An Ontological Analysis of Vague Motion Verbs, with an Application to Event Recognition

Tommaso D'Odorico

Submitted in accordance with the requirements
for the degree of Doctor of Philosophy.

UNIVERSITY OF LEEDS

The University of Leeds
School of Computing

September 2013

The candidate confirms that the work submitted is his own, except where work which has formed part of jointly authored publications has been included. The contribution of the candidate and other authors to this work has been explicitly indicated below. The candidate confirms that appropriate credit has been given within the thesis where reference has been made to the work of others.

Chapter 6, parts of Chapters 3 and 4 and Section 5.2.1 have appeared in publication as follows:
Tommaso D'Odorico (lead author) and B. Bennett (Sec. 3.2, 4.2 and parts of system implementation). Detecting events in video data using a formal ontology of motion verbs. In *Proceedings of the 8th International Conference on Spatial Cognition, Kloster Seeon, Germany, 2012*, volume 7463 of *Lecture Noties in Artificial Intelligence*, Springer, 2012 [37].

Section 3.1, Chapter 4 and parts of Chapter 5 have appeared in publication as follows:
Tommaso D'Odorico (lead author) and B. Bennett (Sec. 3.4 and parts of Sec. 2). Automated reasoning on vague concepts using formal ontologies, with an application to event detection on video data. In *Proceedings of the 11th International Symposium on Logical Formalizations of Commonsense Reasoning, Aya Napa, Cyprus, 2013* [38].

General remarks from the research have also appeared in:
Thora Tenbrink, Tommaso D'Odorico (part of Sec. 4), C. Hertzberg, S. G. Mazman, C. Meneghetti, N. Reshöft and J. Yang. Tutorial report: Understanding spatial thought through language use. *Journal of Spatial Information Science 5(1)*, 2012 [100].

To my grandmother Santina

You alone of all
You in the sky
I wanna know why clouds come in
Between you and I

Mike Scott – The Waterboys, 2007.

Acknowledgements

I would like to thank my supervisor Brandon Bennett for the guidance and encouragement throughout the course of this PhD. I am grateful to him and the School of Computing at the University of Leeds for providing the funding that allowed me to publish and present the results of this work at two international conferences and also for attending a Mind's Eye program meeting in Denver (USA). I would also like to thank Tony Cohn and Vania Dimitrova, for precious advice at regular intervals throughout my time at the School of Computing.

Thanks to Frank Dylla from the University of Bremen and Thora Tenbrink from the University of Bangor, for some interesting conversations about motion verbs and other vague concepts that inspired and motivated me in several aspects of this research. Thanks to Sandeep, for the help and guidance on the Mind's Eye video dataset.

I also acknowledge the financial support from the University Research Scholarship fund, without which this PhD would not have happened, and the funding supporting the Mind's Eye Program (project VIGIL, W911NF-10-C-0083).

I would like to thank Brandon Bennett, John Stell, Anthony Cohn and Katja Markert for the demonstrating and marking opportunities in connection with their undergraduate and masters modules. I thoroughly enjoyed interacting with students and supporting the teaching activities of the School over the last few years.

I would also like to thank my friends and colleagues in the lab, past and present: Dave, David, Ant, Dimoklis, Sam, Claudio and Elaine for the general mutual support, laughs, advice and inspiring discussions. In particular, Sam and Elaine have been invaluable during the final stages of writing-up: nothing beats an early morning or mid-afternoon conversa-

tion over vagueness, language and methodology with some light-hearted critical questioning clarifying thoughts and exposing interesting insights.

These years spent working on this PhD would not have been the same without some of my best, true friends. A great big thank you to Andy: hiker, geocacher, rambler, traveller, PhDer, dinner host, guest, tea-drinker and companion of many endless conversations about research, friendship, journeys, jobs, life and beyond. Really, I would not have got to this point without you. Thanks to Ilaria for supporting and encouraging me, having been through the stages of a PhD long before myself. Thanks to Lisa, Alex, William, Alberto, Heather, Joe and Ed for having been great housemates, superb friends (or both) and, again, always there for a chat and a cup of tea.

I also thank my friends back home for the thoughts and encouragements that keep me going. I know I can always count on you, and I always dream of bike rides, travels, chats and quality time togther: Euro, Alice, Max, Fabrizio, Laura, Chiara, Agnese. In some ways, we never change, and never will.

The friends and the experience gained through the volunteering with St John Ambulance have provided me with invaluable skills, made me feel rewarded and part of the local community. A thoroughly enjoyable commitment on the side of my research, that has enhanced my outlook on life over the past few years. Thank you for the climbing, the late nights (or early mornings?), the support while leading Leeds Links and the countless hours spent on duty.

Some great times have been had with the fellow postgraduates and the staff at the Exams Office: Di, Ron, Chris, Heather, Donna, Niamh and many others. Again, a pleasant, regular break from research I will miss.

Finally, I'd like to thank my parents Lucio and Giovanna for the support throughout the various life stages that led me up to this point. A lot of what I am is because of you, your perseverance in shaping me up with curiosity, sensitivity, agreeableness and thirst for the unconventional. And my grandma Santina, a constant presence during my upbringing. Hope you are finally smiling up there.

Abstract

This research presents a methodology for the ontological formalisation of vague spatial concepts from natural language, with an application to the automatic recognition of event occurrences on video data. The main issue faced when defining concepts sourced from language is vagueness, related to the presence of ambiguities and borderline cases even in simple concepts such as 'near', 'fast', 'big', etc. Other issues specific to this semantic domain are saliency, granularity and uncertainty.

In this work, the issue of vagueness in formal semantics is discussed and a methodology based on supervaluation semantics is proposed. This constitutes the basis for the formalisation of an ontology of vague spatial concepts based on classical logic, Event Calculus and supervaluation semantics. This ontology is structured in layers where high-level concepts, corresponding to complex actions and events, are inferred through mid-level concepts, corresponding to simple processes and properties of objects, and low-level primitive concepts, representing the most essential spatio-temporal characteristics of the real world.

The development of ProVision, an event recognition system based on a logic-programming implementation of the ontology, demonstrates a practical application of the methodology. ProVision grounds the ontology on data representing the content of simple video scenes, leading to the inference of event occurrences and other high-level concepts.

The contribution of this research is a methodology for the semantic characterisation of vague and qualitative concepts. This methodology addresses the issue of vagueness in ontologies and demonstrates the applicability of a supervaluationist approach to the formalisation of vague concepts. It is also proven to be effective towards solving a practical reasoning task, such as the event recognition on which this work focuses.

Contents

List of Figures

Chapter 1

Introduction

Ontologies have become popular in the field of information and cognitive science as a general means for specifying the semantics of the terminology used to represent data. The development of an ontology for the formalisation of concepts describing the physical world is often problematic due to the semantic complexity of most terms sourced from natural language. What does it mean for a man to be 'tall'? Or for a house to be 'near' the railway station? And how can one formally characterise the occurrence of an event where a person is 'picking up' something? Many words have multiple meanings, many others are vague and, often, their applicability in describing a particular object, situation or event is subject to an interpretation by the observer speaking or thinking the word. People indeed hold different beliefs and opinions on the meanings of concepts shaped by their cultural background and prior experience. People also choose and adapt the interpretation of a particular concept according to the specific context surrounding the utterance of the term.

The aim of this work is to develop a methodology for the characterisation of vague concepts in formal ontologies in order to develop an ontology-based automatic reasoning system for the recognition of event occurrences. In particular, this methodology focuses on a specific set of vague spatial concepts and motion verbs and culminates with the implementation of the Prolog-based event recognition system ProVision. This system implements the ontological formalism for the characterisation of vague motion verbs and logically infers the occurrence of certain events in real-world situations, represented by data obtained through the processing

of video scenes. Although the ontology and event recognition system have been developed within the context of motion verbs, the resulting methodology should be general enough for its extension and application to other reasoning tasks.

Motivation for this work stems from interest in formal ontologies, vagueness in natural language and the general aim of combining depth and breadth in an ontology for describing the physical world. This has generated a quest for a logic-based formalism for the representation and reasoning on vague concepts that could also find a practical application in Artificial Intelligence, the results of which are presented here. Motivation also stems from the aim of bridging the gap between the *deep and narrow* approach to ontology formalisation, involving strong axioms and a small number of primitives, and the *broad and shallow* approach, leading to large vocabularies and weak axioms. The approach followed by this research to ontology formalisation is based on the identification of mid-level concepts, defined in terms of basic primitives and also well-suited for facilitating easy definition of a wide range of higher level terminology.

1.1 Domain, Application and Methodology

The specific purpose of developing an ontology for the formalisation of vague motion verbs and their recognition from video through an automatic inference system has stemmed from involvement in the Mind's Eye challenge by DARPA [33, 34]. This project aims to automatically recognise occurrences of actions in video sequences described by the 48 verbs listed in Table 1.1, hereafter referred to as *motion verbs*:

Approach	Arrive	Attach	Bounce	Bury
Carry	Catch	Chase	Close	Collide
Dig	Drop	Enter	Exchange	Exit
Fall	Flee	Fly	Follow	Get
Give	Go	Hand	Haul	Have
Hit	Hold	Jump	Kick	Leave
Lift	Move	Open	Pass	PickUp
Push	PutDown	Raise	Receive	Replace
Run	Snatch	Stop	Take	Throw
Touch	Turn	Walk		

Table 1.1: Motion verbs list

DARPA also provided an extensive collection of videos, each of which contain occurrences of one or more actions described by the verbs in the table above and involving different subjects.

The Mind's Eye challenge is composed of sub-tasks, namely *recognition*, *gap filling* and *anomaly detection*. The recognition task, aimed at recognising any occurrence of a motion verb in the videos, has been the inspiration for this work. Gap filling and anomaly detection tasks, outside the scope of the work presented here, respectively aim at inferring event occurrences over the gaps of an incomplete video sequence, and at detecting unconventional or singular occurrences of events.

Most attempts at performing event recognition tasks of this kind are to be found in Artificial Intelligence within the Vision and Machine Learning research areas. This task has been attempted, for example, through a combination of Machine Learning and Inductive Logic Programming techniques where event models are learnt through the analysis of qualitative spatio-temporal relations between objects [42, 41, 99]. An overview on these approaches can be found in Sec. 2.5.

The approach to event recognition presented in this work is centred around the development of an ontology of vague spatial concepts based on supervaluation semantics [61, 97], and the modelling of each motion verb through the analysis of its most relevant semantic characteristics that can be defined in terms of observable properties. The issue of vagueness arises from the fact that understanding and defining motion verbs in terms of observable properties involves the formalisation of qualitative vague concepts.

For example, verbs such as Chase or Run refer to the concept of an object's motion being *fast*, Pass and Arrive refer to the notion of an object being *near* a boundary expressing the *vicinity* to another object, Fly and Fall may refer to the notion of a trajectory being *mostly* horizontal or vertical, and so on.

Further ambiguities arise from terms that have multiple meanings, such as Pass that may refer to a person giving an item to another person, or crossing a boundary between two areas. Some motion instances may smoothly transition from constituting an occurrence of one verb, for example Walk, into an occurrence of another verb, for example Run. Additionally, the same observed situation may be described by concepts similar

in meaning but different in *saliency*. This is mainly due to a greater specificity of one concept compared to another (e.g. Kick compared to the more general Hit) or a greater emphasis given by a certain concept to a peculiar feature of the action (e.g. the different intentionality in occurrences of Collide or Hit).

The ontology-based approach to event recognition should allow for greater specification and understanding of each verb semantic characteristics, which may not be completely grasped by learning techniques exposed to a finite set of examples. Ideally, a comprehensive formal ontology of motion verbs would require an extensive and systematic analysis of the semantic properties of each verb, drawing for example from studies in linguistics [107, 69]. However, such an extensive analysis is likely to formalise concepts in terms of semantic characteristics that are not definable in terms of observable properties. Some semantic characteristics, in fact, may refer to contextual information or fine-grained and highly detailed properties of objects not detectable in practical applications. Some can be so abstract to be almost impossible to ground on observable facts.

In other words, humans may achieve a deep, thorough understanding of a particular meaning and applicability of a word, but a computer is unlikely to achieve the same understanding, given the limitations of the current state of the art in automated image and video interpretation. In the context of event recognition, this means that machines simply cannot reason on the same set of semantic properties referred to by humans while attempting to describe a situation or event occurrence with qualitative concepts from natural language. This limitation is essentially caused by the fact that machines can only operate on a limited, finite and coarse-grained representation of the world and possess very limited, if any, inbuilt knowledge and prior experience able to guide their judgements.

For this reason, the goal of this research is not to achieve a semantically *exhaustive* characterisation of motion verbs, but rather to develop a methodology for an *effective* characterisation of concepts that would advance an ontological framework from an analytical abstraction of the physical world to a practical, usable reasoning device. In fact, the methodology developed here has led to the implementation of the ontology of motion verbs into the Prolog-based event recognition system ProVision, which grounds objects' primitive properties from the data, and infers mid-level

and higher-level concepts expressing occurrences of events in the video scenes.

Our research also contributes to analysis and considerations on issues of vagueness, saliency, uncertainty and relevancy of context in formal ontologies. Although video processing falls outside the scope of this research, as it lies in within the Vision for Artificial Intelligence area, it constitutes a closely related field. In fact, advances in algorithms and techniques for object detection and classification from video would allow to augment the level of detail with which verbs and semantic properties are formalised in the ontology, allowing ProVision to perform more accurate and/or specific recognition tasks.

Although the specific application domain of motion verbs and the video dataset have an influence on the ontology design and the verb models, efforts have been made to maintain enough generality in the principles and methodology for them to be deployed in reasoning tasks involving other domains.

1.2 Thesis Outline

A formal ontology of motion verbs involves the definition of concepts that may hold at specific time points, occur over certain temporal intervals or refer to particular spatial characteristics of objects and the environment. Most of the related work on formalisms for spatio-temporal reasoning and for the representation of actions and events is reviewed in Chapter 2. This chapter also overviews issues of linguistic studies and classification of verbs. Studies on the nature and characteristics of vagueness are introduced, with an overview of the most prominent logical formalisms that have been proposed for the definition of vague concepts in formal languages.

The specific issues involving the formalisation of an ontology of vague motion verbs are analysed and discussed in Chapter 3. This discussion examines the issue of vagueness and its impact on the formalisation of spatial concepts, followed by an overview of the most relevant semantic characteristics of motion verbs to be formalised as concepts in the ontology. The chapter also analyses issues particularly relevant to the task of event recognition, such as saliency, context, granularity and uncertainty.

The formal ontology of motion verbs is introduced in Chapter 4. The spatio-temporal framework is formalised, followed by the specification of objects in their types and sub-types. The formalism for expressing that propositional expressions hold at a particular time or over a particular interval is established, followed by the definition of primitive and mid-level properties of objects that express their most spatial characteristics. These properties are the bricks building higher-level concepts and the verb models. The chapter closes with considerations on how the appearance of objects expressed by primitive properties can be augmented with a Theory of Appearances.

The analysis of the most relevant characteristics of motion verbs is presented in Chapter 5, where a subset of motion verbs among the ones listed in Table 1.1 is extensively analysed and formalised within the ontological framework defined in Chapter 4. The verbs being modelled broadly involve concepts describing simple motion, such as Move or Walk, proximity, such as Approach or Arrive, relations between objects, such as Follow, or contact, such as Touch or Hit.

Finally, Chapter 6 describes the development of ProVision, the event recognition system based on the ontology resulting from the previous two chapters. The chapter starts with an analysis of the video scenes and the nature and quality of the data available to ProVision to ground the ontology, continues with details about the implementation of parts of the ontology definition and closes with some experimental results and considerations on the evaluation of the system for the recognition of events Approach and Hold.

1.3 Issues Addressed and Contribution

Although there has been work on ontology based definitions of event types, this has hardly been applied to event recognition from video [96], in which, instead, techniques based on Machine Learning dominate. These techniques aim to understand and recognise observable features through the analysis of a large set of training examples. This approach may involve concept semantics, but it is mostly based on a statistical analysis aimed at identifying correlations across training examples. The advantage of such data-driven approach is that it provides an established method-

ology to build an event recognition system relatively quickly, whilst an ontology-based approach requires an extended formalisation and analysis stage before its implementation can be built.

This is a possible cause of the absence of substantial research into designing an ontological framework and corresponding implementation for the task of event recognition. It is believed that ontology-based systems can demonstrate advantages in terms of their generality and greater specification of the meaning of particular features to be recognised, whilst Machine Learning approaches tend to shape their deductions according to the specific set of examples they have been trained on. Ontology-based systems also have the advantage of providing an explanation justifying the reasoning process leading to a particular inference.

The main contribution of this research is a methodology for developing such a framework, and in general for the characterisation of vague concepts in formal ontologies, with the aim of solving practical reasoning tasks such as the Mind's Eye Challenge. Specific contributions are listed below:

- Analysis and discussion of issues regarding vagueness in natural language and the framing of the epistemic stance as an effective model of vagueness for ontology reasoning (Sec. 3.1);
- Development of a method for the formalisation of vague concepts based on supervaluation semantics, leading to the precisification of vague concepts with borderline cases through precisification thresholds;
- Illustration of a methodology for the semantic characterisation of motion verbs (and, more generally, vague concepts) that focuses on observable properties of objects;
- Demonstration of the applicability of the above strategies in ProVision, a Prolog-based implementation of the ontology formalism which performs event recognition from video by inferring high-level qualitative concepts through the grounding of ontology primitives on real data.

Chapter 2

Related work

This chapter summarises the most relevant work related to formal ontologies and the characterisation of concepts from natural language. Several spatio-temporal formalisms, reviewed in Sec. 2.1, constitute a basis for logical languages for the representation of action and events, introduced in Sec. 2.2. Linguistic studies, outlined in Sec. 2.3, contribute to the analysis of the characteristics of words representing states, actions and events in natural language. Vagueness as a linguistic phenomenon has been a matter of study in the philosophical and logical communities and different characterisations and formalisms have been proposed as a result, reviewed in Sec. 2.4. This discussion also continues in Sec. 3.1 with a focus on the specific domain of motion verbs. Finally, Sec. 2.5 outlines how Machine Learning techniques for logic programming may contribute to building a logic-based event recognition system.

2.1 Spatio-temporal reasoning

One of the aims of this project is to develop an ontology for the representation of objects and their interactions in space over a particular time interval. Therefore formalisms of interest to the issues addressed in this work are frameworks with the capability of reasoning about *time* and *space*.

2.1.1 Temporal Reasoning formalisms

Formalisms for temporal reasoning generally define a set of ordered time points or instants $\mathcal{T} = \{t_1, t_2, t_3 \ldots\}$ and a set of intervals ranging over

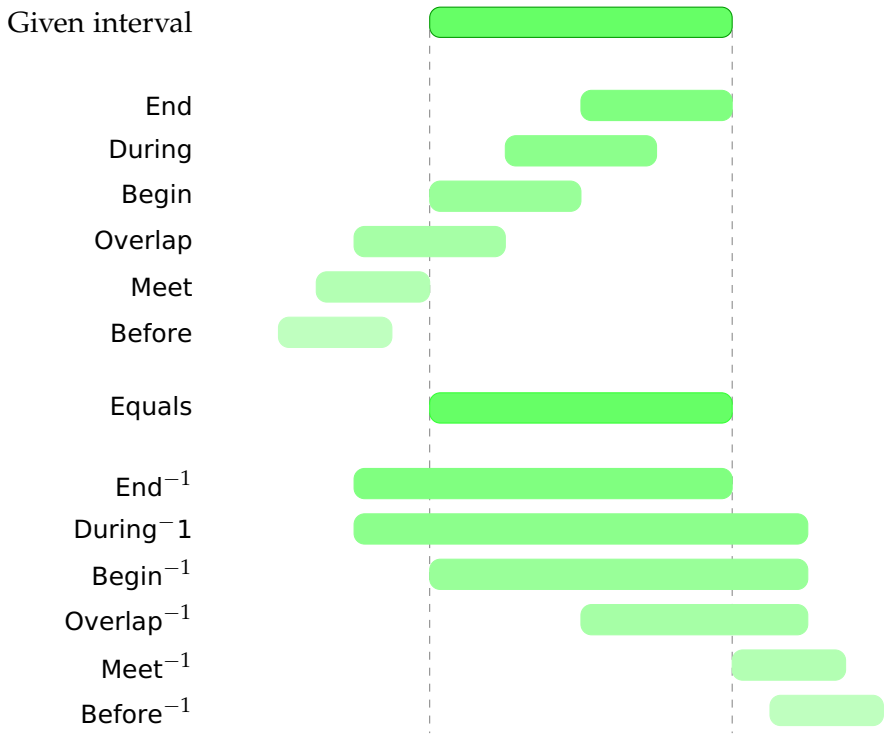


Figure 2.1: Allen temporal intervals

instants in $\mathcal{T}$, for example $[t_s, t_e]$ is a *closed* interval containing the time points between t_s and t_e. Specific formalisms further specify the structure of $\mathcal{T}$, which could be based on a discrete or dense model of time, and the nature of intervals, for example some formalisms admit both closed and open intervals.

Early work on temporal reasoning includes Allen's classical results on temporal intervals [8, 9]. Allen's interval calculus is a reasoning formalism on the relation between intervals. A set of 13 jointly exhaustive and pairwise disjoint relations is defined, such that one and only one particular relation holds between two given intervals $[t_s^1, t_e^1]$ and $[t_s^2, t_e^2]$, for example $\mathit{meet}([t_s^1, t_e^1], [t_s^2, t_e^2])$ or $\mathit{overlap}([t_s^1, t_e^1], [t_s^2, t_e^2])$. This is illustrated in figure 2.1.

Some frameworks introduce *modal operators* $\Box$ and $\Diamond$ of modal logics [22] for reasoning about time points and intervals in possible worlds that can be represented. Early work on modal logics for temporal reasoning is based on the relations defined in Allen's interval calculus [54, 109].

Other works focus on temporal structures and the possibility of reasoning about logical propositions holding at some time instant in the past or at some time instant in the future, either with linear models of time, such as Linear Time Logics LTL, or branching models of time, such as Computer Time Logics CTL and CTL* [43, 44]. Further work focuses on issues of complexity and decidability of some of these modal logics of time, as their expressive power often results in the logic being undecidable. Most of these approaches attempt to define decidable logics of time by restricting their expressive power, in particular the range of admissible relations allowed in the logic [72, 24, 23].

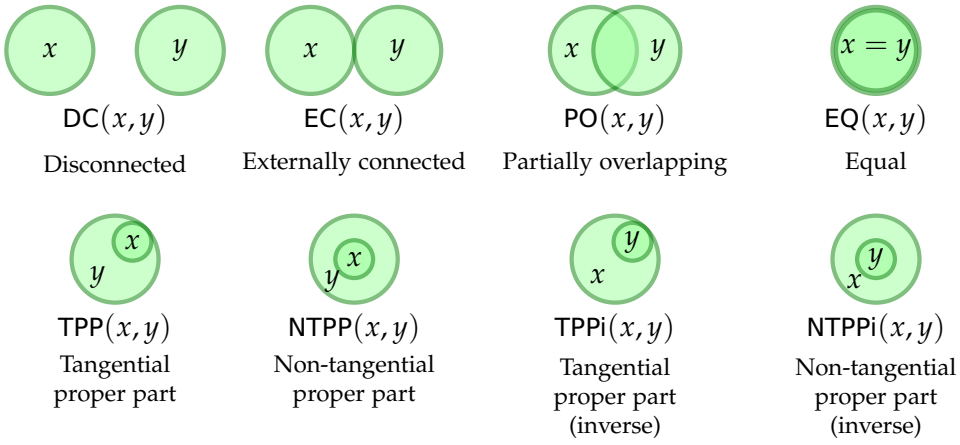
2.1.2 Spatial Reasoning frameworks

Formalisms for spatial reasoning generally define a set of points in space $\mathcal{S}_p$ and/or a set of spatial regions $\mathcal{S}_a$ and focus on defining relations holding between points and regions. Contrarily to the models of time appearing in logical systems for temporal reasoning, in general there is a wider variety of models of points and areas in systems for spatial reasoning. Qualitative spatial reasoning is an area of research whose focus is on identifying qualitative models of space and relations holding between points and regions.

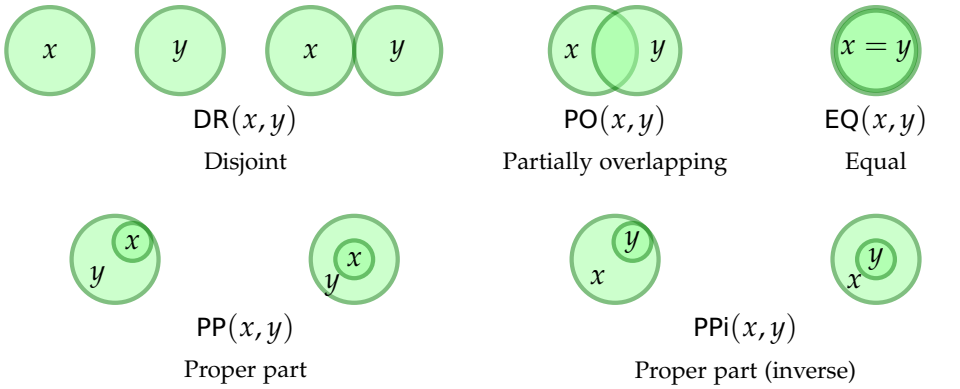
A classic example of a qualitative spatial reasoning formalism is Region Connection Calculus (RCC, [87]), in some ways a counterpart to Allen's temporal calculus applied to spatial regions. RCC examines the topological relations that hold between closed regions of space and identifies two sets of disjoint and exhaustive relations holding between them: a fine-grained set of 8 relations (RCC8) and a coarser set of 5 relations (RCC5), illustrated in figure Fig. 2.2.

RCC has been extended in order to specify relations holding between 2-dimensional concave regions, in particular with calculi RCC23 [30], which treats these regions as one whole part and focuses on the relations between concave regions and their convex hull, and the more expressive RCC62 [114], where concave regions are decomposed into 4 constituent parts (outside, boundary, interior and inside).

Studies on the extensibility of RCC to 3-dimensional regions also exist. The calculus RCC3D [7] defines 13 relations based on the intersection of the 3D interior, exterior and boundary of regions and the intersection of their



(a) RCC8 relations



(b) RCC5 relations

Figure 2.2: Relations in Region Connection Calculus

2-dimensional projection on a reference plane. The calculus VRCC3D+ [92] extends the previous by introducing further relations addressing the ambiguous cases in RCC3D which may arise due to region obscuration in a 3-dimensional space.

A number of studies on Region Connection Calculus has followed, such as completeness and tractability of sets of relations [90, 73] and applications of modal logics to the calculus [12]. This calculus has also been integrated with temporal reasoning frameworks in order to define a modal logic system describing changes in topological relations over time [17]. An overview of modal logics for spatial reasoning is provided in [5, 6].

These frameworks are based on an abstract model of time and space. Our application aims to develop an ontology for spatio-temporal reasoning that can be used to extract and process information from real data. For this reason, calculi based on such abstract temporal and spatial models often need an adaptation in order to deal with issues such as uncertainty and vagueness (see Chapter 3). Spatial and temporal logical formalisms have inspired the spatial and temporal aspects of the ontology defined in Chapter 4, and some particular relations of these calculi have been incorporated, namely Region Connection Calculus.

2.2 Reasoning about Actions and Events

Two important formalisms for reasoning about actions and events are *Situation Calculus* [75, 89] and *Event Calculus* [64, 76].

Situation Calculus is a formalism without an explicit model of time instants, but where the changes in a dynamic world are modelled by a sequence of states. The domain of Situation Calculus comprises actions, situations and objects. The world being modelled is thought of as progressing through several situations, each of which result from some performed action. There are two different interpretations of situations: in [75] a situation represents a state of the world, whilst in [89] a situation represents a sequence of occurrences in the form of a history. Respectively, the initial state of the world or initial sequence prior to any actions is represented by S_0. Actions represent an event which induces a change in the state of the world, altering the current situation. Fluents are temporal entities represented by propositional expressions that may be true or false given a particular situation, and in some ways they model when a particular property of the world, or of specific objects, holds. Within the calculus, an action may have conditions that limit the situations in which it can be executed, and effects of its execution on the fluents.

Event calculus is a formalism where time points are explicit, and makes it possible to state whether a particular proposition is true at a particular time point. Propositions of this kind, whose truth-value can be linked to a specific time point, are called *fluents*. The basic construct of the language is the predicate $\mathsf{HoldsAt}(p, t)$ which is true if fluent p holds at time point t. The calculus also allows for the representation of actions whose execution

may have an effect on the truth-value of certain fluents given a particular time point.

Versatile Event Logic (VEL) is a highly expressive language with a semantics for the representation of spatio-temporal relations, elements of Situation Calculus and Event Calculus for the representation of action and events, and a modal operator to describe alternative histories [18]. VEL defines a spatio-temporal ontology of individuals, times, temporal intervals, states, observable values and events.

The ontology formalised in Chapter 4 and the logical analysis of motion verbs in Chapter 5 have been strongly influenced by characteristics of Event Calculus and Versatile Event Logic. This is particularly true in respect to the definition of an explicit temporal model constituted by time points, the HoldsAt expression expressing truth of a particular fluent at a particular time point and the Occurs construct expressing that a certain event occurs over a time interval.

Ontologies of Processes and Events

Building a logical formalism for reasoning about time and events often involves distinguishing between different classes of temporal entitites that occur over at a particular time or at a particular interval. Operating such a classification has been and still is a matter of debate among the KRR community, Galton [49] provides a summary suggesting a classification of temporal entities in *processes* and *events*:

- Processes represent complex activities, happening over a time interval and generally, but not necessarily, directed towards reaching a goal or intended state. Processes are structured, and may be composed of sub-processes. An example is the process 'cycling from home to work'.

- Events represent simpler activities, generally happening at a time point and generally not directed towards a goal or reaching an intended state. An event can be considered as a 'bounded instantiation of a process' or as a *chunk* of a process. An example is given by a chunk of the process 'cycling from home to work' which represents a person cycling without the context and additional significance carried by the fact that the person is cycling from home to work.

Galton also describes the ways in which events can be described in terms of other events, and how the repetition of an event can be combined in order to form a process. There is a further ontological classification of temporal entities into *types* and *tokens*

- A process- or event-*type* is a temporal entity of which there may exist many temporal instances, for example 'I catch the train to Edinburgh'.
- A process- or event-*token* is a particular temporal occurrence of a process- or event-type, for example 'I catch the train to Edinburgh at 9.50am today'.

The ontology described in Chapter 4 operates the above distinction between event-types and event-tokens.

A process occurring over a particular interval i may or may not occur over sub-intervals of i. Processes for which these sub-intervals can be unboundedly narrow are said to be *homogeneous* (e.g. falling, approaching, drying) whilst processes that do not occur over unboundedly narrow sub-intervals are said to be *granular* (e.g. walking, jumping, picking up). This is because if an homogeneous process such as "falling" occurs over temporal interval i, it is generally possible to describe the situation over interval $i' \subset i$ as an occurrence of the same process. On the other hand, a granular event such as "walking" occurring over interval i is not homogeneous as it is generally the result of a sequence of sub-processes occurring over sub-intervals of i. In other words, if "walking" occurs over i, the process occurring over a very small interval $i' \subset i$ is described, for example, by the constituent sub-process of "a person raising his/her foot".

Processes that have the potential to continue indefinitely (e.g. cycling) are called *open-ended*, and they are often homogeneous. Instances of such processes may still specify a termination (e.g. cycling to the station). Processes which are neither homogeneous nor open-ended are called *closed* (e.g. baking a cake) [49].

2.3 Linguistics

Investigations about the meaning and semantic characterisations of words occurring in natural language are often found in the field of linguistics.

An established classification of verbs and the actions or states they represent is given by Vendler [107, 108], where verbs are classified according to the following categories:

- *Atelic*
  These expressions describe actions or situations not associated with a goal or target. They are further characterised as:
  - *States*
    Static actions or expressions not involving or determining any change in the world, for example 'be tall', 'own a bicycle', 'be alive'...
  - *Activities*
    Dynamic actions involving or determining change, for example 'walk', 'move', 'approach'...
- *Telic*
  These expressions describe actions or situations associated with a goal or target, which is a desired or intended result that an occurrence of this kind is leading to. They are further characterised as:
  - *Achievements*
    Punctual actions whose occurrences happen at a single time instant, for example 'touch', 'win (a game)', 'reach (the bus stop)'...
  - *Accomplishments*
    Durative actions whose occurrences extend over a time interval, for example 'approach Leeds', 'carry a box', 'lift a weight'...

The verb models in Chapter 5 relate to the classification above, especially regarding the distinction between verbs describing static events, whose temporal duration is punctual or near-punctual, and verbs describing dynamic events or complex processes constituted by a particular sequence of events. The process of verb modelling has shown that the classification above is not entirely rigid, as certain expressions may be interpreted in more than one sense.

For example, according to the classification above, the verb 'to touch' represents a punctual action occurring at a time instant. A rigorous anal-

ysis would lead to the formal characterisation of Touch as an event happening at a single time point t. However, some people may interpret the semantics of the verb 'to touch' in a more relaxed way, assigning a duration to the event Touch which encompasses some temporal interval surrounding t and may include, for example, some action prior to Touch which resulted in the event occurrence.

A further study in linguistics about the semantics of verbs is Levin's work on verb classes and alternations [69]. This work analyses the different ways verbs can occur in language sentences, and contributed to the formal characterisation of verb models by clarifying the circumstances under which an expression may occur, or the kinds of objects and contextual information relevant to the interpretation of a particular verb.

Another resource about the semantics of tense and aspect and the treatment of events is [65], which provides an overview on the most philosophical and linguistic aspects of the matter and studies on formal and computational approaches that deal with time and events.

2.4 Vagueness

Vagueness is a phenomenon which manifests itself when attempting a formal definition of natural language concepts, and it is different from uncertainty, arising from insufficient or imprecise knowledge, or generality, arising from lack of specificity. It can be seen as a form of ambiguity, although ambiguity tends to refer to the possibility for a word to have different interpretations, whilst vagueness is concerned with the ways the interpretation of a word can be made precise [103, pag. 110–115].

A classic example of vagueness is given by the Sorites paradox about establishing what constitutes a heap of sand [58, 111]. One may argue that, given a heap, the heap obtained by removing a grain from the original heap would still be considered a heap. However, most would agree on the fact that a single grain of sand does not constitute a heap. If one decides to progressively remove grains from a heap of sand until there is just one grain left, the paradox is shown by the fact such an amount of sand would be considered a heap according to the first observation, and not a heap according to the second.

This example demonstrates a common trait of vague concepts central

to the analysis of most lingusitic forms of vagueness, which is the lack of a crisp *applicability boundary* separating positive and negative instances of a certain concept, i.e. what constitutes a heap and what does not. The example also demonstrates the fact that establishing a precise interpretation of a vague concept is problematic. In fact, one could fix a minimum number n of grains of sand constituting a heap, however this would result in the counterintuitive consequence that the concept 'heap' would apply to a pile of n grains but not to a pile of $n-1$ grains.

A linguistic expression or proposition is vague if it contains one or more constituent parts which are vague. Different syntactic categories of vague terms appearing in language sentences can be identified [15]:

- *Adjectives*, for example *rich*, *heavy*, *red*, *tall*, *elongated*, *steep*...
  Vagueness in adjectives correponds to a blurred boundary for the applicability of the term. Generally, this is due to an underspecification of criteria drawing a clear separation between positive and negative instances of a certain thing characterised with the adjective. For example, it is not very clear how to precisely separate a group of people with very similar heights between *tall* and *not tall* ones.

- *Count nouns*, for example *lake*, *river*, *mountain*,...
  As for adjectives, vagueness in count nouns is also due to a blurred applicability boundary . However, count nouns tend to be more complex as there are many observable properties defining the criteria for the applicability of a term [56]. For example, a count noun such as *lake* could be defined in terms of its size, shape, extension, water volume, tributary, emissary, surrounding environment and many more.

- *Relational Expressions*: for example *near*, *far*, *beside*,...
  These expressions also show blurred applicability boundaries due to unspecified applicability criteria. For example, the distance separating a school and a station in the sentence 'the school is *near* the station'.

The case of adjectives is interesting, with vagueness being particularly pervasive in this category. In general, adjectives are vague if they show borderline cases. A proposed suggestion for the individuation of such adjectives is based on two principles: whether the word can be modified

by the adjective 'very' and whether it allows comparisons (e.g. 'Hardknott Pass is *steeper* than Wrynose Pass'). If an adjectives matches these two criteria, it is called a *degree adjective* [84]. Statistics on the most frequently used adjectives in the British National Corpus [3] show that the most frequent include many degree adjectives, such as 'new', 'good', 'old', 'great', 'high', 'small', 'large', etc. [103, Ch. 6].

Most of the foundational characterisations on origins and nature of vagueness can be found among the philosophical and logic communities [47, 111, 113, 62]. More recently, interest has been growing within the computing and geography communities too [106, 19, 50, 29]. There is an ongoing debate on the nature and characterisation of vagueness, with three accounts emerging from the literature:

- *De dicto* vagueness.
  This is the characterisation most agreed on within the philosophical community, and explains vagueness in terms of linguistic indeterminacy of representation [70, 106]. Given a vague term denoting an object (e.g. 'mountain', for which a demarcation is ambiguous), this view maintains that ambiguity is generated by an indeterminacy in the language expression, and any other type of ambiguity can ultimately be explained in terms of the former. In other words, de dicto vagueness states that there is no such thing as a vague object, but only linguistic expressions that vaguely identify objects. Because of this, vagueness can be addressed by addressing the linguistic ambiguity, thus making the linguistic expressions more precise. For example, one could introduce a new precise noun 'mountain' which refers to a precisely and uniquely determined spatial volume surrounding a terrain prominence.

- *De re* or *ontic* vagueness.
  This characterisation maintains that there exist intrinsically vague objects in the world [102]. In the example noun 'Mount Everest', it is argued that its boundaries are necessarily blurred as there is no fact of the matter about some of its constituent molecules being inside or outside these boundaries. De re vagueness tends to reject the suggestion by *de dicto* theories that vague concepts are, essentially, concepts that can be made more precise, and argues that doing so

would be more akin to characterising a new, sharper concept rather than precisifying a vague one.

- *Epistemic* vagueness.
  This characterisation argues that there is an objectively correct set of criteria for precisely determining the applicability of a vague concept; however, this set of criteria is unknown, due to the uncertain and inconsistent meaning of words and terms in natural language [98, 110, 111].

There is no general agreement on which of the above characterisations identifies exactly the nature and characteristics of vagueness, and it is possible for different views to be applicable at the same time in a particular context. The main focus of this work is concerned with indeterminacy of linguistic expressions, hence adopting the de dicto vagueness as the main characterisation. This does not exclude the possibility that some objects may be considered as inherently vague, therefore allowing for elements of ontic vagueness.

Whilst this work rejects purely epistemic views of vagueness, an epistemic approach in disambiguating vagueness in linguistic expressions can be pragmatically convenient in developing an automated reasoning system. In this respect, an interesting conceptualisation is the *epistemic stance* [68], where an artificial agent is modelled to behave as if the epistemic view was correct. This is further discussed in Sec. 3.1.

The ontology of vague motion verbs of Chapters 4 and 5 involves the definition and formalisation of vague concepts. As "the apparatus of classical logic, within which ontologies have traditionally been defined, cannot by itself account for the meanings of the conceptual terms of natural language", there is the need for a superstructure aimed at mediating between the natural language concepts and their vagueness and a precise classification of these in a formal ontology [13]. The most prominent formalisms proposed in the literature are reviewed below:

- *Egg-Yolk model* [31]
  This is a model for possible interpretation for an axiomatic theory of vague regions, which are viewed as vague objects (de re vagueness). This model is based on the axiomatisation of a theory for crisp and blurred regions [15, Sec. 4.1].

A vague region is interpreted by the egg-yolk model as a pair of nested crisp regions representing its maximal (the egg) and minimal (the yolk) possible extensions. A region is itself crisp if and only if the yolk is equal to the egg.
The egg-yolk semantics suits spatial domains, as it provides a very simple model of indeterminate spatial objects. When a spatial region is interpreted as an egg-yolk pair, this means that the region definitely includes the yolk and is definitely included in the egg.

- *Fuzzy Logic* [116]
  This is a variation on the semantics of classical logics which originated from the theory of fuzzy sets [51]. It has become popular in AI for modelling uncertainty and, also, vagueness.
  Fuzzy logic is an adjustment of classical logic aimed at overcoming the dichotomy between pure truth or pure falsehood. Its interpretation is based on a range of *degrees of truth*, where truth-value predicate p is generally expressed as a value $p_v \in [0, 1]$. It can be considered as a statistical approach to logic, where connectives are interpreted according to statistic functions. There have been spatial applications of fuzzy logic, for example a fuzzified formalisation of RCC [93].

- *Supervaluation Semantics* [47, 61, 111, 97]
  According to supervaluation semantics, a vague language admits a range of different referents for terms, and different truth-values for predicates and sentences. It is based on the principles of classical logic, and it is generally regarded as opposing statistical truth-functional approaches such as fuzzy logic.
  A formula admits multiple models, each obtainable via some form of assignment of referents to terms and truth-values to predicates. Each of these assignments is called a *precisification* and allows one to obtain a precise interpretation of a logical symbol, and there may be admissibility constraints on precisifications. Propositions true in all precisifications are said to be supertrue, and partial assignments leading to imprecise interpretations are also possible. The interpretation of the entire language is given by a *supervaluation*, which is a collection of all assignments for all the precisifications.

Supervaluation semantics has applications in a linguistic context, for instance the meaning of vague adjectives can be mapped by a function to a precise meaning [56].

- *Standpoint Semantics* [11, 14]
  Standpoint Semantics refines and extends supervaluation semantics with the aim of modelling specific vague concepts within a specific application domain. It is based on the fundamental notion of *standpoint*, which is taken every time an assertion is made. A standpoint is partly made of the observer's beliefs about the situation under consideration, and partly of his judgements about the applicability of certain concepts. A precisification then corresponds to a particular standpoint that describes the precisification via a finite set of parameters, specifying thresholds on the value of observable properties characterising vague concepts. This is represented in the syntax in the form of parameterised propositional expressions (Sec. 4.5).

Several other formalisms for vagueness in logics have been proposed, such as a set-theoretical formalisation of *granular partitions* [20] with an application to spatial concepts [21], comparison classes to account for adjectives' context-sensitivity [48], modal logics of vagueness with a focus on the sorites paradox [53] and an algorithmic attempt to disambiguate concepts by measuring the appropriateness of labels attached to terms [67].

The egg-yolk model, although well axiomatisable, appears to be too general, for instance it cannot account for any constraint to be applied on the egg's and/or yolk's shape or extension. For a geographic feature or other real world entities, it would be desirable to have a more structured model.

Fuzzy logic has the potential to model the blurred applicability boundary of a vague concept in terms of degrees of truth of a logical term. However, its semantics disrupts the entailment rules of classical logic, in particular the laws of non-contradiction and excluded middle. In fact, fuzzy models may assign a certain degree of truth to fundamentally false expressions such as $p \wedge \neg p$, rather than interpreting them as definitely false. Similarly, expressions such as $p \vee \neg p$ may be interpreted as true to some degree rather than definitely true [103, pag. 189–203]. Theories of degrees of truth and their relations to vagueness are also discussed in [60].

Supervaluation semantics rejects the principle of bivalence. In fact, predicates with borderline cases are indeed neither true nor false, as they are true in some precisifications and false in others. On the other hand, classical logic entailment rules are preserved; indeed, the law of excluded middle and the principle of non-contradiction hold. Propositions such as 'either Ryan is tall or not tall' and 'Ryan is tall and not tall' are respectively *definitely* true and *definitely* false, regardless of Ryan's height or any interpretation or precisification of the concept 'tall'.

A supervaluationist approach to the sorites paradox, mentioned at the beginning of this section, results in its premise being falsified, thus eliminating the contradiction. Given a series of sorites individuals, for example a series of people $x_1, x_2, \ldots, x_n$, whose heights range from 2.0 to 1.5 metres, arranged in descending order and with a difference of 1mm between any x_i and x_{i+1}'s height, the premise of the sorites paradox states:

$$\forall\, i\, [\, tall(x_i) \rightarrow tall(x_{i+1})\,]$$

In supervaluation semantics, for any possible precisification of *tall*, it will always be the case that $\exists\, i\ [\, tall(x_i) \wedge \neg tall(x_{i+1})\,]$. It follows that the premise of the sorites paradox above is *definitely* false precisely because it is false on all admissible precisifications.

