PARACONSISTENT PROPOSITIONAL INFERENCE
USING RESTRICTED BOLTZMANN MACHINES

by

RYAN T. MCARDLE

(Under the Direction of O. Bradley Bassler)

ABSTRACT

We explore a method proposed in the literature for encoding formulas of propositional logic and identifying models of the formula using Restricted Boltzmann Machines. We find that the method preserves several central properties of classical propositional logic. We extend the formalism to represent the paraconsistent three-valued logics Kleene's Strong Logic of Indeterminacy and Priest's Logic of Paradox and find that this encoding is also faithful to their original presentations. Given the loss of desirable deductive properties when one embraces the Logic of Paradox, we further extend the formalism to encode Priest's Minimally Inconsistent Logic of Paradox, which recovers essential properties that allow one to use the method for paraconsistent inference. Given the success of each of these extensions, we conclude that the formalism for encoding propositional logics shows robust promise for encoding arbitrary logics, suggesting that next steps should investigate extensions to predicate and modal logics.

INDEX WORDS: [Artificial Intelligence, Paraconsistent Logic, Restricted Boltzmann Machine, Multivalued Logic, Artificial Neural Networks, Logic Programming, Variational Principle, Unsupervised Learning]

PARACONSISTENT PROPOSITIONAL INFERENCE USING RESTRICTED BOLTZMANN MACHINES

by

RYAN T. MCARDLE

B.S., University of Georgia, 2018
B.A., University of Georgia, 2018

A Thesis Submitted to the Graduate Faculty of the
University of Georgia in Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE

ATHENS, GEORGIA

2021

PARACONSISTENT PROPOSITIONAL INFERENCE USING RESTRICTED BOLTZMANN MACHINES

by

RYAN T. MCARDLE

Major Professor: O. Bradley Bassler

Committee: Frederick Maier
Sarah Wright

Electronic Version Approved:

Ron Walcott
Dean of the Graduate School
The University of Georgia
May 2021

DEDICATION

For Banksy and Aurora, so that you can eat all the house plants and sticks that your hearts desire.

ACKNOWLEDGMENTS

I'd first like to thank the members of this project's committee, without whom it never would have become a reality. Dr. Sarah Wright was my first philosophy professor at The University of Georgia and sparked interests that ultimately drove my study of artificial intelligence. This project was conceived as a semester project assigned by Dr. Fred Maier, who directed me towards the concept of neuralsymbolic computing and the need for a bridge between connectionist and symbolic representations of knowledge. Finally, Dr. Brad Bassler's years of instruction and mentorship prepared me to complete the necessary work for this project, and his numerous editorial comments have improved the final product significantly. Thank you all for your contributions as instructors, advisors, and friends.

I'd also like to thank all of the friends and family who have been a support system that helped me truly enjoy my time at The University of Georgia. You've provided the critical balance necessary to keep me sane through this whole process, and I appreciate all of your support and friendship over the years. I couldn't have done it without you.

TABLE OF CONTENTS

LIST OF TABLES

CHAPTER 1

INTRODUCTION

§ 1.1 Introduction

The current zeitgeist in artificial intelligence is one dominated by the application of artificial neural networks (ANNs) to solve a wide range of problems. The amount of data and large-scale parallel processing power recently made widely available makes the training of these networks quite efficient compared to any attempts of previous decades. However, their proliferation has brought to the forefront many concerns regarding our ability to understand and ultimately trust these networks that we so often employ in our decision making processes. These networks, due to their complicated structure modeling a massively high-dimensional space, are quite opaque to human interpretation and audit.

There have been advancements made in recent years to address this issue, however correcting our lack of understanding regarding a deep network's inner workings is still a major concern for the field [5], [10]. In general, it is quite difficult for a human to interpret how a trained ANN processes the provided data in the way that it does, or to construct one that will process data via some intended methodology. This can make it difficult to understand what metrics the ANN might be using in its decision process and, when trained on historical data that has been shown to be discriminatory, the ANN will simply replicate these human metrics that can have major impacts on people's lives [2]. As the field continues to apply ANNs, we must develop methods which allow us to faithfully and efficiently audit our ANNs to ensure that their operation remains both under our control and within our approval. It appears that such an approach may be feasible within Restricted Boltzmann Machines (RBMs) and their deep learning counterpart Deep Belief Networks (DBNs).

§ 1.2 Background and Related Works

§ 1.2.1 Restricted Boltzmann Machines

An RBM is a statistical, energy based ANN architecture which, following an unsupervised training process, represents a joint probability distribution over the training data. This distribution can then be used to infer likely values for data that are missing or undefined in the training set [4].

The graphical structure of a Boltzmann Machine consists of two layers of nodes, a "visible" layer and a "hidden" layer. In the Restricted Boltzmann Machine, nodes contained in one layer can be connected only with the nodes contained in the other layer, and all connections between nodes are bidirectional [4]. This restriction offers a significant advantage with regards to the overall computational complexity of the structure, while still allowing one to create faithful models of the input data [16].

We can formally define an RBM in the following way:

Definition 1.2.1. *A Restricted Boltzmann Machine (RBM) is a quadruple $N = (X, H, W, E)$, where X is a set of n visible nodes taking values $\boldsymbol{x}$, H is a set of m hidden nodes taking values $\boldsymbol{h}$, W is the set of $n \times m$ connection weights between each visible node x_i and each hidden node h_j, and E is the energy function:*

$$E(\boldsymbol{x}, \boldsymbol{h}) = -\sum_{i,j} w_{ij} x_i h_j - \sum_i a_i x_i - \sum_j b_j h_j, \tag{1.2.1}$$

where a_i and b_j are the biases of visible and hidden nodes x_i and h_j respectively, and w_{ij} is the connection weight between nodes x_i and h_j.

In general, $\boldsymbol{x} \in \mathbb{R}^n$ and $\boldsymbol{h} \in \mathbb{R}^m$, and these values are often scaled into the range $[0, 1]$. However, for the purposes of this thesis, we constrain ourselves to the binary case for which $\boldsymbol{x} \in \{0, 1\}^n$ and $\boldsymbol{h} \in \{0, 1\}^m$.

From this energy function, we can determine the joint probability distribution of assignment $(\boldsymbol{x}, \boldsymbol{h})$:

$$p(\boldsymbol{x}, \boldsymbol{h}) = \frac{1}{Z} e^{-\frac{1}{\tau} E(\boldsymbol{x}, \boldsymbol{h})} \tag{1.2.2}$$

where

$$Z = \sum_{\boldsymbol{x},\boldsymbol{h}} e^{-\frac{1}{\tau}E(\boldsymbol{x},\boldsymbol{h})} \tag{1.2.3}$$

represents the partition function over all distributions, with 'temperature' τ, which is gradually reduced during the training process in order to minimize the value of the energy function [16][13].

This partition function in general can be quite expensive to compute explicitly. However, given any initial state of the nodes, the minimized energy function of an RBM can be approximated arbitrarily well through a tractable Markov chain Monte Carlo process of alternatively resampling the values of both hidden and visible nodes in an iterative process known as Gibbs Sampling [3]. As real-valued RBMs and the necessary computational methods for minimizing intractible partition functions are outside of the scope of this work, the reader is directed towards [4] and [3] for an introduction to the Gibbs Sampling process and its use in training RBMs.

A standard use of the RBM structure interprets visible nodes as corresponding with training data while the hidden layer would express a level of abstraction corresponding to some shared feature of elements of the input data. Once trained, the architecture will represent a joint probability distribution over a potentially incomplete data set and can then fill in the data set by sampling from the probability distribution [4].

This application has gained attention in recent years when used with the Netflix data set to predict user's movie ratings better than Netflix's own algorithm [12]. This application created visible nodes whose values represented a user's ratings for movies, both known and unknown, and hidden nodes which would represent hidden features shared by movies (inclusion in certain genres, sharing directors or actors, etc.). The algorithm could then predict the user's ratings for unrated movies, filling in the values of missing data points in the input set.

§ 1.2.2 Deep Belief Networks

A Deep Belief Net (DBN) is a neural network architecture in which multiple RBMs are stacked onto one another such that the hidden nodes of one RBM act as the visible nodes of the next. While DBNs do suffer from being more computationally complex and their conditional probabilities may be more

difficult to compute exactly, they benefit from being able to leverage the multiple levels of RBMs in order to model the data at higher levels of abstraction [14]. This makes this structure particularly effective and robust in classification problems, in which it can leverage ontological classification information which can even be transfered between networks [7], [17]. This ontological approach allows for great modularity in the application of these networks and an ability to benefit from prior training for a range of problems, rather than needing to fully retrain each time the problem is adjusted. It is therefore the author's hope that a proof of concept for a faithful representation of propositional logic in a single RBM layer could prompt work with deeper networks, potentially representing logics with a higher potential for abstraction and expression.

§ 1.2.3 Knowledge Representation in RBMs and DBNs

It has been shown in [16] that a knowledge base expressed in propositional logic, when decomposed into a 'Strict Disjunctive Normal Form' (SDNF) in which at most one conjunctive clause holds given an assignment, can be associated with an RBM whose visible nodes correspond to the literals of the knowledge base, whose hidden nodes correspond to the clauses of the SDNF, and whose energy function is determined by the SDNF. The states of the RBM which minimze the energy function are shown to correspond to valuations which will satisfy the knowledge base if it is consistent or provide a maximum satisfiability in the case of weighted logics. It has further been shown that this process can be implemented in reverse in a DBN, isolating a single RBM layer and reverse engineering a logical expression which represents the nodal relationships in said layer and allowing for a process of knowledge extraction from the DBN [14].

The author of [16] points out that while SDNF is more complex and demanding to compute than the normal DNF, which is already quite expensive, this process is efficient for logical implications. As many knowledge bases are already presented in the form of facts and logical implication rules, there is promise that this method could be tractable for real-world knowledge bases. Further, the number of nodes in the RBM associated with a logical implication grows linearly with the number of literals in the implication, so the entire process, including conversion to SDNF, representation as RBM, and Gibbs Sampling to train the RBM, should be tractably efficient in the case of real-world knowledge bases.

§ 1.3 Summary of This Work

In this work, we explore and expand the method for representing propositional statements in RBMs referenced in Sect. 1.2.3.

In Chapter 2, we reproduce the central theorems that allow for the conversion of a sentence into an RBM. We then explore a number of logical properties relevant for deduction in order to ensure that this formalism remains faithful to these properties.

In Chapter 3, we expand the original formalism to encode the three-valued logics Kleene's Strong Logic of Indeterminacy, *K3* , and Priest's Logic of Paradox, *LP* . We again analyze the logical properties and find that they hold as expected.

Given the loss of many desirable deductive properties when one embraces *LP* , in Chapter 4, we further expand the formalism to represent Priest's Minimally Inconsistent Logic of Paradox, LP_m . This extension's restoration of the deductives properties enables one to perform paraconsistent inference within a statistical connectionist network.

Table 1.1 briefly summarizes the properties which are explored and their status in each of the relevant logics, both as we expect them to hold from their original presentation and as they hold under the RBM formalism.[1]

Table 1.1: The status of each property explored in this paper for each of the relevant logics.

Properties in the Relevant Logics								
Property	Propositional		*K3*		*LP*		LP_m	
	Expect	RBM	Expect	RBM	Expect	RBM	Expect	RBM
Transitivity	✓	✓	✓	✓	×	×	✓	✓
Ex Falso Quodlibet	✓	∼	✓	∼	×	×	×	×
Disjunctive Syllogism	✓	✓	✓	✓	×	×	✓	✓
Resolution	✓	✓	✓	✓	×	×	✓	✓
Resolution Refutation	✓	✓	✓	✓	×	×	✓	✓

[1] The marking '∼' indicates that the property holds in a degenerate but philosophically similar sense. For further discussion, see Sect. 2.4.2

CHAPTER 2

PROPOSITIONAL LOGIC

§ 2.1 Chapter Introduction

We reproduce in Sect.2.2 - 2.3 the important theorems and proofs of Tran [16] with minor edits, which will be the basis for our analysis of the RBM encoded propositional logic in Sect. 2.4. For readers who are prepared to accept these results on faith, they may simply read the definitions and theorems, omitting the proofs.

§ 2.2 Strict Disjunctive Normal Form

We establish some preliminary definitions.

Definition 2.2.1. *A conjunctive clause is a conjunction of literals, i.e., a sentence ψ such that:*

$$\psi = x_1 \wedge \ldots \wedge x_n \tag{2.2.1}$$

Definition 2.2.2. *A sentence ψ is in disjunctive normal form (DNF) if it is a disjunction of conjunctive clauses[11]:*

$$\psi = (x_a \wedge \ldots \wedge x_b) \vee \ldots \vee (x_y \wedge \ldots \wedge x_z)\,. \tag{2.2.2}$$

Definition 2.2.3.

- *A "strict DNF" (SDNF) is a DNF where at most one single conjunctive clause is True at a time.*
- *A "full DNF" is a DNF where each variable must appear at least once in every conjunctive clause.*

Tran claims that any propositional well-formed formula can be presented as a full DNF which is also an SDNF and further provides a proof of and process for converting a general logical implication into this form.

Theorem 2.2.4. *A logical implication* $y \leftarrow \bigwedge_{t \in S_T} x_t \wedge \bigwedge_{k \in S_K} \neg x_k$ *where* S_T*,* S_K *respectively are the sets of positive and negative propositions' indices, can be represented as an SDNF having the form:*

$$\left(y \wedge \bigwedge_{t \in S_T} x_t \wedge \bigwedge_{k \in S_K} \neg x_k \right) \vee \bigvee_{p \in S_T \cup S_K} \left(\bigwedge_{t \in S_T . \backslash p} x_t \wedge \bigwedge_{k \in S_K . \backslash p} \neg x_k \wedge x'_p \right)$$

where $S.\backslash p$ *denotes a set* S *where* p *has been removed, and* $x'_p \equiv \neg x_p$ *if* $p \in S_T$ *else* $x'_p \equiv x_p$*.*

Proof. The logical implication $y \leftarrow \bigwedge_{t \in S_T} x_t \wedge \bigwedge_{k \in S_K} \neg x_k$ can be transformed into the disjunction:

$$\left(y \wedge \bigwedge_{t \in S_T} x_t \wedge \bigwedge_{k \in S_K} \neg x_k \right) \vee \left(\bigvee_{t \in S_T} \neg x_t \vee \bigvee_{k \in S_K} x_k \right) \tag{2.2.3}$$

Here, the logical implication holds if and only if either the conjunctive clause holds or the disjunctive clause holds. Consider the disjunctive clause.

$$\gamma \equiv \bigvee_{t \in S_T} \neg x_t \wedge \bigvee_{k \in S_K} x_k \tag{2.2.4}$$

This can be represented as $\gamma \equiv \gamma' \vee x'$, where x' can be either $\neg x_t$ or x_k for any $t \in S_T$ or $k \in S_K$, i.e. γ' is a disjunctive clause obtained by separating x' from γ. We can derive:

$$\gamma \equiv (\neg \gamma' \wedge x') \vee \gamma' \tag{2.2.5}$$

since $(\neg \gamma' \wedge x') \vee \gamma' \equiv (\gamma' \vee \neg \gamma') \wedge (\gamma' \vee x') \equiv \mathit{True} \wedge (\gamma' \vee x')$. With each repeated application of (2.2.5), we can move one variable out of a disjunctive clause and into a new conjunctive clause. We can further see that the disjunctive clause γ holds if and only if either the disjunctive clause γ' holds or the conjunctive clause $(\neg \gamma' \wedge x')$ holds. Therefore, at the end of the this process, the original disjunctive clause

γ is transformed into a union of conjunctive clauses[1]. Applying this process to our original disjunctive form of the logical implication in (2.2.3), we obtain the form:

$$\left(y \wedge \bigwedge_{t \in S_T} x_t \wedge \bigwedge_{k \in S_K} \neg x_k\right) \vee \bigvee_{p \in S_T \cup S_K} \left(\bigwedge_{t \in S_T . \backslash p} x_t \wedge \bigwedge_{k \in S_K . \backslash p} \neg x_k \wedge x'_p\right). \tag{2.2.6}$$

We see that 2.2.6 is in SDNF since the left-hand side of the disjunction is $True$ only in the case that both the antecedent and the consequent on the implication are $True$, and the right-hand side union will have only one $True$ clause for any valuation with makes the antecedent $False$ and the implication consequently $True$. ■

§ 2.3 Equivalence of an SDNF and an RBM

In order to represent some WFF φ as an RBM, we must first define what will be considered to be an equivalence between the two structures:

Definition 2.3.1. *A WFF φ is equivalent to an RBM $\mathcal{N}$ if and only if for any truth assignment over the visible nodes $\boldsymbol{x}$, $s_\varphi(\boldsymbol{x}) = -AE_{rank}(\boldsymbol{x}) + B$, where $s_\varphi(\boldsymbol{x}) \in \{0, 1\}$ is the truth value of φ given $\boldsymbol{x}$ with True $\equiv 1$ and False $\equiv 0$; $A > 0$ and B are constants; $E_{rank}(\boldsymbol{x}) = \min_{\boldsymbol{h}} E(\boldsymbol{x}, \boldsymbol{h})$ is the energy ranking function of $\mathcal{N}$ minimised over all hidden units.*

Definition 2.3.2. *We define preferred valuations of an RBM $\mathcal{N} = (X, H, W, E)$ to be any truth assignment $\boldsymbol{x}_0$ such that:*

$$\min_{\boldsymbol{h}} E(\boldsymbol{x}_0, \boldsymbol{h}) = \min_{\boldsymbol{x}, \boldsymbol{h}} E(\boldsymbol{x}, \boldsymbol{h}). \tag{2.3.7}$$

These valuations are those which minimize the energy function E over all node values.

We can rely on the existence of a preferred valuation for any RBM with fixed weights W and energy function E since $\min_{\boldsymbol{x}, \boldsymbol{h}} E(\boldsymbol{x}, \boldsymbol{h})$ only has a finite set of input states—both the number of nodes and the possible values for each of these nodes are finite. As such, we can identify some non-empty set of

[1] By adapting the definition of SDNF to logical clauses, one could say that γ has been transformed into SDNF.

input valuations $\boldsymbol{x_i}$ which result in the minimized value. This non-empty set is exactly the set of preferred valuations.

We will see when we prove Thm. 2.3.6 that we can identify a form for the energy function such that it is minimized to a value of 0.0 when $\boldsymbol{x}$ is not a model of the sentence φ and minimized to a chosen value $-\epsilon$ if and only if $\boldsymbol{x}$ is a model of the sentence φ.

Definition 2.3.3. *A truth assignment $\boldsymbol{x}$ is called a model of the sentence φ if and only if the truth value of φ given $\boldsymbol{x}$, $s_\varphi(\boldsymbol{x})$ is equal to one. We say that a model $\boldsymbol{x}$ of φ satisfies φ, or symbolically $\vDash_{\boldsymbol{x}} \varphi$.*

We will see that $A = \frac{1}{\epsilon}$ and $B = 0$ satisfy this mapping for the propositional case, i.e., models will map to 1 and non-models will map to 0. When we expand into the minimally inconsistent paraconsistent cases, we will employ the value of B to correct for the penalty imposed on inconsistent models.

We can naturally extend the idea of a model to a set of sentences, which we will call a *knowledge base*:

Definition 2.3.4. *A truth assignment $\boldsymbol{x}$ satisfies a set of sentences $\mathcal{K}$ if and only if $\boldsymbol{x}$ satisfies all $\varphi \in \mathcal{K}$.*

We further note that $\vDash_{\boldsymbol{x}} \mathcal{K}$ if and only if $\vDash_{\boldsymbol{x}} \bigwedge_{\varphi \in \mathcal{K}} \varphi$. As such, there is an equivalence between any set of sentences and a single conjunctive sentences which represents it, and we can therefore encode any knowledge base $\mathcal{K}$ into a single RBM by encoding this conjunction.

We now show that, given some WFF φ, it is possible to generate a Symmetric Connectionist Network (SCN) $\mathcal{N}_{SCN} = (X, H, W)$ and energy function E which will satisfy the definition of equivalence with φ. Combining these two generative processes, we will have a process for converting any formula into an RBM $\mathcal{N} = (X, H, W, E)$.

Lemma 2.3.5. *Let X, H be a set of visible and hidden nodes, respectively. Any SDNF $\varphi \equiv \bigvee_j \left(\bigwedge_{t \in S_{T_j}} x_t \wedge \bigwedge_{k \in S_{K_j}} \neg x_k \right)$ can be mapped onto an energy function*

$$E = -\sum_j \prod_{t \in S_{T_j}} x_t \prod_{k \in S_{K_j}} (1 - x_k)$$

where S_{T_j} and S_{K_j} are respectively the set of T_j indices of positive literals and the set of K_j indices of negative literals.

Proof. By definition, $\varphi \equiv \bigvee_j (\bigwedge_{t \in S_{T_j}} x_t \wedge \bigwedge_{k \in S_{K_j}} \neg x_k)$. Each conjunctive clause $(\bigwedge_{t \in S_{T_j}} x_t \wedge \bigwedge_{k \in S_{K_j}} \neg x_k)$ corresponds to $\prod_{t \in S_{T_j}} x_t \prod_{k \in S_{K_j}} (1 - x_k)$, which maps to 1 if and only if $x_t = 1$ and $x_k = 0$, i.e. $x_t = \textit{True}$ and $x_k = \textit{False}$, for all $t \in S_{T_j}$ and $k \in S_{K_j}$. Since φ is in SDNF, i.e., it is *True* if and only if one conjunctive clause is *True*, and so the sum $\sum_j \prod_{t \in S_{T_j}} x_t \prod_{k \in S_{K_j}} (1 - x_k) = 1$ if and only if the assignment of truth-values for x_t, x_k is a preferred valuation of φ, and 0 otherwise, i.e. E is minimized by preferred valuations. Hence, there exists an energy function $E = -\sum_j \prod_{t \in S_{T_j}} x_t \prod_{k \in S_{K_j}} (1 - x_k)$ such that $s_\varphi(\boldsymbol{x}) = -E_{\textit{rank}}(\boldsymbol{x})$. ■

We are now prepared to convert any formula φ into an RBM using its SDNF.

