Knowing One's Limits

Logical Analysis of Inductive Inference

Nina Gierasimczuk

Knowing One's Limits

Logical Analysis of Inductive Inference

ILLC Dissertation Series DS-2010-11

INSTITUTE FOR LOGIC, LANGUAGE AND COMPUTATION

For further information about ILLC-publications, please contact

Institute for Logic, Language and Computation
Universiteit van Amsterdam
Science Park 904
1098 XH Amsterdam
phone: +31-20-525 6051
fax: +31-20-525 5206
e-mail: illc@uva.nl
homepage: http://www.illc.uva.nl/

Knowing One's Limits

Logical Analysis of Inductive Inference

Academisch Proefschrift

ter verkrijging van de graad van doctor aan de
Universiteit van Amsterdam
op gezag van de Rector Magnificus
prof. dr. D. van den Boom
ten overstaan van een door het college voor
promoties ingestelde commissie, in het openbaar
te verdedigen in de Agnietenkapel
op vrijdag 17 december 2010, te 14.00 uur

door

Nina Gierasimczuk

geboren te Brussel, België.

Promotiecommissie:

Promotores:
Prof. dr. J. F. A. K. van Benthem
Prof. dr. D. H. J. de Jongh

Overige leden:
Prof. dr. P. W. Adriaans
Prof. dr. R. Clark
Prof. dr. V. F. Hendricks
Prof. dr. R. Scha
Dr. S. Smets
Prof. dr. R. Verbrugge

Faculteit der Natuurwetenschappen, Wiskunde en Informatica
Universiteit van Amsterdam
Science Park 904
1098 XH Amsterdam

Cover design by Nina Gierasimczuk.
Printed and bound by Ipskamp Drukkers.

ISBN: 978–90–5776–220–8

Contents

Acknowledgments

I am very grateful to my supervisors for their help and guidance. In particular, I would like to thank Prof. Dick de Jongh for showing me the value of patience and skepticism in scientific research, and for encouraging my interests in formal learning theory. To Prof. Johan van Benthem I am especially grateful for providing me with interesting challenges and opportunities, and for indicating the importance of the balance between setting boundaries and pursuing new ideas.

I am indebted to my co-authors. For the fruitful and inspiring collaboration, and all that I have gained through it I would like to thank: Dr. Alexandru Baltag, Dr. Cédric Dégremont, Prof. Dick de Jongh, Lena Kurzen, Dr. Sonja Smets, Dr. Jakub Szymanik, and Fernando Velázquez-Quesada.

There are also others that contributed to the shape of this book through discussions about various formal and philosophical issues, among them: Dr. Joel Uckelman (he also patiently proofread the dissertation), Umberto Grandi, Prof. Rineke Verbrugge, Yurii Khomski, Dr. Joanna Golińska-Pilarek, and Theodora Achourioti.

I would like to thank all members of faculty and staff, associates, and friends of the Institute of Logic, Language and Computation for creating a greatly exceptional academic and social environment.

To my closest family: my parents, Dr. Iwona and Dariusz, my sisters, Natalia and Marta and my godfather, Ryszard Peryt: I am grateful for recognizing my scientific interests, and for all other forms of appreciation and support. And thank you, Jakub, for being a great companion!

Nina Gierasimczuk
Amsterdam, November 2010

Part I

Setting and Motivation

Chapter 1

Introduction

This book is about change. Change of mind, revision of beliefs, formation of conjectures, and strategies for learning. We compare two major paradigms of formal epistemology that deal with the dynamics of informational states: formal learning theory and dynamic epistemic logic. Formal learning theory gives a computational framework for investigating the process of conjecture change (see, e.g., Jain, Osherson, Royer, & Sharma, 1999). With its central notion of identification in the limit (Gold, 1967), it provides direct implications for the analysis of language acquisition (see, e.g., Angluin & Smith, 1983) and scientific discovery (see, e.g., Kelly, 1996). On the other hand, directions that explicitly involve notions of knowledge and belief have been developed in the area of philosophical logic. After Hintikka (1962) established a precise language to discuss epistemic states, the need of formalizing dynamics of knowledge emerged. The belief-revision AGM framework (Alchourrón, Gärdenfors, & Makinson, 1985) constitutes an attempt to talk about the dynamics of epistemic states. Belief-revision policies thus explained have been successfully modeled in dynamic epistemic logic (see Van Benthem, 2007), which investigates the change in the context of multi-agent systems. Recent attempts to accommodate *iterated* knowledge and belief change is where epistemic logic meets learning theory.

Although the two paradigms are interested in similar and interrelated questions, the communication between formal learning theory and dynamic epistemic logic is difficult, mostly because of the differences in their methodologies. Learning theory is concerned with the global process of convergence in the context of computability. Belief-revision focuses on single steps of revision and constructive manners of obtaining new states, and the perspective here is more logic- and language-oriented.

Learning theory has been formed as an attempt to formalize and understand the process of language acquisition. In accordance with his nativist theory of language and his mathematical approach to linguistics, Chomsky (1965) proposed the existence of what he called a *language acquisition device*, a module that humans are born with, an 'innate facility' for acquiring language. This turned out to be only a step away from the formal definition of language learners as functions, that on ever larger and larger finite samples of a language keep outputting conjectures—grammars (supposedly) corresponding to the language in question. The generalization of this concept in the context of computability theory has taken the learners to be number-theoretic functions that on finite samples of a recursive set output indices that encode Turing machines, in an attempt to find an index of a machine that generates the set. In analogy to a child, who on the basis of finite samples learns to creatively use a language, by inferring an appropriate set of rules, learning functions are supposed to stabilize on a value that encodes a finite set of rules for generating the language.

Learning theory poses computational constraints. Learning functions are most often identified with computational devices, and this leads to assuming their recursivity. There are at least three mutually related reasons why learning theory has been developed in this direction. One comes from cognitive science: Church's Thesis in its psychological version; one is practical: the need of implementing learning algorithms; and finally there is a theoretical one: limiting recursion is in itself a mathematically interesting subject for logic and theoretical computer science.

Church's Thesis says that the human mind can only deal with computable problems. This statement underlies the very popular view about the analogy between minds and Turing machines (for an extensive discussion see Szymanik, 2009). This assumption is compatible with investigations into the implementations of learning procedures as effective algorithms. For similar reasons also the structures that are being learned are often considered to be computable—indeed, they are handled by minds which compute, or by algorithms. However restrictive these computability conditions might seem, learning remains a phenomenon of high complexity. Identification in the limit (Gold, 1967), the classical definition of successful learning, requires that the conjectures of learning functions, after some initial mind-changes, stabilize on the correct hypothesis. This exceeds computable resources, in fact it is an uncomputable, recursive in the limit, condition: there is a step k such that for all steps $n > k$ the computable learning function L outputs the correct hypothesis. Therefore, the question whether a structure is learnable falls outside the range of computable problems. Classes of sets for which such learning functions exist, i.e., learnable classes, constitute the domain of limiting recursion theory, an autonomous topic of research in theoretical computer science.

Summing up, the motivation of language acquisition initially directed learning considerations towards a recursive framework, with agents represented as certain type of number-theoretic functions. The discipline has been restricted to the functions that satisfy the limiting conditions of convergence on certain data structures. One might say that the domain has been taken over by successful, ultimately *reliable* functions (for learning theory in terms of reliability see, e.g., Kelly, 1998a). The observation that reliability is the feature that distinguishes successful learning functions from other possible mind-change policies led to relaxing the recursive paradigm. Learning theory has been re-interpreted as the framework for analyzing the procedural aspects of science, and became a study of information flow and general inquiry. This resulted in the treatment of formal learning theory as the mathematical embodiment of a normative epistemology.[1] In philosophy of science and general epistemology there is no need to assume that theory change is governed by a *computable* function. Immediately after dropping the heavy machinery of computability, learning theory linked to the problematics of knowledge and belief revision (see, e.g., Hendricks, 1995; Jain et al., 1999; Kelly, 1996), with attempts to plug the ready-to-use framework of successful convergence into the considerations of iterated belief-revision.

On the other side, a logical approach to belief-revision has been proposed in the so-called AGM framework (Alchourrón et al., 1985), where the beliefs of an agent are represented as a logically closed set of sentences of a particular language. A (new) belief-representing sentence gets introduced to the set and causes a belief change, which often leads to the necessity of removals to keep the beliefs consistent. AGM theory provides a set of axioms that put some rationality constraints on such revisions and allow the evaluation of various belief-revision policies. Presently, a very promising direction of combining the belief-revision framework with modal logics of knowledge and belief gives us a way to investigate revisions in a more linguistically-detached way. In this thesis we will look at these problems from a recently developed perspective of dynamic epistemic logic.

The framework of dynamic epistemic logic comprises a family of logics of explicit informational actions and corresponding knowledge and belief changes in agents. The information flow consisting of update actions performed in a stepwise manner can be defined as transformations of models. Those transformations can be studied and analyzed explicitly by combining techniques from epistemic, doxastic, and dynamic logic. Being logics, dynamic epistemic systems come with a semantics, but also with syntax: a formal language and a proof theory. Interestingly, like in learning theory, one of the sources is natural language and communication, but others include epistemology, and theories of agency in computer science (in particular Baltag, Moss, & Solecki, 1998; Gerbrandy, 1999a, developed basic update mechanisms that will be used in this thesis). By now many authors see dynamic epistemic logic as a general theory of social information- and preference-driven agency, which has led to growing links with temporal logics, game theory, and other formal theories of interaction (see Van Benthem, 2010). All these more recent themes will return at places in this dissertation.

This thesis brings learning theory and dynamic epistemic logic together on two levels. The first link is *semantic*. We combine local update mechanisms of dynamic epistemic logic, that constitute constructive step-by-step changes of current epistemic states, with the long-term temporal modeling offered by

[1] For the characteristics and history of this line of research see, e.g., the Stanford Encyclopedia of Philosophy entry *Formal Learning Theory* by Oliver Schulte.

learning theory. In terms of benefits of the paradigms, learning theory receives the fine-structure of well-motivated local learning actions[2], and dynamic epistemic logic gets a long-term 'horizon' which it missed (this approach is developed in Chapters 3 and 4). The second link is *syntactic.* Dynamic epistemic logic has its syntax and proof theory, learning theory does not. We show how basic notions of learning theory can be given simple perspicuous qualitative formulations in dynamic epistemic languages (the syntactic link is developed in Chapter 5). In the long run this perspective offers a chance of generic reasoning calculi about inductive learning.

***

The content of this thesis is organized in three parts. Let us give a brief overview of the chapters.

In Part I we introduce the setting and the motivation of the thesis. Chapter 2 gives mathematical preliminaries to the basic frameworks of formal learning theory and logics of knowledge and belief. Chapter 3 is intended to methodologically compare the two frameworks and provide a conceptual 'warming up' for the next part.

Part II is concerned with generally understood definability notions: expressing learnability conditions in the language of epistemic and doxastic logic. Chapter 4 gives a dynamic epistemic logic account of iterated belief-revision. By reinterpreting belief-revision policies as learning methods, we evaluate update, lexicographic and minimal upgrades with respect to their reliability on different kinds of incoming information. We are mainly concerned with identifiability in the limit. In the first part we restrict ourselves to learning from sound and complete streams of positive data. We show that learning methods based on belief revision via conditioning (update) and lexicographic revision are universal, i.e., provided certain prior conditions, those methods are as powerful as identification in the limit. We show that in some cases, these priors cannot be modeled using standard belief-revision models (as based on well-founded preorders), but only using generalized models (as simple preorders). Furthermore, we draw conclusions about the existence of tension between conservatism and learning power by showing that the very popular, most 'conservative' belief-revision method fails to be universal. In the second part we turn to the case of learning from both positive and negative data, and we draw conclusions about iterated belief revision governed by such streams. This enriched framework allows us to consider the occurrence of erroneous information. Provided that errors occur finitely often and are always eventually corrected we show that the lexicographic revision method is still reliable, but more conservative methods fail.

[2] One approach to learning theory, *learning by erasing* (see Section 2.1), uses update-like actions of hypotheses deletion.

In Chapter 5 we are again concerned with learnability properties analyzed in the context of epistemic and doxastic logic. We study both finite identification and identification in the limit. We represent the initial uncertainty of the learner as an epistemic model and characterize the conditions of the emergence of irrevocable knowledge in epistemic and dynamic epistemic logic. Then, we move to the case of identifiability in the limit and we give a doxastic logic characterization of the conditions required for converging to true stable belief. Following recent results on the correspondence between dynamic epistemic and temporal epistemic logics, we also give a characterization of learnability in terms of temporal protocols. We use the fact that the identification of sets can be performed by means of epistemic update. In the general context of learnability of protocols we characterize finite and limiting identification in an epistemic temporal and doxastic temporal language. Our temporal logic based approach to inductive inference gives a straightforward framework for analyzing various domains of learning on a common ground.

Part III consists of concrete case studies developing the general bridge that we built further, while also adding new themes. In Chapter 6 we are concerned with the problem of obtaining and using minimal samples of information that allow reaching certainty (i.e., allow finite identification). With the notion of eliminative power of incoming information, we analyze the computational complexity of finding such minimal samples. The problem of finding minimal-sized samples turns out to be NP-complete. Moreover, in the general case, we show that if we assume learners to be recursive, there are situations in which full certainty can be obtained in a computable way, but it cannot be computably realized by the learner at the first possible moment, i.e., as soon as the objective ambiguity between possibilities disappears. We also investigate different types of preset learners, that are tailored to use the knowledge of such minimal samples in their identification procedure. Differences in computational complexity between reaching certainty and reaching it in the optimal way give a motivation for explicitly introducing a new agent, a teacher, and provide a computational analysis of teachability.

In Chapter 7 we abstract away from the cooperativeness of the learner and the teacher, the property that is uniformly assumed in learning theory. We investigate the interaction between them in a particular kind of supervision learning games based on sabotage games. We are interested in the complexity of teaching, which we interpret in a similar way as in Chapter 6. Assuming the global perspective of the teacher, we treat the teachability problem as deciding whether the learning process can possibly be successful. We interpret learning as a game and hence we identify learnability and teachability with the existence of winning strategies in those games. In this context, we analyze different learning and teaching attitudes, varying the level of the teacher's helpfulness and the learner's willingness to learn. We use sabotage modal logic to reason about these games and, in particular, we identify formulae of the language that characterize the existence of winning strategies in each of the scenarios. We provide the complexity results for the related model-checking problems. They support the intuition that the cooperation of agents facilitates learning. Additionally, we observe the asymmetric nature of the moves of the two players and investigate a version without strict alternation of moves.

Finally, in Chapter 8 we consider another type of inductive inference that consists of iterated epistemic reasoning in multi-agent scenarios. We generalize the Muddy Children puzzle to treat arbitrary quantifiers in Father's announcement. Each child in the puzzle is viewed as a scientist who tries to inductively decide a hypothesis. The interconnection with other scientists can influence the discovery in a positive way. We characterize the property that makes quantifier announcements relevant in an epistemic context. In particular, we show what makes them prone to the occurrence of iteration of epistemic reasoning. The most immediate contribution to dynamic epistemic logic is a concise, linear representation of the epistemic situation of the Muddy Children. Moreover, we give a characterization of the solvability of the Muddy Children puzzle and a uniform way of deciding how many steps of iterated epistemic reasoning are needed for reaching the solution. This explicit, step by step analysis brings us closer to investigating the internal complexity of epistemic problems that the agents are facing and allows a comparison with computational complexity results from the domain of natural language quantifier processing.

Chapter 9 concludes the thesis by giving an overview of results and open questions.

As the reader may have observed from the above overview, the topics of this dissertation are drawn mainly from the domain of logic and theoretical computer science, at points reaching out to game theory and cognitive science. The approach is highly interdisciplinary. Even though the author's goal was to make this thesis self-contained, the reader is still assumed to be acquainted with basics of mathematical logic, computability and complexity theory.

Sources of the chapters

Chapter 3 is based on:

Gierasimczuk, N. (2009). Bridging learning theory and dynamic epistemic logic. *Synthese*, *169*(2), 371–384.

Gierasimczuk, N. (2009). Learning by erasing in dynamic epistemic logic. In *LATA'09: Proceedings of 3rd International Conference on Language and Automata Theory and Applications*, vol. 5457 of *LNCS*, (pp. 362–373). Springer.

Chapter 4 is based on:

Baltag, A., Gierasimczuk, N., & Smets, S. (2010). Truth tracking and belief revision. Manuscript. Presented at NASSLLI'10, Bloomington.

Chapter 5 is based on:

Dégremont, C., & Gierasimczuk, N. (2009). Can doxastic agents learn? On the temporal structure of learning. In X. He, J. F. Horty, & E. Pacuit (Eds.) *LORI'09: Proceedings of 2nd International Workshop on Logic, Rationality, and Interaction*, vol. 5834 of *LNCS*, (pp. 90–104). Springer.

Dégremont, C., & Gierasimczuk, N. (2010). Finite identification from the viewpoint of epistemic update. To appear in *Information and Computation*.

Chapter 6 is based on:

Gierasimczuk, N., & de Jongh, D. (2010). On the minimality of definite finite tell-tale sets in finite identification of languages. *The Yearbook of Logic and Interactive Rationality*, (pp. 26–41). Institute for Logic, Language and Computation, Universiteit van Amsterdam.

Chapter 7 is based on:

Gierasimczuk, N., Kurzen, L., & Velázquez-Quesada, F. R. (2009). Learning and teaching as a game: A sabotage approach. In X. He, J. F. Horty, & E. Pacuit (Eds.) *LORI'09: Proceedings of 2nd International Workshop on Logic, Rationality, and Interaction*, vol. 5834 of *LNCS*, (pp. 119–132). Springer.

Gierasimczuk, N., Kurzen, L., & Velázquez-Quesada, F. R. (2010). Games for learning: A sabotage approach. Submitted to *Logic Journal of the Interest Group in Pure and Applied Logic*.

Chapter 8 is based on:

Gierasimczuk, N., & Szymanik, J. (2010). Muddy Children Playground: Number Triangle, Internal Complexity, and Quantifiers. Presented at *Logic, Rationality and Intelligent Interaction Workshop*, ESSLLI'10, Copenhagen.

Chapter 2

Mathematical Prerequisites

This chapter gathers background information on the two major paradigms discussed and linked in this thesis. First, preliminaries of formal learning theory are given. Then we discuss the basics of dynamic epistemic logic approaches to information and belief change. In both cases the existing literature varies in basic notions and notation. The decisions taken in this chapter should be viewed as defining the framework and laying the grounds for this thesis, rather than restricting the paradigms or indicating a general preference. For exhaustive overviews and references the reader is advised to consult respectively (Jain et al., 1999) and (Van Ditmarsch, Van der Hoek, & Kooi, 2007).

2.1 Learning Theory

Learning theory is concerned with sequences of outputs of recursive functions, focusing on those that stabilize on an appropriate value (Gold, 1967; Putnam, 1965; Solomonoff, 1964a,b). As mentioned in the introduction, the general motivation here is the possibility of inferring general conclusions from partial, inductively given information, as in the case of language learning (inferring grammars from sentences) and scientific inquiry (drawing general conclusions from partial experiments). These processes can be thought of as games between Scientist (Learner) and Nature (Teacher). At the start there is a class of possible worlds, or a class of hypotheses. It is assumed that both Scientist and Nature know what those possibilities are, i.e., they both have access to the initial class. Nature chooses one of those possible worlds to be the actual one. Scientist's aim is to guess which one it is. He receives information about the world in an inductive manner. The stream of data is infinite and contains only and all the elements from the chosen reality. Each time Scientist receives a piece of information he answers with one of the hypotheses from the initial class. We say that Scientist identifies Nature's choice in the limit if after some finite number of guesses his answers stabilize on a correct hypothesis. Moreover, it is required that the same is true for all the possible worlds from the initial class, i.e., regardless of which element from the class is chosen by Nature to be true, Scientist can identify it in the limit. In what follows, the possibilities are taken to be sets of integers, and they will be often called *languages.*

Let $U \subseteq \mathbb{N}$ be an infinite recursive set; we call any $S \subseteq U$ a language. In the general case, we will be interested in indexed families of recursive languages, i.e., classes $\mathcal{C}$ for which a computable function $f : \mathbb{N} \times U \to \{0,1\}$ exists that uniformly decides $\mathcal{C}$, i.e.,

$$f(i,w) = \begin{cases} 1 & \text{if } w \in S_i, \\ 0 & \text{if } w \notin S_i. \end{cases}$$

In large parts of this thesis we will also consider $\mathcal{C}$ to be $\{S_1, S_2, \ldots, S_n\}$, a finite class of finite sets, in which case we will use $I_{\mathcal{C}}$ for the set containing indices of sets in $\mathcal{C}$, i.e., $I_{\mathcal{C}} = \{1, \ldots, n\}$. We will often refer to the setting in which the possible realities are taken to be sets using the terms *language learning* or *set learning*.

The global input for Scientist is given as an infinite stream of data. In learning theory, such streams are often called *texts* (positive presentations).[1]

Definition 2.1.1. *By a* text (positive presentation) ε *of* S *we mean an infinite sequence of elements from* S *enumerating all and only the elements from* S (*allowing repetitions*).

Definition 2.1.2. *We will use the following notation:*

- ε_n *is the* n*-th element of* ε*;*
- $\varepsilon \restriction n$ *is the sequence* $(\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n)$*;*
- $\mathrm{set}(\varepsilon)$ *is the set of elements that occur in* ε*;*
- *Let* U^* *be the set of all finite sequences over* U*. If* $\alpha, \beta \in U^*$*, then by* $\alpha \sqsubset \beta$ *we mean that* α *is a proper initial segment of* β*.*
- L *is a learning function—a recursive map from finite data sequences to indices of hypotheses,* $L : U^* \to \mathbb{N}$*. We will sometimes take the learning function to be* $L : U^* \to \mathbb{N} \cup \{\uparrow\}$*. Then the function is allowed to refrain from giving a natural number answer, in which case the output is marked by* $\uparrow$*, but the function remains recursive.*[2] *We sometimes relax the condition of recursivity of* L *to discuss some cases of non-effective finite identifiability.*

[1] We will be mainly concerned with sequences of positive information, *texts.* They are sometimes also known under the name of *environments* (see, e.g., Jain et al., 1999). The type of information that, besides positive, includes also negative information is usually called an *informant.*

[2] The symbol $\uparrow$ in the context of learning functions should not be read as a calculation that does not stop.

- *Let $T \subseteq \mathbb{N}$ be a finite set. Then $\hat{T}$ is the finite sequence such that* $\mathrm{set}(\hat{T}) = T$ *and* $\mathrm{length}(\hat{T}) = |T|$*, where $|\cdot|$ stands for the cardinality of a set, and $\hat{T}$ enumerates the integers from T in increasing order.*

2.1.1 Finite Identification

Finite identifiability of a class of languages from positive data is defined by the following chain of conditions.

Definition 2.1.3. *A learning function L:*

1. *finitely identifies $S_i \in \mathcal{C}$ on ε iff, when inductively given ε, at some point L outputs a single value i;*

2. *finitely identifies $S_i \in \mathcal{C}$ iff it finitely identifies S_i on every ε for S_i;*

3. *finitely identifies $\mathcal{C}$ iff it finitely identifies every $S_i \in \mathcal{C}$.*

A class $\mathcal{C}$ is finitely identifiable iff there is a learning function L that finitely identifies $\mathcal{C}$.

Example 2.1.4. *Let $\mathcal{C}_1 = \{S_i = \{0, i\} \mid i \in \mathbb{N} - \{0\}\}$. $\mathcal{C}_1$ is finitely identifiable by the following function $L : U^* \to \mathbb{N} \cup \{\uparrow\}$:*

$$L(\varepsilon{\restriction}n) = \begin{cases} \uparrow & \text{if } \mathrm{set}(\varepsilon{\restriction}n) = \{0\} \text{ or } \exists k < n \; L(\varepsilon{\restriction}k) \neq \uparrow, \\ \max(\mathrm{set}(\varepsilon{\restriction}n)) & \text{otherwise.} \end{cases}$$

In other words, L outputs the correct hypothesis as soon as it receives a number different than 0, and the procedure ends.

To see how restrictive this notion is, we can consider a finite class of languages that is not finitely identifiable.

Example 2.1.5. *Let $\mathcal{C}_2 = \{S_i = \{0, \ldots, i\} \mid i \in \{1, 2, 3\}\}$. $\mathcal{C}_2$ is not finitely identifiable. To see that, assume that $S_2 = \{0, \ldots, 2\}$ is chosen to be the actual world. Then the learning function can never conclusively decide that S_2 is the actual language. For all it knows, 3 might appear in the future, so it has to leave the S_3-possibility open.*

A necessary and sufficient condition for finite identifiability has already been formulated in the literature (Lange & Zeugmann, 1992; Mukouchi, 1992).

Definition 2.1.6 (Mukouchi 1992)**.** *A set D_i is a definite finite tell-tale set for $S_i \in \mathcal{C}$ if*

1. *$D_i \subseteq S_i$,*

2. *D_i is finite, and*

3. *for any index j, if $D_i \subseteq S_j$ then $S_i = S_j$.*

On the basis of this notion, finite identifiability can be then characterized in the following way.

Theorem 2.1.7 (Mukouchi 1992)**.** *A class $\mathcal{C}$ is finitely identifiable from positive data iff there is an effective procedure $\mathcal{D} : \mathbb{N} \rightarrow \mathcal{P}^{<\omega}(\mathbb{N})$, given by $n \mapsto \mathcal{D}_n$, that on input i produces a definite finite tell-tale of S_i.*

In other words, each set in a finitely identifiable class contains a finite subset that distinguishes it from all other sets in the class. Moreover, for the effective identification it is required that there is a *recursive* procedure that provides such definite finite tell tale-set.

2.1.2 Identification in the Limit

Let us consider again Example 2.1.5, i.e., take $\mathcal{C}_2 = \{S_i = \{0, \ldots, i\} \mid i \in \{1, 2, 3\}\}$, but now assume that S_2 is the actual language. Then Scientist cannot conclusively decide that it is the case. There is however a way to deal with this kind of uncertainty. Namely, if we allow Scientist to answer each time he gets a new piece of data, we can define the success of learning using the notion of *convergence* to the right answer. After seeing 0, 1 and 2 Learner can keep conjecturing S_2 indefinitely, because in fact 3 will never appear. This leads to the notion of identification in the limit.

Definition 2.1.8 (Identification in the limit (Gold, 1967))**.** *Learning function L:*

1. *identifies $S_i \in \mathcal{C}$ in the limit on ε iff for co-finitely many m, $L(\varepsilon{\restriction}m) = i$;*

2. *identifies $S_i \in \mathcal{C}$ in the limit iff it identifies S_i in the limit on every ε for S_i;*

3. *identifies $\mathcal{C}$ in the limit iff it identifies in the limit every $S_i \in \mathcal{C}$.*

A class $\mathcal{C}$ is identifiable in the limit iff there is a learning function that identifies $\mathcal{C}$ in the limit.

Below we give some examples of classes of languages which are identifiable in the limit. First let us consider an example of a finite class of finite sets.

Example 2.1.9. *Recall the class $\mathcal{C}_2$ from the previous example. $\mathcal{C}_2$ is identifiable in the limit by the following function $L : U^* \rightarrow \mathbb{N}$:*

$$L(\varepsilon{\restriction}n) = \max(\text{set}(\varepsilon{\restriction}n)).$$

We can use the same learning function to identify an infinite class of finite sets.

Example 2.1.10. *Let $\mathcal{C}_3 = \{S_i \mid i \in \mathbb{N} - \{0\}\}$, where $S_n = \{1, \ldots, n\}$.*

The property of identification in the limit of the class $\mathcal{C}_3$ is lost when we enrich it with the set of all natural numbers.

Example 2.1.11. *Let $\mathcal{C}_4 = \{S_i \mid i \in \mathbb{N}\}$, where $S_0 = \mathbb{N}$ and for $n \geq 1$, $S_n = \{1, \ldots, n\}$. $\mathcal{C}_4$ is not identifiable in the limit. To show that this is the case, let us assume that there is a function L that identifies $\mathcal{C}_4$. We will construct a text, ε on which L fails: ε starts by enumerating $\mathbb{N}$ in order: $0, 1, 2, \ldots$, if arriving at a number k, L decides it is S_0, we start repeating k indefinitely. This means we will have a text for S_k. As soon as L decides it is S_k we continue with $k+1, k+2, \ldots$, so we get a text for S_0, etc. This shows that there is a text for a set from $\mathcal{C}_4$ on which L fails.*

We have already seen an infinite class of finite sets that is identifiable in the limit. The next example shows an infinite class of *infinite* sets that is identifiable in the limit.

Example 2.1.12. *Let $\mathcal{C}_5 = \{S_n \mid S_n = \mathbb{N} - \{n\}, n \in \mathbb{N}\}$. $\mathcal{C}_5$ is identifiable in the limit by the learning function $L : U^* \to \mathbb{N}$:*

$$L(\varepsilon\restriction n) = \min(\mathbb{N} - \mathrm{set}(\varepsilon\restriction n)).$$

A characterization of classes that are identifiable in the limit can be given in terms of *finite tell-tale sets*[3] (Angluin, 1980).

Definition 2.1.13 (Angluin 1980)**.** *A set D_i is a finite tell-tale set for $S_i \in \mathcal{C}$ if*

1. *$D_i \subseteq S_i$,*
2. *D_i is finite, and*
3. *for any index j, if $D_i \subseteq S_j$ then $S_j \not\subset S_i$.*

Identifiability in the limit can be then characterized in the following way.

Theorem 2.1.14 (Angluin 1980)**.** *An indexed family of recursive languages $\mathcal{C} = \{S_i \mid i \in \mathbb{N}\}$ is identifiable in the limit from positive data iff there is an effective procedure $\mathcal{D}$, that on input i enumerates all elements of a finite tell-tale set of S_i.*

In other words, each set in a class that is identifiable in the limit contains a finite subset that distinguishes it from all its subsets in the class. Moreover, for the effective identification it is required that there is a *recursive* procedure that enumerates such finite tell-tales.

[3] The notion of *definite* finite tell-tale set from Definition 2.1.6 in the previous section, is a modification and strengthening of the presently discussed, original notion of finite-tell tale set.

2.1.3 Other Paradigms

Learning by Erasing Learning by erasing (Lange, Wiehagen, & Zeugmann, 1996) is an epistemologically intuitive modification of the identification in the limit. It has not drawn much attention in the field of formal learning theory but for our purposes (a comparison with the approach of dynamic epistemic logic) it is interesting. Very often the cognitive process of converging to a correct conclusion consists of eliminating those possibilities that are falsified during the inductive inquiry. Accordingly, in the formal model the outputs of the learning function are negative, i.e., the function each time eliminates a hypothesis, instead of explicitly guessing one that is supposed to be correct. The difference between the definition of this approach and the usual identification is in the interpretation of the conjecture of the learning function. In learning by erasing one assumes an ordering of the initial hypothesis space isomorphic to the natural numbers. This allows one to interpret the actual positive guess of the learning-by-erasing function to be the least hypothesis (in the given ordering) not yet eliminated.

Let us give now the two definitions that shape the notion of learning by erasing.

Definition 2.1.15 (Function stabilization)**.** *In learning by erasing we say that a function stabilizes to number k on environment ε iff for co-finitely many $n \in \mathbb{N}$:*

$$k = \min(\{\mathbb{N} - \{L(\varepsilon\restriction 1), \ldots, L(\varepsilon\restriction n)\}\}).$$

Definition 2.1.16 (Learning by erasing (Lange et al., 1996))**.** *We say that a learning function L:*

1. *learns $S_i \in \mathcal{C}$ by erasing on ε iff L stabilizes to i on ε;*
2. *learns $S_i \in \mathcal{C}$ by erasing iff it learns S_i by erasing from every ε for S_i;*
3. *learns $\mathcal{C}$ by erasing iff it learns every $S_i \in \mathcal{C}$ by erasing.*

A class $\mathcal{C}$ is learnable by erasing iff there is a learning function that learns $\mathcal{C}$ by erasing.

It is easy to observe that in this setting learnability heavily depends on the chosen *enumeration* of languages, since the positive conjecture of the learning function is interpreted as the minimal one that has not yet been eliminated.

Several types of learning by erasing have been proposed. They vary in the condition of which hypotheses the learning function is allowed to remove (for details and results on learning by erasing see Lange et al., 1996).

Function learning Let us now mention another paradigm of learning in the limit—function learning. This falls out of the scope of the language-learning paradigm, but the notion of identification is in its essence very similar. The success of learning is again defined in the limit as convergence to a correct hypothesis.

This time however we take possible realities to be total recursive functions. This can be made concrete in various ways. For instance, it has been considered as a way to model program synthesis in the context of learning and empirical inquiry (see, e.g., Jantke, 1979; Shapiro, 1998); in linguistics, the framework has been used to model language learning in the context of finding an appropriate assignment of deep syntactic structures to syntactic representations (for discussion see Wexler & Cullicover, 1980).

Since we consider a different type of structure here, we have to change the definition of text.[4]

Definition 2.1.17. *A* text of a function*, ε, is any infinite sequence over* $\mathbb{N} \times \mathbb{N}$ *(any infinite sequence of pairs of numbers), such that for each $x \in \mathbb{N}$ there is exactly one y such that (x, y) occurs in the sequence. In other words a text of a function g is any enumeration of the content of the graph of g.*

For text of functions we will use the notation introduced in Definition 2.1.2.

Let us take $\mathcal{C}_f$ to be a class of total recursive functions. For each $g \in \mathcal{C}_f$ we consider Turing machines φ_n which compute g. We take $I_g = \{n \mid \varphi_n \text{ computes } g\}$. Let us now assume that $g \in \mathcal{C}_f$ and ε is a text for g. We specify function identification in the limit by the following definition.

Definition 2.1.18 (Identification in the limit of functions)**.** *We say that a learning function L:*

1. *identifies a function $g \in \mathcal{C}_f$ in the limit on ε iff for co-finitely many m, $L(\varepsilon{\restriction}m) = k$ and $k \in I_g$;*
2. *identifies $g \in C$ in the limit iff it identifies g in the limit on every ε for g;*
3. *identifies C in the limit iff it identifies every $h \in C$ in the limit.*

Function learning differs from language learning in many respects. One of the most important differences lies in the specific properties of possible realities—functions. Namely, environments of functions carry more information than streams of data defined for set learning. In an environment for a total function it is enough to examine a finite fragment of the environment to decide whether a given pair (n, m) is in the whole sequence. That is so because in some finite fragment we can find either (n, m) itself or some (n, m') with $m \neq m'$. In the latter case it follows that (n, m) is not in the sequence. In language learning it is impossible to conclude the non-existence of an element in an environment on the basis of finite examination. This allows the class of all recursive functions to be identifiable in the limit (see Jain et al., 1999). Let us also mention that totality of functions implies that for every n, there is an m, such that (n, m) is an element of an

[4] Similarly to the case of set learning, we take a text to be a positive presentation of a function. We are not concerned here with negative information at all.

environment. Therefore, it makes little difference to the learning if the function is enumerated in order $(g(0), g(1), \ldots)$. In that case learning is equivalent to the ability to guess the next value of the function after a certain time.

2.2 Logics of Knowledge and Belief

Modal logics of epistemic change are used to analyze the information flow in multi-agent systems (see, e.g., Baltag et al., 1998; Van Benthem, Van Eijck, & Kooi, 2006; Gerbrandy, 1999a,b). The approach of *dynamic epistemic logic*, DEL for short, (Plaza, 1989, see also Van Ditmarsch et al., 2007 for a handbook presentation) focuses on formalizing the principles of such epistemic changes.

2.2.1 Epistemic Logic

Let us begin with the notion of epistemic model. In what follows $\mathcal{A} = \{1, \ldots, n\}$ is a finite set of agents and Prop is a countable set of propositional letters.

Definition 2.2.1. *An* epistemic model $\mathcal{M}$ *based on a set of agents* $\mathcal{A}$ *is a triple:*

$$(W, (\sim_i)_{i \in \mathcal{A}}, V),$$

where $W \neq \emptyset$ *is a set of states, for each* $i \in \mathcal{A}$, $\sim_i$ *is a binary equivalence relation on* W*, and* $V : \textsc{Prop} \to \mathcal{P}(W)$ *is a valuation.*

A pair $(\mathcal{M}, w)$*, where* $\mathcal{M} = (W, (\sim_i)_{i \in \mathcal{A}}, V)$ *is an epistemic model and* $w \in W$*, will be called a* pointed epistemic model.

The information that agent i possesses in state w is denoted by

$$\mathcal{K}_i[w] = \{v \in W \mid w \sim_i v\}.$$

It stands for all information within the uncertainty range of agent i with respect to w. Accordingly, the knowledge of agent i in state w consists of those statements that are true in all worlds he considers possible from state w. To explicitly talk about knowledge we will use the language of basic epistemic logic (see, e.g., Blackburn, Rijke, & Venema, 2001).

Definition 2.2.2 (Syntax of $\mathcal{L}_{\mathrm{EL}}$)**.** *The syntax of epistemic language* $\mathcal{L}_{\mathrm{EL}}$ *is defined as follows:*

$$\varphi := \ p \mid \neg\varphi \mid \varphi \vee \varphi \mid K_i\varphi$$

where $p \in \textsc{Prop}$, $i \in \mathcal{A}$*. We will write* $\top$ *for* $p \vee \neg p$ *and* $\bot$ *for* $\neg\top$*.*

Definition 2.2.3 (Semantics of $\mathcal{L}_{\mathrm{EL}}$)**.** *We interpret* $\mathcal{L}_{\mathrm{EL}}$ *in the states of epistemic models as follows.*

$$\begin{array}{lll} \mathcal{M}, w \models p & \text{iff} & w \in V(p) \\ \mathcal{M}, w \models \neg\varphi & \text{iff} & \text{it is not the case that } \mathcal{M}, w \models \varphi \\ \mathcal{M}, w \models \varphi \vee \psi & \text{iff} & \mathcal{M}, w \models \varphi \text{ or } \mathcal{M}, w \models \psi \\ \mathcal{M}, w \models K_i\varphi & \text{iff} & \text{for all } v \text{ such that } w \sim_i v \text{ we have } \mathcal{M}, v \models \varphi \end{array}$$

Let us now provide an axiomatic system for epistemic logic EL (see, e.g., Blackburn et al., 2001).

PL	$\vdash \varphi$ if φ is a substitution instance of a tautology of propositional logic
Nec	if $\vdash \varphi$, then $\vdash K_i\varphi$
K	$\vdash K_i(\varphi \to \psi) \to (K_i\varphi \to K_i\psi)$
T	$\vdash K_i\varphi \to \varphi$
4	$\vdash K_i\varphi \to K_iK_i\varphi$
5	$\vdash \neg K_i\varphi \to K_i\neg K_i\varphi$
MP	if $\vdash \varphi \to \psi$ and $\vdash \varphi$, then $\vdash \psi$

Theorem 2.2.4. *The axiomatic system* EL *is complete with respect to the class of epistemic models.*

Epistemic Update

Epistemic models are static—they represent the informational state of an agent in temporal isolation. We will now make the setting more dynamic by assuming that agents observe some incoming data and are allowed to revise their informational states. We will consider *update* (see Van Benthem, 2007)—a policy that restricts models; each time a piece of data is encountered, it is assumed to be truthful and all worlds of the epistemic model that do not satisfy this new information are eliminated. The definition below formalizes the notion of update with a formula φ.

Definition 2.2.5. *The update of an epistemic model $\mathcal{M} = (W, (\sim_i)_{i\in\mathcal{A}}, V)$ with a formula φ, restricts $\mathcal{M}$ to those worlds that satisfy φ, formally $\mathcal{M} \mid \varphi = \mathcal{M}' := (W', (\sim'_i)_{i\in\mathcal{A}}, V')$,*

1. $W' = \{w \in W \mid w \models \varphi\}$;
2. *for each* $i \in \mathcal{A}$, $\sim'_i = \sim_i\restriction W'$;
3. $V' = V\restriction W'$.

Obviously, the incoming information that triggers update need not be propositional, not even purely linguistic. It can be any *event* that itself has an epistemic structure.[5] Below we consider a quite challenging case of an update with epistemic information.

[5] To consider changes caused by such arbitrary events, the notion of *event model* and *product update* has been introduced (Baltag et al., 1998). The former represents the epistemic content of an event, the latter stands for combining an epistemic model with an event model.

Muddy Children We want to devote some space to the classical logical puzzle which received a considerable amount of attention in dynamic epistemic logic (see, e.g., Van Ditmarsch et al., 2007; Gerbrandy, 1999a; Moses, Dolev, & Halpern, 1986). We discuss it here to give a flavor of complicated epistemic reasoning that can be successfully analyzed within DEL framework. We will return to the puzzle in the last chapter of this thesis, where we also propose a novel representation of this problem.

Example 2.2.6 (Muddy Children Puzzle)**.** *The children, who were playing outside for a while, are called back in by their father. Some of them are dirty, in particular they have mud on their foreheads. The father decides to play with them and says:*

(1) *At least one of you has mud on your forehead.*

And immediately after, he asks:

(**I**) *Can you tell for sure whether or not you have mud on your forehead? If yes, step forward and announce your status.*

Each child can see the mud on others but cannot see his or her own forehead. Nothing happens. After that the father repeats **I**. *Still nothing. But after he repeats the question three times suddenly all children know whether or not they have mud on their forehead. How is that possible?*

The framework of dynamic epistemic logic allows a clear and comprehensive explanation of the underlying phenomena. Let us briefly explain the classical modeling. Assume there are three children, let us call them a, b and c, and assume that, in fact, all of them are muddy. We will take three propositional letters m_a, m_b and m_c that express that the corresponding child is muddy. The initial epistemic model of the situation is depicted in Figure 2.1.

In the model, possible worlds correspond to the 'distribution of mud' on children's foreheads, e.g., $m_a, \neg m_b, \neg m_c$ stands for a being muddy and b and c being clean. Two worlds are joined with an edge labeled with x, if the two worlds are in the uncertainty range of agent x (i.e., if agent x cannot distinguish between the two worlds). We drop the reflexive arrows for each state for clarity of the presentation. The boxed state stands for the actual world. Now, let us see what happens after the first announcement is made.

(1) At least one of you has mud on your forehead.

In propositional logic, this statement has the following form: (1') $m_a \vee m_b \vee m_c$. Since the children trust their father, they all eliminate world w_8 in which (1') is false: none of the children is muddy. In other words, they perform an update with formula (1'). The result is depicted in Figure 2.2.

Now the father asks for the first time:

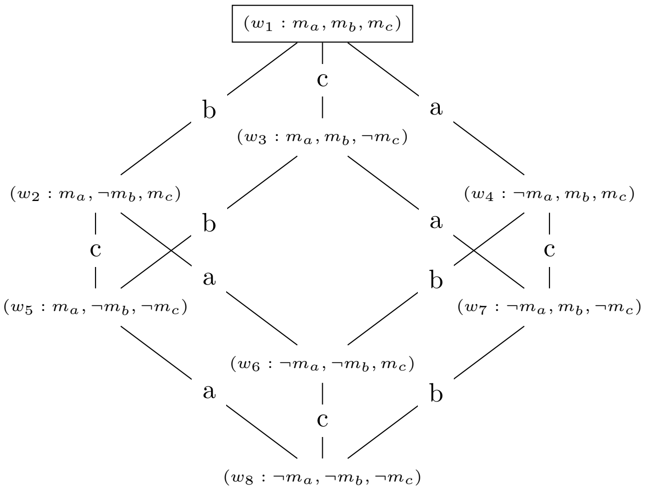

$(w_1 : m_a, m_b, m_c)$
$(w_2 : m_a, \neg m_b, m_c)$
$(w_3 : m_a, m_b, \neg m_c)$
$(w_4 : \neg m_a, m_b, m_c)$
$(w_5 : m_a, \neg m_b, \neg m_c)$
$(w_6 : \neg m_a, \neg m_b, m_c)$
$(w_7 : \neg m_a, m_b, \neg m_c)$
$(w_8 : \neg m_a, \neg m_b, \neg m_c)$
a
b
c

Figure 2.1: Initial epistemic model of the Muddy Children puzzle

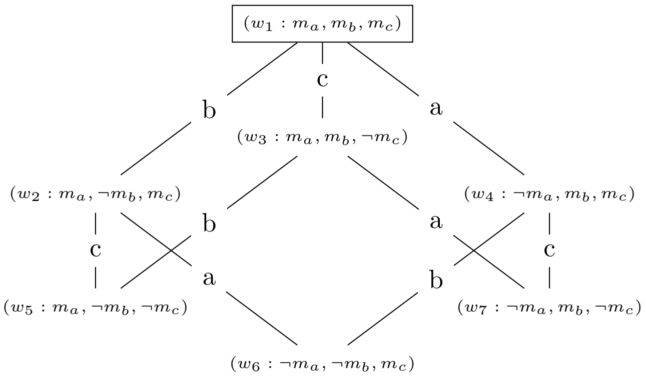

$(w_1 : m_a, m_b, m_c)$
$(w_2 : m_a, \neg m_b, m_c)$
$(w_3 : m_a, m_b, \neg m_c)$
$(w_4 : \neg m_a, m_b, m_c)$
$(w_5 : m_a, \neg m_b, \neg m_c)$
$(w_6 : \neg m_a, \neg m_b, m_c)$
$(w_7 : \neg m_a, m_b, \neg m_c)$
a
b
c

Figure 2.2: Epistemic model after father's announcement

(I) Can you tell for sure whether or not you have mud on your head?

The agents' reasoning can be as follows. In world w_6 agent c knows that he is dirty (there is no uncertainty of agent c between this world and another in which he is clean). Therefore, if the actual world was w_6, agent c would know his state and announce it. The same holds for agents a and b and worlds w_5 and w_7, respectively. But in our story children stay silent. This is in fact equivalent to the announcement that none of the children know whether they are muddy or not. Formally: $\neg(K_a m_a \vee K_a \neg m_a) \wedge \neg(K_b m_b \vee K_b \neg m_b) \wedge \neg(K_c m_c \vee K_c \neg m_c)$. Now all agents eliminate those worlds that do not satisfy this formula: w_5, w_6, w_7. The epistemic model of the next stage is smaller by three worlds (Figure 2.3).

At this stage it is again clear that if one of the w_2, w_3, w_4 was the actual state the respective agent would have announced their knowledge. But in our scenario

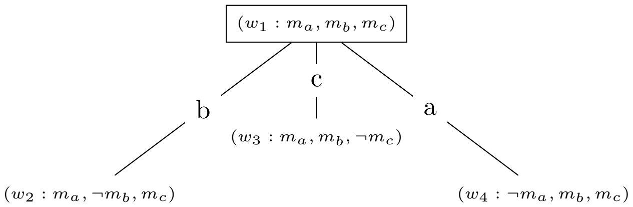

$(w_1 : m_a, m_b, m_c)$
c
b
a
$(w_3 : m_a, m_b, \neg m_c)$
$(w_2 : m_a, \neg m_b, m_c)$
$(w_4 : \neg m_a, m_b, m_c)$

Figure 2.3: Epistemic model in the second stage of epistemic inference

the children still do not respond. Then the father asks again: ‘Can you tell for sure whether or not you have mud on your forehead?’. Now the children base their inference on the silence in the previous step, and come to the conclusion that the actual situation cannot be any of w_2, w_3, w_4. So, they all eliminate the three states, which leaves them all with just one possibility (Figure 2.4). All uncertainty disappears and they all know that they are dirty.

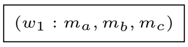


Figure 2.4: Epistemic model in the third stage of epistemic inference

Public Announcement

All announcements made in the above scenario trigger an update of the epistemic model according to Definition 2.2.5. The public character of the announcements makes them influence all agents’ uncertainty ranges. Basic epistemic logic, as defined above, can be extended to account for this type of update with a specific ‘action’ expression of *public announcement*, written as $!\varphi$.

Definition 2.2.7 (Syntax of $\mathcal{L}_{\text{PAL}}$)**.** *The syntax of epistemic language $\mathcal{L}_{\text{PAL}}$ is defined as follows:*

$$\begin{aligned} \varphi &:= \ p \mid \neg\varphi \mid \varphi \vee \varphi \mid K_i\varphi \mid [A]\varphi \\ A &:= \ !\varphi \end{aligned}$$

where $p \in$ PROP, $i \in \mathcal{A}$.

Definition 2.2.8 (Semantics of $\mathcal{L}_{\text{PAL}}$)**.** *For the epistemic fragment $\mathcal{L}_{\text{EL}}$ the interpretation is given in Definition 2.2.3. The remaining clause of $\mathcal{L}_{\text{PAL}}$ is as follows.*

$$\mathcal{M}, w \models [!\varphi]\psi \quad \text{iff} \quad \text{if } \mathcal{M}, w \models \varphi \text{ then } \mathcal{M} \mid \varphi, w \models \psi$$

An axiomatization PAL of $\mathcal{L}_{\text{PAL}}$ can be composed of the previously given axioms of epistemic logic enriched with the following reduction axioms (Plaza, 1989).

$$\begin{array}{ll}
1 & \vdash [!\varphi]p \leftrightarrow (\varphi \to p), \ \text{for } p \in \textsc{Prop} \\
2 & \vdash [!\varphi]\neg\psi \leftrightarrow (\varphi \to \neg[!\varphi]\psi) \\
3 & \vdash [!\varphi](\psi \vee \xi) \leftrightarrow ([!\varphi]\psi \vee [!\varphi]\xi) \\
4 & \vdash [!\varphi]K_i\psi \leftrightarrow (\varphi \to K_i[!\varphi]\psi)
\end{array}$$

Theorem 2.2.9 (Plaza 1989)**.** *The axiomatic system* PAL *is complete with respect to the class of epistemic models.*

The change that epistemic models undergo when subjected to public announcement corresponds to the revision with so-called 'hard' information. Such a revision is reasonable if the information originates from a reliable source.

2.2.2 Doxastic Logic

The notion of irrevocable knowledge defined in the previous subsection is very strong. It implicitly indicates that unless complete certainty is reached, the agent does not form any opinion on the state of the world. In order to talk about weaker informational states, like belief, epistemic models have to be modified to account for the order on states given by agents' doxastic attitudes.

Definition 2.2.10 (Baltag & Smets 2006)**.** *An* epistemic-plausibility model $\mathcal{M}$ *is a triple*

$$(W, (\sim_i)_{i\in\mathcal{A}}, (\leq_i)_{i\in\mathcal{A}}, V),$$

where $W \neq \emptyset$ is a set of states, for each $i \in \mathcal{A}$, $\leq_i$ is a total well-founded preorder[6] *on W, and $V : \textsc{Prop} \to \mathcal{P}(W)$ is a valuation.*

A pair $(\mathcal{M}, w)$, where $\mathcal{M} = (W, (\sim_i)_{i\in\mathcal{A}}, (\leq_i)_{i\in\mathcal{A}}, V)$ an epistemic plausibility model and $w \in W$, is called a pointed epistemic plausibility model.

For each $i \in \mathcal{A}$ we will assume that $\leq_i \subseteq \sim_i$.

Now the language of epistemic logic can be extended to account for belief.

Definition 2.2.11 (Syntax of $\mathcal{L}_{\text{DOX}}$)**.** *The syntax of doxastic-epistemic language $\mathcal{L}_{\text{DOX}}$ is defined as follows:*

$$\varphi := \ p \mid \neg\varphi \mid \varphi \vee \varphi \mid K_i\varphi \mid B_i^{\psi}\varphi$$

where $p \in \textsc{Prop}$, $i \in \mathcal{A}$.

[6] A preorder is a binary relation that is reflexive and transitive. Later we will relax the restriction to well-founded preorders and adjust the relevant definitions.

Definition 2.2.12 (Semantics of $\mathcal{L}_{\text{DOX}}$)**.** *We interpret $\mathcal{L}_{\text{DOX}}$ in the states of doxastic-epistemic models in the following way.*

$$\begin{array}{lll}
\mathcal{M}, w \models p & \text{iff} & w \in V(p) \\
\mathcal{M}, w \models \neg\varphi & \text{iff} & \text{it is not the case that } \mathcal{M}, w \models \varphi \\
\mathcal{M}, w \models \varphi \vee \psi & \text{iff} & \mathcal{M}, w \models \varphi \text{ or } \mathcal{M}, w \models \psi \\
\mathcal{M}, w \models K_i\varphi & \text{iff} & \text{for all } v \text{ such that } w{\sim_i}v \text{ we have } \mathcal{M}, v \models \varphi \\
\mathcal{M}, w \models B_i^{\psi}\varphi & \text{iff} & \text{for all } v \in \mathcal{K}_i[w] \text{ if } v \in \min_{\leq_i}(\mathcal{K}_i[w] \cap \|\psi\|) \text{ then } v \models \varphi
\end{array}$$

We define $\|\varphi\|$ such that $\|\varphi\| = \{w \in W \mid w \models \varphi\}$.

The last clause defines the semantics of the conditional belief operator. An agent is defined to believe φ in state w conditionally on ψ if φ is true in all states that are minimal in the part of the uncertainty range of the agent restricted to those states that make ψ true.

For axiomatizations of $\mathcal{L}_{\text{DOX}}$ the reader is advised to consult (Board, 2004) and (Baltag & Smets, 2008b).

Plausibility Upgrade

Epistemic plausibility models can accommodate public announcements of hard information. Performing update on those structures has an effect analogous to restriction of simple epistemic models. Such a change can of course result in belief change. However, plausibility ordering gives an opportunity to define different, more sophisticated operations on beliefs, operations that do not require state deletion. As we will see in Chapter 4, such revisions are useful if the source of information is not completely trustworthy.