Supervaluation semantics can be affected by *Second-Order vagueness* [112], a form of 'vagueness of vagueness of predicates'. This happens when borderline cases for a certain predicate do not have crisp boundaries themselves, i.e. it is not possible to draw a crisp line between a non-borderline instance of a predicate and a borderline one. Formally, if a predicate p is denoted to be definitely true with notation $D(p)$, a borderline case is simply $\neg D(p) \wedge \neg D(\neg p)$, in other words 'it is not definitely true that p, and it is not definitely true that $\neg p$'. An instance of second-order vagueness can be formalised by $\neg D(D(p)) \wedge \neg D(\neg D(p))$ and higher-order instances follow recursively. Keefe suggests that "any theory of vagueness must recognise and accommodate this phenomenon, and not simply avoid problems with the boundary between p and $\neg p$ by postulating a precise category of in-between cases. Many theories of vagueness have been thought to fail at this hurdle" [61].

Despite higher-order vagueness, the representation of borderline con-

cepts by means of precisifications, together with the preservation of classical logic principles and entailment rules, make supervaluation semantics a prime choice for the development of an ontology of vague concepts. Still, a formalism based on this semantics introduces an additional element, constituted by precisifications, which has to be formalised and represented in the language in order to establish the truth value of borderline predicates.

Standpoint semantics explicitly represents precisifications in the form of threshold parameters embedded in the language syntax, and it is our choice for developing the ontology described in Chapter 4 and model the concepts in Chapter 5.

2.5 Machine Learning and Logic-based Event Recognition

The work presented in this thesis fits within the class of logic-based event recognition systems. A system of this kind processes input in the form of a large set of low-level time-indexed events and aims to infer occurrences of high-level events of interest. Within the domain of this work, the input stream is constituted by the data resulting from the algorithmic processing of images captured by a camera, and the high-level events are the motion verbs aimed to be recognised.

Logic-based event recognition systems present a defined and explicit declarative semantics; it is argued that this provides advantages in terms of traceability of events, validation and extensibility to different settings and domains [81], in contrast to non-logic based event recognition system based on more procedural approaches which are prevalent in industrial applications.

The declarative semantics of a logic-based event recognition system is required to provide capabilities for spatio-temporal representation and reasoning to allow for the definition of high-level events in terms of properties and constraints on the low-level input stream [10]. However, in a purely logic-based system, such as ProVision, the definition of such events is still a manual process which can be time-consuming and lead to errors, lack of specificity and/or lack of generality. The definitions also need constant maintenance in order to reflect changes in the application, or respond to variations in the quality and granularity of the low-level

events extracted from the input stream. For this reason, the most attractive systems in the literature employ Machine Learning techniques for the automatic extraction and refinement of high-level definitions. As these definitions and their underlying formalism often assume the shape of a logic program, most learning-based approaches focus on Inductive Logic Programming (ILP [77, 27, 26]).

2.5.1 Inductive Logic Programming

Inductive Logic Programming (ILP) is a technique for supervised learning of high-level definitions, where a system is trained on a set of positive and negative examples providing the basis on which logical definitions are learned. The reasoning system resulting from this learning process is then tested on a separate set of examples to measure its accuracy.

The aim of ILP is to induce a general theory about the set of training examples in the form of a set of hypothesis expressed within a first-order logic program. First-order logic allows for more expressive power than classical machine learning approaches which induce propositional hypotheses. The learning process of an ILP-based learning system can be also guided by the specification of background knowledge coming, for example, from human expertise [63].

The principal elments of an ILP-based learning system are the following:

- A set of positive and negative examples, respectively E^+ and E^-, in the form of ground facts.
- A hypothesis language $\mathcal{L}_H$ from which hypotheses are defined.
- Background knowledge B, in the form of a set of Prolog-style clauses such as $p \leftarrow l_1, \ldots, l_n$ (where p is the head of the clause and l_i are literals).

The search for hypotheses during the ILP induction phase aims to construct clauses $H \subseteq \mathcal{L}_H$ such that each rule H is complete and consistent. Completeness signifies that all positive examples can be deduced from H and the background knowledge, or $B \wedge H \models E^+$, and consistency signifies that no negative example can be deduced from H, or $B \wedge H \wedge E^- \models \square$ [80].

A general strategy for learning hypotheses starts by choosing a particular example $e^+ \in E^+$, constructing a first-order clause $h \in \mathcal{L}_H$ that entails e^+ and does not entail any $e^- \in E^-$ and eliminating all examples in E^+ covered by h. The process is then iterated over the remaining examples in E^+ to learn a new hypothesis H' refining clause H.

The space of all possible clauses H that entail example e^+ with respect to B is called *version space*, which has the empty clause as the most general clause at its top and the most specific clause entailing e^+ at its bottom. Several strategies may be employed to explore this space: the search may be general-to-specific, i.e. starting at the empty clause and proceeding by specialisation, or specific-to-general, i.e. starting at the bottom clause and proceeding by generalisation. Exhaustive searches of the version space are often impossible due to the exponential complexity of the set of all possible clauses, therefore most ILP systems design search algorithms for pruning the version space and guiding the search. The design of such algorithms constitutes the main challenge of ILP and several have been proposed in the literature, such as FOIL [83], Progol [79] and Aleph [1].

ILP and Event Recognition from Video

An example of a logic-based event recognition system for the detection of events in video scenes using ILP can be found in [42, 41]. The origin of the low-level set of events for this system is constituted by videos filmed by 8 static cameras situated in an airport apron area, providing different views of a same scene where aircraft movement and operations take place (e.g. loading, unloading, refuelling etc.). The videos are processed by tracking algorithms to extract three-dimensional data for objects moving on a ground plane, and the actual set of low-level events constituting the input of the event recognition system results from the conversion of the tracking data into relational data defining spatio-temporal relations holding among objects in the video scene, namely Allen's temporal relations (Sec. 2.1.1) and spatial relations *surrounds*, *touches* and *disconnected*, generalisations of RCC (Sec. 2.1.2).

Most of the challenges in learning event models from this kind of data using ILP are similar to the ones faced by the system presented in this thesis. These are essentially the very large size of the dataset, which results in a very large hypotheses search space, and the noise and uncertainty from

the tracking, which results in hypotheses triggering many false positives. The authors tackle this problem by using a *learning from interpretations* setting, which views each positive example as a set of spatio-temporal facts constituting an *interpretation* [85], and considering a tree-structured *type hierarchy* of objects involved in each event with the introduction of a type-refinement operator in order to extract efficient models for ILP learning.

The search strategy of this event recognition system is based on Progol refinement operator which finds the most specific clause from the training examples, background knowledge and user-defined syntactic biases in the form of *mode declarations* specifying which predicates from the background knowledge are expected in the hypothesis [78]. Hypotheses in the space surrounding this most specific clause are assigned a score (based on the number of positive and negative examples covered, length of the clause etc.) and this space is searched with an A*-based algorithm to identify the hypothesis with a maximum score. The hypothesis thus found can then be augmented with an explicit temporal representation of when the event occurs in the clause head.

Given the large size of the dataset in this system, hypothesis evaluation in this search process can be too time-consuming, and the nosiy character of the data tends to make hypothesis too general and entail many false positives. The authors introduce an object type hiearchy and type-refinement operators which allow to specify the type of arguments in the hypothesis, and conclude that such a *typed ILP* system allows for efficient learning of event models from video due to the acceleration of the hypothesis evaluation stage and the reduction in the number of false positives entailed by each hypothesis.

Event Calculus and ILP

The Event calculus formalism introduced in Sec. 2.2, and further specified in the formalisation of the ontology presented in this thesis in Sec. 4.4, expresses high-level event definitions as first-order logic predicates which can be directly expressed in a logic programming language such as Prolog. Thus, ILP methods are a very good candidate for the automatic learning of such definitions.

For example, in order to learn the fact that event e occurs over temporal interval $[t_1, t_2]$, corresponding to the definition $\mathsf{Occurs}(e, [t_1, t_2])$, one

would have to provide positive examples E^+ and negative examples E^- using the Occurs predicate and a background knowledge B including the stream of low-level events expressed in the form of $\mathsf{HoldsAt}(f, t)$ predicates. Hypotheses learnt by the system will mostly be clauses of the form $\mathsf{Occurs}(e, [t_1, t_2]) \leftarrow \mathsf{HoldsAt}(f_i, t_i), \ldots, \mathsf{HoldsAt}(f_j, t_j)$.

However, the automatic inference of an Event Calculus logic program involves learning hypotheses for which training examples are not available, which means that induction cannot be directly applied to produce the hypotheses [10]. In such cases, abductive logic programming [4, 35, 36] may be used to learn intermediate groud rules using the examples provided in terms of HoldsAt predicates and other Event Calculus rules in the background knowledge B, and inductive logic programming may then generalise the outcome of abduction. An example of a system combining abduction with induction for learning EC programs is the XHAIL system [88]. Briefly this system is based on the construction of preliminary ground hypotheses in a *Kernel Set*. A three-stage process then follows, in which abduction is first used to produce the head of the hypothesis clause, deduction is used to produce the literals in the body of the clause and induction is used to generalise the clauses thus produced in the Kernel Set. A sample application of XHAIL to a transport system network scenario can be found in [10, Sec. 3.3].

2.5.2 Markov Logic Networks

In the context of event recognition from video scenes, the low-level stream of events constituting the input of the recognition system often suffers from quality issues, such as errors and noise in the data leading to incompleteness and/or inconsistency in the representation of positive and negative training examples for the automatic learning of event definitions. This is particularly true when videos are processed automatically by tracking algorithms, but it can also happen for manually annotated data (see Chapter 6).

Logic-based formalisms such as the one presented in this work have the advantage of compactly representing complex definitions in a declarative semantics, but do not naturally handle uncertainty as hypothesis violating even a single formula in the knowledge base are automatically discarded. A combination of the ILP techniques in the previous section

and probabilistic models resulted in systems for Probabilistic ILP (see [86] for an overview).

These evolved in Knowledge-Based Model Construction methods, in which a logic-based language allows the generation of a propositional graphical model on which probabilistic inference is applied. In particular, Markov Logic Networks is a recently developed framework that considers uncertainty in the representation, reasoning and learning of event models [91, 40] and has been applied to the task of event recognition [101].

Essentially, in a Markov Logic Network (MLN), the probability of a world expressed by a hypothesis increases as the number of violated formulae decreases. It follows that a hypothesis violating certain formulae becomes less probable but not impossible as in first-order logic. This is represented in the formalism by associating each first-order logic formula F_i with a weight w_i, where higher values for w_i yield the fact that F_i constitutes a stronger constraint. A set of Markov formulae (F_i, w_i) effectively represents a probability distribution over possible worlds.

In order to produce a MLN graph, all formulas are translated into clausal form, where the weight of each formula is distributed among its clauses, and the clauses are grounded using a finite set of constants C. Each node in the graph is represented by a Boolean variable and corresponds to a possible grounding of a predicate. Ground predicates appearing in the same ground clause F_i are connected to each other and form a clique in the network. Each one of these cliques is associated with clause weight w_i. A ground MLN is composed of nodes corresponding to a set of random variables (ground predicates) and a probability distribution over states can be computed.

In event recognition, the low-level stream of input events provides the set C of constants to ground the clauses for producing the network expressing the knowledge base on which to learn high-level event definitions. Event recognition is performed by querying a ground MLN about a particular high-level definition. The set of random variables of the network is partitioned in sets of query variables, representing the definition of interest, a set of evidence variables, representing the detected series of low-level events, and a set of hidden variables, which correspond to the remaining variables with unknown values. The query is resolved through *conditional inference* that computes the probability of a query variable given

some evidence, for which different algorithms can be employed. The computation of these queries can be subject to enhancements by sampling methods, such as Markov Chain Monte-Carlo algorithms which would walk through the network and draw a set of sample states from the graph of possible states.

The process of learning a Markov Logic Network involves the estimation of the weights of the network and/or the first-order clauses which shape the network structure from a set of training examples. Weight learning involves establishing the weights of the clauses that represent the definition of high-level events translated into clausal form, in which different clauses derived from the same definition may be assigned different weights. The actual weight is calculated by refinement of a likelihood function, which measures how well the probablistic model of a MLN fits the training data [40, Sec. 4.1]. The network structure of a MLN can be learned from the training data through a preliminary Inductive Logic Programming stage followed by weight learning stage [40, Sec. 4.2].

An application of the above principles from Markov Logic Networks and Markov Chain Monte-Carlo algorithms to the task of automated task of learning event classes from video for event recognition can be found in [99]. This application is based on the same aircraft scenario mentioned in Sec. 2.5.1, and event predicates are modelled within a graph of qualitative spatio-temporal predicates representing the interactions between sets of objects. Two events are said to be similar if their graphs are similar. Probablity distributions are then computed over the set of event classes and within each event class as a distribution over the set of qualitative spatio-temporal *event graphs*. Events are generated by sampling event graphs from the distribution over the event classes and constructing a structure called *activity graph* which combines all event graphs and specifies relationships between objects across different graphs. Finally, this activity graph is embedded with concrete objects, spatial positions and temporal intervals. The model thus generated forms a probabilistic framework for finding the most likely interpretation, i.e. the most likely event classes, event graphs and activity graph that generated a particular observed occurrence of an event.

Chapter 3

Issues in Event Classification

The formalisation of an ontology for the definition of vague spatial concepts and motion verbs faces several issues, namely vagueness and the individuation of the most relevant semantic characteristics in the meaning of each concept. The implementation of such an ontology into a system performing automatic event recognition adds further complications, namely uncertainty, saliency and granularity.

In Sec. 3.1 the issue of vagueness is examined in more detail starting from the general overview of Sec. 2.4, and framed into a more spatial context related to the specific set of concepts in our domain. In Sec. 3.2 the most salient characteristics of the meaning of motion verbs are introduced and informally discussed. These include the characterisation of concepts such as distances, trajectories, speed, forces, contact and others. Most of these semantic characteristics and related concepts are analysed formally in Chapter 5. In Sec. 3.3 the issue of saliency of event occurrences and how it is influenced by context and semantics is discussed. Sec. 3.4 introduces the importance of contextual information and how an ontology could be structured in order to incorporate it. The issues of granularity (Sec. 3.5) and uncertainty (Sec. 3.6) bear more relation with the practical task of the implementation of the event recognition system. Respectively, they refer to issues around the level of detail in which things can be looked at and issues with errors, noise or inaccurate representations in real data.

The discussion and considerations involving the issues presented in this chapter have a substantial impact on the methodology guiding the formalisation of the ontology in Chapter 4 and the modelling of verbs

in Chapter 5. The issue at stake, and the purpose of this work, is not the achievement of a semantically exhaustive characterisation of vague motion verbs. It is rather the possibility to achieve an effective one focusing on properties that can be inferred given the issues presented here, especially considering the granularity and uncertainty of the kind of data that the system can realistically expect to operate on. Efforts have been made to maintain the formalism at a general enough level such that one could extend its definitions to take advantage of more detailed or accurate data.

3.1 Vagueness

The main challenge in developing an ontology of vague motion verbs for event recognition is vagueness. Defining the meaning of the verbs listed in Table 1.1 involves the characterisation of spatial concepts, several of which are vague.

For example, recognising the ocurrence of an event such as '*a* is arriving at *b*' with a reasoning system involves the definition of the concept 'arrive' and establishing whether it is applicable to the description of the event being observed. This can be done by characterising the meaning of 'arrive' as '*a* is arriving at *b* if *a* is moving towards *b*, *a* finds itself near *b* and eventually stops at *b*'. Such a formalisation unfolds in the introduction of further vague concepts such as 'moving towards', 'near', 'stop' and *a* being 'at' *b*. This simple example demonstrates how quickly the issue of vagueness can escalate even within a narrow domain.

Vagueness is distinct from *generality*, which has to do with the range of conditions under which a sentence holds or the range of individuals over which it is applicable. A proposition such as 'I earn less than premier league footballer Smith' is general, as it does not go to great lengths in specifying my income, but not vague as it is possible to precisely establish whether my income is less than Smith's. Vagueness is also distinct from *uncertainty*, which has to do with limited or insufficient knowledge about some thing in the world. Vagueness arises from an intrinsic indeterminacy which cannot be made less indeterminate by simply augmenting the precision or obtaining more data [50]. The previous example may represent an uncertain statement if Smith's income were to be undisclosed, nevertheless it is a fact that such information exists, making it possible to establish

whether the proposition 'I earn less than Smith' is true or false.

In its de dicto characterisation, vagueness is a linguistic phenomenon due to the lack of precise criteria for the applicability of concepts, and arises from several classes of linguistic terms. There exist different theories of vagueness with different emphasis on which among objects, language or knowledge is the source of vagueness (see Sec. 2.4), however this work does not aim to enter the debate by reinforcing one particular theory. Instead, a pragmatic approach has been followed, guided by the practical purpose of establishing a reasonably appropriate characterisation of vague concepts, given our specific ontology reasoning task within the domain of motion verbs.

It is possible to identify different kinds of vagueness which affect the precise demarcation of the applicability of a concept [13]:

- *Simple ambiguity*
  Certain concepts admit multiple, and different, meanings or interpretations. For example the verb 'Pass' may mean 'to cross a boundary' but also 'to hand an item to somebody'.

- *Sorites* or *Threshold vagueness*
  A concept's applicability boundary is blurred and depends on the continuous variation of some observable property of the sample to which the concept applies. For example the applicability of the concept 'near' may be decided on the basis of the distance separating a and b. However, establishing a fixed threshold for crisping the boundary has counterintuitive consequences sharply separating very similar instances (see also Sec. 2.4).

- *Deep Ambiguity*
  A concept not only presents a blurred applicability boundary, but there are clusters of different and overlapping observable properties on which this boundary depends, yet it is unclear as to which ones are necessary or relevant. In the previous example, the applicability of 'near' may depend on linear distance between a and b, the viable routes between a and b, the time needed to travel on each one, the type of terrain involved and, often, on a combination of all these.

The problem affects our ontology especially when qualitative concepts are involved. An ontology based on classical logic does not allow for

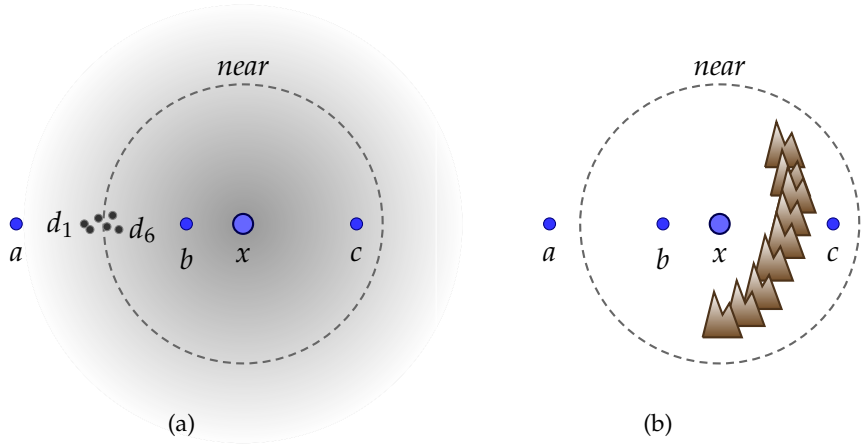


Figure 3.1: Instances and definitions of 'near'

degrees of truth, and each logical proposition can only be evaluated as being true or false. The process of individuating criteria that establish an applicability boundary for vague concepts would clearly lead to the demarcation of true and false instances. The problem lies in the fact that, often, there is no such objective set of criteria as these too often depend on several factors. For example the type of objects involved, characteristics of the environment or the occurrence of certain actions in the past or in the future.

Let us consider the concept 'near' again. This can be formalised with the time-indexed logical predicate $\mathsf{HoldsAt}(near(b,x),t)$, which holds if and only if x is near b at time t. Figure 3.1(a) shows an illustration of the blurred applicability boundary of *near*, represented by the shaded area around x. In some ways, we could say the predicate *near* holds *more* in the immediate proximity of x (for example for point d_6), and *less* as one moves away from x (for example for point d_1). If we consider the linear distance between b and x to be the observable property determining the applicability of *near*, one may define a threshold m_d such that the predicate $near(b,x)$ holds if and only if the linear distance between b and x is less than m_d. Such a crisp boundary is represented by the dashed circle surrounding x. However, this process causes the separation of points $d_1,\ldots,d_6$ between instances for which the predicate *near* holds, such as d_6, and instances for which the predicate does not hold, such as d_1. This is somehow conterintuitive as points $d_1,\ldots,d_6$ are very close to each other, and it would appear strange for some of them to be near x and for some

others not to be.

Figure 3.1(b) illustrates the same example under a slightly different issue. We could imagine that a, b, x and c represent towns on a topographical map, and there is some mountainous terrain between x and c. A precise interpretation of *near* based on linear distance thresholds, as in the previous example represented by the dashed circle, would determine the predicates $near(b, x)$ and $near(c, x)$ to be true and $near(a, x)$ to be false. Given the context, this is also counterintuitive because, despite c's shorter linear distance from x, the geographical features suggest a higher degree of separation and a classification where a is nearer to x than c is.

The methodology that guided the development of the ontology of vague concepts in Chapter 4 and the modelling of motion verbs in Chapter 5 has followed a pragmatic approach similar to the *epistemic stance* of Lawry and Tang [68], a weakened form of epistemic vagueness.

In epistemic vagueness there is an objectively correct set of criteria for precisely determining the applicability of a vague concept, but this set of criteria is unknowable due to the uncertain and inconsistent meaning of words in natural language. For example, in the sentence 'a is near b', epistemicists would believe in the existence of criteria for precisely determining which objects are near b. The epistemic *stance* maintains that decision problems regarding assertions can find it useful to assume an epistemic view of vagueness and thus the existence of a clear dividing line between concept demarcations, even though this dividing line is not necessarily an objective fact. In other words, an artificial agent acting according to the epistemic stance would behave as if the epistemic view was correct. In the previous example, one may not believe in the existence of an objectively determinable boundary for the concept *near*, nonetheless assume such a fact due to the simplification of reasoning tasks involving the concept.

This model of linguistic vagueness — assuming the existence of precise criteria determining the applicability boundary of concepts according to observable properties of objects — essentially constitutes a negation of the premise of the sorites paradox, hence transforming vague concepts into crisp ones. The objectionability and controversy of this model of vagueness are balanced by the substantial simplification it brings to the formal semantics of vague concepts [103, p. 139]. For this reason, an application of supervaluation semantics appears particularly suitable to the model

resulting from the epistemic stance. In this semantics, precise interpretations of vague predicates are expressed by *precisifications* (see Sec. 2.4). In the ontology formalism introduced in Chapter 4, precisifications of vague concepts are modelled with explicit thresholds linked to observable properties relevant to the demarcation of the concept applicability boundary (see Sec. 4.5).

Despite not necessarily believing in an epistemic nature of vagueness, we do believe in the practical utility of reasoning with vague concepts as if they had a precise though indeterminate intepretation. This underlying assumption coupled with a supervaluationist approach has been the guide for the formalisation of vague concepts throughout the rest of this work.

3.2 Verb characterisation

Table. 1.1 includes concepts with varying complexity and difficulty, from actions that appear relatively simple, such as Move or Touch, to actions that unfold in the characterisation of more complex sub-concepts, such as Exchange or Replace.

The inclusion of these concepts in a formal ontology involves the analysis and definition of several vague sub-concepts, for example the fact that two objects are 'near' each other at a particular time, or that an action is 'fast'. The main kinds of vagueness overviewed in the previous section affect the motion verbs and related sub-concepts to be formalised in the ontology.

Some concepts have simple ambiguity. For example Pass may mean 'to cross a boundary' or 'to hand an item to somebody'. Similarly, Exchange may refer to either two people reciprocally giving and receiving an item to and from each other, two people exchanging their respective position or a single person replacing an item with another in a particular location (this latter meaning would be synonymous to Replace).

Defining concepts such as 'near' or 'fast', involved in concepts such as Approach or Run, is a classic example of sorites vagueness. The previous section analysed the example of 'near' and the fact its applicability boundary depends on a continuous variation of an observable property, namely the distance between two objects, but can also depend on other elements of the context in which the action is taking place.

Most concepts not only show instances of sorites vagueness, but also of deep ambiguity, as their characterisation depends on a cluster of observable properties and it is unclear as to which are relevant or necessary. For example, defining the characteristics of an action such as Chase can be done on the basis of the trajectory of the two objects, their distance, their speed or the motivation that triggered one object chasing after the other. Yet, certain aspects may be more or less relevant in particular situations.

Most of the time, all three types of vagueness mentioned above manifest themselves when one tries to unravel a verb into its formal characterisation. Below, the most salient semantic characteristics of objects and actions constituting sources of vagueness are summarised, with a focus on the ones more likely to be visibly observable and capable of being formalised in term of their observable properties. They are examined in greater detail in Chapter 5.

Speed of actions

Verbs such as Run, Chase, Flee, Snatch, Throw and, to some extent, Fall refer to the notion of a particular motion action to be 'fast'. However, given that the type of objects involved and their manner of motion are different, there are different thresholds on and different observable properties playing a part in formalising 'fast'.

For example, Run generally involves a person engaged in a particular manner of motion (see below) which allows the person to move fast. The threshold precisifying the concept 'fast' in the context of an action of type Run is different from the threshold precisifying the fast movement of a vehicle. Additionally, the speed of a runner is different and depends, for instance, on whether the person is a child, adult or elder.

The verb Snatch is of a more subtle kind. In fact, most would agree it is an action which happens 'fast', but this may not always be the case. For example, if a man waiting at a busy station concourse is standing next to its suitcase among a crowd of people and is looking the other way, another person may grab hold of the suitcase and walk away, snatching it without being particularly fast. This is an example of deep ambiguity, and also of the fact that certain properties, such as 'the unawareness of a person', appear extremely challenging to formalise.

Manner of motion

Verbs such as Walk, Run, Dig and Bury refer to motions which are carried out in a particular way, often following a specific pattern. For example, Walk refers to the act of a person lifting one of his feet, moving it not too far ahead, putting it back on the ground, repeating the action with the other foot and periodically repeating this pattern over some time interval. Run refers to a similar pattern where the legs bend differently, the distance between each step is longer, and both feet may be off the ground at the same time instant (the verbs are analysed in Sec. 5.1.2 and 5.1.3. Dig involves a person holding a tool which is suitable for displacing material (e.g. soil), lifting a certain amount of material, displacing it and dropping it somewhere else, and carrying out this motion periodically. Bury involves performing the same pattern of Dig and placing an object in the space resulting by the effect of Dig, then performing an action following a manner of motion opposite to Dig in order to cover the item.

Direction

Verbs such as Approach, Go, Jump, Bounce, Flee, Follow, and Chase refer to the fact a particular motion is oriented towards a specific direction.

Approach denotes a motion where an object is moving in the direction of another. This would involve defining the concept of direction, and a formal way of establishing that the motion of an object is directed towards another object. One possibility would be to measure whether the distance between two objects is decreasing. However, there are examples in which this criterion may not be the most appropriate. Fig. 3.2 shows the position of an object o at time instants t_1 and t_2 approaching d. In Fig. 3.2(a), motion happens along a straight line, hence the distance between o and d is lesser at t_2 than at t_1. In this case, a distance-based criterion for directed motion seem appropriate. In Fig. 3.2(b) instead, the obstacle represented by the blue rectangle has the effect that o approaches d by *increasing* its distance from d at time t_2, still o is approaching d.

Other examples involve establishing whether a motion occurs towards a pre-defined direction, such as Jump or Bounce, where generally objects respectively move suddenly upwards, or periodically repeat an up-and-down motion. However, there exist instances where this may not be the

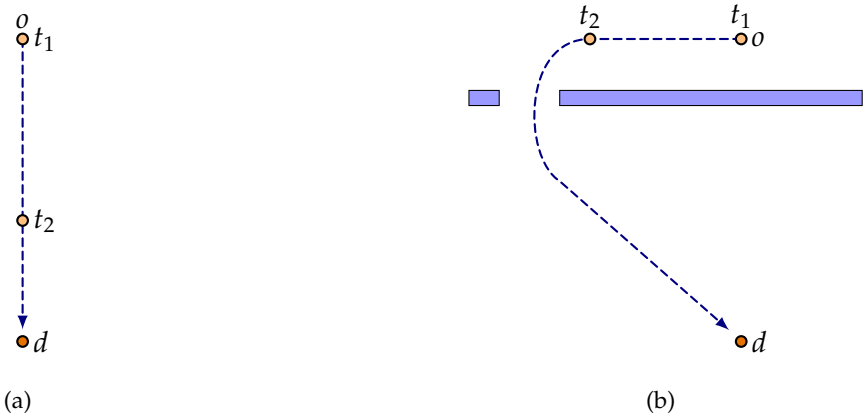


Figure 3.2: Verb Approach – Different routes

defining factor, as in the example where a man is bouncing a ball against a wall in a horizontal direction.

In Flee, Follow and Chase the direction is defined in terms of another object direction. In Follow, a second object is moving in the direction of a first object. However, the second object may not be replicating exactly the same movements of the first, as it may temporarily deviate from a directed route to disguise itself. Flee represents the opposite scenario, where the first object is moving to a direction which is *opposed* to the direction defined by the second object chasing after it.

Distance and boundaries

Verbs such as Approach, Arrive and Leave refer to the observable property of 'distance' and the qualitative concepts of 'near' and 'far'.

The verb Approach refers to an object getting nearer to another over a certain time interval. The concept of 'near' can be formalised by referring to the observable property of distance between two objects. However, we have already seen in the examples in Fig. 3.1(b) and 3.2(b) that there are instances where this observable property may not be the most relevant one. In Fig. 3.2(b), most would agree that o is approaching d over interval $[t_1, t_2]$ knowing that there is an obstacle which determines a particular route that o has to follow.

This example shows that modelling a distance metric is akin to establishing an *effort-space*, a measure expressing how the actions of a subject

would lead it to be closer to its target. In Fig. 3.2(b), such a measure would establish that the effort needed for o to get closer to d at t_2 is less than the effort needed at t_1; it follows that o's motion over $[t_1, t_2]$ has determined a reduction in the effort-space thus constituting an occurence of Approach. An effort-space measure takes into account the characteristics of the object and surrounding environment. For Approach this would be a combination of type of object, possible paths between the object and its target destination, type of terrain etc. For example, in map-based scenarios, certain routes, such as motorways or off-road tracks, may only be applicable to certain types, such as cars and 4x4s, and each has different costs. This effort-based metric can be generalised and extended for other verbs too, such as Arrive, below, or more complex actions such as Follow or Bury.

The verb Arrive and its opposite Leave refer to an object respectively reaching and leaving another object or destination. Formalising this concept for example, involves establishing some form of boundary around such destination and formalising whether the object is approaching and getting close to this boundary, again involving the concept of distance. The individuation of such a boundary depends on different observable properties of the destination.

These verbs are examined in detail in Sec. 5.2

Relations

Verbs such as Follow, Flee and Chase, and to some extent, Throw and Catch refer to a particular relation being established between two objects.

Recognising an instance of Follow involves formalising the fact that there is a relation between two objects such that the motion of the object being followed determines a particular direction or trajectory on the object that is following it. Chase is similar in this respect, with the additional characterisation that the motion of the chaser is likely to be fast with the intention of reaching the chased. The relation holding between the two objects is even more prominent, as the type and actions performed by the chased may have triggered this intention on part of the chaser (e.g. a predator chasing its prey, or a victim of pickpocketing chasing the perpetrator of the theft). These verbs are analysed in Sec. 5.3.

The verbs Throw and Catch also suggest some close relation between a person and the object being thrown or caught. For Throw, the object has

to be in possession of the thrower, and this is accelerated and its motion is directed according to the force and motion of the thrower. Similarly, Catch involves an object moving towards the catcher and terminates with it being in possession of the catcher.

Trajectories and routes

Verbs Fly and Fall involve the formalisation of the notion of trajectory. A particular type of trajectory, in fact, can indicate the specific type of motion and the causes underlying it. In fact, most interpretations would agree on the fact that Fall refers to a primarily vertical motion of an object towards the ground under gravitational force, whilst Fly refers to a primarily horizontal motion of an object under some kind of inner force propelling the object (e.g. an aeroplane) or an inertial force resulting from some previous action or event (e.g. a ball that has been thrown in the air by a person). However, establishing whether a motion is an example of an event of type Fall rather than Fly in terms of its trajectory is vague. The examples in Fig. 3.3 show different types of such motions (assuming they all represent an observation from the same equivalent perspective):

- Fig. 3.3(a) shows a vertical motion, very likely to represent an occurrence of Fall.
- Fig. 3.3(b) shows a motion which is not exactly vertical, but most would agree it is vertical *enough* for it to still represent an occurrence of Fall.
- Fig. 3.3(c) shows a motion which starts mostly horizontal and ends mostly vertical; overall, it still looks like it could represent an instance of Fall.
- Fig. 3.3(d) shows a split motion. It starts as a diagonal motion away from the ground, and probably caused by some force different from gravity, thus constituting an occurrence of Fly. However, the second part seems more akin to an occurrence of Fall. Overall, it would appear as if the objects moved under some inertial force that exhausted itself at half-point hence leaving the object under the influence of gravity. However, if this movement is generated by an inner force of an object capable of flying (e.g. a bird, a helicopter), the entire

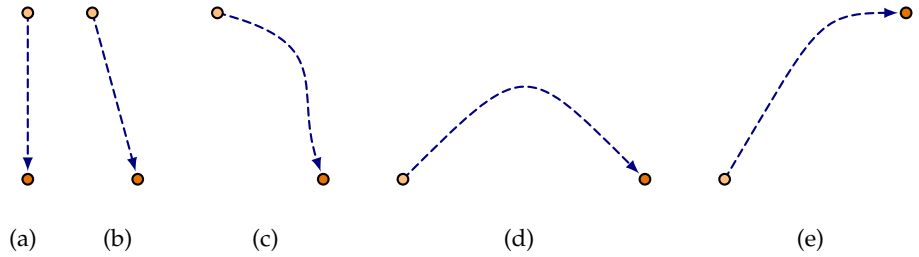


Figure 3.3: Verbs Fly and Fall – Trajectories

occurrence may be classified as Fly, with the second part part of the motion being a controlled descent executed by the object. Still, if the helicopter had broken down, or the bird had been shot, the descent would cease to be controlled and would again be considered as an occurrence of Fall.

- Fig. 3.3(e) shows an object moving away from the ground and eventually assuming a horizontal trajectory (e.g. an aeroplane taking off). This particular motion seems to constitute evidence of the object being capable of an inner force, thus likely to constitute an occurrence of Fly.

The examples above show the complexity of identifying trajectories and associating an occurrence of a motion to its characteristic trajectory, and of individuating the properties of an object that determine certain kinds of motion.

Other verbs are slightly simpler in terms of the trajectory characterising their motion. For example, verbs such as Lift, Raise, PickUp, PutDown and Drop refer to a motion whose trajectory is vertical and is either directed away or towards the the ground. Throw is ambiguous in this respect, as one may throw an item vertically in the air, horizontally or vertically towards the ground.

Contact

Verbs such as Touch and Push and most of the ones listed under the *Possession* section below refer to the fact that two objects are in contact with each other, or the fact such a contact is being established. Contact can be established in different ways. For example Push involves a kind of contact that

allows an object to exert a force on another object in order for the latter to move forward. For example a person pushing something generally places his/her hands flat on the object's most vertical edge, even though sometimes feet may be used instead. A person may be pushing an object even if this results in no movement whatsoever (e.g. pushing a locked door). The verb is analysed in Sec. 5.4.2.

Verbs Hit and Collide also refer to two objects coming into contact in a more disruptive and less controlled manner. They also may involve specific contact parts, for example a person generally participates in an occurrence of Hit by a quick and forceful movement of the hands. Kick, in some ways, is a specialisation of Hit where the contact part involved is the foot. The verbs are discussed in Sec. 5.4.4 and 5.4.5.

Possession

Verbs such as Have, Get, Receive, Give, Hand, PickUp, PutDown, Snatch, Take and Drop refer to the fact that an object (generally a person) is acquiring or relinquishing possession of an object. This is a high-level concept and there are many observable properties that could form part of the formalisation of the concept of possession.

For example, a person may Have an object by holding it in his/her hand. It would follow that an occurrence of Hold would also determine that the event Have is occurring at the same time. An occurrence of Hold may be formalised by specifying that the two objects are in contact by means of the person's hands. However, a person may Have an object by keeping it in a pocket, in which case a more complicated relation has to be formalised.

Similarly, the verbs Give, Receive and Get involve a transfer of possession from one person to another. An event such as Give can be relatively straightforward, for example a person is holding an object and then drops it onto another person's hands, or introduce layers of complexity such as carriers, for example a person gives a document to another person by post.

Size of objects

Verbs such as Catch, Lift, Take, Hold, Throw, Hand, PickUp, Carry and Haul refer to an action involving objects whose size or weight is within certain

ranges.

For example, PickUp refers to some person or, more rarely, some machine lifting an object from the ground or other kind of surface by holding it with hands or other suitable tool (e.g. litter pickers' tongs) in order to gain contact or possession of the object. The object needs to be of a relatively small size compared to the agent performing the pick up action. For example, an adult is able to pick up objects that a child may not. Similarly, a port container shifter will pick up objects with size and weight way out of league of many other agents. Most of the other verbs listed also under the possession section above refer to some contraints in the size of objects being possessed or relinquished.

Carry and Haul represent an example of two verbs describing essentially the same type of action with the difference between them being the size and/or weight of the objects being transported. A person performing an action of type Carry is likely to be transporting an object that can be lifted and moved by the person's strength and ability, although some aiding may be involved (e.g. wheels, a trolley). Haul generally refers to some moving object shifting a substantial load for which great effort and expenditure of energy are required. Sometimes the verb can be used figuratively when applied to people, denoting the struggling efforts of a person in trying to carry a too big item.

Force

Verbs such as Hit, Kick, Fly, Fall, Throw and Push refer to some kind of force that is being exerted from an object towards another and determine motion consequences on the object on the receiving side of this force. One of the ways this force can be exerted is through contact which determines movement or deformation (verbs Push and Hit, see also Sec. 5.4.2 and 5.4.4).

As mentioned earlier during the discussion of trajectories of movement, one can distinguish three main kinds of forces driving a motion:

- *Inner force*, produced by the agent itself. Such an agent would be capable of sustaining a self-propelled motion, such as Walk, or Fly. An agent capable of producing a force of this kind can also transfer it onto other objects, generating an inertial force.

- *Inertial force*, describing a movement originated by an initial impulse

generated by another moving object. The object moving under inertial force is subject to the momentum transferred during such initial impulse. This state lasts for some time until the inertial force is exhausted, mostly by being counteracted by some other force. For example, an agent may throw a ball in the air and the item will be moving under inertial force until gravitational force surpasses it, or a person may kick a box on the floor which will start to slide until the counter-action performed by the resistance between the box and the ground surface will cancel the inertial force effects. The visible transition between a movement driven by inertial force to a movement driven by gravitational force is generally smooth, for example the trajectory of an object may change from predominantly horizontal or oriented upwards to predominantly vertical and oriented downwards.

- *Gravitational force*, describing the way objects are attracted to the ground and is generally observed by a predominantly vertical trajectory with the object accelerating towards the downwards direction.

- *Disruptive force*, this force does not drive movement but is rather intended as causing a deformation or other form of change in the state of the object it is applied to. It is applied by an agent by transferring an inner force as a result of an event where forces involved are of relatively high intensity. For example, an occurence of Hit or Push involves an agent exerting a force onto another object. This force could either determine the motion of the object through inertial force, or act as a disruptive force altering the shape of the object (or, worse, its agent) if this is not capable of moving or is otherwise opposing resistance. This force is discussed for contact verbs in Sec 5.4.3 and 5.4.4.

Objects may be subject to a combination of the forces above, the challenge is to be able to characterise them in terms of the observable characteristics of a certain motion. The motion trajectory is a prime candidate, however the type of objects and events which occurred prior to that particular instance of motion are also relevant (e.g. a ball will be subject to inertial force if somebody has thrown it).

Intentionality

Verbs such as Hit, Collide, Jump seem to recall some kind of intentionality of motion. For example Hit and Collide describe actions which are in essence very similar, as they describe a generally forceful motion leading to two objects establishing contact in a disruptive or violent way. However, language use seems to hint that the subject of Hit is somehow more active in performing the action, conveying the idea that it is somehow responsible for the action which would have been avoidable. On the other hand, Collide seems to refer to some event that happened with no responsibility, or a contact that would have been unavoidable (e.g. 'the asteroid collided with planet Earth'). See Sec. 5.4.3 for a more detailed discussion.

Space and accessibility

Verbs such as Open, Close, Enter and Exit refer to certain characteristics of the space on which the action is being performed.

Enter and Exit, discussed in detail in Sec. 5.2.5 and 5.2.6, describe a motion that leads an object from being *outside* to being *inside* a space or vice versa. Such a space could be an open space as a field or a closed space such as a building. Closed spaces have particular accessibility relations describing the ways one can enter or exit them, i.e. most of the time, a person enters a house through a door. Identifying an occurrence of Enter involves recognising whether a subject is performing certain actions on the object allowing access to a space, for example opening a door and walking through it.

Verbs Open and Close determine a change in the accessibility of a space by an agent. Identifying an occurrence of Open involves defining how the accessibility of a space can be altered. For example, a door delimiting a space with a conventional handle can be opened by pushing down on the handle while at the same time pushing the door to swing it open.

Despite a certain triviality of the examples, recognising the occurrence of these events, especially in the case of closed spaces, is challenging due to the variety of ways and objects through which a space can be accessed or an item can be opened.

Duration

Verbs such as Stop, Touch, Arrive, Leave, Pass, Enter and Exit are vague in their temporal extension or duration, especially because most of them seem not to have any duration at all.

Recalling Vendler's linguistic classification of motion verbs in Sec. 2.3, this set of verbs corresponds to the category of *Achievements*, constituted of actions that happen over a time instant. Nevertheless, some subtle grammatical devices can still assign a duration to events of this kind. Rigorously, the utterance 'The car stopped' refers to the precise single instant in which the car has ceased to move. However, the utterance 'The car *is stopping*' suddenly extends the event over an interval. The vagueness in this instance is what observable properties of the motion lead one to determine when the car *starts* to stop.

This temporal extension issue, relevant to all the verbs listed above and analysed in greater detail in the relevant sections of Chapter 5, is related to the individuation of a suitable interval for event occurrences of achievements. For example, for Arrive or Enter such an interval would span some time before and after an object arrives at a destination or enters a space.

Some of these verbs can also be interpreted as static states. For example Touch, modelled in Sec. 5.4.1, can either describe the static occurrence of two objects being in contact with one another, or the dynamic occurrence of one object moving towards and establishing contact with another. An event of the static kind occurs over an interval lasting as long as the two objects are in a state of connection. Conversely, an event of the dynamic kind occurs over a *narrow* interval preceding and terminating on the connection.

3.3 Saliency

Event occurrences are not described by a single verb, as language allows for varied and sometimes colourful expressions. Expressions describing a particular action generally revolve around a central part of their meaning, and a speaker may wish to highlight details that are *salient* and hide others that are less relevant.

For example, most observers would agree on the fact that an event in which a person is repeatedly impressing motion on a ball using the foot is an occurrence of Kick. At the same time, the person is also performing the action of hitting the ball, thus a description by the verb Hit is acceptable too. Neither description is right or wrong, they simply highlight a different characteristic of the motion. Hit focuses on a fast, forceful force being impressed by contact, Kick focuses on the particular body part through which the force is transmitted. Essentially, events like these can be described by verbs at different levels of *granularity*.

A different scenario is where, for example, there are multiple events happening at the same time but some are *more salient* than others. The variation in the degree of saliency attributed to such events by different observers is often determined by the interest and motivation underlying the observation process itself. Thus language users would describe a situation by filtering out events deemed irrelevant because overshadowed by more relevant ones, either voluntarily or automatically.

For example, a scene may constitute of two vehicles approaching each other in the background while another person is punching another in the foreground. Even though the description of the scene as a co-occurrence of the verb Approach with the verb Hit is correct, most people would almost automatically remove the occurrence of Approach from their observation. Their eyes are so concentrated on the occurrence of Hit some may not even notice the other event.

A greater saliency of one event in respect to another is determined by a mixture of the event semantic characteristics and the context in which the action takes place. In fact, some events seem intrinsically more salient than others. For example Move is the simplest way one could describe an occurrence of motion. A verb specifying a trajectory (Fly), direction (Approach) or force (Fall) is almost guaranteed to be more salient than Move. The context surrounding the event participants completes the picture, and this can become complex very quickly. On a simple level, an action in the foreground of a scene is likely to be more salient than one in the background. But other elements come into play too. An action involving more participants is more likely to attract more attention, as is a participant wearing very bright clothes and a wig.

It seems relatively natural for humans to assess the saliency of events

when uttering descriptions of observations. This is particularly striking in our automated recognition system evaluation data, where there is a marked under-reporting of low-salience events (see Sec. 6.4). One possibility for an intelligent system to mimic a saliency assessment would be to consult a hierarchy of verbs akin to a saliency relation network, describing which verbs are overall more salient than others.. At the same time, the system should recognise contextual elements which impact on the saliency of a specific occurrence, such as position of objects. Still, it seems that, for an automatic recognition system, deciding saliency of event occurrences is not as easy a task as people's innate ability to do so.