Theorem 2.3.6. *Any SDNF $\varphi \equiv \bigvee_j \left(\bigwedge_{t \in S_{T_j}} x_t \wedge \bigwedge_{k \in S_{K_j}} \neg x_k \right)$ can be mapped onto an equivalent RBM with energy function*

$$E = -\sum_j h_j \left(\sum_{t \in S_{T_j}} x_t - \sum_{k \in S_{K_j}} x_k - T_j + \epsilon \right),$$

where $0 < \epsilon < 1$ and S_{T_j} and S_{K_j} are respectively the set of T_j indices of positive literals and the set of K_j indices of negative literals.

Proof. We have seen in Lemma 2.3.5 that an SDNF φ can be mapped onto energy function $E = -\sum_j \prod_{t \in S_{T_j}} x_t \prod_{k \in S_{K_j}} (1 - x_k)$. We will denote T_j as the number of positive propositions in a conjunctive clause j. For each term $\tilde{e}_j(\boldsymbol{x}) = -\prod_{t \in S_{T_j}} x_t \prod_{k \in S_{K_j}} (1 - x_k)$, we define an energy term associated with a hidden variable h_j as $e_j(\boldsymbol{x}, h_j) = -h_j \left(\sum_{t \in S_{T_j}} x_t - \sum_{k \in S_{K_j}} x_k - T_j + \epsilon \right)$ with $0 < \epsilon < 1$ such that $\tilde{e}_j(\boldsymbol{x}) = \frac{e_{j\textit{rank}}(\boldsymbol{x})}{\epsilon}$, where $e_{j\textit{rank}}(\boldsymbol{x}) = \min_{h_j} e_j(\boldsymbol{x}, h_j)$ and $h_j \in \{0, 1\}$. This equation holds because $-\left(\sum_{t \in S_{T_j}} x_t - \sum_{k \in S_{K_j}} x_k - T_j + \epsilon \right) = -\epsilon$ if and only if $x_t = 1$ and $x_k = 0$ for all $t \in S_{T_j}$ and $k \in S_{K_j}$, which makes $\min_{h_j} e_j(\boldsymbol{x}, h_j) = -\epsilon$ with $h_j = 1$. Otherwise, $-\left(\sum_{t \in S_{T_j}} x_t - \sum_{k \in S_{K_j}} x_k - T_j + \epsilon \right) > 0$, and then $\min_{h_j} e_j(\boldsymbol{x}, h_j) = 0$ with $h_j = 0$. By repeating the process on every term $\tilde{e}_j(\boldsymbol{x})$ we can conclude that any SDNF φ is equivalent with an RBM having

the energy function:

$$E = -\sum_j h_j \left(\sum_{t \in S_{T_j}} x_t - \sum_{k \in S_{K_j}} x_k - T_j + \epsilon \right), \tag{2.3.8}$$

where $s_\varphi(\boldsymbol{x}) = -\frac{1}{\epsilon} E_{rank}(\boldsymbol{x})$. ■

In particular, we can now present the core result that the logical implication studied above can be represented by an RBM.

Theorem 2.3.7. *A logical implication* $y \leftarrow \bigwedge_{t \in S_T} x_t \wedge \bigwedge_{k \in S_K} \neg x_k$ *can be represented by an RBM with the energy function:*

$$E = -h_y \left(\sum_{t \in S_T} x_t - \sum_{k \in S_K} x_k + y - T - 1 + \epsilon \right) - \sum_{p \in S_T \cup S_K} h_p \left(\sum_{t \in S_T . \backslash p} x_t - \sum_{k \in S_K . \backslash p} x_k + x'_p - |S_T . \backslash p| - \mathbb{I}_{p \in S_K} + \epsilon \right)$$

where $|S_T . \backslash p|$ *is the cardinality of the set* $S_T . \backslash p$*; if* $p \in S_T$*, then* $x'_p = -x_p$ *and* $\mathbb{I}_{p \in S_K} = 0$*, else* $x'_p = x_p$ *and* $\mathbb{I}_{p \in S_K} = 1$*.*

Proof. We first apply Theorem 2.2.4 to our logical implication and represent it in SDNF:

$$\left(y \wedge \bigwedge_{t \in S_T} x_t \wedge \bigwedge_{k \in S_K} \neg x_k \right) \vee \bigvee_{p \in S_T \cup S_K} \left(\bigwedge_{t \in S_T . \backslash p} x_t \wedge \bigwedge_{k \in S_K . \backslash p} \neg x_k \wedge x'_p \right)$$

We now apply Theorem 2.3.6 to each of our clauses in two parts:

- Applying Theorem 2.3.6 to $\left(y \wedge \bigwedge_{t \in S_T} x_t \wedge \bigwedge_{k \in S_K} \neg x_k \right)$, we create an energy term

$$e_0 = -h_y \left(\sum_{t \in S_T} x_t - \sum_{k \in S_K} x_k + y - T - 1 + \epsilon \right) \tag{2.3.9}$$

- Applying Theorem 2.3.6 to the clauses of the form $(\bigwedge_{t \in S_T.\backslash p} x_t \wedge \bigwedge_{k \in S_K.\backslash p} \neg x_k \wedge x'_p)$, we create for each $p \in S_T \cup S_K$ an energy term

$$e_p = -h_p \left(\sum_{t \in S_T.\backslash p} x_t - \sum_{k \in S_K.\backslash p} x_k + x'_p - |S_T.\backslash p| - \mathbb{I}_{p \in S_K} + \epsilon \right)$$

  Summing over all terms of this form, we create an energy term

$$e_1 = - \sum_{p \in S_T \cup S_K} h_p \left(\sum_{t \in S_T.\backslash p} x_t - \sum_{k \in S - K.\backslash p} x_k + x'_p - |S_T.\backslash p| - \mathbb{I}_{p \in S_K} + \epsilon \right) \tag{2.3.10}$$

Adding together equations (2.3.9) and (2.3.10), we obtain the energy function associated with the RBM that represents our logical implication:

$$E = -h_y \left(\sum_{t \in S_T} x_t - \sum_{k \in S_K} x_k + y - T - 1 + \epsilon \right) - \sum_{p \in S_T \cup S_K} h_p \left(\sum_{t \in S_T.\backslash p} x_t - \sum_{k \in S_K.\backslash p} x_k + x'_p - |S_T.\backslash p| - \mathbb{I}_{p \in S_K} + \epsilon \right)$$

■

§ 2.4 Analysis of RBM Logic Representation

Here, we extend the analysis of Tran [16] and explore some properties of classical propositional logic semantics, namely Transitivity, *Ex Falso Quodlibet* (the "Principle of Explosion"), Disjunctive Syllogism, Resolution, and a modified Resolution Refutation, to see whether they are faithfully recreated in toy models of the method shown above, which we will henceforth refer to as RBM Logic. We will encode preconditions for each of these properties into RBM Logic, analyze the mathematical behavior of their energy functions, and interpret the logical consequences of the preferred valuations appropriately.

§ 2.4.1 Transitivity

The property of Transitivity is foundational to making chains of arguments beginning with premises and inferring towards conclusions. With respect to material implication $\rightarrow$, Transitivity is defined:

Definition 2.4.1. *The logical implication $\rightarrow$ is said to be transitive if and only if:*

$$\mathcal{K} \vDash (P \rightarrow Q) \wedge (Q \rightarrow R) \Rightarrow \mathcal{K} \vDash P \rightarrow R$$

for any knowledge base $\mathcal{K}$ and sentences P,Q, and R. This rule can be expressed syntactically as

$$\frac{P \rightarrow Q,\ Q \rightarrow R}{P \rightarrow R}$$

It can be shown that when represented in RBM Logic, the property of Transitivity for logical implication does hold.

Theorem 2.4.2. *When two logical implications are encoded into RBM Logic using Theorem 2.3.6, the defined RBM and corresponding energy function behave such that the property of Transitivity holds.*

Justification. First, we will define our $\mathcal{K}$ to prime the system for Transitivity.

$$\mathcal{K} \equiv (P \rightarrow Q) \wedge (Q \rightarrow R) \tag{2.4.11}$$

We now must show that, using the RBM energy function, any model of $\mathcal{K}$ is also a model of $P \rightarrow R$. First, we convert (2.4.11) to SDNF:

$$\mathcal{K} \equiv (P \wedge Q \wedge R) \vee (\neg P \wedge Q \wedge R) \vee (\neg P \wedge \neg Q) \tag{2.4.12}$$

Using Theorem 2.3.6 and defining $\epsilon = 0.5$[2], we are able to define an RBM and energy function from (2.4.12):

$$E = -h_1 (P + Q + R - 2.5) - h_2 (-P + Q + R - 1.5) - h_3 (-P - Q + 0.5) \tag{2.4.13}$$

We note here that we make use of the more general Theorem 2.3.6 rather than the more complex 2.3.7, which does deal explicitly with logical implications. This choice has been made primarily for a consistent application of 2.3.6 throughout, which is more naturally employed in the remainder of our proofs. It is expected that a simple summation of implication energy functions should serve to represent the conjunction of implications in a knowledge base, but further analysis is desired to confirm this.

We now consider the truth value assignments $\boldsymbol{x}_i$ which minimize (2.4.13). We reiterate that we have restricted ourselves to the discrete case in this work, i.e. non-real valued logics. We conjecture that this property (and others explored in this way) can be shown to hold in real-valued cases as well, but that there may be more restrictions on the appropriate value of ϵ. For further work exploring the application of RBMs and DBNs to real-valued logics and weighted knowledge bases, see [14],[15] and [16].

Table 2.1: The Transitivity energy function (2.4.13) for each valuation $\boldsymbol{x}_i$.

Transitivity Energy Function							
$\boldsymbol{x}_i$	P	Q	R	Energy Function	Minimized Energy	Model of $\mathcal{K}$	Transitivity $(P \rightarrow R)$
$\boldsymbol{x}_1$	0	0	0	$2.5h_1 + 1.5h_2 - 0.5h_3$	$\mathbf{-0.5}$	**Yes**	**Yes**
$\boldsymbol{x}_2$	0	0	1	$1.5h_1 + 0.5h_2 - 0.5h_3$	$\mathbf{-0.5}$	**Yes**	**Yes**
$\boldsymbol{x}_3$	0	1	0	$1.5h_1 + 0.5h_2 + 0.5h_3$	0.0	No	**Yes**
$\boldsymbol{x}_4$	0	1	1	$0.5h_1 - 0.5h_2 + 0.5h_3$	$\mathbf{-0.5}$	**Yes**	**Yes**
$\boldsymbol{x}_5$	1	0	0	$1.5h_1 + 2.5h_2 + 0.5h_3$	0.0	No	No
$\boldsymbol{x}_6$	1	0	1	$0.5h_1 + 1.5h_2 + 0.5h_3$	0.0	No	**Yes**
$\boldsymbol{x}_7$	1	1	0	$0.5h_1 + 1.5h_2 + 1.5h_3$	0.0	No	No
$\boldsymbol{x}_8$	1	1	1	$-0.5h_1 + 0.5h_2 + 1.5h_3$	$\mathbf{-0.5}$	**Yes**	**Yes**

[2]We will use this value for ϵ implicitly throughout this work. Any value $0 < \epsilon < 1$ will be unable to change the results of our machine by swapping a preferred valuation to a non-preferred valuation or vice versa, so we fix $\epsilon = 0.5$ throughout for ease of calculation.

In Table 2.1, we express the energy functions of each possible truth value assignment $\boldsymbol{x}_i$. This corresponds to setting each of the visible nodes of the RBM to a fixed value, and we then assign values for each h_j such that the energy function for that valuation is minimized. In practice, the range of each of these functions is a subset of the range of values for the energy function (2.4.13) that will be minimized by sampling for both the hidden and visible node values through Gibbs Sampling. Presenting the energy functions for fixed assignments $\boldsymbol{x}_i$ allows us to explore more thoroughly the behaviors of this method with respect to possible valuations, while still being able to identify the models associated with global minima.

Once we identify the set of valuations which minimize to the lowest energy values, the method claims that we have identified the set of models of $\mathcal{K}$. In order to prove our theorem, we must show that Transitivity holds in each of these identified models of $\mathcal{K}$, i.e. $P \to R$.

Observing Table 2.1, we see that the truth assignments $\boldsymbol{x}_1$, $\boldsymbol{x}_2$, $\boldsymbol{x}_4$, and $\boldsymbol{x}_8$ all have a minimized energy function value of -0.5, the lowest of any assignments (note also that this value is $-\epsilon$). Thus, these assignments would be preferred in the Gibbs Sampling minimization training process for the RBM. We also see that these four assignments are the only assignments which are models of $\mathcal{K}$, so the minimization process has properly isolated exactly those truth assignments which satisfy $\mathcal{K}$. Finally, we note that for each of these four assignments Transitivity holds, as the sentence $P \to R$ is semantically entailed. We have therefore shown through the RBM energy function method that $\mathcal{K} \vDash (P \to Q) \wedge (Q \to R) \Rightarrow \mathcal{K} \vDash P \to R$. ■

§ 2.4.2 *Ex Falso Quodlibet*

The property *Ex Falso Quodlibet*, also known as the "Principle of Explosion", states that from a contradiction, anything can be derived. Formally:

Definition 2.4.3 (*Ex Falso Quodlibet*)**.** *The rule of Ex Falso Quodlibet holds in a logic if and only if:*

$$\mathcal{K} \vDash (P \wedge \neg P) \Rightarrow \mathcal{K} \vDash Q$$

for any knowledge base $\mathcal{K}$ and sentences P and Q. This rule can be expressed syntactically as

$$\frac{P,\ \neg P}{Q}$$

Theorem 2.4.4. *When a contradiction is encoded into RBM Logic, the formalism will degenerate into triviality, identifying all possible valuations as preferred valuations, yet none will be models.*

Justification. We first define a contradictory knowledge base:

$$\mathcal{K} \equiv P \wedge \neg P \wedge R \wedge (Q \vee \neg Q)\,. \tag{2.4.14}$$

We include the literal R to explore the systems response to literals of our knowledge base well-founded despite the contradiction. We include the tautology $Q \vee \neg Q$ to explicitly include Q as literal of concern and a visible node in our RBM in order to analyze the system's response to literals with an otherwise unestablished truth value.

We now express (2.4.14) in SDNF:

$$\mathcal{K} \equiv (P \wedge \neg P \wedge R \wedge Q) \vee (P \wedge \neg P \wedge R \wedge \neg Q) \tag{2.4.15}$$

and define our energy function:

$$E = -h_1\,(P - P + R + Q - 2.5) - h_2\,(P - P + R - Q - 1.5)\,.$$

As we can see, each clause of the SDNF in which the contradiction holds has the expression $P - P$ included. These terms will consistently cancel each other out, and our energy function can therefore be simplified to:

$$E = -h_1\,(R + Q - 2.5) - h_2\,(R - Q - 1.5)\,. \tag{2.4.16}$$

We now consider the truth value assignments $\boldsymbol{x}_i$ which minimize (2.4.16). Note that because P and $-P$ have cancelled out of our energy function, the only relevant variable assignments are on Q and R.

Table 2.2: The *Ex Falso* energy function (2.4.16) for each valuation $\boldsymbol{x}_i$.

Ex Falso Quodlibet Energy Function						
$\boldsymbol{x}_i$	Q	R	Energy Function	Minimized Energy	Model of $\mathcal{K}$	*Ex Falso* (Q)
$\boldsymbol{x}_1$	0	0	$2.5h_1 + 1.5h_2$	0.0	No	No
$\boldsymbol{x}_2$	0	1	$1.5h_1 + 0.5h_2$	0.0	No	No
$\boldsymbol{x}_3$	1	0	$1.5h_1 + 2.5h_2$	0.0	No	**Yes**
$\boldsymbol{x}_4$	1	1	$0.5h_1 + 1.5h_2$	0.0	No	**Yes**

Observing Table 2.2, we note that any valuations over Q and R will provide a minimized energy value of 0.0, i.e. all valuations are preferred valuations, yet none are models. By our standard approach, viz., identifying the models of a knowledge base and checking that the property holds in every model, the property vacuously holds (as there are no models). However, one must also acknowledge that this analysis is founded upon the assumption that *both* P *and* $\neg P$ must be encoded into our knowledge base. Since the node x_i associated with P can only have a value of 0 or 1, this case is *prima facie* non-sensical. We therefore interpret these results as the RBM formalism degerating into triviality in the case of contradictory assertion. ■

We note that while the explicit property of *Ex Falso* does not hold, i.e., one cannot derive any proposition that they wish from a contradiction, the philosophical flavor of the property remains, in that the entire logical structure is reduced to trivial non-sense, preventing any form of satisfaction or inference.

One could even argue that this method is preferable to the classical case of contradiction, in that the association of the proposition P with a "physical" node that must have a set value prevents the possibility of $P \wedge \neg P$ holding not just in a syntactic sense, but in something akin to a semantic sense. The node *cannot* hold both values at once, and we conjecture that at best, the node would oscillate between values, preventing the machine from ever stabilizing into a minimal state. This structure seems then to prefer consistency above all else, preventing every valuation from being a model, rather than allowing any proposition to be derived.

We will explore in Ch. 3 an extension of this formalism that enables one to robustly handle the case of contradictory assertions.

§ 2.4.3 Disjunctive Syllogism

An important rule of inference used in classical logic is that of Disjunctive Syllogism.

Definition 2.4.5 (Disjunctive Syllogism)**.** *The rule of Disjunctive Syllogism holds in a logic if and only if*

$$\mathcal{K} \vDash (P \vee Q) \wedge \neg P \Rightarrow \mathcal{K} \vDash Q$$

for any knowledge base $\mathcal{K}$ and sentences P and Q. This rule can be expressed syntactically as

$$\frac{(P \vee Q)\,,\ \neg P}{Q}.$$

Theorem 2.4.6. *Disjunctive Syllogism holds in RBM Logic.*

Justification. We begin by defining a knowledge base

$$\mathcal{K} \equiv (P \vee Q) \wedge \neg P$$

and converting it into SDNF

$$\mathcal{K} \equiv (P \wedge \neg P) \vee (Q \wedge \neg P)\,. \tag{2.4.17}$$

Using Theorem 2.3.6, we create an energy function from (2.4.17)

$$E = -h_1\,(P - P - 0.5) - h_2\,(Q - P - 0.5)$$

and cancel out $P - P$ to get

$$E = -h_1\,(-0.5) - h_2\,(Q - P - 0.5)\,. \tag{2.4.18}$$

We now consider possible truth value assignments $\boldsymbol{x}_i$ and identify the valuations which minimize (2.4.18).

Table 2.3: The Disjunctive Syllogism energy function (2.4.18) for each valuation $\boldsymbol{x}_i$.

Disjunctive Syllogism Energy Function						
$\boldsymbol{x}_i$	P	Q	Energy Function	Minimized Energy	Model of $\mathcal{K}$	Disjunctive Syllogism (Q)
$\boldsymbol{x}_1$	0	0	$0.5h_1 + 0.5h_2$	0.0	No	No
$\boldsymbol{x}_2$	0	1	$0.5h_1 - 0.5h_2$	$\mathbf{-0.5}$	**Yes**	**Yes**
$\boldsymbol{x}_3$	1	0	$0.5h_1 + 1.5h_2$	0.0	No	No
$\boldsymbol{x}_4$	1	1	$0.5h_1 + 0.5h_2$	0.0	No	**Yes**

Observing Table 2.3, we see that $\boldsymbol{x}_2$ is the only valuation with minimial energy, and it is both a model of $\mathcal{K}$ and assigns Q a value of *True*, i.e. Disjunctive Syllogism holds. ■

§ 2.4.4 Resolution

One of the central tools of logic programming is the method of Resolution, which is a process of eliminating complementary literals from conjoined disjunctive clauses through Disjunctive Syllogism.

Definition 2.4.7. *The generalized Resolution rule can be stated as*

$$\frac{l_1 \vee \ldots \vee l_k, \qquad m_1 \vee \ldots \vee m_n}{l_1 \vee \ldots \vee l_{i-1} \vee l_{i+1} \vee \ldots \vee l_k \vee m_1 \vee \ldots \vee m_{j-1} \vee m_{j+1} \vee \ldots \vee m_n}, \tag{2.4.19}$$

where l_i and m_j are complementary literals, i.e. $l_i \equiv \neg m_j$ [11].

Theorem 2.4.8. *The generalized rule of Resolution holds in RBM Logic for resolvents of the form $(P \vee Q) \wedge (\neg P \vee R)$. That is:*

$$\mathcal{K} \vDash (P \vee Q) \wedge (\neg P \vee R) \Rightarrow \mathcal{K} \vDash (Q \vee R) \tag{2.4.20}$$

Justification. For compactness and simplicity's sake, we only show this for short resolvents, as this can be expanded to the generalized rule [11].

We begin by defining a knowledge base on which to test the validity of the Resolution rule

$$\mathcal{K} \equiv (P \vee Q) \wedge (\neg P \vee R)\,,$$

and express it in SDNF:

$$\mathcal{K} \equiv (\neg P \wedge Q \wedge \neg R) \vee (\neg P \wedge Q \wedge R) \vee (P \wedge \neg Q \wedge R) \vee (P \wedge Q \wedge R)\,. \tag{2.4.21}$$

Using Theorem 2.3.6, we define an energy function to represent (2.4.21):

$$\begin{aligned} E = -h_1\,(-P + Q - R - 0.5) - h_2\,(-P + Q + R - 1.5) \\ - h_3\,(P - Q + R - 1.5) - h_4\,(P + Q + R - 2.5) \end{aligned} \tag{2.4.22}$$

and identify the valuations $\boldsymbol{x}_i$ which minimize (2.4.22).

Table 2.4: The Resolution energy function (2.4.22) for each valuation $\boldsymbol{x}_i$.

