Computabilities of Validity and Satisfiability in Probability Logics over Finite and Countable Models

Greg Yang *

December 15, 2014

Abstract

The ϵ-logic (which is called ϵE-logic in this paper) of Kuyper and Terwijn is a variant of first order logic with the same syntax, in which the models are equipped with probability measures and in which the $\forall x$ quantifier is interpreted as "there exists a set A of measure $\geq 1-\epsilon$ such that for each $x \in A, \ldots$." Previously, Kuyper and Terwijn proved that the general satisfiability and validity problems for this logic are, i) for rational $\epsilon \in (0,1)$, respectively Σ_1^1-complete and Π_1^1-hard, and ii) for $\epsilon = 0$, respectively decidable and Σ_1^0-complete. The adjective "general" here means "uniformly over all languages."

We extend these results in the scenario of finite models. In particular, we show that the problems of satisfiability by and validity over finite models in ϵE-logic are, i) for rational $\epsilon \in (0,1)$, respectively Σ_1^0- and Π_1^0-complete, and ii) for $\epsilon = 0$, respectively decidable and Π_1^0-complete. Although partial results toward the countable case are also achieved, the computability of ϵE-logic over countable models still remains largely unsolved. In addition, most of the results, of this paper and of Kuyper and Terwijn, do not apply to individual languages with a finite number of unary predicates. Reducing this requirement continues to be a major point of research.

On the positive side, we derive the decidability of the corresponding problems for monadic relational languages — equality- and function-free languages with finitely many unary and zero other predicates. This result holds for all three of the unrestricted, the countable, and the finite model cases.

Applications in computational learning theory (CLT), weighted graphs, and artificial neural networks (ANN) are discussed in the context of these decidability and undecidability results.

1 Introduction

In the new age of "big data," machine learning and statistical inference have been increasingly applied in the technology sector, and more resources than ever before are poured into advancing our understanding of these techniques. One approach to this end is to reconcile the inductive nature of machine learning with the deductive discipline of logic. Previous attempts include one by computer scientist Leslie Valiant , the creator of the PAC (Probably Approximately Correct) model of computational learning theory (CLT). In *Robust Logics* [26], he tried to combine the PAC model with a fragment of first order logic (FOL) in the context of finite models. The logician H. Jerome Keisler in [7] also investigated a variant of FOL with probabilistic quantifiers of the form $(Px \geq r)$ meaning "holds for x in a set of measure at least r."

Most recently, Terwijn and Kuyper [22] invented a probability logic with a fixed error parameter, called ϵ-logic (or, in this paper, ϵE-logic), that is inspired by features of both Valiant and Keisler's work. This ϵE-logic uses the same syntax as FOL and differ only in that 1) the models are given

*email: gyang@college.harvard.edu

probability measures, and 2) the $\forall$ quantifier has the interpretation of "holds for x in a set of measure at least $1-\epsilon$." In particular, the $\exists$ quantifier keeps the same, non-probabilistic interpretation as in first order logic.

Such an unusual, asymmetric definition was motivated by the key property of ϵE-logic to be learnable through examples, in a sense related to Valiant's PAC-learning model [11, Thm. 2.3.3]: roughly, for any desired error bound ϵ and an example oracle that emits elements of the universe M according to a distribution $\mathcal{D}$, we can learn in time polynomial in $\frac{1}{\epsilon}$ whether $(\mathcal{M}, \mathcal{D}) \models_\epsilon \phi$ or $(\mathcal{M}, \mathcal{D}) \models_\epsilon \neg\phi$. Thus, ϵE-logic has an *inductive* property, in addition to promises of *deductive* properties that would seem to carry over from classical first order logic.

However, it turns out that deductive reasoning in ϵE-logic is computationally much harder than first order logic in the general case. In fact its complexity does not even reside in the arithmetic hierarchy, but rather the analytic one. As a result, there is no algorithm to decide (uniformly over all first order languages) whether a given sentence is valid or satisfiable. The following table summarizes the current knowledge on the satisfiability and validity [1] problems of ϵE-logic (defined below in (2.1.6) and (2.1.5)). Each tuple $\langle n_1, n_2, \ldots \rangle$ represents that the corresponding result requires the language to have at least n_1 unary predicates, at least n_2 binary predicates, and so on. In particular, the value of ω means an infinite number of the corresponding type of predicate is necessary. The empty tuple $\langle \rangle$ means that any language suffices. Finally $\langle * \rangle$ denotes that the set of ϵE-valid sentences coincides with the set of valid sentences in first order logic, so any language that admits a decision algorithm in FOL will do so in ϵE-logic as well. As general FOL validity is Σ^0_1-complete, this means that general ϵE-validity is also Σ^0_1-complete.

	$\epsilon \in (0,1) \cap \mathbb{Q}$	$\epsilon = 0$
ϵE-satisfiability	Σ^1_1-complete $\langle \omega, 6, 2 \rangle$ [10, thm 7.6]	decidable $\langle \rangle$ [10, thm 6.7]
ϵE-validity	Π^1_1-hard $\langle \omega, 3, 2 \rangle$ [12, thm 4.2]	Σ^0_1-complete $\langle * \rangle$ [23, prop 3.2]

Table 1: Current knowledge on general ϵE-satisfiability and ϵE-validity [10, Table 1]

It is still open, however, whether a smaller fragment of ϵE-logic — for example, languages with a finite number of unary predicates, or languages with a single binary predicate — admits an easier complexity.

Notice that the results are not symmetrical as in the case of first order logic, where ϕ is valid iff $\neg\phi$ is not satisfiable. Indeed, ϵE-logic is *paraconsistent* [11, Prop. 2.2.1], because it's possible for $(\mathcal{M}, \mathcal{D}) \models_\epsilon \forall x \phi(x)$ and $(\mathcal{M}, \mathcal{D}) \models_\epsilon \exists x \neg\phi(x)$ to hold at the same time (for example if the set of x satisfying $\neg\phi$ has measure 0 but is not empty).

In this paper, we answer the counterpart questions for ϵE-satisfiability by and validity over finite models and in some cases, countable models (see (2.3.5) for definitions). As noted above, we cannot in general answer the satisfiability question by just answering the validity question, nor vice versa.

In first order logic, Trachtenbrot's theorem [15] asserts that, perhaps counterintuitively, assessing the validity of a theorem over only finite models is Π^0_1-hard. Therefore we do not even have a deductive calculus for this task.

We show that other than the case of $\epsilon = 0$, Trachtenbrot's theorem holds also for ϵE-logic. More precisely, we will establish the following characterizations.

Here, the tuple notation $\langle n_1, n_2, \ldots \rangle$ denotes language requirements, as in the last table, but $\langle * \rangle$ means that the set of ϵE-valid sentences over *finite models* coincides with the corresponding set *over finite models* with regard to FOL.

Like in the unrestricted case, it is still open whether the conditions on signature can be significantly weakened.

[1] This notion of validity is over all probability models. It is called *normal ϵE-validity* in this paper. See definition (2.1.5) and [10, remark before thm 2.6]

	$\epsilon \in (0,1) \cap \mathbb{Q}$	$\epsilon = 0$
finite ϵE-satisfiability	Σ^0_1-complete $\langle \omega, 3 \rangle$ (3.4.9)	decidable $\langle\rangle$ (3.1.3)
finite ϵE-validity	Π^0_1-complete $\langle \omega, 1 \rangle$ (3.5.9)	Π^0_1-complete $\langle * \rangle$ (3.5.9)

Table 2: Summary of results in this paper: the finite model case

Hence, other than the case of finite 0E-satisfiability, the ϵE-satisfiability and ϵE-validity problems in the finite model scenario are as hard as the corresponding problems in ordinary first order logic [15, p. 166]. Therefore, no general deduction mechanism exists for theorems over finite models.

In contrast, for $\epsilon \in [0, 1)$ rational, we show that these problems are decidable over monadic relational languages, which are languages with only unary predicates and no function symbols or equality. This mirrors the characterizations for the corresponding FOL fragments.

Despite these successes, the countable case remains largely unsolved. While models of ϵE-logic have a downward Lowenheim-Skolem theorem transforming them into equivalent continuum-sized models [13, Thm. 4.6], this theorem does not hold when "continuum-sized" is replaced with "countable." So unlike FOL, the set of sentences ϵE-valid over all countable models does not coincide a priori with the set of those over all models.

In the format of the previous tables, we summarize the current knowledge on countable ϵE-validity and satisfiability in table (3).

	$\epsilon \in (0,1) \cap \mathbb{Q}$	$\epsilon = 0$
countable ϵE-satisfiability	unknown	decidable $\langle\rangle$ (3.1.3)
countable ϵE-validity	Σ^0_1-hard $\langle \omega, 1 \rangle$ (3.5.10)	Σ^0_1-complete $\langle * \rangle$ (3.5.4)

Table 3: Summary of results in this paper: the countable case

The outline of the paper is thus: After introducing some notations and prerequisites, we review the basic definitions of ϵE-logic, ϵ-model, the validity and satisfiability problems, and other related notions in section (2). We then define in section (2.2) the dual logic, *ϵF-logic*, in whose terms we phrase many of our results. Briefly, in ϵF-logic, the syntax is once again identical to that of FOL, but the quantifier $\exists x$ is interpreted to mean "there exists a set A of measure $> \epsilon$ such that for each $x \in A, \dots$." Having laid out the analogue concepts over finite models, we dive straight into examples and applications in section (2.4), hoping to motivate our main theorems and future research.

There, we employ ϵE-logic and ϵF-logic to

- model the *approximation concept existence* assumption of PAC learning;
- develop rudimental theories of graphs with weighted vertices and graphs with weighted edges; and
- compute the *linear threshold update rule* in artificial neural networks.

With these examples in mind, we begin our deductions. In section (3.1), we prove the decidability of 0E-satisfiability over both finite models and countable models. In section (3.2), we show that the satisfiability and validity of sentences in any monadic relational language both reduce to linear programming, and thus are decidable whenever $\epsilon \in \mathbb{Q}$. During the development of the reduction, we introduce trees as semantic tools. These ideas are extended in section (3.3) to define the semantics of *q-sentences*, which generalize both ϵE- and ϵF-logics by allowing all forms of quantifiers. This new form of sentences allows us to rigorously state the inter-reduction results of Kuyper and Terwijn between different rational ϵ parameters. Equipped aptly with powerful machinery, in section (3.4) we tackle the Σ^0_1-completeness of the ϵE-satisfiability problem over finite models. The hardness proof involves a painstaking encoding of the halting set in a suitable language. The definability proof utilizes a perturbation lemma that simplifies satisfiability to that by models with rationally-valued distributions. Last but not least, we show in section (3.5) that 0E-validity coincides with FOL validity when both are restricted to finite or countable models. (Of course, in the countable case, the "countably FOL valid" sentences are just the unrestricted valid sentences by Lowenheim-Skolem). With Kuyper's inter-reduction theorem, we deduce that ϵE-validity for finite models is Π^0_1-complete and that for countable models is Σ^0_1-hard for any strong enough language. Finally, we wrap up and mention possible future directions of research in section (4).

1.1 Notation and Prerequisites

Sets of the form $\{1, 2, \ldots, n\}$ will be abbreviated $[\![n]\!]$.

1.1.1 Logic

Script upper case letters $\mathcal{M}$ and $\mathcal{N}$ are used to denote first order models. Their underlying universes are written M and N. All first order languages are assumed to have an at most countable signature.

$\vec{x}$ denotes a finite sequence of variables or parameters. $|\vec{x}|$ denotes the length of this sequence. $\phi(\vec{x}, \vec{y})$ will always represent a formula with free variables $x_1, \ldots, x_n$ and $y_1, \ldots, y_m$, possibly with other bound variables. In the context of a first order model $\mathcal{M}$,

$$\phi(\vec{x}, \vec{y}; \vec{p}) = \phi(x_1, \ldots, x_n, y_1, \ldots, y_m; p_1, \ldots, p_n),$$

represents a formula with free variables $\vec{x}, \vec{y}$ and parameters $p_i \in M$. $\forall\vec{x}$ is the shorthand for the quantifier block $\forall x_1 \forall x_2 \cdots \forall x_n$. Similarly, $\exists\vec{x}$ is the shorthand for the quantifier block $\exists x_1 \exists x_2 \cdots \exists x_n$.

If ϕ is a (formal or informal) sentence, then $\|\phi\| \in \{0, 1\}$ denotes its truth value.

In formulas, we adopt the convention that $\wedge$ is parsed before $\vee$ when written without parentheses. For example,

$$A(x, y) \wedge B(y, z) \vee R(z) \wedge x = z$$

is parsed as

$$(A(x, y) \wedge B(y, z)) \vee (R(z) \wedge x = z).$$

Formulas of the form

$$\phi_1 \wedge \phi_2 \wedge \cdots \wedge \phi_k \to \psi$$

are parsed as

$$(\phi_1 \wedge \phi_2 \wedge \cdots \wedge \phi_k) \to \psi.$$

In addition, we will use square brackets [] in place of parentheses () when doing so improves the readability.

A subset $A \subseteq \mathbb{N}$ is called Σ^0_1 or Σ^0_1-definable if A is the range of a recursive function. A subset $A \subseteq \mathbb{N}$ is called Σ^0_1-hard if for every Σ^0_1 set A', there is a computable many-one reduction from A' to A. A subset $C \subseteq \mathbb{N}$ is called Σ^0_1-complete if C is both Σ^0_1-definable and Σ^0_1-hard.

Dually, a subset $B \subseteq \mathbb{N}$ is called Π^0_1 or Π^0_1-definable (resp. Π^0_1-hard and resp. Π^0_1-complete) if its complement $\mathbb{N} - B$ is Σ^0_1 (resp. Σ^0_1-hard and resp. Σ^0_1-complete).

Please refer to Soare [21] for unexplained concepts in computability.

1.1.2 Measure Theory

Let X be a set. $\mathfrak{P}(X)$ denotes the power set of X. By *a measure μ on X*, we mean a set function $\mu : \mathcal{A} \to [0, \infty]$ that is defined and countably additive on some σ-algebra $\mathcal{A} \subseteq \mathfrak{P}(X)$. The triple $(X, \mathcal{A}, \mu)$ is called a *measure space.* Similarly, by *a finitely additive measure μ on X*, we mean a set function $\mu : \mathcal{A} \to [0, \infty]$ defined on some Boolean algebra $\mathcal{A} \subseteq \mathfrak{P}(X)$, and μ is finitely additive on

$\mathcal{A}$. In both cases, the σ-algebra or Boolean algebra $\mathcal{A}$ on which μ is defined is denoted $\operatorname{dom}\mu$. We say that *μ is everywhere defined* if $\operatorname{dom}\mu = \mathfrak{P}(X)$. We say that *a measure μ on X is extended by μ'* if $\operatorname{dom}\mu' \supseteq \operatorname{dom}\mu$. The μ-measure of a set of elements satisfying some condition ϕ is denoted

$$\mu(x : \phi(x)).$$

When we say "A has measure at least 1/2," we implicitly assume A is first of all measurable, and then that it has measure at least 1/2.

Primarily we will be discussing *probability measures*, i.e. measures μ on X with $\mu(X) = 1$. We use upper case script letters starting from $\mathcal{D}, \mathcal{E}$, etc to name them. In the contexts of probability measures, we will also use the probability notation $\Pr_{x\in\mathcal{D}}[\phi(x)]$ interchangeably with $\mathcal{D}(x : \phi(x))$.

Please refer to Bogachev [4] for unexplained concepts in measure theory.

1.1.3 Linear Programming

A *linear program* is a triple $L = (\vec{x}, E, f)$ where

- $\vec{x}$ is a set of $|\vec{x}| = k$ variables,
- E is a set of $|E| = n$ (weak) linear inequalities

$$e^i : \sum_{j=1}^{k} c^i_j x_j \geq d^i,$$

  and

- f is a linear function, called the *object function*,

$$f(\vec{x}) = \sum_{j=1}^{x} q_j x_j.$$

In general, if assignment $\vec{x} = \vec{a}$ satisfies all inequalities e_i, then we write

$$C\vec{x} \geq \vec{d}.$$

Here C is the matrix $\{c^i_j\}_{i,j}$ with row vectors $\vec{c^i} = (c^i_1, \ldots, c^i_k)$, and $\vec{d} = (d^1, \ldots, d^n)$.

Because each equality $\sum p_j x_j = r$ can be written as two weak inequalities, we can allow E to contain equations as well.

To *maximize* L is to find

$$\max(L) := \max_{C\vec{x}\geq\vec{d}} f(\vec{x}).$$

Likewise, to *minimize* L is to find

$$\min(L) := \min_{C\vec{x}\geq\vec{d}} f(\vec{x}).$$

L is said to be *feasible* if $\{\vec{x} : C\vec{x} \geq \vec{d}\}$ is nonempty. In other words, L is feasible iff $\max(L) > -\infty$ iff $\min(L) < \infty$.

In this paper, we are mainly concerned with the feasibility problem of linear programs. As such, we conveniently identify each program L with its set of inequalities E.

In the *arithmetic model of computation*, the arithmetic operations of addition, multiplication, subtraction, division, and comparison are assumed to take unit time. It is known that maximizing, minimizing, and finding the feasibility of a linear program is polynomial time in the arithmetic model [17]. Because all such arithmetic operations on rational numbers are decidable,

Proposition 1.1.1. *The feasibility problem of linear programs with rational coefficients is decidable.*

We will also briefly cross path with *strict linear programs*. These are linear programs $L = (\vec{x}, E, f)$ where E consists of strict inequalities or equalities but with no weak inequalities.

Please refer to Schrijver [19] for unexplained concepts with regard to linear programming.

2 Probability Logics

Here we formalize the ideas touched in the introduction. First, we revisit the definitions of ϵE-logic, ϵE-model, and related ideas as defined by Terwijn and Kuyper. Then, we introduce ϵF-logic, the dual logic of ϵE-logic. Finally, an abundance of examples and applications are provided to clarify these ideas.

2.1 E-logic

Let

- $\mathcal{L}$ be a first order language, possibly containing equality, of a countable signature;
- $\epsilon \in [0, 1]$;
- $\mathcal{M}$ be a first-order model with universe M;
- $\mathcal{D}$ be a probability measure on M defined on some σ-algebra $\operatorname{dom}\mathcal{D} \subseteq \mathfrak{P}(M)$.

Definition 2.1.1 (ϵE-truth)**.** [2] Let $\phi(\vec{x}; \vec{p}) = \phi(x_1, \ldots, x_n; p_1, \ldots, p_n)$ represent a formula with variables $\vec{x}$ and parameters $p_i \in \mathcal{M}$. We define the notion of ϵ**E-truth**, denoted $(\mathcal{M}, \mathcal{D}) \models_\epsilon \phi$, inductively as follows:

1. For every atomic formula $\phi(\vec{x}; \vec{p})$:
$$(\mathcal{M}, \mathcal{D}) \models_\epsilon \phi(\vec{x}; \vec{p}) \iff \mathcal{M} \models \phi(\vec{x}; \vec{p}).$$
That is, for *all* tuples $(a_1, a_2, \ldots, a_n) \in \mathcal{M}$, $\phi(\vec{a}; \vec{p})$ holds.

2. We treat the logical connectives $\wedge$ and $\vee$ classically. For example, for $\vec{x}, \vec{y}, \vec{z}$ distinct sequences of variables,
$$(\mathcal{M}, \mathcal{D}) \models_\epsilon \phi(\vec{x}, \vec{z}; \vec{p}) \wedge \psi(\vec{y}, \vec{z}; \vec{p})$$
iff for all $\vec{a} \in M^{|\vec{x}|}, \vec{b} \in M^{|\vec{y}|}, \vec{c} \in M^{|\vec{z}|}$,
$$(\mathcal{M}, \mathcal{D}) \models_\epsilon \phi(\vec{a}, \vec{c}; \vec{p}) \wedge \psi(\vec{b}, \vec{c}; \vec{p})$$

3. The existential quantifier is treated classically:
$$(\mathcal{M}, \mathcal{D}) \models_\epsilon \exists \vec{x} \phi(\vec{x}, \vec{y}; \vec{p})$$
iff there exists $\vec{a} \in M^{|\vec{x}|}$ such that
$$(\mathcal{M}, \mathcal{D}) \models_\epsilon \phi(\vec{a}, \vec{y}; \vec{p}).$$

4. The universal quantifier is interpreted probabilistically:
$$(\mathcal{M}, \mathcal{D}) \models_\epsilon \forall x \phi(x, \vec{y}; \vec{p}) \iff \Pr_{a \sim \mathcal{D}}[(\mathcal{M}, \mathcal{D}) \models_\epsilon \phi(a, \vec{y}; \vec{p})] \geq 1 - \epsilon.$$
Note that the universal quantifier in this definition binds a single variable x; in general it's *not* true that
$$(\mathcal{M}, \mathcal{D}) \models_\epsilon \forall \vec{x} \phi(\vec{x}, \vec{y}; \vec{p}) \iff \Pr_{\vec{a} \sim \mathcal{D}^{|\vec{x}|}}[(\mathcal{M}, \mathcal{D}) \models_\epsilon \phi(\vec{a}, \vec{y}; \vec{p})] \geq 1 - \epsilon.$$

5. The case of negation is split in subcases as below:

[2] adapted from [10]

(a) For ϕ atomic, $(\mathcal{M},\mathcal{D}) \models_\epsilon \neg\phi(\vec{x};\vec{p}) \iff (\mathcal{M},\mathcal{D}) \not\models_\epsilon \phi(\vec{x};\vec{p})$.

(b) $\neg$ distributes classically over $\wedge$ and $\vee$, e.g.

$$(\mathcal{M},\mathcal{D}) \models_\epsilon \neg(\phi(\vec{x},\vec{z};\vec{p}) \wedge \psi(\vec{y},\vec{z};\vec{p})) \iff (\mathcal{M},\mathcal{D}) \models_\epsilon \neg\phi(\vec{x},\vec{z};\vec{p}) \vee \neg\psi(\vec{y},\vec{z};\vec{p}).$$

(c) $(\mathcal{M},\mathcal{D}) \models_\epsilon \neg\neg\phi(\vec{x};\vec{p}) \iff (\mathcal{M},\mathcal{D}) \models_\epsilon \phi(\vec{x};\vec{p})$.