3.4 Context

Although many words are vague, this does not usually cause problems for language users. In fact, judging an object as 'fast', two people as 'close' or an action as a 'snatch' rather than a 'take' seems a rather easy task for humans. Sometimes this is due to *partiality*, as people are not necessarily consistent in every judgement they make. At other times people ground judgements on *context*, which consists of relevant information about an observed situation and the observer's prior knowledge and experience. Context also plays a part in picking or disregarding certain parts of the meaning of words, and plays a role in establishing saliency as seen in the previous section. Additionally, while watching a particular scene, a human observers' mind experiences the construction of a narrative story, so that judgement may be oriented towards the confirmation of particular hypotheses on which such a narrative is based.

Earlier studies have attempted at modelling the semantics governing judgements and assertions within dialogues. In particular, Lewis concentrates on sequences of statements and judgements that contribute to building a *conversational score*, which assesses consistency and acceptability of dialogues and evolves according to presupposition and permissibility of statements [52, 71]. Presuppositions are elements of a conversation that participants take for granted. They dynamically evolve over time, as they can be created and destroyed over the course of the dialogue, and generally depend on prior utterances and presuppositions. They need not be explicit, as for example the statement "Even my grandmother could climb

that mountain" generates the presuppositions that "I have a grandmother" and "My grandmother is not particularly good at climbing mountains", and these remain valid until a participant argues against them. Lewis draws a comparison with the score governing the evolution of games, stating that the conversational score of a language game evolves in an almost rule-governed way, with the peculiarity that it also tends to "evolve in such a way as is required in order to make whatever occurs count as correct play".

A closely related concept is the *commitment slate* emerging from studies on mathematical models of dialogue by Hamblin [55], where dialogues, participants and utterances (called *locutions*) are formalised in a semantics aiming at estalibshing which dialogues are legal and which are not. The commitment slate is the set of locutions to which a participant is committed at a particular step of a conversation. The utterance of a locution by a participant is such that all other participants in the dialogue become immediately commited to it. Hamblin's semantics, for example, states that a dialogue is bad if someone pronounces a locution already present in a commitment slate, or that contradicts something in someone's commitment slate. The semantics also specifies conditions under which pronounced locutions may cause commitments to be retracted.

All the above suggests that the applicability of concepts in describing particular situations is determined both statically by established norms and dynamically by a particular communication context.

The potential for such context-awareness is built into the ontology and its application in the form of precisification thresholds: parameters determining the truth-value of a particular predicate according to different situations (see Sec. 4.5). This mechanism would make it possible to reproduce a form of partiality and context-dependency through the envisioned automatic inference of precisification thresholds. For example, the system should establish a different threshold for deciding whether a person is near a destination according to the mode of transport. The threshold value for a person that is driving a car will be different from the value for a person that is walking. Taking the context of a particular vague adjective, noun or verb can be difficult, as one needs to establish what the *relevant* context is [103, Ch. 3]. As seen in the example in Fig. 3.1(b), establishing whether an object is 'near' a place may involve examining the geographi-

cal features of the environment, of which there may be many and not all of them relevant.

A prerequisite for the above is to integrate the representation of context into the ontology. This could resemble a form of knowledge base of information guiding the judgements of vague concepts. This structure will have a static part, essentially constituted by statements describing properties of objects (e.g. the average speed of a walking person, or the average size of a ball) and a dynamic part, constituted by facts acquired by the *experience*. Within our domain, this experience would be built through the observation of scenes where the events of interest are taking place. The experience may be constituted by events that happened in the past, for example the fact that a certain person entered the scene in the foreground, or inferences involving the environment, for example presence of obstacles or routes faster than others. This approach is related to a sketch by Galton on an ontology of history and experience [49]. Galton suggests a dichotomic split between EXP and HIST, the former being the *experiential perspective* representing by the world as it is being observed/experienced, the latter being a form of *historical record* of events representing the past. The link between the two perspectives is that the experience consolidates into the past as time progresses.

Some kinds of contextual information have been mentioned in the overview of semantic characteristics in Sec. 3.2. The most relevant types of context are summarised below:

- *Temporal context*. The applicability of a concept may relate to some event or property that occurred in the past, or that is going to occur in the future. For example, establishing the starting instant of an occurrence of Leave involves establishing whether an object will 'not be near' the starting place at some instant in the future.

- *Spatial context*. Spatial characteristics of the environment influence whether a particular event occurs given a particular motion (see Sec. 3.2 and Fig. 3.2).

- *Action related context*. An event occurrence may occur if another event occurrence is occurring at the same time. A prime example is Chase, performed by a subject in the presence of another object performing Flee. An event occurrence may also be directed towards a goal, and

identifying progression towards the goal also depends on contextual elements (see effort space on pag.39).

- *Object context*. Properties of objects determine different applicability thresholds for vague predicates. In the example mentioned previously, inferring whether someone is moving near a place depends on whether someone is driving or walking.

A contribution towards selecting and analysing the most relevant contextual information given a certain concept is found in studies in the field of linguistics [107, 56, 57, 59, 105, 46].

3.5 Granularity

Granularity has to do with the level of detail in which one is looking at things. Earlier it has been mentioned that an event such as a person kicking a ball can be described at a coarse-grained level of detail as an occurrence of Hit, or at a fine-grained level as an occurrence of Kick.

Granularity also relates to the amount and quality of the information available to perform event recognition. As our system grounds the ontology of motion verbs on data resulting from processing video scenes, such data will not reproduce the details found in the real world. Referring to the example above, recognising an occurrence of Kick ideally requires knowledge about the precise positioning of a person. If the data is very coarse-grained, it most likely will not identify the position of this person's feet. Conversely, very fine-grained data is likely to identify this position precisely. In the former instance, recognising an occurrence of Kick is problematic and the system may recognise such action as an occurrence of the coarser event Hit. In the latter instance, recognising Kick should pose no particular problem for the inference system. In this sense, granularity is also related to uncertainty, discussed in the next section.

Another aspect of granularity refers to the way the system processes the available information. For example the automatic system, in order to infer an occurrence of the event Approach over a time interval, may process the data by sampling an object's position at specific time points within the interval. Figure 3.4 shows a few rather bizarre motion trajectories in which an object x may be approaching or moving away from y:

- In Fig. 3.4(a) x is approaching y with an oscillating motion. The distance between x and y decreases *overall* but does not decrease at each subsequent time instant, as certain parts of the motion actually increase this distance. If x's position is sampled over a sufficiently wide time interval, the system would infer that distance is decreasing hence x is approaching y over the whole motion interval. If the positions are sampled very closely, the system would detect a number of very short occurrences of Approach. In Sec. 6.2.4 a method to overcome this problem is proposed.

- In Fig. 3.4(b) x is initially approaching y then moving away from it. A fine-grained sampling would infer that Approach occurs for the first part of the motion, an inference probably disregarded with a coarser sampling. Although it is true that x is initially approaching y, an observer looking at the bigger picture would probably disregard this occurrence under saliency considerations.

- In Fig. 3.4(c) x is approaching y with an oscillating motion that, ultimately, brings x to be closer to y. However, if the system samples position on the points marked in green, it will not recognise that the distance between x and y is overall decreasing, hence will not recognise Approach

- Finally, in Fig. 3.4(d) x is performing a similar motion through which it is ultimately distancing itself from y. However, if the system samples the positions in green, it will recognise an occurrence of Approach. Such occurrence is not completely wrong as, in some ways, x does get closer to y for some time. However, the general motion pattern does not indicate a particulary strong will of x to get closer to y.

The examples in Fig. 3.4 may seem too extreme and not representative of the real world, and in a way this is true. However they demonstrate that the level of detail has an impact on the inference and the decisions a system or a person can make. Too much focus on fine details may cause one to misjudge the bigger picture. On the other hand, a coarser approach not paying attention to important details may misjudge appearances.

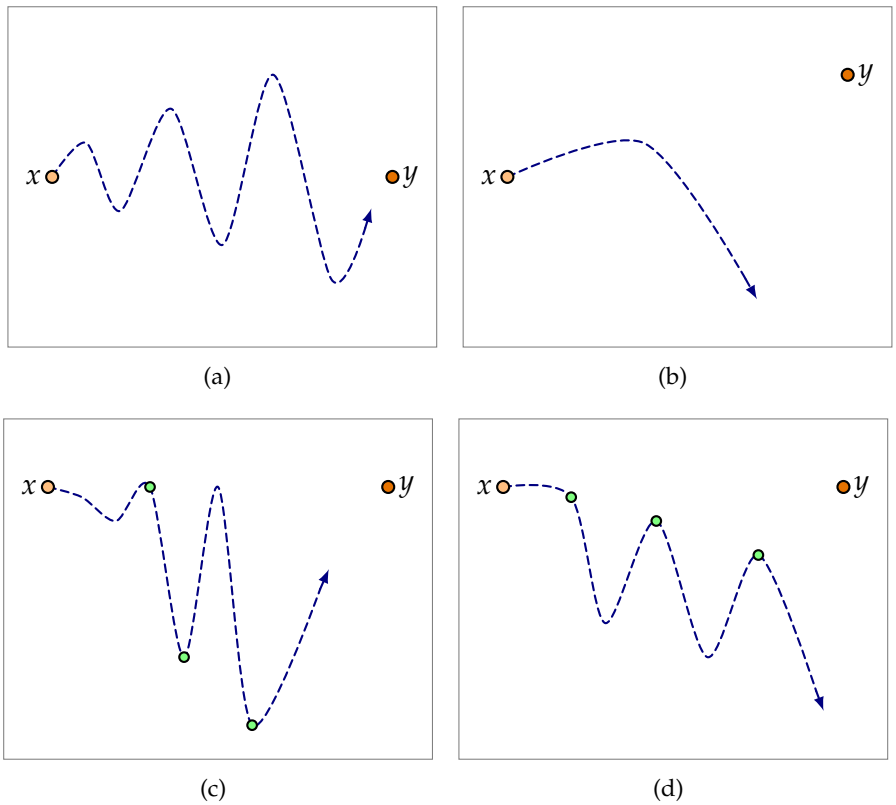


Figure 3.4: Verb Approach – 'Strange' approaches

3.6 Uncertainty

Some verbs refer to very specific and fine-grained properties of objects, characteristics of motion or the surrounding context which may not be at all available to an automatic system grounding the ontology on real data. Referring to the Hit and Kick example, if the position of a person's feet is not specified, recognising an occurrence of Kick will be a tricky task.

Data obtained from video processing carries a varying degree of uncertainty, particularly high if the data is produced automatically, as it is very unlikely that such a representation mirrors the amount and complexity of the information that a human eye may gather from watching the vignette.

Referring to the data available to our system at this stage, the main kinds of uncertainty are summarised below:

- *Errors and noise.* An object may be represented inaccurately in the data. For example, its position may be misplaced by the tracking

algorithm, and may disappear entirely if occluded by another object. Spurious tracks (i.e. detection of objects in positions where there is no actual object) are an undesirable but all too common feature of data resulting from tracking algorithms.

- *Missing objects*. Some objects in the data may not be represented at all. This often applies to static objects not playing an active part in the scene but still relevant as context. This is of course an issue for a verb whose formalisation involves these objects (e.g. an occurrence of Flee will be harder to recognise if the element representing the danger the person is fleeing from is not recognised, such as a fire).

- *Background and Context*. Scene background and other contextual elements, for example obstacles or environmental features determining constraints on the movement of objects, are often unreported in the data. On one hand, this simplifies the data allowing for faster processing. On the other hand evidence from contextual information is useful or even essential for the interpretation of some predicates.

- *Granularity*. As already mentioned, data may not show desired or relevant details for the recognition of a certain occurrence.

- *Spatial representation*. Different algorithms detect and represent objects according to different spatial models. A sophisticated algorithm may detect three-dimensional coordinates. Most traditional ones represent objects as two-dimensional cartesian coordinates. Inferring the fact that a movement is directed towards the third dimension is an issue in this latter case (e.g. a movement towards the background or the foreground for a frontal representation, or a movement towards the sky or the ground for an aerial representation).

Some of the issues listed above may be corrected by the inference system, and in Sec. 4.7 we explain how the ontology can be augmented with such capabilities. For example, it is imaginable that an approximate three-dimensional representation could be inferred from a two-dimensional one. In this case, information about the observer's position and perspective is fundamental, and it would allow the system to improve the recognition of verbs such 'approach' (see Sec. 5.2.1). Similarly, the details about the position of a person's limbs can be inferred by estimating where they are

most likely to be located given the person's position. Some other issues are unavoidable: a system cannot simply infer that there exists an object in the absence of any information or any background knowledge about it.

Chapter 4

Ontology of Motion Verbs

Our ontology builds upon *Event Calculus* [64, 95] and *Versatile Event Logic* (VEL) [18], formalisms designed to reason about actions and events within logic. Given an ordered set of time points $\mathcal{T} = (T, <)$, the most interesting feature of these calculi is the possibility to express that propositional expression p holds at a particular time point $t \in \mathcal{T}$, through the construct $\mathsf{HoldsAt}(p, t)$.

The purpose of our formalism is to describe real world situations, namely objects, their properties and event occurrences. However, the task of automated event detection in which this ontology will be employed presents a few peculiar aspects bearing an influence to some of the design choices outlined in the remainder of this chapter.

Firstly, a computer system can only operate on a representation of the real world and not on the real world itself. In some ways, such a representation constitutes the knowledge grounding the ontology at its lowest-level. This aspect is examined in more detail in Sec. 4.7.

Secondly, higher-level concepts, representing what can be understood about the world from the above knowledge, show different levels of complexity. This determined the structuring of the ontology in broadly three layers: a primitve layer, representing the grounding knowledge, a mid-level layer, concerned with objects' description, and a high-level layer, aimed at understanding complex situations such as processes and events. However, this distinction is not always clear-cut and some predicates may straddle across layers.

Some details of the formalism have been influenced and shaped by the

particular task of event recognition it is going to be applied to, as mentioned in Sec. 1.1, and partly also by the specific characteristics of the data grounding the ontology, described in detail in Sec. 6.1, especially regarding the spatio-temporal model and primitive properties of objects. However, the methodology and general principles allow for the generalisation of this approach to different domains.

For the convenience of the reader, an index detailing the concepts and symbols defined in the ontology is provided at the end of the volume.

The vocabulary of the logical language can be specified by the tuple:

$$\mathcal{V} = \langle \mathcal{T}, \mathcal{I}, \mathcal{S}_p, \mathcal{S}_r, \mathcal{O}, \mathcal{O}_t, \mathcal{PT}, \mathcal{F}, \mathcal{E}, \Sigma \rangle$$

where:

- $\mathcal{T}$ is the set of ordered time points (e.g. $\mathcal{T} = \{t_1, t_2 \ldots\}$);
- $\mathcal{I}$ is the set of time intervals (e.g. $i = [t_1, t_2]$);
- $\mathcal{S}_p$ is the set of spatial points;
- $\mathcal{S}_r$ is the set of spatial regions;
- $\mathcal{O}$ is the set of objects;
- $\mathcal{O}_t$ is the set of object types;
- $\mathcal{PT}$ is the set of *precisification thresholds*;
- $\mathcal{F}$ is a set of fluents;
- $\mathcal{E}$ and Σ are sets of *event-types* and *event-tokens*.

The formalism employs the connectives $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$ and the existential and universal quantifiers $\exists$ and $\forall$ with the semantics of classical first-order logic.

4.1 Temporal Model

The set $\mathcal{T}$ in the vocabulary representes time points or instants. Given that the ontology is going to be implemented and applied to event recognition

tasks from video, this chapter and the following assume that $\mathcal{T} = (T, <)$ is a finite and discrete set of time points ordered by function $<$.

The set $\mathcal{I}$ of time intervals contains sets of time points. In the notation, each set $I \in \mathcal{I}$ is represented as the closed interval $I = [t_1, t_2]$, where t_1 and t_2 are respectively the start and end point of I.

The following expressions allow for comparison and manipulation of time points and intervals:

- $t_1 = t_2$ if and only if t_1 and t_2 are the same time point.
- $t_1 < t_2$ if and only if time point t_1 precedes time point t_2 according to the ordering function in $\mathcal{T}$.
- $t_1 \leqslant t_2$ if and only if $t_1 = t_2$ or $t_1 < t_2$.
- $succ(t)$ is a function expressing the immediate successor of time point t (assuming a discrete model of time).
- $t \in I$ if and only if time point t belongs to interval I (for $t \in \mathcal{T}$ and $I \in \mathcal{I}$).
- $begin(I, t_s) \equiv t_s \in I \wedge \nexists t \in I[t < t_s]$, i.e. time point t_s is the starting instant of interval I, or alternatively $I = [t_s, t]$ for some $t \in \mathcal{T}$.
- $end(I, t_e) \equiv t_e \in I \wedge \forall t \in I[t \leqslant t_e]$, i.e. time point t_e is the ending instant of interval I, or alternatively $I = [t, t_e]$ for some $t \in \mathcal{T}$.
- $I_1 = I_2 \equiv \forall t[t \in I_1 \leftrightarrow t \in I_2]$, i.e. I_1 and I_2 are equal.
- $I_1 \subseteq I_2 \equiv \forall\, t[t \in I_1 \rightarrow t \in I_2]$, i.e. interval I_1 is a subset of interval I_2.
- $I_1 \subsetneq I_2 \equiv I_1 \subseteq I_2 \wedge \exists\, t[t \in I_2 \wedge t \notin I_1]$, i.e. I_1 is a proper subset of I_2.
- $dur(I) = \delta \leftrightarrow \delta = |I|$, i.e. δ is the duration of interval I, expressed as the number of time instants in I.
- $t_2 - t_1 = \delta \equiv dur([t_1, t_2]) = \delta$ (given $t_1 \leqslant t_2$), an alternative form to calculate the duration of the interval between time points t_1 and t_2.

4.2 Spatial Model

The sets $\mathcal{S}_p$ and $\mathcal{S}_r$ are respectively the sets of spatial points and spatial regions. These sets represent the spatial model of the ontology and determine how objects are located in space. In many ways, the exact nature and structure of this spatial representation is influenced by the application domain, as it may orient the ontology towards adopting a particular model of space, and by the data, as it determines how the ontology is grounded. There are several of these possible configurations: space could be represented with two or three dimensions, the set of spatial points could be discrete or dense and different kinds of spatial regions can be considered, such as lines, rectangles, polygons, volumes, multi-polygons, etc. Most aspects relating to the application, implementation and specific representation of space are examined in Chapter 6.

In Sec. 4.2.1 below, the domain-independent structure of the spatial model of our ontology is introduced by specifying general properties and primitives for sets $\mathcal{S}_p$ and $\mathcal{S}_r$. This structure bears no particular bias towards a specific spatial model that may be determined by the data grounding the ontology. Such a specification is sufficient for the introduction of spatial properties of objects in Sec. 4.6. In Sec. 4.2.2 and Sec. 4.2.3 two specifications of this spatial model are presented, respectively for a two- and a three-dimensional representation of points and regions. Although such a specification introduces a data-driven bias in the lower-level primitives of the ontology, the development of the theory of appearances overviewed in Sec. 4.7 would allow for a modular approach with a unified spatial model abstracted from these two specific representations. For instance, the three-dimensional model could be adopted as the model for sets $\mathcal{S}_p$ and $\mathcal{S}_r$ and, if not explicit in the data grounding the ontology, a three-dimensional representation could be inferred from a two-dimensional one by taking into account the observer's position.

4.2.1 Abstract spatial model

The set of spatial points $\mathcal{S}_p = \{p_1, p_2, \ldots\}$ contains all spatial points in the ontology. Characteristics of $\mathcal{S}_p$ such as ordering, dense or discrete, are specified by a particular concrete model such as the ones exemplified in the sub-sections to follow. The set of points $\mathcal{S}_p$ constitutes the basis for the

representation of objects positions in the rest of the ontology.

The set of spatial regions $\mathcal{S}_r = \{R_1, R_2\}$ is a set of sets of connected points in $\mathcal{S}_p$, i.e. $\mathcal{S}_r \subseteq 2^{\mathcal{S}_p}$, such that for each $R \in \mathcal{S}_r$ it holds that $\forall p_1, p_2 \in R\ [conn(p_1, p_2)]$ where *conn* is a primitive relation expressing that points p_1 and p_2 are spatially connected. As for $\mathcal{S}_p$, detailed characteristics of each region $R \in \mathcal{S}_r$ are specified by the particular concrete model adopted, such as whether R is a rectangle, polygon, volume etc. The set of regions $\mathcal{S}_r$ constitutes the basis for the representation of the space occupied by objects and their boundaries.

The following properties allow for comparison and manipulation of spatial points and regions:

- $p_1 = p_2$ (given $p_1, p_2 \in \mathcal{S}_p$) if and only if p_1 and p_2 represent the same spatial position, with the relation being an equivalence (reflexive, symmetric and transitive).
- $dist(p_1, p_2) = d$, where d represents the distance between points p_1 and p_2. The nature of this function is specified by the concrete spatial model.
- $p \in R$ (given $p \in \mathcal{S}_p$ and $R \in \mathcal{S}_r$) if and only if point p belongs to spatial region R.
- $R_1 = R_2 \equiv \forall p[p \in R_1 \leftrightarrow p \in R_2]$, i.e. spatial regions R_1 and R_2 are equal and represent the same set of points.
- $R_1 \subseteq R_2 \equiv \forall\, p[p \in R_1 \rightarrow p \in R_2]$, i.e. region R_1 is a subset of region R_2.
- $R_1 \subsetneq R_2 \equiv R_1 \subseteq R_2 \wedge \exists\, p[p \in R_2 \wedge p \notin R_1]$, i.e. region R_1 is a proper subset of region R_2.
- $R_1 \cap R_2 = R_i \leftrightarrow \forall p[p \in R_i \leftrightarrow p \in R_1 \wedge p \in R_2]$, i.e. region R_i is the intersection of regions R_1 and R_2.
- $R_1 \cup R_2 = R_i \leftrightarrow \forall p[p \in R_i \leftrightarrow p \in R_1 \vee p \in R_2]$, i.e. region R_i is the union of regions R_1 and R_2.
- $boundary(R) = P$ if and only if P is the set of points representing the boundary of region R.

- $interior(R) = P$ if and only if P is the set of points representing the interior region of R.

- The topological constraint that boundary and interior of a region are disjoint $\forall R \in \mathcal{S}_r\ [boundary(R) \cap interior(R) = \emptyset]$.

- The topological constraint that the union of boundary and interior of a region equal the whole region $\forall R \in \mathcal{S}_r\ [boundary(R) \cup interior(R) = R]$.

The advantage of introducing such abstract model is that, in most cases, the particular concrete model of space adopted in a particular domain only affects the definition of the primitive relations between points and regions listed above, whilst most other properties and relations in our ontology introduced in this and the following chapters may only refer to the abstract model.

In the following paragraphs, two possible concrete instances of the abstract model described so far are proposed for two- and three-dimensional representations.

4.2.2 Two-dimensional spatial representation

A two-dimensional, discrete and finite spatial model for the representation of points and regions can be constructed by specifying that the set of spatial points $\mathcal{S}_p$ is a set of ordered pairs of cartesian coordinates such that $\mathcal{S}_p = \{(x, y) \in X \times Y\}$ where X and Y are finite subsets of $\mathbb{N}$.

The set of spatial regions $\mathcal{S}_r$ would then be constituted by the set of sets of connected points in $\mathcal{S}_p$, which would result in a set of lines and areas (respectively one- and two-dimensional regions). For a set $\mathcal{S}_p \subset \mathbb{N} \times \mathbb{N}$ as defined above, the primitive connection relation $conn(p_1, p_2)$ between two points p_1 and p_2 can be defined as follows:

$$conn(p_1, p_2) \equiv \exists\, x_1, x_2 \in X, \exists\, y_1, y_2 \in Y$$
$$[p_1 = (x_1, y_1) \wedge p_2 = (x_2, y_2) \wedge (|x_2 - x_1| = 1 \vee |y_2 - y_1| = 1)]$$

Some of the primitive relations between points and regions can be defined according to this model, for example the identity and distance be-

tween points:

$$p_1 = p_2 \equiv$$
$$\exists\, x_1, x_2 \in X, \exists\, y_1, y_2 \in Y\, [p_1 = (x_1, y_1) \wedge p_2(x_2, y_2) \wedge x_1 = x_2 \wedge y_1 = y_2]$$
$$dist(p_1, p_2) = d \leftrightarrow$$
$$\exists\, x_1, x_2 \in X, \exists\, y_1, y_2 \in Y\, \left[d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}\right]$$

The remaining primitive spatial relations involving points and regions from Sec. 4.2.1 can be defined in a similar fashion by specifying particular constraints about the value of each point coordinates and the set of points that are part of a particular region.

A concrete spatial model where points are ordered, such as the one in this section, admits further primitives, for example the relative position between two points:

- $pos\_x(p) = x \leftrightarrow \exists\, x \in X, y \in Y, z \in Z\, [p = (x, y, z)]$
- $pos\_y(p) = y \leftrightarrow \exists\, x \in X, y \in Y, z \in Z\, [p = (x, y, z)]$
- $pos\_right(p_1, p_2) \leftrightarrow [pos\_x(p_1) > pos\_x(p_2)]$
- $pos\_left(p_1, p_2) \leftrightarrow [pos\_x(p_1) < pos\_x(p_2)]$
- $pos\_above(p_1, p_2) \leftrightarrow [pos\_y(p_1) > pos\_y(p_2)]$
- $pos\_below(p_1, p_2) \leftrightarrow [pos\_y(p_1) < pos\_y(p_2)]$

4.2.3 Three-dimensional spatial representation

A model for a three-dimensional representation of points and regions can be constructed by specifying that the set $\mathcal{S}_p$ is a set of ordered tuples of cartesian coordinates such that $\mathcal{S}_p = \{(x, y, z) \in X \times Y \times Z\}$ where, for example X, Y and Z are finite subsets of $\mathbb{N}$.

The set of spatial regions $\mathcal{S}_r$ would then be constituted by the set of sets of connected points in $\mathcal{S}_p$, which would result in a set of lines, areas and volumes (respectively one-, two- and three-dimensional regions). For a set $\mathcal{S}_p \subset \mathbb{N}^3$ as defined above, the primitive connection relation $conn(p_1, p_2)$

between two points p_1 and p_2 can be defined as follows:

$$\begin{aligned}
&conn(p_1, p_2) \equiv \\
&\quad \exists\, x_1, x_2 \in X, \exists\, y_1, y_2 \in Y, \exists\, z_1, z_2 \in Z \\
&\quad [p_1 = (x_1, y_1, z_1) \wedge p_2 = (x_2, y_2, z_2) \wedge \\
&\quad (|x_2 - x_1| = 1 \vee |y_2 - y_1| = 1 \vee |z_2 - z_1| = 1)]
\end{aligned}$$

The same primitive relations mentioned in the previous paragraphs can be defined according to this model:

$$\begin{aligned}
&p_1 = p_2 \equiv \exists\, x_1, x_2 \in X, \exists\, y_1, y_2 \in Y, \exists z_1, z_2 \in Z \\
&\quad [p_1 = (x_1, y_1, z_1) \wedge p_2(x_2, y_2, z_2) \wedge x_1 = x_2 \wedge y_1 = y_2 \wedge z_1 = z_2] \\
&dist(p_1, p_2) = d \leftrightarrow \exists\, x_1, x_2 \in X, \exists\, y_1, y_2 \in Y, \exists z_1, z_2 \in Z \\
&\quad \left[d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}\right]
\end{aligned}$$

As for the two-dimensional model in the previous section, the remaining primitive spatial relations for points and regions can be defined by specifying particular constraints about the value of each point coordinates and the set of points that are part of a particular region. Models with ordered points may extend the primitives specifying the relative position between two points by examining the z-coordinate:

- $pos\_z(p) = z \leftrightarrow \exists\, x \in X, y \in Y, z \in Z\ [p = (x, y, z)]$
- $pos\_front(p_1, p_2) \leftrightarrow [pos\_z(p_1) > pos\_z(p_2)]$
- $pos\_back(p_1, p_2) \leftrightarrow [pos\_z(p_1) < pos\_z(p_2)]$

4.3 Objects and Types

The sets $\mathcal{O}$ and $\mathcal{O}_t$ in the ontology vocabulary respectively represent objects and object types. Each object $o \in \mathcal{O}$ has a corresponding type in $\mathcal{O}_t$. For every object o, the predicate $\mathsf{type}(o, t)$ is defined, which holds if and only if object o is of type t, for example $\mathsf{type}(o, \mathsf{Person})$. It is assumed that object types are not time-dependent, i.e. the type of an object does not change over time.

Depending on the application and scope of the ontology, the set $\mathcal{O}_t$ can be populated with many object types that can be structured in a variety of hierarchies. The discussion to follow examines the kinds of objects most relevant to the domain described in this work and their structure. The classification is open to addition, modification and further specification as different contexts may present different requirements.

At the top of the hierarchy there are object types AbstractObject and ConcreteObject, forming a partition of Object. An abstract object is essentially an instance with a spatial characterisation but not corresponding to an agent or other entity existing in nature or, more specifically, in the scene under consideration. It is generally time-invariant, i.e. it does not change over time. Conversely, a concrete object is a physical entity relevant to the context and scene under consideration, and instances of ConcreteObject are generally time-dependent, as their properties change over time (for example their position, extension, size,...).

The type AbstractObject is the root of a sub-hierarchy of abstract objects which essentially characterise the different forms of spatial extension. This class is partitioned by subclasses Point, Line and Area. Each member of Point represents a point in space from the set of spatial points $\mathcal{S}_p$.

A member of Line is intended as a collection of objects of type Point and represents a one-dimensional spatial entity. This class is itself specialised by subclasses SimpleLine and CompositeLine, representing respectively segments and collections of consecutive segments.

A member of Area represents a closed spatial area, and it is specialised by classes Rectangle and Circle. This classification is not exhaustive and is open to further structuring to suit the need for the representation of further kinds of spatial areas. For example one may introduce other types of polygons or volumes.

The type ConcreteObject is the root of a sub-hierarchy of concrete objects which characterise the different kinds of agents and other physical entities relevant to the domain under consideration. In general, each member of ConcreteObject has spatial properties such as position, spatial extension, size, etc. Each subclass may specify additional properties relevant to its type. There are many instances of concrete objects in the physical world; this classification includes the ones most relevant to this application.

The type Person and PersonBodyPart are subclasses of ConcreteObject,

respectively representing people and people's body parts. PersonBodyPart may be further specialised by classes such as PersonHand and PersonFoot, with the relation between the two being that a member of PersonBodyPart is attached to a member of Person.

The type ConcreteObject is further specialised in classes Vehicle, Item and Space. These classes admit numerous possibilities for subclassing, for example one may distinguish between different vehicle types (car, bicycle, train...), item types (box, ball, chair,...) and spaces (building, field, car park,...). Specialisations may be useful to group items according to relevant properties, for example the accessibility of a particular space through a door or aperture, or the ways in which an item can be carried.

The following list summarises the hierarchy of object types described so far:

— AbstractObject
A spatial abstract object, generally time-invariant

  – Point
  A point in space, generally corresponding to a point $p \in \mathcal{S}_p$
  – Line
  A one-dimensional line, consisting of a set of Point instances
    - SimpleLine
    A segment
    - CompositeLine
    A collection of consecutive segments as a set of SimpleLine
  – Area
  A two-dimensional spatial area, consisting of a set Point instances, or delimited by a set of Line instances
    - Rectangle
    - Circle

— ConcreteObject
A physical entity. Most instances have properties such as position, extension and size that may change over time

  – Person
  Most instances have properties relating each person to its body parts

  - PersonBodyPart
    - PersonHand
    - PersonFoot
  - Vehicle
  - Item
  - Space
    Most instances have properties characterising the type of space (open, closed) and relations of accessibility

Specific properties of the types of objects listed in this section are discussed in Section 4.6. However, it has been mentioned that most instances of ConcreteObect are characterised by properties that vary over time. Before introducing any of these specific properties, the ontology requires a device allowing one to state that a particular property or predicate holds at a particular time or over a particular interval. This is explained in the next section.

4.4 Fluents

The main aim of the formalism introduced in this chapter is to reason about objects and spatial relations that change over time. Most properties and relations between objects to be introduced in the following sections are time-indexed, and they only hold at particular time instants or over particular intervals.

In order to express this form of time-indexing, the construct HoldsAt is introduced, derived from *Event Calculus* [64, 95] and *Versatile Event Logic* (VEL) [18]. Given the ordered set of time points $\mathcal{T}$, an instant $t \in \mathcal{T}$ and a propositional expression p, the predicate $\mathsf{HoldsAt}(p, t)$ expresses that *proposition p holds at time t*.

In vocabulary $\mathcal{V}$ there is a distinction between two types of time-dependent formal expressions: propositional expressions whose validity can be stated over time (set of *fluents* $\mathcal{F}$) and expressions referring to temporal entitites occurring over some interval (set of *event-types* and *event-tokens* $\mathcal{E}$ and Σ, discussed in section 4.8).

A fluent's truth-value may be established at single time points. Fluents describe either a state that may hold or not hold, or a process that

may be active or inactive at each time point. Given fluent f and notation $\mathsf{HoldsAt}(f,t)$, it is possible to define $\mathsf{HoldsOver}(f,i)$ to express the validity of f over the interval $i = [t_s, t_e] \in \mathcal{I}$:

$$\begin{aligned} &\mathsf{HoldsOver}(f,[t_s,t_e]) \equiv \\ &\quad \forall\, t\big[(t_s \leq t \leq t_e) \rightarrow \mathsf{HoldsAt}(f,t)\big] \end{aligned} \tag{4.1}$$

If $\mathsf{HoldsOver}(f,[t_s,t_e])$ is true for some t_s, t_e, from the definition above it follows that $\mathsf{HoldsOver}(f,[t_1,t_2])$ is also true, for every $[t_1,t_2] \subseteq [t_s,t_e]$.

A predicate holding only on the largest interval is $\mathsf{HoldsOn}(f,i)$, which is true if and only if i is the greatest continuous temporal interval over which f is true, i.e. there does not exist $i' \supset i$ such that $\mathsf{HoldsOn}(f,i')$:

$$\begin{aligned} &\mathsf{HoldsOn}(f,[t_s,t_e]) \equiv \mathsf{HoldsOver}(f,[t_s,t_e]) \ \wedge \\ &\quad \wedge\, \exists\, t_1\Big[t_1 < t_s \wedge \forall\, t\big[t_1 \leqslant t < t_s \rightarrow \neg\mathsf{HoldsAt}(f,t)\big]\Big] \wedge \\ &\quad \wedge\, \exists\, t_2\Big[t_e < t_2 \wedge \forall\, t\big[t_e < t \leqslant t_2 \rightarrow \neg\mathsf{HoldsAt}(f,t)\big]\Big] \end{aligned} \tag{4.2}$$

For clarity of notation, throughout the rest of this chapter and the following, fluents are expressed with lowercase letters (such as position), with capitalised notation reserved for events (such as Approach, see also Sec. 4.8). Additionally, in order to reduce and simplify the structure of definitions, the notations $\mathsf{HoldsAt}(p,t)$ and $\mathsf{HoldsOn}(p,[t_s,t_e])$ have been shortened respectively to $p_{@t}$ and $p_{@[t_s,t_e]}$.

A time-invariant propositional expression p holds or does not hold regardless of a specific time interval, for example property $\mathsf{type}(o,t)$ expressing that object o is of type t. These propositions could still be time-indexed, with the assumption that p is true if and only if $\forall\, t \in \mathcal{T}\ p_{@t}$.

4.5 Precisifications

In addition to time-dependent propositions, most concepts to be introduced in the ontology are vague. Their applicability boundaries are not crisp, as their extension and crispness depends on particular interpretations or the context in which the concepts appear.

A formal method to establish whether a vague concept holds can be obtained through the ideas in Supervaluation Semantics [47, 61], introduced

in greater detail in Sec. 2.4. In this theory, a formula may admit multiple models, each obtainable via an assignment of referents to terms and truth-values to predicates. Such an assignment is called a *precisification*, and allows to obtain a precise interpretation of a vague term. This approach preserves classical logic inference rules, hence it is preferred over multi-valued logics such as Fuzzy Logic [116] for the task of ontology reasoning presented here.

Supervaluation Semantics and the epistemic stance lead to Standpoint semantics [16], an elaboration of supervaluation semantics where the precisification is explicitly embedded in the language syntax. Specifically, the precise criteria governing the extension of a concept's applicability boundary are modelled in terms of *applicability thresholds* for one or more observable properties.

The set of precisification thresholds $\mathcal{PT}$ is constituted of ordered pairs (t, V_t), where V_t is the range of admissible values for threshold t. A precisification P is an assignment of values to precisification thresholds, i.e. $P \subseteq \{(t, v_t) \,|\, (t, V_t) \in \mathcal{PT} \wedge v_t \in V_t\}$ (assuming that, for any P, $\nexists\, (t, v_1), (t, v_2) \in P \wedge v_1 \neq v_2$).

Given a generic predicate p, a precisification P is made explicit in the language syntax as a *parameter* enclosed in square brackets as in $\mathsf{p}[P]$. Definitions of vague predicates parameterised in this manner can explicitly refer to the precisification thresholds embedded in P, therefore establishing a crisp applicability boundary for p. This is demonstrated in the following example. The vague concept *point* p_1 *is near* p_2 can be made precise by specifying a threshold on an observable property, such as the linear distance between points p_1 and p_2. The fluent near can be defined by parameterising its definition with precisification P containing threshold $(minNear, n)$, with $(minNear, V) \in \mathcal{PT}$ and $n \in V$:

$$\begin{aligned} &\mathsf{near}[P](p_1, p_2)_{@t} \equiv \\ &\quad \exists (minNear, \delta) \in P \; \exists d \; [dist(p_1, p_2)_{@t} = d \wedge d < \delta] \end{aligned} \tag{4.3}$$

The simple definition above states that property $near(p_1, p_2)$ holds between spatial points p_1 and p_2 if their distance is smaller than the *minNear* threshold specified by precisification P.

4.6 Object Properties

Section 4.4 introduced the construct to express that a proposition holds at a particular time or over a particular interval, and Section 4.5 introduced the possiblity to specify threshold parameters in order to establish whether a vague predicate holds. It is now possible to introduce several properties and relations between objects that form the basis for building the verb models extensively described in Chapter 5.

The organisation of most concepts listed in this section has partly been shaped according to the specific application domain of recognising occurrences of motion verbs. Therefore, the formalisation of verb models, the methodologies followed in the implementation stage and the data available for testing and evaluation (see Chapter 6) had an influence on the ontology structure, and this is particularly true for primitive concepts, the ones most closely related to the implementation and data. However, efforts have been made to preserve generality so that the ontology and verb models can be extended and/or adapted to different spatio-temporal domains and datasets.

The set of concepts in the ontology is structured in three layers:

- The lowest level, or *primitives* layer, is constituted by simple properties of objects which express their most essential spatio-temporal nature, and bear an intimate connection to the nature of the data available to ground the ontology. In fact, the initial grounding of the ontology happens on this very layer. A prime example is the position of objects.

- The middle level of the ontology is constituted by predicates expressing properties and relations of objects that can be inferred and abstracted from primitives. Within this layer, there is a fluid transition from a lower sub-level expressing mostly quantitative concepts (such as size, extension or distance between objects) and a higher sub-level expressing mostly qualitative concepts often inferred through the quantitave ones (for example proximity or topological relations).

- The highest level of the ontology is constituted by fluents and events describing objects interactions and behaviour, from simple events

such as Move to complex ones such as Exchange. Most of these are introduced and modelled in Chapter 5.

4.6.1 Primitives

As outlined above, the primitives layer is concerned with the most basic properties of objects of type AbstractObject, and properties of instances of ConcreteObject that stem from the data grounding the ontology. For clarity of notation, primitive predicates are prefixed with p_.

Concrete objects

The most prominent primitive properties of concrete objects are position and extension, as this is the most likely way that objects are represented in the data grounding the ontology. Given object o instance of ConcreteObject, the following primitives are defined:

- $\mathsf{p\_position}(o, p)_{@t}$, with $\mathsf{type}(p, \mathsf{Point})$. This asserts that the position of object o is point p at time t.
- $\mathsf{p\_extension}(o, a)_{@t}$, with $\mathsf{type}(a, \mathsf{Area})$. This asserts that the spatial extension of object o is area a at time t.

The idea is that the ontology will be grounded by populating the set of instances of Point and Area, each of which is then linked to the objects, whose position or extension is represented at particular instants by a set of true time-indexed p_position or p_extension fluents.

It is perfectly plausible that only one of these fluents may be grounded for a particular domain. For example, all objects may be represented by points in space, in which case no object is such that $\mathsf{p\_extension}(o, a)_{@t}$ is true for any o, a or t. Conversely, other application domains, such as the one this work is concerned with, see objects represented by their extension, and there is no predicate $\mathsf{p\_position}(o, p)_{@t}$ for any o, a or t.

Subclasses of ConcreteObject can admit additional primitives, specifying characteristics specific to the subclass. For example, given an object o with $\mathsf{type}(o, \mathsf{Person})$, the following primitives associate o with the instances of its body parts:

- $\mathsf{p\_hands}(o, (h_l, h_r))$, with h_l and h_r instances of PersonHand and representing respectively person o's left and right hand.

- p_feet$(o, (f_l, f_r))$, with f_l and f_r instances of PersonFoot and representing respectively person o's left and right foot.

Abstract Objects

Despite their name, the primitives for instances of AbstractObject can also be slightly influenced by the data grounding the ontology. Abstract objects and their primitives connect spatial entities to the spatial model represented by sets $\mathcal{S}_p$ and $\mathcal{S}_r$ in the ontology vocabulary. The structure of these sets is not necessarily identical to the concrete spatial model emerging from a particular real world situation, but it is likely to be very similar.

Given object o of type Point, the following primitive is defined:

- p_point(o, p), which associates point o with spatial point $p \in \mathcal{S}_p$.

Given object o of type Area or Line, the following primitive is defined:

- p_area(o, A), which associates area or line o with region $R \in \mathcal{S}_r$.

Certain subclasses may be characterised by different primitives. For example it is imaginable that if the grounding data contains sets of two-dimensional rectangles, instances o of type Rectangle may define some or all the following primitives:

- p_topleft(o, p), where p is the top left corner of rectangle o.
- p_width(o, w), where w is the width of rectangle o.
- p_height(o, h), where h is the height of rectangle o.

Other domains may define a different set. The above set is not coincidental as it bears a close relation to the spatial model arising from the data grounding the ontology implementation, discussed in Chapter 6.

4.6.2 Middle layer

Within the middle layer of the ontology, one can define predicates expressing properties and relations of objects relevant to a particular domain. In this section the ones relevant to the application and to the verbs modelled in the next chapter are introduced.

The lower level of this layer is mostly populated by *quantitative* relations expressing a precise measurement or property holding at a particular time for a particular object (e.g. position as a point, speed as a value, euclidean distance...). The higher level predicates are an abstraction with a more *qualitative* nature and with a certain degree of vagueness (e.g. speed as slow or fast, relative orientation,...).

In the predicates below, the notation $\mathsf{property}(o_1 \in \mathsf{Class}, o_2 \in \mathsf{Class})$ signifies that relation property has two participants which are both instances of Class. The purpose of this notation is to have the participants' type stand out. The logical formula corresponding to this notation is $\mathsf{property}(o_1, o_2) \equiv \mathsf{type}(o_1, \mathsf{Class}) \wedge \mathsf{type}(o_2, \mathsf{Class}) \wedge \ldots$.
The set $\mathbb{V}$ denotes a set of values; in some applications this could correspond to the actual sets $\mathbb{N}$, $\mathbb{R}$ or a subset of these. The notation $\mathbb{V}$ abstracts from the specific one being employed in a particular domain.

