing and might be examined in future work.

Bloch and Saffiotti make use of fuzzy set theory-based representations for robot maps in [4]. They state that an application of their approach to self-localization and reasoning seems possible. Still, they mainly cover directional information only and the accuracy of the localization is pretty coarse yet. In [3], Bloch investigates the use of a fuzzy set-based framework for spatial reasoning with a focus on the use in image understanding, structure recognition, and computer vision. The author claims to be able to derive useful representations and a reasoning mechanism by making use of connections to mathematical morphology and formal logics. Interestingly, the approach presented allows for quantitative, semi-qualitative, fuzzy, and symbolic representations. We aim to retain the above flexibility in a slightly different way, though, keeping the connection of our qualitative spatial representations to Euclidean space and casting reasoning into classic geometric operations.

In [37], Müller et al. present an application of qualitative spatial representations to robot navigation. They consider the following scenario: In a hospital, a patient should visit a certain room for a medical examination. Since the patient is handicapped, a wheelchair is used to get to the examination room. Normally, the patient would be guided by a nurse, but the hospital is equipped with intelligent power wheelchairs due to time constraints. The nurse is able to instruct the wheelchair where to go so that the patient can automatically be transported to the examination room. To extract the qualitative notions, Müller et al. use methods presented in [38] which mainly base on the approach by Clementini et al. [7] that was already mentioned. They generate qualitative motion vectors by using qualitative distance and orientation relations. Then, these qualitative vectors are generalized to simplify the motion track. By this method they stress on the coarse form and only regard the major directional changes and the overall shape of the course of motion. For a detailed account on the algorithms applied we refer to [38].

The usefulness of fuzzy sets to make use of qualitative fluents in the situation calculus was already shown in [18]. We also defined fuzzy controllers in Golog based on the fuzzy set semantics in [19]. In this paper we revisit the use of fuzzy sets for building qualitative representations, now with a special emphasis on qualitative spatial information, in particular in domestic environments.

## 3 Theoretical Background

### 3.1 The Situation Calculus

The situation calculus is a second order language with equality which allows for reasoning about actions and their effects. The world evolves from an initial situation due to primitive actions. Possible world histories are represented by sequences of actions. The situation calculus distinguishes three different sorts: *actions*, *situations*, and domain dependent *objects*.

A special binary function symbol $do : action \times situation \rightarrow situation$ exists, with $do(a, s)$ denoting the situation which arises after performing action $a$ in situation $s$. The constant $S_0$ denotes the initial situation, i.e. the situation where no actions have yet occurred. We abbreviate the expression $do(a_n, \cdots do(a_1, S_0) \cdots)$ with $do([a_1, \dots, a_n], S_0)$.

The state the world is in is characterized by functions and relations with a situation as their last argument. They are called *functional* and *relational fluents*, respectively. As an example consider the position of a robot operating in a domestic environment. One aspect of the world state is the robot's location $robotLoc(s)$. Suppose the robot is in the kitchen in the initial situation $S_0$. It holds $robotLoc(S_0) = kitchen$. The robot now performs the action $\text{travel}(kitchen, parlour)$. The position of the robot therefore needs to be updated to "parlour": $robotLoc(do(\text{travel}(kitchen, parlour), S_0)) = parlour$.

The third sort of the situation calculus is the sort *action*. Actions are characterized by unique names: for distinct action names $a$ and $b$ it holds that $a(\mathbf{x}) \neq b(\mathbf{x})$ and $a(\mathbf{x}) = a(\mathbf{y}) \supset \mathbf{x} = \mathbf{y}$. For each action one has to specify a *precondition axiom* stating under which conditions it is possible to perform the respective action and *effect axioms* formulating how the action changes the world in terms of the specified fluents. An action precondition axiom has the form $Poss(a(\mathbf{x}), s) \equiv \Phi(\mathbf{x}, s)$ where the binary predicate $Poss : action \times situation$ denotes when an action can be executed, and $\mathbf{x}$ stands for the arguments of action $a$. For our travel action, the precondition axiom may be $Poss(\text{travel}(x, y), s) \equiv robotLoc(s) = x$. After having specified when it is physically possible to perform an action, it remains to state how the respective action changes the world. One has to specify negative and positive effects in terms of the relational fluent $F$, i.e. $\varphi_F^-(\mathbf{x}, s) \supset \neg F(\mathbf{x}, do(a, s))$ and $\varphi_F^+(\mathbf{x}, s) \supset F(\mathbf{x}, do(a, s))$, respectively. The effect axiom for a functional fluents $f$ is $\varphi_f(\mathbf{x}, y, a, s) \supset f(\mathbf{x}, do(a, s)) = y$. However, describing the positive and negative effect says nothing about those effects which do not change the fluent. The problem of describing the non-effects of an action is referred to as the *frame problem*. The number of frame axioms is very large. For relational fluents there exist in the order of $2 \cdot \mathcal{A} \cdot \mathcal{F}$ frame axioms, where $\mathcal{A}$ is the number of actions, and $\mathcal{F}$ the number of relational fluents. McCarthy & Hayes [33] were the first to mention this problem.

A solution to the problem was proposed in [44] with so-called *successor state axioms*. The idea behind these axioms is that, if the truth value of $F$ changes from false to true from situation $s$ to situation $do(a, s)$, then $\varphi_F^+(\mathbf{x}, a, s)$ must have been true. Similarly, for the second axiom. Reiter shows that under consistency assumptions for fluents together with the explanation closure axioms, the normal form axioms for fluent $F$ are logically equivalent to

$$F(\mathbf{x}, do(a, s)) \equiv \varphi_F^+(\mathbf{x}, a, s) \vee F(\mathbf{x}, a, s) \wedge \neg\varphi_F^-(\mathbf{x}, a, s). \tag{1}$$

The above formula is called *successor state axiom for the relational fluent F*. The successor state axiom for the functional fluent $f$ has the form [44]:

$$\begin{aligned} &f(\mathbf{x}, do(a, s)) = y \equiv \\ &\quad \varphi_f(\mathbf{x}, y, s) \vee f(\mathbf{x}, s) = y \wedge \neg\exists y'.\varphi_f(\mathbf{x}, y', a, s) \end{aligned} \tag{2}$$

The background theory must entail the consistency properties

$$\neg\exists \mathbf{x}, a, s.\varphi_F^+(\mathbf{x}, a, s) \wedge \varphi_F^-(\mathbf{x}, a, s) \tag{3}$$

$$\neg\exists \mathbf{x}, y, y', a, s.\varphi_f(\mathbf{x}, y, a, s) \wedge \varphi_f(\mathbf{x}, y', a, s) \wedge y \neq y'. \tag{4}$$

The number of $\mathcal{F}$ successor state axioms together with $\mathcal{A}$ action precondition axioms plus the unique names axioms is far less than the $2 \cdot \mathcal{F} \cdot \mathcal{A}$ explicit frame axioms that would be needed otherwise.

The background theory, also called basic action theory (BAT), is a set of sentences $\mathcal{D}$ consisting of

$$\mathcal{D} = \Sigma \cup \mathcal{D}_{ssa} \cup \mathcal{D}_{ap} \cup \mathcal{D}_{una} \cup \mathcal{D}_{S_0},$$

where

- $\Sigma$ is the set of foundational axioms for situations ensuring, for instance, that no action can be performed before $S_0$. Refer to [44] for details.
- $\mathcal{D}_{ssa}$ is a set of successor state axioms for functional and relational fluents, one for each fluent as given in Eq. 1 for relational fluents, and in Eq. 2 for functional fluents (together with the consistency property Eqs. 3 and 4).
- $\mathcal{D}_{ap}$ is a set of action precondition axioms, one for each action. The set $\mathcal{D}_{ap}$ is the set of precondition axioms of the form $Poss(a(\mathbf{x}), s)$ as described above.
- $\mathcal{D}_{una}$ is the set of unique names axioms for all actions.

– $\mathcal{D}_{S_0}$ is a set of first order sentences that are uniform in $S_0$ and describe the fluent values in the initial situation.[1]

To address the so-called projection problem, i.e. determining if a sentence holds for some future situations, a regression mechanism is used in the situation calculus. Basically, if one wants to prove that a sentence $W$ is entailed by the basic action theory and $W$ mentions a relational fluent $F(\mathbf{t}, do(a,\sigma))$ (with $F(\mathbf{x}, do(a,s)) \equiv \Phi_F(\mathbf{x}, a, s)$ being $F$'s successor state axiom) one determines a logically equivalent formula $W'$ by substituting $\Phi_F(\mathbf{t}, \alpha, \sigma)$ for $F(\mathbf{x}, do(a,\sigma))$. This way, the regression operator $\mathcal{R}$ reduces complex situation terms to terms that only mention $S_0$.

The regression theorem [30] states that $\mathcal{D} \models W$ iff $\mathcal{D}_{S_0} \cup \mathcal{D}_{una} \models \mathcal{R}[W]$, with $W$ a regressable sentence of $\mathcal{L}_{sitcalc}$ and $\mathcal{D}$ a basic action theory. This means that the evaluation of regressable sentences can be reduced to a theorem proving task in the initial theory $\mathcal{D}_{S_0}$ together with unique names axioms for actions. No successor state, precondition or foundational axioms are needed for this task.

### 3.2 The Robot Programming and Plan Language Readylog

Readylog [16, 17] is our variant of Golog [31] and also makes use of Reiter's BATs as described above. The aim of designing the language Readylog was to create a Golog dialect which supports the programming of the high-level control of agents or robots in dynamic real-time domains such as domestic environments or robotic soccer. Readylog borrows ideas from [6, 9, 23, 24, 31] and features the following constructs (see also Fig. 1): (1) sequence $(a; b)$, (2) nondeterministic choice between actions $(a|b)$, (3) solve a Markov Decision Process (MDP) ($solve(p,h)$, $p$ is a Golog program, $h$ is the MDP's solution horizon), (4) test actions $(?(c))$, (5) event-interrupt $(waitFor(c))$, (6) conditionals $(if(c, a_1, a_2))$, (7) loops $(while(c, a_1))$, (8) condition-bounded execution $(withCtrl(c, a_1))$, (9) concurrent execution of programs $(pconc(p_1, p_2))$, (10) probabilistic actions $(prob(val_{prob}, a_1, a_2))$, (11) probabilistic (offline) projection $(pproj(c, a_1))$, and (12) procedures $(proc(name(parameters), body))$. The idea of Golog to combine planning with programming was accounted for in Readylog by integrating decision-theoretic planning; only partially specified programs which leave certain decisions open, which then are taken by the controller based on an optimization theory, are needed.

A nice feature of Golog and Readylog is that its semantics is based on the situation calculus. That means that both languages have a formal semantics and properties of programs can be proved formally. We do not want to get in to the details of the formal semantics here, as only little of it is needed to understand the examples in Sect. 5. Further, we did not introduce formal concepts of stochastic actions and several other concepts that Readylog makes use of. We refer the interested reader to [17] for the complete formal definition of the language. Golog languages come with run-time interpreters usually programmed in Prolog. Also, a Readylog implementation is available in Prolog. However, we are also working towards developing non-prolog implementations for Golog [15, 20].

