Truth-space mass assignments

This item was submitted to Loughborough University's Institutional Repository by the/an author.

Additional Information:


Metadata Record: [https://dspace.lboro.ac.uk/2134/34119](https://dspace.lboro.ac.uk/2134/34119)

Publisher: © Robert Steven Patching

Rights: This work is made available according to the conditions of the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0) licence. Full details of this licence are available at: [https://creativecommons.org/licenses/by-nc-nd/4.0/](https://creativecommons.org/licenses/by-nc-nd/4.0/)

Please cite the published version.
University Library

Author/Filing Title: PATCHING, R.S.

Class Mark: T

Please note that fines are charged on ALL overdue items.

FOR REFERENCE ONLY
Truth-Space Mass Assignments

by

Robert Steven Patching

A Doctoral Thesis

Submitted in partial fulfilment of the requirements for the award of Doctor of Philosophy of Loughborough University

11th May 2005

Copyright © by Robert Steven Patching 2005
Abstract

The theory of mass assignments allows reasoning using probability families that are either imprecise, incomplete or both. The majority of previous work has been with mass assignments defined upon arbitrary domains. This Thesis concentrates on a neglected specialisation of mass assignments, the truth-space mass assignments defined upon the power set of Booleans.

The semantics of truth-space mass assignments and their operators are described, both in relation to general mass assignments and also with other methods of imprecise and incomplete reasoning. New operators are defined for truth-space mass assignments that allow them to be reasoned with in new ways consistent with other logic systems. Alterations are made to existing operators to allow them to act in a more intuitive way.

Using the new and altered operators this Thesis allows mass assignment theory to act as a many-valued logic handling imprecise and incomplete truths. This opens up many new topics of research and potential for applying the method to solve problems in a new way.

Keywords: Artificial Intelligence, Logic, Many-valued logic, Mass assignment theory, Imprecise knowledge, Inconsistency, Probability.
Acknowledgements

The author would like give thanks to several people without whom this thesis would not have been completed.

Firstly I would like to thank my supervisor Chris Hinde and ex-supervisor Steve McCoy who both provided invaluable assistance, and a lot of red pen, in helping me clarify this work. My interest in the area, which eventually lead to this thesis, was sparked during one of Chris Hinde's lectures many years ago.

I would also like to thank my parents, grandparents and the rest of my family who have always supported me in everything I have done. Without their financial help I would not have been able to complete either my undergraduate or postgraduate studies.

Finally my thanks go out to my housemates past and present as well as all my other friends who made my university experience such a positive one.
Contents

Contents i
List of Figures v
List of Tables vii

1 Introduction to the Thesis 1
1.1 Logic Systems 1
1.2 Motivation of the Thesis 2
1.3 Overview of the Thesis 4

2 Logic Systems and Imprecise and Uncertain Logic 8
2.1 Introduction 8
2.2 General Comments on Two-valued Logic 9
2.3 Aristotelian Logic 12
2.4 Boolean Logic 15
2.5 Lukasiewicz Logics 17
2.6 Belnap’s Four-Valued Logic 19
2.7 Fuzzy Logic 28
2.8 The Dempster–Shafer Theory of Evidence 35