(d) $(\mathcal{M},\mathcal{D}) \models_\epsilon \neg\exists x\phi(x,\vec{y};\vec{p}) \iff (\mathcal{M},\mathcal{D}) \models_\epsilon \forall x\neg\phi(x,\vec{y};\vec{p})$.

(e) $(\mathcal{M},\mathcal{D}) \models_\epsilon \neg\forall x\phi(x,\vec{y};\vec{p}) \iff (\mathcal{M},\mathcal{D}) \models_\epsilon \exists x\neg\phi(x,\vec{y};\vec{p})$.

6. The implication symbol $\rightarrow$ reduces to boolean combinations classically:

$$(\mathcal{M},\mathcal{D}) \models_\epsilon \phi(\vec{x},\vec{z};\vec{p}) \rightarrow \psi(\vec{y},\vec{z};\vec{p}) \iff (\mathcal{M},\mathcal{D}) \models_\epsilon \neg\phi(\vec{x},\vec{z};\vec{p}) \vee \psi(\vec{y},\vec{z};\vec{p}).$$

7. The equivalence symbol $\leftrightarrow$ reduces to the conjunction of two implications:

$$(\mathcal{M},\mathcal{D}) \models_\epsilon \phi(\vec{x},\vec{z};\vec{p}) \leftrightarrow \psi(\vec{y},\vec{z};\vec{p})$$

iff

$$(\mathcal{M},\mathcal{D}) \models_\epsilon [\phi(\vec{x},\vec{z};\vec{p}) \rightarrow \psi(\vec{y},\vec{z};\vec{p})] \wedge [\psi(\vec{x},\vec{z};\vec{p}) \rightarrow \phi(\vec{y},\vec{z};\vec{p})]$$

This logic system is called ϵ**E-logic**. When referring to the set of all such logics for $\epsilon \in [0,1]$ or when ϵ is a fixed parameter implicit in the context, we simply use the term **E-logic**.

To make sure that the $\forall$ quantifier makes sense, we need to impose measurability conditions on definable sets. In this paper, *classical models* refer to the models used in ordinary first-order logic. They are distinct from the concept defined here:

Definition 2.1.2. Let $\mathcal{L}$ be a first order language of a countable signature, possibly containing equality, and let $\epsilon \in [0,1]$. Then an ϵ**E-model** for the language $\mathcal{L}$ consists of a classical first-order $\mathcal{L}$-model $\mathcal{M}$ together with a probability measure $\mathcal{D}$ over $\mathcal{M}$ such that:

1. For all formulas $\phi = \phi(x_1,\ldots,x_n)$ and all $a_1,\ldots,a_{n-1} \in \mathcal{M}$, the set

$$\{a_n \in \mathcal{M} : (\mathcal{M},\mathcal{D}) \models_\epsilon \phi(a_1,\ldots,a_n)\}$$

is $\mathcal{D}$-measurable.

2. All relations of arity n are $\mathcal{D}^n$-measurable (including equality, if it is in $\mathcal{L}$), and all functions of arity n are measurable as functions from $(\mathcal{M}^n,\mathcal{D}^n)$ to $(\mathcal{M},\mathcal{D})$. In particular, constants are $\mathcal{D}$-measurable.

A **probability model** is a pair $(\mathcal{M},\mathcal{D})$ that is an ϵE-model for every $\epsilon \in [0,1]$.

Definition 2.1.3. Two ϵE-models $(\mathcal{M},\mathcal{D})$ and $(\mathcal{N},\mathcal{E})$ are ϵ**-elementarily equivalent**, denoted by

$$(\mathcal{M},\mathcal{D}) \equiv_\epsilon (\mathcal{N},\mathcal{E}),$$

iff for every formula ϕ,

$$(\mathcal{M},\mathcal{D}) \models_\epsilon \phi \iff (\mathcal{N},\mathcal{E}) \models_\epsilon \phi.$$

Definition 2.1.4. Two formulas ϕ and ψ are **ϵ-equivalent**, denoted by

$$\phi \equiv_\epsilon \psi$$

iff for every ϵE-model $(\mathcal{M}, \mathcal{D})$,

$$(\mathcal{M}, \mathcal{D}) \models_\epsilon \phi \iff (\mathcal{M}, \mathcal{D}) \models_\epsilon \psi.$$

ϕ and ψ are called **(semantically) equivalent**, written

$$\phi \equiv \psi,$$

if $\phi \equiv_\epsilon \psi$ for all $\epsilon \in [0, 1]$.

Definition 2.1.5. Let ϕ be a first order sentence. We say that ϕ is **ϵE-valid** if for all ϵE-models $(\mathcal{M}, \mathcal{D})$, $(\mathcal{M}, \mathcal{D}) \models_\epsilon \phi$. ϕ is **normally ϵE-valid** iff for all *probability models* $(\mathcal{M}, \mathcal{D})$, $(\mathcal{M}, \mathcal{D}) \models_\epsilon \phi$.

Similarly we define

Definition 2.1.6. A sentence ϕ is said to be **ϵE-satisfiable** if there exists an *ϵE-model* $(\mathcal{M}, \mathcal{D})$ such that $(\mathcal{M}, \mathcal{D}) \models_\epsilon \phi$.

To distinguish between these concepts and the analogue concepts over finite and countable ϵE-models, we also prefix these terms with *unrestricted* or postfix them with *in the unrestricted case*. For example, ϵE-valid means the same thing as *unrestricted ϵE-valid*, or as *ϵE-valid in the unrestricted case*.

Likewise, to make the distinction clear from the corresponding notions in FOL, we say

Definition 2.1.7. A formula ϕ is **classically valid** if every first order model satisfies ϕ.
A formula ϕ is **classically satisfiable** if some first order model satisfies ϕ.

We could define *normally ϵE-satisfiability* in analogy to normally ϵE-validity, but this concept would be equivalent to ϵE-satisfiability [10, Thm 2.6, Prop. 2.7].

Finally, we record the following proposition which will often be implicitly applied.

Proposition 2.1.8 (Terwijn [22])**.** *Every formula ϕ is semantically equivalent to a formula ϕ' in prenex normal form.*

2.2 The Dual Logic, F-logic

Definition 2.2.1. Let $\mathcal{L}$ be a countable first order language, possibly containing equality. Let $\Phi = \Phi(x_1, \ldots, x_n)$ be a first order formula in the language $\mathcal{L}$, and let $\epsilon \in [0, 1]$. If $(\mathcal{M}, \mathcal{D})$ is an ϵE-model, then we define **ϵF-truth**, written $(\mathcal{M}, \mathcal{D}) \vdash_\epsilon \Phi$, by

$$(\mathcal{M}, \mathcal{D}) \vdash_\epsilon \Phi \iff (\mathcal{M}, \mathcal{D}) \not\models_\epsilon \neg\Phi.$$

We call the logic under $\vdash_\epsilon$ **ϵF-logic**.

Suppose for quantifiers $\triangledown_1, \triangledown_2, \ldots, \triangledown_n \in \{\forall, \exists\}$ and a quantifier free formula ψ,

$$\Phi(\vec{y}; \vec{p}) := \triangledown_1 x_1 \triangledown_2 x_2 \cdots \triangledown_n x_n \psi(x_1, x_2, \ldots, x_n, \vec{y}; \vec{p}).$$

Then if $\triangledown'_i$ denotes the dual quantifier of $\triangledown_i$ (interchange $\exists$ with $\forall$),

$$\neg\Phi(\vec{y}; \vec{p}) \equiv_\epsilon \triangledown'_1 x_1 \triangledown'_2 x_2 \cdots \triangledown'_n x_n \neg\psi(x_1, x_2, \ldots, x_n, \vec{y}; \vec{p}).$$

For example, if $\triangledown_i$ is $\forall$ for odd i and $\exists$ for even i, and n is odd, then $(\mathcal{M}, \mathcal{D}) \not\models_\epsilon \neg\Phi$ iff for all x_1,

there exists a set of x_2 with measure strictly greater than ϵ such that,

for all x_3,

$\vdots$

for all x_n,

$\psi(x_1, x_2, \ldots, x_n)$ holds.

With this remark, it's easy to see that ϵF-logic is the *dual logic* of ϵE-logic, in that the quantifier $\forall$ is interpreted classically, while the quantifier $\exists$ is interpreted as "with measure strictly greater than ϵ."

More formally and in parallel with the inductive definition given for E-logic, we can write

1. For every atomic formula $\phi(\vec{x}; \vec{p})$:
$$(\mathcal{M}, \mathcal{D}) \vdash_\epsilon \phi(\vec{x}; \vec{p}) \iff \mathcal{M} \models \phi(\vec{x}; \vec{p}).$$
That is, for *all* tuples $(a_1, a_2, \ldots, a_n) \in \mathcal{M}$, $\phi(\vec{a}; \vec{p})$ holds.

2. We treat the logical connectives $\wedge$ and $\vee$ classically. For example, for $\vec{x}, \vec{y}, \vec{z}$ distinct sequences of variables,
$$(\mathcal{M}, \mathcal{D}) \vdash_\epsilon \phi(\vec{x}, \vec{z}; \vec{p}) \wedge \psi(\vec{y}, \vec{z}; \vec{p})$$
iff for all $\vec{a} \in M^{|\vec{x}|}, \vec{b} \in M^{|\vec{y}|}, \vec{c} \in M^{|\vec{z}|}$,
$$(\mathcal{M}, \mathcal{D}) \vdash_\epsilon \phi(\vec{a}, \vec{c}; \vec{p}) \wedge \psi(\vec{b}, \vec{c}; \vec{p})$$

3. The universal quantifier is treated classically:
$$(\mathcal{M}, \mathcal{D}) \vdash_\epsilon \forall \vec{x} \phi(\vec{x}, \vec{y}; \vec{p})$$
iff for all $\vec{a} \in M^{|\vec{x}|}$
$$(\mathcal{M}, \mathcal{D}) \vdash_\epsilon \phi(\vec{a}, \vec{y}; \vec{p}).$$

4. The existential quantifier is interpreted probabilistically:
$$(\mathcal{M}, \mathcal{D}) \vdash_\epsilon \exists x \phi(x, \vec{y}; \vec{p}) \iff \Pr_{a \sim \mathcal{D}}[(\mathcal{M}, \mathcal{D}) \vdash_\epsilon \phi(a, \vec{y}; \vec{p})] > \epsilon.$$
Note that the existential quantifier in this definition binds a single variable x; in general it's *not* true that
$$(\mathcal{M}, \mathcal{D}) \vdash_\epsilon \exists \vec{x} \phi(\vec{x}, \vec{y}; \vec{p}) \iff \Pr_{\vec{a} \sim \mathcal{D}^{|\vec{x}|}}[(\mathcal{M}, \mathcal{D}) \vdash_\epsilon \phi(\vec{a}, \vec{y}; \vec{p})] > \epsilon.$$

5. The case of negation is split in subcases as below:
   (a) For ϕ atomic, $(\mathcal{M}, \mathcal{D}) \vdash_\epsilon \neg\phi(\vec{x}; \vec{p}) \iff (\mathcal{M}, \mathcal{D}) \not\vdash_\epsilon \phi(\vec{x}; \vec{p})$.
   (b) $\neg$ distributes classically over $\wedge$ and $\vee$, e.g.
   $$(\mathcal{M}, \mathcal{D}) \vdash_\epsilon \neg(\phi(\vec{x}, \vec{z}; \vec{p}) \wedge \psi(\vec{y}, \vec{z}; \vec{p})) \iff (\mathcal{M}, \mathcal{D}) \vdash_\epsilon \neg\phi(\vec{x}, \vec{z}; \vec{p}) \vee \neg\psi(\vec{y}, \vec{z}; \vec{p}).$$
   (c) $(\mathcal{M}, \mathcal{D}) \vdash_\epsilon \neg\neg\phi(\vec{x}; \vec{p}) \iff (\mathcal{M}, \mathcal{D}) \vdash_\epsilon \phi(\vec{x}; \vec{p})$.
   (d) $(\mathcal{M}, \mathcal{D}) \vdash_\epsilon \neg\exists x \phi(x, \vec{y}; \vec{p}) \iff (\mathcal{M}, \mathcal{D}) \vdash_\epsilon \forall x \neg\phi(x, \vec{y}; \vec{p})$.

(e) $(\mathcal{M}, \mathcal{D}) \vdash_\epsilon \neg\forall x\phi(x, \vec{y}; \vec{p}) \iff (\mathcal{M}, \mathcal{D}) \vdash_\epsilon \exists x \neg\phi(x, \vec{y}; \vec{p})$.

6. The implication symbol $\to$ reduces to boolean combinations classically:

$$(\mathcal{M}, \mathcal{D}) \vdash_\epsilon \phi(\vec{x}, \vec{z}; \vec{p}) \to \psi(\vec{y}, \vec{z}; \vec{p}) \iff (\mathcal{M}, \mathcal{D}) \vdash_\epsilon \neg\phi(\vec{x}, \vec{z}; \vec{p}) \vee \psi(\vec{y}, \vec{z}; \vec{p}).$$

7. The equivalence symbol $\leftrightarrow$ reduces to the conjunction of two implications:

$$(\mathcal{M}, \mathcal{D}) \vdash_\epsilon \phi(\vec{x}, \vec{z}; \vec{p}) \leftrightarrow \psi(\vec{y}, \vec{z}; \vec{p})$$

iff

$$(\mathcal{M}, \mathcal{D}) \vdash_\epsilon [\phi(\vec{x}, \vec{z}; \vec{p}) \to \psi(\vec{y}, \vec{z}; \vec{p})] \wedge [\psi(\vec{x}, \vec{z}; \vec{p}) \to \phi(\vec{y}, \vec{z}; \vec{p})]$$

We can similarly define *ϵF-models* for F-logic by replacing condition 1 of definition (2.1.2) with

- For all formulas $\phi = \phi(x_1, \ldots, x_n)$ and all $a_1, \ldots, a_{n-1} \in \mathcal{M}$, the set

$$\{a_n \in \mathcal{M} : (\mathcal{M}, \mathcal{D}) \vdash_\epsilon \phi(a_1, \ldots, a_n)\}$$

  is $\mathcal{D}$-measurable.

However, note that

$$\{a_n \in \mathcal{M} : (\mathcal{M}, \mathcal{D}) \vdash_\epsilon \phi(a_1, \ldots, a_n)\} = M - \{a_n \in \mathcal{M} : (\mathcal{M}, \mathcal{D}) \models_\epsilon \neg\phi(a_1, \ldots, a_n)\}$$

and thus, $(\mathcal{M}, \mathcal{D})$ *is an ϵE-model iff it's also an ϵF-model*. Henceforward we will uniformly adopt the term **ϵ-model** for this use case.

Similarly, it should be clear from definitions (2.1.3) and (2.1.4) that $(\mathcal{M}, \mathcal{D}) \equiv_\epsilon (\mathcal{N}, \mathcal{E})$ iff

$$(\mathcal{M}, \mathcal{D}) \vdash_\epsilon \phi \iff (\mathcal{N}, \mathcal{E}) \vdash_\epsilon \phi,$$

and $\phi \equiv_\epsilon \psi$ iff for every ϵ-model $(\mathcal{M}, \mathcal{D})$,

$$(\mathcal{M}, \mathcal{D}) \vdash_\epsilon \phi \iff (\mathcal{M}, \mathcal{D}) \vdash_\epsilon \psi.$$

ϵF-validity and **ϵF-satisfiability** (along with their synonyms) are defined similar to definitions (2.1.5) and (2.1.6). Due to duality, we have that

Proposition 2.2.2. *ϕ is ϵF-valid iff $\neg\phi$ is ϵE-satisfiable. In general, ϕ is ϵX-$\bigcirc$ iff $\neg\phi$ is not ϵY-$\bigcirc'$, where (X, Y) is a permutation of {E, F}, and $(\bigcirc, \bigcirc')$ is a permutation of {valid, satisfiable}.*

Table (4) states the dual version of table (1).

	$\epsilon \in (0,1) \cap \mathbb{Q}$	$\epsilon = 0$
ϵF-validity	Π^1_1-complete $\langle \omega, 6, 2 \rangle$ [10, thm 7.6]	decidable $\langle\rangle$ [10, thm 6.7]
ϵF-satisfiability	Σ^1_1-hard $\langle \omega, 3, 2 \rangle$ [12, thm 4.2]	Π^0_1-complete $\langle * \rangle$ [23, prop 3.2]

Table 4: Summary of current knowledge on general ϵF-satisfiability and ϵF-validity

2.3 Finite and Countable Concepts

Definition 2.3.1. An ϵ-model $(\mathcal{M}, \mathcal{D})$ is **finite** iff $|\mathcal{M}|$ is finite. Similarly, an ϵ-model $(\mathcal{M}, \mathcal{D})$ is **countable** iff $|\mathcal{M}| \leq \aleph_0$.

Kuyper and Terwijin ahve shown well-formed model-theoretic properties exist for E-logic (and by duality, F-logic). For example, a downward Lowenheim-Skolem theorem always allows one to work on a model of continuum size [13, Thm. 4.6], and a variant of ultrapower construction works for a weakened definition of ϵ-models [13, Sec. 8]. But they are futile in the finite setting for the same reason that their classical counterparts do not work in classical finite model theory.

Despite this difficulty, finite and countable ϵ-models are still very conducive to analysis because from them we automatically get (finite and countable) probability models.

Lemma 2.3.2. *Let $(\mathcal{M}, \mathcal{D})$ be an ϵ-model. If $\mathcal{D}'$ extends $\mathcal{D}$, then $(\mathcal{M}, \mathcal{D}') \equiv_\epsilon (\mathcal{M}, \mathcal{D})$.*

Proof. By induction on formula complexity, we show $(\mathcal{M}, \mathcal{D}') \models_\epsilon \phi \iff (\mathcal{M}, \mathcal{D}) \models_\epsilon \phi$. All cases other than $\forall$ are trivial, as they don't involve measures.

For $\phi(\vec{x}; \vec{p}) = \forall y \psi(y, \vec{x}; \vec{p})$, we have

$$\begin{aligned} & (\mathcal{M}, \mathcal{D}) \models_\epsilon \phi(\vec{x}; \vec{p}) \\ \iff & \mathcal{D}(a \in \mathcal{M} : (\mathcal{M}, \mathcal{D}) \models_\epsilon \psi(a, \vec{x}; \vec{p})) \geq 1 - \epsilon \\ \iff & \mathcal{D}'(a \in \mathcal{M} : (\mathcal{M}, \mathcal{D}) \models_\epsilon \psi(a, \vec{x}; \vec{p})) \geq 1 - \epsilon \\ \iff & (\mathcal{M}, \mathcal{D}') \models_\epsilon \phi(\vec{x}; \vec{p}) \end{aligned}$$

where the middle equivalence derives from the fact that $\mathcal{D}'$ agrees with $\mathcal{D}$ on $\operatorname{dom} D$. □

Lemma 2.3.3 (Tarski [3])**.** *Every finitely additive measure $\mathcal{D}$ on a set X can be extended to a finitely additive measure $\mathcal{D}'$ so that* $\operatorname{dom} \mathcal{D}' = \mathfrak{P}(X)$.

The above two lemmas allow us to "complete" ϵ-models in the following sense:

Proposition 2.3.4. *Let $(\mathcal{M}, \mathcal{D})$ be any finite or countable ϵ-model. $\mathcal{D}$ can be extended to a measure $\mathcal{D}'$ with* $\operatorname{dom} \mathcal{D}' = \mathfrak{P}(M)$. *Therefore, $(\mathcal{M}, \mathcal{D}')$ is a probability model and by (2.3.2),*

$$(\mathcal{M}, \mathcal{D}) \equiv_\epsilon (\mathcal{M}, \mathcal{D}').$$

Proof. The case of finite ϵ-models $(\mathcal{M}, \mathcal{D})$ follows directly from the lemma since $\mathcal{D}$ is countably additive iff $\mathcal{D}$ is finitely additive.

For the case of countable models $(\mathcal{M}, \mathcal{D})$, notice that $\mathcal{D}$ cannot be atomless, or else M would have to be uncountable. Suppose $a_0 \subseteq M$ is an atom. If $M - a_0$ is not null, then the restriction of $\mathcal{D}$ on $M - a_0$ by the same reasoning must not be atomless, and so there is an atom $a_1 \subseteq M - a_0$. By induction, M can be expressed as the disjoint union of at most countable number of atoms and a null set. Hence it suffices to show that (by assuming M is an atom itself) $\mathcal{D}$ extends to $\mathfrak{P}(M)$ when $\mathcal{D}$ is a 0-1 measure.

Suppose not. Then every measure defined on all of $\mathfrak{P}(M)$ is inconsistent with $\mathcal{D}$. In particular, the measure I_x concentrating measure 1 on an element $x \in M$ cannot be a extension of $\mathcal{D}$, and that can happen only if there is $T_x \in \mathcal{D}$ with measure 0 but contains x. Thus there is a countable set $\{T_x : x \in M\}$ satisfying $\mathcal{D}(T_x) = 0$ and $T_x \ni x$. But by countable subadditivity $0 = \sum_x \mathcal{D}(T_x) \geq \mathcal{D}(M) = 1$, which is a contradiction. □

Because every finite or countable ϵ-model $(\mathcal{M}, \mathcal{D})$ can be taken to have $\mathcal{D}$ everywhere defined on $\mathfrak{P}(M)$, we treat $\mathcal{D}$ as a point function on M. It then makes sense to speak of $\mathcal{D}(a)$ for $a \in M$ and in particular, elements of measure zero, or *null elements*.

Finally, we define the main objects of study in this paper.

Definition 2.3.5. For X = F or E:

A sentence ϕ is said to be **finitely ϵX-valid** and is called a *finite ϵX-validity* (resp. **countably ϵX-valid** and *countable ϵX-validity*) if ϕ is ϵX-satisfied by all finite (resp. countable) ϵ-models.