[1] Sentences uniform in $s$ are sentences which do not quantify about situations, nor mention *Poss* or $\sqsubset$. "$\sqsubset$" stands for an ordering relation on situations and is needed in the foundational axioms (see e.g. [44]).

| | |
|---|---|
| *nil* | empty program |
| $\alpha$ | primitive action |
| $\varphi?$ | wait/test action |
| $waitFor(\tau)$ | event-interrupt |
| $[\sigma_1; \sigma_2]$ | sequence |
| **if** $\varphi$ **then** $\sigma_1$ **else** $\sigma_2$ **endif** | conditional |
| **while** $\varphi$ **do** $\sigma$ **endwhile** | loop |
| **withCtrl** $\varphi$ **do** $\sigma$ **endwithCtrl** | guarded execution |
| $\sigma_1 \mid\mid \sigma_2$ | prioritized execution |
| **forever do** $\sigma$ **endforever** | infinite loop |
| **whenever**$(\tau, \sigma)$ | interrupt triggered by continuous function |
| **withPol**$(\sigma_1, \sigma_2)$ | prioritized execution until $\sigma_2$ ends |
| **prob**$(p, \sigma_1, \sigma_2)$ | probabilistic execution of either $\sigma_1$ or $\sigma_2$ |
| **interrupt** | interrupts |
| $pproj(c, \sigma)$ | probabilistic (off-line) projection |
| $\{$**proc** $P_1(\boldsymbol{\vartheta}_1)\sigma_1$ **endproc**$; \cdots ;$ **proc** $P_n(\boldsymbol{\vartheta}_n)\sigma_n$ **endproc**$\}; \sigma_0$ | procedures |
| $solve(h, f, \sigma)$ | initiate decision-theoretic optimization over $\sigma$ |
| $\sigma_1 \mid \sigma_2$ | nondeterministic (dt) choice of programs |
| $(\pi\ \mathbf{x})[\sigma]/pickBest(\mathbf{x}, \sigma, h)$ | nondeterministic (dt) choice of arguments |

**Fig. 1** Overview of Readylog constructs

### 3.3 Fuzzy Sets

A *crisp set* $A$ over a universe of discourse $U$ can be defined by a characteristic function $\mu_A$ as $\mu_A = 1$, if $x \in A$, and $\mu_A = 0$, otherwise. For two sets $A, B \subset U$, the union $A \cup B$ is defined as $\mu_{A \cup B}(x) = 1$ if $x \in A$ or $x \in B$, and $\mu_{A \cup B}(x) = 0$ if $x \notin A$ and $x \notin B$. For the intersection $A \cap B$ it holds: $\mu_{A \cap B} = 1$ if $x \in A$ and $x \in B$, and $\mu_{A \cap B} = 0$ if $x \notin A$ or $x \notin B$. The complement $\bar{A}$ is defined such that $\mu_{\bar{A}}(x) = 1$ if $x \notin A$; $\mu_{\bar{A}}(x) = 0$ if $x \in A$. Crisp set operations enjoy the property of being commutative, associative, and distributive. Further, De Morgan's laws as well as the Law of Contradiction and Excluded Middle hold.

A *fuzzy set* $F$ with a universe of discourse $U$, is characterized by a membership function $\mu_F : U \to [0, 1]$. The membership function provides a measure of the degree of similarity of an element in $U$ to the fuzzy set. A fuzzy set $F$ in $U$ can be represented as a set of ordered pairs of a generic element $x$ and its grade of membership: $F = \{(x, \mu_F(x)) | x \in U\}$. Union, intersection, and complement can be defined the same way as for crisp sets. Note that unlike for crisp sets, the Law of Contradiction and the Excluded Middle do *not* hold, i.e. $A \cup \bar{A} \neq U$ and $A \cap \bar{A} \neq \emptyset$. In general, for set intersection several different so-called t-norms have been proposed, usually written as $\mu_{R \cap S}(x) = \mu_R(x) \star \mu_S(x)$, for set union t-conorms or s-norms, written as $\mu_{R \cup S}(x) = \mu_R(x) \oplus \mu_S(x)$ were formulated (see e.g. [11]). Here, we rely on the min t-norm $\mu_{A \cap B}(x) = \min[\mu_A(x), \mu_B(x)]$, the max t-conorm $\mu_{A \cup B}(x) = \max[\mu_A(x), \mu_B(x)]$ and the complement defined as $\mu_{\bar{A}}(x) = 1 - \mu_A(x)$.

To be able to conclude something useful with fuzzy variables, we need inference rules which define how to reason with variables of this kind. In the following, we give several examples of the type of inference possible by stating examples from [55]: (1) *Categorical rules* like *X is small*; (2) *Entailment rules* like *Mary is very young* and *very young implies young* implies *Mary is young*; (3) *Conjunc-*

*tion/Disjunction rules* like *the pressure is not very high* and/or *the pressure is not very low* implies *the pressure is not very high and/or not very low*; (4) *Compositional rules* like $X$ *is much larger than* $Y$ and $Y$ *is large* implies $X$ *is much larger* $\circ$ *large*; (5) *Negation rules* like *not(Mary is young)* implies *Mary is not young*; (6) *Extension principle* like $X$ *is small* implies $X^2$ *is* $^2$*small* with $^2$small meaning *very small*. Another way to see these rules is by interpreting $X$ and $Y$ as decision variables and $A$ as a soft constraint, allowing for a degree of membership. If there are only two membership values (true or false), then these constraints can be regarded as hard constraints (see e.g. [11]).

## 4 Qualitative Representations with Fuzzy Sets in the Situation Calculus

In this section we introduce fuzzy sets into the language of the situation calculus and introduce fuzzy fluents and the concept of membership in Sect. 4.1, extending our previous work [18]. We define the centre-of-gravity defuzzifier to handle fuzzy fluents, before we introduce qualitative fuzzy fluents for positional information in Sect. 4.2. We finish this section with a one-dimensional example displaying the concept of qualitative distance in Sect. 4.3.

### 4.1 Fuzzy Fluents

**Definition 1 (Reals and Linguistic Terms)** We introduce two new sorts to the situation calculus: *real* and *linguistic*. We do not axiomatize reals here, and assume their standard interpretation together with the usual operations and ordering relations. *Linguistic* terms are a finite set of constant symbols $c_1, \dots, c_k$ in the language. They refer to qualitative classes; examples are close or far. We further require a unique names assumption for these linguistic categories. ■

Now, having introduced reals and linguistic terms into the language of the situation calculus, we can define the degree of membership of a particular value to a given category. For ease of notation we assume that the domain of a particular category is from the domain of real numbers. In general, the domain can be defined arbitrarily.

**Definition 2 (Fuzzy Sets)** Let $c_1, \dots, c_k$ be categories of sort *linguistic*. We introduce a relation $\mathfrak{F} \subseteq$ *linguistic* $\times$ *real* $\times [0,1]$ relating each linguistic term $c$ of the domain, a real number, and a degree of membership in the category $c$ as

$$\begin{aligned} &\forall c, u, \mu_u . \mathfrak{F}(c, u, \mu_u) \equiv \\ &\quad (c = c_1 \supset u = u_{c_1,0} \wedge \mu_u = \mu_{c_1,0} \vee \dots \vee u = u_{c_1,m_1} \wedge \mu_u = \mu_{c_1,m_1}) \vee \dots \vee \\ &\quad (c = c_k \supset u = u_{c_k,0} \wedge \mu_u = \mu_{c_k,0} \vee \dots \vee u = u_{c_k,m_k} \wedge \mu_u = \mu_{c_k,m_k}), \end{aligned}$$

where all $u_{c_i,j}$ and $\mu_{c_i,j}$ are constants of sort real and $\mu_{c_i,j} \in [0,1]$ respectively, i.e. $\forall c, u, \mu_u . \mathfrak{F}(c, u, \mu_u) \supset 0 \leq \mu_u \leq 1$. To ensure that, for each category, each pair $(u, \mu_u)$ is unique, we require unique names for linguistic terms: $\forall c \exists u, \mu_u \forall \mu_{u'} . \mathfrak{F}(c, u, \mu_u) \wedge \mathfrak{F}(c, u, \mu_{u'}) \supset \mu_u = \mu_{u'}$. We further require one of the $u_{c_i,j}$ to equal the centre-of-gravity of the respective category, i.e. $u_{c_i,j} = cog(c_i)$ (cf. Def. 5). ■

Note that the above definition yields a formalization of discrete fuzzy sets as described in Sect. 3.3. That means that all value-membership pairs belonging to a particular linguistic category are enumerated. While we regard a discrete formalization here, our examples (Sect. 4.3) and the implementation make use of a continuous formulation of fuzzy sets. Further note that variables occurring free in the logical sentences are implicitly universally quantified in the following definitions. An example of fuzzy sets (Sect. 4.3) is deferred until we introduced fuzzy fluents.

**Definition 3 (Fuzzy Fluent)** A fuzzy fluent $\mathfrak{f}$ is a functional fluent restricted to take only values from sort *linguistic* or from sort *real*. We write $\mathfrak{f}(\mathbf{x}, s)$ to refer to a fuzzy fluent, and $f(\mathbf{x}, s)$ to refer to a non-fuzzy fluent. ■

To query whether or not a fluent value belongs to a certain category, we introduce, similar to fuzzy control theory, predicates is, $\text{is}_{\complement}$, $\text{is}_{\star}$, and $\text{is}_{\oplus}$. These predicates are true if a fuzzy fluent value belongs to the category in question to a non-zero degree.

**Definition 4 (Membership)**

1. To query if a fuzzy fluent belongs to a given category, we define the predicate is $\subseteq$ *real* $\times$ *linguistic* as

$$\text{is}(\mathfrak{f}(\mathbf{t}, \sigma), \gamma) \doteq \exists u, \mu_u . \mathfrak{f}(\mathbf{t}, \sigma) = u \wedge \mathfrak{F}(\gamma, u, \mu_u) \wedge \mu_u > 0$$

2. Similarly, we define $is_{\complement} \subseteq$ *real* $\times$ *linguistic*, to know if a fuzzy fluent does *not* belong to a certain category

$$\begin{aligned}\text{is}_{\complement}(\mathfrak{f}(\mathbf{t}, \sigma), \gamma) \doteq{}& \neg\exists u, \mu_u . \mathfrak{f}(\mathbf{t}, \sigma) = u \wedge \mathfrak{F}(\gamma, u, \mu_u) \vee \\ & \exists u, \mu_u . \mathfrak{f}(\mathbf{t}, \sigma) = u \wedge \mathfrak{F}(\gamma, u, \mu_u) \wedge \mu_u < 1.\end{aligned}$$

   A fluent value does not belong to a certain category, if either the value in question is not defined in terms of a fuzzy set, or the value exists and its degree of membership is less than 1.
3. For complex queries, for example if a fuzzy fluent value belongs to several overlapping categories at the same time, we define a predicate $\text{is}_{\star} \subseteq$ *real* $\times$ $(\textit{linguistic})^n$ for arbitrary $n$ as

$$\begin{aligned}\text{is}_{\star}(\mathfrak{f}(\mathbf{t}, \sigma), \gamma_0, \ldots, \gamma_n) \doteq{}& \exists u, \mu_{u,0}, \ldots, \mu_{u,n} . \mathfrak{f}(\mathbf{t}, \sigma) = u \wedge \mathfrak{F}(\gamma_0, u, \mu_{u,0}) \\ & \wedge \cdots \wedge \mathfrak{F}(\gamma_n, u, \mu_{u,n}) \wedge (\mu_{u,0} \star \cdots \star \mu_{u,n} > 0).\end{aligned}$$

4. Similarly, for asking whether or not a fuzzy fluent value belongs to one category or the other, we introduce the predicate $is_{\oplus} \subseteq$ *real* $\times$ $(\textit{linguistic})^n$

$$\begin{aligned}\text{is}_{\oplus}(\mathfrak{f}(\mathbf{t}, \sigma), \gamma_0, \ldots, \gamma_n) \doteq{}& \exists u, \mu_{u,0}, \ldots, \mu_{u,n} . \mathfrak{f}(\mathbf{t}, \sigma) = u \wedge \mathfrak{F}(\gamma_0, u, \mu_{u,0}) \\ & \wedge \cdots \wedge \mathfrak{F}(\gamma_n, u, \mu_{u,n}) \wedge (\mu_{u,0} \oplus \cdots \oplus \mu_{u,n} > 0).\end{aligned}$$

■

Note that the $\sigma$'s in the above definition are used as a meta-variable for terms of sort situation. How the predicate "is" is used to query whether or not a fuzzy fluent belongs to a qualitative category will be shown in our one-dimensional robot domain in Sect. 4.3.

