Sandbox Geography

Final

Florian Alfred Twaroch

February 15, 2007

DISSERTATION

Sandbox Geography

How to Structure Space in Formal Models

ausgeführt zum Zwecke der Erlangung des akademischen Grades eines Doktors der technischen Wissenschaften unter der Leitung von

Univ. Prof. Dipl. Ing. Dr. techn. Andrew Frank
E127
Institut für Geoinformation und Kartographie

eingereicht an der Technischen Universität Wien
Fakultät für Informatik

von

Dipl. Ing. Florian Alfred Twaroch
Matr. Nr. 93 25 388
Strohberggasse 15/1/9
1120 Wien

Wien, am 20.02.2007

TU VIENNA

Technische Universität Wien
Vienna University of Technology

DISSERTATION

Sandbox Geography

How to Structure Space in Formal Models

A thesis submitted in partial fulfillment of the requirements for the degree of
Doctor of Technical Sciences

submitted to the Vienna University of Technology
Faculty of Informatics

by

Dipl. Ing. Florian Alfred Twaroch
Strohberggasse 15/1/9
1120 Vienna

Advisory Committee:

Prof. Andrew U. Frank, Ph.D.
Institute for Geoinformation and Cartography
Vienna University of Technology

Prof. Christian Freksa, Ph.D.
Department of Informatics and Mathematics, Cognitive Systems Group
University of Bremen

Vienna, 20.02.2007

For Darja, Luka, and Rebecca,

and my parents Alfred and Christina.

All I Ever Really Need to Know I Learned in Kindergarten.

Most of what I really need to know about how to live, and what to do, and how to be, I learned in Kindergarten. Wisdom was not at the top of the graduate school mountain, but there in the sandpile at Sunday School.

These are the things I learned:

Share everything.

Play fair.

Don't hit people.

Put things back where you found them.

Clean up your own mess.

Don't take things that aren't yours.

Say you're sorry when you hurt somebody.

Wash your hands before you eat.

Flush.

Warm cookies and cold milk are good for you.

Live a balanced life - learn some and think some and draw and paint and sing and dance and play and work every day some.

Take a nap every afternoon.

When you go out into the world, watch out for traffic, hold hands, and stick together.

Robert Fulghum (1993, pp. 6-7)

Acknowledgements

In the first place I would like to thank Andrew Frank for his advice during the whole dissertation project. For letting me try my own ideas and discussing the assets and drawbacks of different approaches. I have learned much more than writing a PhD under his supervision.

I also appreciate the help of George Luger with his pointers to relevant AI literature and his willingness to support the thesis. Furthermore I would like to thank Stephan Winter for joining my first steps into the GI Science community. I also thank the late Karl Kraus for giving me a phone call in January 2003, motivating me to start a PhD project. I like to thank all the colleagues of the community for valuable comments and discussions especially Gilberto Camara, Andrea Rodriguez, Sabine Timpf and Christoph Schlieder. Many thanks for the support to Sue Hespos and Renee Baillargeon who provided me with photos of empirical studies of their websites.

Writing a PhD is quite an egoistic project. Without the support of my family I could never have gone that far. I would like to thank my wife Darja for her love and patience during this project and I am sorry for all the moments I could not share with her when I was writing my PhD. I also missed the times I could not play with Luka and Rebecca. They gave me all the motivation to finish this project. This is also the place to thank my parents Alfred and Christina for their love, for being there whenever needed and for giving me the opportunity to attend a university and for letting me go my own way. I thank all my family and friends for sharing the highs and lows with me in the course of this thesis.

Finally I like to thank all the colleagues at the Institute for Geoinformation and Cartography for many valuable discussions and pointers. I think we shared a good time together for the last four years.

Tudi vam moja družina in prijatelji v Sloveniji se zahvalujem - brez vas ne bi nikoli končal tega projekta. Hvala za vašo ljubezen ki me je spremeljala skozi

težke trenutke, za vaš čas, kadar ste pazili otroke ko so največ nagajali in za vaše razumevanje ko sem bil slabe volje za radi doktorata. Imam vas rad in tudi vam posvečam to delo.

Abstract

"Hard sciences" like physics and geometry define how to build models of spatial reality into a geographical information system. This results in systems lacking user friendliness and suffering from low acceptance because humans conceptualize spatial reality differently. There is a need for formal models of human conceptualizations of the world that are based on common sense respectively naive conceptualizations.

Naive geography studies formal models of the commonsense geographic world. The vision of naive geography is a set of theories that helps to build geographic information systems that can be used without major training by new users. Success in finding this set of theories has been limited as a common problem is to find an axiomatization for the formal models.

The present research formalizes human spatial conceptualizations using algebraic specifications. The theories are based on a set of sorts, operations and axioms. I hypothesize that the change of spatial theories can be modeled by an adaptation of axioms.

Selected examples of developmental psychology serve as an input for building a framework for the acquisition of spatial theories. The *theory theory* of cognitive development motivated sequences of spatial theories presented in this work. For three sequence types of theories theory building mechanisms are identified. The adaptation of theories is based on:

1. **Specialization** considers a new influential parameter. A theory is specialized by constraining it through an axiom. The axioms constrain the theory to a special set of sorts and operations. The more axioms are added the fewer sorts can be described by the specialized theory.

2. **Generalization** is an abstraction step. A theory is found to be a special case of a more general theory. The general theory is obtained by deleting

an axiom of the specialized theory. Other special theories can be derived from a generalized theory by adding axioms. The derived and coexisting theories are special cases of the generalized theory.

3. **Dynamic Weighting** is a mechanism to assign importance to a theory. Theories with higher weights are favored over those with lower weights. Belief revision is the result of the dynamic weighting mechanism that assigns a higher weight to a previously low weighted theory.

The contribution of this thesis is a formal description of spatial theories as found with children. The novelty is in the formal description of the transition from one theory to another. The formal model describes the spatial theories and their change in a framework using algebraic specifications.

The algebraic specifications have been implemented in a purely functional programming language, which makes them executable. The framework allows to simulate the developed change mechanism in accordance with the empirical studies carried out in developmental psychology. It takes abstract perceptions as input, evaluates a set of given theories and responds about the appropriateness of the theories in a given environment. Frequent mismatches between observations in the environment and expectations generated by the theories will elicit changes in the algebraic structure.

In conclusion a set of mechanisms based on three theory building operations is shown to be theoretically capable to construct sequences of ever improved commonsense theories of space. Future work will address the automation of the mechanism in a multi agent environment. The influence of communication processes on spatial concept formation is still an open question of research.

Keywords

spatial cognition, naive geography, ontologies, algebraic modeling, conceptual change

Kurzfassung

Geoinformationssysteme sollen Menschen helfen, räumliche Entscheidungen rascher zu treffen. Die Systeme werden aber ohne die Berücksichtigung naiv kognitiver Theorien implementiert. Der Grund ist ein Mangel an formalen Beschreibungen alltäglicher kognitiver Theorien zur räumlichen Entscheidungsfindung.

Die vorliegende Dissertation ist durch empirische Studien zur Raumkognition von Kindern motiviert. Ausgangspunkt ist die *Theorie Theorie* der kognitiven Entwicklung. Die *Theorie Theorie* besagt, dass Kinder wie Wissenschaftler Theorien über die sie umgebende Umwelt bilden. Die Theorien entstehen durch Beobachtung der Umwelt und dienen der Prädiktion von Phänomenen. Immer wenn Prädiktion und Beobachtung nicht in Übereinstimmung gebracht werden können, wird ein Theoriewechsel ausgelöst.

Der Beitrag dieser Dissertation ist eine formale Beschreibung naiv räumlicher Theorien in Anlehnung an die *Theorie Theorie*. Neu ist die Beschreibung eines Revisionsmechanismus von einer Theorie zur folgenden. Ein formales Modell beschreibt die räumlichen Theorien und den Revisionsmechanismus mittels eines algebraischen Ansatzes. Für drei Sequenzen räumlicher Theorien wurden drei Mechanismen zum algebraischen Theoriewechsel identifiziert:

1. **Spezialisierung** berücksichtigt den Einfluss eines neuen perzeptiven Parameters. Eine Theorie wird spezialisiert durch Hinzufügen eines Axioms. Die Axiome beschränken die Theorie auf eine spezielle Menge von Elementen. Je mehr Axiome hinzugefügt werden, desto weniger Elemente können durch die spezialisierte Theorie beschrieben werden.

2. **Generalisierung** ist ein Abstraktionsschritt. Eine Theorie kann durch mehrmaliges Hinzufügen von Axiomen überspezialisiert werden. Eine überspezialisierte Theorie erhält in Folge durch Beobachtung der Umwelt abwechselnd Bestätigung und Widerspruch. Die überspezialisierte Theorie

kann durch Entfernen eines Axioms in eine generalisierte Theorie übergeführt werden. Die generalisierte Theorie kann durch Hinzufügen von Axiomen wieder in andere spezialisierte Theorien übergeführt werden. Diese abgeleiteten, nebeneinander existierenden Theorien sind Spezialfälle einer generalisierten Theorie.

3. **Dynamische Gewichtung** ist ein Mechanimus, um die Wichtigkeit einer Theorie zu modellieren. Das Gewicht einer Theorie ist ein Mass für die Funktion der Theorie in einer Umwelt. Theorien mit höheren Gewichten werden Theorien mit niedrigeren Gewichten bevorzugt. Theoriewechsel resultieren aus der dynamischen Zuordnung hoher Gewichte auf Theorien, die zunächst niedrig gewichtet waren.

Die algebraischen Spezifikationen wurden mit einer rein funktionalen Programmiersprache implementiert und machen das vorgestellte Model ausführbar. Experimente der Entwicklungspsychologie können mit dem Modell unter Verwendung der Revisionsmechanismen simuliert werden. Als Eingabe dienen dem Modell abstrakte Perzeptionen, die anhand einer vorgegebenen Menge von Theorien verarbeitet werden. Wiederholte Widersprüche zwischen Beobachtungen in der Umwelt und den von den Theorien generierten Erwartungen lösen Theoriewechsel in der algebraischen Struktur aus. Anhand der Simulationen konnte die Plausibilität des Modells überprüft werden.

Es lässt sich der Schluss ziehen, dass Sequenzen naiv räumlicher Theorien durch drei Mechanismen in einem algebraischen Rahmenwerk gebildet werden können. In Zukunft soll die Automatisierung des Mechanismus in einer Multi-Agenten Umgebung erforscht werden. Die Bedeutung von Kommunikationsprozessen zwischen Agenten und deren Einfluss auf räumliche Theoriesequenzen soll untersucht werden.

Schlüsselwörter

Räumliche Kognition, Naive Geographie, Entwicklungspsychologie, Ontologien, Algebraische Modellierung, Konzeptwechsel

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Motivated by the observation of growing children this thesis proposes a model for the revision of spatial theories. These theories are formally described by algebraic specifications in an agent based framework. The mechanism for theory revision is based on adding, deleting and weighting axioms. Simulations that utilize empirical data of developmental psychology are carried out. The chapter introduces the methodology used and the results expected. An outlook to applications is given and the organization of the whole thesis is explained.

1.1 Motivation of the current work

Spatial representations in contemporary geographic information systems are based on "hard sciences" like geometry, mathematics and physics. As a result available geographic information systems are not widely accepted by laymen. People base their everyday spatial reasoning on other conceptualizations.

Recent findings in developmental psychology (Baillargeon, 2004a) suggest that already children as young as 12 months hold concepts about objects, the movement of objects, gravity, inertia, occlusion, containment and many others. The development of these conceptualizations motivated the present thesis that investigates how to formalize spatial concepts as found with children. The focus of the thesis is put on spatial aspects of conceptualizations and their change.

Some developmental psychologists conjecture that infants learn about the world by forming and revising theories (Gopnik et al., 1999). These are not big theories in the sense of Darwin's theory of evolution or Einstein's relativity theory, but small conceptualizations of the world that are adapted in the light of new evidence. Ontologists would rather refer to “theoritas” to distinguish them from fully fledged theories (Casati, 2000).

Efforts to formally describe commonsense theories led to the proposal of naive physics (Hayes, 1978). Hayes formalized the naive understanding of solids and liquids (Hayes, 1985). These ontologies inspired a series of other formal theories about the commonsense world (Hobbs and Moore, 1985).

Naive geography studies formal models of the common sense geographic world. (Mark, 1993; Egenhofer and Mark, 1995; Mark and Egenhofer, 1996). Naive geography is a set of theories that helps to build geographic information systems that can be used without major training by new users. Success in finding this set of common sense theories has been limited, because a common problem is to define axioms for the formal models.

The present thesis is a contribution to naive geography. It investigates a theory for the formal description of spatial theory change with children. Understanding a theory development calculus gives an overall structural framework for naive geography, actually a framework for naive theories of any domain.

The advantage of spatial theories is that they are based on simple observations. The theories can be described by a limited set of operations and axioms. Empirical studies by developmental psychologists provide such data.

The mechanisms children use to develop theories are constant over lifetime. Adults reuse what they have learned as children. I assume that the process of spatial theory development can be compared with the bootstrapping of knowledge in a computer. A small set of given theories is transformed into a complex framework of interacting theories through active exploration of an environment.

The research introduces a theory driven agent that explores its environment. It is a wrapper to a set of mechanisms for theory change. This framework will help to understand the transition of one theory to another. Each theory stands for a conceptual model of the agent. The research works towards a vision of geoinformation science. A geographic information system must be capable of integrating different conceptual models in a single formal system. The long term aim is to treat different representations of space in a uniform way (Frank, 2001).

1.2 Hypothesis and Research Question

The goal of this thesis is a formalization of how people build mental models about spatial phenomena. These mental models will be described as sequences of changing theories. Revision of a (spatial) concept becomes necessary when the predictions generated by a theory do not fit to the according perceptions made in an environment.

In the course of this thesis a theory of the acquisition of spatial concepts is developed based on research in cognitive development. Learning and cognitive development address how people build concepts about the surrounding world. The central question of the present thesis can be stated as how to *formally build* a conceptual schema of space that allows continuous revision whenever new evidence brings up contradiction? The resulting model is a formal specification towards naive geographic information systems (Egenhofer and Mark, 1995).

To formalize the mental models of spatial phenomena the tool of algebraic specification has been chosen. Algebra allows a high level of abstraction. With a pure functional programming language executable models can be built.

Theory revision in the terms of this thesis should be understood as *adaptation*, meaning making a theory fit to observations made in an environment. The refined hypothesis of this thesis states that *a theory of space can be described by a set of axioms. It is possible to adapt the theory by the addition, deletion and dynamic weighting of its axioms.*

The theories under investigation are not necessarily just common sense spatial theories of children. They endure in a revised form in adults (Karmiloff-Smith, 1992; Gopnik and Meltzoff, 1997). Here formal models about spatial conceptualizations together with a mechanism of change apply equally to children and adults. The formal study of early spatial conceptualizations will lead to the identification of elements that are vital to the design of sound geographic information systems.

1.3 Approach and Research Design

Data of empirical experiments carried out in developmental psychology are used to construct a model of a theory driven agent in an environment. These studies provide prelinguistic data and present experiences of infants with table top objects in a cognition laboratory. The interpretations of the studies have been used to develop algebraic specifications.

Series of studies point to developmental processes in the child and thus to theory change. In the course of the thesis a calculus of theory change based on three mechanisms has been developed. The mechanisms have been found in the empirical data and are based on an adaptation of axioms. Theories are built by either adding or deleting axioms.

The theory driven agent can hold several theories about the same phenomena at a time. The agent evaluates the theories using a dynamic weighting mechanism given observations in the environment. The importance of a theory can be determined through its weight.

In order to build the framework of the theory driven agent in an environment previous work has been reviewed in the area of problem solving, belief revision, learning systems and agent theory. A calculus of theory change has been proposed and implemented into an executable computational model. The model is evaluated by the simulation of selected empirical studies.

1.3.1 Conceptual Model - Sandbox Geography

A formal theory will be provided that describes spatial theory change in a computational model. The model describes how spatial beliefs change during infancy towards adult's naive theories of space. In order to simulate this process in a model an artificial environment has been set up.

The theory about the acquisition of spatial concepts is worked out in a sandbox. I am using the metaphor of a sandbox as a place for experimentation; the laws of physics can be investigated by using very simple models. The models in a sandbox do not last, but they can raise new insights in the little engineer's understanding. The objects treated in a sandbox underlie a mesoscopic partitioning (Smith and Mark, 2001), they are on human scale and they belong to categories that geographers form, therefore Sandbox Geography.

The term geography has greek roots and comprises the description of the earth. Before children start to describe large scale space environments they start to describe their immediate surrounding. These are table top spaces. The current research does not investigate geographic phenomena but table top environments. In conformance with the theory theory I assume that the mechanisms used by children are transferred to adults and that the models investigated in this thesis can be transferred to geographic space environments in a later step. The formal models provided are a basis to describe geographic phenomena and stand at the starting point of bootstrapping process.

A theory driven agent is endowed with an initial set of theories about an environment. The agent uses theories and observations of the environment to predict the outcome of spatial operations. Whenever these predictions fail to describe spatial phenomena the agent builds new theories triggered by observations available. New theories are again tested in the environment by observation until the agent loses interest.

The conceptual model has been worked out by the identification of spatial processes in empirical studies of developmental psychology. Previous work in philosophy, psychology, linguistics, geography, computer science, artificial intelligence and robotics has been reviewed in order to build the model of a spatial cognizing agent in an environment. The intention is not to provide a new theory of learning or to build a cognitive architecture.

Sandbox Geography investigates simple spatial situations to find out how space is structured in mental models. The mental models are described as sequences of changing theories. Mechanisms to revise theories from infant's towards adult's conceptualizations of space are provided. The formal description of the mental models together with mechanisms for change represent people's beliefs, i.e. children and adults, about space.

1.3.2 Formalization - Computational Model

The formalization of the conceptual model is carried out using algebraic specifications. The reasons to choose an algebraic approach are multiple. Firstly operations and axioms can be used to describe activities. Operations on the same sorts can be grouped in algebras. The agent based approach is based on activities in an environment. Secondly algebras provide abstract mechanisms that can be used to investigate the transition between different conceptualizations. Thirdly algebraic specifications together with a functional programming language serves as a rapid prototyping tool in several investigations in geographic information science.

An executable model is developed using the functional programming language Haskell. It allows the direct implementation of algebras that have been defined in the conceptual model. The empirical studies designed and implemented can be executed using the prototype.

The formal model helps to keep the conceptual model clean, as one has to be very specific in setting up the model. A sound formalization depends on the decisions which elements, objects and processes are included into the model. The model is an abstract description of reality and contains just the elements necessary to support the hypothesis of the thesis.

The resulting executable formal model is a proof of concept for the theory used. In the present thesis this is the *theory theory* of cognitive development (Gopnik and Meltzoff, 1997; Gopnik et al., 1999). An important follow up step is the testing of the formal model.

1.3.3 Testing the formal model

The testing of the formal model shows whether the spatial cognizing agent reflects the behavior of the subjects involved in the empirical studies carried out in developmental psychology. Major design errors in the conceptual model can be detected by carrying out simulations. The prototypical implementation is a proof of concept for the designed model and verifies the stated hypothesis.

New research questions in other disciplines can be gained by the results. Missing input from the empirical studies for the spatial domain can be identified. The computer simulation further validates the underlying *theory theory* of cognitive development and is a proof for its plausibility.

1.4 Expected Results

The thesis provides a formalization of spatial concepts as found with infants towards those of adults. Sequences of spatial theories are described in a computational model. The focus of the model is on the acquisition of spatial phenomena.

The thesis provides an agent based approach to spatial conceptualizations with infants. In the center of the research there are objects in a small scale space as found in the empirical studies of developmental psychology. The structure of these common sense conceptualizations and the possible level of abstraction are investigated towards mechanisms for conceptual change. The expected results are:

- An abstract model of conceptual change that is grounded in people's real world experience. The aim is to provide formal theories as needed in geographical information systems. Previous work has been mostly based on block worlds and toy space (Frank, 1998; Egenhofer and Rodriguez, 1999). The model built in the present thesis is based on empirical studies. The formalization relies on the interpretations of experiments given by experts, i.e. psychologists.

- A method to build a conceptual schema from empirical studies carried out in developmental psychology. The derivation of axioms for a formal specification of spatial concepts is described. The following steps are necessary:
  - Find empirical studies describing spatial phenomena.
  - Abstract processes and map them to data types and operations.
  - Build axioms to define the behavior of the operations.
  - Simulate the resulting model with a purely functional programming language and compare the behavior of the model with the outcome of the empirical studies.
- A transition mechanism for spatial concepts based on algebraic specifications. The thesis will show that conceptual change can be modeled using algebraic specifications. Revised conceptualizations are obtained by adapting the axioms of an algebra. The adaptation of axioms is based on the addition, deletion and dynamic weighting of axioms.
- An overview of state of the art empirical studies about spatial knowledge as found with infants is provided. The selected body of research is formally described as algebraic theories and their change. The formal theories are a contribution to naive geography rather than being a proposal for a new cognitive architecture.
- The present, developed theories can be formally tested and give the experimenter data that can be cross checked with empirical studies. Thus it can help to keep theories in developmental psychology "clean", because one has to be very specific when building a computational model. Parameters of informal psychological descriptions are validated in a computer model.

In order to answer the research question within a limited time, several aspects have been excluded from the investigations. To avoid misunderstandings I state what this thesis is not about. Points that are not under investigation are:

- The thesis does not provide a new cognitive architecture. Effects due to memorization, strength of stimuli or attention have not been considered in the model. Conclusions about brain activities or other similarities to neural models cannot be drawn.

- It is not my intention to build a model that is exclusively about children. I assume naive theories identified with infants can be found in adults in a revised form (Gopnik and Meltzoff, 1997; Spelke, 2000; Carey, 2004). The presented epistemology and ontology of the agent grow incrementally, comparable with the bootstrapping mechanism of a computer.

- The formal model does not serve as an implementation that can be directly used in a robot or any kind of machine. Low level percepts, such as sensor data from vision or audio devices are abstracted in data types and functions. I deal with observations, empirical studies and physical objects as cognitive products available. Examples for theory construction in robots are the subsumption architecture by Brooks (1986) and the semantic spatial hierarchy by Kuipers (2000).

In summary the expected results are mechanisms to build sequences of spatial theories using an algebraic framework. The mechanisms are tested with a theory driven agent that predicts and observes the outcome of operations on objects in a table top environment. The model is validated by comparing the behavior of the agent in the environment with the behavior of infants in comparable empirical studies of developmental psychology.

1.5 Contribution of the thesis

This thesis contributes to user interfaces and interoperability in geographic information systems. I outline how the findings of this thesis can be useful for the areas mentioned. Firstly a case for user interfaces is discussed, then the contribution towards system interoperability is outlined.

Children who do not yet speak do often point to objects in order to communicate with adults. This pointing paradigm has been implemented in the user interface of a tourist information system (Irschitz, 2004). Empirical tests showed that users readily accepted the interface without any need to learn how to use it. The example shows how adults readily reuse what they have learned as children (see figure 1.1) and how it can be implemented in the user interface of an information system.

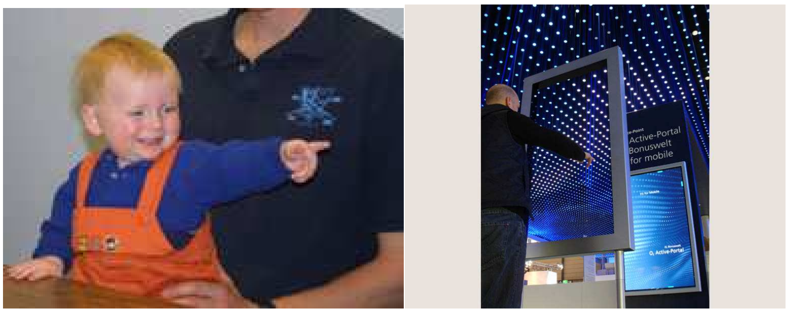


Figure 1.1: User interface with the pointing paradigm
(source: left picture: http://ilabs.washington.edu/news/press_releases/pr_brooks_11.2002.html, right picture: Irschitz (2004))

This thesis contributes to the interoperability of geographic information systems. Semantic interoperability in geographic information systems is about the transition of different mental models. Two system designers may have an intensional description of a part of the world in mind and cast them into a formal model. These models are usually different but describe the same external reality. The assumption in this thesis is that the transition between different formal models is based on the same naive concepts. The transition mechanism is constant through lifetime (Gopnik and Meltzoff, 1997) that is why the results of this research do not refer only to children but also to adults.

A formal investigation of children's mental models of space enables to describe formally naive spatial concepts. The formal treatment allows to find mechanisms of change between the concepts. These mechanisms of change are necessary to yield an automatic transition between concepts that can be used in a computer.

On the one hand future information systems will have to consider the mental models of their users to adapt information representation in according interfaces, on the other hand future information systems will have to merge automatically data from different sources, that have a different conceptual background. The present research is a step towards a deeper formal understanding of how people generate and revise naive spatial conceptualizations.

1.6 Target Audience

The research carried out is related to several disciplines. It is targeted at researchers particularly in the following areas:

- Geoinformation scientists: A method to define the axioms of common sense spatial theories based on the findings of developmental psychology is presented. An algebraic approach together with a functional programming language yields executable models. The proposed theory for the acquisition of spatial concepts is based on empirical data.

- Psychologists and cognitive scientists: Researchers can benefit from the executable computational models that are a validation tool for informal theories. The theories can be tested for formal correctness. Simulations are a common research tool in psychology (Schlesinger and Parisi, 2001) and can be utilized towards new research questions in the spatial domain.

- Artificial intelligence and robotics: Intelligent systems based on naive spatial theories can be built and implemented. The systems will show behaviors as found with infants. Humanoid robots like the infanoids (Hideki and Hiroyuki, 2001) can benefit from the naive spatial theories. Machine learning systems based on the provided mechanism of conceptual change can further be investigated.

- Implementers of vision systems: Reasoners in vision systems require a set of spatial relations for detected objects. Content based image retrieval could be based on the naive theories of objects as found with infants.

- Computer scientists and implementers of geographic information systems: User interfaces need to be based on people's beliefs and expectations about objects in space. The axioms of the common sense spatial theories can be translated into consistency rules for user interfaces. This will lead to systems that represent information as it is naively expected by the layman user.

1.7 Organization of the thesis

The following chapter 2 is dedicated to definitions and the contributing disciplines. The notion of theory as used in the present thesis is introduced. The *theory theory* of cognitive development which has given the motivation for the research project is outlined followed by a comparison of naive with scientific theories. Previous work on computational models of cognitive development is reviewed.

In chapter 3 I review work done by geographic information scientists to provide models for naive geography. I outline how an approach that relies on empirical data of developmental psychology can contribute to this project. I discuss the conceptual aspects of a model for the acquisition of spatial theories and justify the chosen modeling technique.

Chapter 4 gives place for developmental psychology. I describe studies that have been carried out to infer from what children know and learn in the first two years of their lives. Sequences of theories have been identified and serve to generate a mechanism for modeling sequences of spatial theories. Spatial theories for the occlusion, containment and support of objects are described.

In chapter 5 I develop the formal model of an agent that holds spatial theories that can change. Different instantiations of the agent stand for different stages of development. I describe a possible mechanism for change but concentrate on a formal description of changing theories that are necessary to describe the occlusion, containment and support of objects.

In chapter 6 I verify the hypothesis that qualitatively new spatial representations can be gained by the adaptation of axioms in a formal model based on algebraic specifications. The simulations carried out with the model that has been developed in chapter 5 are in accordance with the empirical data presented in chapter 4.

Chapter 7 concludes this thesis. The results and major findings are summarized. An outlook to future research questions is given. The complete Haskell code for the computational model can be found in the appendix of the thesis. Detailed tables for the empirical studies and the derived mechanisms can be found in the appendix. A closer look at the formal tools used throughout the thesis is made. A short introduction to algebra is given and the functional programming paradigm is presented.

Chapter 2

Theories and Theory Forming

The present thesis addresses the question how humans build a spatial concept. I argue that human naive understanding of space is based on a set of theories. These theories underlie qualitative change. On the one hand my goal is to provide a formal description of these spatial theories on the other hand I formalize a mechanism for the qualitative change of theories.

The research has been motivated by an account of developmental psychology, called the *theory theory*. It is a theory how people, especially children build theories of their surroundings. The theory theory proposes a strong parallelism between common sense and scientific theory formation.

A definition of the term theory as used in the present thesis is given and the *theory theory* of cognitive development is introduced. Theories in the scientific and naive realm are discussed and epistemological concerns of theory formation are reviewed. The chapter closes with a review of formal models of cognitive development, outlining different approaches to theory change.

2.1 Theory Theory - Making Sense of the world

Theories are often considered to be well-substantiated explanations (Fellbaum, 1998) and therefore like Einstein's theory of relativity or Darwin's theory of evolution. But the term *theory* can also stand for "a concept of a certain aspect of the world, that is not necessarily yet verified" (Fellbaum, 1998).

Theories of the latter kind can be described by using mathematics. A mathematical theory can be described by a set of formulas, i.e. the axioms of the theory. Einstein's theory of relativity can be written down on a few pages of paper.

I assume that theories are simple in a sense that they can be described by a few rules. Complex theories arise from the combination of several simpler theories. The complexity of a theory does not lie in the size of a theory, e.g. the number of operations used to describe it, but in the interaction of simpler parts.

The ontologist Roberto Casati introduces the term *theorita*, to distinguish small from big theories (Casati, 2000). When I use the term theory in the present thesis I think of small conceptualizations of space: conceptualizations that can be described by a limited set of rules. These theories describe operations with small sized objects that move in space. The theories are a body of rules used to predict spatial properties of objects in an environment. The predictions generated by a theory are also called beliefs in the present thesis.

The *theory theory* proposes that children form theories of the world by building and testing hypothesis. The observation of contradiction between facts and beliefs leads to theory revision (Gopnik and Meltzoff, 1997). Gopnik and Meltzoff (1997) describe theories by structural, functional and dynamic features. The *structural features* describe the theories themselves. Infants seem to hold *abstract theories* that are different from adult theories. Through learning processes these *theories are causally connected* and form new theories. Children hold theories that lead to *ontological commitments* about the world, i.e. accepting a theory leads to expectations grounded in the theory (cf. Kuhn (1962)).

The *functional features* explain what children do when they hold theories. They predict actions or events, e.g infants look at or reach predictably for a moving object. Theories explain why things happen to be as they are. When children observe the same event several times, e.g. a ball falling they lose interest in the event. It seems as they would hold a theory that explains the event.

The change of theories is described by the *dynamic features* of a theory. Change requires to compare predictions with observed actions. Continuous contradiction between observation and prediction will lead to theory revision.

2.2 Scientific Theories vs. Naive Theories

2.2.1 Scientific Theories

Theory theorists hypothesize that the formation of theories in children is analogous to scientific theory revision. The process of theory revision has been a controversial topic in the history of science and discussed by Karl Popper, Thomas

Kuhn and Paul Feyerabend. Among the researchers there are controversies about the influence of society on the scientific project. Progress in knowledge is often presented as one theory building upon the other, this monotony has to be questioned in the course of the thesis.

Kuhn defines normal science as solving puzzles based on paradigms. A paradigm is a common agreement of knowledge shared by a group of scientists. Having a paradigm as a framework, the solutions to problems are known in advance and just have to be worked out. Progress of science requires that paradigms are exchanged (Kuhn, 1962).

The discovery of scientific theories is a multi-staged process. Normal science refines existing theories but will not lead to new theories. *Anomalies* are the recognition that nature does not follow the predictions of the current body of theories and paradigms. Researchers will try to explain the anomaly, by adapting the theories available. For normal science assimilation of new facts into a theory can only be carried out under the constraint that previous facts remain consistent with the theories available (Kuhn, 1962). This protects the paradigm against being given up too quickly.

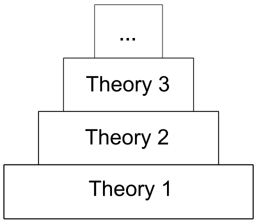


Figure 2.1: Normal science - One theory building upon the other

It is the impossibility to adapt an existing paradigm that leads to a *crisis* in normal science. All efforts fail to correct the existing paradigms. The consequence of a crisis is a *revolution*. A revolution involves the change of paradigms. It is a process that takes place influenced by political, economical and sociological settings of their scientific communities (Kuhn, 1962).

A theory holds until it is disproved (Popper, 1934). A theory such as “Every swan is white” holds until a swan having different colored feathers occurs. Falsified theories have to be rejected and replaced by new theories.

The new theories will allow predictions that were not possible with previous theories and resolve problems that arise out of the anomalies. Theory change influences the way scientists perceive the world. New technology allows to build new sensors, which lead to a new view about the world. One could hypothesize that the emergence of new technical abilities changes process descriptions.

Finding new theories is constrained by the general principle of parsimony. Occam's razor demands to be careful with the available resources and to make theories as simple as possible. The scientist depends on the socio-economic environment and has to be very selective in his experiments to choose among the number of infinite, possible theories.

The monotony of theory revision does not hold as it has been found that the detection of a new theory does not necessarily lead to a rejection of the old theory. The impossibility to proof the fifth axiom out of the given four of the Euclidean geometry lead to the discovery of a plethora of non Euclidean geometries. These coexist to the Euclidean geometry (Blumenthal, 1961). Using this argument Piaget pointed to a coexistence of different competencies in the human organism (Bringuier, 2004).

At a first glance common sense reasoning seems not to be based on the constraints described by the history of science. But empirical research shows analogies between the way children conceptualize the world and the multi-staged process scientists go through (Gopnik and Meltzoff, 1997). The following section discusses how people naively conceptualize the world.

2.2.2 Naive theories

Empirical research in psychology suggests that people build naive theories based on their everyday experience. The behavior and reasoning in experimental spatial situations is consistent across individuals. Therefore one can use the term common sense theory, but should keep in mind that the formalized theory is inferred by an experimenter or scientist.

The formal study of commonsense theories has been started with the intention to build autonomous robot architectures and artificial intelligent systems (Hobbs and Moore, 1985). The formal study of naive physics (Hayes, 1978, 1985) provided commonsense theories for rigid objects and liquids. The commonsense theories about the motion of objects are inconsistent with fundamental principles of classical physics, but show similarities to pre-Newtonian physics (McCloskey, 1983).

The American Institute of Physics compiled a list of children's misconceptions about the world. The list comprises disciplines such as astronomy, space, measurement, force and motion and many more. Examples for objects in motion (see website[1]) are:

- If an object is at rest, no forces are acting on the object.
- A rigid solid object cannot be compressed or stretched.
- Force is a property of an object. An object has force and when it runs out of force it stops moving.

Not only children but also adults hold such naive theories (McCloskey, 1983). Individuals often use more than one commonsense theory to explain a phenomenon. Empirical studies on the coding of object locations give evidence that humans utilize multiple bodies of theories (Newcombe and Huttenlocher, 2003). Other empirical studies on strategy discovery revealed that people use multiple strategies simultaneously (Siegler, 2002; Siegler and Araya, 2005) in various problem domains. Figure 2.2 illustrates the overlapping waves model by Siegler (2002) in which “older, less advanced strategies continue to be used long after newer, more advanced strategies have been discovered”. Multiple strategies are available with age, the x-axis indicates the age in the figure and the y-axis the use of the strategy in percent.

Siegler’s (2002) empirical studies and his model of adaptive strategy choice motivate the conclusion that commonsense theories coexist simultaneously. The use of a theory depends on the context the individual is in. To summarize in the words of Siegler (2002, p. 34): “Knowledge moves consistently from less to more advanced, rather than oscillating aimlessly; knowledge often is reorganized, rather than shifting in superficial ways; and learning is generative, in the sense that early advances form the foundation for later ones.”

2.2.3 Discussion

The reason that naive conceptualizations differ from scientific explanations is that everyday reasoning hardly involves measurements. People estimate how far two objects are from each other, how heavy an object is or how long it takes for an object to travel along a trajectory. People make comparison based on estimates and previous experiences.

[1] http://www.amasci.com/miscon/opphys.html

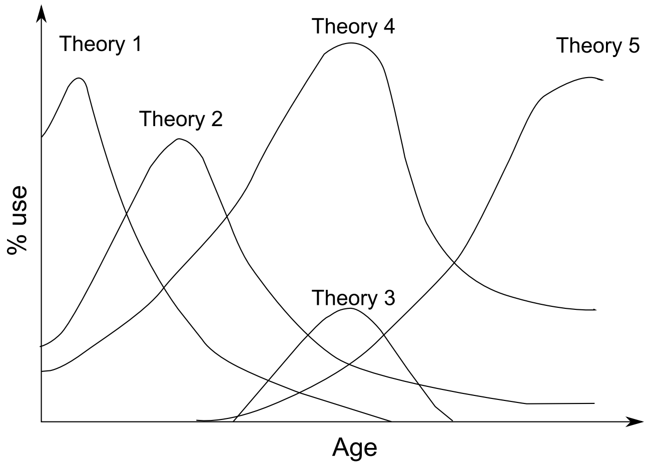


Figure 2.2: Overlapping waves model
(in adaptation to (Siegler, 2002))

People build theories based on the perceptions they make in the environment. The perceptions are different for each individual and that explains why people build different conceptualizations. The situation is described in one of many versions of an ancient parabola where six blind born men were asked to describe an elephant. The blind men conceptualized the elephant as a wall, a tree, a spear, a snake, a fan, and a rope depending on which parts of the elephant they could experience (see figure 2.3).

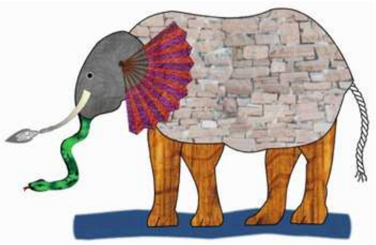


Figure 2.3: Parabola of the elephant
(source: http://www.wordfocus.com/word-act-blindmen.html)

None of them was wrong but also none of them got the whole picture right. In fact if the set of perceptions individuals are exposed to are identical it could be that all individuals would end up in the same conceptualization. Or as theory theorist put it: “If cognitive agents begin with the same initial theory, try to solve the same problems, are presented with similar patterns of evidence over the same period of time, they should converge on the same theories at about the same time" (Gopnik and Meltzoff, 1997, p.26).

Misconceptions in naive theories are a result of the experience an individual made. As long as no contradiction occurs there is no need to change the conceptualization. In a series of studies childrens' and adults' naive conceptualization of a balance scale has been investigated. The subjects in the empirical study were asked if the scale will remain balanced or will tip down. If it tips then they were asked to which side. Cognitive accounts based on conditional rules have been developed. The rule sets never reached the level of explanation that physicist would develop (using the torque rule) (Siegler, 1976; Siegler and Chen, 2002).

That is because in scientific theories measuring in a reproducible way is the tool to objective properties (Feynman, 1998). The process of measuring is influenced by environmental factors, the imprecision of the measuring tool, and the imperfection of the human senses. Based on the measured magnitudes theories are judged as valid.

Naive theories are based on beliefs. These are observations of causal links, and interpretations of the outcome of actions generated out of the naive theories. Because reasoning is based on perceptions, beliefs can be a result of perceptual illusions. Perceptual illusions lead to acceptance of false theories.

2.3 Epistemological Considerations

Epistemology is concerned with the source of knowledge. The revision of theories is very much dependent on the prerequisites. There are two extreme standpoints:

Nativism assumes that all theories are innately given to the organism while empiricists see the environment as the main source of new theories. Nowadays there is common agreement among cognitive researchers that both standpoints are plausible. Interactionist accounts of cognitive development and information processing assume knowledge to be partly coded from birth, to be learned and to be mediated through the environment (Gopnik and Meltzoff, 1997; Newcombe and Huttenlocher, 2003).

2.3.1 Nativism - The Role of Innate Knowledge

Theories may be initially given. New theories are formed by revising given theories. The revision of the theories is triggered by the environment.

Nativists suggest the existence of *core knowledge* with infants that does not underlie radical changes (Spelke, 1990; Spelke et al., 1992; Spelke, 2000). The core knowledge comprehends a set of innate theories. The infant can build *active representations,* i.e. inferences on the core knowledge in order to derive new knowledge (Spelke et al., 1992). The model of a modular brain has been proposed (Fodor, 1987), that encapsulates mental processes, e.g. language (Pinker, 1995) or mathematics into independent modules.

