Research Proposal

Vishal Chakraborty*

January 9, 2018

Title: *Learning in Sparse Background Structures*

Abstract

Learning, defined in a declarative framework, over sparse background structures, of at most poly-logarithmic degree, has been recently shown to be in poly-logarithmic time for First Order Logic (FOL) definable concepts. I will investigate if the claim holds for counting extensions of FOL and Modal Logic. Extensions of FOL with some basic counting and aggregation are expressible in infinitary counting logics which are a central component of relational database management systems. Modal Logic extends classical predicate logic with the use of modal operators there by making it richer. I will also investigate learnability of FOL definable concepts over the field of the Real Numbers as a background structure. With the increasing popularity of Artificial Neural Networks, this result will be significant. Broadly, the contribution of my research is in the field of descriptive complexity theory of machine learning.

1 Context

Machine Learning (ML), today, stands at an interesting juncture. Traditional methods have given way to state of the art techniques; well established doctrines are being challenged constantly; interdisciplinary applications is the new norm. Moreover, with the advent of Big Data, algorithms which are more efficient and accurate than humans are being used for everyday tasks [1]; model human behaviour; provide better medical care [2]; predict and mitigate natural disasters [3]. I want to explore the role of logic as the foundation in ML and the insights it has to offer.

For a well trained logician, FOL is natural. It is the lingua franca in the field of AI and knowledge representation. However, it is not popular among engineers and non-logicians. The logic suffers from some well-known drawbacks. It is, unlike human reasoning, monotonic [4] and computationally difficult. Moreover, it has limited expressivity. Thus, it is natural to ask if other

*University of Cambridge, vc354@cam.ac.uk

logics are better suited for the problem at hand. Extensions of first order logic have more expressive power. For example $FO(C)$—FOL extended with counting quantifiers and simple arithmetic [5]; $FO(Q_u)$—FOL extended with all unary generalised quantifiers [6].

In [7], the two logics — $\mathscr{L}^*_{\infty\omega}(C)$ and $\mathscr{L}^\circ_{\infty\omega}(C)$ have been proposed as a step towards a unifying framework for adding counting to FOL. The authors remark, $FO(C)$ and $FO(Q_u)$ restrict the available arithmetic and the use of free numerical variables respectively thereby making the task of expressing simple properties nontrivial. This motivates the need for "a general framework that subsumes all these logics [namely $FO(C)$ and $FO(Q_u)$], and is at the same time easy to study" [7].

Common lines of investigation in the field include identifying suitably expressive logics to carry out ML, creating new algorithms for learning, and determine the performance limits or complexity theoretic bounds. Previous works (for example, [8], [9], and [7]) have extensively studied FOL and its extensions, and have established many important results on learnability. Particularly, in [10], concepts definable in FOL has been studied over structures with bounded degree. [10] mentions the field of Real Numbers $(\mathbb{R};+;\cdot;<)$ as a possible background structure without much elaboration. This is at the very heart of my investigation.

The are a few general steps in the type of investigation I want to carry out. They are as follows.

1. Choose a background structure B based on the type of data available. Certain data can be best expressed as graphs. In such cases, we can choose B to be finitely labeled graphs. Some data may motivate the use of trees as a background structure.

2. Specify a model, generally parametric, using some formula of a logic, for example first-order logic or second-order logic. The definability of a concept or hypothesis depends on the expressive power of the logic being used.

Stated thus, the aforementioned steps motivate two possible lines of inquiry. First, how does the choice of background structure influence learnability? Second, are some logics better than others for learning in this context?

2 Methodology

An extensive study of the literature points towards the tools that are useful in gaining insight into properties of logics, applicability of algorithmic and statistical learning theories within the declarative framework, and complexity characterization of problems. In this section I will discuss Finite Model Theory and Algorithmic Learning Theory which are the two main broad areas which will be used in the study.

2.1 Finite Model Theory

Finite Model Theory is the primary tool to study the expressive power of logics. It was formulated around 1950. It concerns with the relation between syntax and semantics. Useful methods in Model Theory fail over finite structures. For example, the compactness theorem which allows to construct models for a set of sentences that are finitely consistent, fails. In finite model theory other methodologies are employed to study the expressive power of logics - Locality theorems, Ehrenfeucht-Fraïssé Games to name a few. It is evident in [10] that locality property of logics play an important role in projects similar to mine.