First, we need to define a defuzzifier, a function that computes a single numerical value for a given linguistic category. Note that, while we choose the centre-of-gravity defuzzifier here, our approach is not restricted to this. Instead, any defuzzifier could be used just as well. Then, we define a defuzzifying function that selectively applies the defuzzifier to any linguistic term.

**Definition 5 (Defuzzifying Qualitative Category Values)** Let $\tau$ be a term of $\mathcal{L}_{sitcalc}$. We define a function *defuzz* inductively as:

1. if $\tau$ is an atomic term
   (a) and $\tau$ is of sort *linguistic*, then $\mathit{defuzz}(\tau) = \mathit{cog}(\tau)$
   (b) otherwise $\mathit{defuzz}(\tau) = \tau$
2. if $\tau$ is a non-atomic term of the form $f(\mathbf{t})$ with $\mathbf{t} = t_1, \ldots, t_n$,
   then $\mathit{defuzz}(\tau) = f(\mathit{defuzz}(t_1), \ldots, \mathit{defuzz}(t_n))$

In *defuzz* we make use of the function *cog*, which defines the centre-of-gravity defuzzifier. It is defined as:

$$
\begin{aligned}
&\mathit{cog}(c) = \hat{u} \equiv \\
&\exists u_0, \ldots, u_k, \mu_{u_0}, \ldots, \mu_{u_k}. \mathfrak{F}(c, u_0, \mu_{u_0}) \wedge \cdots \wedge \mathfrak{F}(c, u_k, \mu_{u_k}) \wedge \\
&\quad u_0 \neq \cdots \neq u_k \wedge \forall u^*, \mu^*. (u^* \neq u_0 \wedge \cdots \wedge u^* \neq u_k \wedge \\
&\qquad \mu^* \neq \mu_{u_0} \wedge \cdots \wedge \mu^* \neq \mu_{u_k} \supset \neg \mathfrak{F}(c, u^*, \mu^*)) \wedge \\
&\quad \hat{u} = \sum_{i=0}^{k} u_i \cdot \mu_{u_i} \Big/ \sum_{i=0}^{k} \mu_{u_i}
\end{aligned}
$$

■

A fuzzy fluent's function value is eventually substituted by its defuzzified value when applying the defuzzifying function *defuzz*. This can, for example, be used in a fluent's successor state axiom as we show in the example in Section 4.3. Note that the number $k$ in the definition of the centre-of-gravity defuzzifier above refers to the number of all value-membership pairs defined in the fuzzy set for the linguistic categories plus one value for each category itself (Def. 2). Further note that the number of value-membership pairs is required to be finite. As the above definition of a defuzzifier is not closed under division in general, note that the definition is however well-defined. This is because we postulate that the centre-of-gravity for a qualitative category will be added to the set explicitly. In our implementation, where we make use of continuous fuzzy sets, this requirement can be dropped, as the set then is closed under division.

By now, we defined fuzzy fluents as a specialization of functional fluents operating on reals and linguistic terms, introduced qualitative categories as constants of sort linguistic, and defined a fuzzy set in our domain axiomatization which allows for defining which values make up a qualitative category to which degree. We can further query whether or not a fuzzy fluent belongs to a qualitative category. Moreover, we can ask if a fuzzy fluent belongs to several categories at the same time, or if it belongs to the complementary category.

We have now defined everything we need to reason with qualitative predicates based on fuzzy membership functions. After we introduce qualitative positional information in the next section we illustrate an application with an example in Section 4.3.

### 4.2 Qualitative Positional Information

Based on a representation mechanism for qualitative orientation presented by Hernandez in [25] and a basic method for qualitative distances discussed in 1995 in [26] by Hernandez, Clementini, and Felici in their 1997 paper [7], Clementini et al. present a unified framework which allows for qualitative representation of positional information. This is done by combining the orientation and the distance relation. The position of a *primary object po* is represented by a pair of distance and orientation relations with respect to *ro*, a *reference object*. Both relations depend on a so-called *frame of reference* which accounts for several factors such as the size of objects and different points of view. Thus, to represent positional information, we need to define an orientation and a distance relation.

The orientation relation describes where objects are placed relatively to each other. Based on the fundamental observation of how three points in the plane relate to each other, an orientation relation can be defined in terms of three basic concepts: the *primary object po*, the *reference object ro*, and the *frame of reference* which contains the *point of view*. The point of view and the reference object are connected by a straight line. The view direction is then determined by a vector from the point of view to the reference object. The location of a primary object is expressed with regard to the view direction as one of a set of relations. The number of distinctions made is determined by the *level of granularity*. There are different levels of granularity for orientation relations. On the first level, the point of view and the reference object are connected by a straight line such that the primary object can be to the left, to the right, or on that line. Thus, the first level partitions the plane into two half-planes. On the second level, there would be four partitions, the third level would have eight, and so on (cf. Figs. 2(a) and 2(b)). Based on the frame of reference there is a 'front' side of the reference object. Independent from the level of granularity there is a uniform circular neighbouring structure. In general, at a level of granularity $k$ the set $\{\alpha_0, \alpha_1, \ldots, \alpha_n\}$ denotes the $n + 1$ orientation relations where $n = 2^k - 1$.

The distance relation requires the three elements *ro*, *po*, and the frame of reference. Moreover, a distance relation requires a distance system. A commonly used distance system is the Euclidean space, which is reflexive ($dist(P_1, P_1) = 0$), symmetric ($dist(P_1, P_2) = dist(P_2, P_1)$), and follows the triangle inequality ($dist(P_1, P_2) + dist(P_2, P_3) \leq dist(P_1, P_3)$). The distance of two points expressed in a qualitative way often depends not only on their absolute positions but also on cultural and experimental factors and on the frame of reference. Similar to the orientation relation we can distinguish distances at various levels of granularity (cf. Figs. 2(c) and 2(d)). An arbitrary level $n$ of granularity with $n+1$ distinctions yields the set $Q = \{q_0, q_1, \ldots, q_n\}$ of qualitative distances. Given a reference object *ro*, these distances partition the space around *ro* such that $q_0$ is the distance closest to *ro* and $q_n$ the one farthest away.

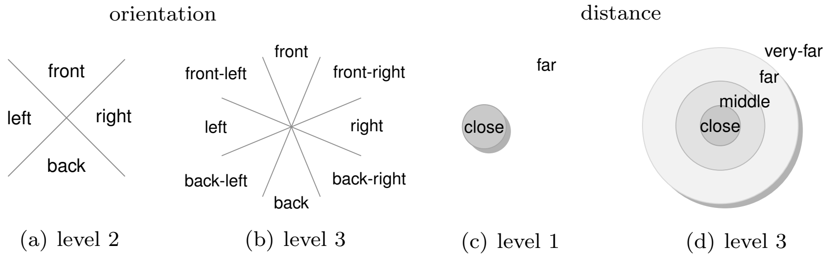



**Fig. 2** Different levels of granularity for orientation and distance according to [7]

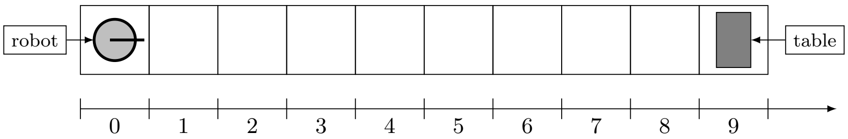


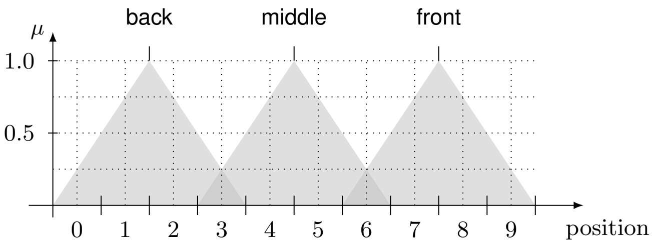

**Fig. 3** The one-dimensional domestic robot world.

From a quantitative point of view, the combined description of a position with the above model using distance and orientation can be seen as the representation of a point in polar coordinates. A point $p$ in polar coordinates is defined by the distance $r$ from the origin to this point and the angle $\varphi$ measured from the horizontal $x$-axis to the line from the origin to $p$ in the counter-clockwise direction. Thus, the position of a point $p$ is described as $(r, \varphi)$. This description directly corresponds to the combination of the distance and the orientation relation.

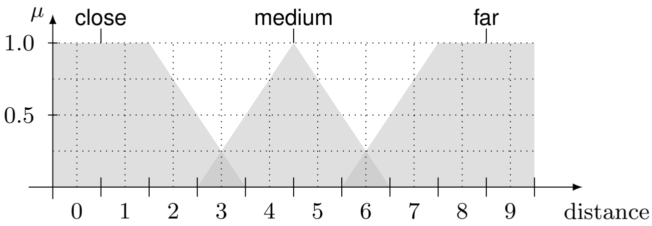
We will make use of this correspondence by relating qualitative positional information given as distance and orientation in the above sense to quantitative positions in Euclidean space. This way we can simply use methods from Euclidean geometry to conduct spatial reasoning. In our example in Section 5.4 we consider simple cases for reasons of simplicity. In future work, however, the reasoning may go beyond transforming one qualitative description to more complicated cases such as computing the composition of two positional relations and alike. For this to work properly, we need to consider some more information associated to positional information in different contexts. Before we will detail this in Sect. 5, we give a one-dimensional example to show how our integration of qualitative positional information in the situation calculus works in the next section.

### 4.3 A One-dimensional Example

To illustrate reasoning with qualitative positional information using linguistic terms and the representations introduced above, consider the following simple example. A robot is situated in a one dimensional room with a length of ten metric units as depicted in Fig. 3. To keep things simple, we restrict ourselves to integer values for positions in the following. We have one single action called gorel($d$) denoting the relative movement of $d$ units of the robot in its world. For sake of simplifying the notation in this example, we assume that this action is always possible, i.e. $Poss(\text{gorel}(d), s) \equiv \top$. The action has impact on the fluent *pos* which




(a) Qualitative positions of a robot in a one-dimensional world.




(b) Qualitative distance in the one-dimensional world.

**Fig. 4** Membership functions for position and distance in our one-dimensional robot domain.

denotes the absolute position of the robot in the world. The position of the table is defined by the macro $pos_{table} = p \doteq p = 9$. In the initial situation, the robot is located at position 0, i.e. $pos(S_0) = 0$. We partition the distance in categories close, medium, and far, and introduce qualitative categories for the position of the robot as back, middle, and front. We give the (fuzzy) definition of those categories below, where we use $(u_i, \mu_{u_i})$ as an abbreviation for $u = u_i \wedge \mu = \mu_{u_i}$. The fuzzy categories for the position of the robot in the world can be defined as

$$
\begin{aligned}
&\mathfrak{F}(position, u, \mu_u) \equiv \\
&\quad (position = \mathsf{back} \supset (0, 0.25) \vee (1, 0.75) \vee (2, 0.75) \vee (3, 0.25) \vee (3/2, 0.5)) \vee \\
&\quad (position = \mathsf{middle} \supset (3, 0.25) \vee (4, 0.75) \vee (5, 0.75) \vee (6, 0.25) \vee (9/2, 0.5)) \vee \\
&\quad (position = \mathsf{front} \supset (6, 0.25) \vee (7, 0.75) \vee (8, 0.75) \vee (9, 0.25)),
\end{aligned}
$$

while the distances can take the values

$$
\begin{aligned}
&\mathfrak{F}(distance, u, \mu_u) \equiv \\
&\quad (distance = \mathsf{close} \supset (0, 1.0) \vee (1, 1.0) \vee (2, 0.75) \vee (3, 0.25) \vee (13/12, 0.5)) \vee \\
&\quad (distance = \mathsf{medium} \supset (3, 0.25) \vee (4, 0.75) \vee (5, 0.75) \vee (6, 0.25) \vee (9/2, 0.5)) \vee \\
&\quad (distance = \mathsf{far} \supset (6, 0.25) \vee (7, 0.75) \vee (8, 1.0) \vee (9, 1.0) \vee (95/12, 0.5)).
\end{aligned}
$$

For readability reasons, we assume in this example that the robot can only move around in integer steps. Restricting to integers requires to use an altered version $cog'(c)$ of the centre-of-gravity defuzzifier formula: $cog'(c) \doteq \lfloor cog(c) \rfloor$. Of course our function *defuzz* has to mention $cog'$ instead of $cog$ then. A graphical illustration of the membership functions for position and distance is given in Fig. 4.