Studies in the realm of spatial cognition revealed that infants may hold object representations that preserve identity and persist over occlusion and time (Spelke, 1990, 2000; Baillargeon, 2004b). Infants seem to possess an early notion of distance (Gopnik et al., 1999, p. 82), they can make predictions whether reaches will make contact to a moving object (Hofsten et al., 1998), and they show reactions to object appearances (Piaget, 1950; Bower, 1974). Gopnik and Meltzoff (1997) suggest that three innate theories are relevant for cognitive development:

1. A theory of appearances that explains object permanence, i.e. objects endure through space and time and do not magically disappear.

2. A theory of actions that explains the difference between actions of the self and others.

3. A theory of object kinds that helps to build categories such as a distinction of living and dead objects.

2.3.2 Empiricism - The Role of the Environment

Theories may be determined by the environment with little initial knowledge. In the empiristic view the focus of cognitive development is on the social setting of the infant. The surrounding conditions consist of parents, relatives and friends but also culture and environment in a wider sense guide the cognitive development (Pine, 1999; Gopnik et al., 1999).

How much children depend on their social environment can be seen with infants attraction to faces. Another example is the well developed early imitation mechanism. It helps the child to distinguish the self from others (Rochat and Hespos, 1996; Meltzoff, 2004). The absence of others can heavily influence social behavior and cognitive development, as demonstrated by Kaspar Hauser 1828 in Nürnberg, Germany.

When adults repeat words that toddlers said, they unconsciously help them at language acquisition. Scaffolding is a follow up theory of Vygotskys' sociocultural empiricism. Parents and teachers scaffold temporarily children's knowledge in order to help them to act independently by giving hints in the right moment of problem solving tasks, asking questions or showing procedures that children can then imitate (Rogoff (1990) as cited in Pine (1999)), e.g. Granott et al. (2002) found evidence that scaffolding appears in problem solving tasks testing peer groups of students.

2.4 Formal Models of Cognitive Development

The theories of nativism and empiricism inspires two types of computational models. The focus of the models is either on reasoning processes over explicitly stored knowledge or on the processing and acquisition of knowledge through the environment. Researchers in language acquisition and semantics have been distinguishing symbolic and grounded models.

Symbolic models are based on a universal, conceptual system. They stress the importance of innate given knowledge. The acquisition of linguistic meaning is a mapping process of new emerging symbols to an available universal system. The universal system is innate or learned pre-linguistically. There is a strict distinction between the lingual and the non-lingual system that develop independently. The role of perception is often neglected in this view (Pinker, 1995; Gasser and Colunga, 1997).

Grounded models stress the importance of perception in the acquisition of linguistic meaning. Gasser and Colunga (1997) differentiate two subtypes of grounded models: models that make a distinction between linguistic meaning, and non-linguistic concepts e.g. Regier (1996), and those that do not e.g. Gasser and Colunga (1997). The later concepts may be learned in three ways: through non-linguistic perceptual and motor experience, through a combination of non-linguistic and linguistic experience, and through linguistic input alone (Gasser and Colunga, 1997).

Mareschal (2003) reviews three groups of computational models for cognitive development. *Symbolic models* represent knowledge in terms of symbols and use grammar rules and syntax to connect symbols to new expressions. Subsymbolic or *connectionist models* encode knowledge in networks analogously to human brain cells. *Dynamic systems* are mostly mathematical models based on differential equations, finite state machines or cellular automata.

A formal model serves to build a sound cognitive theory because all terms have to be defined in order to yield an executable model. The present thesis proposes a formal model of spatial data acquisition motivated by the *theory theory* . Previous formal models that could be used to model spatial theories and their change are reviewed in the subsequent three sections.

2.4.1 Symbolic Models

The work of Young (1976) shows that a theory of sorting can be expressed by a set of rules. The formal model simulates three stages of Piaget's seriation task. In order to change from one stage to another rules are added or deleted from a production system [2]. Other Piagetian tasks such as the identity theory of object concept development in infancy (Piaget, 1950) were modeled using production systems. The model is based on five search behavior rules and three conceptual rules (Luger et al., 1983, 1984).

These early models illustrate that a naive theory can be described by a set of rules in a production system. Theory change can be modeled by adding and deleting rules. It is not required to change the whole theory, i.e. the set of rules.

The models do not automatically proceed from one stage to the other and they also do not reuse old rules. Siegler and Shipley (1995) implemented strategy choice based on a probabilistic cost-efficiency account. The rules in the model are randomly perturbed and tested. Those rules that get more evidence will survive the others will die off. An important aspect of the model is that rules are maintained over a longer period of time and do not immediately disappear of the knowledge base. Old or unused rules are kept in a pool. This allows a later reactivation.

Until lately effects such as memorizing inputs could not be explained by symbolic approaches. The models response to a specific input was always the same no matter if the stimulus was given immediately or with a delay. However a delay of the stimulus has an influence on the outcome in empirical studies of

[2] A production system consists of a set of rules or *productions* that describe which actions have to be taken in order to solve a given problem. Each production has a condition and an action part, such as an if ... then ... else clause. The action part of the rules alter the *working memory* of the production system that describes the current state of the world via patterns. In a *recognize-act control cycle* a given problem description is maintained as patterns in the working memory and matched against the production rules. If several productions fit to a given problem a conflict resolution has to take place (Newell and Simon, 1972).

cognitive development (Thelen et al., 2001). Later models also consider effects of memorizing, consciousness and attention by introducing weighting mechanisms and stochastic such as models by Siegler and collaborators (Shrager and Siegler, 1998; Siegler and Araya, 2005).

The cognitive science community provided a number of architectures. Two representatives that are actively improved are the symbolic cognitive architecture SOAR (States, Operators and Reasoning) (Laird et al., 1987) and the Adaptive Control of Thought, Rational architecture ACT-R (Anderson, 1993). A plethora of variations of these frameworks is available, including models for spatial reasoning and navigation (see the ACT-R website [3]).

2.4.2 Connectionist Models

Connectionist models are inspired by neural activities, they are also known as parallel distributed processing systems. The key idea of neural nets is that knowledge is processed parallel by simply interconnected processors rather than by a single processing unit. Therefore a neural net consists of cells also called units and weighted links that connect the units. The weights describe connection strength. The links between the units are analogous to axons and dendrites in the human brain. Some researchers see therefore a biological grounding in connectionist models.

Neural nets are an attempt to overcome the shortcomings of symbolic models that do not explain *how* knowledge develops. They focus on cognitive development as a process that is just controlled by perception, e.g. data that a robot receives from sensors. The use of explicit symbolic mental representations is omitted (Hiraki et al., 1998; Mareschal, 2003; Schlesinger and Parisi, 2001; Parisi and Schlesinger, 2002; Munkata and McClelland, 2003).

The simplest example of a neural net is a feedforward network. It consists of an input and an output layer. More advanced versions of feedforward networks have several input layers and can also have hidden layers. Feedforward networks are directed and work just in one way (see Figure 2.4).

Recurrent networks allow the definition of paths back to a unit through itself or other units. The networks need not be coded by hand. A backpropagation algorithm can be used together with a set of training data to build the network. The supervised learning algorithm propagates errors from output nodes backward

[3] http://act-r.psy.cmu.edu/publications/index.php

to input nodes by comparing the actual output of the network with the expected output. A reweighting of the links is carried out to adjust the performance of the network. The network is trained until the measurable difference of the actual output and the expected output falls under a threshold.

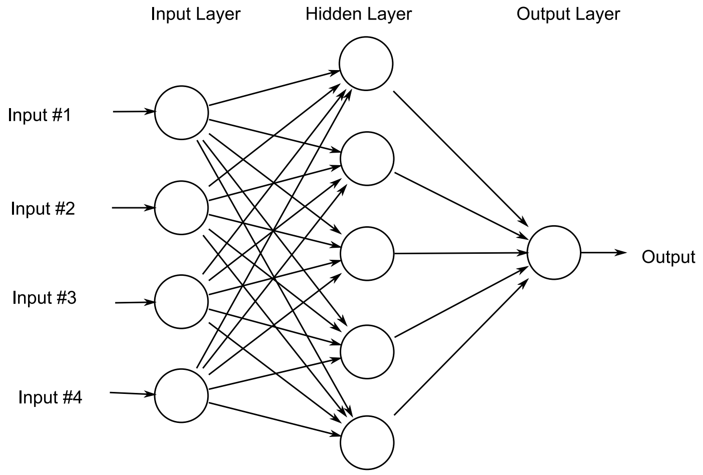


Figure 2.4: Feedforward Network
(source: USGS, http://smig.usgs.gov/SMIG/features_0902/tualatin_ann.fig3.gif)

A number of connectionist models have been proposed to model children's spatial cognition. The development of spatial concepts in linguistics has been modeled (Regier, 1996; Gasser and Colunga, 1997). The neural net mechanism of Terry Regier is able to learn spatial relations in a number of languages without negative evidence. In order to overcome the missing negative evidence he introduces constraints into the neural network, i.e. constrained connectionism (Regier, 1996). The constraints represent knowledge at a price of losing flexibility. The structures have to be hard wired and can not be gained through training or automatic adaptation (Regier, 1996).

Other connectionist networks have been implemented that can track occluded objects in accordance to studies carried out in developmental psychology. The models can predict the position of a moving object after an adequate training phase. Variations of the experimental setup have been tested and implemented in the neural net in order to explain contextual influences (Mareschal, 2000; Schlesinger and Parisi, 2001; Schlesinger and Young, 2003).

Various robot architectures have been implemented using neural nets (Scassellati, 2000). The goal is to build robots that communicate with humans (Hideki and Hiroyuki, 2001) and can build conceptual models of their environments in order to act on objects (Fitzpatrick et al., 2003). Hiraki et al. (1998) focused on spatial cognition and provide a connectionist robot implementation that models the shift of egocentric to allocentric location coding in an object retrieval task.

2.4.3 Dynamic System Models

Dynamic systems can be finite state machines, a set of differential equations, cellular automata or Turing machines (Beer, 2000). Dynamic system theory has been adequate to describe the interaction of multiple cognitive competences, such as perceiving, remembering and acting. Two examples for robot architectures that consider the simultaneous and competitive interaction of spatial competences are the subsumption architecture (Brooks, 1986) and the spatial semantic hierarchy (Kuipers, 1998, 2000). Brooks suggested a subsumption architecture in order to deal with the different interacting levels of competence. Higher levels subsume the roles of lower levels. The lower levels continue to function when new competence is added (Brooks, 1986). The spatial semantic hierarchy is a model of the human cognitive map and a method for robot exploration and map building. The model has five layers that can deal with sensory, control, causal, topological and metrical information (Kuipers, 1998, 2000).

In a dynamic model of Piaget's A-not-B error[4] Thelen et al. (2001) show how goal directed actions such as looking, planning, reaching and remembering can be united in one framework of processes using differential equations.

The equations describe fields to implement the infant's motor encoding. The activation and decay of the different fields are summed up to a single representation field that stands for the motor action of the infant. The dynamic field model of Thelen and collaborators is an abstract model for the dynamics of multiple processes in the brain and body (Thelen et al., 2001).

Figure 2.5 shows the subsequent steps of the A-not-B task from the presentation of the stimulus to the reaching for a hidden object. The star illustrates the object hidden under one of two cups. On the left side of the figure the modeled actions are listed. The infant's planning to reach for the object is illustrated through wave diagrams that have been described by differential equations.

[4] 7-12 months old infants that continuously reach for an object that was hidden at position A continue to reach for that object at position A even if they saw the object being hidden at another location B. This phenomenon called the A-not-B error puzzles psychologists for a long time and is the subject of intensive research.

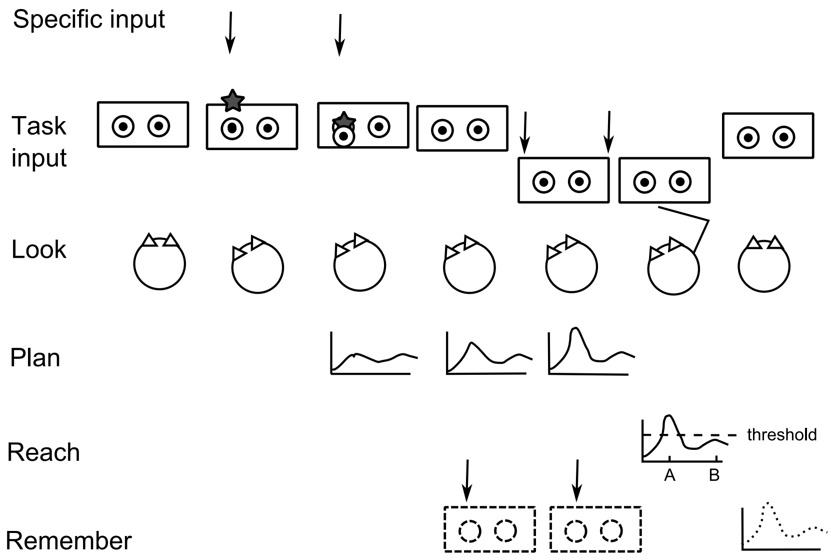


Figure 2.5: Dynamic model of the A-not-B error
(source: Thelen et al. (2001))

Each diagram is a snapshot of the single representation field at a certain point of time. Whenever the amplitude of the wave exceeds a certain threshold the model "reaches" towards a location (see the diagram in the line of the reach action in figure 2.5) indicated by the wavelength and wavenumber. Memorizing has been modeled as an additional term in the representation field that activates or damps the wave depending on previous trials (Thelen et al., 2001).

Dynamic systems models integrate different processes into one framework. Therefore they are an interesting tool for geoinformation scientists concerning the problem of interoperability. However it would require a general approach to describe processes in the various domains of geoinformation using dynamic systems. A first step is the classification of possible differential equations for geographic information (Hofer, 2007).

2.5 Summary

In the present thesis a *theory* is a body of rules to predict spatial properties of objects in an environment. This body of rules is built through observing the world. Theories are revised when the generated predictions (beliefs) do not fit with the observations made in an environment. The thesis describes the change of theories formally, as sequences of theories.

Observation and prediction serve to shape a theory. Theories that explain the world better transform out of their ancestor theories (see figure 2.6). They are preferred to theories that do not explain the world fully.

Scientific and naive theories show commonalities. Firstly the process of naive knowledge discovery is constrained by the human body and the perception of a common shared reality such as scientific knowledge discovery is constrained by socio-economic settings. Secondly the revision of theories advances from simple to complex theories. Thirdly scientific and naive theories coexist simultaneously rather than building one upon the other.

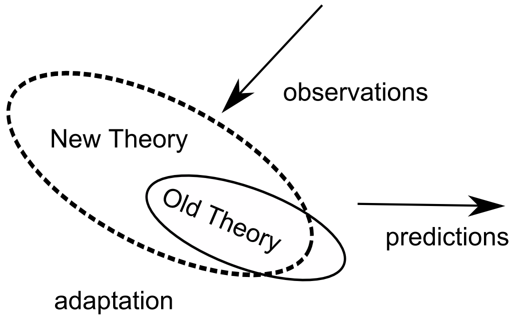


Figure 2.6: The formation of theories

The *theory theory* proposes that children and scientists form theories of the world by building and testing hypotheses. I reviewed computational models of cognitive development as means to formally describe the processes of theory acquisition and theory change. In the following chapter I propose that formal theories of space and their change can be modeled by using an algebraic approach. In terms of the *theory theory* spatial reasoning is based on a set of small theories, that have been developed during childhood and endure with adults in a revised form.

Chapter 3

A Calculus of Spatial Theory Change

I would like to go one step further with the idea of the *theory theory* and argue that human naive understanding of space is based on small theories. These spatial theories underlie qualitative change and build sequences. In the thesis a formal description of spatial theories and their change will be provided as a basis to build geographic information systems grounded in people's commonsense understanding of space.

I am going to concentrate on the change of spatial theories in the light of new evidence. The aim is a formal description of ever qualitatively changing concepts. I assume that spatial concepts are first formed by observing other people acting and operating with objects. At a later stage the own actions on the objects are evaluated to build spatial concepts. In order to build categories such as containers or supporters, means to build an understanding of how containers behave when they move, when they are lifted, and when they are turned around.

When babies start to explore actively their surroundings they are two months old. At this age the infants have developed a sense of self-awareness. The infants then gradually explores the environment by observation - being a simple *spectator.* Once the motor capabilities grow infants get into the role of an *actor* in their environment (Rochat, 2004).

In this chapter a theory driven agent based on algebraic specifications is introduced. The agent is a wrapper for a mechanism that builds sequences of spatial theories based on observations in an environment. The model is an analogy to the developing child that can change its mind based on observations of the environment. The model is deterministic and grounds in previous work in cognitive science. Design decisions towards a calculus of spatial theory change are outlined.

3.1 Spatial Theories

Space has a certain primacy in our lives, humans can not escape the situatedness in a spatial environment. People are in space and their everyday commonsense understanding of space helps them to find ways through the environment. In order to localize objects in an environment, operations with objects are observed and causal effects based on spatial relationships predicted.

Naive geography suggests that spatial, cognitive processes can be described by a set of theories (Egenhofer and Mark, 1995). Studies specifically investigated naive understanding of space and identified a number of "spatial misconceptions" (McCloskey, 1983; Nelson et al., 1992). Egenhofer and Mark (1995) partially listed a set of theories that people have about the surrounding geographic world:

- The earth is flat.
- Maps are more real than experience.
- Boundaries are sometimes entities, sometimes not.
- Topology matters, metric refines.
- Distances are asymmetric.

These commonsense beliefs have to be considered in the user interfaces of geographic information systems. Then geographic information systems will be widely accepted among laymen (Frank, 1993). In order to achieve this there is a need for formal descriptions of naive theories about space. Recent formal descriptions of naive spatial theories have been based on image schemata.

According to Johnson (1987) the concepts of the world are structured in image schemata. Image schemata mentally organize our understanding and reasoning of the world (Johnson, 1987). They are embodied descriptions of the real world such as containers, surfaces and links (see table 2.1).

Image schemata lead to object based models for the use of geographic information systems (Rodriguez and Egenhofer, 1997; Frank and Raubal, 1998; Frank, 1998; Frank and Raubal, 1999; Rodriguez and Egenhofer, 2000; Ruetschi

Container	Balance	Full-Empty	Iteration	Compulsion
Blockage	Counterforce	Process	Surface	Restraint Removal
Enablement	Attraction	Matching	Part-Whole	Mass-Count
Path	Link	Collection	Contact	Center-Periphery
Cycle	Splitting	Merging	Object	Scale

Table 3.1: Partial list of image schemata as defined by Johnson (1987)

and Timpf, 2005). Among the models two research goals can be observed. One direction of research concentrates on improvement of user interfaces, e.g. (Kuhn and Frank, 1991a). Another direction of research carries out the formalization of image schemata with the aim to improve interoperability, e.g. (Frank and Raubal, 1999).

Formalizations have been suggested using predicate calculus, relations, functions and model-based approaches. Predicate calculus is limited by the so called frame problem (McCarthy and Hayes, 1969). Relation tables grow with the square of the number of relations involved. Similar growth rates in complexity apply for function tables. Due to these constraints in complexity, model based approaches seem to be the most promising candidates for a formalization of image schemata (Frank, 1998).

In order to formalize image schemata it is necessary to concentrate on a very specific example, e.g. the axiomatic approach by Rodriguez and Egenhofer (2000). The frame problem can be overcome by introducing a scene and describing changes as subsequent operations on the scene. Spatial semantics is described by the operations used with the model, e.g. the operation *put_in* implies containment (Egenhofer and Rodriguez, 1999; Rodriguez and Egenhofer, 2000). Note that the models have been made up in the mind of the researchers.

Another direction of research formalized image schemata using linguistic approaches (Frank, 1998; Frank and Raubal, 1998). There the question remains to which degree language influences our spatial concepts. Concepts of space do not necessarily have to be reflected in the language we use. Different mechanisms of cognition and perception may work on the concepts before they are externalized through a lingual system.

The present research shows parallelisms to previous work and concentrates on an object based model for spatial theories in a small scale space environment. Mechanisms will be introduced to build sequences of spatial theories and their change. An agent builds new theories based on observations of operations with objects in an environment. In comparison with previous work the proposed model is novel in two aspects:

1. Empirical studies on infants knowledge of the physical object world (Hespos and Baillargeon, 2001a,b; Luo and Baillargeon, 2005; Baillargeon, 2004a,b) are used to build the model. This makes it different from previous work as the modeled environment is not made up in the mind of the researcher. The model can be validated by carrying out a simulation and comparing the simulation with the outcome of the empircial studies.

2. Several developmental psychologists observed that spatial relations are learned in the first two years of life, a phase that is usually before the capability of speaking. The empiricial studies formalized investigate pre-linguistic concepts overcoming the problems with linguistic approaches. The formal theories are a possible explanation of the acquisition of image schemata (Johnson, 1987).

The following section sketches how an agent can be endowed with a set of theories. The model is motivated by infants that learn to know their environment. The following sections serve to explain the elements for a calculus of spatial theory change.

3.2 Theory driven Agent

Agents are an approach to deal with the complexity of building a model for conceptual theory change. The agent based approach allows to reduce reasoning processes to abstract parts and studies the interaction between these parts. Subsequent refinement of the model leads to a better understanding of the underlying cognitive processes.

An agent can stand for a technical concept, a metaphor or a design model (Nwana and Ndumu, 1999). Russell and Norvig (1995) define an agent as “anything that can be viewed as perceiving its environment and is acting upon it through effectors”. This definition has been adopted by researchers in geoinformation science (Raubal, 2001; Krek, 2002) and captures the crucial aspects of agency.

In the course of the thesis the term agent stands for a very generalized concept, for which a generic type of model exists. For a discussion of agent architectures see e.g. (Woolridge, 1999), aspects of embodiment are discussed in Ziemke (2003) and the elements of a learning agent can be found in Russell and Norvig (1995). Agents as an analogy to the developing child are discussed in Schlesinger and Parisi (2001) and Schlesinger (2002). The agent here is a wrapper to cognitive processes that define the change of spatial theories. The agent is situated in an environment.

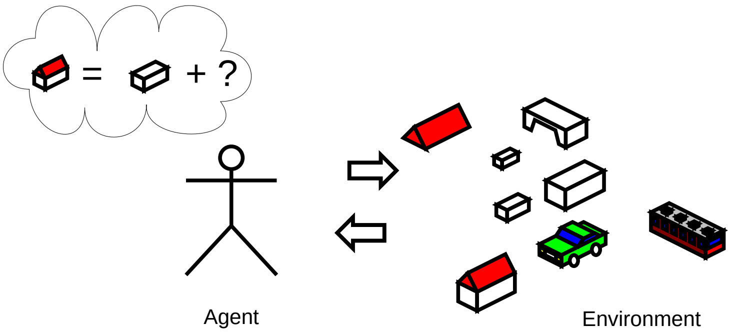


Figure 3.1: Agent acting in an environment

The term theory driven agent specifies the agent as holding theories about its environment. The theories are explicitly given to the agent by the modeler and not built automatically through an inference mechanism. The agent observes operations in the environment and can use the theories to build predictions about the observed operations (see figure 3.1). Frequent mismatches between observation and prediction elicit changes in the knowledge base of the agent. The agent chooses among the available theories the theory that fits its observations best.

3.3 Affordances in the Environment

Affordances in the environment explain why agents are selective among the infinite amount of possible operations that can be carried out over an arbitrary object. There is a strong coupling between the properties and the actions of an object. Objects afford what the agent can do with them.

Gibson's theory of affordances (Gibson, 1979) has been utilized to design agents that make sense of their environment (Raubal, 2001; Viezzer and Nieuwenhuis, 2005). An affordance guides the agents actions it comprehends the object and the subject, i.e. the agent and the environment. This is called an ecological approach.

Objects in the environment afford actions. The affordances depend on the properties of the object and lead to actions. A door handle will afford to push or pull the door.

Affordances have a *functional aspect* because they group objects by their potential use. Objects are described by the operations that can be carried out over them. A stone and a hammer are in the same object category when used to drive a nail in a piece of wood.

Affordances have also a *discriminative aspect.* Things that have the same use may have similar features or object attributes. The hammer and the stone are both rigid and not eatable.

The agent repetitively carries out the afforded actions with the objects in order to learn about their usage and to categorize them. This is analogous to the infant's play. Affordances also depend on the experience the agent has already made. In an early phase of infancy children would investigate any object with their mouth when they are hindered to move their arms (Rochat, 2004). The world is separated into eatable and non eatable, graspable and not graspable, etc. A long period of infancy serves to test and to reason about objects and their affordances.

3.4 Testing Theories

3.4.1 The rational infant

In the book "The Rational Infant", Bower (1989) argues that babies formulate hypothesis and test them. The main part of this subsection is reviewed from Bower's book. The term hypothesis can be used here interchangeably with the definition of theory given in chapter two of the thesis. Hypothesis are used to predict the outcome of an operation. Observation of a different outcome than predicted leads to theory change.

Based on empirical studies Bower states that six to eight week old babies start to verify hypotheses in a Popperian manner. Babies try to prove by disproof. In the empirical study limb movements l of a baby elicit movements of a mobile m (see figure 3.2). This was done based on a contingent reinforcement schedule, whenever the leg moved the mobile movement was elicited.

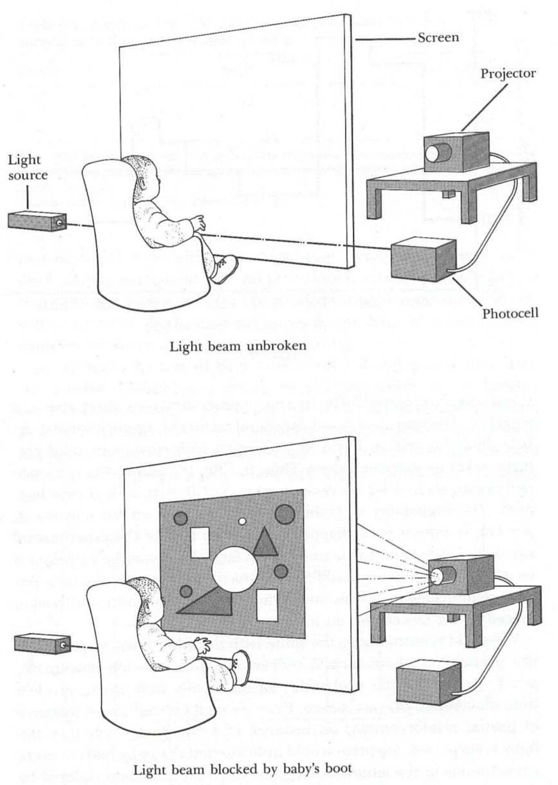


Figure 3.2: Mobile Experiment with contingent reinforcement
(source: Bower (1989))

> "Suppose our baby is lying there making random limb movements. Our baby then notices that the mobile occasionally is turning. He begins to suspect that there is some relationship between limb-movement and mobile movement. He begins [...] to formulate a hypothesis about a possible relation between l and m." (Bower, 1989)

The empirical data gave evidence that the babies' behavior was such as obtaining information for testing the following two inequalities. p stands for the probability, l for limb movements m for mobile movements:

$$p(l \wedge m) > p(\neg l \wedge m)\, and\, p(\neg l \wedge \neg m) > p(l \wedge \neg m)$$

In order to detect the contingency between l and m the baby has to test the positive and the negative instances. Babies should move their limbs in the same extent as not moving. A phase of extensively moving the limbs ($l \wedge m$) followed by a phase of extensively “not moving” the limbs was observed ($\neg l \wedge \neg m$). At this point the babies could hold a theory that $l \rightarrow m$ or $\neg l \rightarrow \neg m$. Here Bower (1989) introduced non contingent reinforcement, i.e. a mobile movement was elicited without a prior leg movements ($\neg l \wedge m$). The baby could reason now that $l \rightarrow m$ is true and $\neg l \rightarrow \neg m$ is false. The babies could be satisfied with the given information but instead of leaving the experiment the babies started again to actively move their limbs such as to disproof the acquired theory of $l \rightarrow m$. Looking at the truth table 3.2 one can see that occurrences of ($l \wedge \neg m$) would falsify the acquired theory.

l	m	$l \rightarrow m$
t	t	t
f	t	t
t	f	f
f	f	t

Table 3.2: Truth table for the mobile experiment by Bower (1989)

The kind of reasoning explained in the paragraph above is perfectly logical. Bower (1989) refers to the adults rational logical system. This logical system is based on two truth values and can be characterized by three axioms:

- the law of identity ($p \rightarrow p$),
- the law of the excluded middle ($p \vee \neg p$) and
- the law of non-contradiction ($\neg(p \wedge \neg p)$).

Piaget proposed that infants hold another system of logic than adults (Piaget 1982 in Bower 1989). By omitting the second and third axiom a new system of logic based on 4 truth values (see figure 3.3) can be won (Belnap, 1977). The truth values are true, false, true or false and true and false. This kind of logic would help the infant to exclude invalid hypotheses from reasoning.

A hypothesis may take any of the four values. The untested hypothesis is at the same time both *true or false*. Observations will lead to hypotheses that are verified as *true* or falsified as *false*. Hypotheses that are *true* and hypothesis that are *false* will be adopted in the reasoning processes of the infant. But observations will also lead to hypotheses that are both *true and false*. A hypothesis that is both *true and false* at the same time can not be used for reasoning and will be isolated of the infants’ reasoning processes (Bower, 1989).

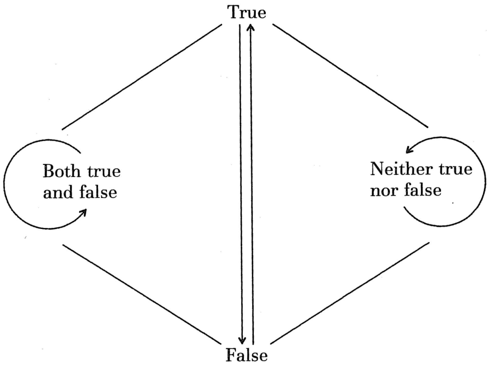


Figure 3.3: 4-valued logic

3.4.2 Theory Change

Theories that are both true and false at the same time need to be adapted. Adaptation is defined as making the theories fit to observations of the outcome of operations in an environment. It is not meant as replacing the whole theory but exchanging part of it following the hypothesis that spatial theories can be described by a set of axioms and that a spatial theory can be adapted by adding, deleting and weighting axioms. In the machine learning literature the process of theory acquisition and change is described as a three step procedure (DeJong (1997) in Baillargeon (2004)).

1. Noticing a contrasting outcome of a theory.
2. Search for the conditions that map onto these outcomes.
3. Build a theory based on the condition-outcome pair using prior knowledge

In adaptation to a model for strategy choice based on rule sets (Siegler and Chen, 2002) the discovery of a new theory can be described as a four step procedure. It depends on the agent's environment, the learning capabilities of the agent and on the prior knowledge:

1. Noticing a new percept. I assume a closed set of available percepts given through an environment, constrained by affordances. Percepts pop up. - The model integrates them into theories.
2. Formulating a theory including the new percept. The formulation of a theory is based on previous theories and percepts. A mechanism could be

based on logical or probabilistic inference. In the course of the thesis I decided not to implement an inference mechanism as there is a number available in the literature and rather focus on theory change mechanisms with algebraic tools.

3. Generalizing the new theory to novel problems by using it consistently after it was formulated. In order to evaluate the appropriateness of a formulated theory it has to explain not only the given data but also future observations. The theory has to be tested. To test a theory, positive and negative evidence must be collected (Bower, 1989). Psychologists observed in early word studies across languages that children show verbalization of success (“There”, “Done it”, “Good”) and failure (“Oh dear”, “no”) (Gopnik and Meltzoff, 1997). Based on the positive and negative evidence a theory can be classified as untested (true or false), true, false or to be adapted (true and false see also figure 3.3).

4. Maintaining the theory, although no further feedback is given. The maintenance of theories without feedback is of great importance. Piaget argues that the formulation of a new theory does not necessarily mean that an old theory has to be abandoned. The old theory coexists to the newly generated theory as special cases of a more general theory (cf. discussion of Euclidean geometry in section 2.2.1) (Bringuier, 2004).

The present approach is a commitment to moderate nativism and symbolic models. Meltzoff terms it “kick start nativism” (Meltzoff, 2004) and Karmiloff Smith proposes the term representational redescription (Karmiloff-Smith, 1992) meaning a set of innate given theories that is triggered by observations of objects, people and operations in an environment. Besides building theories through direct perception of the environment new theories are also built by the combination of other previously acquired theories.

3.4.3 Dynamic Weighting

Feedback is an important mechanism in order to test a theory. If observations support the new theory it will be maintained otherwise it has to be modified or given up. Any learning mechanism for conceptual change needs therefore to introduce a dynamic weighting mechanism.

Spatial reasoning is often supported by direct feedback e.g. an object can be found in a certain location or not. In way finding the destination can be found or not. Positive evidence, i.e. finding objects in locations reinforces the strategies we use in spatial reasoning in order to survive. Negative evidence leads to a decay of strategies.

Newcombe and Huttenlocher (2003) gave empirical evidence for the coexistence of cognitive spatial mechanisms with infants. They describe the coding of object locations by a framework of four competing competences: sensorimotor learning, dead reckoning, cue learning and place learning. Depending how much experience the human has one mechanism is favored to the other, e.g. a movement that is carried out very often such as the daily way to work is coded sensorimotorically rather than using dead reckoning. The strategy of sensorimotor learning has a higher weight than dead reckoning (Newcombe and Huttenlocher, 2003). The authors suggest that any theory of learning needs to implement a dynamic weighting mechanism.

Regier (1996) investigates the acquisition of spatial semantics in children using a dynamic weighting mechanism in a structured neural net. The weighting mechanism treats positive evidence superior to negative evidence. This is done in order to overcome the no-negative evidence problem and to avoid overgeneralization in the learning process of the connectionist model.

Dynamic weighting is vital for the design of a theory driven agent. The mental model of the agent can consider different type of theories. Theories that receive frequent positive feedback will be treated superior to theories that receive no or negative feedback. The dynamic weighting of the theories helps to grade theories based on their conformance with observations in the environment.

3.5 Mechanisms to Structure Theories

In order to treat theories formally it is necessary to describe them in a data structure. Operators are necessary to navigate through this structure. Mechanisms are necessary to create, delete and adapt the theories. Two approaches motivated the formal description of theories in this thesis:

3.5.1 Blending

Cognitive linguists suggest blending as a cognitive mapping between mental spaces. The blend is a space with its own emergent structure evolving from two input spaces (Fauconnier, 1997). Some of the blend space's structure is inherited of the input space's structure. In order to create a blend several conditions must hold: There must be a partial mapping of counterparts between the input spaces (*cross-space mapping*). A *generic-space* maps onto each of the inputs, representing a common structure shared by the input spaces. The *blend-space* is defined as a partial projection of the input spaces onto the blend. Here the blend inherits structure of the input spaces. The three mechanisms of composition, completion and elaboration constitute the emerging structure of the blend (Fauconnier, 1997).

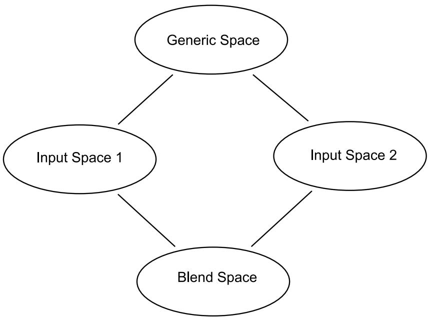


Figure 3.4: Conceptual Model Blending

Blending is an analogy to theory change as defined in the present thesis. Different theories serve as input spaces that can be firstly mapped to more generic theories and secondly blended to more specific theories. In the course of the thesis it will be shown formally how predictions and observations of operations in the environment will lead to more specific theories.

3.5.2 Lattice of Theories

Unfortunately the model of cognitive mapping (Fauconnier, 1997) is not described formally enough in order to be implemented in a computer. A more formal proposal has been provided based on logical theories (Sowa, 1999). The lattice of theories is a generalization hierarchy, where each theory is a generalization of the ones below and a specialization of the ones above it. The topmost theory is a tautology, i.e. all logically true propositions that can be proved from the empty set. New theories are derived from the ones above by inheriting old and adding new axioms. Lower theories are larger in terms of axioms but smaller and thus more specialized in the number of instances they describe (Sowa, 1999).

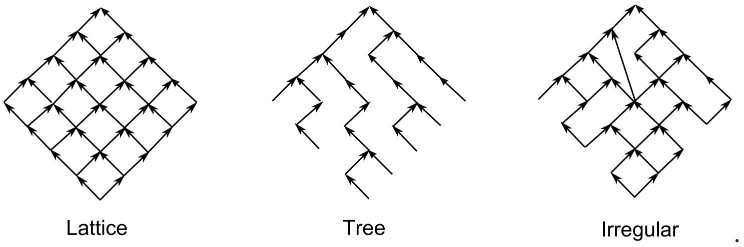


Figure 3.5: Organization of Theories by contraction
(source: (Sowa, 1999))

For example a theory that describes a moveable object inside a container is constrained by two axioms: one axiom for the movability of the object and one axiom for the containment of the object. A generalized theory describes moving objects based on the axiom for movability. Note that the "moveable inside container" theory has one axiom more than the "movable object" theory. An infant observes generally more moving objects than moving objects, inside a container. The "moveable inside container" theory has therefore less instances in the world and receives less evidence. Aristotle termed this property of theories and instances the inverse relationship between intension and extension.

Alchourrón et al. (1985) suggest three operators of contraction, expansion and revision to navigate within a lattice of theories:

Any theory can be *contracted* or reduced to a smaller, simpler theory by deleting one or more axioms. Each contraction step is an upward movement in the lattice of theories. Multiple contraction steps lead to the empty or universal theory. Note that contraction blocks proofs that depend on the deleted axioms.

Any theory can be *expanded* to a bigger theory by adding one or more axioms to it. Each expansion step is an upward movement in the lattice of theories. Multiple expansion steps lead to inconsistent or absurd theories, i.e. theories containing all axioms.

Subsequent contraction and expansion steps lead to theory revision. Figure 3.6 illustrates theory revision. Let a theory with an axiom predict that all objects

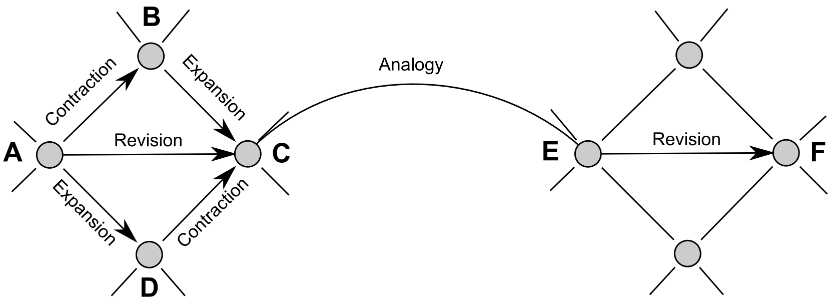


Figure 3.6: Theory revision with the AGM operators
(source: (Sowa, 1999))

that fall down taste like strawberry, and another theory with an axiom predict that all unsupported objects fall down. By adding the axiom of unsupported objects and deleting the axiom of strawberry flavor theory revision takes place. The new theory is based on the support of objects, while the old theory was based on the taste of objects.

In extension to Alchourrón et al. (1985), Sowa (1999) defines analogy as a fourth possibility to theory revision. Analogy requires the detection of structural similarities between theories. Types, relations and individuals that appear in the axioms have to be renamed from the source to the target domain. Analogies lead to new theories by jumps in the lattice of theories (see figure 3.6).

The lattice of theories is a formal approach to structure theories. Algebraic theories can be described with the adaptation of axioms in the lattice of theories. The operators of contraction, expansion, revision, and analogy can be used to model theory change.

3.6 Towards an Algebraic Model

There are controversial discussions what modeling technique to use. Previous research suggests to build either a symbolic or a connectionist model and to choose between a deterministic or stochastic modelling technique. The present research suggests a model that lies somewhere in between.