Abstract Objects

For instances of Point the following predicate can be defined:

- $\mathsf{distance}((o_1 \in \mathsf{Point}, o_2 \in \mathsf{Point}), d \in \mathbb{V})$.
  This relation holds if and only if d is the distance between points o_1 and o_2. Different metrics can be used. In Sec. 4.2 a function expressing distance between points in $\mathcal{S}_p$ has been listed, hence this relation could be defined according to function *dist*:

$$\begin{aligned}&\mathsf{distance}((o_1, o_2), d) \equiv \\ &\quad \mathsf{type}(o_1, \mathsf{Point}) \wedge \mathsf{type}(o_2, \mathsf{Point}) \wedge \\ &\quad \exists\, p_1, p_2 \in \mathcal{S}_p[\mathsf{p\_point}(o_1, p_1) \wedge \mathsf{p\_point}(o_2, p_2) \wedge \mathit{dist}(p_1, p_2) = d]\end{aligned} \tag{4.4}$$

- $o_1 = o_2$, with $o_1, o_2 \in \mathsf{Point}$
  This equality relation holds if and only if o_1 and o_2 are the same point. It is generally defined by referring to the equality relation of the spatial model on $\mathcal{S}_p$ (see Sec. 4.2):

$$\begin{aligned}&o_1 = o_2 \equiv \\ &\quad \exists\, p_1, p_2 \in \mathcal{S}_p[\mathsf{p\_point}(o_1, p_1) \wedge \mathsf{p\_point}(o_2, p_2) \wedge p_1 = p_2]\end{aligned} \tag{4.5}$$

- $o_1 \neq o_2$, with $o_1, o_2 \in \mathsf{Point}$
  This equality relation holds if point o_1 is different from point o_2. This can either be defined as above (by referring to the equality relation of the spatial model) or by stating that their distance is greater than zero:

$$o_1 \neq o_2 \equiv \exists\, p_1, p_2 \in \mathcal{S}_p[\mathsf{p\_point}(o_1, p_1) \wedge \mathsf{p\_point}(o_2, p_2) \wedge p_1 \neq p_2] \quad (4.6)$$

$$o_1 \neq o_2 \equiv \mathsf{distance}((o_1, o_2), d) \wedge d > 0 \quad (4.7)$$

- $\mathsf{position}(p_1 \in \mathsf{Point}, p_2 \in \mathsf{Point})$
  This is a trivial reflexive relation that expresses the position of a point p_1 as a point p_2 equal to itself. It is useful for abstracting further properties on the general class AbstractObject:

$$\mathsf{position}(p_1, p_2) \equiv p_1 = p_2$$

Relevant properties for instances of Area are the following:

- $\mathsf{centroid}(a \in \mathsf{Area}, p \in \mathsf{Point})$
  This relation holds if point p is the centroid of area a. This is a rather low-level property useful for determining the position of an area. It is also heavily dependent on the specific type of area considered, in fact most sub-classes implement the property with a particular definition (e.g. for a Circle point p is simply its center, for a Rectangle point p is the intersection of the two diagonals, etc.). An implementation is described in Sec. 6.2.1.

- $\mathsf{position}(a \in \mathsf{Area}, p \in \mathsf{Point})$
  This relation holds if point p expresses the position of area a. A trivial definition would define $\mathsf{position}(a, p) \equiv \mathsf{centroid}(a, p)$. However, other implementations may abstract from the notion of centroid and take other aspects of a into consideration.

- $\mathsf{distance}((a_1 \in \mathsf{Area}, a_2 \in \mathsf{Area}), d \in \mathbb{V})$
  This relation holds if d is the distance between areas a_1 and a_2. There are many definitions specifying how to interpret the concept of distance between two areas. Trivially, one can refer to their position and

the distance between the points:

$$\begin{aligned}&\mathsf{distance}((a_1, a_2), d) \equiv \\ &\quad \exists\, p_1, p_2[\mathsf{type}(p_1, \mathsf{Point}) \wedge \mathsf{type}(p_2, \mathsf{Point}) \wedge \\ &\quad \mathsf{position}(a_1, p_1) \wedge \mathsf{position}(a_2, p_2) \wedge \mathsf{distance}((p_1, p_2), d)]\end{aligned} \tag{4.8}$$

The same properties could be defined for objects of type Line.

It is possible to abstract general properties applicable to instances of abstract objects:

- $\mathsf{relPosition}((o_1 \in \mathsf{AbstractObject}, o_2 \in \mathsf{AbstractObject}), v \in relPos)$
  where $relPos = \{\text{left}, \text{right}, \text{above}, \text{below}, \text{front}, \text{back}\}$.
  Holds if the relative position v of o_1 in respect to o_2 is among the ones in *relPos*. This is a qualitative property dependent on the spatial model, and may be prone to ambiguity in particular situations. It is possible to define it by referring to the example spatial properties in Sec. 4.2.2 and Sec. 4.2.3 (the front and back relative positions are only available within a three-dimensional model):

  $$\begin{aligned}&\mathsf{relPosition}((o_1, o_2), \text{left}) \equiv \\ &\quad \exists\, p_1, p_2 \in \mathsf{Point}, sp_1, sp_2 \in \mathcal{S}_p[\mathsf{position}(o_1, p_1) \wedge \mathsf{position}(o_2, p_2) \wedge \\ &\quad \mathsf{p\_point}(o_1, sp_1) \wedge \mathsf{p\_point}(o_2, sp_2) \wedge pos\_left(sp_1, sp_2)]\end{aligned} \tag{4.9}$$

  The relations for the remaining relative positionings can be defined in a similar way.

- $\mathsf{interior}(o \in \mathsf{AbstractObject}, a_i \in \mathsf{Area})$
  This relation holds if a_i is the interior area of object o, i.e. a_i contains all the points in o minus the points on o's boundary. In the following definition o is assumed to be an instance of Area as points and lines do not have interior points (this could be changed under different assumptions and spatial models):

  $$\begin{aligned}&\mathsf{interior}(o, a_i) \equiv \mathsf{type}(o, \mathsf{Area}) \wedge \\ &\quad \exists\, R \in \mathcal{S}_r[\mathsf{p\_area}(o, R) \wedge interior(R) = R_i \wedge \mathsf{p\_area}(a_i, R_i)]\end{aligned} \tag{4.10}$$

  The definition recalls the spatial area function *interior* (see Sec. 4.2.1).

- $\mathsf{boundary}(o \in \mathsf{AbstractObect}, l \in \mathsf{Line})$
  This relation holds if line l represents the line formed by the points on the boundary of object o:

$$\begin{aligned}&\mathsf{boundary}(o,l) \equiv \exists\, R_l \in \mathcal{S}_r\, \mathsf{p\_area}(l, R_l) \wedge \\ &\quad [\quad [\mathsf{type}(o, \mathsf{Point}) \wedge \exists\, p \in \mathcal{S}_p[\mathsf{p\_point}(o,p) \wedge R_l = \{p\}]] \vee \\ &\qquad [\mathsf{type}(o, \mathsf{Line}) \wedge \exists\, R_o \in \mathcal{S}_r[\mathsf{p\_area}(o, R_o) \wedge R_o = R_l]] \vee \\ &\qquad [\mathsf{type}(o, \mathsf{Area}) \wedge \exists\, R_o \in \mathcal{S}_r[\mathsf{p\_area}(o, R_o) \wedge \mathit{boundary}(R_o) = R_l]]\ ]\end{aligned} \tag{4.11}$$

  The definition recalls the spatial area function *boundary* (Sec. 4.2.1). From the above it follows that the boundary of a point is a line constituted by a single point.

- $\mathsf{intersection}((a_1 \in \mathsf{Area}, a_2 \in \mathsf{Area}), a_i \in \mathsf{Area})$
  This relation holds if a_i is the non empty intersection of a_1 and a_2 (can be generalised by defining the relation intersection between different objects, for example an area and a line, or two lines):

$$\begin{aligned}&\mathsf{intersection}((a_1, a_2), a_i) \equiv \\ &\quad \mathsf{type}(a_1, \mathsf{Area}) \wedge \mathsf{type}(a_2, \mathsf{Area}) \wedge \mathsf{type}(a_i, \mathsf{Area}) \wedge \\ &\quad \exists R_1, R_2, R_I \in \mathcal{S}_r[\mathsf{p\_area}(a_1, R_1) \wedge \mathsf{p\_area}(a_2, R_2) \wedge \mathsf{p\_area}(a_i, R_i) \wedge \\ &\quad R_1 \cap R_2 = R_i \wedge R_i \neq \emptyset]\end{aligned} \tag{4.12}$$

- $\mathsf{pointInArea}(p \in \mathsf{Point}, a \in \mathsf{Area})$
  This relation holds if point p is part of area a:

$$\begin{aligned}&\mathsf{pointInArea}(p, a) \equiv \\ &\quad \mathsf{type}(p, \mathsf{Point}) \wedge \mathsf{type}(a, \mathsf{Area}) \wedge \\ &\quad \exists\, p_p \in \mathcal{S}_p, R \in \mathcal{S}_r[\mathsf{p\_point}(p, p_p) \wedge \mathsf{p\_area}(a, R) \wedge p_p \in R]\end{aligned} \tag{4.13}$$

- $\mathsf{pointInsideArea}(p \in \mathsf{Point}, a \in \mathsf{Area})$

This relation holds if point p is part of the interior of area a:

$$\begin{aligned}\mathsf{pointInsideArea}(p,a) \equiv \mathsf{type}(p,\mathsf{Point}) \wedge \mathsf{type}(a,\mathsf{Area}) \wedge \\ \exists\, a_i[\mathsf{interior}(a,a_i) \wedge \mathsf{pointInArea}(p,a_i)]\end{aligned} \tag{4.14}$$

Given the low level properties above, higher-level properties involving abstract objects can be defined. For example, the RCC calculus topological relations, illustrated in Fig. 2.2:

- $\mathsf{rcc\_DC}(a_1 \in \mathsf{Area}, a_2 \in \mathsf{Area}) \equiv$
  $\nexists\, a_i \in \mathsf{Area}[\mathsf{intersection}((a_1,a_2),a_i)]$ (4.15)

- $\mathsf{rcc\_EC}(a_1 \in \mathsf{Area}, a_2 \in \mathsf{Area}) \equiv$
  $\exists\, l_1, l_2 \in \mathsf{Line}[\mathsf{boundary}(a_1,l_1) \wedge \mathsf{boundary}(a_2,l_2) \wedge$
  $\exists\, l_i \in \mathsf{Line}[\mathsf{intersection}((l_1,l_2),l_i)]] \wedge$
  $\exists\, i_1, i_2 \in \mathsf{Area}[\mathsf{interior}(a_1,i_1) \wedge \mathsf{interior}(a_2,i_2) \wedge$
  $\nexists\, i_i \in \mathsf{Area}[\mathsf{intersection}((i_1,i_2)i_i)]]$ (4.16)

- $\mathsf{rcc\_PO}(a_1 \in \mathsf{Area}, a_2 \in \mathsf{Area}) \equiv$
  $\exists\, i_1, i_2, i_i \in \mathsf{Area}[\mathsf{interior}(a_1,i_1) \wedge \mathsf{interior}(a_2,i_2) \wedge \mathsf{intersection}((i_1,i_2)i_i)] \wedge$
  $\exists\, p_1, p_2 \in \mathsf{Point}[\mathsf{pointInArea}(p_1,a_1) \wedge \neg\mathsf{pointInArea}(p_1,a_2) \wedge$
  $\mathsf{pointInArea}(p_2,a_2) \wedge \neg\mathsf{pointInArea}(p_2,a_1)]$ (4.17)

- $\mathsf{rcc\_TPP}(a_1 \in \mathsf{Area}, a_2 \in \mathsf{Area}) \equiv$
  $\forall\, p \in \mathsf{Point}[\mathsf{pointInArea}(p,a_1) \rightarrow \mathsf{pointInArea}(p,a_2)] \wedge$
  $\exists\, p \in \mathsf{Point}[\mathsf{pointInArea}(p,a_2) \wedge \neg\mathsf{pointInArea}(p,a_1)] \wedge$
  $\exists\, l_1, l_2 \in \mathsf{Line}[\mathsf{boundary}(a_1,l_1) \wedge \mathsf{boundary}(a_2,l_2) \wedge$
  $\exists\, l_i \in \mathsf{Line}[\mathsf{intersection}((l_1,l_2),l_i)]]$ (4.18)

- $\mathsf{rcc\_NTPP}(a_1 \in \mathsf{Area}, a_2 \in \mathsf{Area}) \equiv$
  $\forall\, p \in \mathsf{Point}[\mathsf{pointInArea}(p,a_1) \rightarrow \mathsf{pointInArea}(p,a_2)] \wedge$
  $\exists\, p \in \mathsf{Point}[\mathsf{pointInArea}(p,a_2) \wedge \neg\mathsf{pointInArea}(p,a_1)] \wedge$
  $\exists\, l_1, l_2 \in \mathsf{Line}[\mathsf{boundary}(a_1,l_1) \wedge \mathsf{boundary}(a_2,l_2) \wedge$
  $\nexists\, l_i \in \mathsf{Line}[\mathsf{intersection}((l_1,l_2),l_i)]]$ (4.19)

- $\mathsf{rcc\_EQ}(a_1 \in \mathsf{Area}, a_2 \in \mathsf{Area}) \equiv$
  $\forall\, p \in \mathsf{Point}[\mathsf{pointInArea}(p,a_1) \leftrightarrow \mathsf{pointInArea}(p,a_2)]$ (4.20)

The above set can be augmented with the inverse relations $\mathsf{rcc\_TPP}^{-1}$ and $\mathsf{rcc\_NTPP}^{-1}$. It has to be noted that the relation intersection needs a further

refinement in order to address situations in which the intersection between lines representing the boundaries of two regions results in a set of isolated points rather than a contiguous line. Additionally, particular types of Area may allow for alternative definitions; for example, the equality relation rcc_EC between two rectangles may be inferred through the comparison of their defining characteristics (for instance corner, width and height).

Concrete Objects

Concrete objects have properties that characterise their essential spatial characteristics, such as position, extension or size, and more abstract properties such as the accessibility of a space or whether an object can be moved. Most of these properties hold according to a specific time instant, and vague concepts are precisified using a precisification containing thresholds that allow for a precise interpretation (see Sec. 4.5).

In general, for most instances of ConcreteObject it is possible to define the following properties:

- $\mathsf{position}(o \in \mathsf{ConcreteObject}, p \in \mathsf{Point})_{@t}$
  Holds if position of object o is represented by point p at time t. Primitive properties establishing the spatial location of object o are either p_position or p_extension, depending on the manner the ontology is grounded (see Sec. 4.6.1):

$$\mathsf{position}(o, p)_{@t} \equiv \mathsf{p\_position}(o, p)_{@t} \vee \exists\, a \in \mathsf{Area}[\mathsf{p\_extension}(o, a)_{@t} \wedge \mathsf{position}(a, p)] \tag{4.21}$$

- $\mathsf{extension}(o \in \mathsf{ConcreteObject}, a \in \mathsf{Area})_{@t}$
  Holds if extension of object o corresponds to area a at time t. For this property to hold, the primitive p_extension for object o has to be grounded in the ontology.

$$\mathsf{extension}(o, a)_{@t} \equiv \mathsf{p\_extension}(o, a)_{@t} \tag{4.22}$$

- $\mathsf{distance}((o_1 \in \mathsf{ConcreteObject}, o_2 \in \mathsf{ConcreteObject}), d \in \mathbb{V})_{@t}$
  Holds if distance between objects o_1 and o_2 is d at time t

$$\begin{aligned}&\mathsf{distance}((o_1, o_2), d)_{@t} \equiv \\ &\quad \exists\, p_1, p_2 \in \mathsf{Point}[\mathsf{position}(o_1, p_1)_{@t} \wedge \mathsf{position}(o_2, p_2)_{@t} \wedge \\ &\quad \mathsf{distance}((p_1, p_2), d)]\end{aligned} \tag{4.23}$$

  The above definition simply recalls the distance relation between points. However, it is imaginable that different metrics are appropriate given specific domains or particular kind of objets. For example, one may consider the distances between the objects' boundaries or more complex calculations.

- $\mathsf{samePosition}(o_1 \in \mathsf{ConcreteObject}, o_2 \in \mathsf{ConcreteObject})_{@t}$
  Holds if concrete objects o_1 and o_2 are at the same position at time t:

$$\begin{aligned}&\mathsf{samePosition}(o_1, o_2)_{@t} \equiv \\ &\quad \exists\, p_1, p_2 \in \mathsf{Point}[\mathsf{position}(o_1, p_1)_{@t} \wedge \mathsf{position}(o_2, p_2)_{@t} \wedge p_1 = p_2]\end{aligned} \tag{4.24}$$

- $\mathsf{nearPosition}[P](o_1 \in \mathsf{ConcreteObject}, o_2 \in \mathsf{ConcreteObject})_{@t}$
  Holds if position of object o_1 is near position of object o_2 at time t. It is a vague concept, hence the parameterisation with precisification P that specifies a suitable threshold for its precise interpretation. The concept nearPosition can be defined by taking into account different characteristics of the objects and the environment in which they are located. The following is a simple definition that holds if their distance at time t is smaller than threshold T_{nearPos} in P:

$$\begin{aligned}&\mathsf{nearPosition}[P](o_1, o_2)_{@t} \equiv \\ &\quad \exists\, (T_{\text{nearPos}}, t_n) \in P[\mathsf{distance}((o_1, o_2), d)_{@t} \wedge d < t_n]\end{aligned} \tag{4.25}$$

- $\mathsf{relPosition}((o_1 \in \mathsf{ConcreteObject}, o_2 \in \mathsf{ConcreteObject}), v \in \mathit{relPos})_{@t}$
  where $\mathit{relPos} = \{\text{left}, \text{right}, \text{above}, \text{below}, \text{front}, \text{back}\}$
  Holds if position of object o_1 in respect to object o_2 at time t is among

the ones listed in *relPos*. A simple definition would just refer to the relative position of the points corresponding to the position of o_1 and o_2 (eq. 4.9):

$$\begin{aligned}&\mathsf{relPosition}((o_1, o_2), v)_{@t} \equiv \\ &\quad \exists p_1, p_2 \in \mathsf{Point}\,[\mathsf{position}(o_1, p_1)_{@t} \wedge \mathsf{position}(o_2, p_2)_{@t} \wedge \\ &\quad \mathsf{relPosition}((p_1, p_2), v)]\end{aligned} \tag{4.26}$$

- $\mathsf{onGround}(o \in \mathsf{ConcreteObject})_{@t}$
  Holds if object o is positioned on the ground at time t. The position of the ground could be established as a primitive when grounding the ontology, or could be inferred from other characteristics of the space.

- $\mathsf{boundary}(o \in \mathsf{ConcreteObject}, a \in \mathsf{Area})_{@t}$
  This property associates object o with its spatial boundary represented by line l at time t, and it is established by looking at o's extension:

$$\begin{aligned}&\mathsf{boundary}(o, a)_{@t} \equiv \\ &\quad \mathsf{extension}(o, a)_{@t} \wedge \exists\, l \in \mathsf{Line}[\mathsf{boundary}(a, l)]\end{aligned} \tag{4.27}$$

- $\mathsf{inViewField}(o \in \mathsf{ConcreteObject})_{@t}$
  This property establishes whether object o is present and visible in the space currently under consideration. A very simple definition would establish whether there exists a position for o at time t:

$$\mathsf{inViewField}(o)_{@t} \equiv \exists\, p \in \mathsf{Point}\,[\mathsf{position}(o, p)_{@t}]$$

  This property is useful for some verb models in the following chapter, however its definition and implementation are quite dependent on the particular model of space resulting from grounding the ontology.

Our ontology allows for the expression of qualitative concepts. A con-

cept which can have a qualitative or quantitative nature, for example, is speed:

- $\mathsf{speed}(o \in \mathsf{ConcreteObject}, s \in \mathbb{V})_{@t}$
  Holds if the speed of object o is some value s at time t. As the speed of an object is a characteristic that results from movement, in order to evaluate the speed at a particular time instant, there is the need to observe the positions of o over a temporal window surrounding t:

$$\begin{aligned}&\mathsf{speed}(o, s)_{@t} \equiv \\ &\quad \exists\, \varepsilon\, \exists t_1, t_2 \in \mathcal{T}, p_1, p_2 \in \mathsf{Point}\ [\ t_1 = t - \frac{\varepsilon}{2} \wedge t_2 = t + \frac{\varepsilon}{2} \wedge \\ &\quad \mathsf{position}(o, p_1)_{@t_1} \wedge \mathsf{position}(o, p_2)_{@t_2} \wedge \\ &\quad \mathsf{distance}((p_1, p_2), d) \wedge s = \frac{d}{t_2 - t_1}]\end{aligned}$$

  The definition above calculates speed as the ratio between the distance and time units over a particular temporal window. The meaning of ε is for this temporal window to be *reasonably small*. As this is essentially a vague concept, the definition can be reformulated with precisifications:

$$\begin{aligned}&\mathsf{speed}[P](o, s)_{@t} \equiv \\ &\quad \exists t_1, t_2 \in \mathcal{T}, p_1, p_2 \in \mathsf{Point}, (W_{\mathsf{speed}}, w) \in P \\ &\quad [\ t_1 = t - \frac{w}{2} \wedge t_2 = t + \frac{w}{2} \wedge \mathsf{position}(o, p_1)_{@t_1} \wedge \mathsf{position}(o, p_2)_{@t_2} \wedge \\ &\quad \mathsf{distance}((p_1, p_2), d) \wedge s = \frac{d}{t_2 - t_1}] \qquad (4.28)\end{aligned}$$

  In the above, precisification P includes a threshold specifying a width of exactly w for the window over which the speed of o is evaluated.

- $\mathsf{speed}(o \in \mathsf{ConcreteObject}, v \in \{\mathrm{slow}, \mathrm{fast}, \mathrm{walkPace}, \mathrm{runPace}, \ldots\})_{@t}$
  This property characterises the speed of o at time t not as an absolute value, but with a qualitative description. The concept of some object's speed being slow or fast is inherently vague, therefore the definitions below are parameterised with precisification P in order to precisely interpret the predicate. Given a speed value $s \in \mathbb{V}$ obtained by the quantitative property speed above, qualitative notions

of speed can be defined:

$$\begin{aligned}
&\mathsf{speed}[P](o, \text{slow})_{@t} \equiv \\
&\quad \exists s \in \mathbb{V}, (T_{\text{slow}}, t_s) \in P \ [\mathsf{speed}[P](o, s)_{@t} \wedge s < t_s] \\
&\mathsf{speed}[P](o, \text{fast})_{@t} \equiv \\
&\quad \exists s \in \mathbb{V}, (T_{\text{fast}}, t_f) \in P \ [\mathsf{speed}[P](o, s)_{@t} \wedge s > t_f] \\
&\mathsf{speed}[P](o, \text{walkPace})_{@t} \equiv \\
&\quad \exists s \in \mathbb{V}, (T^{min}_{\text{walk}}, t^{min}_w)(T^{max}_{\text{walk}}, t^{max}_w) \in P[\mathsf{speed}[P](o, s)_{@t} \wedge t^{min}_w < s < t^{max}_w] \\
&\mathsf{speed}[P](o, \text{runPace})_{@t} \equiv \\
&\quad \exists s \in \mathbb{V}, (T^{min}_{\text{run}}, t^{min}_r)(T^{max}_{\text{run}}, t^{max}_r) \in P[\mathsf{speed}[P](o, s)_{@t} \wedge t^{min}_r < s < t^{max}_r]
\end{aligned}$$

For instances of AbstractObject the RCC topological relations have been introduced. These relations can be applied to instances of ConcreteObject as well, for example:

- $\mathsf{rcc\_DC}(o_1 \in \mathsf{ConcreteObject}, o_2 \in \mathsf{ConcreteObject})_{@t}$
  This relation holds if objects o_1 and o_2 are disconnected at time t. Such a relation can be inferred by examining the objects' positions:

$$\begin{aligned}
&\mathsf{rcc_DC}(o_1, o_2) \equiv \\
&\quad \mathsf{type}(o_1, \mathsf{ConcreteObject}) \wedge \mathsf{type}(o_2, \mathsf{ConcreteObject}) \wedge \\
&\quad \exists a_1, a_2 \in \mathsf{Area} \ [\mathsf{extension}(o_1, a_1)_{@t} \wedge \mathsf{extension}(o_2, a_2)_{@t} \wedge \\
&\quad \mathsf{rcc_DC}(a_1, a_2)]
\end{aligned} \tag{4.29}$$

In principle, the entire set of RCC relations can be formalised using the same approach as in the example above. There may be occasions where the abstract model of spatial areas linked to the space occupied by concrete objects may not allow for such a straightforward application. For example there may be uncertainty or errors in the data leading to imprecise positioning, resulting in non clear-cut topological relations.

Certain subclasses of ConcreteObject define specific properties relevant to the formalisation of certain situations or events, for example:

- $\mathsf{RelOrientation}(p \in \mathsf{Person}, ro)_{@t}$
  $\mathsf{RelOrientation}(v \in \mathsf{Vehicle}, ro)_{@t}$
  Holds if the relative orientation of person p or vehicle v is ro at time t.

Relative orientation could be inferred by looking at certain features of the objects, and it could be represented by *ro* in several ways, for example as left/right, an angle relative to the horizon or a quadrant of the space (this is relevant to the formalisation of verbs such as MoveAhead, Walk and Run, see Sections 5.1.1, 5.1.2 and 5.1.3). A simple way to specify relative orientation would be to refer to the possible relative positions between two objects in the set *relPos* for the property relPosition (see eq. 4.26).

- $\mathsf{sightDistance}(p \in \mathsf{Person}, d \in \mathbb{V})_{@t}$
  Holds if person p can see objects up to distance d at time t (relevant to the formalisation of Follow, Sec. 5.3.1).

- $\mathsf{vehicleBrakesOn}(v \in \mathsf{Vehicle})_{@t}$
  Holds if vehicle v brake lights are on at time t (relevant to the formalisation of Stop, Sec. 5.1.5).

- $\mathsf{movable}(o \in \mathsf{Item}, v \in \{\text{movable}, \text{partMovable}, \text{immovable}\})_{@t}$
  Holds if a particular item o is deemed to be movable, immovable or partially movable (relevant to the formalisation of Push, Sec. 5.4.2).

- $\mathsf{holdable}(o \in \mathsf{Item})_{@t}$
  Holds if a particular item o can be held at time t (relevant to the formalisation of Hold, Sec. 5.4.6).

- $\mathsf{spaceType}(s \in \mathsf{Space}, v \in \{\text{open}, \text{closed}\})$
  Holds if a particular space s is open or closed, assuming this property does not change over time. Open spaces have no particular constraints on how the space can be accessed, whilst closed spaces can only be accessed through specific parts and or in particular ways (relevant to the formalisation of Enter, Sec. 5.2.5).

- $\mathsf{partOf}(s \in \mathsf{Space}, o \in \mathsf{ConcreteObject})_{@t}$
  Holds if object o constitutes a part of space s at time t (relevant to the formalisation of Enter, Sec. 5.2.5).

- $\mathsf{accessible}(s \in \mathsf{Space}, o \in \mathsf{ConcreteObject})_{@t}$
  Holds if a space s can be accessed through object o, normally a part of s, at time t (relevant to the formalisation of Enter, Sec. 5.2.5).

The predicates listed above constitute only a brief example of how the ontology can grow to accommodate for a variety of properties and relations between objects. It is imaginable that in some domains these concepts may be grounded as primitives. This is the case if the data specifies exactly at which time and for which objects a particular property holds. In other domains these properties may have a definition which facilitates inferences about whether they hold or do not hold at particular times by examining whether other properties hold at that time.

4.7 Theory of Appearances

A computer system can only look at a representation of reality, for simplicity hereafter called the *appearance*. Appearances may result from algorithmic processing, such as a computer vision algorithm processing a video scene or manual annotations by human observers. These are the two data sources for grounding the ontology formalised in this chapter. This process is explained in detail in Chapter 6.

Dissimilarities in granularity and precision among different sources of appearances can be huge. In any case, they are always a partial, finite and incomplete representation of the real world and constitute all that is available for a computer system to ground the ontology. Thus the appearance represents everything an ontological system knows about the world. However, some details about the world not accounted for by the appearance may be logically inferred from the appearance itself, and possibly from some prior or acquired knowledge, such as the knowledge base arising from the context sketched in Sec. 3.4. Such augmented representation is hereafter referred to as the *inferred* appearance.

The ontology formalised so far has the potential to incorporate a *Theory of Appearances* layer performing an augmented representation of inferred appearances. This layer is located between the primitives layer and mid-level concepts and its general aim is to augment the knowledge stemming from appearances and adding modularity to the ontology by separating the low-level, data-dependent concepts from the mid- and high-level object properties and event predicates. Further aims of this theory could emerge in particular contexts. In fact, richer inferred appearances may be produced depending on granularity and precision of the data, such as the

inferred appearance of a person's hands and feet if not even shape and posture. Additionally, the Theory of Appearances may aid in correcting errors and noise in the data by discarding 'wrong' appearances, such as spurious tracks produced by a tracking algorithm.

Two specific examples of how a Theory of Appearances could be developed and its advantages are overviewed: the construction of a unified spatial model for the ontology and the issue of occlusion.

A Unified Spatial Model

The spatial models introduced in Sec. 4.2 demonstrate that the definition of ontological primitives concerning the position and relations between spatial points and regions are dependent on the data grounding the ontology. For example, in our domain, the data resulting from the processing of video image data could represent points in space according to a two- or a three-dimensional model, therefore the primitives defining position, connection, distance and topological relations between between points and regions have to be defined accordingly.

It is desirable for the ontology to provide a unified spatial model which minimises the influence carried by the grounding data. Such a model can be based on the three-dimensional model of Sec. 4.2.3 defining properties of sets $\mathcal{S}_p$ and $\mathcal{S}_r$, as this appears to be the most suitable for the representation of objects in space and their interactions. A Theory of Appearances would define a layer within the ontology whose purpose is to infer a representation of points and regions conforming with this unified model. The definition of properties and concepts in layers collocated above it would only have to refer to spatial properties within the unified model framework, thus abstracting higher-level properties from lower-level details of the implementation.

An example of how this could be achieved is demonstrated by assuming that the spatial model in the implementation data is constituted by a finite set of discrete, two-dimensional points. This is the case for the data on which the current implementation of ProVision grounds the ontology, (see Sec. 6.1), and also for the data provided by the automatic video tracking algorithms. Such a set of spatial data points $\mathcal{DP}$ would then be defined as a set of two-dimensional cartesian coordinates such that $\mathcal{DP} = \{(x, y) \in X \times Y\}$, where X and Y are bounded by the number of pixels in each video

frame. The unified ontological spatial model for the set $\mathcal{S}_p$ in the ontology is based on the model in Sec. 4.2.3, i.e. $\mathcal{S}_p = \{(x, y, z) \in X \times Y \times Z\}$. If the position of the observer is known, given a point $dp \in \mathcal{DP}$, it is possible to infer its position in the unified model as a point $p \in \mathcal{S}_p$ with a transformation algorithm. For example, if the observer is a fixed video camera, an algorithm can calculate three-dimensional coordinates based on the focal length of the camera lens. Another possibility is given by homographic transformations and ground plane estimation techniques [28, 32, 94].

The transformation function is defined as $transform_p : \mathcal{DP} \rightarrow \mathcal{S}_p$, such that each data point $dp \in \mathcal{DP}$ is mapped to a corresponding point $p \in \mathcal{S}_p$ in the unified ontological spatial model. It follows that the definition of this function is likely to be the only implementation- and data-dependent element of the ontology, as its definition varies according to the structure and nature of the set of data points $\mathcal{DP}$, the position of the observer (which could be fixed or could change over time) and the particular algorithm involved in the transformation.

In fact, the spatial primitives of the ontological model for $\mathcal{S}_p$ (Sections 4.2.1 and 4.2.3) such as identity, distance etc. need only be defined for points in $\mathcal{S}_p$. The definition of primitives for object types in AbstractObject such as Point and Area would refer to points in $\mathcal{S}_p$ and primitives defined on these points; a particular implementation providing the grounding data would only have to 'plug' the transformation function $transform_p$.

Detection of Occlusion

Another example of how a Theory of Appearances could augment the knowledge given by the data grounding the ontology is given by the issue of occlusion. This problem occurs if an object in space is hidden by another object over some interval, generally due to movement of one or both objects, and, as a consequence, the position of the occluded object is not represented in the data over that interval.

Fig. 4.1 shows a person o walking in space over interval $[t_s, t_e]$ and passing behind a brick wall during interval $[t_1, t_2] \subset [t_s, t_e]$. A tracking algorithm may represent the person's position as spatial area a at each time instant, represented by the thick rectangles. This representation should ideally determine the grounding of ontology primitives $\mathsf{p\_extension}(o, a)_{@t}$ for each $t \in [t_s, t_e]$. From the figure it is evident that such appearance suf-

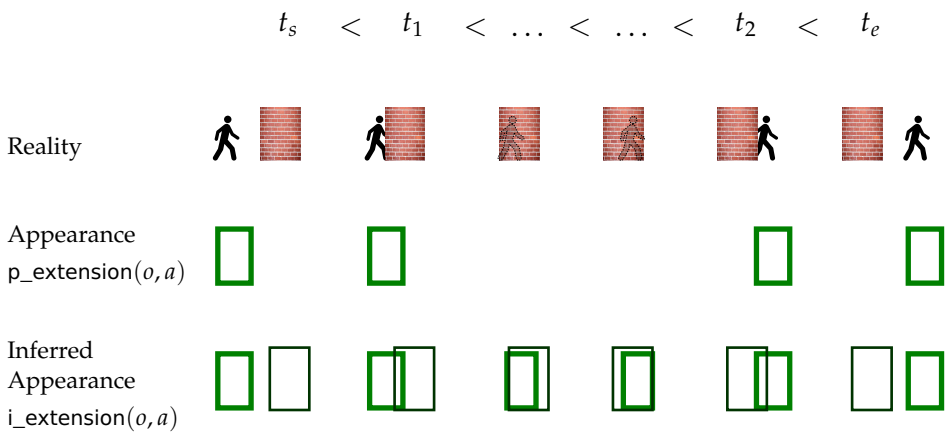


Figure 4.1: Theory of Appearances

fers from occlusion, as the person's extension is not represented between t_1 and t_2. An inferred appearance, extracted via logical inferences based on the p_extension primitives for $t < t_1$ and $t > t_2$, would augment the representation by understanding the occlusion, thus inferring the position of the wall, represented by the thin rectangle, and the position of the person while occluded by the wall. In this example, such inferred appearance would result in the grounding of predicates $\mathsf{i\_extension}(o, a)_{@t}$ for each $t \in [t_s, t_e]$.

It has to be noted that the issue of occlusion also relates to knowing or being able to estimate the position of the observer, as objects may appear hidden according to the perspective from which they are observed. Combining an occlusion detection system with the inference of a unified spatial model that takes the observer's position into account would greatly enhance the chances of successfully addressing this issue.

4.8 Events

An event represents a complex action and there is a distinction between *event-types* and *event-tokens* [18]. An event-type $e \in \mathcal{E}$ is associated with a set of instances of a particular event, for example: 'John approaches Mary', formalised as $\mathsf{Approach}(\text{John}, \text{Mary})$. An event-token $\sigma \in \Sigma$ constitutes an occurrence of a particular event-type over a temporal interval. To express occurrence of event type $e \in \mathcal{E}$ over time interval $I \in \mathcal{I}$, the construct $\mathsf{Occurs}(e, I)$ is introduced. The definition of an event occurrence often

involves specifying a particular sequence of fluents or sub-events that has to hold for the event to occur. For clarity of notation in the formulae to follow, event-types are capitalised as in Approach.

Chapter 5

Verb Models

This chapter illustrates some sample verb models applying the principles outlined in the previous chapter. Definitions in this ontology are not intended as exhaustive semantic characterisations of concepts, such as the ones that would be produced following a systematic linguistic analysis. Our formalisation needs to strike a compromise between the complexity of meaning, the practical task of detecting occurrences of such concepts on a coarse and imprecise representation of the real world and the implementability of these definitions in ProVision, the event recognition system described in the next chapter. For this reason, one cannot formulate very complex definitions, as they would either be too difficult to disambiguate, or impractical to break down and ground on the Theory of Appearances. The formalisations to follow are intended as a demonstration on how to apply such methodology.

The verbs modelled in this chapter, a selection of the ones listed in Table 1.1, are grouped according to their most prominent semantic characteristic, namely: simple motion, proximity, relation and contact. For the convenience of the reader, an index is provided at the end of the volume.

5.1 Simple motion

The verbs in this section concern the motion of objects in space: generic events such as Move, particular modalities of the type of motion (Walk, Run), directional or intentional movement (Go) and cessation of movement (Stop).

5.1.1 Move

According to the Oxford English Dictionary (OED, [2]), the intransitive form of the verb Move most commonly describes a person/thing going from one place, position or state to another, a person/thing changing posture, position or disposition. There is also a transitive use, for which there is no concern in this section, meaning "to change the place or position of someone/something".

The fact that object $o \in \mathcal{O}$ is moving at instant t can be formalised as follows:

$$\begin{aligned}&\mathsf{move}(o)_{@t} \equiv \\ &\quad \exists\, t_s, t_e \in \mathcal{T}, p_s, p_e \in \mathsf{Point}, \varepsilon\, [t_s = t - \frac{\varepsilon}{2} \wedge t_e = t + \frac{\varepsilon}{2} \\ &\quad \wedge \mathsf{position}(o, p_s)_{@t_s} \wedge \mathsf{position}(o, p_e)_{@t_e} \wedge p_s \neq p_e] \end{aligned} \tag{5.1}$$

The definition above refers to the position of o at time instants preceding and following t, respectively t_s and t_e, with the length ε of interval $[t_s, t_e]$ being sufficiently small to avoid considering longer spans over which an object may have moved despite the presence of subintervals where the object remained static. The duration of such interval can be made precise by parameterising the definition with precisification P containing threshold (W_{move}, w):

$$\begin{aligned}&\mathsf{move}[P](o)_{@t} \equiv \\ &\quad \exists\, (W_{\mathsf{move}}, w) \in P\ \exists\, t_s, t_e, p_s, p_e, p_t[(t_s < t < t_e) \wedge (t_e - t_s = w) \wedge \\ &\quad \mathsf{position}(o, p_s)_{@t_s} \wedge \mathsf{position}(o, p_e)_{@t_e} \wedge \\ &\quad \mathsf{position}(o, p_t)_{@t} \wedge p_t \neq p_s \wedge p_t \neq p_e] \end{aligned} \tag{5.2}$$

An alternative definition may make use of the property speed of object o (see Eq. 4.28, on page 81) and specifying a threshold in P:

$$\begin{aligned}&\mathsf{move}[P](o)_{@t} \equiv \\ &\quad \exists\, (T_{\mathsf{moveSpeed}}, t_s) \in P, s\ [\mathsf{speed}[P](o, s)_{@t} \wedge s > t_s] \end{aligned} \tag{5.3}$$

The event Move can now be defined by specifying that it occurs over

an interval if the fluent move holds on the same interval:

$$\mathsf{Occurs}(\mathsf{Move}[P](o), [t_s, t_e]) \equiv \mathsf{move}[P](o)_{@[t_s,t_e]} \tag{5.4}$$

It is useful to specify a few specialised occurrences of Move relevant to the formalisation of several other verbs in this chapter. Of interest are the movements of an object towards a particular direction, for example an object moving *up* or moving *left*. The following events are formalised by examining the relative position of the object at the start and end of an interval, given relation relPosition (see Eq. 4.9):

$$\begin{aligned}
&\mathsf{Occurs}(\mathsf{MoveUp}[P](o), [t_s, t_e]) \equiv \mathsf{Occurs}(\mathsf{Move}[P](o), [t_s, t_e]) \wedge \\
&\quad \exists\, p_s, p_e \in \mathsf{Point}\ [\mathsf{position}(o, p_s)_{@t_s} \wedge \mathsf{position}(o, p_e)_{@t_e} \wedge \\
&\quad \mathsf{relPosition}((p_e, p_s), \mathrm{above})] \qquad (5.5)\\
&\mathsf{Occurs}(\mathsf{MoveDown}[P](o), [t_s, t_e]) \equiv \mathsf{Occurs}(\mathsf{Move}[P](o), [t_s, t_e]) \wedge \\
&\quad \exists\, p_s, p_e \in \mathsf{Point}\ [\mathsf{position}(o, p_s)_{@t_s} \wedge \mathsf{position}(o, p_e)_{@t_e} \wedge \\
&\quad \mathsf{relPosition}((p_e, p_s), \mathrm{below})] \qquad (5.6)\\
&\mathsf{Occurs}(\mathsf{MoveLeft}[P](o), [t_s, t_e]) \equiv \mathsf{Occurs}(\mathsf{Move}[P](o), [t_s, t_e]) \wedge \\
&\quad \exists\, p_s, p_e \in \mathsf{Point}\ [\mathsf{position}(o, p_s)_{@t_s} \wedge \mathsf{position}(o, p_e)_{@t_e} \wedge \\
&\quad \mathsf{relPosition}((p_e, p_s), \mathrm{left})] \qquad (5.7)\\
&\mathsf{Occurs}(\mathsf{MoveRight}[P](o), [t_s, t_e]) \equiv \mathsf{Occurs}(\mathsf{Move}[P](o), [t_s, t_e]) \wedge \\
&\quad \exists\, p_s, p_e \in \mathsf{Point}\ [\mathsf{position}(o, p_s)_{@t_s} \wedge \mathsf{position}(o, p_e)_{@t_e} \wedge \\
&\quad \mathsf{relPosition}((p_e, p_s), \mathrm{right})] \qquad (5.8)\\
&\mathsf{Occurs}(\mathsf{MoveToFront}[P](o), [t_s, t_e]) \equiv \mathsf{Occurs}(\mathsf{Move}[P](o), [t_s, t_e]) \wedge \\
&\quad \exists\, p_s, p_e \in \mathsf{Point}\ [\mathsf{position}(o, p_s)_{@t_s} \wedge \mathsf{position}(o, p_e)_{@t_e} \wedge \\
&\quad \mathsf{relPosition}((p_e, p_s), \mathrm{front})] \qquad (5.9)\\
&\mathsf{Occurs}(\mathsf{MoveToBack}[P](o), [t_s, t_e]) \equiv \mathsf{Occurs}(\mathsf{Move}[P](o), [t_s, t_e]) \wedge \\
&\quad \exists\, p_s, p_e \in \mathsf{Point}\ [\mathsf{position}(o, p_s)_{@t_s} \wedge \mathsf{position}(o, p_e)_{@t_e} \wedge \\
&\quad \mathsf{relPosition}((p_e, p_s), \mathrm{back})] \qquad (5.10)
\end{aligned}$$

A further specialisation of Move involves a subject *moving ahead*, meaning that it is moving towards the direction it is facing according to its relative orientation. This is of course relevant only for objects which can be characterised with a relative orientation, for example people and vehicles.

The ways a relative orientation can be expressed can vary. In the following definition it is assumed that an object can face left, right, towards the front or the back. This is expressed by object property relOrientation and the event MoveAhead occurs over interval $[t_s, t_e]$ if object o moves towards the direction it is facing:

$$\begin{aligned}&\mathsf{Occurs}(\mathsf{MoveAhead}[P](o), [t_s, t_e]) \equiv \\ &\quad [\mathsf{relOrientation}(o, \mathrm{left})_{@[t_s,t_e]} \wedge \mathsf{Occurs}(\mathsf{MoveLeft}[P](o), [t_s, t_e])] \vee \\ &\quad [\mathsf{relOrientation}(o, \mathrm{right})_{@[t_s,t_e]} \wedge \mathsf{Occurs}(\mathsf{MoveRight}[P](o), [t_s, t_e])] \vee \\ &\quad [\mathsf{relOrientation}(o, \mathrm{front})_{@[t_s,t_e]} \wedge \mathsf{Occurs}(\mathsf{MoveToFront}[P](o), [t_s, t_e])] \vee \\ &\quad [\mathsf{relOrientation}(o, \mathrm{back})_{@[t_s,t_e]} \wedge \mathsf{Occurs}(\mathsf{MoveToBack}[P](o), [t_s, t_e])] \vee\end{aligned} \tag{5.11}$$

5.1.2 Walk

Instances of the verb Walk can be seen as specifications of the verb Move which concern the motion of human beings walking in space from one position to another at an appropriate pace. Such meaning is listed by the OED as "to move or travel at a regular and fairly slow pace by lifting and setting down each foot in turn, so that one of the feet is always on the ground" [2]. Figurative or metaphorical meanings where the verb describes the motion or behaviour of individuals other than people will not be considered here.