Lexicographic Upgrade The *lexicographic upgrade* of an epistemic plausibility model $\mathcal{M} = (W, (\sim_i)_{i\in\mathcal{A}}, (\leq_i)_{i\in\mathcal{A}}, V)$ with a formula φ, rearranges the preorders by putting all states satisfying φ to be more plausible then others. Let us take $\leq_i^{\varphi} = \leq_i\restriction \|\varphi\|$, and $\leq_i^{\bar{\varphi}} = \leq_i\restriction \|\neg\varphi\|$.

Definition 2.2.13. *The* lexicographic upgrade *of an epistemic plausibility model $\mathcal{M} = (W, (\sim_i)_{i\in\mathcal{A}}, (\leq_i)_{i\in\mathcal{A}}, V)$ with a formula φ is defined as follows:*

$$\mathcal{M} \Uparrow \varphi := (W, (\sim_i)_{i\in\mathcal{A}}, (\leq_i')_{i\in\mathcal{A}}, V),$$

where for each $i \in \mathcal{A}$ and for all $v, w \in \mathcal{K}_i[w]$:

$$v \leq_i' w \text{ iff } (v \leq_i^{\varphi} w \text{ or } v \leq_i^{\bar{\varphi}} w \text{ or } (v \models \varphi \text{ and } w \models \neg\varphi)).$$

The language of announcements that trigger lexicographic upgrade is given in the following way.

Definition 2.2.14 (Syntax of $\mathcal{L}_{\Uparrow}$)**.** *The syntax of the doxastic-epistemic language $\mathcal{L}_{\Uparrow}$ is defined as follows:*

$$\begin{aligned} \varphi &:= \ p \mid \neg\varphi \mid \varphi \vee \varphi \mid K_i\varphi \mid B_i^{\psi}\varphi \mid [A]\varphi \\ A &:= {\Uparrow}\varphi \end{aligned}$$

where $p \in$ Prop, $i \in \mathcal{A}$.

Definition 2.2.15 (Semantics of $\mathcal{L}_{\Uparrow}$)**.** *For the doxastic-epistemic fragment $\mathcal{L}_{\mathrm{DOX}}$ the interpretation is given in Definition 2.2.12. The remaining clause of $\mathcal{L}_{\Uparrow}$ is as follows.*

$$\mathcal{M}, w \models [{\Uparrow}\varphi]\psi \quad \text{iff} \quad \mathcal{M} \Uparrow \varphi, w \models \psi$$

Conservative Upgrade The *conservative upgrade* (also known as *minimal upgrade* or *elite change*, see Van Benthem, 2007) of an epistemic plausibility model $\mathcal{M} = (W, (\sim_i)_{i\in\mathcal{A}}, (\leq_i)_{i\in\mathcal{A}}, V)$ with a formula φ, rearranges the preorders by making only the most plausible states satisfying φ more plausible than all others, leaving the rest of the preorder the same. Let $\leq_i^{\mathrm{rest}\varphi} = \leq_i \restriction \{t \in S \mid t \notin \min_{\leq_i} \|\varphi\|\}$.

Definition 2.2.16. *The* conservative upgrade *of an epistemic plausibility model $\mathcal{M} = (W, (\sim_i)_{i\in\mathcal{A}}, (\leq_i)_{i\in\mathcal{A}}, V)$ with a formula φ is defined as follows:*

$$\mathcal{M}{\uparrow}\varphi := (W, (\sim_i)_{i\in\mathcal{A}}, (\leq_i')_{i\in\mathcal{A}}, V),$$

where for each $i \in \mathcal{A}$ and for all $v, w \in \mathcal{K}_i[w]$:

$$v \leq_i' w \text{ iff } (v \leq_i^{\mathrm{rest}\varphi} w \text{ or } v \in \min_{\leq_i}\|\varphi\|).$$

Definition 2.2.17 (Syntax of $\mathcal{L}_{\uparrow}$)**.** *The syntax of the doxastic-epistemic language $\mathcal{L}_{\uparrow}$ is defined as follows:*

$$\begin{aligned} \varphi &:= \ p \mid \neg\varphi \mid \varphi \vee \varphi \mid K_i\varphi \mid B_i^{\psi}\varphi \mid [A]\varphi \\ A &:= {\uparrow}\varphi \end{aligned}$$

where $p \in$ Prop, $i \in \mathcal{A}$.

Definition 2.2.18 (Semantics of $\mathcal{L}_{\uparrow}$)**.** *For the doxastic-epistemic fragment $\mathcal{L}_{\mathrm{DOX}}$ the interpretation is given in Definition 2.2.12. The remaining clause of $\mathcal{L}_{\uparrow}$ is as follows.*

$$\mathcal{M}, w \models [{\uparrow}\varphi]\psi \quad \text{iff} \quad \mathcal{M}{\uparrow}\varphi, w \models \psi$$

Complete axiomatization for the logics of the two types of upgrades can be given by a complete axiomatic system for conditional belief complemented with reduction axioms. Van Benthem (2007) gives a detailed discussion on the subject, together with explicitly formulated axioms.

In Chapter 4 we will cover these upgrade methods again in a systematic way. We will compare their reliability in the context of single-agent belief-revision. In this, we will follow other attempts to analyze some classical belief-revision problems within the framework of dynamic epistemic and doxastic logic.

Chapter 3

Learning and Epistemic Change

In the present chapter we show how the paradigms of learning theory and dynamic epistemic logic can be linked. We will discuss the interface between learning theory and dynamic epistemic logic in the context of iterated information change and belief revision.

3.1 Identification as an Epistemic Process

In Chapter 2 we gave the prerequisites of formal learning theory with its central notion of identification. Assuming the reader's familiarity with those standard tools, we will now discuss the epistemology behind finite and limiting identification.

What are the epistemic components of identification in the limit? The entanglement of the notions of knowledge, certainty and belief in limiting learning is widely used in explanations of the paradigm. We quote Gold (1967) in his seminal paper *Language identification in the limit*:

> In the case of identifiability in the limit the learner does not necessarily *know*[1] when his guess is correct. He must go on processing the information forever because there is always the possibility that information will appear which will force him to change his guess.

With time the epistemic metaphor in identification in the limit became even more explicit, involving notions of certainty, justification, possible worlds, etc.:

> [...] Thus the Scientist is never *justified* in feeling *certain* that her last conjecture will be her last.
>
> On the other hand, [identifiability in the limit] does warrant a different kind of *confidence*, namely that systematic application of guessing rule will eventually lead to an accurate, last conjecture [...]. If we *know*

[1] The emphasis is mine.

> that the *actual world* is drawn from [a class identifiable in the limit], then we can be *certain* that our inquiry will ultimately succeed [...]. (Jain et al., 1999, pp. 11–12)

Later, even notions of introspection of knowledge, belief and reliability were introduced:

> This does not entail that [the learner] *knows he knows* the answer, since [...] [the learner] may lack any reason to *believe* that his hypotheses have begun to converge. Nonetheless, to the extent that the *reliability perspective on knowledge* can be sustained, our paradigms concern scientific discovery in the sense of *acquiring knowledge*. (Martin & Osherson, 1998, p. 13)

Finally, the epistemic dominance of limiting identification over certainty has been once summed up in the following way:

> True, there are good reasons for preferring the computable way of deriving *knowledge*. We *know* the results of computations, and only *think we know* the results of trial and error procedures [viz. limiting computation]. There are many reasons for *preferring knowing to thinking* (as Popper, 1966, observed). But that does not change the fact that sometimes *thinking* may be more appropriate. (Kugel, 1986, p. 155)

Our aim is to expose the epistemology that runs the limiting learning process from behind the scenes. Let us start by overviewing the components of identification and discussing their correspondence with the approach of epistemic logic as described in Chapter 2.

Class of hypotheses The procedure of learning starts with a class of hypotheses, a class of possible states of the world. It can be interpreted as the background knowledge of Scientist, his uncertainty range (see, e.g., Martin & Osherson, 1997). Scientist expects that one of the possibilities is true, and in the framework it is guaranteed that he is right—Nature indeed chooses one from the class fixed in the beginning. Among the consequences of such a treatment of background knowledge is that the actual world is always one of the options Scientist considers possible. Another implication is that learning is not simply verifying or falsifying a single hypothesis, although those two processes can be viewed as important components of identification (Gierasimczuk, 2009b). The fact of picking *one from a class* is an important factor in learnability analysis. It allows considering learnability as a property of classes of hypotheses determined by some external properties.

Different nature of data and conclusions The key word "learning" is often used in the context of belief revision and dynamic epistemic logic. There it takes the form of one-step "learning that φ", followed by a modification of the informational state of the agent—usually by various ways of simply accepting φ as it is. In other words, the agent "learned that φ" means that the agent "got to know that φ". In the setting of formal learning theory it requires more effort than that to be declared to have learned something. First of all, the incoming information is by default spread over more than one step. The inductive, step-by-step nature of this inference is essential; the incoming pieces of data are of a different nature than the actual 'thing' being learned. Typically, at each finite step the environment gives only partial information about a potentially infinite set. The relationship between data and hypothesis is like the one between sentences and grammars, natural numbers as such and Turing machines. Namely, if we know the hypothesis, we can infer what kind of possible data are going to appear, but in principle we will not be able to make a conclusive inference from data to hypotheses. Therefore, in learning theory we say that an agent "learned that a hypothesis holds" if he converged to this hypothesis on data that are consistent with the actual world.

Positive, true, and readable data There are three important assumptions that the incoming data can satisfy:

1. Truthfulness (soundness). Scientist receives only true data, no false information is included. This assumption leads to, e.g., the priority of incoming data over the current conjecture and background preferences of Scientist.

2. Positiveness. Scientist receives only elements of positive presentation (*text*) of the object being learned. Alternatively, together with positive also all negative information could be included (*informant*), e.g., for set learning the graph of the characteristic function of the set could be enumerated.

3. Readability. Scientist has a complete clarity about what information he receives. A further step would be to analyze the situation of uncertainty about the incoming information.

4. Completeness. The data that are consistent with the actual world are all eventually enumerated.

In formal learning theory it is usually assumed that the incoming information is readable and complete. The source of data is also taken to be truthful. Occasional errors are rarely taken into account, and in more applied disciplines are interpreted as noise (see, e.g., Grabowski, 1987). In contrast, the general epistemic framework allows erroneous information in form both of mistakes and intentional lies. In this respect the original learning theory conforms more to the assumptions of the philosophy of scientific inquiry (Nature never lies) than to, e.g., conversational situations (see, e.g., Grice, 1975). Another classic requirement put on data is that it is positive, i.e., data enumerates only elements of the language. This assumption is often challenged by involving negative information, data indicating which elements are *not* in the set. This setting boosts the power of learning immensely (see Gold, 1967). It should be noted here that data including both positive and negative samples gives remedy to errors. There is enough expressive power so that any information inconsistent with the actual world can be accounted for truthfully later on in the process.

Inductive, step-by-step process As briefly mentioned in the previous points, the process of restricting the hypothesis space to only those hypotheses that are consistent with the incoming data resembles update or public announcement (Baltag et al., 1998). Can learning in the limit of hypothesis h be viewed as the result of announcing the conjunction of data that lead to stabilization on h? First let us observe that the point of convergence to a correct hypothesis is unknown and in general uncomputable, which makes it also uncomputable to discover which finite sequence resulted in the success of the learning process. Even more importantly, finite sequences of data cannot be seen as a single announcement of a given hypothesis, because which hypothesis is in fact announced by the data heavily depends on the initial hypothesis space. For instance, let us consider two classes: $\mathcal{C}_1 = \{\{1\}, \{2\}\}$ and $\mathcal{C}_2 = \{\{1\}, \{1, 2\}\}$, and let h_1 be a hypothesis corresponding to the set $\{1\}$. In this case the single event of updating with 1 is equivalent to announcing h_1 in case $\mathcal{C}_1$ had been the initial set of hypotheses, but it does not announce h_1 when Scientist has to pick from $\mathcal{C}_2$, since the other hypothesis is still possible.

Infinite procedures The learning theory framework is defined for potentially infinite universes, but even for finite worlds the sequences of data are infinite. The reason for this is that we want to account for situations when Scientist does not know the finiteness or size of the entity he investigates. If the initial class of hypotheses is not drastically restrictive, Scientist can never know whether all the elements have already been enumerated. This leads to infinite procedures and conditions defined in the limit. Our epistemic setting should reflect these properties. It should allow talking about epistemic states as invariant from some point onwards, without specifying when this happens. Such an approach to learning is not unheard of in epistemic logic and belief-revision. There is an ongoing philosophical debate about iterated belief revision, iterated epistemic update, stability of knowledge, etc. (see, e.g., Stalnaker, 2009). As we will show they directly correspond to our limiting processes.

Non-introspective knowledge The success of limiting learning can be defined as reaching an epistemic state that can be called 'knowledge'. What kind of knowledge is it? On the surface it seems to be pretty close to the classical justified true belief (see, e.g., Chisholm, 1982), the definition ascribed to Plato. Indeed, eventually Scientist puts forward a hypothesis that is true, he believes that it is true, and moreover he has some reasons to choose it and those reasons can be viewed as a (often very limited) justification. However, from the perspective of the agent this 'knowledge', preceded by a sequence of belief changes, is strictly operational, the work is always in progress. There seems to be some issue with introspection here—Scientist is not able to point out the successful guess, he does not know whether he will not be forced to change his guess again in the light of future data (for the discussion of the introspection of knowledge in inductive inference see Hendricks, 2003). On the other hand it is more than just a true belief—it is immune to change under new true information.

Single agent As mentioned before, in learning theory the data are assumed to be complete and true. In our view, this is the reason why learners are pretty lonely in this paradigm. Although in principle, science as well as learning seem to be at least a two-player game that includes a teacher and a learner (a sender of the information and a receiver), for many algorithmic reasons the role of the former has been minimized. As a result we are concerned here only with the role of Scientist. Nature can be viewed as an objective, uninvolved source of data. In a sense this constitutes an assumption of fairness. Nature does not intend to help or disturb the process. As a result, learning theory is predominantly a one-agent business. A hint of multi-agency can be associated with team-learning, a framework suggested by Blum & Blum (1975), explicitly introduced by Smith (1982) and since then extensively studied (for an overview see Jain & Sharma, 1996). However, multi-agency understood in this way can be summed up as learners working on their own contributing to some common, bigger goal. The topic of communication and (non-)cooperativeness of the learners is marginal here. Dynamic epistemic and dynamic doxastic logics study these notions of multi-agency explicitly, and this is in fact their main focus (for the benefits of a multi-agent approach to epistemic issues see Van Benthem, 2006).

3.2 Learning via Updates and Upgrades

With the above discussion in mind we can now turn to the question of how learning-theoretic notions can be approached from the perspective of the epistemic framework.[2]

Let us fix $\mathcal{C} = \{S_1, S_2, \ldots\}$ to be a class of sets. It can be interpreted as the initial epistemic model, representing the background knowledge of Scientist

[2] Our considerations are of semantic nature and therefore differ from the computable framework of learning theory. E.g., one of the consequences is that we assume the property of consistency of learning, which in formal learning theory is optional.

together with his uncertainty about which world is the actual one. Let us take the initial epistemic model to be formally defined as

$$\mathcal{M} = (\mathcal{C}, \sim),$$

where $\mathcal{C}$ is the set of worlds and $\sim \subseteq \mathcal{C} \times \mathcal{C}$ is an uncertainty relation for Scientist. For now we do not require any particular preference of the scientist over $\mathcal{C}$—all possibilities are equally plausible. Hence, we can for now assume that $\sim$ is a universal equivalence relation over $\mathcal{C}$. The initial epistemic state of the Scientist is depicted in Figure 3.1. This model corresponds to the starting point of the scientific discovery process. In the beginning Scientist considers all of them possible. Scientist is *given* the class of hypotheses $\mathcal{C}$, i.e., he knows what the alternatives are.

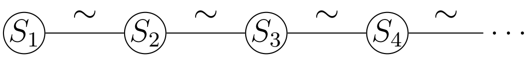


Figure 3.1: Initial epistemic model

Next, Nature decides on some state of the world by choosing one possibility from $\mathcal{C}$. Let us assume that, as a result, S_4 is the chosen world. Then, she decides on some particular environment ε, consistent with S_4. We picture this enumeration in Figure 3.2 below.

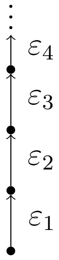


Figure 3.2: Environment ε consistent with S_4

The sequence ε is successively given to Scientist. Let us focus now on the first step of the procedure. A piece of data ε_1 is given to the scientist. In Figure 3.3 Scientist's confrontation with ε_1 is depicted. Scientist can react to this new information by adjusting his epistemic state in different ways.

3.2.1 Learning via Update

Epistemic Update

One way for Scientist to incorporate a new piece of data is to *update*[3] his status with ε_1. This is done by eliminating all the sets that do not include ε_1. We can represent the process formally by the update of $\mathcal{M}$ with ε_1, $(\mathcal{M} \mid \varepsilon_1)$, resulting in a new epistemic model $\mathcal{M}' = (\mathcal{C}', \sim')$, where: $\mathcal{C}' = \{S_n \in \mathcal{C} \mid \varepsilon_1 \in S_n\}$ and $\sim' = \sim\restriction \mathcal{C}'$.

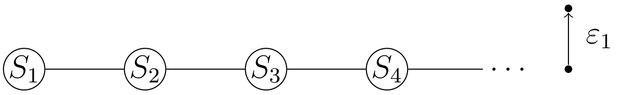


Figure 3.3: Confrontation with data

Scientist tests $\mathcal{C}$ with ε_1. If a set includes the information, it remains as a possibility, if it does not, it is eliminated (see Figure 3.4). Let us assume that ε_1 is not consistent with S_1 and S_3.

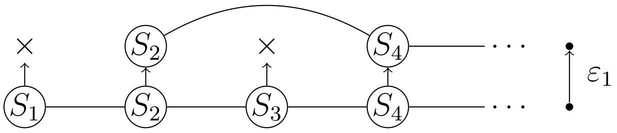


Figure 3.4: Epistemic update

This epistemic update can be iterated infinitely many times along ε resulting in an infinite sequence of models whose result according to the lines of DEL can be called the ε-Generated Epistemic Model (see, e.g., Van Benthem, Gerbrandy, Hoshi, & Pacuit, 2009).

Definition 3.2.1 (Generated epistemic model)**.** *The generated epistemic model* $\mathcal{M}^{\varepsilon}$*, with* $\varepsilon = \varepsilon_1, \varepsilon_2, \varepsilon_3, \ldots$*, is the result of update* $(((\mathcal{M} \mid \varepsilon_1) \mid \varepsilon_2) \mid \varepsilon_3) \mid \ldots$

To stay true to our original learning-theoretic motivation we want to investigate how the epistemic model changes when ε is given in a stepwise fashion. In particular, we would like to focus on its convergence properties. Our modeling involves only the equivalence relation, which mirrors not only the agent's uncertainty, but also indifference with respect to what is the actual world. This approach is especially and, we could argue, exclusively suited for interpreting the rise of irrevocable knowledge. That is, the agent is said to know something if this

[3] The event of update is a simple single-agent version of public announcement (Baltag et al., 1998).

something is true in all worlds in his uncertainty range defined by the equivalence relation. Therefore, we will be particularly interested in the convergence to the state of such knowledge, i.e., in our case in convergence to the situation in which only one, true set is left. Then we will say that the scientist learned with certainty what is the actual world. The possibility of reaching certainty in an epistemic model by the use of updates resembles the setting of finite identifiability. To recall the latter let us give a short example.

Example 3.2.2. *Let us take $\mathcal{C} = \{S_1, S_2, S_3\}$, such that $S_n = \{1, ..., n\}$, for $n \in \{1, 2, 3\}$. Nature makes her choice regarding the identity of the world. Let us assume that, as a result, S_3 is the actual world. Then, Nature chooses an enumeration $\varepsilon = 1, 2, 1, 3, 2, \ldots$. After the first piece of data, 1, the uncertainty range of the scientist includes the whole $\mathcal{C}$. After the second, 2, the scientist eliminates S_1 since it does not contain the event 2 and now he hesitates between S_2 and S_3. The third piece, 1, does not change anything; however, the next one, 3, eliminates S_2. Uncertainty is eliminated. He knows that S_3 is the actual world. Therefore, we can say that he learned it conclusively, with certainty.*

In Chapter 5 we will show that finite identifiability can be modeled within the dynamic epistemic logic framework, with the use of: possible worlds for sets; propositions for the incoming information; and update for the progress in eliminating uncertainty over the hypothesis space.

Plausibility Update

The epistemic, update-based approach as set out above is very restrictive with respect to the outcome of learning. At best, we have been able to account only for finite identification, and not for learning in the limit. In order to move to identification in the limit we need to be able to talk about *sequences of conjectures* of Scientist. Until now this was impossible because the only 'conjecture' that we were able to define was a final irrevocable conclusion. So we want to enrich the framework to account for a *current conjecture*—a hypothesis that is considered appropriate in a given step of the procedure.

Let us consider the following example of a learning scenario, in which the uncertainty is never eliminated.

Example 3.2.3. *In Example 3.2.2 Scientist was very lucky. Let us assume for a moment that nature had chosen $S_2 = \{1, 2\}$, and had fixed the enumeration $\varepsilon = 1, 2, 1, 2, 2, 2, 2, \ldots$ In this case Scientist's uncertainty can never be eliminated.*

This example indicates that the central element of the identification in the limit model is the unavoidable presence of uncertainty. The limiting framework allows, however, introducing some kind of *operational* knowledge (for an account of procedural knowledge see Hoshi, 2009), that is expressed by the stability of the conjectures of the learning function.

To model an algorithmic nature of the learning process that includes the actual guess and other not-yet-eliminated possibilities, we can enrich the epistemic model with some plausibility relation. The relation $\leq$ represents some preference over the set of hypotheses. E.g., if Scientist is an Occamist, the preference would be defined according to the simplicity of hypotheses. In the initial epistemic state the uncertainty of the scientist again ranges over the whole of $\mathcal{C}$. This time however the class is ordered and Scientist's current belief is the most preferred hypothesis.[4] Therefore, we consider the initial epistemic plausibility state of Scientist to be:

$$\mathcal{M} = (\mathcal{C}, \sim, \leq).$$

The procedure of erasing worlds that are inconsistent with successively incoming data is the same as in the previous section. This time however let us introduce the current-guess state which is interpreted as the actual conjecture of the Scientist. It is always the one that is most preferred—the smallest one according to $\leq$. In doxastic logic a set of most preferred hypotheses is almost invariably interpreted as the ones that the agent *believes* in. Let us go back to Example 3.2.3, where Nature chose world S_2. After seeing 2 and eliminating S_1, Scientist's attention focuses on S_2; then S_2 is his current belief. It is the most preferred hypothesis, and as such can be repeated as long as it is consistent with ε. In this particular case, since Nature chose a world consistent with S_2, it will never be contradicted, so Scientist will always be uncertain between S_2 and S_3. However, his preference directs him to believe in the correct hypothesis, without his being aware of the correctness. The belief in a hypothesis may become safe—whatever true information is given, it will not force the scientist to change his mind. And this state of safety while maintaining uncertainty is intuitively the one that occurs in identification in the limit. According to the picture sketched here, we will show (in Chapter 4) that learning in the limit can be modeled within the dynamic doxastic logic framework, using: possible worlds for sets; propositions for incoming information; update for the progress in eliminating uncertainty over the hypothesis space; a plausibility relation for the underlying hypothesis space; in each step of the procedure, the most preferred hypothesis as the actual positive guess of the learning function.

3.2.2 Learning via Plausibility Upgrades

Extending this approach we will also investigate different ways of reacting to the incoming information: except for update we will also consider ways of upgrading the preference relation as a reaction to new data. Upgrades are useful when update is too strong—in the situations in which the source of information is not entirely

[4] For now we do not pose any restriction on the plausibility ordering. The conditions of well-foundedness or connectedness of the plausibility ordering are often assumed of such doxastic situations. As we will see later, in our setting, the well-foundedness of the initial plausibility preorder might not always be possible.

reliable. We want to focus on two types of upgrades: lexicographic and minimal (see Chapter 2). Upgrades can be performed on the epistemic plausibility models step-by-step as in the case of iterated update. Interpreting the minimal hypotheses as the ones that the agents believes in at any finite point of the procedure, again allows considering sequences of conjectures.

3.3 Learning as a Temporal Process

In the above-described paradigm each hypothesis from the given class is associated with the corresponding set of environments. The latter can be seen as possible "streams of events" or "histories" that may occur if the relevant hypothesis is true. A history can in its turn be represented as a branch in the tree of all possible courses of events. Accordingly, hypotheses can be viewed as sets of histories or trees. The intuitive way to deal with hypotheses in a temporal framework is to introduce a temporal model of all the possible streams of information determined by the hypothesis.

Let us consider a set $\mathcal{C} = \{\{1\}, \{1,2\}, \{1,2,3\}\}$ and the corresponding set of hypotheses $I_{\mathcal{C}} = \{h_1, h_2, h_3\}$. We know that h_1 corresponds to the set $\{1\}$, so it is consistent with only one environment $\varepsilon = 1, 1, 1, 1, 1, \ldots$ Therefore, it can be identified with only one possible sequence of events, history H, which is represented by the frame presented in Figure 3.5.

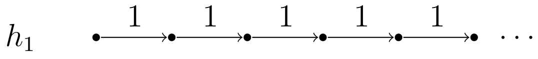


Figure 3.5: History for hypothesis h_1

It is of course different for the hypothesis h_2 which corresponds to the set $\{1,2\}$. Here, possible histories are all ω-sequences over the set $\{1,2\}$, that include at least one occurrence of 1 and 2. Therefore the hypothesis is represented as a binary tree.

Let us put this idea formally. If S is a set, then S^* is the set of all finite sequences over S (all finite strings of elements of S). Let us take a class $\mathcal{C}$ and $S_n \in \mathcal{C}$. The set S_n determines an epistemic temporal logic frame

$$\mathcal{F} = (S_n, H_n, \sim),$$

where $H_n = S_n^*$ is a protocol (says which sequences of events are allowed), that is closed under non-empty prefixes; and $\sim$ is a binary relation on H_n.

Such an epistemic temporal frame indicates which sequences of data can be expected when the corresponding hypothesis is true. This way of thinking allows viewing the class of hypotheses $\mathcal{C}$ as a set of protocols, a forest of temporal frames (see Figure 3.6).

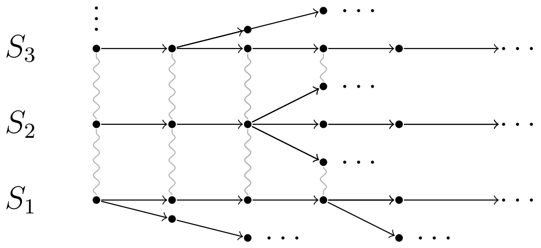


Figure 3.6: Epistemic temporal forest $\mathcal{F}$

To sum up, we interpret hypotheses to be sets of histories, i.e., sets of sequences enumerating events. Therefore, we can reinterpret the possible realities as *sets of functions*. This approach leads to a generalized, uniform view of learnability of various structures. Function learning and set learning become analyzable on a common ground.

To account for identification in the limit, following the argumentation of previous sections it seems to be necessary to enrich the temporal models with plausibility ordering that will account for the beliefs at each level of the temporal forest. The latter can be generated from the initial class as in the previous case. Then the temporal epistemic plausibility frame is given as follows:

$$\mathcal{F}_{\leq} = (S_n, H_n, \sim, \leq).$$

Our aim in all the above described semantic interpretations is to give an epistemic (temporal) characterization of learnability.

3.4 Summary

In this chapter we gave an introduction to our modeling of the process of inductive inference in dynamic epistemic logic and dynamic doxastic logic. For now we avoided formalism in order to first provide motivation and basics of the transition from one framework to another. In particular, we indicated that update is appropriate to analyze the notion of finite identifiability as convergence to knowledge. Learning in the limit, on the other hand, has to be supported by an underlying ordering of the hypothesis space. This indicates that it should be formalized in doxastic logic, where the preference or plausibility relation is a standard element of any model, and identification in the limit is viewed as reaching safe belief. We also proposed to view one component of the learning paradigm, hypotheses and hypothesis spaces, as temporal models. This allows investigating properties of the epistemic revision that requires certain *sequences* of events, conforming to some temporal protocols. We postulate that identifiability can be expressed in temporal logic interpreted over the corresponding epistemic temporal forests.

Part II

Learning and Definability

Chapter 4

Learning and Belief Revision

Learning can be described as a process of acquiring new information. This acquisition can take forms as different as things are that can be learned. We say: 'He learned that she cheated on him' or 'She learned about his disease', but also: 'She learned a language' or 'He finally learned how to behave'. The first two sentences are about a change in informational state induced by accepting a fact, getting to know something. The latter two are different, they describe a situation in which an inductive acquisition process came to a successful end.

The first kind of learning—getting to know about facts—is formalized and analyzed in the domain of belief revision and the diverse frameworks of epistemic and doxastic logics. The main aim here is to formalize the elementary dynamics of knowledge and epistemic attitudes towards incoming information.

The second kind—learning as a process—is studied within the framework of formal learning theory. In this framework a general concept (language, grammar, theory) gets to be identified by an agent on the basis of some elementary data (sentences, results of experiments) over a long period of time. The learning agent is allowed to change his mind on the way, and the process is successful if it results in convergence to an appropriate hypothesis. In a sense this kind of learning is built on top of the first kind, it consists of an iteration of simple getting-to-know events.

In this chapter we propose a way to use the framework of learning theory to evaluate belief-revision policies. Our interest is shared by at least two existing lines of research. Kelly, Schulte, & Hendricks (1995) and Kelly (1998a,b, 2004, 2008) focus on bringing together some classical belief-revision policies (among others those proposed by Boutilier, 1996; Darwiche & Pearl, 1997; Grove, 1988; Spohn, 1988) with the framework of function learning (see Chapter 2, Section 2.1.3, and for more details Blum & Blum, 1975). In this attempt the possible concepts to be learned or discovered are the possible sequential histories. The problem of prediction which seems to be at the heart of this approach is obviously useful for modeling certain kind of scientific inquiry. However in general, changes of epistemic states do not have to happen according to some prescribed sequence.

They are often governed by sequences of facts that are closed under permutation with respect to their informational content.

Martin & Osherson (1997, 1998) have also worked on establishing the connection between learning theory and belief revision. Their attempt has its roots in the classical AGM framework (Alchourrón et al., 1985) and treats belief revision as a two step process: the shrinking of the current belief state to accommodate the new information (belief contraction) and the incorporation of the data (see Levi, 1980). In this approach, the dominant features of modeling inductive learning as iterated belief revision are that the belief state is treated syntactically, as a set of sentences of a given language, and is assumed to be a full-blown theory (closed under the operation of consequence), incoming data get a fully trusted welcome, and last but not least, the agent does not explicitly consider other, perhaps counter-factual, possibilities.

Following Gierasimczuk (2009a,c) and Dégremont & Gierasimczuk (2009) we advance a different line of research. On the inductive inference side, we are interested in the paradigm of language learning which is more general than the aforementioned function learning approach. We assume that the data are observed in a random manner, so that in general predicting the future *sequence* is not feasible, or even relevant. As possible concepts that are inferred we take sets of atomic propositions. Therefore, receiving new data corresponds to getting to know about facts. On the side of belief revision we follow the lines of dynamic epistemic logic (see Van Benthem, 2007). Hence, we interpret current beliefs of the agent (hypothesis) as the content of those possible worlds that he considers most plausible. The revision does not only result in the change of the current hypothesis, but can also induce modification of the agent's plausibility order.

We are mainly concerned with *identifiability in the limit* (Gold, 1967). In the first part we restrict ourselves to learning from sound and complete streams of positive data. We show that learning methods based on belief revision via conditioning (update) and lexicographic revision are universal, i.e., provided certain prior conditions, those methods are as powerful as identification in the limit. Those prior conditions, the agent's prior dispositions for belief revision, play a crucial role here. We show that in some cases, these priors cannot be modeled using standard belief-revision models (as based on well-founded preorders), but only using generalized models (as simple preorders). Furthermore, we draw conclusions about the existence of tension between conservatism and learning power by showing that the very popular, most 'conservative' belief-revision methods, like Boutilier's minimal revision, fail to be universal. In the second part we turn to the case of learning from both positive and negative data. Here, along with information about facts the agent receives negative data about things that do not hold of the actual world. We again assume these streams to be truthful and we draw conclusions about iterated belief revision governed by such streams. This enriched framework allows us to consider the occurrence of erroneous information. Provided that errors occur finitely often and are always eventually corrected we show that the lexicographic revision method is still reliable, but more conservative methods fail. Before we get to the formal content of this chapter, let us first give two additional philosophical motivations for our work.

Evaluation of Belief-Revision Policies The traditional approach to the problem of belief revision (Alchourrón et al., 1985) can be summed up in the following way. The belief is taken to be a set of sentences, often assumed to be closed under some operation of consequence. Then, confronted with an incoming sentence the belief set has to undergo some transformation. If the sentence is consistent with the belief set, it is simply set-theoretically added, and the set is extended to include all consequences. If the sentence contradicts information contained in the set, the latter has to be modified by first removing inconsistency, and only then performing the addition. In general the belief-revision procedure takes the following form:

$$\langle \text{belief set, proposition} \rangle \rightarrow \text{revised belief set;}$$

in other words:

$$\langle S, \alpha \rangle \rightarrow S * \alpha.$$

Intuitively, belief revision, in order to be rational, has to conform to certain general rules. An intuitive set of rules of this kind, or axiomatization, if one prefers, of belief revision has been proposed by Alchourrón et al. (1985). Investigations into the properties of this type of revision led to the following difficulty: often there is more than one way to make a set consistent with some initially inconsistent input. For instance, if the belief set $S = \{\varphi, \varphi \rightarrow \psi\}$, and the incoming information is $\neg\psi$ then dropping either of the sentences in S would make the set S consistent with $\neg\psi$. How do we decide which one should be chosen? This problem indicates the need for some preference order that underlies beliefs and governs the order of potential elimination (Alchourrón et al., 1985). The postulate of ordering the beliefs according to their entrenchment has essentially enriched the framework. The results indicating the necessity of orders led to involving them explicitly as parts of belief states. A system that accounts for the ordering has been provided by Grove (1988), who represented AGM postulates in terms of systems of spheres. This modeling is a direct predecessor of the modal logic based approach to belief, as it presupposes a total well-founded preorder on the initial uncertainty range. The framework then conformed to a new scheme:

$$\langle (\text{belief set}, \leq),\ \text{proposition} \rangle \rightarrow (\text{revised belief set, revised } \leq);$$

in other words:

$$\langle (S, \leq), \alpha \rangle \rightarrow (S * \alpha, \leq^{\alpha}).$$

This approach led to many new questions, among others: the origin and justification of $\leq$; possible ways of transforming $\leq$ and the ways in which they are (or should be) chosen.

Although this approach has been shown to be quite powerful, it is also very controversial in the fact that it is very syntactic. Dependence on the specific language and closing off under a consequence relation present problems when comparing it to linguistic and cognitive reality. In the meantime an alternative, more semantic approach to belief change has been developed, in which modal logic turns out to be very useful (see Stalnaker, 2009). Despite those developments the old uneasiness remains. How are we supposed to judge and choose between different belief-revision policies? Perhaps, by introducing another level in which some new preference relation will order different policies and this preference ordering itself should become a matter of taste or character? Rott (2008) expects that this question will eventually lead to some sort of circularity in the domain of belief-revision theory. It can be argued that this problem could ultimately be solved only by psychological research in human cognitive tendencies. He poses the following two options as answers to this difficulty:

> One option is to insist on divide-and-conquer strategy: Researchers in belief revision should put their efforts into finding out which methods are best in which contexts. [...] [A] second option. This option assumes that there is some level in the belief state hierarchy up to which constraints of rationality reside, but above which, at all higher levels, we are just describing, in an idealized way, various ways that people happen to be. [...] In this perspective, there are no objective standards for rationality beyond the first level as characterized by AGM.

In the present chapter of this book we will challenge this position by showing that applying certain types of rules in certain contexts can be analyzed in terms of whether they can be relied upon in the 'quest for the truth' (the analysis of inductive inference in terms of reliability has been for the first time provided by Kelly, 1996). We will analyze certain belief-revision policies in terms of their dependability and show differences in their learning power. In our framework we can naturally treat the procedural aspect of iterated belief revision, address some intermediate stages of such iterations and relate them to the ultimate success of a belief-revision policy. Hence, belief-revision methods can get evaluated on the basis of their learning power. Finally, it can be argued that the above-mentioned 'different ways that people happen to be' can be traced as evolutionary equilibria that correspond to effectiveness and reliability of methods.

Safety and Stability of Beliefs The classical definition characterizes knowledge as true justified belief (see, e.g., Chisholm, 1982). In a modern setting this has been formalized as the state of certainty, because a decent justification of a theory should eliminate all other possibilities. Such definition is difficult to accept from a philosophical standpoint, and many arguments against it can be (and have been) formulated (see, e.g., Gettier, 1963). One of them is that in fact knowledge is a dynamic phenomenon and it rarely occurs in the form of irrevocable states of certainty. Alternatives oscillate around the concept of knowledge as *safe belief*. The strength of safety is in the guarantee it gives: the safe belief is not endangered by the occurrence of true data. If we restrict our considerations to truthful information, or at least assume that mistakes happen rarely, safety can be reformulated in terms of stability. In other words, knowledge emerges when stability is reached. The need for such a notion appeared in many different frameworks: from reaching an agreement in a conversational situation (see, e.g., Lehrer, 1965, 1990) to the considerations in the domain of philosophy of science (see, e.g., Hendricks, 2001).

In this work we account for and characterize the emergence of both: the restrictive kind of knowledge (certainty) and stable belief. We explicitly formulate the conditions under which certain belief states give raise to the emergence of such epistemic and doxastic states.

Finally, inspired by Nozick (1981), Rott (2004) puts forward that perhaps:

> [...] knowledge [should] be made of still sterner stuff—stuff that also survives (a modest amount of) misinformation.

In the following sections we will show that under the requirement of convergence to stable belief some policies are still reliable if a finite number of errors occur and they are all corrected later in the process.

4.1 Iterated Belief Revision

In our analysis of single agent information-update and belief revision we will redefine the framework of dynamic epistemic logic in order to simplify things. As we are here concerned with the single-agent case and moreover, we take the incoming information to be propositional, we will focus on the notion of *epistemic state*, i.e., *a set of possible worlds.*

Definition 4.1.1. *A* possible world *is a valuation over* Prop*, and it can be identified with a set $s \subseteq$* Prop*. We say that p is true in s (write $s \models p$) if and only if $p \in s$.*

The uncertainty range of an agent is represented as a set of worlds that the agent considers possible.

Definition 4.1.2. *An* epistemic state *is a set $S \subseteq \mathcal{P}($*Prop*$)$ of possible worlds. A pair (S, s), where S is an epistemic state and $s \in S$ is called a* pointed epistemic state*.*

With respect to the setting defined in Chapter 2, epistemic states of the agent i associated to the epistemic model $\mathcal{M} = (W, (\sim_i)_{i \in \mathcal{A}}, V)$ are given by equivalence classes in $W/\!\sim$, in other words, a pointed epistemic state of an agent i is a pair $(\mathcal{K}_i[w], w)$.

In accordance with the semantics of basic epistemic logic, we will interpret the knowledge operator in the usual way.

Definition 4.1.3 (Semantics of $\mathcal{L}_{\mathrm{EL}}$ in epistemic states)**.** *We interpret the single-agent $\mathcal{L}_{\mathrm{EL}}$ in the epistemic states in the following way.*

$$\begin{array}{lll}
S, s \models p & \text{iff} & p \in s \\
S, s \models \neg\varphi & \text{iff} & \text{it is not the case that } s \models \varphi \\
S, s \models \varphi \vee \psi & \text{iff} & s \models \varphi \text{ or } s \models \psi \\
S, s \models K\varphi & \textit{iff} & S \subseteq \|\varphi\|
\end{array}$$

Accordingly, our simplified approach will be extended to the doxastic framework. By enriching the epistemic state with a plausibility relation we consider epistemic plausibility states. To model beliefs, we need to specify some subset $S_0 \subseteq S$ of the epistemic state, consisting of the possible worlds that are consistent with the agent's beliefs. The intuition here is that although the agent considers all worlds in his epistemic state possible, some of them are seen as more 'desirable', those will be given as the minimal ones according to the plausibility order.

Definition 4.1.4. *A prior plausibility assignment $S \mapsto \leq_S$ assigns to any epistemic state some plausibility order based on the original epistemic state.*

Definition 4.1.5. *A* plausibility state *is a pair $(S, \leq)$ of an epistemic state S and a total preorder $\leq$ on S, called a plausibility relation.*

An epistemic state together with some prior plausibility assignment constitute a plausibility state. Here as in the case of plausibility models we will assume the plausibility relations to be arbitrary total preorders. We will sometimes essentially require their non-well-foundedness.

The language $\mathcal{L}_{\mathrm{DOX}}$ is interpreted on plausibility states in the same way as $\mathcal{L}_{\mathrm{EL}}$. The missing clause of belief is given in the following way:

$$S, \leq, s \models B\varphi \text{ iff } \exists w \leq s \; \forall u \leq w \; u \models \varphi.$$

In the case when $\leq$ is well-founded, the usual definition of 'belief as truth in all the most plausible worlds' holds, i.e., if $(S, \leq)$ is a plausibility state, then for all $s \in S$:

$$S, \leq, s \models B\varphi \;\text{ iff }\; \min_{\leq} S \subseteq \|\varphi\|.$$

Our aim now is to reconstruct iterated belief revision in a strict correspondence with identifiability in the limit. We will analyze the epistemic and doxastic properties of limiting learning and the influence of various epistemic attitudes on the process of convergence. Later we will compare the power of learning in the limit with the capabilities of various belief-revision policies. Before we get to them, we need to set some basic notions of incoming data. As mentioned before, we are interested in (possibly indefinite) iterations—to get the kind of full generality we need to consider infinite streams of information.

Our streams of data consist of chunks of information—every such chunk is a finite set of atomic propositions.

Definition 4.1.6. *A positive non-deterministic data stream is an infinite sequence* $\varepsilon = (\varepsilon_1, \varepsilon_2, \ldots)$ *of finite sets* ε_i *of propositions from* $\textsc{Prop}$*.*

The intuition is that, at stage i, the agent observes the data in ε_i. The data set $\emptyset$ corresponds to making no observation. For clarity, we will call finite parts of such data streams *data sequences*.

Definition 4.1.7. *A data sequence is a finite sequence* $\sigma = (\sigma_1, \ldots, \sigma_n)$*, where for every* $0 < i \leq n$*,* σ_i *is a finite subset of* $\textsc{Prop}$*.*

Besides the usual notation given for texts in Definition 2.1.2, we will also use the concatenation of data sequences.

Definition 4.1.8. *Let* σ *and* π *be data sequences. We write* $\sigma * \pi$ *to denote the concatenation of the two strings, i.e., if* $\sigma = (\sigma_1, \ldots, \sigma_n)$ *and* $\pi = (\pi_1, \ldots \pi_k)$*, then* $\sigma * \pi = (\sigma_1, \ldots, \sigma_n, \pi_1, \ldots \pi_k)$*. For simplicity, if* ρ *is a finite set of propositions, then* $\sigma * \rho = (\sigma_1, \ldots, \sigma_n, \rho)$*.*

As explained in Section 2.1, data streams are not entirely arbitrary, they should reflect reality, be consistent with the actual world. The analogy with scientific inquiry can be used here: one can base theories on the results of experiments if the results are assumed to be consistent with reality. This property of data streams will be called 'soundness'.

Definition 4.1.9. *A positive data stream* ε *is* sound *with respect to world* s *iff all data in* ε *are true in* s*, i.e.,* $\text{set}(\varepsilon) \subseteq s$*.*

Another restriction on data streams is that, since they are infinite, they should enumerate all elements that are true in the actual world. In other words, if we wait long enough we will see it all. This property of data streams will be called 'completeness'.

Definition 4.1.10. *A positive data stream* ε *is* complete *with respect to world* s *iff all the true atomic propositions in* s *are in* ε*, i.e., if* $s \subseteq \text{set}(\varepsilon)$*.*

Throughout the most of this chapter we will assume the data streams for some world s to be sound and complete with respect to s, i.e., we will assume that $s = \text{set}(\varepsilon)$.

In standard learning theory such positive, sound and complete data streams are called 'texts' (see Chapter 2). They are restricted only to the streams in which all the observed data ε_i are either singletons $\{p\}$ (consisting of a positive atom $p \in \textsc{Prop}$) or $\emptyset$ ('no observation'). The above definitions allow observing more than one atomic fact at a time. In our formalism each piece of information ranges over finite ε_i, and therefore the classical learning theory setting is equivalent to ours.

Until now learning methods have been described generally as ways of converting epistemic states into belief sets in a way dependent on the incoming information. In order to approach the subject of learning as an iterated belief-revision process, we will now turn to the more constructive part of our paradigm—the belief-revision methods themselves. The long-term aim that we have in mind is to define and investigate learning methods that are governed by belief-revision policies.

We define a belief-revision method as a function that, given some data sequence, transforms plausibility states.

Definition 4.1.11. *A* belief-revision method *is a function R that given any plausibility state $(S, \leq)$ and a data sequence $\sigma = (\sigma_1, \ldots, \sigma_n)$ (of any finite length n), outputs a new plausibility state*

$$R((S, \leq), \sigma) := (S^\sigma, \leq^\sigma).$$

Our notion of belief-revision method is more general than the one of classical belief-revision policies. The latter are memory-free, can account by default only for one step of revision. Hence, each time they take only one piece of incoming information. The above definition makes our methods dependent on a finite history of events, but obviously it accounts for the classical policies as a special case.

As in the case of learning methods there are some basic requirements that belief-revision methods might be expected to fulfill. This time most of the properties will be defined in terms of belief operator B, as given in Section 4.1. First we give two versions of data-retention, the property that states that beliefs are expected to reflect the incoming information.

Definition 4.1.12. *A belief-revision method is* weakly data-retentive *if after the revision the most recent piece of data is believed, i.e., for $\sigma = (\sigma_1, \ldots, \sigma_n)$, we have*

$$\text{if } p \in \sigma_n \text{ then } (S^\sigma, \leq^\sigma) \models Bp.$$

Definition 4.1.13. *A belief-revision method is* strongly data-retentive *if all the observed data are believed, i.e., if $\sigma = (\sigma_1, \ldots, \sigma_n)$ then for every $1 \leq i \leq n$:*

$$\text{if } p \in \sigma_i \text{ then } (S^\sigma, \leq^\sigma) \models Bp.$$

In the case of belief-revision methods we can define two types of conservatism. Unlike general learning methods, belief-revision methods output the whole revised plausibility state. So, conservatism can take a weak form in which the belief itself does not change if the new piece of data has already been believed, or a strong form in which the whole plausibility state does not change under new information, that has been already believed.

Definition 4.1.14. *A belief-revision method is* weakly conservative *if it keeps the same belief when it is confirmed by the new information, i.e., for every finite* $\rho \subseteq \textsc{Prop}$ *such that* $(S^\sigma, \leq^\sigma) \models B(\bigwedge \rho)$ *and for every formula* θ*, we have that:*

$$(S^\sigma, \leq^\sigma) \models B\theta \text{ iff } (S^{\sigma * \rho}, \leq^{\sigma * \rho}) \models B\theta.$$

Definition 4.1.15. *A belief-revision method is* strongly conservative *if it does not change the plausibility state when the new data has already been believed, i.e., for every finite* $\rho \subseteq \textsc{Prop}$ *s.t.* $(S^\sigma, \leq^\sigma) \models B(\bigwedge \rho)$*, we have*

$$(S^\sigma, \leq^\sigma) = (S^{\sigma * \rho}, \leq^{\sigma * \rho}).$$

We define the notion of data-drivenness as in the case of learning methods:

Definition 4.1.16. *A belief-revision method is data-driven if it is both weakly data-retentive and weakly conservative.*

As mentioned before, belief-revision methods work on whole plausibility states. This allows a refined notion of keeping track of past events. History-independent belief-revision methods do not distinguish between the same plausibility states that have different pasts.

Definition 4.1.17. *A belief-revision method is* history-independent *if its output at any stage depends only on the previous output and the most recently observed data, i.e., for every finite* $\rho \subseteq \textsc{Prop}$ *and all finite data sequences* σ, π*, we have*

$$\text{if } (S^\sigma, \leq^\sigma) = (S^\pi, \leq^\pi) \text{ then } (S^{\sigma * \rho}, \leq^{\sigma * \rho}) = (S^{\pi * \rho}, \leq^{\pi * \rho}).$$

4.2 Iterated DEL-AGM Belief Revision

All revision methods satisfying the AGM postulates are data-driven. This follows from the second AGM postulate: If S is a belief state and $S * \varphi$ represents the set of beliefs resulting from revising S with new belief φ, then φ belongs to $S * \varphi$ (Alchourrón et al., 1985). However, as we will see below, AGM methods are not necessarily strongly data-retentive, nor strongly conservative. Below we will consider three basic qualitative belief-revision methods that met considerable attention within dynamic epistemic logic research: conditioning (update), lexicographic revision (radical upgrade) and minimal revision (conservative upgrade) (for details see Chapter 2). We will investigate the properties of homogeneous iterated revision, i.e., sequences of revisions governed by one particular belief-revision policy.[1]

4.2.1 Conditioning

We want to focus now on the conditioning revision method, which corresponds to *update* in dynamic epistemic logic (see Van Benthem, 2007, and Chapter 2). To briefly recall the notion, update operates by deleting those worlds that do not satisfy all the new data. The minimal requirement for rational application of update is that the incoming information is truthful. We redefine the notion of update for our epistemic states in the following way.

Definition 4.2.1. Conditioning *is a belief-revision method* `Cond` *that takes an epistemic state S together with a prior plausibility assignment $\leq_S$, i.e., a plausibility state, and a finite set of propositions ρ and outputs a new plausibility state in the following way:*

$$\texttt{Cond}((S, \leq_S), \rho) = (S^\rho, \leq_S^\rho),$$

where $S^\rho = \{s \in S | s \models \bigwedge \rho\}$, and $\leq_S^\rho = \leq_S \upharpoonright S^\rho$.

The conditioning revision method is obviously weakly data-retentive. Moreover, one can say that it treats the incoming information very seriously—it deletes all worlds inconsistent with it. The deletion cannot be reversed—in this sense conditioning is the ultimate way to memorize things. Below we prove that conditioning is strongly data-retentive.

Proposition 4.2.2. *Conditioning revision method on $(S, \leq_S)$ is strongly data-retentive.*

Proof. Let us take $\sigma = (\sigma_1, \ldots, \sigma_n)$ and assume that $\texttt{Cond}((S, \leq), \sigma) = (S^\sigma, \leq^\sigma)$. We have to show that the conditioning revision method `Cond` is strongly data-retentive, i.e., if , for every $1 \leq i \leq n$:

$$\text{if } p \in \sigma_i \text{ then } (S^\sigma, \leq^\sigma) \models Bp.$$

Each time the new information σ_i comes in all worlds that do not satisfy it are eliminated, therefore $S^\sigma = \| \bigwedge \bigcup \sigma \|$. Hence for every world $s \in S^\sigma$, we have that $s \models \bigwedge \bigcup \sigma$. So in the resulting model every proposition that ever occurred in σ is believed. □

Conditioning, being an AGM revision method, is weakly conservative. We will show that it is not strongly conservative.

[1] An alternative, complementary view is to alternate belief-revision policies depending on the status of the incoming information. In such a case the level of doubt (or conservatism) with which one can accept the incoming data depends on the level of the reliability of the incoming information (see Baltag & Smets, 2008a; Van Benthem, 2007). We will not be concerned with such heterogeneous policies, but we view them as an interesting topic for future work.

Proposition 4.2.3. *Conditioning is not strongly conservative.*

Proof. Let us take a sequence of data σ and assume that $\texttt{Cond}((S, \leq), \sigma) = (S^\sigma, \leq^\sigma)$. We have to show that the conditioning revision method Cond is not strongly conservative, i.e., it is not necessarily the case that it keeps the same plausibility state when the new data is already believed. In other words for every finite $\rho \subseteq \textsc{Prop}$ such that $(S^\sigma, \leq^\sigma) \models B(\bigwedge \rho)$, we have

$$(S^\sigma, \leq^\sigma) = (S^{\sigma*\rho}, \leq^{\sigma*\rho}).$$

Let us consider the following example. Assume that $S^\sigma = \{\{p, q\}, \{p\}\}$, $\rho = \{q\}$, and the plausibility gives the following order: $\{p, q\} \leq^\sigma \{p\}$. Then clearly

$$(S^\sigma, \leq^\sigma) \models B(\bigwedge \rho).$$

However, after receiving ρ, the revision method Cond will eliminate world $\{p\}$ and therefore:

$$(S^\sigma, \leq^\sigma) \neq (S^{\sigma*\rho}, \leq^{\sigma*\rho}).$$

□

We have shown that apart from being data-driven (weakly data-retentive and weakly conservative) conditioning is strongly data-retentive but not strongly conservative.

Conditioning on Epistemic States Conditioning can change the underlying plausibility order only by deletion of possible worlds. If update is performed on epistemic states that lack plausibility structure, in some cases, while the range of uncertainty of the agent shrinks upon new data, the emergence of full certainty can occur. Conditioning can be considered successful if the actual guess is *finitely identified* (see Chapter 5 and Dégremont & Gierasimczuk, 2009). In this case the iteration on any data stream consistent with any world s allows eliminating uncertainty in a finite number of steps.