The theory driven agent is based on algebraic specifications. Three reasons gave the way to this modeling strategy. First algebra is a mathematical sound framework. Second together with a type safe functional programming language algebraic specifications allows rapid prototyping, and third algebras offer mechanisms for abstraction.

In a series of articles in the science magazine infants learning of speech with algebraic rules has been discussed (Marcus et al., 1999; Shastri, 1999; Seidenberg and Elman, 1999). Seven month old infants were habituated to sequences of syllables having solely the pattern ABA or the pattern ABB. In a test phase infants were confronted with new sequences, having both patterns ABA and ABB. It was found that infants showed a preference for the unhabituated, unknown pattern, suggesting that the infants are capable of using an algebraic rule such as "the first item X is the same as the third item Y" in the given task (Marcus et al., 1999; Marcus, 2001).

Shastri (1999) showed that a connectionist network architecture can acquire algebraic rules given in an appropriate presentation. The problem of learning algebraic rules could be reduced to finding spatiotemporal patterns in the nodes of the connectionist network. The proposed model could learn from a small number of examples, generalizing to new data without being given negative evidence.

Today the view is intertwined, cognition works like a neural net but also like a symbol processor that abstracts at a higher level (e.g. Kuipers (2000)). When modeling neural networks researchers give up understanding the way knowledge is encoded. The network's behavior during simulation is observed and conclusions about the operation are made (Regier, 1996). Knowledge about a certain fact or phenomena is a state of the network at a certain timepoint.

The present research suggests a model that is well determined at any given timepoint. Algebraic specifications serve to describe formally spatial theories. Objects that move, objects that contain each other, objects that can rest on each other are all described by different algebras. Each algebra groups common objects that behave similarly under certain operations, e.g. all objects that move freely vs. those objects that move under a constraint, e.g. a rubber-band or in the vicinity of an attraction field such as a magnet (see figure 3.7).

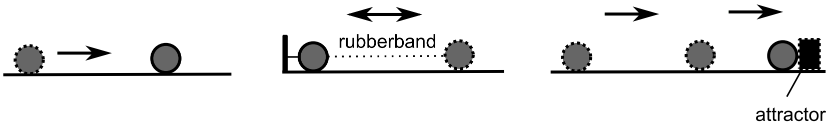


Figure 3.7: Movement without (a) and under constraints of (b) a rubber-band and (c) magnetism

Such explicit descriptions of spatial theories also allow to describe how changes from simple theories to more complex theories may work. The distinction between freely moving objects and objects on a rubber-band may be made by just adapting an axiom for an operation that anticipates the movement of the solid object:

- A freely moving object moves from A to B
- An object on a rubberband moves from A to B and then back towards A.

For the description of an object in the vicinity of a magnet further constraints have to be considered. If the object is magnetic at all it could be repelled or attracted. In the subsequent chapter I discuss three selected examples for sequences of theories in order to find mechanisms that explain how to advance from simple to more complex theories. The paradigm of symbolic description is a necessary abstraction step to investigate these mechanisms.

3.7 Summary

In the course of the thesis a simple model of theory acquisition and change based on observations of the environment is proposed. A theory driven agent that is exposed to an environment has been introduced. Theories are explicitly given to the agent and not built by the agent. The agent holds algebraic theories about an environment and tests these theories (see figure 3.1). Frequent mismatches between observation and prediction elicit changes in the algebraic structure. Affordances limit the operations available to the agent. The agent chooses among the available algebraic theories the theory that fits best to its observations.

The present approach is a commitment to symbolic modeling. An explicit representation of naive theories is the aim of the research. Therefore the algebraic approach has been favored over connectionist models and stochastic methods.

The formal theories developed in this thesis are a contribution to the naive geography project. The present research approach concentrates on empirical data that are prelinguistic in order to overcome language constraints and artificial environments. This is novel in comparison with previous work carried out utilizing image schemata. The following chapter provides empirical data from studies carried out in developmental psychology for commonsense spatial theories.

Chapter 4

Sequences of Theories

Theories develop in sequences. In the following chapter I am going to review empirical data that supports sequences of spatial theories. Incrementally growing theories for the occlusion, containment and support of objects are presented. Based on empirical studies three mechanisms to build sequences of theories have been worked out.

4.1 Empirical Studies

In this thesis I refer to the type of empirical studies that interpret the behavior of infants in a laboratory. As infants cannot communicate what they know about the world, special designed studies have to be carried out to make statements about children's knowledge. Figure 4.1 shows the setup of an empirical study.

The toddler sits on the lap of the parent and observes a test condition. Other stimuli are blocked away from the toddler. A test condition is carried out by an experimenter. A second researcher observes the behavior of the infant, not knowing the infant nor the objectives of the empirical study. The observer cannot see the tested stimulus.

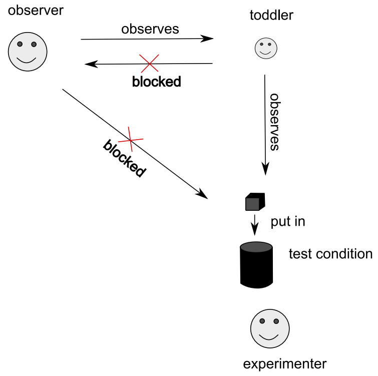


Figure 4.1: Design of a preferential looking study

Passive measure studies interpret indications for the excitement of an infant (heartbeat rate, frequency of pacifier sucking or preferential looking) as a reaction to a novel event (Bower, 1974; Rochat, 2004). The researchers exploit the fact that infants prefer to attend longer to unknown events than to known phenomena (see figure 4.2). Spontaneous looking time declines when the same stimulus is repeatedly presented to an infant. The subsequent presentation of a new stimulus leads to an increase of looking time as it represents a novel event (see figure 4.4). This behavior occurs with children and adults and is utilized by researchers to infer which conceptions the subjects hold about the world.

Figure 4.2: A 3 month old child looking at a novel object

Figure 4.3 illustrates a preferential looking study that tests infants' knowledge of solidity. A group of infants of approximately the same age is repeatedly exposed to the following stimulus. A ball is falling down behind a screen. The screen is lifted by the experimenter and the scene reveals a ball lying on the ground. With every trial a decrease of looking time can be measured (figure 4.4). The infants lose interest in the event, because they know it already. This phase of the experiment is also called the habituation phase.

In order to infer the infants' knowledge of object solidity a platform is introduced in the test phase of the experiment. If infants have a notion of solidity they should expect that a ball that falls down and hits a platform will rest on the platform. A violation of this expectation, i.e. a novel event, should lead to an increased looking time.

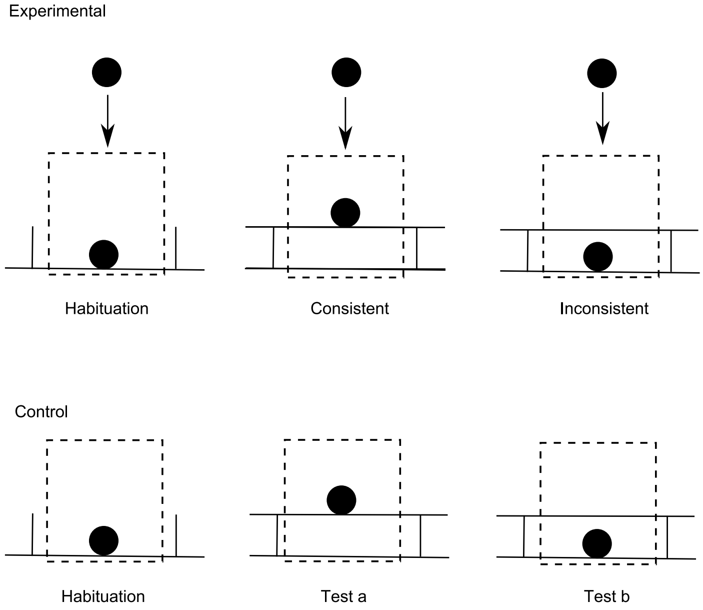


Figure 4.3: Preferential looking study to test 4 month old notion of solidity (source: Spelke et al. (1992))

Spelke et al. (1992) test two groups of children. Both groups are habituated the same way. In every trial the infants see a ball falling down behind a screen. Then the screen is revealed by the experimenter. To the first group of infants a ball is shown resting on the top of the platform. To the second group of children a ball is shown lying under a platform, hurting the principle of solidity, as if the ball would magically pass through the platform.

The group of children that was exposed to the second inconsistent test case showed an increase of looking time (see dotted line in the graph for mean looking time in figure 4.4). Spelke et al. (1992) interpret this increase of looking time as evidence that children in the age of 4 months have a notion of solidity.

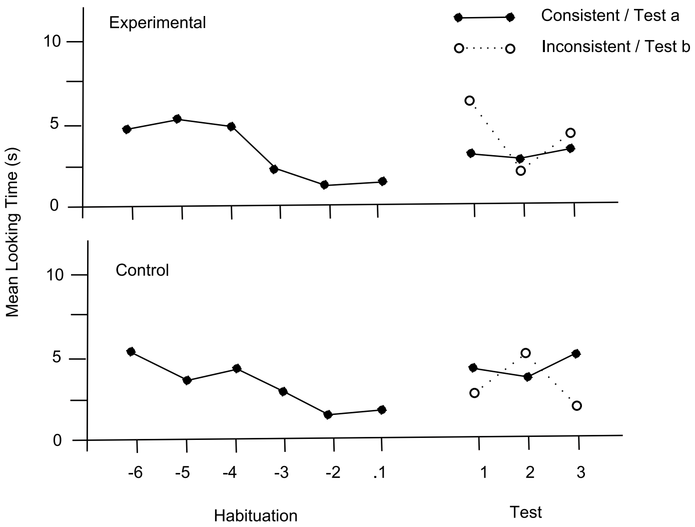


Figure 4.4: Mean looking times in an empirical study on object solidity
(source: Spelke et al. (1992))

In order to eliminate training effects or effects caused by the setup of the experiment, an unhabituated group of children is tested as well. Preferential looking studies appear under different names in the literature like "violation of expectation method" or "visual preference for novelty method" (Baillargeon, 2004a; Rochat, 2004). They have been used to verify concepts of object solidity with infants a few days after their birth. The method is also suitable for testing adults. Further studies tested infants knowledge of collision, occlusion, containment, inertia, gravity, and other events (Spelke et al., 1992; Hespos and Baillargeon, 2001a,b; Baillargeon, 2004b; Luo and Baillargeon, 2005; Rochat, 2004). Older children have been tested using alternative methods such as: the observation of predictive hand reaching (von Hofsten et al., 2000) , neuroscientific methods (Johnson, 1999), and early word studies (MacWhinney, 2000). The following table lists investigated spatial phenomena.

study / image schema	**literature**	**spatial relation**	**influences investigated**
PATH	Spelke et al. (1992)Rochat and Hespos (1996) Hofsten et al. (1998) Kim and Spelke (1999)	AT	gravity, inertia, obstacles
OCCLUSION	Bower (1974) Bower (1989) Baillargeon (2004b) Baillargeon (2004a)Hespos and Baillargeon (2001b) Luo and Baillargeon (2005)	BEHIND	number of objects, windows, size, transparency
CONTAINER	Hespos and Baillargeon (2001a) Hespos and Baillargeon (2001b)Hespos and Spelke (2004)	IN	open/closed, size, transparency
SURFACE	Spelke et al. (1992) McCloskey (1983)	ON	contact, amount of contact, shape
COVER	Piaget (1950) Wang et al. (2005)	UNDER	movement, size, transparency

Table 4.1: Classification of empircial studies by spatial relations

Three sections are dedicated to empirical studies about the occlusion, containment and support of solids. The studies exhibit the sequential development of knowledge. This development will be modeled as sequences of theories.

All studies use the violation of expectation paradigm. The researcher assumes that the infant holds an expectation, in the terms of this thesis a theory about the occlusion, containment and support of solids. A violation of this expectation leads to an increased looking time in the studies.

4.2 Occlusion of Solids

Objects that magically disappear in space have been in the center of psychological research for decades. The permanence of objects has been investigated, i.e. the question if children maintain a representation of objects when they are out of sight or hidden by another object - an occluder.

An object that moves behind an occluder gets hidden. Adults have knowledge about spatial relations between stationary objects. They assume that objects have properties and they know laws that rule the movement and perception of objects. Based on this knowledge they can predict when an object will be visible to the observer. They can explain when an object disappears and reappears (Gopnik and Meltzoff, 1997).

Children do not possess the same knowledge as adults. Many studies have been carried out to investigate object permanence in infancy. There is evidence that the concept of object occlusion is acquired in the first ten months of life (Luo and Baillargeon, 2005). Within the studies two research questions are addressed:

- When is an object that reappears from behind an occluder the same object that disappeared?
- When is an object behind an occluder hidden?

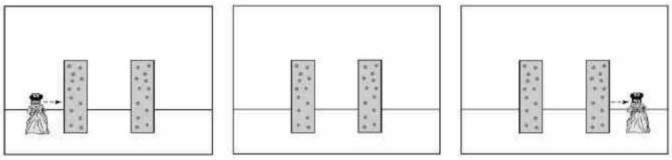


Figure 4.5: Test stimuli for occlusion of an object
(source: Aguiar and Baillargeon (1999))

Empircial studies that investigated whether children expect an object to be hidden when moving behind an occluder have been carried out by Baillargeon and collaborators. Different types of occluders have been used to test if young infants expect an object to be hidden behind an occluder (see figure 4.5). The violation of expectation method was applied for testing. The following results were revealed.

With 2.5 months children seem to distinguish objects just based on the spatial relation behind. In the empircial study they seem to ignore windows in the occluder, as well as the height and width of the occluder. A window in the occluder is not recognized before the age of 3 months. Infants expect a higher object to be hidden behind a lower occluder with about 3.5 months. But expectations about wider objects not being hidden behind smaller occluders do not appear in the studies before the age of 7 months. Figure 4.6 shows the stimuli for testing whether infants know that a wider object should be hidden behind a windowless, taller (in height), smaller (in width) occluder.

The upper photo series (figure 4.6) shows a solid object that is narrower than an occluder. In the second upper photo a screen is set up and the object is lowered. The third upper photo shows the object hidden behind the occluder. The series illustrates the physically possible outcome of a hiding operation.

The photo series in the figure below shows an impossible outcome of the operation carried out. In the first photo below it can be seen that the object behind the occluder is wider than the occluder. In the second photo a screen is set up and the object is lowered. In the third photo the object is shown hidden behind the occluder. The series illustrates the physically impossible outcome of a hiding operation.

If children show a sign of surprise when exposed under controlled conditions to the second series of stimuli then developmental psychologists infer that infants consider width as an occlusion constraint. Infants younger than 7 months are not surprised when exposed to the impossible stimuli presented here.

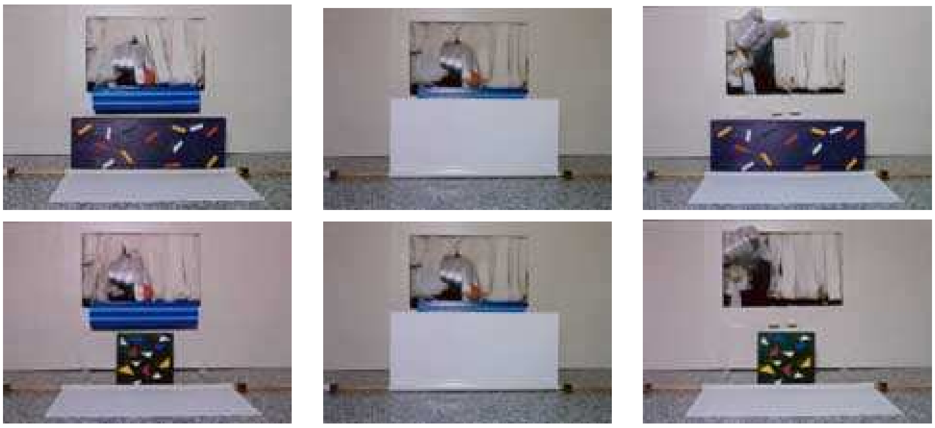


Figure 4.6: Testing the width of an object as occlusion variable
(source: Renee Baillargeon - infant cognition lab,
http://www.psych.uiuc.edu/˜rbaillar/ICL/welcome.html)

Transparency as an occlusion variable is considered at around 7.5 months of age. It is up to the age of a year that children can predict when a moving object that disappeared behind an occluder should reappear on the other side based on judgments about the speed of the object and width of the occluder (Mareschal, 2000). Table 4.2 summarizes these findings.

percept	**expectation** **B occludes A if**	**age** (months)
spatial relation	A is *behind* B	< 2.5
Structure of solid (doorways and windows)	B has no window	3
height of solid	height A $<$ height B	3.5
width of solid	width A $<$ width B	7
transparency of solid	B is not transparent	7.5

Table 4.2: Solid Occlusion Theory Sequence

4.3 Containment of Solids

Although the mouth may be one of the first containers that is experienced, the development may be slow due to the difficulty that arises when connecting the self to the external world. A number of studies have been carried out using the violation of expectation paradigm. Empirical studies about infants ability to distinguish between objects that can contain and those that cannot, motivate a sequence of theories (Hespos and Baillargeon, 2001a,b; Baillargeon, 2004a; Hespos and Spelke, 2004).

Within the studies two research questions are addressed:

- When is an object that reappears from inside a container the same object that disappeared?
- When is an object inside a container hidden?

Infants at the age of 2.5 months already seem to know that solids that have an opening may act as containers (Hespos and Baillargeon, 2001a), as well as that a solid in a container shares the movement with the according container. The width of an object is not considered as a containment variable before the age of 4 to 6 months.

Another series of studies has been carried out to reveal prelingual knowledge with infants about containment (Hespos and Spelke, 2004). Concepts of space do not necessarily have to be reflected in the language we use. Objects that can be moved inside a container may be distinguished between objects that cannot be moved inside a container. Korean adult speakers use the verb "kitta" focusing the movability of solid objects (tight-fit) while English adult speakers focus the containment relation of the involved solids (Choi et al., 1999). Figure 4.7 illustrates the difference of a loose-fit container on the left side vs. a tight fit container event on the right side. This functional distinction between loose-fit and tight-fit containers is not supported in the English language although adults show looking behavior according to both concepts in violation of expectation studies (Bloom, 2004; Hespos and Spelke, 2004).

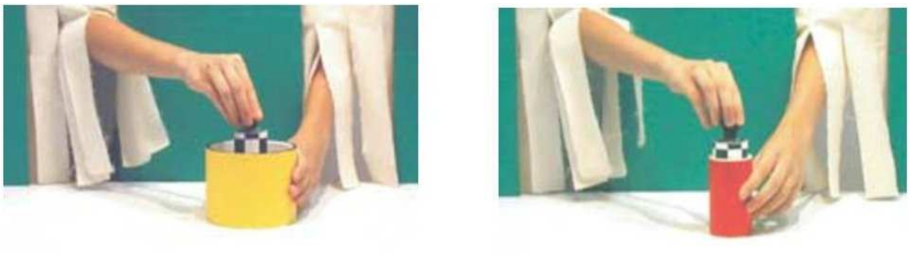


Figure 4.7: Loose-in vs. Tight-in Containment
(source: Hespos and Spelke (2004))

Empirical studies have shown that both English and Korean five month old infants can perfectly maintain the difference between "loose-fit" and "tight-fit" containers. After repeated presentation of a tight-fit container subjects have been exposed to a loose-fit container (or vice versa). An increase of spontaneous looking time has been observed indicating that subjects distinguish between both types of containment (Hespos and Spelke, 2004). Hespos and Baillargeon (2001a,b) further revealed that infants younger than 7.5 months are not surprised when exposed to the impossible stimulus presented in figure 4.8.

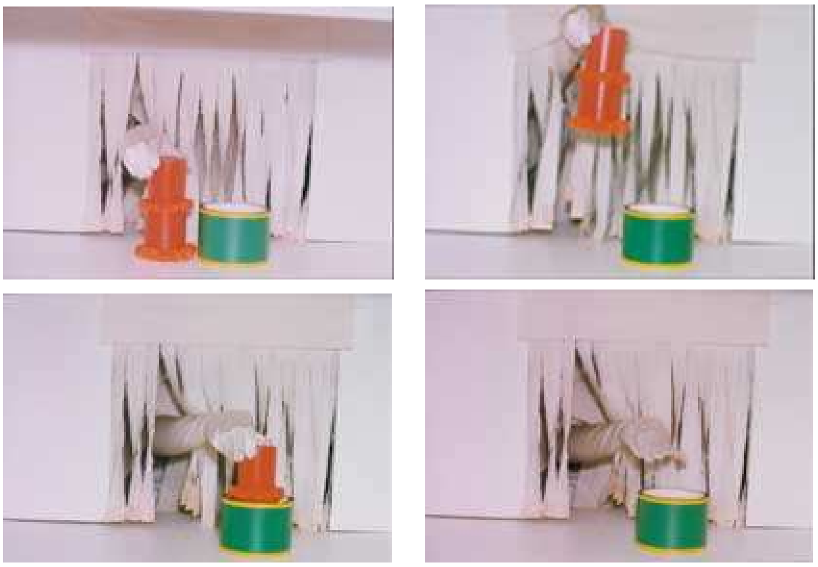


Figure 4.8: Violation of expectation study to test height as a constraint in containment events

(source: Renee Baillargeon - infant cognition lab,
http://www.psych.uiuc.edu/˜rbaillar/ICL/welcome.html)

In a series of violation of expectation experiments 7 months old infants have been tested under controlled conditions when they expect a taller object to be hidden in a smaller container. If infants know that height constrains the containment of an object they should be surprised by the outcome of the operation illustrated in figure 4.8. If they do not consider height as a constraint of occlusion events they should not show signs of surprise when observing the operation carried out shown in figure 4.8. The latter was found with infants 7 months old. Hespos and Baillargeon (2001b) interpret the results of the studies such that 7.5 months old children consider height as a containment variable.

percept	**expectation** **A is in B if**	**age** (months)
movement of solid	A shares movement with container B	2.5, 3.5
opening of solid	B has an opening	2.5, 3.5
width of solid	width A < width B	4 - 6
movement and spatial relation	movability of A inside B	5
height of solid	height A < height B	7.5
transparency	B is not transparent	10

Table 4.3: Solid Containment Theory Sequence

The developmental sequence of a containment theory for solid objects is summarized in table 4.3. It is not before 10 months that infants anticipate a contained solid to be visible in a transparent container (Baillargeon, 2004a).

4.4 Support of Solids

Children start very early to observe that objects fall down such as pacifiers, teddy bears, and bottles. Empirical studies have been carried out testing the knowledge about the support of objects (Baillargeon, 2004b). The violation of expectation paradigm (Luo and Baillargeon, 2005) has been used again. Within the studies the following research question is addressed:

- When is an object supported by another object?

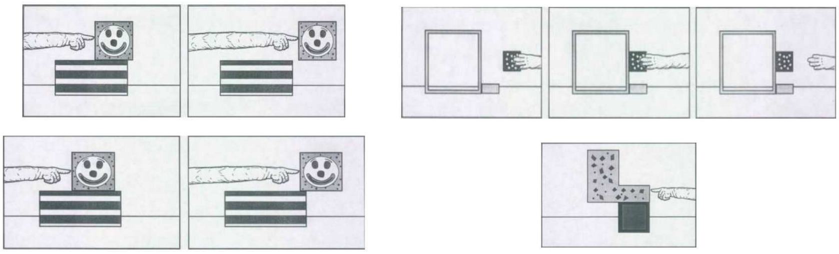


Figure 4.9: Stimuli for the object support experiments

(source: Baillargeon (2000))

Figure 4.9 illustrates the stimuli used in the empirical studies. On the top left side of the figure an object is released in midair and does not fall. Infants as young as 3 months are surprised by this event indicating that they have a notion of object support. On the top right side of the figure the two objects have contact via their side surfaces. Though adults would not conceptualize this situation as object support, infants younger than 4.5 to 5 months do not show any sign of surprise when object B does not fall (Luo and Baillargeon, 2005). After that they consider that the supporter has to have contact on the top surface with the supported object.

With 5.5 months infants start to distinguish between objects that are movable on the supporter via objects that are not movable when put on another object, such as a ring on a zylinder (see figure 4.10). The behavior could be observed with infants that grow up in Korean speaking environments where the relation is distinguished by language. Although the distinction is not supported in English speaking environments, English infants distinguish between loose-fit and tight-fit supporters (Choi et al., 1999; Hespos and Spelke, 2004).

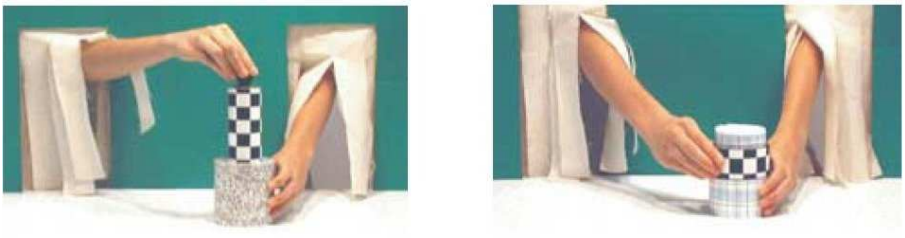


Figure 4.10: Loose-in vs. Tight-in Containment
(source: Hespos and Spelke (2004))

With 6.5 months infants detect that object support is also dependent on the amount of top contact as illustrated in figure 4.9 on the bottom left. A further series of experiments gives evidence that around 12.5 months old children will also consider the shape (see figure 4.9 bottom right) of a solid to have influence on the support relation. The test case can be seen on the bottom right of the figure 4.9. Table 4.4 summarizes the findings.

percept	**expectation** **A *is ON* B if**	**age** (months)
-	A has contact with B	<=3
type of contact	A has contact on top of B	4.5 - 5.5
movement	A is movable on B	5.5
amount of contact	amount of contact A B > experienced threshold	6.5
shape of supported solid	shape of B supported by A	12.5

Table 4.4: Solid Support Theory Sequence

4.5 Theory Change

The presented empirical studies serve to support the hypothesis of the thesis that spatial theories can be described by algebraic specifications and that change can be modeled by adapting the axioms of the algebraic specifications. An adaptation is not as radical as a change. It makes a theory fit to the observations of the environment. In order to adapt a theory three mechanisms of theory change will be introduced in the following section. The mechanisms have been found in the empirical data previously presented.

4.5.1 Specialization - Detecting Constraints

The developmental sequence for the support of solids presented in section 4.4 lists more and more constraints. An initial theory of solid support considers firstly just the contact between the solids. An advanced theory for solid support can be gained by distinguishing between the type of contact. A solid is just supported if it contacts the supporting solid on the top surface. The axiom “A has contact with B” is extended to the axiom “A has contact with B and A has contact on top of B”.

Figure 4.11 illustrates how the simple theory is extended by adding an axiom. “+” and “-” indicate the truth value of the axiom. “+” stands for true while “-” stands for false. Further constraints are a consideration of the amount of contact, and the shape of the supported solid object.

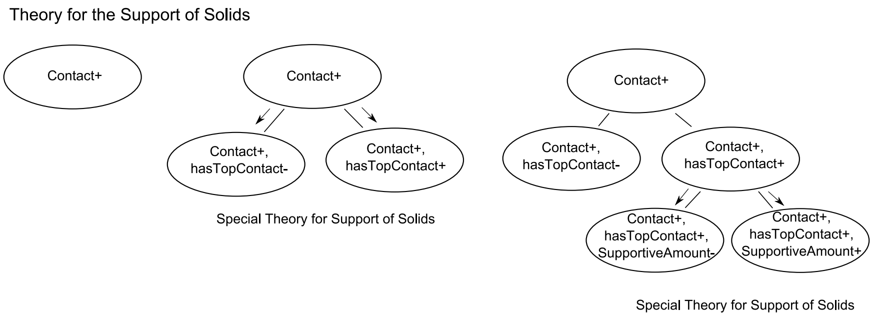

Theory for the Support of Solids
Contact+
Contact+
Contact+, hasTopContact-
Contact+, hasTopContact+
Special Theory for Support of Solids
Contact+
Contact+, hasTopContact-
Contact+, hasTopContact+
Contact+, hasTopContact+, SupportiveAmount-
Contact+, hasTopContact+, SupportiveAmount+
Special Theory for Support of Solids

Figure 4.11: Specialization of a theory

Figure 4.11 further shows that a specialized theory can be won by adding an axiom to the current theory. The step of adding an axiom with further constraints will be called a specialization step in the course of the thesis. Table 4.5 illustrates the specialization steps in the theory sequence for the support of solids.

PERCEPTION	**axiom -** **A *is* *ON* B if**	**age** (months)	**CONCEPTUAL CHANGE**
Contact	A has contact with B	<=3	
Type of contact	A has contact on top with B	4.5-5.5	SPECIALIZATION Top and side contact
Amount of contact	amount of contact A B > experienced threshold	6.5	SPECIALIZATION amount of contact supportive and unsupportive

Table 4.5: Theory Sequence for the Support of Solids by Specialization

The empirical data for the occlusion of solids also point to the mechanism of specialization. An excerpt of the theory sequence is summarized in table 4.6, the full sequences can be found in the appendix of the thesis. The sequence of occlusion theories emerges through subsequently adding more and more axioms.

PERCEPTION	**axiom - A is occluded by B if**	**age** (months)	**CONCEPTUAL CHANGE**
-	A is *behind* B	< 2.5	-
Structure of occluder (presence of doorways and windows)	B has no windows	3	SPECIALIZATION no Window B
height of the objects involved	$A < B$	3.5	SPECIALIZATION $A < B$

Table 4.6: Theory Sequence for the Occlusion of Solids by Specialization

Both theory sequences motivate the introduction of a *specialization* step. The specialization of a theory is the adaptation of one of its axioms by adding further constraints. The theory is split into two specialized sub-theories. In summary a theory can be constrained by adding subsequently more axioms to it. Change is achieved by considering a further constraint. The changed theory will be called a specialized theory.

4.5.2 Generalization - Detecting a Special Case

Having acquired specialized theories it motivates the introduction of generalized theories. Generalization is defined - following the definition of induction in the problem solving literature - as the combination of particular instances through observation to more general laws. Induction tries to find regularity and coherence between observations (Polya, 1973). An example in the history of science is the detection of non-Euclidean geometries that made the Euclidean geometry a special case of more general geometries (see discussion in section 2.2.1).

Evidence for the generalization of theories could also be found in the empirical data. A generalization step deletes an axiom of a theory to generate a more abstract theory. Special cases can be won by carrying out specialization steps on the general theory. The generalization of a theory will be illustrated by two examples.

The theory of occlusion advances through several specialization steps until the detection of transparency. Until then the theory for the occlusion of solid objects can generate an expectation about a hidden object considering the spatial relation behind, windows in the occluder, and the size of the involved objects.

Transparent objects will lead to contradiction. Although a solid object is behind a bigger, windowless occluder it can be still visible, due to the transparency of the occluder.

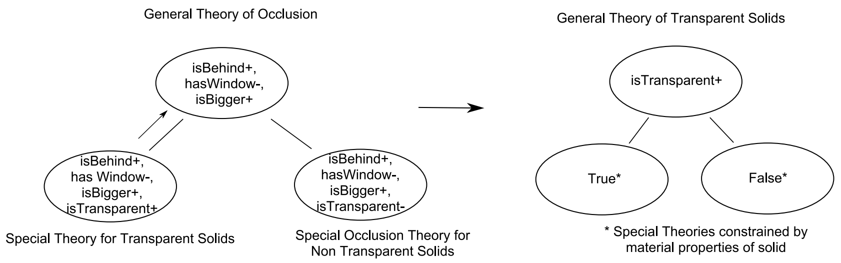

General Theory of Occlusion
isBehind+, hasWindow-, isBigger+
isBehind+, has Window-, isBigger+, isTransparent+
isBehind+, hasWindow-, isBigger+, isTransparent-
Special Theory for Transparent Solids
Special Occlusion Theory for Non Transparent Solids
General Theory of Transparent Solids
isTransparent+
True*
False*
* Special Theories constrained by material properties of solid

Figure 4.12: Generalization Step in Theory Sequence for Object Occlusion

With the detection of transparency the current theory can be identified as a special case for non transparent occluders. The current theory (figure 4.12 top) is a general theory for object occlusion having two specialized theories for transparent and non transparent occluders (figure 4.12 bottom). The transparency of solid objects can again be defined as a new general theory that depends on the material properties of the solid.

A second example for the generalization of theories is the detection of movability under a spatial relation. As pointed out in section 4.3 the containment of a solid object may be defined through the opening of the container, the shared movement of container and contained solid object and the size of the involved objects. The distinction of loose vs. tight fit containers requires the detection of motion inside the container.

A theory that describes a movable solid object inside a container is a special case of a more general theory of containment (see figure 4.13). The theory can predict when an object is inside a loose-fit container but it does not suffice to anticipate tight-fit containment. In order to obtain a general theory about containment the “movability axiom” has to be deleted of the loose-fit container theory. The general theory of containment can then be specialized with a new axiom towards a theory of tight-fit containment.

Figure 4.13 illustrates the generalization step for a containment theory sequence of solid objects. The specialized theories can be transformed to a more general theory by deleting an axiom. The general theory can be again specialized by adding another axiom. The movability of an object can be defined in a new general theory depending on the spatial relation of the involved objects and their physical properties.

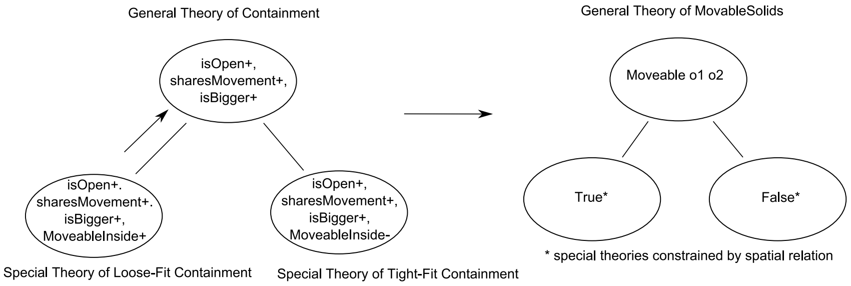

General Theory of Containment
isOpen+,
sharesMovement+,
isBigger+
isOpen+.
sharesMovement+.
isBigger+,
MoveableInside+
Special Theory of Loose-Fit Containment
isOpen+,
sharesMovement+,
isBigger+,
MoveableInside-
Special Theory of Tight-Fit Containment
General Theory of MovableSolids
Moveable o1 o2
True*
False*
* special theories constrained by spatial relation

Figure 4.13: Generalization Step in Theory Sequence for Object Containment

Theory sequences - as described in the present section - motivate the introduction of a *generalization* step. A general theory can be won by the deletion of the axiom of a special theory. Change is achieved by omitting a constraint. The changed theory will be called a generalized theory in the course of the thesis.

4.5.3 Weighting - The Importance of a Theory

Weighting is a crucial mechanism that explains the coexistence of several theories. Naive theories are not replaced but rather fade out by using weights. Dynamic weighting allows to describe how the infant can hold several theories at a time and switch between them. Here a weight is defined as the ratio of successfully predicted to totally observed actions in the environment.

Figure 4.14 explains how the weights can influence the choice of a naive theory. The importance of the theory, i.e. its weight is indicated by the size of the surrounding ellipses in the figure. On the left side the case of solid object support is illustrated whereas the right side shows solid object containment.

The presented empirical data point to developmental sequences, e.g. (Baillargeon, 2004a), that are in accordance with the figure. Some sequences can be influenced by specially designed learning trials, which again points to a weighting mechanism. Generally children learn first that contact is important for object support then they learn that the type of contact is important, followed by the consideration of the amount of contact. The theory that uses just the contact information to predict the support of two solids will lose importance after a while so it is weighted lower than the theory that considers contact and type of contact. On the left side of figure 4.14 a snapshot of theories is shown at a timepoint where the theory for solid object support explains best operations in the environment that are dependent on the parameters of contact, type of contact and amount of contact.

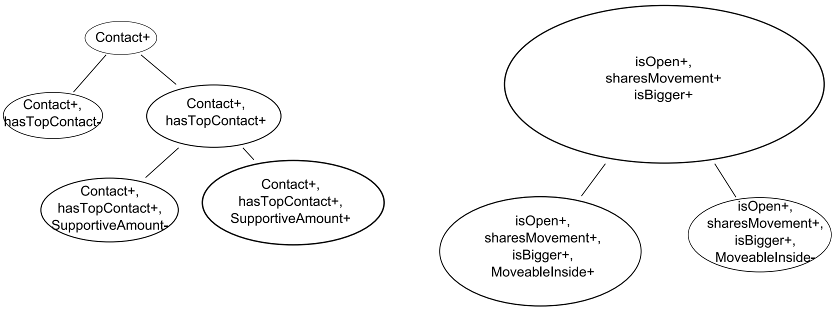

Contact+
Contact+, hasTopContact-
Contact+, hasTopContact+
Contact+, hasTopContact+, SupportiveAmount-
Contact+, hasTopContact+, SupportiveAmount+
isOpen+, sharesMovement+ isBigger+
isOpen+, sharesMovement+, isBigger+, MoveableInside+
isOpen+, sharesMovement+, isBigger+, MoveableInside-

Figure 4.14: Weighting of a theory

On the right side of figure 4.14 the weighting for theories on solid object containment is illustrated. The snapshot shows a timepoint where the agent holds a theory for containment that depends on the opening of the container, the shared movement and the size of the involved objects. The agent also holds two specialized theories that additionally depend on the movability of the object inside. Both specialized theories have lower weights (smaller ellipses in the figure) than the general theory of solid object containment. The figure also shows that the loose-fit containment theory (movability in the container) has been more often observed than the tight-fit containment theory.

With a dynamic weighting mechanism a gradual theory revision can be described in dependence on experience infants made in the environment. Infants may receive evidence that lead to “wrong means-end relations” . They could generate a theory like all objects that taste like strawberry fall down. Such a theory may be supported for some time by tasting falling strawberries but the major part of evidence in the environment will point to unsupported objects that fall down. The “strawberry” theory will gradually fade out. Note that it still continues to exist and may be reused at a later point of time.

The mechanism of *weighting* has been identified in the empirical data as a method to assign importance to a theory and facilitate its choice. The weight determines the importance of a theory by comparing the successfully predicted with the totally observed actions carried out in the environment. Theories with high weights fit better to the agents current observation data than those with lower weights.

4.5.4 Summary

The empirical studies point to three mechanisms of theory change.

1. **Specialization** considers a new influential parameter. A theory is specialized by constraining it through an axiom. The axioms constrain the theory to a special set of sorts and operations. The more axioms are added the fewer sorts can be described by the specialized theory.

2. **Generalization** is an abstraction step. A theory is found to be a special case of a more general theory. A number of other special theories may exist that can be derived from the newly created generalized theory. Coexisting theories are special cases of generalized theories.

3. **Dynamic Weighting** is a mechanism to assign importance to a theory. Theories with higher weights are favored to those with lower weights. Belief revision is the result of the dynamic weighting mechanism that assigns a higher weight to a previously low weighted theory.

4.6 Lattice of Infant's Theories

The mechanisms presented in the previous section serve to connect theories in a partial ordering, specifically a lattice. Recently Frank (2006) proposed a taxonomic lattice of distinctions, the idea is here extended to axioms. The novel contribution is a weighting mechanism based on observations of the environment in combination with a lattice. The infant's theories are described in a weighted lattice. I am going to give examples how weighted lattices describe theories the infant holds.

In order to store all naive theories an infant can hold each node in the lattice corresponds to a theory. The agent holds a theory of object containment, and after observation the agent builds theories for loose-fit and tight-fit containers. These are combinations of the containment and movability theories that can be built by *joining* operations. Further evidence may point to more general theories that can be built by *meeting* operations.

Not all theories created in that way are useful. I distinguish three types of theories: absurd, plausible and established theories. Absurd theories can never be observed in the environment, such as a theory of an object being contained and not contained at the same time ({isIn+,isIn-}) or the bottom theory of the lattice that fulfills all axioms at the same time. Absurd theories exist but will not receive evidence and have therefore weight 0 (figure 4.15).