3 Mass Assignment Theory 40
3.1 Introduction 40
3.2 The Least Prejudiced Distribution 41
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.3</td>
<td>Type-1_R and Type-2_R Restrictions</td>
<td>42</td>
</tr>
<tr>
<td>3.4</td>
<td>The Restrictiveness Ordering (≤_R)</td>
<td>44</td>
</tr>
<tr>
<td>3.5</td>
<td>The Complement of a Mass Assignment</td>
<td>45</td>
</tr>
<tr>
<td>3.6</td>
<td>Mass Assignment Operators</td>
<td>45</td>
</tr>
<tr>
<td>3.7</td>
<td>Individual Mass Assignment 'Meet' and 'Join'</td>
<td>47</td>
</tr>
<tr>
<td>3.8</td>
<td>Families of Mass Assignments</td>
<td>49</td>
</tr>
<tr>
<td>3.9</td>
<td>FRIL's Multiplicative Knowledge 'Meet' and 'Join'</td>
<td>53</td>
</tr>
<tr>
<td>3.10</td>
<td>Discussion on MA Operators</td>
<td>55</td>
</tr>
<tr>
<td>3.11</td>
<td>Support Functions</td>
<td>56</td>
</tr>
<tr>
<td>3.12</td>
<td>Semantic Unification</td>
<td>58</td>
</tr>
<tr>
<td>4</td>
<td>Truth-Space</td>
<td>61</td>
</tr>
<tr>
<td>4.1</td>
<td>The Domain of Truth-Space Mass Assignments</td>
<td>61</td>
</tr>
<tr>
<td>4.2</td>
<td>Truth-Space and Belnap Domains</td>
<td>61</td>
</tr>
<tr>
<td>4.3</td>
<td>The Knowledge Ordering of the Truth Domain (≤_K)</td>
<td>63</td>
</tr>
<tr>
<td>4.4</td>
<td>The Truth Ordering of the Truth Domain (≤_T)</td>
<td>65</td>
</tr>
<tr>
<td>4.5</td>
<td>The Bilattice of the Truth-MA Domain</td>
<td>66</td>
</tr>
<tr>
<td>4.6</td>
<td>Discussion on Truth-Space</td>
<td>67</td>
</tr>
<tr>
<td>5</td>
<td>Truth-Space Mass Assignments</td>
<td>69</td>
</tr>
<tr>
<td>5.1</td>
<td>Overview</td>
<td>69</td>
</tr>
<tr>
<td>5.2</td>
<td>The Partial Order Limits</td>
<td>71</td>
</tr>
<tr>
<td>5.3</td>
<td>The Truth-MA Knowledge Ordering(≤_K)</td>
<td>73</td>
</tr>
<tr>
<td>5.4</td>
<td>The Truth-MA Truth Ordering (≤_T)</td>
<td>74</td>
</tr>
<tr>
<td>5.5</td>
<td>Truth Mass Assignment Operators</td>
<td>77</td>
</tr>
<tr>
<td>5.6</td>
<td>Truth Negation</td>
<td>80</td>
</tr>
<tr>
<td>5.7</td>
<td>Discussion on Truth-MAs</td>
<td>85</td>
</tr>
<tr>
<td>6</td>
<td>Understanding MA Operators</td>
<td>91</td>
</tr>
<tr>
<td>6.1</td>
<td>Introduction</td>
<td>91</td>
</tr>
<tr>
<td>6.2</td>
<td>The Tableau</td>
<td>92</td>
</tr>
<tr>
<td>6.3</td>
<td>The Contribution Function</td>
<td>93</td>
</tr>
</tbody>
</table>
CONTENTS

6.4 The Distribution Function ................................ 96
6.5 Combining the Functions .................................. 98
6.6 The Consistency Function ................................ 99
6.7 The Certainty Function .................................. 101
6.8 Discussion ................................................ 103

7 Semantic Unification and Inconsistency ............... 105
7.1 Overview .................................................. 105
7.2 Notation ................................................... 106
7.3 Interval Semantic Unification ......................... 106
7.4 Mass Assignments as Claims ......................... 108
7.5 Semantic Unification as a Valuation ............. 109
7.6 Support Functions as Questions .................... 111
7.7 General Unitary Case for the Original SU .... 111
7.8 Updated Interval Semantic Unification .......... 116
7.9 Discussion ................................................ 120
7.10 More Claims ............................................. 121
7.11 The Final Version of Interval Semantic Unification ..... 123
7.12 Extending the Claims ................................ 127
7.13 Point-Value Semantic Unification ............. 129
7.14 Conclusions ............................................. 134

8 Limits of Truth-Space MA Operators ............... 136
8.1 Introduction ............................................. 136
8.2 The Limits ............................................... 137
8.3 Limits and Maximals .................................. 138
8.4 Limits and the Truth Maximals .................... 139
8.5 Limits and the Knowledge Maximals ............. 141
8.6 Limits and the Restrictiveness Maximals .... 143
8.7 Limits and the Falsity Maximals ................. 147
8.8 The Envelope of Valid Results ................. 148
8.9 Limits of Point-Value Operators ............ 149
## CONTENTS