Finite Model Theory was founded as an independent field of logic from consideration of problems in theoretical computer science. Finite structures like databases and finite graphs are the main objects of study in this field. Many problems in theoretical computer science (complexity theory and database theory) can be expressed as problems in mathematical logic over finite structures. This is the principal motivation for the use of finite model theory [11].

2.2 Algorithmic Learning Theory

Algorithmic Learning Theory provides a mathematical framework to study machine learning problems and algorithms. It includes measures like Vapnik Chervonenkis (VC) dimension analysis and Probably Approximately Correct (PAC) learning to analyze [12, 13] the complexity, richness and flexibility of functions.

Let H be a family of sets and C a set. Their intersection is defined as the set family $H \cap C = \{h \cap C \mid h \in H\}$. The set C is said to be shattered by H if $H \cap C$ contains all the subsets of C, i.e., $H \cap C = 2^C$. The VC dimension of H is the largest integer D such that there exitsts a set C with cardinality D that is shattered by H. A set family with finite VC dimension is also PAC learnable. Families of sets definable in FOL over the real field (and many other structures) have finite VC dimension and are thus PAC learnable. In the 1990's, these results were central in database theory to understand expressiveness of languages that use operations on numbers and strings.

The main theorem in [10] states that concepts definable in FOL can be learned in "probably approximately correct" sense over structures with atmost poly-logarithmic degree in poly-logarithmic time. I investigate the case of the field of the Real Numbers as a background structure. This is significant because widely used learning models like Neural Networks and Vector Machines can be thought of as approximating some real valued function. Preliminary research indicates that the Tarski-Saidenburg property [14], quantifier elimination using the Cylindrical Algebraic Decomposition (CAD) algorithm will be helpful.

3 Preliminary Investigation

3.1 Learning over the Real Field as a Background Structure

Learning problems can be classified into two type- parameter learning and model learning. In the context of learning in a declarative framework, for the parameter learning problem it is assumed that there is a FO formula $\psi(\boldsymbol{x};\boldsymbol{y})$ with two kinds of free variables: the instance variables $\boldsymbol{x}$ and the parameter variables $\boldsymbol{y}$. The goal is to find a tuple $\boldsymbol{v}$ such that the function $[\![\psi(\boldsymbol{x};\boldsymbol{v})]\!]^{\mathscr{B}}$ (defined in the following section) on some background structure B with local access [1] is consistent with the supplied training sequence $T = (\boldsymbol{x}_i, t_i = C(\boldsymbol{x}_i)_{i=1}^{N}$ where C is some unknown concept the algorithm is trying to learn. In model learning, the task is to find the suitable formula $\varphi(\boldsymbol{x};\boldsymbol{y})$ such the corresponding function defined as $[\![\psi(\boldsymbol{x};\boldsymbol{y})]\!]^{\mathscr{B}}$ generalizes the unknown concept C well.

3.1.1 Background from Mathematical Logic

The vocabulary σ is a set of constant symbols denoted $c_1, c_2, c_3, \dots, c_n, \dots$; predicate symbols denoted $P_1, P_2, \dots, P_n, \dots$; function symbols denoted $f_1, f_2, \dots, f_n, \dots$. Let A be a set. A σ model denoted

$$\mathscr{A} = \langle A, \{c_i^{\mathscr{A}}\}, \{P_i^{\mathscr{A}}\}, \{f_i^{\mathscr{A}}\}\rangle$$

consists of a universe, the set $U(\mathscr{A}) = A$ along with an interpretation for each constant symbol $c_i \in \sigma$ as an element $c_i^{\mathscr{A}} \in A$; each k-ary predicate $P_i \in \sigma$ as a $P_i^{\mathscr{A}} \subseteq A^k$; each k-ary function symbol $f_i \in \sigma$ as a function $f_i^{\mathscr{A}} : A^k \longrightarrow A$.

In our case, $\sigma = \{0, 1, <, \cdot, +\}$ and the σ model is the real field $\mathscr{R} = \langle \mathbb{R}, 0^{\mathscr{R}}, 1^{\mathscr{R}}, <^{\mathscr{R}}, +^{\mathscr{R}}, \cdot^{\mathscr{R}}\rangle$ where the constant symbols $0^{\mathscr{R}}$ and $1^{\mathscr{R}}$, binary relation $<^{\mathscr{R}}$, and the binary functions $+^{\mathscr{R}}$ and $\cdot^{\mathscr{R}}$ have the usual meaning. To avoid excessive use of notation, we will omit the superscript with the name of the model and use the same symbol for both- a symbol in the vocabulary and its interpretation in the model. Thus, we shall write $\mathscr{R} = \langle \mathbb{R}, 0, 1, <, +, \cdot\rangle$.