In the definition of the successor state axiom of the fluent *pos*, we have to handle its qualitative categories. We need to apply the function $defuzz(c)$ to the qualitative term which yields always a quantitative representative:

$$\begin{aligned} &pos(do(a,s)) = y' \equiv y' = defuzz(y) \wedge \\ &\quad a = \text{gorel}(d) \wedge y = pos(s) + d' \vee a \neq \text{gorel}(d) \wedge y = pos(s). \end{aligned}$$

We want to evaluate the robot's position and its distance to the table. Therefore we define a functional fluent *dist* which returns the distance between the robot and the table:

$$\begin{aligned} &dist(do(a,s)) = d \equiv \\ &\quad \exists p_1 . pos_{table} = p_1 \wedge \exists p_2 . pos(do(a,s)) = p_2 \wedge d = p_1 - p_2. \end{aligned}$$

Now that we have linguistic terms for position and distance, we want to showcase that qualitative statements can in fact be used easily and that their integration in our reasoning framework yields correct results for the same. For that, we consider three main uses of qualitative notions: (1) in specifications of the initial situation, (2) linguistic terms in action arguments, and (3) fluents taking a qualitative category as the result of an action.

#### *4.3.1 Linguistic Terms in the Initial Situation*

Suppose the robot's position in situation $S_0$ is characterized by the linguistic term $\mathsf{back}$ and the table is located at position 9, i.e. $\mathcal{D}_{S_0} = \{pos(S_0) = \mathsf{back}, dist(S_0) = 9\}$. Suppose now that the robot travels 4 units to the right. Then we can show that

$$\mathcal{D} \models \text{is}(pos(do(\text{gorel}(4), S_0)), \mathsf{middle}) \wedge \text{is}(dist(do(\text{gorel}(4), S_0)), \mathsf{medium}).$$

*Proof Sketch* Using regression and the successor state axiom for the fluent *dist* we apply the centre-of-gravity $cog'(\mathsf{back}) = 1$ if the value of a fuzzy fluent is a linguistic term in the initial situation. It thus holds in $S_0$ that $\mathcal{D} \models \text{is}(dist(S_0), \mathsf{far})$. By performing the action gorel(4) the robot moves four positions to the right. The proposition holds because $pos(do(\text{gorel}(4), S_0)) = 1 + 4 = 5$ and $\mathfrak{F}(\mathsf{middle}, 5, 0.75)$ has a non-zero membership value. The quantitative distance from 5 to 9 equals 4 units or $\mathsf{medium}$ distance, as is given by $\mathfrak{F}(\mathsf{medium}, 4, 0.75)$. □

#### *4.3.2 Qualitative Statements in Action Arguments*

Suppose now that the robot's control program contains the action gorel($\mathsf{far}$) mentioning the qualitative term $\mathsf{far}$. At which position will the robot end up in situation $s = do(\text{gorel}(\mathsf{far}), S_0)$? It follows that

$$\mathcal{D} \models \text{is}(pos(do(\text{gorel}(\mathsf{far}), S_0)), \mathsf{front})$$

i.e. the robot ends up in the $\mathsf{front}$ part of its world after executing gorel($\mathsf{far}$).

*Proof Sketch* Determining the robot's position in situation $do(\text{gorel}(\mathsf{far}), S_0)$ we again use regression. It is sufficient to show that $\mathcal{D}_{S_0} \models \text{is}(\mathcal{R}[pos(do(gorel(\mathsf{far}), S_0))], \mathsf{front})$ which is—according to the successor state axiom above—regressed to $pos(S_0) = cog'(\mathsf{back}) \wedge \mathfrak{F}(\mathsf{far}, 7, 0.75) \wedge d' = cog'(\mathsf{far}) \wedge \text{is}(y = cog'(\mathsf{back}) + cog'(\mathsf{far}), \mathsf{front}) \equiv \text{is}(y = 1 + 7, \mathsf{front}) \equiv y = 8 \wedge \mathfrak{F}(\mathsf{front}, 8, 0.75) \wedge 0.75 > 0$. Hence, we can infer that the robot ends up at position $\mathsf{front}$. □

### *4.3.3 Using Qualitative Categories for Fluents as a Result of an Action*

Assume that apart from gorel($x$) there is another action go($x$) which makes the robot move directly to position $x$. The successor state axiom of go($x$) is given as $pos(do(a,s)) = y \equiv a = \mathrm{go}(x) \wedge y = x \vee a \neq \mathrm{go}(x) \wedge y = pos(s)$. What happens if we put in a qualitative category there, i.e. at which position will the robot end up in situation $s = do(\mathrm{go}(\mathsf{front}), S_0)$? It turns out that we have

$$\mathcal{D} \models \mathrm{is}(pos(do(\mathrm{go}(\mathsf{front}), S_0)), \mathsf{front})$$

i.e. the robot ends up in the front part of its world after executing go(front).

*Proof Sketch* When regressing a formula that contains a linguistic term, the defuzzification function (e.g. centre-of-gravity $cog'(c)$) is applied if the result of a previous successor state axiom assigned a qualitative term to the fuzzy fluent. Then, $\mathcal{D} \models \mathrm{is}(pos(do(\mathrm{go}(\mathsf{front}), S_0)), \mathsf{front})$ iff $\mathcal{D}_{S_0} \models \mathrm{is}(\mathcal{R}[pos(do(\mathrm{go}(\mathsf{front}), S_0))], \mathsf{front})$ which is regressed to $pos(S_0) = cog'(\mathsf{back}) \wedge x = \mathsf{front} \wedge u = cog'(\mathsf{front}) \wedge \mathrm{is}(u, \mathsf{front}) \equiv \mathrm{is}(u = 7, \mathsf{front}) \equiv u = 7 \wedge \mathfrak{F}(\mathsf{front}, 7, 0.75) \wedge 0.75 > 0$. Thus, it can be inferred that the robot will end up at position front. □

## 5 An Extension for Spatial Domestic Environments

In this section, we extend the qualitative notations of the previous section with information about their spatial context. In Sect. 5.1, we introduce the domestic robot domain, before we introduce and formalize the concept of frames in Sect. 5.2. Sect 5.3 shows the formalization of the domain in the situation calculus. In particular, we show how different reference frames can be transformed into each other and how we deploy unit fuzzy sets to formalize the concept of distance and orientation in the domestic domain. In Sect. 5.4 we present a high-level Readylog controller for the fetch-and-carry task in the domestic domain. There, we show (a) how domestic high-level controllers can be formulated in Readylog in a straightforward way, and (b) how the qualitative information are seamlessly integrated into the high-level program of the robot.

### 5.1 The Domestic Robot Domain

Our target domain is the Domestic Robot Domain. In this domain, a service robot is instructed by a human operator via natural language to fulfil tasks such as *Fetch&Carry* in an apartment environment. Fig. 5 shows an example domain. In this domain, there are several rooms and several pieces of furniture or objects. The human-machine interaction should be as natural as possible. Therefore, our goal is to support instructions such as "*get me the left cup on table2*" or "*bring me a coke to the living room*". A closer look at such instructions reveals that mainly qualitative positional information is used by the human instructor, i.e. qualitative distance and orientation. Further, to be able to put the information given by the human instructor into the right context, the concept of a frame of reference possibly including a point of view mentioned already is required. For example, "far" in the context of the living room refers to a larger distance than "far" in

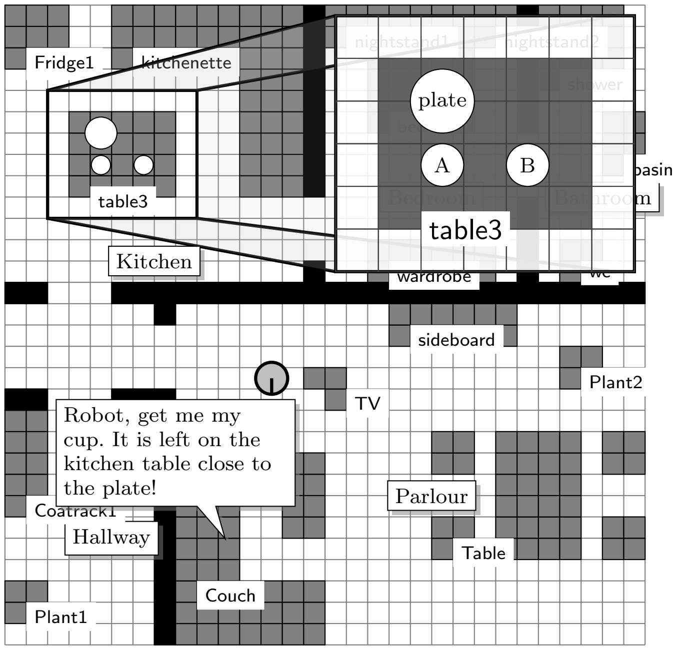


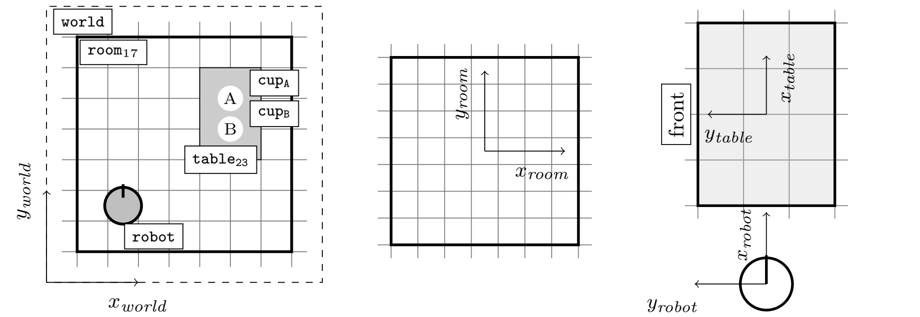

**Fig. 5** The domestic robot domain

the bath room. To cope with these types of contexts and different points of view, we introduce the concept of *frames* in the next section and formalize them in the situation calculus.

For positional information in our domestic domain, we need qualitative distance and orientation which we already introduced in Section 4.2. However, it makes sense to fix things like the distance system and the number of granularity levels for the indoor environment. We do so by specifying membership functions for distance and orientation, respectively. That means we fix a level of granularity of 5 for the distance relation with the categories very-close, close, medium-far, far, and very-far. Of course, in different frames, these categories have different scales. As the parlour is larger than the bath room, we have to scale these categories according to their respective frame. Our concept of *frame* is doing exactly that. This, however, requires, that the categories for distance are defined on a unit scale. We show our definitions of qualitative unit distance in Sect. 5.3. For the qualitative orientation, we select a level of granularity of 3, meaning that we distinguish $2^3 = 8$ different qualitative orientations. In order to relate the orientation to its context, each frame needs to have a distinguished front side. Our concept of *frame* in the next sections covers this as well.