Given object $o \in \mathcal{O}$ of type Person, the predicate footstepAheadL holds on $[t_s, t_e]$ if o keeps the right foot on the ground throughout the interval while moving the left foot up and down within the interval, and at the same time moving it ahead according to o's relative orientation:

$$\begin{aligned}&\mathsf{footstepAheadL}[P](o)_{@[t_s,t_e]} \equiv \\ &\quad \exists\, f_l, f_r, t_1[(t_s < t_1 < t_e) \wedge \mathsf{type}(o, \mathsf{Person}) \wedge \mathsf{p\_feet}(o, (f_l, f_r)) \wedge \\ &\quad \mathsf{onGround}(f_r)_{@[t_s,t_e]} \wedge \mathsf{onGround}(f_l)_{@t_s} \wedge \\ &\quad \mathsf{Occurs}(\mathsf{MoveUp}[P](f_l), [t_s, t_1]) \wedge \mathsf{Occurs}(\mathsf{MoveDown}[P](f_l), [t_1, t_e]) \wedge \\ &\quad \mathsf{Occurs}(\mathsf{MoveAhead}[P](f_l), [t_s, t_e]) \wedge \mathsf{onGround}(f_l)_{@t_e}]\end{aligned} \tag{5.12}$$

Similarly, the predicate footstepAheadR defines the same movement

with the left foot on the ground and the right foot moving ahead:

$$\begin{aligned}
&\mathsf{footstepAheadR}[P](o)_{@[t_s,t_e]} \equiv \\
&\quad \exists\, f_l, f_r, t_1[(t_s < t_1 < t_e) \wedge \mathsf{type}(o, \mathsf{Person}) \wedge \mathsf{p_feet}(o, (f_l, f_r)) \wedge \\
&\quad \mathsf{onGround}(f_l)_{@[t_s,t_e]} \wedge \mathsf{onGround}(f_r)_{@t_s} \wedge \\
&\quad \mathsf{Occurs}(\mathsf{MoveUp}[P](f_r), [t_s, t_1]) \wedge \mathsf{Occurs}(\mathsf{MoveDown}[P](f_r), [t_1, t_e]) \wedge \\
&\quad \mathsf{Occurs}(\mathsf{MoveAhead}[P](f_r), [t_s, t_e]) \wedge \mathsf{onGround}(f_r)_{@t_e}]
\end{aligned} \tag{5.13}$$

The predicate footstepAhead generalises the previous two and holds on $[t_s, t_e]$ if any of footstepAheadL or footstepAheadR hold on the same interval. It also specifies that the speed of o's motion is walkPace:

$$\begin{aligned}
&\mathsf{footstepAhead}[P](o)_{@[t_s,t_e]} \equiv \\
&\quad \exists\, t_1[t_s < t_1 < t_e \wedge \mathsf{speed}[P](o, \mathrm{walkPace})_{@[t_s,t_e]} \wedge \\
&\quad [\ [\mathsf{footstepAheadL}[P](o)_{@[t_s,t_1]} \wedge \mathsf{footstepAheadR}[P](o)_{@[t_1,t_e]}] \\
&\qquad \vee \\
&\quad\ [\mathsf{footstepAheadR}[P](o)_{@[t_s,t_1]} \wedge \mathsf{footstepAheadL}[P](o)_{@[t_1,t_e]}]\ ]]
\end{aligned} \tag{5.14}$$

It is now possible to define the occurrence of $\mathsf{Walk}(o)$ over $[t_s, t_e]$ by expressing that there exists a set of ordered, adjacent and consecutive time intervals $(I, <)$ such that t_s and t_e correspond respectively to the start of the first interval and the end of the last interval, and the predicate footstepAhead holds on each interval $i \in I$:

$$\begin{aligned}
&\mathsf{Occurs}(\mathsf{Walk}[P](o), [t_s, t_e]) \equiv \mathsf{type}(o, \mathsf{Person}) \wedge \\
&\quad \exists\, (I, <)\, \exists\, I_s, I_e \in I\, [\mathit{begin}(I_s, t_s) \wedge \mathit{end}(I_e, t_e) \wedge \\
&\qquad \forall\, i \in I - \{I_s, I_e\}\, [I_s < i < I_e] \wedge \forall\, i \in I\, [\mathsf{footstepAhead}[P](o)_{@i}] \wedge \\
&\qquad \forall\, i_1, i_2 \in I[(i_1 < i_2 \wedge \neg \exists\, i' \in I\, [i_1 < i' < i_2]) \rightarrow \\
&\qquad\quad (\mathit{end}(i_1, t_1^e) \wedge \mathit{begin}(i_2, t_2^s) \wedge t_1^e = t_2^s)]]
\end{aligned} \tag{5.15}$$

5.1.3 Run

Similarly to Walk, the verb Run can also be characterised as a specification of Move which describes the fast motion of human beings. The OED describes it as "to go with quick steps on alternate feet, never having both

feet on the ground at the same time; to make one's way or cover the ground in this manner" [2].

Given object $o \in \mathcal{O}$ of type person the fluent runstepAhead defined below holds over the interval $[t_s, t_e]$ if o's feet are moving:

$$\begin{aligned}
&\mathsf{runstepAhead}[P](o)_{@[t_s,t_e]} \equiv \exists\, f_l, f_r, t_1, t_2 [(t_s < t_1 < t_2 < t_e) \wedge \\
&\quad \mathsf{type}(o, \mathsf{Person}) \wedge \mathsf{p_feet}(o, (f_l, f_r)) \wedge \mathsf{onGround}(f_r)_{@[t_s,t_1]} \wedge \\
&\quad \mathsf{MoveUp}[P](f_l)_{@[t_s,t_1]} \wedge \mathsf{MoveDown}[P](f_l)_{@[t_1,t_2]} \wedge \\
&\quad \mathsf{MoveAhead}[P](f_l)_{@[t_s,t_2]} \wedge \mathsf{onGround}(f_l)_{@[t_2,t_e]} \wedge \mathsf{MoveUp}[P](f_r)_{@[t_1,t_2]} \wedge \\
&\quad \mathsf{MoveDown}[P](f_r)_{@[t_2,t_e]} \wedge \mathsf{MoveAhead}[P](f_r)_{@[t_1,t_e]}] \wedge \\
&\quad \mathsf{speed}[P](o, \mathrm{runPace})_{@[t_s,t_e]}
\end{aligned} \tag{5.16}$$

It is now possible to define an occurrence of the event Run over interval $[t_s, t_e]$ as a periodical occurrence of runstepAhead:

$$\begin{aligned}
&\mathsf{Occurs}(\mathsf{Run}[P](o), [t_s, t_e]) \equiv \mathsf{type}(o, \mathsf{Person}) \wedge \\
&\quad \exists\, (I, <) \exists\, I_s, I_e \in I\, [\mathit{begin}(I_s, t_s) \wedge \mathit{end}(I_e, t_e) \wedge \\
&\qquad \forall\, i \in I - \{I_s, I_e\}\, [I_s < i < I_e] \wedge \forall\, i \in I\, [\mathsf{runstepAhead}[P](o)_{@i}] \wedge \\
&\qquad \forall\, i_1, i_2 \in I [(i_1 < i_2 \wedge \neg \exists\, i' \in I\, [i_1 < i' < i_2]) \rightarrow \\
&\qquad\quad (\mathit{end}(i_1, t_1^e) \wedge \mathit{begin}(i_2, t_2^s) \wedge t_1^e = t_2^s)]]
\end{aligned} \tag{5.17}$$

So far the definitions of the events Walk and Run are rather similar, their most prominent differentiation is the specification that the speed of object o is walk_pace for Walk and run_pace for Run. More detailed characterisations of the two verbs may be obtained by analysing the posture of the human body and the way legs, knees and feet bend and lift. Typically, a walking person's legs tend to be straight or slightly bent around the knee, with the thigh directed towards the ground. This would suggest that the angle between thigh and leg behind the knee is flat or slightly narrower than flat, and the angle between the thigh and a line perpendicular to the ground through the body would be smaller than 45°, with little variation during the course of the walk. On the other hand, a running person's thighs, legs and knees show a more articulated positioning, the angle between thigh and leg behind the knee can get narrower during the run whilst the angle between the thigh and the body's vertical line widens

as the thigh projects itself forward.

5.1.4 Go

The verb Go also suggests a generic motion in space and it is very close to Move. However, common usages of the verb seem to suggest some form of directionality or intentionality to such motion, for example a direction or position towards which the motion tends. The OED in fact lists three main meanings:

1. motion irrespective of the point of departure or destination,
2. motion *from* a place (often the speaker's position or "the point at which he mentally places himself")
3. motion *towards* a destination or direction (away from the speaker).

The meanings outlined above would suggest that one of the key features peculiar to Go is the consistency of an object's movement along a particular direction, if not even towards a specific destination.

The meaning corresponding to the first item on the list (motion irrespective of point of departure or destination) is formalised below with the event occurrence Go_1. Given object $o \in \mathcal{O}$, the event $\mathsf{Go}_1(o)$ occurs over the interval $[t_s, t_e]$ if o is moving and the direction of such movement is constant throughout the interval. This latter aspect is what ultimately differentiates an occurence of Go from an occurrence of the simpler verb Move. The direction of movement could be expressed by stating that the relative orientation of object o does not change through the interval.

$$\begin{aligned}&\mathsf{Occurs}(\mathsf{Go}_1[P](o), [t_s, t_e]) \equiv \\ &\quad \mathsf{Occurs}(\mathsf{Move}[P](o), [t_s, t_e]) \wedge \exists\, ro_s\, \mathsf{relOrientation}(o, ro_s)_{@t_s} \wedge \\ &\quad \forall\, t\, [t_s < t \leqslant t_e \rightarrow \exists\, ro_t\, (\mathsf{relOrientation}(o, ro_t)_{@t} \wedge ro_s = ro_t)]\end{aligned} \tag{5.18}$$

The second and third meanings previously listed have different formalisations according to whether the point of departure or destination is explicitly or implicitly stated, a fact that has a direct effect on the arity of the event predicate. The predicates $\mathsf{Go}_2(o_1, o_2)$ and $\mathsf{Go}_3(o_1, o_2)$ have both arity 2 with o_2 being the explicit object from which o_1 is respectively moving towards or away from (fluents moveTowards and MoveAwayFrom are

introduced in Sec. 5.2.1):

$$\mathsf{Occurs}(\mathsf{Go}_2[P](o_1,o_2),[t_s,t_e]) \equiv \\ \mathsf{Occurs}(\mathsf{Move}[P](o_1),[t_s,t_e]) \wedge \mathsf{Occurs}(\mathsf{MoveAwayFrom}(o_1,o_2),[t_s,t_e]) \tag{5.19}$$

$$\mathsf{Occurs}(\mathsf{Go}_3[P](o_1,o_2),[t_s,t_e]) \equiv \\ \mathsf{Occurs}(\mathsf{Move}[P](o_1),[t_s,t_e]) \wedge \mathsf{moveTowards}(o_1,o_2)_{@[t_s,t_e]} \tag{5.20}$$

Given the above definitions for Go_2 and Go_3 with arity 2 it is possible to formalise occurrences of the same predicates with arity 1. In this instance o_2 is assumed to exist as o':

$$\mathsf{Occurs}(\mathsf{Go}_2[P](o),[t_s,t_e]) \equiv \exists\, o' \,[\mathsf{Occurs}(\mathsf{Go}_2[P](o,o'),[t_s,t_e]] \tag{5.21}$$

$$\mathsf{Occurs}(\mathsf{Go}_3[P](o),[t_s,t_e]) \equiv \exists\, o' \,[\mathsf{Occurs}(\mathsf{Go}_3[P](o,o'),[t_s,t_e]] \tag{5.22}$$

In addition to the definitions of Go_2 and Go_3 above, it is possible to refer to the fact that the motion of o_1 may be directed towards or away from the position of the observer. Identifying such a position is heavily context-dependent; within the application domain it would seem reasonable to treat the observer as a virtual object positioned at the location of the camera filming the scene. The predicates $\mathsf{Go}_2^{\mathsf{s}}$ and $\mathsf{Go}_3^{\mathsf{s}}$ formalise the motion of o respectively from or towards the position of such observer:

$$\mathsf{Occurs}(\mathsf{Go}_2^{\mathsf{s}}[P](o),[t_s,t_e]) \equiv \\ \exists\, o_s[\mathsf{observer}(o_s)_{@t_s} \wedge \mathsf{Occurs}(\mathsf{Go}_2[P](o,o_s),[t_s,t_e])] \tag{5.23}$$

$$\mathsf{Occurs}(\mathsf{Go}_3^{\mathsf{s}}[P](o),[t_s,t_e]) \equiv \\ \exists\, o_s[\mathsf{observer}(o_s)_{@t_s} \wedge \mathsf{Occurs}(\mathsf{Go}_3[P](o,o_s),[t_s,t_e])] \tag{5.24}$$

The meanings formalised by the definitions of meanings Go_1, Go_2 and Go_3 (and relative sub-characterisations) all contribute to the definition of the generic event predicate Go, which holds over an interval if any of the events corresponding to either sub-meaning occur over the same interval:

$$\mathsf{Occurs}(\mathsf{Go}[P](o),[t_s,t_e]) \equiv \mathsf{Occurs}(\mathsf{Go}_1[P](o),[t_s,t_e]) \vee \\ \mathsf{Occurs}(\mathsf{Go}_2^{\mathsf{s}}[P](o),[t_s,t_e]) \vee \mathsf{Occurs}(\mathsf{Go}_3^{\mathsf{s}}[P](o),[t_s,t_e]) \vee$$

$$\mathsf{Occurs}(\mathsf{Go}_2[P](o), [t_s, t_e]) \vee \mathsf{Occurs}(\mathsf{Go}_3[P](o), [t_s, t_e]) \vee \tag{5.25}$$

5.1.5 Stop

This verb can be used transitively or intransitively. The general meaning of the intransitive form is "to come to a stand, cease to move or act", in particular "to cease from onward movement, to come to a stand or position of rest" [2], and such action can be said of people or inanimate things. The transitive form is associated with several meanings of which the most relevant to the application domain is "to bring to a stand", in particular "to arrest the onward movement of a person/thing; to bring to a stand or state of rest".

The intransitive form is the most relevant to the domain, and its formalisation appears relatively straightforward to formalise, as it involves identifying the fact that a particular object is ceasing its motion. The transitive form instead is more vague and complex, as it generally involves identifying the different ways in which an object may be causing the cessation of another object's motion. For this reason this section is focused on the intransitive form.

Nevertheless, whilst the fact that an object is ceasing motion may be objectively identifiable, its temporal extension is vague. Opinions may range from an interpretation where Stop is a punctual event happening in the instant at which the object stops moving to interpretations where the event spans an interval of variable length ending at such instant. The issue is individuating a criterion that establishes where the interval begins. In other words, it seems relatively objective to say that an object *has* stopped; identifying when the object *began* stopping is not. A very un-sophisticated criterion would formalise Stop as a fixed-duration event (e.g. n seconds before the cessation of movement), more elaborate ones would examine the object's motion pattern (e.g. sharp monotonic deceleration, speed lower than a particular threshold or the object reaching a particular position). Such criteria are very likely to depend on the type of object being considered, for example the event "a car stopping" may begin when its brake lights are activated, or "a person stopping" may begin on his/her last footstep. However, there is plenty of scope for some objects with random motion patterns to escape most of these criteria, for example

a very fast animal jumping from one place to another.

Another aspect to consider is the fact that there may be object motion patterns that exhibit short intervals of idleness between longer intervals of movement, and it is essentially a matter of granularity to consider such short intervals as occurrences of Stop or rather consider Stop to occur only when the encompassing motion pattern actually ceases. For example, let us consider a child playing on a swing, where the general pattern is for the child and the swing to oscillate back and forth between two positions. One could observe that between oscillations there is a very short interval in which the objects appear not to move. A very fine-grained interpretation of the meaning of Stop would assert that the event occurs over such short intervals, a coarse-grained interpretation may assert that the event begins when oscillations start covering a shorter distance (or even just on the last oscillation, for an even more coarse-grained interpretation) and ends when the swing and child are perpendicular to the ground. Another example is given by a person transporting objects from one location to another with short rests in between. A fine-grained view would have Stop occurring just before each rest, whilst a coarse-grained view would have the event occur only just before the last shuttle.

Capturing the peculiarities outlined so far in a logic formalisation, and many others which may be relevant in specific situations, is a challenging task. Below is a suggestion which is essentially based on the criterion of monotonic deceleration, with two refinements concerning vehicles and the stopping or walking motions of persons, and also provides a threshold relevant to the granularity aspect outlined in the previous paragraph.

Given object $o \in \mathcal{O}$, the event Stop occurs over the interval $[t_s, t_e]$ if o is moving over $[t_s, t_e]$, stopping holds on $[t_s, t_e]$ and the object is idle (i.e. not moving) for an interval starting at t_e and with a minimum length specified by precisification threshold T^{stop}_{idle}.

$$\begin{aligned}\mathsf{Occurs}(\mathsf{Stop}[P](o), [t_s, t_e]) \equiv \exists\, (T^{stop}_{idle}, t_i) \in P\ \exists\, t_2\, [t_e < t_2 \wedge \\ dur([t_e, t_2]) > t_i \wedge \mathsf{Occurs}(\mathsf{Move}[P](o), [t_s, t_e]) \wedge \\ \neg\mathsf{Occurs}(\mathsf{Move}[P](o), [t_e, t_2]) \wedge \mathsf{stopping}[P](o)_{@[t_s,t_e]}] \end{aligned} \tag{5.26}$$

Threshold T^{stop}_{idle} represents a simple solution for the granularity problem for periodical or oscillating motions, and provides a mechanism to allow

for fine- or coarse-grained interpretations, respectively corresponding to lower and higher values for the threshold.

The purpose of predicate stopping is to formalise the actual possible ways in which an object may stop its motion. The following definition refers to two specialised predicates for vehicle and persons and the generic predicate monoDecel:

$$\begin{aligned}&\mathsf{stopping}[P](o)_{@t} \equiv \\ &\quad [\mathsf{type}(o, \mathsf{Vehicle}) \wedge \mathsf{stoppingVehicle}(o)_{@t}] \vee \\ &\quad [\mathsf{type}(o, \mathsf{Person}) \wedge \mathsf{stoppingPerson}[P](o)_{@t}] \vee \mathsf{monoDecel}[P](o)_{@t}\end{aligned} \tag{5.27}$$

The predicate stoppingVehicle holds at time t if there is an instant t_1 prior to t, the vehicle brake lights are lit and the vehicle is also monotonically decelerating over the interval $[t_1, t]$:

$$\begin{aligned}&\mathsf{stoppingVehicle}(o)_{@t} \equiv \\ &\quad \exists\, t_1\, [t_1 < t \wedge \mathsf{vehicleBrakesOn}(o)_{@[t_1,t]} \wedge \mathsf{monoDecel}[P](o)_{@[t_1,t]}]\end{aligned} \tag{5.28}$$

The above predicate is not concerned about whether the vehicle will actually stop at some instant following t. In fact, it seems plausible to affirm that a vehicle is stopping over a particular interval only because it *looks like* it is stopping (i.e. slowing down and with its brakes lights on) and not because of a firm belief that it will eventually stop.

The predicate stoppingPerson distinguishes between the fact that person o may be walking or running in the interval $[t_1, t_2]$ surrounding t. The predicate holds at t if the walking or running step performed at $[t_1, t_2]$ is not followed by further steps for an interval whose minimum length is specified by T_{idle}^{stop}:

$$\begin{aligned}&\mathsf{stoppingPerson}[P](o)_{@t} \equiv \exists\, (T_{idle}^{stop}, i) \in P\, \exists\, t_1, t_2, t_3 \\ &\quad [(t_1 < t < t_2 \wedge \mathsf{footstepAhead}[P](o)_{@[t_1,t_2]}) \wedge \\ &\qquad (t_2 < t_3 \wedge dur([t_2, t_3]) > i \wedge \neg \mathsf{footstep\_ahead}[P](o)_{@[t_2,t_3]}) \vee \\ &\quad (t_1 < t < t_2 \wedge \mathsf{runstepAhead}[P](o)_{@[t_1,t_2]}) \wedge \\ &\qquad (t_2 < t_3 \wedge dur([t_2, t_3]) > i \wedge \neg \mathsf{runstepAhead}[P](o)_{@[t_2,t_3]})]\end{aligned} \tag{5.29}$$

The predicate monoDecel holds true if a moving object o is monoton-

ically decelerating. The following definition holds at time t if there is a sufficiently large interval $[t_1, t_2]$ around t (whose width is precisified by threshold T_{md}) such that the speed of o is monotonically decreasing at every point.

$$\mathsf{monoDecel}[P](o)_{@t} \equiv \exists\, t_1, t_2 \exists\, (T_{md}, t_{md}) \in P\, [t_2 - t_1 \geqslant t_{md} \wedge t_1 < t < t_2 \wedge \forall\, t', t''[t_1 < t' < t'' < t_2 \wedge (\mathsf{speed}(o, s')_{@t'} \wedge \mathsf{speed}(o, s'')_{@t''} \rightarrow s'' < s')]] \tag{5.30}$$

5.2 Proximity

This section formalises motion verbs which refer to the concept of proximity or nearness of one object to another object or position: Approach, Pass, Arrive, Leave, Enter and Exit. Some of these verbs, particularly Enter and Exit, allow a modelling through notions of topology.

One of the common issues across most verbs in this section is temporal extension, a problem already encountered in the formalisation of Stop (see Sec. 5.1.5). In fact, drafting a formal model for these verbs involves establishing the temporal interval over which such events occur. This process is hampered by the ambiguity over the precise instant at which an event starts or ends. It appears that the primary source of this ambiguity stems from the notion of proximity between two objects which most of these verbs refer to. Indeed, this is an intrinsecally vague concept, as there are many criteria that can be employed to define the time instants over which a predicate such as "a is near b" holds. This is particularly true for the verbs Enter, Exit, Arrive and Leave; an approach based on the individuation of object proximity boundaries has been followed.

5.2.1 Approach

The dictionary lists two main forms for the verb: an intransitive one meaning "to come nearer [...] in space" and a transitive one meaning "to come near *to*"[2]. Most formalisations of this latter meaning are likely to analyse the motion of subject and object in order to determine whether the subject is getting closer to the object. Below, this transitive meaning is discussed and Approach is formalised with arity 2. The intransitive meaning is considered as a specialised case of the transitive meaning at the end of this

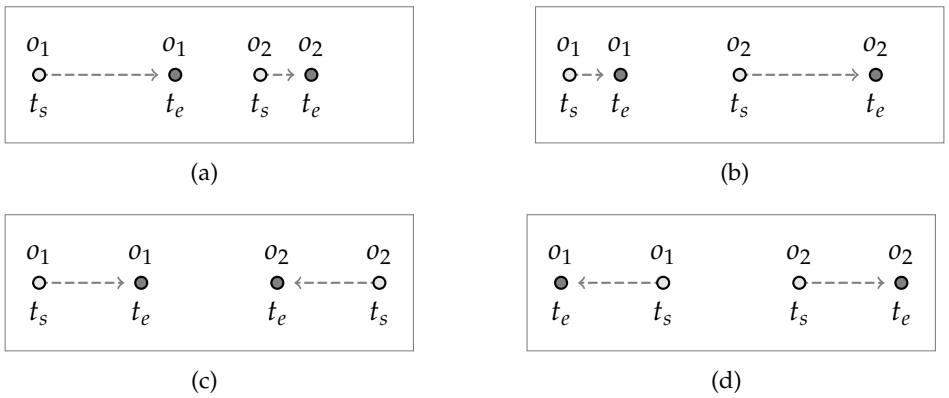


Figure 5.1: Verb Approach – Objects' positions

section.

Let us consider two objects $o_1, o_2 \in \mathcal{O}$ moving in space over the interval $[t_s, t_e] \in \mathcal{I}$. Fig. 5.1 shows some possible positions for o_1 and o_2 at t_s and t_e. An average observer would most likely assert the following:

- In Fig. 5.1(a), o_1 is approaching o_2, but o_2 is not approaching o_1 as o_2 is moving in the opposite direction;
- In Fig. 5.1(b), despite o_1 moving towards o_2, o_1 is not approaching o_2 as their distance does not decrease;
- In Fig. 5.1(c), o_1 is approaching o_2 *and* o_2 is approaching o_1 as they are both actively moving towards each other, and their distance decreases as well;
- In Fig. 5.1(d), clearly neither o_1 is approaching o_2 nor o_2 is approaching o_1 as they are heading towards opposite directions.

This is a simplification of the relative movement for two generic objects in space and does not take into account futher semantic properties that may be relevant.

Given the observations above about objects $o_1, o_2 \in \mathcal{O}$, the fact that o_1 is approaching o_2 can be defined by specifying that o_1 has to be both getting closer and moving towards o_2.

The fluent getCloserTo holds at time point t if and only if the distance between two objects $o_1, o_2 \in \mathcal{O}$ over an interval surrounding t monotoni-

cally decreases :

$$\begin{aligned}&\mathsf{getCloserTo}(o_1, o_2)_{@t} \equiv \\ &\quad \exists\, t_s, t_e \big[(t_s < t < t_e) \wedge \forall\, t', t'' \big[(t_s \leq t' < t'' \leq t_e) \rightarrow \exists\, d_s, d_e \\ &\quad \big[\mathsf{distance}((o_1, o_2), d_s)_{@t'} \wedge \mathsf{distance}((o_1, o_2), d_e)_{@t''} \wedge d_e < d_s\big]\,\big]\,\big] \end{aligned} \tag{5.31}$$

The fluent moveTowards holds at time point t if and only if the distance between o_1 and the start point of o_2 monotonically decreases over an interval surrounding t, i.e. o_1 is heading in the direction of o_2 irrespective of o_2's movements:

$$\begin{aligned}&\mathsf{moveTowards}(o_1, o_2)_{@t} \equiv \\ &\quad \exists\, t_s, t_e, p_{2s} \big[(t_s < t < t_e) \wedge \mathsf{position}(o_2, p_{2s}))_{@t_s} \wedge \\ &\quad \forall\, t', t'' \big[(t_s \leq t' < t'' \leq t_e) \rightarrow \exists\, d_s, d_e, p_{1s}, p_{1e} \\ &\qquad \big[\mathsf{position}(o_1, p_{1s})_{@t'} \wedge \mathsf{position}(o_1, p_{1e})_{@t''} \wedge \mathsf{distance}((p_{1s}, p_{2s}), d_s) \wedge \\ &\qquad \mathsf{distance}((p_{1e}, p_{2s}), d_e) \wedge d_s > d_e\big]\,\big]\big] \end{aligned} \tag{5.32}$$

It is now possible to define the event Approach:

$$\begin{aligned}&\mathsf{Occurs}(\mathsf{Approach}(o_1, o_2), [t_s, t_e]) \equiv \\ &\quad \mathsf{getCloserTo}(o_1, o_2)_{@t_s, t_e} \wedge \mathsf{moveTowards}(o_1, o_2)_{@t_s, t_e}\big] \end{aligned} \tag{5.33}$$

The definitions above have been implemented in the ontology with the introduction of precisification thresholds in Sec. 6.2.2.

The event $\mathsf{MoveAwayFrom}(o_1, o_2)$, expressing the occurrence of an event opposite in meaning to Approach and referred to by some verb definitions in the rest of this chapter, can be defined in a similar fashion by reversing the the definitions above and expressing that the distance between o_1 and o_2 ought to increase over the temporal intervals.

The intransitive meaning of Approach refers to the the subject's act of "coming nearer". Such a meaning may not necessarily refer to an explcit reference object that the subject is moving towards or getting closer to. However, a similarity with the verb Go (discussed in Sec. 5.1.4) can be drawn; the intransitive meanings of Go often refer to the observer's position. A human observer need not actually exist, this could be a virtual entity placed at the position where a human observer would be most

likely to be located (in a video scene this often corresponds to the camera location). For this reason, below is a formalisation of Approach with arity 1 as a particular case of the transitive form, where the implicit object refers to the observer's position:

$$\begin{aligned}&\mathsf{Occurs}(\mathsf{Approach}(o), [t_s, t_e]) \equiv \\ &\quad \exists\, o_s [\mathsf{observer}(o_s)_{@t_s} \wedge \mathsf{Occurs}(\mathsf{Approach}(o, o_s), [t_s, t_e]]\end{aligned} \tag{5.34}$$

The above definitions refer to the rather simple concepts of distance and position. Depending on the data grounding the ontology, one could employ a finer characterisation taking into account, for example, the type of objects involved, the terrain surrounding them, the different paths one object could take, the presence of constraints blocking a particular path, the effort required for each path etc.

5.2.2 Pass

The dictionary lists several meanings for the verb Pass, as it can be used in a variety of contexts and applied to different types of objects in a number of figurative ways, most of which loosely refer to an object's transition from a state to another. Several meanings do not refer directly to an act of motion; for the ones that do Pass "differs from Move" as it "expresses the effect [of the act of motion] rather than the cause" [2].

This section focuses on the meanings of the verb most closely related to acts of motion. Specifically, two forms can be identified [2]:

1. transitive, meaning "to go by or move past [...]; to go beyond (a point or place) [...]; to outrun or outdistance; (of a vehicle or its driver) to drive past, overtake; to get through, across, or over".

2. intransitive, meaning "to proceed, move forward, depart; to move along under a force; to go by or past".

There is also another motion-related meaning commonly used in natural language, which is "to cause to go from one person to another; to hand over, transfer". This particular meaning is very close to the meaning of Give and Hand, hence it is not discussed here.

In the application context, the most common example of the transitive form of $\mathsf{Pass}(o_1, o_2)$ is given by an object o_1 which moves towards an object

o_2 (not necessarily static) and *goes beyond* it, concept itself open to several interpretations. A reasonable simple formalisation would specify that o_1 is approaching o_2 over an interval $[t_1, t]$ and moving away from o_2 over the adjacent interval $[t, t_2]$. It would result that, at instant t, o_1 is either at the location of o_2 or at the point where the distance between o_1 and o_2 over interval $[t_1, t_2]$ is minimum.

There are two main sources of ambiguity in the the formalisation of Pass outlined above, namely temporal extension and directionality. The temporal extension issue is a recurring theme of similar almost-punctual events (such as Stop, see Sec 5.1.5), and has to do with the non-trivial task of establishing when the event $\mathsf{Pass}(o_1, o_2)$ starts and ends. Essentially, the interval $[t_1, t_2]$ needs to be bounded such that the event occurs in the instants immediately preceding and following the instant t identified in the previous paragraph. For the verb Stop the monotonic deceleration criterion has been employed; however, for the verb Pass, it appears more challenging to identify a similar meaningful, simple and measurable criterion. One could argue that $\mathsf{Pass}(o_1, o_2)$ starts when o_1 is *near* o_2 and ends when o_1 is *not near* o_2, thus refocusing the ambiguity on the formalisation of *near*. Resolving this is very complex due to deep ambiguity, as there are many observable properties concurring to establishing whether two objects are near each other (distance, objects' size, spatial extension...) and most of these properties generate instances of sorites vagueness (see Sec. 3.1). A simple formalisation would formalise the predicate with precisification P as $\mathsf{nearPosition}[P](o_1, o_2)$, holding when the linear distance between o_1 and o_2 is below a threshold specified by P, as in Eq. 4.25 on page 79.

Such a definition of nearPosition presents several limitations as it does not take contextual information into account, most importantly the type of o_1 and o_2 and their size. The mode of travel is also relevant, as further details such as transport links and terrain type may influence the degree of nearness. However, the disambiguation of the sorites vagueness associated with the linear distance observable property in nearPosition by means of a fixed threshold seems appropriate within the formalisation of Pass, as it serves the purpose of delimiting its applicability boundary. The formalisation of Pass proposed below simplifies this issue by requiring that

o_1 has to reach the position of o_2 before moving forward.

$$\begin{aligned}\mathsf{Occurs}(\mathsf{Pass}[P](o_1,o_2),[t_s,t_e]) \equiv \exists\, t_1,t_2,t_3,t_4\,[t_1 < t_s < t_2 \leqslant t_3 < t_e < t_4 \wedge \\ \mathsf{Occurs}(\mathsf{Approach}(o_1,o_2),[t_1,t_2]) \wedge \neg\mathsf{nearPosition}[P](o_1,o_2)_{@[t_1,t_s]} \wedge \\ \mathsf{nearPosition}[P](o_1,o_2)_{@[t_s,t_e]} \wedge \mathsf{samePosition}(o_1,o_2)_{@[t_2,t_3]} \wedge \\ \mathsf{Occurs}(\mathsf{MoveAwayFrom}(o_1,o_2),[t_3,t_4]) \wedge \neg\mathsf{nearPosition}[P](o_1,o_2)_{@[t_e,t_4]}]\end{aligned} \tag{5.35}$$

The directionality issue is related to the fact that most instances of $\mathsf{Pass}(o_1,o_2)$ are likely to exhibit a motion pattern for o_1 which is consistenly directed towards a particular direction or path. The definition above, for example, would hold if o_1 were to move towards o_2 till very near it but without reaching its position, and then move away from o_2 towards the way it came from. Such an occurrence would be unlikely to be described as an occurrence of Pass by the average observer. In other words, it has to be that either o_1 keeps moving along the same direction it approached o_2, or it follows a particular path on which o_2 is situated. In practice, extending definition (5.35) to formalise this aspect is difficult, as there may be many admissible changes in direction over the course of o_1's motion which may incorporate appropriate occurrences of Pass. For example if o_1 is a train and o_2 a station on the line positioned on a sharp bend, o_1's direction would change when passing o_2, but this is appropriate and expected given the particular path constraining o_1's motion.

The intransitive meaning of $\mathsf{Pass}(o)$ refers to the act of object o appearing and moving through the scene under observation. A manifestation of this motion over interval $[t_s,t_e]$ would have o enter the field of view at t_s, move forward with a consistent direction through the interval and exit the field of view at t_e. The general assumption is that occurrences where o does not enter and exit the field of view at the interval boundaries, or where o does not move forward consistently (i.e. pauses or changes in direction) would be unlikely to be described by $\mathsf{Pass}(o)$.

The previous example introduces the concept of a 'field of view' that can be shaped in several ways. For instance, if the observed situation is filmed by a fixed camera, the field of view may be defined as a rectangle corresponding to the scene captured by the camera. More complex formalisations may define the field of view more precisely as the focal point

of the scene, i.e. the area of the scene where an observer would be likely to focus his/her attention. This can be determined by analysing object's positions and where most of the activity is taking place. In relation to the meaning of $\mathsf{Pass}(o)$, a too specific formalisation may not be appropriate, as it is imaginable that a situation where o enters, moves along and exits the scene away from where most of the activity is taking place would still be classified as an occurrence of $\mathsf{Pass}(o)$. The following definition formalises such sense of Pass:

$$\begin{aligned}\mathsf{Occurs}(\mathsf{Pass}[P](o), [t_s, t_e]) \equiv \exists\, t_1, t_2\, [t_1 < t_s \wedge t_e < t_2 \wedge \\ \neg\mathsf{inViewField}(o)_{@[t_1,t_s]} \wedge \mathsf{inViewField}(o)_{@[t_s,t_e]} \wedge \\ \mathsf{Occurs}(\mathsf{Move}[P](o), [t_s, t_e]) \wedge \neg\mathsf{inViewField}(o)_{@[t_e,t_2]}\end{aligned} \tag{5.36}$$

The definition above does not specify the characteristics of o's motion while it is in the field of view. This is because there are many possible motions exhibiting the consistency one would expect to see during an occurence of Pass (for example o may move in a straight line, or on a big loop which would take it out of the field of view on the direction it came from), hence it is left unspecified at this stage.

5.2.3 Arrive

The dictionary lists several meanings for the verb Arrive, the most common use within the application domain is generally for an object "to come to the end of a journey, to a destination, or to some definite place; to come upon the scene, make one's appearance" [2].

This meaning suggests a directional motion, however the specific manner of motion is not specified but is often dependent on the type of objects involved [69]. For example, if object o is moving along a known path, one could state that o has *arrived* when its motion has terminated at the end of such path in the direction o has been travelling from. If o is an object of type person, it is likely that o's motion is an occurrence of verbs Walk or Run, even though there are instances where o is transported on a vehicle (car, train...), in which case its motion would be described by the vehicle's motion.

The verb Arrive is intransitive, but often an indirect object or location specifies the destination or place where motion terminates. An occurrence

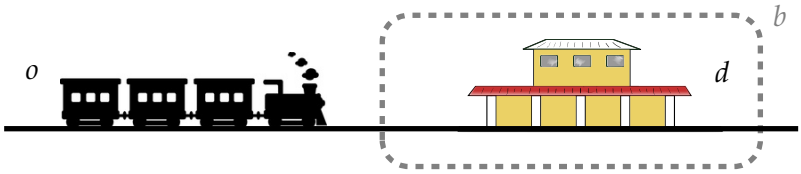


Figure 5.2: Verb Arrive

of the verb where the indirect object d is specified, such as 'the train arrived *at the station*' would be formalised with arity 2, as in $\mathsf{Arrive}(o, d)$. Another use of Arrive is not followed by an indirect object, such as 'John has arrived', which would be formalised with arity 1, as in $\mathsf{Arrive}(o)$. However, an indirect object is almost always implicit and can be determined by analysing contextual information. For example, John may have been travelling on a particular route and come to its (known) terminus, or the utterance may refer to a meeting and the fact that John has reached the meeting location.

The verb shares the same issue of temporal extension already discussed in the formalisation of Stop (Sec. 5.1.5) and Pass (Sec. 5.2.2), namely the fact that events of this kind are punctual or near-punctual events, i.e. their occurrences generally span a small temporal interval whose duration and precise individuation is questionable. One could argue that, in general, an occurrence of $\mathsf{Arrive}(o, d)$ starts when o is in the vicinity of d and ends when o is at d's position. In order to establish whether o is in the vicinity of d, one could assume that there exists a spatial area surrounding d which defines its proximity boundary; the event $\mathsf{Arrive}(o, d)$ would start when o enters such area. For example, let us consider a train o arriving at a station d, illustrated in Fig. 5.2. Most formalisations would agree on the fact that the interval over which the event $\mathsf{Arrive}(o, d)$ occurs would terminate on the instant the train stops at the station's platform. A possibility to establish the starting instant of the interval would be to draw a spatial area constituting the proximity boundary around d and affirm that the event $\mathsf{Arrive}(o, d)$ starts when the train enters such area (this is illustrated in Fig. 5.2 by the dashed rectangle b).

A predicate expressing this is $\mathsf{nearBoundary}(d, b)_{@t}$, which holds if and only if b is an instance of Area representing the proximity boundary of object d at time t. There are many criteria and observable properties lead-

ing to the demarcation of proximity boundaries for spatial objects, hence such a concept is necessarily vague. In general, boundaries for concrete and atomic objects with a precise spatial extension (e.g. people, vehicles, small items) are less ambiguous than boundaries for abstract, complex or extended objects (e.g. cities, stations, mountains). In the former kind of objects, their spatial extension is often precisely known, and a simple proximity boundary may include the region within a fixed distance d from such spatial extension. The latter kind of objects, instead, have an indeterminate spatial extension which may vary according to different criteria appropriate within the utterance context. For example, a city within a large conurbation may have a precise boundary established by law (*de iure* boundary, e.g. the local authority), nevertheless people in the conurbation may refer to buildings and inhabitants in surrounding towns as being in the city even if, legally, they are not (*de facto* boundary, not unique and generally not precisely determined). In the train and station example, one may consider the *de iure* boundary assigned to the railway station by the land register, but other *de facto* boundaries may exist. For example one could consider the line past which a train approaching the station can be seen from the platform, and establishing such line is subject to a moltitude of criteria (there may be many platforms, different fields and angles of view, etc.). Furthermore, other criteria for the proximity boundary may be appropriate when considering different contexts, e.g. a person arriving at a station, where one may choose to draw such boundary around the station entrance.

It is likely that an ontology will incorporate many formalisations of nearBoundary, each of them specialised to define the proximity boundary of a particular type of object according to its specifities, as briefly demonstrated in the examples above. Nevertheless, for the purposes of this work, it is imaginable to formalise a very general and still usable definition of the concept which establishes the boundary b as a crisp area around the position of object o, according to the precisification threshold T_b:

$$\begin{aligned}&\mathsf{nearBoundary}[P](d,b)_{@t} \equiv \\ &\quad \exists\,(T_b,t_b) \in P,\ b \in \mathsf{Area}, p_d \in \mathsf{Point}, \delta\,[\mathsf{position}(d,p_d)_{@t} \wedge \\ &\quad \forall\, p \in \mathsf{Point}[\mathsf{pointInArea}(p,b) \rightarrow \mathsf{distance}((p,p_d),\delta) \wedge \delta \leqslant t_b] \end{aligned} \tag{5.37}$$

The principles outlined above can be applied in the following formalisation of the concept $\mathsf{Arrive}(o, d)$, occurring when object o arrives at destination d (which can be another object or a spatial location):

$$\begin{aligned}&\mathsf{Occurs}(\mathsf{Arrive}[P](o,d),[t_s,t_e]) \equiv \\ &\quad \exists t_1,t_2,t_3 \in \mathcal{T}, b \in \mathsf{Area}[t_1 < t_s < t_2 \leqslant t_3 < t_e \wedge \\ &\quad \mathsf{nearBoundary}[P](d,b)_{@[t_1,t_e]} \wedge \mathsf{moveTowards}(o,b)_{@[t_1,t_s]} \wedge \\ &\quad \mathsf{Occurs}(\mathsf{Enter}(o,b),[t_s,t_2]) \wedge \mathsf{Occurs}(\mathsf{Stop}[P](o),[t_3,t_e])] \end{aligned} \tag{5.38}$$

The above definition states that the event $\mathsf{Arrive}(o, d)$ holds over interval $[t_s, t_e]$ if o is moving towards o's boundary b in the preceeding interval $[t_1, t_s]$, o is entering boundary b over subinterval $[t_s, t_2]$ (the verb Enter is defined in Sec. 5.2.5) and stops at the end of the interval over $[t_3, t_e]$.

As discussed earlier in this section, an occurrence of Arrive with arity 1, i.e. not followed by the indirect object specifying the destination of motion, such as in "John has arrived", may refer to an implicit indirect object that can be determined by analysing the context in which the occurrence takes place. This leads to the formalisation of $\mathsf{Arrive}(o)$, which occurs over interval $[t_s, t_e]$ if there exists an object d constituting the destination for an occurrence of $\mathsf{Arrive}(o, d)$ over the same interval:

$$\mathsf{Occurs}(\mathsf{Arrive}_1[P](o),[t_s,t_e]) \equiv \exists\, d \in \mathcal{O}[\mathsf{Occurs}(\mathsf{Arrive}[P](o,d),[t_s,t_e]) \tag{5.39}$$

The above definition could be refined by individuating a link between d and the route or path followed by object o.

However, at the beginning of this section a slightly particular submeaning of Arrive has been listed: "to come upon the scene, make one's appearance". This meaning also leads to a formalisation of $\mathsf{Arrive}(o)$ with arity 1, which mirrors the formalisation of $\mathsf{Pass}(o)$ in eq. 5.36, except for the fact that here o stops once it has entered the scene:

$$\begin{aligned}&\mathsf{Occurs}(\mathsf{Arrive}_2[P](o),[t_s,t_e]) \equiv \exists\, t_1,t_2\, [t_1 < t_s < t_2 < t_e \wedge \\ &\quad \neg\mathsf{inViewField}(o)_{@[t_1,t_s]} \wedge \mathsf{InViewField}(o)_{@[t_s,t_e]} \wedge \\ &\quad \mathsf{Occurs}(\mathsf{Move}[P](o),[t_s,t_e]) \wedge \mathsf{Occurs}(\mathsf{Stop}[P](o),[t_2,t_e]) \end{aligned} \tag{5.40}$$

A general formalisation for $\mathsf{Arrive}(o)$ can then be defined as the disjunction of the two variants defined above:

$$\begin{aligned}&\mathsf{Occurs}(\mathsf{Arrive}[P](o), [t_s, t_e]) \equiv \\ &\quad \mathsf{Occurs}(\mathsf{Arrive}_1[P](o), [t_s, t_e]) \vee \mathsf{Occurs}(\mathsf{Arrive}_2[P](o), [t_s, t_e])\end{aligned} \tag{5.41}$$

5.2.4 Leave

Given the meanings listed by the dictionary, the most relevant to the application domain is "To go away from, quit (a place, person or thing)"[2]. There is also a meaning close to this, which is "to deviate from (a line of road, etc.)", however the discussion is not concerned with this particular sub-meaning. In this sense, the verb Leave can be framed as the opposite of Arrive; it follows that most of the considerations on the pecularities of Arrive exposed in the previous section apply to Leave as well. The main difference between the two is that whilst Arrive is intransitive, with an optional indirect object specifying the place the subject is arriving at, Leave is transitive, with the optional direct object specifying the place the subject is departing from.