Likewise, a sentence ϕ is said to be **finitely ϵX-satisfiable** and is called a *finite ϵX-satisfiable* (resp. **countably ϵX-satisfiable** and *countable ϵX-satisfiable*) if ϕ is ϵX-satisfied by some finite (resp. countable) ϵ-models.

In addition, to distinguish between these concepts and the similar concepts in FOL, we say that a formula ϕ is *finitely classically valid* if ϕ is satisfied by all finite first order models; a formula ϕ is *finitely classically satisfiable* if ϕ is satisfied by some finite first order model.

Note that by the Lowenheim-Skolem theorem, what would be the concept of "countably classically valid" coincides with unrestricted validity in first order logic. This is not true for ϵE- or ϵF-logic unconditionally. Terwijn and Kuyper provide a counterexample in [13, exmp 4.5].

2.4 Examples and Applications

Here the goal is two-fold: 1) we clarify concepts developed in the previous sections through examples, and 2) we also note possible applications of ϵE- and ϵF-logic, in part to motivate the main theorems. For the second point, we feel it is illuminating to mention results in later sections. Readers are encouraged to check that these anachronism are correctly applied after perusing their respective expositions.

We first exhibit some examples that highlight the difference in semantics between classical first order logic and our ϵE- and ϵF-logics.

Example 2.4.1. Let $\phi := \exists x \forall y[x = y]$ where $=$ is true equality. Classically, a model $\mathcal{M} \models \phi$ iff M is a singleton. This also holds in ϵF-logic for all $\epsilon < 1$. In ϵE-logic, $(\mathcal{M}, \mathcal{D}) \models_\epsilon \phi$ iff there is a singleton subset $\{a\} \subseteq M$ such that $\mathcal{D}(\{a\}) \geq 1 - \epsilon$.

Example 2.4.2. Let $\phi := \forall x \exists y[x = y]$ where $=$ is true equality. Classically, ϕ holds in every nonempty model. This is true also for ϵE-logic. But $(\mathcal{M}, \mathcal{D}) \vdash_\epsilon \phi$ iff every element of M has $\mathcal{D}$-measure greater than ϵ. In particular, when $\epsilon = \frac{1}{n}$, M must have less than n elements; when $\epsilon = 0$, M is at most countable.

Example 2.4.3. Let $\phi := \exists x \forall y[x \neq y]$ where $=$ is true equality. ϕ is a contradiction in classical first order logic and in ϵF-logic, but $(\mathcal{M}, \mathcal{D}) \models_\epsilon \phi$ iff there is a singleton subset $\{a\} \subseteq M$ with $\mathcal{D}$-measure less than ϵ.

Example 2.4.4. Let $\phi := \forall x \exists y[x \neq y]$ where $=$ is true equality. Classically and in ϵE-logic, $\mathcal{M} \models \phi$ iff $|M| \geq 2$. In ϵF-logic, $(\mathcal{M}, \mathcal{D}) \vdash_\epsilon \phi$ iff every singleton subset of M has $\mathcal{D}$-measure at most $1 - \epsilon$.

Example 2.4.5. Let $\psi(x)$ be a formula with a single free variable. In first order logic,

$$\psi(x),\ \forall y[x = y \to \psi(y)],\ \exists y[x = y \wedge \psi(y)]$$

are equivalent formulas.

In ϵE-logic, for any parameter $a \in M$,

$$(\mathcal{M}, \mathcal{D}) \models_\epsilon \psi(a) \iff (\mathcal{M}, \mathcal{D}) \models_\epsilon \exists y[y = a \wedge \psi(y)]$$

but $(\mathcal{M}, \mathcal{D}) \models_\epsilon \forall y[y = a \to \psi(y)]$ whenever $\{b : b \neq a\} \subseteq M$ has inner $\mathcal{D}$-measure at least $1 - \epsilon$.

Likewise, in ϵF-logic, for any parameter $a \in M$,

$$(\mathcal{M}, \mathcal{D}) \vdash_\epsilon \psi(a) \iff (\mathcal{M}, \mathcal{D}) \vdash_\epsilon \forall y[y = a \to \psi(y)]$$

but $(\mathcal{M}, \mathcal{D}) \not\vdash_\epsilon \exists y[y = a \wedge \psi(y)]$ whenever $\{a\} \subseteq M$ has $\mathcal{D}$-measure $\leq \epsilon$.

In first order logic, a common way to assess whether a theorem ψ follows from a set of axioms T is to pass the sentence $\tau := \neg(T \to \psi)$ to some resolution algorithm R. T implies ψ iff R resolves τ to a contradiction.

But the obvious analogue for ϵE-logic cannot work: $(\mathcal{M}, \mathcal{D}) \models_\epsilon \phi \to \psi$ is not equivalent to

$$(\mathcal{M}, \mathcal{D}) \models_\epsilon \phi \implies (\mathcal{M}, \mathcal{D}) \models_\epsilon \psi$$

because of the paraconsistency of ϵE-logic. However, from definition (2.1.1), $(\mathcal{M}, \mathcal{D}) \models_\epsilon \phi \to \psi$ iff

$$(\mathcal{M}, \mathcal{D}) \models_\epsilon \neg\phi \vee \psi$$

which is equivalent to

$$(\mathcal{M}, \mathcal{D}) \not\models_\epsilon \neg\phi \implies (\mathcal{M}, \mathcal{D}) \models_\epsilon \psi.$$

Rephrasing using ϵF-logic then,

Proposition 2.4.6 (deduction theorem)**.** $(\mathcal{M}, \mathcal{D}) \models_\epsilon \phi \to \psi$ *iff*

$$(\mathcal{M}, \mathcal{D}) \vdash_\epsilon \phi \implies (\mathcal{M}, \mathcal{D}) \models_\epsilon \psi.$$

By duality, $(\mathcal{M}, \mathcal{D}) \vdash_\epsilon \phi \to \psi$ *iff*

$$(\mathcal{M}, \mathcal{D}) \models_\epsilon \phi \implies (\mathcal{M}, \mathcal{D}) \vdash_\epsilon \psi.$$

Thus, axioms T, interpreted inside ϵF-logic, (metalogically) imply theorem ψ, interpreted inside ϵE-logic, iff

$$T \to \psi$$

is an ϵE-validity.

By results (3.4.2) and (3.5.6) of Kuyper and Terwijn that we record in later sections, we can even interpret each quantifier in T (resp. in ψ) in αF-logics (resp. αE-logics) for different αs, in this sense:

> Let T be in prenex normal form $\triangledown_1 x_1 \cdots \triangledown_n x_n \phi(\vec{x})$. For every $\triangledown_i = \exists$, we translate $\triangledown_i x_i$ as "there exists a set A_i with measure $> \alpha_i$ such that for each $x_i \in A_i$, ...;" The quantifier $\forall$ is interpreted as usual. Each α_i can be any arbitrary rational number in $[0, 1]$, independent of what other α_js are.
> Similarly, let ψ be in prenex normal form $\triangledown_1 x_1 \cdots \triangledown_m x_m \phi'(\vec{x})$. For every $\triangledown_i = \forall$, we translate $\triangledown_i x_i$ as "there exists a set B_i with measure $\geq 1 - \beta_i$ such that for each $x_i \in B_i$, ..." The quantifier $\exists$ is interpreted as usual. Each β_i can be any arbitrary rational number in $[0, 1]$, independent of what other β_js are. (∇)
> The implication $T \implies \psi$ (resp. $\psi \implies T$) under these translations can be expressed as some sentence in ϵE-logic (resp. ϵF-logic).

As a corollary, if T is a conjunction of sentences $\{\Lambda_j\}_{j=1}^l$ and ψ is a disjunction of sentences $\{\Gamma_i\}_{i=1}^k$, then for any finite $\{\alpha_j\}_{j=1}^l, \{\beta_i\}_{i=1}^k$ of rational numbers in $[0, 1]$, there is some sentence Φ such that the following are equivalent:

- for all finite $(\mathcal{M}, \mathcal{D})$, if $(\mathcal{M}, \mathcal{D})$ simultaneously α_jF-satisfies each Λ_j, then

$$(\mathcal{M}, \mathcal{D}) \models_{\beta_i} \Gamma_i \qquad \text{for some } i \in [\![k]\!].$$

- Φ is finitely ϵE-valid.

Though such a form of deduction theorem may not seem useful at first, it nevertheless allows the expression of statements regarding many types of mathematical objects, including concept classes in CLT, graphs with weighted vertices, graphs with weighted edges, and artificial neural networks.

In what follows, we give examples of such expressions and their meanings in both ϵE-logic and ϵF-logic. For ease of reading, we will write $\underset{\geq 1-\epsilon}{\forall}$ for $\forall$ in the context of ϵE-logic and $\underset{>\epsilon}{\exists}$ for $\exists$ in the context of ϵF-logic. As noted, in a single sentence, the ϵ parameter may vary, so it is meaningful to write $\underset{>0}{\exists} x \underset{>1/2}{\exists} y \underset{>\epsilon}{\exists} z$. Each boolean symbol whose meaning has not been defined is a shorthand for the usual composition of symbols $\neg, \wedge, \vee$.

These examples will hopefully give a rough picture of the implications of our decidability and undecidability results.

Example 2.4.7 (PAC learning). Define the following:

- an *example space* is a probability space $(X, \mathcal{D})$ with probability distribution $\mathcal{D}$;
- a *concept class* $\mathcal{C}$ is a collection $\{U \subseteq X\}$ of subsets of X, identified with their indicator functions;
- for $U \in \mathcal{C}$, an example oracle $\text{EX}(X, \mathcal{D}, U)$ is a device that randomly returns a pair $(x \in X, U(x))$ for every invocation. The pair is sampled according to $\mathcal{D}$.

In the basic PAC learning model [25], we are given X, $\mathcal{C}$, and an example oracle EX that emits elements of X according to an unknown distribution $\mathcal{D}$ and unknown concept $U \in \mathcal{C}$. We wish to efficiently find a concept close enough to U, in the following sense:

> We have a probabilistic algorithm that, for all error parameters ϵ and δ, for all distributions $\mathcal{D}$ on X, in time polynomial in $\frac{1}{\epsilon}$ and $\frac{1}{\delta}$, returns $U' \in \mathcal{C}$ with
>
> $$\Pr_{x \sim \mathcal{D}}[U'(x) \neq U(x)] < \epsilon$$
>
> more than $1 - \delta$ of the time. Such a U' is called an *approximation concept.*

Typically, X is taken to be $\mathfrak{B}_s = \{0, 1\}^s$ for some s, and $\mathcal{C}$ is a subclass of the boolean functions on $\mathfrak{B}_n$.

It is important to note that this learning model assumes

$$\text{for the given EX, some approximation concept } U' \text{ exists in } \mathcal{C}. \tag{$\wp$}$$

In particular, this happens if EX labels elements according to some concept in $\mathcal{C}$. But in practice this may not always be the case [3]. In ϵE-logic with a language having monadic predicate $P_1, \ldots, P_s$ and $\mathfrak{c}$, we can express this assumption.

In what follows, for a general vector w, w_i will represent the ith value of w; $\mathcal{M}$ is a first order model over universe $\mathfrak{B}_s$, and

$$P_i^{\mathcal{M}}(v) = v_i, \qquad \mathfrak{c}^{\mathcal{M}}(v) = \text{the label generated by EX;}$$

$\mathcal{D}$ is a probability measure everywhere defined over M.

1. (Point class). The point concept class over $\mathfrak{B}_s$ is the collection

$$\{\{v\} : v \in \mathfrak{B}_s\}.$$

[3] hence the research into *agnostic learning*; see [5] for an overview

In other words, each concept labels exactly one point of $\mathfrak{B}_s$ as 1 and the rest as 0. Assumption ($\wp$) holds for the point class iff

$$(\mathcal{M},\mathcal{D}) \models_\epsilon \exists x \underset{\geq 1-\epsilon}{\forall} y[\mathfrak{c}(y) \leftrightarrow \bigwedge_{i=1}^{s}(P_i(x) \leftrightarrow P_i(y))].$$

(Of course it is much more convenient to use equality, but we refrain in order to take advantage of the decidability of monadic relational languages)

2. (Parity class). For each $v \in \mathfrak{B}_s$, let

$$O_v(x) := v \cdot w \mod 2$$

where $(-)\cdot(-) \mod 2$ is the dot product in the vector space $(\mathbb{Z}_2)^s$.

The parity concept class over $\mathfrak{B}_s$ is the collection

$$\{O_v : v \in \mathfrak{B}_s\}.$$

Then assumption ($\wp$) holds for the parity class iff

$$(\mathcal{M},\mathcal{D}) \models_\epsilon \exists x \underset{\geq 1-\epsilon}{\forall} y[\mathfrak{c}(y) \leftrightarrow \bigoplus_{i=1}^{s}(P_i(x) \wedge P_i(y))].$$

3. (Conjunction class). The conjunction concept C_w represented by a vector $w \in \{-1,0,1\}^s$ labels $v \in \mathfrak{B}_s$ as 1 iff for each $i \in [\![s]\!]$ such that $w_i \neq 0$, $v_i = (w_i+1)/2$. The conjunction concept class is the collection of C_w over all $w \in \{-1,0,1\}^s$.

Assumption ($\wp$) holds for the conjunction class iff

$$(\mathcal{M},\mathcal{D}) \models_\epsilon \exists x \exists y \underset{\geq 1-\epsilon}{\forall} z \left(\mathfrak{c}(z) \leftrightarrow \bigwedge_{i=1}^{s}[(P_i(x) \wedge P_i(y) \to P_i(z)) \wedge (\neg P_i(x) \wedge \neg P_i(z) \to \neg P_i(z))]\right)$$

Here we code each conjunction concept C_w using 2 bitstrings x and y: if $x_i = y_i = 1$, $w_i = 1$; if $x_i = y_i = 0$, $w_i = -1$; otherwise $w_i = 0$.

4. (1-Decision lists). Let Z be a triple (α, β, b) where $\alpha, \beta \in \mathfrak{B}_s$ and $b \in \{0,1\}$. A 1-decision list DL_Z represented by Z is the decision procedure that, on input $v \in \mathfrak{B}_s$, runs as follows

if $v_1 = \alpha_1$ **then** output β_1
else if $v_2 = \alpha_2$ **then** output β_2
⋮
else if $v_s = \alpha_s$ **then** output β_s
else output b
end if.

Let

$$\begin{aligned}\psi_i(x,y,w) := {} & [P_1(x) \oplus P_1(w)] \wedge [P_2(x) \oplus P_2(w)] \wedge \cdots \\ & \wedge [P_{i-1}(x) \oplus P_{i-1}(w)] \wedge [P_i(x) \leftrightarrow P_i(w)] \\ & \to [\mathfrak{c}(w) \leftrightarrow P_i(y)].\end{aligned}$$

ψ_i represents a computation of $\mathrm{DL}_{(x,y,b)}$ (for any b) on input w that proceeds to the ith **if** statement before returning.

Let

$$\phi(x,y,z,w) := [P_1(x) \oplus P_1(w)] \wedge [P_2(x) \oplus P_2(w)] \wedge \cdots \\ \wedge [P_{s-1}(x) \oplus P_{s-1}(w)] \wedge [P_s(x) \oplus P_s(w)] \\ \to [\mathfrak{c}(w) \leftrightarrow P_1(z)]$$

ϕ represents a computation of $\mathrm{DL}_{(x,y,z_1)}$ on input w that proceeds to the **else** statement.

Then assumption ($\wp$) holds for the 1-decision lists iff

$$(\mathcal{M}, \mathcal{D}) \models_\epsilon \exists x \exists y \exists z \underset{\geq 1-\epsilon}{\forall} w \left(\phi(x,y,z,w) \wedge \bigwedge_{i=1}^{s} \psi_i(x,y,w) \right).$$

VC dimension is an important quantity of concept classes studied in CLT. We will not define it here (consult [6]) but wish to mention to the computational learning theorists that, as these examples illustrate, when a concept class $\mathcal{C}$ over $\mathfrak{B}_s$ has VC dimension $f(s)$, assumption ($\wp$) can be expressed over $(\mathcal{M}, \mathcal{D})$ by a formula with $O(f(s))$ number of quantifiers (exercise!).

On the other hand, in ϵF-logic, we can express certain interesting conditions on the probability space $(M, \mathcal{D})$.

1. "more than half the time, the label is 1":

$$(\mathcal{M}, \mathcal{D}) \vdash_{\frac{1}{2}} \underset{>1/2}{\exists} x \mathfrak{c}(x).$$

2. "the probability that the ith bit is 1 is strictly between ϵ and $1-\epsilon$":

$$\textsc{uniform}^i_{|\frac{1}{2}-\epsilon|} := (\mathcal{M}, \mathcal{D}) \vdash_\epsilon \forall x \underset{>\epsilon}{\exists} y [P_i(x) \oplus P_i(y)].$$

   As a straightforward generalization, the quantifier-free part of the expression can be swapped out for a more complicated boolean combination.

3. "most bits are irrelevant to the concept — the label only depends on 2 fixed bits":

$$\textsc{atteff}_2 := (\mathcal{M}, \mathcal{D}) \vdash_\epsilon \bigvee_{i<j} \forall x \forall y [(P_i(x) \leftrightarrow P_i(y)) \wedge (P_j(x) \leftrightarrow P_j(y)) \to (\mathfrak{c}(x) \leftrightarrow \mathfrak{c}(y))].$$

   This is the setting for *attribute efficient learning* [9].

As noted before, each ϵ in these examples can vary over the rationals at will. Imagine we want to find if $\textsc{uniform}^i_{\delta_1}$ and $\textsc{uniform}^j_{\delta_2}$ along with $\textsc{atteff}_2$ would imply ($\wp$) for some concept class $\mathcal{C}$. We may do so by querying for the ϵE-validity of a two-part sentence

$$T \to \phi$$

derived from applying note (∇).

By the decidability of monadic relational languages (3.2.8 and 3.2.11) established later, these kinds of questions are all decidable, as long as all statements use only unary predicates.

It is also possible to add a BIT relation to ϵE- and ϵF-logics along the lines of the corresponding relation in classical finite model theory [15]. One can then express concept classes over all size parameter s with one single sentence. However, the complexity of deduction then becomes unknown.

Example 2.4.8 (graphs with weighted vertices). Let $\mathcal{L}$ be a language with a single binary relation $E(\cdot,\cdot)$, standing for the edge relation between two vertices. As in first order logic, any ϵ-model of this language is automatically a directed graph with at most one edge between every pair of vertices (including loops). Moreover, in such ϵ-models, to each vertex of the graph structure is assigned a weight in $[0,1]$ such that the total sum of all vertex weights is 1; however, the edges are not weighted. Such graphs with weighted vertices can be used to model, among many things, cities with populations and in general the PageRank algorithm.

Because in ϵF-logic, $\forall$ is interpreted classically, we can express quite a few properties of graphs:

1. loopless:
$$\forall x \neg E(x,x).$$

2. undirected:
$$\forall x \forall y [E(x,y) \leftrightarrow E(y,x)].$$

3. complete:
$$\forall x \forall y [E(x,y) \wedge E(y,x)].$$

4. bipartite with a fixed partition (if $\mathcal{L}$ has a unary predicate A):
$$\forall x \forall y [(A(x) \leftrightarrow A(y)) \to \neg E(x,y) \wedge \neg E(y,x)].$$
Similarly, k-coloring can be expressed as well.

5. in a simple graph, "every 1-neighborhood collectively has weight more than ϵ" (if $\mathcal{L}$ has equality):
$$\forall x \underset{>\epsilon}{\exists} y [y = x \vee E(x,y)].$$

6. "every directed triangle has collective measure more than ϵ" (if $\mathcal{L}$ has equality):
$$\forall x \forall y \forall z [E(x,y) \wedge E(y,z) \wedge E(z,x) \to \underset{>\epsilon}{\exists} w (w = x \vee w = y \vee w = z)].$$

7. "every element has a unique successor of positive measure" (when $\mathcal{L}$ has equality):
$$\forall x \underset{>0}{\exists} y [E(x,y) \wedge \forall z (E(x,z) \to z = y)].$$

8. "there exists a set of 'initial' vertices A collectively with weight more than ϵ such that every $v \in A$ is connected to every vertex in the entire graph":
$$\underset{>\epsilon}{\exists} x \forall y E(x,y).$$

On the other hand, in ϵE-logic, we can express the likes of the following.

1. "there is a clique of size k":
$$\exists \vec{x} \left(\bigwedge_{1 \le i < j \le k} E(x_i, x_j) \wedge E(x_j, x_i) \right).$$
In general, for any fixed graph G, we can express the existence of a subgraph isomorphic to G.

2. "there is a subgraph isomorphic to G that carries weight $\geq 1-\epsilon$":

$$\exists \vec{x}\left[\mathrm{ISO}_G(\vec{x}) \wedge \underset{\geq 1-\epsilon}{\forall} y \bigvee_{i=1}^{k} y = x_i\right],$$

where ISO_G is a formula expressing $\vec{x}$ is isomorphic to G.

3. in a simple graph, "there is a single vertex v that is connected to at least $1-\epsilon$ (by weight) of other vertices":

$$\exists v \underset{\geq 1-\epsilon}{\forall} x E(x, v).$$

In ϵE-logic, we can also make weakened version of universal statements from the ϵF-examples by replacing $\forall$ with $\underset{\geq 1-0}{\forall} = \underset{\geq 1}{\forall}$. The transformed sentences will then quantify over all elements of positive measure, rather than all elements. For instance, the sentences given for the loopless, undirected, complete, and bipartite properties all carry over to apply to the subgraph consisting of all nonnull vertices.

This weakened quantifier suffices in most cases. In fact, later on, our proof of the undecidability of finite ϵE-satisfiability (3.4.4) depends heavily on this competency of $\underset{\geq 1}{\forall}$.