4.2.2 Lexicographic Revision

Lexicographic revision corresponds to radical upgrade in dynamic epistemic logic. When facing new information, it does not delete states, it just makes all the worlds satisfying the new piece of data more plausible than all the worlds that do not satisfy it and within the two parts, the old order is kept.

Definition 4.2.4. Lexicographic revision *is a belief-revision method* Lex *that takes an epistemic state S together with a prior plausibility assignment $\leq_S$, i.e., a plausibility state, and a finite set of propositions ρ and outputs a new plausibility state in the following way:*

$$\texttt{Lex}((S, \leq_S), \rho) = (S, \leq_S^\rho),$$

where for all $t, w \in S$:

$$t \leq_S^{\rho} w \textit{ iff } (t \leq_S^{\rho} w \textit{ or } t \leq_S^{\bar{\rho}} w \textit{ or } (t \in \| \bigwedge \rho \| \wedge w \in \| \neg \bigwedge \rho \|)),$$

where: $\leq_S^{\rho} = \leq_S \restriction \| \bigwedge \rho \|$, *and* $\leq_S^{\bar{\rho}} = \leq_S \restriction \| \neg \bigwedge \rho \|$.

Lexicographic revision is not strongly data-retentive on arbitrary sequences of data. However, if the data sequence is sound with respect to a world in the epistemic state, strong data retention holds. Moreover, this type of revision is not strongly conservative. Let us go through the arguments for each case.

Proposition 4.2.5. *Lexicographic revision is not strongly data-retentive on arbitrary data streams.*

Proof. Let us take a finite sequence of data $\sigma = (\sigma_1, \ldots, \sigma_n)$ and assume that $\mathtt{Lex}((S, \leq), \sigma) = (S, \leq^{\sigma})$. We have to show that the lexicographic revision method is not strongly data-retentive, i.e. it is not the case that for every $1 \leq i \leq n$:

$$\text{if } p \in \sigma_i \text{ then } (S, \leq^{\sigma}) \models Bp.$$

Let us take $S = \{\{p\}, \{q\}\}$, $\sigma = (\{p\}, \{q\})$, and assume any initial ordering on S, e.g., $\{p\} \leq \{q\}$. First $\sigma_1 = \{p\}$ comes in, and p starts to be believed. After receiving $\sigma_2 = \{q\}$ the most plausible state becomes $\{q\}$, so p is no longer believed, i.e., $\neg B \bigwedge \sigma_1$. □

Observe that σ in the above proof is not sound and complete with respect to any possible world in S. Therefore, in the learning-theoretic setting that we described in Chapter 2, σ cannot possibly appear in the first place. In this sense the lexicographic revision is especially well suited to learning and scientific inquiry—it's behaviour improves on data streams that are assumed to be consistent with reality. Let us see that it is so.

Proposition 4.2.6. *Lexicographic revision method on $(S, \leq_S)$ is strongly data-retentive on data sequences that are sound with respect to some $s \in S$.*

Proof. We have to show that the lexicographic revision method $\mathtt{Lex}$ is strongly data-retentive on sound data sequences. Let us take a plausibility state $(S, \leq_S)$, $s \in S$ and σ—a data sequence that is sound with respect to s, i.e., $\text{set}(\sigma) \subseteq s$. After reading σ, for all the worlds t that are most plausible with respect to $\leq_S$ in S it is the case that $\| \bigwedge \bigcup \sigma \| \subseteq t$, $t \models B \bigwedge \bigcup \sigma$. It is so because by assumption there is at least one such world, s. □

While adhering to the desired form of strong data-retention, lexicographic revision is not strongly conservative.

Proposition 4.2.7. *The lexicographic revision is not strongly conservative.*

Proof. Let us take a sequence of data σ and assume that $\texttt{Lex}((S,\leq),\sigma) = (S,\leq^{\sigma})$. We have to show that the lexicographic revision method `Lex` is not strongly conservative, i.e., it is not necessarily the case that it keeps the same plausibility state when the new data is already believed. Formally, for every finite $\rho \subseteq \textsc{Prop}$ such that $(S^{\sigma},\leq^{\sigma}) \models B(\bigwedge \rho)$, we have

$$(S,\leq^{\sigma}) = (S,\leq^{\sigma * \rho}).$$

Let us consider the following example. Assume that $S = \{\{p,q\},\{p\},\{q\}\}$, $\rho = \{q\}$, and the plausibility gives the following order: $\{p,q\} \leq^{\sigma} \{p\} \leq^{\sigma} \{q\}$. Then clearly $(S,\leq^{\sigma}) \models B(\bigwedge \rho)$. However, after getting ρ, the revision method will put world $\{q\}$ to be more plausible than $\{p\}$, and therefore

$$(S,\leq^{\sigma}) \neq (S,\leq^{\sigma * \rho}).$$

□

Summary Lexicographic revision is strongly data-retentive on streams that are sound with respect to some possible world. On the other hand, it is not strongly conservative.

4.2.3 Minimal Revision

The minimal revision method corresponds to conservative upgrade in dynamic epistemic logic (see Van Benthem, 2007). The most plausible worlds satisfying the new data become the most plausible overall. In the remaining part the old order is kept.

Definition 4.2.8. *Minimal revision is a belief-revision method* `Mini` *that takes an epistemic state S together with a prior plausibility assignment $\leq_S$ (i.e., a plausibility state) and a finite set of propositions ρ and outputs a new plausibility state in the following way:*

$$\texttt{Mini}((S,\leq_S),\rho) = (S,\leq_S^{\rho}),$$

where for all $t, w \in S$:

$$t \leq_S^{\rho} w \textit{ iff } t \leq_S^{\mathrm{rest}\rho} w \textit{ or } t \in \min_{\leq_S}(S,\leq_S)$$

where: $\leq_S^{\mathrm{rest}\rho} = \leq_S \upharpoonright \{t \in S \mid t \notin \min_{\leq_S} \| \bigwedge \rho \| \}$.

The minimal revision method is not strongly data-retentive (not even on sound data streams). But, on the other hand, it is strongly conservative.

Proposition 4.2.9. *Minimal revision on $(S, \leq_S)$ is not strongly data-retentive on all data sequences that are sound with respect to some $s \in S$.*

Proof. Let us take a sequence of data $\sigma = (\sigma_1, \ldots, \sigma_n)$ and assume that $\texttt{Mini}((S, \leq), \sigma) = (S, \leq^\sigma)$. We have to show that `Mini` is not strongly data-retentive, i.e., is not the case that for every $1 \leq i \leq n$:

$$\text{if } p \in \sigma_i \text{ then } (S, \leq^\sigma) \models Bp.$$

Let us take $S = \{\{p\}, \{q\}, \{p, q\}\}$, $\sigma = (\{p\}, \{q\})$ a data sequence consistent with world $\{p, q\}$, and assume that the initial ordering on S is $\{q\} \leq \{p\} \leq \{p, q\}$. After receiving $\sigma_1 = \{p\}$ the plausibility ordering becomes $\{p\} \leq^{\sigma_1} \{q\} \leq^{\sigma_1} \{p, q\}$. Then $\sigma_2 = \{q\}$ comes in—now our method gives the ordering $\{q\} \leq^{(\sigma_1, \sigma_2)} \{p\} \leq^{(\sigma_1, \sigma_2)} \{p, q\}$. So p is no longer believed although it was included in σ_1, i.e., after the second piece of data $\neg B(\bigwedge \sigma_1)$. □

Proposition 4.2.10. *Minimal revision is strongly conservative.*

Proof. Let us take a sequence of data σ and assume that $\texttt{Mini}((S, \leq), \sigma) = (S, \leq^\sigma)$. We have to show that the minimal revision method is strongly conservative, i.e., it keeps the same plausibility state when the new data is already believed. Formally, for every finite $\rho \subseteq \textsc{Prop}$ such that $(S, \leq^\sigma) \models B(\bigwedge \rho)$, we have

$$(S, \leq^\sigma) = (S, \leq^{\sigma * \rho}).$$

Let us take $\rho \subseteq \textsc{Prop}$ such that $(S, \leq^\sigma) \models B(\bigwedge \rho)$, we have to show that

$$(S, \leq^\sigma) = (S, \leq^{\sigma * \rho}).$$

Let us assume that $(S, \leq^\sigma) \neq (S, \leq^{\sigma * \rho})$. This means that after receiving ρ the plausibility order has been rearranged. By the definition of `Mini`, this could happen only in the case when among the most plausible in $(S, \leq^\sigma)$ there was no world t such that $t \in \|\rho\|$. But then also $(S, \leq^\sigma) \not\models B(\bigwedge \rho)$. Contradiction. □

The precise relation between the minimal revision method and the notion of conservatism is an interesting subject of further investigation. Our definition of strong conservatism indicates that minimal revision is the only strongly conservative belief-revision method. Hence, the concepts of minimal revision and strongly conservative revision are equivalent.

4.3 Learning Methods

In learning theory the learner is taken to be a function that on each finite sequence of data outputs a conjecture—a hypothesis from the initially given set of possibilities. We will follow this intuition here—our learning is performed by a function that, given some initial uncertainty range and a data sequence outputs a belief set. The latter contains worlds that are in some way consistent with the received sequence. In other words, a learning method is a way of converting any epistemic state, S, into a *belief set* $S_0 \subseteq S$ on the basis of any given data sequence.[2]

Definition 4.3.1. *A learning method is a function* L *that assigns some belief set* $L(S, (\sigma_1, \ldots, \sigma_n)) \subseteq S$ *to an epistemic state* S *(of any epistemic model) and a data sequence* $\sigma = (\sigma_1, \ldots, \sigma_n)$ *(of any finite length* n*).*

In contrast to our setting, in learning theory the learning functions are assumed to be deterministic, i.e., $L(S, \sigma_1, \ldots, \sigma_n) \subseteq S$ is either a singleton $\{h\}$ (for some $h \in S$, called the current hypothesis) or $\emptyset$ (when no conjecture is made).

In principle learning methods can be completely arbitrary. However, (to keep learners reasonable) we might well want them to satisfy certain requirements. Let us now define and briefly describe such basic properties of learning methods.

The first feature of learning methods is their data-retention. It formalizes the intuition that new beliefs should be consistent with the information received. Minimal requirement of this kind is that the belief set accounts for the most recent piece of data.

Definition 4.3.2. *A learning method is* weakly data-retentive *if for all data sequences* $\sigma = (\sigma_1, \ldots, \sigma_n)$*, we have that if* $L(S, \sigma) \neq \emptyset$ *then:*

$$\sigma_n \subseteq \bigcap L(S, \sigma).$$

The maximal requirement of data-retention is that the current conjecture always accounts for all data that have been encountered.

Definition 4.3.3. *A learning method is* strongly data-retentive *if for all data sequences* $\sigma = (\sigma_1, \ldots, \sigma_n)$ *and for every* $1 \leq i \leq n$*, we have that if* $L(S, \sigma) \neq \emptyset$*, then:*

$$\sigma_i \subseteq \bigcap L(S, \sigma).$$

In formal learning theory the corresponding property is known under the name of consistent learning.

Another quite intuitive assumption is that agents change their beliefs only when the observed information clearly contradicts their current beliefs. In other words, unless the agent is forced to, he does not change his mind.

[2] Learning functions output 'absolute beliefs'. In this respect they seem to be closer to the AGM-style belief revision than to DEL operations which account for 'conditional beliefs'. In fact, this analogy is not full, because learning functions are allowed to base their conjectures on whole sequences of events.

Definition 4.3.4. *A learning method is* weakly conservative *if for all data sequences $\sigma = (\sigma_1, \ldots, \sigma_n)$ and a finite $\rho \subseteq$ Prop, we have:*

$$\textit{if } \rho \subseteq \bigcap L(S,\sigma) \textit{ then } L(S,\sigma) = L(S, \sigma * \rho).$$

The analogous concepts: conservatism, in learning theory has been shown to restrict the class of languages identifiable in the limit, as has been consistency (see, e.g., Jain et al., 1999). We will not go into details of these arguments here. Let us just mention that our concept of learning method is different from that of the learning function in formal learning theory. These are assumed to output integers that are indices of sets in the initial class. Our learning methods are working directly on sets and output the entire set corresponding to current beliefs. In this respect our approach is more 'semantic' and accordingly learning methods may turn out to be more powerful.

For brevity's sake we will sometimes combine the two weak conditions of conservatism and data-retention together under the name of data-drivenness.

Definition 4.3.5. *A learning method is* data-driven *if it is both weakly data-retentive and weakly conservative.*

Last but not least, an important aspect of learning methods is their memory concerning past conjectures. Below we define the limit case of a memory-free learning method.

Definition 4.3.6. *A learning method is* memory-free *if, at each stage, the new belief set depends only on the previous belief set and the new data, i.e., for any finite $\rho \subseteq$ Prop:*

$$\text{if } L(S,\sigma) = L(S',\sigma') \text{ then } L(S,\sigma * \rho) = L(S', \sigma' * \rho).$$

A condition analogous to this in Definition 4.3.6 has been considered in formal learning theory and is known under the name of memory limitations (see, e.g., Jain et al., 1999).

4.4 Belief-Revision-Based Learning Methods

Finally, we are ready to put all the pieces together and describe learning that is based on belief-revision methods. We build a learning method from a belief-revision strategy in the following way. We take an epistemic state, put some plausibility order on it, and simulate a certain belief-revision method while receiving new information. The answer of the learning method each time consists of the most plausible worlds in the plausibility state. Such a learning method still outputs just the belief states but it bases its conjectures on the constructive work executed in the background by the belief-revision method.

Definition 4.4.1. *A belief-revision method R, together with a prior plausibility assignment $S \mapsto \leq_S$, generates a learning method L_R, called a* belief-revision-based learning method*, and given by:*

$$L_R(S, \sigma) := \min_{\leq_S} R(S, \leq_S, \sigma),$$

where $\min_{\leq'}(S', \leq')$ is the set of all the least elements of S' with respect to $\leq'$ (if such least elements exist) or $\emptyset$ (otherwise).

Now, an interesting set of questions arises. Is it the case that data-retention and conservatism of belief-revision method is inherited by the corresponding belief-revision-based learning methods? Do history-independent belief-revision methods generate memory-free learning methods? Below we list and discuss several dependencies between belief-revision methods and learning methods generated from them.

Proposition 4.4.2. *If a belief-revision method R is weakly data-retentive then the generated learning method L_R is weakly data-retentive.*

Proof. Let us take a belief-revision method R and some epistemic state together with a prior plausibility assignment $(S, \leq_S)$. Assume that R is weakly data-retentive, i.e., if $\sigma = (\sigma_1, \ldots, \sigma_n)$ is a data sequence then:

$$\forall p \in \sigma_n \; (S^\sigma, \leq_S^\sigma) \models Bp.$$

We need to show that if $L_R(S, \sigma) \neq \emptyset$, then $\sigma_n \subseteq \bigcap L_R(S, \sigma)$. Let us then assume that $L_R(S, \sigma) \neq \emptyset$, i.e., there is a $\leq_S^\sigma$ minimal element in S^σ. Then in every world minimal with respect to $\leq_S$ every p from σ_n holds:

$$\forall p \in \sigma_n \; \min_{\leq_S^\sigma}(S^\sigma, \leq_S^\sigma) \subseteq \|p\|,$$

where $\|p\|$ stands for the set of possible worlds that include p. Therefore, in every minimal world the conjunction of the σ_n holds:

$$\min_{\leq_S^\sigma}(S^\sigma, \leq_S^\sigma) \subseteq \| \bigwedge \sigma_n \|,$$

or equivalently:

$$\sigma_n \subseteq \bigcap \min_{\leq_S^\sigma}(S^\sigma, \leq_S^\sigma).$$

Since $(S^\sigma, \leq_S^\sigma) = R((S, \leq_S), \sigma) = L_R(S, \sigma)$, we have that

$$\sigma_n \subseteq \bigcap L_R(S, \sigma).$$

□

The next two propositions are proved in a similar way.

Proposition 4.4.3. *If R is data-retentive then the induced learning method L_R is data-retentive.*

Proposition 4.4.4. *If a belief-revision method R is weakly conservative then the induced learning method L_R is weakly conservative.*

It remains to show how belief-revision and learning methods relate to each other with respect to their memory limitations.

Proposition 4.4.5. *A learning method generated from a history-independent belief-revision method does not have to be memory-free.*

Proof. We prove this proposition by showing an example—a belief-revision method that is history-independent but the learning method that it induces is not memory-free. Let R be the lexicographic revision method (that corresponds to *lexicographic upgrade* in DEL, see Chapter 2), all the worlds satisfying the new data become more plausible than all the worlds not satisfying them; and within the two zones, the old order is kept. R is clearly history-independent. Each time the revision takes into account only the last output in the form of an epistemic plausibility state and the new incoming information. To see that L_R is not memory-free consider the following two plausibility orders on $S = S' = \{\{p\}, \{q\}, \{p, q\}\}$. Assume that for some σ and σ':

1. $R((S, \leq_S), \sigma)$ gives the plausibility order: $\{p\} <_S \{p, q\} <_S \{q\}$;

2. $R((S', \leq'_S), \sigma')$ gives the plausibility order: $\{p\} <_{S'} \{q\} <_{S'} \{p, q\}$.

It is easy to observe that $L_R(S, \sigma) = L_R(S', \sigma')$. Assume now that the next observation $\rho = \{q\}$. Then clearly $L_R(S, \sigma * \rho) = \{p, q\}$, while $L_R(S', \sigma' * \rho) = \{q\}$. Therefore, for the belief-revision method R there is a data sequence ρ such that:

$$L_R(S, \sigma) = L_R(S', \sigma'), \text{ but } L_R(S, \sigma * \rho) \neq L_R(S', \sigma' * \rho).$$

□

Summary Let us briefly summarize the results we have obtained so far. Data-retention and weak conservatism are preserved when a learning method is generated from a belief-revision method. However history-free belief-revision methods are still able to remember more than just the last conjecture of the generated learning method. This is so, because they 'keep' the whole plausibility order for 'further use'.

4.5 Convergence

What does it mean for a learning method to be *reliable* with respect to the initial epistemic state S? It means that it is possible to rely upon it to find the real world in finite time, no matter what the real world is, as long as it belongs to the given initial epistemic state S and as long as the data stream is sound and complete (for a discussion of reliability in belief-revision see Kelly et al., 1995). In this section we investigate reliability with respect to convergence to the correct belief. The expected result is not knowledge understood as full certainty, but rather a kind of belief that is guaranteed to persist under true information. In this setting, an agent can be right in believing something but he might not know it.

Identification in the limit guarantees the *convergence* to the right hypothesis, i.e., at a finite stage the answers of the learning method *stabilize* on the correct conjecture.[3]

Definition 4.5.1. *An epistemic state S is* identified in the limit *on positive data by learning method L if and only if for every world $s \in S$ and every sound and complete positive data stream for s, there exists a finite stage after which L outputs the singleton $\{s\}$ from then on.*[4]

In general we can attribute identifiability to the epistemic states by requiring that there is a learning method that identifies the state.

Definition 4.5.2. *An epistemic state is* identifiable in the limit (resp. finitely identifiable) *on positive data if there exists a learning method that can identify it in the limit (resp. finitely identify it) on positive data.*

Particular learning methods differ in their power. The most powerful among them are those that are universal, i.e., they can identify in the limit every class identifiable in the limit.

Definition 4.5.3. *A learning method L is* universal *on positive data if and only if it can identify in the limit on positive data every epistemic state that is identifiable in the limit.*

We are especially interested in learning methods that are generated from belief-revision policies. For brevity's sake we will use the notion of identification in the limit while talking about belief-revision policies. By a belief-revision method identifying S in the limit, we mean that the belief-revision method together with some prior plausibility assignment generates a learning method that identifies S in the limit (as given in Definition 4.5.1).

[3] In this chapter we will focus on identification in the limit. Finite identification is investigated in the context of epistemic logic in Chapter 5.

[4] In terms of belief, it means that the agent's conjectures stabilize to the complete true belief about the actual world.

Definition 4.5.4. *An epistemic state S is identified in the limit on positive data by a belief-revision method R if there exists a prior plausibility assignment $S \mapsto \leq_S$ such that the generated belief-revision-based learning method L_R identifies S in the limit on positive data.*

The above definition requires the existence of an appropriate initial plausibility assignment. In principle it can be a completely arbitrary preorder. However, we might want this prior plausibility assignment to satisfy certain assumptions of cognitive realism or rationality. The properties that are often required of such priors are *well-foundedness* and *totality*. Well-foundedness assures that a minimal state is always exists and it is possible to point to it as to the current belief. Totality guarantees that whenever two possibilities are considered, they are comparable with respect to the plausibility assignment. With respect to identifiability in the limit one can accordingly demand a prior plausibility assignment to satisfy those as standard assumptions of preference relations in doxastic epistemic logic (see, e.g., Dégremont, 2010).

Definition 4.5.5. *An epistemic state S is* standardly identified in the limit *on positive data by a belief-revision method R if there exists a (total) well-founded prior plausibility assignment $S \mapsto \leq_S$ such that the induced belief-revision-based learning method L identifies S in the limit on positive data.*

We define the analogous notion of universality for standard identifiability.

Definition 4.5.6. *A revision method is* standardly universal *on positive data if it can standardly identify in the limit on positive data every epistemic state that is identifiable.*

Our aim now is to show that some of the DEL-AGM revision methods generate a universal learning methods. The main technical difficulty of this part is the construction of the appropriate prior plausibility order. To define it we will use the concepts of locking sequences introduced by Blum & Blum (1975) and finite tell-tale sets proposed by Angluin (1980). For the latter we will use the simple non-computable version. We will refine the classical notion of finite tell-tales and use it in the construction of the suitable prior plausibility assignment that, together with conditioning and lexicographic revision, will generate universal learning method.

The first observation is that if convergence occurs, then there is a finite sequence of data that 'locks' the corresponding sequence of conjectures on a correct answer. This finite sequence is called a 'locking sequence'.

Definition 4.5.7 (Blum & Blum 1975)**.** *Let an epistemic state S, a possible world $s \in S$, a learning method L and a finite data sequence of propositions, σ, be given. The sequence σ is called a* locking sequence *for s and L if:*

1. $\text{set}(\sigma) \subseteq s$;

2. $L(S, \sigma) = \{s\}$*;*

3. for any data sequence α*, if* $\mathrm{set}(\alpha) \subseteq s$*, then* $L(S, \sigma) = L(S, \sigma * \alpha)$*.*

Lemma 4.5.8 (Blum & Blum 1975)**.** *If a learning method* L *identifies possible world* s *in the limit then there exists a locking sequence for* L *on* s*.*

The characterization of identifiability in the limit (see Theorem 2.1.14) can be generalized to account for arbitrary classes, by dropping the assumption of computability. It requires the existence of finite sets that allow drawing a conclusion without the risk of overgeneralization.

Lemma 4.5.9 (Angluin 1980)**.** *Let* S *be an epistemic state over a set* $\textsc{Prop}$ *of atomic sentences, such that* $\textsc{Prop}$ *and* S *are at most countable. If* S *is identifiable in the limit on positive data, then there exists a total map* $D : S \to \mathcal{P}^{<\omega}(\textsc{Prop})$*, given by* $s \mapsto D_s$*, such that* D_s *is a finite tell-tale for* s*, i.e.,*

1. D_s *is finite,*

2. $D_s \subseteq s$*,*

3. if $D_s \subseteq t \subseteq s$ *then* $t = s$*.*

Proof. Let S be an epistemic state over a set $\textsc{Prop}$ of atomic sentences, such that $\textsc{Prop}$ and S are at most countable. Let us also assume that S is identifiable in the limit on positive data by the learning method L, i.e., for every world $s \in S$ and every sound and complete positive data stream for s, there exists a finite stage after which L outputs the singleton $\{s\}$ from then on. By Lemma 4.5.8, for every $s \in S$ we can take a locking sequence σ_s for L on s. For any $s \in S$ we define $D_s := \mathrm{set}(\sigma_s)$.

1. D_s is finite because locking sequences are finite.

2. $D_s \subseteq s$, because $\mathrm{set}(\sigma_s) \subseteq S$.

3. if $D_s \subseteq t \subseteq s$ then $t = s$. Assume that there are $s, t \in S$, such that $s \neq t$ and $D_s \subseteq t \subseteq s$. Let us take a positive sound and complete data stream ε for t, such that for some $n \in \mathbb{N}$, $\varepsilon \restriction n = \sigma_s$. Because σ_s is a locking sequence for L on s and $\mathrm{set}(\varepsilon) = t \subseteq s$, L converges to s on ε. Therefore, L does not identify t, a state from S. Contradiction.

This concludes the proof. □

We will use the notion of finite 'tell-tale' to construct an ordering of S. The aim is to find a way of assigning the prior plausibility order that allows reliable belief revision. We will base the construction on finite tell-tales, but we will introduce one additional condition (see point 4 of Definition 4.5.10, below).

Definition 4.5.10. *Let S be a countable epistemic state with an injective map $i : S \to \mathbb{N}$, and D' be a total map such that $D' : S \to \mathcal{P}^{<\omega}(\textsc{Prop})$, given by $s \mapsto D'_s$ having the following properties:*

1. *D'_s is finite,*
2. *$D'_s \subseteq s$,*
3. *if $D'_s \subseteq t \subseteq s$ then $t = s$,*
4. *if $D'_s \subseteq t$ but $s \not\subseteq t$ then $i(s) < i(t)$.*

We call D' an ordering tell-tale map, *and D'_s an* ordering tell-tale set *of s.*

Definition 4.5.11. *For $s, t \in S$, we put*

$$s \preceq_{D'} t \text{ if and only if } D'_s \subseteq t.$$

We take $\leq_{D'}$ to be the transitive closure of the relation $\preceq_{D'}$.

Lemma 4.5.12. *For any identifiable epistemic state S and any ordering tell-tale map D', the relation $\leq_{D'}$ is an order, i.e., $\leq_{D'}$ is reflexive, transitive and antisymmetric.*[5]

Before we give the proof let us introduce the notion of a proper cycle in $\leq_{D'}$.

Definition 4.5.13. *A* proper cycle *in $\leq_{D'}$ is a sequence of worlds $s_1, \ldots, s_n$, with $n \geq 2$, and such that:*

1. *D'_{s_i} is included in s_{i+1} (for all $i = 1, \ldots, n-1$).*
2. *$s_1 = s_n$, but*
3. *$s_1 \neq s_2$.*

Proof. The fact that $\leq_{D'}$ is a preorder is trivial: reflexivity follows from the fact that D'_s is always included in s, and transitivity is imposed by construction (by taking the transitive closure). We need to prove that $\leq_{D'}$ is antisymmetric. In order to do that we will show (by induction on n) that $\leq_{D'}$ does not contain proper cycles of any length $n \geq 2$.

1. For the initial step ($n = 2$): Suppose we have a proper cycle of length 2. As we saw, this means that there exist two states s_1, s_2 such that $s_1 \neq s_2$, D'_{s_1} is included in s_2 and D'_{s_2} is included in s_1. There are three cases:

[5] We use D', to distinguish from the original tell-tale function D (satisfying only the conditions of Angluin's Theorem).

Case 1: s_1 is included in s_2. In this case, D'_{s_2} is included in s_1, and s_1 is included in s_2, so (by Condition 3 of Definition 4.5.10), we have that $s_1 = s_2$. Contradiction.

Case 2: s_2 is included in s_1. This case is similar: D'_{s_1} is included in s_2 and s_2 is included in s_1, so (by Condition 3 of Definition 4.5.10), we have that $s_2 = s_1$. Contradiction.

Case 3: s_1 is not included in s_2, and s_2 is not included in s_1. In this case, from the assumption that D'_{s_1} is included in s_2, and that s_1 is not included in s_2, we can infer (by Condition 4 of Definition 4.5.10), that $i(s_1) < i(s_2)$. But, in a completely similar manner (from D'_{s_2} included in s_1, and s_2 not included in s_1), we can also infer that $i(s_2) < i(s_1)$. Putting these together, we get $i(s_1) < i(s_2) < i(s_1)$. Contradiction.

2. For the inductive step $(n + 1)$: Suppose $s_1, s_2, ..., s_{n+1}$ is a proper cycle of length $n + 1$. We consider two cases:

Case 1: There exists k with $1 \leq k < n$ such that s_k is included in s_{k+1}. If $1 < k$, then it is easy to see that the sequence $s_1, \ldots, s_{k-1}, s_{k+1}, \ldots$ (obtained by deleting s_k from the above proper cycle of length $n + 1$) is also a proper cycle, but of smaller length (n). Contradiction. Similarly, if $k = 1$, it is easy to see that the sequence $s_1, s_3, ..., s_{n+1}$ (obtained by deleting s_2) is a proper cycle of smaller length (n). Contradiction.

Case 2: s_k is not included in s_{k+1} for any $1 \leq k < n$. In this case, we have that for all $1 \leq k < n$, D'_{s_k} is included in s_{k+1} but s_k is not included in s_{k+1}. By Condition 4 of Definition 4.5.10, it follows that we have $i_{s_k} < i_{s_{k+1}}$, for all $k = 1, \ldots, n$. By the transitivity of $\leq_{D'}$, it follows that $i_{s_1} < i_{s_{n+1}}$. But by Condition 2 of Definition 4.5.10, $s_1 = s_{n+1}$, hence $i_{s_1} > i_{s_{n+1}}$. Contradiction.

□

We will now show that $\leq_{D'}$, used by the conditioning revision method, guarantees convergence to the right belief whenever the underlaying epistemic state is identifiable in the limit.

Theorem 4.5.14. *The conditioning-based learning method is universal on positive data.*

Proof. We have to show that an epistemic model S is identifiable in the limit iff S is identifiable in the limit by the conditioning-based learning method. Obviously, if S is identifiable in the limit by the conditioning-based learning method, then S is identifiable in the limit. We will therefore focus on the other direction, i.e., we will show that if S is identifiable in the limit by any learning method, then it is identifiable in the limit by the conditioning-based learning method.

First let us assume that S, an epistemic state, is identifiable in the limit and hence it is at most countable. Let us then take an injective map $i : S \to \mathbb{N}$. By Lemma 4.5.9 we can assume the map D that gives tell-tales for any $s \in S$. On the basis of D we will now construct a new map $D' : S \to \mathcal{P}^{\leq\omega}(\textsc{Prop})$. We will proceed step by step according to the enumeration of S given by i.

1. For s_1 we set $D'(s_1) := D(s_1)$.
2. For s_n: For every $k < n$ such that $D_{s_n} \subseteq s_k$ and $s_n \not\subseteq s_k$, we choose an atomic proposition p_k such that $p_k \in s_n$ and $p_k \notin s_k$. We define *Rest* in the following way.

   $$\mathit{Rest} := \{p_k \mid k < n \ \& \ p_k \in s_n \ \& \ p_k \notin s_k \ \& \ D_{s_n} \subseteq s_k \ \& \ s_n \not\subseteq s_k\}.$$

   Then, we set $D'_{s_n} = D_{s_n} \cup \mathit{Rest}$.

We have to check if D' satisfies conditions of Definition 4.5.10.

1. D'_s is finite, because D_s and *Rest* are both finite.
2. $D'_s \subseteq s$, because D_s and *Rest* are subsets of s.
3. If $D'_s \subseteq t \subseteq s$ then $t = s$, because then $D_s \subseteq D'_s \subseteq t \subseteq s$, and hence, by the definition the finite tell-tale set $t = s$.

What remains is to check the condition 4: If $D'(s) \subseteq t$ and $s \not\subseteq t$ then $i(s) < i(t)$. Let us assume the contrary: $D'(s) \subseteq t$ and $s \not\subseteq t$ and $i(t) \leq i(s)$. There are two possibilities:

1. $i(t) = i(s)$, but then $s = t$ and hence $s \subseteq t$. Contradiction.
2. $i(t) < i(s)$. Then, there is a proposition $p \in D'(s)$ such that $p \in s$ and $p \notin t$. Therefore, by the inductive step of the construction of D', $D'(s) \not\subseteq t$. Contradiction.

We now have that D' satisfies all conditions of Definition 4.5.10, and therefore it is an ordering tell-tale map. Hence, by Lemma 4.5.12, the corresponding $\leq_{D'}$ is an order.

It remains to show that S is identifiable in the limit by the learning method generated from the conditioning belief-revision method and the prior plausibility assignment $\leq_{D'}$. Let us then take any $s \in S$ and the corresponding $D'(s)$. Since $D'(s) \subseteq s$ for every ε—a sound and complete positive data stream for s, there is $n \in \mathbb{N}$ for which $D'(s) \subseteq \mathrm{set}(\varepsilon\restriction n)$. Our aim is now to demonstrate that after receiving the elements of $\varepsilon\restriction n$, s is the minimal element in $S^{\varepsilon\restriction n}$ with respect to $\leq_{D'}$. Let us assume for contradiction that it is not, i.e., there is $t \in S^{\varepsilon\restriction n}$ such that $t \neq s$ and $t \leq_{D'} s$. Since $t \in S^{\varepsilon\restriction n}$, we get that $D'(s) \subseteq t$, but then, by Definition

4.5.10, $s \leq_{D'} t$. And because by Lemma 4.5.12, $\leq_{D'}$ is antisymmetric we get that $s = t$. Contradiction.

To see that the conditioning process stabilizes on $\{s\}$, it is enough to observe that ε is sound with respect to s, and therefore no further information from ε can eliminate s (because conditioning is weakly conservative). So for any $k > n$, $\min_{\leq_{D'}} \mathtt{Cond}((S, \leq'_D), \varepsilon \restriction k) = \{s\}$. □

Theorem 4.5.15. *The lexicographic belief-revision method is universal on positive data streams.*

The proof is analogous to the proof of Theorem 4.5.14. As far as simple beliefs are concerned, radical upgrades with true information do exactly what updates do. The only difference is that the rest of the doxastic structure might not stabilize, but only the minimal elements stabilize (on worlds indistinguishable from the real one).

The preorder defined in the proof of Theorem 4.5.14 is not necessarily well-founded. It is impossible to improve on this without losing the universality property. This is why as background setting we need generalized plausibility models, in which the plausibility is a preorder, without assuming well-foundedness. Belief can still be defined as 'truth in all the states that are plausible enough' (this requires three quantifiers: For every state s there exists some state $t \leq s$ such that φ is true in all states $w \leq t$). This is equivalent to the standard definition in the case that there exist minimal states (i.e., states $\leq$ than all others).

Let us now turn to the negative result concerning the minimal revision method.

Proposition 4.5.16. *Minimal revision is not universal.*

Proof. Let us give a counter-example, an epistemic state that is identifiable in the limit, but is not identifiable by the minimal revision method:

$$S = \{\{p\}, \{q\}, \{p, q\}\}.$$

The epistemic state S is identifiable in the limit by the conditioning revision method: just assume the ordering $\{p\} < \{q\} < \{p, q\}$. However, there is no ordering that will allow identification in the limit of S by the minimal revision method. If $\{p, q\}$ occurs in the ordering before $\{p\}$ (or before $\{q\}$), then the minimal revision method fails to identify $\{p\}$ ($\{q\}$, respectively). If both $\{p\}$ and $\{q\}$ precede $\{p, q\}$ in the ordering then the minimal revision method fails to identify $\{p, q\}$ on any data stream consisting of singletons of propositions from $\{p, q\}$. On all such data streams for $\{p, q\}$ the minimal state will alternate between $\{p\}$ and $\{q\}$, or stabilize on one of them. The last case is that at least one of $\{p\}$ and $\{q\}$ is equiplausibile to $\{p, q\}$. In such case $\{p, q\}$ is not identifiable because for any single proposition from $\{p, q\}$ there is more than one possible world consistent with it. □

Proposition 4.5.17. *There is no standardly universal belief-revision method.*

Proof. There is an epistemic state S that is identifiable in the limit by a learning method, but is not standardly identified in the limit by any belief-revision method, i.e., there is no belief-revision method that would, together with a well-founded order $\leq$ generate a learning method that identifies S in the limit. The following epistemic state constitutes such counter-example:

$$S = \{s_n = \{p_k \mid k \geq n\} \mid n \in \mathbb{N}\}.$$

S is identifiable in the limit by learning method L, that is defined in the following way:

$$L(S, \sigma) = s_n \text{ iff } n \text{ is the smallest such that } \mathrm{set}(\sigma) \subseteq s_n.$$

Moreover, S is identifiable in the limit by a revision-based learning method. We take the conditioning revision method and $\leq \subseteq S \times S$ defined in the following way: For any $s_n, s_m \in S$, $s_n \leq s_m$ iff $n \geq m$. It is easy to observe that $\leq$ is not well-founded.

Let us now assume that S is standardly identifiable in the limit, i.e., there is a belief-revision method R and a well-founded order $\leq$ on S, such that the learning method generated from R and $\leq$ identifies S in the limit. If $\leq$ is well-founded we can choose the $\leq$-minimal element. Let as assume that it is s_k for some $k \in \mathbb{N}$. Obviously, for all $n > k$, $s_n \subseteq s_k$, in fact there are infinitely many $n \in \mathbb{N}$ such that $s_k \leq s_n$ and $s_n \subseteq s_k$. Therefore, all positive, sound and complete data steams for such s_n are also sound with respect to s_k. If we accept the minimal assumption of data-drivenness of belief-revision methods, we can easily see that R will not change the $\leq$-minimal state from s_k to any of s_n, for any sound and complete data streams for s_n. Therefore R fails to identify in the limit s_n, for all $n > k$. □

Summary In this section we considered a notion of reliability of a belief-revision method. We used the concept of identifiability in the limit to define success of an iterated belief-revision process. We have shown that some belief-revision methods are universal, i.e., they identify in the limit all epistemic states that are identifiable by arbitrary learning methods. Such very powerful learning methods are generated from conditioning and lexicographic revision (update and conservative (radical) upgrade in dynamic epistemic logic). More conservative methods turn out not to be universal. This indicates the existence of a tension between learning power and conservatism. We can see that the weakness of the minimal revision method lies in ignoring information that is already believed. Universal belief-revision methods perform operations on plausibility states even if they do not influence the current beliefs immediately. These operations pay off as the process continues.

4.6 Learning from Positive and Negative Data

We will now extend our framework to account for revising with negation. Let us consider the stream ε that consists of both positive and negative data:[6]

$$\mathrm{set}(\varepsilon) \subseteq \textsc{Prop} \cup \{\overline{p} \mid p \in \textsc{Prop}\}.$$

All notions defined in Sections 4.1 and 4.5 (soundness and completeness of a stream, identifiability in the limit, universality, etc.) are analogous for this case.

Let us recall the definition of the epistemic state, together with the additional explanation how to interpret the negative information.

Definition 4.6.1. *Let* Prop *be the a (possibly infinite) set of atomic propositions. A* possible world *is a valuation over* Prop*, and it can be identified with a set* $s \subseteq \textsc{Prop}$*. We say that* p *is true in* s *(write* $s \models p$*) iff* $p \in s$*, we say that* $\overline{p}$ *is true in* s *(write* $s \models \overline{p}$*) iff* $p \notin s$*.*

Proposition 4.6.2. *Conditioning and lexicographic revision generate standardly universal learning methods for positive and negative data.*

Proof. In fact, any ω-type order on S gives a suitable prior plausibility assignment. Let us take $s \in S$. Since $\leq$ is ω-type it is well-founded, there are only finitely many more plausible worlds. For each such world $t \in S$ we collect a $p_n \in t$ such that $p_n \in s \dot{-} t$ ($\dot{-}$ stands for the symmetric difference of two sets). Then we construct a finite data sequence σ enumerating the all the information obtained in this manner, including p_n if $p_n \in s$ or $\bar{p}_n$ if $p_n \notin s$. Obviously σ is an initial segment of some data stream ε for s, hence $\mathrm{set}(\sigma)$ is enumerated in finite time by every data stream ε for s. After $\mathrm{set}(\sigma)$ has been observed all worlds that are more plausible than s will be deleted (in the case of conditioning) or will become less plausible than s. Hence, conditioning and lexicographic revision generate universal learning methods. □

Proposition 4.6.3. *Minimal revision is not universal for positive and negative data.*

Proof. We will give a counterexample, an epistemic state that is identifiable in the limit on positive and negative data streams, but is not identifiable in the limit by the minimal revision method. Let us first introduce the sets crucial for constructing the counterexample. Let $S_{\mathbb{N}} := \{p_i \mid i \in \mathbb{N}\}$, $S_{pos} := \{S_i = \{p_0, \ldots, p_i\} \mid i \in \mathbb{N}\}$, $S_{neg} := \{T_i = S_{\mathbb{N}} - \{p_0 \ldots p_i\} \mid i \in \mathbb{N}\}$. Now we define our epistemic state in the following way:

$$S := \{S_{\mathbb{N}}, \emptyset\} \cup S_{pos} \cup S_{neg}.$$

First let us observe that S is countable, and hence it is identifiable in the limit from positive and negative data (from the proof of Proposition 4.6.2). We will now

[6] In learning theory such streams are called 'informants', see Jain et al., 1999.

show that for any total preorder $\leq$ on S there is a set in S that is not identifiable in the limit by the minimal revision method. We will consider three basic cases: $\emptyset < S_{\mathbb{N}}$, $S_{\mathbb{N}} < \emptyset$ and $S_{\mathbb{N}} \sim \emptyset$.

1. $\emptyset < S_{\mathbb{N}}$. Let $B \subset S$ be the set of all C such that $S_{\mathbb{N}} < C$. There are two cases:

   (a) $B \neq \emptyset$. Then there is a set C such that $\emptyset < S_{\mathbb{N}} < C$ and $C \in S_{pos} \cup S_{neg}$. Then C is not identifiable in the limit by the minimal revision method.

   (b) $B = \emptyset$. Then all sets from S_{pos} are at least as plausible as $S_{\mathbb{N}}$. Then $S_{\mathbb{N}}$ is not identifiable in the limit.

2. $S_{\mathbb{N}} < \emptyset$. Again, let $B \subset S$ be the set of all C such that $\emptyset < C$. Let us again consider two cases.

   (a) $B \neq \emptyset$. Then there is a set C such that $S_{\mathbb{N}} < \emptyset < C$ and $C \in S_{pos} \cup S_{neg}$. Then C is not identifiable in the limit by the minimal revision method.

   (b) $B = \emptyset$. Then all sets from S_{neg} are at least as plausible as $\emptyset$. Then $\emptyset$ is not identifiable in the limit.

3. $\emptyset \sim S_{\mathbb{N}}$. With this assumption the elements of $S_{pos} \cup S_{neg}$ can find themselves in one of the three parts of the preorder. We can have elements that are more plausible than $\emptyset$ (we will call the set of such sets B_1), equally plausible as $\emptyset$ (set of those will be called B_2) or less plausible than $\emptyset$ (B_3). Since our epistemic set is infinite, one of B_1, B_2 and B_3 has to be infinite. Let us again consider three cases:

   (a) B_1 is infinite. Then B_1 has to contain infinitely many sets from S_{pos}, in which case $S_{\mathbb{N}}$ is not identifiable, or infinitely many sets from S_{neg}, in which case $\emptyset$ is not identifiable.

   (b) B_2 is infinite. Then the argument from the above case holds, here for B_2.

   (c) B_3 is infinite. Then B_3 has to contain infinitely many sets from S_{pos}, in which case all sets from $S_{pos} \cap B_3$ are not identifiable, or infinitely many sets from S_{neg}, in which case all sets from $S_{neg} \cap B_3$ are not identifiable.

□

4.6.1 Erroneous Information

If data streams are known to be sound and complete with respect to the actual world, the most economical strategy is to shrink the uncertainty range by deleting those possibilities that contradict the data. This strategy is based on total trust of the information source. However, in belief revision errors might be encountered (in the form of mistakes or lies). Eliminating worlds that contradict the incoming information is then risky and irrational. It is better to change beliefs via some upgrade method, that does not have any built-in mechanism of deletion. Let us compare the performance of upgrading strategies on erroneous data.

To consider errors, we will give up the soundness of data streams, i.e., we will allow data that are false in the real world. To still keep the identification of the real world possible, the data streams are required to be 'fair': there are only finitely many errors, and every error is eventually corrected.

Definition 4.6.4. *A stream ε of positive and negative data is* fair *with respect to the world s iff:*

- *ε is complete with respect to s,*
- *there is $n \in \mathbb{N}$ such that for all $k \geq n$, all the data in $s \models \bigwedge \varepsilon_k$, and*
- *for every $i \in \mathbb{N}$ and for every $\varphi \in \varepsilon_i$ such that $s \not\models \varphi$, there exists some $k \geq i$, such that $\overline{\varphi} \in \varepsilon_k$.*

Notions defined in Subsection 4.5 (identifiability in the limit, universality, etc.) are defined analogously for fair data streams.

We will now demonstrate that lexicographic revision deals with errors in a skillful manner. Before we get to that we will introduce and discuss the notion of *propositional upgrade* (which is a special case of generalized upgrade, see Baltag & Smets, 2009b). Such an upgrade is a transformation of an epistemic-plausibility state that can be given by any finite sequence of mutually disjoint propositional sentences $x_1, \ldots, x_n$. The corresponding propositional upgrade $(x_1, \ldots, x_n)$ acts on an epistemic-plausibility state $(S, \leq_S)$ by changing the preorder $\leq_S$ as follows: all worlds that satisfy x_1 become less plausible than all satisfying x_2, all the worlds satisfying x_2 become less plausible than all x_3 worlds, etc., up to the worlds which satisfy x_n. Moreover, for any k such that $1 \leq k \leq n$, among the worlds satisfying x_k the old order $\leq_S$ is kept. In particular, our lexicographic revision is a special case of such propositional upgrade, namely in these terms lexicographic revision with φ can be identified with the propositional upgrade $(\neg\varphi, \varphi)$.

Lemma 4.6.5. *The class of propositional upgrades is closed under sequential composition.*

Proof. We need to show that the sequential composition of any two propositional upgrades gives a propositional upgrade. Let us take $X := (x_1, \ldots, x_n)$ and $Y := (y_1, \ldots, y_m)$. The sequential composition XY is equivalent to the following propositional upgrade:

$$(x_1 \wedge y_1, \ldots, x_1 \wedge y_n, x_1 \wedge y_2, \ldots, x_n \wedge y_2, \ldots, x_1 \wedge y_m, \ldots, x_n \wedge y_m).$$

To show this let us take an arbitrary epistemic-plausibility state $(S, \leq_S)$ and apply upgrades X and Y successively. First, we apply to $(S, \leq_S)$ the upgrade X.

We obtain the new preorder $\leq_S^X$, in which all worlds satisfying x_1 are less plausible than all x_2-worlds, etc., and within each such partition the old order $\leq_S$ is kept. Now, to this new epistemic-plausibility state we apply the second upgrade, Y, obtaining the new preorder $\leq_S^{XY}$, in which all y_1-worlds are less plausible than all y_2-worlds, etc. However, since the upgrade Y has been applied to the preorder $\leq_S^X$ we also know that the new preorder $\leq_S^{XY}$ has the following property: for each j, such that $1 \leq j \leq m$, within the partition given by y_j, we have that all x_1-worlds are less plausible than all x_2-worlds, etc. At the same time in each j and k, such that $1 \leq j \leq m$ and $1 \leq k \leq n$, in the partition $(y_j \wedge x_k)$ the preorder $\leq_S$ is maintained.

Putting this together, we get that $\leq_S^{XY}$ has the following structure:

$$\|(x_1 \wedge y_1)\| \geq_S^{XY} \ldots \geq_S^{XY} \|(x_n \wedge y_1)\| \geq_S^{XY}$$

$$\|(x_1 \wedge y_2)\| \geq_S^{XY} \ldots \geq_S^{XY} \|(x_n \wedge y_2)\| \geq_S^{XY} \ldots \geq_S^{XY} \|(x_n \wedge y_m)\|.$$

Moreover, within each such partition, the old preorder $\leq_S$ is kept.

The final observation is that the above setting can be obtained directly by the propositional upgrade of the following form:

$$(x_1 \wedge y_1, \ldots, x_1 \wedge y_n, x_1 \wedge y_2, \ldots, x_n \wedge y_2, \ldots, x_1 \wedge y_m, \ldots, x_n \wedge y_m).$$

□

Now we are ready to show that lexicographic revision is well-behaved on fair streams.

Proposition 4.6.6. *Lexicographic revision generates a standardly universal belief-revision-based learning method for fair streams of positive and negative data.*

Proof. First let us recall that lexicographic revision, Lex, is standardly universal on positive and negative data. For the above conjecture it is left to be shown that it retains its power on fair streams. It is sufficient to show that lexicographic revision is 'error-correcting': the effect of revising with the stream $\varphi, \sigma, \overline{\varphi}$ is exactly the same as with the stream $\sigma, \overline{\varphi}$, where σ is a sequence of propositions. The proof uses the properties of sequential composition for propositional upgrade.

Let us assume that length$(\sigma) = n$. In terms of generalized upgrade we need to demonstrate that the sequential composition $(\neg\varphi, \varphi)(\neg\sigma_1, \sigma_1)\ldots(\neg\sigma_n, \sigma_n)(\varphi, \neg\varphi)$ is equivalent to $(\neg\sigma_1, \sigma_1)\ldots(\neg\sigma_n, \sigma_n)(\varphi, \neg\varphi)$.

From Lemma 4.6.5 we know that propositional upgrade is closed under sequential composition. Hence, in the equivalence to be shown, we can replace the composition $(\neg\sigma_1, \sigma_1)\ldots(\neg\sigma_n, \sigma_n)$ by only one generalized upgrade, which we will denote by $(x_1, \ldots, x_m)$. Now, we have to show that: $(\neg\varphi, \varphi)(x_1, \ldots, x_m)(\varphi, \neg\varphi)$ is equivalent to: $(x_1, \ldots, x_m)(\varphi, \neg\varphi)$.

By the proof of Lemma 4.6.5, the composition $(x_1, \ldots, x_n)(\varphi, \neg\varphi)$ has the following form:

$$(x_1 \wedge \varphi, \ldots, x_n \wedge \varphi, x_1 \wedge \neg\varphi, \ldots, x_n \wedge \neg\varphi).$$

Accordingly, the other upgrade, $(\neg\varphi,\varphi)(x_1,\ldots,x_n)(\varphi,\neg\varphi)$, has the following form:

$$(\neg\varphi \wedge x_1 \wedge \varphi, \varphi \wedge x_1 \wedge \varphi, \ldots, \neg\varphi \wedge x_n \wedge \varphi, \varphi \wedge x_n \wedge \varphi, \neg\varphi \wedge x_1 \wedge \neg\varphi, \varphi \wedge x_1 \wedge \neg\varphi, \ldots,$$

$$\neg\varphi \wedge x_n \wedge \neg\varphi, \varphi \wedge x_n \wedge \neg\varphi).$$

Let us observe that some of the terms in the above upgrade are inconsistent. We can eliminate them since they correspond to empty subsets of the epistemic-plausibility state. We obtain:

$$(x_1 \wedge \varphi, \ldots, x_n \wedge \varphi, x_1 \wedge \neg\varphi, \ldots, x_n \wedge \neg\varphi).$$

The observation that the two propositional upgrades turn out to be the same concludes the proof. □

Proposition 4.6.7. *Conditioning and minimal revision are not universal for fair streams.*

Proof. Conditioning does not tolerate errors at all. On any ε_i such that $\varepsilon_i \nsubseteq s$ conditioning will remove s and it does not provide a way to revive it. Minimal revision, as it has been shown, is not universal on regular positive and negative data streams, which are a special case of fair streams. □

Summary In this section we have shown how the framework of iterated belief revision can be enriched by the use of negative information. First, we investigated positive and negative information that is sound and complete. In this case, both conditioning and lexicographic revision are standardly universal, i.e., there are well-founded total orders that, together with either of the two mentioned belief-revision methods, generate universal learning methods. Minimal revision again turns out to be insufficient. Secondly, we define fair data streams that use both positive and negative information. Such fair streams contain a finite number of errors and every error is eventually corrected later in the stream. The conditioning revision method again proves to be universal on fair streams, because it overrides inconsistent information. Conditioning and minimal revision lack this error-correcting property.

4.7 Conclusions and Perspectives

We have considered iterated belief-revision policies of conditioning, lexicographic and minimal revision. We have identified certain features of those methods relevant in the context of iterated revision: data-retention, conservatism, and history-independence. We defined learning methods based on those revision policies and we have shown how the aforementioned properties influence the learning process. Throughout this chapter we have been mainly interested in convergence to the actual world on the basis of infinite data streams. In the setting of positive, sound, and complete data streams we have exhibited that conditioning and lexicographic revision generate universal learning methods. Minimal revision fails to be universal, and the crucial property that makes it weaker is its strong conservatism. Moreover, we have shown that the full power of learning cannot be achieved when the underlying prior plausibility assignment is assumed to be well-founded. In the case of positive and negative information, both conditioning and lexicographic revision are universal. Minimal revision again is not. Finally, in the setting of fair streams (containing a finite number of errors that all get corrected later in the stream) lexicographic revision again turns out to be universal. Both conditioning and minimal revision lack the 'error-correcting' property.

Future work consists in multi-level investigation of the relationship between learning theory, belief revision, and dynamic epistemic logic. There surely are many links still to be found, with interesting results for everyone involved. What seems to be especially interesting is the multi-agent extension of our results. In terms of the efficiency of convergence it would enrich the multi-agent approach to information flow, an interesting subject for epistemic and doxastic logic. The interactive aspect would probably be appreciated in formal learning theory, where the single-agent perspective is clearly dominating. Another way to extend the framework is to allow revision with more complex formulae. This would perhaps link to the AGM approach, and to the philosophical investigation into the process of scientific inquiry, where possible realities have a more 'theoretical' character.