The problem of establishing a duration for an occurrence of Leave can be tackled in the same way, that is by establishing a proximity boundary around the object or area the subject is departing from. Given objects $o, s \in \mathcal{O}$, the following definition formalises an occurrence of $\mathsf{Leave}(o, s)$ over the interval $[t_s, t_e]$ by stating that o leaves object s if o is at the position of s at instant t_s, moves away from s over an interval extending beyond t_e and exits the proximity boundary of s at the end of $[t_s, t_e]$ (the verb Exit is defined in Sec. 5.2.6):

$$\begin{aligned}&\mathsf{Occurs}(\mathsf{Leave}[P](o, s), [t_s, t_e]) \equiv \\ &\quad \exists\, t_1, t_2 \in \mathcal{T},\, b \in \mathsf{Area}[t_s < t_1 < t_e < t_2 \wedge \mathsf{nearBoundary}[P](s, b)_{@[t_s, t_e]} \wedge \\ &\quad \mathsf{samePosition}(o, s)_{@t_s} \wedge \mathsf{Occurs}(\mathsf{MoveAwayFrom}(o, s), [t_s, t_2]) \wedge \\ &\quad \mathsf{Occurs}(\mathsf{Exit}(o, b), [t_1, t_e])]\end{aligned} \tag{5.42}$$

As for Arrive, the verb Leave may not necessarily be followed by a direct object, as in the example 'John left'. Such object is often implicit and can be determined by analysing the context surrounding the occurrence

of Leave, which may contain information about the fact that John is at a particular place, or is supposed to start travelling along a definite route. This can be formalised in the definition of $\mathsf{Leave}(o)$ with arity 1, which occurs over interval $[t_s, t_e]$ if there exists an object s participating in an occurrence of $\mathsf{Leave}(o, s)$ over the same interval:

$$\mathsf{Occurs}(\mathsf{Leave}_1[P](o), [t_s, t_e]) \equiv \exists\, s \in \mathcal{O}[\mathsf{Occurs}(\mathsf{Leave}[P](o, s), [t_s, t_e]) \tag{5.43}$$

Given the fact that so far Leave has been considered as being the opposite of Arrive, it is conceivable that an occurrence of Leave with an unspecified direct object could also refer to the subject's departure or disappearance from the scene, hence representing the opposite of the meaning of $\mathsf{Arrive}_2(o)$ formalised in eq. 5.40. The following definition formalises an occurrence of $\mathsf{Leave}_2(o)$ over interval $[t_s, t_e]$, by stating that o is within the field of view and moves over interval $[t_s, t_e]$ and disappears from the field of view in an interval following t_e:

$$\begin{aligned}&\mathsf{Occurs}(\mathsf{Leave}_2[P](o), [t_s, t_e]) \equiv \\ &\quad \exists\, t_1\, [t_e < t_1 \wedge \mathsf{inViewField}(o)_{@[t_s,t_e]} \wedge \\ &\quad \mathsf{Occurs}(\mathsf{Move}[P](o), [t_s, t_e]) \wedge \neg \mathsf{inViewField}(o)_{@[t_e,t_1]}\end{aligned} \tag{5.44}$$

A general formalisation for $\mathsf{Leave}(o)$ can then be defined as the disjunction of the two variants defined above:

$$\begin{aligned}&\mathsf{Occurs}(\mathsf{Leave}[P](o), [t_s, t_e]) \equiv \\ &\quad \mathsf{Occurs}(\mathsf{Leave}_1[P](o), [t_s, t_e]) \vee \mathsf{Occurs}(\mathsf{Leave}_2[P](o), [t_s, t_e])\end{aligned} \tag{5.45}$$

5.2.5 Enter

Among the meanings listed by the dictionary, the one that seems most appropriate to the motion verbs domain is the transitive form of Enter with the sense of "to go or come into (a closed space); to go within the bounds of"[2].

It is possible to distinguish between two broad categories of objects or areas that one can enter: *open* and *closed* spaces.

In general, open spaces are areas delimited by vague or crisp bound-

aries which, most of the times, pose no particular constraints on the way in which one can enter the area. For example, an area delimited by a vague bondary such as a town centre can be accessed in a multitude of ways: a pedestrian may be walking and step into the area over the course of his walking motion, or a vehicle may be travelling on a road that crosses the area. Accessibility constraints may still arise given the context and type of objects involved (e.g. a vehicle generally is constrained to move along roads, and only certain roads enter the town centre), but they are not imposed by the area itself.

On the other hand, most closed spaces carry an accessibility specification of the objects that can enter such space and the ways in which they are allowed to do so. For example, a house is normally entered only by people, and they must do so through a door or some other designated passage. There are occasions in which these constraints are violated, such as 'the lorry entered (into) the house kitchen' in the context of an accident, or 'Santa Claus entered through the chimney', but these uses are less frequent and imply some sense of anormality, in respect to the fact that expected conventions have not been followed. Modelling an occurrence of Enter in the context of a closed space carries additional steps, which involve the individuation of such accessibility constraints and the verification of whether they are met by the subject performing the action.

The distinction between the two modalities an object o_1 can enter the space of another object o_2 is formalised in the following definition of Enter, where if o_2 is a closed space then o_1 has to enter o_2 through a sub-part s_2 as specified by the relation accessible:

$$\begin{aligned}
&\mathsf{Occurs}(\mathsf{Enter}(o_1, o_2), [t_s, t_e]) \equiv \\
&\quad [\mathsf{spaceType}(o_2, \mathrm{closed}) \wedge \exists s_2 \in \mathsf{ConcreteObject}[\mathsf{partOf}(o_2, s_2) \wedge \\
&\qquad \mathsf{accessible}(o_2, s_2) \wedge \mathsf{Occurs}(\mathsf{Enter}_{\mathsf{area}}(o_1, s_2), [t_s, t_e])]] \vee \\
&\quad [\mathsf{spaceType}(o_2, \mathrm{open}) \wedge \mathsf{Occurs}(\mathsf{Enter}_{\mathsf{area}}(o_1, o_2), [t_s, t_e])]
\end{aligned} \tag{5.46}$$

The definition above introduces the sub-concept $\mathsf{Enter}_{\mathsf{area}}$, which represents the actual motion event that occurs when an object o_1 enters the space of another object o_2. The verb Enter, and particularly this sub-concept of the verb, suffers from the issue of temporal extension common across this section. In order to tackle the problem, a formalisation of $\mathsf{Enter}_{\mathsf{area}}$ based

on the changes in the topological relations between objects is proposed; this results in the decomposition of the event occurrence into stages, leading to the identification of start and end instants. The topological relations between objects are introduced in Sec. 4.6 and are based on the RCC calculus. Given objects o_1 and $o_2 \in \mathcal{O}$, an event $\mathsf{Enter}_{\mathsf{area}}(o_1, o_2)$ occurs over $[t_s, t_e]$ if objects o_1 and o_2 are disconnected prior to t_s, externally connected in the first part of $[t_s, t_e]$, partially overlapping in the final part of $[t_s, t_e]$ and o_1 is a tangential proper part of o_2 following t_e:

$$\begin{aligned}&\mathsf{Occurs}(\mathsf{Enter}_{\mathsf{area}}(o_1, o_2), [t_s, t_e]) \equiv \\ &\quad \exists t_1, t_2, t_3 \, [t_1 < t_s < t_2 < t_e < t_3 \wedge \mathsf{rcc\_DC}(o_1, o_2)_{@[t_1, t_s]} \wedge \\ &\quad \mathsf{rcc\_EC}(o_1, o_2)_{@[t_s, t_2]} \wedge \mathsf{rcc\_PO}(o_1, o_2)_{@[t_2, t_e]} \wedge \mathsf{rcc\_TPP}(o_1, o_2)_{@[t_e, t_3]}]\end{aligned} \quad (5.47)$$

The particular choice of topological relations in the definition above implies the assumption that the space occupied by o_2 is capable of fully containing o_1 by the time an occurrence of $\mathsf{Enter}_{\mathsf{area}}(o_1, o_2)$ terminates. This appears to be a reasonable assumption given that when an object enters a space, a speaker generally means that the object will ultimately be fully contained in such space. However, there may be instances where this is not necessarily the case and for which the definition above is likely to fail.

The definitions above constitute an over-simplification of the complexity of the verb Enter, particularly regarding the ways an object o_1 may enter a closed space o_2. For instance, a house may be entered through a door which has to be unlocked and opened with a set of defined motions, for example by pushing down a handle and pressing against the frame, before the subject is actually able to enter the interior of the area occupied by the house. It is easy to see that there is a multitude of different kinds of closed spaces, each with its on accessibility peculiarities. The approach followed above assumes that, if o_2 is a closed space, then it can be accessed by one of its sub-parts s_2 which o_1 can enter as any other open space.

5.2.6 Exit

The general meaning of Exit is "to leave, to get out of (a building, road, etc.)"[2]. As for Arrive and Leave, Enter and Exit are essentially opposites, hence most of the aspects discussed for Enter in the previous section apply to Exit as well.

The following definition of $\mathsf{Exit}(o_1, o_2)$ distinguishes between open or closed spaces:

$$\begin{aligned}
&\mathsf{Occurs}(\mathsf{Exit}(o_1, o_2), [t_s, t_e]) \equiv \\
&\quad [\mathsf{spaceType}(o_2, \text{closed}) \rightarrow \exists s_2 \in \mathsf{ConcreteObject}[\mathsf{partOf}(o_2, s_2) \wedge \\
&\qquad \mathsf{accessible}(o_2, s_2) \wedge \mathsf{Occurs}(\mathsf{Exit}_{\mathsf{area}}(o_1, s_2), [t_s, t_e])]] \vee \\
&\quad [\mathsf{spaceType}(o_2, \text{open}) \rightarrow \mathsf{Occurs}(\mathsf{Exit}_{\mathsf{area}}(o_1, o_2), [t_s, t_e])]
\end{aligned} \tag{5.48}$$

The definition of $\mathsf{Exit}_{\mathsf{area}}$ mirrors the definition of $\mathsf{Enter}_{\mathsf{area}}$ (eq. 5.47) by specifying a reversed sequence of topological relation changes between o_1 and o_2. An instance of $\mathsf{Exit}_{\mathsf{area}}(o_1, o_2)$ occurs over interval $[t_s, t_e]$ if o_1 is a non-tangential proper part of o_2 in an interval preceeding t_s, is a tangential proper part over the start of $[t_s, t_e]$, is partially overlapping with o_2 at the end of $[t_s, t_e]$ and is externally connected to o_2 in an interval following t_e:

$$\begin{aligned}
&\mathsf{Occurs}(\mathsf{Exit}_{\mathsf{area}}(o_1, o_2), [t_s, t_e]) \equiv \\
&\quad \exists t_1, t_2, t_3 \, [t_1 < t_s < t_2 < t_e < t_3 \wedge \mathsf{rcc_NTPP}(o_1, o_2)_{@[t_1, t_s]} \wedge \\
&\quad \mathsf{rcc_TPP}(o_1, o_2)_{@[t_s, t_2]} \wedge \mathsf{rcc_PO}(o_1, o_2)_{@[t_2, t_e]} \wedge \mathsf{rcc_EC}(o_1, o_2)_{@[t_e, t_3]}]
\end{aligned} \tag{5.49}$$

5.3 Relation

The verbs in this section identify a type of motion which is closely connected to a relation or connection between two objects. The verb Follow refers to an event occurrence where the motion of an object is causing another object to move in a way so that the second object does not lose sight of the first. Chase is a specification of this where the action is fast-paced, and Flee characterises the motion of an object in a direction opposite to something following, chasing or otherwise perceived as a danger.

5.3.1 Follow

The dictionary lists Follow as a transitive verb meaning "To go or come after (a person or other object in motion); to move behind in the same direction"[2]. There are several ways in which this can be formally characterised.

A good indication that an object is following another is the fact that both objects are moving along the same trajectory. For example, if a car is moving along a route, another vehicle may be following the former by travelling on exactly the same roads at a slightly later time.

However, different characterisations may be appropriate, especially for instances where the second object is not so explicitly or openly following the first. Let us consider the case of a person a inconspicuously following an unaware person b in an urban environment. Rather than passing on exactly the same route walked by b at the same pace keeping a fixed distance between them, a is much more likely to disguise its movements by stopping frequently, crossing the road, slowing down or even taking diversions (e.g. side streets, parallel roads) whilst at the same time maintaining an awareness of the position and movements of b. A way to characterise such occurrences of Follow would be to affirm that b is moving along a route or direction and a is moving within a boundary of b's route, such that the distance between a and b is approximately constant or within reasonable bounds (not too close, not too far).

The above characterisation introduces three elements that have to be formalised: the fact that a is moving along the same route of b, the fact that a is moving within the boundaries of b's route allowing for diversions, and the fact that the a tries to keep its distance from b within certain bounds (ideally, a desires to keep b within its line of sight).

Given objects o_1 and o_2, the predicate $\mathsf{followRoute}(o_1, o_2)$ holds on interval $[t_s, t_e]$ if o_1 is passing on the same route as o_2, by stating that for every instant $t \in [t_s, t_e]$, if o_2 is at position p_2 at instant t, then there exists a later instant $t_p > t$ such that o_1 is at p_2 at t_p. In general, t_p is thought to be both not too close and not too far in the future from t. The defnition below specifies two ways to delimit such bound: either there exist thresholds $T^{Follow}_{min\_time}$ and $T^{Follow}_{max\_time}$ respectively specifying the minimum and maximum duration of the interval $[t, t_p]$, or there exist thresholds $T^{Follow}_{min\_dist}$ and $T^{Follow}_{max\_dist}$ respectively specifying the minimum and maximum distance between the positions of o_1 and o_2 at t and at t_p:

$$
\begin{aligned}
&\mathsf{followRoute}[P](o_1, o_2)_{@[t_s,t_e]} \equiv \\
&\quad \forall\, t \in \mathcal{T}\, [t_s \leqslant t \leqslant t_e \wedge \mathsf{position}(o_2, p_2)_{@t} \rightarrow \\
&\qquad \exists\, t_p \in \mathcal{T}, (T^{follow}_{min\_time}, \delta^t_{min}), (T^{follow}_{max\_time}, \delta^t_{max}) \in P\, [t < t_p \wedge
\end{aligned}
$$

$$\delta^t_{min} \leqslant t_p - t \leqslant \delta^t_{max} \wedge \mathsf{position}(o_1, p_{1p})_{@t_p} \wedge p_{1p} = p_2] \vee$$
$$\exists\, t_p \in \mathcal{T}, (T^{follow}_{min\_dist}, \delta^d_{min}), (T^{follow}_{max\_dist}, \delta^d_{max}) \in P\, [t < t_p \wedge$$
$$\mathsf{position}(o_1, p_{1p})_{@t_p} \wedge p_{1p} = p_2 \wedge$$
$$\mathsf{position}(o_1, p_1)_{@t} \wedge \mathsf{distance}((p_1, p_2), d) \wedge \delta^d_{min} \leqslant d \leqslant \delta^d_{max} \wedge$$
$$\mathsf{position}(o_2, p_{2p})_{@t_p} \wedge \mathsf{distance}((p_{1p}, p_{2p}), d_p) \wedge \delta^d_{min} \leqslant d_p \leqslant \delta^d_{max}]] \tag{5.50}$$

The definition above is rigid in respect to the fact that o_2 has to find itself at the precise position previously occupied by o_1. This can suit the previously mentioned example where a vehicle is openly following another, but does not allow for subtler instances of Follow where o_2's movements do not strictly mirror o_1's.

A possibility would be to extend the definition of $\mathsf{followRoute}(o_1, o_2)$ by specifying that if o_2 is at a certain position p_2 at t, then o_1 will be within a certain boundary of p_2 at t_p. The predicate $\mathsf{followBoundary}(o_1, o_2)$ below introduces threshold T^{follow}_{bdry} to specify the maximum distance between o_1's position at t_p and the position occupied by o_2 at t:

$$\mathsf{followBoundary}[P](o_1, o_2)_{@[t_s,t_e]} \equiv$$
$$\mathsf{Occurs}(\mathsf{Move}[P](o_2), [t_s, t_e]) \wedge \forall\, t \in \mathcal{T}\, [t_s \leqslant t \leqslant t_e \wedge \mathsf{position}(o_2, p_2)_{@t} \rightarrow$$
$$\exists\, t_p \in \mathcal{T}, (T^{follow}_{min\_dist}, \delta^d_{min}), (T^{follow}_{max\_dist}, \delta^d_{max}), (T^{follow}_{bdry}, \delta_{bdry}) \in P, d_b, d, d_p$$
$$[t < t_p \wedge \mathsf{position}(o_1, p_{1p})_{@t_p} \wedge \mathsf{distance}((p_{1p}, p_2), d_b) \wedge d_b < \delta_{bdry} \wedge$$
$$\mathsf{position}(o_1, p_1)_{@t} \wedge \mathsf{distance}((p_1, p_2), d) \wedge \delta^d_{min} \leqslant d \leqslant \delta^d_{max} \wedge$$
$$\mathsf{position}(o_2, p_{2p})_{@t_p} \wedge \mathsf{distance}((p_{1p}, p_{2p}), d_p) \wedge \delta^d_{min} \leqslant d_p \leqslant \delta^d_{max}]] \tag{5.51}$$

The occurrence of the event $\mathsf{Follow}(o_1, o_2)$ over interval $[t_s, t_e]$ can now be formalised by specifying that either fluent $\mathsf{followRoute}(o_1, o_2)$ or fluent $\mathsf{followBoundary}(o_1, o_2)$ hold on the same interval. The definition below also introduces the predicate $\mathsf{sightDistance}(o_1, d)$, which holds if d is the maximum distance for an object to be within the line of sight of o_1, and specifies that if such a value exists, the threshold $T^{follow}_{max\_dist}$ in precisification P will assume value d:

$$\mathsf{Occurs}(\mathsf{Follow}[P](o_1, o_2), [t_s, t_e]) \equiv$$

$$[\exists\, d\, \mathsf{sightDistance}(o_1, d) \rightarrow \exists\, (T^{follow}_{max\_dist}, \delta) \in P\, [\delta = d]] \wedge$$
$$[\mathsf{followRoute}[P](o_1, o_2)_{@[t_s,t_e]} \vee \mathsf{followBoundary}[P](o_1, o_2)_{@[t_s,t_e]}] \quad (5.52)$$

5.3.2 Chase

The dictionary lists Chase as a transitive verb meaning "To pursue with a view to catching"[2]. In many ways, it can be seen as an occurrence of Follow with a further specification that the object behind is actively trying to reach the object in front. As for Follow, the object being chased may or may not be aware of the object performing the chase, and similarly the latter may be more or less conspicuous in the act of chasing.

Most of the considerations in the previous section about the ways in which an object a can follow another object b apply to Chase as well. However, because the meaning of Chase suggests that a intends to ultimately reach b's position, and at the same time that b is fleeing or moving away from a, there has to be a specific characterisation of this intention. Such behaviour appears particularly difficult to identify through the analysis of a and b's motion patterns. For example, one may argue that a is effectively chasing b if a is following b and the distance separating them progressively decreases through the interval over which the chase occurs. However, there may be instances where such distance reduction does not happen due to the chase being ineffective. In fact, if a is frantically following b and at the same time the distance between them increases, an observer would probably understand that b manages to flee from a, but also that a would still be chasing b. This consideration leads to the conclusion that a reduction in the distance between a and b over the course of a chase is too specific for the definition of Chase.

Another aspect that may characterise a's intentional behaviour suggested by the verb is that a is taking shortcuts or moving in such a way in order to gain an advantage over b. Unfortunately, identifying such behaviour would require extensive knowledge of the environment in which a and b are moving, and of the strategies that a may put in place to accomplish its goal of catching b.

The considerations above point out that the verb Chase has a strong intentionality component which is not easily characterisable by analysing the two objects' motion. A compromise may be reached by defining an

occurrence of $\mathsf{Chase}(o_1, o_2)$ over $[t_s, t_e]$ as an occurrence of $\mathsf{Follow}(o_1, o_2)$ over the same interval where object o_1 is moving fast:

$$\begin{aligned}&\mathsf{Occurs}(\mathsf{Chase}[P](o_1, o_2), [t_s, t_e]) \equiv \\ &\quad \mathsf{Occurs}(\mathsf{Follow}[P](o_1, o_2), [t_s, t_e]) \wedge \mathsf{speed}[P](o_1, \mathsf{fast})_{@[t_s, t_e]}\end{aligned} \tag{5.53}$$

Of course the above definition is not particularly general or complete, as there may be instances where o_1 is chasing o_2 by slowly and carefully watching o_2's moves, or where o_1 stops or slows down during the process. Nevertheless, the fact that an occurrence of Follow is performed at a fast speed is probably a good indication that such occurrence is in fact a Chase.

5.3.3 Flee

The dictionary lists Flee as a verb that can be used transitively to mean "To run away from or as from danger; to take flight; to try to escape or seek safety by flight"[2].

An occurrence of the transitive form of flee, formalised as $\mathsf{Flee}(o_1, o_2)$, would be characterised by the fact that object o_1 is trying to escape or distance itself from object o_2 perceived as a danger. A common example would be given by the fact that o_2 is chasing o_1, or that o_1 begins a fast motion away from o_2, at which point o_2 may or may not initiate a chase.

$$\begin{aligned}&\mathsf{Occurs}(\mathsf{Flee}[P](o_1, o_2), [t_s, t_e]) \equiv \\ &\quad \mathsf{Occurs}(\mathsf{MoveAwayFrom}(o_1, o_2), [t_s, t_e]) \wedge \mathsf{speed}[P](o_1, \mathsf{fast})_{@[t_s, t_e]}\end{aligned} \tag{5.54}$$

There are occurrences in which the verb is used intransitively, as in $\mathsf{Flee}(o)$. This is likely to refer to instances in which object o is fleeing from something unspecified. The object perceived as danger may or may not be present in the scene under consideration. A possible definition for this occurrence is by stating that there exist an object o_d which o is fleeing from.

$$\mathsf{Occurs}(\mathsf{Flee}[P](o), [t_s, t_e]) \equiv \exists\, o_d(\mathsf{Occurs}(\mathsf{Flee}[P](o, o_d), [t_s, t_e])) \tag{5.55}$$

However, it is possible to conceive examples where o is fleeing a scene for no apparent or observable reason or trigger.

5.4 Contact

The verbs in this section describe events which involve objects being in or establishing a static or dynamic form of contact with each other.

The section starts by examining Touch, the most generic of the set, and moves forward by analysing Push, a specific type of occurrence where the act of establishing contact with an object has the effect of causing a subsequent act of motion. For verbs Hit and Collide the contact is established in a forceful and/or disruptive way, and for Kick contact involves a specific body part. For Hold contact serves the purpose of supporting an object and fixing its relative position.

The analysis of the events briefly listed above involves various factors, such as identifying specific body parts, establishing when an act of contact begins and ends, and also characterising the forces involved in the dynamic contact between two objects, and under which circumstances these can be consider forceful or disruptive. This latter aspect is particularly prevalent for verbs Hit and Collide. The formalisation of Hit also involves the analysis of whether it is possible or sensible to identify one particular object as performing an active or intentional role in the action.

5.4.1 Touch

The verb Touch can be used in many concrete and figurative instances to express the fact that an object is in contact with another. The dictionary lists it as a transitive verb with several specifications of such meaning, the most relevant of which are "To put the hand or finger, or some other part of the body, upon, or into contact with (something); to touch (a thing) with the hand or other part, or with some instrument; to come into, or be in, contact with; to be in contact with, or immediately adjacent to" [2]. From this it follows that most of the concrete occurrences of Touch will involve two objects being in contact with each other; with further specifications of such manifestations when people are involved in the event.

There are two main interpretations shaping the temporal nature of Touch. One of these considers an occurrence of Touch as a *static event* or *state*, and the event-type $\mathsf{Touch}(o_1, o_2)$ occurring on a temporal interval where objects o_1 and o_2 are in contact. For example, if two boxes are positioned in such a way that one of the boxes' side is immediately above,

below, to the left or to the right of the side of the other box, the event would be occurring on the entire interval in which the boxes are in this position. In general, the formalisation of this interpretation inolves individuating the edges or boundaries of the two objects and specifying that the boundaries are in the same spatial location. Referring to the topology of the two objects, this corresponds to establishing whether they are externally connected. This interpretation can be formalised as the fluent touch that holds over an instant t if two objects o_1 and o_2 are connected. The following simple definition specifies that $\mathsf{touch}(o_1, o_2)$ holds at time t if o_1 and o_2 are externally connected:

$$\mathsf{touch}(o_1, o_2)_{@t} \equiv \mathsf{rcc\_EC}(o_1, o_2)_{@t} \tag{5.56}$$

The other interpretation considers Touch as a *dynamic event*, with the event-type $\mathsf{Touch}(o_1, o_2)$ occurring over an extended interval. An occurrence of $\mathsf{Touch}(o_1, o_2)$ generally has o_1 playing the active role in the action (meaning that subject o_1 touches object o_2 rather than vice versa). For example, person o_1 raises his hand and lays it on person o_2's shoulder, or person o_1 puts his hand against door o_2. Often the active object can be identified by analysing the degree of dynamicity of the two, with the trivial case having o_1 as dynamic and o_2 static (more advanced measures are imaginable, e.g. degree of intentionality). This may not always be the case, for example in occurrences where both objects are equally dynamic with no particular prominence on o_1 or o_2, such as two people o_1 and o_2 walking towards each other placing each other's hand on each other's shoulder at the same time. In this instance, it would be reasonable to have both event-types $\mathsf{Touch}(o_1, o_2)$ and $\mathsf{Touch}(o_2, o_1)$ occurring on the same interval.

Another aspect of occurrences of Touch involves identifying specific and predictable contact parts through which an object o_1 comes into contact with o_2. This is particularly true for objects of type person; people tend to touch things by using their hands or fingers. It follows that the formalisation of an instance of $\mathsf{Touch}(o_1, o_2)$ where o_1 is a person will generally specify the position and the motion of the person's hands towards another object. Instances in which these contact parts exist but the contact does not happen through them, as one may predict, are generally un-characteristic and it is likely that they may be classified as occurrences of other, more

specific contact verbs. For example, if a person o_1 is walking in space and comes into contact with a lamp post through the body trunk, this is generally classified as an occurrence of Hit or Collide rather than Touch. Some other instances may be more subtle, for example if a person o_1 is putting his/ her foot against an object o_2, this may be classified as an occurrence of Kick if the motion is fast and o_2 moves away from o_1 as a result of the action (for example o_2 being a ball), or as an occurrence of Touch if o_1's motion is slow and/or o_2 is static (for example o_2 being a wall).

The temporal extension of the dynamic interpretation of Touch is hard to establish. This issue has been encountered for many other verbs in this chapter, and it relates to the lack of precise and objective criteria identifying the instants in which the occurrence starts and ends. A very strict physical interpretation of $\mathsf{Touch}(o_1, o_2)$ as a punctual event would have the occurrence spanning an interval containing only the single instant in which o_1 and o_2 become connected. Despite the advantage of clearing any ambiguity, such an interpretation is rather unrealistic as an observer would generally assign a duration to the event.

Some interpretations may regard an occurrence of $\mathsf{Touch}(o_1, o_2)$ as terminating just on or very shortly after the instant in which o_1 has established contact with o_2, regardless of the fact that the *state* of o_1 being in contact with o_2 may persist after such instant. Other interpretations instead may regard the occurrence as extending beyond that instant and throughout the entire time in which o_1 and o_2 are in contact. Given that the interpretation of Touch presented here focuses on the dynamic event in which two objects come into contact, the formalisation will concentrate on the former type, where the occurrence does not persist after the dynamic part of the event.

Establishing a starting instant for an occurrence of $\mathsf{Touch}(o_1, o_2)$ is even harder. A possibility would be to identify the start of the particular aspect of o_1's motion that results into o_1 establishing contact with o_2, which should not occur too far in the past prior to the contact. For example, if o_1 and o_2 are two people side by side and o_1 lays a hand on o_2's shoulder, the event occurrence would start on the instant in which o_1's hand initiates movement towards o_2, and terminate when the hand is in contact with the shoulder. Whilst this interpretation may work reasonably well for objects which are static prior to the event, it may prove more diffi-

cult to apply to more dynamic objects that are already performing acts of motion prior to coming in contact with each other. For example, a ball may have been rolling on the ground for a while and slowly come into contact with another object. In this case, because of the impossibility of establishing a clear start of the motion that led the ball to touch the object, a punctual interpretation of Touch may seem appropriate. However, other interpretations may still be reasonable. For example, $\mathsf{Touch}(o_1, o_2)$ may start a predefined amount of time before the contact (e.g. 1 or 2 seconds), or when o_1 is *very near* o_2.

The following definition is a disjunction between two formalisations of an occurrence of $\mathsf{Touch}(o_1, o_2)$ over interval $[t_s, t_e]$. The first disjunct states that if object o_1 is a person then one of his/her hands will move towards o_2 over $[t_s, t_e]$, the second disjunct instead limits the width of $[t_s, t_e]$ to a reasonably small threshold T_{Touch} in precisification P and specifies that o_1 moves towards o_2 over $[t_s, t_e]$. In both cases, the definition states that o_1 and o_2 come into contact at t_e and such contact must persist for at least an interval following t_e.

$$
\begin{aligned}
&\mathsf{Occurs}(\mathsf{Touch}[P](o_1, o_2), [t_s, t_e]) \equiv \\
&\quad [\mathsf{type}(o_1, \mathsf{Person}) \wedge \exists\, t_1, t_2 \in \mathcal{T} [t_1 = succ(t_e) \wedge t_1 < t_2 \wedge \\
&\quad \exists\, h_l, h_r\, [\mathsf{p\_hands}(o_1, (h_l, h_r)) \wedge \neg \mathsf{touch}(o_1, o_2)_{@[t_s, t_e]} \wedge \\
&\qquad [[\mathsf{moveTowards}(h_l, o_2)_{@[t_s, t_e]} \wedge \mathsf{touch}(h_l, o_2)_{@[t_1, t_2]}] \vee \\
&\qquad\ [\mathsf{moveTowards}(h_r, o_2)_{@[t_s, t_e]} \wedge \mathsf{touch}(h_r, o_2)_{@[t_1, t_2]}]] \ ] \vee \\
&\quad \exists\, (T_{Touch}, \varepsilon) \in P\, [t_e - t_s < \varepsilon \wedge \neg \mathsf{touch}(o_1, o_2)_{@[t_s, t_e]} \wedge \\
&\qquad \mathsf{Occurs}(\mathsf{Move}[P](o_1), [t_s, t_e]) \wedge \mathsf{touch}(o_1, o_2)_{@[t_1, t_2]}]]
\end{aligned}
\tag{5.57}
$$

5.4.2 Push

The meaning most likely to match the occurrences of Push observed within the domain of this work is the transitive use of the verb meaning "to exert force upon or against (a body) so as to move it away; to move along by exerting a continuous force; to move forward or advance (a force) against opposition or difficulty"[2]. The rest of this section ignores other more or less figurative meanings of the verb, even though the characterisation of a subject's own behaviour, such as "to make one's way with force or persistence", would look interesting to investigate.

From the above it can be gathered that an occurrence of $\texttt{Push}(o_1, o_2)$ is characterised by a subject o_1 exerting a force against another object o_2, with the expected outcome being the motion of o_2 in a direction away from o_1. For example, a person may push a box by placing his/her hands on the item and pressing against it, for the purpose of moving it towards a particular position.

However, the above characterisation is slightly too specific, as there are instances in which the action performed by o_1 does not have the result of causing o_2 to move, or at least not in the way one would expect. For example, if a box is anchored to the ground, the subject's actions cause no actual displacement, except for possibly a slight deformation of the box edges. If a person is pushing against a door in order to cause it to swing open and the door is locked, trivially the door will remain in position with no observable movement.

Taking the above into account, a more general characterisation may specify that o_1 is exerting a force on o_2 with the intention of causing o_2 to move, even if the occurrence of Push is not followed by the expected movement of o_2. Identifying this is not trivial, as an object o_1 exerting a force against o_2 may appear static to an observer. However, it generally happens that o_1 will move itself, or part of itself, towards o_2 over an interval preceding the instant in which the force starts being exerted (e.g. a person pushing a door will lift his/her hand, move it towards the door and place it on the door frame or handle). Moreover, the nature, type or size of o_2 could aid to establish whether o_2 can be moved by being pushed. In fact, objects can be classified in three broad categories in respect to this capability:

- *Immovable* objects, whose size or type inhibit their transition to a state of motion as a result of someone or something pushing on them (e.g. a house, a wall, a lamp post...).
- *Movable* objects, whose size, type and absence of physical constraints allow them to start to move if subject to an external force (e.g. a vehicle on wheels, an unanchored box, an unlocked door, a person...)
- *Potentially movable* objects. This category is constituted by almost all movable objects which had constraints put on them to prevent movement, which could nevertheless still happen under disproportionate

force (e.g. a car with a handbrake on, an anchored box, a locked door...)

There are further aspects to consider too. In some instances, o_2 may be a very small or particularly constrained object resulting in an almost negligible movement following a push (e.g. a person pushing a button to call a lift would result in the button slightly moving away from the person given the groove it is sliding into). Also, most instances are likely to see o_1 exerting force on o_2 through specific or predictable contact parts. As for Touch, this is particularly true for people, which generally exert force onto objects by placing their hands against the side of the object closer to them, or another specific part of the object that is intended for pushing (e.g. a door handle, or trolley's arm). However, there may exist instances where the force is exerted in an unconventional way, (e.g. using feet or the entire body trunk). Finally, an effect that may follow an occurrence of o_1 pushing o_2 is the deformation of o_2's shape. Identifying such deformation may prove useful in instances where o_2 does not move as a result, as it would constitute evidence of the force exerted by o_1 against o_2. Immovable objects are unlikely to change shape, and it is potentially movable objects that are the most likely to show deformation, depending on their rigidity and quality of their anchors or other constraints inhibiting their movement.

Regarding the temporal extension of an occurrence of $\mathsf{Push}(o_1, o_2)$, its starting point can be established at the instant o_1 begins to exert the force, and its ending can be established at the instant o_1 ceases to exert such force. Given the considerations above about the ways in which such a force can be exerted and the effects it may cause, a general formalisation would state that the event occurs over $[t_s, t_e]$ if o_2 is a movable or potentially movable object, o_1 is in contact with o_2 at t_s, this contact is maintained throughout $[t_s, t_e]$ and either o_2 moves away from o_1 over $[t_s, t_e]$ (and possibly beyond t_e) or there is a deformation of o_2's shape over $[t_s, t_e]$.

$$
\begin{aligned}
&\mathsf{Occurs}(\mathsf{Push}[P](o_1, o_2), [t_s, t_e]) \equiv \\
&\quad \exists, t_1, t_2, t_3 [t_1 < t_s \leqslant t_2 \wedge t_e \leqslant t_3 \wedge \\
&\quad \mathsf{Occurs}(\mathsf{Touch}[P](o_1, o_2)[t_1, t_s]) \wedge \mathsf{touch}(o_1, o_2)_{@[t_s, t_e]} \wedge \\
&\quad [[\mathsf{movable}(o_2, \mathrm{movable})_{@[t_s, t_3]} \wedge \mathsf{Occurs}(\mathsf{MoveAwayFrom}(o_2, o_1), [t_2, t_3])] \vee
\end{aligned}
$$

$$[\mathsf{movable}(o_2, \text{partMovable})_{@[t_s,t_3]}] \vee [\mathsf{Occurs}(\mathsf{deform}(o_2), [t_2, t_e])]]] \tag{5.58}$$

The above definition introduces the event $\mathsf{deform}(o)$ which holds at any time instant t at which object o is subject to a deformation of its shape. This event is currently not defined and an implementation of its definition would be likely to characterise the process of deformation by analysing changes in the shape or area occupied by o.

Definition 5.58 lacks an actual characterisation of the fact that o_1 is exerting some form of force directed or impressed at o_2. This is a rather high-level concept which is hard to identify by observation, hence the definition has focused on formalising Push by specifying a more observable sequence of events.

5.4.3 Collide

The verb Collide is semantically very similar to the verb Hit, discussed in the next section. In fact, it is listed by the dictionary as a transitive verb meaning "to bring or come into collision or violent contact, strike or dash together" [2]. In many ways, it is a specialisation of Touch where the movement of the two objects is fast, sudden, forceful and/or the forces involved have disruptive consequences. Force and speed are the main characterising themes.

As opposed to the characterisation of $\mathsf{Hit}(o_1, o_2)$ of Sec. 5.4.4, where it is argued that object o_1 plays an active role in the action, occurrences of Collide generally do not exhibit greater importance, effort or intentionality on part of either of the objects participating in the action. From this it follows that most occurrences of Collide are symmetrical; if an occurrence of $\mathsf{Collide}(o_1, o_2)$ happens at a particular interval, then it is generally also true that $\mathsf{Collide}(o_2, o_1)$ happens on the same interval.

The assumptions above seem to agree with the ways the words 'collide' and 'collision' are generally used in natural language. They are formal words referring to the abstract notion of a disruptive contact rather than the particular way in which it happened or its consequences. This semantics is not always clear and can be subtle. For example, police or emergency services attending the scene of road accident often refer to the event as a 'road traffic collision', referring to the fact that two or more

vehicles collided. It is unlikely for such an occurrence of Collide to be symmetrical with no greater blame or responsibility placed on any one vehicle; accidents indeed involve some kind of wrongdoing however subtle this may be (except in very rare cases, for example a bug in the traffic light control system which gives simultaneous green lights to conflicting traffic paths). The semantics of the words 'collision' and 'collide' deliberatly avoid the attribution of any form of active role or responsibility on any one of the subjects involved in the event. This is often appropriate, as it serves the purpose of focusing on the disruptive nature of the event and its serious consequences that have to be dealt with by the emergency services. At a later stage, investigations are aimed at individuating any blame or responsibility, thus attributing a particularly active role on one subject and allowing for a more specific utterance such as Hit.

In general, the formalisation of an occurrence of $\mathsf{Collide}(o_1, o_2)$ would specify the existence of high and/or disruptive forces in the instants surrounding the contact between the two objects:

$$\begin{aligned}
&\mathsf{Occurs}(\mathsf{Collide}[P](o_1, o_2), [t_s, t_e]) \equiv \\
&\quad \mathsf{Occurs}(\mathsf{Touch}[P](o_1, o_2), [t_s, t_e]) \wedge \\
&\quad [\mathsf{speed}[P](o_1, \mathsf{fast})_{@[t_s,t_e]} \vee \mathsf{speed}[P](o_2, \mathsf{fast})_{@[t_s,t_e]} \vee \\
&\quad \mathsf{disruptiveForce}(o_1)_{@t_e} \vee \mathsf{disruptiveForce}(o_2)_{@t_e}]
\end{aligned} \tag{5.59}$$

The definition above states that $\mathsf{Collide}(o_1, o_2)$ occurs over interval $[t_s, t_e]$ if $\mathsf{Touch}(o_1, o_2)$ occurs over the same interval and either of the two objects is fast or causes some disruptive force formalised by the predicate disruptiveForce. This is a high level concept whose definition is hard to formalise, especially by referring to observable properties. Some clues may be given by changes in the state of o_1, o_2 or both: if a car collides with a wall, the car will have an altered shape, and possibly the wall as well if the impact force is great enough; however, if a person collides with a wall, there are generally no noticeable changes in the shape or motion of the wall.

5.4.4 Hit

The dictionary lists Hit as a transitive verb meaning "to reach or get at with a blow; to give a blow to (something aimed at); to strike with aim or intent; (of a missile or moving body): to come upon with forcible impact; to strike"[2]. An occurrence of the verb is generally characterised by the presence of a moving subject coming into contact with a direct object at a high force or speed.

In many ways, occurrences of Hit can be framed as specific instances of the more general verbs Touch, describing the act of an object coming into contact with another, and/or Push, itself a specification of Touch whose effect is to cause movement or deformation in the object being touched. A formalisation of $\mathsf{Hit}(o_1, o_2)$ is likely to describe it as specialising Touch with a movement that could be described as fast, sudden and/or forceful.

In particular, this verb bears a very close similarity with the verb Collide discussed in the previous section. Both verbs describe the forceful and potentially disruptive act of an object clashing with another, however it seems that an occurrence of $\mathsf{Hit}(o_1, o_2)$ conveys the fact that subject o_1 has *more involvement* or is *more active* than o_2. For this reason, and given the discussion and formalisation of Collide in Sec. 5.4.3, it seems appropriate to consider Hit as a particular specialisation of Collide where the action is not symmetric and o_1 is the active subject.

Such active role of o_1 in the action could be described by several characteristics. One of the most trivial of these is the case in which object o_2 is static, for example a ball or a missile hitting the ground, or a car hitting a wall. In this instance, o_1 is the only moving object and the act is characterised by its movement directed towards o_2 which suddenly stops when o_1 and o_2 come into contact. In some instances, there may be an effect or consequence causing changes in the state of o_2. For example, if the force generated by car o_1 hitting wall o_2 is very high, it could result in a deformation of o_2. Similarly, if o_1 is the white ball that has just being set in motion by a player on a pool table, and it comes into contact with ball o_2 in its path, the consequence of $\mathsf{Hit}(o_1, o_2)$ is for o_2 to start moving in a particular direction, and possibly o_1 diverting its motion as well.

The active role played by o_1 may also be conveyed by the fact that some occurrences of $\mathsf{Hit}(o_1, o_2)$ involve specific contact parts. For example, if a

person hits another person with his fist, the active subject would be the one whom the hand that came in contact with the person belongs to.

Another characterisation of the active role played by o_1 may refer to a certain *intentionality* of o_1 in causing a particular effect of o_2, which is particularly true for animate objects such as people, or items capable of generating force and/or propelling themselves such as vehicles, many of which are directly controlled by people anyway. For example: if car o_1 runs a red light and subsequently clashes with car o_2, it would be o_1 (or, rather, his driver) that deliberately caused the action by crossing o_2's path despite not being allowed to do so. In this instance the occurrence would be described by $\mathsf{Hit}(o_1, o_2)$ rather than $\mathsf{Hit}(o_2, o_1)$. The degree of intentionality of an object in the act of performing a particular action is a very high level concept whose semantics are not easily captured by observation.

An occurrence of $\mathsf{Hit}(o_1, o_2)$ can now be formalised as a specific occurrence of Touch where motion is fast. Below, o_1 is characterised as playing the active role by stating that either o_2 is static or o_1 is moving faster than o_2 (predicate faster can be defined by comparing the two object's speeds, see pag. 81):

$$\begin{aligned}&\mathsf{Occurs}(\mathsf{Hit}[P](o_1, o_2), [t_s, t_e]) \equiv \\ &\quad \mathsf{Occurs}(\mathsf{Collide}[P](o_1, o_2), [t_s, t_e]) \wedge \mathsf{speed}[P](o_1, \mathsf{fast})_{@[t_s,t_e]} \wedge \\ &\quad [\neg \mathsf{move}[P](o_2)_{@[t_s,t_e]} \vee \mathsf{faster}(o_1, o_2)_{@[t_s,t_e]}]\end{aligned} \tag{5.60}$$

The formalisation above simplifies and/or ignores several aspects regarding contact parts, intentionality or consequences of the occurrence on the state of o_2 (such as movement, deformation, etc). A refined formalisation should refer to the concept of *force* and characterise occurrences of Hit as motions in which the force transferred from o_1 towards o_2 is a significant amount and with specific consequences.

5.4.5 Kick

The verb Kick is listed by the dictionary as a transitive verb meaning "to strike (anything) with the foot; to impel, drive, or move, by or as by kicking"[2]. This meaning specifically identifies a person as the subject performing the action, given that a foot is a body part that is only attached to people.

An occurrence of Kick can be characterised as a specific occurrence of Hit, itself a specialisation of Touch. In fact, $\mathsf{Kick}(o_1, o_2)$ generally describes the fact that subject o_1 is actively and, often, intentionally touching or pushing o_2 with significant force by moving his/her foot towards o_2.

In most instances, consequences of o_1 kicking o_2 are very similar to the consequences described in the discussion and formalisation of Push in Sec. 5.4.2. If o_2 is a movable object then it is expected that o_2 will move away from o_1 or alter his shape by deformation. However, it is conceivable for an occurrence of Kick to describe o_1 kicking an immovable object o_2, a situation not considered in the formalisation of Push. In this situation, there is generally no deformation or other observable consequence in o_2's state.

The following definition states that $\mathsf{Kick}(o_1, o_2)$ occurs over interval $[t_s, t_e]$ if o_1 is a person and it occurs that one of his feet hits object o_2 over the same interval and, additionally, either it occurs that $\mathsf{Push}(o_1, o_2)$ or o_2 is an immovable object:

$$\begin{aligned}
&\mathsf{Occurs}(\mathsf{Kick}[P](o_1, o_2), [t_s, t_e]) \equiv \\
&\quad \mathsf{type}(o_1, \mathsf{Person}) \wedge \exists\, f_l, f_r [\mathsf{p_feet}(o_1, (f_l, f_r)) \wedge \\
&\quad [\mathsf{Occurs}(\mathsf{Hit}[P](f_l, o_2), [t_s, t_e]) \vee \mathsf{Occurs}(\mathsf{Hit}[P](f_r, o_2), [t_s, t_e])] \wedge \\
&\quad [\mathsf{Occurs}(\mathsf{Push}[P](o_1, o_2), [t_s, t_e]) \vee \mathsf{movable}(o_2, \mathrm{immovable})_{@t_s, t_e}] \qquad (5.61)
\end{aligned}$$

As already mentioned for Hit and Collide, a refined characterisation of Kick would take the forces involved in the event into account, by specifying the nature and characteristics of a forceful impact and the effects this has on o_2. However, there may be instances of Kick where the impact is not regarded as particularly forceful, such as a person gently kicking a ball with no particular energy expenditure.