8.10 Example of Limits for Semantic Unification 152  
8.11 Racing Pundit Example 157  
8.12 Discussion on Limit Results 159  

9 Conclusion 162  
  9.1 Contribution of Thesis 162  
  9.2 Applicability of the Thesis 165  
  9.3 Future Work 166  
  9.4 Final Comments 167  

A Calculating the Limits of Truth-Space MAs 169  
  A.1 Limit Mass Assignments 169  
  A.2 Maximising a Single Cell 170  
  A.3 Maximising the Current Focal Set 170  
  A.4 Considering the Other Focal Elements 179  

B Publications 181  

Bibliography 183
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>FOUR’s Truth Meet Operator ((\land))</td>
<td>21</td>
</tr>
<tr>
<td>2.2</td>
<td>FOUR’s Truth Join Operator ((\lor))</td>
<td>21</td>
</tr>
<tr>
<td>2.3</td>
<td>FOUR’s Knowledge Meet Operator ((\otimes))</td>
<td>22</td>
</tr>
<tr>
<td>2.4</td>
<td>FOUR’s Knowledge Join Operator ((\oplus))</td>
<td>23</td>
</tr>
<tr>
<td>2.5</td>
<td>FOUR’s Truth Negation Operator ((\neg))</td>
<td>23</td>
</tr>
<tr>
<td>2.6</td>
<td>A Bilattice Representation of FOUR</td>
<td>25</td>
</tr>
<tr>
<td>2.7</td>
<td>Example Fuzzy Set for the Concept 'Tall'</td>
<td>30</td>
</tr>
<tr>
<td>2.8</td>
<td>Example Fuzzy Set for the Concept 'Not Tall'</td>
<td>31</td>
</tr>
<tr>
<td>2.9</td>
<td>Example Fuzzy Set for the Union of 'Tall' and 'Not Tall'</td>
<td>32</td>
</tr>
<tr>
<td>2.10</td>
<td>Example Fuzzy Set for the Intersection of 'Tall' and 'Not Tall'</td>
<td>33</td>
</tr>
<tr>
<td>2.11</td>
<td>Dempster-Shafer Assignment 1</td>
<td>37</td>
</tr>
<tr>
<td>2.12</td>
<td>Dempster-Shafer Assignment 2</td>
<td>37</td>
</tr>
<tr>
<td>2.13</td>
<td>Dempster’s Rule of Combination</td>
<td>37</td>
</tr>
<tr>
<td>3.1</td>
<td>Example Probability Families for (m_A \leq_R m_B)</td>
<td>44</td>
</tr>
<tr>
<td>3.2</td>
<td>Generic Mass Assignment Table</td>
<td>46</td>
</tr>
<tr>
<td>3.3</td>
<td>Example Multiplicative Mass Assignment Meet</td>
<td>54</td>
</tr>
<tr>
<td>4.1</td>
<td>The Knowledge Meet of the Truth Domain ((\otimes))</td>
<td>64</td>
</tr>
<tr>
<td>4.2</td>
<td>The Knowledge Join of the Truth Domain ((\oplus))</td>
<td>64</td>
</tr>
<tr>
<td>4.3</td>
<td>The Truth Meet of the Truth Domain ((\land))</td>
<td>65</td>
</tr>
<tr>
<td>4.4</td>
<td>The Truth Join of the Truth Domain ((\lor))</td>
<td>66</td>
</tr>
</tbody>
</table>
List of Figures