Chapter 5

Epistemic Characterizations of Identifiability

In this chapter we will further investigate the connection between formal learning theory and modal-temporal logics of belief change. We will again focus on the language-learning paradigm, in which languages are treated as sets of positive integers.[1] In the previous chapter we focused on the semantic analysis of identifiability in the limit. Now, we will devote more attention to the syntactic counterparts of our logical approach to identifiability, focusing on both finite identifiability and identifiability in the limit. We will show how the previously chosen semantics can be reflected in an appropriate syntax for knowledge, belief, and their changes over time. The corresponding notions of learning theory and dynamic epistemic logic are given in Chapter 2.

Our approach to inductive learning in the context of dynamic epistemic and epistemic temporal logic is as follows. As in the previous chapter, we take the initial class of sets to be possible worlds in an epistemic model, which mirrors Learner's initial uncertainty over the range of sets. The incoming pieces of information are taken to be events that modify the initial model. We will show that iterated update on epistemic models based on finitely identifiable classes of sets is bound to lead to the emergence of irrevocable knowledge. In a similar way identifiability in the limit leads to the emergence of stable belief. Next, we observe that the structure resulting from updating the model with a sequence of events can be viewed as an epistemic temporal forest. We explicitly focus on protocols that are assigned to worlds in set-learning scenarios. We give a temporal characterization of forests that are generated from learning situations of finite identifiability and identifiability in the limit. We observe that a special case of this protocol-based setting, in which only one stream of events is allowed in each state, can be used to model the function-learning paradigm. We show that the simple setting of iterated epistemic update cannot account for all possible learning situations. In

[1] In this chapter we are concerned with logical characterizations of learning, hence we will often refer to *languages* of certain logics. To avoid confusion for the time being we will replace the name *language learning* with *set learning* (see Section 2.1).

the end we conclude our considerations and present possible directions of further work.

5.1 Learning and Dynamic Epistemic Logic

Following our observations about the power of the conditioning revision method (Chapter 4) we will still be concerned with epistemic update. To recall the idea let us consider some simple examples of single-agent propositional update.

Example 5.1.1. *Let us take a single-agent epistemic model* $\mathcal{M} = \langle W, \sim, V \rangle$*, where* $W = \{w_1, w_2, w_3\}$*,* $\sim = W \times W$*,* $\textsc{Prop} = \{p_1, p_2, p_3, p_4\}$ *and the valuation* $V : \textsc{Prop} \to \mathcal{P}(W)$ *is defined in the following way* $V(p_1) = \{w_1, w_2, w_3\}$*,* $V(p_2) = \{w_1, w_2\}$*,* $V(p_3) = \{w_2, w_3\}$*,* $V(p_4) = \{w_3\}$*, in other words:* $w_1 \models p_1 \wedge p_2 \wedge \neg p_3 \wedge \neg p_4$*,* $w_2 \models p_1 \wedge p_2 \wedge p_3 \wedge \neg p_4$*, and* $w_3 \models p_1 \wedge p_3 \wedge p_4 \wedge \neg p_2$*. Let us assume that* w_2 *is the actual world, and that the agent receives propositional information that is consistent with* w_2 *in the following order:* p_1, p_2, p_3*. Receiving* p_1 *does not change anything—every world satisfies* p_1*. Then* p_2 *comes in, eliminating* w_3*, since* $w_3 \not\models p_2$*. The agent is now uncertain only between* w_1 *and* w_2*. The last information* p_3 *allows deleting* w_1 *because* $w_1 \not\models p_3$*. The uncertainty of the agent now disappears—the only possibility left is* w_2*. Moreover, whatever true (consistent with the actual world* w_2*) information comes in,* w_2 *cannot be eliminated.*

In fact, if any of the worlds is the actual one, and the agent will receive truthful and complete propositional information about it, he will be able to eventually eliminate all other worlds, and therefore gain full certainty about his situation.

Example 5.1.2. *Let us again take a similar epistemic model, this time with the following valuation* $V(p_1) = \{w_1, w_2, w_3\}$*,* $V(p_2) = \{w_1, w_2\}$*,* $V(p_3) = \{w_1\}$*. Now, only one world, namely* w_1*, can get identified by receiving truthful and complete propositional information. In case* w_2 *(or* w_3*) is the actual world, the agent will never be able to eliminate* w_1 *(or* w_1 *and* w_2*), and therefore the uncertainty will always remain.*

5.1.1 Dynamic Epistemic Learning Scenarios

In Examples 5.1.1 and 5.1.2, the uncertainty range of the agent is revised as new pieces of data (in the form of propositions) are received. The information comes from a completely trusted source, and as such causes the agents to eliminate the worlds that do not satisfy it. In learning theory it is common to assume the truthfulness of incoming data, and therefore, in principle, it is justified to use epistemic update as a way to perform the inquiry (for such interpretation of update see Van Benthem, 2006). It is important to note that public announcement is not the main notion of dynamic epistemic logic. Our update-based approach to learning gives the first connection, but dynamic epistemic logic can typically also deal with varieties of 'soft information' that is less trusted (see Section 2.2).

In this section we will present single-agent learning scenarios in the framework of doxastic epistemic logic. We base our investigations on the learning-theoretic framework defined in Section 2.1.

First, the initial learning model is a simple epistemic model whose worlds correspond to the initial class of sets.

Definition 5.1.3. *Let $\mathcal{C} = \{S_1, S_2, \ldots\}$ be a class of sets such that for all $i \in \mathbb{N}$, $S_i \subseteq \mathbb{N}$. Our* initial learning model *$\mathcal{M}_\mathcal{C}$ is a triple:*

$$\langle W_\mathcal{C}, \sim, V_\mathcal{C} \rangle,$$

where $W_\mathcal{C} = \mathcal{C}$, $\sim = W_\mathcal{C} \times W_\mathcal{C}$, $V_\mathcal{C} : \textsc{Prop} \cup \textsc{nom} \to \mathcal{P}(W_\mathcal{C})$, such that $S_i \in V_\mathcal{C}(p_n)$ iff $n \in S_i$ and for each set $S_i \in \mathcal{C}$, we take a nominal $i \in \textsc{nom}$ and we set $V_\mathcal{C}(i) = \{S_i\}$.

In words, we identify states of the model with sets, we also assume that our agent does not have any particular initial information or preference over the possibilities. The interpretation of the propositional letters is as follows. Let $\mathcal{C} = \{S_1, S_2, \ldots\}$ be a class of sets, and let $U = \bigcup \mathcal{C}$ be the universal set of $\mathcal{C}$. For every piece of data $n \in U$ we take a propositional letter p_n. The nominals correspond to indices of sets. They can be interpreted as finite descriptions of sets or as theories that describe possible sequences of events.

In the previous chapter we analyzed our central topic of *iterated* update. The definitions of data streams, data sequences and related notions remain the same for this chapter. We will be concerned with sound and complete data streams (see Section 4.1).

5.1.2 Finite Identification in DEL

The research in dynamic epistemic and dynamic doxastic logic often touches the subject of converging to some desired states: (common) knowledge or (joint) true belief (see, e.g., Baltag & Smets, 2009a). In this respect it is concerned with multi-agent versions of the belief-revision problem. In this section we will show how to use the notion of finite identification to characterize convergence to irrevocable knowledge. To establish the first connection we will restrict ourselves to the single-agent case.

Definition 5.1.4. *Iterated epistemic update of model $\mathcal{M}$ with an infinite data stream ε* stabilizes *to $\mathcal{M}'$ iff there is an $n \in \mathbb{N}$, such that for all $m \geq n$, $\mathcal{M}^{\varepsilon \restriction m} = \mathcal{M}'$. In such cases we will sometimes write that the generated epistemic model $\mathcal{M}^\varepsilon$ stabilizes to $\mathcal{M}'$.*

In our considerations we will use the characterization of finite identifiability of sets from positive data (Mukouchi, 1992). First we recall the notion of the finite definite tell-tale set.

Definition 5.1.5 (Mukouchi 1992)**.** *A set D_i is a* definite finite tell-tale set *of $S_i \in \mathcal{C}$ if*

1. *$D_i \subseteq S_i$,*
2. *D_i is finite, and*
3. *for any index j, if $D_i \subseteq S_j$ then $S_i = S_j$.*

The non-computable case of finite identifiability can be then characterized in the following way.

Theorem 5.1.6 (Mukouchi 1992)**.** *A class $\mathcal{C}$ is finitely identifiable from positive data if and only if for every set $S_i \in \mathcal{C}$ there is a definite finite tell-tale set D_i.*

We are now ready to show that epistemic update performed on finitely identifiable class of sets leads to irrevocable knowledge.

Theorem 5.1.7. *The following are equivalent:*

1. *$\mathcal{C}$ is finitely identifiable.*
2. *For every $S_i \in W_{\mathcal{C}}$ and every data stream ε for S_i the generated epistemic model $\mathcal{M}_{\mathcal{C}}^{\varepsilon}$ stabilizes to $\mathcal{M}'_{\mathcal{C}} = \langle W'_{\mathcal{C}}, \sim', V_{\mathcal{C}} \rangle$, where $W'_{\mathcal{C}} = \{S_i\}$ and $\sim' = \{(S_i, S_i)\}$.*

Proof. The proof of this assertion consists mainly in understanding our earlier semantic definitions and arguments, and seeing that they conform to a simple syntactic pattern definable in epistemic logic. Nevertheless, for once, we add some explicit formal detail to show how this works.

(1 ⇒ 2) Let us assume that $\mathcal{C}$ is finitely identifiable. Then, by Theorem 5.1.6, for every set $S_i \in \mathcal{C}$ there is a finite definite tell-tale set $D_i \subseteq S_i$ such that D_i is not a subset of any other set in $\mathcal{C}$. Let us then take one S_i and the corresponding finite definite tell-tale set D_i. For every data stream ε for S_i there is a finite initial segment, $\varepsilon \upharpoonright m$, such that $D_i \subseteq \text{set}(\varepsilon \upharpoonright m)$. Then by stage m every S_j such that $i \neq j$ has been eliminated by the update.

(2 ⇒ 1) Let us assume that for every $S_i \in W_{\mathcal{C}}$ and a data stream ε for S_i, the generated epistemic model $\mathcal{M}_{\mathcal{C}}^{\varepsilon}$ stabilizes to $\mathcal{M}'_{\mathcal{C}} = \langle W'_{\mathcal{C}}, \sim', V_{\mathcal{C}} \rangle$, where $W'_{\mathcal{C}} = S_i$ and $\sim' = \{(S_i, S_i)\}$. Assume that $\mathcal{C}$ is not finitely identifiable. Therefore, by Theorem 5.1.6, there is a set $S_i \in \mathcal{C}$ such that every finite subset of S_i is included in some $S_j \in \mathcal{C}$ such that $i \neq j$. Then for all n, if $\mathcal{M}_{\mathcal{C}}^{\varepsilon\upharpoonright n} = \langle W_{\mathcal{C}}^{\varepsilon\upharpoonright n}, \sim^{\varepsilon\upharpoonright n}, V_{\mathcal{C}}^{\varepsilon\upharpoonright n} \rangle$ then $\{S_i, S_j\} \subseteq W_{\mathcal{C}}^{\varepsilon\upharpoonright n}$, so $\mathcal{M}_{\mathcal{C}}^{\varepsilon}$ clearly does not stabilize to $\mathcal{M}'_{\mathcal{C}} = \langle W'_{\mathcal{C}}, \sim', V_{\mathcal{C}} \rangle$, where $W'_{\mathcal{C}} = \{S_i\}$ and $\sim' = \{(S_i, S_i)\}$. Contradiction. □

With respect to the language of epistemic logic $\mathcal{L}_{\mathrm{EL}}$ given in Definition 2.2.3, the following corollary corresponds to the semantic characterization in Theorem 5.1.7.

Corollary 5.1.8. *The following are equivalent:*

1. *$\mathcal{C}$ is finitely identifiable.*
2. *For every $S_i \in W_{\mathcal{C}}$ and every data stream ε for S_i the generated epistemic model $\mathcal{M}^{\varepsilon}_{\mathcal{C}}$ stabilizes to $\mathcal{M}'_{\mathcal{C}} = \langle W'_{\mathcal{C}}, \sim', V_{\mathcal{C}}\rangle$, where $W'_{\mathcal{C}} = \{S_i\}$ and $\mathcal{M}'_{\mathcal{C}}, S_i \models K\, i$.*

Proof. From Theorem 5.1.7 we know that 1 is equivalent to:

# For all $S_i \in W_{\mathcal{C}}$ and every data stream ε for S_i the generated epistemic model $\mathcal{M}^{\varepsilon}_{\mathcal{C}}$ stabilizes to $\mathcal{M}'_{\mathcal{C}} = \langle W'_{\mathcal{C}}, \sim', V_{\mathcal{C}}\rangle$, where $W'_{\mathcal{C}} = \{S_i\}$ and $\sim' = \{(S_i, S_i)\}$.

($\# \Rightarrow 2$) Let us take $S_i \in W_{\mathcal{C}}$ and data stream ε for S_i and assume that the generated epistemic model $\mathcal{M}^{\varepsilon}_{\mathcal{C}}$ stabilizes to $\mathcal{M}'_{\mathcal{C}} = \langle W'_{\mathcal{C}}, \sim', V_{\mathcal{C}}\rangle$, where $W'_{\mathcal{C}} = \{S_i\}$ and $\sim' = \{(S_i, S_i)\}$. Then, by definition of the semantics of $\mathcal{L}_{EL}$, $\mathcal{M}', S_i \models K\, i$, since it is true that for all $S_j \in \mathcal{K}[S_i]$, we have that $\mathcal{M}'_{\mathcal{C}}, S_j \models i$.

($2 \Rightarrow \#$) Let us assume that for every $S_i \in W_{\mathcal{C}}$ and every data stream ε for S_i the generated epistemic model $\mathcal{M}^{\varepsilon}_{\mathcal{C}}$ stabilizes to $\mathcal{M}'_{\mathcal{C}} = \langle W'_{\mathcal{C}}, \sim', V_{\mathcal{C}}\rangle$, where $W'_{\mathcal{C}} = \{S_i\}$ and $\mathcal{M}'_{\mathcal{C}}, S_i \models K\, i$. This means that for all $S_j \in \mathcal{K}[S_i]$ we have that $\mathcal{M}'_{\mathcal{C}}, S_j \models i$. But from definition of the valuation $V_{\mathcal{C}}$ we know that S_i is the only state in $W_{\mathcal{C}}$ that validates i. Therefore $\sim' = \{(S_i, S_i)\}$. □

The above results provide a characterization of the outcome of finite identification in the simple language of epistemic logic. To incorporate more dynamics into the syntactic counterpart of finite learning we can use the language $\mathcal{L}_{\mathrm{PAL}}$.[2]

Corollary 5.1.9. *The following are equivalent:*

1. *$\mathcal{C}$ is finitely identifiable.*
2. *For every $S_i \in W_{\mathcal{C}}$ and every data stream ε for S_i there is an $n \in \mathbb{N}$ such that for all $m \geq n$, $\mathcal{M}_{\mathcal{C}}, S_i \models [!(\bigwedge \mathrm{set}(\varepsilon{\upharpoonright}m))]\, K\, i$.*

Proof. This equivalence follows directly from Corollary 5.1.8 and the semantics of $\mathcal{L}_{\mathrm{PAL}}$, Definition 2.2.8. □

[2] To bring out the future behavior more explicitly in our syntax, we could also formulate this result in terms of *repeated announcement*, in a version of public announcement logic that also allows Kleene star. We forego such extensions here.

5.1.3 Identification in the Limit and DDL

In Chapter 4 we extensively discussed the interrelation between identifiability in the limit, update (conditioning), and the notion of belief. Now, on the basis of those results we can give the following corollary.

Corollary 5.1.10. *The following are equivalent:*

1. *$\mathcal{C}$ is identifiable in the limit.*
2. *There is a plausibility preorder $\leq\, \subseteq W_{\mathcal{C}} \times W_{\mathcal{C}}$ such that for every $S_i \in W_{\mathcal{C}}$ and every data stream ε for S_i in the generated epistemic model $\mathcal{M}^{\varepsilon}_{\mathcal{C}}$, $\min_{\leq} W^{\varepsilon}_{\mathcal{C}}$ stabilizes to $\{S_i\}$.*
3. *There exists a plausibility preorder $\leq\, \subseteq W_{\mathcal{C}} \times W_{\mathcal{C}}$ such that for every $S_i \in W_{\mathcal{C}}$ and every data stream ε for S_i there is $n \in \mathbb{N}$ such that for all $m \geq n$, $(\mathcal{M}_{\mathcal{C}}, \leq), S_i \models [!(\bigwedge \mathrm{set}(\varepsilon\restriction m))]B\, i$.*

Clause 3 gives a characterization in public announcement logic with the operator of absolute belief.[3] The plausibility order used in the above corollary is defined and discussed in Section 4.5. It is based on the characterization of identification in the limit and the concept of the finite tell-tale set (see Section 2.1).

Let us additionally note that the last clauses of Corollaries 5.1.8, 5.1.9, and 5.1.10 describe the *persistence* of the relevant doxastic-epistemic states. In fact, under the conditions of update, it is the case that as soon as the desired doxastic-epistemic state is reached it cannot be lost later in the process.

Until now we have shown how to model learning scenarios in dynamic epistemic and doxastic logic. In order to explicitly express the possibility of convergence as a temporal property, we will view the structure generated by iterated epistemic update as a temporal branching model. In this we follow the recently established bridge between dynamic epistemic and epistemic temporal logic (see Van Benthem et al., 2009).

5.2 Learning and Temporal Logic

We have shown how basic results connecting dynamic epistemic logic and learning theory can be given syntactic formulations in terms of the K and B operators. However, we are still missing a crucial dimension, the *temporal* one. Implicitly, we already considered the temporal aspects, since in fact the knowledge and beliefs stabilized only after some finite sequences of announcements or other

[3] Given the reduction axioms of dynamic doxastic logic for ‘factual formulae’, such as the nominal i, an equivalent formulation would be: There is a plausibility preorder $\leq\, \subseteq W_{\mathcal{C}} \times W_{\mathcal{C}}$ such that for every $S_i \in W_{\mathcal{C}}$ and every data stream ε for S_i there is $n \in \mathbb{N}$ such that for all $m \geq n$, $(\mathcal{M}_{\mathcal{C}}, \leq), S_i \models B^{(\bigwedge \mathrm{set}(\varepsilon\restriction m))}\, i$.

informative events. This long-term aspect could be formalized in extensions of public announcement logic with program operations, in particular, Kleene iteration. While this seems an interesting line to pursue, we feel that this still does not do justice to another striking logical feature of learning theory: its resemblance to *temporal logics*. In what follows, we will show how to establish the connection, taking advantage of some recent developments that have linked dynamic epistemic logic to epistemic temporal logics, via the crucial notion of a *protocol*.

To make this connection, we need to turn to the more general version of DEL based on *event models* and *product update* (Baltag et al., 1998). We will just give the absolute basics here, referring mainly to the literature.

5.2.1 Event Models and Product Update

Iterated update can be placed in a more general perspective. Obviously, the incoming information does not have to be propositional. It does not even have to be purely linguistic. It can be any *event* that itself has an epistemic structure. To consider changes caused by such arbitrary events, we will now introduce the notion of event model, which represents the epistemic and informational content of what 'happens'.

Definition 5.2.1. *An* event model *is a triple:*

$$\mathcal{E} = \langle E, (\sim_a^{\mathcal{E}})_{a\in\mathcal{A}}, \mathtt{pre}\rangle,$$

where $E \neq \emptyset$ is a set of events; for every agent $a \in \mathcal{A}$, $\sim_a^{\mathcal{E}}$ is a binary equivalence relation on E, and $\mathtt{pre} : E \to \mathcal{L}_{EL}$, is a precondition function where $\mathcal{L}_{EL}$ is a set of formulae of some epistemic language. A pair $(\mathcal{E}, e)$, where $e \in E$ is called a pointed event model.

For every agent $a \in \mathcal{A}$, the relation $\sim_a^{\mathcal{E}}$ encodes that agent's epistemic information about the event taking place. The precondition function `pre` maps events to epistemic formulae. An event will be executable in some state only if that state satisfies the precondition of this event.

The effect of updating an epistemic model $\mathcal{M}$ with an event model $\mathcal{E}$ can be computed according to the *product update.*

Definition 5.2.2. *Let $\mathcal{M} = \langle W, (\sim_a)_{a\in\mathcal{A}}, V\rangle$ be an epistemic model and $\mathcal{E} = \langle E, (\sim_a^{\mathcal{E}})_{a\in\mathcal{A}}, \mathtt{pre}\rangle$ be an event model. The* product update *of $\mathcal{M}$ with $\mathcal{E}$ gives a new epistemic model $\mathcal{M} \otimes \mathcal{E} = \langle W', (\sim'_a)_{a\in\mathcal{A}}, V'\rangle\}$, where:*

1. $W' = \{(w, e) \mid w \in W \;\&\; e \in E \;\&\; w \models \mathtt{pre}(e)\}$;
2. $(w, e) \sim'_a (w', e')$ iff $w \sim_a w'$ and $e \sim_a^{\mathcal{E}} e'$;
3. *and the valuation is as follows:* $(w, e) \in V'(p)$ iff $w \in V(p)$.

Illustrations of the strength of product update can be found in (Baltag & Moss, 2004; Van Benthem, 2010; Van Benthem & Dégremont, 2010; Dégremont, 2010).

5.2.2 Dynamic Epistemic Logic Protocols

By making a step from dynamic epistemic logic into epistemic temporal logic we can analyze the temporal structure of update. Redefining the iterated epistemic update in terms of protocols (see Fagin, Halpern, Moses, & Vardi, 1995; Parikh & Ramanujam, 2003) will bring us closer to the temporal setting. A protocol specifies sequences of events that are admissible in certain epistemic situations. In this section, following Van Benthem et al. (2009), we will give the definition of local protocols and epistemic models generated with respect to a protocol. By doing this we prepare the grounds for our learning-theoretic setting.

A protocol P maps states in an epistemic model to sets of finite and infinite sequences of event models closed under taking prefixes. It defines the admissible runs of some informational process: In general, not every sequence of events may be possible at a given state.

Let $\mathbb{E}$ be the class of all event models. Every state of the epistemic model is assigned a set of sequences (infinite and finite) of event models closed under taking finite prefixes, an element of the set

$$\mathrm{Prot}(\mathbb{E}) = \{\mathtt{P} \subseteq \mathcal{P}(\mathbb{E}^* \cup \mathbb{E}^\omega) \mid \mathtt{P} \text{ is closed under finite prefixes}\}.$$

Definition 5.2.3. *Let us take an epistemic model $\mathcal{M} = \langle W, (\sim_a)_{a\in\mathcal{A}}, V\rangle$. A* local protocol *for $\mathcal{M}$ is a function $P : W \to \mathrm{Prot}(\mathbb{E})$.*

Until now we have been concerned with the $\varepsilon\restriction n$-generated epistemic model $\mathcal{M}$, where $\varepsilon\restriction n$ is some sequence of propositions. We will now provide an analogous notion of a model generated from a sequence of event models but according to some specific local protocol.

Definition 5.2.4. *Let $\mathcal{M} = \langle W, (\sim_a)_{a\in\mathcal{A}}, V\rangle$ be an epistemic model. We define the* $(P, \varepsilon\restriction n)$-generated epistemic model $\mathcal{M}^{P,\varepsilon\restriction n}$ *inductively, as follows:*

$$\begin{array}{lcl} \mathcal{M}^{P,\varepsilon\restriction 0} & = & \mathcal{M} \\ \mathcal{M}^{P,\varepsilon\restriction n+1} & = & \langle W^{P,\varepsilon\restriction n+1}, \sim^{P,\varepsilon\restriction n+1}, V^{P,\varepsilon\restriction n+1}\rangle, \textit{ where:} \\ & & W^{P,\varepsilon\restriction n+1} := \{s \mid s \in W^{P,\varepsilon\restriction n}; s \models \mathtt{pre}(\varepsilon_{n+1}) \;\&\; \varepsilon\restriction n+1 \in P(s)\}; \\ & & \sim^{P,\varepsilon\restriction n+1} := \sim^{P,\varepsilon\restriction n}\restriction W^{P,\varepsilon\restriction n+1}; \\ & & V^{P,\varepsilon\restriction n+1} := V^{P,\varepsilon\restriction n}\restriction W^{P,\varepsilon\restriction n+1}. \end{array}$$

The protocol-based approach to update has a straightforward temporal interpretation. The question is how iterated product update can be interpreted in epistemic temporal logics, which are widely used to study the evolution of a system over time focusing on the information that agents possess. And this perspective is exactly what we need.

5.2.3 Dynamic Epistemic and Epistemic Temporal Logic

Epistemic temporal logics are interpreted on epistemic temporal forests (see, e.g., Parikh & Ramanujam, 2003).

Definition 5.2.5. An epistemic temporal model $\mathcal{H}$ *is a tuple:*

$$\langle W, \Sigma, H, (\sim_a)_{a\in\mathcal{A}}, V\rangle,$$

where $W \neq \emptyset$ *is a countable set of initial states;* Σ *is a countable set of events;* $H \subseteq W\Sigma^*$ *is a set of histories (sequences of events starting at states from* W*) closed under non-empty finite prefixes; for each* $a \in A$, $\sim_a \subseteq H \times H$ *is an equivalence relation; and* $V : \textsc{Prop} \to \mathcal{P}(H)$ *is a valuation. We write* wh *to denote some finite history starting in the state* w*.*

We sometimes refer to the $\langle W, \Sigma, H\rangle$-part of an ETL model as the *temporal protocol* this model is based on. We refer to the information of an agent a at h with $\mathcal{K}_a[wh] = \{vh' \in H \mid wh \sim_a vh'\}$.

The question is now how to make the step from dynamic epistemic logic to epistemic temporal logic. The relation between the two frameworks has already been studied (see, e.g., Van Benthem & Liu, 2004; Van Benthem & Pacuit, 2006). In particular, it has been observed that iterated epistemic update in dynamic epistemic logic generates epistemic temporal forests satisfying certain properties (see Van Benthem et al., 2009). We will refer to this construction by $\texttt{For}(\mathcal{M}, P)$ and define it below.

We construct the forest by induction, starting with the epistemic model and and then checking which events can be executed according to the precondition function and to the protocol. Finally, the new information partition is updated at each stage according to the product update. Since product update describes purely epistemic change, the valuation stays the same as in the initial model.

Definition 5.2.6 (ETL forest generated by a DEL protocol)**.** *Each epistemic model* $\mathcal{M} = \langle W, (\sim_a^{\mathcal{M}})_{a\in\mathcal{A}}, V^{\mathcal{M}}\rangle$ *and a local protocol* $P : W \to \mathrm{Prot}(\mathbb{E})$ *generates an ETL forest* $\texttt{For}(\mathcal{M}, P)$ *of the form:*

$$\mathcal{H} = \langle W^{\mathcal{H}}, \mathbb{E}, H, (\sim_a)_{a\in\mathcal{A}}, V\rangle, \text{ where:}$$

1. $W^{\mathcal{H}} := W$;

2. H *is defined inductively as follows:*

   $H_0 := W^{\mathcal{H}}$;
   $H_{n+1} := \{(we_1 \ldots e_{n+1}) \mid (we_1 \ldots e_n) \in H_n; \mathcal{M}^{\varepsilon\restriction n}, w \models \texttt{pre}(e_{n+1})$
   and $(e_1 \ldots e_{n+1}) \in P(w)\}$;
   $H := \bigcup_{0\leq k<\omega} H_k$;

3. *If* $w, v \in W^{\mathcal{H}}$, *then* $w \sim_a v$ *iff* $w \sim_a^{\mathcal{M}} v$;

4. $whe \sim_a vh'e'$ *iff* $whe, vh'e' \in H_k$, $wh \sim_a vh'$, e *and* e' *are states in an event model* $\mathcal{E}$ *and* $e \sim_a^{\mathcal{E}} e'$;

5. *Finally, $wh \in V(p)$ iff $w \in V^{\mathcal{M}}(p)$.*

The correspondence between the iterated product update and an epistemic temporal forest relies on some properties of epistemic temporal agents. To be precise, it has been shown that the structures of iterated DEL update are in fact epistemic temporal forests that satisfy the following conditions: perfect recall, synchronicity, uniform no miracles and propositional stability. Let us introduce those epistemic multi-agent assumptions.

Definition 5.2.7. *Let us take $\mathcal{H} = \langle W, \Sigma, H, (\sim_a)_{a\in\mathcal{A}}, V\rangle$ to be an epistemic temporal model.*

Perfect Recall *$\mathcal{H}$ satisfies perfect recall iff*

$$\textit{for all } whe, vh'f \in H \textit{ if } \mathcal{K}_a[whe] = \mathcal{K}_a[vh'f], \textit{ then } \mathcal{K}_a[wh] = \mathcal{K}_a[vh'].$$

The condition of perfect recall expresses that agents do not forget past information as further events take place.

Synchronicity *$\mathcal{H}$ satisfies synchronicity iff*

$$\textit{for all } wh, vh' \in H \textit{ if } \mathcal{K}_a[wh] = \mathcal{K}_b[vh'], \textit{ then } \mathrm{length}[wh] = \mathrm{length}[vh'].$$

Synchronicity is satisfied if the agents have access to some external discrete clock and thus can keep track of the time.

Uniform-No-Miracles *$\mathcal{H}$ satisfies uniform no miracles iff*

$$\textit{for all } wh, vh' \in H \textit{ such that } wh \sim_a vh'$$
$$\textit{and for all } e_1, e_2 \in \Sigma \textit{ with } whe_1, vh'e_2 \in H$$
$$\textit{if there are } sh'', th''' \in H \textit{ such that } sh''e_1 \sim_a th'''e_2, \textit{ then } whe_1 \sim_a vh'e_2.$$

Uniform-No-Miracles means that if an agent cannot distinguish between a history terminating with e_1 and a history whose last event is e_2, then at any time if he is unable to distinguish between two histories wh and vh' then he is still unable to distinguish between whe_1 and vhe_2. This property characterizes local ‘updaters’ that do not take into account the whole history but that proceed in a step-by-step manner.

Propositional Stability *$\mathcal{H}$ satisfies propositional stability iff for all $wh, whe \in H$ we have $p \in V(whe)$ iff $p \in V(wh)$.*

The following result says that the iterated product update of an epistemic model $\mathcal{M}$ according to a protocol P generates an epistemic temporal forest that validates the above-mentioned epistemic properties.

Theorem 5.2.8 (Van Benthem et al. 2009)**.** *An ETL-model $\mathcal{H}$ is isomorphic to the forest generated by the sequential product update of an epistemic model according to some state-dependent DEL-protocol iff it satisfies perfect recall, synchronicity, uniform-no-miracles and propositional stability.*

5.2.4 Learning in a Temporal Perspective

Let us now see how the above construction can be used to analyze learning scenarios.

Learning Event Models In our learning setting the incoming information has a purely propositional character. A simple *event learning model* can be obviously associated with every such piece of data in the following way.

Definition 5.2.9. *Let $\mathcal{C} = \{S_1, S_2, \ldots\}$ be a class of sets and, as before, $U = \bigcup \mathcal{C}$ is the universal set of $\mathcal{C}$. Let $\mathtt{E} : \mathbb{N} \to \mathbb{E}$ be a function that transforms an integer into an event model in the following way: for each $n \in \mathbb{N}$, $\mathtt{E}(n) = \mathcal{E}_n = \langle \{e\}, \sim^{\mathcal{E}_n}, \mathtt{pre}_{\mathcal{E}} \rangle$, where $\sim = \{(e, e)\}$ and $\mathtt{pre}_{\mathcal{E}}(e) = p_n$. Similarly, if $S \subseteq \mathbb{N}$, $\mathtt{E}(S) = \{\mathtt{E}(n) \mid n \in S\}$.*

In other words, for every piece of data n from U we take a propositional letter p_n. Then for each p_n we take a simple public announcement event model. By making the conceptual transition from the simple propositional update to the event models we want to show that our framework conforms to the general setting described in the previous section.

Local Set-Learning Protocol Intuitively, given a state $S_i \in W_{\mathcal{C}}$, our protocol P should authorize at S_i any ω-sequence that enumerates S_i and nothing more. Our set-learning scenarios allow any enumeration of elements of a given set. Therefore, the corresponding local protocol can be defined in the following way.

Definition 5.2.10. *Let $\mathcal{C} = \{S_1, S_2, \ldots\}$ be a class of sets and $U = \bigcup \mathcal{C}$ be the universal set of $\mathcal{C}$. For every $S_i \in W_{\mathcal{C}}$, the* set-learning local protocol, *$P(S_i)$, is the smallest subset of $(\mathtt{E}(U))^\omega$ that contains:*

$$\{f : \omega \to \mathtt{E}(S_i) \mid f \textit{ is surjective}\},$$

and that is closed under non-empty finite prefixes.

Set-learning local protocols restrict the admissible sequences of events only in terms of content and not in terms of ordering. It is easy to observe that such a local protocol can replace the sets in learning scenarios. In principle we can then skip the precondition check and instead decide whether an event can take place just on the basis of the protocols. We will return to this issue in the end of this chapter.

To sum up we will now complement our definition of the initial learning model (Definition 5.1.3) with the local set-learning protocol.

Definition 5.2.11. *Let $\mathcal{C} = \{S_1, S_2, \ldots\}$ be a class of sets such that for all $i \in \mathbb{N}$, $S_i \subseteq \mathbb{N}$. The* initial learning model with local protocol *consists of:*

1. *an epistemic model* $\mathcal{M}_\mathcal{C} = \langle W_\mathcal{C}, \sim, V_\mathcal{C}\rangle$*, where* $W_\mathcal{C} = \mathcal{C}$*,* $\sim\, = W_\mathcal{C} \times W_\mathcal{C}$*,* $V_\mathcal{C} : \textsc{Prop} \cup \textsc{nom} \to \mathcal{P}(W_\mathcal{C})$*, such that* $S_i \in V_\mathcal{C}(p_n)$ *iff* $n \in S_i$ *and for each set* $S_i \in \mathcal{C}$*, we take a nominal* i *and we set* $V_\mathcal{C}(i) = \{S_i\}$*.*

2. *for each* $S_i \in W_\mathcal{C}$*, a set-learning local protocol* $P(S_i)$*.*

Now we are ready to define how our initial learning model and the local set-learning protocol generate an epistemic temporal forest. We define the additional set of designated propositional letters based on the previously used set of nominals $\textsc{nom}$, $\textsc{Prop}_{\textsc{nom}} := \{q_i \mid i \in \textsc{nom}\}$, and we assume that $\textsc{Prop}_{\textsc{nom}} \subseteq \textsc{Prop}$.

Definition 5.2.12 (Epistemic Temporal Learning Forest)**.** *A learning model* $\mathcal{M}_\mathcal{C} = \langle W_\mathcal{C}, \sim^\mathcal{M}, V^\mathcal{M}_\mathcal{C}\rangle$ *together with the local set-learning protocol* $P : W \to$ Prot($\mathbb{E}$) *generates an ETL forest* $\texttt{For}(\mathcal{M}, P)$ *of the form:*

$$\mathcal{H} = \langle W^\mathcal{H}, \mathbb{E}, H, \sim, V\rangle, \text{ where:}$$

1. $W^\mathcal{H} := W_\mathcal{C}$*,*

2. H *is defined inductively as follows:*

   $H_0 := W^\mathcal{H}$;
   $H_{n+1} := \{(we_1 \ldots e_{n+1}) \mid (we_1 \ldots e_n) \in H_n;\ \mathcal{M}_C^{P,\varepsilon\restriction n}, w \models \texttt{pre}(e_{n+1})$ *and* $(e_1 \ldots e_{n+1}) \in P(w)\}$;
   $H := \bigcup_{0 \le k < \omega} H_k$;

3. *If* $w, v \in W^\mathcal{H}$*, then* $w \sim v$ *iff* $w \sim^\mathcal{M} v$*;*

4. $whe \sim_a vh'e'$ *iff* $whe, vh'e' \in H_k$*,* $wh \sim vh'$*, and* $e = e'$*;*

5. *Finally, the valuation* $V : \textsc{Prop} \cup \textsc{Prop}_{\textsc{nom}} \to \mathcal{P}(H)$ *is defined in the following way:*

   - *for every* $p \in \textsc{Prop}$*,* $wh \in V(p)$ *iff* $w \in V^\mathcal{M}_\mathcal{C}(p)$*;*
   - *for every* $q_i \in \textsc{Prop}_{\textsc{nom}}$*,* $wh \in V(q_i)$ *iff* $w \in V^\mathcal{M}_\mathcal{C}(i)$*.*

The above construction is in strict correspondence with the general case of generated epistemic temporal forest of Definition 5.2.6. Our concept allows a slight simplification in point 4 because of the very simple structure of our public announcement events.

At this point we have the temporal structures that correspond to the learning situation. The next step is to give a temporal characterization of forests that satisfy the identifiability condition.

5.2.5 Finite Identifiability in ETL

In this section we will give a general characterization of finite identification in the language of an epistemic temporal logic (see Emerson & Halpern, 1986; Fagin et al., 1995; Parikh & Ramanujam, 2003). The aim of this section is to give a formula of epistemic temporal logic that characterizes learnable classes of sets.

Epistemic Temporal Language

Syntax The syntax of our epistemic temporal language $\mathcal{L}_{\mathrm{ETL}^*}$ is defined in the following way.

$$\varphi := p \mid \neg\varphi \mid \varphi \vee \varphi \mid K\varphi \mid F\varphi \mid A\varphi$$

where p ranges over a countable set of proposition letters $\textsc{Prop}$. $K\varphi$ reads: 'the agent knows that φ'. Symbol F stands for future, and we define G to mean $\neg F \neg$. $A\varphi$ means: 'in all infinite continuations conforming to the protocol, φ holds'.

Semantics $\mathcal{L}_{\mathrm{ETL}^*}$ is interpreted over epistemic temporal frames, $\mathcal{H}$, and pairs of the form (ε, h), the former being a maximal, infinite history in our trees, and the latter a finite prefix of ε (see Van der Meyden & Wong, 2003; Parikh & Ramanujam, 2003).

Definition 5.2.13. *We give the semantics of $\mathcal{L}_{\mathrm{ETL}^*}$. We skip the boolean clauses. We take $h \sqsubseteq h'$ to mean that h is an initial segment of h', and $p \in \textsc{Prop}$.*

$$\begin{array}{lll}
\mathcal{H}, \varepsilon, wh \models p & \text{iff} & wh \in V(p) \\
\mathcal{H}, \varepsilon, wh \models K\varphi & \text{iff} & \textit{for all } \varepsilon',\ vh' \textit{ if } vh' \in K[wh] \textit{ then } \mathcal{H}, \varepsilon', vh' \models \varphi \\
\mathcal{H}, \varepsilon, wh \models F\varphi & \text{iff} & \textit{there is } \sigma \in \Sigma^* \text{ s.t. } wh' = wh\sigma \text{ and } \mathcal{H}, \varepsilon, wh' \models \varphi \\
\mathcal{H}, \varepsilon, wh \models A\varphi & \text{iff} & \textit{for all } \varepsilon' \in P(w) \text{ such that } wh \sqsubset \varepsilon' \text{ we have } \mathcal{H}, \varepsilon', wh \models \varphi
\end{array}$$

The modality 'A' refers to the particular infinite sequences that belong to the chosen protocol associated to w. It can be viewed as an operator that performs a global update on the overall temporal structure, 'accepting' only those infinite histories that conform to the protocol.

To give a temporal characterization of finite identifiability we need to express the following idea. In our epistemic temporal forest, for any starting, bottom node S_i it is the case that for all branches in the future there will be a point after which the agent will know that he started in S_i, which means that he will remain certain about the partition of the tree he is in. The designated propositional letters from $\textsc{Prop}_{\textsc{nom}}$ correspond to the partitions, which can also be viewed as underlying theories that allow predicting further events.[4] Formally, with respect to finite identifiability of sets, the following theorem holds.

[4] The characterization involving designated propositional letters can be replaced with one that uses nominals as markers of bottom nodes. For such an approach see Dégremont & Gierasimczuk, 2009.

Theorem 5.2.14. *The following are equivalent:*

1. *$\mathcal{C}$ is finitely identifiable.*

2. *For all $S_i \in W_{\mathcal{C}}$ and $\varepsilon \in P(S_i)$ the learner's knowledge about the initial state stabilizes to S_i on ε in the generated forest $\texttt{For}(\mathcal{M}_{\mathcal{C}}, P)$.*

3. *$\texttt{For}(\mathcal{M}_{\mathcal{C}}, P) \models q_i \to AFGKq_i$.*

Proof. (1 ⇔ 2) This equivalence restates the earlier result (Theorem 5.1.7) in terms of epistemic temporal forests.

(2 ⇔ 3) Let us first observe that in our generated epistemic temporal forest $\texttt{For}(\mathcal{M}_{\mathcal{C}}, P)$ the following holds:

$$S_i h \sim S_j h' \text{ iff } S_i \sim S_j \text{ and } h = h'. \tag{5.1}$$

Now let us analyze the structure of Clause 3. $\texttt{For}(\mathcal{M}_{\mathcal{C}}, P) \models q_i$ stands for a choice of the partition of the forest, and hence, implicitly, for the initial node S_i; then, the temporal prefix AF stands for: 'on every infinite continuation of S_i consistent with the protocol there is a point'. Hence, it expresses that for all ε for S_i there is a special finite point, a point in which the epistemic temporal fragment of the formula GKq_i holds. Following this observation, to conclude the proof it suffices to show the following proposition:

Proposition 5.2.15. *Let $S_i \in W^{\mathcal{H}}$ and $S_i h \in H$. The following are equivalent:*

1. *For all $\sigma \in \Sigma^*$, such that $S_i h\sigma \in H$, $\mathcal{K}[S_i h\sigma] = \{S_i h\sigma\}$;*

2. *$\texttt{For}(\mathcal{M}_{\mathcal{C}}, P), S_i h \models GKq_i$.*

(1 ⇒ 2) Assume that $\mathcal{K}[S_i h] = \{S_i h\}$. By the definition of the valuation V, we get that $\texttt{For}(\mathcal{M}_{\mathcal{C}}, P), S_i h \models q_i$. Then, by the assumption and the semantics of K, we get that $\texttt{For}(\mathcal{M}_{\mathcal{C}}, P), S_i h \models Kq_i$. Finally, since $\texttt{For}(\mathcal{M}_{\mathcal{C}}, P)$ satisfies Perfect Recall and by the definition of protocol P, we get that $\texttt{For}(\mathcal{M}_{\mathcal{C}}, P), S_i h \models GKq_i$.

(2 ⇒ 1) Now, assume that $\texttt{For}(\mathcal{M}_{\mathcal{C}}, P), S_i h \models GKq_i$. Then, by the semantics of K and by (5.1) we get that for all $\sigma \in \Sigma^*$, such that $S_i h\sigma \in H$, $\mathcal{K}[S_i h\sigma] = \{S_i h\sigma\}$. □

In Chapter 4 we mentioned the adequacy of epistemic models and update with respect to the modeling of finite identification. However, we have been mostly concerned with identification in the limit. In the next section we will explore the language of a temporal logic that can express the condition of identifiability in the limit.

5.2.6 Identification in the Limit and DETL

In order to give a temporal characterization of identifiability in the limit we need to be able to express beliefs of the learner. Therefore, our temporal forests should include a plausibility ordering. In Chapter 4 we have shown that conditioning (update) is a universal learning method from truthful data. In other words, in the case of identifiability in the limit, eliminating the worlds of an epistemic plausibility model is enough to reach stable and true belief. This allows considering very specific temporal structures that result from updating a doxastic epistemic model with purely propositional information.[5]

Definition 5.2.16. An epistemic plausibility temporal forest $\mathcal{H}$ *is a tuple:*

$$\langle W, \Sigma, H, (\sim_a)_{a\in\mathcal{A}}, (\leq_a)_{a\in\mathcal{A}}, V\rangle,$$

where $W \neq \emptyset$ is a countable set of initial states; Σ is a countable set of events; $H \subseteq W\Sigma^*$ is a set of histories (sequences of events starting at states from W) closed under non-empty finite prefixes; for each $a \in A$, $\sim_a \subseteq H \times H$ is an equivalence relation, $\leq_a \subseteq H \times H$ is a plausibility preorder; and $V : \textsc{Prop} \to \mathcal{P}(H)$ is a valuation. We write wh to denote some finite history starting in the state w.

Doxastic Epistemic Temporal Language

Syntax Our doxastic epistemic temporal language of $\mathcal{L}_{\mathrm{DETL}^*}$ is defined by the following inductive syntax.

$$\varphi := p \mid \neg\varphi \mid \varphi \vee \varphi \mid K\varphi \mid B\varphi \mid F\varphi \mid A\varphi$$

where p ranges over a countable set of proposition letters $\textsc{Prop}$. $K\varphi$ reads: 'the agent knows that φ', and $B\varphi$: 'the agent believes that φ'. Symbol F stands for future, G is defined as $\neg F\neg$. $A\varphi$ means: 'in all continuations φ'.

$\mathcal{L}_{\mathrm{DETL}^*}$ is interpreted over epistemic plausibility temporal forests, its semantics is for the most part the same as $\mathcal{L}_{\mathrm{ETL}^*}$. Below we give the semantics of the missing clause, the belief operator B.

Definition 5.2.17.

$$\mathcal{H}, wh \models B\varphi \quad \text{iff} \quad \textit{for all } vh',\ \textit{if } vh' \in \min\nolimits_{\leq} \mathcal{K}[wh],\ \textit{then } \mathcal{H}, vh' \models \varphi$$

We again start with an initial learning epistemic model that corresponds to a class of sets and a local set-learning protocol. This time we want to add a plausibility ordering to generate an epistemic plausibility temporal forest. The construction is defined in the following way:

[5] For more complex actions performed on plausibility models in the context of the comparison between dynamic doxastic and doxastic temporal logic see Van Benthem & Dégremont, 2010.

Definition 5.2.18 (Learning Forest). *A learning model* $\mathcal{M}_{\mathcal{C}} = \langle W_{\mathcal{C}}, \sim^{\mathcal{M}}, V_{\mathcal{C}}^{\mathcal{M}} \rangle$ *together with the local set-learning protocol* $P : W \to \text{Prot}(\mathbb{E})$ *and a plausibility preorder* $\leq^{\mathcal{M}} \subseteq W_{\mathcal{C}} \times W_{\mathcal{C}}$ *generates an DETL forest* $\texttt{For}(\mathcal{M}, P, \leq)$ *of the form:*

$$\mathcal{H} = \langle W^{\mathcal{H}}, \mathbb{E}, H, \sim, \leq, V \rangle, \text{ where:}$$

1. $W^{\mathcal{H}}$, $\mathbb{E}$, H, $\sim$ *and* V *are defined as in the generated epistemic temporal forest, Definition 5.2.12;*
2. *If* $w, v \in W^{\mathcal{H}}$ *and* $wh, vh' \in H$, *then* $wh \leq vh'$ *iff* $wh \sim vh'$ *and* $w \leq^{\mathcal{M}} v$.

As in the case of finite identifiability we will now provide a formula of doxastic epistemic temporal logic that characterizes identifiability in the limit.

Theorem 5.2.19. *The following are equivalent:*

1. $\mathcal{C}$ *is identifiable in the limit.*
2. *There exists a plausibility preorder* $\leq \subseteq W_{\mathcal{C}} \times W_{\mathcal{C}}$ *such that for all* $S_i \in W_{\mathcal{C}}$ *and* $\varepsilon \in P(S_i)$ *the learner's belief about the initial state stabilizes to* S_i *on* ε *in the generated forest* $\texttt{For}(\mathcal{M}_{\mathcal{C}}, P, \leq)$.
3. *There exists a plausibility preorder* $\leq \subseteq W_{\mathcal{C}} \times W_{\mathcal{C}}$ *such that* $\texttt{For}(\mathcal{M}_{\mathcal{C}}, P, \leq) \models q_i \to AFGBq_i$.

Proof. ($1 \Leftrightarrow 2$) This equivalence follows from the existence of an appropriate preorder, defined in Section 4.5, and its adaptation to the notion of epistemic temporal forest.

($2 \Leftrightarrow 3$) The proof has a strategy similar to the proof of Theorem 5.2.14. This time the crucial observation is that in $\texttt{For}(\mathcal{M}_{\mathcal{C}}, P, \leq)$, $S_i h \leq S_j h'$ iff $S_i \leq S_j$ and $h = h'$. □

Let us observe that the last clauses of Theorems 5.2.14 and 5.2.19 can be strengthened to exclude the condition of persistence of the doxastic-epistemic states. In our setting, once such a state is reached, it cannot disappear. In the above characterizations this can be reflected by dropping the temporal operator G.

The above theorems give simple syntactic temporal characterizations of finite and limiting learning in doxastic epistemic temporal logic. We do not provide any proof theory for these notions, any 'logic of learning'. However, we do perceive this as an interesting direction for future work. Moreover, we are especially interested in giving temporal characterizations of various learning-theoretic facts, e.g., the existence of tell-tale sets or the locking-sequence lemma (see Chapter 4). Further questions concern modifications of our syntactic temporal characterization and observing what notions of learning can be obtained this way.

5.2.7 Further Questions on Protocols

Uniform-No-Miracles states that any two histories that are not distinguishable from an agent's perspective cannot get distinguished by extending them with the same event (or two indistinguishable event states). In our learnability context a strengthening of this rule seems interesting.

Let us consider the problem of identification in a more general perspective. Objects to be learned do not have to be sets, in particular their protocols do not have to be order-independent. Except for sets, formal learning theory is also concerned, for example, with learnability of functions (see Section 2.1.3). Possible realities can even be more general, they can be classes of functions (scenarios of this kind are at the heart of many inductive inference games, as the card game *Eleusis*, see, e.g., Romesburg, 1978). Then the worlds can be identified with protocols that allow certain sequences of events that can be defined by some logical formula. In particular, events might be assumed to occur in a certain order. Let us consider the following example.

Example 5.2.20. *Let us take two possible worlds: w_1 and w_2 such that:*

1. *the protocol for w_1 allows all infinite sequences that contain all even numbers, and additionally require that whenever a number is 8 then the successor should be 10;*
2. *the protocol for w_2 allows all infinite sequences that contain all even numbers, and additionally require that whenever a number is 8 then the successor should be 6.*

As long as the learner receives even numbers different than 10 he cannot distinguish between the two states, e.g., the two sequences, h, h', are in both protocols:

- $h : 2, 4, 6, 8$
- $h' : 4, 2, 6, 8$

Therefore, we can say that whichever of the two is enumerated, $w_1 \sim w_2$. However, complementing both of them with the same event, 10, leads to 'a miracle'—two hypotheses get to be distinguished.

In principle, there is no reason why such 'miraculous' classes of hypotheses should be excluded from learnability considerations. Such cases show a strength of the protocol based temporal approach over the one-step simple DEL update. The latter is well-suited for set learning, because set-learning protocols are permutation closed and in this sense they are reducible to the precondition check. This is why we turned to a more liberal setting of epistemic temporal logic in which the 'miracle' of order-dependence is possible. What we observed is that with a protocol we can obtain not only factual, but also genuine 'procedural information' in the model. Therefore, sometimes we can distinguish between hypotheses not because a new fact comes in, but because of *the way in which* it comes in.

In general, thinking about learnability in terms of protocols leads to a setting in which the possible realities are identified with sets of scenarios of what should be expected to happen in the future. In this sense, the most general realities are sets—they allow any possible enumeration of their content. Functions allow only one particular sequence of events. In between there are a variety of possibilities for defining protocols that can be characterized in an arbitrary way. In general, our results in the previous sections are only the beginning of the logical study of the richness of possible learning protocols.

5.3 Conclusions and Perspectives

Our work provides a translation of scenarios from formal learning theory into the domain of dynamic epistemic logic and epistemic temporal logic. In particular, we characterized the process of identification in the syntax of dynamic doxastic epistemic logic. Moreover, in the more general context of learnability of protocols, we characterized learning in the syntax of a doxastic epistemic temporal language. Hence, we showed that the proposal of expressing learnability in languages of modal-temporal logics of knowledge and belief (see Van Benthem, 2010) can be made precise.

Our results again show that the two prominent approaches, learning theory and epistemic modal-temporal logics, can be joined together in order to describe the notions of belief and knowledge involved in inductive inference. We believe that bridging the two approaches benefits both sides. For formal learning theory, to create a logic for it is to provide additional syntactic insight into the process of inductive learning. For logics of epistemic and doxastic change, it enriches their present scope with different learning scenarios, i.e., not only those based on the incorporation of new data but also on generalization.

Moreover, as we indicated in the last section of this chapter the temporal logic based approach to inductive inference gives a straightforward framework for analyzing various domains of learning on a common ground. In terms of protocols, sets can be seen as classes of specific histories—their permutation-closed complete enumerations. Functions, on the other hand, can be seen as 'realities' that allow only one particular infinite sequence of events. We can think of many intermediate concepts that may be the object of learning. Interestingly, the identification of protocols, that seems to be a generalization of the set-learning paradigm provides what has been the original motivation for epistemic temporal logic from the start: identifying the current history that the agent is in, including its order of events, repetitions, and other constraints.