In some cases, when $\epsilon > 0$, the quantifier $\underset{>\epsilon}{\exists}$ can also be replaced (per note (∇)) with $\underset{\geq \epsilon}{\forall}$ without affecting the intended semantics very much. For instance, the property "every 1-neighborhood collectively has weight more than ϵ" differs very little from "every 1-neighborhood collectively has weight at least ϵ" in most imaginable applications.

Example 2.4.9 (graphs with weighted edges). Instead of assigning a measure to vertices, often we want to assign numbers to edges of a graph, for example in a MAX-FLOW or a path-finding problem.

Let $\mathcal{L}$ be a language with binary relations $I(\cdot,\cdot)$, $C(\cdot,\cdot)$, and $D(\cdot,\cdot)$. Here $I(x, y)$ represents that the codomain of edge x equals the domain of edge y; $D(x, y)$ (resp. $C(x, y)$) represent that edges x and y have the same domain (resp. codomain). Thus, any graph with weighted edges $\{e_i\}_{i\in M}$ such that the total weight equals 1 is automatically a probability model in $\mathcal{L}$.

Conversely, suppose $(\mathcal{M}, \mathcal{D})$ has an everywhere defined measure $\mathcal{D}$ and classically satisfies the axioms of

- "C and D are equivalence relations":

$$\begin{gathered} \forall x C(x,x) \\ \forall x \forall y C(x,y) \leftrightarrow C(y,x) \\ \forall x \forall y \forall z C(x,y) \wedge C(y,z) \rightarrow C(z,x) \end{gathered} \tag{EQR}$$

  along with the analogues for D.

- "incidence relation respects domain and codomain":

$$\begin{gathered} \forall x \forall y (C(x,y) \rightarrow \forall z[I(x,z) \leftrightarrow I(y,z)]) \\ \forall x \forall y (D(x,y) \rightarrow \forall z[I(z,x) \leftrightarrow I(z,y)]). \end{gathered} \tag{IDC}$$

- "domain and codomain respect incidence relation":

$$\begin{gathered} \forall x \forall y (I(x,y) \rightarrow \forall z[C(z,x) \leftrightarrow I(z,y)]) \\ \forall x \forall y (I(x,y) \rightarrow \forall z'[D(z',y) \leftrightarrow I(x,z')]) \end{gathered} \tag{DCI}$$

- "domain and codomain are unique":

$$\begin{aligned}&\forall x \forall y \forall z[I(x,y) \wedge I(x,z) \to D(y,z)]\\&\forall x \forall y \forall z[I(y,x) \wedge I(z,x) \to C(y,z)]\end{aligned} \tag{!DC}$$

Let V_D and V_C be respectively the equivalence classes modulo $D^{\mathcal{M}}$ and $C^{\mathcal{M}}$. For every element a, denote its equivalence class in V_D by $[\![a]\!]_D$ and that in V_C by $[\![a]\!]_C$. By axiom (DCI), $[\![a]\!]_D \in V_D$ can be identified with $[\![b]\!]_C \in V_C$, $[\![a]\!]_D \sim [\![b]\!]_C$, if $I^{\mathcal{M}}(b,a)$. Thus we can form the vertex set $V = (V_D \cup V_C)/\sim$. Axiom (IDC) implies that the relation $I^{\mathcal{M}}$ induces a relation $I_C^{\mathcal{M}}$ on $V_C \times M$. By factoring through the identification $\sim$, the relation $I_C^{\mathcal{M}}$ can be treated as a relation on $V \times M$ with the property that, for every element a of M, $I_C^M([\![a]\!]_D, a)$. But axiom (!DC) says that this $I_C^{\mathcal{M}}$ is in fact a function $M \to V$. We therefore retrieve the domain function $\mathrm{dom}: M \to V, a \mapsto [\![a]\!]_D$. By the same reasoning, we also derive the codomain function $\mathrm{cod}: M \to V, a \mapsto [\![a]\!]_C$. These data then uniquely determine a graph with edges M and vertices V.

As ϵF-logic interprets $\forall$ classically, it can convey the axioms along with many of the usual properties of graphs.

1. loopless:
$$\forall x \neg I(x,x).$$
2. "no more than one edge per pair of vertices" (if $\mathcal{L}$ has equality):
$$\forall x \forall y[D(x,y) \wedge C(x,y) \to x = y].$$
3. bidirectional: "each edge x has a corresponding edge with positive weight that goes in the opposite direction":
$$\forall x \underset{>0}{\exists} y[I(x,z) \wedge I(z,x)].$$
On the other hand, we cannot express *un*directedness without changing the axioms. The reader is encouraged to work out the axioms for a simple graph with weighted edges.
4. complete: "for any two edges x and y, there is an edge with positive weight that connects x to y":
$$\forall x \forall y \underset{>0}{\exists} z[I(x,z) \wedge I(z,y)].$$
5. "every directed triangle has collective weight $> \epsilon$" (if $\mathcal{L}$ has equality):
$$\forall x \forall y \forall z[I(x,y) \wedge I(y,z) \wedge I(z,x) \to \underset{>\epsilon}{\exists} w(w = x \vee w = y \vee w = z)].$$
For any fixed k, we can also make the analogous statement for k-cycles.
6. "every length k path from x to y has weight $> \epsilon$" (if $\mathcal{L}$ has equality):
$$\textsc{pathmin}_\epsilon^k(x,y) := \forall \vec{x} \left(\bigwedge_{i=1}^{k-1} I(x_i, x_{i+1}) \to \underset{>\epsilon}{\exists} y \left(\bigvee_{i=1}^{k} y = x_i \right) \right).$$
For any fixed graph G of size k, the constructions in this and the last items generalize to make statements of the form "any subgraph isomorphic to G has total weight $> \epsilon$."

As with example (2.4.9), in ϵE-logic, for any fixed graph G, we can express the existence of a subgraph isomorphic to G. We can also assert that some such subgraph has weight $\geq 1 - \epsilon$. These properties may be desirable when working with MAX-FLOW problems.

Finally, we can also transform statements in ϵF-logic into weaker statements in ϵE-logic by replacing $\forall$ with $\underset{\geq 1}{\forall}$ and $\underset{>\epsilon}{\exists}$ with $\underset{\geq \epsilon}{\forall}$. With emphasis, we note that all axioms (EQR), (DCI), (IDC), and (!DC) of graphs with weighted edges are universal sentences. Therefore, as long as the presence of zero-weight edges present no difficulty, we can also investigate implications

$$T \implies \psi$$

with ψ interpreted in ϵF-logic.

Example 2.4.10 (artificial neural networks). *Artificial neural network (ANN)* is a very popular biologically inspired technique in machine learning that is often used in pattern recognition [16][18]. Each ANN is a directed graph in which each edge e has weight $\mathfrak{w}(e)$. Its nodes are called *neurons* and its edges are called *connections*. If neuron η connects to neuron ζ via connection e, we say η *feeds into* ζ via e (written $\eta \xrightarrow{e} \zeta$), η is the *presynaptic neuron* of e, and ζ is the *postsynaptic neuron* of e. Each neuron is either *activated* or not. Its state at time $t+1$ depends on the activation states at time t of the neurons that feed into it. The exact update rule may vary in different neural networks, but usually it is implemented as a *linear threshold function*:

Each neuron η has a *threshold value* $\mathfrak{T}$ such that η is activated at time $t+1$ iff

$$\sum_{\zeta \xrightarrow{e} \eta} \mathfrak{w}(e) \cdot \|\text{“}\zeta \text{ activated at time } t\text{”}\| > \mathfrak{T}.$$

Like in the previous example, ANNs can be represented by finite ϵ-models $(\mathcal{M}, \mathcal{D})$ with $\operatorname{dom} D = \mathfrak{P}(M)$ of the language $\mathcal{L}$ with binary relations I, D, and C. The measure $\mathcal{D}(a)$ of each element a of M correspond to the weight $\mathfrak{w}(a)$ in the ANN. If the threshold $\mathfrak{T}$ is fixed across all neurons, then the linear threshold update rule can be expressed in ϵF-logic.

We introduce new predicates $\textsc{actv}_t(x)$ that represents whether the presynaptic neuron of edge x is activated at time t. It satisfies the following relations for each t.

1. “Suppose x and y have the same presynaptic neuron. Then $\textsc{actv}_t(x)$ holds iff $\textsc{actv}_t(y)$ holds”:

$$\forall x \forall y (D(x, y) \to [\textsc{actv}_t(x) \leftrightarrow \textsc{actv}_t(y)]).$$

2. the linear threshold update rule:

$$\forall x (\textsc{actv}_{t+1}(x) \leftrightarrow \underset{>\mathfrak{T}}{\exists} y [I(y, x) \wedge \textsc{actv}_t(y)]).$$

In a typical usage of ANN, there are two sets of distinguished neurons $\mathfrak{I}$ and $\mathfrak{O}$ called *input neurons* and *output neurons*. At the beginning, each neuron of $\mathfrak{I}$ is activated or deactivated according to an input bitstring, for example derived from a digital image. All other neurons are not activated. After some time t, the activation states of the neurons of $\mathfrak{O}$ are returned as a bitstring. Continuing our example, we might desire the output of 1 from every output neuron iff the image is of a butterfly.

Imagine we are interested in whether some property Φ of ANN implies some property Ψ. If we can phrase Φ as a sentence to be interpreted under ϵF-logic and Ψ as a sentence to be interpreted under ϵE-logic, then we can answer this question by querying for the ϵE-validity of

$$\Phi \wedge \Lambda \to \Psi,$$

where Λ is the conjunction of the axioms of the graph from example (2.4.9) and the axioms of $\textsc{actv}_t$ from this example.

Unlike the example of PAC learning, we cannot say with certainty whether any or all of the theories of finite graphs with weighted vertices, finite graphs with weighted edges, or finite artificial neural networks are decidable. The main theorems of this paper will establish that the naive method is out of the picture: there is no general deduction mechanism for ϵE- or ϵF-logics when restricted to finite ϵ-models. In particular, this result holds even when restricted to first order languages with a finite number of binary relations and an infinite number of unary predicates. But that is not enough to determine the exact computability of the above theories, each of which uses only a finite number of unary predicates. (Even for ANN, in almost all use cases, only a finite number of $\textsc{actv}_t$ predicates are considered). Weakening the language requirement remains a major research area in ϵE- and ϵF-logics.

Related to the issue of decidability is expressability. We note in passing that despite these examples, ϵE- and ϵF-logics still have nontrivial limitations in expression power. These limitations derive in many cases from the limitations of first order logic itself. A detailed discussion of impossibility results in expressability is outside the scope of this paper, but we mention that quite a few techniques for first order logic, like locality, carry over to our probability logics. The interested reader is advised to consult [15].

3 Validities and Satisfiabilities

3.1 Finite and Countable 0E-Satisfiabilities

In contrast to first order logic, where Trachtenbrot's theorem implies that an effective calculus for deducing true theorems over finite models cannot exist, we show here that finite and countable 0F-validities are decidable. Moreover, the results of this subsection apply to any first order language. Consequently, finite and countable 0E-satisfiability are also decidable regardless of language.

(Recall that $\vec{x}$ is a shorthand for a sequence of variables $x_1, x_2, \ldots, x_n$ for some $n \geq 0$, and $\forall\vec{x}$ is a shorthand for $\forall x_1 \forall x_2 \cdots \forall x_n$).

Lemma 3.1.1 (validity conversion)**.** *Let $\nabla_i \in \{\forall, \exists\}$ represent quantifiers, and*

$$\phi := \nabla_1 x_1 \nabla_2 x_2 \cdots \nabla_n x_n \psi(\vec{x})$$

where ψ is a quantifier free formula. Suppose $I = \{i_1, i_2, \ldots, i_k\} \subseteq \{1, \ldots, n\}$ is an enumeration of all indices i such that $\nabla_i = \forall$. Define

$$\phi^*(y) := \forall x_{i_1} \forall x_{i_2} \cdots \forall x_{i_k} \psi(\underline{\vec{y}}, x_{i_1}, \underline{\vec{y}}, x_{i_2}, \underline{\vec{y}}, \ldots, \underline{\vec{y}}, x_{i_k}, \underline{\vec{y}}),$$

where each $\underline{\vec{y}}$ denote a block $y, y, \ldots, y$ of y repeated some number of times, depending on the location of $\underline{\vec{y}}$. In other words, in ϕ^*, all x_j with $j \notin I$ has been substituted with the free variable y.

Let

$$\phi' := \forall y \phi^*(y).$$

Then ϕ is finitely 0F-valid iff ϕ' is finitely classically valid. ϕ is countably 0F-valid iff ϕ' is classically valid.

Proof. Let's consider the finite validity claim of the theorem. The countable validity portion is almost exactly the same.

(*ϕ finitely 0F-valid $\implies$ ϕ' finitely classically valid*) Let $\mathcal{V}_k$ be a classical model of size k, with universe $\{1, 2, \ldots, k\}$. Define measures $\mathcal{E}_{i,k}$ on it such that $\mathcal{E}_{i,k}(i) = 1$ and $\mathcal{E}_{i,k}(j) = 0$, $\forall j \neq i$. If $(\mathcal{V}_k, \mathcal{E}_{i,k}) \vdash_0 \phi$, then all the $x_j, j \notin I$ (i.e. all those with an existential quantifier) must be interpreted as i since i has measure 1. Hence $(\mathcal{V}_k, \mathcal{E}_{i,k}) \vdash_0 \phi$ implies $\mathcal{V}_k \models \phi^*(i)$.

Since ϕ is finitely 0F-valid, for any fixed k, $(\mathcal{V}_k, \mathcal{E}_{i,k}) \vdash_0 \phi$ and thus $\mathcal{V}_k \models \phi^*(i)$ hold for all $1 \leq i \leq k$. Therefore,

$$\mathcal{V}_k \models \forall y \phi^*(y) \implies \mathcal{V}_k \models \phi'$$

Now vary k, and we conclude that ϕ' is classically finitely valid.

Note that the above reasoning did not use finiteness in an essential way. In fact, slightly modifying the argument shows that ϕ 0F-valid for all 0-models of size κ implies ϕ' classically valid for all first order models of size κ.

(ϕ' *finitely classically valid* $\implies$ ϕ *finitely 0F-valid*) Let $(\mathcal{M}, \mathcal{D})$ be a 0-model with the universe $M = \{1, \ldots, k\}$. By proposition (2.3.4), we can take $\mathcal{D}$ to be defined on all subsets of M. Since ϕ' is satisfied by all classical models, $\mathcal{M} \models \phi' \implies \mathcal{M} \models \forall y \phi^*(y)$. Because $\mathcal{M}$ is finite (this is the only place where the finiteness is used; substitute countability for the countable case), there must be an element $a \in M$ with positive measure. Then $\mathcal{M} \models \phi^*(a)$, meaning that all the $\exists$ bindings (with the interpretation under $\vdash_\epsilon$ of measure strictly positive) in ϕ are realized by a. Thus $(\mathcal{M}, \mathcal{D}) \vdash_0 \phi$. Since $(\mathcal{M}, \mathcal{D})$ is an arbitrary finite 0-model, ϕ is finitely 0F-valid.

□

Lemma 3.1.2. *Let*

$$\phi(\vec{y}) := \forall \vec{x} \psi(\vec{x}, \vec{y})$$

be a universal formula, where ψ is quantifier free.

The following are equivalent:

1. *ϕ is finitely classically valid.*
2. *ϕ is classically valid.*

Proof. Certainly, (2) $\implies$ (1). It suffices to show (1) $\implies$ (2).

Because for any first order model $\mathcal{M}$, $\mathcal{M} \models \phi(\vec{y}) \iff \mathcal{M} \models \forall \vec{y} \phi(\vec{y})$, we assume that ϕ is a sentence $\forall \vec{x} \psi(\vec{x})$. Let $n = |\vec{x}|$.

Suppose ϕ is satisfied by all finite (classical) models but there is an infinite model $\mathcal{M} \models \neg\phi$. Then there is a tuple $\vec{a} \in M^n$ such that $\mathcal{M} \models \neg\psi(\vec{a})$. We form a finite model $\mathcal{M}'$ containing $\{a_i\}_{i=1}^n$ such that $\mathcal{M}' \models \neg\psi(\vec{a})$, which would yield a contradiction.

Let $F_0 := \{a_i\}_{i=1}^n \cup \{c^{\mathcal{M}} : c \text{ is a constant symbol that appears in } \phi\}$. Given F_i, set

$$F_{i+1} := \{f^{\mathcal{M}}(\xi) : \xi \in F_i, f \text{ is a function symbol that appears in } \phi\}$$

Then we define the universe of our model to be

$$M' = \bigcup_{i=0}^{k} F_k \cup \{r\}$$

where k is maximal number of times any function symbol appears in ψ, and r is an arbitrary new element. F_0 is obviously finite, and given F_i is finite, $|F_{i+1}| \leq |F_i| \cdot |\text{length}(\phi)|$ is finite. Thus each F_i is finite and so M' is finite.

The relations in $\mathcal{M}'$ will be the relations of $\mathcal{M}$ restricted to M'. For each function symbol f in the language, define

$$f^{\mathcal{M}'}(\xi) = \begin{cases} f^{\mathcal{M}}(\xi) & \text{if } \xi \in F_i \text{ for some } i < k \\ r & \text{otherwise} \end{cases}$$

if f appears in ψ, and otherwise arbitrary. Finally, for each constant symbol c, the interpretation is

$$c^{\mathcal{M}'} = \begin{cases} c^{\mathcal{M}} & \text{if } c^{\mathcal{M}} \in H \\ r & \text{otherwise} \end{cases}$$

It's easy to check that if t is a term that appears in ψ, then $t^{\mathcal{M}} = t^{\mathcal{M}'}$, and if R is an n-ary relation that appears in ψ then $R^{\mathcal{M}}(\vec{\xi}) = R^{\mathcal{M}'}(\vec{\xi})$ for any $\vec{\xi} \in M'^n$. Thus by induction $\mathcal{M}'$ as constructed satisfies $\neg\psi(\vec{a})$ as desired. □

But universal classical validities reduce to propositional tautologies: for each prime formula π with n arguments, and each n-tuple $\vec{x}$ of variables in the language, form a propositional variable $p_{\pi,\vec{x}}$. Then a universal formula $\forall\vec{y}\psi(\vec{y},\vec{z})$ with ψ quantifier-free is valid iff ψ^{prop}, the propositional formula where all instances of prime formulas $\pi(\vec{x})$ are replaced by the propositional variable $p_{\pi,\vec{x}}$, is a propositional tautology. Since propositional validity is decidable, this combined with (3.1.2) yields

Theorem 3.1.3. *For any first order language, the set of finitely 0F-valid formulas coincides with the set of countably 0F-valid formulas. They are both decidable. Therefore, the set of finitely 0E-satisfiable sentences coincides with the set of countably 0E-satisfiable sentences, and they are both decidable.*

3.2 Monadic Relational Language

Let $\mathcal{L}$ be a first order language

- with no equality,
- with no function symbols,
- with no relation of arity at least 2, and
- with at most a finite number of unary predicates $P_1, P_2, \ldots, P_s$.

We call $\mathcal{L}$ a *monadic relational language*. In this subsection, we show that for any such language, unrestricted, countable, and finite ϵE-satisfiabilites and ϵF-satisfiabilities are all decidable, ergo the computability of ϵE-validities as well.

The essence of the proofs in this section resides in the fact that each ϵ-model in such a language "has only a finite amount of information": They are partitioned by the monadic predicates into a finite number of indistinguishable parts, and the measures of these parts uniquely determine the models up to ϵ-elementary equivalence. This observation allows us to reduce these satisfiability problems to linear programming.

Lemma 3.2.1. *Let $\mathcal{L}$ be monadic relational with unary predicates $P_1, P_2, \ldots, P_s$. Suppose $(\mathcal{M},\mathcal{D})$ is an ϵ-model in $\mathcal{L}$. Then there is a finite probability model $(\mathcal{N},\mathcal{E})$ such that $(\mathcal{M},\mathcal{D}) \equiv_\epsilon (\mathcal{N},\mathcal{E})$.*

Furthermore, we can take the universe to be some subset $N \subseteq \mathfrak{P}([\![s]\!])$ and require that

for every $a, b \in N$, if $P_l^{\mathcal{N}}(a)$ holds when and only when $P_l^{\mathcal{N}}(b)$ holds, then $a = b$.

Proof. For each $U \subseteq [\![s]\!]$, define the subset

$$M_U := \{a \in M : \forall l \in [\![s]\!],\ P_l^{\mathcal{M}}(a) \iff l \in U\}.$$

$\{M_U\}_{U\subseteq[\![s]\!]}$ partitions $(\mathcal{M},\mathcal{D})$ into at most 2^s disjoint parts. It should be immediate that for any formula $\phi(x)$, any $U \subseteq [\![s]\!]$, and $a, b \in M_U$,

$$(\mathcal{M},\mathcal{D}) \models_\epsilon \phi(a) \iff (\mathcal{M},\mathcal{D}) \models_\epsilon \phi(b). \tag{Δ}$$

Now we define $(\mathcal{N},\mathcal{E})$. Let

$$N := \{U : M_U \neq \emptyset\}$$

and let $\mathcal{E}$ be defined on points U by

$$\mathcal{E}(U) = \mathcal{D}(M_U).$$

Then $\sum_{U\in N} \mathcal{E}(U) = 1$, so $\mathcal{E}$ is a probability measure.

Finally, define the interpretations $P_l^{\mathbb{N}}$ on N by

$$P_l^N(U) :\iff l \in U.$$

Since all subsets of N are $\mathcal{E}$-measurable, $(\mathcal{N},\mathcal{E})$ is a probability model.

For ϵ-elementary equivalence, we show the stronger claim that:

For any quantifier-free formula $\phi(\vec{x}, \vec{y})$ with $j = |\vec{x}|$ and $k = |\vec{y}|$, every $\vec{U} \in N^k$, and every sequence of quantifiers $\nabla_1, \ldots, \nabla_j \in \{\exists, \forall\}$,

$$(\mathcal{N}, \mathcal{E}) \models_\epsilon \nabla_1 x_1 \cdots \nabla_j x_j \phi(\vec{x}, \vec{U})$$

iff for all (and, by (Δ), for any) $\vec{a} \in M_{\vec{U}} := M_{U_1} \times M_{U_2} \times \cdots \times M_{U_k}$,

$$(\mathcal{M}, \mathcal{D}) \models_\epsilon \nabla_1 x_1 \cdots \nabla_j x_j \phi(\vec{x}, \vec{a}).$$

We proceed by induction on the number j of quantifiers. The case of $j = 0$ is immediate by our construction.