Resolution Energy Function							
$\boldsymbol{x}_i$	P	Q	R	Energy Function	Minimized Energy	Model of $\mathcal{K}$	Resolution $(Q \vee R)$
$\boldsymbol{x}_1$	0	0	0	$0.5h_1 + 1.5h_2 + 1.5h_3 + 2.5h_4$	0.0	No	No
$\boldsymbol{x}_2$	0	0	1	$1.5h_1 + 0.5h_2 + 0.5h_3 + 1.5h_4$	0.0	No	**Yes**
$\boldsymbol{x}_3$	0	1	0	$-0.5h_1 + 0.5h_2 + 2.5h_3 + 1.5h_4$	$\mathbf{-0.5}$	**Yes**	**Yes**
$\boldsymbol{x}_4$	0	1	1	$0.5h_1 - 0.5h_2 + 1.5h_3 + 0.5h_4$	$\mathbf{-0.5}$	**Yes**	**Yes**
$\boldsymbol{x}_5$	1	0	0	$1.5h_1 + 2.5h_2 + 0.5h_3 + 1.5h_4$	0.0	No	No
$\boldsymbol{x}_6$	1	0	1	$2.5h_1 + 1.5h_2 - 0.5h_3 + 0.5h_4$	$\mathbf{-0.5}$	**Yes**	**Yes**
$\boldsymbol{x}_7$	1	1	0	$0.5h_1 + 1.5h_2 + 1.5h_3 + 0.5h_4$	0.0	No	**Yes**
$\boldsymbol{x}_8$	1	1	1	$1.5h_1 + 0.5h_2 + 0.5h_3 - 0.5h_4$	$\mathbf{-0.5}$	**Yes**	**Yes**

Observing Table 2.4, we see that $\boldsymbol{x}_3$, $\boldsymbol{x}_4$, $\boldsymbol{x}_6$, and $\boldsymbol{x}_8$ are the preferred valuations with minimized energy value -0.5. We also note that each of these valuations is a model of $\mathcal{K}$ and that the generalized Resolution rule holds for each one. ■

§ 2.4.5 Resolution Refutation

A standard method of checking for entailment or consistency in logic programming is Resolution Refutation. This process takes a knowledge base $\mathcal{K}$ and a query Q, creates a new knowledge base $\mathcal{K}' \equiv \mathcal{K} \cup \{\neg Q\}$ and repeatedly applies Resolution to the sentences of the new knowledge base $\mathcal{K}'$. If the empty clause is derived through this process, then $\mathcal{K}'$ is shown to be inconsistent, and thus $\mathcal{K} \vDash Q$ is proven. Because this

refutation can be made for any knowledge base and query, Resolution is considered a refutation-complete inference technique [1].

We now consider a similar process in the RBM Logic. We first define our idea of Resolution Refutation within this method.

Definition 2.4.9. *Resolution Refutation in the RBM Logic will be defined as the process of adding the query Q to the knowledge base $\mathcal{K}$, then creating and minimizing the resulting energy function based upon $\mathcal{K} \cup \{Q\}$.*

Notice that this does differ from the standard Resolution Refutation process in that Q is added to $\mathcal{K}$, rather than $\neg Q$. This convention of definition allows the following resulting theorem to be more intuitive.

Theorem 2.4.10. *Given a knowledge base $\mathcal{K}$ and a query Q, the RBM Logic will prefer no valuations if $\mathcal{K} \cup \{Q\}$ is inconsistent and will prefer models in which $Q \equiv_{\boldsymbol{x}_i} True$ if $\mathcal{K} \cup \{Q\}$ is consistent.*

Justification.

Claim 1: The RBM Logic will prefer no models if $\mathcal{K} \cup \{Q\}$ is inconsistent

Subproof. We define a simple knowledge base:

$$\mathcal{K} \equiv P \wedge (P \rightarrow Q)$$

and inconsistent query:

$$\neg Q.$$

We add our query to our knowledge base and get

$$\mathcal{K}' \equiv P \wedge (P \rightarrow Q) \wedge \neg Q,$$

which we then convert into SDNF

$$\mathcal{K}' \equiv (P \wedge \neg P \wedge \neg Q) \vee (P \wedge Q \wedge \neg Q)\,. \tag{2.4.23}$$

We note here that each of our conjunctive clauses contains an explicit contradiction, i.e. $P \wedge \neg P$ and $Q \wedge \neg Q$. As such, none of the clauses in (2.4.23) can actually be satisfied, as is to be expected in the inconsistent case. Further, the SDNF of $\mathcal{K}'$ amounts to conjoining the query into each of our conjunctive clauses in the SDNF of $\mathcal{K}$.

We now use (2.4.23) to define our energy function:

$$E = -h_1\left(P - P - Q - 1 + 0.5\right) - h_2\left(P + Q - Q - 2 + 0.5\right),$$

from which we cancel out contradictions and simplify to:

$$E = -h_1\left(-Q - 0.5\right) - h_2\left(P - 1.5\right). \tag{2.4.24}$$

We now consider the possible valuations $\boldsymbol{x}_i$ and identify those which minimize the energy function (2.4.24).

Table 2.5: The inconsistent Refutation energy function (2.4.24) for each valuation $\boldsymbol{x}_i$.

Inconsistent Refutation Energy Function						
$\boldsymbol{x}_i$	P	Q	Energy Function	Minimized Energy	Model of $\mathcal{K}'$	Model of $\mathcal{K}$
$\boldsymbol{x}_1$	0	0	$0.5h_1 + 1.5h_2$	0.0	No	No
$\boldsymbol{x}_2$	0	1	$1.5h_1 + 1.5h_2$	0.0	No	No
$\boldsymbol{x}_3$	1	0	$0.5h_1 + 0.5h_2$	0.0	No	No
$\boldsymbol{x}_4$	1	1	$1.5h_1 + 0.5h_2$	0.0	No	**Yes**

As we expected, there are no assignments which would serve as a model of $\mathcal{K} \cup \{Q\}$. We can also see from Table 2.5 that all assignments have the same minimized energy, and as such none are selected as preferred assignments. □

Claim 2: The RBM Logic will prefer valuations $\boldsymbol{x}_i$ in which $Q \equiv_{\boldsymbol{x}_i} \mathit{True}$ if $\mathcal{K} \cup \{Q\}$ is consistent.

Subproof. We use the same knowledge base as before, but instead now offer Q as our query. Therefore,

$$\mathcal{K}' \equiv P \wedge (P \to Q) \wedge Q,$$

and when converted into SDNF:

$$\mathcal{K}' \equiv (P \wedge \neg P \wedge Q) \vee (P \wedge Q \wedge Q)\,. \tag{2.4.25}$$

We now define our energy function to represent (2.4.25)

$$E = -h_1\,(P - P + Q - 2 + 0.5) - h_2\,(P + Q + Q - 3 + 0.5)$$

and simplify it to

$$E = -h_1\,(Q - 1.5) - h_2\,(P + 2Q - 2.5)\,. \tag{2.4.26}$$

We now consider possible evaluations $\boldsymbol{x}_i$ and identify those which minimize energy function (2.4.26).

Table 2.6: The consistent Refutation energy function (2.4.26) for each valuation $\boldsymbol{x}_i$

Consistent Refutation Energy Function						
$\boldsymbol{x}_i$	P	Q	Energy Function	Minimized Energy	Model of $\mathcal{K}'$	Model of $\mathcal{K}$
$\boldsymbol{x}_1$	0	0	$1.5h_1 + 2.5h_2$	0.0	No	No
$\boldsymbol{x}_2$	0	1	$0.5h_1 + 0.5h_2$	0.0	No	No
$\boldsymbol{x}_3$	1	0	$1.5h_1 + 1.5h_2$	0.0	No	No
$\boldsymbol{x}_4$	1	1	$0.5h_1 - 0.5h_2$	$\mathbf{-0.5}$	**Yes**	**Yes**

We see then that $\boldsymbol{x}_4$ is the valuation with minimum energy and the only valuation which serves as a model for $\mathcal{K} \cup \{Q\}$. □

■

§ 2.5 Chapter Conclusion

We have therefore seen that each of the studied properties either hold exactly as they would be expected to hold in standard propositional logic, or (as in the case of Resolution Refutation) hold in a slightly modified but similar fashion.

Study of these properties has also yielded insight into the function of the RBM formalism. In particular, in the case of contradictions, one may note that the inclusion of a complementary pair of literals in a single conjunctive clause of the SDNF leads to the associated hidden node never activating. In the energy function term for such an h_j, the canceling out of a positive literal x_i, which initially contributed to an increased $|T_j|$, forces the coefficient of h_j to remain negative regardless of any possible binary assignment $\boldsymbol{x}_i$. As such, h_j must always receive a value of 0 in order to minimize the energy function. Since this value never varies, its assignment will have no influence on the value of the energy function, and no valuation can be preferred by this term. In the case of a necessarily contradictory knowledge base, every potential model for $\mathcal{K}$ must include a pair of these contradictory literals, and the result is that every term associated with some h_j will contain the complementary pair, forcing $h_j = 0$ for all j. In this situation, no truth assignment will be preferred, and none could be considered a viable model.

The inability to robustly respond to contradictory assertions is undesirable when one considers representing real-world knowledge bases, and we will focus in the coming chapters on extending the capabilities of the formalism to address contradiction.

CHAPTER 3

PARACONSISTENT LOGIC

§ 3.1 Chapter Introduction

While an encoding of propositional logic into a connectionist method is powerful, there have also been many extensions of propositional logic which are also worth considering. One such extension is into three-valued so-call paraconsistent logics. These logics seek to address issues that may arise within propositional logic if one were to allow the possibility of contradictory values for a given sentence.

§ 3.2 Paraconsistent Logics

Recall our exploration of *Ex Falso Quodlibet*, "The Principle of Explosion." If one were to come across a situation in which they were applying propositional logic to a given domain and encounter a situation in which their system derives both a sentence and its negation, suddenly every possibly sentence (and its negation!) could become $True$. In practical applications, this is obviously far from ideal. The contradictory values most likely speak to some error in the knowledge base or representation of the domain, rather than that all of possibility is actively $True$.

The paraconsistent logics are a method of addressing exactly this sort of issue. The simplest approach is to introduce a third truth value which corresponds neither with $True$ nor $False$, but the status of this value depends upon the system which is in use.

We establish our definiton of a *logic* and *designated values*.

Definition 3.2.1. *A logic is a triple $\mathfrak{L} = \langle L, V, D \rangle$ where L is a langauge consisting of an alphabet and formation rules (which leads to a set of well-formed formulas W of the language), V is a set of values which can be assigned to the well-formed formulas of L by a valuation function $\nu : W \to V$, and $D \subseteq V$ are*

the designated values. We say a well-formed formula $\varphi \in W$ is satisfied by a valuation ν if and only if $\nu(\varphi) \in D$.

In the classical two-valued case, the set $V = \{True, False\}$ and the set $D = \{True\}$, i.e., $True$ is the only value that can be used to satisfy any sentence. When we introduce new values $v \in V$, we must determine whether or not they will also be inserted into our set D and count as designated values.

§ 3.2.1 *K3*

One way to understand a sentence that is assigned this additional value would be that the sentence is *neither $True$ nor $False$*. It would represent a gap in our knowledge such that we cannot confirm that the sentence is either. This is the approach taken for the system *K3*, also known as Kleene's Strong Logic of Indeterminacy, which *does not* include the new valuation, which we will call $Neither$ or N, in the designated values D.

Definition 3.2.2. *K3 is a three-valued logic in which a sentence s can have a valuation $\nu(s) \in \{-1, 0, 1\}$, where $\nu(s) = -1$ corresponds with "s is False", $\nu(s) = 1$ corresponds with "s is True", and $\nu(s) = 0$ corresponds with "s is Neither (True nor False)." In this logic, the designated values $D = \{True\}$.*

Because the value $Neither$ or N is not given the designated status, while contradictory values are allowed to exist without resulting in the explosion of the entire system, a contradictory valuation is not permitted for satisfaction of a sentence.

Because of this additional value, we must reconsider the truth tables for each of our standard logical connectives and how they function under *K3*. To ease reading, we will use the notation that $F = -1$, $T = 1$, and $N = 0$, as well as bolding the designated values for the connective.

Table 3.1: The negation connective in *K3*.

A	$\neg A$
F	$\mathbf{T}$
N	N
T	F

Table 3.2: The conjunction connective in *K3*.

$A \wedge B$		B		
		F	N	T
A	F	F	F	F
	N	F	N	N
	T	F	N	$\mathbf{T}$

Table 3.3: The disjunction connective in *K3*.

$A \vee B$		B		
		F	N	T
A	F	F	N	$\mathbf{T}$
	N	N	N	$\mathbf{T}$
	T	$\mathbf{T}$	$\mathbf{T}$	$\mathbf{T}$

Table 3.4: The implication connective in *K3*.

$A \rightarrow B$		B		
		F	N	T
A	F	$\mathbf{T}$	$\mathbf{T}$	$\mathbf{T}$
	N	N	N	$\mathbf{T}$
	T	F	N	$\mathbf{T}$

§ 3.2.2 *LP*

A different approach to understanding a sentence that is assigned this additional value would be that the sentence is *both $True$ and $False$*. It would represent a glut in our knowledge such that we can confirm that the sentence is both. This is the approach taken for the system *LP*, or Graham Priest's Logic of Paradox, which *does* include the new valuation, which we will call $Both$ or B, in the designated values D.

Definition 3.2.3. *LP is a three-valued logic in which a sentence s can have a valuation $\nu(s) \in \{-1, 0, 1\}$, where $\nu(s) = -1$ corresponds with "s is $False$", $\nu(s) = 1$ corresponds with "s is $True$", and $\nu(s) = 0$ corresponds with "s is $Both$ ($True$ and $False$)." In this logic, the designated values $D = \{True, Both\}$.*

Because the value $Both$ or B is given the designated status, contradictory values are allow to exist without resulting in the explosion of the entire system, and a contradictory valuation is permitted for satisfaction of a sentence.

We now reconsider the truth tables for each of our standard logical connectives and how they function under *LP*. To ease reading, we again use the notation that $F = -1, T = 1$, and $B = 0$, and continue to bold the designated valuations of each connective.

Table 3.5: The negation connective in *LP*.

A	$\neg A$
F	$\mathbf{T}$
B	$\mathbf{B}$
T	F

Table 3.6: The conjunction connective in *LP*.

$A \wedge B$		B		
		F	B	T
A	F	F	F	F
	B	F	$\mathbf{B}$	$\mathbf{B}$
	T	F	$\mathbf{B}$	$\mathbf{T}$

Table 3.7: The disjunction connective in *LP*.

$A \vee B$		B		
		F	B	T
A	F	F	$\mathbf{B}$	$\mathbf{T}$
	B	$\mathbf{B}$	$\mathbf{B}$	$\mathbf{T}$
	T	$\mathbf{T}$	$\mathbf{T}$	$\mathbf{T}$

Table 3.8: The implication connective in *LP*.

$A \rightarrow B$		B		
		F	B	T
A	F	$\mathbf{T}$	$\mathbf{T}$	$\mathbf{T}$
	B	$\mathbf{B}$	$\mathbf{B}$	$\mathbf{T}$
	T	F	$\mathbf{B}$	$\mathbf{T}$

We note that these tables are structurally identical to Tables 3.1-3.4 discussed for the logic $K3$, noting that the value B is substituted for the value N in each place. The difference between these logics is founded

in whether or not the third value is designated and the consequences that this has on inference in either system.

§ 3.3 Paraconsistent Logics in RBMs

We now turn our attention to encoding the two discussed paraconsistent logics into our RBM method for representing logic. The core of our focus will be in modifying the energy function that is defined for the network, in particular the valuation function that is used within that energy function and identifying the proper SDNF in each context.

In general, *K3* and *LP* are defined as predicate logics, including both predicates and quantifiers. The encoding of these features into RBM remains an open question, and as such, we limit ourselves in this thesis to the propositional logic analogues of the more general languages.

We will first address *K3* , as it requires simpler modifications than the *LP* case. We will then use the intuition gained in the simpler case to address *LP* .

§ 3.3.1 *K3* in RBMs

In order to represent *K3* in an RBM, we must expand the domain of our valuation function to include the value $Neither$ and ensure that literals with this value are not used to satisfy any sentences. Since the only addition is the value $Neither$, which cannot be used to satisfy any of the disjuncts in the SDNF, we simply must account for the presence of this new value when defining our energy function. We will do this by passing the valuation of each atom through a function which assigns 1 to designated values and 0 to non-designated values.

K3 Strict Disjunctive Normal Form

In order to present a *K3* sentence φ in SDNF, no change is needed from the propositional case. $True$ remains the only value which can satisfy a clause of the SDNF, and as such we have simply introduced more ways in which the valuation can fail to satisfy the sentence. These do not need to be represented in our SDNF form, and as such the SDNF of a sentence will be identical in the classical propositional logic and in *K3* .

K3 Energy Function

We first recall the Heaviside Function:

Definition 3.3.1. *The Heaviside Function is a function* $\mathcal{H} : \mathbb{R} \to \{0, 1\}$ *such that*

$$\mathcal{H}(x) \coloneqq \begin{cases} 0 & \textit{if } x < 0 \\ 1 & \textit{if } x \geq 0 \end{cases} \tag{3.3.1}$$

and define a useful function which we shall refer to as the Dual Heaviside Function:

Definition 3.3.2. *The Dual Heaviside Function is a function* $\mathcal{H}^* : \mathbb{R} \to \{0, 1\}$ *such that*

$$\mathcal{H}^*(x) \coloneqq \begin{cases} 0 & \textit{if } x \leq 0 \\ 1 & \textit{if } x > 0 \end{cases}. \tag{3.3.2}$$

This Dual Heaviside Function has the property that $\mathcal{H}^*(0) = 0$, where $\mathcal{H}(0) = 1$. We note that this function has the property of mapping the valuations in *K3* to their designated value status, i.e., $\mathcal{H}^*(\nu(x)) = 1$ if and only if $\nu(x) = 1$ or $True$.

We must now convert our SDNF representation of a *K3* WFF φ into an energy function. In order to do so, we now define the K3 Designated Value Function:

Definition 3.3.3. *The Designated Values Function for a K3 valuation* ν *is a function* $D_v^{K3} : \{-1, 0, 1\} \to \{0, 1\}$ *where:*

$$D_v^{K3}(x) \coloneqq \begin{cases} 0 & \textit{if } \nu(x) = -1 \textit{ or } \nu(x) = 0 \\ 1 & \textit{if } \nu(x) = 1 \end{cases} \tag{3.3.3}$$

This function has the property of mapping a sentence of *K3* to its truth assignment's status as a designated value, i.e., whether it can be used to satisfy a model. We now show:

Lemma 3.3.4.

$$D_v^{K3}(x) = \mathcal{H}^*(\nu_{K3}(x)), \tag{3.3.4}$$

where we use $\nu_{K3}(x)$ *to indicate the valuation of a formula* x *in K3 .*

Proof. If we consider the *K3* truth values $\nu_{K3}(x) \in \{-1, 0, 1\}$ and apply the Dual Heaviside Function to them, we see that $\mathcal{H}^*(\nu_{K3}(x)) = 1$ if and only if $x = True$ and equals 0 otherwise. ■

We recall our mapping of a sentence presented in SDNF to an RBM for the propositional logic case:

Theorem 2.3.6. *Any SDNF* $\varphi \equiv \bigvee_j \left(\bigwedge_{t \in S_{T_j}} x_t \wedge \bigwedge_{k \in S_{K_j}} \neg x_k \right)$ *can be mapped onto an equivalent RBM with energy function*

$$E = -\sum_j h_j \left(\sum_{t \in S_{T_j}} x_t - \sum_{k \in S_{K_j}} x_k - T_j + \epsilon \right),$$

where $0 < \epsilon < 1$ *and* S_{T_j} *and* S_{K_j} *are respectively the set of* T_j *indices of positive literals and the set of* K_j *indices of negative literals.*

Implicit in this definition of the energy function is that the summations $\sum_{t \in S_{T_j}} x_t$ and $\sum_{k \in S_{K_j}} x_k$ are over the *values* of each literal, i.e. $\nu(x_t)$ and $\nu(x_k)$.

In order to adjust this energy function so that it will continue to function in the *K3* context, we must transform this process of valuation to align with the designated values of *K3* . Our Designated Values Function, Def. 3.3.3, does just that.

We are now prepared to define our energy function for the *K3* context.

Theorem 3.3.6. *Any K3 SDNF* $\varphi \equiv \bigvee_j \left(\bigwedge_{t \in S_{T_j}} x_t \wedge \bigwedge_{k \in S_{K_j}} \neg x_k \right)$ *can be mapped onto an equivalent RBM with energy function*

$$E = -\sum_j h_j \left(\sum_{t \in S_{T_j}} D_v^{K3}(x_t) - \sum_{k \in S_{K_j}} D_v^{K3}(x_k) - T_j + \epsilon \right),$$

where $0 < \epsilon < 1$ *and* S_{T_j} *and* S_{K_j} *are respectively the set of* T_j *indices of positive literals and the set of* K_j *indices of negative literals.*

By replacing the implicit valuation function ν with the Designated Values Function, we expand the domain of values to include the tertiary $Neither$ value and map it to the same value as $False$, putting the two non-designated values on equal footing while leaving the $True$ value to behave normally. Both non-designated values will fail to satisfy the positive literals in $\sum_{t \in S_{T_j}} D_v^{K3}(x_t)$, and the designated value will fail to satisfy the negative literals in $-\sum_{k \in S_{K_j}} D_v^{K3}(x_k)$, acting against the satisfaction of the conjunctive clause associated with node h_j.

Some reflection should convince the reader that the proof of this theorem follows that given for the analogous Thm. 2.3.6. Notice, however, that the function $s_\varphi(\boldsymbol{x})$ is now defined as the designated status of the sentence $\boldsymbol{x}$, rather than the strict truth status.

One must also convince oneself that the notions of a model and a preferred valuation of a formula carries over into the current context, since this is what allows the energy function analagous to that in Lemma 2.3.5 to be defined here.

Rather than spell out all of the details of a proof here, we rather look at properties of the RBM model of *K3* in Sect. 3.4 and confirm these properties as evidence supporting the validity of this theorem.

§ 3.3.2 *LP* in RBMs

We now turn our attention to the *LP* case.

LP Strict Disjunctive Normal Form

In order to represent *LP* in an RBM, we must take a dual approach to the one outlined in Sect. 3.3.1, i.e., atoms with the new value of $Both$ should be permitted to satisfy any clauses and qualify the valuation as a model. However, we must now put the tertiary valuation on an equal footing with the value $True$, as it is also a designated value.