Plausible theories are theoretically possible theories but their occurrence has not yet been observed. Every observation will increase the weight of the theory. An observed theory has been given evidence through observations in the environment. An object is inside another object and movable. Plausible theories have a low weight, such as the theory of non movable objects in a container in figure 4.15 ({isIn+,isMoveable-}).

Established theories have strong evidence. They have been observed frequently, such as an object being movable {movable+} or inside another {in+}. Therefore these theories have a high weight, indicated by the bigger ellipses in figure 4.15. Theories with higher weights are preferred to theories with lower weights and guide the construction of new theories.

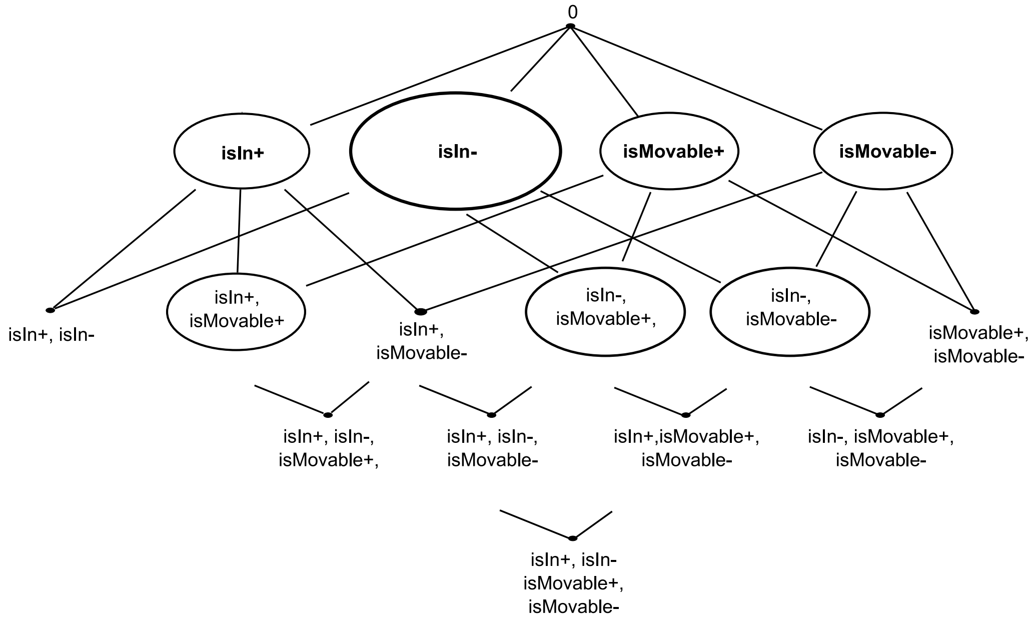


Figure 4.15: Lattice of loose-fit and tight-fit container.

In summary three mechanisms identified in the empirical data of this chapter lead to a weighted lattice of naive theories.

1. A specialization step is movement down in the lattice of theories.
2. A generalization step is a movement up in the lattice of theories.
3. Setting weights in the lattice of theories determines the importance of a theory as a basis to choose between several competing theories and to model gradual theory revision.

Starting with m pairwise mutual exclusive inputs one ends up with n = 2*m inputs. The lattice would then have 2^n elements. The problem of computability arises. However there is some empircial evidence that infants learn domain specific. Furthermore I assume that the number of possible elements is constrained by affordances. Thus the number of contributing elements in such a lattice is small.

4.7 Summary

Empirical studies of developmental psychology have been presented to identify mechanisms to build sequences of spatial theories. The studies are based on the violation of expectation paradigm. It assumes that infants generate beliefs out of their knowledge about the physical world. Infants hold theories about the world (Gopnik and Meltzoff, 1997; Gopnik et al., 1999; Meltzoff, 2004).

The infants react with measurable signs of surprise whenever exposed to novel stimuli. Different stimuli contradicting common-sense knowledge about the physical world are presented to the infants at different stages of age. A sign of surprise proves the developmental psychologist that infants have a notion about the tested knowledge. So developmental sequences for the support, occlusion and containment of solids are identified.

I used these sequences to generate naive theories about the support, occlusion and containment of solids. During the modeling process with the empirical data three theory changing mechanisms could be identified. Specialization steps constrain theories by adding more axioms. Generalization steps change theories by deleting axioms and adding new specializations. Weighting serves as a mechanism to assign the importance to a theory in the light of evidence provided by the environment.

The three mechanisms serve to build weighted lattices of naive theories. The following chapter 5 will provide an abstract model for theory change triggered by the environment. The model will be used to carry out simulations based on the empirical data described in this chapter.

Chapter 5

Abstract Representation - A Model in Haskell

The following chapter verifies the hypothesis that spatial theories can be described by algebraic specifications and that a change mechanism for new theories can be based on the adaptation and weighting of axioms. The algebraic specifications serve to provide a formal model of a theory driven agent. The agent's mental model is based on theory sequences conforming to the *theory theory* of cognitive development.

Theories are described by algebraic specifications (for details see appendix). As they are based on constructive axioms it has been suggested to say model based (executable) specification method, rather than using the term algebraic specifications. The purely functional programming language Haskell has been used to implement the specifications. Three mechanisms have been formally modeled with the empirical data presented in the previous chapter:

- Specialization: Adding a new instance, i.e. a specialized theory to an existing class.
- Generalization: Adding a new class, i.e. a generalized theory. In order to use the general theory an instance has to be created. For all new theories that can be derived from the generalized theory instances have to be added.
- Weighting: Calculating the weight of a theory, i.e. counting successful anticipations vs. observations of operations in the environment.

These mechanims served to implement a theory driven agent that observes the outcome of operations in an environment. The agent advances through sequences of spatial theories. The agent's mental model is based on two theory forming cycles that are described by a sequence of `observe-test-build-use` functions.

5.1 Theory Driven Agent and It's Operations

The current model represents the internal operations of a theory driven agent. Although the agent model is very simple it can be shown how theories can be chosen based on observations of an environment. The model has been tested with empirical data in a simulation in chapter 6.

Each theory driven agent has a unique identifier. The agent holds observations made in the environment, a list of current theories, and a list of all possible theories, called the potential. The potential is necessary to make the agent learn. The current model does not build theories automatically.

```
data Agent = Agent ID Observation Theories Potential State
data State = Observe | Build | Test | Use | GiveUp deriving (Show,Eq)
type Potential = Theories
```

The agent has been implemented as a finite state machine (see figure 5.1). One of the four states: `Observe, Build, Test, Use` determines the agent's next step in order to carry out sequences of mental actions. The state transition diagram in figure 5.1 illustrates how the agent advances through different theory constructing episodes.

Each episode has a starting point with high interest into the problem domain and an ending point with low interest into the problem solving domain. The end point or point of low interest is reached when the agent found a satisfying explanation, i.e. a theory that explains the environment or when the opposite happens namely the agent is unable to find a theory that fits the observations of the environment. The starting point of the diagram can be interpreted as high interest into a domain under investigation towards the end point of the diagram, that stands for a timepoint having low interest into the domain of investigation.

Currently this research focuses on describing redescription of spatial theories rather than modelling the agent's emotional and attentional parameters. Therefore currently measures for the level of interest, attention and the level of frustration have not been cosidered in the agent's model. Extensions are possible in future models.

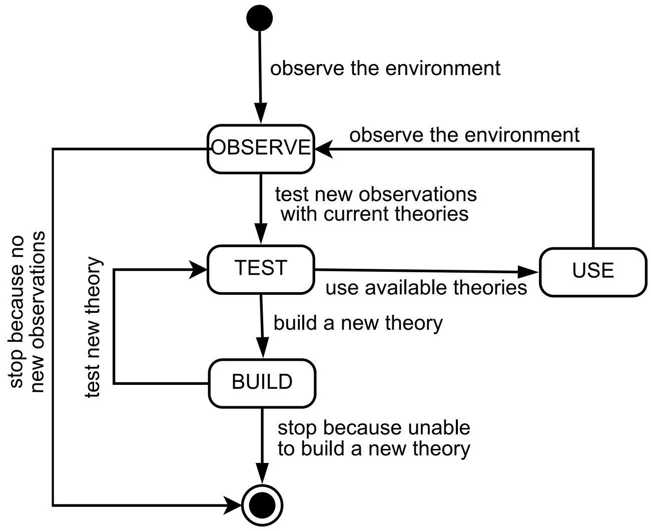


Figure 5.1: State transition diagram for the agent

The agent starts by *observing* the environment holding a simple theory. The newly observed data are *tested* with the simple theory the agent holds. If the theory can explain the observed data the agent continues to *use* the simple theory and *observes* the environment until no new data arrives into the model.

If the theory does not fit the observed data the agent will try to *build* a new theory based on the perceived input and the old theory. The old and the newly built theory are then *tested* with the previously observed data. If the observed data conforms the new theory the agent continues to *use* the simple theory and *observes* the environment until no new data arrives into the model.

In the case that the new theory does not fit the observed data the agent loops back to the build theory state. If the agent can not build a new theory the model stops. Building a new theory has not been implemented in the present model. It could be based on a deterministic or stochastic inferencing engine.

The current research concentrates on describing the overall process of testing theories. The implementation of a theory inferencing engine is future work. At the present theories are selected from a set of possible theories that have been hard coded. The hard coded theories are the agent's potential. This is what the agent could learn if it perceived observations that lead to theory change.

The following four sections describe the mental operations of the agent in detail. The agent utilizes an *observe-build-test-use* cycle that makes it advance through sequences of theories given observation data from the environment. This has been motivated by previous work of Siegler and Chen (2002). The agent's actions are a reaction to observations of the environment and beliefs generated.

```
class Agents ag where
    observe           :: Observation -> ag -> ag
    buildTheories    :: ag -> ag
    testTheories     :: ag -> ag
    useTheories      :: ag -> ag
```

5.1.1 Observe Function

The agent observes the outcome of operations carried out in an environment. These observations are described in the model in an abstract form. Generally an observation is described by a data type consisting of a list of affordances and a list of experiments.

The affordances are modeled using strings. They serve to connect the observations to the appropriate theory in a simple matching process. The observations coupled with affordances control which theories will be chosen out of the agent's memory.

```
Observation = O [Affordance] ExpSerie deriving Show
```

An experiment involves two objects and the outcome of the experiment. The outcome of the experiment is modeled as a Boolean variable. The mapping to Boolean values is in accordance with observations of psychologists that found across languages that young children verbalize success ("There", "Done it", "Good") and failure ("Oh dear", "no") Gopnik and Meltzoff (1997).

```
type Exp = (Obj,Obj,Bool)   -- an experiment
type ExpSerie = [Exp]       -- a series of experiments
```

The agent's observe function serves to internalize the observations made in the environment into it's mental model. The observation function has influence on the state of the agent. Depending on the amount of information that arrives into the agent's internal memory, the agent will change it's internal state.

```
observe (O aff []) (Agent iD o t p s)  = (Agent iD (O aff []) t p GiveUp)
observe exps (Agent iD o t p Observe) = (Agent iD exps t p Test)
observe exps (Agent iD o t p Build)    = (Agent iD o t p Build)
```

Initially an agent is in the state `Observe`. An agent that receives data from the environment will change it's state to `Test`. The new data have to be tested with the theories that are held in the agent's mental model. However if the agent receives no input from the environment it cannot test it's theories and will change the state to `GiveUp`. An agent that is in the state `Build` has no need to acquire new observations. An agent that is in the state `Build` tries to build theories. The agent therefore has no need to acquire other types of observations. The agent will firstly resolve the conflict generated by previous observation types in a theory building process and continue to observe new information at a later point of time.

5.1.2 BuildTheories Function

An agent that is in the state `Build` tries to build theories. This happens whenever the observations of operations in the environment stand in contradiction with expectations derived from theories (beliefs). The changes between the theories have been classified, but will not be carried out automatically by the model. This step could be done in future work by an inference machine based on many-valued logic or stochastic reasoning. Instead I provide all the possible theories an agent can have and the agent chooses among them based on affordances given by the environment.

```
buildTheories (Agent iD o t p Build)
   |newt == [] = (Agent iD o t p GiveUp)                    -- error
   |otherwise  = (Agent iD o (t++newt) p Test) where
        t’ = [pt|pt<-p, (inSide (getObsAff o) (getTAff pt))]
        newt = [nt|nt<-t’,not (elem nt t)]            -- new theories
buildTheories ag = ag
```

The decision to store the affordances with the theories has been made due to model simplification. I do not implement a multiple belief reasoner in this research. The affordances that come with the input data to the model are utilized to match the appropriate theories that are hard coded. The affordances serve to match theories to perceived inputs. The building process is here reduced to choose a finite set of theories among all available theories.

5.1.3 TestTheories Function

The task remains to choose the appropriate theory and test its usefulness. By a dynamic weighting mechanism theories are graded. The gradation of the theories changes the state of the agent. An agent that is successful in testing a newly acquired theory changes it's state to `Use`. An agent that unsuccesfully tests a theory will change it's state back to `Build`. The agent tries again to find another explanation, i.e a theory. This process is iterative if the agent cannot find any theory that explains the observed data it will set it's state to `GiveUp`.

```
testTheories (Agent iD o t p Test) = (Agent iD o wt p newState) where
                        newState  = determineState wt
                          -- weighted theories
                        wt = [(T i a op (eval-
Exp op test data'))|(T i a op w)<-t]
                          -- create as much test data as theories
                        test data■ = replicate (length t) test data'
                        test data' = getExpSerie o
    testTheories ag = ag
```

The code sample above illustrates the `testTheories` function. The function creates multiple copies of the observation data for each theory in the memory of the agent. The set of observation data is used to test the current theories in the memory of the agent. The implementation of this evaluation step is described in the following section.

5.1.3.1 Evaluation Method

The agent observes the execution of operations in the environment and compares them with beliefs generated from theories. The agent can distinguish between four types of theories. In terms of a four valued logic (Belnap, 1977):

- Theories can be *true or false*. The agent has to acquire test data in order to evaluate the theory's usefulness.
- Theories can be only *true*. The agent can infer that the theory fully explains the observed test data.
- Theories can be only *false*. The theory received no positive evidence, but it is still kept in the agent's memory as some other data may verify the theory.
- Theories can be both *true and false*. The agent needs has to adapt the theory as some elements of the theory's axioms do not predict outcomes of operations well enough.

A mechanism that is purley based on logic (cf. (Bower, 1989)) seemed too rigid to model theory change. The immediate rejection after one falsification of a theory contradicts Kuhn's (1962) resistance against theory change that could also be observed with infants (Gopnik and Meltzoff, 1997). Therefore the evaluation of the theories is based on a dynamic weighting mechanism. Dynamic weighting allows to model gradual theory change and has been proposed in the cognitive science literature (see Regier (1996) or Newcombe and Huttenlocher (2003)).

```
class (Show a, Show b) => Evaluation a b where
    testExp  :: a -> b -> b
    testExps :: a -> [b] -> [b]       -- Evaluates a series of experiments
    testExps f e = map (testExp f) e

    evalExp  :: a -> [b] -> Float     -- Number of hits in an experiment is a
                                      -- utility score -> 0.0 - 1.0
```

Actually the number of objects can be arbitrary. If an experiment type with another number of objects is introduced an instance has to be added to the evaluation class. The evaluation function has to be of the form `[o]->Bool`.

```
--
Evaluation of Experiments having two objects, mapped on an equivalence class
instance Evaluation (Object->Object->Bool) (Object,Object,Bool) where
    testExp f e@(o1,o2,observation) = (o1,o2,evidence) where
        evidence                = observation == belief
        belief                  = f o1 o2

    evalExp f e = no_hits / no_exps where
        no_hits = fromIntegral (length hits)
        no_exps = fromIntegral (length e)
        hits = [(o1,o2,t)|(o1,o2,t)<-(testExps f e),t==True]
```

The `testExp` function tests a single observation with a theory, it compares observations with beliefs and returns a Boolean value. The `TestExps` function has been created to test a series of theories using the `map` function. Finally the `evalExp` function counts successful hits vs. number of all trials and returns a utility score for the theory between `0.0` and `1.0`. A theory can have a maximal utility of `1.0`.

- Theories that are untested have a weight of 0.0.
- Theories that are always tested as false have a weight of 0.0.
- Theories that are always tested as true have a weight of 1.0
- Theories that are tested as true and false have a weight between 0.0 - 1.0

When specialized theories start to oscillate around certain utility scores and the weight for the previously existing generalized theory grows in the same extent then a generalization step has to be carried out. I am going to show this behavior in the simulation data in chapter 6. Theories that converge towards `1.0` should be firstly specialized when new contradictions are detected.

5.1.3.2 Determine State Function

The weighting of the theory influences the agent's state. Whenever one of the theories can explain all the observed data there is no need for the agent to search for a further explanation. The agent sets it's state to `Use.`

```
determineState :: [Theory] -> State
determineState [] = error "Cannot find any theory that fits the given data."
determineState t
    |any (==1.0) $ map (getWeight) t = Use      -- no need to change
    |otherwise                       = Build    -- try a better explanation
```

An agent that cannot perfectly explain the world will be set to the state `Build`. The agent will try to resolve the conflicts out of the given data by combining the contradicting percepts to new theories. The model uses a simple heuristic here, but the focus is not on building the inference machine but to describe the overall process of theory formation and validation.

The way how the calculation is done is not so important. The agent has to be equipped with a mechanism to compare beliefs with observations and a mechanism to score the theories. In the current implementation the weighting mechanism does not treat positive feedback superior to negative feedback. This remains to be done in future work.

5.1.4 UseTheories Function

The `useTheories` function serves to start the next theory testing cycle, by either setting the agent in the state `Observe` or by breaking the loop because no new percepts are found. The function could be extended in future work to implement cognitive effects such as the strength of a stimulus or memory effects by reweighting the theories after they have been tested and before a new cycle starts.

```
useTheories (Agent iD o t p s)
      |s == Use      = (Agent iD o t p Observe)       --
apply theory to novel data
      |s == Build    = (Agent iD o t p Build)         -- build a new theory
```

```
       |s == GiveUp   = error "No percepts or theo-
ries found that fit the data."
```

The useTheories function can be extended in future work to consider measures for the agent's attention, the agent's ability to memorize, and the emotional state of the agent. E.g. Low attention could be modeled by decreasing weights of a newly acquired theory with the useTheories function. This has been omitted because the focus of the research is on the qualitative redescription of the spatial theories.

5.1.5 Simulation Function

The simulation function links the four mental operations of the agent. Intermediate results are connected to a result string. A time limit of two times the length of the observation list constrains the model.

```
sim :: Time -> [Observation] -> Agent -> String
sim time [] ag = error "End of Simulation - no data in the environment"
sim time obs ag
    |time > (2* length obs) = "Agent getting bored -
no new data \n CHECK Simulation time limit"
    |otherwise             = show "Time: " ++ show (time') ++ "\n" ++
                              "----------------------------\n" ++
                              "observing environment ...   \n" ++
                              "----------------------------\n" ++
                              show (observe obs' ag) ++
                              "----------------------------\n" ++
                              "building theories ...     \n" ++
                              "----------------------------\n" ++
                               show (buildTheories $ (observe obs' ag)) ++
                              "----------------------------\n" ++
                              "testing theories ...     \n" ++
                              "----------------------------\n" ++
                               show (testTheories $ buildTheories
                                                $ (observe obs' ag)) ++
                              "----------------------------\n" ++
                              "using theories ...     \n" ++
                              "----------------------------\n" ++
                               show (useTheories $ testTheories
                                                 $ buildTheories
                                                 $ (observe obs' ag)) ++
                              "----------------------------\n" ++
                              (sim time' obs ag') where
                                  ag' = useTheories $ testTheories
                                                    $ buildTheories
                                                    $ (observe obs' ag)
                                  time' = time + 1
                                  obs' = (O r_aff r_obs) -- reduce the data
                                  r_aff = take no_obs (getObsAff obs)
                                  r_obs = take no_obs (getExpSerie obs)
                                  no_obs = length (getExpSerie $ getObserva-
tion ag)
```

The simulation function is initialized with a timepoint, a list of observations, an agent and returns a string that contains the result of the simulation at a later timepoint. The agent advances through different states until a time limit exceeds or no further input is given to the model. The simulation function is defined recursively and invokes the chain of functions described above.

5.1.6 Summary

In summary a framework for testing sequences of theories in an environment has been presented. The framework utilizes a single agent. The agent uses four mental operations to advance through sequences of theories triggered by perceptual input of the environment. The model stops to explore the environment in the following situations:

- The agent repetitively observes the same type of data. The observation of the environment does not yield a new theory. The `buildTheories` function just builds theories that are already in the mind of the agent. The model uses a time limit to exit this cycle. It can be interpreted as the agent being bored leaving the experiment.
- The `buildTheories` function cannot match the perceptual data to a theory. The observations of the objects in the environment do not afford any actions. No theory in the mind of the agent fits to the perceptual input data. This can be interpreted as the agent being angry or frustrated because it did not find any explanation.

In the following section the algebraic descripition of the theories and perceptual data are given.

5.2 Formal Description of Theories and Testdata

5.2.1 Theories are Classes

Theories are specified using Haskell classes. The class specifies which operations belong to a theory. A theory must have at least one expectation function in order to predict beliefs about the environment.

Classes cannot be used directly because they are abstract specifications. To execute the model implementations have to be provided for the classes. This is done via the definition of instances and data types.

A theory is described by a data type having a list of affordances. Affordances determine which theories will be chosen among the possible theories. Affordances are described by String types.

Furthermore the data type `Theory` consists of an operation that is carried out in the environment. For the chosen examples it is a binary function that maps two objects to an equivalence class. A weight, i.e. a floating type value describes the importance of the theory. The value of the weight can be between 0.0 and 1.0.

```
type Affordance     = String
type Operation      = (Object->Object->Bool)
type Weight         = Float
data Theory         = T [Affordance] Operation Weight deriving Show
type Theories       = [Theory]
```

In the Haskell code theories are specified by classes having different implementations, i.e. instances. Each instance holds at least one expectation function that is utilized to generate the agent's beliefs. In order to hold the different expectation functions within the same data structure partially initialized functions have been implemented. For each new theory a data type `Redescription` has been defined.

```
data Redescription1 = R1 deriving (Eq,Show)
data Redescription2 = R2 deriving (Eq,Show)
data Redescription3 = R3 deriving (Eq,Show)
data Redescription4 = R4 deriving (Eq,Show)
data Redescription5 = R5 deriving (Eq,Show)
class Theory c o where
    expect_fun :: r -> o -> o -> Bool
```

The expectation function `expect_fun` is implemented for each redescription (Haskell instance) that is defined for a certain theory (Haskell class). Expectation functions in the course of the thesis are: `isIn, isOn` and `isOccluded`. The data type `Redescription` is used as an abstract identifier for the theory. With partial initialization the appropriate implementation of a theory can be overloaded when necessary. Partially initialized expectation functions can be managed within the same data structure.

```
instance Theory Redescription1 Object where
    expect_fun R1 o1 o2 = 1st implementation
instance Theory Redescription2 Object where
    expect_fun R2 o1 o2 = 2nd implementation
..
-- data structure for partially initialized expectation functions
theory_list = [expect_fun Redescription1,
               expect_fun Redescription2,
               .. ]
```

5.2.2 Test data are functions

In order to test a theory in the current framework test data have to be provided. These are percepts made in the environment. The current model describes percepts as functions. The functions have been chosen in a way that they cause the model to advance through sequences of theories.

The test data for the current model have to be opposite to the stimuli used in the empirical studies described in chapter 4 of the thesis. Psychologists test violations of expectations. The expectations have been build by observations made previously. Because what the model learns depends on what the model perceived before the test data have to desribe observations that were made before the detection of the empirically tested violations.

The agent that detects a violation like an object that disappears behind a transparent occluder has to have seen before many times that an object reappeared behind a transparent occluder. In the current model that is based on dynamic weighting the number of observations pointing to reappearance have to be higher than the number of observations that point to dissappearance when object moved behind a transparent occluder.

The test data contain affordances. Affordances serve to provide the connection between perceptual data and theories. The affordance guides which percepts belong to a theory or which theory has to be chosen based on given perceptual data.

The model is object based and the test data does not consider partial relationships between objects. In the test data an object is fully hidden or hidden not at all. All objects treated in the model are in a table top space. The observations of the objects are perfect, i.e. free of any erroneous influences.

The subsequent sections introduce formal theories for the occlusion, containment and support of objects. In order to carry out a simulation with a theory driven agent test data were created that caused changes in the theory sequences. The agent advances from one theory to the other. The description of the theories will always be followed by a description of the test data.

5.3 Occlusion of Solids

5.3.1 Theories

Empirical data that were described in section 4.2 served to build the available Haskell implementation. The theory sequence for the occlusion of objects shows how a theory is stepwise constrained by adding more and more axioms. The number of possible test cases that fit to the world get more and more small.

The class occluder provides a specification for an expectation function `isOccluded` that permits to predict whether an object is occluded by another object given a set of percepts. The outcome of the operation is mapped on a Bool variable. The first instantiation of the theory considers just percepts about an object being behind another.

```
class Occluders r o where
    isOccluded :: r -> o -> o -> Bool
-- 1st Theory: An object behind another is occluded.
instance Occluders Redescription1 Object where
    isOccluded R1 o1 occ = isBehind o1 occ
```

A second instantiation constrains the theory by adding an axiom. Beside the spatial relation behind, a window in the occluding object is considered in the theory. The object is occluded by the occluder when there is no window in the occluder.

```
-- 2nd Theory: Occluders with windows do not work.
instance Occluders Redescription2 Object where
    isOccluded R2 o1 occ = isOccluded R1 o1 occ &&
                           (not $ hasWindow occ)
```

The third instantiation considers the size of the objects. The object behind the windowless occluder has to be smaller than the occluder in width and height in order to be fully hidden. The theory is again spezialized by adding an axiom.

```
-- 3rd Theory: The size of the occluded object matters.
instance Occluders Redescription3 Object where
    isOccluded R3 o1 occ = isOccluded R2 o1 occ &&
                           o1 < occ
```

Transparency creates a new general theory and splits the theory of occlusion into two special theories. Objects that are not transparent are a special case of a more general occlusion theory (`Redescription3`) that distinguishes between transparent and non transparent occluders. The second special theory is triggered by an object that has been observed behind a transparent occluder. It will always be visible no matter if the occluder has a window or is smaller than the object.

```
-- 4th Theory: A transparent occcluder does not hide the occluded object.
instance Occluders Redescription4 Object where
    isOccluded R4 o1 occ = isOccluded R3 o1 occ &&
                           isTransparent’ R2 occ

-- ------------------------------------------------------------------------
--                  GENERALIZING TO A NEW THEORY OF TRANSPARENCY
-- ------------------------------------------------------------------------
class Transparency r o where
        isTransparent :: r -> o -> Bool

-- Theory of Transparency
instance Transparency Redescription1 Object where
        isTransparent’ R1 o1 = isTransparent o1
-- Theory of nonTransparency
instance Transparency Redescription2 Object where
        isTransparent’ R2 o1 = not $ isTransparent o1
```

Finally movement also constrains occlusion. Moving objects are sometimes occluders and sometimes they are not. If the occluder is moved or the occluder and the object do not share the same location the object behind the occluder is not hidden.

```
instance Occluders Redescription5 Object where
    isOccluded R5 o1 occ = isOccluded R4 o1 occ &&
                           (not $ (moved occ && (not $ samePos o1 occ)))
```

5.3.2 Test Data

The test data have been created so that they cause a conflict in the agent’s mental model. The first test data set for the occlusion of a solid object consists of an object behind another second object. The objects are treated as integers in the implementation. Objects move behind others. They disappear before the occluder and reappears unchanged on the other side of the occluder. This will be in accordance with a theory that considers just the spatial relation behind. But observations of objects that reappear in gaps between occluders will elicit a new specialized theory that also considers gaps and windows. This new theory may be contradicted by the following test cases illustrated in table 5.1.

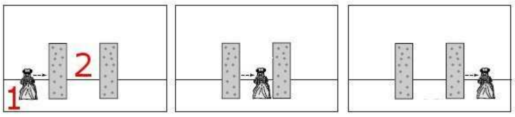
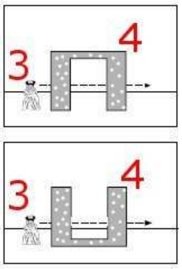
redescription	o1	o2	conflict	picture
R1	1	2	behind	 expectation: behind o1 o2 == True
R2	3	4	window	 expectation: hasWindow o2 == False

Table 5.1: Theory Sequence and test data for the occlusion of solids

The first example shows an object 1 that moves behind an object 2 being shown in the gap between the objects (table 5.1 above). The second type of observation is illustrated in table 5.1 below. Object 3 is moved behind an occluder 4 with a window, being shown in the window. Both cases can not be predicted with a theory that considers just the spatial relation behind. As windows reveal the objects that are behind a new theory is suggested that is constrained by the spatial relation behind and the presence of windows.

```
e_occlusion = [(1,2,True)]
e_occlusion1 = [(3,4,False)] -- contradict R1, conflict window
isBehind 1 2 = True
hasWindow 2  = False
isTransparent 2 = False
moved 2 = False
```

```
samePos 1 2 = True
```

The new theory that considers the spatial relation behind and the presence of gaps and windows will be revised with the detection of size. Objects that are bigger than the occluding objects will be seen and objects that are smaller than the occluder will be hidden. Table 5.2 shows above an object 6 that is hidden behind a smaller object 5. The size of the integer value has been utilized as the size of the object. More observations will point to smaller objects being hidden behind bigger, windowless occluders.

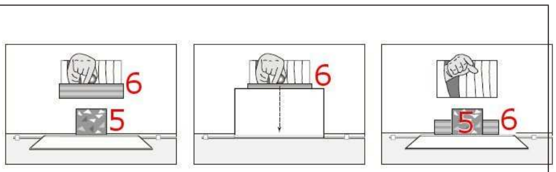
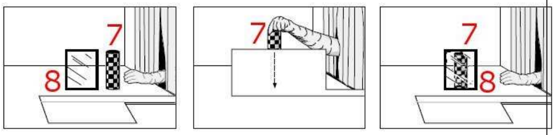
redescription	**o1**	**o2**	**conflict**	**picture**
R3	6	5	size	 expectation - width: o1 < o2 == True
R4	7	8	transparency	 expectation: isTransparent o2 == False

Table 5.2: Theory Sequence and test data for the occlusion of solids

The naive occlusion theory will again be revised when tested with data shown in table 5.2 below. The transparent object 8 will not occlude object 7. The anticipation function of the occlusion theory will be in conflict with the illustrated test case.

5.4 Containment of Solids

5.4.1 Theories

The empirical studies in section 4.3 point to a simple theory for the containment of solid objects that depends on the existence of an opening in the container object. The algebraic description below defines an expectation function `isIn` considering that fact. A second theory considers the shared movement of the container and the contained object in its expectation function. In another specialization step the theory advances to considering the size of the container and the contained object.

```
class Containment r o where
    isIn  :: r -> o -> o -> Bool
-- 1st Theory: the container has to be open.
instance Containment Redescription1 Object where
    isIn R1 obj co = isOpen co
-- 2nd Theory: The object in the container shares the movement with the con-
tainer
instance Containment Redescription2 Object where
    isIn R2 obj co = isIn R1 obj co &&
                     ((not $ moved co) || (moved co && samePosAfter obj co))
-- 3rd Theory: The contained object has to be smaller.
instance Containment Redescription3 Object where
    isIn R3 obj co = isIn R2 obj co &&
                     (obj < co)
```

The perception of movement of the contained object inside the container leads to a generalization of the theory. The according algebras are described below where `hasOpening (object)` is a perception that determines whether the object has an opening, the ’`<`’ operator compares the size of the objects and `fit` determines whether the contained object can be moved inside the container.

The empirical data in section 4.3 point to a generalization step in the theory of object containment. The distinction of loose-fit and tight-fit containers requires to introduce new data types, classes and instances for the generalized theories. The Haskell code below illustrates the specialization step.

```
-- Theory of Loose-Fit Containment
instance Containment Redescription4 Object where
    isIn R4 obj co = isIn R3 obj co &&
                     fit R1 obj co
-- Theory of Tight-fit Containment
instance Containment Redescription5 Object where
    isIn R5 obj co = isIn R3 obj co &&
                    fit R2 obj co
-- ------------------------------------------------------------------------
--                          GENERALIZING TO A NEW THEORY
-- ------------------------------------------------------------------------
-- Loose and tight fit containment require Fit as generalized super class
class Fit r o where
    fit :: r -> o -> o -> Bool
instance Fit Redescription1 Object where
   fit R1 o1 o2 = movable o1 o2
instance Fit Redescription2 Object where
   fit R2 o1 o2 = not $ movable o1 o2
```

Observation will lead to cases where the new axiom `fit` will be both verified and falsified. Thus the theory for loose-fit container and the theory for tight-fit container will receive alternating evidence dependent on observations in the environment. The general theory of containment (`Redescription 3`) and the new general theory of movability will receive support with each observation.

5.4.2 Test Data

In order to test the sequence of containment theories five types of observation lists have been created, each creating a conflict that will lead to theory adaptation. As the notation is abstract two tables illustrate the test cases. Integers have again been used as a representation for solid objects. The model observes and predicts the outcome of the visualized operations. The pictures show the violations that should be detected by the model using the newly acquired theory.

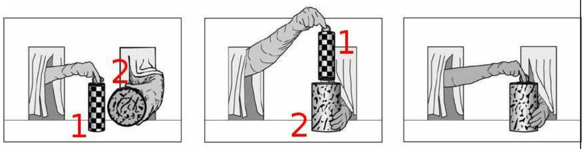
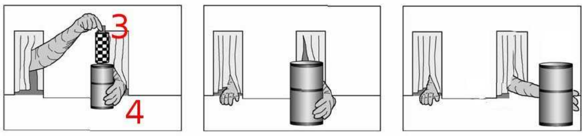
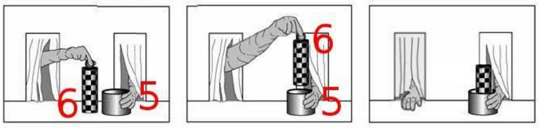
redescription	**o1**	**o2**	**conflict**	**picture**
R1	1	2	opening	 expectation: isOpen o2 == True
R2	3	4	movement	 expectation: moved o2 && samePosAfter o1 o2
R3	6	5	size	 expectation: o1 < o2

Table 5.3: Theory Sequence and test data for the containment of solids

Frequent observations of objects that are put into a container will lead to a theory that points to an opening of the container. Solid object 2 has an opening (`isOpen 2 == True`) and the observation of the objects 1 and 2 leads to the anticipation that 1 is in 2 which is in accordance with the observed outcome of the operation.

Solid objects in containers share the movement with their container. This observation will lead to a theory that does not only consider the opening but also the shared movement of the objects. Although object 4 has an opening it does not share the movement with object 3, the testcase contradicts the agent's theory of containment. Therefore object 4 is not a container or the theory of containers needs to be adapted. The agent percepts are described by a set of functions such as `isOpen o, moved o, samePosAfter o1 o2, movable o1 o2`.

```
e_containment = [(1,2,True)]
e_containment1 = [(3,4,False)] -- contradict R1, conflict movement
e_containment2 = [(6,5,False)] -- contradict R2, conflict size
isOpen 2 = True
moved _ = False
samePosAfter 1 2 = True
movable 1 2 = True
```

It is not sufficient to specify a container by an opening and a shared movement. An object that is bigger than the container will not fit into the container. The test case in table 5.3 illustrates on the bottom the observation that leads to a theory revision. Object 6 will not fit into object 5 and most often the model will be confronted with observations of the form `(6,5,False)`. Until the model does not hold a theory that considers size as an important parameter for the containment of objects it will not detect the violation shown in the figure that represents a case of the observation`(6,5,True)`.

redescription	**o1**	**o2**	**conflict**	**picture**
R4	7	8	loose-fit	 expectation: moveableInside == True
R5	9	10	tight-fit	 expectation: moveableInside == False

Table 5.4: Theory Sequence and test data for the containment of solids

The movability inside the container is a percept that leads to theory generalization. The perceptual data are extended to consider movement inside the container via the function `movableInside o1 o2`. Both observational data will be provided to the model cases of observations `(7,8,True)` and cases of observations `(9,10,False)`.

5.5 Support of Solids

5.5.1 Theories

The Redescription for the support of solids is implemented as a sequence of theories that goes through four specialization steps. The initial theory for object support relies only on a very generic definition of contact. I discuss the code below and explain the functions and introduced data types.

```
class Supporters r o where
    isOn :: r -> o -> o -> Bool
-- 1st Theory considers just the contact between the objects
instance Supporters Redescription1 Object where
        isOn R1 o1 o2 = hasContact o1 o2
```

In a specialization step the theory of a generic contact will be transformed. A new instance will be added that considers solids that touch via their top surface implementing the `hasTopContact o1 o2` operation. It defines the type of contact observed between two objects. Possible outcomes are the values `True` (objects touch via top/bottom surface) and `False` (objects touch in another way, e.g. on their side surfaces).

```
-- 2nd Theory specifies the type of contact
instance Supporters Redescription2 Object where
        isOn R2 o1 o2 = isOn R1 o1 o2 &&
                        hasTopContact o1 o2
```

Another specialization step transforms the theory for support of solids. The new theory also considers the amount of contact. A new instance will be added the considers not only objects that touch via their top surface but also the amount of contact between the objects. The operation `getAmountContact o1 o2` is a perception that determines the amount of contact between two solids in the environment.

```
-- 3rd Theory considers the amount of contact
instance Supporters Redescription3 Object where
        isOn R3 o1 o2 = isOn R2 &&
                        getAmountContact o1 o2
-- 4th Theory considers the shape of the supported object
instance Supporters Redescription4 Object where
        isOn R4 o1 o2 = isOn R3 &&
                        getShape o1 o2
```

The 4th theory is further constrained by the shape of the supported object. A distinction between supportive and non supportive shapes leads again to theory specialization. A new instance is added that considers the shape of the solid. The `getShape o1 o2` operation distinguishes between supportive and unsupportive shapes in the environment.

5.5.2 Test Data

The test data introduced in this subsection will elicit sequences of occlusion theories in the model. An object that loses contact with it's supporter will not be supported. Solid objects are described as integer values. The object 1 that loses contact with object 2 and falls down is observed as `(1,2,True)` by the model.

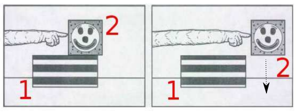
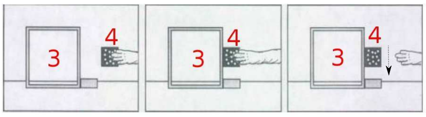
rede-scription	**o1**	**o2**	**conflict**	**picture**
R1	1	2	contact	 expectation: contact o1 o2 == True
R2	3	4	type	 hasTopcontact o1 o2 == True

Table 5.5: Theory Sequence and test data for the support of solids

Arbitrary contact will not confirm the theory of object support. Observations that distinguish side from top contact such as object 4 not being supported by object 3 will lead to theory revision. The code sample below illustrates two different experiment series and the according perceptual data.

```
e_support = [(1,2,True)]   -- contradict, conflict contact
e_support1 = [(3,4,False)] -- contradict R1, conflict side contact
```

```
hasContact 1 2 = True
hasContact 3 4 = True
hasContact _ _ = False
hasTopContact 3 4 = True
hasContact _ _ = False
getAmountContact 5 6   = True
getAmountContact _ _   = False
getShape 7 8   = True
getShape _ _   = False
```

Further observations will stress the amount of contact, e.g. object 6 lying on object 5. Frequent observations of object 6 falling down of object 5 (`(5,6,False)`) will elicit theory change towards a specialized theory of object support that considers the amount of contact. The figures below show the violation that should be detected after the new theory has been acquired.