Suppose that our claim is proved for $j = j' \geq 0$. The case of $\psi(\vec{U}) := \exists y \nabla_1 x_1 \cdots \nabla_{j'} x_{j'} \phi(\vec{x}, y, \vec{U})$ does not involve measures and is obvious. For $\psi(\vec{U}) := \forall y \nabla_1 x_1 \cdots \nabla_{j'} x_{j'} \phi(\vec{x}, y, \vec{U})$, $(\mathcal{N}, \mathcal{E}) \models_\epsilon \psi(\vec{U})$ iff

$$W := \{V : (\mathcal{N}, \mathcal{E}) \models_\epsilon \nabla_1 x_1 \cdots \nabla_{j'} x_{j'} \phi(\vec{x}, V, \vec{U})\}, \quad \mathcal{E}(W) \geq 1 - \epsilon.$$

By induction hypothesis,

$$W = \{V : \forall a \in M_V,\ \forall \vec{b} \in M_{\vec{U}},\ (\mathcal{M}, \mathcal{D}) \models_\epsilon \nabla_1 x_1 \cdots \nabla_{j'} x_{j'} \phi(\vec{x}, a, \vec{b})\}.$$

Therefore, for all $\vec{b} \in M_{\vec{U}}$,

$$\bigcup_{V \in W} M_V = \{a : (\mathcal{M}, \mathcal{D}) \models_\epsilon \nabla_1 x_1 \cdots \nabla_{j'} x_{j'} \phi(\vec{x}, a, \vec{b})\}$$

and thus

$$\begin{aligned} & \Pr_{a \sim \mathcal{D}}[(\mathcal{M}, \mathcal{D}) \models_\epsilon \nabla_1 x_1 \cdots \nabla_{j'} x_{j'} \phi(\vec{x}, a, \vec{b})] \\ = \ & \Pr_{a \sim \mathcal{D}}[a \in \bigcup_{V \in W} M_V] \\ = \ & \sum_{V \in W} \mathcal{D}(M_V) \\ = \ & \sum_{V \in W} \mathcal{E}(V) \\ = \ & \mathcal{E}(W) \\ \geq \ & 1 - \epsilon. \end{aligned}$$

implying

$$(\mathcal{M}, \mathcal{D}) \models_\epsilon \forall y \nabla_1 x_1 \cdots \nabla_{j'} x_{j'} \phi(\vec{x}, y, \vec{U}).$$

The converse direction follows by reversing this line of reasoning and applying (Δ).

The claim starting from “Furthermore” follows by our construction. □

We introduce the concept of ϵE- and ϵF-trees to help us analyze satisfiability of sentences.

Definition 3.2.2. Let M be a set. A **tree in M with height** n is defined as a tree T with n levels (from 1 to n) with the following properties

1. all nodes are subsets of M.
2. if node V is at level $k < n$, then V has a child $V^{\succ x}$ for each $x \in V$, and these are all of V's children.
3. if node V is at level n, then V has no children; V is called a *leaf node* of T.

The unique root of T is denoted $\operatorname{root} T$. A **bran** of a tree in M with height n is defined as a sequence $\langle (a_i, V_i) \rangle_{i=1}^n$ of pairs, where for each i,

1. $V_i \subseteq M$ is a node of T at level i,
2. $a_i \in V_i$, and
3. $V_{i+1} = V_i^{\succ a_i}$ if $i < n$.

We will also write $\langle a_i \in V_i \rangle_{i=1}^n$ for a bran, which should not cause any confusion.

Definition 3.2.3. Let $Q = \langle \triangledown_1, \triangledown_2, \ldots, \triangledown_n \rangle$ be a sequence of quantifiers from $\{\exists, \forall\}$. Let $(\mathcal{M}, \mathcal{D})$ be an ϵ-model. An ϵ**E-tree in** $(\mathcal{M}, \mathcal{D})$ **with levels** Q is defined as a tree in M with height n such that:

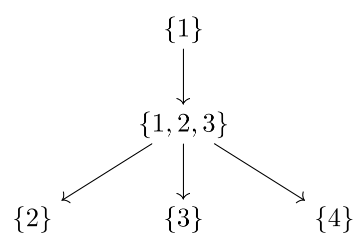
1. if $\triangledown_k = \exists$, then all nodes at level k are nonempty subsets of M; level k is called a $\exists$-level.
2. if $\triangledown_k = \forall$, then all nodes at level k are $\mathcal{D}$-measurable subsets of M with $\mathcal{D}$-measure at least $1 - \epsilon$; level k is called a $\forall$-level.

Let $\Phi := \triangledown_1 x_1 \cdots \triangledown_n x_n \phi(\vec{x})$ where ϕ is quantifier-free. An ϵ**E-tree in** $(\mathcal{M}, \mathcal{D})$ **for** Φ is defined as an ϵE-tree in $(\mathcal{M}, \mathcal{D})$ with levels $\langle \triangledown_1, \ldots, \triangledown_n \rangle$ with the additional property that

(S) for every bran $\langle a_i \in V_i \rangle_{i=1}^n$,
$$\mathcal{M} \models \phi(\vec{a}).$$

When Φ (or Q) and $(\mathcal{M}, \mathcal{D})$ are clear from the context, we will simply use the term ϵ**E-tree**.

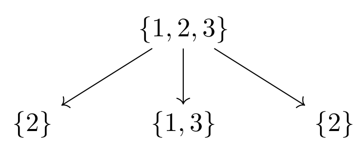
Example 3.2.4. If $Q = \langle \exists, \forall, \exists \rangle$, $M = [\![4]\!]$, and $\mathcal{D}$ is the uniform distribution, then



is an $\frac{1}{2}$E-tree in $(\mathcal{M}, \mathcal{D})$ with levels Q.

If $\Phi = \exists x \forall y \exists z [x + y = z]$ (where $x + y = z$ should be treated as a relation over (x, y, z)), then the above tree is also an $\frac{1}{2}$E-tree in $(\mathcal{M}, \mathcal{D})$ for Φ.

Note that nodes across a level need not be distinct. For example, if $\Phi = \forall x \exists y [x \neq y]$, then



would be a valid $\frac{1}{2}$E-tree in $(\mathcal{M}, \mathcal{D})$ for Φ.

Similarly, we define

Definition 3.2.5. Let $Q = \langle \triangledown_1, \triangledown_2, \ldots, \triangledown_n \rangle$ be a sequence of quantifiers from $\{\exists, \forall\}$. Let $(\mathcal{M}, \mathcal{D})$ be an ϵ-model. An ϵ**F-tree in** $(\mathcal{M}, \mathcal{D})$ **with levels** Q is defined as a tree in M with height n such that:

1. if $\triangledown_k = \exists$, then all nodes at level k are $\mathcal{D}$-measurable subsets of M with $\mathcal{D}$-measure greater than ϵ; level k is called a $\exists$-level.

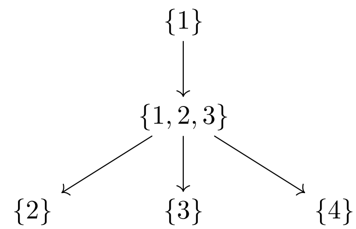
2. if $\triangledown_k = \forall$, then all nodes at level k are M; level k is called a $\forall$-level.

Let $\Phi := \triangledown_1 x_1 \cdots \triangledown_n x_n \phi(\vec{x})$ where ϕ is quantifier-free. An **ϵF-tree in $(\mathcal{M}, \mathcal{D})$ for Φ** is defined as an ϵF-tree in $(\mathcal{M}, \mathcal{D})$ with levels $\langle \triangledown_1, \dots, \triangledown_n \rangle$ with the additional property that

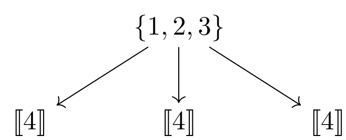
(S) for every bran $\langle a_i \in V_i \rangle_{i=1}^n$,
$$\mathcal{M} \models \phi(\vec{a}).$$

When Φ and $(\mathcal{M}, \mathcal{D})$ are clear from the context, we will simply use the term **ϵF-tree**.

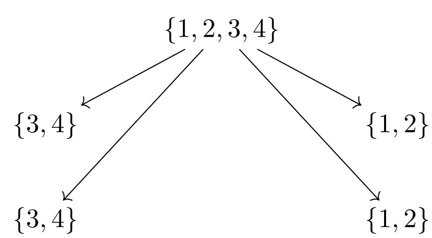
Example 3.2.6. With the same setting as above of $Q = \langle \exists, \forall, \exists \rangle$, $M = [\![4]\!]$, and $\mathcal{D}$ the uniform distribution,



is *not* a $\frac{1}{2}$F-tree in $(\mathcal{M}, \mathcal{D})$ with levels Q because the root $\{1\}$ has measure $\frac{1}{4} < \frac{1}{2}$ and the second level node is not all of M. However, the following *is* a $\frac{1}{2}$F-tree with levels in $\langle \exists, \forall \rangle$:



If $\Phi = \forall x \exists y [x \neq y]$, then



is a $\frac{1}{4}$F-tree for Φ, but not a $\frac{1}{2}$F-tree. As an exercise, the reader should verify that there is no $\frac{3}{4}$F-tree for Φ in our choice of $(\mathcal{M}, \mathcal{D})$. Note once again that, as this example illustates, nodes across the same level in an ϵF-tree need not be distinct.

It should be apparent from the definitions and examples that

Proposition 3.2.7. *For any first order language $\mathcal{L}$, let Φ be an $\mathcal{L}$-sentence and $(\mathcal{M}, \mathcal{D})$ be an ϵ-model of $\mathcal{L}$. Then $(\mathcal{M}, \mathcal{D}) \models_\epsilon \Phi$ iff there exists an ϵE-tree in $(\mathcal{M}, \mathcal{D})$ for Φ. Similarly, $(\mathcal{M}, \mathcal{D}) \vdash_\epsilon \Phi$ iff there exists an ϵF-tree in $(\mathcal{M}, \mathcal{D})$ for Φ.*

Therefore, Φ is (unrestricted/finitely/countably) ϵE-satisfiable iff there exists an ϵE-tree in some (unrestricted/finite/countable) $(\mathcal{M}, \mathcal{D})$ for Φ. This statement holds also when F substitutes E.

Now we are ready to tackle the decidability of monadic relational languages.

Theorem 3.2.8. *Let $\mathcal{L}$ be a monadic relational first order language and $\epsilon \in [0,1]$ be a rational number. Then ϵE-satisfiability in $\mathcal{L}$ is decidable for the unrestricted, the countable, and the finite cases.*

Proof. By lemma (3.2.1), all three ϵE-satisfiability follow from the finite case, so we will prove the latter. In particular, it suffices to consider only satisfaction by probability models $(\mathcal{M}, \mathcal{D})$ with universe $M \subseteq \mathfrak{P}(\llbracket s \rrbracket)$, with $\mathcal{D}$ everywhere defined, and with the property that

$$\forall a, b \in M,\ [\forall l \in \llbracket s \rrbracket, P_l^{\mathcal{M}}(a) \iff P_l^{\mathcal{M}}(b)] \implies a = b.$$

Call such models *simple models*.

Let Φ be a sentence in $\mathcal{L}$, without loss of generality in prenex normal form

$$\Phi := \triangledown_1 x_1 \cdots \triangledown_n x_n \phi(\vec{x})$$

where $\phi(\vec{x})$ is quantifier-free and each $\triangledown_i$ is a quantifier.

By proposition (3.2.7), Φ is ϵE-satisfiable by simple models iff there is an ϵE-tree in some simple $(\mathcal{M}, \mathcal{D})$ for Φ. But all such trees are trees in some subset of $\mathfrak{P}(\llbracket s \rrbracket)$ with height n, which are finite in number. Thus, we only need to devise an algorithm that, for any $\dot{M} \subseteq \mathfrak{P}(\llbracket s \rrbracket)$ and any tree T in $\dot{M}$ with height n, tests whether there exists simple model $(\mathcal{M}, \mathcal{D})$ with universe $\dot{M}$ such that T is an ϵE-tree in $(\mathcal{M}, \mathcal{D})$ for Φ — and it suffices to verify definition (3.2.3). But $\dot{M}$ is the universe of some such simple model $(\mathcal{M}, \mathcal{D})$ iff $\dot{M}$ is so with the interpretations

$$\forall U \in \dot{M},\ P_l^{\mathcal{M}}(U) \iff l \in U.$$

Thus, assuming this structure of $\mathcal{M}$, we can check conditions (1) and (S) of definition (3.2.3) immediately, in finite time.

Finally, for any surviving T and $\mathcal{M}$ with universe $\dot{M}$, we find whether there exists a probability measure $\mathcal{D}$ everywhere defined such that T satisfies condition (2). $(\mathcal{M}, \mathcal{D})$ fulfills this condition iff for every node $V \subseteq \dot{M}$ at a $\forall$-level in T,

$$\sum_{a \in V} \mathcal{D}(a) \geq 1 - \epsilon. \tag{$\diamond_V$}$$

This is equivalent, then, to the feasibility of the linear program LP with variables μ_a for every $a \in \dot{M}$ and inequalities

1. $(\diamond_V)$ with μ_a replacing $\mathcal{D}(a)$,
2. $\mu_a \geq 0$, for each $a \in \dot{M}$, and
3. $\sum_{a \in \dot{M}} \mu_a = 1$.

Evidently, because ϵ is rational, all coefficients in LP are rational. By proposition (1.1.1), LP is solvable in finite time.

Thus condition (2) can be verified effectively.

□

The corresponding result for ϵF-logic will proceed similarly, except that the linear program will involve strict inequalities. The following result of Carver is needed to prove the next lemma.

Proposition 3.2.9 (Carver [19])**.** *Let A be a matrix and let b be a column vector. There exists a vector x with $Ax < b$ iff $y = 0$ is the only solution for*

$$y \geq 0,\ yA = 0,\ yb \leq 0.$$

Lemma 3.2.10. *Let A and B be matrices and let b and c be column vectors. If all entries in A, B, b, c are rational, then there is an algorithm that decides the feasibility of the system*

$$Ax < b, \ Bx = c.$$

Proof. Solve $Bx = c$ for x (for example by Gaussian elimination). If there is no solution x, then the system is not feasible. If there is exactly one solution x, we check whether x satisfies $Ax < b$ and return the result. Finally, if the solution set forms an affine plane of dimension d, then there exist d indices $i_1, \ldots, i_d$ such that each coordinate x_k is a linear combination of $x_{i_1}, \ldots, x_{i_d}$ and 1. Substituting these equations into $Ax < b$ yields a new (strict) linear program $A'x' < b'$ with rational coefficients where $x' = (x_{i_1}, x_{i_2}, \ldots, x_{i_d})$. Evidently the feasibility of the original system is equivalent to the feasibility of $A'x' < b'$, which can be solved via Carver's theorem and proposition (1.1.1). □

Now we are ready to characterize the monadic relational fragment of ϵF-logic. The method of proof follows roughly the same path as for theorem (3.2.8).

Theorem 3.2.11. *Let $\mathcal{L}$ be a monadic relational first order language and $\epsilon \in [0, 1)$ be a rational number. Then ϵF-satisfiability in $\mathcal{L}$ is decidable for the unrestricted, the countable, and the finite cases.*

Proof. Let Φ be a sentence in prenex normal form.

Again, by lemma (3.2.1), it suffices to consider only the finite case and only satisfaction by probability models $(\mathcal{M}, \mathcal{D})$ with universe $M \subseteq \mathfrak{P}([\![s]\!])$, with $\mathcal{D}$ everywhere defined, and with the property that

$$\forall a, b \in M, \ [\forall l \in [\![s]\!], P_l^{\mathcal{M}}(a) \iff P_l^{\mathcal{M}}(b)] \implies a = b.$$

Call such models *simple models*.

Just as in the proof of theorem (3.2.8), it's enough to check in finite time, for each $\dot{M} \subseteq \mathfrak{P}([\![s]\!])$ and each tree T in $\dot{M}$, whether there is a simple $(\mathcal{M}, \mathcal{D})$ with universe $\dot{M}$ such that T is an ϵF-tree in $(\mathcal{M}, \mathcal{D})$ for Φ. If $(\mathcal{M}, \mathcal{D})$ is some such model, then T' is an ϵF-tree in $(\mathcal{M}', \mathcal{D}')$ for Φ, where

- M' is the set of nonnull elements of $\dot{M}$,
- $\mathcal{D}'$ is the restriction of $\mathcal{D}$ to M', and
- T' is derived from T by restricting every node $V \subseteq \dot{M}$ of T to M'.

Therefore we may consider only $\mathcal{D}$ that is everywhere positive.

Again, we can assume $\mathcal{M}$ to have interpretations

$$P_l^{\mathcal{M}}(U) \iff l \in U$$

for each $U \in \dot{M}$ and $l \in [\![s]\!]$. So conditions (2) and (S) of definition (3.2.5) can be easily verified.

Finally, for any surviving T and $\mathcal{M}$ with universe $\dot{M}$, we find whether there exists a probability measure $\mathcal{D}$ everywhere defined such that T satisfies condition (1). Thus T is an ϵF-tree in $(\mathcal{M}, \mathcal{D})$ iff for every node $V \subseteq \dot{M}$ at an $\exists$-level in T,

$$\sum_{a \in V} \mathcal{D}(a) > \epsilon. \qquad (\circ_V)$$

The existence of such $\mathcal{D}$ is equivalent to the feasibility of the strict linear program LP with variables μ_a for every $a \in \dot{M}$ and inequalities

1. $(\circ_V)$ with μ_a replacing $\mathcal{D}(a)$,

2. $\mu_a > 0$, for each $a \in \dot{M}$, and

3. $\sum_{a \in \dot{M}} \mu_a = 1$.

By lemma (3.2.10), this is decidable.

□

By duality, we can phrase theorems (3.2.8) and (3.2.11) thus

Corollary 3.2.12. *Let $\mathcal{L}$ be any monadic relational first order language and $\epsilon \in (0,1)$ be a rational number. Then for $X = E$ or F, (unrestricted/countable/finite) ϵX-satisifiability and validity are both decidable.*

3.3 q-Sentence, q-Trees, and q-Satisfiability

Here we generalize the various concepts of ϵE and ϵF like trees and satisfiability. These generalizations will allow us to express computability reduction results in the next section.

Definition 3.3.1. For any first order language $\mathcal{L}$, a **q-sentence in $\mathcal{L}$** is defined as a string of the form

$$\triangledown_1 x_1 \cdots \triangledown_n x_n \phi(\vec{x})$$

where ϕ is a quantifier-free $\mathcal{L}$-formula in n variables, and for each i,

$$\triangledown_i \in \mathrm{QSET} := \{\exists, \forall\} \cup \{\mathfrak{Q}^{\geq}_{\epsilon}\}_{\epsilon \in \mathbb{Q} \cap [0,1]} \cup \{\mathfrak{Q}^{>}_{\epsilon}\}_{\epsilon \in \mathbb{Q} \cap [0,1]}.$$

Quantifiers of the form $\mathfrak{Q}^{\geq}_{\epsilon}$ are called **weak q-quantifiers**. Quantifiers of the form $\mathfrak{Q}^{>}_{\epsilon}$ are called **strong q-quantifiers**.

If q-sentence Φ has no strong q-quantifiers and no $\forall$, then Φ is called a **qE-sentence**. In the same way, if Φ has no weak q-quantifiers and no $\exists$, then Φ is called a **qF-sentence**.

Definition 3.3.2. Let $\Phi := \triangledown_1 x_1 \cdots \triangledown_n x_n \phi(\vec{x})$ be a first order logic sentence, with $\triangledown_i \in \{\exists, \forall\}$ and ϕ quantifier-free. The **ϵE-coercion of Φ** is defined as the q-sentence

$$\epsilon\text{E-coerce}(\Phi) := \triangledown'_1 x_1 \cdots \triangledown'_n x_n \phi(\vec{x})$$

where

$$\triangledown'_i = \begin{cases} \exists & \text{if } \triangledown_i = \exists \\ \mathfrak{Q}^{\geq}_{1-\epsilon} & \text{if } \triangledown_i = \forall. \end{cases}$$

Likewise, the **ϵF-coercion of Φ** is defined as the q-sentence

$$\epsilon\text{F-coerce}(\Phi) := \triangledown'_1 x_1 \cdots \triangledown'_n x_n \phi(\vec{x})$$

where

$$\triangledown'_i = \begin{cases} \forall & \text{if } \triangledown_i = \forall \\ \mathfrak{Q}^{>}_{\epsilon} & \text{if } \triangledown_i = \exists. \end{cases}$$

For an arbitrary first order sentence Φ, its ϵE-coercion is the coercion of the equivalent prenex normal form. Similarly for ϵF-coercion.

Clearly, both coercion functions are computable. In addition, every q-sentence in the image of ϵE-coerce is a qE-sentence, and every q-sentence in the image of ϵF-coerce is a qF-sentence.

Definition 3.3.3. Let $Q = \langle \triangledown_1, \triangledown_2, \ldots, \triangledown_n \rangle$ be a sequence of quantifiers from QSET. Let $\mathcal{M}$ be a first order model and $\mathcal{D}$ be a probability measure on its universe M. An **q-tree in $(\mathcal{M}, \mathcal{D})$ with levels Q** is defined as a tree in M with height n such that:

1. if $\triangledown_k = \exists$, then all nodes at level k are nonempty subsets of M; level k is called a $\exists$-level.
2. if $\triangledown_k = \forall$, then all nodes at level k are M; level k is called a $\forall$-level.
3. if $\triangledown_k = \mathfrak{Q}^{\geq}_{\epsilon}$, then all nodes at level k are subsets of M with $\mathcal{D}$-measure at least ϵ; level k is called a $\mathfrak{Q}^{\geq}_{\epsilon}$-level.
4. if $\triangledown_k = \mathfrak{Q}^{>}_{\epsilon}$, then all nodes at level k are subsets of M with $\mathcal{D}$-measure greater than ϵ; level k is called a $\mathfrak{Q}^{>}_{\epsilon}$-level.