5.4.6 Hold

The dictionary lists Hold as a transitive verb meaning "to keep from getting away; to keep from falling, to uphold, sustain, support or maintain in or with the hand, arms, etc.; to have or keep within it"[2].

From the above it follows that general occurrences of $\mathsf{Hold}(o_1, o_2)$ are constituted by an object o_1 whose actions or state prevent o_2 from mov-

ing away from o_1. More specific occurrences would classify the active or passive nature of o_1 and/or o_2.

In fact, o_1 may be intentionally and actively holding o_2 from moving away from itself, or may be passively holding o_2 due to its persistent state. Similarly, o_2 may be actively or passively moving away from o_1, as o_2 may be capable of propelling itself in order to move away from o_2, or may be subject to an external force that would cause such movement (e.g. gravity). For example, a picture hook attached to a wall holding a picture frame in place is an example where the persistent state of the (passive) picture hook is holding the frame in position, which would otherwise be subject to gravity thus be drawn away from the hook. Conversely, a policeman holding a thief after a chase is actively performing specific actions aimed at keeping the other person in position, which would otherwise actively propel itself in order to get away.

There are also situations where there is no immediate possibility for o_2 to move away from o_1 but still there is the potential for o_2 to do so. For example, let us consider a person o_1 holding a ladder o_2 set against a wall on which another person o_3 is standing on. If the ladder is reasonably stable, o_2 would not normally move away from o_1 and/or o_3, however the context may indicate a relatively high probability for this to happen, hence o_1 is holding o_2 to prevent that eventuality. This example also shows that occurrences of Hold could be subject to a form of transitive relation. In fact, person o_1 is holding ladder o_2 in place, and ladder o_2 is holding o_3 in place too. One could then argue that a reasonable interpretation of this situation would also result in o_1 holding o_3 in place via o_2.

Most passive occurrences of Hold are generally characerised as a static event, where objects establish contact and maintain it throughout the event occurrence, with an almost constant relative distance between them. Subjects may be completely static, as in the picture frame example above, or dynamic, for example a person walking and holding a set of keys in his hand. The theme these occurrences have in common is that the relation and relative positioning of the two objects is not subject to major variations over the interval.

The discussion to follow focuses on static occurrences of $\mathsf{Hold}(o_1, o_2)$, where o_1 is a person holding another person or object o_2 through specific contact parts of o_1, generally hands or fingers. Specific contact parts of o_2

may be also involved, such as an item handles or its outer edges, however this may sometimes be difficult to identify for particular objects, such as the ladder in the example above. The definition below formalises the fluent $\mathsf{hold}(o_1, o_2)$ as holding at time t if o_1 is a person, its left and right hands are h_l and h_r, o_2 is an object that can be held and it is either the case that one of o_1's hands are touching o_2 at time t:

$$\begin{aligned}&\mathsf{hold}(o_1, o_2)_{@t} \equiv \\ &\quad \exists\, h_l, h_r[\mathsf{type}(o_1, \mathsf{Person}) \wedge \mathsf{p\_hands}(o_1, (h_l, h_r)) \wedge \mathsf{holdable}(o_2)_{@t} \wedge \\ &\quad [\mathsf{touch}(h_l, o_2)_{@t} \vee \mathsf{touch}(h_r, o_2)_{@t}]\end{aligned} \tag{5.62}$$

The predicate $\mathsf{holdable}(o)$ should specify whether an object o has certain characteristics that allow it to be held by a person at a particular time. These could be a combination of his size, shape and or the fact that o_2 at time t would actually or potentially move. Refined definitions would characterise the specific ways in which a person's hand or fingers hold particular objects, a challenging task given the variety of shapes and possibilities for people to position their fingers on an object in order to establish effective contact. These issues are ignored in the formalisation of event-type $\mathsf{Hold}(o_1, o_2)$ below, where it is simply specified that it occurs over interval $[t_s, t_e]$ if fluent $\mathsf{hold}(o_1, o_2)$ holds throughout the interval:

$$\mathsf{Occurs}(\mathsf{Hold}(o_1, o_2), [t_s, t_e]) \equiv \mathsf{hold}(o_1, o_2)_{@[t_s, t_e]} \tag{5.63}$$

In Sec. 6.2.3 an implementation of a definition of Hold in the logic-programming system ProVision is proposed, where the occurrence of the event is inferred by estimating a 'holding position' in which an object being held is most likely to be located.

Chapter 6

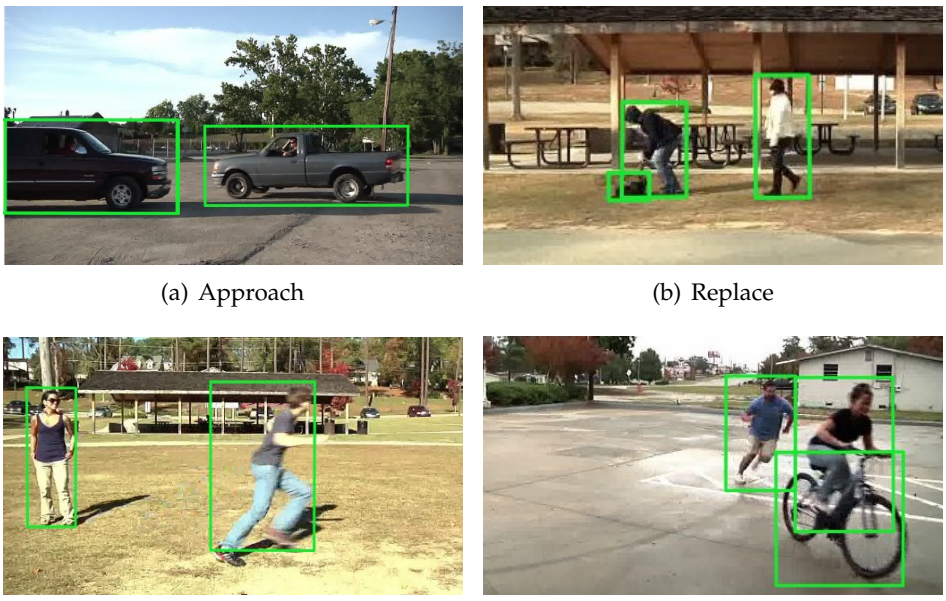
Event Recognition

Our event detection system ProVision is a logic-programming implementation of the formal ontology emerging from Chapters 4 and 5. It is designed as a module of a wider framework for event analysis and detection, whose input is a video and whose output is a high-level description of the events occurring in it. Within this framework, the initial processing of video frames, not described in this paper, is performed by trackers and classifiers that output a structured description of the relevant objects. The resulting data represents the information grounding the ontology, and it is described in Sec. 6.1. ProVision infers higher-level predicates defined in the ontology and produces a list of event occurrences detailing which events occur in each video. Certain technical implementative aspects are presented in Sec. 6.2, where some spatio-temporal concepts are defined and two sample verbs are modeled. Section 6.3 describes the result of some experimental recognition tests, discussed in Sec. 6.4.

6.1 Source Data

The video sequences which constitute the data for the implementation and evaluation of our formalism have been provided by DARPA as the development video dataset. This dataset contains 1302 video sequences in MPEG format, hereafter called *vignettes*, with a resolution of 1280x720 pixels and variable duration, generally between 5 and 20 seconds.

Portrayed subjects are mostly people, vehicles (cars, bicycles and motorbikes) and other objects such as boxes, balls, small items and sometimes



(a) Approach (b) Replace

(c) Flee (d) Chase

Figure 6.1: Vignettes and annotated objects

animals. The scene background is a generic urban or semi-urban outdoor environment, such as streets, recreation parks or car parks. The camera filming the scene is in a frontal, central position and is static throughout the scene, recording 25 frames per second. Each vignette is meant to represent the occurrence of a specific motion verb, included in the video file name, even though other motion verbs may be represented in the scene as well. There are generally 10 to 30 vignettes per motion verb. Fig. 6.1 shows some example vignettes and the associated verb for each vignette.

The automatic event recognition system ProVision does not operate on the actual video files, but on an *annotation* of the video. These annotations are constituted by XML files in Viper format [39, 66, 74] called *vignette annotations*, one per video file. Each annotation file is structured in a first part containing general information about the video, a second part containing information about the *tracked objects* in each video and a third part containing *event annotations*.

Listing 6.1 on page 136 shows the relevant sections of the XML annotation file in Viper format describing the vignette in Fig. 6.1(a):

- The first part of the file describes the number of frames in the vi-

gnette (220), the width and height in pixels (1280 by 720) and other information not reported in the example.

- The second part describes *tracked objects*, recognised by either a manual or automatic annotator whose position is represented at each video frame. In our annotations the position is represented as a rectangular *bounding box*, illustrated in Fig. 6.1 by the green rectangles. The file shows two objects, each started by the tag `<object>` with the following properties:
  - The first object is present at frames 91 to 219, has id 0 and is of type vehicle (attributes `framespan`, `id` and `name`). The position is represented by a set of bounding boxes (tags `data:bbox`) each of which is described at each frame with top left coordinate, width and height (attributes `framespan`, `x`, `y`, `width` and `height`). For example the position of this object at frame 91 is represented by box with coordinates $(1021, 325)$, width 259 and height 238.
  - Similarly, the second object is present from frame 121 to 219, has id 1 and is of type vehicle. Its position at frame 121 is represented by bounding box with coordinates $(1, 317)$, width 257, height 260
- The third part of the file describes *event annotations*, started by tag `<content ... name="Event">`. These state which events occur in the vignette and is referred as *ground truth* for evaluation purposes (see also Sec. 6.3). For example, the file contains information about the fact an event of type `approach` occurs between frames 121 and 219 with participants `Vehicle:0` and `Vehicle:1`. An event participant is represented by a string formed by its type, a colon and its id number.

The information shown in Listing 6.1 is parsed by the ProVision system in order to produce an equivalent representation more suitable for the Prolog language. Each vignette annotation is represented by the Prolog fact `annotationset(+X)` where `X` is an associative list such as the one shown in Listing 6.2. This list characterises the file uniquely (property `input`), specifies the framespan and each object with the term `tracklet`. Each tracklet term is composed of a list specifying `id`, which also contains the type of object, and a list of bounding boxes at each frame. The event

```
<attribute name="NUMFRAMES">
  <data:dvalue value="220"/>
</attribute>
<attribute name="H-FRAME-SIZE">
  <data:dvalue value="1280"/>
</attribute>
<attribute name="V-FRAME-SIZE">
  <data:dvalue value="720"/>
</attribute>
<object framespan="91:219" id="0" name="Vehicle">
  <attribute name="bbox">
  <data:bbox framespan="91:91" height="238" width="259" x="1021" y="325"/>
   <data:bbox framespan="92:92" height="238" width="263" x="1017" y="325"/>
   ...
   <data:bbox framespan="219:219" height="207" width="196" x="714" y="341"/>
  </attribute>
</object>
<object framespan="121:219" id="1" name="Vehicle">
  <attribute name="bbox">
   <data:bbox framespan="121:121" height="260" width="257" x="1" y="317"/>
   <data:bbox framespan="122:122" height="260" width="264" x="1" y="317"/>
   ...
   <data:bbox framespan="219:219" height="260" width="694" x="-1" y="317"/>
  </attribute>
</object>
<content framespan="122:219" id="0" name="Event">
  <attribute name="name">
    <data:svalue value="approach"/>
  </attribute>
  <attribute name="subjects">
    <data:svalue value="Vehicle:0,Vehicle:1"/>
  </attribute>
</content>
<content framespan="122:219" id="5" name="Event">
  <attribute name="name">
    <data:svalue value="approach"/>
  </attribute>
  <attribute name="subjects">
    <data:svalue value="Vehicle:1,Vehicle:0"/>
  </attribute>
</content>
<content framespan="214:219" id="1" name="Event">
  <attribute name="name">
    <data:svalue value="arrive"/>
  </attribute>
  <attribute name="subjects">
    <data:svalue value="Vehicle:0"/>
  </attribute>
</content>
```

Listing 6.1: Sample annotation file – XML

```
annotationset(
    [ stage         = 'probase.pl',
      input         = input([dataset=1,vignette_id=1,origin=ground_rev]),
      framespan     = span(1,220),
      tracklet_list = [
        tracklet( [ id                = 'Vehicle':0,
                    bounding_box_list = [ 91-91:[238,259,1021,325],
                                          92-92:[238,263,1017,325],
                                           ...
                                          219-219:[207,196,714,341] ] ] ),
        tracklet( [ id                = 'Vehicle':1,
                    bounding_box_list = [ 121-121:[260,257,1,317],
                                          122-122:[260,264,1,317],
                                          ...
                                          219-219:[260,694,-1,317] ] ] )
        ],
        event_list  = [
          event( [  type              =  approach / 2,
                    participants      =  ['Vehicle':1,'Vehicle':0],
                    framespan         =  span(122,219)   ] ),
          event( [  type              =  approach / 2,
                    participants      =  ['Vehicle':0,'Vehicle':1],
                    framespan         =  span(122,219)   ] )
        ]
    ] ).
```

Listing 6.2: Sample annotation – Prolog

list is composed of `event` terms, each of which lists the type and arity (e.g. `approach/2`), the participants and the frame interval over which the event occurs.

For each vignette, there are essentially two classes of annotation files available to the system:

- *Hand-annotated* data produced by several human annotators. Each annotator received a set of vignettes for which he/she manually specified the coordinates of each oject's bounding box at each frame and added the event occurrences believed to be occurring in the vignette.
- *Tracked* data automatically generated by tracking algorithms (whose generation and analysis is outside the scope of this work).

Annotation files are subject to the issue of uncertainty discussed in Sec. 3.6. Specifically, this data does not have any contextual or background information, as the environment in which the objects act is not represented

in any way. It is very coarse-grained, as each object is identified only by a rectangular bounding box and parts of objects which may be of interest are not represented, for example people's hands and feet. Additionally, objects are represented on a two-dimensional plane and information relative to the third dimension is not represented explicitly.

Regarding data reliability, there is a wide gap between hand-annotated and tracked data. Hand-annotated data, despite its coarseness and exclusion of contextual information, is mostly reliable as rectangles are drawn around objects with overall good precision. On the other hand, tracked data is not only coarse and lacking in any contextual information, but is also subject to errors. In fact some objects in the annotation may not relate to any real object in the video, similarly objects in the video may have a partial or non-existent representation in the annotation, and the bounding boxes may not represent the position accurately.

The current stage of the development of ProVision is not focused on the detection and management of issues arising from the data such as the ones briefly overviewed above, especially regarding the unreliability of tracked data. This is within the scope of future work with the development of the Theory of Appearances (see Sec. 4.7). Throughout of the rest of this chapter, any reference to video annotations data is to be assumed to refer to hand-annotated data.

6.2 Ontology Implementation

The temporal model resulting from this data is given by a finite set of ordered frames, each of which corresponds to a time instant in our ontology (see Sec. 4.1). In the fragments of code to follow, each time instant corresponds to a frame, usually denoted by a variable such as `F`. Intervals are represented by the term `span(Fs,Fe)`, where `Fs` and `Fe` are respectively the first and last frame in the interval.

The spatial model resulting from this data is a cartesian coordinate system of two-dimensional coordinates representing pixels within the video image frame. The origin of this system is at point $(0, 0)$ at the top-left corner of the frame, increasing in the downward and rightward direction. The maximum x- and y-coordinates are bounded by the video frame size.

In the code listings below, each point is represented as a list of coordi-

nates [X,Y], and each spatial area corresponds to a rectangular bounding box represented by the term bbox(H,W,X,Y), where H, W, X and Y are respectively the height, width and x- and y-coordinate of its top-left corner.

This section will illustrate how the information from the vignette annotations explained in the previous section is used by the ProVision system to ground the ontology and infer whether mid-level predicates hold at a particular time and the intervals over which events occur.

Temporally-indexed predicates

In order to express that a predicate p holds at time t, the ontology in Chapter 4 and the verb models in Chapter 5 make use of the Event Calculus constructs HoldsAt(p,t) and its abbreviation $p_{@t}$. Event occurrences are represented as Occurs(e,i).

In the Prolog code, these formalisations are wrapped inside the term infer(Ha, Context), where Ha is an instance of a temporally indexed predicate involving HoldsAt or Occurs. The following fragments of code express that predicate Predicate holds at frame F, or that event Event occurs over interval span(Fs,Fe):

```
infer( holds_at(Predicate, F), Context).
infer( occurs(Event, span(Fs, Fe), Context).
```

The construct HoldsOn in Eq. 4.2 is defined within infer by the following Prolog code:

```
infer( holds_on( Predicate, span(Fs, Fe) ), Context ) :-
  infer( holds_at( Predicate, Fs), Context ),
  Previous is Fs -1,
  \+( infer( holds_at( Predicate, Previous ), Context ) ),
  end_of_holds_on_span( Predicate, Fs, Fe, Context ).

end_of_holds_on_span(Predicate, Fs, Fe, Context ) :- !,
  Next is Fs + 1,
  ( infer(holds_at( Predicate, Next), Context ) ->
    end_of_holds_on_span( Predicate, Next, Fe, Context ) ;
    Fe = Fs ),
  !.
```

The variable Context is intialised and updated by the system, and represents a form of contextual information available to the predicate definition. Currently, it is a term embedding a list with details about the anno-

tation set under examination (i.e. an `annotationset` term such as the one illustrated in Listing 6.2), the precisification (see Sec. 4.5) and the frame span of the current vignette:

```
inference_context( [annotationset=A, precisification=P, framespan=S ] )
```

Precisifications

Precisifications are represented as a contextual element rather than a parameter of a particular predicate, for ease of implementation. Each precisification is essentially a list of thresholds and their values. For example, precisification $P = \{(T, t)\}$ is represented in the code by the list `[ T = t ]`.

An initial precisification is initialised by the system and contains the necessary thresholds for the inference of the implemented definitions. The fragment below shows an example of a `default_precisification` fact which initialises such precisification:

```
default_precisification( [ movement_detection_window = 10,
                           towards_min_speed         = 0.2,
                           closer_min_speed          = 0.2,

                           holding_pos_top_height    =  0.3,
                           holding_pos_bottom_height =  1.0,
                           holding_pos_left_width    = -0.5,
                           holding_pos_right_width   =  1.5,

                           hold_rel_pos_height_tolerance  = 0.1,
                           hold_rel_pos_width_tolerance   = 0.2,

                           merge_threshold           = 15,
                           filter_threshold          = 30  ] ).
```

The precisification thresholds listed above are to be found in definitions appearing later in this section.

6.2.1 Ontology Grounding

Primitive properties of objects are grounded by ProVision on the contents of the `annotationset` term asserted by the system after parsing the Viper XML annotation data (see example in Listing 6.2). The term in fact lists objects and their bounding boxes, corresponding to their spatial extension.

Given that spatial points are represented by the list of coordinates `[X,Y]` and spatial areas by the term `bbox`, the following fragment defines the primitive properties `p_point` and `p_extension` of abstract objects `Point` and `Area` (see pag. 72):

```
p_point( Point, [X, Y] ) :-
  Point = [ X, Y ].
p_extension( Area, bbox(H,W,X,Y) ) :-
  Area = bbox(H,W,X,Y).
```

The code above is rather trivial given the simple representation chosen for points and areas, but it allows for generality and modularity should the representation be extended in the future.

The properties expressing the centroid and position of a spatial area (see pag. 74) are defined in the fragment below:

```
position(Area, Point) :-
  centroid(Area, Point).

centroid(Area, [Cx, Cy] ) :-
  p_extension(Area, bbox(H,W,X,Y) ) :-
  Cx is X + W/2,
  Cy is Y + H/2.
```

In this implementation, an object of type `ConcreteObject` corresponds to a tracked object in the annotation data, and its primitive spatial property is `p_extension` (see pag. 71). This is the only one defined in our implementation, given that objects are represented by their extension in the form of bounding boxes. The following fragment grounds the property expressing that object `Ob` has extension `Area` at frame `F`:

```
infer( holds_at( p_extension(Ob, Area), F), Context) :-
  ensure_ground_object(Ob, Context), ground(F), !
  obatval( Ob, bounding_box_list, BBlist),
  member( T1-T2 : [H,W,X,Y], BBlist ),
  T1 =< F, F =< T2,
  Area = bbox(H,W,X,Y).
```

The definition above ensures object `Ob` is grounded in the current context as a term of the form `tracklet(+L)` as in the code in Listing 6.2. The list of bounding boxes is extracted from the object and the requested bounding box at frame `F` is searched within the list and unified with `Area`.

In the code above, predicate `obatval( +Ob, +Key, -Value)`, also appearing in definitions to be found later on, unifies `Value` with the right

hand side of the term `[ Key = Value ]` member of the associative list `Ob`.

Given the primitive property expressing extension of generic objects above, the property $\mathsf{position}(o \in \mathsf{ConcreteObject}, p \in \mathsf{Point})_{@t}$ can be inferred by the system (see pag. 78):

```
infer( holds_at( position(Ob, Point), F), Context) :-
  infer( holds_at( p_extension(Ob, Area), F), Context),
  position(Area, Point).
```

The grounding of primitive properties briefly outlined in the previous definitions allows for the inference of higher level properties. For example, the property distance for points and generic objects can be inferred:

```
distance(Point1, Point2, Dist) :-
  p_point(Point1, Coords1),
  p_point(Point2, Coords2),
  euclid_distance(Coords1, Coords2, Dist).

euclid_distance( [X1, Y1], [X2, Y2], Dist) :-
  Dist is sqrt( (X1 - X2) * (X1 - X2) + (Y1 - Y2) * (Y1 - Y2) ).

infer( holds_at( distance(Ob1, Ob2, Dist) , F), Context) :-
  infer( holds_at( position(Ob1, Point1), F), Context),
  infer( holds_at( position(Ob2, Point2), F), Context),
  distance( Point1, Point2, Dist).
```

6.2.2 The Verb Approach

The verb approach has been modelled in Sec. 5.2.1 by defining fluents getCloserTo and moveTowards. The definitions of these fluents need some modifications for their implementation in ProVision.

Eq. 5.31 defines the fluent $\mathsf{getCloserTo}(o_1, o_2)_{@t}$ by expressing that there exists an interval $[t_s, t_e]$ surrounding t over which the distance between o_1 and o_2 decreases at each subsequent time point. Eq. 5.32 defines fluent $\mathsf{moveTowards}(o_1, o_2)_{@t}$ in a similar fashion.

In order to infer whether the fluents hold in the ontology implementation described so far, the interval $[t_s, t_e]$ surrounding t has to be precisely individuated. Additionally, negligible reductions in distance between o_1 and o_2 should not determine an occurrence of Approach, as these generally do not constitute significant occurrences of the verb and may signify noise or unrelated movement that may appear from the data.

For this reason, precisification P is added as a parameter to the definitions. P contains thresholds (T_w, w) and (T_s, s) respectively specifying the size of the *detection window* (i.e. the interval $[t_s, t_e]$ surrounding t) and the minimum speed at which o_1 has to be getting closer or moving towards o_2 over this window in order for the predicates to hold at time t. The revised definitions are below:

$$\begin{aligned}
&\mathsf{getCloserTo}[P](o_1, o_2)_{@t} \equiv \\
&\quad \exists\,(T_w, w), (T_s, s) \in P\ \exists\, t_s, t_e, d_s, d_e [t - t_s = t_e - t = w \\
&\quad \wedge \mathsf{distance}((o_1, o_2), d_s)_{@t_s} \wedge \mathsf{distance}((o_1, o_2), d_e)_{@t_e} \wedge \frac{d_s - d_e}{t_e - t_s} > s \qquad (6.1) \\
&\mathsf{moveTowards}[P](o_1, o_2)_{@t} \equiv \\
&\quad \exists\,(T_w, w), (T_s, s) \in P\ \exists\, t_s, t_e, p_{1s}, p_{2s}, p_{1e}\ [t - t_s = t_e - t = w \\
&\quad \wedge \mathsf{position}(o_1, p_{1s})_{@t_s} \wedge \mathsf{position}(o_2, p_{2s})_{@t_s} \wedge \mathsf{position}(o_1, p_{1e})_{@t_e} \\
&\quad \wedge \mathsf{distance}((p_{1s}, p_{2s}), d_s) \wedge \mathsf{distance}((p_{1e}, p_{2s}), d_e) \wedge \frac{d_s - d_e}{t_e - t_s} > s \qquad (6.2)
\end{aligned}$$

The implementation in Prolog requires a predicate to establish the detection window given the threshold in the precisification. The following predicate unifies `WinStart` and `WinEnd` with the start and end frame of the detection window at frame `F` (the predicate `obatval` extracts the value of a threshold from precisification list `P`):

```
detection_window( move, P, F, WinStart, WinEnd) :-
  obatval(P, movement_detection_window, Win),
  WinStart is max( 1, F - Win),
  WinEnd is F + Win.
```

The `infer` clause for predicate `getCloserTo` ensures objects `Ob1` and `Ob2` are grounded and distinct, calculates the detection window, infers the distance between `Ob1` and `Ob2` at the start and end of the window and checks whether the average speed of `Ob1` is greater than the value of threshold `closer_min_speed` over the window:

```
infer( holds_at( getCloserTo( Ob1, Ob2), F),  Context ) :-
  ensure_ground_object( Ob1, Context ),
  ensure_ground_object( Ob2, Context ),
  \+( Ob1 = Ob2 ),
  ensure_ground_frame( F, Context ),
  obatval( Context, precisification, P ),
  detection_window( move, P, F, WinStart, WinEnd),
```

```
  infer( holds_at( distance(Ob1, Ob2, DistStart), WinStart ), Context ),
  infer( holds_at( distance(Ob1, Ob2, DistEnd), WinEnd ), Context ),
  ClosingSpeed is (DistStart - DistEnd) / (WinEnd - WinStart),
  obatval( P, closer_min_speed, MinSpeed ),
  ClosingSpeed > MinSpeed.
```

Similarly, the `infer` clause for predicate `moveTowards` ensures objects are ground and checks whether the average speed at which `Ob1` has been moving towards the initial position of `Ob2` is greater than the value of threshold `towards_min_speed`:

```
infer( holds_at( moveTowards( Ob1, Ob2), F),  Context ) :-
  ensure_ground_object( Ob1, Context ),
  ensure_ground_object( Ob2, Context ),
  \+( Ob1 = Ob2 ),
  ensure_ground_frame( F, Context ),
  obatval( Context, precisification, P ),
  detection_window( move, P, F, WinStart, WinEnd),
  infer( holds_at( distance(Ob1, Ob2, DistStart), WinStart), Context ),
  infer( holds_at( position(Ob1, Point1End), WinEnd ), Context ),
  infer( holds_at( position(Ob2, Point2start), WinStart ), Context ),
  distance( Point1End, Point2Start, DistEnd ),
  SpeedTowards is (DistStart - DistEnd) / (WinEnd - WinStart),
  obatval( P, towards_min_speed, MinSpeed ),
  SpeedTowards > MinSpeed.
```

The occurrence of the event `Approach` is inferred by simpy inferring whether the two fluents hold over the interval `Span`:

```
infer( occurs( approach( Ob1, Ob2 ), Span ), Context ) :-
    infer( holds_on( moveTowards( Ob1, Ob2), Span),  Context ),
    infer( holds_on( getCloserTo( Ob1, Ob2), Span),  Context ).
```

6.2.3 The Verb Hold

The general meaning of `Hold` is that a person is carrying or supporting an object with his/her hands, with the position being mostly stationary, even though there are occasions where, for example, a person may be walking *and* holding an object at the same time. The verb has been modelled in Sec. 5.4.6 using the primitive `p_hands` that specifies the position of the hands of an object of type `Person`. However, given the coarse-grained nature of the data available for ontology grounding, and the fact the hand

position is not specified, the position of a subject's hands cannot be extracted precisely.

In the implementation of Hold for event recognition purposes the definition for $\mathsf{hold}(o_1, o_2)_{@t}$ has been rewritten by referring to a *holding position*, representing the location an object o_2 is most likely to be at when held by o_1 at time t:

$$
\begin{aligned}
&\mathsf{holdingPosition}[P](o_1, o_2)_{@t} \equiv \\
&\quad \exists\,(T_h, t_h), (B_h, b_h), (L_w, l_w), (R_w, r_w), (Tol, tol) \in P, a_1, a_2, A_1, A_2, \\
&\quad p_1^{tl}, p_2^{tl}, h_1, h_2, w_1, w_2[\mathsf{extension}(o_1, a_1)_{@t} \wedge \mathsf{extension}(o_2, a_2)_{@t} \wedge \\
&\quad \mathsf{p\_area}(a_1, A_1) \wedge \mathsf{p\_topleft}(A_1, p_1^{tl}) \wedge \mathsf{p\_width}(A_1, w_1) \wedge \\
&\qquad \mathsf{p\_height}(A_1, h_1) \wedge pos\_x(p_1^{tl}, x_1) \wedge pos\_y(p_1^{tl}, y_1) \wedge \\
&\quad \mathsf{p\_area}(a_2, A_2) \wedge \mathsf{p\_topleft}(A_2, p_2^{tl}) \wedge \mathsf{p\_width}(A_2, w_2) \wedge \\
&\qquad \mathsf{p\_height}(A_2, h_2) \wedge pos\_x(p_2^{tl}, x_2) \wedge pos\_y(p_2^{tl}, y_2) \wedge \\
&\quad [(y_2 + h_2) - (y_1 + b_h \cdot h_1)] < tol \cdot h_2 \wedge \\
&\quad [(y_1 + t_h \cdot h_1) - y_2] < tol \cdot h_2 \wedge \\
&\quad [(x_1 + l_w \cdot w_1) - x_2] < tol \cdot w_2 \wedge \\
&\quad [(x_2 + w_2) - (x_1 + r_w \cdot w_1)] < tol \cdot w_2] \qquad (6.3)
\end{aligned}
$$

The precisification thresholds T_h and B_h determine the top- and bottom-most y-coordinates of the holding position shaded in the illustration in Fig. 6.2. Thresholds L_w and R_w determine the left- and right-most x-coordinates. Threshold *Tol* is a tolerance value used to allow for positions that fall outside this area to be still classified as holding positions.

The definitions inferring an occurrence of event Hold on this formalisation have been implemented in ProVision with the code below:

```
infer( holds_at( holdingPosition(Ob1, Ob2), F), Context) :-
      obatval(Context, precisification, P),
      obatval(P, holding_pos_top_height, T_H),
      obatval(P, holding_pos_bottom_height, B_H),
      obatval(P, holding_pos_left_width, L_W),
      obatval(P, holding_pos_right_width, R_W),
      obatval(P, hold_rel_pos_height_tolerance, TolH),
      obatval(P, hold_rel_pos_width_tolerance, TolW),
      infer( holds_at( p_extension(Ob1, Area1), F), Context),
      infer( holds_at( p_extension(Ob2, Area2), F), Context),
```

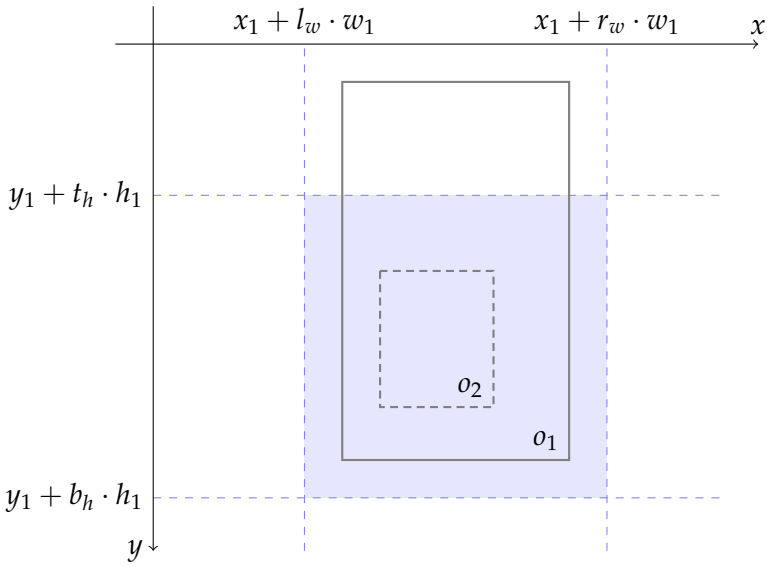
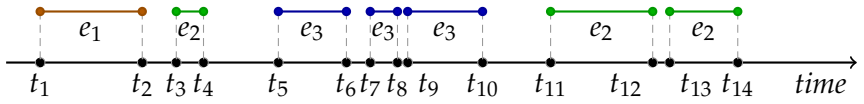


Figure 6.2: Implementation of Hold – Estimation of a 'holding position'

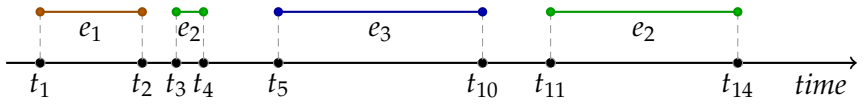
```
    p_extension(Area1, bbox(H1, W1, X1, Y1)),
    p_extension(Area2, bbox(H2, W2, X2, Y2)),
      ( ( Y2 + H2 )          - ( Y1 + (B_H * H1) ) )  < ( TolH * H2 ),
      ( ( Y1 + (T_H * H1) )  - Y2 )                   < ( TolH * H2 ),
      ( ( X1 + ( L_W * W1 ) ) - X2 )                  < ( TolW * W2 ),
      ( ( X2 + W2)           - (X1 + ( R_W * W1 ) ) ) < ( TolW * W2 ).

infer( holds_at( hold(Ob1, Ob2), F), Context) :-
    ensure_ground_object(Ob1, Context),
    ensure_ground_object(Ob2, Context),
    type( Ob1, 'Person'),
    type( Ob2, 'Other'),
    ensure_ground_frame(F, Context),
    infer( holds_at( holdingPosition(Ob1, Ob2), F), Context)

infer( occurs( hold(Ob1, Ob2), Span), Context) :-
    infer( holds_on( hold(Ob1, Ob2), Span), Context).
```

6.2.4 Occurrence Smoothing

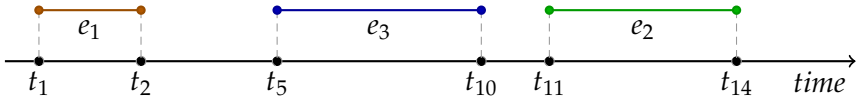
When testing the event recognition system with the standard definition for the constructs HoldsAt and HoldsOn, in several occasions several short isolated occurrences were produced, probably due to the fact that some



(a) Original sequence



(b) Merged Sequence



(c) Filtered Sequence

Figure 6.3: Merging and filtering fluents and event occurrences

vignette annotations, or some actual objects in the video, do not move particularly fluidly.

This problem can be overcome by enhancing HoldsAt, HoldsOn and Occurs in order to extend the truth-value of a particular fluent or event over small temporal gaps, likely to form part of a long occurrence span, and falsify short isolated occurences, likley to constitute spurious ones. This idea of *occurrence smoothing* is illustrated in Fig. 6.3.

The first stage, corresponding to the transition between Fig. 6.3(a) and Fig. 6.3(b) is performed by the new constructs HoldsAtM and OccursM. The former establishes that fluent f holds at time point t if t is part of a wider interval $[t_1, t_2]$ with duration smaller than precisification threshold T_m and where f holds at t_1 and t_2. Similarly, the latter joins together separate occurrences of an event-type e which are separated by an interval with duration smaller than threshold T_e:

$$\begin{aligned}
&\mathsf{HoldsAtM}[P](f,t) \equiv \\
&\quad \exists\,(T_m, t_m) \in P, t_1, t_2 \in \mathcal{T}[t_1 \leqslant t \leqslant t_2 \wedge \\
&\quad \mathsf{HoldsAt}(f,t_1) \wedge \mathsf{HoldsAt}(f,t_2) \wedge t_2 - t_1 < t_m] \qquad (6.4) \\
&\mathsf{OccursM}[P](e,[t_s,t_e]) \equiv
\end{aligned}$$

$$\mathsf{Occurs}(e,[t_s,t_e]) \vee \exists\ (T_e,t_e) \in P, t_1, t_2 \in \mathcal{T}[(t_s < t_1 < t_2 < t_e) \wedge \mathsf{Occurs}(e,[t_s,t_1]) \wedge \mathsf{Occurs}(e,[t_2,t_e]) \wedge t_2 - t_1 < t_m] \tag{6.5}$$

The second stage, corresponding to the transition between Fig. 6.3(b) and Fig. 6.3(c) is performed by the constructs HoldsAtF and OccursF. These are built on constructs HoldsAtM and OccursM and filter isolated occurrences of very little duration. $\mathsf{HoldsAtF}(f,t)$ holds if and only if f holds at time t part of an interval $[t_1,t_2]$ longer than threshold T_f, and similarly for OccursF:

$$\begin{aligned}
&\mathsf{HoldsAtF}[P](f,t) \equiv \\
&\quad \exists (T_f,t_f) \in P, t_1, t_2 \in \mathcal{T}[t_1 \leqslant t \leqslant t_2 \wedge t_2 - t_1 > t_f \wedge \\
&\qquad \forall\, t'[(t_1 \leqslant t' \leqslant t_2) \rightarrow \mathsf{HoldsAtM}(f,t')]] \qquad (6.6)\\
&\mathsf{OccursF}[P](e,[t_s,t_e]) \equiv \\
&\quad \exists (T_f,t_f) \in P[\mathsf{OccursM}(e,[t_s,t_e]) \wedge t_e - t_s > t_f] \qquad (6.7)
\end{aligned}$$

The constructs HoldsOverM, HoldsOnM, HoldsOverF and HoldsOnF can be defined following the same scheme in the definitions for HoldsOver and HoldsOn in Eq. 4.1 and Eq. 4.2.

The definitions above have been implemented in ProVision, and the thresholds corresponding to T_m and T_f in the equations above have been included in the standard precisification illustrated in the code on pag. 140 as thresholds `merge_threshold` and `filter_threshold`.

This mechanism of occurrence smoothing has been employed to produce a list of event occurrences described by verbs Approach and Hold and obtain the results discussed in the next section.

6.3 Experimental Results

This section outlines the methodology with which the event recognition system ProVision has been tested given the implementation of the ontology in the previous section.

From Sec. 6.1 it can be recalled that hand-annotated vignette annotations include event annotations which state a series of events occurring in each vignette. These are represented in the `annotationset` term within the list `event_list` shown in Listing 6.2 on pag. 137. These hand-annotated

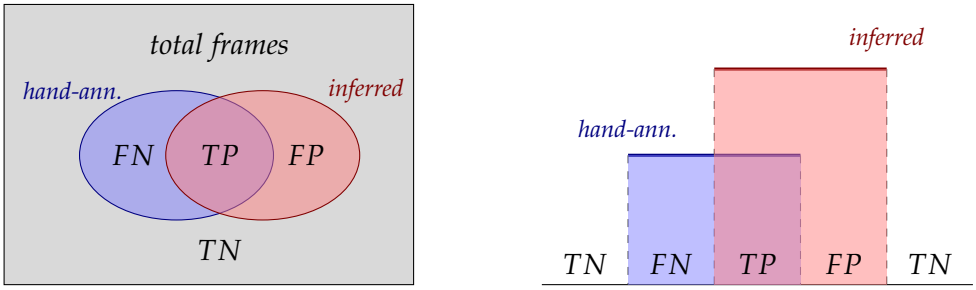


Figure 6.4: Evaluation – Frame classification

event occurrences constitute the *ground truth* used for evaluating the performance of ProVision. The methodology surrounding the production of such ground truth for evaluation purposes is outside the scope of this work, and has just been made available for this project. It has been produced by human observers who watched a subset of vignettes and annotated the events which, in their opinion, believed to be occurring in the scene.

Each frame F part of an event occurrence recognised by ProVision is categorised in one of the following sets, also illustrated in Fig. 6.4:

- TP (True Positives): if F is within the span of an inferred event occurrence *also* in the ground truth;
- FP (False Positives): if F is within the span of an inferred event occurrence *not* in the ground truth;
- TN (True Negative): if no inferred event occurrences nor occurrences in the ground truth involve F;
- FN (False Negative): if F is within the span of a ground truth occurrence but ProVision produced no inferred occurrence involving F.

At the end of the frame categorisation, each set is such that $|TP| + |FP| + |TN| + |FN| = T$ where T is the total number of frames in the vignette. The measures Prec (Precision), Rec (Recall), Fv (F-value), Mcc (Matthews Correlation Coefficient) and occurrence rates $TP^{\%}$, $FP^{\%}$, $TN^{\%}$,

$FN^{\%}$ are calculated according to the following formulae:

$$\begin{gathered}
\text{Prec} = \frac{|TP|}{|TP| + |FP|} \qquad \text{Rec} = \frac{|TP|}{|TP| + |FN|} \qquad \text{Fv} = 2 \cdot \frac{\text{Prec} \cdot \text{Rec}}{\text{Prec} + \text{Rec}} \\
\text{Mcc} = \frac{(|TP| \cdot |TN|) - (|FP| \cdot |FN|)}{\sqrt{(|TP| + |FP|)(|TP| + |FN|)(|TN| + |FP|)(|TN| + |FN|)}} \\
TP^{\%} = \frac{|TP_i|}{|TP_i| + |FN_i|} \qquad FP^{\%} = \frac{|FP_i|}{|TN_i| + |FP_i|} \\
TN^{\%} = \frac{|TN_i|}{|TN_i| + |FP_i|} \qquad FN^{\%} = \frac{|FN_i|}{|TP_i| + |FN_i|}
\end{gathered} \tag{6.8}$$

Values for Prec and Rec range between 0 and 1, whilst values for Mcc range between -1 and 1. The values for Prec and Rec are set to 1 if the denominator is 0, values for Mcc are set to 0 if the denominator is 0 [82].

Given that recognition of an occurrence of a particular verb is tested over a set of n vignettes, an overall system accuracy figure for the recognition of the verb over the set is obtained by summing the values $|TP|$, $|FP|$, $|TN|$ and $|FN|$ relative to each vignette, thus calculating a global value for each category. The statistic measures in the formulae above are then computed on these global values.

6.3.1 Sample Statistics and Baseline Accuracy

Tracked data tends to abund with error and noise; the detection tests discussed in this section have been carried out on hand-annotated data, as ProVision is not yet fully capable of managing tracked data effectively.

Tests on verbs Approach and Hold have been run on two particular sets of vignettes:

- *Whole set*, or W: all 1302 vignettes in the development dataset.
- *Restricted set*, or R: only vignettes whose hand-annotated annotation file reports an occurrence of the event being tested.

The frequency of occurrences of verbs Approach and Hold in the ground truth annotations are reported in Table 6.1 detailing the sample statistics and displaying, for each verb and set, the total number of frames across the vignettes, the number of Positive and Negative frames (i.e. the frames

Verb	Set	Vignettes	Frames	Positives	Negatives	PosRate	NegRate
Approach	W	1302	595,110	5,254	589,856	0.88 %	99.12 %
Approach	R	70	33,340	5,254	28,086	15.76 %	84.24 %
Hold	W	1302	595,110	26,034	569,076	4.37 %	95.63 %
Hold	R	98	42,682	26,034	16,648	61.00 %	39.00 %

Table 6.1: Sample statistics

Verb	Set	Bl	Prec	Rec	Fv	Mcc	$TP^{\%}$	$FP^{\%}$	$TN^{\%}$	$FN^{\%}$
Approach	W	A	0.010	0.969	0.020	0.028	96.90	86.69	13.32	3.10
Approach	W	S	0.016	0.917	0.031	0.077	91.74	50.36	49.64	8.26
Approach	W	N	1.000	0.000	0.000	0.000	0.00	0.00	100.00	100.00
Approach	R	A	0.159	0.969	0.273	0.023	96.90	95.61	4.39	3.10
Approach	R	S	0.280	0.917	0.429	0.347	91.74	44.16	55.84	8.26
Approach	R	N	1.000	0.000	0.000	0.000	0.00	0.00	100.00	100.00
Hold	W	A	0.044	1.000	0.084	0.012	100.00	99.68	0.32	0.00
Hold	W	S	0.068	0.750	0.125	0.116	75.01	46.71	53.29	24.99
Hold	W	N	1.000	0.000	0.000	0.000	0.00	0.00	100.00	100.00
Hold	R	A	0.610	1.000	0.758	0.000	100.00	100.00	0.00	0.00
Hold	R	S	0.605	0.750	0.670	−0.019	75.01	76.65	23.35	24.99
Hold	R	N	1.000	0.000	0.000	0.000	0.00	0.00	100.00	100.00

Bl: baseline algorithm (All, Some or None);
Set: vignette set (Whole or Restricted).

Table 6.2: Baseline statistics

within or outside the span of a hand-annotated occurrence of the verb) and their occurrence rates PosRate and NegRate are reported.