With $\mathscr{B}$ as a background structure, we fix a logic L and use formulas in this logic to denote parametric models. Such a formula, say $\psi(\boldsymbol{x};\boldsymbol{y})$, has two types of free variables: $\boldsymbol{x} = (x_1, x_2, \dots x_k)$ that are the instance variables of the formula and $\boldsymbol{y} = (y_1, y_2, \dots y_l)$ the parameter variables of the model. The instance space $\mathbb{U}$ of the model is $U(\mathscr{B})^k$, that is, B^k. The parameter $\boldsymbol{y}$ is in B^l. For each $\boldsymbol{p} \in B^l$, the formula $\psi(\boldsymbol{x};\boldsymbol{y})$ defines a function $[\![\psi(\boldsymbol{x};\boldsymbol{p})]\!]^{\mathscr{B}} : B^k \longrightarrow \{0,1\}$ as follows

$$[\![\psi(\boldsymbol{x};\boldsymbol{p})]\!]^{\mathscr{B}}(u) = \begin{cases} 1 & \text{if } \mathscr{B} \vDash \psi(\boldsymbol{u};\boldsymbol{p}) \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

This function is interpreted as a concept or hypothesis over the instance space. The symbol $\vDash$ denotes satisfiability. $\mathscr{B} \vDash \psi(\boldsymbol{u};\boldsymbol{p})$ means that the background structure $\mathscr{B}$ satisfies ψ if the

[1] Local access of the Real Field, for example, means that in a abstract computation model, the algorithm can store a number in a single memory cell and has access to all the operations of the structure.

variables $\boldsymbol{x}$ and $\boldsymbol{y}$ assume the value $\boldsymbol{u}$ and $\boldsymbol{p}$ in $\mathbb{R}^k$ and $\mathbb{B}^l$ respectively. $[\![\psi(\boldsymbol{x};\boldsymbol{p})]\!]^{\mathscr{B}}$ is often called the L-definable hypothesis.

Example 1. Graphs
Let B, the background structure be an $\{E, C\}$ structure, where E is a binary and C a unary relation symbol. B can be viewed as a directed graph in which some vertices are coloured. The free variables of the FO formula that defines a universal labeling of B ranges over the set of vertices of the graph.

3.1.2 ANN in the declarative framework

A Neural Network is a network of interconnected neurons found in biological ("real") brains. Neurons are nerve cells capable of transmitting electric impulses. They interact with attached neurons via synapses. Artificial neural networks (ANNs) are abstract or mechanical realizations of (biological) neural networks. They can be regarded as rudimentary approximations of parts of the biological brain. Much of the terminology in the artificial domain has been inspired by their biological counterparts - for example, neuron and synapse. ANNs, are ubiquitous [15, 16, 17]. They have found applications in many domains – classification, localisation, and detection being a few of them.

Neurons are the primary building block of ANNs. A neuron i comprises of the following:

- set of links (synapse) characterized by weights. For the k^{th} synapse of the neuron, its input signal is denoted as x_k and the corresponding weight as w_{ik}.
- linear combiner which adds up all the input signals weighted by their corresponding synaptic weights and the bias b_i. The bias has the effect of increasing or decreasing the input, u_i, to the activation function.
- activation function for limiting the amplitude of the output y_i of the neuron.

Mathematically, to define a neuron i we need to provide the following two equations.

$$u_i = \sum_{j=1}^{N} x_j w_{ij} \tag{2}$$

$$y_i = \varphi_i(u_i + b_i) \tag{3}$$

The bias b_i can be thought of as a synapse with a fixed input $x_0 = 1$ and weight $w_{i0} = 1$. We rewrite (3) as $y_i = \varphi(v_i)$ where $v_i = u_i + b_i = \sum_{j=0}^{N} x_j w_{ij}$.

The architecture of neural networks can be extremely varied depending on the learning task and the algorithm used. The following are the three different classes of network architecture.