In the same spirit as Defn. 3.3.3, we therefore define the *LP* designated value function:

Definition 3.3.7. *The Designated Values Function for an LP valuation ν is a function* $D_v^{LP} : \{-1, 0, 1\} \to \{0, 1\}$ *where:*

$$D_v^{LP}(x) \coloneqq \begin{cases} 0 & \textit{if } \nu(x) = -1 \\ 1 & \textit{if } \nu(x) = 1 \textit{ or } \nu(x) = 0 \end{cases} \tag{3.3.5}$$

This function has the property of mapping a sentence of *LP* to its truth assignment's status as a designated value, i.e., whether it can be used to satisfy a model. We now show:

Lemma 3.3.8.

$$D_v^{LP}(x) = \mathcal{H}(\nu_{LP}(x)) \tag{3.3.6}$$

Proof. If we consider the *LP* truth values $\nu_{LP}(x) \in \{-1, 0, 1\}$ and apply the Heaviside Function to them, we see that $\mathcal{H}(\nu_{LP}(x)) = 0$ if and only if $x = False$ and equals 1 otherwise. ■

We next extend the definition of SDNF naturally to the case of *LP*.

Definition 3.3.9. *A "strict DNF" (SDNF) is a DNF where at most one single conjunctive clause has a designated value at a time.*

It is no longer sufficient for only one clause to simply be $True$, but rather only one clause can receive a designated value. We conjecture that this extension to the *LP* case works as a general extension of SDNF

that could be used to apply this formalism to any desired logic, providing other appropriate extensions are made.

However, this extensions brings additional complications. Consider *LP* in relation to *K3* . Both logics introduce an additional valuation over the classical two-valued case, but in *LP* this additional value is also designated. As such, while a valuation could have resulted in a conjunctive clause in an SDNF being evaluated as $Neither$ (and consequentially unsatisfied) in the *K3* case without affecting that clause's satisfaction of the SDNF condition, in the *LP* case a conjunctive clause could receive the value $Both$, resulting in the satisfaction of an additional conjunct and thus violating the SDNF condition. Because the SDNF condition is more restrictive on acceptable forms in the *LP* case than in our prior cases, special care must be taken to represent a sentence in the *LP* SDNF. We now construct a formalism that will allow this.

We claim that *LP* as defined lacks the expressive capability of presenting an arbitrary sentence in SDNF. We explore an example to develop the reader's intuition and highlight the expressive failings.

Consider the sentence $\Psi = (P \vee Q)$. Ψ is in standard DNF, and we wish to present it in SDNF. We must create an equivalent disjunction of conjunctive clauses that partitions models of Ψ such that any model satisfies only one of the conjunctive clauses.

Consider the approach in the classical two-valued case. To begin constructing our SDNF, we propose that SDNF (Ψ) have the form $(P \vee \ldots)$, i.e., this first disjunct will be the (trivially conjunctive) clause P, which captues all models of Ψ for which $P = True$. When we construct our next clause, we must find a sentence that will be satisfied by models of Ψ *but not by models of the first clause*. So, the form of our second clause will be $(\neg P \wedge \ldots)$ to satisfy this condition. We then move on to satisfying Ψ by extending this second clause to be $(\neg P \wedge Q)$. This clause captures the remainder of the models of Ψ, so we have constructed SDNF $(\Psi) = (P \vee (\neg P \wedge Q))$, which can be read in English as 'Either P is $True$, or P is not $True$ and Q is $True$.'

Now consider the same process in *LP* . We propose the same form of the SDNF, i.e. the first clause of the sentence will be the clause P, which captures all models of Ψ for which $P = True\ or\ Both$, or where P is *satisfied*. When constructing our second clause, the first conjunct must now capture the idea that

'P *is not satisfied*,' rather than the classical 'P is not $True$.' In the classical case, $\neg P$ sufficed; whenever $\neg P$ was designated, P was non-designated. This is not the case in LP. Consider the case for which $\nu(P) = Both$. $\nu(\neg P) = Both$ as well, so $\neg$ no longer captures the necessary expression 'is not satisfied' or 'is strictly $False$'.

We conjecture that no combination of LP connectives can be used to express the sentiment that a sentence is not satisfied, and LP is therefore incapable of expressing an arbitrary sentence in SDNF. We therefore develop an extended language to handle this issue, and it is important to note that although this approach is motivated by the previous conjecture, it does not depend on it being true.

Since we still wish to transform sentences of LP into SDNF so that they can be encoded using the RBM formalism, we use the Heaviside function and its dual as inspiration to introduce additional connectives, extending the language in such a way that we can generate an SDNF representation of sentences. Since we have now extended the language LP by introducing additional connectives into the alphabet, the SDNFs are formulas in the extended language, and valuations will also have to be extended accordingly. However, the original LP WFFs will be logically equivalent to the sentences in the new language, which we will refer to as LP^S, and this process of translation into a new language does not change the fact that it is the original LP WFF (as well as the LP^S WFF!) which is encoded into the RBM[1].

We now introduce our additional connectives.

Table 3.9: The Heaviside connective in LP^S.

A	$H(A)$
F	F
B	$\mathbf{T}$
T	$\mathbf{T}$

We allow these connectives to operate quite similarly to the unary negation connective $\neg$, noting however, that while the negation connective does not require us to extend the valuations for our language, the unary connectives H and H^* are in an extended language, and so require extension of the LP valuations.

[1]Note that the *necessity* of this new language relies upon the truth of our conjecture that LP lacks the ability to express the sentiment 'is not satisfied' or its negation 'is strictly $True$.' In the case that this conjecture is disproven, one can take the new primitive connectives introduced in this section to simply be short-hand for the appropriate sentences which capture their truth tables. As this would allow one to contain the SDNF within LP without translating into a new language and improve the elegance of this formalism, disproof of the conjecture is most welcome.

Table 3.10: The Dual Heaviside connective in LP^S .

A	$H^*(A)$
F	F
B	F
T	$\mathbf{T}$

Tables 3.9 and 3.10 should be read as rules for these extensions of valuations. The most crucial point is that a sentence of the form $H(P)$ or $H^*(P)$ will qualify as a literal and as such a conjunction of such sentences will qualify as a conjunctive clause as defined in Defn. 2.2.1.

We return now to our goal of expressing the *LP* sentence $\Psi = (P \vee Q)$ in SDNF. We have so far constructed SDNF $(\Psi) = (P \vee (\ldots))$ and now must insert a sentence which captures 'P is not satisfied' or 'P is strictly $False$' into our second clause. We propose the sentence $\neg H(P)$, which is only satisfied in the case that $P = False$! This is the exact behavior we desired, and we can then continue to construct our second clause by including the conjunct Q to satisfy Ψ, just as we did in the classical case. We have therefore successfully constructed the sentence SDNF $(\Psi) = (P \vee (\neg H(P) \wedge Q))$. In English, 'Either P is *satisfied*, or P is *not satisfied* and Q is *satisfied*.'

Employing our new connectives, we now present an SDNF form for each of the connectives of *LP* in order to give the reader an intuition about this process. We assume here that each of the sentences A and B are atomic, so these examples should not be taken as a definition of a general recursive method for converting an arbitrary sentence to SDNF. As a result of the fact that it is easier for paraconsistent sentences to satisfy the SDNF criterion than it is in the two-valued case that SDNF representations exist for all paraconsistent sentences, as it turns out, a general algorithmic process—particularly an efficient one—for converting paraconsistent sentences to SDNF remains an open problem. As such, we will propose SDNF forms for the paraconsistent statements throughout and invite the reader to confirm their validity for themselves[2]. The reader can check the claims made below against the truth tables in Sect. 3.2.2.

[2]Recall that the SDNF for a given sentence is, in general, non-unique. As such, one could define other representations for the connectives which would be logically equivalent.

$$\text{SDNF}(A) = A \tag{3.3.7}$$

$$\text{SDNF}(\neg A) = \neg A \tag{3.3.8}$$

$$\text{SDNF}(A \wedge B) = A \wedge B \tag{3.3.9}$$

$$\text{SDNF}(A \vee B) = A \vee (\neg H(A) \wedge B) \tag{3.3.10}$$

$$\text{SDNF}(A \rightarrow B) = \neg A \vee (H^*(A) \wedge B) \tag{3.3.11}$$

We note useful identities:

$$H(\neg x) = \neg H^*(x) \tag{3.3.12}$$

and

$$H^*(\neg x) = \neg H(x), \tag{3.3.13}$$

and require of our *LP* SDNFs that any negations be moved outside of H or H^*. This will allow for an easier conversion to the energy function.

We will often see $H^*(x)$ make an appearence in the the SDNF of sentences where a model requires that a literal be assigned specifically either the value $Both$ or the classical $True$, rather than accepting either equally. For example, consider the presence of the following in an SDNF:

$$H(x) \wedge \neg H^*(x).$$

One can check that this expression will only be satisfied by a valuation ν in the event that $\nu(x) = Both$.

LP Energy Function

We now assume we have an SDNF representation of an *LP* WFF φ and convert it into an energy function.

We again recall our mapping of a sentence presented in SDNF to an RBM for the propositional logic case:

Theorem 2.3.6. *Any SDNF* $\varphi \equiv \bigvee_j \left(\bigwedge_{t \in S_{T_j}} x_t \wedge \bigwedge_{k \in S_{K_j}} \neg x_k \right)$ *can be mapped onto an equivalent RBM with energy function*

$$E = -\sum_j h_j \left(\sum_{t \in S_{T_j}} x_t - \sum_{k \in S_{K_j}} x_k - T_j + \epsilon \right),$$

where $0 < \epsilon < 1$ *and* S_{T_j} *and* S_{K_j} *are respectively the set of* T_j *indices of positive literals and the set of* K_j *indices of negative literals.*

Just as before, we must transform this process of valuation to align with the designated values for *LP* . We use our *LP* Designated Values Function, Def. 3.3.7, and are now prepared to define our energy function for the *LP* context.

Theorem 3.3.11. *Any* LP^S *SDNF*

$$\varphi \equiv \bigvee_j \left(\bigwedge_{t \in S_{T_j}} x_t \wedge \bigwedge_{u \in S_{U_j}} H^* (x_u) \wedge \bigwedge_{k \in S_{L_j}} \neg H (x_l) \wedge \bigwedge_{l \in S_{K_j}} \neg x_k \right)$$

can be mapped onto an equivalent RBM with energy function

$$E = -\sum_j h_j \left(\sum_{t \in S_{T_j}} D_v^{LP} (x_t) + \sum_{u \in S_{U_j}} D_v^{K3} (x_u) \right.$$
$$\left. - \sum_{k \in S_{K_j}} D_v^{LP} (x_k) - \sum_{l \in S_{L_j}} D_v^{K3} (x_l) - T_j - U_j + \epsilon \right),$$

where $0 < \epsilon < 1$ *and* S_{T_j} *and* S_{K_j} *are respectively the set of* T_j *indices of positive literals and the set of* K_j *indices of negative literals acted upon by* H*, and* S_{U_j} *and* S_{L_j} *are respectively the set of* U_j *indices of literals acted upon by* H^* *and the set of* L_j *indices of literals acted upon by* $\neg H$*.*

The reader may wish to convince themselves that the SDNF general form does in fact capture the full range of expressions which may appear in the LP^S SDNF by seeing that other representations can be reduced to those included in the general SDNF expression by employing the identities described above.

We have again expanded the domain of values to include the tertiary $Both$ value and map it to the same value as $True$, putting the two designated values on equal footing while leaving the $False$ value to behave normally. Both count towards satisfying the positive literals in $\sum_{t \in S_{T_j}} D_v^{LP}(x_t) + \sum_{u \in S_{U_j}} D_v^{K3}(x_u)$, and both count against satisfying the negative literals in $-\sum_{k \in S_{K_j}} D_v^{LP}(x_k) - \sum_{l \in S_{L_j}} D_v^{K3}(x_l)$.

We note that the use of the term $D_v^{K3}(x)$ may seem somewhat odd, as ν is an *LP* valuation, and we are applying the *K3* valuation function. Note, however, that the values in both *LP* and *K3* are the set $\{-1, 0, 1\}$. As such, the values for each logic are the domain of the designated value functions, so the function $D_v^{K3}(x)$ can be applied to values of *LP*; it simply returns 1 only in the case that $\nu(x) = 1$.

As in the previous section, we omit explicit proof of this theorem and instead will further explore and confirm the validity of this defintion in Sect. 3.4.

§ 3.4 Analysis of Paraconsistent Logics in RBMs

We now explore our defined methods of converting paraconsistent logical sentences into RBM form in order to ensure that the representations behave as expected for each logic. To do this, we will follow the same process executed in Sect. 2.4, in which we represent the antecedents for various logical properties as an RBM-encoded knowledge base and ensure that the consequences are as expected.

§ 3.4.1 *K3*

We first attend to the *K3* case.

Transitivity

We recall Defn. 2.4.1:

Definition 2.4.1. *The logical implication $\rightarrow$ is said to be transitive if and only if:*

$$\mathcal{K} \vDash (P \rightarrow Q) \wedge (Q \rightarrow R) \Rightarrow \mathcal{K} \vDash P \rightarrow R$$

for any knowledge base $\mathcal{K}$ and sentences P, Q, and R. This rule can be expressed syntactically as

$$\frac{P \rightarrow Q,\ Q \rightarrow R}{P \rightarrow R}$$

Theorem 3.4.2. *When two K3 implications are encoded into an RBM, the property of Transitivity holds.*

Justification. We will follow the same method used to prove Thm. 2.4.2, i.e., define a knowledge base with two implications, represent it using an RBM and energy function, and show that for all models which result in minimal energy, Transitivity holds.

We define our knowledge base $\mathcal{K}$:

$$\mathcal{K} \equiv (P \rightarrow Q) \wedge (Q \rightarrow R)\,. \tag{3.4.14}$$

We then convert $\mathcal{K}$ into SDNF:

$$\mathcal{K} \equiv (P \wedge Q \wedge R) \vee (\neg P \wedge Q \wedge R) \vee (\neg P \wedge \neg Q)\,. \tag{3.4.15}$$

Using our new *K3* context energy function in Thm. 3.3.6, we transform (3.4.15) to an energy function.

$$\begin{aligned} E = &-h_1 \left(D_v^{K3}(P) + D_v^{K3}(Q) + D_v^{K3}(R) - 2.5\right) \\ &- h_2 \left(-D_v^{K3}(P) + D_v^{K3}(Q) + D_v^{K3}(R) - 1.5\right) \\ &- h_3 \left(-D_v^{K3}(P) - D_v^{K3}(Q) + 0.5\right) \end{aligned} \tag{3.4.16}$$

We now consider the truth value assignments $\boldsymbol{x}_i$ which minimize (3.4.16) in Table 3.11.

Table 3.11: The *K3* Transitivity energy function (3.4.16) for each valuation $\boldsymbol{x}_i$.

K3 Transitivity Energy Function							
$\boldsymbol{x}_i$	P	Q	R	Energy Function	Minimized Energy	Model of $\mathcal{K}$	Transitivity $(P \to R)$
$\boldsymbol{x}_1$	-1	-1	-1	$2.5h_1 + 1.5h_2 - 0.5h_3$	$\mathbf{-0.5}$	**Yes**	**Yes**
$\boldsymbol{x}_2$	-1	-1	0	$2.5h_1 + 1.5h_2 - 0.5h_3$	$\mathbf{-0.5}$	**Yes**	**Yes**
$\boldsymbol{x}_3$	-1	-1	1	$1.5h_1 + 0.5h_2 - 0.5h_3$	$\mathbf{-0.5}$	**Yes**	**Yes**
$\boldsymbol{x}_4$	-1	0	-1	$2.5h_1 + 1.5h_2 - 0.5h_3$	$\mathbf{-0.5}$	**Yes**	**Yes**
$\boldsymbol{x}_5$	-1	0	0	$2.5h_1 + 1.5h_2 - 0.5h_3$	$\mathbf{-0.5}$	**Yes**	**Yes**
$\boldsymbol{x}_6$	-1	0	1	$1.5h_1 + 0.5h_2 - 0.5h_3$	$\mathbf{-0.5}$	No	**Yes**
$\boldsymbol{x}_7$	-1	1	-1	$1.5h_1 + 0.5h_2 + 0.5h_3$	0.0	No	**Yes**
$\boldsymbol{x}_8$	-1	1	0	$1.5h_1 + 0.5h_2 + 0.5h_3$	0.0	No	**Yes**
$\boldsymbol{x}_9$	-1	1	1	$0.5h_1 - 0.5h_2 + 0.5h_3$	$\mathbf{-0.5}$	**Yes**	**Yes**
$\boldsymbol{x}_{10}$	0	-1	-1	$2.5h_1 + 1.5h_2 - 0.5h_3$	$\mathbf{-0.5}$	**Yes**	**Yes**
$\boldsymbol{x}_{11}$	0	-1	0	$2.5h_1 + 1.5h_2 - 0.5h_3$	$\mathbf{-0.5}$	**Yes**	**Yes**
$\boldsymbol{x}_{12}$	0	-1	1	$1.5h_1 + 0.5h_2 - 0.5h_3$	$\mathbf{-0.5}$	**Yes**	**Yes**
$\boldsymbol{x}_{13}$	0	0	-1	$2.5h_1 + 1.5h_2 - 0.5h_3$	$\mathbf{-0.5}$	**Yes**	**Yes**
$\boldsymbol{x}_{14}$	0	0	0	$2.5h_1 + 1.5h_2 - 0.5h_3$	$\mathbf{-0.5}$	**Yes**	**Yes**
$\boldsymbol{x}_{15}$	0	0	1	$1.5h_1 + 0.5h_2 - 0.5h_3$	$\mathbf{-0.5}$	**Yes**	**Yes**
$\boldsymbol{x}_{16}$	0	1	-1	$1.5h_1 + 0.5h_2 + 0.5h_3$	0.0	No	**Yes**
$\boldsymbol{x}_{17}$	0	1	0	$1.5h_1 + 0.5h_2 + 0.5h_3$	0.0	No	**Yes**
$\boldsymbol{x}_{18}$	0	1	1	$0.5h_1 - 0.5h_2 + 0.5h_3$	$\mathbf{-0.5}$	**Yes**	**Yes**
$\boldsymbol{x}_{19}$	1	-1	-1	$1.5h_1 + 2.5h_2 + 0.5h_3$	0.0	No	No
$\boldsymbol{x}_{20}$	1	-1	0	$1.5h_1 + 2.5h_2 + 0.5h_3$	0.0	No	No
$\boldsymbol{x}_{21}$	1	-1	1	$0.5h_1 + 1.5h_2 + 0.5h_3$	0.0	No	**Yes**
$\boldsymbol{x}_{22}$	1	0	-1	$1.5h_1 + 2.5h_2 + 0.5h_3$	0.0	No	No
$\boldsymbol{x}_{23}$	1	0	0	$1.5h_1 + 2.5h_2 + 0.5h_3$	0.0	No	No
$\boldsymbol{x}_{24}$	1	0	1	$0.5h_1 + 1.5h_2 + 0.5h_3$	0.0	No	**Yes**
$\boldsymbol{x}_{25}$	1	1	-1	$0.5h_1 + 2.5h_2 + 1.5h_3$	0.0	No	No
$\boldsymbol{x}_{26}$	1	1	0	$0.5h_1 + 2.5h_2 + 1.5h_3$	0.0	No	No
$\boldsymbol{x}_{27}$	1	1	1	$-0.5h_1 + 0.5h_2 + 1.5h_3$	$\mathbf{-0.5}$	**Yes**	**Yes**

Observing the possible valuations $\boldsymbol{x}_i$, we recognize that the set of valuations which minimize the energy function (to $-\epsilon$) is exactly the set of valuations which satisfy $\mathcal{K}$. Further, this set is a subset of those models for which $(P \to R)$ holds. We have therefore shown that Transitivity holds in each of the models identified by this method.

■

Ex Falso Quodlibet

We recall Defn. 2.4.3:

Definition 2.4.3. *The rule of Ex Falso Quodlibet holds in a logic if and only if:*

$$\mathcal{K} \models (P \wedge \neg P) \Rightarrow \mathcal{K} \models Q$$

for any knowledge base $\mathcal{K}$ and sentences P and Q. This rule can be expressed syntactically as

$$\frac{P,\ \neg P}{Q}$$

A motivation for the development of the *K3* system was to be able to work with sentences that can neither be proven $True$ nor $False$ without reducing the system into trivial nonsense. It is crucial that our method reproduce this quality as well by failing to satisfy *Ex Falso* as a rule.

Further, as *K3* is inspired by the intuitionistic approach, one should expect that contradictory sentences are never derived. As such, the system should not accept any positive, contradictory valuation as a model. The value of the knowledge base could be at best underdetermined or $Neither$, and there would therefore be no models.

Theorem 3.4.4. *When a K3 knowledge base is encoded into an RBM, the formalism will degenerate into triviality, identifying all possible valuations as preferred valuations, yet none will be models.*

Justification. We must show that if contradictory literals P and $\neg P$ are satisfied in the same knowledge base, there will be no selected models of the knowledge base. We follow the same procedure from Thm. 2.4.4.

We first define a contradictory knowledge base:

$$\mathcal{K} \equiv P \wedge \neg P \wedge R. \tag{3.4.17}$$

Because there exist no tautologies in the *K3* system, we are unable to take our previous approach of introducing a tautology to the knowledge base in order to include an otherwise undefined atom to our RBM. While less elegant, one can instead introduce an additional visible node to the RBM which does not factor into the value of the energy function. We will see that consequently the additional node is not influenced by the contradiction into exemplifying *Ex Falso Quodlibet*. We are still able to include the atom R to evaluate the entailment of non-contradictory subsets of the knowledge base while failing to entail the knowledge base in total.