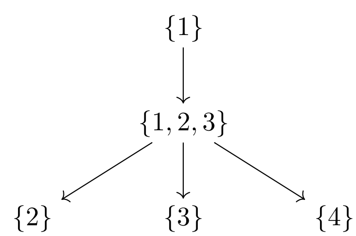
Let $\Phi := \triangledown_1 x_1 \cdots \triangledown_n x_n \phi(\vec{x})$ be a q-sentence. A **q-tree in** $(\mathcal{M}, \mathcal{D})$ **for** Φ is defined as a q-tree in $(\mathcal{M}, \mathcal{D})$ with levels $\langle \triangledown_1, \ldots, \triangledown_n \rangle$ with the additional property that

(S) for every bran $\langle a_i \in V_i \rangle_{i=1}^n$,

$$\mathcal{M} \models \phi(\vec{a}).$$

When Φ and $(\mathcal{M}, \mathcal{D})$ are clear from the context, we will simply use the term **q-tree**.

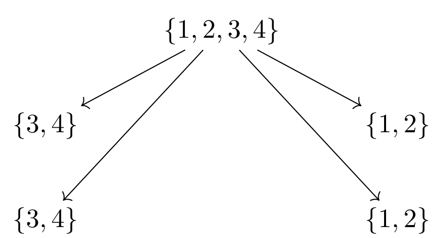
Example 3.3.4. Let $M = [\![4]\!]$ and $\mathcal{D}$ be the uniform distribution. The following q-tree in $(\mathcal{M}, \mathcal{D})$



is a q-tree with levels Q for $Q = \langle \exists, \mathfrak{Q}^{\geq}_{3/4}, \mathfrak{Q}^{>}_{0} \rangle$, $\langle \mathfrak{Q}^{\geq}_{1/4}, \exists, \exists \rangle$, and $\langle \mathfrak{Q}^{>}_{0}, \mathfrak{Q}^{>}_{1/2}, \mathfrak{Q}^{\geq}_{0} \rangle$, but not for $Q = \langle \exists, \forall, \exists \rangle$ or $\langle \mathfrak{Q}^{>}_{1/2}, \mathfrak{Q}^{>}_{3/4}, \mathfrak{Q}^{>}_{1/4} \rangle$. Thus T can be an ϵE-tree with levels Q but not necessarily be a q-tree with levels Q, as $\forall$ is interpreted differently.

It is also a q-tree for Φ if $\Phi = \exists x \mathfrak{Q}^{>}_{1/2} y \exists z[x + y = z]$ but not if $\Phi = \exists x \forall y \exists z[x + y = z]$.

The following q-tree in $(\mathcal{M}, \mathcal{D})$



is a q-tree with levels Q for $Q = \langle \forall, \exists \rangle$ and $\langle \mathfrak{Q}^{\geq}_{1}, \mathfrak{Q}^{>}_{1/4} \rangle$ but not for $Q = \langle \exists, \forall \rangle$ or $\langle \forall, \mathfrak{Q}^{>}_{1/2} \rangle$.

It is also a q-tree for Φ if $\Phi = \forall x \exists y[x \neq y]$ or $\mathfrak{Q}^{\geq}_{1} x \mathfrak{Q}^{>}_{0} y[x \neq y]$ but not if $\Phi = \forall x \forall y[x \neq y]$ or $\forall x \mathfrak{Q}^{\geq}_{3/4} y[x \neq y]$.

Clearly, q-trees are generalizations of both ϵE-trees and ϵF-trees: An ϵE-tree in $(\mathcal{M}, \mathcal{D})$ for Φ is exactly a q-tree in $(\mathcal{M}, \mathcal{D})$ for ϵE-coerce(Φ). Likewise for ϵF-trees.

Definition 3.3.5. A pair $(\mathcal{M}, \mathcal{D})$ of first order model $\mathcal{M}$ and probability measure $\mathcal{D}$ on M is said to **q-satisfy** a q-sentence $\Phi := \triangledown_1 x_1 \cdots \triangledown_n x_n \phi(\vec{x})$, written

$$(\mathcal{M}, \mathcal{D}) \models^{\mathrm{q}} \Phi,$$

if there exists a q-tree in $(\mathcal{M}, \mathcal{D})$ for Φ.

A q-sentence Φ is said to be **q-satisfiable** if some $(\mathcal{M}, \mathcal{D})$ q-satisfies Φ.

Again, q-satisfiability is just a generalization of ϵE- and ϵF-satisfibiliy: for a first order sentence Ψ,

$$(\mathcal{M}, \mathcal{D}) \models_\epsilon \Psi \iff (\mathcal{M}, \mathcal{D}) \models^{\mathrm{q}} \epsilon\text{E-coerce}(\Psi).$$

The obvious analogue holds for ϵF-satisfiability as well.

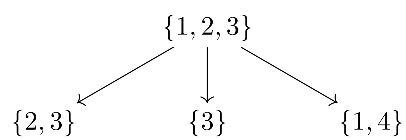
Definition 3.3.6. Let $Q = \langle \nabla_1, \ldots, \nabla_n \rangle$ and $Q' = \langle \nabla'_1, \ldots, \nabla'_m \rangle$. Let $(\mathcal{M}, \mathcal{D})$ be a pair of first order model and probability measure. Suppose T and T' are q-trees in $(\mathcal{M}, \mathcal{D})$ respectively with levels Q and Q'. The **wedge product** $T \wedge T'$ is defined as the q-tree with levels $\langle \nabla_1, \ldots, \nabla_n, \nabla'_1, \ldots, \nabla'_m \rangle$ (thus of height $n + m$) constructed as follows:

> For every leaf node V in T and every element $a \in V$, set a copy of T' as the subtree under a.

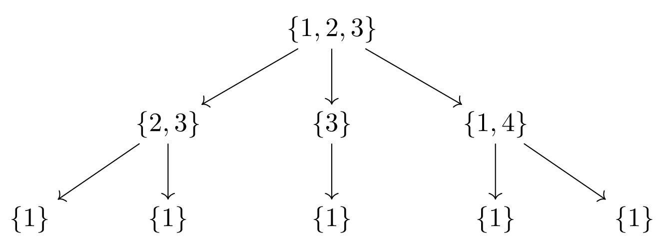
The expression $T_1 \wedge T_2 \wedge \cdots \wedge T_k$ is parsed as

$$(((T_1 \wedge T_2) \wedge \cdots) \wedge T_k).$$

Example 3.3.7. Let T be



and T' be the singleton tree $\{1\}$. Then $T \wedge T'$ is the q-tree



Proposition 3.3.8. *Let $\Phi := \nabla_1 x_1 \cdots \nabla_n x_n \phi(\vec{x})$ and $\Phi' := \nabla'_1 y_1 \cdots \nabla'_m y_m \phi'(\vec{y})$ be q-sentences. Let $(\mathcal{M}, \mathcal{D})$ be a pair of first order model and probability measure. Suppose T and T' are q-trees in $(\mathcal{M}, \mathcal{D})$ respectively for Φ and Φ'. Then $T \wedge T'$ is a q-tree in $(\mathcal{M}, \mathcal{D})$ for the q-sentence*

$$\nabla_1 x_1 \cdots \nabla_n x_n \nabla'_1 y_1 \cdots \nabla'_m y_m [\phi(\vec{x}) \wedge \phi'(\vec{y})].$$

Proof. We verify definition (3.3.3). Conditions (1), (2), (3), and (4) follow easily from the respective conditions on T and T'.

Each bran of $T \wedge T'$ is a concatenation

$$\langle a_1 \in V_1,\ \ldots,\ a_n \in V_n,\ b_1 \in W_1,\ \ldots,\ b_m \in W_m \rangle$$

of a bran $\langle a_i \in V_i \rangle_{i=1}^n$ of T and a bran $\langle b_i \in W_i \rangle_{i=1}^m$ of T'. Thus

$$(\mathcal{M}, \mathcal{D}) \models^{\mathrm{q}} \phi(\vec{a}) \wedge \phi'(\vec{b})$$

by T and T''s property (S). So condition (S) holds for $T \wedge T'$ as well. ∎

This proposition immediately yields

Proposition 3.3.9. *Let Φ and Φ' be as above. A pair $(\mathcal{M}, \mathcal{D})$ simultaneously q-satisfies Φ and Φ' iff*

$$(\mathcal{M}, \mathcal{D}) \models^{\mathrm{q}} \nabla_1 x_1 \cdots \nabla_n x_n \nabla'_1 y_1 \cdots \nabla'_m y_m [\phi(\vec{x}) \wedge \phi'(\vec{y})].$$

3.4 Finite ϵE-satisfiability

In this subsection we will show that ϵE-satisfiability is Σ_1^0-complete for rational ϵ strictly between 0 and 1. The main reason that we would like to work with rational ϵ is the following set of tools provided by Kuyper and Terwijn:

Lemma 3.4.1 (Kuyper-Terwijn inter-reduction [13][10])**.** *Let*

- *$\mathcal{L}$ be a countable first-order language not containing function symbols or equality,*
- *$\mathcal{L}'$ be the language obtained by adding an infinite number of unary predicates to $\mathcal{L}$, and*
- *ϵ_0, ϵ_1 be rational such that*
  1. *$0 \le \epsilon_0 \le \epsilon_1 < 1$, or*
  2. *$0 < \epsilon_1 \le \epsilon_0 \le 1$.*

Then there is a computable f mapping $\mathcal{L}$-sentences to $\mathcal{L}'$-sentences such that ϕ is ϵ_0E-satisfiable iff $f(\phi)$ is ϵ_1E-satisfiable.

More generally [4], this reduction works "per quantifier": For any $\epsilon \in (0,1) \cap \mathbb{Q}$, there exists a computable function $\mathcal{F}_\epsilon$ mapping qE-sentences in $\mathcal{L}$ to $\mathcal{L}'$-sentences such that the following are equivalent:

1. *there exists a pair $(\mathcal{M}, \mathcal{D})$ such that*
$$(\mathcal{M}, \mathcal{D}) \models^q \Phi.$$
2. *$\mathcal{F}_\epsilon(\Phi)$ is ϵE-satisfiable.*

Even though the theorem only applies to full satisfiability, the proof works exactly the same for finite (and countable) satisfiability, because the only model construction in the proof is the duplication of a given satisfying model a finite number of times, which preserves finiteness (and countability). Thus

Lemma 3.4.2. *Let $\mathcal{L}$ and $\mathcal{L}'$ be defined as above. For any $\epsilon \in (0,1) \cap \mathbb{Q}$, there exists a computable function $\mathcal{F}_\epsilon$ mapping qE-sentences in $\mathcal{L}$ to $\mathcal{L}'$-sentences such that the following are equivalent:*

1. *there exists a pair $(\mathcal{M}, \mathcal{D})$ with M finite such that*
$$(\mathcal{M}, \mathcal{D}) \models^q \Phi.$$
2. *$\mathcal{F}_\epsilon(\Phi)$ is finitely ϵE-satisfiable.*

In particular, we are interested in the following case

Lemma 3.4.3. *Let $\mathcal{L}$ and $\mathcal{L}'$ be defined as above and fix rational $\epsilon \in (0,1)$. There is a computable function f_ϵ such that, for any finite set of rationals $J \subseteq \mathbb{Q} \cap [0,1]$ and $\mathcal{L}$-sentences $\{\Psi_\alpha\}_{\alpha \in J}$, the following are equivalent:*

1. *there exists a pair $(\mathcal{M}, \mathcal{D})$ such that, for each $\alpha \in J$, $(\mathcal{M}, \mathcal{D})$ is a finite α-model and*
$$(\mathcal{M}, \mathcal{D}) \models_\alpha \Psi_\alpha.$$
2. *$f_\epsilon(\{\Psi_\alpha\}_{\alpha \in J})$ is finitely ϵE-satisfiable.*

[4] see [13, remark 2.14].

Proof. We have

$$(\mathcal{M}, \mathcal{D}) \models_\alpha \Psi_\alpha \iff (\mathcal{M}, \mathcal{D}) \models^q \epsilon\text{E-coerce}(\Psi_\alpha).$$

By repeated applications of proposition (3.3.9), $(\mathcal{M}, \mathcal{D})$ simultaneously q-satisfies all ϵE-coerce(Ψ_α) iff $(\mathcal{M}, \mathcal{D}) \models^q \Gamma$ for some q-sentence Γ. Now apply lemma (3.4.2). □

Therefore, we could compute the simultaneous E-satisfiability over different error parameters $\alpha \in [0, 1]$ by using only one fixed $\epsilon \in (0, 1)$. This property of $\epsilon \in (0, 1)$ turns out to be powerful enough to allow the encoding of the halting set by sentences in ϵE-logic. It also distinguishes the case of $\epsilon \in (0, 1)$ from the case of $\epsilon = 0$, for which lemma (3.4.3) is not applicable: whereas the E-satisfiability of the latter is decidable by theorem (3.1.3), that of the former, as will be shown next, is Σ^0_1-complete.

We now commence the first half of the completeness proof.

Theorem 3.4.4. *Let $\mathcal{L}$ be any countable first-order language with an infinite number of unary predicates and at least three binary predicates. Finite ϵE-satisfiability for $\mathcal{L}$-sentences is Σ^0_1-hard for rational $1 > \epsilon > 0$.*

Proof. The main idea of the proof, as remarked above, is that we want to reduce the halting problem to finite ϵE-satisfiability. Specifically we will show that there is a reduction from the set of Turing machines that halt on empty input (which is Σ^0_1-complete) to the set of finite ϵE-satisfiables. The proof is loosely based on the proof of Trachtenbrot's theorem in Libkin [15, p. 166] and the proof of Σ^1_1-hardness of ϵE-satisfaction in Kuyper [10, Thm. 7.6]. Since by theorem (3.4.3) there is a reduction between finite ϵ_0E- and ϵ_1E-satisfiability in $\mathcal{L}$ for any rational $\epsilon_0, \epsilon_1 \in (0, 1)$, it suffices to establish the case of $\epsilon = \frac{1}{2}$.

Suppose that $M = (Q, \nabla, \delta, q_0, Q_a, Q_r)$ is a single-tape Turing machine, where

- Q is the set of states,
- ∇ is the tape alphabet,
- q_0 is the initial state,
- Q_a and Q_r are respectively the sets of accepting and rejecting states, and
- $\delta : Q \times \nabla \to Q \times \nabla \times \{\mathbf{L}, \mathbf{R}\}$ is the transition function.

Since we are only interested in Turing machines with empty input, ∇ can be assumed WLOG to be $\{0, 1\}$ with 0 representing the blank symbol.

In what follows, we break into three sections the proof for encoding the halting of M as a finite $\frac{1}{2}$E-satisfiability problem. Section 1 describes the first order language used. Section 2 constructs the sentence Ψ which is finitely $\frac{1}{2}$E-satisfiability iff M halts. Finally section 3 proves that Ψ indeed has such a property.

Part 1 (The vocabulary). We define vocabulary

$$\sigma := \{\underline{\min}, \underline{\max}, N(\cdot), \dot{=}, <, R(\cdot,\cdot), T(\cdot,\cdot), H(\cdot,\cdot), (S_q(\cdot))_{q \in Q}\}.$$

(The constants $\underline{\min}, \underline{\max}$ in σ can be replaced by unary predicates, so the theorem as stated will still stand.)

The intuition behind this vocabulary, which will be formalized by the axioms below, is as follows:

- Elements satisfying N will be "roughly" a set of positive measure, linearly ordered (by $<$) elements that will be our measure of time and space, such that $\underline{\min}$ and $\underline{\max}$ are the minimal and the maximal elements of this chain — "Roughly", because using ϵE-logic, we cannot specify that elements of measure 0 do not satisfy N (in fact, we cannot say anything about elements of measure 0). A nontrivial part of this proof is used to maneuver around these "phantom elements".

- $\doteqdot$ is a binary relation mimicking equality. We avoid using true equality so that we can use the computable reductions of theorem (3.4.3).
- $T(p,t)$, where $p,t \in N$, represents that at time t, there is a 1 at position p on the tape.
- $H(p,t)$, where $p,t \in N$, represents that at time t, the head of the machine is at position p.
- $S_q(t)$, where $t \in N$, represents that at time t, the state of the machine is q.
- R is an auxiliary relation that is used to force certain measures to be equal. Its purpose will become more clear over the course of the proof.

Part 2 (The encoding sentence). In this section we define the sentence Ψ that encodes whether M will halt. Ψ will be of the form

$$\Psi := f(\{\mathrm{T}_\alpha\}_{\alpha \in J})$$

where

- $J = \{0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}\}$
- f is the reduction function f_ϵ from lemma (3.4.3) for J and $\epsilon = \frac{1}{2}$, and
- T_α is a sentence for each $\alpha \in J$.

Thus Ψ is finitely ϵ_0E-satisfiable iff each T_α is finitely αE-satisfiable.

For a formula ϕ in prenex normal form, we recursively define ϕ^N (called ϕ *relativized to* N):

1. if ϕ is quantifier-free, then $\phi^N = \phi$
2. if $\phi(\vec{y}) = \forall x \psi(x, \vec{y})$, then $\phi(\vec{y})^N = \forall x(N(x) \to \psi(x, \vec{y})^N)$
3. if $\phi(\vec{y}) = \exists x \psi(x, \vec{y})$, then $\phi(\vec{y})^N = \exists x(N(x) \land \psi(x, \vec{y})^N)$.

($\mathbf{T_0}$). T_0 will consist of the conjunction of the following sentences (because $\forall x$ here should be interpreted as "for almost all x", or, as we only deal with finite models here, as "for all x with positive measure," we will write $\underset{+}{\forall}$ for the sake of clarity):

1. All axioms of equality:
   (a) $\doteqdot$ is an equivalence relation:
   $$\underset{+}{\forall} x(x \doteqdot x)$$
   $$\underset{+}{\forall} x \underset{+}{\forall} y(x \doteqdot y \to y \doteqdot x)$$
   $$\underset{+}{\forall} x \underset{+}{\forall} y \underset{+}{\forall} z(x \doteqdot y \land y \doteqdot z \to x \doteqdot z)$$
   (b) the indiscernability of identicals: for each atomic formula π,
   $$\pi(a_1, \ldots, a_n) \land \bigwedge_{i=1}^{n} a_i \doteqdot b_i \to \pi(b_1, \ldots, b_n)$$
2. $<$ is a linear order on all elements of N with nonzero measure:
   $$(\underset{+}{\forall} x \underset{+}{\forall} y(x \doteqdot y \lor x < y \lor y < x))^N$$
   $$(\underset{+}{\forall} x \underset{+}{\forall} y(\neg x \doteqdot y \to (x < y \leftrightarrow \neg y < x)))^N$$
   $$(\underset{+}{\forall} x \underset{+}{\forall} y \underset{+}{\forall} z(x < y \land y < z \to x < z))^N$$

3. $\underline{\min}$ and $\underline{\max}$ are respectively minimal and maximal in $<$:

$$N(\underline{\min}) \wedge N(\underline{\max})$$
$$(\underset{+}{\forall} x(x \doteq \underline{\min} \vee \underline{\min} < x))^N$$
$$(\underset{+}{\forall} x(x \doteq \underline{\max} \vee x < \underline{\max}))^N$$

4. Initially, M is in state q_0, the head is in the first position, and the tape has all zeros:

$$S_{q_0}(\underline{\min})$$
$$\underset{+}{\forall} p(p \doteq \underline{\min} \leftrightarrow H(p, \underline{\min}))$$
$$(\underset{+}{\forall} p \neg T(p, \underline{\min}))^N$$

5. For any time t, M is in a unique state:

$$\left(\underset{+}{\forall} t \left(\bigvee_{q \in Q} S_q(t) \wedge \bigwedge_{q,q' \in Q} \neg(S_q(t) \wedge S_{q'}(t))\right)\right)^N$$

6. A set of sentences encoding the transition function δ.

   First we define a binary relation $\succ$. The expression $t' \succ t$ is a shorthand for the conjunction of

   - "t' is greater than t"
   $$t < t'$$
   - "for all s, s is less than t' iff s is at most t"
   $$\underset{+}{\forall} s(s < t' \leftrightarrow (s < t \vee s \doteq t))$$
   - "for all s, s is greater than t iff s is at least t'"
   $$\underset{+}{\forall} s(t < s \leftrightarrow (t' < s \vee t' \doteq s)).$$

   Thus $t' \succ t$ says that "t' is a successor or t."

   For any formula $\phi(t, \vec{x})$, let $\phi(t+1, \vec{x})$ be defined as the shorthand for the following relativized implication

   $$[\underset{+}{\forall} t'(t' \succ t \rightarrow \phi(t', \vec{x}))]^N.$$

   Hence $\phi(t+1, \vec{x})$ states that "ϕ holds for the successor of t":

   Similarly, let $\phi(t-1, \vec{x})$ be defined as the shorthand for

   $$[\underset{+}{\forall} t'(t \succ t' \rightarrow \phi(t', \vec{x}))]^N.$$

   The expression $\phi(t-1, \vec{x})$ asserts that "ϕ holds for the predecessor of t."

These shorthands are well-defined when applied to multiple variables. For example, $\phi(p-1, t+1)$ is the shorthand for

$$\begin{aligned} & [\underset{+}{\forall} p'(p \succ p' \to \phi(p', t+1))]^N \\ \mapsto \quad & [\underset{+}{\forall} p'(p \succ p' \to [\underset{+}{\forall} t'(t' \succ t \to \phi(p', t'))]^N)]^N \\ \equiv \quad & [\underset{+}{\forall} p' \underset{+}{\forall} t'(p \succ p' \wedge t' \succ t \to \phi(p', t'))]^N \end{aligned}$$

where in the middle $\mapsto$ means "expands into."