- Single Layer Feedforward Networks: In these kind of networks, the input layer of source nodes are connected to an output layer of neurons but not vice versa.These networks are acyclic.

- Multilayer Feedforward Networks: These networks have one or more hidden layers of computation besides the layer of output neurons. The neurons in the hidden layers are called hidden neurons or units. With more hidden layers, the network is able to extract higher-order statistics.
  The input nodes are connected to the hidden units of the first hidden layer. The output signals from this layer feeds into the next layer. Generally, neurons in a layer have as inputs output signals of the neurons of the previous layer only. The output signals of the last layer of the network (the output layer) is the output of the entire neural network. The neurons in this layer are called output neurons.

- Recurrent Networks: These networks are cyclic in nature and may contain one or more hidden layers.

Function approximation is one of the most common learning tasks for ANNs. The main idea behind function approximation is to describe complicated functions using a collection of simpler functions. Let $F(x)$ be a real valued real function accepting vectors $\boldsymbol{x} = [x_1, \dots x_k] \in \mathbb{R}^k$. Function approximation aims to describe the behaviour of $F(\boldsymbol{x})$ by a linear combination of simpler functions $f_i(\boldsymbol{x})$

$$\hat{F}(\boldsymbol{x}, \boldsymbol{w}) = \sum_{i=1}^{N} w_i f_i(\boldsymbol{x}) \tag{4}$$

3.1.3 Definitions

A ANN has a graph structure with *neurons* as its nodes and *connections* as the weighted edges. In this section, we define feed forward artificial neural networks as directed acyclic graphs. A directed acyclic graph (DAG) $G = (V, E)$ is defined as usual with a set of vertices V and a binary edge relation $E \subseteq V \times V$, with the requirement that E contains no cycles. For a DAG G, we write I^G for the set of input nodes of G defined by $I^G = \{i \in V \mid (j, i) \notin V\}$ and O^G for the set of output nodes defined by $O^G = \{i \in V \mid (i, j) \notin E$. They are also commonly called source nodes and sink nodes respectively. Where graph G is determined by context and no ambiguity arises, we drop the superscript from I and O.

Definition 1. An artificial neural network (ANN) is a tuple (G, b, Φ) consisting of

- a DAG $G = (V, E)$
- a tuple $b \in \mathbb{R}^V$ assigning to each node $i \in V$ a bias b_i; and
- $\Phi \in (\mathbb{R} \longrightarrow \mathbb{R})^V$ assigning to each node $i \in V$ an activation function $\phi_i : \mathbb{R} \longrightarrow \mathbb{R}$.

Given a ANN $A = (G, B, \Phi)$ along with $x \in \mathbb{R}^I$, an assignment of values to the input nodes of G, Given the values $\boldsymbol{x}$ and $\boldsymbol{w}$, the ANN A can be seen as computing a tuple $\boldsymbol{y} \in \mathbb{R}^{\mathbb{O}}$ where O is the set of output nodes of G, given by $y_i = n_i$. Thus, we can associate with A the function $\hat{F}_A : \mathbb{R}^I \times \mathbb{R}^E \longrightarrow \mathbb{R}^O$ defined by $\hat{F}_A(\boldsymbol{x}, \boldsymbol{w}) = \boldsymbol{y}$.

The learning task can then be understood as, for an unknown function $F : \mathbb{R}^I \longrightarrow \mathbb{R}^O$, find a suitable value for the weights w such that $\hat{F}_A(x, w)$ is a good approximation for $F(x)$. To be precise, we are given a finite training set T consisting of pairs of the form $(x, F(x))$ and the aim is to find a tuple that minimizes the error:

$$\sum_{(\boldsymbol{x},F(\boldsymbol{x}))\in T} \left|\hat{F}_A(\boldsymbol{x}, w) - F(\boldsymbol{x})\right| \tag{5}$$

Definition 2. A N-layered ANN is an artificial neural network (G, B, Φ) where the set of vertices V of the DAG $G = (V, E)$ is partitioned into sets $L_0 = I^G$, L_1, $L_2 \dots L_N = O^G$ such that for all $(u, v) \in E$, there are layers L_i, L_{i+1} s.t $u \in L_i$ and $v \in L_{i+1}$.