Further directions include extending our approach to other types of identification, e.g., identification of functions; finding a modal framework for learning from both positive and negative information; studying systematically the effects of different restrictions on protocols. We are also interested in investigating various constraints one can enforce on learning functions (e.g., consistency, conservatism or set-drivenness) and comparing them to those of epistemic and doxastic agents in doxastic epistemic temporal logics. Modal logics of belief change are a natural framework to study a variety of notions that underly such concepts of learnability. Another important restriction on learning functions is computability. In the next chapter we will be concerned with computable learning functions in the case of finite identification—the convergence to irrevocable knowledge.

Part III

Learning and Computation

Chapter 6

On the Complexity of Conclusive Update

To finitely identify a language means to be able to recognize it with certainty after receiving some (specific) finite sample of the language. Such a finite sample that suffices for finite identification is called *definite finite tell-tale set* (DFTT, for short, see Lange & Zeugmann, 1992; Mukouchi, 1992). One can interpret such a DFTT as the collection of the most characteristic (from a certain point of view) elements of the set. It has also a different connotation that is based on the *eliminative power* of its elements. We can think of the information that is carried by a particular sample of the language in a negative way, as showing which of the hypotheses are inconsistent with the information that has arrived, and thereby eliminating them. A set S is finitely identifiable if its finite subset has the power of eliminating all possibilities under consideration which are different from S.

From the characterization of finite identifiability (Mukouchi, 1992), we know that if a class of languages is finitely identifiable, then the identification can be done on the basis of corresponding DFTTs, i.e., finite subsets of the original languages that contain a sample sufficient for finite identifiability. We will call a learner that explicitly uses some DFTTs in the process of identification a *preset learner*. The name derives from the fact that such a learner is equipped with extra information about the DFTTs prior to the identification. A number of issues emerge when analyzing computational properties of the definite finite tell-tales used in identification. Since DFTTs are by no means unique, it can obviously be useful to obtain small definite tell-tales. In this context we distinguish two notions of minimality for DFTTs. A *minimal* DFTT is a DFTT that cannot be further reduced without losing its eliminative power with respect to a class of languages. A *minimal-size* DFTT of a set S, is a DFTT that is one of those which are smallest among all possible DFTTs of S. In order to investigate the computational complexity of finding such minimal DFTTs, we will have to restrict ourselves to finite classes of languages. Even though it is a very heavy restriction, it creates the possibility of grasping important aspects of the complexity of finite identification. We will next move back to more general cases and investigate how the use of the class of all (minimal) DFTTs can influence the speed of finite identification.

In the previous chapters we linked the notion of finite identification with the convergence to irrevocable knowledge. The idea of eliminating possibilities that are inconsistent with the incoming data is essentially the same as in the concept of *update* in dynamic epistemic logic (see, e.g., Van Ditmarsch et al., 2007). Presently we discuss, given the epistemic state S, the computational complexity of:

1. deciding whether convergence to irrevocable knowledge via update is possible (whether S is finitely identifiable);

2. given that the class is finitely identifiable, finding minimal samples that allow eliminating uncertainty (finding minimal DFTTs);

3. given that the class is finitely identifiable, finding minimal-size samples that allow eliminating uncertainty (finding minimal-size DFTTs).

We argue that the investigations into the complexity of finite identification give a new perspective on the complexity of the emergence of the resulting state, the state of full certainty, that corresponds to the K operator in S5 systems of epistemic logic (see Chapter 2).

The computational tasks can also be interpreted as a motivation for explicitly introducing a new actor, a teacher. Her role is to decide whether learning (given a certain learnability condition) can be successful, and if it can be, to find and provide to the learner minimal samples that will lead to the emergence of knowledge. The analysis of complexity of those tasks assumes a number of conditions on the teacher and the learner: helpfulness of the teacher and eagerness to learn of the learner. Those are not controversial, however they constitute only one of many possible learning and teaching attitudes (other profiles are discussed, in a different setting, in Chapter 7).

The plan of this chapter is as follows. We first recall some relevant learnability notions and the definition and characterization of finite identifiability. We will introduce the notion of *preset learner* that performs identification on the basis of some DFTTs, and characterize the notion using the concept of *subset-driven* learning. Then we will ask the question of how difficult it is to find DFTTs of various kinds. We present the refined notions of *minimal* DFTT, and *minimal-size* DFTT. We show that the problem of finding a minimal-size DFTT is NP-complete, while the problem of finding any minimal DFTT is PTIME computable. Therefore, it can be argued that it is harder for a teacher to provide a minimal-size optimal sample, than just any minimal sufficient information. Then we analyze the possibility of a recursive function that explicitly provides all minimal DFTTs of a finite language. We call the type of finite identification that requires the existence of such a function *strict preset finite identification.* For the case of finite classes of finite languages we apply a computational complexity analysis—here finding the set of all minimal DFTTs turns out to be NP-hard. In the more general case of infinite classes of finite languages, we also show that there are recursively finitely identifiable classes which are not recursively strict preset finitely identifiable. In the end we compare finite identification with the concept of fastest finite identification and show classes for which recursive finite identifiers exist, but which cannot be recursively finitely identified in the fastest way. That is so because for those classes no recursive function exists that gives access to all minimal DFTTs for each language in the class.

6.1 Basic Definitions and Characterization

Let $U \subseteq \mathbb{N}$ be an infinite recursive set, we call any $S \subseteq U$ a language. In the general case, we will be interested in any class of languages that forms an indexed family of recursive languages, i.e., a class $\mathcal{C}$ for which a computable function $f : \mathbb{N} \times U \to \{0, 1\}$ exists that uniformly decides $\mathcal{C}$, i.e.:

$$f(i, w) = \begin{cases} 1 & \text{if } w \in S_i, \\ 0 & \text{if } w \notin S_i. \end{cases}$$

In large parts of this chapter we will also consider $\mathcal{C}$ to be $\{S_1, S_2, \ldots, S_n\}$, a finite class of finite sets, in which case we will use $I_{\mathcal{C}}$ for the set containing indices of sets in $\mathcal{C}$, i.e., $I_{\mathcal{C}} = \{1, \ldots, n\}$.

The notation and basic definition are as introduced in Chapter 2. We recall those that are most important for the content of the present chapter.

Finite identifiability of a class of languages from positive data is defined by the following chain of conditions.

Definition 6.1.1. *A learning function L:*

1. *finitely identifies $S_i \in \mathcal{C}$ on ε iff, when inductively given ε, at some point L gives a single output i;*

2. *finitely identifies $S_i \in \mathcal{C}$ iff it finitely identifies S_i on every ε for S_i;*

3. *finitely identifies $\mathcal{C}$ iff it finitely identifies every $S_i \in \mathcal{C}$.*

A class $\mathcal{C}$ is finitely identifiable iff there is a learning function L that finitely identifies $\mathcal{C}$.

The correspondence between the learning-theoretical setting and the epistemic framework is set to be as in Chapter 4. Namely we take $U :=$ Prop an infinite, countable set of propositions, we call any $s \subseteq$ Prop a possible world. A set of possible worlds $S = \{s_1, s_2, \ldots\}$ is an epistemic state. Throughout a large part of this chapter the epistemic states are taken to be finite. Otherwise we assume them to be indexed families of recursive sets of propositions. Accordingly, the *text (positive presentation)* ε of s_i is a sound and complete infinite sequence of propositions from Prop allowing repetitions, that are satisfied in s_i. For simplicity we will continue here with the number-theoretical framework, but we would like the reader to bear in mind that the epistemic interpretation of these results is straightforward.

Let us recall the necessary and sufficient condition for finite identifiability (Lange & Zeugmann, 1992; Mukouchi, 1992). It involves a modified, stronger notion of finite tell-tale (Angluin, 1980), namely the *definite finite tell-tale set*.

Definition 6.1.2 (Mukouchi 1992)**.** *A set D_i is a definite finite tell-tale set for $S_i \in \mathcal{C}$ if*

1. $D_i \subseteq S_i$,
2. D_i *is finite, and*
3. *for any index j, if $D_i \subseteq S_j$ then $S_i = S_j$.*

Finite identifiability can be then characterized in the following way.

Theorem 6.1.3 (Mukouchi 1992)**.** *A class $\mathcal{C}$ is finitely identifiable from positive data iff there is an effective procedure $\mathcal{D} : \mathbb{N} \to \mathcal{P}^{<\omega}(\mathbb{N})$, given by $n \mapsto \mathcal{D}_n$, that on input i produces a definite finite tell-tale of S_i.*

In other words, each set in a finitely identifiable class contains a finite subset that distinguishes it from all other sets in the class. Moreover, for the effective identification it is required that there is a *recursive* procedure that provides such DFTTs.

Let us first observe that if a language S_i contains a DFTT, then every text for S_i enumerates all elements of this DFTT in finite time.

Proposition 6.1.4. *If ε is a text for $S_i \in \mathcal{C}$ and S is a finite subset of S_i (in particular a DFTT of S_i), then there is an $n \in \mathbb{N}$, such that* set$(\varepsilon\restriction n)$ *is a superset of S.*

Proof. Let us take a finite $S \subseteq S_i$, and ε—a text for S_i. Assume for contradiction there is no $n \in \mathbb{N}$ such that set$(\varepsilon\restriction n)$ is a superset of S. Then that means that there is $k \in \mathbb{N}$ such that $k \in S \subseteq S_i$ and for all n, $\varepsilon_n \neq k$. This contradicts the definition of text. □

Theorem 6.1.3 gives the characterization of finite identification in terms of a recursive procedure that *generates* DFTTs. Below we present a new way of tell-tale sets being given—by a decision procedure. We will call such a procedure a *dftt-function*.

Definition 6.1.5. *Let $\mathcal{C}$ be an indexed family of recursive sets. The* dftt-function *for $\mathcal{C}$ is a recursive function $f_{\mathsf{dftt}} : \mathcal{P}^{<\omega}(\mathbb{N}) \times \mathbb{N} \to \{0, 1\}$, such that:*

1. *if $f_{dftt}(S, i) = 1$, then S is a DFTT of S_i;*

2. *for every $i \in \mathbb{N}$ there is a finite $S \subseteq \mathbb{N}$, such that $f_{dftt}(S, i) = 1$.*

A first observation about the dftt-function is that it cannot attribute two $i, j \in \mathbb{N}$, such that $i \neq j$ to one finite set S .

Proposition 6.1.6. *Let $\mathcal{C}$ be a class of languages and f_{dftt} be a dftt-function for $\mathcal{C}$. Then there is no finite $S \subseteq \mathbb{N}$ such that for some $i, j \in \mathbb{N}$, such that $i \neq j$ and $f_{dftt}(S, i) = 1$ and $f_{dftt}(S, j) = 1$.*

Proof. Assume that there is a finite $S \subseteq \mathbb{N}$ and $i, j \in \mathbb{N}$ such that $f_{\text{dftt}}(S, j) = 1$ and $f_{\text{dftt}}(S, i) = 1$. Then, by definition of f_{dftt}, S is a DFTT of both S_i and S_j. By the definition of DFTT, $i = j$. □

Now we will show that in fact the condition given in Theorem 6.1.3 is equivalent to the existence of such dftt-function.

Theorem 6.1.7. *A class $\mathcal{C}$ is finitely identifiable from positive data iff there is a dftt-function for $\mathcal{C}$.*

Proof. ($\Rightarrow$) Let us assume that $\mathcal{C}$ is finitely identifiable. Then, by Theorem 6.1.3 there is an effective procedure $\mathcal{D} : \mathbb{N} \to \mathcal{P}^{<\omega}(\mathbb{N})$, given by $n \mapsto \mathcal{D}_n$, that on input i produces all elements of a definite finite tell-tale of S_i. Let $S \subseteq \mathbb{N}$ be a finite set and $i \in \mathbb{N}$. We define $f : \mathcal{P}^{<\omega}(\mathbb{N}) \times \mathbb{N} \to \{0, 1\}$ in the following way:

$$f(S, i) = \begin{cases} 1 & \text{if } \mathcal{D}_i = S; \\ 0 & \textit{otherwise}. \end{cases}$$

Let us observe that f is a dftt-function for $\mathcal{C}$:

1. f is recursive: given S and i the function f uses $\mathcal{D}$ to produce a DFTT of S_i, and then compares the obtained $\mathcal{D}_i$ with S. Such a $\mathcal{D}_i$ always exists by the definition of $\mathcal{D}$;

2. if $f(S, i) = 1$ then S is obviously a DFTT of S_i;

3. for all $i \in \mathbb{N}$ there is a finite $S \subseteq \mathbb{N}$ such that $f(S, i) = 1$, by the definition of $\mathcal{D}$.

($\Leftarrow$) Let us take a class $\mathcal{C}$ and assume that there is a dftt-function for $\mathcal{C}$. Let us take $S_i \in \mathcal{C}$. The standard text ε^{st} for S_i is defined in the following way:

$$\varepsilon_0^{\text{st}} = \mu n(n \in S_i), \text{ and}$$

$$\varepsilon_n^{\text{st}} = \begin{cases} n & \text{if } n \in S_i; \\ \varepsilon_{n-1}^{\text{st}} & \text{otherwise}. \end{cases}$$

We define $\mathcal{D} : \mathbb{N} \to \mathcal{P}^{<\omega}(\mathbb{N})$ in the following way: On input i, $\mathcal{D}$ constructs the standard text for S_i in a step by step manner. At each step n, $\mathcal{D}$ performs a search for a $S \subseteq \mathrm{set}(\varepsilon^{\mathrm{st}}{\restriction}n)$ such that $f_{\mathrm{dftt}}(S, i) = 1$. The first one found in this manner is taken to be $\mathcal{D}_i$. By Definition of f_{dftt} and Proposition 6.1.4, $\mathcal{D}$ is recursive and $\mathcal{D}_i$ is a DFTT of S_i. □

We have shown that DFTTs for a finitely identifiable class can be given in two equivalent ways. It is important to remember that f_{dftt} may not recognize all DFTTs of a given language, but it is guaranteed to 'know about' at least one.

6.2 Preset Learning

Let us now turn to our central notion of *preset learning.* Intuitively, a preset learning function uses a recursive decision function, such as the dftt-function defined in the previous section, as a guide in the process of identification.

Definition 6.2.1. *Let $\mathcal{C} = \{S_i \mid i \in \mathbb{N}\}$ be a class of languages, ε—a text for some $S_i \in \mathcal{C}$, and $f : \mathcal{P}^{<\omega}(\mathbb{N}) \times \mathbb{N} \to \{0, 1\}$ be a recursive function. A* preset learning function L based on f *is defined in the following way:*

$$L(\varepsilon{\restriction}n) = \begin{cases} \mu j \ \ \mathrm{set}(\varepsilon{\restriction}n) \subseteq S_j & \textit{if for that } j\ \exists S \subseteq \mathrm{set}(\varepsilon{\restriction}n)\ f(S, j) = 1 \\ & \&\ \forall k < n\ L(\varepsilon{\restriction}k) = \uparrow; \\ \uparrow & \textit{otherwise.} \end{cases}$$

It is easy to see that on any text for a language in the class such a function has at most one integer value. In general we will call such learning functions (at most) once defined.

Definition 6.2.2. *A learning function L is* (at most) once defined *on $\mathcal{C}$ iff for any text ε for a language from $\mathcal{C}$ and $n, k \in \mathbb{N}$ such that $n \neq k$: $L(\varepsilon{\restriction}n) = \uparrow$ or $L(\varepsilon{\restriction}k) = \uparrow$.*

Proposition 6.2.3. *Every preset learning function is (at most) once defined.*

Proof. Let $\mathcal{C}$ be a class of languages. Assume that L is a preset learning function based on recursive f, and, for contradiction, that L is not (at most) once defined on $\mathcal{C}$. Then, there is ε—a text for some $S_i \in \mathcal{C}$ and $\ell, n \in \mathbb{N}$ such that $\ell \neq n$, $L(\varepsilon{\restriction}\ell) \neq \uparrow$ and $L(\varepsilon{\restriction}n) \neq \uparrow$. Assume that $\ell < n$. Since L is total, there is an i such that $L(\varepsilon{\restriction}n) = i$, and so, by the definition of L, $\exists S \subseteq \mathrm{set}(\varepsilon{\restriction}n)\ f(S, i) = 1$ and $\forall k < n\ L(\varepsilon{\restriction}k) = \uparrow$. The latter gives a contradiction with the assumption that $L(\varepsilon{\restriction}\ell) \neq \uparrow$. □

Moreover, we can show that if for every $i \in \mathbb{N}$, f judges at least one finite $S \subseteq S_i$ positively then the preset learning function based on f is recursive.

Proposition 6.2.4. *Let L be a preset learning function based on f. If f satisfies Condition 2 of Definition 6.1.5, i.e., for all $i \in \mathbb{N}$ there is a finite $S \subseteq \mathbb{N}$ such that $f(S, i) = 1$, then L is recursive on any finitely identifiable class.*

We will now show that preset learners can identify every finitely identifiable class.

Proposition 6.2.5. *If a class $\mathcal{C}$ is finitely identifiable then it is finitely identified by a preset learner.*

Proof. Assume that $\mathcal{C}$ is finitely identifiable, then by the Theorem 6.1.7, there is a dftt-function f_{dftt} for $\mathcal{C}$. We will show that L, the recursive preset learner based on f_{dftt} finitely identifies $\mathcal{C}$. First, let us observe that by Proposition 6.2.4, L is recursive. By Proposition 6.2.3 we have that for any ε text for some $S_i \in C$, L is (at most) once defined. Let us take ε a text for $S_i \in \mathcal{C}$. By the definition of text and DFTT for all $j \neq i$, there is no $n \in \mathbb{N}$ and no S such that $S \subseteq \text{set}(\varepsilon{\restriction}n)$ & $f(S, j) = 1$. By the definition of f_{dftt} we get that $\exists S \subseteq S_i\ f(S, i) = 1$, and by Proposition 6.1.4 there $\exists n, S\ (S \subseteq \text{set}(\varepsilon{\restriction}n)$ & $f(S, i) = 1)$. Take the smallest such n. Then $L(\varepsilon{\restriction}n) = i$. □

We have shown that a preset learning function based on a dftt-function can identify any finitely identifiable class. In the next section we will discuss some further properties of preset learning.

Set-Drivenness and Subset-Drivenness The fact that the preset learner based on a dftt-function is universal with respect to finite identification indicates that for finite learning it is enough to care at each step only about the content of the finite sequence presented so far. In particular, a preset learner only checks whether the sequence includes a subset with certain properties. It does not pay attention to the order of elements and repetitions. Learning functions that work this way are called set-driven.

Definition 6.2.6 (Wexler & Cullicover 1980)**.** *Let $\mathcal{C}$ be an indexed family of recursive sets. A learning function L is said to be* set-driven *with respect to $\mathcal{C}$ iff for any two texts ε_1 and ε_2 for some languages in $\mathcal{C}$ and any two $n, k \in \mathbb{N}$, if* $\text{set}(\varepsilon_1{\restriction}n) = \text{set}(\varepsilon_2{\restriction}k)$, *$L(\varepsilon_1{\restriction}n) \neq \uparrow$ and $L(\varepsilon_2{\restriction}k) \neq \uparrow$ (i.e., they both have a natural number value), then $L(\varepsilon_1{\restriction}n) = L(\varepsilon_2{\restriction}k)$.*

It has been shown that set-drivenness does not restrict the power of finite identification.[1] This is different from the general case of identification in the limit, where set-drivenness does restrict the power of identification.

[1] In their proof of Theorem 6.2.7, Lange & Zeugmann construct a learner very similar to our preset learning function. We would like to thank the anonymous reviewer of *The 23rd Annual Conference of Learning Theory 2010* for pointing us in this direction.

Theorem 6.2.7 (Lange & Zeugmann 1996)**.** *A class $\mathcal{C}$ is finitely identifiable if and only if $\mathcal{C}$ is finitely identified by a set-driven learner.*

We will show that any preset learning function based on a dftt-function is set-driven.

Theorem 6.2.8. *Let $\mathcal{C}$ be a class of languages, and f_{dftt} be a dftt-function for $\mathcal{C}$. If L is a preset learning function based on f_{dftt}, then L is set-driven with respect to $\mathcal{C}$.*

Proof. Take $\mathcal{C}$, f_{dftt} and L as specified in the theorem. Assume that ε_1 and ε_2 are texts for some languages in $\mathcal{C}$; $n, k \in \mathbb{N}$; $\text{set}(\varepsilon_1 \restriction n) = \text{set}(\varepsilon_2 \restriction k)$; $L(\varepsilon_1 \restriction n) \neq \uparrow$ and $L(\varepsilon_2 \restriction k) \neq \uparrow$ (they both have an integer value). Assume that $L(\varepsilon_1 \restriction n) = i$. Then, by the definition of L, $\text{set}(\varepsilon_1 \restriction n) \subseteq S_i$, $\exists S \subseteq \text{set}(\varepsilon_1 \restriction n)\ f(S, i) = 1$ and $\forall \ell < n\ L(\varepsilon_1 \restriction \ell) = \uparrow$. The same holds for $L(\varepsilon_2 \restriction k)$ and some $j \in \mathbb{N}$. We have to show that $i = j$.

Assume for contradiction that $i \neq j$. Then, since $\text{set}(\varepsilon_1 \restriction n) = \text{set}(\varepsilon_2 \restriction k)$, $\exists S \subseteq \text{set}(\varepsilon_1 \restriction n) \subseteq \text{set}(\varepsilon_2 \restriction k)\ f(S, i) = 1$ and $\text{set}(\varepsilon_2 \restriction k) \subseteq S_j$. This means that there is a finite set S that is a DFTT for S_i and at the same time $S \subseteq S_j$ for some $j \neq i$. This gives a contradiction with the definition of DFTT. □

A stronger notion of set-drivenness is possible. Definition 6.2.6 restricts the condition to those situations in which L gives an integer value. The alternative concept is as follows.

Definition 6.2.9 (Wexler & Cullicover 1980)**.** *Let $\mathcal{C}$ be an indexed family of recursive sets. A learning function L is said to be* strongly set-driven *with respect to $\mathcal{C}$ iff for any two texts ε_1 and ε_2 for some languages in $\mathcal{C}$ and any two $n, k \in \mathbb{N}$, if* $\text{set}(\varepsilon_1 \restriction n) = \text{set}(\varepsilon_2 \restriction k)$*, then $L(\varepsilon_1 \restriction n) = L(\varepsilon_2 \restriction k)$.*

The preset learner based on an f_{dftt} for $\mathcal{C}$ is not strongly set-driven with respect to $\mathcal{C}$. Consider the folowing simple example. Let $\mathcal{C} = \{S_1 = \{1, 2, 4\}, S_2 = \{1, 3, 4\}\}$, and let $\varepsilon_1 = \langle 1, 2, 4, \ldots \rangle$ and $\varepsilon_2 = \langle 1, 4, 2 \ldots \rangle$ be two texts for S_1. Let us compare the outputs of the learning function in the two cases of the initial segments of ε_1 and ε_2: $L(\langle 1, 2, 4 \rangle) = \uparrow$ and $L(\langle 1, 4, 2 \rangle) = 1$. The content of the two sequences is the same but the outputs of L are different.

From Theorem 6.2.7 we know that set-drivenness does not restrict finite identifiability. The notion of preset learning leads to a concept of subset-driven learning, that is itself related to set-driven learning.[2]

Definition 6.2.10. *Let $\mathcal{C}$ be an indexed family of recursive sets. A learning function L is* subset-driven *with respect to $\mathcal{C}$ iff for any two texts ε_1 and ε_2 for some languages in $\mathcal{C}$, and any $n, k \in \mathbb{N}$:*

[2] In fact, if one considers (instead of once-defined functions) functions that keep outputting the same value after having given a value once, then the concepts of strongly set-driven and subset-driven coincide. This would also make the anomaly vanish that preset learners are not strongly set-driven.

- *If* $L(\varepsilon_1 \restriction n) = \downarrow$ *and* $\text{set}(\varepsilon_1 \restriction n) \subseteq \text{set}(\varepsilon_2 \restriction k)$ *and for all* $\ell < k$*,* $L(\varepsilon_2 \restriction \ell) = \uparrow$*, then* $L(\varepsilon_1 \restriction n) = L(\varepsilon_2 \restriction k)$*.*

In other words, assume that a subset-driven learning function on an initial segment $\varepsilon \restriction n$ gives an integer answer. Then if there is some other text that at some point enumerates all elements of $\varepsilon \restriction n$, and up to that point no answer was given, then the function is bound to give the same integer answer.

Theorem 6.2.11. *Let $\mathcal{C}$ be a class of languages, and f_{dftt} be a dftt-function for $\mathcal{C}$. If L is a preset learning function based on f_{dftt}, then L is subset-driven with respect to $\mathcal{C}$.*

Proof. Take $\mathcal{C}$, f_{dftt} and L as specified in the theorem. Assume that ε_1 and ε_2 are texts for some languages in $\mathcal{C}$; $L(\varepsilon_1 \restriction n) = \downarrow$ and $\text{set}(\varepsilon_1 \restriction n) \subseteq \text{set}(\varepsilon_2 \restriction k)$ and for all $\ell < k$, $L(\varepsilon_2 \restriction \ell) = \uparrow$. We have to show that then $L(\varepsilon_1 \restriction n) = L(\varepsilon_2 \restriction k)$.

By the fact that $L(\varepsilon_1 \restriction n) = \downarrow$, we know that there is $i \in \mathbb{N}$ such that $\text{set}(\varepsilon_1 \restriction n) \subseteq S_i$ and that $\exists S \subseteq \text{set}(\varepsilon_1 \restriction n)\ f(S, i) = 1$. Then $\exists S \subseteq \text{set}(\varepsilon_1 \restriction n) \subseteq \text{set}(\varepsilon_2 \restriction k)$ such that $f(S, i) = 1$, i.e., $\text{set}(\varepsilon_2 \restriction k)$ includes a finite set S such that S is a DFTT for S_i. Hence, by the definitions of text and DFTT, $\text{set}(\varepsilon_2 \restriction k) \subseteq S_i$. Moreover, one of the assumptions is that for all $\ell < k$, $L(\varepsilon_2 \restriction \ell) = \uparrow$. Therefore, $L(\varepsilon_2 \restriction \ell) = i$. □

The connection between subset-driven finite identifiers and preset learners is even tighter. Every subset-driven learning function that finitely identifies a class is a preset learner (with respect to some f).

Theorem 6.2.12. *Assume $\mathcal{C}$ is a class of languages and $\mathcal{C}$ is finitely identified by a subset-driven learning function L. Then L is a preset learner (with respect to some f).*

Proof. Let subset-driven learning function L finitely identify $\mathcal{C}$. We define f in the following way:

$$f(s, i) = 1 \text{ iff, for some } T \subseteq S,\ L(\hat{T}) = i.$$

We show that L_f, preset learner with respect to f, is equal to L.

Let us take ε a text for some language in $\mathcal{C}$ and take n such that for all $k < n$, $L(\varepsilon \restriction k) = \uparrow$. It is sufficient to show that in this case $L(\varepsilon \restriction n) = L_f(\varepsilon \restriction n)$.

First, assume that $L(\varepsilon \restriction n) = \uparrow$. Then, since L is subset-driven, for no $S \subseteq \text{set}(\varepsilon \restriction n)$, $L(\hat{S}) \neq \uparrow$ (otherwise $L(\varepsilon \restriction n)$ would have the same value). So for all $S \subseteq \text{set}(\varepsilon \restriction n)$ and all i, $f(S, i) = 0$. Hence, $L_f(\varepsilon \restriction n) = \uparrow$.

Next, assume that $L(\varepsilon \restriction n) = i$. Then, because of the set-drivenness of L, $f(\text{set}(\varepsilon \restriction n), i) = 1$ and $L_f(\varepsilon \restriction n) = i$ immediately follows. □

Having established the set- and subset-drivenness of preset finite identifiers, we will now turn to investigating the complexity of finding DFTTs that govern the preset finite identification. Until now we have focused on the availability of any DFTT for each language from a class. Of course, a language can have many different DFTTs. In the next section we will distinguish different types of DFTT and discuss their usefulness for finite identification.

6.3 Eliminative Power and Complexity

Identifiability in the limit (Gold, 1967) of a class of languages guarantees the existence of a reliable strategy that allows for convergence to a correct hypothesis for every language from the class. The exact moment at which a correct hypothesis has been reached is not known and in general can be uncomputable. Things are different for finite identifiability. Here, the learning function is allowed to answer only once. Hence, the conjecture is based on certainty. In other words, the learner must know that the answer she gives is true, because there is no opportunity for a change of mind later.

Knowing that one hypothesis is true means being able to exclude all other possibilities. In this section we define the notion of *eliminative power* of a piece of information, which reflects the informative strength of data with respect to a certain class of sets.

Definition 6.3.1. *Let us take $\mathcal{C}$ an indexed class of recursive languages, and $x \in \bigcup \mathcal{C}$.* The eliminative power of x with respect to $\mathcal{C}$ *is determined by a function $El_{\mathcal{C}} : \bigcup \mathcal{C} \to \mathcal{P}(\mathbb{N})$, such that:*

$$El_{\mathcal{C}}(x) = \{i \,|\, x \notin S_i \;\&\; S_i \in \mathcal{C}\}.$$

Additionally, we will write $El_{\mathcal{C}}(X)$ for $\bigcup_{x \in X} El_{\mathcal{C}}(x)$.

In other words, function $El_{\mathcal{C}}$ takes x and outputs the set of indices of all the sets in $\mathcal{C}$ that are inconsistent with x, and therefore in the light of x they can be "eliminated". We can now characterize finite identifiability in terms of the eliminative power.

Proposition 6.3.2. *A set D_i is a definite tell-tale set of $S_i \in \mathcal{C}$ iff*

1. *$D_i \subseteq S_i$,*
2. *D_i is finite, and*
3. *$El_{\mathcal{C}}(D_i) = \mathbb{N} - \{i\}$.*

Moreover, from Theorem 6.1.3 we know that finite identifiability of an indexed family of recursive languages requires that every set in a class has a DFTT. Formally:

Theorem 6.3.3. *A class $\mathcal{C}$ is finitely identifiable from positive data iff there is an effective procedure $\mathcal{D} : \mathbb{N} \to \mathcal{P}^{<\omega}(\mathbb{N})$, given by $n \mapsto \mathcal{D}_n$, that on input i produces a finite set $D_i \subseteq S_i$, such that*

$$El_{\mathcal{C}}(D_i) = \mathbb{N} - \{i\}.$$

6.3.1 The Complexity of Finite Identifiability Checking

As has already been mentioned in the introduction to this chapter, we aim to analyze the computational complexity of finding DFTTs. In order to do that we restrict ourselves to finite classes of finite sets. One may ask about the purpose of further reduction of sets that are already finite. In fact, if a finite class of finite sets is finitely identifiable, then each element of the class is already its own DFTT. However, finite sets can be much larger than their DFTTs. For example, we can take a class of the following form:

$$\mathcal{C} = \{S_i = \{2i, 2^i \text{ first odd natural numbers}\} \mid i = 1, \ldots, n\}.$$

In this case reduction to the minimal information that suffices for finite identification, $2i$, makes a significant difference in the complexity of the process of learning. The learner can simply disregard all odd numbers and wait for an even number.

Theorem 6.3.4. *Checking whether a finite class of finite sets is finitely identifiable is polynomial with respect to the number of sets in the class and the maximal cardinality of sets in the class.*

Proof. The procedure consists of computing $El_{\mathcal{C}}(x)$ and checking whether for each $S_i \in \mathcal{C}$, $El(S_i) = I_{\mathcal{C}} - \{i\}$, where $I_{\mathcal{C}} = \{i \mid S_i \in \mathcal{C}\}$.

Let us first focus on computing $El_{\mathcal{C}}(x)$ for $x \in \bigcup \mathcal{C}$. We take a class $\mathcal{C}$ and assume that $|\mathcal{C}| = m$, and that the largest set in $\mathcal{C}$ has n elements.

In the first steps we have to obtain $\bigcup \mathcal{C}$. After this, we list for each element of $\bigcup \mathcal{C}$ the indices of the sets to which the element does not belong. In this step we have computed $El_{\mathcal{C}}(x)$ for each $x \in \bigcup \mathcal{C}$. All components of this procedure can clearly be performed in polynomial time with respect to m and n. It remains to check whether for all $S_i \in \mathcal{C}$, $\bigcup_{x \in S_i} El_{\mathcal{C}}(x) = I_{\mathcal{C}} - \{i\}$. This involves essentially only the operation of sum. □

From this analysis we conclude that checking whether a finite class of finite sets is finitely identifiable is a quite easy, polynomial task. Nevertheless, as we saw in the example in the beginning of this section, it can be time consuming if n and m are large numbers.

6.3.2 Minimal Definite Finite Tell-Tale

We are now ready to introduce one of the two nonequivalent notions of minimality of the DFTTs. We will call D_i a minimal DFTT of S_i in $\mathcal{C}$ if and only if all the elements of the sets in D_i are essential for finite identification of S_i in $\mathcal{C}$, i.e., taking an element out of the set D_i will decrease its eliminative power with respect to $\mathcal{C}$, and hence it will no longer be a DFTT. We will observe that a language can have many minimal DFTTs of different cardinalities. This will give us a cause to introduce another notion of minimality—minimal-size DFTT.

Learning functions are bound to be guided by the elements that are presented to them in texts. In order to converge quickly there is no reason for the learner to look especially for a *certain* minimal or minimal-size DFTT, because those might not appear soon enough in the text. However, being able to recognize *all* minimal DFTTs can intuitively guarantee that the right answer occurs as soon as it is objectively possible. If it is not the time of convergence but the memory of the learner that we want to spare, having access to all minimal-size DFTTs is obviously useful. Finding the minimal-size DFTTs can certainly be attributed to an efficient teacher, who looks for an optimal sample that allows identification.

Definition 6.3.5. *Let us take a finitely identifiable indexed family of recursive languages $\mathcal{C}$, and $S_i \in \mathcal{C}$. A* minimal DFTT *of S_i in $\mathcal{C}$ is a $D_i \subseteq S_i$, such that*

1. *D_i is a DFTT for S_i in $\mathcal{C}$, and*

2. *$\forall X \subset D_i$ $El_{\mathcal{C}}(X) \neq I_{\mathcal{C}} - \{i\}$.*

Theorem 6.3.6. *Let $\mathcal{C}$ be a finitely identifiable finite class of finite sets. Finding a minimal DFTT of $S_i \in \mathcal{C}$ can be done in polynomial time.*

Proof. Assume that the class $\mathcal{C}$:

1. is finitely identifiable;

2. is finite;

3. consists only of finite sets.

From the assumptions 1 and 3, we know that for each $S_i \in \mathcal{C}$ a DFTT exists, in fact S_i is its own DFTT.

The following procedure yields a minimal DFTT for each $S_i \in \mathcal{C}$.

We want to find a set $X \subseteq S_i$ such that

$$El(X) = I_{\mathcal{C}} - \{i\}, \text{ but } \forall Y \subset X \ El(Y) \neq I_{\mathcal{C}} - \{i\}.$$

First we set $X := S_i$.

We look for the minimal $x \in X$ such that $El(X - \{x\}) = I_{\mathcal{C}} - \{i\}$. If there is no such element, X is the desired DFTT. If there is such an x, we set $X := X - \{x\}$, and repeat the procedure.

Let $|S_i| = n$, where $|\cdot|$ stands for cardinality. The number of comparisons needed for finding a minimal DFTT of S_i in $\mathcal{C}$ is bounded by n^2. □

Example 6.3.7. *Let us consider the class*

$$\mathcal{C} = \{S_1 = \{1,3,4\}, S_2 = \{2,4,5\}, S_3 = \{1,3,5\}, S_4 = \{4,6\}\}.$$

The procedure of finding minimal DFTTs for sets in $\mathcal{C}$ is as follows.

x	$El_{\mathcal{C}}(x)$
1	$\{2,4\}$
2	$\{1,3,4\}$
3	$\{2,4\}$
4	$\{3\}$
5	$\{1,4\}$
6	$\{1,2,3\}$

Table 6.1: Eliminative power of the elements in $\bigcup\mathcal{C}$ with respect to $\mathcal{C}$

set	a minimal DFTT
$\{1,3,4\}$	$\{3,4\}$
$\{2,4,5\}$	$\{4,5\}$
$\{1,3,5\}$	$\{3,5\}$
$\{4,6\}$	$\{6\}$

Table 6.2: DFTTs of $\mathcal{C}$

1. *We construct a list of the elements from $\bigcup\mathcal{C}$.*

2. *With each element x from $\bigcup\mathcal{C}$ we associate $El_{\mathcal{C}}(x) = \{i \mid x \notin S_i\}$, i.e., the set of indices of sets to which x does not belong (names of sets that are inconsistent with x). Table 6.1 shows the result of the two first steps for $\mathcal{C}$.*

3. *The next step is to find minimal DFTTs for every set in the class $\mathcal{C}$. As an example, let us take the first set $S_1 = \{1,2,3\}$. We order elements of S_1, and take the first element of the ordering. Let it be 1. We compute $El_{\mathcal{C}}(S_1 - \{1\})$, it turns out to be $\{2,3,4\}$. We therefore accept the set $\{3,4\}$ as a smaller DFTT for S_1. Then we try to further reduce the obtained DFTT, by checking the next element in the ordering, let it be 3. $El_{\mathcal{C}}(\{3,4\} - \{3\}) = \{4\} \neq \{2,3,4\}$, so 3 cannot be subtracted without loss of eliminative power. We perform the same check for the last singleton $\{4\}$. It turns out that $\{3,4\}$ cannot further be reduced. We give $\{3,4\}$ as a minimal DFTT of S_1.*[3]

4. *We perform the same procedure for all the sets in $\mathcal{C}$. As a result we get minimal DFTTs for each $S_i \in \mathcal{C}$ presented in Table 6.2.*

[3] Checking only singletons is enough because the eliminative power of sets is defined as the sum of the eliminative power of its elements.

6.3.3 Minimal-Size Definite Finite Tell-Tale

Minimal DFTTs of a language include all information that is enough to exclude other possibilities and involve no redundant data. We can use the notion of eliminative power to construct a procedure for finding minimal-size DFTTs of a finitely identifiable finite class of finite sets $\mathcal{C}$. Minimal-size DFTTs are the minimal DFTTs of smallest cardinality.

We assume that $|\mathcal{C}| = m$. To find a DFTT of minimal size for the set $S_i \in \mathcal{C}$, one has to perform a search through all the subsets of S_i starting from singletons, looking for the first X_i, such that $El(X_i) = I_{\mathcal{C}} - \{i\}$.

DFTTs of minimal size need not be unique. Which one is encountered first depends on the manner of performing the search. Below we describe a possible way of searching for minimal-size DFTTs on the example discussed before.

Example 6.3.8. *Let us consider again the class from Example 6.3.7, namely*

$$\mathcal{C} = \{S_1 = \{1,3,4\}, S_2 = \{2,4,5\}, S_3 = \{1,3,5\}, S_4 = \{4,6\}\}.$$

1. *We construct a list of the elements from $\bigcup\mathcal{C}$.*

2. *With each element x from $\bigcup\mathcal{C}$ we associate $El_{\mathcal{C}}(x) = \{i \mid x \notin S_i\}$, i.e., the set of hypotheses for sets to which x does not belong (names of sets that are inconsistent with x). Table 6.1 presents the result of the two first steps for $\mathcal{C}$.*

3. *The next step is to find minimal-size DFTTs for every set in the class $\mathcal{C}$. As an example, let us take the first set $S_1 = \{1,3,4\}$. We are looking for $X \subseteq S_1$ of minimal size, such that $El_{\mathcal{C}}(X) = I_{\mathcal{C}} - \{1\}$.*

   (a) *We look for $\{x\}$ such that $x \in S_1$ and $El_{\mathcal{C}}(\{x\}) = \{2,3,4\}$. There is no such singleton.*

   (b) *We look for $\{x,y\}$ such that $x, y \in S_1$ and $El_{\mathcal{C}}(\{x\}) = \{2,3,4\}$. There are two such sets: $\{1,4\}$ and $\{3,4\}$.*

4. *We perform the same procedure for all $S_i \in \mathcal{C}$. As a result we get minimal-size DFTTs for each of $\mathcal{C}$, the result is presented in Table 6.3.*

Let us now compare the two resulting reductions of sets from $\mathcal{C}$ (see Table 6.4). The case of S_2 shows that the two procedures give different outcomes.

Running time Let us now analyze and discuss the running time of this procedure. First we need to compute $El_{\mathcal{C}}(x)$ for $\bigcup\mathcal{C}$. From the Theorem 6.3.4 we know that it can be done in polynomial time. Now, let us approximate the number of steps needed to find a minimal-size DFTT of a chosen set $S_i \in \mathcal{C}$. We again assume that $|\mathcal{C}| = m$, and S_i has n elements. In the procedure described above we

set	minimal-size DFTTs
$\{1,3,4\}$	$\{1,4\}$ or $\{3,4\}$
$\{2,4,5\}$	$\{2\}$
$\{1,3,5\}$	$\{1,5\}$ or $\{3,5\}$
$\{4,6\}$	$\{6\}$

Table 6.3: Minimal-size DFTTs of $\mathcal{C}$

set	a minimal DFTT	minimal-size DFTTs
$\{1,3,4\}$	$\{3,4\}$	$\{1,4\}$ or $\{3,4\}$
$\{2,4,5\}$	$\{4,5\}$	$\{2\}$
$\{1,3,5\}$	$\{3,5\}$	$\{1,5\}$ or $\{3,5\}$
$\{4,6\}$	$\{6\}$	$\{6\}$

Table 6.4: A comparison of minimal and minimal-size DFTTs of $\mathcal{C}$

performed a search through, in the worst case, all combinations from 1 to $|S_i|$, to find the right set $X \subseteq S_i$, such that $El_{\mathcal{C}}(X)$ satisfies the condition of eliminating all hypothesis but h_i. So, for each set S_i, the number of comparisons that have to be performed is:

$$n + \frac{n!}{2!(n-2)!} + \frac{n!}{3!(n-3)!} + \ldots + 1 = 2^{n-1}$$

Computational Complexity It is costly to find minimal-size DFTTs. As we have seen above, our procedure leads to a complete search through the large space of all subsets of a language. We call this computational problem the Minimal-size DFTT Problem, and define it formally below. In words, the problem can be phrased as checking whether $S_i \in \mathcal{C}$ has a DFTT of size k or smaller.

Definition 6.3.9 (Minimal-size DFTT Problem)**.**

Instance *A finite class of finite sets $\mathcal{C}$, a set $S_i \in \mathcal{C}$, and a positive integer $k \leq |S_i|$.*

Question *Is there a minimal DFTT $X_i \subseteq S_i$ of size $\leq k$?*

We are going to show that the Minimal-size DFTT Problem is NP-complete by pointing out that it is equivalent to the Minimum Cover Problem, which is know to be NP-complete (Karp, 1972). Let us recall it below.

Definition 6.3.10 (Minimal Cover Problem)**.**

Instance: *Collection P of subsets of a finite set F, positive integer $k \leq |P|$.*

Question: *Does P contain a cover for X of size k or less, i.e., a subset $P' \subseteq P$ with $|P'| \leq k$ such that every element of X belongs to at least one member of P'?*

Theorem 6.3.11. *The* Minimal-size DFTT *Problem is NP-complete.*

Proof. First, let us observe that by Theorem 6.3.3, Minimal-size DFTT Problem is equivalent to the following Problem:

Definition 6.3.12 (Minimal-size DFTT′ Problem)**.**

Instance: *Collection $\{El(x) \mid x \in S_i\}$, positive integer $k \leq |S_i|$.*

Question: *Does $\{El(x) \mid x \in S_i\}$ contain a cover for $I_{\mathcal{C}} - \{i\}$ of size k or less, i.e., a subset $Y_i \subseteq \{El(x) \mid x \in S_i\}$ with $|Y_i| \leq k$ such that every element of $\{El(x) \mid x \in S_i\}$ belongs to at least one member of Y_i?*

It is easy to observe that Minimal-size DFTT′ Problem is a notational variant of Minimum Cover Problem, i.e., we take $F = I_{\mathcal{C}}$, $P = \{El(x) \mid x \in S_i\}$ (and therefore $|P| = |S_i|$), and $X = I_{\mathcal{C}} - \{i\}$. Therefore Minimal-size DFTT′ Problem is NP-complete. Since the Minimal-size DFTT′ Problem is equivalent to the Minimal-size DFTT Problem, we conclude that the Minimal-size DFTT Problem is also NP-complete. □

According the our previous considerations, the Minimal-size DFTT Problem may have to be solved by an optimal ('good') teacher, who is expected to give only relevant information to guarantee fast learning. In this sense our result shows that the task of providing the most useful information for finite identification is NP-complete.

6.4 Preset Learning and Fastest Learning

Let us now return to the concept of the *preset learner*. This concept is based on the intuition that it is easier to identify a complicated, large finite structure or an infinite language solely on the basis of their DFTTs, treating those as finite symptoms of the underlying structure. In particular, the use of minimal DFTTs and their influence on the speed of finite identification gives rise to an interesting set of questions. A very natural one is how DFTTs can be used by such preset learning functions. In this section we introduce the notion of *fastest learner* that finitely identifies a language S_i as soon as objective 'ambiguity' between languages has been lifted. In other words, we will define the extreme case of a finite learner who decides on the right language as soon as *any* DFTT has been enumerated.

Let us again take a finitely identifiable class $\mathcal{C}$, and $S_i \in \mathcal{C}$. Now, consider the collection $\mathbb{D}_i$ of all DFTTs of $S_i \in \mathcal{C}$.

Definition 6.4.1. *Let $\mathcal{C}$ be an indexed family of recursive sets. $\mathcal{C}$ is* finitely identifiable in the fastest way *if and only if there is a learning function L s.t.:*

$$L(\varepsilon\restriction n) = i \quad \textit{iff} \quad \exists D_i^j \in \mathbb{D}_i \; D_i^j \subseteq \text{set}(\varepsilon\restriction n) \;\& \\ \neg\exists D_i^k \in \mathbb{D}_i \; D_i^k \subseteq \text{set}(\varepsilon\restriction n-1).$$

We will call such L a fastest learning function*.*

Intuitively, the fastest learner has to explicitly store all DFTTs of all languages in the given class. Then he makes his conjectures on the basis of the occurrence of the DFTTs in the given text. However, we do not have to provide a set of all DFTTs of all languages explicitly. Rather, we will define them to be accessible via a decision procedure.

Definition 6.4.2. *Let $\mathcal{C}$ be an indexed family of recursive sets. The* complete dftt-function *for $\mathcal{C}$ is a recursive function $f_{c\text{-}dftt} : \mathcal{P}^{<\omega}(\mathbb{N}) \times \mathbb{N} \to \{0,1\}$, such that:*

1. *$f_{c\text{-}dftt}(S,i) = 1$ if and only if S is a DFTT of S_i;*
2. *for every $i \in \mathbb{N}$ there is a finite $S \subseteq \mathbb{N}$, such that $f_{c\text{-}dftt}(S,i) = 1$.*

Theorem 6.4.3. *A class $\mathcal{C}$ is finitely identifiable in the fastest way if and only if there is a complete dftt-function for $\mathcal{C}$.*

Proof. ($\Rightarrow$) Let us take a class $\mathcal{C}$ and assume that it is finitely identifiable in the fastest way, i.e., there is a learning function L that finitely identifies $\mathcal{C}$, and satisfies the condition of Definition 6.4.1. We define $f : \mathcal{P}^{<\omega}(\mathbb{N}) \times \mathbb{N} \to \{0,1\}$ in the following way:

$$f(S,i) = \begin{cases} 1 & \text{if } \exists T \subseteq S \; L(\hat{T}) = i, \\ 0 & \text{otherwise.} \end{cases}$$

First, let us observe that f is recursive because L is recursive and there are only finitely many $T \subseteq S$.

Now we have to show that f is a complete dftt-function for $\mathcal{C}$. Let us observe that for every $i \in \mathbb{N}$ there is a finite $S \subseteq \mathbb{N}$, such that $f(S,i) = 1$. This is so because L finitely identifies $\mathcal{C}$, and so, it finitely identifies an $S_i \in C$ on a text that enumerates S_i in increasing order. It remains to show that:

$$f(S,i) = 1 \text{ iff } \; S \text{ is a DFTT for } S_i.$$

($\Rightarrow$) Assume that $f(S,i) = 1$, then $\exists T \subseteq S \; L(\hat{T}) = i$. By the definition of fastest learner L we have that $\exists D_i^j \in \mathbb{D}_i \; D_i^j \subseteq \text{set}(\hat{T})$, i.e., there is a DFTT of S_i included in T and hence also in S. Since $S \subseteq S_i$, S then has to be a DFTT for S_i as well.

($\Leftarrow$) Assume that S is a DFTT for S_i and, for contradiction, that $f(S, i) = 0$. Then it means that $\forall T \subseteq S \; L(\hat{T}) \neq i$. Take ε any text for S_i and let $\varepsilon' := \hat{S} * \varepsilon$. ε' is clearly a text for S_i, but L is not the fastest learner on ε'. Contradiction.

($\Leftarrow$) Assume that there is a complete dftt-function, $f_{\text{c-dftt}}$, for $\mathcal{C}$. We define L to be the preset learning function based on $f_{\text{c-dftt}}$. Then, by Proposition 6.2.5 L finitely identifies $\mathcal{C}$. We have to show that L is the fastest learner, i.e., for any $S_i \in \mathcal{C}$ and any text ε for S_i:

$$\begin{aligned} L(\varepsilon\restriction n) = i \quad \text{iff} \quad & \exists D_i^j \in \mathbb{D}_i \; D_i^j \subseteq \text{set}(\varepsilon\restriction n) \; \& \\ & \neg\exists D_i^k \in \mathbb{D}_i \; D_i^k \subseteq \text{set}(\varepsilon\restriction n-1). \end{aligned}$$

($\Rightarrow$) Assume that $L(\varepsilon\restriction n) = i$ and, for contradiction, that the right-hand side of the above equivalence does not hold. Then there are two possibilities.

1. $\forall D_i^j \in \mathbb{D}_i \; D_i^j \not\subseteq \text{set}(\varepsilon\restriction n)$. But then from the assumption that $L(\varepsilon\restriction n) = i$, by the definition of L we have also that $\text{set}(\varepsilon\restriction n) \subseteq S_i$ and $\exists S \subseteq \text{set}(\varepsilon\restriction n)$ such that $f_{\text{c-dftt}}(S, i) = 1$. Hence, by the definition of $f_{\text{c-dftt}}$, S is a DFTT of S_i and $S \subseteq \text{set}(\varepsilon\restriction n)$. Contradiction.

2. $\exists D_i^k \in \mathbb{D}_i \; D_i^k \subseteq \text{set}(\varepsilon\restriction n-1)$. Since $L(\varepsilon\restriction n) = i$, by the definition of L we have that $\forall \ell < n \; L(\varepsilon\restriction \ell) = \uparrow$ and hence (a) $\forall \ell < n-1 \; L(\varepsilon\restriction \ell) = \uparrow$. Moreover, by the definition of $f_{\text{c-dftt}}$, (b) $f_{\text{c-dftt}}(\text{set}(\varepsilon\restriction n-1), i) = 1$ and by the definition of DFTT, (c) $\text{set}(\varepsilon\restriction n-1) \subseteq S_i$. From (a), (b) and (c) we can conclude that $L(\varepsilon\restriction n-1) = i$. This contradicts the fact that L is (at most) once defined.

($\Leftarrow$) Assume that for ε a text for $S_i \in C$ and $n \in \mathbb{N}$ the following holds:

$$\exists D_i^j \in \mathbb{D}_i \quad D_i^j \subseteq \text{set}(\varepsilon\restriction n) \; \& \tag{1}$$

$$\neg\exists D_i^k \in \mathbb{D}_i \quad D_i^k \subseteq \text{set}(\varepsilon\restriction n-1) \tag{2}$$

Then, by (2), for all $k < n$ there is no S such that $S \subseteq \text{set}(\varepsilon\restriction k)$ and $f(S, i) = 1$. Hence, by the definition of L for all $k < n$, $L(\varepsilon\restriction k) \neq i$. Since ε is a text for S_i it can not enumerate any DFTT of some different set in $\mathcal{C}$, hence we have that for all $k < n$, $L(\varepsilon\restriction k) = \uparrow$. By (1) and the definition of $f_{\text{c-dftt}}$ we get that $\exists S \subseteq \text{set}(\varepsilon\restriction n) \; f_{\text{c-dftt}}(S, i) = 1$. So, $L(\varepsilon\restriction n) = i$.

This completes the proof. □

The above theorem gives a condition of fastest finite identifiability. For any finite set and $i \in \mathbb{N}$, the function f decides whether S is a DFTT of S_i. In other words, the class of all DFTTs of $\mathcal{C}$, $\{\mathbb{D}_i \mid S_i \in \mathcal{C}\}$, is uniformly decidable. In fact, the function that the fastest preset learner uses does not have to give a positive answer every time it sees a DFTT, it is enough if the function signals the occurrence of every minimal DFTT.

Definition 6.4.4. *Let $\mathcal{C}$ be an indexed family of recursive sets. The* min dftt-function *for $\mathcal{C}$ is a recursive function $f_{min\text{-}dftt} : \mathcal{P}^{<\omega}(\mathbb{N}) \times \mathbb{N} \to \{0,1\}$, such that:*

1. *$f_{min\text{-}dftt}(S,i) = 1$ if and only if S is a minimal DFTT of S_i;*
2. *for every $i \in \mathbb{N}$ there is a finite $S \subseteq \mathbb{N}$, such that $f_{min\text{-}dftt}(S,i) = 1$.*

Theorem 6.4.5. *A class $\mathcal{C}$ is finitely identifiable in the fastest way iff there is a min-dftt-function for $\mathcal{C}$.*

Proof. ($\Rightarrow$) Assume that a class $\mathcal{C}$ is finitely identifiable in the fastest way. Then, by Theorem 6.4.3, there is an effective function $f_{\text{c-dftt}} : \mathcal{P}^{<\omega}(\mathbb{N}) \times \mathbb{N} \to \{0,1\}$ such that $f_{\text{c-dftt}}(S,i) = 1$ iff S is a DFTT for S_i and for every $i \in \mathbb{N}$ there is a finite set $S \subseteq \mathbb{N}$ such that $f_{\text{c-dftt}}(S,i) = 1$.