Now we turn to the task of encoding the transition function.

Suppose the transition function has rule $\delta(q, w) = (q', w', S)$ for some $q \in Q; w, w' \in \nabla; S \in \{\mathbf{L}, \mathbf{R}\}$. For each q and w, we define the sentence

$$\rho_{q,w} := [\underset{+}{\forall} p \underset{+}{\forall} t(\textsc{cond}(p, t) \to \textsc{trans}(p, t))]^N$$

where COND and TRANS are constructed as follows.

(a) $\textsc{cond}(p, t)$ checks that "the machine M at time t has state q, has its head pointing at cell p, and the character under the head is w": explicitly, $\textsc{cond}(p, t)$ is the conjunction of the following:

  i. "The state of M is q at time t"

$$S_q(t)$$

  ii. "The head of M is above cell p"

$$H(p, t)$$

  iii. "The character at cell p is w" (exactly one of the following sentences belongs to the conjunction, depending on which condition is satisfied)

$$\begin{cases} T(p, t) & \text{if } w = 1 \\ \neg T(p, t) & \text{if } w = 0 \end{cases}$$

(b) $\textsc{trans}(p, t)$ asserts that "M at time $t+1$ has state q' and has moved S from cell p; the cell at p now contains the symbol w'": explicitly, $\textsc{trans}(p, t)$ is the conjunction of the following (in every set of alternatives, exactly one of the sentences belongs to the conjunction, depending on which condition is satisfied):

  i. "The state of M is q' at time $t+1$"

$$S_{q'}(t+1)$$

  ii. "At time $t+1$: If $S = \mathbf{R}$, the head of M is at cell $p+1$. If $S = \mathbf{L}$ and $p = \underline{\min}$, the head of M is at cell $\underline{\min}$. Otherwise, the head of M is at cell $p-1$."

$$\begin{cases} H(p+1, t+1) & \text{if } S = \mathbf{R} \\ (p \doteqdot \underline{\min} \to H(p, t+1)) \wedge (\neg p \doteqdot \underline{\min} \to H(p+1, t+1) & \text{if } S = \mathbf{L} \end{cases}$$

  iii. "At time $t+1$, cell p contains symbol w'"

$$\begin{cases} T(p, t+1) & \text{if } w' = 1 \\ \neg T(p, t+1) & \text{if } w' = 0 \end{cases}$$

iv. "All cells other than those involved in (6(b)ii) are unaffected".

$$\begin{cases} \underset{+}{\forall} p'[\neg p' \doteqdot p \wedge \neg p' \succ p \rightarrow (T(p',t) \leftrightarrow T(p',t+1))] & \text{if } S = \mathbf{R} \\ \underset{+}{\forall} p'[\neg p' \doteqdot p \wedge \neg p \succ p' \rightarrow (T(p',t) \leftrightarrow T(p',t+1))] & \text{if } S = \mathbf{L} \end{cases}$$

7. We assert that at time $\underline{\max}$, M arrives at an accepting or rejecting state:

$$\bigvee_{q \in Q_a \cup Q_r} S_q(\underline{\max})$$

8. For reasons that will become clear later, for all x, y of positive measure, we need to let $R(x,y)$ hold only when y is not in N:

$$\underset{+}{\forall} x \underset{+}{\forall} y (R(x,y) \rightarrow \neg N(y))$$

This finishes the description of the sentence T_0.

($\mathbf{T}_{\frac{1}{4}}$ and $\mathbf{T}_{\frac{3}{4}}$). $\mathrm{T}_{\frac{1}{4}}$ and $\mathrm{T}_{\frac{3}{4}}$ are respectively the sentences $\forall x(x = \underline{\min})$ and $\forall x(x \neq \underline{\min})$. $f_{\frac{1}{4}}(\mathrm{T}_{\frac{1}{4}})$ and $f_{\frac{3}{4}}(\mathrm{T}_{\frac{3}{4}})$ are simultaneously $\frac{1}{2}$E-satisfiable iff the measure of $\underline{\min}$ is at least $\frac{3}{4}$ and the measure of all other elements is at least $\frac{1}{4}$. Thus $\mathrm{T}_{\frac{1}{4}}$ and $\mathrm{T}_{\frac{3}{4}}$ force the measure of $\underline{\min}$ to be exactly $\frac{1}{4}$.

($\mathbf{T}_{\frac{1}{2}}$). $\mathrm{T}_{\frac{1}{2}}$ is the following conjunction of sentences (because the $\forall x$ here should be interpreted as "for a set of x with measure at least $\frac{1}{2}$", we write $\underset{\geq 1/2}{\forall}$ for the sake of clarity):

1. the set of elements in N takes up measure exactly $\frac{1}{2}$:

$$\underset{\geq 1/2}{\forall} x N(x) \wedge \underset{\geq 1/2}{\forall} x \neg N(x)$$

2. we want each element of N to have measure equal to the total measure of all greater elements. This is where the padding relation R is used.

   First, for $\phi(x)$ a formula of a single free variable x, we define the following shorthand

$$\ulcorner \underset{>0}{\forall} x \in N(\phi(x)) \urcorner := \underset{\geq 1/2}{\forall} x(N(x) \wedge \phi(x))$$

   The RHS says "for a set X of measure at least $1/2$, $X \subseteq N$ and every element of X satisfies ϕ." If we assume that (1) is $\frac{1}{2}$E-satisfied, then $\ulcorner \underset{>0}{\forall} x \in N(\phi(x)) \urcorner$ is equivalent to "all elements in N of positive measure must satisfy ϕ."

   Secondly, for formula $\psi(y, \vec{x})$, we define the following shorthand

$$\ulcorner \Pr_y[\psi(y,\vec{x})] = 1/2 \urcorner := [\underset{\geq 1/2}{\forall} y \psi(y,\vec{x})] \wedge [\underset{\geq 1/2}{\forall} y' \neg \psi(y',\vec{x})]$$

   The conjuncts on the right respectively assert that "the probability of y satisfying $\psi(y,\vec{x})$ is at least $1/2$" and "the probability of y' not satisfying $\psi(y',\vec{x})$ is at least $1/2$." Hence $\ulcorner \Pr_y[\psi(y,\vec{x})] = 1/2 \urcorner$ says that "the probability of y satisfying $\psi(y,\vec{x})$ is exactly $1/2$.

   Finally, we define the actual sentences in the conjunction of $\mathrm{T}_{\frac{1}{2}}$:

(Recall that $\wedge$ has precedence over $\vee$ for parsing)

$$\ulcorner \underset{>0}{\forall} x \in N(x \doteq \underline{\max} \vee \ulcorner \Pr_y[R(x,y) \vee N(y) \wedge x < y] = 1/2 \urcorner) \urcorner$$
$$\ulcorner \underset{>0}{\forall} x \in N(x \doteq \underline{\max} \vee \ulcorner \Pr_y[R(x,y) \vee x \doteq y] = 1/2 \urcorner) \urcorner$$

If clause (8) of T_0 is 0E-satisfied ("If $R(x,y)$ holds then y has measure 0 or is not in N"), then the disjuncts $R(x,y)$ and $N(y) \wedge x < y$ are disjoint. Therefore, the above two sentences together express "for all x in N that's not $\underline{\max}$ and not null, the measure of the strict final segment of x is the same as the probability of x itself; they are both $(\frac{1}{2} - \Pr_y[R(x,y)])$." Forcing the equality of these two measures is the purpose of the predicate $R(\cdot,\cdot)$, which has otherwise no uses.

This concludes the construction of the reducing sentence Ψ.

Part 3 (The reduction). Now we show that M halts if and only if Ψ has a finite E-satisfying model.

(M halts $\implies$ Ψ satisfiable). Suppose M halts in time m. We define the finite satisfying model $(\mathcal{W}, \mathcal{D})$ thus:

- Let the universe W be the set $\{1, 2, \ldots, 2m\}$.
- Let $\doteq^{\mathcal{W}}$ be true equality $=$.
- Let $N^{\mathcal{W}}$ be $\{1, 2, \ldots, m\}$
- Let $a <^{\mathcal{W}} b$ for $a, b \in W$ be defined to agree with the natural ordering on W.
- Let $\underline{\min}^{\mathcal{W}} = 1$ and $\underline{\max}^{\mathcal{W}} = m$.
- Define the measure $\mathcal{D}(i) = \mathcal{D}(i+m) = 2^{-i-1}$ for $i \in [1, m-1]$, and define $\mathcal{D}(m) = \mathcal{D}(2m) = 2^{-m}$.
- Define $R^{\mathcal{W}}(i,j)$ iff $i \geq j - m \geq 1$
- Define $H^{\mathcal{W}}(i,j)$ iff $j \leq m$ and M's head is at position i at time j.
- Define $T^{\mathcal{W}}(i,j)$ iff $j \leq m$ and the tape's symbol at position i at time j is 1.
- Define $S_q^{\mathcal{W}}(i)$ iff M is in state q at time i.

Since all elements of $\mathcal{W}$ have positive measure, all $\forall$ quantifiers in T_0 are interpreted classically. Therefore one can easily check that $(\mathcal{W}, \mathcal{D})$ 0E-satisfies T_0.

As $\underline{\min} = 1$ has measure $\frac{1}{4}$, $\mathrm{T}_{\frac{1}{4}}$ and $\mathrm{T}_{\frac{3}{4}}$ are satisfied.

$N^{\mathcal{W}}$ obviously has measure $1/2$, so the first clause of $\mathrm{T}_{\frac{1}{2}}$ is satisfied.

Finally, consider clauses (2) in $\mathrm{T}_{\frac{1}{2}}$.

For a fixed $i < m$, the measure of $\{j : R^{\mathcal{W}}(i,j)\}$ is $\sum_{k=1}^{i} 2^{-k-1} = 2^{-1} - 2^{-i-1}$, and the measure of $j \in N^{\mathcal{W}}$ such that $i < j$ is $\sum_{k=i+1}^{m-1} 2^{-k-1} + 2^{-m} = 2^{-i-1}$. Thus, for this fixed i,

$$\Pr_{j \sim \mathcal{D}}[R(i,j) \vee N(j) \wedge i < j] = \frac{1}{2}.$$

Letting i vary, we can conclude that "for all x in $N_0^{\mathcal{W}}$, either x is $\underline{\max}^{\mathcal{W}}$ or the probability of y such that $R(x,y) \vee N(y) \wedge x < y$ holds is exactly $1/2$." In other words, the following clause in $\mathrm{T}_{\frac{1}{2}}$

$$\ulcorner \underset{>0}{\forall} x \in N(x \doteq \underline{\max} \vee \ulcorner \Pr_y[R(x,y) \vee N(y) \wedge x < y] = 1/2 \urcorner) \urcorner$$

holds in $(\mathcal{W}, \mathcal{D})$.

Similarly, as $\mathcal{D}(i) = 2^{-i-1}$ for any $i < m$,

$$\Pr_{j \sim \mathcal{D}}[R(i,j) \vee i \doteqdot j] = \frac{1}{2},$$

so the following clause in $\mathrm{T}_{\frac{1}{2}}$

$$\ulcorner \forall_{>0} x \in N(x \doteqdot \underline{\max} \vee \ulcorner \Pr_y[R(x,y) \vee x \doteqdot y] = 1/2 \urcorner) \urcorner$$

is $\frac{1}{2}$E-satisfied by $(\mathcal{W}, \mathcal{D})$.

Therefore $(\mathcal{W}, \mathcal{D})$ $\frac{1}{2}$E-satisfies all of $\mathrm{T}_{\frac{1}{2}}$, as desired.

(M halts $\Longleftarrow$ Ψ satisfiable). Let $(\mathcal{W}, \mathcal{D})$ be a finite $\frac{1}{2}$-model. By proposition (2.3.4), we can assume $\mathcal{D}$ is defined on all subsets of $\mathcal{W}$ and is a probability model. Suppose $(\mathcal{W}, \mathcal{D}) \models_{\frac{1}{2}} \Psi$. We wish to show that the Turing machine M halts. Our strategy will be to show that $\underline{\min}^{\mathcal{W}}$ and $\underline{\max}^{\mathcal{W}}$ have positive measures, and every element between them has positive measure. Then checking the sentences T_0 encoding M becomes straightforward, as $\forall$ is interpreted classically on this linear chain.

Let $N_0^{\mathcal{W}}$ be the subset of elements of $N^{\mathcal{W}}$ of positive measure. By the axioms of equality of T_0, the restriction $\doteqdot_0^{\mathcal{W}}$ of $\doteqdot^{\mathcal{W}}$ to $N_0^{\mathcal{W}}$ is an equivalence relation and satisfies the indiscernability of identicals. Therefore $\mathcal{W} \equiv \mathcal{W}/\doteqdot_0^{\mathcal{W}}$ as first order models, and for all ϵ, $(\mathcal{W}, \mathcal{D})$ is ϵ-elementarily equivalent to the probability model $(\mathcal{W}/\doteqdot_0^{\mathcal{W}}, \mathcal{D}')$, where $\mathcal{D}'$ is defined by assigning each equivalence class $[a]$ (as a point of $\mathcal{W}/\doteqdot_0^{\mathcal{W}}$) the $\mathcal{D}$-measure of $[a]$ as a set. We are thus justified in assuming that $\doteqdot$ is true equality on $N_0^{\mathcal{W}}$ henceforth.

By $\mathrm{T}_{\frac{1}{4}}$ and $\mathrm{T}_{\frac{3}{4}}$ we know that $\mathcal{D}(\underline{\min}^{\mathcal{W}}) = \frac{1}{4}$, so $\underline{\min}^{\mathcal{W}} \in N_0^{\mathcal{W}}$. By clause (1) of $\mathrm{T}_{\frac{1}{2}}$ we know that $\mathcal{D}(N^{\mathcal{W}}) = \frac{1}{2}$.

Any $\forall$ quantifier relativized to N can be interpreted classically on $N_0^{\mathcal{W}}$. Thus, by clauses (2) of T_0, $<^{\mathcal{W}}$ defines a linear order on $N_0^{\mathcal{W}}$, and, by clauses (3) of T_0, $\underline{\min}^{\mathcal{W}}$ is the unique minimal element. However, it is not immediate whether $\underline{\max}^{\mathcal{W}}$ has positive measure and thus is the unique maximal element of $N_0^{\mathcal{W}}$.

But by clause (8) of T_0, we have $(\mathcal{W}, \mathcal{D}) \models_0 \forall x \forall y (R(x,y) \to \neg N(y))$. Then, for a fixed x of positive measure, the set of y where $R^{\mathcal{W}}(x,y)$ holds intersects $N^{\mathcal{W}}$ with measure 0. Thus, by clauses (2) of $\mathrm{T}_{\frac{1}{2}}$, if $a \in N_0^{\mathcal{W}}$ is not $\underline{\max}^{\mathcal{W}}$, then the equations

$$\Pr_{y \sim \mathcal{D}}[R(a,y)] + \Pr_{y \sim \mathcal{D}}[N(y) \wedge a < y] = \frac{1}{2} \qquad \text{and}$$
$$\Pr_{y \sim \mathcal{D}}[R(a,y)] + \Pr_{y \sim \mathcal{D}}[a \doteqdot y] = \frac{1}{2}$$

hold. For such an a,

$$\Pr_{y \sim D}[N(y) \wedge a < y] = \Pr_{y \sim \mathcal{D}}[a \doteqdot y] = \mathcal{D}(a). \qquad (\star)$$

Lemma 3.4.5. $\underline{\max}^{\mathcal{W}} \in N_0^{\mathcal{W}}$

Proof. Let $\underline{2}^{\mathcal{W}}$ denote the immediate successor of $\underline{\min}^{\mathcal{W}}$ in $N_0^{\mathcal{W}}$; it exists since $1/4 = \mathcal{D}(\underline{\min}^{\mathcal{W}}) < \mathcal{D}(N^{\mathcal{W}}) = 1/2$ and $\underline{\min}^{\mathcal{W}}$ is minimal in the finite set $N_0^{\mathcal{W}}$. Suppose $\underline{2}^{\mathcal{W}} \neq \max^{\mathcal{W}}$, and

$$\tau := \mathcal{D}(\underline{2}^{\mathcal{W}}) \quad \text{and} \quad \xi := \Pr_{y \sim \mathcal{D}}[N(y) \wedge \underline{2}^{\mathcal{W}} < y].$$

Then we have

$$\tau + \xi = \Pr_{y \sim \mathcal{D}}[N(y) \wedge \underline{\min}^{\mathcal{W}} < y] = \mathcal{D}(\underline{\min}^{\mathcal{W}}) = \frac{1}{4}, \qquad \text{and}$$
$$\tau = \xi, \qquad \text{by } (\star).$$

Hence $\mathcal{D}(\underline{2}^{\mathcal{W}}) = \tau = \xi = \frac{1}{8}$.

In general, if the $(n-1)$-fold successor $\underline{n}^{\mathcal{W}}$ of $\underline{\min}^{\mathcal{W}}$ has probability 2^{-n-1} for each $n < m$, then the $(m-1)$-fold successor $\underline{m}^{\mathcal{W}}$ of $\underline{\min}^{\mathcal{W}}$ exists, and a) is either $\underline{\max}^{\mathcal{W}}$, or b) satisfy the following equations

$$\Pr_{y\in\mathcal{D}}[N(y) \wedge \underline{m-1}^{\mathcal{W}} < y] = \Pr_{y\in\mathcal{D}}[N(y) \wedge \underline{m}^{\mathcal{W}} < y] + \mathcal{D}(\underline{m}^{\mathcal{W}}) \qquad \text{and}$$
$$\mathcal{D}(\underline{m}^{\mathcal{W}}) = \Pr_{y\in\mathcal{D}}[N(y) \wedge \underline{m}^{\mathcal{W}} < y] \qquad \text{by } (\star),$$

which, along with

$$\begin{aligned} & 1/2 - 2^{-m} + \Pr_{y\in\mathcal{D}}[N(y) \wedge \underline{m-1}^{\mathcal{W}} < y] \\ = \; & \mathcal{D}(\underline{\min}) + \sum_{n=2}^{m-1} \mathcal{D}(\underline{n}^{\mathcal{W}}) + \Pr_{y\in\mathcal{D}}[N(y) \wedge \underline{m-1}^{\mathcal{W}} < y] \\ = \; & \Pr_{y\in\mathcal{D}}[N(y)] = 1/2, \end{aligned}$$

imply $\Pr_{y\in\mathcal{D}}[N(y) \wedge \underline{m-1}^{\mathcal{W}} < y] = 2^{-m}$ and thus $\mathcal{D}(\underline{m}^{\mathcal{W}}) = 2^{-m-1}$.

Now assume for the sake of contradiction that $\mathcal{D}(\underline{\max}^{\mathcal{W}}) = 0$. Then $\underline{m}^{\mathcal{W}} = \underline{\max}^{\mathcal{W}}$ for no finite m. This would mean that the m-fold successor of $\underline{\min}^{\mathcal{W}}$ exists for all finite m. But $N_0^{\mathcal{W}}$ is finite, so this cannot be true. Therefore $\underline{\max}^{\mathcal{W}}$ must have positive measure, as desired. ■

We have thus shown that $N_0^{\mathcal{W}}$ is a linear chain ordered by $<^{\mathcal{W}}$, with minimal element $\underline{\min}^{\mathcal{W}}$ and maximal element $\underline{\max}^{\mathcal{W}}$. This structure allows us to interpret the sentences in T_0 encoding the Turing machine M classically.

Indeed, we can construct the computation history of M as follows:

At time t,

— the tape has a 1 at position p iff $T(p,t)$ holds,

— the head of M is above cell p iff $H(p,t)$ holds, and

— the state of M is q iff $S_q(t)$ holds.

Now, aided by the verbal translation provided in the description of T_0, we can verify that

- Initially, M is in state q_0, the head is in the first position, and the tape has all 0s (clauses (4) of T_0).
- At any time, M is in one state and one state only (clause (5) of T_0).
- The tape and M's head position and state are updated correctly according to δ (clauses (6) of T_0). In particular, all cells not specified by the update rule have the same symbol after the update (clauses (6(b)iv)).
- At time $\underline{\max}^{\mathcal{W}}$, M is in either an accepting or rejecting state (clause (7) of T_0).

Hence, the $\frac{1}{2}$E-satisfaction of Ψ implies M halts.

□

Finally, to complete our proof that finite ϵE-satisfiability is Σ_1^0-complete, we prove

Theorem 3.4.6. *For both $X = E$ and $X = F$, finite ϵX-satisfiability is Σ_1^0-definable for rational $\epsilon \in (0,1)$ and any first order language.*

Before the proof, we will need the following perturbation results, which allow us to "shake up" the probability measures of each ϵ-model into a nicer form.

Lemma 3.4.7. *Suppose $(M, \operatorname{dom}\mathcal{D}, \mathcal{D})$ is a measure space such that M is finite and $\mathcal{D}$ is defined on all subsets of M. Then for any $\delta > 0$, there exists a measure $\mathcal{D}'$ defined on all subsets of M such that*

- *all values of $\mathcal{D}'$ are rational,*
- *$\mathcal{D}(S) = \mathcal{D}'(S)$ whenever $\mathcal{D}(S)$ is rational and positive,*
- *$\max_{S\subseteq M} |\mathcal{D}(S) - \mathcal{D}'(S)| < \delta$, and*
- *$\mathcal{D}(S) > 0$ for all $S \subseteq M$*

Proof. WLOG let $M = \{1, 2, \dots, m\}$. Then $\mathcal{D}$ is uniquely determined by its values on $i \in M$. Let $\mathbf{p} = \langle p_i\rangle_{i=1}^m$ represent this vector. Thus $\mathcal{D}(S) = \mathbf{v}_S \cdot \mathbf{p}$ where $\mathbf{v}_S$ is the vector whose value at position i is 1 if $i \in S$ and 0 otherwise.