## 5.2 Positional Fuzzy Fluents and Positional Frames

For each domain object which should be reasoned about in a qualitative fashion, we need to know the object's coordinate and its reference frame. For example, $\texttt{table}_{23}$ could be either defined to be at a certain position in room $\texttt{room}_{17}$, or its position could be instantiated with a global world coordinate. In the former case,




(a) World or global frame (b) Room frame (c) Table and robot frame

**Fig. 6** Examples of different frame in our domain.

we would attach frame *room*$_{17}$ to the positional information of the table, in the latter we would attach frame *world*. Fig. 6 is showing different example frames. Attaching its frame to an object allows for transforming the object's coordinates to any other given frame.

Associated with a frame is a local Cartesian coordinate system $\mathbb{R}^2$ of an object in the world. The origin of a frame's coordinate system is defined with respect to a super-ordinated frame, with the *world* being the most general frame. For each frame $f_s$ denoting the source frame, we need to specify, how the frame's coordinate system can be transformed into the target frame $f_t$. The parameters we need are the origin $(x_o, y_o)$ of the target system expressed in the coordinates of the source frame, the angle $\theta_o$ between the source and the target frame and $m_o$ a scaling factor between the units of the respective distance systems.

As objects can be moved in the world, these parameters are not rigid and have to be defined in terms of fluents. We therefore require without loss of generality that for each object in the world, there is a sentence of the following form in $\mathcal{D}_{ssa}$:

$$\mathit{frmparam}(f_s, f_t, s) = (x_o, y_o, \theta_o, m_o) \doteq \mathit{pos}_{f_s}(f_t, s) = (x_o, y_o, \theta_o, m_o)$$

with $\mathit{pos}_{f_s}$ defining the position, angle, and scaling factor of an object $f_s$ expressed in the frame $f_t$. Note that *frmparam* is defined as a macro. The domain axiomatizer has to provide sentences about the initial position of an object and a successor state axiom describing how the position of the respective object changes. Further note that the basic action theory is just used, not extended. Therefore, all results for BATs still apply.

To give an example, consider Fig. 6(a). To express the position of the table in the world, we need to add the sentence $\mathit{pos}_{table_{23}}(room_{17}, S_0) = (5, 4, 0, 1)$ to $\mathcal{D}_{S_0}$ and $\mathit{frmparam}(table_{23}, room_{17}, s) = (x_o, y_o, \theta_o, m_o) \doteq \mathit{pos}_{table_{23}}(room_{17}, s) = (x_o, y_o, \theta_o, m_o)$ to $\mathcal{D}_{ssa}$. Now, we can keep track of the position of the table w.r.t. its coordinate in the kitchen. Defining the position of objects w.r.t. a superordinate frame also allows to derive, say, the coordinate of a cup on the kitchen table, even if the table was moved around in the kitchen.

To convert between a source and a target frame, we define a function *chfrm* as

$$
\begin{aligned}
&\mathit{chfrm}(x_s, y_s, f_s, f_t, s) = (x, y) \doteq \\
&\quad \exists x_o, y_o, \theta_o, m_o . \mathit{frmparam}(f_s, f_t, s) = (x_o, y_o, \theta_o, m_o) \wedge \\
&\quad [f_s \neq f_t \wedge (x, y)^T = \begin{pmatrix} \cos\theta_o & -\sin\theta_o \\ \sin\theta_o & \cos\theta_o \end{pmatrix} \cdot \begin{pmatrix} x_o + x_s \\ y_o + y_s \end{pmatrix} \vee \\
&\quad f_s = f_t \wedge (x, y) = (x_s, y_s)]
\end{aligned}
$$

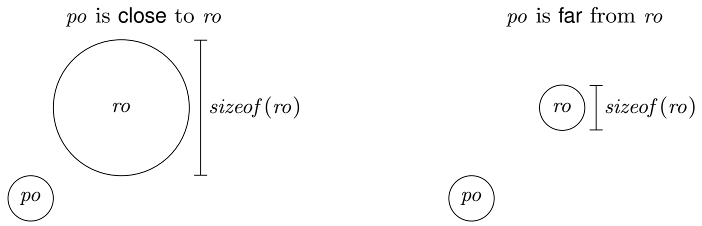
The scaling factor $m_o$ is used to determine the unit lengths of each interval of the distance system. Each frame requires a distinct front side in order to provide a standard point of view. We assume the standard view point along the $y$-axis of the frame's coordinate system.

Note, that the above definition of *chfrm* directly corresponds to a coordinate transformation in Euclidean space in terms of translation and rotation as well as scaling. This is exactly the correspondence that we exploit to retain a simple way to reason with positional information. Enabling the direct relation between the qualitative description and a coordinate based numerical representation does exactly this.

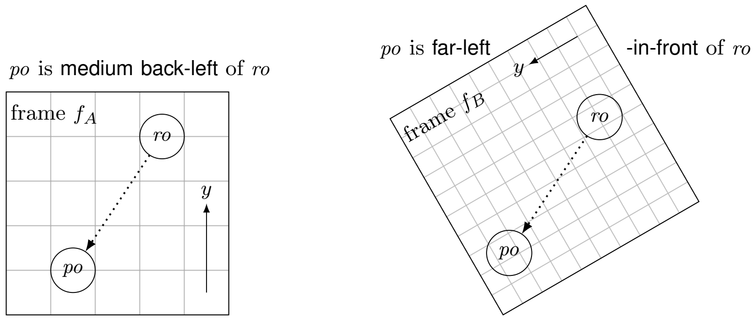
Before we show an extended example of the coordinate transformations making use of *frmparam* and *chfrm* in the next section, we show that our concept of *frame* satisfies the properties of the frame of reference *FofR* as given in [7]. According to [7], the frame of reference for the distance relation is the tuple $\mathit{FofR} = (D, S, T)$ with $D$ a distance system, $S$ a distance scaling factor, and $T$ the type of relation. The type can be either (1) *intrinsic*, i.e. the distance is determined by an inherent characteristics of the reference object such as its size, shape, or topology; (2) *extrinsic*, i.e. the distance is determined by some external factor, for instance, the arrangement of objects or a measure for the costs involved in travelling (e.g. time); and (3) *deictic*, i.e. the distance is determined by an external point of view, e.g. the viewpoint of an observer perceiving an object.

From that we can derive the positional frame of reference, defining the distance and orientation of the primary object *po*, the reference object *ro*, and a point of view *pov*. If the relation between *po* and *ro* is *intrinsic*, then the scale is depending only on properties of *ro* such as the size or the weight of the reference object. Fig. 7(a) is showing an example. In the left-hand figure *po* is close to *ro* because *ro*'s size is huge, hence the distance between both objects related to *ro* is small. In the right-hand figure on the other hand, *ro* is small, and although the quantitative distance between the centres of both objects is the same, *po* is now far away from *po*. In the *extrinsic* case shown in Fig. 7(b), the relation between objects is only dictated by the reference frame and its size. In the left-hand case of Fig.7(b), *po* is at a medium distance on the back-left side of *ro*. Note that, in the right-hand figure, the scale is different and the $\theta$ between $f_A$ and $f_B$ is about 30 degrees. Hence, *po* is far-left-in-front of *ro*. The third case in Fig. 7(c) shows the *deictic* case including an external point of view *pov*. Therefore, in the left figure, *po* is left of *po*, while in the right figure, *po* is right of *ro*.

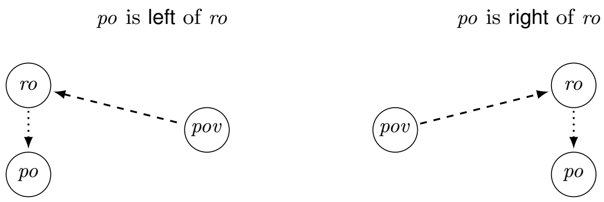
These cases are all captured with our concept of *frame* as given above together with fluents defining the respective coordinate transformation between two different reference systems by translation, rotation, and scaling. For example, in the




(a) An *intrinsic* distance relation $dist(po, ro)$. Here the relation depends on the size of $ro$.




(b) *Extrinsic* relations $dist(po, ro)$ for distance and $\theta(po, ro)$ for orientation.



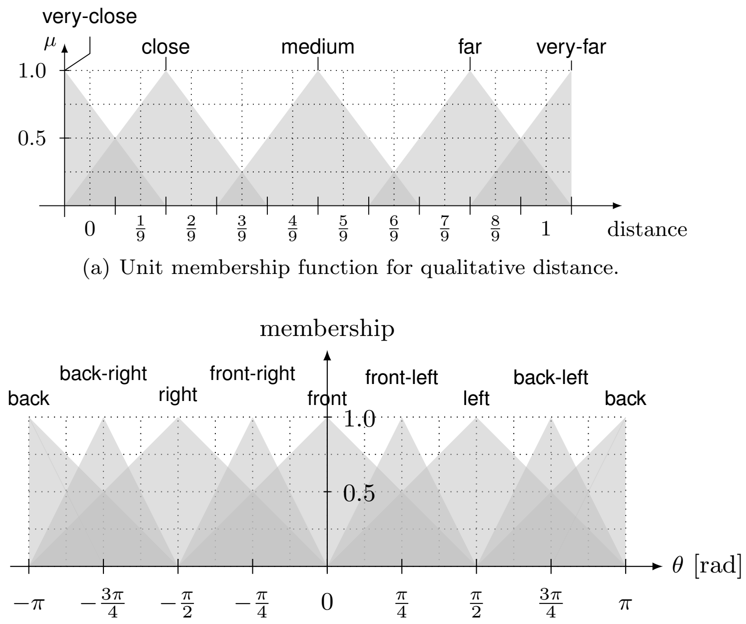

(c) A *deictic* orientation relation $\theta(po, ro)$. Depending on the position of $pov$, $po$ is either left or right of $ro$.

**Fig. 7** Different frames of reference

first case depicted in Fig. 7, we only need to adapt the scaling, while in Fig. 7(b) we need to adopt translation, rotation, and scaling. Finally, for the last case shown in Fig. 7(c), we need to express both, the *po* and the *ro* in the coordinate system of *pov*. Then, we can establish the relation between *po* and *ro* as seen from *pov*.

### 5.3 Axiomatizing the Domestic Robot Domain

In this section, we axiomatize the domestic robot domain based on the introduced qualitative distance and orientation relations together with their respective frames. We start with the required fuzzy sets. We now fix our distance and orientation system for the domestic robot domain. A number of 5 levels of granularity for the distance and 3 levels of granularity for the orientation seems to be sufficient. With that, we can define the fuzzy sets on which our qualitative positional fluents




(a) Unit membership function for qualitative distance.



(b) Membership function for qualitative orientation at level of granularity 3.

**Fig. 8** Membership functions for distance and orientation in our domestic robot domain.

are based. We omit the formal definition of the fuzzy sets $\mathfrak{F}(distance, u, \mu_u)$ and $\mathfrak{F}(orientation, u, \mu_u)$. Instead, we show their continuous membership functions in Fig. 8. The formal definition of both fuzzy sets is similar to the ones for position and distance in Sect. 4.3. The only differences to Fig. 4 are that we use 5 different categories, that the $x$-axis has unit scale, and that we enlarged the categories for front, left, back, and right.

Our domain, as depicted in Fig. 5, consists of the rooms *hallway*, *kitchen*, *parlour*, *bedroom*, and *bathroom*. Further, we have different tables, the kitchenette, the nightstands, or the coatrack, on which objects can be placed or from which objects can be taken. Each of these objects have their own local frame, which is defined w.r.t. the room where they are located in. We assume a room's origin to be at its centre. The origin of the world is also at the centre of Fig. 5, i.e. at position $(15, 15)$. Therefore, the *parlour* has its origin at world coordinate $(4, -7)$. In the following, we give some examples for the *chfrm* predicate, omitting a tedious and not very exciting complete axiomatization of all conversions between all room, object and world frames.