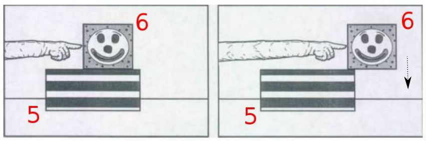
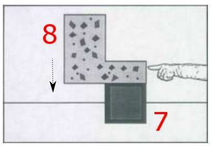
redescription	**o1**	**o2**	**conflict**	**picture**
R3	5	6	amount	 hasSupportiveAmountContact o1 o2 == True
R4	7	8	shape	 hasSupportiveShape o2 == True

Table 5.6: Theory Sequence and test data for the support of solids

On the bottom of table 5.6 the test data for the theory revision towards a theory of object support that considers shape can be found. Object 8 is supported by object 7. Observations of the form `(7,8,True)` will raise the evidence into a theory that considers the shape of the supported object.

5.6 Summary

The abstract model of an algebra based agent that advances through sequences of ever improving spatial theories in an environment has been presented. The implementation of three mechanisms has been suggested to change algebraic Re-descriptions based on observations in an environment. Each mechanism has been explained in analogy to a study carried out in developmental psychology. Starting at the formalization of theories and test data I continue to describe a simple theory driven agent in an environment to carry out simulations. The presented framework has these properties:

- The model builds sequences of perpetually improving spatial theories based on perceptual input.
- The model depends on the perceptual input. The observation of a contradicting outcome of an operation in the environment with the one predicted by the theories held in the agent's knowledge base leads to theory change.
- Changed theories are firstly retested with older observations before theories are used with new data.
- The model makes errors of commission and omission as tested by Baillargeon (2004a) and Luo and Baillargeon (2005). I illustrate this with an example. The model is in a conflict when
  - an occlusion theory *without* the transparency constraint has been developed and an object appears behind a transparent occluder (error of omission). The influence of transparency is omitted with the occlusion theory. The omission of transparency makes the agent belief that objects disappear when moving behind a transparent occluder.
  - an occlusion theory *with* the transparency constraint has been developed and a solid disappears behind a transparent occluder (error of commission). The commitment to transparency as an influencing factor of a occlusion theory causes the model conflict with the predicted belief. The commitment to transparency makes the agent belief that objects cannot disappear behind transparent occluders.
- Equal or similar weights point to coexisting theories and theory generalization.

- The model is domain specific - support, occlusion and containment are developed in separate sequences.

Theories are human interpretations of given data (Kuhn, 1962, p. 137). The interpretation of these data help to articulate a theory but do not lead to theory change. Children that frequently observe the same type of problem will not develop new knowledge without having given the possibility to experiment or additional evidence (Siegler, 1976). The test data that have been illustrated in this chapter will be used in a simulation of empirical studies described in chapter 4. I am going to show qualitatively that the model is in analogy to empirical studies by carrying out the simulations. The created test data raise conflict situations in order to keep the agent developing a sequence of theories.

Chapter 6

Simulation of Theory Sequences

The following computational model is based on the studies presented in chapter 4 of the thesis and has been implemented with the functional programming language Haskell. In order to carry out a simulation a theory driven agent has been defined in chapter 5. The agent is a wrapper to an algebraic theory change mechanism. The mechanism compares observations of the outcome of operations in an environment with anticipations about the results of operations. The anticipations are generated out of the theories the agent holds at the time of observation. They are the beliefs of the agent that will guide the conceptual change. Whenever anticipation and observation do not fit the agent will try to build a new theory. The major point in this chapter is that frequent mismatches between observations and expectations will elicit changes in the algebraic structure of spatial theories. The model is in accordance with empirical data described in chapter 4.

6.1 The Sandbox

In a simulation empirical studies of developmental psychology with an agent in an environment are carried out in analogy to an infant playing in a sandbox. The simulations show that dependent on the perceptual input (like toys in a sandbox) different instantiations of the cognizing agents (different children), having different initial knowledge come to build different spatial theories. The spatial theories differ in the level of details, considering different number of percepts. The general agent's behavior is in accordance with infant's behavior in empirical studies carried out in developmental psychology.

- The agent will leave the experiment if there is no variation in the experiment or nothing new can be reasoned out of the given test data (agent state: GiveUp).
- An agent that is initialized without any test data will request test data, noting being unable to test it's concepts (program error).
- An agent that is always confronted with the same test data will not be able to advance to a new concept (agent loops until time limit exceeds).
- A variation in the percepts will lead to new conflicts that elicit the adaptation of concepts. The adaptation is based on a dynamic weighting mechanism.

In the following three sections simulations for different theory sequences are carried out. The simulated sequences will be compared to the outcome of empirical studies. Further examples can be found in the appendix.

6.2 Occlusion of Solids

The acquisition of a sequence of theories for the occlusion of solids has been simulated. The empirical studies have been described in section 4.2 and the fomalization of the theories in section 5.3. The simulated results of the model are in accordance with the findings of Hespos and Baillargeon (2001b),Baillargeon (2004b) and Luo and Baillargeon (2005).

Different agents are instantiated with test data in order to carry out a simulation. Agents that are confronted with the same data over a long period of time will stop acquisition of new concepts and leave the experiment. The variation in the test data will lead the agents to change their theories.

```
ag1 :: Agent
ag1 = Agent 1
            (O [] [])                 -- no observations
            [Occlusion.theories !! 0]  -- holds unweighted theories
            Occlusion.theories         -- potentially possible theories
            Observe                    -- state: willing to explore
```

The piece of code below illustrates an agent that observes the environment. The agent receives an observation of the type `[1,2,True]`. The observation has to be read as solid 1 occludes solid 2. The truth value indicates that solid 1 is hidden by the occluder solid 2.

```
Loading package haskell98-1.0 ... linking ... done.
"Time: "1
-------------------------------
observing environment ...
-------------------------------
...............................
Agent id: 1
Detected percepts:"isBehind"
Observations: [(1,2,True)]
Current theories: T 1 ["isBehind"] Function 0.0
State: Test
...............................
```

The first instantation of the agent holds a simple theory about the occlusion of solids, this can be seen in the status line “Current theories”. The theory has not been tested before therefore the weight has been set to `0.0`. The agent received an observation and has a theory about occlusion. Therefore the agent changes it’s state to `Test` (see output above).

```
-------------------------------
testing theories ...
-------------------------------
...............................
Agent id: 1
Detected percepts:"isBehind"
Observations: [(1,2,True)]
Current theories: T 1 ["isBehind"] Function 1.0
State: Use
...............................
```

The theory predicts a solid to be hidden by another if it is behind the other solid. The observed test cases - in the sample code above there is just one - can be explained by the current theory, the weight is set to `1.0`. The agent changes it’s state to `Use`.

```
-------------------------------
using theories ...
-------------------------------
...............................
Agent id: 1
Detected percepts:"isBehind"
```

```
Observations: [(1,2,True)]
Current theories: T 1 ["isBehind"] Function 1.0
State: Observe
..............................
```

No conflict appeared in the observation cycle and no new theory has been built. The agent changes it's state to `Observe` in order to acquire new test data for the current theories.

```
------------------------------
"Time: "2
------------------------------
observing environment ...
------------------------------
..............................
Agent id: 1
Detected percepts:"isBehind", "hasWindow"
Observations: [(1,2,True),(3,4,False)]
Current theories: T 1 ["isBehind"] Function 1.0
State: Test
..............................
```

The invocation of the observe function leads to new test data. In time cycle 2 the agent observes a solid 3 being behind a solid 4. Solid 4 has a window. The old observation data stay in the memory of the agent. As new test data arrived the agent changes its state to `Test`.

```
------------------------------
testing theories ...
------------------------------
..............................
Agent id: 1
Detected percepts:"isBehind", "hasWindow"
Observations: [(1,2,True),(3,4,False)]
Current theories: T 1 ["isBehind"] Function 0.5
State: Build
..............................
```

The agent still holds a theory that considers an object to be occluded if it is behind the occluder. The expectation that the solid 3 is hidden by solid 4 cannot be generated by the theory `T1`. The weight of the theory is set to `0.5`, meaning the agent can explain half of the observed phenomena with the current theory. Therefore the agent changes it's state to `Build`.

```
------------------------------
"Time: "3
```

```
------------------------------
observing environment ...
------------------------------
..............................
Agent id: 1
Detected percepts:"isBehind", "hasWindow"
Observations: [(1,2,True),(3,4,False)]
Current theories: T 1 ["isBehind"] Function 0.5
State: Build
..............................
```

Now the agent chooses an occlusion theory among all possible theories that considers the perceptual influence of windows. The agent specializes the theory by adding an axiom to the previous theory, holding now the old theory `T1` tested with a weight of `0.5` and the new theory `T2`. `T2` is still untested and therefore holds the weight 0.0. The agent therefore changes it's state to `Test`.

```
------------------------------
building theories ...
------------------------------
..............................
Agent id: 1
Detected percepts:"isBehind", "hasWindow"
Observations: [(1,2,True),(3,4,False)]
Current theories: T 1 ["isBehind"] Function 0.5
T 2 ["isBehind","hasWindow"] Function 0.0
State: Test
..............................
```

In the test phase the agent detects that the new theory cannot only explain the new test data (3,4,False) but also the old observation of the type (1,2,True). The new theory receives more positive evidence. The theory can predict all observed cases, therefore its weight is set to one. The old theory remains in the agent's memory being less important with a weight of `0.5`. The agent holds a theory that sufficiently explains its surroundings and therefore set it's state to `Use`.

```
------------------------------
testing theories ...
------------------------------
..............................
Agent id: 1
Detected percepts:"isBehind"
"hasWindow"
Observations: [(1,2,True),(3,4,False)]
Current theories: T 1 ["isBehind"] Function 0.5
T 2 ["isBehind","hasWindow"] Function 1.0
```

```
State: Use
...............................
```

The observation of the environment and the detection of new percepts lead to further theories. As long as a theory predicts sufficiently the agent will set it's state from `Use` to `Observe` in order to acquire new test data. The agent actively tries to find data that could falsify it's theories.

```
-------------------------------
using theories ...
-------------------------------
...............................
Agent id: 1
Detected percepts:"isBehind", "hasWindow"
Observations: [(1,2,True),(3,4,False)]
Current theories: T 1 ["isBehind"] Function 0.5
T 2 ["isBehind","hasWindow"] Function 1.0
State: Observe
...............................
```

After seven iterations of the simulation the agent detects - based on the given perceptual input - an occlusion theory that considers the spatial relation behind, windows in the occluder, and the size of the hidden object. The agent stops to observe the environment as no new perceptual data are entered.

```
-------------------------------
"Time: "7
-------------------------------
...
-------------------------------
using theories ...
-------------------------------
...............................
Agent id: 1
Detected percepts: "isBehind", "hasWindow", "size",
Observations: [(1,2,True),(3,4,False),(6,5,False)]
Current theories: T 1 ["isBehind"] Function 0.33
T 2 ["isBehind","hasWindow"] Function 0.67
T 3 ["isBehind","hasWindow","size"] Function 1.0
State: Observe
...............................
-------------------------------
Agent getting bored - no new data
CHECK Simulation time limit
```

The simulation is in accordance with the study carried out by Hespos and Baillargeon (2001b),Baillargeon (2004b) and Luo and Baillargeon (2005). The behavior of the agent in the experiment can be simulated in analogy to the tested children. Therefore it can be assumed that the model is plausible.

6.3 Containment of Solids

In order to simulate the development of a containment concept studies by Hespos and Baillargeon (2001a,b) have been formalized and simulated. The studies have been described in section 4.3 and their formalization in 5.4. An agent has been initialized with a simple theory of containment that considers an object to be inside a container if the agent has observed that the object was put in another object with an opening. Agents that are confronted with the same data over a long period of time will stop acquisition of new concepts and leave the experiment. The variation in the test data will lead the agent to change it's concept.

```
ag3 :: Agent
ag3 = Agent 3
            (O [] [])                    -- no observations
            [Containment.theories !! 0]  -- holds unweighted theories
            Containment.theories         -- potentially possible theories
            Observe                      -- state: willing to explore
```

The agent observes that an object is put into another and that it is true that solid 1 is inside solid 2. This knowledge is expressed by the observation status line "Observations: [1,2,True]". Solid has an opening that is detected by the agent.

```
Loading package haskell98-1.0 ... linking ... done.
"Time: "1
------------------------------
observing environment ...
------------------------------
..............................
Agent id: 3
Detected percepts:"isOpen"
Observations: [(1,2,True)]
Current theories: T 21 ["isOpen"] Function 0.0
State: Test
..............................
```

The agent holds an untested theory about the containment of objects. The weight of the theory is `0.0`. Therefore the agent sets it's state to `Test`.

```
------------------------------
testing theories ...
------------------------------
```

```
...............................
Agent id: 3
Detected percepts:"isOpen"
Observations: [(1,2,True)]
Current theories: T 21 ["isOpen"] Function 1.0
State: Use
...............................
```

The theory can predict the outcome of the observed operation. The weight of the theory is therefore calculated with 1.0. The agent holds a theory that can sufficiently explain its environment and changes it's state to `Use`.

```
-------------------------------
using theories ...
-------------------------------
...............................
Agent id: 3
Detected percepts:"isOpen"
Observations: [(1,2,True)]
Current theories: T 21 ["isOpen"] Function 1.0
State: Observe
...............................
```

In the same way as in the occlusion experiments the agent continues to observe the environment. The detection of new percepts leads to new theories.

```
-------------------------------
"Time: "8
-------------------------------
observing environment ...
-------------------------------
...............................
Agent id: 3
Detected percepts:"isOpen", "size", "movement", "loosefit", "tightfit"
Observations: [(1,2,True),(4,3,False),(5,6,False),(7,8,False),(9,10,True)]
Current theories: T 21 ["isOpen"] Function 0.4
T 22 ["isOpen","size"] Function 0.6
T 23 ["isOpen","size","movement"] Function 0.8
T 24 ["isOpen","size","movement","loosefit"] Function 0.8
T 25 ["isOpen","size","movement","loosefit","tightfit"] Function 0.6
State: Build
...............................
```

The agent continues to observe it's environment. The code sample above shows an agent after the eighth iteration step. The agent holds five different types of observations in its memory and dected a set of five theories. Opposite to the example of the occlusion sequence the agent does not hold a theory that fully explains all containment events at this point of time. This can be seen in the code example as no theory has a weight of `1.0`.

```
-------------------------------
building theories ...
-------------------------------
...............................
Agent id: 3
Detected percepts:"isOpen", "size", "movement", "loosefit", "tightfit"
Observations: [(1,2,True),(4,3,False),(5,6,False),(7,8,False),(9,10,True)]
Current theories: T 21 ["isOpen"] Function 0.4
T 22 ["isOpen","size"] Function 0.6
T 23 ["isOpen","size","movement"] Function 0.8
T 24 ["isOpen","size","movement","loosefit"] Function 0.8
T 25 ["isOpen","size","movement","loosefit","tightfit"] Function 0.6
State: GiveUp
...............................
```

Theory `T23` explains containment based on axioms that consider the opening of the container, size of the involved objects and the shared movement of container and object inside. However observation points to solids that can contain an object being movable inside and solids containing another solid not being movable inside. The agent stops testing the theory because no new theory can be built out of the given percepts.

```
-------------------------------
testing theories ...
-------------------------------
...............................
Agent id: 3
Detected percepts:"isOpen", "size", "movement", "loosefit", "tightfit"
Observations: [(1,2,True),(4,3,False),(5,6,False),(7,8,False),(9,10,True),(9,10,True)]
Current theories: T 21 ["isOpen"] Function 0.4
T 22 ["isOpen","size"] Function 0.6
T 23 ["isOpen","size","movement"] Function 0.8
T 24 ["isOpen","size","movement","loosefit"] Function 0.6
T 25 ["isOpen","size","movement","loosefit","tightfit"] Function 0.6
State: GiveUp
...............................
-------------------------------
using theories ...
-------------------------------
```

```
*** Exception: No percepts or theories found that fit the data.
*NewSim>
```

The code example above shows that a generalization step is necessary. The general theory of containment T23 received the highest weight. To distinguish between loose-fit and tight-fit containment it was necessary to build a new general theory about movable objects (see chapter 5). As theory building is not automated, new code would have to be added and then the simulation restarted. Automation however is possible as the weights indicate which general theories have to be built. The more percepts the agent receives the stronger will be the general theories (T23), the specialized theories (T24,T25) will get lower weights but still will grow.

The simulation is in accordance with studies carried out by Hespos and Baillargeon (2001a,b). The behavior of the agent in the experiment can be simulated in analogy to the children tested. One can therefore assume that the model is plausible.

6.4 Support of Solids

A sequence of support theories has been described and formally modelled (see section 4.4 and 5.5). The model is based on studies carried out by Baillargeon (1994, 2004b). An agent has been initialized with a simple theory of object support that just considers the contact between two solids.

```
ag2 :: Agent
ag2 = Agent 2
          (O [] [])                -- no observations
          [Support.theories !! 0]  -- holds unweighted theories
          Support.theories         -- potentially possible theories
          Observe                  -- state: willing to explore
```

As in the two sequences before the agent cycles through the observe-build-test-use functions to build theories about it's environment. When a theory explains all given data the agent seeks for new test data and percepts that falsify the current theories. The detection of new percepts and contradicting test cases lead to the choice of new theories.

```
Loading package haskell98-1.0 ... linking ... done.
"Time: "1
------------------------------
```

```
observing environment ...
-------------------------------
...............................
Agent id: 2
Detected percepts:"Contact"
Observations: [(1,2,True)]
Current theories: T 10 ["Contact"] Function 0.0
State: Test
...............................
-------------------------------
testing theories ...
-------------------------------
...............................
Agent id: 2
Detected percepts:"Contact"
Observations: [(1,2,True)]
Current theories: T 10 ["Contact"] Function 1.0
State: Use
...............................
-------------------------------
using theories ...
-------------------------------
...............................
Agent id: 2
Detected percepts:"Contact"
Observations: [(1,2,True)]
Current theories: T 10 ["Contact"] Function 1.0
State: Observe
...............................
-------------------------------
"Time: "2
-------------------------------
observing environment ...
-------------------------------
...............................
Agent id: 2
Detected percepts:"Contact"
"TopContact"
Observations: [(1,2,True),(3,4,False)]
Current theories: T 10 ["Contact"] Function 1.0
State: Test
...............................
```

After 11 iteration cycles the agent stops observing the environment. The agent holds a theory that can predict the support of a solid through another by considering the perceptual parameters type of contact, amount of contact, and shape of the involved solids. As no new contradictions occur the agent gets bored as he holds a theory that sufficiently explains the environment.

```
------------------------------
"Time: "11
------------------------------
...
------------------------------
using theories ...
------------------------------
...............................
Agent id: 2
Detected percepts:"Contact", "TopContact", "AmountContact", "Shape"
Observations: [(1,2,True),(3,4,False),(5,6,False),(7,8,False),(9,10,True)]
Current theories: T 10 ["Contact"] Function 0.4
T 11 ["Contact","TopContact"] Function 0.6
T 12 ["Contact","TopContact","AmountContact"] Function 0.8
T 13 ["Contact","TopContact","AmountContact","Shape"] Function 1.0
State: Observe
...............................
------------------------------
Agent getting bored - no new data
 CHECK Simulation time limit
*NewSim>
```

The simulation is in accordance with the study carried out by Baillargeon (1994). The behavior of the agent in the experiment can be simulated in analogy to the tested children. Therefore it can be assumed that the model is plausible.

6.5 Summary

Sandbox Geography is a formal theory for the acquisition and adaption of spatial concepts. In the simulation of empirical studies I could qualitatively show that the formalized model conforms to the empirical data described in chapter 4 of the thesis. I provided the implementation of an agent as a wrapper to a cognitive changing mechanism for spatial theories. The mechanism evaluates theories by comparing observations with generated anticipations. The comparison of observations and expectations is based on counting success and failure of operations carried out in an environment.

The model considers that an agent would leave the experiment when new instances do not occur. Because of conflict cases the agent must adapt it's concepts. The acquisition of the new concept has to be done manually by the researcher. The simulated data point to the theory constructing operations of specialization and generalization.

Chapter 7

Conclusions and Future Work

> "... disconfirmation can be decisive, but confirmation is just an invitation for further investigation."[1]

This chapter summarizes the present thesis. It concludes the methodology used to design a spatial cognizing agent in a sandbox world. Important findings are discussed and analyzed. Future research directions are identified providing a number of new questions.

7.1 Summary

This research formally shows how observations of an environment lead to a new theory based on a set of empirical studies. Studies have been chosen that describe spatial aspects of the world as sequences of naive theories. The spatial theories have been modeled using algebraic specifications.

The thesis started with a notion about theory, giving different viewpoints and reviewing the literature. The *theory theory* of cognitive development was introduced that states that the world can be explained by a set of theories that changes in the light of new evidence. With the tools of observation, prediction and adaptation theories are manipulated by the growing infant. The research was connected to the history of science and naive theories. Chapter 2 closed with a review of computational models for cognitive development to position the present

[1] Marcus (2001)

interdisciplinary approach to other available models in cognitive science and to consider recent findings in the proposed model.

Chapter 3 started with a discussion of research approaches towards a model that contributes to naive geography. The chapter outlined the characteristics of a model of a theory driven agent that learns spatial concepts in an environment. Affordances, rationality and feedback were discussed as a means to evaluate a theory in an environment. The sequence of theories is dependent on the innate equipment of the agent, the learning mechanism, and the cues given by the environment. The novelty of this approach lies in the use of prelinguistic empirical data to build formal models about the acquisition of image schemata.

In chapter 4 empirical studies that describe sequences of theories for the occlusion, support and containment of objects were reviewed. The general setup of the empirical studies used was explained and the results and interpretations were summarized. Mechanisms to build sequences of theories were derived from identifying regularities in the empirical data. In the modeling process three types of theory changing mechanisms have been found:

1. **Specialization** considers a new influential parameter. A theory is specialized by constraining it through an axiom. The axioms constrain the theory to a special set of sorts and operations. The more axioms are added the fewer sorts can be described by the specialized theory.

2. **Generalization** is an abstraction step. A theory is found to be a special case of a more general theory. A number of other special theories may exist that can be derived from the newly created generalized theory. The coexisting theories are special cases of the generalized theory.

3. **Dynamic Weighting** is a mechanism to assign importance to a theory. Theories with higher weights are favored to those with lower weights. Belief revision is the result of the dynamic weighting mechanism that assigns a higher weight to a previously low weighted theory.

Empirical studies on sequences of theories for the equality, occlusion, support and containment of objects were formally modeled in chapter 5. A mechanism for the evaluation of spatial theories in an environment based on a weighting mechanism was presented. In a detect-build-test-use cycle the agent evaluated theories in an environment and reasoned about their accordance. Theories were not built automatically by the agent. The agent chose among possible theories. In future work this step will be replaced by an inferencing mechanism. Different instantiations of the agent stand for differently developed theories

In chapter 6 the hypothesis was verified that qualitatively new spatial representations can be gained by the adaptation of axioms in a formal model based on algebraic specifications. Simulations were carried out using the agent based model. Observations were created according to the empirical studies of developmental psychology and tested with the model. The model's behavior was compared to the behavior of infants in empirical studies. The simulated outcomes were in accordance with the empirical data.

7.2 Results and major findings

The present thesis discusses the acquisition of spatial concepts based on empirical studies of developmental psychology. Space has a certain primacy in our lives as its understanding is crucial to survive. On a daily basis we have to find objects in space and find our ways through different environments. The acquisition of spatial concepts therefore starts at birth, i.e. the moment we are set in an environment having some innate concepts.

The major result of this research is a framework for the acquisition of spatial theories in an environment that starts with some innate theories. A theory driven agent advances through sequences of spatial theories based on perceptual input. The model is based on algebraic specifications. Three mechanisms of conceptual changes have been classified.

Theories can be constrained by adding axioms to more specialized theories. Theories can be abstracted to more general theories by deleting axioms. Theories can be evaluated by a dynamic weighting mechanism that compares observations with predictions of the outcome of operations in the environment. The proposed framework shows the following properties. It is

1. modular: A conceptualization of the world is built using algebras. Each algebra is derived from an empirical study of cognitive development. Initial algebras have been defined to describe the world.

2. dynamic: Theories are expressed by algebras. Theory revision is based on three mechanisms. They change the behavior of the model and thus the spatial conceptualizations of the agent in the environment.

3. action-driven: The observations and predictions of the outcome of actions[2] in an environment lead to the formation of new theories. The theories held by the agent serve to anticipate the outcome of the actions. A comparison of anticipation with observation leads the agent to evaluate it's concepts by further observation and if necessary by adapting them.

The proposed model for theory change is based on the beliefs and observations of the agent. The agent's model has been derived from infant's expectations about table top objects in empirical studies. The *theory theory* of cognitive development suggests that children's mechanism of theory revision is also used by adults. The present model therefore does not only apply to children but also to adults.

The formal framework about people's beliefs of space is a contribution to naive geography. The novelty of the present approach lies in the use of prelinguistic empirical data for deriving a model from the acquisition of spatial theories. The grounding in empirical studies of developmental psychology makes the formalism cognitively plausible.

7.3 Future work and open questions

The current model is able to simulate an agent that evaluates spatial theories with an environment. The agent is however not able to build automatically these representations. Future work will address more comprehensive methods to build theories automatically from the given percepts.

Discovering new theories could be based on statistics (Seidenberg and Elman, 1999). While Seidenberg and Elman (1999) illustrate statistic learning of patterns in a lingual task recent work by Gopnik and collaborators suggest that learning theories in tasks where children require conditional reasoning can be simulated with a Bayesian Belief Network mechanism (Gopnik and Schulz, 2004; Gopnik, 2005). Cause-effect relationships between variables in a Bayesian network can be defined by statistical analyses from observed data.

Extensions of the present model using a stochastic approach are under consideration. The mechanism would require the automatic creation of new instances and classes in the Haskell code. Manipulation of code during runtime is possible using Template Haskell. Template Haskell however has not been deeply investigated in the course of the research and remains a topic of future research.

[2] The actions of others and later the agent's own actions

Future research will investigate the properties of the current algebras. Especially the identification of morphisms in the present data is planned. Morphisms describe analogies, but you do not often find them in the empirical data, e.g. there is an analogy between an object being visible in a transparent container and an object being visible behind a transparent occluder. Although the developmental sequences point to a domain specific acquisition of two different concepts of transparency (one for containment and one for occlusion), adults' naive transparency theories may be explained by analogy, in the sense of a representational rediscription as proposed by Karmiloff-Smith (1992). The body of empirical data for this kind of investigation will be investigated in future research.

The thesis offered a simple model for the acquisition of spatial concepts in small scale spaces. I conjecture that the bootstrapping of knowledge in the sense of the *theory theory* (Gopnik and Meltzoff, 1997) has implications on the way we build concepts of large scale spaces. Elements of small scale space representations may then be transferred in an adapted form to large scale space representations, perhaps by exploiting morphisms. A large scale space is a space that is explored by wandering through it. Mobility occurs around the age of twelve months, and has strong influence on cognitive development (Thelen et al., 2001; Hannaford, 2002). New tests and empirical studies with children and adults are required in order to test the hypothesis that large scale space representations bootstrap from small scale space representations. Some pointers in this direction have been given by Gattis (2003). The change towards a mobile agent has a high priority on the agenda for future research.

Another open research question are communication processes between agents that hold different concepts. This research would go into the direction of a cognitive framework. In a multi-agent framework agents may hold different conceptualizations of an environment based on their previous experiences. Given the ability to communicate agents could take over the beliefs of other agents. A number of open research questions remains to be solved. How can agents communicate that do not hold the same conceptualizations? I assume that learning to communicate spatial concepts already need the concepts in advance (see (Hespos and Spelke, 2004; Bloom, 2004)).

In a multi-agent framework a model about the beliefs of other agents' beliefs is necessary. Psychologists call this the theory of mind. A formal model of the theory of mind is vital to consider incomplete knowledge of the object concept

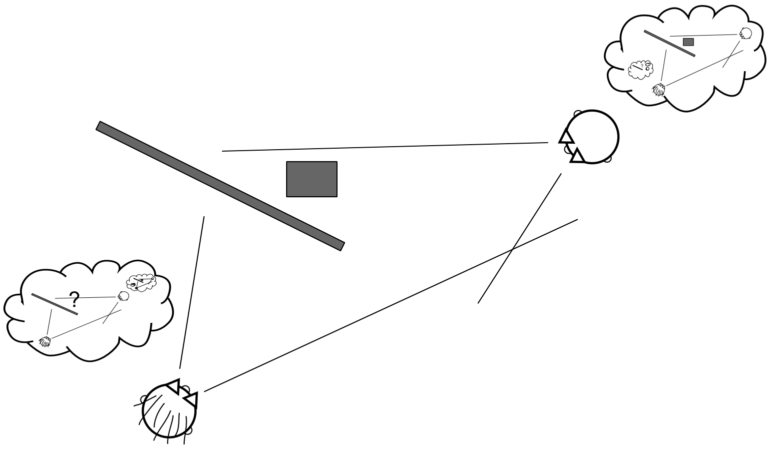


Figure 7.1: The theory of mind

of an agent (see figure 7.1). Towards a more realistic model for the acquisition of spatial theories the proposed agent will be gradually endowed with beliefs, desires, intentions and even emotions. Towards a cognitive architecture memorizing, attention and strength of stimuli should be considered.

7.4 Conclusion

The current approach uses symbols and rules. By formally modeling and simulating empirical studies I have shown that spatial theories can be described by a set of axioms and three theory change mechanisms. Other researchers confirm the use of “rule-like” descriptions for naive theories (Siegler, 1976; Shrager and Siegler, 1998; Siegler and Araya, 2005) as an adequate modeling technique. The vehicle of algebra seems the right tool to describe spatial theories and their change for their use in a computer. I want to stress again that I do not argue here that children think in algebras or are rational in their reasoning (Bower, 1989). The symbolic approach is a tool to describe sequences of changing spatial theories in a computer close to the behavior of people.

Geoinformation systems are more than desktop applications and location based services. Under the paradigm of spatial cognition roadsigns, sketches, verbal descriptions and map representations of robots belong to the realm of geoinformation. The major challenge for spatial cognition is to find out how spatial representations interact. Mechanisms to match different representations will be necessary to describe how new representations develop out of existing representations. A better formal understanding of this matching process will also contribute to system interoperability.

The GI community needs sound formal definitions of the communication between two systems that are based on different conceptualizations. That enables the systems (software agents) to negotiate the concepts held by one system to the concepts held by the other system. An automatic mechanism seems still far based on the current investigations.

The mass of people holds naive concepts of space, physics and any other area of human knowledge. Some people may have advanced knowledge in their field of expertise. But the main part of our everyday reasoning is based on commonsense concepts.

Commonsense concepts start to develop in childhood and underlie frequent changes. Some of these concepts grow stably and should therefore be considered in user interfaces. Not only the naive theories but also a mechanism of change should be considered, e.g. distinguishing novices from expert users. Findings that children can handle three objects at a glance and that adults operate on seven plus or minus two objects easily (Miller, 1956) are just the start. If the aim is that the majority of people uses geoinformation, more formal models of changing commonsense spatial theories are needed urgently.

Bibliography

Aguiar, A. and R. Baillargeon (1999). 2.5-month-old infants' reasoning about when objects should and should not be occluded. *Cognitive Psychology 39*, 116–157.

Alchourrón, C. E., P. Gärdenfors, and D. Makinson (1985). On the logic of theory change: Partial meet contraction and revision functions. *J. Symb. Log. 50*(2), 510–530.

Anderson, J. R. (1993). Problem solving and learning. *American Psychologist 48*, 35–44.

Baillargeon, R. (1994). How do infants learn about the physical world? *Current Directions in Psychological Science 3*, 133–140.

Baillargeon, R. (2000). Infants' understanding of the physical world. Technical report, Infant Cognition Lab - www.psych.uiuc.edu/ rbaillar/ICL/ (access date: 15.12.2006).

Baillargeon, R. (2004a). Infants' physical world. *Current Directions in Psychological Science 13*, 89–94.

Baillargeon, R. (2004b). Infants' reasoning about hidden objects: Evidence for event-general and event-specific expectations. *Developmental Science 7*(4), 391 – 424.

Beer, R. D. (2000). Dynamical approaches to cognitive science. *Trends in Cognitive Science 4*(3), 91–99.

Belnap, N. D. (1977). *Modern uses of multiple-valued logic*, Chapter A Useful Four-Valued Logic, pp. 5–37. Dordrecht, NL, Reidel.

Bird, R. (1998). *Introduction to Functional Programming Using Haskell* (Second Edition ed.). Prentice Hall Series in Computer Science. Hemel Hempstead, UK: Prentice Hall Europe.

Bloom, P. (2004). Children think before they speak. *Nature 430*, 410–411.

Blumenthal, L. M. (1961). *A Modern View of Geometry.* Dover Publications, Inc.

Bower, T. (1974). *Development In Infancy.* San Francisco: W. H. Freeman and Company.

Bower, T. (1989). *The Rational Infant - Learning In Infancy.* New York: W. H. Freeman and Company.

Bringuier, J. C. (2004). *Jean Piaget - Ein Selbstporträt in Gesprächen*. Beltz Taschenbuch. Translation from the french original by Enrico Heinemann and Reinhard Tiffert.

Brooks, R. A. (1986). A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation RA-2*(1), 14–23.

Carey, S. (2004). Bootstrap knowledge. *Deadalus, Winter*, 59 – 68.

Casati, R. (2000). *Time and Motion of Socio-Economic Units*, Chapter The Structure of Shadows, pp. 99–109. London: Taylor and Francis.

Choi, S., L. McDonough, M. Bowerman, and J. M. Mandler (1999). Early sensitivity to language-specific spatial categories in english and korean. *Cognitive Development 14*, 241–268.

Egenhofer, M. J. and D. M. Mark (1995). Naive geography. In A. U. Frank and W. Kuhn (Eds.), *Lecture Notes in Computer Science (COSIT '95, Semmering, Austria)*, Volume 988, pp. 1–15. Springer Verlag.

Egenhofer, M. J. and A. M. Rodriguez (1999). Relation algebras over containers and surfaces: An ontological study of a room space. *Journal of Spatial Cognition and Computation 1*(2), 155–180.

Fauconnier, G. (1997). *Mappings in Thought and Language*. Cambridge University Press.

Fellbaum, C. (1998). *WordNet : An Electronic Lexical Database - Language, Speech, and Communication* (2nd edition ed.). Cambridge, Massachussets: MIT Press.

Feynman, R. (1998). *The Character of Physical Law*. Cambridge, Mass.: The MIT Press.

Fitzpatrick, P., G. Metta, L. Natale, R. Sajit, and G. Sadini (2003). Learning about objects through action - initial steps towards artificial cognition. In *International Conference on Robotics & Automation*, pp. 3140–3145. IEEE.

Fodor, J. A. (1987). *The Modularity of Mind: An Essay on Faculty Psychology* (5 ed.). Cambridge, Mass.: MIT Press.

Frank, A. U. (1993). The use of geographical information systems: The user interface is the system. In H. M. Medyckyj-Scott, David Hearnshaw (Ed.), *Human Factors in Geographical Information Systems*, pp. 3–14. London and Florida: Belhaven Press.

Frank, A. U. (1998). Specifications for interoperability: Formalizing spatial relations 'in','auf' and 'an' and the corresponding image schemata 'container', 'surface'and 'link'. In *1st Agile-Conference*, Enschede, Netherlands. ITC.

Frank, A. U. (1999). One step up the abstraction ladder: Combining algebras - from functional pieces to a whole. In C. Freksa and D. M. Mark (Eds.), *Spatial Information Theory - Cognitive and Computational Foundations ofGeographic Information Science (Int. Conference COSIT'99, Stade, Germany)*, Volume 1661 of *Lecture Notes in Computer Science*, pp. 95–107. Berlin: Springer-Verlag.

Frank, A. U. (2001). Tiers of ontology and consistency constraints in geographic information systems. *International Journal of Geographical Information Science 15*(7 (Special Issue on Ontology of Geographic Information)), 667–678.

Frank, A. U. (2006). Distinctions produce a taxonomic lattice: Are these the units of mentalese? In B. Bennett and C. Fellbaum (Eds.), *International Conference on Formal Ontology in Information Systems*, Volume 150 of *Frontiers in Artificial Intelligence and Applications*, Baltimore, Maryland, pp. 27–38. IOS Press.

Frank, A. U. and D. Medak (1997). Executable axiomatic specification using functional language - case study: Base ontology for a spatio-temporal database. Technical report, Institute for Geoinformation and Cartography, Vienna University of Technology.

Frank, A. U. and M. Raubal (1998). Specifications for interoperability: Formalizing image schemata for geographic space. In *8th International Symposium on Spatial Data Handling. SDH '98*, Vancouver, Canada.

Frank, A. U. and M. Raubal (1999). Formal specification of image schemata - a step towards interoperability in geographic information systems. *Spatial Cognition and Computation 1*(1), 67– 101.

Fulghum, R. (1993). *All I really need to know I learned in Kindergarten.* New York: Ballantine Books.

Gasser, M. and E. Colunga (1997). Playpen: Toward an architecture for modeling the development of spatial cognition. Technical report, Computer Science and Linguistics Department, Indiana University, Bloomington, IN 47405, USA.

Gattis, M. (2003). Are reasoning in small and large scale space different? In *International workshop on Spatial and Visual Components in Mental Reasoning about Large-Scale Spaces.*

Gibson, J. J. (1979). *The Ecological Approach to Visual Perception.* Company, Boston.: Houghton Mifflin.

Gopnik, A. (2005). *A Brand New Ball Game: Bayes Net and Neural Learning Mechanisms in Young Children*, Chapter In Attention and Performance XXI: Processes of Change in Brain and Cognitive Development. Oxford University Press (in press).

Gopnik, A. and A. N. Meltzoff (1997). *Words, Thoughts, and Theories.* Learning, Development, and Conceptual Change. Cambridge, Massachusetts: MIT Press.

Gopnik, A., A. N. Meltzoff, and P. K. Kuhl (1999). *The Scientist in the Crib - What Early Learning Tells us about the Mind* (1 ed.). New York: Perennial - HarperCollins.

Gopnik, A. and L. Schulz (2004). Mechanism of theory formation in young children. *TRENDS in Cognitive Science 8*(8), 371–377.

Granott, N., K. W. Fischer, and J. Parziale (2002). *Microdevelopment: Transition Processes in Development and Learning*, Chapter Bridging to The Unknown: A Transition Mechanism in Learning and Development, pp. 1–41. Cambridge, U.K.: Cambridge University Press.

Hannaford, C. (2002). *Bewegung - das Tor zum Lernen* (5th ed.). Kirchzarten, VAK Verlags GmbH. Translation: Smart Moves - Why Learning is not All in Your Head.

Hayes, P. (1985). The second naive physics manifesto. In J. R. Hobbs and R. C. Moore (Eds.), *Formal Theories of the Commonsense World*, Ablex Series in Artificial Intelligence, pp. 1–36. Norwood, New Jersey: Ablex Publishing Corporation.

Hayes, P. J. (1978). The naive physics manifesto. In D. Mitchie (Ed.), *Expert Systems in the Microelectronic Age*, pp. 242–270. Edinburgh: Edinburgh University Press.

Hespos, S. J. and R. Baillargeon (2001a). Infants' knowledge about occlusion and containment events: A surprising discrepancy. *Psychological Science 12*(2), 141–147.

Hespos, S. J. and R. Baillargeon (2001b). Reasoning about containment events in very young infants. *Cognition 78*, 207 – 245.

Hespos, S. J. and E. S. Spelke (2004). Conceptual precursors to language. *Nature 430*, 453–455.

Hideki, K. and Y. Hiroyuki (2001). A robot that learns to communicate with human caregivers. In *International Workshop on Epigenetic Robotics, EpiRob 2001*, Lund, Sweden, pp. 47–52.

Hiraki, K., A. Sashima, and S. Phillips (1998). From egocentric to allocentric spatial behavior: A computationalmodel of spatial development. *Adaptive Behaviour 6*(3/4), 371–391.

Hobbs, J. and R. C. Moore (1985). *Formal Theories of the Commonsense World*. Ablex Series in Artificial Intelligence. Norwood, NJ: Ablex Publishing Corp.

Hofer, B. (2007, 11-13 April 2007). A classification of spatial processes based on pdes. In *GISRUK 2007*, Maynooth Ireland.