Let $\langle S_j\rangle_{j=1}^k$ be an enumeration of all $S \subseteq M$ such that $\mathcal{D}(S)$ is rational and positive. We can then form the matrix $\mathbf{R}$ with row vectors $\mathbf{v}_{S_j}$ and the column vector $\mathbf{q} = \langle \mathcal{D}(S_j)\rangle_{j=1}^k$. Immediately, we have

$$\mathbf{Rp} = \mathbf{q}.$$

Since all entries of $\mathbf{R}$ and $\mathbf{q}$ are rational, by Gaussian elimination, we can reduce the associated matrix $\mathbf{R}|\mathbf{q}$ to row echelon form $\mathbf{R}'|\mathbf{q}'$ with all rational entries. It's then clear that we can perturb each value of $\mathbf{p}$ by less than $\delta/|M|$ to get $\mathbf{p}'$ such that 1) each entry of $\mathbf{p}'$ is positive and rational, and 2) $\mathbf{R}'\mathbf{p}' = \mathbf{q}'$ and thus $\mathbf{Rp}' = \mathbf{q}$ and all positive rational values of $\mathcal{D}$ are unaffected. Extending the point measure $\mathbf{p}'$ linearly to a measure over all subsets of M gives the desired result. □

Lemma 3.4.8. *Let $\epsilon \in (0, 1)$ be rational. If $(\mathcal{M}, \mathcal{D})$ is a finite ϵ-model, then for some measure $\mathcal{D}'$ with $\operatorname{dom}\mathcal{D}' = \mathfrak{P}(M)$ such that $\mathcal{D}'(x)$ is rational and positive for all $x \in M$, $(\mathcal{M}, \mathcal{D}')$ is ϵ-elementarily equivalent to $(\mathcal{M}, \mathcal{D})$.*

Proof. By proposition (2.3.4) we may assume that $\mathcal{D}$ is defined on all subsets of M. Using lemma (3.4.7) with

$$\delta = \frac{1}{2}\min\left(|1-\epsilon|, \min_{\substack{S\subseteq M\\ \mathcal{D}(S)\notin\mathbb{Q}}} |\mathcal{D}(S) - (1-\epsilon)|\right) > 0,$$

there is a measure $\mathcal{D}'$ defined on all subsets of M such that $\mathcal{D}'$ has all rational and positive values, $\mathcal{D}'$ differs from $\mathcal{D}$ only on sets of irrational $\mathcal{D}$-measure, and this difference is uniformly bounded by δ. In particular, $\mathcal{D}(S) \geq 1-\epsilon \iff \mathcal{D}'(S) \geq 1-\epsilon$, so (by an easy induction argument) $(\mathcal{M}, \mathcal{D})$ is ϵ-elementary equivalent to $(\mathcal{M}, \mathcal{D}')$. Therefore $(\mathcal{M}, \mathcal{D}') \models_\epsilon \phi$. □

For the following proof, we do not actually need $\mathcal{D}'(x)$ to be positive, but this lemma provides an alternative justification for the assumption in the proof of (3.2.11).

Proof of thm (3.4.6). Let $\langle \mathbf{t}^i\rangle_{i\geq 1}$ be an effective enumeration of all finite sequences of ω (for example by Gödel's β function). For each finite sequence $\mathbf{t}^i$ let $|\mathbf{t}^i|$ denote the length of the sequence, let $\mathbf{t}^i_j$ denote the jth element of $\mathbf{t}^i$, for $1 \leq j \leq |\mathbf{t}^i|$, and let $\|\mathbf{t}^i\|$ denote the sum of its elements $\sum_{j=1}^{|\mathbf{t}^i|} \mathbf{t}^i_j$.

Let σ be the vocabulary used in ϕ. To test whether a sentence ϕ is ϵE-satisfiable by a finite model, we inspect $\mathbf{t}^i$ in sequence for $i = 1, 2, \dots$. For each $\mathbf{t}^i$ we form all classical models with signature σ on $|\mathbf{t}^i|$ elements $\{1, 2, \dots, |\mathbf{t}^i|\}$. There are only a finite number of them since σ is finite. We turn these classical models into ϵ-models by imbuing them with the measure $\mathcal{D}$ defined by $\mathcal{D}(j) = \mathbf{t}^i_j/\|\mathbf{t}^i\|$. We can then mechanically check whether any of them satisfy ϕ.

If $(\mathcal{M}, \mathcal{D}) \models_\epsilon \phi$ for some finite ϵ-model $(\mathcal{M}, \mathcal{D})$, then WLOG we can assume $M = \{1, 2, \ldots, m\}$ for some m, $\mathcal{D}$ to be defined on all subsets of $\mathcal{M}$ (by proposition (2.3.4)), and $\mathcal{D}$ to have all rational values (by lemma (3.4.8)). Thus there exists an integer r such that for each $1 \leq i \leq m$, $\mathcal{D}(i) = r_i/r$ and r_i is an integer. Our algorithm described above will then terminate before $k+1$ outer loops, where $\mathbf{t}^k = \langle r_i \rangle_{i=1}^m$.

This shows that the ϵE-satisfiability problem for rational ϵ is Σ_1^0-definable. The ϵF-satisfiability case is resolved identically. □

Combined with theorem (3.4.4), this result shows

Corollary 3.4.9. *Let $\mathcal{L}$ be a countable first order language with an infinite number of unary relations and at least three binary relations. For any rational $\epsilon \in (0,1)$, finite ϵE-satisfiability is Σ_1^0-complete. Equivalently, finite ϵF-validity is Π_1^0-complete.*

3.5 Finite and Countable ϵE-validity

Like the unrestricted case, the set of finitely (resp. countably) 0E-valid sentences also coincides with the set of finitely (resp. countably) classically valid sentences. With Kuyper's inter-reduction result for ϵE-validities and the Σ_1^0-definability derived from the last section, we can characterize ϵE-validity over finite models precisely as Π_1^0-complete whenever $\epsilon \in \mathbb{Q}$.

Theorem 3.5.1. *For any countable first order language, the set of finitely 0E-valid sentences is exactly the set of finitely classically valid sentences. The set of countably 0E-valid sentences is exactly the set of classically valid sentences.*

Proof. Obviously every finitely classically valid sentence is a finitely 0E-valid sentence.

Now suppose ϕ is finitely 0E-valid. Then for any n, $(\mathcal{M}, \mathcal{D}) \models_0 \phi$ for all classical models $\mathcal{M}$ of size n and $\mathcal{D}$ the uniform distribution. But in such 0-models, $\forall$ has the same interpretation as classically. Hence all finite classical models satisfy ϕ, as desired.

The proof works the same for countable validities, except that for countably infinite models, we instead ascribe the exponential distribution $\mathcal{D}(n) = \frac{1}{2^n}$. □

Immediately,

Corollary 3.5.2. *Let $\mathcal{L}$ be any first order language and $\mathcal{S}$ be a set of sentences in $\mathcal{L}$. The following are equivalent:*

- *The set of finitely (resp. unrestricted) classically valid sentences in $\mathcal{S}$ is decidable.*
- *The set of finitely (resp. countably) 0E-valid sentences in $\mathcal{S}$ is decidable.*

In any first order language with at least one binary relation, the set of finitely classically valid sentences is Π_1^0-complete [15, p. 166]. Therefore,

Corollary 3.5.3. *In any first order language with at least one binary relation, the problem of determining whether a sentence is finitely 0E-valid is Π_1^0-complete.*

Likewise, as classical validity in any language with at least one binary relation is Σ_1^0-complete [24], we have in the countable case

Corollary 3.5.4. *In any first order language with at least one binary relation, the problem of determining whether a sentence is countably 0E-valid is Σ_1^0-complete.*

There exist computable reductions for ϵE-validity just like in the case of ϵE-satisfiability (see proposition (3.4.1)):

Proposition 3.5.5 (Kuyper inter-reduction [12]). *Let*

- $\mathcal{L}$ *be a countable first-order language not containing function symbols or equality, and*
- $\mathcal{L}'$ *be the language obtained by adding an infinite number of unary predicates to* $\mathcal{L}$.

Then, for all rational $0 \leq \epsilon_0 \leq \epsilon_1 < 1$, *the set of normally* ϵ_0*E-valid* $\mathcal{L}$*-sentences many-one reduces via a computable function to the set of normally* ϵ_1*E-valid* $\mathcal{L}$*-sentences.*

More generally [5], *this reduction works "per quantifier": For any* $\epsilon \in (0,1) \cap \mathbb{Q}$, *there exists a computable function* $\mathcal{F}_\epsilon$ *mapping qF-sentences in* $\mathcal{L}$ *to* $\mathcal{L}'$*-sentences such that the following are equivalent:*

1. *there exists a pair* $(\mathcal{M}, \mathcal{D})$ *such that*

$$(\mathcal{M}, \mathcal{D}) \models^{q} \Phi.$$

2. $\mathcal{F}_\epsilon(\Phi)$ *is* ϵ*F-satisfiable.*

Again, the proof for this theorem and the construction of the reduction function given in [12] carry over almost identically when we restrict our attention from "normally ϵE-valid" (validity over all probability models) to "finitely ϵE-valid" (validity over all finite probability models — which is equivalent to the validity over all finite models by (2.3.4)): the method of proof is the duplication of a given model a finite number of times, and this procedure preserves finiteness of ϵ-models. This observation remains true in considering countable ϵ-models.

Therefore, we have, by duality, the following two ϵF-analogues of (3.4.2) and (3.4.3).

Lemma 3.5.6. *Let* $\mathcal{L}$ *and* $\mathcal{L}'$ *be defined as above. For any* $\epsilon \in (0,1) \cap \mathbb{Q}$, *there exists a computable function* $\mathcal{F}_\epsilon$ *mapping qF-sentences in* $\mathcal{L}$ *to* $\mathcal{L}'$*-sentences such that the following are equivalent:*

1. *there exists a pair* $(\mathcal{M}, \mathcal{D})$ *with* M *finite such that*

$$(\mathcal{M}, \mathcal{D}) \models^{q} \Phi.$$

2. $\mathcal{F}_\epsilon(\Phi)$ *is finitely* ϵ*F-satisfiable.*

Lemma 3.5.7. *Let* $\mathcal{L}$ *and* $\mathcal{L}'$ *be defined as above and fix rational* $\epsilon \in (0,1)$. *There is a computable function* f_ϵ *such that, for any finite set of rationals* $J \subseteq \mathbb{Q} \cap [0,1]$ *and* $\mathcal{L}$*-sentences* $\{\Psi_\alpha\}_{\alpha \in J}$, *the following are equivalent:*

1. *there exists a pair* $(\mathcal{M}, \mathcal{D})$ *such that, for each* $\alpha \in J$, $(\mathcal{M}, \mathcal{D})$ *is a finite* α*-model and*

$$(\mathcal{M}, \mathcal{D}) \vdash_\alpha \Psi_\alpha.$$

2. $f_\epsilon(\{\Psi_\alpha\}_{\alpha \in J})$ *is finitely* ϵ*F-satisfiable.*

These two lemmas were used in the example section to express sentences over different error parameters.

In (3.5.5), letting $\epsilon_0 = 0$, we obtain a computable reduction from finite 0E-validity to finite ϵE-validity for any $\epsilon \in (0,1)$. By corollary (3.5.3), we get

Corollary 3.5.8. *For any language with an infinite number of unary predicates and at least one binary predicate, finite* ϵ*E-validity is* Π^0_1*-hard for rational* $\epsilon \in (0,1)$.

[5] see [12, remark below thm 3.3].

By theorem (3.4.6) for the case of X = F, finite ϵF-satisfiability is Σ_1^0-definable for any language. By duality, finite ϵE-validity is Π_1^0-definable. Hence, in combination with the above corollary, this implies

Theorem 3.5.9. *For any language with an infinite number of unary predicates and at least one binary predicate, finite ϵE-validity is Π_1^0-complete for rational $\epsilon \in (0, 1)$.*

Along the same lines, for the countable case, we have

Corollary 3.5.10. *For any language with an infinite number of unary predicates and at least one binary predicate, countable ϵE-validity is Σ_1^0-hard for rational $\epsilon \in (0, 1)$.*

Finally, we mention a theorem of Terwijn.

Proposition 3.5.11 (Terwijn [23])**.** *Let ϕ be a sentence. ϕ is finitely classically valid iff ϕ is countably ϵE-valid for every $\epsilon > 0$.*

4 Future Work

4.1 The Countable Case

As displayed by table (3), we still do not know much about ϵE-logic over countable ϵ-models. Most egregiously we have no idea of the computability of its satisfiability problem. Looking over the entries of tables (1), (2), and (3), the pattern seems to favor the possibility of countable ϵE-satisfiabilty being Π_1^0-hard or even complete. Obviously our proof for the finite case would not carry over, but it is conceivable that replacing the halting set with a Π_1^0-complete set would work out naturally. Alternatively, we could look at the dual problem of reducing the halting set to countable ϵF-validity.

4.2 Reducing Language Requirement

In our results, the requirements of an infinite number of unary predicates and at least three binary predicates are likely not optimal. In classical first order logic, these requirements can be collapsed to the single requirement of one binary predicate through graph theoretic or set theoretic encodings. However, in ϵE- and ϵF- logic, these methods do not seem to play well with the additional structure of a probability space. In any case, for our theorems to be more relevant to applications, the number of unary predicates must be brought down to a finite number, whether strengthening our undecidability or breaking into decidability.

4.3 q-Logic and Trees

We developed q-sentences and other q-concepts only to arrive at results for ϵE- and ϵF-logic, but they can as well be studied on their own. In particular, an obvious definition of q-logic would make it a stronger version of Keisler's probability logic which only allows the quantifiers $\mathfrak{Q}_\epsilon^{\geq}$ and $\mathfrak{Q}_\epsilon^{>}$. Keisler's work [7] can then be applied in most aspects to such a q-logic.

Similarly, one could investigate the algebraic structure of ϵE-, ϵF-, and q-trees, whose properties we have not fully exploited. It should not be hard to see that, for a fixed ϵ-model $(\mathcal{M}, \mathcal{D})$ and a fixed (q-)sentence Φ, there is a natural partial order and a join operation on trees $(\mathcal{M}, \mathcal{D})$ for Φ of each class. Instinctively one could ask, under what circumstance does a meet operation exists? Deeper research into the semilattice structure of these trees could reveal information on the computability of ϵE and ϵF fragments not discussed here.

4.4 Irrational ϵ

There are several insufficiencies in current techniques with regard to deducing facts about irrational ϵs:

(i) Terwijn and Kuyper's proofs of the inter-reduction theorems fundamentally require the ratio ϵ_0/ϵ_1 to be rational. So while this would imply we have inter-reductions between, say $\frac{1}{\sqrt{2}}$ and $\frac{1}{2\sqrt{2}}$, we cannot say much about the relative difficulties of $\frac{1}{2}$ and $\frac{1}{\sqrt{2}}$. Surpassing the obstacles to generalize to irrational ϵ would necessitate brand new methods, which could also eliminate the infinite unary predicate restriction.

(ii) Our proof of the Σ^0_1-hardness of finite ϵE-satisfiability depends crucially on the ability of $\frac{1}{2}$E-logic to force the measures of two sets to be equal. Without a reduction from $\frac{1}{2}$E- to ϵE-validity, we cannot conclude that ϵE-validity is Σ^0_1-hard.

(iii) In order for a linear program to be solved in finite time, all arithmetic operations over the field generated by its coefficients must be total computable. This holds for the rational and general number fields, but not for the field of computable reals, for which the comparison function is not total. Thus our proof of decidabilities in monadic relational languages do not carry over to the general case.

Fortunately, as our example section illustrate, in many cases only the relative magnitude of ϵ matters. This observation also boosts the likelihood that our computability results hold for general ϵ as well.

4.5 Classical Model Theory Techniques

Given the applications in section (2.4) and the initial motivation of ϵE-logic, we see that it has an intimate connection with computational learning theory. At the intersection of CLT and classical model theory is the concept of VC dimension [6][14][1][2], which we mentioned briefly. It could be possible to reconcile the results of these two disciplines in ϵE- and ϵF-logics, to the benefit of all involved.

As our examples also hinted, many techniques in finite model theory could be converted to versions for our probability logics. One could also experiment with adding new means of expressions like the BIT relation, a canonical ordering, counting operators, etc. An analogue of descriptive complexity could be developed; given the abundance of probabilistic quantifiers in classes like BPP, PP, PCP, and so on, it is indeed plausible that one could equate one of these complexity classes with a description class in ϵE-logic.

A version of Ehrenfeucht-Fraïssé games [15], another common tool in finite model theory, for ϵE-logic could also have connections to the malicious advisary learning model [8][20].

4.6 Computational Complexity

In addition to developing descriptive complexity, one could also explore typical[6] -case complexity in ϵE-logic.

5 Acknowledgements

I owe the inspiration for this topic to a conversation with Leslie Valiant, who provided me with a copy of his work *Robust Logics* [26]. His class CS228 at Harvard cultured my appreciation of

[6] to be distinguished from *average-case* complexity.

computational learning theory and equipped me with the knowledge for much of the material in the application section. The first draft of this essay was born as the final project for the course.

I thank Rutger Kuyper for his enthusiastic assistance throughout my research despite being an ocean away. My specific interest in ϵ-logic started when I looked for derivative works of Valiant's *Robust Logics* and discovered Kuyper's thesis. He helped gather for me all current materials on ϵ-logic and answered my questions readily, making what could be a rocky journey very smooth.

Finally, and most importantly, I am indebted to Nate Ackerman for the intellectual conversations, mathematical guidance, and detailed comments on life, liberty, and an early draft of this paper. Without him, I would not have seen half of the mathematical world as I have now; without him, I would not have anywhere to echo my love and amazement of this magical world; and without him, I would not be able to amuse myself with his Ackermanic similes.

References

[1] M. Aschenbrenner, A. Dolich, D. Haskell, D. Macpherson, and S. Starchenko. Vapnik-Chervonenkis density in some theories without the independence property, I. *ArXiv e-prints*, September 2011.

[2] M. Aschenbrenner, A. Dolich, D. Haskell, D. Macpherson, and S. Starchenko. Vapnik-Chervonenkis density in some theories without the independence property, II. *Notre Dame Journal of Formal Logic*, 54(3-4):311–363, 2013.

[3] Garrett Birkhoff. *Lattice Theory*. American Mathematical Society, revised ed. edition, 1948.

[4] Vladimir I. Bogachev. *Measure Theory*. Springer, 2007.

[5] Michael J. Kearns, Robert E. Schapire, and Linda M. Sellie. Toward efficient agnostic learning. *Machine Learning*, 17(2-3):115–141, 1994.

[6] Michael J. Kearns and Umesh Virkumar Vazirani. *An Introduction to Computational Learning Theory*. MIT Press, Jan 1994.

[7] H. J. Keisler. *Chapter XIV: Probability Quantifiers*, volume 8 of *Perspectives in Mathematical Logic*, pages 507–556. Springer-Verlag, New York, 1985.

[8] Adam R. Klivans, Philip M. Long, and Rocco A. Servedio. Learning halfspaces with malicious noise. *J. Mach. Learn. Res.*, 10:2715–2740, December 2009.

[9] Adam R. Klivans and Rocco A. Servedio. Toward attribute efficient learning of decision lists and parities. In John Shawe-Taylor and Yoram Singer, editors, *Learning Theory*, volume 3120 of *Lecture Notes in Computer Science*, pages 224–238. Springer Berlin Heidelberg, 2004.

[10] Rutger Kuyper. Computational aspects of satisfiability in probability logic. *to appear in Mathematical Logic Quarterly*.

[11] Rutger Kuyper. Probability logic. Master's thesis, Radboud University Nijmegen, 2011.

[12] Rutger Kuyper. Computational hardness of validity in probability logic. In Sergei Artemov and Anil Nerode, editors, *Logical Foundations of Computer Science*, volume 7734 of *Lecture Notes in Computer Science*, pages 252–265. Springer Berlin Heidelberg, 2013.

[13] Rutger Kuyper and Sebastiaan A. Terwijn. Model theory of measure spaces and probability logic. *The Review of Symbolic Logic*, 6:367–393, 9 2013.

[14] Michael C. Laskowski. Vapnik-chervonenkis classes of definable sets. *Journal of the London Mathematical Society*, s2-45(2):377–384, 1992.

[15] Leonid Libkin. *Elements of Finite Model Theory.* Springer, 2004.

[16] David J. C. MacKay. *Information Theory, Inference, and Learning Algorithms.* Cambridge University Press, 2003.

[17] Nimrod Megiddo. On the complexity of linear programming. In Truman Fassett Bewley, editor, *Advances in Economic Theory*, pages 225–268. Cambridge University Press, 1987. Cambridge Books Online.

[18] Stuart J. Russel and Peter Norvig. *Artificial Intelligence: A Modern Approach.* Prentice Hall, 2010.

[19] Alexander Schrijver. *Theory of Linear and Integer Programming.* Wiley, 1998.

[20] Rocco A. Servedio. Smooth boosting and learning with malicious noise. *J. Mach. Learn. Res.*, 4:633–648, December 2003.

[21] Robert I. Soare. *Recursively Enumerable Sets and Degrees.* Springer, 1987.

[22] Sebastiaan A. Terwijn. Probabilistic logic and induction. *Journal of Logic and Computation*, 15(4):507–515, 2005.

[23] Sebastiaan A. Terwijn. Decidability and undecidability in probability logic. In Sergei Artemov and Anil Nerode, editors, *Logical Foundations of Computer Science*, volume 5407 of *Lecture Notes in Computer Science*, pages 441–450. Springer Berlin Heidelberg, 2009.

[24] A. M. Turing. On computable numbers, with an application to the entscheidungsproblem. *Proceedings of the London Mathematical Society*, s2-42(1):230–265, 1937.

[25] L. G. Valiant. A theory of the learnable. *Commun. ACM*, 27(11):1134–1142, November 1984.

[26] Leslie G. Valiant. Robust logics. In *Proceedings of the Thirty-first Annual ACM Symposium on Theory of Computing*, STOC '99, pages 642–651, New York, NY, USA, 1999. ACM.