Our world is $30 \times 30$ square units large. Therefore, $\delta^{world}_{\text{very far}} = \sqrt{2} \cdot 30 \approx 42.42$. The parlour has a size of $22 \times 16$ square units and $\delta^{parlour}_{\text{very far}} \approx 27.20$ That means that our distance relation has to be multiplied with $m = 1.56$ to convert very-far distances from the *parlour* into the *world* frame. Therefore assuming that the parlour does not move in the world,

$$frmparam(parlour, world, s) = (4, -7, 0, 1.56).$$

To convert a coordinate from the parlour-frame to the world-frame we simply apply the *chfrm*-predicate. Then, the coordinate $(1,2)_{parlour} = (5,-5)_{world}$ as

$$(5,-5)^T = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 4+1 \\ -7+2 \end{pmatrix}.$$

The function *chfrm* can also be used for deictic relations between objects. Recall, that for a deictic relation, a separate point of view is required (see Fig. 7). For now, we assume that the view direction is always along the positive $y$-axis of the frame coordinate system. With this assumption, categories such as left or right are uniquely determined. Note, however, that this assumption can be easily overcome by adding a standard view point to the definition of the frame, e.g. by a predicate $pov(frame, x, y)$, denoting that the standard point of view of frame *frame* is along the line between the origin of the frame coordinate system and the point $(x, y)$.

To keep track of the position of objects in our world, we need a number of fuzzy positional fluents, one for each object. Each fluent describes the object's position w.r.t. a frame. As we want our robot to fulfil fetch-and-carry tasks in this domain, we need actions like $goto(x, y)$, $grab(object)$, or $drop(object)$, and an action $move(object, x, y)$ to be able to model that objects in the world have been moved or carried around. The positional fluent for the kitchen table could thus be:

$$\begin{aligned}
&pos_{kitchentable}(kitchen, do(a, s)) = (x, y, \theta, m) \equiv \\
&\quad \exists x', y', \theta'.a = move(kitchentable, x', y', \theta') \land x = x' \land y = y' \land \theta = \theta' \land \\
&\qquad \exists x'', y'', \theta''.pos_{kitchentable}(kitchen, s) = (x'', y'', \theta'', m) \lor \\
&\quad \neg\exists x', y', \theta'.a \neq move(kitchentable, x', y', \theta') \land \\
&\qquad pos_{kitchentable}(kitchen, s) = (x, y, \theta, m).
\end{aligned}$$

In the initial situation, the position of the kitchen table is:

$$pos_{kitchentable}(kitchen, S_0) = (-1, -0.5, 1.56, 1).$$

Similarly, we can define and keep track of the position of the robot:

$$\begin{aligned}
&pos_{robot}(world, do(a, s)) = (x, y, \theta, m) \equiv \\
&\quad \exists x', y'.a = goto(x, y) \land x = x' \land y = y' \land \\
&\qquad \exists x'', y''.pos_{robot}(world, s) = (x'', y'', \theta, m) \lor \\
&\quad \neg\exists x'.y'.a = goto(x, y) \land pos_{robot}(world, s) = (x, y, \theta, m)
\end{aligned}$$

with its initial position given by $pos_{robot}(world, S_0) = (-2.5, -2.5, 0, 1)$. Note that in the definitions of the position of the kitchen table and the robot, we carry over the value of the scaling factor $m$ from the previous situation. Of course, we need further fluents to describe our domain. For example, to decide whether or not we can pickup an object: picking it up is only possible if the robot does not already have something in its gripper:

$$\begin{aligned}
&Poss(pickup(obj), s) \equiv \forall x.\neg holding(x, s) \text{ with} \\
&\quad holding(x, do(a, s)) \equiv a = pickup(x) \lor a \neq drop(x) \land holding(x, s).
\end{aligned}$$

To be able to quantify over the positions of objects in our world, we define a macro *object_at_pos* as

$$\begin{aligned}
&object_at_pos(s) = (x, y) \doteq \\
&\quad \exists f.pos_{kitchen}(f, s) = (x', y', \theta', m') \wedge chfrm(x', y', f, world, s) = (x, y) \vee \\
&\quad \exists f.pos_{kitchentable}(f, s) = (x', y', \theta', m') \wedge chfrm(x', y', f, world, s) \vee \cdots
\end{aligned}$$

which is a disjunction over all object positions (transformed into world coordinates) in a particular situation $s$. This macro is true, if any object is located at the position $(x, y)$. As our definition of *chfrm* captures also the case where source and target frame are the same, the above definition is sound and works also in the case where $f = world$. With that we are, for instance, able to query if there is any object at position $(5, 5)$:

$$\mathcal{D} \models \exists x, y.object\_at\_pos(s) = (x, y) \wedge x = 5 \wedge y = 5.$$

Next, we define a function *dist*, which yields the distance between two coordinates and returns the value in the scale of a given frame

$$\begin{aligned}
&dist(x_1, y_1, f_1, x_2, y_2, f_2, f_t, s) = d \equiv \\
&\quad \exists x_1', y_1'.chfrm(x_1, y_1, f_1, world, s) = (x_1', y_1') \wedge \\
&\quad \exists x_2', y_2'.chfrm(x_2, y_2, f_2, world, s) = (x_2', y_2') \wedge \\
&\quad \exists x_o, y_o, \theta_o, m_o.frmparam(world, f_t, s) = (x_o, y_o, \theta_o, m_o) \wedge \\
&\quad \exists d'.d' = \sqrt{(x_1' - x_2')^2 + (y_1' - y_2')^2} \wedge d = d' \cdot m_o,
\end{aligned}$$

where $f_1$ is the frame of the first coordinate, $f_2$ is the frame of the second coordinate, and $f_t$ is the target frame; and a function *ori* that yields the angle between two objects:

$$\begin{aligned}
&ori(x_1, y_1, f_1, x_2, y_2, f_2, f_t, s) = \theta \equiv \\
&\quad \exists x_1', y_1'.chfrm(x_1, y_1, f_1, f_t, s) = (x_1', y_1') \wedge \\
&\quad \exists x_2', y_2'.chfrm(x_2, y_2, f_2, f_t, s) = (x_2', y_2') \wedge \\
&\quad \exists \theta_1. \arctan(y_1'/x_1') = \theta_1 \wedge \exists \theta_2. \arctan(y_2'/x_2') = \theta_2 \wedge \theta = \theta_1 - \theta_2.
\end{aligned}$$

Now, we have everything in place to fetch an object in our domestic world.

### 5.4 Fetching a Cup: A Domestic Robot Example

Next, we want to instruct our robot for a fetch-and-carry task. An example scenario is shown in Fig. 5. Suppose the following situation. The user is sitting on the couch in her parlour, watching TV. The good robot servant is quietly staying aside waiting for instructions. At some point, the instructor is commanding: "*Robot, get me my cup. It is left on the kitchentable, close to the plate.*" Having a closer look at this instruction, the following objects and frames are referred to with it:

$$\text{Robot, get me } \underbrace{\textit{my cup.}}_{po} \text{ It is } \underbrace{\textit{left}}_{ori} \text{ on the } \underbrace{\textit{kitchentable}}_{frame}, \underbrace{\textit{close}}_{dist} \text{ to the } \underbrace{\textit{plate}}_{ro}.$$

The first part of the instruction is a deictic hint. As shown in Fig. 7(a), the position of the primary object *cup* is dependent on a reference object *ro* and a point of view *pov*. As we mentioned earlier, both the *ro* and the *pov* are given by a frame, the kitchentable, in this case. That means that the *ro* is the origin of the kitchen table's coordinate system, the *pov* is along the $y$-axis. The second part is an example for the intrinsic case. Here the distance relation depends on the reference object *plate* and close depends on the size of the reference object *ro*. The correct scaling is defined in the frame parameter of "plate". Assume, that our natural language processing (NLP) software is capable to extract the above marked information.[2] The hints of the human instructor tell us something about the object we seek and can be formalized as follows:

$$\begin{aligned}\mathcal{D} \models\ & \exists x_1, y_1.object\_at\_pos(s) = (x_1, y_1) \wedge \\ & \exists x_2, y_2, \theta_2, m_2.pos_{kitchentable}(kitchen, s) = (x_2, y_2, \theta_2, m_2) \wedge \\ & \mathrm{is}(ori(x_1, y_1, world, x_2, y_2, kitchen, kitchentable, s), \mathsf{left}) \wedge \\ & \exists x_3, y_3, \theta_3, m_3.pos_{plate}(kitchentable, s) = (x_3, y_3, \theta_3, m_3) \wedge \\ & \mathrm{is}(dist(x_1, y_1, world, x_3, y_3, plate, kitchentable, s), \mathsf{close})\end{aligned} \tag{5}$$

That means that we are looking for an object with the coordinates $(x_1, y_1)$ that is on the left side of the table and that is, measured in the scale of the plate, close to the plate. The only object that meets these conditions is $cup_A$. A plan that our robot should therefore make up for this case should be something similar to this:

$$\begin{gathered} do([goto(kitchen), approach(kitchentable), pickup(cup_A), \\ goto(parlour), approach(human), drop(cup_A)], S_0). \end{gathered}$$

The required information is: the kitchentable is in the kitchen, the robot has to conclude that the object left close to the plate is cup A, the position of the human needs to be known, and cup A needs to be dropped without spilling etc. Of course, a sophisticated robot controller is required to execute this simple-looking plan on a real robot in the real world. In the following, we abstract from many of the complications that arise during the execution. However, in the following, we want to concentrate on planning the above action sequence in an abstract yet flexible way, thereby making use of the integrated qualitative spatial representations and reasoning facilities. Having all the ingredients such as the position of all mentioned objects and their frames, the control program shown in Alg. 1 is doing the job.

From the NLP system we expect to get a set $h_{ori} = \{(ro_1, f_1, c_1), \ldots, (ro_k, f_k, c_k)\}$ with statements qualifying the orientation between the primary object *po* and a reference object $ro_i$ together with a reference frame $f_i$ and the qualitative orientation category $c_i$ describing the relation between *po* and $ro_i$. Similarly for distance information, we get a set $h_{dist} = \{(ro_{k+1}, f_{k+1}, c_{k+1}), \ldots, (ro_n, f_n, c_n)\}$. From

[2] In fact, it should not be too hard for the NLP component installed on our @Home robot, to extract these information. See, for instance, [50] for an overview of the control software that is running on our robots.

**Algorithm 1**: A Readylog program making use of the qualitative positional notions for the "*Fetch&Carry*" task. We omit some specification details to retain reasonable clarity.