We define $f_{\text{min-dftt}} : \mathcal{P}^{<\omega}(\mathbb{N}) \times \mathbb{N} \to \{0,1\}$ in the following way:

$$f_{\text{min-dftt}}(S,i) = \begin{cases} 1 & \text{if } f_{\text{c-dftt}}(S,i) = 1 \ \& \ \neg\exists T \subset S \ f_{\text{c-dftt}}(T,i) = 1, \\ 0 & \text{otherwise.} \end{cases}$$

The function $f_{\text{min-dftt}}$ is recursive, because S is finite and $f_{\text{c-dftt}}$ is recursive.
($\Rightarrow$) Assume that there is an effective function $f_{\text{min-dftt}} : \mathcal{P}^{<\omega}(\mathbb{N}) \times \mathbb{N} \to \{0,1\}$ such that $f_{\text{min-dftt}}(S,i) = 1$ iff S is a minimal DFTT for S_i and for every $i \in \mathbb{N}$ there is a finite set $S \subseteq \mathbb{N}$ such that $f_{\text{min-dftt}}(S,i) = 1$.

We define $f_{\text{c-dftt}} : \mathcal{P}^{<\omega}(\mathbb{N}) \times \mathbb{N} \to \{0,1\}$ in the following way:

$$f_{\text{c-dftt}}(S,i) = \begin{cases} 1 & \text{if } \exists T \subseteq S \ f_{\text{min-dftt}}(T,i) = 1, \\ 0 & \text{otherwise.} \end{cases}$$

The function $f_{\text{c-dftt}}$ is recursive, because S is finite and $f_{\text{min-dftt}}$ is recursive. □

6.4.1 Strict Preset Learning

The reader may have expected a stronger notion of preset learner for which a recursive function F exists such that that for each S_i, $F(i)$ is a finite set of DFTTs for S_i. As announced before we will now be interested in learners that have at their *direct* disposal all minimal DFTTs of languages from the given class. Obviously, such a situation is only generally possible in the case of classes of finite languages. We will consider both possibilities: the one of finite classes of finite languages and that of infinite classes of finite languages. We define strict preset finite identifiability in the following way.

Definition 6.4.6. *A finitely identifiable class $\mathcal{C}$ is* strict preset finitely identifiable *iff there is a recursive function $F : \mathbb{N} \to \mathcal{P}^{<\omega}(\mathbb{N})$ such that $F(i)$ outputs the set of all minimal DFTTs of S_i.*

Computational Complexity of Strict Preset Learning Let us again go back to the case of a finite class of finite sets. To compute the set min-$\mathbb{D}_i$ of all minimal DFTTs of $S_i \in \mathcal{C}$ we need to perform the procedure for finding a minimal DFTT for all possible orderings of elements in S_i. Therefore the simple procedure described earlier (in Section 6.3.2) has to be performed $n!$ times. This indicates that finding the set of all minimal DFTTs is in general quite costly. We show that in fact the problem is NP-hard.

Proposition 6.4.7. *Finding min-$\mathbb{D}_i$ of $S_i \in \mathcal{C}$ is NP-hard.*

Proof. It is easy to observe that once we enumerate D^i of S_i, we can find a minimal-size DFTT of S_i in polynomial time by simply picking one of the smallest sets in D^i. This means that the Minimal-size DFTT Problem for S_i can be polynomially reduced to the problem of finding D^i for S_i. □

We will now move to the case of infinite classes of finite sets. We will compare the two notions: strict preset learning and fastest learning in the more general setting of recursive sets. We will proceed with a number of examples. First we will show that there are classes that are finitely identifiable both in the fastest and in the preset way, but that are not strict preset finitely identifiable.

In the following we will use the manner of speech where we will say that e is a Turing machine if we mean that e is an integer that codes a Turing machine and $f(a) = \{b, c\}$ will mean that $f(a)$ codes the finite set containing just b and c. Let us recall the notion of Kleene's T-predicate.

Definition 6.4.8 (T-predicate (Kleene, 1943))**.** *$T(e, x, y)$ holds iff e is a Turing machine that on input x performs computation y.*

Let us recall that with the use of the T-predicate the Halting Problem can be defined in the following way:

$$\exists y T(e, e, y) \iff \varphi_e(e) \downarrow .$$

In other words, the question of whether Turing machine e stops on the input e is equivalent to the question of existence of a computation y performed by e on input e.

Let us start with an example of a finitely identifiable class for which a recursive preset learner exists, but which is not strict preset finitely identifiable.

Theorem 6.4.9. *There exists a class $\mathcal{C}$ that is finitely identifiable, but for which no recursive function F exists such that for each i, $F(i)$ is the set of all minimal DFTTs for S_i.*

Proof. Let us consider the following class of finite sets $\mathcal{C} = \{S_i \mid i \in \mathbb{N}\}$:

$$S_i = \{2i, 2(\mu y T(i, i, y)) + 1\}.$$

Obviously, the class $\mathcal{C}$ is finitely identifiable. Moreover there exists a recursive fastest learner L that finitely identifies $\mathcal{C}$ and can be defined in the following way:

$$L(\varepsilon{\restriction}n) = \begin{cases} i & \text{if } 2i \in \mathrm{set}(\varepsilon{\restriction}n), \\ \mu\ell\ T(\ell,\ell,k) & \text{if } 2k+1 \in \mathrm{set}(\varepsilon{\restriction}n). \end{cases}$$

We can easily observe that $|S_i| = 2 \iff \varphi_i(i)\downarrow$. Minimal DFTTs of S_i are $\{2i\}$, and, in case $\varphi_i(i)\downarrow$, also $\{2(\mu y T(i,i,y))+1\}$. It is clear that no total recursive function F such that

$$g(i) = \{\{2i\}, \{2(\mu y T(i,i,y))+1\}\}$$

can be given, because its existence would solve the Halting Problem. □

Therefore, the strict preset learner for the above class cannot be recursive. Since the recursive fastest learner exists (defined in the proof above), the fastest learner cannot be strict preset. Hence, we can conclude that even in the case of classes of finite languages strict preset finite identification is properly included in finite identification and in fastest finite identification. A similar result can be shown for the minimal-size strict preset finite identifiability, i.e., when the learning function requires the set of all the minimal-size DFTTs.

Proposition 6.4.10. *There exists a class $\mathcal{C}$ that is finitely identifiable, but for which no recursive function F exists such that for each i, $F(i)$ is the set of all minimal-size DFTTs for S_i.*

Proof. The argument is analogous to the one given in the proof of Theorem 6.4.9. Let $j : \mathbb{N}^2 \to \mathbb{N}$ be a recursive pairing function (bijection) with inverses j_1 and j_2. We now consider the class $\mathcal{C} = \{S_i \mid i \in \mathcal{C}\}$:

$$S_i = \{3j_1(i), 3j_2(i)+1, 3(\mu y T(i,i,y))+2\}.$$

The set of all minimal-size DFTTs of S_i is $\{\{3(\mu y T(i,i,y))+2\}\}$ in case $\varphi_i(i)\downarrow$ and $\{\{3j_1(i)\}, \{3j_2(i)+1\}\}$ in case $\varphi_i(i)\uparrow$. Therefore the minimal-size DFTTs of S_i cannot be given by a total recursive function, because its existence would solve the Halting Problem. □

6.4.2 Finite Learning and Fastest Learning

We have established that recursive fastest identifiability does not require strict preset learnability even in the finite cases. We will now turn to the more general question whether every finitely identifiable class has a fastest learner. The answer is negative—there are finitely identifiable classes of languages which cannot be finitely identified in the fastest way.

Definition 6.4.11 (Smullyan 1958)**.** *Let $A, B \subset \mathbb{N}$. A separating set is $C \subset \mathbb{N}$ such that $A \subset C$ and $B \cap C = \emptyset$. In particular, if A and B are disjoint then A itself is a separating set for the pair, as is B. If a pair of disjoint sets A and B has no computable separating set, then the two sets are* recursively inseparable.

The following theorem says that there are effectively finitely identifiable classes of languages for which there is no effective fastest learner and no recursive function that could enumerate a minimal DFTT for each language.

Theorem 6.4.12. *There exists a class $\mathcal{C}$ that is finitely identifiable but is not finitely identifiable in the fastest way (and moreover there is no recursive function that gives a minimal-size DFTT for each language).*

Proof. Let A and B be two disjoint r.e. recursively inseparable sets. Let $x \in A$ be equivalent to $\exists y\, Rxy$ with R recursive and let $x \in B$ be equivalent to $\exists y\, Sxy$ with S recursive. We assume that for each x there is at most one y such that Rxy and at most one y such that Sxy.

The class of languages $\{S_i \mid i \in \mathbb{N}\}$ is defined as follows:

$$S_i = \{2i, 2i+1\} \cup \{2j \mid Rji\} \cup \{2j+1 \mid Sji\}.$$

The idea is that $\{2i, 2i+1\}$ is the exclusive domain of S_i except that, for some usually much larger m, Rim or Sim may be true, and then $2i \in S_m$ or $2i+1 \in S_m$, respectively. There can be at most one such m, and for that m only one of Rim or Sim can be true. However, since A and B are recursively inseparable there can be no recursive function f that makes this latter choice for each i.

It is clear that to decide definitely that the language is S_i it suffices to encounter $2i$ and $2i+1$, so $\{2i, 2i+1\}$ is a DFTT for S_i. But, if $i \in A$, then $\{2i+1\}$ is a DFTT for S_i, and if $i \in B$, then $\{2i\}$ is a DFTT for S_i. To be more precise, $\{2i+1\}$ is a DFTT for S_i if $i \notin B$ and $\{2i\}$ is a DFTT for S_i if $i \notin A$. But no recursive learner can decide on this, so there cannot be a recursive fastest learner, or a function that gives, for each i, a minimal DFTT for S_i. □

In general there is no reason for preset learners to use only minimal DFTTs. They can have a set of DFTTs for each S_i and only use those. The class given in the proof of Theorem 6.4.12, for which no recursive fastest learner is possible can obviously have a preset learner but the DFTTs do not include the minimal ones.

The above theorem shows that fastest finite identifiability is properly included in finite identification and hence also in preset finite identification. Hence, we have shown the existence of yet another kind of learning, even more demanding than finite identification. Speaking in terms of conclusive update, our considerations show that in some cases, even if computable convergence to certainty is possible, it is not computable to conclude that at the first moment in which objective ambiguity disappears.

In the light of these discoveries about preset learning, we can give a computational justification for introducing multi-agency to this setting. It seems to be justifiable to switch the perspective from the single agent, learning-oriented view, to the two agent game of learner and *teacher*. The responsibility of effective learning, in the line with natural intuitions, is in the hands of the teacher, whose computational task is to find samples of information that guarantee optimal learning. Intuitively, it is not very surprising that the task of finding such minimal samples can be more difficult than the complexity of the actual learning. As such, computing the minimal(-size) DFTTs seems to go beyond the abilities of the learner and is not necessary in order to be rational or successful. However, such a task is natural to be performed by a teacher.

6.5 Conclusions and Perspectives

We used the characterization of finite identification of languages from positive data to discuss the complexity of optimal learning and teaching strategies in finite identification. We introduced two notions: minimal DFTT and minimal-size DFTT. By viewing the informativeness of examples as their power to eliminate certain conjectures, we have checked the computational complexity of 'finite teachability' from minimal DFTT and minimal-size DFTT. In the former case the problem turns out to be PTIME computable, while the latter falls into the NP-complete class of problems. This suggests that it is easy to teach in a way that avoids irrelevant information but it is potentially difficult to teach in the most effective way. We also conceptually extended the characterization of finite identification and introduced the notion of preset learner. We compared variations of the latter with the idea of fastest finite identification. In particular, we focused on the notion of strict preset finite identifiers that have at their disposal all minimal DFTTs of every language in the class. Even in the setting of classes of finite sets this type of learning turns out to be restrictive with respect to finite identifiability. We have exhibited a recursively finitely identifiable class that cannot be recursively finitely identifiable in the fastest way. Hence we have established a new more restrictive kind of finite identification.

Links between finite identification and dynamic epistemic logic have already been established and described in Chapters 3–5. For dynamic epistemic logic the restriction to finite sets of finite languages is very natural, so our analysis of its complexity can be applied to strengthen this connection. The complexity of fastest finite identification corresponds to the complexity of fastest conclusive update in dynamic epistemic logic. It gives a new measure of computational complexity of certainty— a measure that corresponds to the question of how difficult it is to reach the state of irrevocable knowledge.

The main assumptions of learning theory and hence also of the present chapter is the cooperative nature of the interaction between the learner and the teacher. In the next chapter we will reflect upon other possible learning attitudes.

Chapter 7

Supervision and Learning Attitudes

In philosophy, logic and psychology the word ‘learning’ is used to denote a variety of phenomena. In this chapter, as in the previous ones, we take the phrase ‘to learn’ to mean ‘to acquire information in order to arrive at a certain (correct) conclusion’. In this we follow the lines of learning theory (see, e.g., Jain et al., 1999) where learning consists of a sequence of mind changes that should lead to a correct conclusion. All belief states visited on the way there, together with the correct one, are drawn from some given set of possibilities that constitutes the initial uncertainty range. To be more specific, let us try to describe in those terms the process of language learning. Agent L’s (Learner’s) aim is to ‘arrive’ at a grammar that correctly describes his native language. This scenario can be represented by a graph in which the vertices represent possible grammars and an edge from vertex s_1 to vertex s_2 labeled with α stands for a possibility of a mind change from grammar s_1 to s_2 that is triggered by the incoming information α. The properties of such a graph are determined by features of the initial range of possibilities, the language being learned, and the nature of agent’s learning strategy. Then, the process of learning can be represented simply as a ‘journey’ of L through the graph until he finally reaches the correct grammar. Obviously, other inductive inference processes can also be described in this way.

The setting can be enriched by the presence of another agent, let us call her Teacher, T, who decides which data are presented to L, in which order, etc. In other words, we can introduce another player who supervises the process of learning and manipulates the data in order to influence the speed and accuracy of convergence. The analysis of the role of the teacher has been increasingly present in formal learning theory (e.g., see Angluin, 1987 for the *minimally adequate teacher* in learning from queries and counterexamples, and Balbach & Zeugmann, 2009 for recent developments in teachability theory).

In this chapter, we investigate the interaction between Learner and Teacher in a particular kind of *supervision* learning game that is played on a graph. Learner’s information state changes while he moves around the graph, from one conjecture to another. Teacher, having a global perspective, knows the structure of the graph, and by providing certain information eliminates some initially possible mind changes of L. We are interested in the complexity of teaching, which we interpret in a similar way as in Chapter 6. Assuming the global perspective of Teacher, we identify the teachability problem with deciding whether the success of the learning process is possible. We interpret learning as a game and hence we identify learnability and teachability with the existence of winning strategies in a certain type of game. In this context, we analyze different Learner and Teacher attitudes, varying the level of Teacher's helpfulness and Learner's willingness to learn.

We interpret our learning game within the existing framework of *sabotage games* (Van Benthem, 2005). We start by recalling sabotage games and *sabotage modal logic*. Then, we explore *variations of the winning condition* of the game, providing sabotage modal logic formulae that characterize the existence of a winning strategy and their model-checking complexity (in other words: the complexity of deciding whether a player has a winning strategy in the game) in each case. Then we observe the asymmetry of the players' roles, and we allow Teacher to skip moves and we analyze how such removal of strict alternation of the moves affects our previous results. Finally, we recapitulate the work and discuss possible extensions.

With respect to the previous chapter, which also deals with computational complexity of teaching, we will now, in a way, take a step back. We will lose the detailed view on the content of epistemic states. Instead, we will gain the global picture of the process on conjecture change, and will be able to see the influence of intentional attitudes on the complexity of teaching.

7.1 Sabotage Games

As we already mentioned, our perspective on learning leads naturally to the framework of sabotage games. Sabotage games are useful for reasoning about various interactive processes involving random breakdowns or intentional obstruction in a system, from the failures of server networks to the logistics of traveling by public transport. We argue that it can also be interpreted positively, as some form of learning. But before we get to that, we first introduce the general, basic framework of sabotage games.

A sabotage game is played in a directed multi-graph, with two players, *Runner* and *Blocker*, moving in alternation, with Runner moving first. Runner moves by making a single transition from the *current* vertex. Blocker moves by deleting *any* edge from the graph. Runner wins the game if he is able to reach a designated goal vertex; otherwise Blocker wins.

To define the game formally let us first introduce the structure in which sabotage games take place, a directed multi-graph (see, e.g., Balakrishnan, 1997).

Definition 7.1.1. *A* directed multi-graph *is a pair $G = (V, E)$ where V is a set of vertices and $E : V \times V \to \mathbb{N}$ is a function indicating the number of edges between any two vertices.*

The sabotage game is defined in the following way.

Definition 7.1.2 (Löding & Rohde 2003a)**.** *A* sabotage game

$$\mathtt{SG} = \langle V, E, v, v_g \rangle$$

is given by a directed multi-graph (V, E) and two vertices $v, v_g \in V$. Vertex v represents the initial position of Runner and v_g represents the goal state (the aim of Runner).

Each match consists of a sequence of positions and is played as follows:

1. *the initial position $\langle E_0, v_0 \rangle$ is given by $\langle E, v \rangle$;*
2. *round $k + 1$ from position $\langle E_k, v_k \rangle$ consists of:*
   (a) *Runner moving to some v_{k+1} such that $E(v_k, v_{k+1}) > 0$, and then*
   (b) *Blocker removing an edge (v, v') such that $E_k(v, v') > 0$.*

   The new position is $\langle E_{k+1}, v_{k+1} \rangle$, where $E_{k+1}(v, v') := E_k(v, v') - 1$ and, for every $(u, u') \neq (v, v')$, $E_{k+1}(u, u') := E_k(u, u')$;
3. *the match ends if a player cannot make a move or if Learner reaches the goal state, which is the only case in which he wins.*

In other words, Blocker removes an edge between two states v, v' by decreasing the value of $E(v, v')$ by 1. As we will see later, this description of the game based on the above definition of multi-graphs can lead to some technical problems when we want to interpret modal logic over these structures. Therefore, we will now present an alternative definition, which we later show to be equivalent with respect to the existence of a winning strategy[1].

Definition 7.1.3. *Let $\Sigma = \{a_1, \ldots a_n\}$ be a finite set of labels. A* directed labeled multi-graph *is a tuple $G^{\Sigma} = (V, \mathcal{E})$ where V is a set of vertices and $\mathcal{E} = (\mathcal{E}_{a_1}, \ldots, \mathcal{E}_{a_n})$ is a collection of binary relations $\mathcal{E}_{a_i} \subseteq V \times V$ for each $a_i \in \Sigma$.*

In the above definition, the labels from Σ are used to represent multiple edges between two vertices; $\mathcal{E}$ is simply an ordered collection of binary relations on V with labels drawn from Σ. Accordingly, the modified definition of sabotage game is as follows.

[1] In what follows we take the size of any multi-graph $G = (V, E)$ to be bounded by: $|V| + \max\{E(v, w) \mid v, w \in V\} \cdot |V^2|$.

Definition 7.1.4. *A* labeled sabotage game

$$\mathtt{SG}^{\Sigma} = \langle V, \mathcal{E}, v, v_g \rangle$$

is given by a directed labeled multi-graph $(V, \mathcal{E})$ *and two vertices* $v, v_g \in V$*. Vertex* v *represents the initial position of Runner and* v_g *represents the goal state.*[2]

Each match is played as follows:

1. *the initial position* $\langle \mathcal{E}^0, v_0 \rangle$ *is given by* $\langle \mathcal{E}, v \rangle$*;*
2. *round* $k+1$ *from position* $\langle \mathcal{E}^k, v_k \rangle$ *with* $\mathcal{E}^k = (\mathcal{E}^k_{a_1}, \ldots, \mathcal{E}^k_{a_n})$*, consists of:*
   (a) *Runner moving to some* v_{k+1} *such that* $(v_k, v_{k+1}) \in \mathcal{E}^k_{a_i}$ *for some* $a_i \in \Sigma$*, and then*
   (b) *Blocker removing an edge* (v, v') *with label* a_j *(*$(v, v') \in \mathcal{E}^k_{a_j}$*) for some* $a_j \in \Sigma$*.*

   The new position is $\langle \mathcal{E}^{k+1}, v_{k+1} \rangle$*, where* $\mathcal{E}^{k+1}_{a_j} := \mathcal{E}^k_{a_j} - \{(v, v')\}$ *and* $\mathcal{E}^{k+1}_{a_i} := \mathcal{E}^k_{a_i}$ *for all* $i \neq j$*;*
3. *The match ends if a player cannot make a move or if Runner reaches the goal state, which is the only case in which he wins.*

It is easy to see that both versions of sabotage games have the *history-free determinacy property*: if one of the players has a winning strategy then (s)he has a winning strategy that depends only on the current position. Then, each round can be viewed as a transition from a sabotage game $\mathtt{SG}^{\Sigma} = \langle V, \mathcal{E}^k, v_k, v_g \rangle$ to another sabotage game $\mathtt{SG}'^{\Sigma} = \langle V, \mathcal{E}^{k+1}, v_{k+1}, v_g \rangle$, since all previous moves become irrelevant. We will use this fact through the whole paper.

It is easy to see that in *labeled* sabotage games, the label of the edge removed by Blocker is irrelevant with respect to the existence of a winning strategy. What matters is the number of edges that is left.

Observation 7.1.5. *Let* $\mathtt{SG}^{\Sigma} = \langle V, \mathcal{E}, v_0, v_g \rangle$ *and* $\mathtt{SG}'^{\Sigma} = \langle V, \mathcal{E}', v_0, v_g \rangle$ *be two labeled sabotage games that differ only in the labels of their edges, that is,*

$$\textit{for all } (v, v') \in V \times V, \ |\{\mathcal{E}_{a_i} \mid (v, v') \in \mathcal{E}_{a_i}\}| = |\{\mathcal{E}'_{a_i} \mid (v, v') \in \mathcal{E}'_{a_i}\}|,$$

where $|\cdot|$ *stands for cardinality. Then Runner has a winning strategy in* $\mathtt{SG}^{\Sigma}$ *iff he has a wining strategy in* $\mathtt{SG}'^{\Sigma}$*.*

The existing results on sabotage have been given for the non-labeled version of the game. In what follows we show that the problems of deciding whether Runner has a winning strategy in sabotage games $\mathtt{SG}$ and $\mathtt{SG}^{\Sigma}$ are polynomially equivalent.

[2] We will sometimes talk about edges and vertices of $\mathtt{SG}^{\Sigma} = \langle V, \mathcal{E}, v, v_g \rangle$, meaning edges and vertices of its underlying directed (labeled) multi-graph $(V, \mathcal{E})$.

By doing this we establish that our modification of the definition makes only a slight difference and that the previous contribution is valid for our notion. We start by formalizing the two problems.

Definition 7.1.6 (Sabotage Decision Problem)**.**

Instance *Sabotage game* $\mathtt{SG} = \langle V, E, v_0, v_g\rangle$*.*

Question *Does Runner have a winning strategy in* $\mathtt{SG}$*?*

The Σ-Sabotage Decision Problem is very similar. The only difference is that it is concerned with slightly modified structures — labeled sabotage games.

Definition 7.1.7 (Σ-Sabotage Decision Problem)**.**

Instance *Labeled sabotage game* $\mathtt{SG}^{\Sigma} = \langle V, \mathcal{E}, v_0, v_g\rangle$*.*

Question *Does Runner have a winning strategy in* $\mathtt{SG}^{\Sigma}$*?*

Theorem 7.1.8. Sabotage *and* Σ-Sabotage *are polynomially equivalent.*

Proof. The two problems can be polynomially reduced to each other.

($\Rightarrow$) Sabotage can be reduced to Σ-Sabotage. Given a sabotage game $\mathtt{SG} = \langle V, E, v_0, v_g\rangle$, let m be the maximal number of edges between any two vertices in the graph, i.e.:

$$m := \max\{E(u, u') \mid (u, u') \in (V \times V)\}.$$

Then, we define the labeled sabotage game $f(\mathtt{SG}) := \langle V, \mathcal{E}, v_0, v_g\rangle$, where $\mathcal{E} := (\mathcal{E}_1, \dots, \mathcal{E}_m)$ and each $\mathcal{E}_i$ is given by $\mathcal{E}_i := \{(u, u') \in V \times V \mid E(u, u') \geq i\}$.

We have to show that Runner has a winning strategy in $\mathtt{SG}$ iff he has one in $f(\mathtt{SG})$. The proof is by induction on n — the number of edges in $\mathtt{SG}$, i.e., $n = \sum_{(v,v') \in V \times V} E(v, v')$. Note that by definition of f, $f(\mathtt{SG})$ has the same number of edges, i.e., $n = \sum_{i=1}^{i=m} |\mathcal{E}_i|$.

The base case

Straightforward. In both games Runner has a winning strategy iff $v_0 = v_g$.

The inductive case

($\Rightarrow$) Suppose that Runner has a winning strategy in the game $\mathtt{SG} = \langle V, E, v_0, v_g\rangle$ with $n+1$ edges. Then, there is some $v_1 \in V$ such that $E(v_0, v_1) > 0$ and Runner has a winning strategy for all games $\mathtt{SG}' = \langle V, E', v_1, v_g\rangle$ that result from Blocker removing any edge (u, u') with $E(u, u') > 0$. Note that all such games $\mathtt{SG}'$ have just n edges, so by the induction hypothesis Runner has a winning strategy in $f(\mathtt{SG}')$. But then, by Observation 7.1.5, Runner also has a winning strategy in all games $f(\mathtt{SG})'$ that result from removing an arbitrary edge from $f(\mathtt{SG})$. This is so because for any removed edge (u, u'), the only possible difference between $f(\mathtt{SG}')$ and $f(\mathtt{SG})'$ is in the labels of the edges between u and u' (in $f(\mathtt{SG}')$ the removed label was the largest, in $f(\mathtt{SG})'$ the removed label is arbitrary). Now, by definition of f, choosing v_1 is also a legal move for Runner in $f(\mathtt{SG})$ and, since he can win every $f(\mathtt{SG})'$, he has a winning strategy in $f(\mathtt{SG})$.

($\Leftarrow$) Runner having a winning strategy in $f(\mathtt{SG})$ means that he can choose some v_1 with $(v_0, v_1) \in \mathcal{E}_i$ for some $i \leq m$ such that he has a winning strategy in all games $f(\mathtt{SG})'$ resulting from Blocker's move. Choosing v_1 is also a legal move of Runner in $\mathtt{SG}$. Suppose that Blocker replies by choosing (v, v'). Let us call the resulting game $\mathtt{SG}'$. By assumption and Observation 7.1.5, Runner also has a winning strategy in the game $f(\mathtt{SG}')$ which is the result from Blocker choosing $((v, v'), E(v, v'))$. Since $f(\mathtt{SG})' = f(\mathtt{SG}')$, we can apply the inductive hypothesis.

($\Leftarrow$) Σ-Sabotage can be reduced to Sabotage. Given a labeled sabotage game $\mathtt{SG}^{\Sigma} = \langle V, \mathcal{E}, v, v_g \rangle$ with $\Sigma = \{a_1, \ldots a_m\}$, define the sabotage game $f'(\mathtt{SG}^{\Sigma}) := \langle V, E, v, v_g \rangle$, where $E(v, v') := |\{\mathcal{E}_{a_i} \mid (v, v') \in \mathcal{E}_{a_i}\}|$.

Showing that Runner has a winning strategy in $\mathtt{SG}^{\Sigma}$ iff he has one in $f(\mathtt{SG}^{\Sigma})$ is straightforward, and can be done by induction on the number of edges in $\mathtt{SG}^{\Sigma}$, i.e., on $n := \sum_{a \in \Sigma} |\mathcal{E}_a|$.

Finally, let us observe that both f and f' that encode the procedures of transforming one type of graph to another, are polynomial, so the proof is complete. □

7.2 Sabotage Modal Logic

Sabotage modal logic (SML) has been introduced by Van Benthem (2005) to investigate the complexity of reachability-type problems in dynamic structures, such as the graph of our sabotage games. Besides the standard modalities, it also contains 'transition-deleting' modalities for reasoning about model change that occurs when a transition (an edge) is removed. To be more precise, we have formulae of the form $\diamondminus\varphi$, expressing that it is possible to delete a pair from the accessibility relation such that φ holds in the resulting model at the current state.

Definition 7.2.1 (SML Language; Syntax). *Let* Prop *be a countable set of propositional letters and let Σ be a finite set of labels. Formulae of the language of sabotage modal logic are given by:*

$$\varphi := p \mid \neg\varphi \mid \varphi \vee \varphi \mid \Diamond_a \varphi \mid \diamondminus_a \varphi$$

with $p \in$ Prop and $a \in \Sigma$. The formula $\boxminus_a \varphi$ is defined as $\neg\diamondminus_a \neg\varphi$, and we will write $\Diamond\varphi$ for $\bigvee_{a \in \Sigma} \Diamond_a \varphi$ and $\diamondminus\varphi$ for $\bigvee_{a \in \Sigma} \diamondminus_a \varphi$.

The sabotage modal language is interpreted over Kripke models, that are here called *sabotage models*.

Definition 7.2.2 (Löding & Rohde 2003b). *Given a countable set of propositional letters* Prop *and a finite set $\Sigma = \{a_1, \ldots, a_n\}$, a* sabotage model *is a tuple $M = \langle W, (R_{a_i})_{a_i \in \Sigma}, Val \rangle$ where W is a non-empty set of worlds, each $R_{a_i} \subseteq W \times W$*

is an accessibility relation and $Val : \textsc{Prop} \to \mathcal{P}(W)$ *is a propositional valuation function. We will call a pair* (M, w) *with* $w \in W$ *a* pointed sabotage model.

To get to the semantics of sabotage modal language, we first have to define the model that results from removing an edge.

Definition 7.2.3. *Let* $M = \langle W, R_{a_1}, \dots R_{a_n}, Val\rangle$ *be a sabotage model. The model* $M^{a_i}_{(v,v')}$ *that results from removing the edge* $(v, v') \in R_{a_i}$ *is defined as*

$$M^{a_i}_{(v,v')} := \langle W, R_{a_1}, \dots R_{a_{i-1}}, R_{a_i} \setminus \{(v, v')\}, R_{a_{i+1}}, \dots R_{a_n}, Val\rangle.$$

Definition 7.2.4 (SML; Semantics)**.** *Given a sabotage model*

$$M = \langle W, (R_a)_{a\in\Sigma}, Val\rangle$$

and a world $w \in W$*, atomic propositions, negations, disjunctions and standard modal formulae are interpreted as usual. For the case of 'transition-deleting' formulae, we have*

$$(M, w) \models \Diamond\!\!\!\!-_a\varphi \text{ iff } \exists v, v' \in W \ ((v, v') \in R_a \ \& \ (M^a_{(v,v')}, w) \models \varphi).$$

One SML result that is of great importance to us is the SML model checking complexity (combined complexity model checking, see Vardi, 1982). We will use it to reason about the difficulty of our learning scenarios.

Theorem 7.2.5 (Löding & Rohde 2003b)**.** *The computational complexity of model checking for SML is PSPACE-complete.*

7.3 Sabotage Learning Games

In this section, we reinterpret the sabotage game in the broader perspective of learning. We introduce variants of the winning condition of the game. For each variant, we will provide a sabotage modal logic formula characterizing the existence of a winning strategy. We also prove complexity results for model checking in each case. We will work with previously introduced labeled sabotage games, using the labeling of the edges to represent different kinds of information changes that take Learner from one state into another.

7.3.1 Three Variations

A sabotage learning game is defined as follows.

Definition 7.3.1. *A* sabotage learning game *(*`SLG`*) is a labeled sabotage game between* Learner *(L, taking the role of Runner) and* Teacher *(T, taking the role of Blocker). We distinguish three different versions,* `SLGhe`*,* `SLGhu` *and* `SLGue`*. Moves allowed for both players are those of the sabotage game. There is also no difference in the arena in which the game is played. However, the winning conditions vary from version to version (Table 7.1).*

Game	**Winning Condition**
SLGue	L wins iff L reaches the goal state, T wins otherwise.
SLGhu	T wins iff L reaches the goal state, L wins otherwise.
SLGhe	L and T win iff L reaches the goal state. Both lose otherwise.

Table 7.1: Sabotage Learning Games

The different winning conditions correspond to different levels of Teacher's helpfulness and Learner's willingness to learn. We can then have the cooperative case with *Helpful* Teacher and *Eager* Learner (SLGhe). But there are two other possibilities that we will be interested in: *Unhelpful* Teacher with *Eager* Learner (SLGue), and *Helpful* Teacher with *Unwilling* Learner (SLGhu).

Having defined the games representing various types of Teacher and Learner attitudes, we now show how sabotage modal logic can be used for reasoning about players' strategic powers in these games.

7.3.2 Sabotage Learning Games in Sabotage Modal Logic

Sabotage modal logic turns out to be useful for reasoning about graph-like structures where edges can disappear; in particular, it is useful for reasoning about sabotage learning games. In order to interpret the logic on our graphs we need to transform the arena of a labeled sabotage game into a sabotage model in which formulae of the logic can be interpreted. In fact, for each SLG we can construct a pointed sabotage model in the following straightforward way.

Definition 7.3.2. *Let* $\mathtt{SG}^{\Sigma} = \langle V, \mathcal{E}, v_0, v_g \rangle$ *be a sabotage game and* $\mathcal{E} = (\mathcal{E}_a)_{a \in \Sigma}$. *The* pointed sabotage model $(M(\mathtt{SG}^{\Sigma}), v_0)$ *over the set of atomic propositions* $\textsc{Prop} := \{goal\}$ *is given by*

$$M(\mathtt{SG}^{\Sigma}) := \langle V, \mathcal{E}, \mathit{Val} \rangle,$$

where $\mathit{Val}(goal) := \{v_g\}$.

In the light of this construction, sabotage modal logic becomes useful for reasoning about players' strategic powers in sabotage learning games. Each winning condition in Table 7.1 can be expressed by a formula of SML that characterizes the existence of a winning strategy, that is, the formula is true in a given pointed sabotage model if and only if the corresponding player has a winning strategy in the game represented by the model.

Unhelpful Teacher and Eager Learner ($\texttt{SLGue}$) Let us first consider $\texttt{SLGue}$, the original sabotage game (Van Benthem, 2005). For any $n \in \mathbb{N}$, we define the formula $\gamma_n^{\textsf{ue}}$ inductively as follows:

$$\gamma_0^{\textsf{ue}} := goal, \qquad \gamma_{n+1}^{\textsf{ue}} := goal \vee \Diamond \boxminus \gamma_n^{\textsf{ue}}.$$

The following result is Theorem 7 of Löding & Rohde (2003b) rephrased for labeled sabotage games. We provide a detailed proof to show how our *labeled* definition avoids a technical issue present in the original proof.

Theorem 7.3.3. *Learner has a winning strategy in the* $\texttt{SLGue}$

$$\texttt{SG}^{\Sigma} = \langle V, \mathcal{E}^0, v_0, v_g \rangle$$

if and only if $(M(\texttt{SG}^{\Sigma}), v_0) \models \gamma_n^{\textsf{ue}}$*, where* n *is the number of edges of* $\texttt{SG}^{\Sigma}$*.*

Proof. The proof is by induction on n.

The base case

($\Rightarrow$) If L has a winning strategy in a game $\texttt{SG}^{\Sigma}$ with no edges, then he should be already in the winning state, that is $v_0 = v_g$. Thus, $(M(\texttt{SG}^{\Sigma}), v_0) \models goal$ and hence, $(M(\texttt{SG}^{\Sigma}), v_0) \models \gamma_0^{\textsf{ue}}$.

($\Leftarrow$) If $(M(\texttt{SG}^{\Sigma}), v_0) \models \gamma_0^{\textsf{ue}}$ then $(M(\texttt{SG}^{\Sigma}), v_0) \models goal$. Since v_g is the only state where $goal$ holds, then we have $v_0 = v_g$, and therefore L wins $\texttt{SG}^{\Sigma}$ immediately.

The inductive case

($\Rightarrow$) Suppose that $\texttt{SG}^{\Sigma}$ has $n+1$ edges, and assume L has a winning strategy. There are two possibilities: L's current state is the goal state (that is, $v_0 = v_g$), or it is not.

In the first case, we get $(M(\texttt{SG}^{\Sigma}), v_0) \models goal$ and hence $(M(\texttt{SG}^{\Sigma}), v_0) \models \gamma_{n+1}^{\textsf{ue}}$. In the second case, since L has a winning strategy in $\texttt{SG}^{\Sigma}$, there is some state $v_1 \in V$ reachable from v_0, i.e., for some $a_i \in \Sigma$ (that is, $(v_0, v_1) \in \mathcal{E}^0_{a_i}$) such that in all games $\texttt{SG}^{\Sigma}_{(u,u'),a_j} = \langle V, \mathcal{E}^1_{(u,u'),a_j}, v_1, v_g \rangle$ that result from removing edge (u, u') from the relation labeled a_j, L as a winning strategy.[3]

All such games have n edges, so by inductive hypothesis we have

$$(M(\texttt{SG}^{\Sigma}_{(u,u'),a_j}), v_1) \models \gamma_n^{\textsf{ue}}.$$

for every edge (u, u') and label a_j. Now, the key observation is that each M-image of the game that results from L moving to v_1 and T removing edge (u, u') with label a_j, is exactly the model that results from removing edge (u, u') with a_j from the *model* $M(\texttt{SG}^{\Sigma})$.[4] Then, for all such (u, u') and a_j, we have

$$(M(\texttt{SG}^{\Sigma})^{a_j}_{(u,u')}, v_1) \models \gamma_n^{\textsf{ue}}.$$

[3] The collection $\mathcal{E}^1_{(u,u'),a_j}$ is given by $(\mathcal{E}^0_{a_1}, \ldots, \mathcal{E}^0_{a_j} - \{u, u'\}, \ldots, \mathcal{E}^0_{a_{|\Sigma|}})$.

[4] In the original definition of a sabotage game this is not the case. In the game, the edges are implicitly ordered by numbers (the existence of an edge labeled with k implies the existence of edges labeled with $1, \ldots, k-1$); in the model, this is not the case. When we remove an edge from a *game* we always remove the one with the highest label, but when we remove an edge from a model we remove an arbitrary one: the operations of removing an edge and turning a game into a model do not commute.

It follows that $(M(\mathtt{SG}^\Sigma), v_1) \models \boxminus \gamma_n^{\mathtt{ue}}$ and therefore $(M(\mathtt{SG}^\Sigma), v_0) \models \Diamond \boxminus \gamma_n^{\mathtt{ue}}$, that is, $(M(\mathtt{SG}^\Sigma), v_0) \models \gamma_{n+1}^{\mathtt{ue}}$.

($\Leftarrow$) Suppose that $(M(\mathtt{SG}^\Sigma), v_0) \models \mathit{goal} \vee \Diamond \boxminus \gamma_n^{\mathtt{ue}}$. Then, v_0 is the goal state or else there is a state v_1 accessible from v_0 such that $(M(\mathtt{SG}^\Sigma), v_1) \models \boxminus \gamma_n^{\mathtt{ue}}$, that is, $(M(\mathtt{SG}^\Sigma)^{a_i}_{(u,u')}, v_1) \models \gamma_n^{\mathtt{ue}}$, for all edges (u, u') and labels a_j. By inductive hypothesis, L has a winning strategy in each game that correspond to each pointed model $(M(\mathtt{SG}^\Sigma)^{a_i}_{(u,u')}, v_1)$. But these games are exactly those that result from removing any edge from the game $\langle V, \mathcal{E}^0, v_0, v_g \rangle$ after L moves from v_0 to v_1. Hence, L has a winning strategy in $\langle V, \mathcal{E}^0, v_0, v_g \rangle$, the game that corresponds to the pointed model $(M(\mathtt{SG}^\Sigma), v_0)$, as required. □

Helpful Teacher and unwilling Learner (SLGhu) Now consider SLGhu, the game with helpful Teacher and unwilling Learner. We define $\gamma_n^{\mathtt{hu}}$ inductively, as

$$\gamma_0^{\mathtt{hu}} := \mathit{goal}, \qquad \gamma_{n+1}^{\mathtt{hu}} := \mathit{goal} \vee (\Diamond \top \wedge \Box \diamondminus \gamma_n^{\mathtt{hu}}).$$

In this case, Teacher has to be sure that Learner does not get stuck before he has reached the goal state — this is why the conjunct $\Diamond \top$ is needed in the definition of $\gamma_{n+1}^{\mathtt{hu}}$. We show that this formula corresponds to the existence of a winning strategy for Teacher.

Theorem 7.3.4. *Teacher has a winning strategy in the* SLGhu

$$\mathtt{SG}^\Sigma = \langle V, \mathcal{E}^0, v_0, v_g \rangle$$

if and only if $(M(\mathtt{SG}^\Sigma), v_0) \models \gamma_n^{\mathtt{hu}}$, *where n is the number of edges of* $\mathtt{SG}^\Sigma$.

Proof. Similar to the proof of Theorem 7.3.3. □

Helpful Teacher and eager Learner (SLGhe) Finally, for SLGhe, the corresponding formula is defined as

$$\gamma_0^{\mathtt{he}} := \mathit{goal}, \qquad \gamma_{n+1}^{\mathtt{he}} := \mathit{goal} \vee \Diamond \diamondminus \gamma_n^{\mathtt{he}}.$$

Theorem 7.3.5. *Teacher and Learner have a joint winning strategy in* SLGhe

$$\mathtt{SG}^\Sigma = \langle V, \mathcal{E}^0, v_0, v_g \rangle$$

if and only if $(M(\mathtt{SG}^\Sigma), v_0) \models \gamma_n^{\mathtt{he}}$, *where n is the number of edges of* $\mathtt{SG}^\Sigma$.

Proof. L and T have a joint winning strategy if and only if there is a path from v_0 to v_g. From left to right this is obvious. From right to left, if there is such a path, then there is also one without cycles[5], and a joint winning strategy is the one that follows the path and at each step removes the edge that has just been used (it is essential that L moves first). The theorem follows by observing that $\gamma_n^{\mathtt{he}}$ expresses the existence of such path. □

The above results for the three scenarios are summarized in Table 7.2.

[5] If this is not the case, i.e., if it is essential that L uses a path twice, removing used edges could cause L to be stuck somewhere away from the goal.

Game	Winning Condition in SML	Winner
SLGue	$\gamma_0^{\mathtt{ue}} := \mathit{goal}, \quad \gamma_{n+1}^{\mathtt{ue}} := \mathit{goal} \vee \Diamond \boxminus \gamma_n^{\mathtt{ue}}$	Learner
SLGhu	$\gamma_0^{\mathtt{hu}} := \mathit{goal}, \quad \gamma_{n+1}^{\mathtt{hu}} := \mathit{goal} \vee (\Diamond\top \wedge (\Box\diamondminus\gamma_n^{\mathtt{hu}}))$	Teacher
SLGhe	$\gamma_0^{\mathtt{he}} := \mathit{goal}, \quad \gamma_{n+1}^{\mathtt{he}} := \mathit{goal} \vee \Diamond\diamondminus\gamma_n^{\mathtt{he}}$	Both

Table 7.2: Winning Conditions for SLG in SML

7.3.3 Complexity of Sabotage Learning Games

We have characterized the existence of a winning strategy in our three versions of SLGs by means of sabotage modal logic formulae. In this section, we investigate the complexity of deciding whether such formulae are true in a given pointed model, i.e., the complexity of checking whether there is a winning strategy in the corresponding game.

By Theorem 7.2.5, the model checking problem of sabotage modal logic is PSPACE-complete. This gives us PSPACE upper bounds for the complexity of deciding whether a player can win a given game. In the three cases, we can also give tight lower bounds.

Unhelpful Teacher and eager Learner (SLGue) For SLGue, which can be identified with the standard sabotage game, PSPACE-hardness is shown by reduction from Quantified Boolean Formula (Löding & Rohde, 2003b).

Theorem 7.3.6 (Löding & Rohde 2003b)**.** SLGue *is PSPACE-complete.*

Helpful Teacher and unwilling Learner (SLGhu) Whereas at first sight, SLGhu and SLGue might seem to be duals of each other, the relationship between them is more complex due to the different nature of the players' moves: Learner moves locally by choosing an state accessible *from the current one*, while Teacher moves globally by removing *an arbitrary edge*. Nevertheless, we can show PSPACE-hardness for SLGhu. In the proof we will use the Quantified Boolean Formula (QBF) problem, known to be PSPACE-complete.

Definition 7.3.7 (Quantified Boolean Formula Problem)**.**

Instance *Let φ be an instance of* QBF*, i.e., a formula:*

$$\varphi := \exists x_1 \forall x_2 \exists x_3 \ldots Q x_n \psi$$

where Q is $\exists$ for n odd, and $\forall$ for n even, and ψ is a quantifier-free formula in conjunctive normal form.

Question *Is φ satisfiable?*

Theorem 7.3.8. `SLGhu` *is PSPACE-complete.*

Proof. From Theorem 7.2.5 and Theorem 7.3.4 it follows that `SLGhu` is in PSPACE. PSPACE-hardness of `SLGhu` is proved by showing that the Quantified Boolean Formula (QBF) problem, can be polynomially reduced to `SLGhu`.[6]

We will construct a directed game arena for $\texttt{SLGhu}_\varphi$ such that Learner has a winning strategy in the game iff the formula φ is satisfiable.

The ∃-gadget. Figure 7.1 represents the situation in which Learner chooses the assignment for x_i when i is odd. This part corresponds to assigning the value to an existentially quantified variable. Learner starts in A, and moves either left or right; if he wants to make x_i true, then he moves to $\bar{X}_i$, otherwise to X_i. Let us assume that he moves right, towards $\bar{X}_i$. Then Teacher has exactly four moves to remove all the edges leading to the dead-end #. At this point, Teacher cannot remove any edge in some other place in the graph without losing. So, Learner reaches $\bar{X}_i$, and Teacher is forced to remove the edge that leads from B to X_i, because otherwise Teacher would allow Learner to reach the dead-end #. At this point Learner moves towards B, and in the next step exits the gadget. Moving back towards $\bar{X}_i$ would cause him to lose, because then Teacher could remove the edge between B and $\bar{X}_i$ and Learner would be forced to enter the goal.

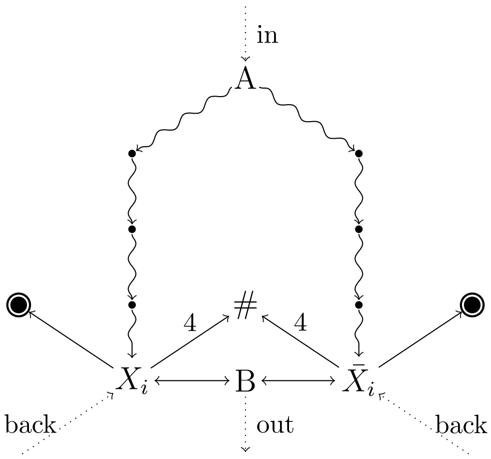


Figure 7.1: The ∃-gadget

The ∀-gadget. Figure 7.2 represents the situation in which Teacher chooses the assignment for x_i when i is even. This part corresponds to assigning the value to a universally quantified variable.

Let us assume that Teacher wants to make x_i false. Then she leads Learner towards X_i by successively removing the edges between C and $\bar{X}_i$. When Learner already is on the path to X_i, Teacher starts removing an edge going from X_i to

[6] The proof uses the same strategy to the one of Theorem 7.3.6 of Löding & Rohde (2003b). We would like to thank Frank Radmacher for suggestions about this proof.

the dead-end #. When Learner reaches X_i, he chooses to go towards B (because the other option is a goal). Then Teacher removes the edge that goes from B to $\bar{X}_i$, and Learner leaves the gadget.

Let us now assume that Teacher wants to make x_i true, and therefore wants Learner to reach $\bar{X}_i$. First she removes three of four edges from $\bar{X}_i$ to the dead-end #. Then Learner reaches C. Let us consider two cases:

1. Learner moves to $\bar{X}_i$. Teacher removes the last edge between $\bar{X}_i$ and the dead-end #, Learner moves to B, Teacher removes the edge to X_i and Learner leaves the gadget.

2. Learner moves to D. Teacher removes the four edges from X_i to the dead-end #, and then eliminates the remaining edge between $\bar{X}_i$ and the dead-end #. Learner leaves the gadget.

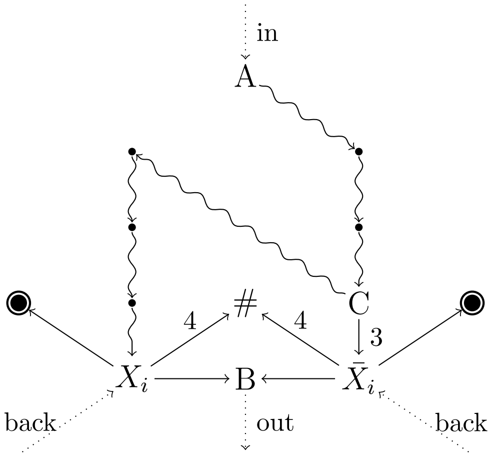


Figure 7.2: The ∀-gadget

The verification gadget. Figure 7.3 represents the situation in which Teacher chooses one clause from ψ. If Teacher chooses c, then Learner can choose one literal x_i from c. There are edges from x_i to an ∃-gadget if i is odd, and to a ∀ -gadget otherwise, leading directly to X_i if x_i is positive in c, and to $\bar{X}_i$ otherwise. So, if the chosen assignment satisfies ψ, then for all clauses there is at least one literal which is true, and leads to the opposite truth value in the corresponding gadget, from which in turn Learner can get to the dead-end # (there are four edges left) and win the game.

For the converse, if the chosen assignment does not satisfy ψ, then Learner gets to a corresponding point in a proper gadget, Teacher removes the edge from this point to B, and the only option left for Learner is to enter the goal, which means that he loses the game.

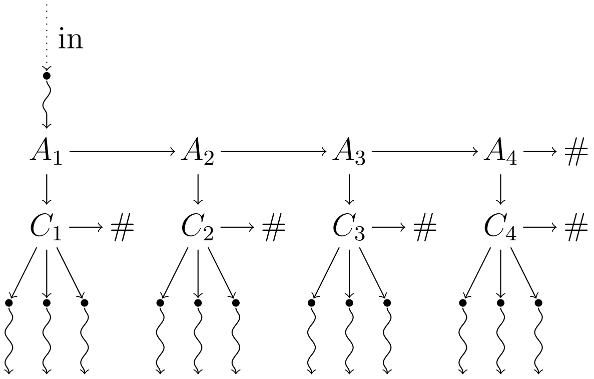


Figure 7.3: The verification gadget

With the above considerations, we can observe that Learner has a winning strategy in $\texttt{SLGhu}_\varphi$ iff φ is satisfiable. Moreover, the representation can clearly be done in polynomial time with respect to the size of φ. This finishes the proof. □

Helpful Teacher and eager Learner (SLGhe) Finally, let us have a look at SLGhe. This game is different from the two previous ones: L and T win or lose together. Then, a winning strategy for each of them need not take into account all possible moves of the other. This suggests that this version should be less complex than SLGue and SLGhu.

As mentioned in the proof of Theorem 7.3.5, L and T have a joint winning strategy if and only if the goal vertex is reachable from L's position. Thus, determining whether they can win SLGhe is equivalent to solving the REACHABILITY (ST-CONNECTIVITY) problem, which is known to be non-deterministic logarithmic space complete (NL-complete) (Papadimitriou, 1994).

Theorem 7.3.9. SLGhe *is NL-complete.*

Proof. Polynomial equivalence of SLGhe and REACHABILITY follows from the argument given in the proof of Theorem 7.3.5. □

Table 7.3 summarizes the complexity results for the different versions of SLG. The complexity of these problems can be interpreted as the difficulty of deciding whether certain aims of Teacher can be fulfilled. We can attribute the question of the existence of the winning strategy to Teacher, as she already has the global perspective on the situation anyway. Our results say how difficult it is for her to decide whether there is a chance of success. The results agree with our intuition, as coming up with a strategy to teach is easier if Learner and Teacher cooperate. Following our interpretation it is the easiest for Teacher to check whether the teaching will work out if Teacher assumes eagerness of Learner and she herself does her best to ensure that he succeeds. Moreover, the remaining two cases turn out to be equally difficult — it is as difficult to decide whether Teacher can force an unwilling Learner to learn as it is to decide whether an eager Learner can learn in the presence of an unhelpful Teacher.

From the perspective of the standard sabotage games, our complexity result for `SLGhu` means that with an additional *safety*[7] winning condition, sabotage games are PSPACE-complete.