In the declarative framework of learning, for a given ANN, suppose that we have a fixed FOL formula $\psi(\boldsymbol{x}; \boldsymbol{w})$. The task is to find a tuple $\boldsymbol{v} \in \mathbb{R}^E$ such that the function $[\![\psi(\boldsymbol{x}; \boldsymbol{v})]\!]^{\mathscr{R}}$, defined as in Eq.1, but over $\mathscr{R}$ as the background structure, where $\boldsymbol{u}$ is the input, is consistent with the training set T (or at least approximately consistent).

3.2 Modal Logic in declarative framework of learning

In the context of declarative framework of learning, this proposal investigates two issues in the field: characterizing learning over the Real Field as a background structure, and identifying suitable logics. In the following, I address the latter.

[9] suggests exploring the suitability of logics like second-order, modal and temporal in the declarative framework of learning as a potential line of investigation. Modal Logics feature modalities (though $\Diamond$ and $\Box$) that capture dependence of knowledge on parameters such as information state of agents, time, and place. Modal logic (ML) has been studied extensively under philosophical logic since the early 1900s. The connections between ML and certain areas of theoretical computer science were explored much later. Van Benthem and later others, studied ML from a “classical model theory” perspective. It has been shown that ML satisfies preservation theorems that are analogous to classical theorems for first-order logic, FOL [18]. This is an anomaly because expressive logics [19], generally, do not.

[20] proposed a novel methodology for modal semantics, now known as “Kripke Semantics”. Kripke’s model is motivated by Leibniz’s idea that a statement is necessarily true if it is true in “all possible worlds.” In this model, the worlds are related by an accessibility relation R and the statement necessarily φ is true in world w (or a description state s) if and only if ψ is true at all worlds w' accessible from w. Formally, a Kripke model $K = (W, R, V)$ where W is a non-empty set of “worlds”, R is a binary accessibility relation on W, and V is a function assigning to each propositional variable p a set $V(p)$ of possible worlds. (K, w) denotes a world w in the model K.

Locality properties in FOL can be thought of as expressing some kind of indistinguishability. This notion is captured in Modal Logics using bisimilarity [11]. For two Kripke structures $K_1 = (W_1, R_1, V_1)$ and $K_2 = (W_2, R_2, V_2)$, (K_1, w_1) and (K_2, w_2) are bisimilar if there is a binary relation $R \subseteq S_1 \times S_2$ such that $(w_1, w_2) \in R$; if $(p, q) \in R$ then $V_1(p) \iff V_2(q)$ for all $a \in \Sigma$; if $(p, q) \in R$

and $(p, r) \in R_1$ then there is $s \in W_2$ such that $(r, s) \in R$ and $(q, s) \in R_2$; if $(p, q) \in R$ and $(q, s) inR_2$ then there is $r \in W_1$ such that $(r, s) \in R$ and $(p, r) \in R_1$.

[21] studies expressibility and definability in modal logics by investigating specific classes of structures defined through constraints on underlying frames. [22] presents normal forms for formuals of different modal systems. These provide insight into the properties of modal logic that will be used in my research.

4 Objectives

This proposal has two goals:

- First, establish the learnability results, stated at the onset, with the real field as a background structure;
- the second strand of this research addresses extending the learnability result in [10] to a counting extension of FOL, the logic introduced in [7] in particular, as well as Modal Logic.

4.1 Deliverables

I want to write an elaborate survey of the existing literature besides attending workshops, conferences, and relevant lectures at the University to stay updated on new methods and results in the field. In keeping with the motivation and significance of this project as stated above, I set the following target end results by the end of the first year.

- a paper on learnability in $\mathscr{L}^{\circ}_{\infty\omega}(C)$ and Modal Logic over finite background structures
- a paper on the generalised complexity bounds for learning over the field of the Real Numbers
- a report on novel lines of investigation in the field and their significance

References

[1] J. M. Kanter and K. Veeramachaneni. "Deep feature synthesis: Towards automating data science endeavors". In: *2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. 2015, pp. 1–10.

[2] R. Burbidge et al. "Drug design by machine learning: support vector machines for pharmaceutical data analysis". In: *Computers And Chemistry* 26.1 (2001), pp. 5–14.

[3] A. Musaev, D. Wang, and C. Pu. "LITMUS: A Multi-Service Composition System for Landslide Detection". In: *IEEE Transactions on Services Computing* 8.5 (Sept. 2015), pp. 715–726.