We convert (3.4.17) into SDNF:

$$\mathcal{K} \equiv P \wedge \neg P \wedge R, \tag{3.4.18}$$

and then define our energy function:

$$E = -h_1 \left(D_v^{K3}(P) - D_v^{K3}(P) + D_v^{K3}(R) - 1.5 \right), \tag{3.4.19}$$

which can be simplified to the representation:

$$E = -h_1 \left(D_v^{K3}(R) - 1.5 \right). \tag{3.4.20}$$

We now consider all possible truth valuations $\boldsymbol{x}_i$ and identify those that minimize (3.4.20). As the energy function is a function of only the atom R, much of this table will be redundant. However, this redundancy reinforces that no possible valuations over the literals will satisfy the knowledge base.

Studying Table 3.12, we note that this is analgous to the behavior of the classical RBM to a contradiction studied in Sect. 2.4.2, i.e., all valuations are preferred yet none are models. We similarly interpret these results as the formalism degenerating to triviality, while reiterating the formalism's inability to encode the contradictory values *a priori*. While there is a tertiary value, it encodes that the sentence is $Neither\ True\ nor\ False$, not $Both$.

■

Table 3.12: The *K3 Ex Falso* energy function (3.4.20) for each valuation $\boldsymbol{x}_i$.

K3 Ex Falso Quodlibet Energy Function							
$\boldsymbol{x}_i$	P	Q	R	Energy Function	Minimized Energy	Model of $\mathcal{K}$	*Ex Falso* (Q)
$\boldsymbol{x}_1$	-1	-1	-1	$1.5h_1$	0.0	No	No
$\boldsymbol{x}_2$	-1	-1	0	$1.5h_1$	0.0	No	No
$\boldsymbol{x}_3$	-1	-1	1	$0.5h_1$	0.0	No	No
$\boldsymbol{x}_4$	-1	0	-1	$1.5h_1$	0.0	No	No
$\boldsymbol{x}_5$	-1	0	0	$1.5h_1$	0.0	No	No
$\boldsymbol{x}_6$	-1	0	1	$0.5h_1$	0.0	No	No
$\boldsymbol{x}_7$	-1	1	-1	$1.5h_1$	0.0	No	**Yes**
$\boldsymbol{x}_8$	-1	1	0	$1.5h_1$	0.0	No	**Yes**
$\boldsymbol{x}_9$	-1	1	1	$0.5h_1$	0.0	No	**Yes**
$\boldsymbol{x}_{10}$	0	-1	-1	$1.5h_1$	0.0	No	No
$\boldsymbol{x}_{11}$	0	-1	0	$1.5h_1$	0.0	No	No
$\boldsymbol{x}_{12}$	0	-1	1	$0.5h_1$	0.0	No	No
$\boldsymbol{x}_{13}$	0	0	-1	$1.5h_1$	0.0	No	No
$\boldsymbol{x}_{14}$	0	0	0	$1.5h_1$	0.0	No	No
$\boldsymbol{x}_{15}$	0	0	1	$0.5h_1$	0.0	No	No
$\boldsymbol{x}_{16}$	0	1	-1	$1.5h_1$	0.0	No	**Yes**
$\boldsymbol{x}_{17}$	0	1	0	$1.5h_1$	0.0	No	**Yes**
$\boldsymbol{x}_{18}$	0	1	1	$0.5h_1$	0.0	No	**Yes**
$\boldsymbol{x}_{19}$	1	-1	-1	$1.5h_1$	0.0	No	No
$\boldsymbol{x}_{20}$	1	-1	0	$1.5h_1$	0.0	No	No
$\boldsymbol{x}_{21}$	1	-1	1	$0.5h_1$	0.0	No	No
$\boldsymbol{x}_{22}$	1	0	-1	$1.5h_1$	0.0	No	No
$\boldsymbol{x}_{23}$	1	0	0	$1.5h_1$	0.0	No	No
$\boldsymbol{x}_{24}$	1	0	1	$0.5h_1$	0.0	No	No
$\boldsymbol{x}_{25}$	1	1	-1	$1.5h_1$	0.0	No	**Yes**
$\boldsymbol{x}_{26}$	1	1	0	$1.5h_1$	0.0	No	**Yes**
$\boldsymbol{x}_{27}$	1	1	1	$0.5h_1$	0.0	No	**Yes**

Disjunctive Syllogism

We recall Defn. 2.4.5:

Definition 2.4.5. *The rule of Disjunctive Syllogism holds in a logic if and only if*

$$\mathcal{K} \vDash (P \vee Q) \wedge \neg P \Rightarrow \mathcal{K} \vDash Q$$

for any knowledge base $\mathcal{K}$ and sentences P and Q. This rule can be expressed syntactically as

$$\frac{(P \vee Q)\,,\ \neg P}{Q}.$$

We note that by Kleene's defintion of the disjunction in *K3*, Disjunctive Syllogism does hold, and we seek now to show that this property does hold in our logic [6].

Theorem 3.4.6. *When an K3 knowledge base is encoded into an RBM, Disjunctive Syllogism holds.*

Justification. We must show that when both statements $(P \vee Q)$ and $\neg P$ are satisfied in the same knowledge base, the sentence Q is also satisfied. We follow the same procedure from Thm. 2.4.6.

We first define a knowledge base to represent our situation:

$$\mathcal{K} \equiv (P \vee Q) \wedge \neg P. \tag{3.4.21}$$

We then express (3.4.21) in SDNF:

$$\mathcal{K} \equiv (P \wedge \neg P) \vee (Q \wedge \neg P)\,. \tag{3.4.22}$$

and define our energy function:

$$E = -h_1 \left(D_v^{K3}(P) - D_v^{K3}(P) - 0.5\right) - h_2 \left(D_v^{K3}(Q) - D_v^{K3}(P) - 0.5\right), \tag{3.4.23}$$

which can be simplified to the expression:

$$E = -h_1 (-0.5) - h_2 \left(D_v^{K3}(Q) - D_v^{K3}(P) - 0.5\right), \tag{3.4.24}$$

We now consider all possible truth valuations $\boldsymbol{x}_i$ and identify those that minimize (3.4.24).

Table 3.13: The *K3* Disjunctive Syllogism energy function (3.4.24) for each valuation $\boldsymbol{x}_i$.

K3 Disjunctive Syllogism Energy Function						
$\boldsymbol{x}_i$	P	Q	Energy Function	Minimized Energy	Model of $\mathcal{K}$	Disjunctive Syllogism (Q)
$\boldsymbol{x}_1$	-1	-1	$0.5h_1 + 0.5h_2$	0.0	No	No
$\boldsymbol{x}_2$	-1	0	$0.5h_1 + 0.5h_2$	0.0	No	No
$\boldsymbol{x}_3$	-1	1	$0.5h_1 - 0.5h_2$	$\mathbf{-0.5}$	**Yes**	**Yes**
$\boldsymbol{x}_4$	0	-1	$0.5h_1 + 0.5h_2$	0.0	No	No
$\boldsymbol{x}_5$	0	0	$0.5h_1 + 0.5h_2$	0.0	No	No
$\boldsymbol{x}_6$	0	1	$0.5h_1 - 0.5h_2$	$\mathbf{-0.5}$	**Yes**	**Yes**
$\boldsymbol{x}_7$	1	-1	$0.5h_1 + 0.5h_2$	0.0	No	No
$\boldsymbol{x}_8$	1	0	$0.5h_1 + 0.5h_2$	0.0	No	No
$\boldsymbol{x}_9$	1	1	$0.5h_1 + 0.5h_2$	0.0	No	**Yes**

Studying Table 3.13, we see that the subset of valuations $\boldsymbol{x}_i$ which are selected by the minimization process is the same subset of valuations which model the knowledge base. Futher, in each of these models, Q is assigned the value $True$, and as such, Disjunctive Syllogism holds.

■

Resolution

We recall the defintion of Resolution.

Definition [. *2.4.7] The generalized Resolution rule can be stated as*

$$\frac{l_1 \vee \ldots \vee l_k, \qquad m_1 \vee \ldots \vee m_n}{l_1 \vee \ldots \vee l_{i-1} \vee l_{i+1} \vee \ldots \vee l_k \vee m_1 \vee \ldots \vee m_{j-1} \vee m_{j+1} \vee \ldots \vee m_n},$$

where l_i and m_j are complementary literals, i.e. $l_i \equiv \neg m_j$ [11].

We again will prove that Resolution holds only in the case of short resolvents and claim that the argument will generalize for longer resolvents as well.

Theorem 3.4.8. *In the K3 context of RBM Logic, the rule of Resolution holds for resolvents of the form* $(P \vee Q) \wedge (\neg P \vee R)$. *That is:*

$$\mathcal{K} \vDash (P \vee Q) \wedge (\neg P \vee R) \Rightarrow \mathcal{K} \vDash (Q \vee R)$$

Justification. We begin our proof by defining the relevant knowledge base,

$$\mathcal{K} \equiv (P \vee Q) \wedge (\neg P \vee R)\,, \tag{3.4.25}$$

and presenting $\mathcal{K}$ in SDNF:

$$\mathcal{K} \equiv (\neg P \wedge Q \wedge \neg R) \vee (\neg P \wedge Q \wedge R) \vee (P \wedge \neg Q \wedge R) \vee (P \wedge Q \wedge R)\,. \tag{3.4.26}$$

Using Thm. 3.3.6, we transform (3.4.26) into an energy function:

$$\begin{aligned} E = &-h_1\left(-D_v^{K3}(P) + D_v^{K3}(Q) - D_v^{K3}(R) - 0.5\right) \\ &- h_2\left(-D_v^{K3}(P) + D_v^{K3}(Q) + D_v^{K3}(R) - 1.5\right) \\ &- h_3\left(D_v^{K3}(P) - D_v^{K3}(Q) + D_v^{K3}(R) - 1.5\right) \\ &- h_4\left(D_v^{K3}(P) + D_v^{K3}(Q) + D_v^{K3}(R) - 2.5\right). \end{aligned} \tag{3.4.27}$$

We now calculate the value of (3.4.27) for all possible valuations over the atoms.

From Table 3.14, we see that the valuations which minimize the energy are those which model $\mathcal{K}$, and also that these models are a subset of those valuations which entail $Q \vee R$, i.e., which entail Resolution. ■

§ 3.4.2 Resolution Refutation

We recall our defintion of Resolution Refutation in the context of RBM Logic.

Table 3.14: The *K3* Resolution energy function (3.4.27) for each valuation $\boldsymbol{x}_i$.

K3 Resolution Energy Function							
$\boldsymbol{x}_i$	P	Q	R	Energy Function	Minimized Energy	Model of $\mathcal{K}$	Resolution $(Q \vee R)$
$\boldsymbol{x}_1$	-1	-1	-1	$0.5h_1 + 1.5h_2 + 1.5h_3 + 2.5h4$	0.0	No	No
$\boldsymbol{x}_2$	-1	-1	0	$0.5h_1 + 1.5h_2 + 1.5h_3 + 2.5h4$	0.0	No	No
$\boldsymbol{x}_3$	-1	-1	1	$1.5h_1 + 0.5h_2 + 0.5h_3 + 1.5h4$	0.0	No	**Yes**
$\boldsymbol{x}_4$	-1	0	-1	$0.5h_1 + 1.5h_2 + 1.5h_3 + 2.5h4$	0.0	No	No
$\boldsymbol{x}_5$	-1	0	0	$0.5h_1 + 1.5h_2 + 1.5h_3 + 2.5h4$	0.0	No	No
$\boldsymbol{x}_6$	-1	0	1	$1.5h_1 + 0.5h_2 + 0.5h_3 + 1.5h4$	0.0	No	**Yes**
$\boldsymbol{x}_7$	-1	1	-1	$-0.5h_1 + 0.5h_2 + 2.5h_3 + 1.5h4$	$\mathbf{-0.5}$	**Yes**	**Yes**
$\boldsymbol{x}_8$	-1	1	0	$-0.5h_1 + 0.5h_2 + 2.5h_3 + 1.5h4$	$\mathbf{-0.5}$	**Yes**	**Yes**
$\boldsymbol{x}_9$	-1	1	1	$0.5h_1 - 0.5h_2 + 1.5h_3 + 0.5h4$	$\mathbf{-0.5}$	**Yes**	**Yes**
$\boldsymbol{x}_{10}$	0	-1	-1	$0.5h_1 + 1.5h_2 + 1.5h_3 + 2.5h4$	0.0	No	No
$\boldsymbol{x}_{11}$	0	-1	0	$0.5h_1 + 1.5h_2 + 1.5h_3 + 2.5h4$	0.0	No	No
$\boldsymbol{x}_{12}$	0	-1	1	$1.5h_1 + 0.5h_2 + 0.5h_3 + 1.5h4$	0.0	No	**Yes**
$\boldsymbol{x}_{13}$	0	0	-1	$0.5h_1 + 1.5h_2 + 1.5h_3 + 2.5h4$	0.0	No	No
$\boldsymbol{x}_{14}$	0	0	0	$0.5h_1 + 1.5h_2 + 1.5h_3 + 2.5h4$	0.0	No	No
$\boldsymbol{x}_{15}$	0	0	1	$1.5h_1 + 0.5h_2 + 0.5h_3 + 1.5h4$	0.0	No	**Yes**
$\boldsymbol{x}_{16}$	0	1	-1	$-0.5h_1 + 0.5h_2 + 2.5h_3 + 1.5h4$	$\mathbf{-0.5}$	**Yes**	**Yes**
$\boldsymbol{x}_{17}$	0	1	0	$-0.5h_1 + 0.5h_2 + 2.5h_3 + 1.5h4$	$\mathbf{-0.5}$	**Yes**	**Yes**
$\boldsymbol{x}_{18}$	0	1	1	$0.5h_1 - 0.5h_2 + 1.5h_3 + 0.5h4$	$\mathbf{-0.5}$	**Yes**	**Yes**
$\boldsymbol{x}_{19}$	1	-1	-1	$1.5h_1 + 2.5h_2 + 0.5h_3 + 1.5h4$	0.0	No	No
$\boldsymbol{x}_{20}$	1	-1	0	$1.5h_1 + 2.5h_2 + 0.5h_3 + 1.5h4$	0.0	No	No
$\boldsymbol{x}_{21}$	1	-1	1	$2.5h_1 + 1.5h_2 - 0.5h_3 + 0.5h4$	$\mathbf{-0.5}$	**Yes**	**Yes**
$\boldsymbol{x}_{22}$	1	0	-1	$1.5h_1 + 2.5h_2 + 0.5h_3 + 1.5h4$	0.0	No	No
$\boldsymbol{x}_{23}$	1	0	0	$1.5h_1 + 2.5h_2 + 0.5h_3 + 1.5h4$	0.0	No	No
$\boldsymbol{x}_{24}$	1	0	1	$2.5h_1 + 1.5h_2 - 0.5h_3 + 0.5h4$	$\mathbf{-0.5}$	**Yes**	**Yes**
$\boldsymbol{x}_{25}$	1	1	-1	$0.5h_1 + 1.5h_2 + 1.5h_3 + 0.5h4$	0.0	No	**Yes**
$\boldsymbol{x}_{26}$	1	1	0	$0.5h_1 + 1.5h_2 + 1.5h_3 + 0.5h4$	0.0	No	**Yes**
$\boldsymbol{x}_{27}$	1	1	1	$1.5h_1 + 0.5h_2 + 0.5h_3 - 0.5h4$	$\mathbf{-0.5}$	**Yes**	**Yes**

Definition 2.4.9. *Resolution Refutation in the RBM Logic will be defined as the process of adding the query Q to the knowledge base $\mathcal{K}$, then creating and minimizing the resulting energy function based upon $\mathcal{K} \cup \{Q\}$.*

We now present our theorem and prove it using the same method as Thm. 2.4.10.

Theorem 3.4.10. *Given a knowledge base $\mathcal{K}$ and a query Q, the RBM Logic in the K3 context will prefer no valuations if $\mathcal{K} \cup \{Q\}$ is inconsistent and will prefer models in which $Q \equiv True$ if $\mathcal{K} \cup \{Q\}$ is consistent.*

Justification.

Claim 1: The RBM Logic will prefer no models if $\mathcal{K} \cup \{Q\}$ is inconsistent

Subproof. We recall the system's response to an inconsistent knowledge base explored in Thm. 3.4.4, i.e., the system's inability to select for any valuation as a model of the knowledge base. Invoking that property here, we see that this claim is trivial. □

Claim 2: The RBM Logic will prefer valuations $\boldsymbol{x}_i$ in which $Q \equiv_{\boldsymbol{x}_i} True$ if $\mathcal{K} \cup \{Q\}$ is consistent.

Subproof. We define a simple knowledge base:

$$\mathcal{K} \equiv P \wedge (P \rightarrow Q)\,, \tag{3.4.28}$$

and offer the sentence Q as our query:

$$\mathcal{K}' \equiv P \wedge (P \rightarrow Q) \wedge Q,$$

We convert $\mathcal{K}'$ into SDNF:

$$\mathcal{K}' \equiv (P \wedge \neg P \wedge Q) \vee (P \wedge Q \wedge Q)\,. \tag{3.4.29}$$

and define our energy function to represent (3.4.29)

$$\begin{aligned} E = -h_1 \left(D_v^{K3}(P) - D_v^{K3}(P) + D_v^{K3}(Q) - 2 + 0.5\right) \\ - h_2 \left(D_v^{K3}(P) + D_v^{K3}(Q) + D_v^{K3}(Q) - 3 + 0.5\right), \end{aligned}$$

which simplifies to

$$E = -h_1 \left(D_v^{K3}(Q) - 1.5 \right) - h_2 \left(D_v^{K3}(P) + 2D_v^{K3}(Q) - 2.5 \right). \tag{3.4.30}$$

We now consider possible evaluations $\boldsymbol{x}_i$ and identify those which minimize energy function (3.4.30). We see then that $\boldsymbol{x}_9$ is the sole valuation which minimizes the energy and the only valuation which serves

Table 3.15: The *K3* Resolution Refutation energy function (3.4.30) for each valuation $\boldsymbol{x}_i$.

K3 Resolution Refutation Energy Function						
$\boldsymbol{x}_i$	P	Q	Energy Function	Minimized Energy	Model of $\mathcal{K}$	Model of $\mathcal{K} \cup Q$)
$\boldsymbol{x}_1$	-1	-1	$1.5h_1 + 2.5h_2$	0.0	No	No
$\boldsymbol{x}_2$	-1	0	$1.5h_1 + 2.5h_2$	0.0	No	No
$\boldsymbol{x}_3$	-1	1	$0.5h_1 + 0.5h_2$	0.0	No	No
$\boldsymbol{x}_4$	0	-1	$1.5h_1 + 2.5h_2$	0.0	No	No
$\boldsymbol{x}_5$	0	0	$1.5h_1 + 2.5h_2$	0.0	No	No
$\boldsymbol{x}_6$	0	1	$0.5h_1 + 0.5h_2$	0.0	No	No
$\boldsymbol{x}_7$	1	-1	$1.5h_1 + 1.5h_2$	0.0	No	No
$\boldsymbol{x}_8$	1	0	$1.5h_1 + 1.5h_2$	0.0	No	No
$\boldsymbol{x}_9$	1	1	$0.5h_1 - 0.5h_2$	$\mathbf{-0.5}$	**Yes**	**Yes**

as a model for both $\mathcal{K}$ and $\mathcal{K} \cup \{Q\}$. □

■

§ 3.4.3 LP

We next attend to the *LP* case.

Transitivity

See Sect. 3.4.1 for the recollection of Transitivity.

Theorem 3.4.11. *When two LP implications are encoded into an RBM, the property of Transitivity does not hold.*

Proof. We will follow the same method used to prove Thm. 3.4.2, i.e., define a knowledge base with two implications and represent it using an RBM and energy function . We then show that there exists some model which results in minimal energy for which Transitivity fails to hold.

We define our knowledge base,

$$\mathcal{K} \equiv (P \rightarrow Q) \wedge (Q \rightarrow R)\,. \tag{3.4.31}$$

We then convert $\mathcal{K}$ into SDNF,

$$\begin{aligned}\mathcal{K} \equiv (\neg P \wedge \neg Q) \vee (\neg P \wedge H^{*}(Q) \wedge R) \\ \vee (H^{*}(P) \wedge Q \wedge \neg Q) \vee (H^{*}(P) \wedge Q \wedge H^{*}(Q) \wedge R)\,.\end{aligned} \tag{3.4.32}$$

Using our new *LP* context energy function in Thm. 3.3.11, we transform (3.4.32) to an energy function.

$$\begin{aligned}E = -h_1\left(-D_v^{K3}(P) - D_v^{K3}(Q) + 0.5\right) \\ - h_2\left(-D_v^{K3}(P) + D_v^{K3}(Q) + D_v^{LP}(R) - 1.5\right) \\ - h_3\left(D_v^{K3}(P) + D_v^{LP}(Q) - D_v^{K3}(Q) - 1.5\right) \\ - h_4\left(D_v^{K3}(P) + D_v^{LP}(Q) + D_v^{K3}(Q) + D_v^{LP}(R) - 3.5\right)\,.\end{aligned} \tag{3.4.33}$$

We now consider the truth value assignments $\boldsymbol{x}_i$ which minimize (3.4.33) in Table 3.16.

Observing the possible valuations $\boldsymbol{x}_i$, we direct the reader towards valuation $\boldsymbol{x}_{22}$. We note that the minimzed energy is -0.5, which is both $-\epsilon$ and the minimal energy over all valuations. As such, it is a preferred valuation and also a model of $\mathcal{K}$. However, $(P \rightarrow Q)$ fails to hold, and we have therefore identified a model of $\mathcal{K}$ for which Transitivity fails.

■

Table 3.16: The *LP* Transitivity energy function (3.4.33) for each valuation $\boldsymbol{x}_i$.