Game	Winning Condition	Complexity
`SLGue`	Learner wins iff he reaches the goal state, Teacher wins otherwise	PSPACE-complete
`SLGhu`	Teacher wins iff Learner reaches the goal state, Learner wins otherwise	PSPACE-complete
`SLGhe`	Both players win iff Learner reaches the goal state. Both lose otherwise	NL-complete

Table 7.3: Complexity Results for Sabotage Learning Games

7.4 Relaxing Strict Alternation

As mentioned above, in sabotage (learning) games the players' moves are asymmetric: Learner moves locally (moving to a vertex accessible *from the current one*) while Teacher moves globally (removing *any* edge from the graph, and thereby manipulating the space in which Learner is moving). Intuitively, both for a helpful and an unhelpful Teacher, it is not always necessary to react to a move of Learner by giving immediate feedback (here, by removing an edge). This leads us to a variation of a `SLG` in which Learner's move need not in principle be followed by Teacher's move, i.e., Teacher has the possibility of skipping a move.

Definition 7.4.1. *A sabotage learning game without strict alternation (for T) is a tuple* $\texttt{SLG}^* = \langle V, \mathcal{E}, v_0, v_g\rangle$. *Moves of Learner are as in* `SLG` *and, once he has chosen a vertex* v_1*, Teacher can choose between removing an edge, in which case the next game is given as in* `SLG`*, and doing nothing, in which case the next game is* $\langle V, \mathcal{E}, v_1, v_g\rangle$. *Again, there are three versions with different winning conditions, now called* $\texttt{SLG}^*\texttt{ue}$, $\texttt{SLG}^*\texttt{hu}$ *and* $\texttt{SLG}^*\texttt{he}$.

After defining the class of games $\texttt{SLG}^*$, the natural question arises of how the winning abilities of the players change from `SLG` to $\texttt{SLG}^*$, since in the latter

[7] Safety concerns those properties that we want to hold throughout the process, in this case it is L being away from the goal. Safety is usually contrasted with reachability, which requires the system to get into a certain configuration at some moment, here L reaching the goal (see, e.g., Radmacher & Thomas, 2008).

Teacher can choose between removing an edge or doing nothing. In the rest of this section, we show that for all three winning conditions (SLG*ue, SLG*hu, SLG*he), the winning abilities of the players remain the same as in the case in which players move in strict alternation. This is surprising in the SLGhu case. It is by no means obvious that if Teacher has a winning strategy in the game without strict alternation then she also has one in the regular version of the game, because we might expect that removing an edge instead of skipping the move could result in blocking the way to the goal.

We start with the case of an unhelpful Teacher and an eager Learner, SLG*ue. Note that although in this new setting matches can be infinite, e.g., in the game with unwilling L, if T skips her moves indefinitely, L will just keep moving, and hence the game will not be finished in finite time. However, in fact if Learner can win the game, he can do so in a finite number of rounds. We start with a lemma stating that if Learner can win some SLGue in some number of rounds, then he can do so also if the underlying multi-graph has additional edges.

Definition 7.4.2. *Let $\Sigma = \{a_1, \dots a_n\}$ be a finite set of labels. For directed labeled multi-graphs, $G^\Sigma = (V, \mathcal{E})$ and $G'^\Sigma = (V', \mathcal{E}')$, we say that G^Σ is a* subgraph *of G'^Σ if $V \subseteq V'$ and $\mathcal{E}_{a_i} \subseteq \mathcal{E}'_{a_i}$ for all labels $a_i \in \Sigma$.*

Lemma 7.4.3. *If Learner has a strategy for winning the* SLGue *$\langle V, \mathcal{E}, v_0, v_g\rangle$ in at most n rounds, then he can also win any* SLGue *$\langle V, \mathcal{E}', v_0, v_g\rangle$ in at most n rounds, where $(V, \mathcal{E})$ is a subgraph of $(V, \mathcal{E}')$.*

Proof. The proof is by induction on n. In the inductive step, for the case that T removes an edge which was not in the original multi-graph, note that the resulting graph is a supergraph of the original one. Then we can use the inductive hypothesis. □

Theorem 7.4.4. *Let us consider the* SLG *$\langle V, \mathcal{E}, v_0, v_g\rangle$ with $(V, \mathcal{E})$ being a directed labeled multi-graph and $v, v_g \in V$. Learner has a winning strategy in the corresponding* SLGue *iff he has a wining strategy in the corresponding* SLG*ue.

Proof. From left to right, we show by induction on n that if L can win the SLGue in at most n rounds, then he can also win the SLG*ue in at most n rounds. In the inductive step, in the case that T responds by not removing any edge, we first use Lemma 7.4.3 and then can apply the inductive hypothesis.

The direction from right to left is immediate: if L has a winning strategy for SLG*ue, then he can also win the corresponding SLGue by using the same strategy. □

The case of helpful Teacher and unwilling Learner is more interesting. One might expect that the additional possibility of skipping a move gives more power to Teacher, since she could avoid removals that would have made the goal unreachable from the current vertex. However, we can show that this is not the case. First, we state the following lemmas.

Lemma 7.4.5. *Consider the* `SLG*hu` $\langle V, \mathcal{E}, v_0, v_g\rangle$*. If there is a path from* v_0 *to* v_g *and there is no path from* v_0 *to a state from where* v_g *is not reachable, then Teacher has a winning strategy.*

Proof. Let us assume that all states reachable from v_0 are on paths to v_g. Then, even if T refrains from removing any edge, L will be on a path to the goal. Now, either the path to the goal does not include a loop or it does. If it does not then T can simply wait until L arrives at the goal. If it does, T can remove the edges that lead to the loops in such a way that v_g is still reachable from any vertex. □

Lemma 7.4.6. *For all* `SLG*hu` $\langle V, \mathcal{E}, v_0, v_g\rangle$*, if Teacher has a winning strategy and there is an edge* $(v, v') \in \mathcal{E}_a$ *for some* $a \in \Sigma$ *such that no path from* v_0 *to* v_g *uses* (v, v')*, then Teacher also has a winning strategy in* $\langle V, \mathcal{E}', v_0, v_g\rangle$*, where* $\mathcal{E}'$ *results from removing* (v, v') *from* $\mathcal{E}_a$*.*

Proof. If v is not reachable from v_0, it is easy to see that the claim holds. Assume that v is reachable from v_0. T's winning strategy should prevent L from moving from v to v' (otherwise L wins). Hence, T can also win if (v, v') is not there. □

Theorem 7.4.7. *If Teacher has a winning strategy in the* `SLG*hu` $\langle V, \mathcal{E}, v_0, v_g\rangle$*, then she also has a winning strategy in which she removes an edge in each round.*

Proof. The proof is by induction on the number of edges $n = \sum_{a\in\Sigma} |\mathcal{E}_a|$.

The base case

Straightforward: there is no round because L cannot move.

The inductive case

Assume that T has a winning strategy in `SLG*hu` $\langle V, \mathcal{E}, v_0, v_g\rangle$ with $\sum_{a\in\Sigma} |\mathcal{E}_a| = n + 1$.

If $v_0 = v_g$, it is obvious. Otherwise, since T can win, there is some $v_1 \in V$ such that $(v_0, v_1) \in \mathcal{E}_a$ for some $a \in \Sigma$ and for all such v_1 we have:

1. There is a path from v_1 to v_g, and
2. (a) T can win $\langle V, \mathcal{E}, v_1, v_g\rangle$, or

   (b) there is a $((v, v'), a) \in (V \times V) \times \Sigma$ such that $(v, v') \in \mathcal{E}_a$ and T can win $\langle V, \mathcal{E}', v_1, v_g\rangle$ where $\mathcal{E}'$ is the result from removing (v, v') from $\mathcal{E}_a$.

If 2b holds, since $\sum_{a\in\Sigma} |\mathcal{E}'_a| = n$, we are done — we use the inductive hypothesis to conclude that T has a winning strategy in which she removes an edge in each round (in particular, her first choice is $((v, v'), a)$). Let us show that 2b holds.

If there is some $(v, v') \in V \times V$ such that $(v, v') \in \mathcal{E}_a$ for some $a \in \Sigma$ and this edge is not part of any path from v_1 to v_g then by Lemma 7.4.6, T can remove this edge and 2*b* holds, so we are done.

If all edges in $(V, \mathcal{E})$ belong to a path from v_1 to v_g, from 1, there are two cases: either there is only one, or there is more than one path from v_1 to v_g.

In the first case (only one path) (v_0, v_1) can be chosen since it cannot be part of the *unique* path from v_1 to v_g. Assume now that there is more than one path from v_1 to v_g. Let $p = (v_1, v_2, \ldots, v_g)$ be the/a shortest path from v_1 to v_g. This path cannot contain any loops. Then, from this path take v_i such that i is the smallest index for which it holds that from v_i there is a path $(v_i, v'_{i+1}, \ldots v_g)$ to v_g that is at least as long as the path following p from v_i (i.e., $(v_i, v_{i+1}, \ldots, v_g)$). Intuitively, when following path p from v_1 to v_g, v_i is the first point at which one can deviate from p in order to take another path to v_g (recall that we consider the case where every vertex in the graph is part of a path from v_1 to v_g). Now it is possible for T to choose $(v_i, v'_{i+1}) \in \mathcal{E}_a$. Let $\mathcal{E}'$ be the resulting set of edges after removing (v_i, v'_{i+1}) from $\mathcal{E}_a$. Then we are in the game $\langle V, \mathcal{E}', v_1, v_g \rangle$. Note that due to the way we chose the edge to be removed, in the new graph it still holds that from v_0 there is no path to a vertex from which v_g is not reachable (this holds because from v_i the goal v_g is still reachable). Then by Lemma 7.4.5, T can win $\langle V, \mathcal{E}', v_1, v_g \rangle$, which then implies 2*b*.

Hence, we conclude that 2*b* is the case and thus using the inductive hypothesis, T can win $\langle V, \mathcal{E}, v_0, v_g \rangle$ also by removing an edge in every round. □

Corollary 7.4.8. *Teacher has a* `SLG*hu`*-winning strategy in* $\langle V, \mathcal{E}, v_0, v_g \rangle$ *iff she has a* `SLGhu`*-winning strategy.*

As the reader might have noticed, the result that if Teacher can win a `SLG*hu`, then she can also win the corresponding `SLGhu`, relies on the fact that Learner is the first to move. For instance, in a graph with two vertices and and one edge — leading from the initial vertex to the goal vertex — if Teacher was to move first, she can win the `SLG*hu` only by skipping the move.

Finally, let us consider the case of helpful Teacher and eager Learner.

Theorem 7.4.9. *If Learner and Teacher have a joint* `SLG*he`*-winning strategy in* $\langle V, \mathcal{E}, v_0, v_g \rangle$*, then they have a joint* `SLGhe`*-winning strategy*

Proof. If the players have a joint `SLG*he`-winning strategy, then there is an acyclic path from v_0 to v_g, which L can follow. At each round, T has to remove the edge that has just been used by L. □

Let us briefly conclude this section. We have shown that allowing Teacher to skip moves does not change the winning abilities of the players. Using these results, both the complexity and definability results from the previous section also apply to their versions without strict alternation, in which Teacher can skip a move.

7.5 Conclusions and Perspectives

We have provided a game theoretical approach to learning that takes into account different levels of cooperativeness between Learner and Teacher in a game of perfect information based on sabotage games. Because of our new interpretation we were able to define sabotage learning games with three different winning conditions. Then, following the strategy of Löding & Rohde (2003b), we have shown how sabotage modal logic can be used to reason about these games and, in particular, we have identified formulae of the language that characterize the existence of winning strategies in each of the two remaining cases. We also provided complexity results for the model-checking problem of these formulae. Our complexity results support the intuitive claim that cooperation of agents facilitates learning. Moreover, in our framework it turns out to be as difficult to decide whether a Teacher can force an unwilling Learner to learn as it is to decide whether an eager Learner can learn in the presence of an unhelpful Teacher.

Viewed from the perspective of the standard sabotage games, our complexity result for the game between a helpful Teacher and an unwilling Learner means that also with a *safety* winning condition, sabotage games are PSPACE-complete.

From the game-theoretical perspective, sabotage learning games can be extended to more general scenarios by relaxing the strict alternation. The moves of the players are of a different nature. Learner's moves can be seen as internal ones, moving to a state that is reachable *from the current one*, while Teacher's moves can be interpreted externally, removing *any* edge of the underlying graph. Once this asymmetry is observed, it becomes natural to ask what happens if from time to time Learner's move is not followed by Teacher's move (e.g., Learner can perform several changes of his information state before Teacher makes a restriction). Our results of Section 7.4 show that if we allow Teacher to skip a move, the winning abilities of the players do not change with respect to the original versions of the games. In the case of helpful Teacher and unwilling Learner, the result is quite surprising since it says that if Teacher can force Learner to learn in the game with non-strict alternation, then she can also do it when she is forced to remove edges in each round. This result crucially depends on the fact that Learner is the first to move, and does not hold in case Teacher starts the game.

In this chapter, we have described the learning process purely as *changes in information states*, without going further into their epistemic and/or doxastic interpretation.

We understand successful learning as the ability to reach an appropriate information state, not taking into account what happens afterwards. Formal learning theory that treats the inductive inference type of learning situates our work close to the concept of *finite identification* (Mukouchi, 1992) treated extensively in previous chapters. In particular, we are not concerned with the stability of the resulting state. *Identification in the limit* (Gold, 1967) extends finite identification by looking beyond reachability in order to describe 'ongoing behavior'. Fixed-point logics, such as the modal μ-calculus (Kozen, 1983; Scott & Bakker, 1969), can provide us with tools to express this notion of learnability. Further work involves investigating how fixed points can enrich sabotage-based learning analysis.

Moreover, in natural learning scenarios, e.g., language learning, the goal of the learning process is concealed from Learner. An extension of the framework of randomized sabotage games (Klein, Radmacher, & Thomas, 2009) could then be used to model the interaction between Learner and Teacher.

Chapter 8

The Muddy Scientists

Imagine you are one of ten prisoners locked up for extensive use of logic. To make you even more miserable, the guard comes up with a puzzle. He gathers all ten of you and says: 'Each of you will be assigned a random hat, either black or white. You will be lined up single file where each can see the hats in front of him but not behind. Starting with the prisoner in the back of the line and moving forward, you must each, in turn, say only one word which must be 'black' or 'white'. If the word you uttered matches your hat color you are released, if not, you are killed on the spot. You have half an hour to pray for your life.' Then he leaves. One of the prisoners says: 'I have a plan! If you agree on it, 9 of us 10 will definitely survive, and the remaining one has a 50/50 chance of survival.' What does he have in mind?

Most probably the strategy that he wants to implement is as follows. First, the prisoners have to agree on the following meaning of the utterance of the one who is the last in the line. If he says 'white', it means that he sees an even number of white hats in front of him. If he says 'black' it means that he sees an odd number of white hats in front of him. Hence, his utterance has nothing to do with what he thinks his own hat is. There is a 50/50 chance of the total number black hats being odd or even, and a 50/50 chance of his hat being black or white, so his chance of survival is the same. However, after this utterance the prisoner that stands in front of him knows for sure the color of his hat—he compares the utterance of his predecessor with the number of white hats he sees in front of him. If the parity is the same, he concludes that his hat is black, otherwise it is white. He makes his guess aloud. Now the person in front of her takes into account the first announcement and the second utterance, sees the number of white hats in front of her, and now she is also certain about her hat's color, etc.

This epistemic scenario shows the power of multi-agent information exchange. A very simple *quantitative* public announcement carries powerful *qualitative* information. Agents can easily deduce nontrivial facts from implicit and indirect information. Obviously, the information has to be relevant to make certain deductions possible. For example, in the above scenario announcing 'At least 5 hats are white' is not as effective as announcing the parity.

In this chapter we deal with the question of what makes such announcements relevant in epistemic context. We will be concerned with the simpler multi-agent scenario of the so-called Muddy Children puzzle, where, in contrast to the above-described Top-Hat puzzle, each agent has information about the state of all other agents.

8.1 The Muddy Children Puzzle

Yet another thought experiment—you are now out of prison, visiting a relative, who has three children[1]. While you are having coffee in the living-room, the kids are playing outside. When they come back home, their father says:

(1) At least one of you has mud on your forehead.

Then he asks them:

(**I**) Can you tell for sure whether you have mud on your forehead? If yes, step forward and announce your status.

Each child can see the mud on others but cannot see his or her own forehead. Nothing happens. The father insists—he repeats (**I**). Still nothing. But after he repeats the question for the third time suddenly all muddy children know that they have mud on their forehead. You are quite amazed by their telepathic skills. You ask yourself—how is that possible?

A considerable amount of philosophical and logical literature has been devoted to describe the epistemic phenomena that make such 'miraculous' inferences work. The framework of dynamic epistemic logic allows a clear and comprehensive explanation of the underlying phenomena (see Van Ditmarsch et al., 2007; Gerbrandy, 1999a; Moses et al., 1986). The classical modeling of the Muddy Children puzzle has been explained in Chapter 2. The DEL representation is extensive—the size of the epistemic model is exponential with respect to the number of children. This seems to be essential, as children are in fact asked whether or not they *themselves* are muddy. Therefore, even the worlds in which the same number of children is muddy, e.g., $(w_2 : m_a, \neg m_b, m_c)$ and $(w_3 : m_a, m_b, \neg m_c)$ have to be distinguished.

The similarity between the Muddy Children puzzle and the Top-Hats problem is striking: in both cases agents need to reason about their properties on the basis of some general quantitative statement; the settings differ with respect to the observational power of the agents. It looks like the possibility of convergence to knowledge in such problems depends on the trade-off between the internal structure of epistemic information and the amount of information provided by

[1] We assume that it is commonly know among them that they are truthful and that they are perfect logical reasoners.

the public announcement. To see these differences in full light let us consider the following two cases:

- The Top-Hats puzzle: announcing ‘an even number of hats are white’ allows epistemic reasoning that solves the puzzle for any configuration; announcing ‘at least one hat is black’ allows solving the problem only in a very limited number of cases.
- The Muddy Children puzzle: announcing ‘at least one of you has mud on your forehead’ allows epistemic reasoning that solves the puzzle for any configuration (see Van Ditmarsch et al., 2007), while announcing parity leads to an immediate one-step solution that does not involve any epistemic reasoning.

Hence, it is fair to say that in some sense parity announcements bring more information than existential announcements, at least with respect to the above-mentioned epistemic frameworks. In this chapter we study the informational content of various announcements in an epistemic context. We will generalize the Muddy Children problem and consider the ‘muddy inferences’ as depending on this kind of background information (public announcement).

8.2 Muddy Children Generalized

Let us recall the first announcement made by the father:

1. **At least one** of you has mud on your forehead.

Sentence (1) can be seen as a *background assumption* that makes the epistemic multi-agent inferential process possible. Unlike other assumptions of the puzzle (e.g., truthfulness of the agents, their perfect reasoning skills, etc.), it is factual, external with respect to the agents and, as such, explicitly present in the formulation of the puzzle. It also has a different status than the events taking place after the announcement. In the modeling explained in Chapter 2 the first announcement and next epistemic updates have the same status. The implicit idea is that the agents first construct the exhaustive representation of the situation, then they modify the model according to the background assumption and after that proceed to performing the updates of their epistemic reasoning. In fact the question about the epistemic states of the agents is asked *after* the background assumption has been introduced. The quantifier announcement prepares the ground for epistemic reasoning, and enforces some structure on the situation.

The background information has the form of a simple quantifier sentence, with the quantifier ‘At least one’. However, as we have already seen the background assumption can be altered with the use of different quantifiers, e.g., ‘at most three’, ‘an even number of’, etc. We will have a look at various quantifiers in the background assumption of the Muddy Children puzzle, investigate their multi-agent inferential power and associate the latter with some properties of generalized quantifiers.

8.2.1 Generalized Quantifiers

The information provided by the father in the Muddy Children scenario has the following form:

$$Q \text{ of you have mud on your forehead.}$$

As a matter of fact, Q may be substituted by various quantifiers, like 'At least one', 'An even number', 'One third' and so on. Of course not all quantifiers guarantee the convergence to knowledge. As we said before, the informational power of an announcement depends on the quantifier. This is not very surprising—it is well understood in linguistics that expressivity of a language is to a huge extent determined by the employed quantifier constructions (see, e.g., Peters & Westerståhl, 2006). In principle, the Muddy Children situation can be modeled as $M = (U, A)$, where U is the set of children and $A \subseteq U$ is the set of children that are muddy. Of course after father's announcement some models are no longer possible. Only those satisfying the quantifier sentence, i.e., $M \models Q_U(A)$, should be considered. Therefore, the model of a given Muddy Children scenario consists of structures satisfying the quantifier sentence. The agent's goal is to pinpoint one of them—the actual world. To explain this idea in more detail let us start with introducing the notion of generalized quantifiers.

Definition 8.2.1 (Mostowski 1957)**.** *A generalized quantifier Q of type (1) is a class of structures of the form $M = (U, A)$, where A is a subset of U. Additionally, Q is closed under isomorphism, i.e., if M and M' are isomorphic*[2]*, then $(M \in Q \iff M' \in Q)$.*

Let us give some examples of generalized quantifiers in the context of Muddy Children scenarios. The classical Muddy Children puzzle with the father saying 'At least one of you has mud on your forehead' involves the existential generalized quantifier:

$$\exists = \{(U, A) : A \subseteq U \ \&\ A \neq \emptyset\}.$$

The variation with the father using the cardinal quantifier 'Exactly m of you...' gives rise to the following class of models:

$$\exists^{=m} = \{(U, A) : A \subseteq U \ \&\ |A| = m\}.$$

Furthermore, the father may know the Top-Hat puzzle and use a divisibility announcement of the form 'A number divisible by k of you...'. This situation

[2] Two models M and M' are isomorphic iff there is a bijective map f between M and M' that preserves their structure.

can be captured by divisibility quantifiers:

$$\mathsf{D_k} = \{(U, A) : A \subseteq U \;\&\; |A| = k \times n\}, \text{where } n \in \mathbb{N}.$$

Finally, the father may say 'Most of you. . . '. Then the children need to pick a model from the following class:

$$\mathsf{Most} = \{(U, A) : A \subseteq U \;\&\; |A| > |U - A|\}.$$

Isomorphism closure One of the relevant properties of generalized quantifiers is that they are *closed under isomorphism* (see Definition 8.2.1). This means that our public announcements are logical in the sense that they cannot distinguish between isomorphic models. This leads to a theoretical discomfort—if generalized quantifiers cannot distinguish between the situation in which only Ann is muddy and the situation in which only Bob is muddy, it might be impossible to use this tool to analyze the epistemic situation of the Muddy Children puzzle—somehow the agents are able to distinguish qualitatively between the models. In their epistemic reasoning they can identify the particular world they are in. The additional power stemming from their observations and their epistemic skills makes such a solution possible.

Number triangle Isomorphism closure gives rise to so-called *number triangle* representation of quantifiers (see Van Benthem, 1986). Every element of a generalized quantifier of type (1) may be represented as a pair of natural numbers (k, n), where $k = |U - A|$ and $n = |A|$. In other words the first number stands for the cardinality of the complement of A and the second number stands for the cardinality of A. The following definition gives a formal counterpart of this notion.

Definition 8.2.2. *Let Q be a type* (1) *generalized quantifier. For any numbers $k, n \in \mathbb{N}$ we define a* quantifier relation Q*:*

$$\mathsf{Q}(k, n) \iff \textit{there are } U,\ A \subseteq U \textit{ such that}$$
$$|U| = n + k, |A| = n, \textit{ and } Q_U(A).$$

Proposition 8.2.3. *If Q is a type* (1) *generalized quantifier, then for all U and all $A \subseteq U$ we have:*

$$Q_U(A) \iff \mathsf{Q}(|U - A|, |A|).$$

If we restrict ourselves to finite universes, we can represent all that is relevant for type (1) generalized quantifiers in the structure called number triangle. This construct simply enumerates all finite models of type (1). The node labeled (k, n) stands for a model in which $|U - A| = k$ and $|A| = n$. Now, every generalized quantifier of type (1) can be represented by putting '+' at those (k, n) that belong to Q and '–' at the rest. For example, the quantifier 'At least one' in number triangle representation is shown in Figure 8.1. This handy representation will play a crucial role in our investigations.

```
                (0,0)                                  —
            (1,0)  (0,1)                            −     +
        (2,0)  (1,1)  (0,2)                      −     +     +
    (3,0)  (2,1)  (1,2)  (0,3)                −     +     +     +
(4,0)  (3,1)  (2,2)  (1,3)  (0,4)          −     +     +     +     +
.................................          .........................
```

Figure 8.1: Number triangle and the representation of 'At least 1'

Monotonicity An important intuition about some natural language quantifiers is that they say that some sets are 'large enough'. We would expect that quantifiers are closed under some operations that change the size of those sets. The simplest among such operations is the one of taking subsets and supersets. Intuitively, monotonicity is about quantifiers being closed under these operations. The possible announcements of the father may differ with respect to this factor.

Definition 8.2.4. *A generalized quantifier Q of type* (1) *is said to be* upward monotone (increasing)*, just in case for any two sets X and Y, if X is a subset of Y, then $Q_U(X)$ entails $Q_U(Y)$ for every U.*

For example, the quantifier 'every child' is monotone increasing. For example, the first sentence below entails the second as the set of children that have mud on the forehead is a subset of children that are dirty.

1. Every child has mud on the forehead.
2. Every child is dirty.

Definition 8.2.5. *A generalized quantifier Q of type* (1) *is said to be* downward monotone (decreasing)*, just in case for any two sets X and Y, if X is a subset of Y, then $Q_U(Y)$ entails $Q_U(X)$ for every U.*

A quantifier Q is said to be downward monotone (decreasing) if the entailment holds in the other direction. An example of a monotone decreasing quantifier is 'no child' as the first sentence below entails the second:

1. No child is dirty.
2. No child has mud on the forehead.

Similarly a kind of monotonicity can be defined for the complement of the predicate. This property is called extension and guarantees that if a given finite model satisfies a quantifier, then extending $U - A$ cannot change this.

Definition 8.2.6. *A quantifier Q of type* (1) *satisfies* extension *iff for all models M and M', with the universes U and U', respectively: $A \subseteq U \subseteq U'$ implies $Q_U(A) \Rightarrow Q_{U'}(A)$.*

It is not surprising that monotonicity is one of the key properties of quantifiers, both in logic and linguistics. In model theory it contributes to definability (see, e.g., Väänänen, 2002); in linguistics it is used, among other applications, to explain the phenomenon of negative polarity items (see, e.g., Ladusaw, 1979). Moreover, there are good reasons to believe that it is a crucial feature for processing natural language quantifiers, as has already been suggested by psychologists, e.g., Johnson-Laird (1983) as well as linguists and logicians, e.g., Barwise & Cooper (1981), and empirically supported by Geurts (2003). There are also strong links with learnability of quantifiers (see Clark, 2010; Gierasimczuk, 2007, 2009b; Tiede, 1999).

CE quantifiers It is relevant in the context of Muddy Children puzzle to introduce the notion of such a quantifiers.

Definition 8.2.7 (Mostowski 1957)**.** *A* generalized quantifier Q *of type* (1,1) *is a class of structures of the form* $M = (U, A, B)$*, where* A, B *are subsets of* U*. Additionally,* Q *is closed under isomorphism, i.e., if* M *and* M' *are isomorphic, then* $(M \in Q \iff M' \in Q)$*.*

One may argue that the quantifiers used in the puzzle, as many other natural language quantifiers, are in fact of type (1,1) as they bind the two predicates 'children' and 'muddy'. This leads to the formalization of the form 'Q of the CHILDREN are MUDDY'. We would have then two sets A and B—the first stands for the set of children, the second for the set of muddy objects. Our considerations in this chapter explicitly concern quantifiers of type (1), but they also account for (1,1) quantifiers that satisfy certain additional restrictions. So-called CE-quantifiers (Väänänen, 2002) are those quantifiers of type (1,1) that, in addition to the isomorphism closure, satisfy the properties of extension and conservativity:

Definition 8.2.8. *Let* $M = (U, A, B)$ *and* $M' = (U', A', B')$ *be finite models, and* $A, B \subseteq U \subseteq U'$*:*

1. Q *of type* $(1, 1)$ *satisfies* extension *iff* $Q_U(A, B)$ *implies* $Q_{U'}(A, B)$*.*
2. Q *of type* $(1, 1)$ *is* conservative *iff* $Q_U(A, B)$ *iff* $Q_U(A, A \cap B)$*.*

Those properties quite intuitively capture natural language quantifier constructions, and have been even proposed as natural language universals (Barwise & Cooper, 1981). Also, they reduce quantifiers to being dependent only on two cardinalities: $|A - B|$ and $|A \cap B|$. Therefore, the relevant finite models can again be represented as pairs of integers, and such quantifiers can be depicted in the number triangle (see Väänänen, 2002).

8.3 Quantifiers as Background Assumptions

In this section we will investigate the behavior of various generalized quantifiers in the multi-agent inferential situation of the Muddy Children puzzle. We will investigate its *solvability*. We will say that a Muddy Children situation is 'solvable' if it entails the possibility of all agents converging to knowledge about their own status in a finite number of steps.

8.3.1 Increasing Quantifiers

Let us first consider a class of quantifiers that is closest to the classical version of the Muddy Children puzzle, namely: 'At least m', where $m \in \mathbb{N}$. In the puzzle this takes the form of the announcement:

At least k of you have mud on your foreheads.

Quantifiers of this form are monotone increasing and satisfy extension—once the quantifier is true in a model, adding new elements to A or $U - A$ will not change its logical value.[3] The number triangle representation gives us always a downward triangle starting in a point $(0, m)$.

How do those quantifiers behave in the Muddy Children situation? In Table 8.1 rows are labeled with the total number of children and columns with the number of muddy children. The number at the coordinates (c, m) says how many steps are needed to solve the Muddy Children puzzle for the muddy children (immediately after all children know their status).

	0	1	2	3	4	5
1	x	1	x	x	x	x
2	x	1	2	x	x	x
3	x	1	2	3	x	x
4	x	1	2	3	4	x
5	x	1	2	3	4	5
6	x	1	2	3	4	5 …

	0	1	2	3	4	5
1	x	x	x	x	x	x
2	x	x	1	x	x	x
3	x	x	1	2	x	x
4	x	x	1	2	3	x
5	x	x	1	2	3	4
6	x	x	1	2	3	4 …

Table 8.1: 'At least one' and 'At least two'

The numbers of steps needed to solve the puzzle form a triangle, with the values increasing horizontally to the right. When increasing the parameter k in the quantifier 'At least k' the whole triangle simply moves to the right and downwards.

[3] In the case of (1,1) quantifiers this property corresponds to upward monotonicity in the left argument, which is also called *persistence* (see Peters & Westerståhl, 2006).

From the conceptual analysis we can extract some more information about the structure of the solutions. In Table 8.2 a number is superscripted by '+' if in the final step of the reasoning some children infer their status from other children's epistemic announcements (from other children saying that they already know). The lack of the plus means that the final step of the reasoning is drawn simultaneously by all agents.

	0	1	2	3	4	5
1	x	1	x	x	x	x
2	x	1^+	2	x	x	x
3	x	1^+	2^+	3	x	x
4	x	1^+	2^+	3^+	4	x
5	x	1^+	2^+	3^+	4^+	5
6	x	1^+	2^+	3^+	4^+	5^+ …

	0	1	2	3	4	5
1	x	x	x	x	x	x
2	x	x	1	x	x	x
3	x	x	1^+	2	x	x
4	x	x	1^+	2^+	3	x
5	x	x	1^+	2^+	3^+	4
6	x	x	1^+	2^+	3^+	4^+ …

Table 8.2: 'At least one' and 'At least two'

Now we are ready to give a general fact about this type of background assumption.

Proposition 8.3.1. *Let us take a Muddy Children situation, with n the number of children, $m \leq n$ the number of muddy children. The Muddy Children puzzle with the background assumption 'At least k of you have mud on your forehead' can be solved in $m - (k - 1)$ steps, where $k \leq m$.*

Using a similar background assumption with inner negation

At least k of you do not have mud on your foreheads

also makes the puzzle solvable. 'At least k not' behaves as 'At least k', but depends on the number of clean children. In general, inner negation works this way for other quantifiers.

A simplifying observation about a similar class of upward monotone quantifiers that satisfy extension is as follows:

Observation 8.3.2. *Let us take a Muddy Children situation, with n the number of children, $m \leq n$ the number of muddy children. The Muddy Children puzzle with the background assumption 'More than k of you have mud on your forehead' can be solved in $m - k$ steps, where $k \leq m$.*

8.3.2 Decreasing Quantifiers

Let us now consider another natural class, downward monotone quantifiers that satisfy extension: 'At most k', where $k \in \mathbb{N}$. In the puzzle this takes form of the announcement:

At most k of you have mud on your foreheads.

Like in the case of increasing quantifiers we provide a table with the numbers of steps needed for solving the puzzle in respective cases. By doing this we indicate how the situation changes with the parameter k. In Table 8.3 we can observe

	0	1	2	3	4
1	?	?	x	x	x
2	2	1^+	x	x	x
3	2	1^+	x	x	x
4	2	1^+	x	x	x
5	2	1^+	x	x	x …

	0	1	2	3	4
1	?	?	x	x	x
2	?	?	?	x	x
3	3	2^+	1^+	x	x
4	3	2^+	1^+	x	x
5	3	2^+	1^+	x	x …

Table 8.3: ‘At most one’ and ‘At most two’

that the numbers of steps needed to solve the puzzle form a block, with the values increasing horizontally to the left. When increasing the parameter k in the quantifier ‘At least k’ the whole blocks moves to the right and downwards revealing the next column on the left. Also, in case when the parameter k in the quantifiers is larger or equal to the number of muddy children, the puzzle is not solvable! When the block of numbers moves downward together with k, it leaves a trace consisting of question marks behind that correspond to the unsolvable situations.

We can characterize the solvability of Muddy Children situations with the quantifier ‘At most k’ in the following way.

Proposition 8.3.3. *Let us take a Muddy Children scenario, with n the number of children, $m \leq n$ the number of muddy children. If $n > k$ then the Muddy Children puzzle with the background assumption ‘At most k of you have mud on your forehead’ can be solved in $(k + 1) - m$ steps. If $n \leq k$ the situation is not solvable.*

Like in the previous case we give a simplifying observation about a similar class of downward monotone quantifiers that satisfy extension:

Observation 8.3.4. *Let us take a Muddy Children situation, with n the number of children, $m \leq n$ the number of muddy children. The Muddy Children puzzle with the background assumption ‘Less than k of you have mud on your forehead’ can be solved in $k - m$ steps, where $k \leq m$. If $m < k$ the situation is not solvable.*

8.3.3 Cardinal and Parity Quantifiers

Some kinds of quantifiers allow one-step immediate solvability for all agents. Taking into consideration what they already know, the announcement gives them full certainty about their state. This takes place for example when the number of muddy children is explicitly announced with the use of the quantifier ‘Exactly k’, where $k \in \mathbb{N}$. The announcement of:

Exactly k of you have mud on your foreheads

always leads to immediate answers (see Table 8.4).

	0	...	k	k+1	...
1	x	x	1	x	x
2	x	x	1	x	x
3	x	x	1	x	x
4	x	x	1	x	x
5	x	x	1	x	x ...

Table 8.4: ‘Exactly k’

Proposition 8.3.5. *Every Muddy Children scenario with a background assumption of the form ‘Exactly k’ is solvable in* 1 *step.*

There are other, more interesting quantifiers with this property, e.g., divisibility quantifiers: ‘A number divisible by k’, where $k \in \mathbb{N}$. An example of such an announcement for $k = 2$ is:

An even number of you have mud on your foreheads.

In Table 8.5 you can see that the columns that include solvable scenarios are isolated and consists only of 1s. Moreover, if the number k in the quantifier ‘Divisible by k’ increases the gaps between the columns.

	0	1	2	3	4
1	1	x	1	x	1
2	1	x	1	x	1
3	1	x	1	x	1
4	1	x	1	x	1
5	1	x	1	x	1 ...

	0	1	2	3	4	5	6
1	1	x	x	1	x	x	1
2	1	x	x	1	x	x	1
3	1	x	x	1	x	x	1
4	1	x	x	1	x	x	1
5	1	x	x	1	x	x	1 ...

Table 8.5: ‘Even’ and ‘Divisible by 3’

A relevant fact is as follows.

Proposition 8.3.6. *Let us take a Muddy Children scenario. The Muddy Children puzzle with the background assumption ‘The number of you that have mud on your forehead is ℓ mod k’, for any $\ell, k \in \mathbb{N}$, can be solved in* 1 *step.*

Additionally, it is worth noting that provided that every agent sees all other agents, and in fact the number of muddy children is $k \times m$, the announcement of quantifier ‘A number divisible by k’ is equivalent to announcing ‘Exactly $k \times m$’.

All and No There are two simple natural language quantifiers that also allow one step immediate solvability for all agents. Quantifiers ‘All’ and ‘No’ inform all the agents directly about the status of all agents, and therefore their own. Trivially, the announcement of:

All of you have mud on your foreheads

or

None of you have mud on your foreheads

always leads to immediate answers (see Table 8.6).

	0	1	2	3	4
1	x	1	x	x	x
2	x	x	1	x	x
3	x	x	x	1	x
4	x	x	x	x	1
5	x	x	x	x	x …

	0	1	2	3	4	5
1	1	x	x	x	x	x
2	1	x	x	x	x	x
3	1	x	x	x	x	x
4	1	x	x	x	x	x
5	1	x	x	x	x	x …

Table 8.6: ‘All’ and ‘No’

Proposition 8.3.7. *Every Muddy Children scenario with a background assumption of the form ‘All’ and ‘No’ is solvable in* 1 *step.*

8.3.4 Proportional Quantifiers

Proportional quantifiers indicate the ratio between the number of elements in the predicate and the total number of elements. The first that comes to mind is ‘Exactly $\frac{1}{k}$’, where $k \in \mathbb{N}$. Update with this information will be survived by cardinalities that are divisible by k. In those situations, where $|A| = k \times \ell$, for some $\ell \in \mathbb{N}$, it is equivalent to the cardinal quantifier ‘Exactly ℓ’ (see previous section). However, there are also more interesting cases of upward monotone proportional quantifiers. Such class is, e.g., ‘More than $\frac{1}{k}$’, where $k \in \mathbb{N}$. An example of such announcement could be:

Most of you have mud on your foreheads.

If we agree to interpret ‘Most’ as ‘More than half’, then the solvability of the Muddy Children puzzle with this quantifier is depicted in on the left in Table 8.7. The table on the right shows the pattern for the quantifier ‘More than one third’.

The patterns in Table 8.7 might at first sight seem complex, but as a matter of fact it is quite easy to observe that the pattern consists of smaller parts resembling simple increasing quantifiers that satisfy extension (see Section 8.3.1). In fact these muddy situations are reducible to those given by quantifiers ‘More than k’.

	0	1	2	3	4	5
1	x	1	x	x	x	x
2	x	x	1	x	x	x
3	x	x	1^+	2	x	x
4	x	x	x	$\mathbf{1^+}$	**2**	x
5	x	x	x	$\mathbf{1^+}$	$\mathbf{2^+}$	**3**
6	x	x	x	x	1^+	2^+ …

	0	1	2	3	4	5
1	x	1	x	x	x	x
2	x	1	2	x	x	x
3	x	x	$\mathbf{1^+}$	**2**	x	x
4	x	x	$\mathbf{1^+}$	$\mathbf{2^+}$	**3**	x
5	x	x	$\mathbf{1^+}$	$\mathbf{2^+}$	$\mathbf{3^+}$	**4**
6	x	x	x	1^+	2^+	3^+ …

Table 8.7: ‘More than half’ and ‘More than one third’

In a given situation, when $|U| = n$, ‘More than $\frac{1}{k}$’ is of course equivalent to ‘More than $\frac{n}{k}$’. This can be also observed in Table 8.7, the examples of ‘intervals’ of the kind described in the proposition are printed bold.

Observation 8.3.8. *The Muddy Quantifier ‘More than $\frac{1}{k}$’ consists of intervals $q_0, q_1, \ldots$ such that:*

1. *q_0 consists of $k - 1$ rows in the table, and for $i > 0$, q_i consists of k rows.*
2. *q_i is the segment of size k of the table for ‘More than i’ starting in the i-th row.*

The number of steps needed to solve this puzzle is then characterized in the following way.

Proposition 8.3.9. *Let us take a Muddy Children situation, with n the number of children, $m \leq n$ number of muddy children. The Muddy Children puzzle with the background assumption ‘More than $\frac{1}{k}$ of you have mud on your forehead’ can be solved in $\lceil m - \frac{n}{k} \rceil$ steps.*

All above-mentioned propositions lead to a uniform perspective on quantifier announcements in the Muddy Children puzzle. In the next sections we will describe the epistemic reasoning using the tools of generalized quantifier theory.

8.4 Iterated Epistemic Reasoning

The number of steps needed to solve the puzzle seems to be pretty arbitrary. It is, however, clearly determined by the epistemic structures that can be defined on the basis of the number triangle. In order to understand how this works, let us have a look at a concrete Muddy Children scenario. Let us assume that we have three agents a, b, and c. All possibilities with respect to the size of the set of muddy children are enumerated in the third level of the number triangle. Let us also assume at this point that the actual situation is that agents a, b are muddy and c is clean. Therefore, with respect to our representation the real world is $(1, 2)$, one child is clean and two are muddy:

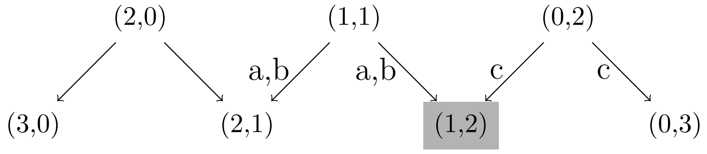
(3,0) (2,1) (1,2) (0,3)

Now, let us focus on what the agents can observe. Agent a sees one muddy child and one clean child. The same holds for agent b. Therefore their observational state can be encoded as (1,1). Accordingly, the observational state of c is (0,2). In general, if the number of agents is n, each agent can observe $n-1$ agents. Therefore, in our case what agents observe is in the second level of the number triangle. So, in order to model what an agent sees, and what the actual state can be, we need levels 2 and 3. Level 2 includes states that encode all possible observations of facts in level 3. Level 3 lists facts—observations extended by the state of the respective agent.

(2,0) (1,1) (0,2)

(3,0) (2,1) (1,2) (0,3)

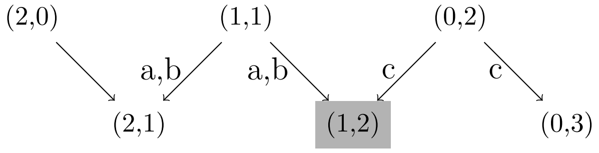
Now, the question that each of the agents is facing is whether they are muddy or not. For example, agent a has to decide whether he should *extend* his observation state, $(1,1)$, to the left state $(2,1)$ (a decides that he is clean) or to the right state $(1,2)$ (a decides that he is muddy). The same holds for agent b. The situation of agent c is similar, his observational state is $(0,2)$ and it has two potential extensions $(1,2)$ and $(0,3)$. In general, we will say that every observational state has two possible successors.



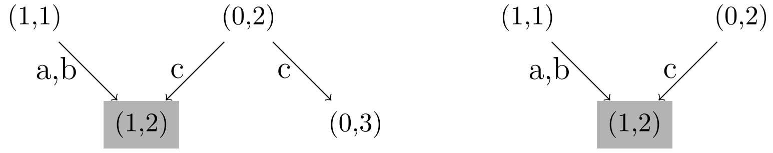
Given this representation, we can now analyze what happens in the Muddy Children situation. First, the announcement is given:

At least one of you is muddy.

According to the number triangle representation, this allows eliminating those factual states that represent finite models that are not in the quantifier. In this case it is $(3,0)$. The new model is as follows.



The father asks: 'Can you tell for sure whether or not you have mud on your forehead?' In our graph, this question means: 'Do any of you have only one successor?' All agents know that $(3, 0)$ has just been eliminated. Agent a considers it possible that the actual state is $(2, 1)$, i.e., that two agents are clean and one is muddy, so that he himself would have to be clean. But then he knows that there would have to be an agent whose observational state is $(2, 0)$—there has to be a muddy agent that observes two clean ones. For this hypothetical agent the uncertainty disappeared just after quantifier announcement (for $(2, 0)$ there is only one successor left). So, when it becomes clear that no one knows and the father asks the question again, the world $(2, 1)$ gets eliminated and the only possibility for agent a is now $(1, 2)$ via the right successor, and this indicates that he has to be muddy. Agent b is in exactly the same situation. They both can announce that they know. And since c witnessed the whole process he knows that the only way for them to know was to be in $(1, 1)$ and decide on $(1, 2)$. So, *on the basis of their announcement of knowledge*, c also knows that he is supposed to take the left successor, to arrive at the right conclusion—c is clean.



This epistemic reasoning took two steps. If the actual world was $(2, 1)$ some agent's observation would be $(2, 0)$, and this agent would know his status after the first announcement, and the rest of the agents would follow. Accordingly, for $(0, 3)$ this would have taken three steps. This can be summed up in the following way: the quantifier breaks the perfect 'uncertainty structure' of the model, and the farther the actual state is from this break, the longer it takes to solve the puzzle.

Epistemic Modeling Let us now generalize the situation analyzed in the previous section. First let us formalize the intuition levels: observational and factual, and the nature of agents.

Levels In general, if there are n agents, we take the nth level of the triangle, i.e., finite models with $|U| = n$, enumerating all possible settings (up to isomorphism). This level will be called the *factual level.* It constitutes the uncertainty domain of the children. We can say that the children's uncertainty is captured by the variety of models belonging to the generalized quantifier formalizing the background assumption. Moreover, in the puzzle every child sees all other children, but not himself, so every possible observation consists of $n - 1$ children. Therefore, level $n - 1$ of the number triangle can be interpreted as enumerating every possible observation of the children. We will call it the *observational level.* Each observation can be extended to one of the two factual states that are the closest below—to the left if the observer in question is clean or to the right if he is muddy.

Agents There can be any number of Muddy Children—the puzzle is a real *multi*-agent scenario. This is why it is so surprising to realize that it can be collapsed to two agents, i.e., two local states. Obviously every agent is in one of the two groups: either among the muddy children or among the clean ones. But this gives much more—in fact there are at most two possible observational states that the agents might be in. Every clean child observes the same as all other clean children, and every muddy child observes the same as all other muddy children.

Proposition 8.4.1. *Every agent's observation is encoded by one of at most two states in the observational level. Those two are neighboring states.*

Proof. For the first part. Assume that the total number of children is n, the number of muddy children is m. Let us pick an agent and call him a. There are two possibilities:

1. a is muddy, then a's observational state is $(n - m, m - 1)$.
2. a is clean, then a's observational state is $((n - m) - 1, m)$.

The two relevant observational states neighbor each other in the model because they are the only two states that can be extended to the actual state $(n-m, m)$. □

Moreover, the actual world clearly determines the number of agents perceiving the same situation.

Proposition 8.4.2. *Given a certain situation (c, m) there are c children that are in the observational state $(c - 1, m)$ and m children in the observational state of $(c, m - 1)$. In case $m = 0$ all children are in the observational state $(c - 1, 0)$. In case $c = 0$ all children are in the observational state $(0, m - 1)$.*

8.4.1 An Epistemic Model Based on the Number Triangle

In this section we will link our modeling of the Muddy Children puzzle to the existing approaches based on dynamic epistemic logic. The correspondence is not straightforward. However, it is possible to formalize our models in a similar way.

Definition 8.4.3. *The* Muddy Children model *for n children is a quadruple $\mathcal{M}_n^{MC} = (S_o, S_f, R_m, R_{\bar{m}})$, where*

- $S_o = \{(c, m) \mid c + m = n - 1\}$ *(the observational states),*
- $S_f = \{(c, m) \mid c + m = n\}$ *(the factual states),*
- $R_m \subseteq S_o \times S_f$*, such that* $R_m((c_1, m_1), (c_2, m_2))$ *iff* $m_2 = m_1 + 1$*,*

- $R_{\bar{m}} \subseteq S_o \times S_f$, *such that* $R_{\bar{m}}((c_1, m_1), (c_2, m_2))$ *iff* $c_2 = c_1 + 1$.

In other words, the Muddy Children model is a two-row fragment of the number triangle with a double successor relation. An agent having access from an observational state (c, m) to the factual state $(c, m + 1)$ corresponds to the possibility that he is muddy. Every such two states are in the relation R_m. Analogously, for $R_{\bar{m}}$.

Let us observe that the size of such models is linear with respect to the number of children. To be precise:

Observation 8.4.4. *If* $n \in \mathbb{N}$ *is the number of children, then the Muddy Children model has* $2n + 1$ *states.*

This is a significant improvement with respect to the classical modeling in which an exponential number of states is required (cf. Van Ditmarsch et al., 2007; Fagin et al., 1995).

In this setting generalized quantifiers can be interpreted as propositional letters evaluated over the factual states of the Muddy Children model. For any generalized quantifier of type (1) we take a propositional letter q. Now, let $\mathcal{M}_n^{MC} = (S_o, S_f, R_m, R_{\bar{m}})$ be a Muddy Children model, and $(c, m) \in S_f$, then the semantics of q can be defined in the following way:

$$\mathcal{M}_n^{MC}, (c, m) \models q \text{ iff } (c, m) \in \mathsf{Q}.$$

Every variant of the Muddy Children puzzle comes with a quantifier that constitutes the background assumption of the puzzle. Therefore we can assume that the models get cut by the quantifier before the epistemic reasoning starts. This cut is in fact an *update* of the factual level of the Muddy Children model with a corresponding 'quantifier' letter q. Below we define what happens to the general Muddy Children model when a quantifier is introduced.

Definition 8.4.5. *Having the Muddy Children model* $\mathcal{M}_n^{MC} = (S_o, S_f, R_m, R_{\bar{m}})$ *and a generalized quantifier* Q *of type (1), we define the quantifier update of* $\mathcal{M}_n^{MC}$ *with the quantifier* Q *as resulting in the* Q-Muddy Children model $\mathcal{M}_n^{QMC} = (S'_o, S'_f, R'_m, R'_{\bar{m}})$ *in the following way:*

- $S'_o = \{(c, m) \mid (c, m) \in S_o \;\&\; (\mathsf{Q}(c + 1, m) \vee \mathsf{Q}(c, m + 1))\}$,
- $S'_f = \{(c, m) \mid (c, m) \in S_f \;\&\; \mathsf{Q}(c, m)\}$,
- $R'_m = R_m \restriction (S'_o \times S'_f)$,
- $R'_{\bar{m}} = R_{\bar{m}} \restriction (S'_o \times S'_f)$.

At this point let us observe that for some quantifiers Q the Q-Muddy Children models have a fixed size, independent of the number of children. Such a quantifier is for example 'At most k'. In all situations in which the number of children n is larger than k the puzzle is solvable. If fact, in this situation for any $n > k$, $|S'_f| = k$. This shows that some quantifier announcements leave behind them a fixed number of possible worlds.

The epistemic information can be expressed with formulae evaluated in the observational states. We can say that an agent in an observational state (c, m) knows that φ_m (that he is muddy) if and only if the only successor of (c, m) is $(c, m + 1)$. Moreover, if an epistemic announcement eliminates an observational state it also eliminates all its successors. As our modeling is Muddy-Children specific, we do not give here a full epistemic language. The semantic analysis of the announcements is aimed at emphasizing the analogy with so-called *unsuccessful announcements* in DEL (see Van Ditmarsch et al., 2007).

8.5 Muddy Children Solvability

By reinterpreting the Muddy Children puzzle within the semantics of quantifiers we can associate every finite model with the number of steps needed to solve the puzzle, if it is solvable at all.

Definition 8.5.1. *A* muddy quantifier *is a pair $Q^{MC} = (Q, f_Q)$, where Q is a quantifier and $f_Q : Q \to \mathbb{N}$ is a function that assigns to a pair of numbers representing $M \in Q$ the number of steps needed to solve the Muddy Children puzzle with the background assumption containing quantifier Q.*

Below we list some examples of muddy quantifiers in the number triangle representation according to the previously given enumeration of interesting cases.[4]

First let us consider the quantifier 'At least k'. It is easy to observe that increasing k causes the downward triangle to move down along the $(0, 0)$–$(0, n)$ axis.

This quantifier allows solving the Muddy Children puzzle for any configuration of 'muddiness'. However, within a certain level, the farther from a minus the longer it takes.

Now let us have a look at the quantifier 'At most k'.

The question-marks occur in place of models that satisfy the quantifier, but for which it is impossible to solve the Muddy Children puzzle. For example, if one child is clean and one child is muddy (the actual world is $(1, 1)$) the Muddy Children situation does not lead to a solution if the announcement is:

At most two of you are muddy.

[4] As before, the index + marks the solutions in which at least one agent infers his status from the announcement of knowledge of other agents. The lack of + marks the situations in which the agents discover their status simultaneously.

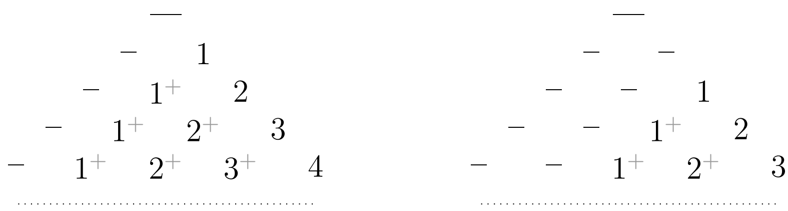


Figure 8.2: Increasing muddy-quantifiers ‘At least 1’ and ‘At least 2’

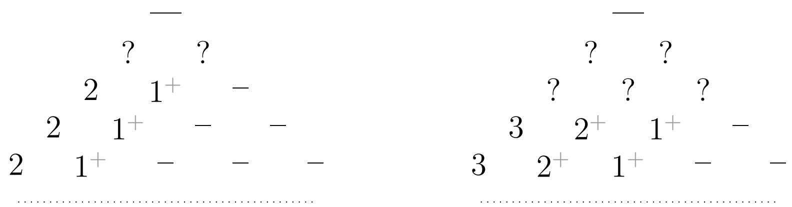


Figure 8.3: Decreasing muddy-quantifiers ‘At most 1’ and ‘At most 2’

Again, the farther from a minus the longer it takes to solve the puzzle.