Hofsten, C. v., P. Vishton, E. S. Spelke, Q. Feng, and K. Rosander (1998). Predictive action in infancy: Tracking and reaching for moving objects. *Cognition 67*, 255–285.

Hudak, P. and J. H. Fasel (1992). A gentle introduction to haskell. *ACM Sigplan Notices 27*(5), 1–53.

Hudak, P. and M. P. Jones (1994). Haskell vs. ada vs. c++ vs. awk vs. ... an experiment in software prototyping productivity. Technical report, Yale University, Department of Computer Science.

Hughes, J. (1989). Why functional programming matters. *Computer Journal 32*(2), 98–107.

Irschitz, O. (2004). R.O.S.I. (real time orientation super interactive) the new dimension of tourism information. In *CORP*, Vienna. Multimedia Plan.

Johnson, M. (1987). *The Body in the Mind: The Bodily Basis of Meaning, Imagination, and Reason.* University of Chicago Press.

Johnson, M. H. (1999). *Developmental Cognitive Neuroscience: Relating Brain Development to Cognitive Change*, Chapter 2, pp. 23–41. London, Great Britain: Arnold, a member of the Hodder Headline Group.

Karmiloff-Smith, A. (1992). *Beyond Modularity.* MIT Press.

Kim, I.-K. and E. S. Spelke (1999). Perception and understanding of effects of gravity and inertia on object motion. *Developmental Science 2*(3), 339–362.

Krek, A. (2002). *An Agent-Based Model for Quantifying the Economic Value of Geographic Information.*, Volume 26 of *Geoinfo Series.* Institute for Geoinformation and Cartography, Vienna University of Technology.

Kuhn, T. S. (1962). *Die Struktur wissenschaftlicher Revolutionen* (1. Auflage, November 2003 ed.). Suhrkamp Taschenbuch Wissenschaft. translated from the English Original "The Structure of Scientific Revolutions", 2. revised edition, 1976.

Kuhn, W. and A. Frank (1991a). *Cognitive and Linguistic Aspects of Geographic Space*, Chapter A Formalization of Metaphors and Image-Schemas in User Interfaces, pp. 419–434. Kluwer, Academic Publishers.

Kuhn, W. and A. U. Frank (1991b). A formalization of metaphors and image-schemas in user interfaces. In A. U. Frank and D. M. Mark (Eds.), *Cognitive and Linguistic Aspects of Geographic Space*, Volume 63 of *NATO ASI Series D.* Dordrecht, The Netherlands: Kluwer Academic Publishers.

Kuipers, B. (1998). A hierarchy of qualitative representations for space. In C. Freksa, C. Habel, and K. Wender (Eds.), *Spatial Cognition: An Interdisciplinary Approach to Representing and Processing Spatial Knowledge*, Volume 1404 of *LNCS*, pp. 337–350.

Kuipers, B. (2000). The spatial semantic hierarchy. *Artificial Intelligence 119*, 191–233.

Laird, J., A. Newell, and P. Rosenbloom (1987, September). Soar: An architecture for general intelligence. *Artificial Intelligence 33*(1), 1–64.

Liskov, B. and J. Guttag (1989). *Abstraction and Specification in Program Development.* The MIT Electrical Engineering and Computer Science Series. Cambridge, MA: MIT Press.

Liskov, B. and S. Zilles (1978). An introduction to formal specifications of data abstractions. *Current Trends in Programming Methodology. Prentice-Hall*(1), 1–32.

Loeckx, J., H.-D. Ehrich, and W. Markus (1996). *Specification of Abstract Data Types.* Chichester, UK and Stuttgart: Wiley, John and Teubner, B.G.

Luger, G. F., T. Bower, and J. G. Wishart (1983). A model of the conceptual development of the early infant object concept. *Perception 12*, 21–34.

Luger, G. F., J. G. Wishart, and T. Bower (1984). Modelling the stages of the identity theory of object-concept development in infancy. *Perception 13*, 97–115.

Luo, Y. and R. Baillargeon (2005). When the ordinary seems unexpected: Evidence for rule-based physical reasoning in young infants. *Cognition 95*(3), 297 – 328.

MacWhinney, B. (2000). *The CHILDES Project: Tools for Analyzing Talk* (Third Edition. ed.). Mahwah, NJ: Lawrence Erlbaum Associates.

Marcus, G. F. (2001). *The Algebraic Mind: Integrating Connectionism and Cognitive Science.* MIT Press.

Marcus, G. F., S. Vijayan, S. Rao, and P. M. Vishton (1999). Rule learning in seven month-old infants. *Science 283*, 77–80.

Mareschal, D. (2000). Object knowledge in infancy: Current controversies and approaches. *Trends in Cognitive Sciences 4*(11), 408–416.

Mareschal, D. (2003). *Encyclopaedia of Cognitive Science*, Chapter Computational Models of Cognitive Development, pp. 533–39. London, UK: Macmillan.

Mark, D. M. (1993). *Human Factors in Geographical Information Systems*, Chapter Human Spatial Cognition, pp. 51–60. Belhaven Press.

Mark, D. M. and M. J. Egenhofer (1996). Common-sense geography: Foundations for intuitive geographic informationsystems. In *GIS/LIS '96*, Betsheda, pp. 935–941. American Society for Photogrammetry and Remote Sensing.

McCarthy, J. and P. J. Hayes (1969). Some philosophical problems from the standpoint of artificial intelligence. In B. Meltzer and D. Michie (Eds.), *Machine Intelligence 4*, pp. 463–502. Edinburgh University Press.

McCloskey, M. (1983). Naive theories of motion. In D. Genter and A. L. Stevens (Eds.), *Mental Models*, pp. 299 – 324. Lawrence Erlbaum Associates.

Meltzoff, A. N. (2004). *Theories of Infant Development*, Chapter The Case for Developmental Cognitive Science: Theories of People and Things, pp. 145–173. Oxford: Blackwell Publishing.

Miller, G. A. (1956). The magic number seven, plus or minus two - some limits in our capacity for processing information. *Psychological Review 63*, 81–97.

Munkata, Y. and J. L. McClelland (2003). Connectionist models of development. *Developmental Science 6*(4), 413–429.

Nelson, B. D., R. H. Aron, and M. A. Francek (1992). Clarification of selected misconceptions in physical geography. *Journal of Geography 91*(2), 76–80.

Newcombe, N. S. and J. Huttenlocher (2003). *Making Space: The Development of Spatial Representation and Reasoning.* Learning, Development, and Conceptual Change. Cambridge, Massachusetts: MIT Press.

Newell, A. and H. A. Simon (1972). *Human problem solving.* Prentice Hall.

Nwana, H. S. and D. T. Ndumu (1999). A perspective on software agent research. *The Knowledge Engineering Review 14*(2), 1–18.

Parisi, D. and M. Schlesinger (2002). Artificial life and piaget. *Cognitive Development 17*, 1301–1321.

Piaget, J. P. (1950). *Der Aufbau der Wirklichkeit beim Kinde* (2. Auflage, 1998 ed.), Volume 2 of *Gesammelte Werke, Studienausgabe.* Klett-Cotta. translated from the french original ”La construction du rèel chez l'enfant” by Sandberger, J.U.; Thirion, C. and Wundberg, H.

Pine, K. (1999). *Exploring Developmental Psychology*, Chapter Theories of Cognitive Development, pp. 41–61. Arnold London, Sydney, Auckland - copublished Oxford University Press, NewYork.

Pinker, S. (1995). *The Language Instinct.* New York: Harper Perennial.

Polya, G. (1973). *How to Solve It.* Princeton University Press.

Popper, K. R. (1934). *Logik der Forschung* (8. Auflage (1984) ed.). J.C.B. Mohr (Paul Siebeck) Tübingen.

Raubal, M. (2001). *Agent-based Simulation of Human Wayfinding: A Perceptual Model for UnfamiliarBuildings.* Ph.d. thesis, Vienna University of Technology.

Regier, T. (1996). *The Human Semantic Potential - Spatial Language and Constrained Connectionism.* Cambridge, Massachusetts London, England: MIT Press.

Rochat, P. (2004). *The Infant's World.* Cambridge, Massachusetts, and London, England: Harvard University Press.

Rochat, P. and S. J. Hespos (1996). Tracking and anticipation of invisible transformations by 4- to 8-month-old infants. *Cognitive Development 11*, 3–17.

Rodriguez, A. M. and M. J. Egenhofer (1997). Image-schemata-based spatial inferences: The container-surface algebra. In S. Hirtle and A. Frank (Eds.), *Spatial Information Theory - A Theoretical Basis for GIS, International Conference COSIT'97*, Volume Vol. 1329 of *LNCS*, Laurel Highlands, PA, pp. 35–52. Springer-Verlag.

Rodriguez, A. M. and M. J. Egenhofer (2000). A comparison of inferences about containers and surfaces in small-scale and large scale spaces. *Journal of Visual Languages and Computing 11*(6), 639–662.

Ruetschi, U.-J. and S. Timpf (2005). Using image schemata to represent meaningful spatial configurations. In R. e. a. Meersman (Ed.), *On the Move to Meaningful Internet Systems 2005: OTM 2005 Workshop*, Volume 3762 of *LNCS*, Berlin, Heidelberg, pp. 1047–1055. Springer-Verlag.

Russell, S. J. and P. Norvig (1995). *Artificial Intelligence*. Prentice Hall Series in Artificial Intelligence. Englewood Cliffs, NJ: Prentice Hall.

Scassellati, B. (2000). Theory of mind for a humanoid robot. In *IEEE/RSJ International Conference on Humanoid Robots*.

Schlesinger, M. (2002). A lesson from robotics: Modeling infants as autonomous agents. In C. Prince, Y. Demiris, Y. Marom, H. Kozima, and H. Balkenius (Eds.), *Second International Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems*, Sweden, pp. 133–140. Lund University Cognitive Studies.

Schlesinger, M. and D. Parisi (2001). The agent approch: A new direction for computational models of development. *Developmental Review 21*, 121–146.

Schlesinger, M. and M. Young (2003). Examining the role of prediction in infants' physical knowledge. In R. Alterman and D. Kirsh (Eds.), *Proceedings of the Twenty-Fifth Annual Meeting of the Cognitive Science Society*, Boston, pp. 1047–1052. Cognitive Science Society.

Schrage, M. M., A. van Ijzendoorn, and L. C. van der Gaag (2005). Haskell ready to dazzle the real world. In *Haskell '05: Proceedings of the 2005 ACM SIGPLAN Workshop on Haskell*, New York, NY, USA, pp. 17–26. ACM Press.

Seidenberg, M. S. and J. L. Elman (1999). Do infants learn grammar with algebra or statistics? *Science 284*, 433–435.

Shastri, L. (1999). Infants learning algebraic rules. *Science 285*, 1673–1674.

Shrager, J. and R. S. Siegler (1998). Scads: A model of children's strategy choices and strategy discoveries. *Psychological Science 9*(5), 405–410.

Siegler, R. and R. Araya (2005). A computational model of conscious and unconscious strategy discovery. In R. V. Kail (Ed.), *Advances in Child Development and Behavior*, Volume 33, pp. 1–42. Elsevier Academic Press.

Siegler, R. and C. Shipley (1995). *Variation, selection, and cognitive change*, Chapter Developing cognitive competence: New approaches to process modeling, pp. 31–76. Hillsdale, NJ: Erlbaum.

Siegler, R. S. (1976). Three aspects of cognitive development. *Cognitive Psychology 8*, 481–520.

Siegler, R. S. (2002). Microgenetic studies of self-explaination. In N. Granott and J. Parziale (Eds.), *Microdevelopment - Transaction Processes in Development and Learning*, pp. 31–58. Cambridge University Press.

Siegler, R. S. and Z. Chen (2002). Development of rules and strategies: Balancing the old and the new. *Journal of Experimental Child Psychology 81*, 446–457.

Smith, B. and D. M. Mark (2001). Geographical categories: An ontological investigation. *International Journal of Geographical Information Science 15*(7 (Special Issue - Ontology in the Geographic Domain)), 591–612.

Sowa, J. F. (1999). *Knowledge Representation: Logical, Philosophical and Computational Foundations.* Pacific Grove, CA: Brooks Cole Publishing Co.

Spelke, E. S. (1990). Principles of object perception. *Cognitive Science 14*, 29–56.

Spelke, E. S. (2000). Core knowledge. *American Psychologist November 2000*, 1233–1243.

Spelke, E. S., K. Breinlinger, J. Macomber, and K. Jacobson (1992). Origins of knowledge. *Psychological Review 99*, 605–632.

Thelen, E., G. Schöner, C. Scheier, and L. Smith (2001). The dynamics of embodiment: A field theory of infant perservative reaching. *Behavioral and Brain Sciences 24*(1), 1–86.

Thompson, S. (1996). *Haskell - The Craft of Functional Programming.* International Computer Science Series. Harlow, UK: Addison-Wesley. Inv.No. 602.171.

Viezzer, M. and C. H. M. Nieuwenhuis (2005). Learning affordance concepts: Some seminal ideas. Technical report, Thales Research and Technology Nederland School of Computer Science, The University of Birmingham.

von Hofsten, C., Q. Feng, and E. S. Spelke (2000). Object representation and predictive action in infancy. *Developmental Science 3*(2), 193–205.

Wang, S.-h., B. Renee, and P. Sarah (2005). Detecting continuity violations in infancy: A new account and new evidence from covering and tube events. *Cognition 95*, 129 – 173.

Woolridge, M. (1999). *Intelligent Agents. Multiagent Systems - A Modern Approach to Distributed Artificial Intelligence.* Cambridge, Massachusetts: MIT Press.

Young, R. M. (1976). *Seriation by Children: An Artificial Intelligence Analysis of a Piagetian Task.* Birkhauser.

Ziemke, T. (2003, October 23rd - 25th, 2005). What's that thing called embodiment? In *25th Annual Meeting of the Cognitive Science Society*, Brighton, UK. Lawrence Erlbaum.

Appendix A

Theory Change and Development

A.1 Occlusion of objects

PERCEPTION	**Axiom - A is occluded by B if**	**Age**	**CONCEPTUAL CHANGE**
-	A is *behind* B	< 2.5 months	-
structure of occluder (presence of doorways and windows)	previous + has no window B	3 months	SPECIALIZATION
height of the objects involved	previous + height A < height B	3.5 months	SPECIALIZATION
width of object	previous + width A < width B	7 months	SPECIALIZATION
transparency	previous + not transparent B	7.5 months	GENERALIZATION
	transparent B		

Table A.1: Solid Occlusion Theory Sequence

A.2 Support of objects

PERCEPTION	**Axiom -** A *is ON* B if	**Age**	**CONCEPTUAL CHANGE**
-	previous + A has contact with B	<=3 months	-
type of contact	previous + A is on top of B	4.5-5.5 months	SPEZIALIZATION
movability inside	previous + movable on B	5.5 months	GENERALIZATION
	previous + not movable on B	5.5 months	
amount of contact	previous + amount of contact is supportive	6.5 months	SPECIALIZATION amount of contact
shape of the supported object	previous + shape of B	12.5 months	SPECIALIZATION shape of supported object

Table A.2: Solid Support Theory Sequence

A.3 Containment of objects

PERCEPTION	**Axiom - A is in B if**	**Age**	**CONCEPTUAL CHANGE**
movement of the solid	A shares movement with container B	2.5, 3.5 months	-
opening of the solid	previous + B has an opening	2.5, 3.5 months	SPECIALIZATION
functional distinction between containers	previous + A is movable in B	5 months	GENERALIZATION
	previous + A not movable in container B	5 months	
size of the solid (width 4-6 months, height 7.5 months)	previous + A < B	7.5 months	SPECIALIZATION
transparency	previous + not transparent container B	10 months	GENERALIZATION
	transparent	10 months	

Table A.3: Solid Containment Theory Sequence

Appendix B

Formal Tools and Methods

This chapter describes the formal methods and tools used in the present thesis. In order to build a formal theory means to specify precisely the terms and relations of an underlying conceptual model. Errors in the conceptual model can be detected in the process of formalization. Abstraction helps to keep the formal theory clean by avoiding unnecessary details.

The model was implemented using the functional programming language Haskell. Classes represent algebras and thus theories. Polymorphism has been used to overload the *expectation functions* of a theory. Partial initialization of functions has been utilized to treat all expectation functions in the same way. These features will be outlined in the section about functional programming.

The hypothesis is based on axiomatic specification. The aim is to verify that a spatial representation can be built upon algebras. A change in these representations can be reflected in an adaptation of underlying axioms.

B.1 Formal specifications

Domain experts in psychology or philosophy usually provide informal descriptions about human behavior. The interpretations of empirical experiments provide information about the actions, behaviors and expectations of humans in a certain setting. In order to describe these in a computer, *formal specifications* are required. In a *formal specification* a problem or task is described in terms of actions, behaviors and expected results (Liskov and Guttag, 1989).

A formal specification can be seen as a layer between the concept, one holds in his mind and an infinite number of possible computer programs for the concept.

The formal specification provides a mathematical description of the concept.

In the present thesis we use algebraic specifications as a formal method. The term algebra as used has been issue of controversies (Frank and Medak, 1997). I use a weak notion of the mathematical definition. In order to make the specifications executable only constructive axioms are allowed. Therefore it has been suggested to say model based (executable) specification method, rather than using the term algebraic specifications.

B.1.1 Algebras

Algebras describe mathematical structures. There are well known algebras, like the algebra for natural numbers, the Boolean algebra or the linear algebra for vector calculations. An algebra groups operations that are applied to the same data type e.g. the Boolean algebra has operations that are all applied to truth values. I use a definition of computer science and refer to many sorted algebra that can be structured into three parts:

1. A set of *sorts* that identifies involved objects.
2. A set of *operations* that describes what can be done with the objects and groups them by their functionality.
3. A set of *axioms* that defines the behavior of the operations.

Informally *sorts* stand for types, objects or carriers. They abstract from individual values to a set of values. An algebra that depends just on one sort is called a single-sorted algebra in comparison with many-sorted algebra that depend on many different types.

Operations of data abstraction are classified into constructors and observers. They are carried out just over defined sorts. Liskov and Guttag (1989) distinguish four types of operations. *Primitive constructor operations* create sorts without taking sorts of their type as input. *Constructor operations* take sorts of their type as input arguments and create other sorts of their type. *Mutator operations* modify sorts of their type. *Observer operations* return the properties of their sorts.

Axioms describe the behavior of operations. This allows to predict the outcome of an operation. Given a set of axioms one can predict which sorts can be constructed out of a given algebra.

B.1.2 Abstract Data Types with Algebraic Specifications

The given example serves as an illustration of how to define an abstract data type using algebraic specifications. The data structure of a stack can be modeled with a many-sorted algebra. The example given below can be found in the literature (Liskov and Zilles, 1978; Frank, 1999). The sorts used are a stack of elements a and the element a.

```
Algebra Stack (stack of a, a)


Operations:


create :: stack of a                     -- primitive constructor
push :: stack of a -> a -> stack of a    -- constructor
top :: stack of a -> a                   -- observer
pop :: stack of a -> stack of a          -- mutator


Axioms:


top (push s a) = a     -- axiom 1
pop (push s a) = s    -- axiom 2
top (create) = error    -- axiom 3
pop (create) = error    -- axiom 4
```

The primitive constructor operation `create` serves to construct an empty stack, while the constructor operation `push` takes a stack of elements and an element as input arguments and returns a stack of elements with the element added. `Top` returns the topmost element of the stack, being an observer operation. `Pop` is an example for a mutator operation, returning the stack with the top element being removed.

The axioms define the behavior of the operations. Axiom 1 specifies that the top element of a stack of elements is the element that has been recently pushed on the stack. Axiom 2 states that after having pushed an element on the stack of elements pop returns the same stack of elements as before the execution of the operation push. The behavior of the operations top and pop is undefined for an empty stack, therefore the error sort is added for the definition of axiom 3 and axiom 4.

B.1.3 Summary

Algebraic specifications are based on the mathematical sound theory of algebra. Algebraic specifications have been used to define abstract data types for the spatial and temporal domain (Kuhn and Frank, 1991b; Frank and Medak, 1997; Frank and Raubal, 1999; Raubal, 2001; Krek, 2002).

In the present thesis spatial theories as found with infants are described with algebraic specifications. Three theory change mechanisms are implemented using the functional programming paradigm. Together with a purely functional programming language algebraic specifications lead to executable specifications.

B.2 Functional Programming

Functional programs consist entirely of functions (Hughes, 1989). Each function takes a number of input types and returns a single output type. Constants are functions, which always return the same value. Even the program itself is a function. Because a function call can have no other effect than producing a result, functional programs are said to have *no side effects*.

```
General function pattern:
fktname :: par1typ -> ... -> parntyp -> fkttyp
```

A number of implementations for functional programming languages is available. An incomplete list of available functional programming languages follows:λ-calculus, Lisp, ML, SML, Hope, Miranda, OPAL, Haskell, Gofer. For the present thesis Haskell has been chosen as a tool.

Functional programming is grounded in the declarative programing paradigm. In comparison with imperative programming languages like C++ or Java, functional programming languages do not depend on the sequence of commands. Expressions are evaluated comparable with the evaluation of cells in a spreadsheet application.

The evaluation process consists of alternating *expansion* and *simplification* steps. Depending on the evaluation strategy applicative order and normal order evaluation languages can be distinguished. Function application is the operator with the highest priority.

B.2.1 Data Types and Strong Typing

Functional programming languages support a set of built in types. Expressions are evaluated and their result is associated with a type e.g. `Int, Integer, Char, String , and Bool.` These are primitive types or base types.

Programming languages that support data types have two advantages. Firstly the programmer does not have to bother with the representation of a certain type in the memory of the computer. Secondly the compiler can assist the programmer by type-checking the meaning of expressions with a type inference mechanism. Type errors can be detected at the earlier stage of compile time rather at the runtime of the program.

Types can be composed to a fixed set, a tuple or to a set of undefined size, namely a list. Lists are the most common and often used type in a functional programming language. Predefined functions are usually available in functional programming languages. Tuples, lists and functions can be further combined to some more complex data types as lists of lists, lists of tuples and lists of functions, e.g. a string is defined as a list of characters. Haskell lists are enclosed in squared brackets - `[ ]`.

```
-42 :: Int
String = [Char]
Tuple = (Float,Float)
ArbitraryList = [Type]
```

User defined data types extend functional programming languages so that data types of any complexity can be described. Haskell allows the definition of algebraic data types and abstract data types. In the most general form a user defined data type consists of a type name (typeName) and n constructor functions (Con_i, i = 1 .. n), each followed by a number of types. The arity of the constructor function can range from 0 to n.

```
data typeName
    = Con1 t11 ... t1k1 |
       Con2 t21 ... t2k2 |
            ...
       Conn tn1 ... tnkn
```

A new type can be formed just by enumerating its elements, e.g. the type Season has the constructor functions `Spring, Summer, Fall and Winter`. The constructor functions are of arity 0. The type is called an *enumeration type*.

```
data Season = Spring | Summer | Fall | Winter
```

When the constructor function has an arity equal or bigger than one, a *product type* can be defined. In the following example the product type Person has a constructor function Person. The constructor takes the types ID, Name and Age as input, they have been defined as aliases to available primitive types. The constructor behaves like a function that has the signature `(ID -> Name -> Age -> Person)`.

```
type ID = Int
type Name = String
type Age = Int
data Person :: Person ID Name Age
```

The combination of nullary and unary constructor functions lead to *sum types.* All types can be defined as sum types. Sum types allow recursive definitions and the use of type variables. Therefore they can be of arbitrary complexity. The implementation of a binary tree can be found below.

```
data Tree a = Leaf a |
              Node a (Tree a) (Tree a)
```

B.2.2 Polymorphism

Polymorphism has been used to overload a theory with different implementations. *Polymorphism* is the property of a function that can be applied to arguments of different types. *Parametric polymorphism* enables the reuse of code by defining a function for a data structure independent on a given parameter type, e.g. the implementation of the length of a list is not dependent on the parameter type. The code sample below illustrates the recursive definition of function to calculate the length of a list. The implementation can serve different parameter types, such as `Float, Int, Char`. Using polymorphism in programming leads to less code and thus shorter programs (Hudak and Fasel, 1992).

```
length :: [a] -> Int
length [] = 0
length (x:xs) = 1 + length (xs)
```

Ad-hoc polymorphism or *overloading* gathers different types in one operation together. Different implementations for the operation depending on the type have to be provided. Haskell introduces the concept of type classes for ad-hoc polymorphism, which will be explained in section B.3.2.

B.2.3 High Order Functions

High order functions are used to describe preinitialzed theories. A functional or high order function is a function whose arguments are functions or whose result is a function. The map function illustrates how a high order function works. Map takes a function with the signature `(a -> b)` and a list of arguments `[a]` and applies the input function to each argument in the argument list. It returns the list of evaluated arguments.

```
map :: (a -> b) ->[a] -> [b]
map f xs = [f x |x<-xs]
```

Other examples are the filter or fold functionals that can be found in the Haskell language report (Hudak et al. 1992). Functional programming languages allow partial function application. A function that receives fewer arguments than needed will be partially evaluated. Its result is a new function that waits for missing input arguments.

```
> map (<3) [1,2,3]
> [True,True,False]
```

A function that takes the input arguments one after the other, is called curried. Only curried functions allow partial function application. An uncurried function bundles its input arguments in a tuple. In order to uncurry a curried function two operations are necessary:

```
curry ::  ((a,b) -> c) -> (a -> b -> c))
uncurry :: (a -> b -> c) -> ((a,b) -> c)
```

New functions can also be created by using function composition. The composition of two continuous functions yields to a continuous function. Functional programming languages offer function composition via an operator.

```
(.) :: (b->c) -> (a->b) -> (a -> c)
(f .g) x = f (g x)
```

High order functions distinguish functional programming from other programming languages. The partial application of functions and the possibility to define functions of functions permit abstract specifications. As a result generic code can be written that is modular and highly reuseable.

B.3 Haskell

The functional programming language Haskell was invented by Haskell B. Curry. It is based on a Turing complete computability system, the λ-calculus. Haskell extends the λ-calculus by syntax and makes it thus executable.

Haskell is a purely functional programming language. That means that expressions written in Haskell do not cause side-effects. When values of expressions do not depend on unknown states a programming language is *referentially transparent*.

Haskell is based on the Hindley Miller type system and supports the following predefined types in its standard prelude: integer (`Int`), floating point numbers (`Float`), Boolean values (`Bool`), characters (`Char`), lists (`[a]`), strings (`String`) and tuples. Haskell expressions are usually evaluated lazy. An introduction to Haskell can be found in the literature (Hughes, 1989; Hudak and Fasel, 1992; Thompson, 1996; Bird, 1998).

B.3.1 Syntax

Function names and variables start with small letters, types, constructors and modules start with capital letters. Spaces and brackets separate names. Arguments (of functions) are separated by spaces. Spaces have a higher precedence than any other operations.

Expressions hardly need to be bracketed in Haskell as the layout rule defines top levels of a program. Code that is to be bracketed can be simply indented. Every indentation ”opens” a new bracket.

```
Function application:
f (x) = f x
Code example - quicksort algorithm
qsort :: (Ord a) => [a] -> [a]
qsort [] = []
qsort(x:xs) = qsort [i|i<-xs,i<=x] ++ [x] ++ qsort[i|i<-xs,i>x]
```

Code written in functional programming languages is short. Factors between 5 - 20 in terms of smaller code length, compared with imperative programming languages have been reported (Hudak and Jones, 1994; Schrage et al., 2005). The example implementation of the quicksort algorithm illustrates this.

B.3.2 Classes and Instances

Classes are a specific feature of the Haskell programming language. A type class is equivalent to an algebra of types. The elements of a type class are called the instances of the class. Haskell predefines a number of classes shown in figure 5.1, in its prelude file. User defined classes can be built in the same way as illustrated in the examples below.

```
        Eq
        / \
       /   \
     Ord    \     Text
     / \     \    /
    /   \     \  /
   /    Enum  Num
  /       \   / \
Ix         \ /   \
  \      Real Fractional
   \      / \   / \
    \    /   \ /   \
  Integral RealFrac Floating
               \   /
                \ /
              RealFloat
```

Figure B.1: Hierarchy of Haskell classes

On the top of the hierarchy there is the class `Eq`. It defines the collection of types with which the equality of two elements can be tested. The class declaration of `Eq` is given below. It consists of a class Name followed by a signature. The signature is a list of names and their types. The equality operation `(==)` takes two types and returns a Boolean value.

```
class Eq a where
(==) :: a -> a -> Bool
```

In order to make an arbitrary type a member of the class `Eq` an implementation has to be provided. Members of a type class are called `instances`. Haskell defines `instances` for `Int, Float, Bool` and `Char` for `Eq`. Furthermore a default implementation for the class Eq is given that can be overwritten with a new implementation.

The equality between two different types may differ. The equality of two natural numbers may be implemented by simply comparing their values, while the equality of two strings may be based on comparing the length of the two strings. Different instances may be defined. The appropriate implementation will be *overloaded* for the corresponding type. This way type classes implement the ad-hoc polymorphism mentioned earlier.

```
instance Eq Int where
    (==) a b = a == b
instance Eq String where
    (==) a b = length a == length b
```

A user can define algebraic types in a three step procedure. Firstly a class *declaration* has to be provided. Secondly a *representation* for a data type has to be defined using sum types or type aliases. Thirdly an *implementation* has to be given through instances of the class.

```
data Point = Point Float Float
class Points p where
getX :: p -> Float
getY :: p -> Float
instance Points Point where
getX (Point x y) = x
getY (Point x y) = y
```

Haskell also offers a mechanism of inheritance, similar to object oriented programming languages. The class Ord can be derived from the class Eq. Ord defines the class of ordered types. The class Ord defines the operations to compare types like <, <=, >, >= . The definition of equality, defined by the == operator is inherited of the class Eq.

```
class Eq a => Ord a where
(<),(<=),(>),(>=) :: a -> a -> Bool
...
```

In the sample code above the => operator indicates that Ord is derived from Eq. The operator => refers to the context of a class. In the example that means for any type a that is declared and implemented belonging to Ord there has to be also a declaration and implementation belonging to Eq.

In the present thesis classes represent spatial theories. The classes can depend on a single or multiple parameter. Using classes hierarchies for theories can be built.

B.3.3 Modularization

Haskell programs can be split into different modules. Modules are parts of a computer program that can be maintained independently. Modules can be reused and avoid unnecessary copying of code. Each module can be compiled on its own. Modules help to deal complexity by splitting the problem into simple parts that can be studied individually.

Each theory has been implemented in an own module. A module contains the code for a theory, perceptions and test data. The mechanism to evaluate theories has been implemented as its own module. Modules can be exchanged in order to improve and extend the model.

B.4 Summary

Spatial theories are described with algebraic specifications. A change in the algebraic structure is reflected in an adaptation of axioms. We use algebra in its simplest definition as a set of sorts, operations, and axioms (Loeckx et al., 1996). The advantage of using algebra for modeling is its mathematical sound- and compactness, e.g the reuse of code by defining sub algebras and combining different algebras (Frank, 1999). The functional programming paradigm with algebraic specifications was used to carry out a prototypical implementation. Especially high order functions and polymorphism have been utilized to implement the model.

Appendix C

Haskell Code

C.1 Objects - Definitions

```
module Definitions where

-- ---------------------------------------------------------------------------
-- OBJECTS
-- ---------------------------------------------------------------------------

type Object = Int     -- objects are just numbered entities

-- ---------------------------------------------------------------------------
-- EXPERIMENTS
-- ---------------------------------------------------------------------------

type ID         = Int
type Time       = Int
type ValExp     = Float
type Affordance = String
type Exp        = (Object,Object,Bool)
type ExpSerie   = [Exp]
data Observation = O [Affordance] ExpSerie deriving Show

maxNumberIterations = 4

-- ---------------------------------------------------------------------------
-- THEORIES
-- ---------------------------------------------------------------------------

type Weight    = Float
type Operation = (Object->Object->Bool)

data Theory   = T ID [Affordance] Operation Weight deriving Show
type Theories = [Theory]
```

```
instance Eq Theory where
   (==)  (T i a op w) (T i' a' op' w') = i == i'

data Redescription1 = R1 deriving (Eq,Show) -- stages of theories
data Redescription2 = R2 deriving (Eq,Show)
data Redescription3 = R3 deriving (Eq,Show)
data Redescription4 = R4 deriving (Eq,Show)
data Redescription5 = R5 deriving (Eq,Show)

data Redescription6 = LooseFit deriving (Show)
data Redescription7 = TightFit deriving (Show)

-- ---------------------------------------------------------------------------
-- Access to Observations and Theories
-- ---------------------------------------------------------------------------
class Observations obs where
    getObsAff   :: obs -> [Affordance]
    getExpSerie :: obs -> ExpSerie

instance Observations Observation where
    getObsAff (O a e)   = a
    getExpSerie (O a e) = e

instance Observations [Observation] where
    getObsAff list =  concat [getObsAff o |o<-list]
    getExpSerie list = concat [getExpSerie o |o<-list]

class Th theory where
    getTAff   :: theory -> [Affordance]

instance Th Theory where
    getTAff (T i a o w)   = a


instance (Th t) => Th [t] where
    getTAff list   = concat [getTAff t1 | t1<-list]

getWeight :: Theory -> Weight
getWeight (T i a o w) = w

getOpList :: [Theory] -> [(Object->Object->Bool)]
getOpList tlist = [op|(T i a op w)<-tlist]

-- ----------------------------------------------------------------------------
-- useful auxilliary functions
-- ----------------------------------------------------------------------------

-- 'showing' a function
instance Show (a->b) where
    show x = "Function"
```

```
showlist :: (Show a) => [a] -> String
showlist []    = ""
showlist list = show (head list) ++ "\n" ++ showlist (tail list)

inSide ::(Eq a) => [a] -> [a] -> Bool
inSide alist blist = and [(elem b alist)|b<-blist]
```

C.2 Support of Objects

```
-- ----------------------------------------------------------------------------
--                       S U P P O R T   T H E O R I E S
-- ----------------------------------------------------------------------------
{- File: Support.hs, Support of Objects
   Autor: Florian Twaroch
   Relevant Experiments: Luo & Baillargeon 2004
   Date: 18.10.2006
   Revision: -
-}
-- ----------------------------------------------------------------------------
--                       S U P P O R T   T H E O R I E S
-- ----------------------------------------------------------------------------
module Support where

import Definitions



class Supporters r o where
    isOn :: r -> o -> o -> Bool

-- 1st Theory considers just the contact between the objects
instance Supporters Redescription1 Object where
        isOn R1 o1 o2 = getContact o1 o2

-- ----------------------------------------------------------------------------
--                            REPLACING THE THEORY
-- ----------------------------------------------------------------------------

-- 2nd Theory specifies the type of contact
instance Supporters Redescription2 Object where
        isOn R2 o1 o2 = isOn R1 o1 o2 &&
                        hasTopContact o1 o2


-- 3rd Theory considers the amount of contact
instance Supporters Redescription3 Object where
        isOn R3 o1 o2 = isOn R2 o1 o2 &&
                        hasSupportiveAmountContact o1 o2

-- 4th Theory considers the shape of the supported object
instance Supporters Redescription4 Object where
        isOn R4 o1 o2 = isOn R3 o1 o2 &&
                        hasSupportiveShape o2


-- ----------------------------------------------------------------------------
--                  P E R C E P T I O N S // T E S T D A T A
-- ----------------------------------------------------------------------------
-- Support Experiment
-- conflicts in contact, amount and shape necessary

e_support, e_support1,e_support2,e_support3,e_support4 :: ExpSerie

e_support = [(1,2,False)]  -- contradict, conflict contact

e_support1 = [(3,4,False)] -- contradict C1, conflict TopContact

e_support2 = [(5,6,False)] -- contradict C2, conflict amount

e_support3 = [(7,8,False)] -- contradict C3, conflict shape

e_support4 = [(9,10,True)] -- input set free of contradictions

-- ---------------------------------------------------------------------------
--      PERCEPTIONS - Support, these are access operations to sensors.
-- ---------------------------------------------------------------------------

-- 1,2 conflict Contact
getContact 3 4   = True
getContact 5 6   = True
getContact 7 8   = True
getContact 9 10  = True
getContact _ _   = False

-- 3,4 conflict TopContact
hasTopContact 5 6   = True
hasTopContact 7 8   = True
hasTopContact 9 10  = True
hasTopContact _ _   = False

-- 5,6 conflict Amount
hasSupportiveAmountContact 3 4   = True
hasSupportiveAmountContact 7 8   = True
hasSupportiveAmountContact 9 10  = True
hasSupportiveAmountContact _ _   = False

-- 7,8 conflict Shape
hasSupportiveShape 2   = True
hasSupportiveShape 4   = True
hasSupportiveShape 6   = True
hasSupportiveShape 10  = True
hasSupportiveShape _   = False

-- ----------------------------------------------------------------------------
--                      S U P P O R T   T H E O R I E S
-- ----------------------------------------------------------------------------
aff :: [Affordance]
aff = ["Contact", "TopContact", "AmountContact","Shape",""]
```

```
theories :: Theories
theories = [T 10 (take 1 aff) (isOn R1) 0.0,
            T 11 (take 2 aff) (isOn R2) 0.0,
            T 12 (take 3 aff) (isOn R3) 0.0,
            T 13 (take 4 aff) (isOn R4) 0.0
            ]

testdata :: [Observation]
testdata = [O [aff !! 0] e_support,
            O [aff !! 1] e_support1,
            O [aff !! 2] e_support2,
            O [aff !! 3] e_support3,
            O [] e_support4             -- nub duplicates in affordances
            ]

-- ----------------------------------------------------------------------------
--                   E N D   S U P P O R T   T H E O R I E S
-- ----------------------------------------------------------------------------
```