Baseline statistics for the recognition of event occurrences, reported in Table 6.2, have been generated by implementing three very simple baseline detection algorithms, :

- *All*: $\mathsf{Approach}(o_1, o_2)$ occurs at every interval where two distinct objects o_1 and o_2 are present. $\mathsf{Hold}(o_1, o_2)$ occurs at every interval where two objects o_1 and o_2 are present.

- *Some*: $\mathsf{Approach}(o_1, o_2)$ occurs at every interval where two distinct objects o_1 and o_2 are present and o_1 is moving. $\mathsf{Hold}(o_1, o_2)$ occurs at every interval where two distinct objects o_1 and o_2 are present, o_1 is of type Person and o_2 is of type Other.

- *None*: $\mathsf{Approach}(o_1, o_2)$ and $\mathsf{Hold}(o_1, o_2)$ never occur in any interval.

6.3.2 Detection Results

The precisification shown on pag. 140 specifies the value of several thresholds essential to the implementation of the definitions for verbs Approach and Hold. At the current stage, these values are set by default when the system is initialised even though their automatic inference is envisioned in future stages. For this reason, recognition tests have been repeated several times in order to assess the impact of different threshold values on the recognition accuracy.

For the verb Approach tests have been run with the following threshold values:

- `movement_detection_window` = $t_w \in \{5, 7, 10, 12, 13, 14\}$;
- `towards_min_speed` = $t_s \in \{0.2, 0.5, 0.7, 0.9\}$;
- `closer_min_speed` equal to `towards_min_speed`;
- `merge_threshold` = $\delta_m \in \{5, 10, 15, 17, 20, 22, 25, 30, 35, 40\}$;
- `filter_threshold` = $\delta_f \in \{5, 7, 10, 12, 15, 17, 20, 22, 25, 30, 35, 40, 50, 60, 70, 80\}$.

For the verb Hold tests have been run with the following threshold values:

- `holding_pos_top_height` = $t_t \in \{0.25, 0.35, 0.45\}$;
- `holding_pos_bottom_heigh` = $t_b \in \{0.85, 1.00, 1.15\}$;
- `holding_pos_left_width` = $t_l \in \{-0.1, -0.25, -0.5\}$;
- `holding_pos_right_width` = $t_r = 1 - t_l$;
- `hold_rel_pos_height_tolerance` = $tol \in \{0.1, 0.2, 0.3\}$;
- `hold_rel_pos_width_tolerance` equal to the previous values.

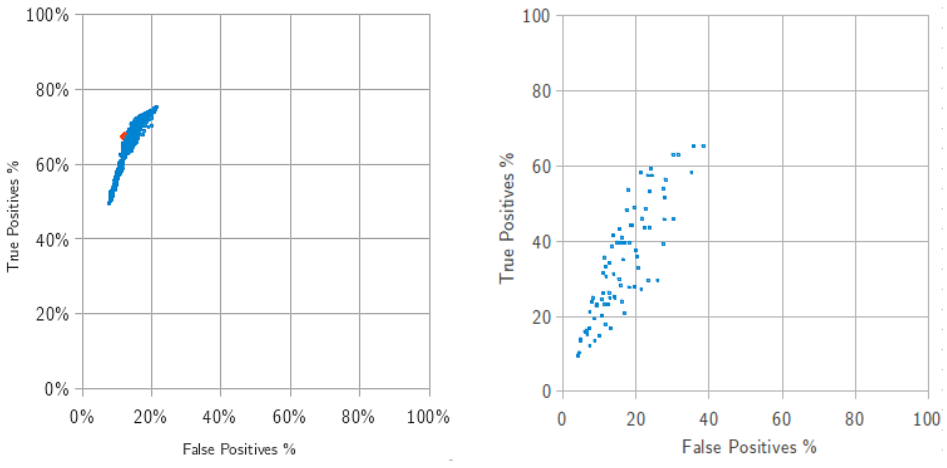
After an initial test phase, the set of threshold values has been progressively reduced, focusing on values yielding the best accuracy results. Experimental results for the recognition of occurrences of Approach are reported in Table 6.3, which reports accuracy statistics for increasing values of threshold δ_f. Table 6.4 reports accuracy statistics for occurrences of

Set	t_w	t_s	δ_m	δ_f	Prec	Rec	Fv	Mcc	$TP^{\%}$	$FP^{\%}$	$TN^{\%}$	$FN^{\%}$
W	10	0.2	15	30	0.042	0.673	0.079	0.144	67.26	13.64	86.36	32.74
W	10	0.2	15	40	0.047	0.625	0.087	0.149	62.50	11.30	88.70	37.50
W	10	0.2	15	50	0.052	0.575	0.096	0.152	57.46	9.27	90.73	42.54
W	10	0.2	15	60	0.057	0.524	0.103	0.153	52.38	7.69	92.31	47.62
W	10	0.2	15	70	0.060	0.459	0.106	0.147	45.89	6.45	93.55	54.11
W	10	0.2	15	80	0.062	0.417	0.108	0.143	41.66	5.59	94.41	58.34
R	10	0.2	15	30	0.506	0.673	0.578	0.492	67.26	12.27	87.73	32.74
R	10	0.2	15	40	0.507	0.625	0.560	0.471	62.51	11.37	88.63	37.50
R	10	0.2	15	50	0.513	0.575	0.542	0.452	57.46	10.19	89.81	42.54
R	10	0.2	15	60	0.520	0.524	0.522	0.432	52.38	9.03	90.97	47.62
R	10	0.2	15	70	0.520	0.459	0.488	0.400	45.89	7.93	92.07	54.11
R	10	0.2	15	80	0.505	0.417	0.457	0.369	41.66	7.64	92.36	58.34

Table 6.3: Accuracy statistics for the recognition of Approach

Set	t_t	t_b	t_l	tol	Prec	Rec	Fv	Mcc	$TP^{\%}$	$FP^{\%}$	$TN^{\%}$	$FN^{\%}$
W	0.25	0.85	−0.1	0.2	0.153	0.415	0.224	0.196	41.47	10.50	89.50	58.53
W	0.25	1.15	−0.5	0.3	0.096	0.653	0.167	0.166	65.30	28.16	71.84	34.70
W	0.25	0.85	−0.1	0.3	0.145	0.537	0.228	0.217	53.71	14.52	85.48	46.30
W	0.25	0.85	−0.25	0.3	0.142	0.582	0.228	0.225	58.17	16.07	83.93	41.83
R	0.35	0.85	−0.1	0.3	0.830	0.357	0.499	0.269	35.69	11.40	88.60	64.31
R	0.25	1.15	−0.5	0.3	0.727	0.653	0.688	0.264	65.30	38.33	61.67	34.70
R	0.25	1.00	−0.5	0.3	0.740	0.652	0.693	0.287	65.23	35.85	64.15	34.77
R	0.25	0.85	−0.25	0.3	0.811	0.582	0.678	0.363	58.17	21.25	78.75	41.83

Table 6.4: Accuracy statistics for the recognition of Hold



(a) Approach – Restricted set (b) Hold – Restricted set

Figure 6.5: Evaluation – ROC curves

Hold with different threshold values. The precisification yielding the best accuracy value for each statistic figure is underlined.

ROC curve graphs showing the overall detection accuracy over different precisifications are shown in Fig. 6.5 [25, 45]. Each dot on the graph represents a couple of $TP^{\%}$ and $FP^{\%}$ values associated with the event detection results for a specific choice of thresholds. In general, precisifications yielding high $TP^{\%}$ values have the undesirable effect of yielding high $FP^{\%}$ values too; figures showing point concentrations skewed towards topmost and leftmost areas of the graph denote algorithms with good overall performances.

6.4 Considerations

Results obtained by the event recognition tests, carried out as explained in the previous section, show that ProVision recognised 67.26% of true positive frames (against 12.27% of false positives) for the verb Approach and 65.30% of true positive frames (against 38.33% of false positives) for Hold across the restricted set of vignettes. These detection rates yield Precision and Recall figures of 0.506 and 0.673 for Approach and 0.727 and 0.653 for Hold.

When the whole set of vignettes is considered, true and false positives rates do not show significant variations, while Precision and Mcc figures decrease sensibly. This is due to the distribution of event occurrences in hand-annotated data shown in Table 6.1. In fact, occurrences of Approach and Hold only involve 0.88% and 4.37% of frames respectively. Given this occurrence rate, even small $FP^{\%}$ values yield high a high absolute number of false positives, hence the rapid deterioration of Precision, Mcc and Fv values.

The Mcc figure shows some peculiar behaviour in expressing the accuracy of event recognition for event e performed on a vignette v where no event of type e is recognised *and* there is no event annotation for e in the ground truth. In this situation, given the number of frames f in vignette v, the recognition statistics would result in a confusion matrix where $|TP| = |FP| = |FN| = 0$ and $|TN| = f$. Computing the accuracy figures in Eq. 6.8 would have them set at their limit value as the denominators are 0, thus yielding $\text{Prec} = \text{Rec} = \text{Fv} = 0.5$ and $\text{Mcc} = 0$.

A glance at the confusion matrix for the above case may give the impression of a very good or even optimal result, given that nothing has been recognised and there was nothing to be recognised. However, the accuracy figures paint a different picture by classifying the result as just 'average', meaning that the system has not scored badly but not even that well. This may appear slightly unfair but, in some ways, it simply recognises the fact that such vignette is a trivial example, where the system is just required to 'do the bare minimum'.

The issue in our dataset is that there are plenty of such instances, especially when recognition tests regard the whole set of 1302 vignettes. In fact, the sample statistics in Table 6.1 show that only 0.88% of frames are marked with an occurrence of the event Approach in the ground truth, and 4.37% are marked with an occurrence of Hold. The problem is that accuracy results for vignettes not involving either event will lower the overall average score as seen above. And, of course, even a small amount of false positives recognised in such vignettes will lower scores even further.

By examining the annotation files for some vignettes, it appeared that sample statistics for Approach are affected by under-reporting and/or inconsistent reporting of the event in the ground truth. This may have been caused by the fact that human observers who produced the data may have interpreted the meaning of each verb slightly or markedly differently. For example, the event Approach has mostly been marked as a verb with arity 2 identifying an object moving towards and getting closer to another, but also as a verb with arity 1 identifying an object approaching the foreground represented by the camera location.

The issue of saliency is very relevant to this particular problem (see Sec. 3.3). In fact, several examples show that human annotators did not report occurrences of Approach in the ground truth for scenes where more salient and semantically richer events dominate the foreground. In such instances, ProVision would still report an occurrence of the verb, as long as the implemented definitions for the verb hold for any two objects. However, these would count as false positives, thus impacting negatively on the accuracy figures.

This problem in the ground truth could be overcome either by a consistent, reliable re-annotation process, or by a change in perspective in the evaluation methodology.

A reliable re-annotation could be produced by specifying a meaning for each verb as precisely as possible, then having human annotators marking *every* occurrence of the verb so described. However, this would create a biased ground truth as it would orient the annotators towards adhering to the verb meaning intended by the system developers.

A change in perspective in the evaluation methodology would evaluate the system *a posteriori* rather than on a hand-annotated ground truth produced *a priori*. Such methodology would see a first stage where ProVision performs event recognition on a set of vignettes, and a second stage where human annotators are asked whether they think that the events recognised by the system for a particular vignette actually occur in the scene. In fact, people may not notice an event not deemed particularly salient, nevertheless agree with the fact it is happening if asked explicitly.

Both processes (re-annotation and evaluation *a posteriori*) are relatively consuming in terms of time and human resources required, and they would not resolve the issue of saliency. In fact, a system aiming to be intelligent, such as ProVision, should be able to operate a saliency classification similar to the one naturally performed by humans when observing a scene and focusing on the more prominent events. Such a classification could be based on a pre-established hierarchy of verbs and a dynamic reasoning stage in which different aspects of the scene and objects involved are considered (positions, movement range, etc.). Incorporating such intelligent behaviour in ProVision would constitute a deciding feature towards a more reliable event recognition system.

The verb Hold is not affected by under-reporting to the same extent of Approach. The high false positives detection rate for this verb is rather caused by the fact that an occurrence of Hold is inherently more difficult to detect when the position of the two objects is only described by their bounding boxes, hence the relatively high number of cases where two objects' relative positioning is mistaken for an occurrence of Hold. This could be improved by access to more precise and fine-grained data, or a better estimation of the position of a person's hands.

Regarding hand-annotated occurrences of events not recognised by ProVision, there are several vignettes annotated with an occurrence of Approach where instances seem particularly difficult to recognise given the quality of the data and its associated spatial model. For example, vi-

Figure 6.6: A difficult approach

gnettes where the objects move along the z-axis (i.e. the direction towards or away from the camera) are challenging as that type of motion is not detected yet. This could be improved by enhancing the spatial model with an approximation of the z-coordinate either through an analysis on changes in the size of the bounding boxes, or through more specific algorithms.

An example of a challenging vignette on which our system fails to recognise a hand-annotated occurrence of `Approach` is shown in Fig. 6.6, where two people turn around and approach by 'bumping' into each other. This occurrence is challenging due to a movement along z which, even with the improvements outlined above, appears very hard to detect: the distances involved are very small and the subjects are jumping, thus impairing the detection of any significant changes in the boxes' height.

Chapter 7

Conclusions

The models presented in Chapter 5 demonstrate the escalating complexity of defining natural language concepts in ontologies. The attempt of defining a single motion verb, or even a particular meaning of a verb, often unfolds a variety of sub-concepts, interpretations and sources of ambiguity. The set of verbs on which the methodology and application of this research has focused is very particular in this respect. Most verbs and related sub-concepts suffer from vagueness, in particular multiple meanings (e.g. Approach, Sec. 5.2.1), borderline cases (e.g. near) and deep ambiguity. The interpretation of some concepts may change in different contexts or among different speakers, such as the characterisation of what distinguishes an occurence of Hit and an occurrence of Collide. One may even question the reason or motivation behind the need to discriminate between nearly synonymous concepts, such as Lift or Raise which would seem to describe very similar events. The task of recognising occurrences of these semantic descriptions on real data adds further issues such as saliency and uncertainty, mainly due to the imprecise representation of the world in the data grounding the ontology.

In summary, the main challenge faced during this research on modelling motion verbs and, more generally, spatial concepts has been the formulation of a formal description identifying the semantic properties of each concept that can be recognised by observation. Some of these properties, such as position and distance, admit a reasonably clean, precise and straightforward formalisation, although they are still subject to some degree of interpretation. Many higher-level properties, instead, are

problematic to recognise by grounding inferences on lower-level observable properties. A prime example of such a property is the previously mentioned degree of intentionality or responsibility of an agent in participating in an action for occurrences of Hit and Collide (Sec. 5.4.3 and 5.4.4).

This challenge is constituted by two aspects. The first one involves addressing the vagueness of predicates with borderline cases. The applicability boundary of these predicates depends on one or several observable properties, and the inclusion of these concepts in formal ontologies requires the presence of a device able to precisify borderline cases. This has been addressed with a supervaluationist approach and the parameterisation of some definitions in order to specify precisification thresholds. This approach, however, is a partial solution to the issue of vagueness, as a new element is introduced, which is the need to design an effective mechanism for establishing appropriate threshold values given the particular context of predication. At this stage, such values are set manually during the experimental stage, in order to isolate the thresholds leading to the best recognition results. Ideally, an inference system should be able to automatically infer threshold values relative to a particular context; this would constitute the natural extension of the formalism developed so far (see Sec. 7.2 on future work).

The second aspect of the challenge involves issues of uncertainty arising from imprecision and unreliability in the data on which observable properties are to be inferred. Despite the primary purpose of this research being the demonstration of a methodology for the formalisation of vague concepts in ontologies oriented to practical reasoning tasks, rather than the advance in techniques for event recognition, nevertheless issues related to the quality of the data had an influence on verb modelling. In fact, this process has to strike a compromise between formalising the semantic properties most relevant to the meaning of a verb and formalising the observable properties most likely to be inferrable given the average accuracy and granularity of the data available to a reasoning system..

Such compromise is not easy to achieve, as certain characteristics are inherently hard to detect, such as the frequently mentioned intentionality of an agent. After all, the representation of the world available to the event recognition system was only constituted by rectangular bounding

boxes around some objects. For certain properties, this data simply cannot provide enough evidence for the recognition of verbs with very specific meanings defined in terms of very abstract characteristics.

These two aspects of this research challenge determine, as a consequence, that the formalism illustrated in Chapters 4 and 5 may not possess the elegant characteristics of completeness, generality and consistency that other theories for knowledge representation and reasoning show. One may wonder if it is even possible to define what constitutes an acceptable balance between these desirable characteristics of logical formalisms and their practical effectiveness in understanding and interpreting real data.

Despite the challenging aspects described above, the verb modelling process has resulted in the development of the automatic event recognition system ProVision. The system is capable of inferring certain object properties from primitives and its ontological core allows for generality as it can be extended with the specification of further definitions. Although this process may be inelegant in some of its parts, as the implementation of definitions in Chapters 4 and 6 requires the consideration of practical, low-level issues such as errors and granularity, it is effective in the context of event recognition, and can be improved by further refinements of the ontological model. Still, much greater advances would be achievable by operating on more accurate and fine-grained data. Indeed, further developments of this research are closely related to the improvement in Vision techniques for video processing, an area outside the scope of this project. The preliminary results discussed in Chapter 6 for two simple verbs where ProVision operated on the available data have been encouraging.

In summary, the research has demonstrated the potential applicability of formal semantics to the development of an ontology for describing physical objects and their interactions, and to a concrete reasoning task such as event recognition. A practical supervaluationist method to ground vague concepts on observable properties has been outlined, and it has been stressed that there is a balance between *exhaustive* and *effective* semantic characterisations of concepts. Despite the attractiveness of enriching this formalism further by specifying very detailed semantic characteristics, this could in fact run the risk of specifying characteristics of objects and events whose manifestations may be too challenging to recognise, even on data constituting a very accurate representation of the world.

Given the very particular nature of some of the events to be recognised in this task, the main strength of this approach is to provide for a greater specification of each verb's semantic characteristics, which may not be completely understood by Machine Learning techniques. Indeed, this ontology and its implementation ProVision can integrate a detailed semantic characterisation of concepts and allow for augmentation of inference capabilities by broadening the ontology with additional definitions and deepening it by further specifications and structuring. Moreover, the methodology underlying this approach has the potential to be generalised to other domains and automated reasoning tasks involving qualitative vague concepts.

7.1 Contributions

The research focuses on vagueness in natural language, the methodology and techniques for the formalisation of vague spatial concepts, processes and events in ontologies and the application of the resulting formalism to the task of event recognition.

Specific original contributions are summarised below:

- The analysis and investigation on vagueness in natural language and issues of context-dependency, carried out in Chapter 3, demonstrated the usefulness of an epistemic model of vagueness to the concrete task of reasoning about vague spatial concepts and related event occurrences. In fact, most concepts have been formalised assuming that there exists a crisp boundary separating borderline instances. The applicability of this model is further substantiated by the formalisation and application of the ontology and the verb models in Chapters 4 and 5. The investigation on issues of saliency, uncertainty and granularity provided an insight on the main challenges in performing reasoning tasks on finite and limited representations of real-world scenarios.

- The formalisation of the ontology in Chapter 4 applied the principles of supervaluation semantics to the formalisation of vague concepts, following from the above considerations on epistemic vagueness. Vague predicates are formalised through the parameterisation of def-

initions with precisifications, specifying which observable properties affect the applicability of concepts. This method does not address the issue of vagueness entirely, as further work is needed in order to infer appropriate precisifications within the ontology, however it provides a practical technique for reasoning with vague concepts.

- The verb models in Chapter 5 provide further insight on the formalisation of processes and events in formal ontologies. The result is a methodolgy that focuses on the semantic characterisation of concepts in terms of mid-level predicates that can be ground on observable properties and primitives.
- The implementation of ProVision and the preliminary experimental results in Chapter 6 connect the three aspects above and demonstrate the practical applicability of formal ontologies to a reasoning task such as event recognition from video. Despite the limitations of the dataset available for our evaluation, it is possible to generalise the approach that led to this implementation to other domains and reasoning tasks.

The contribution from Chapter 6 and a selection of the discussions, analysis and formalisations in Chapters 3, 4 and 5 have been published in the proceedings of the 8th International Conference on Spatial Cognition [37]. A contribution focused on the formal analysis of Chapters 4 and 5 has also been published in the proceedings of Commonsense 2013, 11th International Symposium on Logical Formalisations of Commonsense Reasoning [38].

7.2 Future Work

There are many directions in which the research carried out so far could move forward. These include developments on the logical formalism, practical techniques in order to address issues of uncertainty in the data and the extension of the methodology to a different set of spatial concepts or to non-spatial domains. The most important aspects that would lead to an advance of the results presented here are summarised below:

- The approach based on supervaluation semantics and precisification thresholds can be extended by designing an inference mechanism

seamlessly precisifying predicate definitions by inferring the value of thresholds given the current context. This is likely to involve further reasoning on object properties, past and future events, background and other relevant information. This would ultimately lead to the development of an ontology of contextual information guiding the resolution of ambiguities. It will have to address the limitations constituted by uncertainty in the data and the limited knowledge and experience of an automatic system.

- The application of Machine Learning techniques would be of particular interest for the automatic production of event definitions, and could also provide a mechanism for the aforementioned automatic inference of precisification thresholds. Techniques basic on Inductive and Abductive Logic Programming for Event Calculus and Markov Logic Networks, overviewed in Sec. 2.5, are of particular interest as they suit the first-order logic nature of the ontological formalism underlying ProVision.

- The development of a Theory of Appearances (Sec. 4.7) would partially address issues of uncertainty arising from the data grounding the ontology. This module is intended as a refinement of a representation of the world through the discovery of obscure or unclear information that this representation may hide. This could lead, for example, to the extraction of a three-dimensional representation from a two-dimensional one, to the detection of phenomenons of occlusion and to the correction of errors in the data.

- Issues of saliency, discussed in Sec. 3.3 and 6.4, can be addressed by the development of a semantic hierarchy within the ontology specifying which observable properties of objects and events are more relevant within the context of an observed situation. This could be coupled with an inference mechanism establishing saliency by identifying prominent objects or events deemed more relevant than others, for example people in the foreground or complex structured actions.

- The evaluation strategy for this and other reasoning tasks could be revised as explained in Sec. 6.4. For example, rather than evaluat-

ing the performance of the system by assessing whether a computer agrees with the human interpretation of a concept or complex scene (*a priori*), one could consider assessing whether a human agrees with the interpretation provided by the machine (*a posterori*). In fact, even though machines may not assess saliency in the same way humans do, humans may still agree with their assessment.

- The ontology could be broadened by applying the characterisation methodology to other spatial or non-spatial domains and similar reasoning tasks
- The ontology could be deepened with a further specification of concepts' semantic characteristics. This particular process is likely to require more accurate data detailing fine-grained observable properties of objects, for example obtained through advanced tracking algorithms extracting people's features such as body posture and positioning [104, 115].

Bibliography

[1] The aleph manual. University of Oxford, Department of Computer Science, http://www.cs.ox.ac.uk/activities/machlearn/Aleph/.

[2] Oxford english dictionary. OED Online. Oxford University Press. http://www.oed.com.

[3] The British National Corpus, version 3 (BNC XML edition). Distributed by Oxford Computing Services on behalf of the BNC Consortium, http://www.natcorp.ox.ac.uk/, 2007.

[4] R. A. Kowalski A.C. Kakas and F. Toni. Abductive logic programming. *Journal of Logic and Computation*, 2(6):719–770, 1992.

[5] Marco Aiello, Ian E. Pratt-Hartmann, and Johan F. A. K. van Benthem, editors. *Handbook of Spatial Logics*. Springer, Dordrecht, The Netherlands, 2007.

[6] Marco Aiello and Johan F. A. K. van Benthem. A Modal Walk Through Space. *Journal of Applied Non-Classical Logics*, 12(3-4):319–364, 2002.

[7] J. Albath, J. Leopold, C. Sabharwal, and A. Maglia. Rcc-3d: Qualitative spatial reasoning in 3d. In *Proceedings of the 23rd International Conference on Computer Applications in Industry and Engineering (CAINE 2010), Las Vegas, NV, Nov. 8-10, 2010*, pages 74–79, 2010.

[8] James F. Allen. Maintaining Knowledge about Temporal Intervals. *Communications of the ACM*, 26(11):832–843, 1983.

[9] James F. Allen. Towards a General Theory of Action and Time. *Artificial Intelligence*, 23(2):123–154, 1984.

[10] Alexander Artikis, A. Skarlatidis, F. Portet, and G. Paliouras. Logic-based event recognition. *The Knowledge Engineering Review*, 27(04):469–506, December 2012.

[11] Brandon Bennett. Standpoint semantics - A model theory for vague languages.

[12] Brandon Bennett. Modal Logics for Qualitative Spatial Reasoning. *Logic Journal of the IGPL*, 4(1):23–45, 1996.

[13] Brandon Bennett. Modes of Concept Definition and Varieties of Vagueness. *Applied Ontology*, 1(1):17–26, 2005.

[14] Brandon Bennett. Statistical Standpoint Semantics. 2009.

[15] Brandon Bennett. *Methods for Handling Imperfect Spatial Information*, chapter Spatial Vagueness. Springer, 2010.

[16] Brandon Bennett. Possible Worlds and Possible Meanings: a Semantics for the Interpretation of Vague Languages. In *Commonsense 2011: Tenth International Symposium on Logical Formalizations of Commonsense Reasoning*, AAAI Spring Symposium, Stanford University, 2011. AAAI.

[17] Brandon Bennett, Anthony G. Cohn, Frank Wolter, and Michael Zakharyaschev. Multi-Dimensional Modal Logic as a Framework for Spatio-Temporal Reasoning. *Applied Intelligence*, 17(3):239–251, 2002.

[18] Brandon Bennett and Antony P. Galton. A Unifying Semantics for Time and Events. *Artificial Intelligence*, 153(1-2):13–48, 2004.

[19] Brandon Bennett, David Mallenby, and Allan Third. An Ontology for Grounding Vague Geographic Terms. In *Proceedings of the Fifth International Conference on Formal Ontology in Information Systems (FOIS 2008), Saarbrücken, Germany, October 31st - November 3rd, 2008*, pages 280–293, 2008.

[20] Thomas Bittner and Barry Smith. A Theory of Granular Partitions. In M. Duckham, M. F. Goodchild, and M. F. Worboys, editors, *Foundations of Geographic Information Science*, pages 117–151. Taylor & Francis Books, London, 2003.

[21] Thomas Bittner and Barry Smith. Vague reference and Approximating Judgements. *Spatial Cognition and Computation*, 3(2):137–156, 2003.

[22] Patrick Blackburn, Johan F. A. K. van Benthem, and Frank Wolter, editors. *Handbook of Modal Logic*. Elsevier, New York, NY, USA, 2006.

[23] Davide Bresolin. *Proof Methods for Interval Temporal Logics*. PhD thesis, Dipartimento di Matematica e Informatica (Department of Mathematics and Computer Science), Universitá degli Studi di Udine (University of Udine, Italy), 2006.

[24] Davide Bresolin, D. della Monica, V. Goranko, A. Montanari, and G. Sciavicco. Decidable and Undecidable Fragments of Halpern and Shoham's Interval Temporal Logic: Towards a Complete Classification. In *Logic for*

Programming, Artificial Intelligence and Reasoning, 15th International Conference, LPAR 2008, Doha, Qatar, November 22-27, 2008, volume 5330 of *Lecture Notes in Computer Science*, pages 590–604. Springer, 2008.

[25] Christopher D. Brown and Herbert T. Davis. Receiver operating characteristics curves and related decision measures: a tutorial. *Chemometrics and Intelligent Laboratory Systems*, 80(1):24–38, January 2006.

[26] L. Callens, G. Carrault, F. Portet M.-O. Cordier, É. Fromont, and R. Quiniou. Intelligent adaptive monitoring for cardiac surveillance. In *Proceedings of European Conference on Artificial Intelligence (ECAI)*, 2008.

[27] G. Carrault, M. Cordier, R. Quiniou, and F. Wang. Temporal abstraction and inductive logic programming for arrythmia recognition from electrocardiograms. *Artificial Intelligence in Medicine*, 28:231–263, 2003.

[28] Anoop Cherian, Vassilios Morellas, and Nikolaos Papanikolopoulos. Accurate 3d ground plane estimation from a single image. In *IEEE International Conference on Robotics and Automation ICRA 2009, Kobe, 12th-17th May 2009*, 2009.

[29] Petr Cintula, Christian Fermuller, Lluis Godo, and Petr Hajek, editors. *Understanding Vagueness. Logical, Philosophical and Linguistic Perspectives*, volume 36 of *Studies in Logic*. College Publications, 2011.

[30] Anthony G. Cohn, Brandon Bennett, John Gooday, and Nicholas Mark Gotts. Qualitative spatial representation and reasoning with the region connection calculus. *GeoInformatica*, 1(3):275–316, October 1997.

[31] Anthony G. Cohn and Nicholas Mark Gotts. Representing Spatial Vagueness: a Mereological Approach. In L.C. Aiello, J. Doyle, and S. C. Shapiro, editors, *Proceedings of the 5th International Conference on Principles of Knowledge Representation and Reasoning (KR-96)*, pages 230–241, 1996.

[32] D. Conrad and G.N. De Souza. Homography-based ground plane detection for mobile robot navigation using a modified em algorithm. In *2010 IEEE Conference on Robotics and Automation, Anchorage, Alaska, USA, May 3-8 2010*, pages 910–915, 2010.

[33] DARPA: Defense Advanced Research Projects Agency. Mind's Eye Project Homepage,. `http://www.darpa.mil/Our_Work/I2O/Programs/Minds_Eye.aspx`. Last visited January 2012.

[34] DARPA: Defense Advanced Research Projects Agency. DARPA Mind's Eye Program. Project specification, March 2010.

[35] M. Denecker and A. Kakas. Special issue: Abductive logic programming. *Journal of Logic Programming*, 44(1–3):1–4, July 2000.

[36] Marc Denecker and Antonis Kakas. *Computational Logic: Logic Programming and Beyond*, volume 2407 of *Lecture Notes in Computer Science*, chapter Abduction in Logic Programming, pages 402–436. Springer, 2002.

[37] Tommaso D'Odorico and Brandon Bennett. Detecting events in video data using a formal ontology of motion verbs. In Cyrill Stachniss, Kerstin Schill, and David H. Uttal, editors, *Spatial Cognition VIII - Proceedings of the 8th International Conference on Spatial Cognition, Kloster Seeon, Germany, August 31 - September 3 2012.*, volume 7463 of *Lecture Notes in Artificial Intelligence*, pages 338–357, Berlin Heidelberg, 2012. Springer.

[38] Tommaso D'Odorico and Brandon Bennett. Automated reasoning on vague concepts using formal ontologies, with an application to event detection on video data. In Loizos Michael, Charlie Ortiz, and Benjamin Johnston, editors, *Commonsense 2013 - Proceedings of the 11th International Symposium on Logical Formalizations of Commonsense Reasoning, Aya Napa, Cyprus, May 27 - 29 2013.*, 2013.

[39] D. Doermann and D. Mihalcik. Tools and Techniques for Video Performance Evaluation. In *Proceedings of the 15th International Conference on Pattern Recognition*, volume 4, pages 167–170, 2000.

[40] P. Domingos and D. Lowd. *Markov Logic: An Interface Layer for Artificial Intelligence*. Morgan & Claypool Publishers, 2009.

[41] Krishna S. R. Dubba. *Learning Relational Event Models from Video*. PhD thesis, 2012.

[42] Krishna S. R. Dubba, Anthony G. Cohn, and David C. Hogg. Event Model Learning from Complex Videos using ILP. In *proceedings of ECAI 2010 - 19th European Conference on Artificial Intelligence, Lisbon, Portugal, August 16-20*, volume 215 of *Frontiers in Artificial Intelligence and Applications*, pages 93–98. IOS Press, 2010.

[43] E. Allen Emerson and Joseph Y. Halpern. Decision Procedures and Expressiveness in the Temporal Logic of Branching Time. *Journal of Computer and System Sciences*, 30(1):1–24, February 1985.

[44] E. Allen Emerson and Joseph Y. Halpern. "Sometimes" and "not never" revisited: on Branching versus Linear Time Temporal Logic. *Journal of the ACM*, 33(1):151–178, January 1986.

[45] Tom Fawcett. ROC Graphs: Notes and Practical Considerations for Researchers. Technical Report No. HPL-2003-4, HP Laboratories, March 2004.

[46] Tim Fernando. Representing Events and Discourse; comments on Hamm, Kamp and van Lambalgen. *Theoretical Linguistics*, 32(1):57–64, 2006.

[47] K. Fine. Vagueness, truth and logic. *Synthese*, 30:263–300, 1975.

[48] Silvia Gaio. Granular Models for Vague Predicates. volume 183 of *Frontiers in Artificial Intelligence and Applications*. IOS Press, 2008.

[49] Antony Galton. Experience and History: Processes and their Relations to Events. *Journal of Logic and Computation*, 18(3):323–340, June 2008.

[50] Antony Galton. Spatial and Temporal Knowledge Representation. *Earth Science Informatics*, 3(3):169–187, 2009.

[51] Joseph A. Goguen. The Logic of Inexact Concepts. *Synthese*, 19(3-4):325–373, 1969.

[52] Verena Gottschling. Keeping the Conversational Score: Constraints for an Optimal Contexualist Answer? In Elke Brendel and Christoph Jager, editors, *Contextualisms in Epistemology*, pages 153–172. Springer, 2005.

[53] Joseph Y. Halpern. Intransitivity and Vagueness. AAAI Press, 2004.

[54] Joseph Y. Halpern and Yoav Shoham. A Propositional Modal Logic of Time Intervals. *Journal of the ACM*, 38(4):935–962, 1991.

[55] Charles Leonard Hamblin. *Fallacies*. Methuen & Co., 1970.

[56] Hans Kamp. Two Theories about Adjectives. In Edward L. Keenan, editor, *Formal Semantics of Natural Language*, pages 123–155. Cambridge University Press, 1975.

[57] Hans Kamp. Events, Instants and Temporal Reference. In R. Bäuerle, U. Egli, and A. von Stechow, editors, *Semantics from Different Points of View*, volume 6 of *Springer Series in Language and Communication*, pages 376–417. Springer-Verlag, 1979.

[58] Hans Kamp. The paradox of the heap. *Aspects of Philosophical Logic*, pages 225–277, 1981. Dordrecht: Reidel.

[59] Hans Kamp. A Calculus for First Order Discourse Representation Structures. *Journal of Logic, Language and Information*, 5(3-4):297–348, October 1996.

[60] Rosanna Keefe. *Theories of Vagueness*. Cambridge University Press, 2000.

[61] Rosanna Keefe. Vagueness: supervaluationism. *Philosophy Compass*, 3(2):315–324, March 2008.

[62] Rosanna Keefe and Peter Smith. *Vagueness: a reader*. MIT press, 1997.

[63] Stasinos Konstantopoulos, Rui Camacho, Nuno A. Fonseca, and Vitor Santos Costa. *Artificial Intelligence for Advanced Problem Solving Techniques*, chapter Induction as a search, pages 158–205. Idea Group, Hershey, PA, USA, 2008.

[64] Robert A. Kowalski and Marek J. Sergot. A Logic-based Calculus of Events. *New Generation Computing*, 4(1):67–95, 1986.

[65] Michiel Van Lambalgen and Fritz Hamm. *The Proper Treatment of Events*. Wiley-Blackwell, May 2008.

[66] Language and Media Processing Laboratory. ViPER: The Video Performance Evaluation Resource. Last visited January 2012.

[67] Jonathan Lawry. Appropriateness measures: an uncertainty model for vague concepts. *Synthese*, 161(2):255–269, 2008.

[68] Jonathan Lawry and Yongchuan Tang. Uncertainty modelling for vague concepts: a prototype theory approach. *Artificial Intelligence*, 173:1539–1558, 2009.

[69] Beth Levin. *English Verb Classes and Alternations - A Preliminary Investigation*. The University of Chicago Press, 1993.

[70] D. Lewis. *On the Plurality of Worlds*. Blackwell, Oxford, 1986.

[71] David K. Lewis. Scorekeeping in a Language Game. *Journal of Philosophical Logic*, 8(1):339–359, January 1979.

[72] Kamal Lodaya. Sharpening the Undecidability of Interval Temporal Logic. In Jifeng He and Masahiko Sato, editors, *Advances in Computing Science - ASIAN 2000, 6th Asian Computing Science Conference, Penang, Malaysia, November 25-27, 2000*, volume 1961 of *Lecture Notes in Computer Science*, pages 290–298. Springer, 2000.

[73] Carsten Lutz and Frank Wolter. Modal Logics of Topological Relations. *Logical Methods in Computer Science*, 2(2):1–31, 2006.

[74] Vladimir Y. Mariano, Junghye Min, Jin-Hyeong Park, Rangachar Kasturi, David Mihalcik, Huiping Li, David Doermann, and Thomas Drayer. Performance Evaluation of Object Detection Algorithms. In *Proceedings of the 16th International Conference on Pattern Recognition*, volume 3, pages 965–969, 2002.

[75] John McCarthy and Patrick J. Hayes. Some Philosophical Problems from the Standpoint of Artificial Intelligence. In B. Meltzer and D. Michie, editors, *Machine Intelligence*, volume 4, pages 463–502. Edinburgh University Press, 1969.

[76] Rob Miller and Murray Shanahan. The event calculus in classical logic - alternative axiomatisations. *Electronic Transactions in Artificial Intelligence*, 3(A):77–105, 1999.

[77] S. Muggleton. Inductive logic programming. *New Generation Computing*, 8(4):295–318, 1991.

[78] S. Muggleton. Inverse entailment and progol. *New Generation Computing*, 13(3-4):245–286, 1995.

[79] Stephen Muggleton and Christopher H. Bryant. Theory completion using inverse entailment. In *Inductive Logic Programming, 10th International Conference, ILP 2000, London, UK, July 24-27, 2000, Proceedings*, volume 1866 of *Lecture Notes in Computer Science*, pages 130–146. Springer, 2000.

[80] Stephen Muggleton and Luc De Raedt. Inductive logic programming: Theory and methods. *Journal of Logic Programming*, 19(20):629–679, 1994.

[81] Adrian Paschke. Eca-ruleml: An approach combining eca rules with temporal interval-based kr event/action logics and transactional update logics. Technical Report 11, Technische Universität München, November 2005.

[82] David M.W. Powers. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. *Journal of Machine Learning Technologies*, 2(1):37–63, February 2011.

[83] J.R. Quinlan and R.M. Cameron Jones. Induction of logic programs: Foil and related systems. *New Generation Computing*, 13:287–312, 1995.

[84] Randolph Quirk, Sidney Greenbaum, and Geoffrey Leech. *A Grammar of Contemporary English*. Longman, Harlow, 1972.

[85] L. De Raedt. Logical settings for concept-learning. *Artificial Intelligence*, 95(1):187–201, 1997.

[86] Luc De Raedt and Kristian Kersting. Probabilistic inductive logic programming. In Luc de Raedt, Paolo Frasconi, Kristian Kersting, and Stephen Muggleton, editors, *Probabilistic Inductive Logic Programming - Theory and Applications*, volume 4911 of *Lecture Notes in Computer Science*, pages 1–27. Springer, 2008.

[87] David A. Randell, Zhan Cui, and Anthony G. Cohn. A Spatial Logic Based on Regions and Connection. In Bernhard Nebel, Charles Rich, and William R. Swartout, editors, *Proceedings of the 3rd International Conference on Principles of Knowledge Representation and Reasoning (KR'92). Cambridge, MA, October 25-29, 1992*, pages 165–176. Morgan Kaufmann, 1992.

[88] O. Ray. Non-monotonic abductive inductive learning. *Journal of Applied Logic*, 7(3):329–340, 2009.

[89] Raymond Reiter. The Frame Problem in the Situation Calculus: a Simple Solution (sometimes) and a Completeness Result for Goal Regression.

In Vladimir Lifshitz, editor, *Artificial Intelligence and Mathematical Theory of Computation: papers in honour of John McCarthy*, pages 359–380. Academic Press Professional Inc., San Diego, CA, USA, 1991.

[90] Jochen Renz and Bernhard Nebel. On the Complexity of Qualitative Spatial Reasoning: A Maximal Tractable Fragment of Region Connection Calculus. *Artificial Intelligence*, 108(1–2):69–123, 1999.

[91] M. Richardson and P. Domingos. Markov logic networks. *Machine Learning*, 62(1–2):107–136, 2006.

[92] Chaman L. Sabharwal, Jennifer L. Leopold, and Nathan Eloe. A more expressive 3d region connection calculus. In *Proceedings of the 17th International Conference on Distributed Multimedia Systems, DMS 2011, 18th-20th October 2011, Florence, Italy.*, pages 307–311. Knowledge Systems Institute, 2011.

[93] Steven Schockaert, Martine de Cock, and Etienne E. Kerre. Spatial Reasoning in a Fuzzy Region Connection Calculus. *Artificial Intelligence*, 173(2):258–298, 2009.

[94] Stehpen Se and Michael Brady. Ground plane estimation, error analysis and applications. *Robotics and Autonomous Systems*, 39:59–71, 2002.

[95] Murray Shanahan. The Event Calculus Explained. In *Artificial Intelligence Today: Recent Trends and Developments*, volume 1600 of *Lecture Notes in Computer Science*, pages 409–430. Springer Berling / Heidelberg, 1999.

[96] Jeffrey Mark Siskind. Reconstructing force-dynamic models from video sequences. *Art*, 151, 2003.

[97] Nicholas J. J. Smith. *Vagueness and Degrees of Truth*. Oxford University Press, November 2008.

[98] R. A. Sorensen. *Vagueness and Contradiction*. Oxford University Press, 2001.

[99] Muralikrishna Sridhar, Anthony G. Cohn, and David C. Hogg. Unsupervised Learning of Event Classes from Video. In *proceedings of the 24th AAAI Conference on Artificial Intelligence, AAAI 2010, Atlanta, Georgia, USA, July 11-15*, AAAI 2010, pages 1631–1638, 2010.

[100] Thora Tenbrink, Tommaso D'Odorico, C. Hertzberg, S. Güzin Mazman, C. Meneghetti, N. Reshöft, and J. Yang. Tutorial report: Understanding spatial thought through language use. *Journal of Spatial Information Science*, 5(1):107–114, 2012.

[101] Son D. Tran and Larry S. Davis. Event modeling and recognition using markov logic networks. In *Computer Vision - ECCV 2008*, volume 5303 of *Lecture Notes in Computer Science*, pages 610–623. Springer, 2008.

[102] Michael Tye. Vague objects. *Mind*, XCIX(396):535–557, October 1990.

[103] Kees van Deemter. *Not Exactly. In Praise of Vaueness*. Oxford University Press, 2010.

[104] Michael van den Bergh, Esther Koller-Meier, and Luc Van Gool. Fast body posture estimation using volumetric features. In *IEEE Visual Motion Computing (MOTION)*, January 2008.

[105] Jan van Eijck and Hans Kamp. Representing Discourse in Context. In *Handbook of Logics and Linguistics*, pages 179–237. Elsevier, 1996.

[106] A. C. Varzi. Vagueness in georgraphy. *Philosophy and Geography*, 4:49–65, 2001.

[107] Zeno Vendler. Verbs and Times. *The Philosophical Review*, 66(2):143–160, April 1957.

[108] Zeno Vendler. *Linguistics in Philosophy*. Cornell University Press, Ithaca, NY, USA, first edition, 1967.

[109] Yde Venema. A Modal Logic for Chopping Intervals. *Journal of Logic and Computation*, 1(4):453–476, 1991.

[110] Timothy Williamson. Vagueness and Ignorance. *Proceedings of the Aristotelian Society, Supplementary Volumes*, 66:145–177, 1992.

[111] Timothy Williamson. *Vagueness*. The Problems of Philosophy. Routledge, 1998 edition, 1994.

[112] Timothy Williamson. On the Structure of Higher-Order Vagueness. *MIND, New Series*, 108(429):127–143, January 1999.

[113] Timothy Williamson. Vagueness in Reality. In Michael J. Loux and Dean W. Zimmerman, editors, *The Oxford Handbook of Metaphysics*. Oxford University Press, 2003.

[114] Jihong O. Yang, Quian Fu, and Dayou Liu. A model for representing topological relations between simple concave regions. In *Computational Science – ICCS 2007*, volume 4487 of *Lecture Notes in Computer Science*, pages 160–167. Springer, 2007.

[115] Angela Yao, Juergen Gall, and Luc Van Gool. Coupled action recognition and pose estimation from multiple views. *International Journal of Computer Vision (IJCV)*, 100(1):16–37, 2012.

[116] Lotfi A. Zadeh. Fuzzy Logic and Approximate Reasoning. *Synthese*, 30(3-4):407–428, September 1975.

Index

This index is intended as a reference for the symbolic notation introduced throughout the thesis. Numbers in **bold** refer to the page where the concept, symbol or relation is first introduced, defined or substantially discussed.