**proc** fetch_and_carry$(f, x_t, y_t, f_t)$
  $(\pi\ x, y, \theta, m, x_1, y_1, x_2, y_2)[pos_f = (x, y, \theta, m)?$; approach$(x, y)]$;
  **if** $\exists x, y.\Phi(x, y)$ **then**
    $(\pi\ x', y', \theta', m', obj)[\Phi(x', y') \wedge pos_{obj} = (x', y', \theta', m')?$;
      pickup$(obj)$ ;
      approach$(x_t, y_t, f_t)$ ;
      drop$(obj)]$
  **else**
    fail
  **endif**
**endproc**

**proc** approach$(x, y, frame)$
  **if** $\exists x_1, y_1,$ is$(dist(x, y, frame, x_1, y_1, frame, frame), \mathsf{close})$ **then**
    $(\pi\ x_1, y_1)[$is$(dist(x, y, frame, x_1, y_1, frame, frame), \mathsf{close}$; goto$(x_1, y_1)]$;
  **else**
    fail
  **endif**
**endproc**

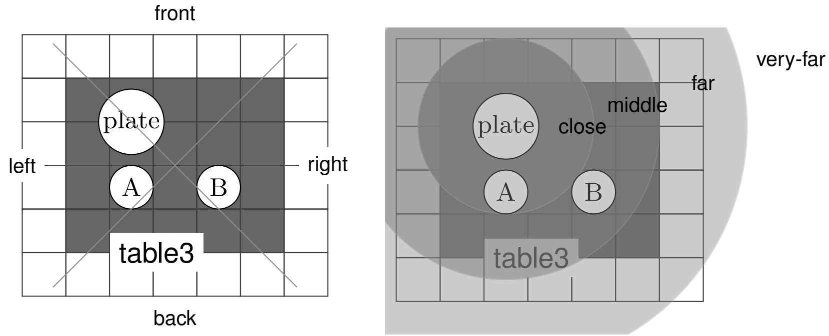
these sets, we construct a macro $\Phi(x, y)$ as:

$$\begin{aligned}
&\Phi(x, y) \doteq object_at_pos(s) = (x, y)\wedge \\
&\quad \exists x_1, y_1, \theta_1, m_1.pos_{ro_1}(f_1, s) = (x_1, y_1, \theta_1, m_1) \wedge \\
&\qquad \text{is}(ori(x, y, world, x_1, y_1, f_1, f_t, s), c_1) \wedge \cdots \wedge \\
&\quad \exists x_k, y_k, \theta_k, m_k.pos_{ro_k}(f_k, s) = (x_k, y_k, \theta_k, m_k) \wedge \\
&\qquad \text{is}(ori(x, y, world, x_k, y_k, f_k, f_t, s), c_k) \wedge \\
&\quad \exists x_{k+1}, y_{k+1}, \theta_{k+1}, m_{k+1}.pos_{ro_{k+1}}(f_{k+1}, s) = (x_{k+1}, y_{k+1}, \theta_{k+1}, m_{k+1}) \wedge \\
&\qquad \text{is}(dist(x, y, world, x_{k+1}, y_{k+1}, f_{k+1}, f_t), c_{k+1}) \wedge \cdots \wedge \\
&\quad \exists x_n, y_n, \theta_n, m_n.pos_{ro_n}(f_n, s) = (x_n, y_n, \theta_n, m_n) \wedge \\
&\qquad \text{is}(dist(x, y, world, x_n, y_n, f_n, f_t, s), c_n)
\end{aligned}$$

For our example these sets would be $h_{ori} = \{kitchentable, kitchen, \mathsf{left}\}$ and $h_{dist} = \{plate, plate, \mathsf{close}\}$ with the target frame $f_t = kitchentable$ (abusing notation slightly). It should be obvious that $\mathcal{D} \models \exists x, y.\Phi(x, y)$ yields Eq. 5.

Now let us showcase how the execution of the program given in Alg. 1 leads to a successful retrieval of the cup as desired. We can assume that the robot has initial knowledge about the positions of rooms and furniture therein. In terms of our specification of *frames* it would comprise the following sentences:

- $frmparam(kitchen, world, s) = (-8.0, 8.5, 0, 2.23)$
- $frmparam(kitchentable, kitchen, s) = (-1.5, -0.5, 0, 3.17)$
- $frmparam(kitchentable, world, s) = (-9.5, 8.0, 0, 7.07)$




(a) left on table

(b) close to plate

**Fig. 9** Details on coord-transform to answer the request in the domestic robot domain

Firstly, the robot retrieves the position of the frame mentioned in the user's request, namely the *kitchentable*. It then moves to that very frame using the method approach. Note that we are using READYLOG's features here to pick a position close to the frame already deploying a first use of the qualitative notions established in this paper.

Then, the robot picks positions[3] that meet the hints given in the initial request ensuring to satisfy $\Phi(x, y)$. As laid out before, $\Phi$ contains the structured information that our NLP could retrieve from the user utterance as given in Eq. 5. Evaluating different objects, $cup_A$ is the only one that satisfies the given specifications as follows.

The predicate is$(ori(x_1, y_1, world, x_2, y_2, kitchen, kitchentable, s), \mathsf{left})$ can only be satisfied for $cup_A$ with $pos_{cup_A}(world) = (-10.5, 7.5)$ and $pos_{kitchentable}(kitchen) = (-9.5, 8.0)$ [4] since $ori(-10.5, 7.5, world, -9.5, 8.0, kitchen, kitchentable, s)$ is then computed as $atan2(-0.5, -1.0) = -2.68$ and $\mathfrak{F}(\mathsf{left}, -2.68, 0.65) \wedge 0.65 > 0$ indicates that this angle does in fact belong to the category left.[5] Analogously, to satisfy the predicate is$(dist(x_1, y_1, world, x_3, y_3, plate, kitchentable, s), \mathsf{close})$ extracted from the distance hint the robot computes the distance for $cup_A$ with $pos_{cup_A}(world) = (-10.5, 7.5)$ and $pos_{plate}(plate) = (0.0, 0.0)$ as $dist(-1, 1, kitchentable, -1, -0.5, kitchentable, plate, s)$ resulting in 1.5 which, after being normalised with the scaling factor for the plate results in $\mathfrak{F}(\mathsf{close}, 0.23, 0.75) \wedge 0.75 > 0$ which in turn indicates that the distance between the cup and the plate does belong to the category close (w.r.t. the plate's size).

We give an illustration of applying the qualitative categories while respecting the corresponding frames both for the orientation and the distance hint in Fig. 9.

Since $cup_A$ is the only object satisfying both the conditions extracted as hints from the user utterance the robot continues executing the program with $cup_A$ as the object to grab and to deliver to the target position. Hence, by following the

[3] Recall that $\pi$ stands for a nondeterministic choice of arguments.

[4] For sake of readability, we leave out the angle and the scaling factor in the positional fluents.

[5] Note that we have formally defined the entries of the membership function only for integer values. However, we assume here that real values are possible also.

program from Alg. 1 the robot eventually comes up with the action sequence

$$do([goto(-9.5, 5.0), pickup(cup_A), goto(-2.0, -8.0), drop(cup_A)], S_0).$$

as claimed initially. It is able to do so satisfying the qualitative spatial description given as hints by the user. This is due to the use of the underlying mechanics that we defined and formally introduced to the situation calculus.

## 6 Discussion and Future Work

In this paper, we showed how qualitative spatial reasoning about positional information in a domestic environment can be conducted. The basic idea is to combine the situation calculus and qualitative representations based on fuzzy sets. We represent the different qualitative categories we want to reason about as fuzzy sets. This is appealing as it is possible that an object falls into several categories at the same time. Fuzzy set theory gives us a formal account for that. For our reasoning engine, we use the situation calculus as a powerful calculus to reason about action and change. We embed fuzzy sets into the situation calculus. To this end, we introduce linguistic categories and fuzzy fluents, which are special functional fluents that can take qualitative, linguistic terms as function values. These linguistic terms are evaluated based on fuzzy sets and related to quantitative representations with a special defuzzifier function, the centre-of-gravity defuzzifier in our case.

For the domestic environment, we extended our notions to account for different contexts of qualitative fluents in indoor environments. Our notion allows for coping, for example, with the fact that different objects have different sizes and that "far" w.r.t. a large room has a different quantitative scale than "far" on the kitchen table. Therefore, we introduced the concept of frames and defined it formally. The *frame* concept needed to be extended to the positional information of domain objects. Otherwise it would be impossible to evaluate spatial relations in the right context. This lead to the notions of fuzzy positional fluents that always carry their contextual frame with them. Finally, we presented a high-level controller in the robot programming and plan language Readylog for the domestic fetch-and-carry task. The controller shows that it is easy to integrate the formal notions presented in this paper for real-world applications. While it was shown that high-level controllers programmed in Readylog can be deployed beneficially also in domestic environments (e.g. [17, 48, 49]), the presented work here lays the theoretical foundations for controllers dealing with qualitative positional information that frequently appear in real world domestic settings. It is due to future work to apply the work presented in this paper on a real robot.

The assumption for the domestic environments we make in this paper is that it is sufficient to have one fixed unit distance and orientation relation for all the different contexts. Of course, we adopt our distance relation according to the size of the room, but still the number of categories remains fixed. It would be interesting, though, for future work to drop this assumption and learn which categories are required in what context by interacting with the human instructor, along the lines of e.g. [46], where fuzzy distances between cities are learnt in a GIS context. Looking at the table scenario, the instruction that is actually meant by the human instructor is more like "cup A is closer to the plate than cup B" or "cup A is closer to the plate than any other object". Fuzzy logic also offers this kind of reasoning facilities. For example, as we pointed out in Sect. 3.3, the compositional rule is able to reason that *X is much larger* ∘ *large* if *X is much larger than Y* and *Y is large*. Together with so-called hedges (a kind of intensifier of a fuzzy category), we could draw the required conclusions. We proposed a situation calculus semantics for hedges in [19]. This needs to be integrated into Readylog for future work to be able to reason about such kind of human instruction, as well. Our example also reveals another interesting problem that we want to address in our future work. When approaching the table, we were assuming that we could pick up the target object from the front side of the table. This is, in general, not true. In some situations the robot might need to approach the table from a different angle. Then, the qualitative positions and orientations of primary and reference objects to each other are changing. Our framework, however, is able to keep track of the positions. Together with Readylog's capabilities to perform decision-theoretic planning making use of a optimization theory, we could plan the optimal approach angle to grab the object in question. It would be also interesting to integrate human-machine interaction into our robot controller. If, for example, the robot is not able to detect the object in question based on the hints that the human instructor is giving, it could start a dialogue with the human to get more evidence about the object. In our controller representation in Alg. 1, the dialogue system in lines 9 and 16 of the robot controller could be used in the failure cases.

**Acknowledgements** The project was supported in parts by the International Bureau of the German Federal Ministry of Education and Research under grant SUA 07/031. A. Ferrein is supported by the Alexander von Humboldt Foundation with a Feodor Lynen fellowship. We thank the anonymous reviewers for their helpful comments on an earlier version of this paper.

## References

1. Arbib, M.: Schema theory. In: The handbook of brain theory and neural networks, pp. 830–834. MIT Press (1998)
2. Bhatt, M., Rahayu, J.W., Sterling, G.: Qualitative spatial reasoning with topological relations in the situation calculus. In: G. Sutcliffe, R. Goebel (eds.) Proceedings of the Nineteenth International Florida Artificial Intelligence Research Society Conference (FLAIRS), pp. 713–718. AAAI Press (2006)
3. Bloch, I.: Spatial reasoning under imprecision using fuzzy set theory, formal logics and mathematical morphology. International Journal of Approximate Reasoning **41**(2), 77–95 (2006). Advances in Fuzzy Sets and Rough Sets
4. Bloch, I., Saffiotti, A.: On the representation of fuzzy spatial relations in robot maps. In: B. Bouchon-Meunier, L. Foulloy, R. Yager (eds.) Intelligent Systems for Information Processing, pp. 47–57. Elsevier, NL (2003)
5. Bolloju, N.: Formulation of qualitative models using fuzzy logic. Decision Support Systems **17**(4), 275–298 (1996)
6. Boutilier, C., Reiter, R., Soutchanski, M., Thrun, S.: Decision-theoretic, high-level agent programming in the situation calculus. In: Proceedings of the 17th National Conference on Artificial Intelligence (AAAI-00), pp. 355–362. AAAI Press/The MIT Press (2000)
7. Clementini, E., Felice, P.D., Hernandez, D.: Qualitative representation of positional information. Artificial Intelligence **95**(2), 317–356 (1997)
8. Cohn, A.G., Hazarika, S.M.: Qualitative spatial representation and reasoning: An overview. Fundamenta Informaticae **46**(1-2), 1–29 (2001)
9. De Giacomo, G., Lésperance, Y., Levesque, H.J.: ConGolog, A concurrent programming language based on situation calculus. Artificial Intelligence **121**(1–2), 109–169 (2000)
10. Dominey, P.F., Boucher, J.D.: Developmental stages of perception and language acquisition in a perceptually grounded robot. Cognitive Systems Research **6**(3), 243–259 (2005)