C.3 Occlusion of Objects

```
-- ---------------------------------------------------------------------------
--                      O C C L U S I O N  T H E O R I E S
-- ---------------------------------------------------------------------------
{- File: Occlusion.hs, Occlusion of Objects
   Autor: Florian Twaroch
   Relevant Experiments: Luo & Baillargeon 2004
   Date: 18.10.2006
   Revision: -
-}
-- ---------------------------------------------------------------------------
--                      O C C L U S I O N  T H E O R I E S
-- ---------------------------------------------------------------------------

module Occlusion where

import Definitions



class Occluders r o where
    isOccluded :: r -> o -> o -> Bool

-- 1st Theory: An object behind another is occluded.
instance Occluders Redescription1 Object where
    isOccluded R1 o1 occ = isBehind o1 occ

-- 2nd Theory: Occluders with windows do not work.
instance Occluders Redescription2 Object where
    isOccluded R2 o1 occ = isOccluded R1 o1 occ &&
                           (not $ hasWindow occ)


-- 3rd Theory: The size of the occluded object matters.
instance Occluders Redescription3 Object where
    isOccluded R3 o1 occ = isOccluded R2 o1 occ &&
                           o1 < occ

-- 4th Theory: A transparent occcluder does not hide the occluded object.
instance Occluders Redescription4 Object where
    isOccluded R4 o1 occ = isOccluded R3 o1 occ &&        -- Spatial Relation
                           isTransparent' R2 occ           -- windows, Baillargeon 2004
                                                           -- dimensions,width & height
-- ---------------------------------------------------------------------------
--                  GENERALIZING TO A NEW THEORY OF TRANSPARENCY
-- ---------------------------------------------------------------------------
class Transparency r o where
isTransparent' :: r -> o -> Bool

-- Theory of Transparency
```

```
instance Transparency Redescription1 Object where
isTransparent' R1 o1 = isTransparent o1

-- Theory of nonTransparency
instance Transparency Redescription2 Object where
isTransparent' R2 o1 = not $ isTransparent o1


-- ----------------------------------------------------------------------------
--                  SPECIALIZING THE OCCLUSION THEORY
-- ----------------------------------------------------------------------------

-- 5th theory: The occluder moved and the occluder and the object
-- share the same pos
instance Occluders Redescription5 Object where
    isOccluded R5 o1 occ = isOccluded R4 o1 occ &&
   (not $ (moved occ && (not $ samePos o1 occ)))

-- ----------------------------------------------------------------------------
--                  P E R C E P T I O N S // T E S T D A T A
-- ----------------------------------------------------------------------------

-- Occlusion Experiments
-- conflicts in windows, size, transparency and movement are necessary

e_occlusion, e_occlusion1,e_occlusion2,e_occlusion3,e_occlusion4 :: ExpSerie

e_occlusion  = [(1,2,True)]

e_occlusion1 = [(3,4,False)]  -- contradict C1, conflict window

e_occlusion2 = [(6,5,False)]  -- contradict C2, conflict size 6 is bigger than 4

e_occlusion3 = [(7,8,False)]  -- contradict C3, conflict transparency

e_occlusion4 = [(9,10,False)] -- contradict C4, conflict movement

-- ---------------------------------------------------------------------------
--       PERCEPTIONS - Occlusion, these are access operations to sensors.
-- ---------------------------------------------------------------------------

isBehind  1 2  = True
isBehind  3 4  = True
isBehind  6 5  = True
isBehind  7 8  = True
isBehind  9 10 = True
isBehind  _ _ = False


hasWindow 4 = True
hasWindow _ = False
```

```
isTransparent 8 = True
isTransparent _ = False

-- did the object moved
moved 10 = True
moved _  = False

-- position after movement
samePos 1 2 = True
samePos 3 4 = True
samePos 6 5 = True
samePos 7 8 = True
samePos _ _ = False


-- ---------------------------------------------------------------------------
--                      O C C L U S I O N  T H E O R I E S
-- ---------------------------------------------------------------------------
-- This list describes what the objects afford
aff :: [Affordance]
aff = ["isBehind",
        "hasWindow",
        "size",
        "isTransparent",
        "movedSamePos"
      ]

theories :: Theories
theories = [T 1 (take 1 aff) (isOccluded R1) 0.0,
            T 2 (take 2 aff) (isOccluded R2) 0.0,
            T 3 (take 3 aff) (isOccluded R3) 0.0,
            T 4 (take 4 aff) (isOccluded R4) 0.0,
            T 5 (take 5 aff) (isOccluded R5) 0.0 ]

testdata :: [Observation]
testdata = [O [aff !! 0] e_occlusion,      -- type of observations
            O [aff !! 1] e_occlusion1,     -- describes just the type of percept
            O [aff !! 2] e_occlusion2,
            O [aff !! 3] e_occlusion3,
            O [aff !! 4] e_occlusion4 ]

-- ---------------------------------------------------------------------------
--                   E N D  O C C L U S I O N  T H E O R I E S
-- ---------------------------------------------------------------------------
```

C.4 Coverage of Objects

```
-- ---------------------------------------------------------------------------
--                       C O V E R   T H E O R I E S
-- ---------------------------------------------------------------------------
{- File: Support.hs, Support of Objects
   Autor: Florian Twaroch
   Relevant Experiments: Luo & Baillargeon 2004
   Date: 18.10.2006
   Revision: -
-}
-- ---------------------------------------------------------------------------
--                       C O V E R   T H E O R I E S
-- ---------------------------------------------------------------------------

module Cover where


import Definitions



class Covers c o where
    isUnder :: c -> o -> o -> Bool

-- 1. Theory: The object under the cover moves iff the cover is down.
instance Covers Concept1 Object where
    isUnder C1 o1 cover = covered o1 cover &&
                          ( (not $ moved cover) ||
                            ((not $ liftedBeforeMoved cover) && samePosAfter o1 cover)
                          )

-- 2nd. Theory: The object under the cover iff it is smaller than the cover.
instance Covers Concept2 Object where
    isUnder C2 o1 cover = isUnder C1 o1 cover &&
                          (o1 < cover)

-- 3rd. Theory: The object is hidden under the cover iff the cover is not transparent
--              or the previous axioms
instance Covers Concept3 Object where
    isUnder C3 obj cover = (not $ isTransparent cover) &&
                           isUnder C2 obj cover


-- ---------------------------------------------------------------------------
--              P E R C E P T I O N S // T E S T D A T A
-- ---------------------------------------------------------------------------

-- Cover Experiments
e_cover, e_cover1, e_cover2, e_cover3 :: ExpSerie
```

```
e_cover  = [(1,2,True)]

e_cover1 = [(4,3,False)]

e_cover2 = [(5,6,False)]

e_cover3 = [(7,8,True)]

-- ---------------------------------------------------------------------------
--       PERCEPTIONS - Support, these are access operations to sensors.
-- ---------------------------------------------------------------------------

covered 1 2 = True
covered 4 3 = True
covered 5 6 = True
covered 7 8 = True
covered _ _ = False


moved 2 = True
moved 3 = False
moved 6 = False
moved 8 = False
moved _ = error "no perception defined"

liftedBeforeMoved 2   = False
liftedBeforeMoved 3   = True
liftedBeforeMoved 6   = True
liftedBeforeMoved 8   = True
liftedBeforeMoved  _  = error "no perception defined"


samePosAfter 1 2 = True
samePosAfter _ _ = error "no perception defined"


isTransparent 6 = True
isTransparent _ = False

-- ----------------------------------------------------------------------------
--                      C O V E R   T H E O R I E S
-- ----------------------------------------------------------------------------
aff :: [Affordance]
aff = [""
      ]

theories :: [([Affordance],(Object->Object->Bool))]
theories = [(aff,isUnder C1),
            (aff,isUnder C2),
            (aff,isUnder C3)]
```

```
testdata :: [([Affordance],ExpSerie)]
testdata = [(aff,e_cover),
            (aff,e_cover1),
            (aff,e_cover2),
            (aff,e_cover3)]

-- ----------------------------------------------------------------------------
--                    E N D   C O V E R   T H E O R I E S
-- ----------------------------------------------------------------------------
o1,o2 :: Object
o1 = 4
o2 = 3

test :: Bool
test = isUnder C1 o1 o2
```

C.5 Containment of Objects

```
-- ---------------------------------------------------------------------------
--                   C O N T A I N M E N T   T H E O R I E S
-- ---------------------------------------------------------------------------
{- File: Containment.hs, Containment of Rigid Objects
   Autor: Florian Twaroch
   Relevant Experiments: Luo & Baillargeon 2004
   Date: 18.10.2006
   Revision: -
-}
-- ---------------------------------------------------------------------------
--                   C O N T A I N M E N T   T H E O R I E S
-- ---------------------------------------------------------------------------

module Containment where


import Definitions


class Containment r o where
    isIn  :: r -> o -> o -> Bool

-- 1st Theory: the container has to be open.
instance Containment Redescription1 Object where
    isIn R1 obj co = isOpen co

-- 2nd Theory: The object in the container shares the movement with the container
instance Containment Redescription2 Object where
    isIn R2 obj co = isIn R1 obj co &&
                     ((not $ moved co) || (moved co && samePosAfter obj co))

-- 3rd Theory: The contained object has to be smaller.
instance Containment Redescription3 Object where
    isIn R3 obj co = isIn R2 obj co &&
                     (obj < co)

-- LooseFit Containment
instance Containment Redescription4 Object where
    isIn R4 obj co = isIn R3 obj co &&
                     fit R1 obj co

-- Tightfit Containment
instance Containment Redescription5 Object where
    isIn R5 obj co = isIn R3 obj co &&
                     fit R2 obj co


-- ---------------------------------------------------------------------------
--                            GENERALIZING TO A NEW THEORY
-- ----------------------------------------------------------------------------
-- Loose and tight fit containment require Fit as generalized super class

class Fit c o where
    fit :: c -> o -> o -> Bool

instance Fit Redescription1 Object where
   fit R1 o1 o2 = moveable o1 o2

instance Fit Redescription2 Object where
   fit R2 o1 o2 = not $ moveable o1 o2


-- ----------------------------------------------------------------------------
--                  P E R C E P T I O N S // T E S T D A T A
-- ----------------------------------------------------------------------------

-- Support Experiment
-- conflicts in size, amount and shape necessary

e_containment, e_containment1, e_containment2 :: ExpSerie
e_containmentLR1, e_containmentTR1 :: ExpSerie

e_containment    = [(1,2,True)]

e_containment1   = [(3,4,False)] -- contradict R1, conflict movement

e_containment2   = [(6,5,False)] -- contradict R2, conflict size

e_containmentLR1 = replicate 3 (7,8,False)  -- contradict R3, conflict moveable inside

e_containmentTR1 = replicate 10 (9,10,True)

-- ---------------------------------------------------------------------------
--        PERCEPTIONS - Support, these are access operations to sensors.
-- ---------------------------------------------------------------------------

isOpen 2 = True
isOpen 4 = True
isOpen 5 = True
isOpen 8 = True
isOpen 10 = True
isOpen _ = False -- error "Container has no opening!"

-- did the object move
moved 4  = True
moved 8  = True
moved 10 = True
moved _  = False

-- position after movement
```

```
samePosAfter 1 2  = True
samePosAfter 4 3  = False
samePosAfter 6 5  = True
samePosAfter 7 8  = True
samePosAfter 9 10 = True
samePosAfter _ _  = False

-- moveable inside container
moveable 1 2  = True
moveable 3 4  = True
moveable 6 5  = True
moveable 9 10 = False
moveable _ _  = False

-- ----------------------------------------------------------------------------
--                    C O N T A I N M E N T   T H E O R I E S
-- ----------------------------------------------------------------------------
aff :: [Affordance]
aff = ["isOpen",
       "movement",
       "size",
       "loosefit",
       "tightfit"
      ]

theories :: Theories
theories = [T 21 (take 1 aff) (isIn R1) 0.0,
            T 22 (take 2 aff) (isIn R2) 0.0,
            T 23 (take 3 aff) (isIn R3) 0.0,
            T 24 (take 4 aff) (isIn R4) 0.0,
            T 25 (take 5 aff) (isIn R5) 0.0
           ]

testdata :: [Observation]
testdata = [O [aff !! 0] e_containment,
            O [aff !! 1] e_containment1,
            O [aff !! 2] e_containment2,
            O [aff !! 3] e_containmentLR1,
            O [aff !! 4] e_containmentTR1
            ]

-- ----------------------------------------------------------------------------
--                  E N D   C O N T A I N M E N T   T H E O R I E S
-- ----------------------------------------------------------------------------
```

C.6 Balance Scale Task

```
{-
Author: Florian Twaroch
Topic: Sandbox Geography, PhD
       Balance Scale Task
       relevant experiments: Siegler 1976, Siegler 1983, etc.
Date: August 2005
Revision: 09.12.2005

Comment:

       Robert Sieglers Balance Scale experiment (1976) identified four
       naive rules for a fulcrum to be in balance. Children might experience
       the balance scale on a playground on a swing.

                     0==0==0==0===o===0==0==0==0
                                  |
                                 |||

                     fulcrum with 0 (1,2,..) weights

       Axioms:: empty fulcrum == Balanced
       When testing ensure that left and right arm have equal length !!!

-}

module BalanceScale where

import Definitions


type Left = [Int]
type Right = [Int]
data Fulcrum = F Left Right

-- Null stands for guessing the solution
data Side = LeftSide | RightSide | Balanced | Null deriving (Eq,Show)

class Fulcrums c where
   tip::c->Fulcrum->Side->Bool

instance Show Fulcrum where
show (F l r) = show (reverse l) ++ "-o-" ++ show r


instance Fulcrums Fulcrum where
   tip c (F [] []) side = Balanced == side


-- -----------------------------------------------------------------------------
-- Concept I starts with an initial rule about balance scales based on the
-- weight of the items piled on the fulcrum. An alternative would
-- be to conceptualize the distance instead of the weight as the influencing
-- element.
-- ----------------------------------------------------------------------------
instance Fulcrums Concept1 where
   tip C1 (F l r) side = weight (F l r) == side

instance Fulcrums Concept2 where
   tip C2 (F l r) side = distance (F l r) == side

-- the weight is determined by summing up the items seen on each side
-- of the fulcrum
weight::Fulcrum->Side
weight (F l r)
    |sum l > sum r  = LeftSide
    |sum l < sum r  = RightSide
    |otherwise  = Balanced

distance::Fulcrum->Side
distance (F l r)
    |dist l > dist r    = LeftSide
    |dist l < dist r    = RightSide
    |otherwise          = Balanced

-- determines the length of a list, depending on the position
-- of the weight, maximal outer position
dist::[Int]->Int
dist [] = 0
dist l
    |last l == 0 = dist(init l)
    |otherwise = length l

-- ----------------------------------------------------------------------------
-- Concept III is a generalization step out of the concepts I and II.
-- Both weight and distance influence the balance scale.
-- ----------------------------------------------------------------------------
instance Fulcrums Concept3 where
   tip C3 (F l r) side
    |weight (F l r) == Balanced = distance (F l r) == side
    |otherwise                  = weight (F l r) == side


-- ----------------------------------------------------------------------------
-- Concept IV is a trial to create a more complex naive theory by introducing
-- special cases (specialization) and generalization steps.
-- The theory makes use of guessing, that is why it does not perform as good as
-- naive theory III in the given test cases.
-- -- ----------------------------------------------------------------------------
instance Fulcrums Concept4 where
   tip C4 (F l r) side
```

```
    |weight (F l r) == Balanced = distance (F l r) == side
    |otherwise                  = testWeight (F l r) == side

-- Because distance alone does not lead to a decision
-- weight must be tested once again.
testWeight::Fulcrum->Side
testWeight (F l r)
    |distance (F l r) == Balanced = weight (F l r)
    |otherwise                    = testDistanceANDWeight (F l r)

testDistanceANDWeight::Fulcrum->Side
testDistanceANDWeight (F l r)
    |distance (F l r) == weight (F l r) = weight (F l r)
    |otherwise                          = Null  -- can not decide and must guess

-- ---------------------------------------------------------------------------
-- Concept5 represents the torque rule to describe weights on a fulcrum.
-- To develop the concept measurement is required.
-- ---------------------------------------------------------------------------
instance Fulcrums Concept5 where
   tip C5 (F l r) side = cross (F l r) == side

-- The rule should be expressed in terms of weight and distance.
cross::Fulcrum->Side
cross (F l r)
| cp l > cp r = LeftSide            -- cp ... the cross product of mass and distance
| cp l < cp r = RightSide
| otherwise   = Balanced where
cp::[Int]->Int
cp ls = sum (zipWith (*) ls [1..])

-- ---------------------------------------------------------------------------
--                  P E R C E P T I O N S // T E S T D A T A
-- ---------------------------------------------------------------------------

-- test cases according to Siegler 1976

f1,f2,f3,f4,f5,f6 :: Fulcrum

f1 = F [2,1,0,0] [2,1,0,0] -- Balance
f2 = F [0,2,1,0] [1,1,0,0] -- Weight
f3 = F [0,0,3,0] [0,3,0,0] -- Distance
f4 = F [0,2,2,0] [0,0,0,2] -- Conflict Weight
f5 = F [0,0,3,0] [2,3,0,0] -- Conflict Distance
f6 = F [0,3,0,0] [6,0,0,0] -- Conflict Balance

-- --------------------------------------------------------------------------
--       PERCEPTIONS - Support, these are access operations to sensors.
-- --------------------------------------------------------------------------

-- observations
```

```
o1,o2,o3,o4,o5,o6::[(Fulcrum,Side,Bool)]

o1 = [(f1,Balanced,True)]

o2 = [(f2,LeftSide,True)]

o3 = [(f3,LeftSide,True)]

o4 = [(f4,LeftSide,True)]

o5 = [(f5,LeftSide,True)]

o6 = [(f6,Balanced,True)]

-- ----------------------------------------------------------------------------
--               B A L A N C E   S C A L E   T H E O R I E S
-- ----------------------------------------------------------------------------
aff :: [Affordance]
aff = [""]

theories :: [([Affordance],(Fulcrum->Side->Bool))]
theories = [(aff,tip C1),
            (aff,tip C2),
            (aff,tip C3),
            (aff,tip C4),
            (aff,tip C5)]

testdata :: [([Affordance],[(Fulcrum,Side,Bool)])]
testdata = [(aff,o1),
            (aff,o2),
            (aff,o3),
            (aff,o4),
            (aff,o5),
            (aff,o6)]

-- ----------------------------------------------------------------------------
--           E N D   B A L A N C E   S C A L E   T H E O R I E S
-- ----------------------------------------------------------------------------
```

C.7 Simulation

```
module NewSim where

import Definitions

import Support
import Occlusion
import Containment

import List

-- States of the agent
data State = Observe | Build | Test | Use | GiveUp deriving (Show,Eq)
type Potential = Theories  -- The agent could theoretically build up all possible theories.
                           -- The model is simplified here as I did not automate the process
                           -- of theory building.
                           -- Building theories means to choose out of the potential.

data Agent = Agent ID Observation Theories Potential State

class Agents ag where
    observe          :: Observation -> ag -> ag    -- get observations of the environment
    buildTheories    :: ag -> ag
    testTheories     :: ag -> ag
    useTheories      :: ag -> ag

instance Agents Agent where
    observe (O aff []) (Agent iD o t p s)  = (Agent iD (O aff []) t p GiveUp)
    observe exps (Agent iD o t p Observe) = (Agent iD exps t p Test)
    observe exps (Agent iD o t p Build)    = (Agent iD o t p Build)

    -- The potential desribes all possible theories the agent could make
    -- through an inferencing process. They are hard coded and not build.
    -- The agent chooses among them.

    buildTheories (Agent iD o t p Build)
       |newt == [] = (Agent iD o t p GiveUp) -- error "No new theories detected !"
       |otherwise  = (Agent iD o (t++newt) p Test) where
            t' = [pt|pt<-p, (inSide (getObsAff o) (getTAff pt))]
            newt = [nt|nt<-t',not (elem nt t)]  -- New theories are just those
    buildTheories ag = ag                       -- that the agent haven't had before.

    -- testing observed data with current theory
    testTheories (Agent iD o t p Test) = (Agent iD o wt p newState) where
                        newState  = determineState wt
                        -- weighted theories
                        wt = [(T i a op (evalExp op testdata'))|(T i a op w)<-t]
                        -- need as much testdata as theories
                        testdata'' = replicate (length t) testdata'
                        testdata' = getExpSerie o
```

```haskell
    testTheories ag = ag

    useTheories (Agent iD o t p s)
       |s == Use      = (Agent iD o t p Observe)       -- apply theory to novel data
       |s == Build    = (Agent iD o t p Build)         -- build a new theory
       |s == GiveUp   = error "No percepts or theories found that fit the data."




instance Show Agent where
    show (Agent iD obs theories p state) = "............................\n" ++
                                         "Agent id: " ++ show (iD) ++ "\n" ++
                                         "Detected percepts:" ++ showlist (getObsAff obs) ++ "\n" ++
                                         "Observations: " ++ show (getExpSerie obs) ++ "\n" ++
                                         "Current theories: " ++ showlist (theories) ++ "\n" ++
                                         "State: " ++ show state ++ "\n" ++
                                         "............................\n"

-- -----------------------------------------------------------------------------
--                       E V A L U A T I O N   M E T H O D
-- -----------------------------------------------------------------------------

class (Show a, Show b) => Evaluation a b where
    testExp  :: a -> b -> b
    testExps :: a -> [b] -> [b]       -- Evaluates a serie of experiments.
    testExps f e = map (testExp f) e

    evalExp  :: a -> [b] -> Float     -- Number of hits in an experiment is a
                                      -- utility score -> 0.0 - 1.0


-- Evaluation of Experiments having two objects, mapped on an equivalence class

instance Evaluation (Object->Object->Bool) (Object,Object,Bool) where
    testExp f e@(o1,o2,observation) = (o1,o2,evidence) where
        evidence                = observation == belief
        belief                  = f o1 o2

    evalExp f e = no_hits / no_exps where
        no_hits = fromIntegral (length hits)
        no_exps = fromIntegral (length e)
        hits = [(o1,o2,t)|(o1,o2,t)<-(testExps f e),t==True]


-- Theories that fully explain the world are immediatelly
-- without any efforts undertaken to search for another theory (explaination).
determineState :: [Theory] -> State
determineState [] = error "Can not find any theory that fits the given data."
determineState t    -- Agent holds a theory that fully explains the given data.
    |any (==1.0) $ map (getWeight) t = Use       -- therefore no need to change
    |otherwise                        = Build     -- otherwise tries a better explaination.


-- ---------------------------------------------------------------------------
--                             S I M U L A T I O N
-- ---------------------------------------------------------------------------

ag1 :: Agent
ag1 = Agent 1
            (O [] [])                  -- no observations
            [Occlusion.theories !! 0]  -- holds unweighted theories
            Occlusion.theories         -- potentially possible theories
            Observe                    -- state: willing to explore


ag2 :: Agent
ag2 = Agent 2
            (O [] [])                  -- no observations
            [Support.theories !! 0]    -- holds unweighted theories
            Support.theories           -- potentially possible theories
            Observe                    -- state: willing to explore


ag3 :: Agent
ag3 = Agent 3
            (O [] [])                    -- no observations
            [Containment.theories !! 0]  -- holds unweighted theories
            Containment.theories         -- potentially possible theories
            Observe                      -- state: willing to explore


sim :: Time -> [Observation] -> Agent -> String
sim time [] ag = error "End of Simulation - no data in the environment"
sim time obs ag
    |time > (length obs + 2) = "Agent getting bored - no new data\n" ++
                              "CHECK Simulation time limit"
    |otherwise             = show "Time: " ++ show (time') ++ "\n" ++
                              "----------------------------\n" ++
                              "observing environment ...   \n" ++
                              "----------------------------\n" ++
                              show (observe obs' ag) ++
                              "----------------------------\n" ++
                              "building theories ...       \n" ++
                              "----------------------------\n" ++
                               show (buildTheories $ (observe obs' ag)) ++
                              "----------------------------\n" ++
                              "testing theories ...        \n" ++
                              "----------------------------\n" ++
                               show (testTheories $ buildTheories
                                                   $ (observe obs' ag)) ++
                              "----------------------------\n" ++
                               "using theories ...           \n" ++
                               "-----------------------------\n" ++
                                show (useTheories $ testTheories
                                                  $ buildTheories
                                                  $ (observe obs' ag)) ++
                               "-----------------------------\n" ++
                               (sim time' obs ag') where
                                   ag' = useTheories $ testTheories $ buildTheories $ (observe obs' ag)
                                   time' = time + 1
                                   obs' = (O r_aff r_obs)          -- reduce the data
                                   r_aff = take time' (getObsAff obs)
                                   r_obs = take time' (getExpSerie obs)

-- Simulation of Occlusion of Solids
f = putStrLn (sim 0 (Occlusion.testdata) ag1)

-- Simulation of Support of Solids
l = putStrLn (sim 0 (Support.testdata) ag2)

-- Simulation of Containment of Solids
o = putStrLn (sim 0 (Containment.testdata) ag3)
```

Appendix D

Simulation Output

D.1 Simulation - Theories “Occlusion of Solids”

```
*NewSim> simulate "Occlusion"
Loading package haskell98-1.0 ... linking ... done.
"Time: "1
----------------------------
observing environment ...
----------------------------
............................
Agent id: 1
Detected percepts:"isBehind"

Observations: [(1,2,True)]
Current theories: T 1 ["isBehind"] Function 0.0

State: Test
............................
----------------------------
building theories ...
----------------------------
............................
Agent id: 1
Detected percepts:"isBehind"

Observations: [(1,2,True)]
Current theories: T 1 ["isBehind"] Function 0.0

State: Test
............................
----------------------------
testing theories ...
----------------------------
............................
```

```
Agent id: 1
Detected percepts:"isBehind"

Observations: [(1,2,True)]
Current theories: T 1 ["isBehind"] Function 1.0

State: Use
............................
----------------------------
using theories ...
----------------------------
............................
Agent id: 1
Detected percepts:"isBehind"

Observations: [(1,2,True)]
Current theories: T 1 ["isBehind"] Function 1.0

State: Observe
............................
----------------------------
"Time: "2
----------------------------
observing environment ...
----------------------------
............................
Agent id: 1
Detected percepts:"isBehind"
"hasWindow"

Observations: [(1,2,True),(3,4,False)]
Current theories: T 1 ["isBehind"] Function 1.0

State: Test
............................
----------------------------
building theories ...
----------------------------
............................
Agent id: 1
Detected percepts:"isBehind"
"hasWindow"

Observations: [(1,2,True),(3,4,False)]
Current theories: T 1 ["isBehind"] Function 1.0

State: Test
............................
----------------------------
testing theories ...
----------------------------
```

```
............................
Agent id: 1
Detected percepts:"isBehind"
"hasWindow"

Observations: [(1,2,True),(3,4,False)]
Current theories: T 1 ["isBehind"] Function 0.5

State: Build
............................
----------------------------
using theories ...
----------------------------
............................
Agent id: 1
Detected percepts:"isBehind"
"hasWindow"

Observations: [(1,2,True),(3,4,False)]
Current theories: T 1 ["isBehind"] Function 0.5

State: Build
............................
----------------------------
"Time: "3
----------------------------
observing environment ...
----------------------------
............................
Agent id: 1
Detected percepts:"isBehind"
"hasWindow"

Observations: [(1,2,True),(3,4,False)]
Current theories: T 1 ["isBehind"] Function 0.5

State: Build
............................
----------------------------
building theories ...
----------------------------
............................
Agent id: 1
Detected percepts:"isBehind"
"hasWindow"

Observations: [(1,2,True),(3,4,False)]
Current theories: T 1 ["isBehind"] Function 0.5
T 2 ["isBehind","hasWindow"] Function 0.0

State: Test
```

```
............................
----------------------------
testing theories ...
----------------------------
............................
Agent id: 1
Detected percepts:"isBehind"
"hasWindow"

Observations: [(1,2,True),(3,4,False)]
Current theories: T 1 ["isBehind"] Function 0.5
T 2 ["isBehind","hasWindow"] Function 1.0

State: Use
............................
----------------------------
using theories ...
----------------------------
............................
Agent id: 1
Detected percepts:"isBehind"
"hasWindow"

Observations: [(1,2,True),(3,4,False)]
Current theories: T 1 ["isBehind"] Function 0.5
T 2 ["isBehind","hasWindow"] Function 1.0

State: Observe
............................
----------------------------
"Time: "4
----------------------------
observing environment ...
----------------------------
............................
Agent id: 1
Detected percepts:"isBehind"
"hasWindow"
"size"

Observations: [(1,2,True),(3,4,False),(6,5,False)]
Current theories: T 1 ["isBehind"] Function 0.5
T 2 ["isBehind","hasWindow"] Function 1.0

State: Test
............................
----------------------------
building theories ...
----------------------------
............................
Agent id: 1
```

```
Detected percepts:"isBehind"
"hasWindow"
"size"

Observations: [(1,2,True),(3,4,False),(6,5,False)]
Current theories: T 1 ["isBehind"] Function 0.5
T 2 ["isBehind","hasWindow"] Function 1.0

State: Test
............................
----------------------------
testing theories ...
----------------------------
............................
Agent id: 1
Detected percepts:"isBehind"
"hasWindow"
"size"

Observations: [(1,2,True),(3,4,False),(6,5,False)]
Current theories: T 1 ["isBehind"] Function 0.33333334
T 2 ["isBehind","hasWindow"] Function 0.6666667

State: Build
............................
----------------------------
using theories ...
----------------------------
............................
Agent id: 1
Detected percepts:"isBehind"
"hasWindow"
"size"

Observations: [(1,2,True),(3,4,False),(6,5,False)]
Current theories: T 1 ["isBehind"] Function 0.33333334
T 2 ["isBehind","hasWindow"] Function 0.6666667

State: Build
............................
----------------------------
"Time: "5
----------------------------
observing environment ...
----------------------------
............................
Agent id: 1
Detected percepts:"isBehind"
"hasWindow"
"size"
```

```
Observations: [(1,2,True),(3,4,False),(6,5,False)]
Current theories: T 1 ["isBehind"] Function 0.33333334
T 2 ["isBehind","hasWindow"] Function 0.6666667

State: Build
............................
----------------------------
building theories ...
----------------------------
............................
Agent id: 1
Detected percepts:"isBehind"
"hasWindow"
"size"

Observations: [(1,2,True),(3,4,False),(6,5,False)]
Current theories: T 1 ["isBehind"] Function 0.33333334
T 2 ["isBehind","hasWindow"] Function 0.6666667
T 3 ["isBehind","hasWindow","size"] Function 0.0

State: Test
............................
----------------------------
testing theories ...
----------------------------
............................
Agent id: 1
Detected percepts:"isBehind"
"hasWindow"
"size"

Observations: [(1,2,True),(3,4,False),(6,5,False)]
Current theories: T 1 ["isBehind"] Function 0.33333334
T 2 ["isBehind","hasWindow"] Function 0.6666667
T 3 ["isBehind","hasWindow","size"] Function 1.0

State: Use
............................
----------------------------
using theories ...
----------------------------
............................
Agent id: 1
Detected percepts:"isBehind"
"hasWindow"
"size"

Observations: [(1,2,True),(3,4,False),(6,5,False)]
Current theories: T 1 ["isBehind"] Function 0.33333334
T 2 ["isBehind","hasWindow"] Function 0.6666667
T 3 ["isBehind","hasWindow","size"] Function 1.0
```

```
State: Observe
............................
----------------------------
"Time: "6
----------------------------
observing environment ...
----------------------------
............................
Agent id: 1
Detected percepts:"isBehind"
"hasWindow"
"size"
"isTransparent"

Observations: [(1,2,True),(3,4,False),(6,5,False),(7,8,False)]
Current theories: T 1 ["isBehind"] Function 0.33333334
T 2 ["isBehind","hasWindow"] Function 0.6666667
T 3 ["isBehind","hasWindow","size"] Function 1.0

State: Test
............................
----------------------------
building theories ...
----------------------------
............................
Agent id: 1
Detected percepts:"isBehind"
"hasWindow"
"size"
"isTransparent"

Observations: [(1,2,True),(3,4,False),(6,5,False),(7,8,False)]
Current theories: T 1 ["isBehind"] Function 0.33333334
T 2 ["isBehind","hasWindow"] Function 0.6666667
T 3 ["isBehind","hasWindow","size"] Function 1.0

State: Test
............................
----------------------------
testing theories ...
----------------------------
............................
Agent id: 1
Detected percepts:"isBehind"
"hasWindow"
"size"
"isTransparent"

Observations: [(1,2,True),(3,4,False),(6,5,False),(7,8,False)]
Current theories: T 1 ["isBehind"] Function 0.25
```

```
T 2 ["isBehind","hasWindow"] Function 0.5
T 3 ["isBehind","hasWindow","size"] Function 0.75

State: Build
............................
----------------------------
using theories ...
----------------------------
............................
Agent id: 1
Detected percepts:"isBehind"
"hasWindow"
"size"
"isTransparent"

Observations: [(1,2,True),(3,4,False),(6,5,False),(7,8,False)]
Current theories: T 1 ["isBehind"] Function 0.25
T 2 ["isBehind","hasWindow"] Function 0.5
T 3 ["isBehind","hasWindow","size"] Function 0.75

State: Build
............................
----------------------------
"Time: "7
----------------------------
observing environment ...
----------------------------
............................
Agent id: 1
Detected percepts:"isBehind"
"hasWindow"
"size"
"isTransparent"

Observations: [(1,2,True),(3,4,False),(6,5,False),(7,8,False)]
Current theories: T 1 ["isBehind"] Function 0.25
T 2 ["isBehind","hasWindow"] Function 0.5
T 3 ["isBehind","hasWindow","size"] Function 0.75

State: Build
............................
----------------------------
building theories ...
----------------------------
............................
Agent id: 1
Detected percepts:"isBehind"
"hasWindow"
"size"
"isTransparent"
```

```
Observations: [(1,2,True),(3,4,False),(6,5,False),(7,8,False)]
Current theories: T 1 ["isBehind"] Function 0.25
T 2 ["isBehind","hasWindow"] Function 0.5
T 3 ["isBehind","hasWindow","size"] Function 0.75
T 4 ["isBehind","hasWindow","size","isTransparent"] Function 0.0

State: Test
............................
----------------------------
testing theories ...
----------------------------
............................
Agent id: 1
Detected percepts:"isBehind"
"hasWindow"
"size"
"isTransparent"

Observations: [(1,2,True),(3,4,False),(6,5,False),(7,8,False)]
Current theories: T 1 ["isBehind"] Function 0.25
T 2 ["isBehind","hasWindow"] Function 0.5
T 3 ["isBehind","hasWindow","size"] Function 0.75
T 4 ["isBehind","hasWindow","size","isTransparent"] Function 1.0

State: Use
............................
----------------------------
using theories ...
----------------------------
............................
Agent id: 1
Detected percepts:"isBehind"
"hasWindow"
"size"
"isTransparent"

Observations: [(1,2,True),(3,4,False),(6,5,False),(7,8,False)]
Current theories: T 1 ["isBehind"] Function 0.25
T 2 ["isBehind","hasWindow"] Function 0.5
T 3 ["isBehind","hasWindow","size"] Function 0.75
T 4 ["isBehind","hasWindow","size","isTransparent"] Function 1.0

State: Observe
............................
----------------------------
"Time: "8
----------------------------
observing environment ...
----------------------------
............................
Agent id: 1
```

```
Detected percepts:"isBehind"
"hasWindow"
"size"
"isTransparent"
"movedSamePos"

Observations: [(1,2,True),(3,4,False),(6,5,False),(7,8,False),(9,10,False)]
Current theories: T 1 ["isBehind"] Function 0.25
T 2 ["isBehind","hasWindow"] Function 0.5
T 3 ["isBehind","hasWindow","size"] Function 0.75
T 4 ["isBehind","hasWindow","size","isTransparent"] Function 1.0

State: Test
............................
----------------------------
building theories ...
----------------------------
............................
Agent id: 1
Detected percepts:"isBehind"
"hasWindow"
"size"
"isTransparent"
"movedSamePos"

Observations: [(1,2,True),(3,4,False),(6,5,False),(7,8,False),(9,10,False)]
Current theories: T 1 ["isBehind"] Function 0.25
T 2 ["isBehind","hasWindow"] Function 0.5
T 3 ["isBehind","hasWindow","size"] Function 0.75
T 4 ["isBehind","hasWindow","size","isTransparent"] Function 1.0

State: Test
............................
----------------------------
testing theories ...
----------------------------
............................
Agent id: 1
Detected percepts:"isBehind"
"hasWindow"
"size"
"isTransparent"
"movedSamePos"

Observations: [(1,2,True),(3,4,False),(6,5,False),(7,8,False),(9,10,False)]
Current theories: T 1 ["isBehind"] Function 0.2
T 2 ["isBehind","hasWindow"] Function 0.4
T 3 ["isBehind","hasWindow","size"] Function 0.6
T 4 ["isBehind","hasWindow","size","isTransparent"] Function 0.8

State: Build
```

```
............................
----------------------------
using theories ...
----------------------------
............................
Agent id: 1
Detected percepts:"isBehind"
"hasWindow"
"size"
"isTransparent"
"movedSamePos"

Observations: [(1,2,True),(3,4,False),(6,5,False),(7,8,False),(9,10,False)]
Current theories: T 1 ["isBehind"] Function 0.2
T 2 ["isBehind","hasWindow"] Function 0.4
T 3 ["isBehind","hasWindow","size"] Function 0.6
T 4 ["isBehind","hasWindow","size","isTransparent"] Function 0.8

State: Build
............................
----------------------------
"Time: "9
----------------------------
observing environment ...
----------------------------
............................
Agent id: 1
Detected percepts:"isBehind"
"hasWindow"
"size"
"isTransparent"
"movedSamePos"

Observations: [(1,2,True),(3,4,False),(6,5,False),(7,8,False),(9,10,False)]
Current theories: T 1 ["isBehind"] Function 0.2
T 2 ["isBehind","hasWindow"] Function 0.4
T 3 ["isBehind","hasWindow","size"] Function 0.6
T 4 ["isBehind","hasWindow","size","isTransparent"] Function 0.8

State: Build
............................
----------------------------
building theories ...
----------------------------
............................
Agent id: 1
Detected percepts:"isBehind"
"hasWindow"
"size"
"isTransparent"
"movedSamePos"
```

```
Observations: [(1,2,True),(3,4,False),(6,5,False),(7,8,False),(9,10,False)]
Current theories: T 1 ["isBehind"] Function 0.2
T 2 ["isBehind","hasWindow"] Function 0.4
T 3 ["isBehind","hasWindow","size"] Function 0.6
T 4 ["isBehind","hasWindow","size","isTransparent"] Function 0.8
T 5 ["isBehind","hasWindow","size","isTransparent","movedSamePos"] Function 0.0

State: Test
............................
----------------------------
testing theories ...
----------------------------
............................
Agent id: 1
Detected percepts:"isBehind"
"hasWindow"
"size"
"isTransparent"
"movedSamePos"

Observations: [(1,2,True),(3,4,False),(6,5,False),(7,8,False),(9,10,False)]
Current theories: T 1 ["isBehind"] Function 0.2
T 2 ["isBehind","hasWindow"] Function 0.4
T 3 ["isBehind","hasWindow","size"] Function 0.6
T 4 ["isBehind","hasWindow","size","isTransparent"] Function 0.8
T 5 ["isBehind","hasWindow","size","isTransparent","movedSamePos"] Function 1.0

State: Use
............................
----------------------------
using theories ...
----------------------------
............................
Agent id: 1
Detected percepts:"isBehind"
"hasWindow"
"size"
"isTransparent"
"movedSamePos"

Observations: [(1,2,True),(3,4,False),(6,5,False),(7,8,False),(9,10,False)]
Current theories: T 1 ["isBehind"] Function 0.2
T 2 ["isBehind","hasWindow"] Function 0.4
T 3 ["isBehind","hasWindow","size"] Function 0.6
T 4 ["isBehind","hasWindow","size","isTransparent"] Function 0.8
T 5 ["isBehind","hasWindow","size","isTransparent","movedSamePos"] Function 1.0

State: Observe
............................
----------------------------
```

```
"Time: "10
----------------------------
observing environment ...
----------------------------
............................
Agent id: 1
Detected percepts:"isBehind"
"hasWindow"
"size"
"isTransparent"
"movedSamePos"

Observations: [(1,2,True),(3,4,False),(6,5,False),(7,8,False),(9,10,False)]
Current theories: T 1 ["isBehind"] Function 0.2
T 2 ["isBehind","hasWindow"] Function 0.4
T 3 ["isBehind","hasWindow","size"] Function 0.6
T 4 ["isBehind","hasWindow","size","isTransparent"] Function 0.8
T 5 ["isBehind","hasWindow","size","isTransparent","movedSamePos"] Function 1.0

State: Test
............................
----------------------------
building theories ...
----------------------------
............................
Agent id: 1
Detected percepts:"isBehind"
"hasWindow"
"size"
"isTransparent"
"movedSamePos"

Observations: [(1,2,True),(3,4,False),(6,5,False),(7,8,False),(9,10,False)]
Current theories: T 1 ["isBehind"] Function 0.2
T 2 ["isBehind","hasWindow"] Function 0.4
T 3 ["isBehind","hasWindow","size"] Function 0.6
T 4 ["isBehind","hasWindow","size","isTransparent"] Function 0.8
T 5 ["isBehind","hasWindow","size","isTransparent","movedSamePos"] Function 1.0

State: Test
............................
----------------------------
testing theories ...
----------------------------
............................
Agent id: 1
Detected percepts:"isBehind"
"hasWindow"
"size"
"isTransparent"
"movedSamePos"
```

```
Observations: [(1,2,True),(3,4,False),(6,5,False),(7,8,False),(9,10,False)]
Current theories: T 1 ["isBehind"] Function 0.2
T 2 ["isBehind","hasWindow"] Function 0.4
T 3 ["isBehind","hasWindow","size"] Function 0.6
T 4 ["isBehind","hasWindow","size","isTransparent"] Function 0.8
T 5 ["isBehind","hasWindow","size","isTransparent","movedSamePos"] Function 1.0

State: Use
............................
----------------------------
using theories ...
----------------------------
............................
Agent id: 1
Detected percepts:"isBehind"
"hasWindow"
"size"
"isTransparent"
"movedSamePos"

Observations: [(1,2,True),(3,4,False),(6,5,False),(7,8,False),(9,10,False)]
Current theories: T 1 ["isBehind"] Function 0.2
T 2 ["isBehind","hasWindow"] Function 0.4
T 3 ["isBehind","hasWindow","size"] Function 0.6
T 4 ["isBehind","hasWindow","size","isTransparent"] Function 0.8
T 5 ["isBehind","hasWindow","size","isTransparent","movedSamePos"] Function 1.0

State: Observe
............................
----------------------------
"Time: "11
----------------------------
observing environment ...
----------------------------
............................
Agent id: 1
Detected percepts:"isBehind"
"hasWindow"
"size"
"isTransparent"
"movedSamePos"

Observations: [(1,2,True),(3,4,False),(6,5,False),(7,8,False),(9,10,False)]
Current theories: T 1 ["isBehind"] Function 0.2
T 2 ["isBehind","hasWindow"] Function 0.4
T 3 ["isBehind","hasWindow","size"] Function 0.6
T 4 ["isBehind","hasWindow","size","isTransparent"] Function 0.8
T 5 ["isBehind","hasWindow","size","isTransparent","movedSamePos"] Function 1.0

State: Test
```

```
............................
----------------------------
building theories ...
----------------------------
............................
Agent id: 1
Detected percepts:"isBehind"
"hasWindow"
"size"
"isTransparent"
"movedSamePos"

Observations: [(1,2,True),(3,4,False),(6,5,False),(7,8,False),(9,10,False)]
Current theories: T 1 ["isBehind"] Function 0.2
T 2 ["isBehind","hasWindow"] Function 0.4
T 3 ["isBehind","hasWindow","size"] Function 0.6
T 4 ["isBehind","hasWindow","size","isTransparent"] Function 0.8
T 5 ["isBehind","hasWindow","size","isTransparent","movedSamePos"] Function 1.0

State: Test
............................
----------------------------
testing theories ...
----------------------------
............................
Agent id: 1
Detected percepts:"isBehind"
"hasWindow"
"size"
"isTransparent"
"movedSamePos"

Observations: [(1,2,True),(3,4,False),(6,5,False),(7,8,False),(9,10,False)]
Current theories: T 1 ["isBehind"] Function 0.2
T 2 ["isBehind","hasWindow"] Function 0.4
T 3 ["isBehind","hasWindow","size"] Function 0.6
T 4 ["isBehind","hasWindow","size","isTransparent"] Function 0.8
T 5 ["isBehind","hasWindow","size","isTransparent","movedSamePos"] Function 1.0

State: Use
............................
----------------------------
using theories ...
----------------------------
............................
Agent id: 1
Detected percepts:"isBehind"
"hasWindow"
"size"
"isTransparent"
"movedSamePos"
```

```
Observations: [(1,2,True),(3,4,False),(6,5,False),(7,8,False),(9,10,False)]
Current theories: T 1 ["isBehind"] Function 0.2
T 2 ["isBehind","hasWindow"] Function 0.4
T 3 ["isBehind","hasWindow","size"] Function 0.6
T 4 ["isBehind","hasWindow","size","isTransparent"] Function 0.8
T 5 ["isBehind","hasWindow","size","isTransparent","movedSamePos"] Function 1.0

State: Observe
............................
----------------------------
Agent getting bored - no new data
CHECK Simulation time limit
```