LP Transitivity Energy Function							
$\boldsymbol{x}_i$	P	Q	R	Energy Function	Minimized Energy	Model of $\mathcal{K}$	Transitivity $(P \to R)$
$\boldsymbol{x}_1$	-1	-1	-1	$-0.5h_1 + 1.5h_2 + 1.5h_3 + 3.5h_4$	$\mathbf{-0.5}$	**Yes**	**Yes**
$\boldsymbol{x}_2$	-1	-1	0	$-0.5h_1 + 0.5h_2 + 1.5h_3 + 2.5h_4$	$\mathbf{-0.5}$	**Yes**	**Yes**
$\boldsymbol{x}_3$	-1	-1	1	$-0.5h_1 + 0.5h_2 + 1.5h_3 + 2.5h_4$	$\mathbf{-0.5}$	**Yes**	**Yes**
$\boldsymbol{x}_4$	-1	0	-1	$-0.5h_1 + 1.5h_2 + 0.5h_3 + 2.5h_4$	$\mathbf{-0.5}$	**Yes**	**Yes**
$\boldsymbol{x}_5$	-1	0	0	$-0.5h_1 + 0.5h_2 + 0.5h_3 + 1.5h_4$	$\mathbf{-0.5}$	**Yes**	**Yes**
$\boldsymbol{x}_6$	-1	0	1	$-0.5h_1 + 0.5h_2 + 0.5h_3 + 1.5h_4$	$\mathbf{-0.5}$	**Yes**	**Yes**
$\boldsymbol{x}_7$	-1	1	-1	$0.5h_1 + 0.5h_2 + 1.5h_3 + 1.5h_4$	0.0	No	**Yes**
$\boldsymbol{x}_8$	-1	1	0	$0.5h_1 - 0.5h_2 + 1.5h_3 + 0.5h_4$	$\mathbf{-0.5}$	**Yes**	**Yes**
$\boldsymbol{x}_9$	-1	1	1	$0.5h_1 - 0.5h_2 + 1.5h_3 + 0.5h_4$	$\mathbf{-0.5}$	**Yes**	**Yes**
$\boldsymbol{x}_{10}$	0	-1	-1	$-0.5h_1 + 1.5h_2 + 1.5h_3 + 3.5h_4$	$\mathbf{-0.5}$	**Yes**	**Yes**
$\boldsymbol{x}_{11}$	0	-1	0	$-0.5h_1 + 0.5h_2 + 1.5h_3 + 2.5h_4$	$\mathbf{-0.5}$	**Yes**	**Yes**
$\boldsymbol{x}_{12}$	0	-1	1	$-0.5h_1 + 0.5h_2 + 1.5h_3 + 2.5h_4$	$\mathbf{-0.5}$	**Yes**	**Yes**
$\boldsymbol{x}_{13}$	0	0	-1	$-0.5h_1 + 1.5h_2 + 0.5h_3 + 2.5h_4$	$\mathbf{-0.5}$	**Yes**	**Yes**
$\boldsymbol{x}_{14}$	0	0	0	$-0.5h_1 + 0.5h_2 + 0.5h_3 + 1.5h_4$	$\mathbf{-0.5}$	**Yes**	**Yes**
$\boldsymbol{x}_{15}$	0	0	1	$-0.5h_1 + 0.5h_2 + 0.5h_3 + 1.5h_4$	$\mathbf{-0.5}$	**Yes**	**Yes**
$\boldsymbol{x}_{16}$	0	1	-1	$0.5h_1 + 0.5h_2 + 1.5h_3 + 1.5h_4$	0.0	No	**Yes**
$\boldsymbol{x}_{17}$	0	1	0	$0.5h_1 - 0.5h_2 + 1.5h_3 + 0.5h_4$	$\mathbf{-0.5}$	**Yes**	**Yes**
$\boldsymbol{x}_{18}$	0	1	1	$0.5h_1 - 0.5h_2 + 1.5h_3 + 0.5h_4$	$\mathbf{-0.5}$	**Yes**	**Yes**
$\boldsymbol{x}_{19}$	1	-1	-1	$0.5h_1 + 2.5h_2 + 0.5h_3 + 2.5h_4$	0.0	No	No
$\boldsymbol{x}_{20}$	1	-1	0	$0.5h_1 + 1.5h_2 + 0.5h_3 + 1.5h_4$	0.0	No	**Yes**
$\boldsymbol{x}_{21}$	1	-1	1	$0.5h_1 + 1.5h_2 + 0.5h_3 + 1.5h_4$	0.0	No	**Yes**
$\boldsymbol{x}_{22}$	1	0	-1	$0.5h_1 + 2.5h_2 - 0.5h_3 + 1.5h_4$	$\mathbf{-0.5}$	**Yes**	No
$\boldsymbol{x}_{23}$	1	0	0	$0.5h_1 + 1.5h_2 - 0.5h_3 + 0.5h_4$	$\mathbf{-0.5}$	**Yes**	**Yes**
$\boldsymbol{x}_{24}$	1	0	1	$0.5h_1 + 1.5h_2 - 0.5h_3 + 0.5h_4$	$\mathbf{-0.5}$	**Yes**	**Yes**
$\boldsymbol{x}_{25}$	1	1	-1	$1.5h_1 + 1.5h_2 + 0.5h_3 + 0.5h_4$	0.0	No	No
$\boldsymbol{x}_{26}$	1	1	0	$1.5h_1 + 0.5h_2 + 0.5h_3 - 0.5h_4$	$\mathbf{-0.5}$	**Yes**	**Yes**
$\boldsymbol{x}_{27}$	1	1	1	$1.5h_1 + 0.5h_2 + 0.5h_3 - 0.5h_4$	$\mathbf{-0.5}$	**Yes**	**Yes**

Ex Falso Quodlibet

See Sect. 3.4.1 for our recollection of the property *Ex Falso Quodlibet*.

Similarly to *K3* , a motivation for *LP* is to be able to work with sentences which have been identified as both $True$ and $False$, without the reduction to triviality. This is a critical property for modeling *LP* using RBMs.

Theorem 3.4.12. *When an LP knowledge base is encoded into an RBM, Ex Falso Quodlibet does not hold.*

Proof. We must show that if contradictory literals P and $\neg P$ are satisfied in the same knowledge base, one is not able to derive an arbitrary sentence Q. We follow the same procedure from Thm. 2.4.4.

We first define a contradictory knowledge base:

$$\mathcal{K} \equiv P \wedge \neg P \wedge R \wedge (Q \vee \neg Q)\,. \tag{3.4.34}$$

We again include the literal R to explore the systems response to literals of our knowledge base well-founded despite the contradiction. We include the tautology $Q \vee \neg Q$ to explicitly include Q as literal of concern and a visible node in our RBM in order to analyze the systems response to otherwise unfounded literals.

We then express (3.4.34) in SDNF:

$$\mathcal{K} \equiv (P \wedge \neg P \wedge R \wedge Q) \vee (P \wedge \neg P \wedge R \wedge \neg H\,(Q) \wedge \neg Q) \tag{3.4.35}$$

and define our energy function:

$$\begin{aligned} E = &-h_1 \left(D_v^{LP}\,(P) - D_v^{K3}\,(P) + D_v^{LP}\,(R) + D_v^{LP}\,(Q) - 2.5\right) \\ &- h_2 \left(D_v^{LP}\,(P) - D_v^{K3}\,(P) + D_v^{LP}\,(R) - D_v^{LP}\,(Q) - D_v^{K3}\,(Q) - 1.5\right), \end{aligned} \tag{3.4.36}$$

We now consider all possible truth valuations $\boldsymbol{x}_i$ and identify those that minimize (3.4.36).

Studying Table 3.17, we see that those valuations $\boldsymbol{x}_i$ which minimize the energy function are not only those same valuations which model (3.4.34), but also that in each of these models, R must have the value 0 or 1, restricting it to designated values, while Q can take on the values -1, 0, or 1, i.e., we have maintained the knowledge that we have outside of our contradiction, while not restricting the values of other possible sentences. Thus, *Ex Falso Quodlibet* does not hold for *LP* in RBM Logic. Further, we do not have the same degeneracy witnessed in the classical and *K3* cases; the property fails in a robust and nondegenerate sense.

Table 3.17: The *LP Ex Falso* energy function (3.4.36) for each valuation $\boldsymbol{x}_i$.

LP Ex Falso Quodlibet Energy Function							
$\boldsymbol{x}_i$	P	Q	R	Energy Function	Min. Energy	Model of $\mathcal{K}$	*Ex Falso* (Q)
$\boldsymbol{x}_1$	-1	-1	-1	$2.5h_1 + 1.5h_2$	0.0	No	No
$\boldsymbol{x}_2$	-1	-1	0	$1.5h_1 + 0.5h_2$	0.0	No	No
$\boldsymbol{x}_3$	-1	-1	1	$1.5h_1 + 0.5h_2$	0.0	No	No
$\boldsymbol{x}_4$	-1	0	-1	$1.5h_1 + 2.5h_2$	0.0	No	**Yes**
$\boldsymbol{x}_5$	-1	0	0	$0.5h_1 + 1.5h_2$	0.0	No	**Yes**
$\boldsymbol{x}_6$	-1	0	1	$0.5h_1 + 1.5h_2$	0.0	No	**Yes**
$\boldsymbol{x}_7$	-1	1	-1	$1.5h_1 + 3.5h_2$	0.0	No	**Yes**
$\boldsymbol{x}_8$	-1	1	0	$0.5h_1 + 2.5h_2$	0.0	No	**Yes**
$\boldsymbol{x}_9$	-1	1	1	$0.5h_1 + 2.5h_2$	0.0	No	**Yes**
$\boldsymbol{x}_{10}$	0	-1	-1	$1.5h_1 + 0.5h_2$	0.0	No	No
$\boldsymbol{x}_{11}$	0	-1	0	$0.5h_1 - 0.5h_2$	$\mathbf{-0.5}$	**Yes**	No
$\boldsymbol{x}_{12}$	0	-1	1	$0.5h_1 - 0.5h_2$	$\mathbf{-0.5}$	**Yes**	No
$\boldsymbol{x}_{13}$	0	0	-1	$0.5h_1 + 1.5h_2$	0.0	No	**Yes**
$\boldsymbol{x}_{14}$	0	0	0	$-0.5h_1 + 0.5h_2$	$\mathbf{-0.5}$	**Yes**	**Yes**
$\boldsymbol{x}_{15}$	0	0	1	$-0.5h_1 + 0.5h_2$	$\mathbf{-0.5}$	**Yes**	**Yes**
$\boldsymbol{x}_{16}$	0	1	-1	$0.5h_1 + 2.5h_2$	0.0	No	**Yes**
$\boldsymbol{x}_{17}$	0	1	0	$-0.5h_1 + 0.5h_2$	$\mathbf{-0.5}$	**Yes**	**Yes**
$\boldsymbol{x}_{18}$	0	1	1	$-0.5h_1 + 0.5h_2$	$\mathbf{-0.5}$	**Yes**	**Yes**
$\boldsymbol{x}_{19}$	1	-1	-1	$2.5h_1 + 1.5h_2$	0.0	No	No
$\boldsymbol{x}_{20}$	1	-1	0	$1.5h_1 + 0.5h_2$	0.0	No	No
$\boldsymbol{x}_{21}$	1	-1	1	$1.5h_1 + 0.5h_2$	0.0	No	No
$\boldsymbol{x}_{22}$	1	0	-1	$1.5h_1 + 2.5h_2$	0.0	No	**Yes**
$\boldsymbol{x}_{23}$	1	0	0	$0.5h_1 + 1.5h_2$	0.0	No	**Yes**
$\boldsymbol{x}_{24}$	1	0	1	$0.5h_1 + 1.5h_2$	0.0	No	**Yes**
$\boldsymbol{x}_{25}$	1	1	-1	$1.5h_1 + 3.5h_2$	0.0	No	**Yes**
$\boldsymbol{x}_{26}$	1	1	0	$0.5h_1 + 1.5h_2$	0.0	No	**Yes**
$\boldsymbol{x}_{27}$	1	1	1	$0.5h_1 + 1.5h_2$	0.0	No	**Yes**

■

Disjunctive Syllogism

See Sect. 3.4.1 for our recollection of Disjunctive Syllogism.

Priest has claimed that this property does not hold in *LP* [8]. We seek now to prove that this property does not hold in our logic.

Theorem 3.4.13. *When an LP knowledge base is encoded into an RBM, Disjunctive Syllogism does not hold.*

Proof. We must show that when both statements $(P \vee Q)$ and $\neg P$ are satisfied in the same knowledge base, one is not able to derive the sentence Q. We follow the same procedure from Thm. 2.4.6.

We first define a knowledge base to represent our situation:

$$\mathcal{K} \equiv (P \vee Q) \wedge \neg P. \tag{3.4.37}$$

We then express (3.4.37) in SDNF:

$$\mathcal{K} \equiv (P \wedge \neg P) \vee (\neg H(P) \wedge Q \wedge \neg P). \tag{3.4.38}$$

and define our energy function:

$$E = -h_1 \left(D_v^{LP}(P) - D_v^{K3}(P) - 0.5\right) - h_2 \left(-D_v^{LP}(P) + D_v^{LP}(Q) - D_v^{K3}(P) - 0.5\right), \tag{3.4.39}$$

We now consider all possible truth valuations $\boldsymbol{x}_i$ and identify those that minimize (3.4.39).

Table 3.18: The *LP* Disjunctive Syllogism energy function (3.4.39) for each valuation $\boldsymbol{x}_i$.

LP Disjunctive Syllogism Energy Function						
$\boldsymbol{x}_i$	P	Q	Energy Function	Minimized Energy	Model of $\mathcal{K}$	Disjunctive Syllogism (Q)
$\boldsymbol{x}_1$	-1	-1	$0.5h_1 + 0.5h_2$	0.0	No	No
$\boldsymbol{x}_2$	-1	0	$0.5h_1 - 0.5h_2$	$\mathbf{-0.5}$	**Yes**	**Yes**
$\boldsymbol{x}_3$	-1	1	$0.5h_1 - 0.5h_2$	$\mathbf{-0.5}$	**Yes**	**Yes**
$\boldsymbol{x}_4$	0	-1	$-0.5h_1 + 1.5h_2$	$\mathbf{-0.5}$	**Yes**	No
$\boldsymbol{x}_5$	0	0	$-0.5h_1 + 0.5h_2$	$\mathbf{-0.5}$	**Yes**	**Yes**
$\boldsymbol{x}_6$	0	1	$-0.5h_1 + 0.5h_2$	$\mathbf{-0.5}$	**Yes**	**Yes**
$\boldsymbol{x}_7$	1	-1	$0.5h_1 + 2.5h_2$	0.0	No	No
$\boldsymbol{x}_8$	1	0	$0.5h_1 + 1.5h_2$	0.0	No	**Yes**
$\boldsymbol{x}_9$	1	1	$0.5h_1 + 1.5h_2$	0.0	No	**Yes**

Studying Table 3.18, we see that valuations $\boldsymbol{x}_4$ is a model of 3.4.37 which is selected by the minimization process, however this valuation does not entail Q, and as such the rule of Disjunctive Syllogism does not hold for all models of our knowledge base.

■

Resolution and Resolution Refutation

Because the properties of Defn. 2.4.7 Resolution and Defn. 2.4.9 Resolution Refutation require as a foundation the rule of Disjunctive Syllogism to hold, it is trivial to show that they will not hold in the RBM Logic representation of *LP*.

§ 3.5 Chapter Conclusion

We have seen a viable extension of the classical RBM Logic case into the two paraconsistent logics *K3* and *LP*. These extensions required not only that we introduce additional possible values for each of the nodes, but also additional designated values which could satisfy clauses and knowledge bases in the case of *LP*.

We further identified that one may have to extend an arbitrary language in order to represent sentences of that language in SDNF. We have presented an approach that could serve as inspiration for a generalized method for addressing this problem in other languages, i.e., introducing additional connectives that allow one to express the satisfiability of a sentence.

We note that while *K3* maintains the properties that were explored in Sect. 2.4, the encoding of *LP* —faithful to its original presentation—loses the properties of Transitivity, Disjunctive Syllogism, Resolution, and Resolution Refutation. As these properties are all central to our ability to use logic for deductive purposes, especially in a computational context, one hopes that the introduction of a robust response to contradiction could maintain these properties. In the next chapter, we seek to restore these deductive qualities within our formalism.

CHAPTER 4

MINIMALLY INCONSISTENT LOGIC

§ 4.1 Chapter Introduction

While the paraconsistent logic *LP* is interesting in its own right and enables one to consider paraconsistent knowledge bases, its loss of classical results, most notably Disjunctive Syllogism, is unappealing. As such, Priest introduced his Minimally Inconsistent Logic of Paradox LP_m to help address this issue. In short, LP_m imposes an ordering onto the valuations which corresponds to their degree of inconsistency and accepts as models only those valuations which are minimally inconsistent [9]. In this language, so long as a knowledge base can be satisfied fully within the domain of consistency, the langauge acts exactly as classical logic; the language otherwise prefers models which minimize the inconsistencies (and therefore deductive casualites).

§ 4.2 LP_m : Minimally Inconsistent LP

As we are concerned with only the propositional subset of LP_m and have elsewhere used a notation that differs from Priest's, we here use Priest's presentation as a guide for our own presentation of LP_m , rather than following it exactly. The structure and consequential properties of the logic will remain. We also note that we will only define the notion of inconsistency as it relates to the atomic sentences, rather than those of higher rank. In future work, the author hopes to expand this method of RBM Logic representation in such a way that sentences of higher rank can act functionally as atomic sentences at higher levels of abstraction, and as such we feel this will not be a limitation on the system, though additional work may be required to formalize the more general approach.

Given the language *LP* , we extend the langauge to LP_m via the following.

We first define the notion of an *atomic subformula*. Although the general definition of this concept requires more explication, the following will suffice for this thesis.

Definition 4.2.1. *A sentence Ψ is called an atomic subformula of the sentence φ if and only if Ψ occurs in the sentence φ and Ψ does not itself contain any connectives. We will call φ_A the set of atomic subformulas of the sentence φ.*

We define a new helper function over a sentence φ and an *LP* valuation ν:

Definition 4.2.2.

$$B_\nu(\varphi) \equiv \begin{cases} 1 & \textit{if } \nu(\varphi) = 0 \\ 0 & \textit{else} \end{cases} \tag{4.2.1}$$

This equation has the property of evaluating whether the sentence has been assigned the paraconsistent value $Both$ in the given valuation.

Now, given a knowledge base $\mathcal{K}$ and associated valuation ν, we define:

Definition 4.2.3.

$$\mathbb{B}(\langle \mathcal{K}, \nu \rangle) \equiv \sum_{\alpha \in K_A} B_\nu(\alpha), \tag{4.2.2}$$

where K is the conjunction which encodes the knowledge base $\mathcal{K}$ into a single sentence.

We note that this helper function has the property of counting the total number of atoms α in the sentence K which are assigned the $Both$ or inconsistent value by the given valuation. When applied to a knowledge base and valuation for a given interpretation, this function will allow the ordering of models. If there are no atoms which are shown to be inconsistent, then $\mathbb{B}(\langle \mathcal{K}, \nu \rangle)$ will have its minimal value of 0. If every atom is inconsistent, $\mathbb{B}(\langle \mathcal{K}, \nu \rangle)$ will return a maximal value for the knowledge base.

We now refine our definition of a model to incorporate our ordering by inconsistency:

Definition 4.2.4. *$\mathfrak{A} = \langle \mathcal{K}, \nu \rangle$ is a minimally inconsistent (mi) model of Σ ($\mathfrak{A} \models_m \Sigma$) iff $\mathfrak{A} \models \Sigma$ and if $\mathbb{B}(\mathfrak{A}') < \mathbb{B}(\mathfrak{A})$ and $\mathcal{K} = \mathcal{K}'$, then $\mathfrak{A}' \not\models \Sigma$.*

We further refine our notion of satisfaction:

Definition 4.2.5. *A knowledge base $\mathcal{K}$ is satisfied by a valuation ν if and only if $\mathfrak{A} = \langle \mathcal{K}, \nu \rangle$ is an mi model of $\mathcal{K}$.*

§ 4.3 Minimally Inconsistent *LP* in RBMs

We now turn our attention to encoding this new definition of satisfaction into our RBM Logic energy function. To do so, we must introduce a penalty on those valuations which increase the inconsistency of the model.

§ 4.3.1 Extension from *LP* in RBMs

We note that the function Defn. 4.2.3 functions as a count of inconsistency for a given $\mathcal{K}$ and ν. We must now modify this measure so that it can be added to our energy function such that it penalizes non-minimal models without increasing the energy so much that the valuation is no longer recognized as a model.

We note that in non-degenerate cases, models have their energy function minimized to a value of $-\epsilon$, while non-models receive an energy value of 0.0. Therefore, we must define our penalty $p_{\mathcal{K}}(\nu)$ for a given valuation ν on a knowledge base $\mathcal{K}$ such that $p_{\mathcal{K}}(\nu) < \epsilon$. We propose:

Definition 4.3.1. *The penalty for inconsistency $p_{\mathcal{K}}(\nu)$ for a valuation will be defined:*

$$p_{\mathcal{K}}(\nu) \equiv \frac{\mathbb{B}(\langle \mathcal{K}, \nu \rangle)}{|K_A| + 1}\epsilon, \tag{4.3.3}$$

where K_A represents the set of atomic formulas in the finite conjunction that represents the knoweledge base $\mathcal{K}$.

Consider a maximally inconsistent interpretation of $\mathcal{K}$, such that every atom is overdetermined. In this case, $\mathbb{B}(\langle \mathcal{K}, \nu \rangle) = |K_A|$, and thus $p_{\mathcal{K}}(\nu) = \frac{|K_A|}{|K_A|+1}\epsilon$. While close to ϵ (closer than any non-maximally

inconsistent interpretation), this added penalty is not greater than ϵ and the energy for a model penalized by this amount would still be negative. In the case of a consistent interpretation, $\mathbb{B}\left(\langle \mathcal{K}, \nu \rangle\right) = 0 = p_{\mathcal{K}}\left(\nu\right)$, and there is no penalty, reducing to our classical case, as desired.

We note that SDNFs can be handled in this context in exactly the same way as in Sect. 3.3.2, extending the language LP_m to a new language LP_m^S . We also note that an adjustment must be made to Defn. 2.3.1 regarding the equivalence of an SDNF and an RBM to properly handle this case.