11. Dubois, D., Prade, H.: An introduction to fuzzy systems. Clinica Chimica Acta **270**(1), 3–29 (1998)
12. Dutta, S.: Qualitative spatial reasoning: A semi-quantitative approach using fuzzy logic. In: A.P. Buchmann, O. Günther, T.R. Smith, Y.F. Wang (eds.) Proceedings of the First Symposium on Design and Implementation of Large Spatial Databases (SSD-89), *Lecture Notes in Computer Science*, vol. 409, pp. 345–364. Springer (1989)
13. Dylla, F., Kreutzmann, A.: Agent Control by Adaptive Neighborhoods. In: M. Bhatt, H. Guesgen, S. Hazarika (eds.) Proceedings of the International Workshop on Spatio-Temporal Dynamics, co-located with the European Conference on Artificial Intelligence (ECAI-10), pp. 55–60. ECAI Workshop Proceedings., and SFB/TR 8 Spatial Cognition Report Series (2010). URL http://www.cosy.informatik.uni-bremen.de/events/ecai10/
14. Dylla, F., Moratz, R.: Exploiting qualitative spatial neighborhoods in the situation calculus. In: C. Freksa, M. Knauff, B. Krieg-Brückner, B. Nebel, T. Barkowsky (eds.) Spatial Cognition IV. Reasoning, Action, and Interaction, *Lecture Notes in Computer Science*, vol. 3343, pp. 304–322. Springer (2005)
15. Ferrein, A.: golog.lua: Towards a non-prolog implementation of GOLOG for embedded systems. In: G. Hoffmann (ed.) Proceedings of the AAAI Spring Symposium on Embedded Reasoning, (SS-10-04), pp. 20–28. AAAI Press (2010)
16. Ferrein, A.: Robot controllers for highly dynamic environments with real-time constraints. Künstliche Intelligenz **24**(2), 175–178 (2010)
17. Ferrein, A., Lakemeyer, G.: Logic-based robot control in highly dynamic domains. Robotics and Autonomous Systems, Special Issue on Semantic Knowledge in Robotics **56**(11), 980–991 (2008)
18. Ferrein, A., Schiffer, S., Lakemeyer, G.: A fuzzy set semantics for qualitative fluents in the situation calculus. In: Proceedings of the International Conference on Intelligent Robotics and Applications (ICIRA'08), *Lecture Notes in Computer Science*, vol. 5314, pp. 498–509. Springer (2008)
19. Ferrein, A., Schiffer, S., Lakemeyer, G.: Embedding fuzzy controllers into golog. In: Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ-IEEE-09), pp. 894–899. IEEE Press (2009)
20. Ferrein, A., Steinbauer, G.: On the way to high-level programming for resource-limited embedded systems with GOLOG. In: N. Ando, S. Balakirsky, T. Hemker, M. Reggiani, O. von Stryk (eds.) Proceedings of the 2nd International Conference on Simulation, Modeling and Programming for Autonomous Robots (SIMPAR 2010), *Lecture Notes in Computer Science*, vol. 6472, pp. 229–240. Springer (2010)
21. Freksa, C.: Using Orientation Information for Qualitative Spatial Reasoning. In: A.U. Frank, I. Campari, U. Formentini (eds.) Theories and methods of spatio-temporal reasoning in geographic space, pp. 162–178. Springer (1992)
22. Freksa, C., Zimmermann, K.: On the utilization of spatial structures for cognitively plausible and efficient reasoning. In: IEEE International Conference on Systems Man and Cybernetics, pp. 261–266. IEEE Press (1992)
23. Grosskreutz, H.: Probabilistic projection and belief update in the pGOLOG framework. In: Proceedings of the 2nd Cognitive Robotics Workshop (CogRob'00) at the 14th European Conference on Artificial Intelligence (ECAI'2000), pp. 34–41 (2000)
24. Grosskreutz, H., Lakemeyer, G.: cc-Golog – An Action Language with Continous Change. Logic Journal of the IGPL **11**(2), 179–221 (2003)
25. Hernandez, D.: Relative representation of spatial knowledge: The 2-d case. In: D.M. Mark, A.U. Frank (eds.) Cognitive and Linguistic Aspects of Geographic Space, pp. 373–385. Kluwer, Dordrecht (1991)
26. Hernandez, D., Clementini, E., Felice, P.D.: Qualitative distances. In: W. Kuhn, A. Frank (eds.) Spatial Information Theory: a theoretical basis for GIS, no. 988 in Lecture Notes in Computer Science, pp. 45–58. Springer (1995)
27. Isermann, R.: On fuzzy logic applications for automatic control, supervision, and fault diagnosis. IEEE Transactions on Systems, Man, and Cybernetics, Part A **28**(2), 221–235 (1998)
28. Jacobs, S., Ferrein, A., Lakemeyer, G.: Controlling unreal tournament 2004 bots with the logic-based action language golog. In: R.M. Young, J.E. Laird (eds.) Proceedings of the First Artificial Intelligence and Interactive Digital Entertainment Conference, pp. 151–152. AAAI Press (2005)
29. Levesque, H., Lakemeyer, G.: Cognitive Robotics. In: F. van Harmelen, V. Lifschitz, B. Porter (eds.) Handbook of Knowledge Representation, chap. 23, pp. 869–886. Elsevier (2008)

30. Levesque, H., Pirri, F., Reiter, R.: Foundations for a calculus of situations. Electronic Transactions of AI (ETAI) **2**(3–4), 159–178 (1998)
31. Levesque, H.J., Reiter, R., Lespérance, Y., Lin, F., Scherl, R.B.: Golog: A logic programming language for dynamic domains. Journal of Logic Programming **31**(1-3), 59–84 (1997)
32. Liu, H., Brown, D.J., Coghill, G.M.: Fuzzy qualitative robot kinematics. IEEE Transactions on Fuzzy Systems **16**(3), 808–822 (2008)
33. McCarthy, J., Hayes, P.: Some philosophical problems from the standpoint of artificial intelligence. Machine Intelligence **4**, 463–502 (1969)
34. McDermott, D., Davis, E.: Planning routes through uncertain territory. Artificial Intelligence **22**(2), 107–156 (1984)
35. Mendel, J.: Fuzzy logic systems for engineering: a tutorial. Proceedings of the IEEE **83**(3), 345–377 (1995)
36. Moratz, R., Dylla, F., Frommberger, L.: A relative orientation algebra with adjustable granularity. In: Proceedings of the Workshop on Agents in Real-Time and Dynamic Environments (IJCAI 05) (2005)
37. Müller, R., Röfer, T., Lankenau, A., Musto, A., Stein, K., Eisenkolb, A.: Coarse qualitative descriptions in robot navigation. In: C. Freksa, W. Brauer, C. Habel, K.F. Wender (eds.) Spatial Cognition II, Integrating Abstract Theories, Empirical Studies, Formal Methods, and Practical Applications, *Lecture Notes in Computer Science*, vol. 1849, pp. 265–276. Springer (2000)
38. Musto, A., Stein, K., Eisenkolb, A., Röfer, T., Brauer, W., Schill, K.: From motion observation to qualitative motion representation. In: C. Freksa, W. Brauer, C. Habel, K.F. Wender (eds.) Spatial Cognition II, Integrating Abstract Theories, Empirical Studies, Formal Methods, and Practical Applications, *Lecture Notes in Computer Science*, vol. 1849, pp. 115–126. Springer (2000)
39. Nordvik, J.P., Smets, P., Magrez, P.: Fuzzy qualitative modeling. In: B. Bouchon, L. Saitta, R.R. Yager (eds.) Uncertainty and Intelligent Systems: Proceedings of the 2nd International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, *Lecture Notes in Computer Science*, vol. 313, pp. 231–238. Springer (1988)
40. Passino, M., Yurkovich, S.: Fuzzy Control. Addison-Wesley-Longman (1998)
41. Piaget, J.: The construction of reality in the child. Journal of Consulting Psychology **19**(1), 77–77 (1955)
42. Pommerening, F., Wölfl, S., Westphal, M.: Right-of-Way Rules as Use Case for Integrating GOLOG and Qualitative Reasoning . In: Proceedings of the 32nd Annual German Conference on AI (KI 2009), pp. 468–475. Springer-Verlag New York Inc (2009)
43. Randell, D.A., Cui, Z., Cohn, A.: A Spatial Logic Based on Regions and Connection. In: Proceedings of the Third International Conference on Principles of Knowledge Representation and Reasoning (KR'92), pp. 165–176. Morgan Kaufmann (1992)
44. Reiter, R.: Knowledge in Action. Logical Foundations for Specifying and Implementing Dynamical Systems. MIT Press (2001)
45. Renz, J., Nebel, B.: Qualitative spatial reasoning using constraint calculi. Handbook of Spatial Logics pp. 161–215 (2007)
46. Robinson, V.B.: Individual and multipersonal fuzzy spatial relations acquired using human-machine interaction. Fuzzy Sets and Systems **113**(1), 133–145 (2000)
47. Saffiotti, A.: Fuzzy logic in autonomous robotics: behavior coordination. In: Proceedings of the IEEE International Conference on Fuzzy Systems, pp. 573–578. IEEE Press (1997)
48. Schiffer, S., Ferrein, A., Lakemeyer, G.: Football is coming home. In: Proceedings of the 2006 International Symposium on Practical Cognitive Agents and Robots (PCAR'06), pp. 39–50. ACM, New York, NY, USA (2006)
49. Schiffer, S., Ferrein, A., Lakemeyer, G.: Fuzzy Representations and Control for Domestic Service Robots in GOLOG. In: L. Iocchi, J.R. del Solar, T. van der Zant (eds.) Domestic Service Robots in the Real World. Workshop Proceedings of the International Conference on Simulation, Modeling and Programming for Autonomous Robots (SIMPAR 2010), pp. 183–192. Darmstadt, Germany (2010)
50. Schiffer, S., Niemüller, T., Doostdar, M., Lakemeyer, G.: Allemaniacs@home 2009 team description. In: Proceedings CD RoboCup 2009. Graz, Austria (2009)
51. Schockaert, S., Cornells, C., De Cock, M., Kerre, E.: Fuzzy spatial relations between vague regions. In: Proc. 3rd International IEEE Conference on Intelligent Systems, pp. 221–226 (2006)

52. Sugeno, M., Yasukawa, T.: A fuzzy-logic-based approach to qualitative modeling. IEEE Transactions on Fuzzy Systems **1**(1), 7–31 (1993)
53. Tikk, D., Biro, G., Gedeon, T., Koczy, L., Yang, J.D.: Improvements and critique on Sugeno's and Yasukawa's qualitative modeling. IEEE Transactions on Fuzzy Systems, **10**(5), 596–606 (2002)
54. Yen, J., Lee, J.: Fuzzy logic as a basis for specifying imprecise requirements. IEEE International Conference on Fuzzy Systems pp. 745–749 (1993)
55. Zadeh, L.: Knowledge representation in fuzzy logic. IEEE Transaction on Knowledge and Data Engineering **1**(1), 89–100 (1989)
56. Zadeh, L.A.: The concept of a linguistic variable and its application to approximate reasoning - I. Information Sciences **8**(3), 199–249 (1975)
57. van der Zant, T., Wisspeintner, T.: Robocup x: A proposal for a new league where robocup goes real world. In: A. Bredenfeld, A. Jacoff, I. Noda, Y. Takahashi (eds.) RoboCup, *LNCS*, vol. 4020, pp. 166–172. Springer (2005)
58. van der Zant, T., Wisspeintner, T.: Robotic Soccer, chap. RoboCup@Home: Creating and Benchmarking Tomorrows Service Robot Applications, pp. 521–528. I-Tech Education and Publishing (2007)