D.2 Simulation - Theories “Containment of Solids”

```
*NewSim> simulate "Containment"
Loading package haskell98-1.0 ... linking ... done.
"Time: "1
----------------------------
observing environment ...
----------------------------
............................
Agent id: 3
Detected percepts:"isOpen"

Observations: [(1,2,True)]
Current theories: T 21 ["isOpen"] Function 0.0

State: Test
............................
----------------------------
building theories ...
----------------------------
............................
Agent id: 3
Detected percepts:"isOpen"

Observations: [(1,2,True)]
Current theories: T 21 ["isOpen"] Function 0.0

State: Test
............................
----------------------------
testing theories ...
----------------------------
............................
Agent id: 3
Detected percepts:"isOpen"

Observations: [(1,2,True)]
Current theories: T 21 ["isOpen"] Function 1.0

State: Use
............................
----------------------------
using theories ...
----------------------------
............................
Agent id: 3
Detected percepts:"isOpen"

Observations: [(1,2,True)]
Current theories: T 21 ["isOpen"] Function 1.0
```

```
State: Observe
............................
----------------------------
"Time: "2
----------------------------
observing environment ...
----------------------------
............................
Agent id: 3
Detected percepts:"isOpen"
"movement"

Observations: [(1,2,True),(3,4,False)]
Current theories: T 21 ["isOpen"] Function 1.0

State: Test
............................
----------------------------
building theories ...
----------------------------
............................
Agent id: 3
Detected percepts:"isOpen"
"movement"

Observations: [(1,2,True),(3,4,False)]
Current theories: T 21 ["isOpen"] Function 1.0

State: Test
............................
----------------------------
testing theories ...
----------------------------
............................
Agent id: 3
Detected percepts:"isOpen"
"movement"

Observations: [(1,2,True),(3,4,False)]
Current theories: T 21 ["isOpen"] Function 0.5

State: Build
............................
----------------------------
using theories ...
----------------------------
............................
Agent id: 3
Detected percepts:"isOpen"
"movement"
```

```
Observations: [(1,2,True),(3,4,False)]
Current theories: T 21 ["isOpen"] Function 0.5

State: Build
............................
----------------------------
"Time: "3
----------------------------
observing environment ...
----------------------------
............................
Agent id: 3
Detected percepts:"isOpen"
"movement"

Observations: [(1,2,True),(3,4,False)]
Current theories: T 21 ["isOpen"] Function 0.5

State: Build
............................
----------------------------
building theories ...
----------------------------
............................
Agent id: 3
Detected percepts:"isOpen"
"movement"

Observations: [(1,2,True),(3,4,False)]
Current theories: T 21 ["isOpen"] Function 0.5
T 22 ["isOpen","movement"] Function 0.0

State: Test
............................
----------------------------
testing theories ...
----------------------------
............................
Agent id: 3
Detected percepts:"isOpen"
"movement"

Observations: [(1,2,True),(3,4,False)]
Current theories: T 21 ["isOpen"] Function 0.5
T 22 ["isOpen","movement"] Function 1.0

State: Use
............................
----------------------------
using theories ...
```

```
----------------------------
............................
Agent id: 3
Detected percepts:"isOpen"
"movement"

Observations: [(1,2,True),(3,4,False)]
Current theories: T 21 ["isOpen"] Function 0.5
T 22 ["isOpen","movement"] Function 1.0

State: Observe
............................
----------------------------
"Time: "4
----------------------------
observing environment ...
----------------------------
............................
Agent id: 3
Detected percepts:"isOpen"
"movement"
"size"

Observations: [(1,2,True),(3,4,False),(6,5,False)]
Current theories: T 21 ["isOpen"] Function 0.5
T 22 ["isOpen","movement"] Function 1.0

State: Test
............................
----------------------------
building theories ...
----------------------------
............................
Agent id: 3
Detected percepts:"isOpen"
"movement"
"size"

Observations: [(1,2,True),(3,4,False),(6,5,False)]
Current theories: T 21 ["isOpen"] Function 0.5
T 22 ["isOpen","movement"] Function 1.0

State: Test
............................
----------------------------
testing theories ...
----------------------------
............................
Agent id: 3
Detected percepts:"isOpen"
"movement"
```

```
"size"

Observations: [(1,2,True),(3,4,False),(6,5,False)]
Current theories: T 21 ["isOpen"] Function 0.33333334
T 22 ["isOpen","movement"] Function 0.6666667

State: Build
............................
----------------------------
using theories ...
----------------------------
............................
Agent id: 3
Detected percepts:"isOpen"
"movement"
"size"

Observations: [(1,2,True),(3,4,False),(6,5,False)]
Current theories: T 21 ["isOpen"] Function 0.33333334
T 22 ["isOpen","movement"] Function 0.6666667

State: Build
............................
----------------------------
"Time: "5
----------------------------
observing environment ...
----------------------------
............................
Agent id: 3
Detected percepts:"isOpen"
"movement"
"size"

Observations: [(1,2,True),(3,4,False),(6,5,False)]
Current theories: T 21 ["isOpen"] Function 0.33333334
T 22 ["isOpen","movement"] Function 0.6666667

State: Build
............................
----------------------------
building theories ...
----------------------------
............................
Agent id: 3
Detected percepts:"isOpen"
"movement"
"size"

Observations: [(1,2,True),(3,4,False),(6,5,False)]
Current theories: T 21 ["isOpen"] Function 0.33333334
```

```
T 22 ["isOpen","movement"] Function 0.6666667
T 23 ["isOpen","movement","size"] Function 0.0

State: Test
............................
----------------------------
testing theories ...
----------------------------
............................
Agent id: 3
Detected percepts:"isOpen"
"movement"
"size"

Observations: [(1,2,True),(3,4,False),(6,5,False)]
Current theories: T 21 ["isOpen"] Function 0.33333334
T 22 ["isOpen","movement"] Function 0.6666667
T 23 ["isOpen","movement","size"] Function 1.0

State: Use
............................
----------------------------
using theories ...
----------------------------
............................
Agent id: 3
Detected percepts:"isOpen"
"movement"
"size"

Observations: [(1,2,True),(3,4,False),(6,5,False)]
Current theories: T 21 ["isOpen"] Function 0.33333334
T 22 ["isOpen","movement"] Function 0.6666667
T 23 ["isOpen","movement","size"] Function 1.0

State: Observe
............................
----------------------------
"Time: "6
----------------------------
observing environment ...
----------------------------
............................
Agent id: 3
Detected percepts:"isOpen"
"movement"
"size"
"loosefit"

Observations: [(1,2,True),(3,4,False),(6,5,False),(7,8,False)]
Current theories: T 21 ["isOpen"] Function 0.33333334
```

```
T 22 ["isOpen","movement"] Function 0.6666667
T 23 ["isOpen","movement","size"] Function 1.0

State: Test
............................
----------------------------
building theories ...
----------------------------
............................
Agent id: 3
Detected percepts:"isOpen"
"movement"
"size"
"loosefit"

Observations: [(1,2,True),(3,4,False),(6,5,False),(7,8,False)]
Current theories: T 21 ["isOpen"] Function 0.33333334
T 22 ["isOpen","movement"] Function 0.6666667
T 23 ["isOpen","movement","size"] Function 1.0

State: Test
............................
----------------------------
testing theories ...
----------------------------
............................
Agent id: 3
Detected percepts:"isOpen"
"movement"
"size"
"loosefit"

Observations: [(1,2,True),(3,4,False),(6,5,False),(7,8,False)]
Current theories: T 21 ["isOpen"] Function 0.25
T 22 ["isOpen","movement"] Function 0.5
T 23 ["isOpen","movement","size"] Function 0.75

State: Build
............................
----------------------------
using theories ...
----------------------------
............................
Agent id: 3
Detected percepts:"isOpen"
"movement"
"size"
"loosefit"

Observations: [(1,2,True),(3,4,False),(6,5,False),(7,8,False)]
Current theories: T 21 ["isOpen"] Function 0.25
```

```
T 22 ["isOpen","movement"] Function 0.5
T 23 ["isOpen","movement","size"] Function 0.75

State: Build
............................
----------------------------
"Time: "7
----------------------------
observing environment ...
----------------------------
............................
Agent id: 3
Detected percepts:"isOpen"
"movement"
"size"
"loosefit"

Observations: [(1,2,True),(3,4,False),(6,5,False),(7,8,False)]
Current theories: T 21 ["isOpen"] Function 0.25
T 22 ["isOpen","movement"] Function 0.5
T 23 ["isOpen","movement","size"] Function 0.75

State: Build
............................
----------------------------
building theories ...
----------------------------
............................
Agent id: 3
Detected percepts:"isOpen"
"movement"
"size"
"loosefit"

Observations: [(1,2,True),(3,4,False),(6,5,False),(7,8,False)]
Current theories: T 21 ["isOpen"] Function 0.25
T 22 ["isOpen","movement"] Function 0.5
T 23 ["isOpen","movement","size"] Function 0.75
T 24 ["isOpen","movement","size","loosefit"] Function 0.0

State: Test
............................
----------------------------
testing theories ...
----------------------------
............................
Agent id: 3
Detected percepts:"isOpen"
"movement"
"size"
"loosefit"
```

```
Observations: [(1,2,True),(3,4,False),(6,5,False),(7,8,False)]
Current theories: T 21 ["isOpen"] Function 0.25
T 22 ["isOpen","movement"] Function 0.5
T 23 ["isOpen","movement","size"] Function 0.75
T 24 ["isOpen","movement","size","loosefit"] Function 1.0

State: Use
............................
----------------------------
using theories ...
----------------------------
............................
Agent id: 3
Detected percepts:"isOpen"
"movement"
"size"
"loosefit"

Observations: [(1,2,True),(3,4,False),(6,5,False),(7,8,False)]
Current theories: T 21 ["isOpen"] Function 0.25
T 22 ["isOpen","movement"] Function 0.5
T 23 ["isOpen","movement","size"] Function 0.75
T 24 ["isOpen","movement","size","loosefit"] Function 1.0

State: Observe
............................
----------------------------
"Time: "8
----------------------------
observing environment ...
----------------------------
............................
Agent id: 3
Detected percepts:"isOpen"
"movement"
"size"
"loosefit"
"tightfit"

Observations: [(1,2,True),(3,4,False),(6,5,False),(7,8,False),(9,10,True)]
Current theories: T 21 ["isOpen"] Function 0.25
T 22 ["isOpen","movement"] Function 0.5
T 23 ["isOpen","movement","size"] Function 0.75
T 24 ["isOpen","movement","size","loosefit"] Function 1.0

State: Test
............................
----------------------------
building theories ...
----------------------------
```

```
............................
Agent id: 3
Detected percepts:"isOpen"
"movement"
"size"
"loosefit"
"tightfit"

Observations: [(1,2,True),(3,4,False),(6,5,False),(7,8,False),(9,10,True)]
Current theories: T 21 ["isOpen"] Function 0.25
T 22 ["isOpen","movement"] Function 0.5
T 23 ["isOpen","movement","size"] Function 0.75
T 24 ["isOpen","movement","size","loosefit"] Function 1.0

State: Test
............................
----------------------------
testing theories ...
----------------------------
............................
Agent id: 3
Detected percepts:"isOpen"
"movement"
"size"
"loosefit"
"tightfit"

Observations: [(1,2,True),(3,4,False),(6,5,False),(7,8,False),(9,10,True)]
Current theories: T 21 ["isOpen"] Function 0.4
T 22 ["isOpen","movement"] Function 0.6
T 23 ["isOpen","movement","size"] Function 0.8
T 24 ["isOpen","movement","size","loosefit"] Function 0.8

State: Build
............................
----------------------------
using theories ...
----------------------------
............................
Agent id: 3
Detected percepts:"isOpen"
"movement"
"size"
"loosefit"
"tightfit"

Observations: [(1,2,True),(3,4,False),(6,5,False),(7,8,False),(9,10,True)]
Current theories: T 21 ["isOpen"] Function 0.4
T 22 ["isOpen","movement"] Function 0.6
T 23 ["isOpen","movement","size"] Function 0.8
T 24 ["isOpen","movement","size","loosefit"] Function 0.8
```

```
State: Build
............................
----------------------------
"Time: "9
----------------------------
observing environment ...
----------------------------
............................
Agent id: 3
Detected percepts:"isOpen"
"movement"
"size"
"loosefit"
"tightfit"

Observations: [(1,2,True),(3,4,False),(6,5,False),(7,8,False),(9,10,True)]
Current theories: T 21 ["isOpen"] Function 0.4
T 22 ["isOpen","movement"] Function 0.6
T 23 ["isOpen","movement","size"] Function 0.8
T 24 ["isOpen","movement","size","loosefit"] Function 0.8

State: Build
............................
----------------------------
building theories ...
----------------------------
............................
Agent id: 3
Detected percepts:"isOpen"
"movement"
"size"
"loosefit"
"tightfit"

Observations: [(1,2,True),(3,4,False),(6,5,False),(7,8,False),(9,10,True)]
Current theories: T 21 ["isOpen"] Function 0.4
T 22 ["isOpen","movement"] Function 0.6
T 23 ["isOpen","movement","size"] Function 0.8
T 24 ["isOpen","movement","size","loosefit"] Function 0.8
T 25 ["isOpen","movement","size","loosefit","tightfit"] Function 0.0

State: Test
............................
----------------------------
testing theories ...
----------------------------
............................
Agent id: 3
Detected percepts:"isOpen"
"movement"
```

```
"size"
"loosefit"
"tightfit"

Observations: [(1,2,True),(3,4,False),(6,5,False),(7,8,False),(9,10,True)]
Current theories: T 21 ["isOpen"] Function 0.4
T 22 ["isOpen","movement"] Function 0.6
T 23 ["isOpen","movement","size"] Function 0.8
T 24 ["isOpen","movement","size","loosefit"] Function 0.8
T 25 ["isOpen","movement","size","loosefit","tightfit"] Function 0.6

State: Build
............................
----------------------------
using theories ...
----------------------------
............................
Agent id: 3
Detected percepts:"isOpen"
"movement"
"size"
"loosefit"
"tightfit"

Observations: [(1,2,True),(3,4,False),(6,5,False),(7,8,False),(9,10,True)]
Current theories: T 21 ["isOpen"] Function 0.4
T 22 ["isOpen","movement"] Function 0.6
T 23 ["isOpen","movement","size"] Function 0.8
T 24 ["isOpen","movement","size","loosefit"] Function 0.8
T 25 ["isOpen","movement","size","loosefit","tightfit"] Function 0.6

State: Build
............................
----------------------------
"Time: "10
----------------------------
observing environment ...
----------------------------
............................
Agent id: 3
Detected percepts:"isOpen"
"movement"
"size"
"loosefit"
"tightfit"

Observations: [(1,2,True),(3,4,False),(6,5,False),(7,8,False),(9,10,True)]
Current theories: T 21 ["isOpen"] Function 0.4
T 22 ["isOpen","movement"] Function 0.6
T 23 ["isOpen","movement","size"] Function 0.8
T 24 ["isOpen","movement","size","loosefit"] Function 0.8
```

```
T 25 ["isOpen","movement","size","loosefit","tightfit"] Function 0.6

State: Build
............................
----------------------------
building theories ...
----------------------------
............................
Agent id: 3
Detected percepts:"isOpen"
"movement"
"size"
"loosefit"
"tightfit"

Observations: [(1,2,True),(3,4,False),(6,5,False),(7,8,False),(9,10,True)]
Current theories: T 21 ["isOpen"] Function 0.4
T 22 ["isOpen","movement"] Function 0.6
T 23 ["isOpen","movement","size"] Function 0.8
T 24 ["isOpen","movement","size","loosefit"] Function 0.8
T 25 ["isOpen","movement","size","loosefit","tightfit"] Function 0.6

State: GiveUp
............................
----------------------------
testing theories ...
----------------------------
............................
Agent id: 3
Detected percepts:"isOpen"
"movement"
"size"
"loosefit"
"tightfit"

Observations: [(1,2,True),(3,4,False),(6,5,False),(7,8,False),(9,10,True)]
Current theories: T 21 ["isOpen"] Function 0.4
T 22 ["isOpen","movement"] Function 0.6
T 23 ["isOpen","movement","size"] Function 0.8
T 24 ["isOpen","movement","size","loosefit"] Function 0.8
T 25 ["isOpen","movement","size","loosefit","tightfit"] Function 0.6

State: GiveUp
............................
----------------------------
using theories ...
----------------------------
*** Exception: No percepts or theories found that fit the data.
*NewSim>
```

D.3 Simulation - Theories “Support of Solids”

```
Loading package haskell98-1.0 ... linking ... done.
>simulate "Support"
"Time: "1
----------------------------
observing environment ...
----------------------------
............................
Agent id: 2
Detected percepts:"Contact"

Observations: [(1,2,False)]
Current theories: T 10 ["Contact"] Function 0.0

State: Test
............................
----------------------------
building theories ...
----------------------------
............................
Agent id: 2
Detected percepts:"Contact"

Observations: [(1,2,False)]
Current theories: T 10 ["Contact"] Function 0.0

State: Test
............................
----------------------------
testing theories ...
----------------------------
............................
Agent id: 2
Detected percepts:"Contact"

Observations: [(1,2,False)]
Current theories: T 10 ["Contact"] Function 1.0

State: Use
............................
----------------------------
using theories ...
----------------------------
............................
Agent id: 2
Detected percepts:"Contact"

Observations: [(1,2,False)]
Current theories: T 10 ["Contact"] Function 1.0
```

```
State: Observe
............................
----------------------------
"Time: "2
----------------------------
observing environment ...
----------------------------
............................
Agent id: 2
Detected percepts:"Contact"
"TopContact"

Observations: [(1,2,False),(3,4,False)]
Current theories: T 10 ["Contact"] Function 1.0

State: Test
............................
----------------------------
building theories ...
----------------------------
............................
Agent id: 2
Detected percepts:"Contact"
"TopContact"

Observations: [(1,2,False),(3,4,False)]
Current theories: T 10 ["Contact"] Function 1.0

State: Test
............................
----------------------------
testing theories ...
----------------------------
............................
Agent id: 2
Detected percepts:"Contact"
"TopContact"

Observations: [(1,2,False),(3,4,False)]
Current theories: T 10 ["Contact"] Function 0.5

State: Build
............................
----------------------------
using theories ...
----------------------------
............................
Agent id: 2
Detected percepts:"Contact"
"TopContact"
```

```
Observations: [(1,2,False),(3,4,False)]
Current theories: T 10 ["Contact"] Function 0.5

State: Build
............................
----------------------------
"Time: "3
----------------------------
observing environment ...
----------------------------
............................
Agent id: 2
Detected percepts:"Contact"
"TopContact"

Observations: [(1,2,False),(3,4,False)]
Current theories: T 10 ["Contact"] Function 0.5

State: Build
............................
----------------------------
building theories ...
----------------------------
............................
Agent id: 2
Detected percepts:"Contact"
"TopContact"

Observations: [(1,2,False),(3,4,False)]
Current theories: T 10 ["Contact"] Function 0.5
T 11 ["Contact","TopContact"] Function 0.0

State: Test
............................
----------------------------
testing theories ...
----------------------------
............................
Agent id: 2
Detected percepts:"Contact"
"TopContact"

Observations: [(1,2,False),(3,4,False)]
Current theories: T 10 ["Contact"] Function 0.5
T 11 ["Contact","TopContact"] Function 1.0

State: Use
............................
----------------------------
using theories ...
```

```
----------------------------
............................
Agent id: 2
Detected percepts:"Contact"
"TopContact"

Observations: [(1,2,False),(3,4,False)]
Current theories: T 10 ["Contact"] Function 0.5
T 11 ["Contact","TopContact"] Function 1.0

State: Observe
............................
----------------------------
"Time: "4
----------------------------
observing environment ...
----------------------------
............................
Agent id: 2
Detected percepts:"Contact"
"TopContact"
"AmountContact"

Observations: [(1,2,False),(3,4,False),(5,6,False)]
Current theories: T 10 ["Contact"] Function 0.5
T 11 ["Contact","TopContact"] Function 1.0

State: Test
............................
----------------------------
building theories ...
----------------------------
............................
Agent id: 2
Detected percepts:"Contact"
"TopContact"
"AmountContact"

Observations: [(1,2,False),(3,4,False),(5,6,False)]
Current theories: T 10 ["Contact"] Function 0.5
T 11 ["Contact","TopContact"] Function 1.0

State: Test
............................
----------------------------
testing theories ...
----------------------------
............................
Agent id: 2
Detected percepts:"Contact"
"TopContact"
```

```
"AmountContact"

Observations: [(1,2,False),(3,4,False),(5,6,False)]
Current theories: T 10 ["Contact"] Function 0.33333334
T 11 ["Contact","TopContact"] Function 0.6666667

State: Build
............................
----------------------------
using theories ...
----------------------------
............................
Agent id: 2
Detected percepts:"Contact"
"TopContact"
"AmountContact"

Observations: [(1,2,False),(3,4,False),(5,6,False)]
Current theories: T 10 ["Contact"] Function 0.33333334
T 11 ["Contact","TopContact"] Function 0.6666667

State: Build
............................
----------------------------
"Time: "5
----------------------------
observing environment ...
----------------------------
............................
Agent id: 2
Detected percepts:"Contact"
"TopContact"
"AmountContact"

Observations: [(1,2,False),(3,4,False),(5,6,False)]
Current theories: T 10 ["Contact"] Function 0.33333334
T 11 ["Contact","TopContact"] Function 0.6666667

State: Build
............................
----------------------------
building theories ...
----------------------------
............................
Agent id: 2
Detected percepts:"Contact"
"TopContact"
"AmountContact"

Observations: [(1,2,False),(3,4,False),(5,6,False)]
Current theories: T 10 ["Contact"] Function 0.33333334
```

```
T 11 ["Contact","TopContact"] Function 0.6666667
T 12 ["Contact","TopContact","AmountContact"] Function 0.0

State: Test
............................
----------------------------
testing theories ...
----------------------------
............................
Agent id: 2
Detected percepts:"Contact"
"TopContact"
"AmountContact"

Observations: [(1,2,False),(3,4,False),(5,6,False)]
Current theories: T 10 ["Contact"] Function 0.33333334
T 11 ["Contact","TopContact"] Function 0.6666667
T 12 ["Contact","TopContact","AmountContact"] Function 1.0

State: Use
............................
----------------------------
using theories ...
----------------------------
............................
Agent id: 2
Detected percepts:"Contact"
"TopContact"
"AmountContact"

Observations: [(1,2,False),(3,4,False),(5,6,False)]
Current theories: T 10 ["Contact"] Function 0.33333334
T 11 ["Contact","TopContact"] Function 0.6666667
T 12 ["Contact","TopContact","AmountContact"] Function 1.0

State: Observe
............................
----------------------------
"Time: "6
----------------------------
observing environment ...
----------------------------
............................
Agent id: 2
Detected percepts:"Contact"
"TopContact"
"AmountContact"
"Shape"

Observations: [(1,2,False),(3,4,False),(5,6,False),(7,8,False)]
Current theories: T 10 ["Contact"] Function 0.33333334
```

```
T 11 ["Contact","TopContact"] Function 0.6666667
T 12 ["Contact","TopContact","AmountContact"] Function 1.0

State: Test
............................
----------------------------
building theories ...
----------------------------
............................
Agent id: 2
Detected percepts:"Contact"
"TopContact"
"AmountContact"
"Shape"

Observations: [(1,2,False),(3,4,False),(5,6,False),(7,8,False)]
Current theories: T 10 ["Contact"] Function 0.33333334
T 11 ["Contact","TopContact"] Function 0.6666667
T 12 ["Contact","TopContact","AmountContact"] Function 1.0

State: Test
............................
----------------------------
testing theories ...
----------------------------
............................
Agent id: 2
Detected percepts:"Contact"
"TopContact"
"AmountContact"
"Shape"

Observations: [(1,2,False),(3,4,False),(5,6,False),(7,8,False)]
Current theories: T 10 ["Contact"] Function 0.25
T 11 ["Contact","TopContact"] Function 0.5
T 12 ["Contact","TopContact","AmountContact"] Function 0.75

State: Build
............................
----------------------------
using theories ...
----------------------------
............................
Agent id: 2
Detected percepts:"Contact"
"TopContact"
"AmountContact"
"Shape"

Observations: [(1,2,False),(3,4,False),(5,6,False),(7,8,False)]
Current theories: T 10 ["Contact"] Function 0.25
```

```
T 11 ["Contact","TopContact"] Function 0.5
T 12 ["Contact","TopContact","AmountContact"] Function 0.75

State: Build
............................
----------------------------
"Time: "7
----------------------------
observing environment ...
----------------------------
............................
Agent id: 2
Detected percepts:"Contact"
"TopContact"
"AmountContact"
"Shape"

Observations: [(1,2,False),(3,4,False),(5,6,False),(7,8,False)]
Current theories: T 10 ["Contact"] Function 0.25
T 11 ["Contact","TopContact"] Function 0.5
T 12 ["Contact","TopContact","AmountContact"] Function 0.75

State: Build
............................
----------------------------
building theories ...
----------------------------
............................
Agent id: 2
Detected percepts:"Contact"
"TopContact"
"AmountContact"
"Shape"

Observations: [(1,2,False),(3,4,False),(5,6,False),(7,8,False)]
Current theories: T 10 ["Contact"] Function 0.25
T 11 ["Contact","TopContact"] Function 0.5
T 12 ["Contact","TopContact","AmountContact"] Function 0.75
T 13 ["Contact","TopContact","AmountContact","Shape"] Function 0.0

State: Test
............................
----------------------------
testing theories ...
----------------------------
............................
Agent id: 2
Detected percepts:"Contact"
"TopContact"
"AmountContact"
"Shape"
```

```
Observations: [(1,2,False),(3,4,False),(5,6,False),(7,8,False)]
Current theories: T 10 ["Contact"] Function 0.25
T 11 ["Contact","TopContact"] Function 0.5
T 12 ["Contact","TopContact","AmountContact"] Function 0.75
T 13 ["Contact","TopContact","AmountContact","Shape"] Function 1.0

State: Use
............................
----------------------------
using theories ...
----------------------------
............................
Agent id: 2
Detected percepts:"Contact"
"TopContact"
"AmountContact"
"Shape"

Observations: [(1,2,False),(3,4,False),(5,6,False),(7,8,False)]
Current theories: T 10 ["Contact"] Function 0.25
T 11 ["Contact","TopContact"] Function 0.5
T 12 ["Contact","TopContact","AmountContact"] Function 0.75
T 13 ["Contact","TopContact","AmountContact","Shape"] Function 1.0

State: Observe
............................
----------------------------
"Time: "8
----------------------------
observing environment ...
----------------------------
............................
Agent id: 2
Detected percepts:"Contact"
"TopContact"
"AmountContact"
"Shape"

Observations: [(1,2,False),(3,4,False),(5,6,False),(7,8,False),(9,10,True)]
Current theories: T 10 ["Contact"] Function 0.25
T 11 ["Contact","TopContact"] Function 0.5
T 12 ["Contact","TopContact","AmountContact"] Function 0.75
T 13 ["Contact","TopContact","AmountContact","Shape"] Function 1.0

State: Test
............................
----------------------------
building theories ...
----------------------------
............................
```

```
Agent id: 2
Detected percepts:"Contact"
"TopContact"
"AmountContact"
"Shape"

Observations: [(1,2,False),(3,4,False),(5,6,False),(7,8,False),(9,10,True)]
Current theories: T 10 ["Contact"] Function 0.25
T 11 ["Contact","TopContact"] Function 0.5
T 12 ["Contact","TopContact","AmountContact"] Function 0.75
T 13 ["Contact","TopContact","AmountContact","Shape"] Function 1.0

State: Test
............................
----------------------------
testing theories ...
----------------------------
............................
Agent id: 2
Detected percepts:"Contact"
"TopContact"
"AmountContact"
"Shape"

Observations: [(1,2,False),(3,4,False),(5,6,False),(7,8,False),(9,10,True)]
Current theories: T 10 ["Contact"] Function 0.4
T 11 ["Contact","TopContact"] Function 0.6
T 12 ["Contact","TopContact","AmountContact"] Function 0.8
T 13 ["Contact","TopContact","AmountContact","Shape"] Function 1.0

State: Use
............................
----------------------------
using theories ...
----------------------------
............................
Agent id: 2
Detected percepts:"Contact"
"TopContact"
"AmountContact"
"Shape"

Observations: [(1,2,False),(3,4,False),(5,6,False),(7,8,False),(9,10,True)]
Current theories: T 10 ["Contact"] Function 0.4
T 11 ["Contact","TopContact"] Function 0.6
T 12 ["Contact","TopContact","AmountContact"] Function 0.8
T 13 ["Contact","TopContact","AmountContact","Shape"] Function 1.0

State: Observe
............................
----------------------------
```

```
"Time: "9
----------------------------
observing environment ...
----------------------------
............................
Agent id: 2
Detected percepts:"Contact"
"TopContact"
"AmountContact"
"Shape"

Observations: [(1,2,False),(3,4,False),(5,6,False),(7,8,False),(9,10,True)]
Current theories: T 10 ["Contact"] Function 0.4
T 11 ["Contact","TopContact"] Function 0.6
T 12 ["Contact","TopContact","AmountContact"] Function 0.8
T 13 ["Contact","TopContact","AmountContact","Shape"] Function 1.0

State: Test
............................
----------------------------
building theories ...
----------------------------
............................
Agent id: 2
Detected percepts:"Contact"
"TopContact"
"AmountContact"
"Shape"

Observations: [(1,2,False),(3,4,False),(5,6,False),(7,8,False),(9,10,True)]
Current theories: T 10 ["Contact"] Function 0.4
T 11 ["Contact","TopContact"] Function 0.6
T 12 ["Contact","TopContact","AmountContact"] Function 0.8
T 13 ["Contact","TopContact","AmountContact","Shape"] Function 1.0

State: Test
............................
----------------------------
testing theories ...
----------------------------
............................
Agent id: 2
Detected percepts:"Contact"
"TopContact"
"AmountContact"
"Shape"

Observations: [(1,2,False),(3,4,False),(5,6,False),(7,8,False),(9,10,True)]
Current theories: T 10 ["Contact"] Function 0.4
T 11 ["Contact","TopContact"] Function 0.6
T 12 ["Contact","TopContact","AmountContact"] Function 0.8
```

```
T 13 ["Contact","TopContact","AmountContact","Shape"] Function 1.0

State: Use
............................
----------------------------
using theories ...
----------------------------
............................
Agent id: 2
Detected percepts:"Contact"
"TopContact"
"AmountContact"
"Shape"

Observations: [(1,2,False),(3,4,False),(5,6,False),(7,8,False),(9,10,True)]
Current theories: T 10 ["Contact"] Function 0.4
T 11 ["Contact","TopContact"] Function 0.6
T 12 ["Contact","TopContact","AmountContact"] Function 0.8
T 13 ["Contact","TopContact","AmountContact","Shape"] Function 1.0

State: Observe
............................
----------------------------
"Time: "10
----------------------------
observing environment ...
----------------------------
............................
Agent id: 2
Detected percepts:"Contact"
"TopContact"
"AmountContact"
"Shape"

Observations: [(1,2,False),(3,4,False),(5,6,False),(7,8,False),(9,10,True)]
Current theories: T 10 ["Contact"] Function 0.4
T 11 ["Contact","TopContact"] Function 0.6
T 12 ["Contact","TopContact","AmountContact"] Function 0.8
T 13 ["Contact","TopContact","AmountContact","Shape"] Function 1.0

State: Test
............................
----------------------------
building theories ...
----------------------------
............................
Agent id: 2
Detected percepts:"Contact"
"TopContact"
"AmountContact"
"Shape"
```

```
Observations: [(1,2,False),(3,4,False),(5,6,False),(7,8,False),(9,10,True)]
Current theories: T 10 ["Contact"] Function 0.4
T 11 ["Contact","TopContact"] Function 0.6
T 12 ["Contact","TopContact","AmountContact"] Function 0.8
T 13 ["Contact","TopContact","AmountContact","Shape"] Function 1.0

State: Test
............................
----------------------------
testing theories ...
----------------------------
............................
Agent id: 2
Detected percepts:"Contact"
"TopContact"
"AmountContact"
"Shape"

Observations: [(1,2,False),(3,4,False),(5,6,False),(7,8,False),(9,10,True)]
Current theories: T 10 ["Contact"] Function 0.4
T 11 ["Contact","TopContact"] Function 0.6
T 12 ["Contact","TopContact","AmountContact"] Function 0.8
T 13 ["Contact","TopContact","AmountContact","Shape"] Function 1.0

State: Use
............................
----------------------------
using theories ...
----------------------------
............................
Agent id: 2
Detected percepts:"Contact"
"TopContact"
"AmountContact"
"Shape"

Observations: [(1,2,False),(3,4,False),(5,6,False),(7,8,False),(9,10,True)]
Current theories: T 10 ["Contact"] Function 0.4
T 11 ["Contact","TopContact"] Function 0.6
T 12 ["Contact","TopContact","AmountContact"] Function 0.8
T 13 ["Contact","TopContact","AmountContact","Shape"] Function 1.0

State: Observe
............................
----------------------------
"Time: "11
----------------------------
observing environment ...
----------------------------
............................
```

```
Agent id: 2
Detected percepts:"Contact"
"TopContact"
"AmountContact"
"Shape"

Observations: [(1,2,False),(3,4,False),(5,6,False),(7,8,False),(9,10,True)]
Current theories: T 10 ["Contact"] Function 0.4
T 11 ["Contact","TopContact"] Function 0.6
T 12 ["Contact","TopContact","AmountContact"] Function 0.8
T 13 ["Contact","TopContact","AmountContact","Shape"] Function 1.0

State: Test
............................
----------------------------
building theories ...
----------------------------
............................
Agent id: 2
Detected percepts:"Contact"
"TopContact"
"AmountContact"
"Shape"

Observations: [(1,2,False),(3,4,False),(5,6,False),(7,8,False),(9,10,True)]
Current theories: T 10 ["Contact"] Function 0.4
T 11 ["Contact","TopContact"] Function 0.6
T 12 ["Contact","TopContact","AmountContact"] Function 0.8
T 13 ["Contact","TopContact","AmountContact","Shape"] Function 1.0

State: Test
............................
----------------------------
testing theories ...
----------------------------
............................
Agent id: 2
Detected percepts:"Contact"
"TopContact"
"AmountContact"
"Shape"

Observations: [(1,2,False),(3,4,False),(5,6,False),(7,8,False),(9,10,True)]
Current theories: T 10 ["Contact"] Function 0.4
T 11 ["Contact","TopContact"] Function 0.6
T 12 ["Contact","TopContact","AmountContact"] Function 0.8
T 13 ["Contact","TopContact","AmountContact","Shape"] Function 1.0

State: Use
............................
----------------------------
```

```
using theories ...
----------------------------
............................
Agent id: 2
Detected percepts:"Contact"
"TopContact"
"AmountContact"
"Shape"

Observations: [(1,2,False),(3,4,False),(5,6,False),(7,8,False),(9,10,True)]
Current theories: T 10 ["Contact"] Function 0.4
T 11 ["Contact","TopContact"] Function 0.6
T 12 ["Contact","TopContact","AmountContact"] Function 0.8
T 13 ["Contact","TopContact","AmountContact","Shape"] Function 1.0

State: Observe
............................
----------------------------
Agent getting bored - no new data
CHECK Simulation time limit
*NewSim>
```

Biography of the Author

Florian Twaroch was born in Vienna, Austria on June 13, 1975. He received his high school diploma with honors from the ”Katholische Privatschule Friesgasse”, Vienna, Austria in 1993. He entered the Vienna University of Technology in 1993 to study Surveying and Geoinformation.

During this time being a student Florian had several part time jobs as a surveyor and programmer. From 1996 to 1998 he did work freelance in several small projects, developing software for companies involved in the fields of civil engineering, surveying and archeology. Florian graduated with a master thesis on model based reconstruction of buildings and their administration in topographic information systems in June 2001.

From July 2001 to February 2003 he worked for the Advanced Computer Vision GmbH - ACV, a subsidiary of the Austrian Research Centers. He was employed as a scientific project manager and involved in the development of GIS software in distributed environments. He investigated algorithms in the fields of GIS and computational geometry.

Since 2003 Florian has been a Project Assistant at the Institute for Geoinformation and Cartography at the Vienna University of Technology, where he has been involved in the project management, the administrative and technical execution of EC-funded projects. The projects have been dealing with decision support and information systems for trans-European web portals, based on proprietary and open standards software. He carried out research on interoperability of GIS with OGC web services. In feasibility studies he also investigated marketing instruments and business models for the portals. He co-organized two workshops in the course of the projects.

Florian has been teaching courses in surveying for space planners and architects from 2004 to 2007 at the Vienna University of Technology. Florian has authored and co-authored several research papers published in conference proceedings. Florian Twaroch is married and has two children. He is a candidate for the “Doktor der technischen Wissenschaften” degree from the Vienna University of Technology in 2007.