Definition 4.3.2. *A WFF φ is equivalent to an RBM $\mathcal{N}$ if and only if for any truth assignment over the visible nodes $\boldsymbol{x}$, $s_\varphi\left(\boldsymbol{x}\right) = -AE_{rank}\left(\boldsymbol{x}\right) + B$, where $s_\varphi\left(\boldsymbol{x}\right) \in (-1, 1]$ is the inconsistency-penalized truth value of φ given $\boldsymbol{x}$ with True $\equiv 1$ and False $\equiv 0$, and each truth value is reduced by $p_\varphi\left(\boldsymbol{x}\right)$; $A > 0$ and B are constants; $E_{rank}\left(\boldsymbol{x}\right) = \min_{\boldsymbol{h}} E\left(\boldsymbol{x}, \boldsymbol{h}\right)$ is the energy ranking function of $\mathcal{N}$ minimised over all hidden units.*

The increased range of $s_\varphi\left(\boldsymbol{x}\right)$ allows us to capture both whether the valuation ν is a (non-minimally inconsistent) model of φ and also how inconsistent of a valuation it is. $s_\varphi\left(\boldsymbol{x}\right) > 0$ corresponds to (non-mi) models, while $s_\varphi\left(\boldsymbol{x}\right) < 0$ corresponds to non-models. $s_\varphi\left(\boldsymbol{x}\right) = 1$ corresponds to fully consistent models, while $s_\varphi\left(\boldsymbol{x}\right) \approx -1$ corresponds to fully inconsistent models.

We are now prepared to present our LP_m energy function defintion.

Theorem 4.3.3. *Any LP_m^S SDNF*

$$\varphi \equiv \bigvee_j \left(\bigwedge_{t \in S_{T_j}} x_t \wedge \bigwedge_{u \in S_{U_j}} H^*\left(x_u\right) \wedge \bigwedge_{k \in S_{L_j}} \neg H\left(x_l\right) \wedge \bigwedge_{l \in S_{K_j}} \neg x_k \right)$$

can be mapped onto an equivalent RBM with energy function

$$E = -\sum_j h_j \left(\sum_{t \in S_{T_j}} D_v^{LP}(x_t) + \sum_{u \in S_{U_j}} D_v^{K3}(x_u) - \sum_{k \in S_{K_j}} D_v^{LP}(x_k) - \sum_{l \in S_{L_j}} D_v^{K3}(x_l) - T_j - U_j + \epsilon \right) + p_\varphi(\nu),$$

where $0 < \epsilon < 1$ *and* S_{T_j} *and* S_{K_j} *are respectively the set of* T_j *indices of positive literals and the set of* K_j *indices of negative literals, and* S_{U_j} *and* S_{L_j} *are respectively the set of* U_j *indices of literals acted upon by* H^* *and the set of* L_j *indices of literals acted upon by* $\neg H$.

As before, we omit explicit proof and explore the behavior of this formalism below.

§ 4.4 Analysis of Minimally Inconsistent LP in RBMs

We now explore the RBM Logic encoding of LP_m by analyzing the same properties earlier explored.

§ 4.4.1 Transitivity

We recall Defn. 2.4.1:

Definition 2.4.1. *The logical implication* $\rightarrow$ *is said to be transitive if and only if:*

$$\mathcal{K} \vDash (P \rightarrow Q) \wedge (Q \rightarrow R) \Rightarrow \mathcal{K} \vDash P \rightarrow R$$

for any knowledge base $\mathcal{K}$ *and sentences* P, Q, *and* R. *This rule can be expressed syntactically as*

$$\frac{P \rightarrow Q,\ Q \rightarrow R}{P \rightarrow R}$$

Theorem 4.4.2. *When two* LP_m *implications are encoded into an RBM, the property of Transitivity holds.*

Justification. We will follow the same method used to prove Thm. 2.4.2, i.e., define a knowledge base with two implications, represent it using an RBM and energy function, and show that for all models which

result in a negative minimized energy, Transitivity holds. Further, we will see that those valuations with minimal energy are the minimally inconsistent models.

We recall our *LP* Transitivity SDNF:

$$\mathcal{K} \equiv (\neg P \wedge \neg Q) \vee (\neg P \wedge H^*(Q) \wedge R) \vee (H^*(P) \wedge Q \wedge \neg Q) \vee (H^*(P) \wedge Q \wedge H^*(Q) \wedge R). \tag{3.4.32}$$

and transform our SDNF into an energy function,

$$\begin{aligned} E = -h_1 \left(-D_v^{K3}(P) - D_v^{K3}(Q) + 0.5\right) \\ - h_2 \left(-D_v^{K3}(P) + D_v^{K3}(Q) + D_v^{LP}(R) - 1.5\right) \\ - h_3 \left(D_v^{K3}(P) + D_v^{LP}(Q) - D_v^{K3}(Q) - 1.5\right) \\ - h_4 \left(D_v^{K3}(P) + D_v^{LP}(Q) + D_v^{K3}(Q) + D_v^{LP}(R) - 3.5\right) + p_{\mathcal{K}}(\nu). \end{aligned} \tag{4.4.5}$$

We now consider the truth value assignments $\boldsymbol{x}_i$ which minimize (4.4.5).

Observing the possible valuations $\boldsymbol{x}_i$ in Table 4.1, we identify that -0.5 (our chosen ϵ) is the minimum energy value for any valuation. We consider the set of valuations which minimize the function to this value and recognize that this is exactly the set of minimally consistent models. This set is a subset of models of KB, which all have a negative minimized energy value. This is further a subset of those valuations for which $(P \rightarrow Q)$ holds. We have therefore shown that this method picks out the minimally inconsistent models of the knowledge base and that Transitivity holds in each of these models.

■

§ 4.4.2 *Ex Falso Quodlibet*

We recall Defn. 2.4.3:

Table 4.1: The LP_m Transitivity energy function (4.4.5) for each valuation $\boldsymbol{x}_i$.

LP_m Transitivity Energy Function								
$\boldsymbol{x}_i$	P	Q	R	Energy Function	Min. Energy	mi Model	Model of $\mathcal{K}$	Trans. $(P \to R)$
$\boldsymbol{x}_1$	-1	-1	-1	$-0.5h_1 + 1.5h_2 + 1.5h_3 + 3.5h_4$	$\mathbf{-0.5}$	**Yes**	**Yes**	**Yes**
$\boldsymbol{x}_2$	-1	-1	0	$-0.5h_1 + 0.5h_2 + 1.5h_3 + 2.5h_4 + \frac{1}{8}$	-0.375	No	**Yes**	**Yes**
$\boldsymbol{x}_3$	-1	-1	1	$-0.5h_1 + 0.5h_2 + 1.5h_3 + 2.5h_4$	$\mathbf{-0.5}$	**Yes**	**Yes**	**Yes**
$\boldsymbol{x}_4$	-1	0	-1	$-0.5h_1 + 1.5h_2 + 0.5h_3 + 2.5h_4 + \frac{1}{8}$	-0.375	No	**Yes**	**Yes**
$\boldsymbol{x}_5$	-1	0	0	$-0.5h_1 + 0.5h_2 + 0.5h_3 + 1.5h_4 + \frac{2}{8}$	-0.250	No	**Yes**	**Yes**
$\boldsymbol{x}_6$	-1	0	1	$-0.5h_1 + 0.5h_2 + 0.5h_3 + 1.5h_4 + \frac{1}{8}$	-0.375	No	**Yes**	**Yes**
$\boldsymbol{x}_7$	-1	1	-1	$0.5h_1 + 0.5h_2 + 1.5h_3 + 1.5h_4$	0.000	No	No	**Yes**
$\boldsymbol{x}_8$	-1	1	0	$0.5h_1 - 0.5h_2 + 1.5h_3 + 0.5h_4 + \frac{1}{8}$	-0.375	No	**Yes**	**Yes**
$\boldsymbol{x}_9$	-1	1	1	$0.5h_1 - 0.5h_2 + 1.5h_3 + 0.5h_4$	$\mathbf{-0.5}$	**Yes**	**Yes**	**Yes**
$\boldsymbol{x}_{10}$	0	-1	-1	$-0.5h_1 + 1.5h_2 + 1.5h_3 + 3.5h_4 + \frac{1}{8}$	-0.375	No	**Yes**	**Yes**
$\boldsymbol{x}_{11}$	0	-1	0	$-0.5h_1 + 0.5h_2 + 1.5h_3 + 2.5h_4 + \frac{2}{8}$	-0.250	No	**Yes**	**Yes**
$\boldsymbol{x}_{12}$	0	-1	1	$-0.5h_1 + 0.5h_2 + 1.5h_3 + 2.5h_4 + \frac{1}{8}$	-0.375	No	**Yes**	**Yes**
$\boldsymbol{x}_{13}$	0	0	-1	$-0.5h_1 + 1.5h_2 + 0.5h_3 + 2.5h_4 + \frac{2}{8}$	-0.250	No	**Yes**	**Yes**
$\boldsymbol{x}_{14}$	0	0	0	$-0.5h_1 + 0.5h_2 + 0.5h_3 + 1.5h_4 + \frac{3}{8}$	-0.125	No	**Yes**	**Yes**
$\boldsymbol{x}_{15}$	0	0	1	$-0.5h_1 + 0.5h_2 + 0.5h_3 + 1.5h_4 + \frac{2}{8}$	-0.25	No	**Yes**	**Yes**
$\boldsymbol{x}_{16}$	0	1	-1	$0.5h_1 + 0.5h_2 + 1.5h_3 + 1.5h_4 + \frac{1}{8}$	0.125	No	No	**Yes**
$\boldsymbol{x}_{17}$	0	1	0	$0.5h_1 - 0.5h_2 + 1.5h_3 + 0.5h_4 + \frac{2}{8}$	-0.250	No	**Yes**	**Yes**
$\boldsymbol{x}_{18}$	0	1	1	$0.5h_1 - 0.5h_2 + 1.5h_3 + 0.5h_4 + \frac{1}{8}$	-0.375	No	**Yes**	**Yes**
$\boldsymbol{x}_{19}$	1	-1	-1	$0.5h_1 + 2.5h_2 + 0.5h_3 + 2.5h_4$	0.000	No	No	No
$\boldsymbol{x}_{20}$	1	-1	0	$0.5h_1 + 1.5h_2 + 0.5h_3 + 1.5h_4 + \frac{1}{8}$	0.125	No	No	**Yes**
$\boldsymbol{x}_{21}$	1	-1	1	$0.5h_1 + 1.5h_2 + 0.5h_3 + 1.5h_4$	0.000	No	No	**Yes**
$\boldsymbol{x}_{22}$	1	0	-1	$0.5h_1 + 2.5h_2 - 0.5h_3 + 1.5h_4 + \frac{1}{8}$	-0.375	No	**Yes**	No
$\boldsymbol{x}_{23}$	1	0	0	$0.5h_1 + 1.5h_2 - 0.5h_3 + 0.5h_4 + \frac{2}{8}$	-0.250	No	**Yes**	**Yes**
$\boldsymbol{x}_{24}$	1	0	1	$0.5h_1 + 1.5h_2 - 0.5h_3 + 0.5h_4 + \frac{1}{8}$	-0.375	No	**Yes**	**Yes**
$\boldsymbol{x}_{25}$	1	1	-1	$1.5h_1 + 1.5h_2 + 0.5h_3 + 0.5h_4$	0.000	No	No	No
$\boldsymbol{x}_{26}$	1	1	0	$1.5h_1 + 0.5h_2 + 0.5h_3 - 0.5h_4 + \frac{1}{8}$	-0.375	No	**Yes**	**Yes**
$\boldsymbol{x}_{27}$	1	1	1	$1.5h_1 + 0.5h_2 + 0.5h_3 - 0.5h_4$	$\mathbf{-0.5}$	**Yes**	**Yes**	**Yes**

Definition 2.4.3. *The rule of Ex Falso Quodlibet holds in a logic if and only if:*

$$\mathcal{K} \vDash (P \wedge \neg P) \Rightarrow \mathcal{K} \vDash Q$$

for any knowledge base $\mathcal{K}$ and sentences P and Q. This rule can be expressed syntactically as

$$\frac{P,\ \neg P}{Q}$$

We note that an important characteristic of LP_m is that it does not hold. This is the critical property that contradiction does not reduce the entire set of sentences to trivially $True$.

Theorem 4.4.4. *When an LP_m knowledge base is encoded into an RBM, Ex Falso Quodlibet does not hold.*

Proof. We must show that if contradictory literals P and $\neg P$ are satisfied in the same knowledge base, one is not able to derive an arbitrary sentence Q. We follow the same procedure from Thm. 2.4.4.

We recall our *Ex Falso LP* SDNF

$$\mathcal{K} \equiv (P \wedge \neg P \wedge R \wedge Q) \vee (P \wedge \neg P \wedge R \wedge \neg H(Q) \wedge \neg Q) \tag{3.4.35}$$

and define our energy function:

$$\begin{aligned} E = &-h_1 \left(D_v^{LP}(P) - D_v^{K3}(P) + D_v^{LP}(R) + D_v^{LP}(Q) - 2.5 \right) \\ &- h_2 \left(D_v^{LP}(P) - D_v^{K3}(P) + D_v^{LP}(R) - D_v^{LP}(Q) - D_v^{K3}(Q) - 1.5 \right) + p_{\mathcal{K}}(\nu), \end{aligned} \tag{4.4.7}$$

We now consider all possible truth valuations $\boldsymbol{x}_i$ and identify those that minimize (4.4.7).

Studying Table 4.2, we see that those valuations $\boldsymbol{x}_i$ which minimize the energy function are the minimally inconsistent valuations which model (3.4.34). In each of these models, R must have the value 1, restricting it not only to designated values, but to the minimally inconsistent designated value. Q can take

Table 4.2: The LP_m *Ex Falso* energy function (4.4.7) for each valuation $\boldsymbol{x}_i$.

LP_m *Ex Falso Quodlibet* Energy Function								
$\boldsymbol{x}_i$	P	Q	R	Energy Function	Min. Energy	mi Model of $\mathcal{K}$	Model of $\mathcal{K}$	*Ex Falso* (Q)
$\boldsymbol{x}_1$	-1	-1	-1	$2.5h_1 + 1.5h_2$	0.000	No	No	No
$\boldsymbol{x}_2$	-1	-1	0	$1.5h_1 + 0.5h_2 + \frac{1}{8}$	0.125	No	No	No
$\boldsymbol{x}_3$	-1	-1	1	$1.5h_1 + 0.5h_2$	0.000	No	No	No
$\boldsymbol{x}_4$	-1	0	-1	$1.5h_1 + 2.5h_2 + \frac{1}{8}$	0.125	No	No	**Yes**
$\boldsymbol{x}_5$	-1	0	0	$0.5h_1 + 1.5h_2 + \frac{2}{8}$	0.25	No	No	**Yes**
$\boldsymbol{x}_6$	-1	0	1	$0.5h_1 + 1.5h_2 + \frac{1}{8}$	0.125	No	No	**Yes**
$\boldsymbol{x}_7$	-1	1	-1	$1.5h_1 + 3.5h_2$	0.000	No	No	**Yes**
$\boldsymbol{x}_8$	-1	1	0	$0.5h_1 + 2.5h_2 + \frac{1}{8}$	0.125	No	No	**Yes**
$\boldsymbol{x}_9$	-1	1	1	$0.5h_1 + 2.5h_2$	0.000	No	No	**Yes**
$\boldsymbol{x}_{10}$	0	-1	-1	$1.5h_1 + 0.5h_2 + \frac{1}{8}$	0.125	No	No	No
$\boldsymbol{x}_{11}$	0	-1	0	$0.5h_1 - 0.5h_2 + \frac{2}{8}$	-0.250	No	**Yes**	No
$\boldsymbol{x}_{12}$	0	-1	1	$0.5h_1 - 0.5h_2 + \frac{1}{8}$	$\mathbf{-0.375}$	**Yes**	**Yes**	No
$\boldsymbol{x}_{13}$	0	0	-1	$0.5h_1 + 1.5h_2 + \frac{2}{8}$	0.250	No	No	**Yes**
$\boldsymbol{x}_{14}$	0	0	0	$-0.5h_1 + 0.5h_2 + \frac{3}{8}$	-0.125	No	**Yes**	**Yes**
$\boldsymbol{x}_{15}$	0	0	1	$-0.5h_1 + 0.5h_2 + \frac{2}{8}$	-0.250	No	**Yes**	**Yes**
$\boldsymbol{x}_{16}$	0	1	-1	$0.5h_1 + 2.5h_2 + \frac{1}{8}$	0.125	No	No	**Yes**
$\boldsymbol{x}_{17}$	0	1	0	$-0.5h_1 + 0.5h_2 + \frac{2}{8}$	-0.250	No	**Yes**	**Yes**
$\boldsymbol{x}_{18}$	0	1	1	$-0.5h_1 + 0.5h_2 + \frac{1}{8}$	$\mathbf{-0.375}$	**Yes**	**Yes**	**Yes**
$\boldsymbol{x}_{19}$	1	-1	-1	$2.5h_1 + 1.5h_2$	0.000	No	No	No
$\boldsymbol{x}_{20}$	1	-1	0	$1.5h_1 + 0.5h_2 + \frac{1}{8}$	0.125	No	No	No
$\boldsymbol{x}_{21}$	1	-1	1	$1.5h_1 + 0.5h_2$	0.000	No	No	No
$\boldsymbol{x}_{22}$	1	0	-1	$1.5h_1 + 2.5h_2 + \frac{1}{8}$	0.125	No	No	**Yes**
$\boldsymbol{x}_{23}$	1	0	0	$0.5h_1 + 1.5h_2 + \frac{2}{8}$	0.250	No	No	**Yes**
$\boldsymbol{x}_{24}$	1	0	1	$0.5h_1 + 1.5h_2 + \frac{1}{8}$	0.125	No	No	**Yes**
$\boldsymbol{x}_{25}$	1	1	-1	$1.5h_1 + 3.5h_2$	0.000	No	No	**Yes**
$\boldsymbol{x}_{26}$	1	1	0	$0.5h_1 + 1.5h_2 + \frac{1}{8}$	0.125	No	No	**Yes**
$\boldsymbol{x}_{27}$	1	1	1	$0.5h_1 + 1.5h_2$	0.000	No	No	**Yes**

on the values -1 or 1, i.e., its value is not determined by our knowledge base, but a consistent value for the atom is prefered. We have maintained the knowledge of R that we have outside of our contradiction, while not deciding the values of other atoms, i.e., Q. Thus, *Ex Falso Quodlibet* does not hold for LP_m in RBM Logic, and it fails in the same robust way as the *LP* case.

■

§ 4.4.3 Disjunctive Syllogism

We recall Defn. 2.4.5:

Definition 2.4.5. *The rule of Disjunctive Syllogism holds in a logic if and only if*

$$\mathcal{K} \vDash (P \vee Q) \wedge \neg P \Rightarrow \mathcal{K} \vDash Q$$

for any knowledge base $\mathcal{K}$ and sentences P and Q. This rule can be expressed syntactically as

$$\frac{(P \vee Q)\,,\ \neg P}{Q}.$$

Priest's motivation for defining LP_m is at least in part to maintain the property of Disjunctive Syllogism wherever possible, i.e., in those cases in which overdetermination is not necessary, one can still rely on Disjunctive Syllogism for inference. We now show that the same is true in the LP_m context for RBM Logic.

Theorem 4.4.6. *When an LP_m knowledge base is encoded into an RBM, Disjunctive Syllogism does hold.*

Justification. We must show that when both statements $(P \vee Q)$ and $\neg P$ are satisfied in the same knowledge base, one is able to derive the sentence Q in the minimally inconsistent models. We follow the same procedure from Thm. 2.4.6.

We recall our Disjunctive Syllogism *LP* SDNF

$$\mathcal{K} \equiv (P \wedge \neg P) \vee (\neg H\,(P) \wedge Q \wedge \neg P) \tag{3.4.38}$$

and define our energy function:

$$E = -h_1\left(D_v^{LP}(P) - D_v^{K3}(P) - 0.5\right) - h_2\left(-D_v^{LP}(P) + D_v^{LP}(Q) - D_v^{K3}(P) - 0.5\right) + p_{\mathcal{K}}(\nu), \tag{4.4.9}$$

We now consider all possible truth valuations $\boldsymbol{x}_i$ and identify those that minimize (4.4.9).

Table 4.3: The LP_m Disjunctive Syllogism energy function (4.4.9) for each valuation $\boldsymbol{x}_i$.

LP_m Disjunctive Syllogism Energy Function							
$\boldsymbol{x}_i$	P	Q	Energy Function	Minimized Energy	mi Model of $\mathcal{K}$	Model of $\mathcal{K}$	Disjunctive Syllogism (Q)
$\boldsymbol{x}_1$	-1	-1	$0.5h_1 + 0.5h_2$	0.000	No	No	No
$\boldsymbol{x}_2$	-1	0	$0.5h_1 - 0.5h_2 + \frac{1}{6}$	-0.333	No	**Yes**	**Yes**
$\boldsymbol{x}_3$	-1	1	$0.5h_1 - 0.5h_2$	$\mathbf{-0.500}$	**Yes**	**Yes**	**Yes**
$\boldsymbol{x}_4$	0	-1	$-0.5h_1 + 1.5h_2 + \frac{1}{6}$	-0.333	No	**Yes**	No
$\boldsymbol{x}_5$	0	0	$-0.5h_1 + 0.5h_2 + \frac{2}{6}$	-0.166	No	**Yes**	**Yes**
$\boldsymbol{x}_6$	0	1	$-0.5h_1 + 0.5h_2 + \frac{1}{6}$	-0.333	No	**Yes**	**Yes**
$\boldsymbol{x}_7$	1	-1	$0.5h_1 + 2.5h_2$	0.000	No	No	No
$\boldsymbol{x}_8$	1	0	$0.5h_1 + 1.5h_2 + \frac{1}{6}$	0.166	No	No	**Yes**
$\boldsymbol{x}_9$	1	1	$0.5h_1 + 1.5h_2$	0.000	No	No	**Yes**

Studying Table 4.3, we see that valuation $\boldsymbol{x}_3$ is the minimally inconsistent model of 3.4.37 which is selected by the minimization process. This model does entail Q, and as such the rule of Disjunctive Syllogism holds for the minimally inconsistent models of our knowledge base.

■

§ 4.4.4 Resolution

Because we have salvaged the property of Disjunctive Syllogism by extending LP to LP_m , we are now able to explore the property of Resolution in the LP_m context. We recall the defintion of Resolution.