Divisibility quantifiers in the Muddy Children setting do not involve much inference—every situation is solvable in one step (see Figure 8.4). Moreover, there is no ‘+’ superscript anywhere. This indicates that the answers are simultaneously given by all the agents.

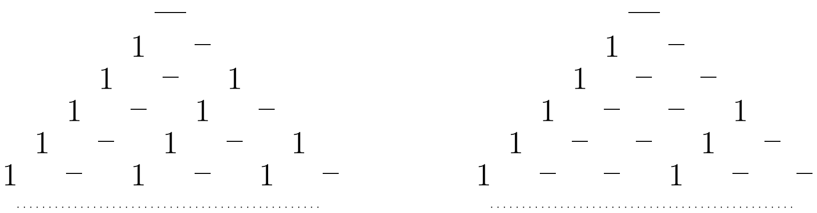


Figure 8.4: Muddy-quantifiers ‘Divisible by 2’ and ‘Divisible by 3’

Finally, let us have a look at upward monotone proportional quantifiers. They create more complicated patterns (see Figure 8.5).

Notice the similarity between these patterns and those of Figure 8.5. The number of steps increases to the right, together with the distance from a minus.

Function f_{Q} in Definition 8.5.1 gives the number of steps needed to solve the puzzle. It follows from the structure of the epistemic models that underlie the reasoning:

Proposition 8.5.2. *Let Q be a generalized quantifier, and n be the number of children. Then the corresponding muddy quantifier is $Q^{MC} = (Q, f_Q)$, where the*

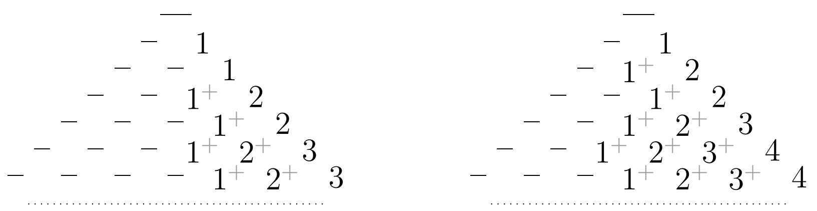


Figure 8.5: 'More than half' and 'More than one third'

partial function $f_Q : Q \rightharpoonup \mathbb{N}$ is defined in the following way.

$$f_Q((n-m,m)) = \min(\mu_{x \leq n-m}\ (n-m-x, m+x) \notin \mathsf{Q}, \mu_{y \leq m}\ (n-m+y, m-y) \notin \mathsf{Q}).$$

In other words, the function assigns a value x to $(u-k,k)$ in the level u of the number triangle if $(u-k,k) \in \mathsf{Q}$ and there is $(u-\ell,\ell)$ in the level u such that $(u-\ell,\ell) \notin \mathsf{Q}$. Moreover, the value x encodes the distance from the nearest $(u-\ell,\ell)$ such that $(u-\ell,\ell) \notin \mathsf{Q}$.

Concerning the assignment of the number of steps needed for solving the puzzle, we can also ask what is the structure of those steps. Namely, we can characterize situations in which some agents infer their status from the announcements of other agents, in contrast to the cases in which it happens simultaneously (we use '+'-superscripts to identify those situations). The definition of the partial function $f_{\mathsf{Q}}^{+} : Q \rightharpoonup \{+\}$ can be given in the following way. $f_Q^{+}((n-m,m)) = +$ iff:

1. $f_Q((n-m,m))$ is defined, and
2. $m \neq 0$ and $m \neq n$ and some agent considers two factual worlds possible.

The above discussion leads to an observation that solving the Muddy Children Puzzle is possible if the announcement of the quantifier leaves one observational state with just one successor. Therefore we can characterize Muddy Children Solvability in the following way:

Theorem 8.5.3 (Muddy Children Solvability)**.** *Let n be the number of children, $m \leq n$ the number of muddy children, and Q be the background assumption. A Muddy Children situation is solvable iff $(n-m,m) \in \mathsf{Q}$ and there is an $\ell \leq n$ such that $(n-\ell,\ell) \notin \mathsf{Q}$.*

The theorem is easy to verify with the use of the epistemic modeling explained in the previous section.

8.6 Discussion

Internal complexity and plausible modeling One of the main aims of applying logic in artificial intelligence and cognitive science is to model possible inferential strategies of an agent. An immediate plausibility test is the computational complexity of the proposed model. However, by ‘complexity’ we do not mean the computational complexity of model checking of the logic in general, but rather an agent’s internal complexity (cf. Aucher, 2010). Namely, how many computational resources (e.g., time and working memory) an agent needs to carry out a solution procedure suggested by the logical model. If the model demands that an agent has to perform intractable computations, then it might be not a correct description of the cognitive task an agent is faced with. Even though humans can not always deal with epistemic reasonings very efficiently (see Meijering, Van Maanen, Van Rijn, & Verbrugge, 2010), we do not expect them to come up with epistemic representations that lead to intractable problems, if only there is a simpler possibility (cf. Van Rooij, 2008; Szymanik, 2009).

The framework of dynamic epistemic logic as explained in Chapter 2 allows a clear and comprehensive explanation of the underlying phenomena. However, the representation of the problem is extensive—the size of the epistemic model to be considered is exponential with respect to the number of children. Unfortunately, this is not a desirable property of a cognitive model. It violates working memory restrictions humans are subjected to, and therefore it is hard to believe that subjects’ mental computations are based on that logical representation. The representation of the puzzle proposed in this chapter does not share the above-mentioned exponential-size problem. In our modeling of the Muddy Children puzzle the local perspectives of the agents are taken explicitly. The observational states encode the content of their background knowledge. Then we have the decision process—each agent’s task is to decide his ‘muddiness’. In order to decide whether both possibilities are open, first the quantifier has to be computed, the task that for most everyday quantifiers is easy (see Szymanik, 2009; Szymanik & Zajenkowski, 2010). Then the iterated epistemic reasoning takes place. Our models determine how many steps of the epistemic reasoning are needed to know which state holds. This explicit, step by step analysis brings us closer to investigating the internal complexity of epistemic problems that the agents are facing. The mental representations implicitly postulated here are more local (and therefore linear in size).

Isomorphism closure The size of our models is clearly connected to the properties of generalized quantifiers. One can argue that the latter have a major expressive weakness—they are closed under isomorphism. There are many important aspects of situations that they simply ‘overlook’. Our work indicates that this property can increase the informational power of a message in certain situations. Clearly, announcements that do not ‘trim’ all agents’ uncertainty simultaneously, announcements that are agent-specific, do not have the power of leading to a successful epistemic iteration. If the announcement includes only a determiner that is not closed under isomorphism, e.g.:

Agent a is muddy.

all worlds that make a clean disappear, together with the uncertainty of a himself. But this elimination is not the same for all agents—others cannot infer anything more no matter how many times they are asked to do so.

The choice of epistemic representation As mentioned above, our number triangle-based modeling of Muddy Children situations uses structures that are linear in size with respect to the number of agents. This significant improvement with respect to the classical DEL approach begs for an explanation. In particular, it is interesting which aspects of the epistemic situation, if any, have to be dropped or redefined. Our framework is clearly compatible with the one of DEL in terms of (the structure and the number of) steps needed for completion of epistemic reasoning. Their semantic interpretation is different, because the model behind them is significantly changed. Hence, the question remains what is the source of these concise models. We view the goal of the Muddy Children reasonings in terms of the individual learning of the agents, and not in the emergence of some general properties, like common knowledge. We believe that this difference is one of perspective, rather than one of core content of our work. What makes our models essentially smaller is that they 'internalize' more assumptions about agents' reasoning capabilities, e.g., agents become implicitly aware of the binary nature of the situation. Moreover, our choice of epistemic modeling is particularly suited for discussing 'efficiency' of various quantifier announcements. A generalization of this observation indicates that the domain of the classical epistemic model can be partitioned by an equivalence relation, and the efficiency of the announcements depends on their 'compatibility' with this partition. In other words, all assertions that are used in the scenario either remove or retain whole partition cells. In that case, clearly, the update process will terminate with a number of steps measured by the number of equivalence classes, and not with the size of the actual model. In particular, in the Muddy Children puzzle the equivalence classes are given by the worlds connected by permutations of individuals, and all relevant assertions, both the Father's announcement and the children's subsequent 'silence', respect that equivalence relation. This perspective on the *generic assertions* of the Muddy Children puzzle can be linked to the rationality assertions of game solution procedures (see Van Benthem, 2007). Another interesting interpretation of the partition is the one with the notion of *issue* in dynamic epistemic logic of questions (Van Benthem & Minica, 2010): by choosing an optimal issue, we can speed up the learning processes dramatically. Moreover, it is important to note that the problem of making epistemic models more concise has recently been considered in the context of abstraction techniques for Kripke models (Wang, 2010). Investigating our modelling in this context constitutes an interesting topic for future research.

The local, internal perspective on the puzzle provides an interesting link with formal epistemology. An agent in the Muddy Children puzzle can be seen as a scientist who tries to inductively decide a hypothesis, tries to discover what the actual world is like. Our analysis shows that even if the agents have limited observational capacities, the presence and interconnection with other scientists doing similar research can influence the discovery in a positive way.

8.7 Conclusions and Perspectives

In this chapter we characterized solvability of the Muddy Children puzzle with arbitrary generalized quantifier announcements. We introduced a new kind of logical modeling of the puzzle based on the idea of the number triangle. In our approach the representation that an agent has to come up with is exponentially smaller than in other models based on dynamic epistemic logic. Therefore, it seems that the model can be attractive in all those applications where an agent's internal complexity of the problem is crucial, like cognitive science modeling or designing multi-agent systems in the domain of artificial intelligence.

There are many further methodological questions concerning our logical modeling. First is that of the generality of our approach. Can we extend it in a way that allows more flexibility of logical theories, like DEL? A possible direction would be to associate explicitly our local representations with computational procedures, e.g., by viewing the representation in terms of automata theory (cf. Van der Meyden, 1996; Su, Sattar, Governatori, & Chen, 2005). Secondly, our work includes extension of public announcements to arbitrary generalized quantifiers. This in itself leads to a number of important issues, e.g., what is the epistemic logic of quantifier public announcements?

Our work generates many directions of follow-up research. For instance, we could consider situations with many predicates (e.g., children having spots of different colors on their foreheads), manipulate the observational power of the children or restrict their abilities to infer higher-order epistemic states to account for well-known processing bottlenecks (see, e.g., Verbrugge, 2009). Finally, distinguishing between factual and observational states in the proposed epistemic modeling can be used to investigate other types of epistemic inferences and puzzles, for example Russian Cards or the Top-Hat puzzle. In general, we hope that this fresh view on the old puzzle will motivate new developments in the study of agents' local perspective in multi-agent intelligent interaction.

Part IV

Conclusions

Chapter 9

Conclusions and Outlook

In Chapter 1 we defined as the aim of this thesis "to link learning theory with logics of knowledge and belief". Let us now look back to see to what extent we have fulfilled this goal.

We started off (in Chapter 3) with a methodological analysis of both frameworks, in particular with analyzing the basic learning-theoretic setting in terms of dynamic epistemic logic. We observed a compatibility with respect to basic epistemological assumptions, but we also pointed out some difficulties, in particular the difference in the perspective on the mind-change process. Learning theory focuses on computational problems and on a global picture of sequences of conjectures. Dynamic epistemic logic zooms in on particular steps of revision, providing a more constructive, logical approach. We built on what the two have in common, i.e., on the initial uncertainty of the agent. The translation of the basic learning theory setting into the semantics of modal logic exposes the epistemic grounds of inductive inference. It also shows the restrictions of the domain of epistemic problems covered by learning theory. In particular, the learning theory setting is conceptually limited to the single agent case.

Part II of this thesis is directly concerned with the problem of translation between the paradigms. The idea is to express learnability in epistemic and doxastic logic, accounting for the temporal aspects of learning as well. In Chapter 4 we approached the problem from the perspective of identifiability in the limit, the dominant notion of learning theory. Learning in the limit is the kind of learning of which success does not depend on reaching certainty. It does not concern the state of irrevocable knowledge, but rather stable true belief. Hence, we linked it with the belief-revision problem and with doxastic logic. Using a characterization of identifiability in the limit we defined learnability as the reachability of safe belief. We also showed the properties of various belief-revision policies with respect to their ability to converge to the true belief. First, we restricted ourselves to learning from sound and complete streams of positive data. We showed that learning methods based on belief revision via conditioning (update) and lexicographic revision are universal, i.e., provided certain prior conditions, those methods are as powerful as identification in the limit. Those prior conditions, the agent's prior dispositions for belief revision, play a crucial role here. We showed that in some cases, these priors cannot be modeled using standard belief-revision models (as those are based on well-founded preorders), but only using generalized models (based on simple preorders). Furthermore, we drew conclusions about the existence of a tension between conservatism and learning power by showing that the very popular, most 'conservative' belief-revision method fails to be universal. In the second part we turned to the case of learning from both positive and negative data. Here, along with positive information the agent receives negative data about facts that do not hold of the actual world. We again assumed these streams to be truthful and we drew conclusions about iterated belief revision governed by such streams. This enriched framework allows us to consider the occurrence of erroneous information. Provided that errors occur finitely often and are always eventually corrected we show that the lexicographic revision method is still reliable, but more conservative methods fail.

In Chapter 5 we were also concerned with expressing learnability in the epistemic framework, this time focusing only on update. We advanced a different approach—we investigated both finite identifiability and identifiability in the limit as properties of epistemic and doxastic models. We characterized the outcome of finite identification in the language of epistemic logic and dynamic epistemic logic. As a corollary from our results obtained in Chapter 4, we also characterized the outcome of identification in the limit in doxastic logic. Then, focusing on the procedural aspect of learning we observed that the iterated update of finite identification generates an epistemic temporal forest. We used the latter to characterize finite identifiability in epistemic temporal logic. Then, we again extended our results to the case of identifiability in the limit, indicating the temporal conditions for the success of this type of learning. As a result, we show what properties the initial uncertainty range needs to have, to guarantee reaching the state of irrevocable knowledge and true stable belief. Both irrevocable knowledge and true stable belief concern the complete description of the actual world. Finally, we mentioned that our temporal setting accounts for more than just language learning. All other types of learning, e.g., function-learning, can be analyzed in this temporal setting.

In Part II our aim was to establish a connection between the two frameworks. As we all know, well-established relationships are usually based on mutual profit, motivation and inspiration exchange. The research presented in Part III of this book was meant to live up to this kind of expectation. In Chapter 6, inspired by the dynamic epistemic logic interpretation of certainty, we investigated it in the learning-theoretic setting. We focused on the distinction between objective certainty (the objective lack of alternatives to the actual world) and subjective, introspective certainty (with the agent being aware of the unambiguity). Assuming the computability of agents, we showed that there are classes of languages that are finitely identifiable, i.e., the agent can always eventually conclude certainty; but no computable agent can always conclude it as soon as it is objectively possible, i.e., as soon as the data exclude all other possibilities. For learning theory this means that the domain of finite identifiability properly contains the domain of fastest finite identification—a well-motivated new kind of finite identification. In the same chapter, on the basis of learning theory, we also investigated the complexity of obtaining minimal conclusive samples of information, the minimal descriptions that include enough information to eliminate uncertainty. Our results may, for instance, find applications in the analysis of efficient communication in epistemic games. In particular, the task of finding a minimal-size sample eliminating uncertainty turned out to be NP-complete and hence, most probably, more difficult than finding any minimal sample, which can be done in polynomial time. As the task of finding a minimal-size sample could be delegated to a helpful teacher, our results provide an indirect computational motivation for introducing another agent, a teacher, to the simple setting of finite identification. Therefore, computational analysis shows that teaching in the most efficient way is very demanding.

The aim of Chapter 7 was to enrich both learning theory and dynamic epistemic logic with new learning-related concepts. We questioned the learning-theoretic dogma of learner-teacher cooperativeness by analyzing the computational complexity of various learning and teaching attitudes. It turned out that the question of teachability of a concept (the possibility to direct the learner to the desired state in a graph) is NP-complete under the assumption of non-cooperativeness of the learner. Our epistemic interpretation of Sabotage Games provided in that chapter, is based on deletion, as are epistemic update and the framework of learning by erasing, discussed in Chapter 2, but it gives an account of the progress of mind-change that differs from both. It allows thinking of incoming information in a more local way—as the removal of possible mind-changes, rather than the hypotheses themselves. This is a step towards modeling a learner to have limited access to the whole range of possibilities.

In Chapter 7 we explicitly enriched learning situations with a second agent, the teacher. In Chapter 8 we considered a multi-agent scenario of the Muddy Children puzzle. Inspired by results on the learnability of generalized quantifiers, we reinterpreted the agents of the puzzle as scientists. They are supposed to decide a hypothesis on the basis of a background assumption (a generalized quantifier) and inductive epistemic inference. The most immediate contribution to dynamic epistemic logic is a concise, linear representation of the epistemic situation of Muddy Children and a characterization of the solvability of the generalized version of the Muddy Children puzzle (with arbitrary quantifiers). Viewing the puzzle from the perspective of learning theory, leads one to extend the incoming factual, propositional information by some epistemic, indirect information. It sheds light on a different kind of learning performed by a team of learners with interconnected observational power. Moreover, our approach gives additional insights into the efficiency of learning.

Overall, in this thesis we focused on building a connection between formal learning theory and dynamic epistemic logic. The semantic link provided dynamic epistemic logic with a uniform framework for considering iterated actions. The semantic analysis underlying this logical view on inductive inference led us to give syntactic chracterizations of learnability in doxastic epistemic modal-temporal logics. Further topics of the thesis, taken from the domains of computability, games, and multi-agency, strengthen the connection by providing additional insights into the process of epistemic and doxastic change.

There are many open questions and directions for further work. They are all discussed in detail in the corresponding chapters of the thesis. Let us here underline some of them that seem especially interesting from the perspective of the interrelation between learning theory and logics of knowledge and belief.

The first open problem concerns the transfer of the *computational aspects* of learning theory to the frameworks of belief revision and epistemic change. The task of revising beliefs has a direct psychological interpretation. This cognitive link leads us towards the computational framework. Hence, a theoretically interesting and an empirically inspiring direction is to establish a broad learning-theory-like computational platform for investigating the restriction to computable (and perhaps also tractable, cf. Chapter 8) belief-revision. Chapter 4 gives a general setting that could be adapted to such considerations.

The second direction is to incorporate non-trivial *epistemic multi-agency* into the framework of learning theory. Such multi-agency is the very central concept of game theory and epistemic logic. In learning theory, team learning has been considered in the context of improving learnability, but these considerations have been limited by conditions of true information and perfect observation. From our considerations in Chapter 8 we can see that there are very simple scenarios in which quite substantial observational limitations of the agents can still lead to successful learning.

The third further direction concerns a generalization of learning-theoretic approach to account for a great *variety of input information.* As we showed (e.g., in Chapter 4) the character of the incoming information can heavily influence convergence. Hence, it is reasonable to account for different levels of trustworthiness in formal learning models, by linking different kinds of incoming information with different operations on models.

The fourth topic of further work concerns analyzing particular learning algorithms, properties of learning functions, and properties of learnable classes in epistemic and doxastic temporal logic. In particular, we would like to get the *temporal characterizations of protocols* that govern certain types of learning or lead to particular learning effects. The topic is especially interesting, because the temporal treatment of hypotheses gives a framework for analyzing different types of structures on a common ground, as we argued in Chapter 5.

Bibliography

Alchourrón, C. E., Gärdenfors, P., & Makinson, D. (1985). On the logic of theory change: Partial meet contraction and revision functions. *The Journal of Symbolic Logic*, *50*(2), 510–530.

Angluin, D. (1980). Inductive inference of formal languages from positive data. *Information and Control*, *45(2)*, 117–135.

Angluin, D. (1987). Learning regular sets from queries and counterexamples. *Information and Computation*, *75*(2), 87–106.

Angluin, D., & Smith, C. H. (1983). Inductive inference: Theory and methods. *ACM Computing Surveys*, *15*(3), 237–269.

Aucher, G. (2010). An internal version of epistemic logic. *Studia Logica*, *94*(1), 1–22.

Balakrishnan, V. K. (1997). *Graph Theory*. McGraw-Hill.

Balbach, F. J., & Zeugmann, T. (2009). Recent developments in algorithmic teaching. In A. H. Dediu, A.-M. Ionescu, & C. Martín-Vide (Eds.) *LATA'09: Proceedings of 3rd International Conference on Language and Automata Theory and Applications*, vol. 5457 of *Lecture Notes in Computer Science*, (pp. 1–18). Springer.

Baltag, A., & Moss, L. (2004). Logics for epistemic programs. *Synthese*, *139*(2), 165–224.

Baltag, A., Moss, L. S., & Solecki, S. (1998). The logic of public announcements, common knowledge, and private suspicions. In I. Gilboa (Ed.) *TARK'98: Proceedings of the 7th Conference on Theoretical Aspects of Rationality and Knowledge*, (pp. 43–56). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Baltag, A., & Smets, S. (2006). Dynamic belief revision over multi-agent plausibility models. In G. Bonanno, W. van der Hoek, & M. Wooldridge (Eds.) *LOFT'06: Proceedings of 7th Conference on Logic and the Foundations of Game and Decision Theory*, (pp. 11–24). University of Liverpool.

Baltag, A., & Smets, S. (2008a). The logic of conditional doxastic actions. In R. van Rooij, & K. Apt (Eds.) *New Perspectives on Games and Interaction. Texts in Logic and Games*. Amsterdam University Press.

Baltag, A., & Smets, S. (2008b). A qualitative theory of dynamic interactive belief revision. In G. Bonanno, W. van der Hoek, & M. Wooldridge (Eds.) *LOFT'08: Proceedings of 8th Conference on Logic and the Foundations of Game and Decision Theory*, no. 3 in Texts in Logic and Games, (pp. 9–58). Amsterdam University Press.

Baltag, A., & Smets, S. (2009a). Group belief dynamics under iterated revision: fixed points and cycles of joint upgrades. In *TARK'09: Proceedings of the 12th Conference on Theoretical Aspects of Rationality and Knowledge*, (pp. 41–50). ACM.

Baltag, A., & Smets, S. (2009b). Learning by questions and answers: From belief-revision cycles to doxastic fixed points. In H. Ono, M. Kanazawa, & R. de Queiroz (Eds.) *Logic, Language, Information and Computation*, vol. 5514 of *Lecture Notes in Computer Science*, (pp. 124–139). Springer.

Barwise, J., & Cooper, R. (1981). Generalized quantifiers and natural language. *Linguistics and Philosophy*, *4*, 159–219.

Van Benthem, J. (1986). *Essays in Logical Semantics*. Dordrecht: D. Reidel.

Van Benthem, J. (2005). An essay on sabotage and obstruction. In D. Hutter, & W. Stephan (Eds.) *Mechanizing Mathematical Reasoning, Essays in Honor of Jörg H. Siekmann on the Occasion of His 60th Birthday*, vol. 2605 of *Lecture Notes in Computer Science*, (pp. 268–276). Springer.

Van Benthem, J. (2006). One is a lonely number: on the logic of communication. In Z. Chatzidakis, P. Koepke, & W. Pohlers (Eds.) *LC'02: Proceedings of Logic Colloquium 2002*, vol. 27 of *Lecture Notes in Logic*, (pp. 96–129). Cergy-Pontoise: ASL & A.K. Peters.

Van Benthem, J. (2007). Dynamic logic for belief revision. *Journal of Applied Non-Classical Logics*, *2*, 129–155.

Van Benthem, J. (2007). Rational dynamics and epistemic logic in games. *International Game Theory Review*, *9:1*, 13–45. Erratum reprint, Volume 9:2, 377–409.

Van Benthem, J. (2010). *Logical Dynamics of Information and Interaction*. Cambridge: Cambridge University Press.

Van Benthem, J., & Dégremont, C. (2010). Bridges between dynamic doxastic and doxastic temporal logics. In G. Bonanno, B. Löwe, & W. van der Hoek (Eds.) *LOFT'08: Revised Selected Papers of 8th Conference on Logic and the Foundations of Game and Decision Theory*, vol. 6006 of *Lecture Notes in Computer Science*, (pp. 151–173). Springer.

Van Benthem, J., Van Eijck, J., & Kooi, B. (2006). Logics of communication and change. *Information and Computation*, *204*(11), 1620–1662.

Van Benthem, J., Gerbrandy, J., Hoshi, T., & Pacuit, E. (2009). Merging frameworks for interaction: DEL and ETL. *Journal of Philosophical Logic*, *38*(5), 491–526.

Van Benthem, J., & Liu, F. (2004). Diversity of logical agents in games. *Philosophia Scientiae*, *8(2)*, 163–178.

Van Benthem, J., & Minica, S. (2010). Questions and issue management. Toward a dynamic logic of questions. To appear in the Journal of Philosophical Logic.

Van Benthem, J., & Pacuit, E. (2006). The tree of knowledge in action: Towards a common perspective. In G. Governatori, I. Hodkinson, & Y. Venema (Eds.) *AiML'06: Proceedings of Advances in Modal Logic 2006*, vol. 6, (pp. 87–106). Edmonton: King's College.

Blackburn, P., Rijke, M. D., & Venema, Y. (2001). *Modal Logic*. No. 53 in Cambridge Tracts in Theoretical Computer Science. Cambridge University Press.

Blum, L., & Blum, M. (1975). Toward a mathematical theory of inductive inference. *Information and Control*, *28*, 125–155.

Board, O. (2004). Dynamic interactive epistemology. *Games and Economic Behavior*, *49*(1), 49–80.

Boutilier, C. (1996). Iterated revision and minimal change of conditional beliefs. *Journal of Philosophical Logic*, *25*, 263–305.

Chisholm, R. (1982). *Knowledge as Justified True Belief. The Foundations of Knowing*. Minneapolis: University of Minnesota Press.

Chomsky, N. (1965). *Aspects of the Theory of Syntax*. MIT Press.

Clark, R. (2010). On the learnability of quantifiers. In J. van Benthem, & A. ter Meulen (Eds.) *Handbook of Logic and Language*, (pp. 909–922). Elsevier. 2nd edition.

Darwiche, A., & Pearl, J. (1997). On the logic of iterated belief revision. *Artificial Intelligence*, *89*, 1–29.

Dégremont, C. (2010). *The temporal mind. Observations on the logic of belief change in interactive systems*. Ph.D. thesis, Universiteit van Amsterdam.

Dégremont, C., & Gierasimczuk, N. (2009). Can doxastic agents learn? On the temporal structure of learning. In X. He, J. F. Horty, & E. Pacuit (Eds.) *LORI'09: Proceedings of 2nd International Workshop on Logic, Rationality, and Interaction*, vol. 5834 of *Lecture Notes in Computer Science*, (pp. 90–104). Springer.

Van Ditmarsch, H., Van der Hoek, W., & Kooi, B. (2007). *Dynamic Epistemic Logic*. Springer Netherlands.

Emerson, E. A., & Halpern, J. Y. (1986). "Sometimes" and "not never" revisited: on branching versus linear time temporal logic. *Journal of ACM*, *33*(1), 151–178.

Fagin, R., Halpern, J. Y., Moses, Y., & Vardi, M. Y. (1995). *Reasoning About Knowledge*. MIT Press.

Gerbrandy, J. (1999a). *Bisimulations on planet Kripke*. Ph.D. thesis, Universiteit van Amsterdam.

Gerbrandy, J. (1999b). Dynamic epistemic logic. In L. Moss, J. Ginzburg, & M. De Rijke (Eds.) *Logic, Language, and Computation*, vol. 2. CSLI Publications.

Gettier, E. L. (1963). Is justified true belief knowledge? *Analysis*, *23*(6), 121–123.

Geurts, B. (2003). Reasoning with quantifiers. *Cognition*, *86*, 223–251.

Gierasimczuk, N. (2007). The problem of learning the semantics of quantifiers. In *TBiLLC'05: 6th International Tbilisi Symposium on Logic, Language, and Computation. Revised Selected Papers*, vol. 4363 of *Lecture Notes in Artificial Intelligence*, (pp. 117–126). Springer.

Gierasimczuk, N. (2009a). Bridging learning theory and dynamic epistemic logic. *Synthese*, *169*(2), 371–384.

Gierasimczuk, N. (2009b). Identification through inductive verification. In *TBiLLC'07: 7th International Tbilisi Symposium on Logic, Language, and Computation. Revised Selected Papers*, vol. 5422 of *Lecture Notes in Computer Science*, (pp. 193–205). Berlin, Heidelberg: Springer-Verlag.

Gierasimczuk, N. (2009c). Learning by erasing in dynamic epistemic logic. In *LATA'09: Proceedings of 3rd International Conference on Language and Automata Theory and Applications*, vol. 5457 of *Lecture Notes in Computer Science*, (pp. 362–373). Springer.

Gold, E. (1967). Language identification in the limit. *Information and Control*, *10*, 447–474.

Grabowski, J. (1987). Inductive inference of functions from noised observations. In K. Jantke (Ed.) *Analogical and Inductive Inference*, vol. 265 of *Lecture Notes in Computer Science*, (pp. 55–60). Springer Berlin/Heidelberg.

Grice, H. P. (1975). Logic and conversation. In P. Cole, & J. Morgan (Eds.) *Syntax and semantics*, vol. 3. New York: Academic Press.

Grove, A. (1988). Two modellings for theory change. *Journal of Philosophical Logic*, *17*, 157–170.

Hendricks, V. (1995). AGM-identification of first order structures. Tech. rep., Department of Philosophy, Carnegie Mellon University, PA, USA.

Hendricks, V. (2001). *The Convergence of Scientific Knowledge: A View from The Limit*. Dordrecht: Kluwer Academic Publishers.

Hendricks, V. (2003). Active agents. *Journal of Logic, Language and Information*, *12*(4), 469–495.

Hintikka, J. (1962). *Knowledge and Belief. An Introduction to the Logic of the Two Notions*. Cornell University Press.

Hoshi, T. (2009). *Epistemic Dynamics and Protocol Information*. Ph.D. thesis, Stanford University.

Jain, S., Osherson, D., Royer, J. S., & Sharma, A. (1999). *Systems that Learn*. Chicago: MIT Press.

Jain, S., & Sharma, A. (1996). Team learning of recursive languages. In *PRICAI'96: Proceedings of the 4th Pacific Rim International Conference on Artificial Intelligence*, (pp. 324–335). London, UK: Springer-Verlag.

Jantke, K. P. (1979). Natural properties of strategies identifying recursive functions. *Elektronische Informationverarbeitung und Kybernetik*, *15*, 487–496.

Johnson-Laird, P. N. (1983). *Mental Models: Toward a Cognitive Science of Language, Inference and Consciousness*. Harvard University Press.

Karp, R. M. (1972). Reducibility among combinatorial problems. In R. E. Miller, & J. W. Thatcher (Eds.) *Complexity of Computer Computations*, (pp. 85–103). Plenum Press.

Kelly, K. (1996). *The Logic of Reliable Inquiry*. Oxford: Oxford University Press.

Kelly, K., Schulte, O., & Hendricks, V. (1995). Reliable belief revision. In *Proceedings of the 10th International Congress of Logic, Methodology, and Philosophy of Science*, (pp. 383–398). Kluwer Academic Publishers.

Kelly, K. T. (1998a). Iterated belief revision, reliability, and inductive amnesia. *Erkenntnis*, *50*, 11–58.

Kelly, K. T. (1998b). The learning power of belief revision. In *TARK'98: Proceedings of the 7th Conference on Theoretical Aspects of Rationality and Knowledge*, (pp. 111–124). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Kelly, K. T. (2004). Learning theory and epistemology. In I. Niiniluoto, M. Sintonen, & J. Smolenski (Eds.) *Handbook of Epistemology*. Dordrecht: Kluwer.

Kelly, K. T. (2008). Ockham's razor, truth, and information. In P. Adriaans, & J. van Benthem (Eds.) *Handbook of the Philosophy of Information*. Elsevier.

Kleene, S. C. (1943). Recursive predicates and quantifiers. *Transactions of the American Mathematical Society*, *53*(1), 41–73.

Klein, D., Radmacher, F. G., & Thomas, W. (2009). The complexity of reachability in randomized sabotage games. In F. Arbab, & M. Sirjani (Eds.) *FSEN'09: Proceedings of 3rd International Conference on Fundamentals of Software Engineering*, vol. 5961 of *Lecture Notes in Computer Science*.

Kozen, D. (1983). Results on the propositional μ-calculus. *Theoretical Computer Science*, *27*, 333–354.

Kugel, P. (1986). Thinking may be more than computing. *Cognition*, *22*(2), 137–198.

Ladusaw, W. (1979). *Polarity Sensitivity as Inherent Scope Relations*. Phd thesis, University of Texas.

Lange, S., Wiehagen, R., & Zeugmann, T. (1996). Learning by erasing. In *ALT'96: Proceeding of 7th International Workshop on Algorithmic Learning Theory*, Lecture Notes in Artificial Intelligence, (pp. 228–241). Springer-Verlag.

Lange, S., & Zeugmann, T. (1992). Types of monotonic language learning and their characterization. In *COLT'92: Proceedings of the 5th Annual ACM Conference on Computational Learning Theory*, (pp. 377–390). ACM.

Lange, S., & Zeugmann, T. (1996). Set-driven and rearrangement-independent learning of recursive languages. *Mathematical Systems Theory*, *29*(6), 599–634.

Lehrer, K. (1965). Knowledge, truth and evidence. *Analysis*, *25*(5), 168–175.

Lehrer, K. (1990). *Theory of Knowledge*. Routledge, London.

Levi, I. (1980). *The Enterprise of Knowledge*. Cambridge, MA: MIT Press.

Löding, C., & Rohde, P. (2003a). Solving the sabotage game is PSPACE-hard. In *MFCS'03: Proceedings of the 28th International Symposium on Mathematical Foundations of Computer Science*, vol. 2474 of *Lecture Notes in Computer Science*, (pp. 531–540). Springer.

Löding, C., & Rohde, P. (2003b). Solving the sabotage game is PSPACE-hard. Tech. rep., Aachener Informatik Berichte, RWTH Aachen.

Martin, E., & Osherson, D. (1997). Scientific discovery based on belief revision. *The Journal of Symbolic Logic*, *62*(4), 1352–1370.

Martin, E., & Osherson, D. (1998). *Elements of Scientific Inquiry*. Cambridge: MIT Press.

Meijering, B., Van Maanen, L., Van Rijn, H., & Verbrugge, R. (2010). The facilitative effect of context on second order social reasoning. In R. Catrambone, & S. Ohlsson (Eds.) *CogSci'10: Proceedings of the 32nd Annual Conference of the Cognitive Science Society*.

Van der Meyden, R. (1996). Finite state implementations of knowledge-based programs. In *FSTTCS'96: Proceedings of the Annual Conference on Foundations of Software Technology and Theoretical Computer Science 1996*, (pp. 262–273).

Van der Meyden, R., & Wong, K. (2003). Complete axiomatizations for reasoning about knowledge and branching time. *Studia Logica*, *75*(1), 93–123.

Moses, Y., Dolev, D., & Halpern, J. Y. (1986). Cheating husbands and other stories: A case study of knowledge, action, and communication. *Distributed Computing*, *1*(3), 167–176.

Mostowski, A. (1957). On a generalization of quantifiers. *Fundamenta Mathematicae*, *44*, 12–36.

Mukouchi, Y. (1992). Characterization of finite identification. In *AII'92: Proceedings of the International Workshop on Analogical and Inductive Inference*, (pp. 260–267). Springer-Verlag.

Nozick, R. (1981). *Philosophical Explanations*. Harvard University Press.

Papadimitriou, C. M. (1994). *Computational Complexity*. Reading, MA: Addison-Wesley.

Parikh, R., & Ramanujam, R. (2003). A knowledge based semantics of messages. *Journal of Logic, Language and Information*, *12*(4), 453–467.

Peters, S., & Westerståhl, D. (2006). *Quantifiers in Language and Logic*. Oxford: Oxford University Press.

Plato (360 B.C.). *Theaetetus*.

Plaza, J. (1989). Logics of public communications. In M. Emrich, M. Pfeifer, M. Hadzikadic, & Z. Ras (Eds.) *Proceedings of the 4th International Symposium on Methodologies for Intelligent Systems*, (pp. 201–216).

Popper, K. (1966). *The Open Society and its Enemies*. Princeton University Press.

Putnam, H. (1965). Trial and error predicates and the solution to a problem of Mostowski. *The Journal of Symbolic Logic*, *30*(1), 49–57.

Radmacher, F. G., & Thomas, W. (2008). A game theoretic approach to the analysis of dynamic networks. *Electronic Notes in Theoretical Computer Science*, *200*(2), 21–37.

Romesburg, C. H. (1978). Simulating scientific inquiry with the card game eleusis. *Science Education*, *5*(63), 599–608.

Van Rooij, I. (2008). The tractable cognition thesis. *Cognitive Science: A Multidisciplinary Journal*, *32*(6), 939–984.

Rott, H. (2004). Stability, strength and sensitivity: Converting belief into knowledge. *Erkenntnis*, *61*(2-3).

Rott, H. (2008). A new psychologism in logic? Reflections from the point of view of belief revision. *Studia Logica*, *88*(1), 113–136.

Scott, D., & Bakker, W. D. (1969). A theory of programs. Unpublished manuscript, IBM, Vienna.

Shapiro, E. (1998). *Algorithmic Program Debugging*. Cambridge, MA: MIT Press.

Smith, C. H. (1982). The power of pluralism for automatic program synthesis. *Journal of ACM*, *29*(4), 1144–1165.

Smullyan, R. M. (1958). Undecidability and recursive inseparability. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, *4*(7–11), 143–147.

Solomonoff, R. (1964a). A formal theory of inductive inference. Part I. *Information and Control*, *7*(1), 1–22.

Solomonoff, R. (1964b). A formal theory of inductive inference. Part II. *Information and Control*, *7*(2), 224–254.

Spohn, W. (1988). A general non-probabilisistic theory of inductive reasoning. *Uncertainty in Artificial Intelligence*, *4*, 149–159.

Stalnaker, R. (2009). Iterated belief revision. *Erkenntnis*, *70*(2), 189–209.

Su, K., Sattar, A., Governatori, G., & Chen, Q. (2005). A computationally grounded logic of knowledge, belief and certainty. In *AAMAS'05: Proceedings of the 4th International Joint Conference on Autonomous Agents and Multiagent Systems*, (pp. 149–156).

Szymanik, J. (2009). *Quantifiers in TIME and SPACE. Computational Complexity of Generalized Quantifiers in Natural Language*. Ph.D. thesis, Universiteit van Amsterdam.

Szymanik, J., & Zajenkowski, M. (2010). Comprehension of simple quantifiers. Empirical evaluation of a computational model. *Cognitive Science*, *34*(3), 521–532.

Tiede, H.-J. (1999). Identifiability in the limit of context-free generalized quantifiers. *Journal of Language and Computation*, *1*, 93–102.

Väänänen, J. (2002). On the expressive power of monotone natural language quantifiers over finite models. *Journal of Philosophical Logic*, *31*, 327–358.

Vardi, M. Y. (1982). The complexity of relational query languages. In *STOC'82: Proceedings of the 14th Annual ACM Symposium on Theory of Computing*, (pp. 137–146). New York, NY, USA: ACM Press.

Verbrugge, R. (2009). Logic and social cognition. *Journal of Philosophical Logic*, *38*(6), 649–680.

Wang, Y. (2010). *Epistemic Modelling and Protocol Dynamics*. Ph.D. thesis, Universiteit van Amsterdam.

Wexler, K., & Cullicover, P. (1980). *Formal Principles of Language Acquisition*. Cambridge, MA: MIT Press.

Abstract

The thesis links learning theory with logics of knowledge and belief.

Following the introduction and mathematical preliminaries, Chapter 3 contains a methodological analysis of both frameworks, in particular it analyzes the basic learning-theoretic setting in terms of dynamic epistemic logic.

In Chapter 4 we use learning theory to evaluate dynamic epistemic logic-based belief-revision policies. We investigate them with respect to their ability to converge to the true belief for sound and complete streams of positive data, streams of positive and negative data, and erroneous fair information. We show that some belief-revision methods are universal on certain types of data, i.e., they have full learning power.

Chapter 5 is concerned with expressing identification in the limit and finite identifiability in the languages of modal and temporal logics of epistemic and doxastic change. We characterize learnability by formulas of various logics of knowledge and belief.

In Chapter 6 we investigate the notion of definite finite tell-tale set in finite identifiability of languages, in particular the computational complexity of finding various kinds of minimal DFTTs. Assuming the computability of learning functions we show that there are classes of languages that are finitely identifiable, but no computable agent can always conclude it as soon as it is objectively possible.

In Chapter 7 we analyze different levels of cooperativeness between the learner and the teacher in a game of perfect information based on sabotage games. We give formulas of sabotage modal logic that characterize the existence of winning strategies in such games. We show that non-cooperative case is PSPACE-complete, and that relaxing the strict alternation of the moves of the two players does not influence the winning conditions.

In Chapter 8 we generalize the Muddy Children puzzle, to account for arbitrary quantifier announcements. We characterize the solvability of the generalized version of the Muddy Children puzzle and we propose a new representation of the epistemic situation of Muddy Children scenarios. Our modeling is linear with respect to the number of agents, and is more concise than the one used in the classical dynamic epistemic approach.

Overall, we focus on building a connection between formal learning theory and dynamic epistemic logic. We provide dynamic epistemic logic with a uniform framework for considering iterated actions. On the other hand, this leads to a logical view on inductive inference and to syntactic characterizations of learnability in modal and temporal logics. Further topics of the thesis, taken from the domains of computability, games, and multi-agency, strengthen the connection by providing additional computational, logical and philosophical insights into the process of epistemic and doxastic change.

Samenvatting

Deze dissertatie verbindt leertheorie met logica's van kennis en geloof.

Na de inleiding en het wiskundige voorwerk volgt Hoofdstuk 3 met een methodologische analyse van beide gebieden. Daarbij analyseert dit hoofdstuk de leertheoretische werkwijze in termen van de dynamisch-epistemische logica.

In Hoofdstuk 4 gebruiken we leertheorie om strategieën voor geloofsrevisie te evalueren die gebaseerd zijn op dynamisch-episteische logica. We beoordelen deze strategien op hun mogelijkheden om naar waar geloof te convergeren op correcte en volledige stromen van positieve gegevens, op stromen van positieve en negatieve gegevens, en op beperkt foutenbevattende informatie. We laten zien dat geschikte geloofsrevisiemethoden universeel zijn op bepaalde soorten gegevens, d.w.z. dat ze dan volledige leerkracht hebben.

Hoofdstuk 5 houdt zich bezig met het uitdrukken van identificatie in de limiet en eindige identificeerbaarheid in modaal-logische en tijdslogische talen voor epistemische en doxastische verandering. We karakteriseren leerbaarheid daarbij door middel van formules van verschillende logica's van kennis en geloof.

In Hoofdstuk 6 onderzoeken we het begrip (eindige telltale-verzameling) uit de theorie van eindige identificeerbaarheid van talen. In het bijzonder onderzoeken we de computationele complexiteit van het vinden van diverse soorten van minimale DFTTs. Onder de aanname van de berekenbaarheid van de leerfuncties laten we zien dat er klassen van talen zijn die eindig identificeerbaar zijn, terwijl toch geen berekenbare agent altijd de juiste identificatie kan maken zodra dat objectief mogelijk is.

In Hoofdstuk 7 onderzoeken we verschillende niveaus van samenwerkingsbereidheid tussen leerling en leraar in een spel van perfecte informatie gebaseerd op sabotage-spelen. We geven formules van de sabotage-modale logica aan die het bestaan van winnende strategien in dergelijke spelen karakteriseren. We laten zien dat het niet-samenwerkingsgeval PSPACE-volledig is, en dat verzwakking van de eis dat de zetten van de twee spelers strikt alternerend zijn de condities voor winnen niet benvloeden.

In Hoofdstuk 8 generaliseren we de informatie-puzzel van de modderige kinderen door aankondigingen met willekeurige kwantoren toe te laten. We karakteriseren de gevallen waarin de gegeneraliseerde puzzel oplosbaar is en we stellen een nieuwe representatie voor van scenario's met modderige kinderen. Onze modelleermethode is lineair met betrekking tot het aantal agenten en is compacter dan die van de klassieke dynamisch-epistemische benadering.

Door het proefschrift heen zijn we er steeds op gericht een verbinding aan te brengen tussen formele leertheorie en dynamisch-epistemische logica. We geven aldus aan de dynamisch-epistemische logica een uniform kader voor het beschouwen van herhaalde handelingen. Aan de andere zijde leidt dit verband tot een logische visie op inductieve inferentie en tot syntactische karakteriseringen van leerbaarheid in modale en temporele logica's. De verdere thema's uit de dissertatie, afkomstig uit de gebieden van berekenbaarheid, speltheorie, en de theorie van meer-agentsystemen, versterken de verbinding door additionele computationele, logische en filosofische inzichten te geven in processen van epistemische en doxastische verandering.

Titles in the ILLC Dissertation Series:

ILLC DS-2001-01: **Maria Aloni**
Quantification under Conceptual Covers

ILLC DS-2001-02: **Alexander van den Bosch**
Rationality in Discovery - a study of Logic, Cognition, Computation and Neuropharmacology

ILLC DS-2001-03: **Erik de Haas**
Logics For OO Information Systems: a Semantic Study of Object Orientation from a Categorial Substructural Perspective

ILLC DS-2001-04: **Rosalie Iemhoff**
Provability Logic and Admissible Rules

ILLC DS-2001-05: **Eva Hoogland**
Definability and Interpolation: Model-theoretic investigations

ILLC DS-2001-06: **Ronald de Wolf**
Quantum Computing and Communication Complexity

ILLC DS-2001-07: **Katsumi Sasaki**
Logics and Provability

ILLC DS-2001-08: **Allard Tamminga**
Belief Dynamics. (Epistemo)logical Investigations

ILLC DS-2001-09: **Gwen Kerdiles**
Saying It with Pictures: a Logical Landscape of Conceptual Graphs

ILLC DS-2001-10: **Marc Pauly**
Logic for Social Software

ILLC DS-2002-01: **Nikos Massios**
Decision-Theoretic Robotic Surveillance

ILLC DS-2002-02: **Marco Aiello**
Spatial Reasoning: Theory and Practice

ILLC DS-2002-03: **Yuri Engelhardt**
The Language of Graphics

ILLC DS-2002-04: **Willem Klaas van Dam**
On Quantum Computation Theory

ILLC DS-2002-05: **Rosella Gennari**
Mapping Inferences: Constraint Propagation and Diamond Satisfaction

ILLC DS-2002-06: **Ivar Vermeulen**
A Logical Approach to Competition in Industries

ILLC DS-2003-01: **Barteld Kooi**
Knowledge, chance, and change

ILLC DS-2003-02: **Elisabeth Catherine Brouwer**
Imagining Metaphors: Cognitive Representation in Interpretation and Understanding

ILLC DS-2003-03: **Juan Heguiabehere**
Building Logic Toolboxes

ILLC DS-2003-04: **Christof Monz**
From Document Retrieval to Question Answering

ILLC DS-2004-01: **Hein Philipp Röhrig**
Quantum Query Complexity and Distributed Computing

ILLC DS-2004-02: **Sebastian Brand**
Rule-based Constraint Propagation: Theory and Applications

ILLC DS-2004-03: **Boudewijn de Bruin**
Explaining Games. On the Logic of Game Theoretic Explanations

ILLC DS-2005-01: **Balder David ten Cate**
Model theory for extended modal languages

ILLC DS-2005-02: **Willem-Jan van Hoeve**
Operations Research Techniques in Constraint Programming

ILLC DS-2005-03: **Rosja Mastop**
What can you do? Imperative mood in Semantic Theory

ILLC DS-2005-04: **Anna Pilatova**
A User's Guide to Proper names: Their Pragmatics and Semanics

ILLC DS-2005-05: **Sieuwert van Otterloo**
A Strategic Analysis of Multi-agent Protocols

ILLC DS-2006-01: **Troy Lee**
Kolmogorov complexity and formula size lower bounds

ILLC DS-2006-02: **Nick Bezhanishvili**
Lattices of intermediate and cylindric modal logics

ILLC DS-2006-03: **Clemens Kupke**
Finitary coalgebraic logics

ILLC DS-2006-04: **Robert Špalek**
Quantum Algorithms, Lower Bounds, and Time-Space Tradeoffs

ILLC DS-2006-05: **Aline Honingh**
The Origin and Well-Formedness of Tonal Pitch Structures

ILLC DS-2006-06: **Merlijn Sevenster**
Branches of imperfect information: logic, games, and computation

ILLC DS-2006-07: **Marie Nilsenova**
Rises and Falls. Studies in the Semantics and Pragmatics of Intonation

ILLC DS-2006-08: **Darko Sarenac**
Products of Topological Modal Logics

ILLC DS-2007-01: **Rudi Cilibrasi**
Statistical Inference Through Data Compression

ILLC DS-2007-02: **Neta Spiro**
What contributes to the perception of musical phrases in western classical music?

ILLC DS-2007-03: **Darrin Hindsill**
It's a Process and an Event: Perspectives in Event Semantics

ILLC DS-2007-04: **Katrin Schulz**
Minimal Models in Semantics and Pragmatics: Free Choice, Exhaustivity, and Conditionals

ILLC DS-2007-05: **Yoav Seginer**
Learning Syntactic Structure

ILLC DS-2008-01: **Stephanie Wehner**
Cryptography in a Quantum World

ILLC DS-2008-02: **Fenrong Liu**
Changing for the Better: Preference Dynamics and Agent Diversity

ILLC DS-2008-03: **Olivier Roy**
Thinking before Acting: Intentions, Logic, Rational Choice

ILLC DS-2008-04: **Patrick Girard**
Modal Logic for Belief and Preference Change

ILLC DS-2008-05: **Erik Rietveld**
Unreflective Action: A Philosophical Contribution to Integrative Neuroscience

ILLC DS-2008-06: **Falk Unger**
Noise in Quantum and Classical Computation and Non-locality

ILLC DS-2008-07: **Steven de Rooij**
Minimum Description Length Model Selection: Problems and Extensions

ILLC DS-2008-08: **Fabrice Nauze**
Modality in Typological Perspective

ILLC DS-2008-09: **Floris Roelofsen**
Anaphora Resolved

ILLC DS-2008-10: **Marian Counihan**
Looking for logic in all the wrong places: an investigation of language, literacy and logic in reasoning

ILLC DS-2009-01: **Jakub Szymanik**
Quantifiers in TIME and SPACE. Computational Complexity of Generalized Quantifiers in Natural Language

ILLC DS-2009-02: **Hartmut Fitz**
Neural Syntax

ILLC DS-2009-03: **Brian Thomas Semmes**
A Game for the Borel Functions

ILLC DS-2009-04: **Sara L. Uckelman**
Modalities in Medieval Logic

ILLC DS-2009-05: **Andreas Witzel**
Knowledge and Games: Theory and Implementation

ILLC DS-2009-06: **Chantal Bax**
Subjectivity after Wittgenstein. Wittgenstein's embodied and embedded subject and the debate about the death of man.

ILLC DS-2009-07: **Kata Balogh**
Theme with Variations. A Context-based Analysis of Focus

ILLC DS-2009-08: **Tomohiro Hoshi**
Epistemic Dynamics and Protocol Information

ILLC DS-2009-09: **Olivia Ladinig**
Temporal expectations and their violations

ILLC DS-2009-10: **Tikitu de Jager**
"Now that you mention it, I wonder...": Awareness, Attention, Assumption

ILLC DS-2009-11: **Michael Franke**
Signal to Act: Game Theory in Pragmatics

ILLC DS-2009-12: **Joel Uckelman**
More Than the Sum of Its Parts: Compact Preference Representation Over Combinatorial Domains

ILLC DS-2009-13: **Stefan Bold**
Cardinals as Ultrapowers. A Canonical Measure Analysis under the Axiom of Determinacy.

ILLC DS-2010-01: **Reut Tsarfaty**
Relational-Realizational Parsing

ILLC DS-2010-02: **Jonathan Zvesper**
Playing with Information

ILLC DS-2010-03: **Cédric Dégremont**
The Temporal Mind. Observations on the logic of belief change in interactive systems

ILLC DS-2010-04: **Daisuke Ikegami**
Games in Set Theory and Logic

ILLC DS-2010-05: **Jarmo Kontinen**
Coherence and Complexity in Fragments of Dependence Logic

ILLC DS-2010-06: **Yanjing Wang**
Epistemic Modelling and Protocol Dynamics

ILLC DS-2010-07: **Marc Staudacher**
Use theories of meaning between conventions and social norms

ILLC DS-2010-08: **Amélie Gheerbrant**
Fixed-Point Logics on Trees

ILLC DS-2010-09: **Gaëlle Fontaine**
Modal Fixpoint Logic: Some Model Theoretic Questions

ILLC DS-2010-10: **Jacob Vosmaer**
Logic, Algebra and Topology. Investigations into canonical extensions, duality theory and point-free topology.