[4] John McCarthy. “Circumscription- A form of non-monotonic reasoning”. In: *Artificial Intelligence* 13.1 (1980). Special Issue on Non-Monotonic Logic, pp. 27–39.

[5] Kousha Etessami. “Counting Quantifiers, Successor Relations, and Logarithmic Space”. In: *Journal of Computer and System Sciences* 54.3 (1997), pp. 400–411. ISSN: 0022-0000. DOI: 10.1006/jcss.1997.1485.

[6] Lauri Hella. “Logical Hierarchies in PTIME”. In: *Inf. Comput.* 129.1 (Aug. 1996), pp. 1–19.

[7] Leonid Libkin. “Logics with Counting and Local Properties”. In: *ACM Trans. Comput. Logic* 1.1 (July 2000), pp. 33–59. ISSN: 1529-3785.

[8] M. Grohe and G. Turán. “Learnability and Definability in Trees and Similar Structures”. In: *Theory of Computing Systems* 37.1 (Jan. 2004), pp. 193–220.

[9] M. Grohe and M. Ritzert. “Learning first-order definable concepts over structures of small degree”. In: *2017 32nd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*. 2017, pp. 1–12.

[10] Martin Grohe, Christof Löding, and Martin Ritzert. “Learning MSO-definable hypotheses on string”. In: (Aug. 2017).

[11] Leonid Libkin. *Elements Of Finite Model Theory (Texts in Theoretical Computer Science. An Eatcs Series)*. SpringerVerlag, 2004. ISBN: 3540212027.

[12] Avrim Blum, John Hopcroft, and Ravindran Kannan. “Foundations of Data Science”. In: (). URL: https://www.cs.cornell.edu/jeh/book.pdf.

[13] Wojciech Kazana and Luc Segoufin. “First-order query evaluation on structures of bounded degree”. In: *Logical Methods in Computer Science* 7.2 (2011). DOI: 10.2168/LMCS-7(2:20)2011. URL: https://doi.org/10.2168/LMCS-7(2:20)2011.

[14] Erich Grädel. *Finite model theory and its applications*. Berlin New York: Springer, 2007.

[15] M. R. G. Meireles, P. E. M. Almeida, and M. G. Simoes. “A comprehensive review for industrial applicability of artificial neural networks”. In: *IEEE Transactions on Industrial Electronics* 50.3 (2003), pp. 585–601.

[16] Bing Cheng and D. M. Titterington. “[Neural Networks: A Review from Statistical Perspective]: Rejoinder”. In: *Statist. Sci.* 9.1 (Feb. 1994), pp. 49–54.

[17] A. Graves, A.-r. Mohamed, and G. Hinton. “Speech Recognition with Deep Recurrent Neural Networks”. In: *ArXiv e-prints* (Mar. 2013). arXiv: 1303.5778.

[18] Eric Rosen. “Modal Logic over Finite Structures”. In: *Journal of Logic, Language and Information* 6.4 (Oct. 1997), pp. 427–439. ISSN: 1572-9583. DOI: 10.1023/A:1008275906015. URL: https://doi.org/10.1023/A:1008275906015.

[19] Yuri Gurevich. “Toward logic tailored for computational complexity”. In: *Computation and Proof Theory: Proceedings of the Logic Colloquium held in Aachen, July 18–23, 1983 Part II*. Ed. by Egon Börger et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 1984, pp. 175–216. ISBN: 978-3-540-39119-7. DOI: 10.1007/BFb0099486. URL: https://doi.org/10.1007/BFb0099486.

[20] Saul A. Kripke. "A Completeness Theorem in Modal Logic". In: *J. Symbolic Logic* 24.1 (Mar. 1959), pp. 1–14. URL: https://projecteuclid.org:443/euclid.jsl/1183733464.

[21] Anuj Dawar and Martin Otto. "Modal characterisation theorems over special classes of frames". In: *Annals of Pure and Applied Logic* 161.1 (2009), pp. 1–42. ISSN: 0168-0072. DOI: https://doi.org/10.1016/j.apal.2009.04.002. URL: http://www.sciencedirect.com/science/article/pii/S016800720900092X.

[22] Kit Fine. "Normal forms in modal logic." In: *Notre Dame J. Formal Logic* 16.2 (Apr. 1975), pp. 229–237. DOI: 10.1305/ndjfl/1093891703. URL: https://doi.org/10.1305/ndjfl/1093891703.