Definition 2.4.7. *The generalized Resolution rule can be stated as*

$$\frac{l_1 \vee \ldots \vee l_k, \qquad m_1 \vee \ldots \vee m_n}{l_1 \vee \ldots \vee l_{i-1} \vee l_{i+1} \vee \ldots \vee l_k \vee m_1 \vee \ldots \vee m_{j-1} \vee m_{j+1} \vee \ldots \vee m_n},$$

where l_i and m_j are complementary literals, i.e. $l_i \equiv \neg m_j$ [11].

We continue to prove only that Resolution holds in the case of short resolvents and claim that the argument will hold for longer resolvents as well.

Theorem 4.4.8. *In the LP_m context of RBM Logic, the rule of Resolution holds for resolvents of the form $(P \vee Q) \wedge (\neg P \vee R)$. That is:*

$$\mathcal{K} \vDash (P \vee Q) \wedge (\neg P \vee R) \Rightarrow \mathcal{K} \vDash (Q \vee R)$$

Justification. We begin our proof by defining the relevant knowledge base,

$$\mathcal{K} \equiv (P \vee Q) \wedge (\neg P \vee R)\,, \tag{4.4.10}$$

and presenting $\mathcal{K}$ in SDNF:

$$\begin{aligned}\mathcal{K} \equiv (P \wedge \neg P) \vee (P \wedge H^*(P) \wedge R) \\ \vee (\neg H(P) \wedge Q \wedge \neg P) \vee (\neg H(P) \wedge Q \wedge H^*(P) \wedge R)\,.\end{aligned} \tag{4.4.11}$$

Using Thm. 4.3.3, we transform (4.4.11) into an energy function:

$$\begin{aligned}E = &-h_1\left(D_v^{LP}(P) - D_v^{K3}(P) - 0.5\right) \\ &- h_2\left(D_v^{LP}(P) + D_v^{K3}(P) + D_v^{LP}(R) - 2.5\right) \\ &- h_3\left(-D_v^{LP}(P) + D_v^{LP}(Q) - D_v^{K3}(P) - 0.5\right) \\ &- h_4\left(-D_v^{K3}(P) + D_v^{LP}(Q) + D_v^{K3}(P) + D_v^{LP}(R) - 2.5\right) + p_{\mathcal{K}}(\nu)\,,\end{aligned} \tag{4.4.12}$$

We now calculate the value of (4.4.12) for all possible valuations over the atoms.

Table 4.4: The LP_m Resolution energy function (4.4.12) for each valuation $\boldsymbol{x}_i$.

LP_m Resolution Energy Function								
$\boldsymbol{x}_i$	P	Q	R	Energy Function	Min. Energy	mi Model	Model of $\mathcal{K}$	Res. $(Q \vee R)$
$\boldsymbol{x}_1$	-1	-1	-1	$0.5h_1 + 2.5h_2 + 0.5h_3 + 2.5h_4$	0.000	No	No	No
$\boldsymbol{x}_2$	-1	-1	0	$0.5h_1 + 1.5h_2 + 0.5h_3 + 1.5h_4 + \frac{1}{8}$	0.125	No	No	**Yes**
$\boldsymbol{x}_3$	-1	-1	1	$0.5h_1 + 1.5h_2 + 0.5h_3 + 1.5h_4$	0.000	No	No	**Yes**
$\boldsymbol{x}_4$	-1	0	-1	$0.5h_1 + 2.5h_2 - 0.5h_3 + 1.5h_4 + \frac{1}{8}$	-0.375	No	**Yes**	**Yes**
$\boldsymbol{x}_5$	-1	0	0	$0.5h_1 + 1.5h_2 - 0.5h_3 + 0.5h_4 + \frac{2}{8}$	-0.250	No	**Yes**	**Yes**
$\boldsymbol{x}_6$	-1	0	1	$0.5h_1 + 1.5h_2 - 0.5h_3 + 0.5h_4 + \frac{1}{8}$	-0.375	No	**Yes**	**Yes**
$\boldsymbol{x}_7$	-1	1	-1	$0.5h_1 + 2.5h_2 - 0.5h_3 + 1.5h_4$	$\mathbf{-0.500}$	**Yes**	**Yes**	**Yes**
$\boldsymbol{x}_8$	-1	1	0	$0.5h_1 + 1.5h_2 - 0.5h_3 + 0.5h_4 + \frac{1}{8}$	-0.375	No	**Yes**	**Yes**
$\boldsymbol{x}_9$	-1	1	1	$0.5h_1 + 1.5h_2 - 0.5h_3 + 0.5h_4$	$\mathbf{-0.500}$	**Yes**	**Yes**	**Yes**
$\boldsymbol{x}_{10}$	0	-1	-1	$-0.5h_1 + 1.5h_2 + 1.5h_3 + 2.5h_4 + \frac{1}{8}$	-0.375	No	**Yes**	No
$\boldsymbol{x}_{11}$	0	-1	0	$-0.5h_1 + 0.5h_2 + 1.5h_3 + 1.5h_4 + \frac{2}{8}$	-0.250	No	**Yes**	**Yes**
$\boldsymbol{x}_{12}$	0	-1	1	$-0.5h_1 + 0.5h_2 + 1.5h_3 + 1.5h_4 + \frac{1}{8}$	-0.375	No	**Yes**	**Yes**
$\boldsymbol{x}_{13}$	0	0	-1	$-0.5h_1 + 1.5h_2 + 0.5h_3 + 1.5h_4 + \frac{2}{8}$	-0.250	No	**Yes**	**Yes**
$\boldsymbol{x}_{14}$	0	0	0	$-0.5h_1 + 0.5h_2 + 0.5h_3 + 0.5h_4 + \frac{3}{8}$	-0.125	No	**Yes**	**Yes**
$\boldsymbol{x}_{15}$	0	0	1	$-0.5h_1 + 0.5h_2 + 0.5h_3 + 0.5h_4 + \frac{2}{8}$	-0.250	No	**Yes**	**Yes**
$\boldsymbol{x}_{16}$	0	1	-1	$-0.5h_1 + 1.5h_2 + 0.5h_3 + 1.5h_4 + \frac{1}{8}$	-0.125	No	**Yes**	**Yes**
$\boldsymbol{x}_{17}$	0	1	0	$-0.5h_1 + 0.5h_2 + 0.5h_3 + 0.5h_4 + \frac{2}{8}$	-0.250	No	**Yes**	**Yes**
$\boldsymbol{x}_{18}$	0	1	1	$-0.5h_1 + 0.5h_2 + 0.5h_3 + 0.5h_4 + \frac{1}{8}$	-0.375	No	**Yes**	**Yes**
$\boldsymbol{x}_{19}$	1	-1	-1	$0.5h_1 + 0.5h_2 + 2.5h_3 + 2.5h_4$	0.000	No	No	No
$\boldsymbol{x}_{20}$	1	-1	0	$0.5h_1 - 0.5h_2 + 2.5h_3 + 1.5h_4 + \frac{1}{8}$	-0.375	No	**Yes**	**Yes**
$\boldsymbol{x}_{21}$	1	-1	1	$0.5h_1 - 0.5h_2 + 2.5h_3 + 1.5h_4$	$\mathbf{-0.500}$	**Yes**	**Yes**	**Yes**
$\boldsymbol{x}_{22}$	1	0	-1	$0.5h_1 + 0.5h_2 + 1.5h_3 + 1.5h_4 + \frac{1}{8}$	0.125	No	No	**Yes**
$\boldsymbol{x}_{23}$	1	0	0	$0.5h_1 - 0.5h_2 + 1.5h_3 + 0.5h_4 + \frac{2}{8}$	-0.250	No	**Yes**	**Yes**
$\boldsymbol{x}_{24}$	1	0	1	$0.5h_1 - 0.5h_2 + 1.5h_3 + 0.5h_4 + \frac{1}{8}$	-0.375	No	**Yes**	**Yes**
$\boldsymbol{x}_{25}$	1	1	-1	$0.5h_1 + 0.5h_2 + 1.5h_3 + 1.5h_4$	0.000	No	No	**Yes**
$\boldsymbol{x}_{26}$	1	1	0	$0.5h_1 - 0.5h_2 + 1.5h_3 + 0.5h_4 + \frac{1}{8}$	-0.375	No	**Yes**	**Yes**
$\boldsymbol{x}_{27}$	1	1	1	$0.5h_1 - 0.5h_2 + 1.5h_3 + 0.5h_4$	$\mathbf{-0.500}$	**Yes**	**Yes**	**Yes**

From Table 4.4, we see that the valuations which minimize the energy are those which model $\mathcal{K}$, and also that these models are a subset of those valuations which entail $(Q \vee R)$, i.e., which entail Resolution.

■

§ 4.4.5 Resolution Refutation

We further have the ability to investigate the process of Resolution Refutation within the context of encoding LP_m . We recall our defintion of Resolution Refutation in the context of RBM Logic.

Definition 2.4.9. *Resolution Refutation in the RBM Logic will be defined as the process of adding the query Q to the knowledge base $\mathcal{K}$, then creating and minimizing the resulting energy function based upon $\mathcal{K} \cup \{Q\}$.*

We now present our theorem and justify it using the same method as Thm. 2.4.10.

Theorem 4.4.10. *Given a consistent knowledge base $\mathcal{K}$ and a query Q, the RBM Logic in the LP_m context will prefer the classical models for which $Q \equiv \mathit{True}$ if $\mathcal{K} \cup \{Q\}$ is consistent and minimally inconsistent models of $\mathcal{K} \cup \{Q\}$ if $\mathcal{K} \cup \{Q\}$ is not consistent.*

Justification.

Claim 1: The RBM Logic will prefer the classical models for which $Q \equiv \mathit{True}$ if $\mathcal{K} \cup \{Q\}$ is consistent.

Subjustification. We define a simple knowledge base:

$$\mathcal{K} \equiv P \wedge (P \rightarrow Q)\,, \tag{4.4.13}$$

and offer the sentence Q as our query:

$$\mathcal{K}' \equiv P \wedge (P \rightarrow Q) \wedge Q,$$

We convert $\mathcal{K}'$ into SDNF:

$$\mathcal{K}' \equiv (P \wedge \neg P \wedge Q) \vee (P \wedge H^* (P) \wedge Q) . \tag{4.4.14}$$

and define our energy function to represent (4.4.14).

$$E = -h_1 \left(D_v^{LP} (P) - D_v^{K3} (P) + D_v^{LP} (Q) - 1.5 \right) \\ - h_2 \left(D_v^{LP} (P) + D_v^{K3} (P) + D_v^{LP} (Q) - 2.5 \right) + p_{\mathcal{K}} (\nu) . \tag{4.4.15}$$

We now consider possible evaluations $\boldsymbol{x}_i$ and identify those which minimize energy function (4.4.15).

Table 4.5: The LP_m Resolution Refutation energy function (4.4.15) for each valuation $\boldsymbol{x}_i$.

LP_m Resolution Refutation Energy Function						
$\boldsymbol{x}_i$	P	Q	Energy Function	Minimized Energy	mi Model of $\mathcal{K}'$	Model of $\mathcal{K}'$
$\boldsymbol{x}_1$	-1	-1	$1.5h_1 + 2.5h_2$	0.000	No	No
$\boldsymbol{x}_2$	-1	0	$0.5h_1 + 1.5h_2 + \frac{1}{6}$	0.166	No	No
$\boldsymbol{x}_3$	-1	1	$0.5h_1 + 1.5h_2$	0.000	No	No
$\boldsymbol{x}_4$	0	-1	$0.5h_1 + 1.5h_2 + \frac{1}{6}$	0.166	No	No
$\boldsymbol{x}_5$	0	0	$-0.5h_1 + 0.5h_2 + \frac{2}{6}$	-0.166	No	**Yes**
$\boldsymbol{x}_6$	0	1	$-0.5h_1 + 0.5h_2 + \frac{1}{6}$	-0.333	No	**Yes**
$\boldsymbol{x}_7$	1	-1	$1.5h_1 + 0.5h_2$	0.000	No	No
$\boldsymbol{x}_8$	1	0	$0.5h_1 - 0.5h_2 + \frac{1}{6}$	-0.333	No	**Yes**
$\boldsymbol{x}_9$	1	1	$0.5h_1 - 0.5h_2$	$\mathbf{-0.500}$	**Yes**	**Yes**

We see then that $\boldsymbol{x}_9$ is the sole valuation which minimizes the energy and the only valuation which serves as a minimally inconsistent model for both $\mathcal{K}$ and $\mathcal{K} \cup \{Q\}$. □

Claim 2: The RBM Logic will prefer the minimally inconsistent models if $\mathcal{K} \cup \{Q\}$ is inconsistent.

Subjustification. We define a simple knowledge base:

$$\mathcal{K} \equiv P \wedge (P \rightarrow Q)\,, \tag{4.4.16}$$

and offer the inconsistent sentence Q as our query:

$$\mathcal{K}' \equiv P \wedge (P \rightarrow Q) \wedge \neg Q.$$

We convert $\mathcal{K}'$ into SDNF:

$$\mathcal{K}' \equiv (P \wedge \neg P \wedge \neg Q) \vee (P \wedge H^{*}(P) \wedge Q \wedge \neg Q)\,. \tag{4.4.17}$$

and define our energy function to represent (4.4.17).

$$\begin{aligned} E = &-h_1\left(D_v^{LP}(P) - D_v^{K3}(P) - D_v^{K3}(Q) - 0.5\right) \\ &- h_2\left(D_v^{LP}(P) + D_v^{K3}(P) + D_v^{LP}(Q) - D_v^{K3}(Q) - 2.5\right) + p_{\mathcal{K}}(\nu)\,. \end{aligned} \tag{4.4.18}$$

We now consider possible evaluations $\boldsymbol{x}_i$ and identify those which minimize energy function (4.4.18).

We see then that both $\boldsymbol{x}_4$ and $\boldsymbol{x}_8$ are the minimally inconsistent valuations which satisfy $\mathcal{K} \cup \{Q\}$. We also note that these are the valuations which minimize the energy function, and the formalism therefore prefers the minimally inconsistent models when an inconsistent query is added into the knowledge base.

□

■

§ 4.5 Chapter Conclusion

We have, where possible, successfully restored our desirable deductive properties for paraconsistent RBM Logic by encoding LP_m , the original language which sought to restore the deductive properties lost

Table 4.6: The LP_m Resolution Refutation energy function (4.4.18) for each valuation $\boldsymbol{x}_i$.

LP_m Resolution Refutation Inconsistent Energy Function						
$\boldsymbol{x}_i$	P	Q	Energy Function	Minimized Energy	mi Model of $\mathcal{K}'$	Model of $\mathcal{K}'$
$\boldsymbol{x}_1$	-1	-1	$0.5h_1 + 2.5h_2$	0.000	No	No
$\boldsymbol{x}_2$	-1	0	$0.5h_1 + 1.5h_2 + \frac{1}{6}$	0.166	No	No
$\boldsymbol{x}_3$	-1	1	$1.5h_1 + 2.5h_2$	0.000	No	No
$\boldsymbol{x}_4$	0	-1	$-0.5h_1 + 0.5h_2 + \frac{1}{6}$	$\mathbf{-0.333}$	**Yes**	**Yes**
$\boldsymbol{x}_5$	0	0	$-0.5h_1 + 0.5h_2 + \frac{2}{6}$	-0.166	No	**Yes**
$\boldsymbol{x}_6$	0	1	$0.5h_1 + 2.5h_2 + \frac{1}{6}$	0.166	No	No
$\boldsymbol{x}_7$	1	-1	$0.5h_1 + 0.5h_2$	0.000	No	No
$\boldsymbol{x}_8$	1	0	$0.5h_1 - 0.5h_2 + \frac{1}{6}$	$\mathbf{-0.333}$	**Yes**	**Yes**
$\boldsymbol{x}_9$	1	1	$1.5h_1 + 0.5h_2$	0.000	No	No

in LP. By penalizing the energy function of inconsistent valuations proportionally to their measure of inconsistency, one is able to respond to contradictory cases while preferring those models which minimize the inconsistency or even are fully consistent. This method of penalizing a given valuation could also be generalized for other metrics of preference for certain models.

CHAPTER 5

CONCLUSION AND FUTURE WORK

We have explored here the formalism for encoding propositional logic using Restricted Boltzmann Machines, and extended this formalism to encode three paraconsistent logics as well.

We found that the original formalism faithfully recreates a number of expected behaviors of classical logic, noting that while the property of *Ex Falso Quodlibet* does not explicitly hold in the same fashion as classical propositional logic, the system reduces into a philosophically similar degeneracy if one assumes that the necessary contradiction can even be encoded in the first place.

We found that the extension of the formalism to non-binary logics is viable, provided a method for identifying whether an arbitrary sentence receives a designated status or not is expressable. In the *LP* case, we explored a method through which one can extend the expressability of a language in order to represent a sentence in the necessary form for encoding into an RBM.

We further explored a method through which one can introduce a penalty to certain valuations, allowing us to extend the formalism even further to capture the concept of minimal inconsistency, allowing us to reclaim the deductive properties that are lost when one extends from the two-valued classical logic to the three-valued *LP*.

The general formalism presented here seems quite robust, easily encoding a number of additional properties required to express certain logics. The methods employed in order to express a sentence in SDNF and identify when a sentence receives a designated value in the extensions to K_3, LP, and LP_m are readily generalizable for encoding additional propositional logics.

We further conjecture that this formalism may be a solid foundation for the mathematical expression of logics in general. If one were to develop a method for encoding the predicate quantifiers $\forall$ and $\exists$, it may be possible to express a variational calculus for identifying models of predicate logic, perhaps

providing insight into issues of deciadability or at least interesting manifestations of undecidability within the system. Further, unpublished work has begun to explore a representation of modal logic in which a given world is represented by a minimized RBM, and the worlds which are accessible by this world are represented by an unminimized RBM connected to the first by a matrix which captures the relational information represented by the modal quantifiers. Further development of the formalism is needed, but there is apparent promise.

Given the prior inability of the mathematical logic community to ground mathematics in formal logic throughout the 20th century, perhaps the time has come to reverse this goal and instead ground formal logic in mathematical structures.

BIBLIOGRAPHY

[1] R. J. Brachman and H. J. Levesque, *Knowledge representation and reasoning*. Morgan Kaufmann, 2004.

[2] A. Caliskan, J. Bryson, and A. Narayanan, "Semantics derived automatically from language corpora contain human-like biases," English, *Science*, vol. 356, no. 6334, pp. 183–186, Apr. 2017, ISSN: 0036-8075. DOI: 10.1126/science.aal4230.

[3] G. Casella and E. I. George, "Explaining the gibbs sampler," *The American Statistician*, vol. 46, no. 3, pp. 167–174, 1992, ISSN: 00031305. [Online]. Available: http://www.jstor.org/stable/2685208.

[4] A. Fischer and C. Igel, "An introduction to restricted boltzmann machines," in *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, L. Alvarez, M. Mejail, L. Gomez, and J. Jacobo, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 14–36, ISBN: 978-3-642-33275-3.

[5] R. Guidotti, A. Monreale, S. Ruggieri, F. Turin, F. Giannotti, and D. Pedreschi, "A survey of methods for explaining black box models.," *ACM COMPUTING SURVEYS*, vol. 51, no. 5, n.d. ISSN: 03600300. [Online]. Available: http://proxy-remote.galib.uga.edu/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=edswsc&AN=000457121600006&site=eds-live.

[6] S. C. Kleene, *Introduction to Metamathematics*. North Holland, 1952.

[7] H. Larochelle and Y. Bengio, "Classification using discriminative restricted boltzmann machines," in *Proceedings of the 25th International Conference on Machine Learning*, ser. ICML '08, Helsinki, Finland: Association for Computing Machinery, 2008, pp. 536–543, ISBN: 9781605582054. DOI:

10.1145/1390156.1390224. [Online]. Available: https://doi.org/10.1145/1390156.1390224.

[8] G. Priest, "The logic of paradox," *Journal of Philosophical Logic*, vol. 8, no. 1, pp. 219–241, 1979, ISSN: 00223611, 15730433. [Online]. Available: http://www.jstor.org/stable/30227165.

[9] ——, "Minimally inconsistent lp," *Studia Logica: An International Journal for Symbolic Logic*, vol. 50, no. 2, pp. 321–331, 1991, ISSN: 00393215, 15728730. [Online]. Available: http://www.jstor.org/stable/20015581.

[10] M. T. Ribeiro, S. Singh, and C. Guestrin, ""why should i trust you?"" *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16*, 2016. DOI: 10.1145/2939672.2939778. [Online]. Available: http://dx.doi.org/10.1145/2939672.2939778.

[11] S. J. Russell and P. Norvig, *Artificial Intelligence a Modern Approach*, 3rd ed. Prentice Hall, 2010.

[12] R. Salakhutdinov, A. Mnih, and G. Hinton, "Restricted boltzmann machines for collaborative filtering," in *Proceedings of the 24th International Conference on Machine Learning*, ser. ICML '07, Corvalis, Oregon, USA: Association for Computing Machinery, 2007, pp. 791–798, ISBN: 9781595937933. DOI: 10.1145/1273496.1273596. [Online]. Available: https://doi.org/10.1145/1273496.1273596.

[13] P. Smolensky, "Information processing in dynamical systems: Foundations of harmony theory," *Parallel Distributed Process*, vol. 1, Jan. 1986.

[14] S. Tran and A. S. d'Avila Garcez, "Deep logic networks: Inserting and extracting knowledge from deep belief networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 2, pp. 246–258, Feb. 2018. DOI: 10.1109/TNNLS.2016.2603784. [Online]. Available: https://openaccess.city.ac.uk/id/eprint/19150/.

[15] S. Tran, "Representation decomposition for knowledge extraction and sharing using restricted boltzmann machines," 2016. [Online]. Available: https://openaccess.city.ac.uk/id/eprint/14423/.

[16] ——, "Propositional knowledge representation in restricted boltzmann machines," *ArXiv*, May 2017.

[17] H. Wang, D. Dou, and D. Lowd, "Ontology-based deep restricted boltzmann machine," in *Database and Expert Systems Applications*, S. Hartmann and H. Ma, Eds., Cham: Springer International Publishing, 2016, pp. 431–445, ISBN: 978-3-319-44403-1.