4.5  The Truth Domain's Truth Negation Operator (¬)  . 66
4.6  The Truth-Space Bilattice  . . . . . . . . . . . . . 67
5.1  Double Hasse Diagram of the Truth-MA Limits  . . . 72
7.1  The Original Semantic Unification Operator  . . . . 108
7.2  An Example of the Original Semantic Unification Op-
erator  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 109
7.3  The Updated Semantic Unification Operator  . . . . 117
7.4  The Final Semantic Unification Operator  . . . . . 124
7.5  Multiplicative Mass Assignment Example  . . . . . . . 128
8.1  A Hypothetical Tableau 1  . . . . . . . . . . . . . . . 144
8.2  A Hypothetical Tableau 2  . . . . . . . . . . . . . . . 145
8.3  A Hypothetical Tableau 3  . . . . . . . . . . . . . . . 146
8.4  A Hypothetical Envelope of Valid Results  . . . . . 149
8.5  A Consistency Envelope  . . . . . . . . . . . . . . . . . 151
8.6  A Certainty Envelope  . . . . . . . . . . . . . . . . . 151
8.7  A Consistency and Certainty Envelope  . . . . . . 152
8.8  Limits for the Example SU (A | B)  . . . . . . . . . 155
8.9  Point-Value Limits for the Example SU (A | B)  . . 156
A.1  α partially dependent on β (α → β)  . . . . . . . . . . 172
A.2  Codependency of α and β (α ↔ β)  . . . . . . . . . . 172
A.3  Independency of α and β (α || β)  . . . . . . . . . . 172
A.4  γ Codependent on α and δ (α ↔ γ ∧ γ ↔ δ)  . . . . 174
List of Tables

2.1 Boolean Operators ............................................. 17
2.2 Lukasiewicz Logic Operators ................................. 18
2.3 The Truth-Values of \( \mathcal{FOLR} \) ......................... 19
2.4 \( \mathcal{FOLR} \)'s Main Operators and Comparators ........ 24

8.1 Permutations for Maximising Truth-Space MAs ........... 138
8.2 Limits that Create the Truth Maximals .................... 140
8.3 Limits that Create the Knowledge Maximals ............... 142
8.4 Limits that Create the Restrictiveness Maximals .......... 143
8.5 Limits that Potentially Create the Falsity Maximals ...... 147
8.6 Limits for Example \( A \ | \ B \) ................................. 154
Chapter 1

Introduction to the Thesis

1.1 Logic Systems

There are many systems of logic, they are all concerned with the process of inference. It is the process of constructing new assertions from existing ones according to some formal rules. Traditionally logic has been part of philosophy. However, since the mid-19th Century logic has also been studied as part of mathematics.

The field of logic is extremely large ranging from discussion about ‘what is true?’ and causality to incorrect forms of argumentation (fallacies) and paradoxes. The ancient Greek philosophers highlighted many of these paradoxes in the third century BC. However, even after the formalisation of two-valued logic by George Boole in the 1850s [13, 14] these paradoxes still made reasoning about many common situations difficult. The main problem was that early logic systems required complete and consistent knowledge about the situation.

The first many-valued logic was developed by Lukasiewicz in the early 1920s to address some of these paradoxes [36, 37, 39]. Since then numerous other many-valued logic systems have been proposed, see Rescher [47] and Hänle [27] for surveys of many-valued logics.
CHAPTER 1. INTRODUCTION TO THE THESIS

They attempt to provide reasoning systems that can handle situations difficult for classical two-valued logic. These situations range from the inherent vagueness of human concepts to imprecision and inconsistency found when collecting information.

One of these many-valued logic systems is that of Belnap's \textsc{four} \cite{belnap1, belnap2}. Previous work has shown that this logic is an extension of Kleene's three-valued logic \cite{kleene}, which in turn is an extension of classical propositional logic. Consequently it provides a powerful method of logical reasoning that can handle inconsistent statements and statements for which the truth is not known.

There is one major shortcoming of \textsc{four} and that is its inability to represent partial information and vague concepts. One method that can handle these concepts is mass assignment theory developed by Baldwin \cite{baldwin1, baldwin2, baldwin3} in 1991. The basic concept of mass assignment theory involves assigning probability mass to sets of possible events from a particular universe of discourse. This is similar to the Dempster–Shafer theory of evidence \cite{dempster1, shafer} with the most obvious difference being that probability mass can be associated with the empty set, denoting inconsistency. There are other differences detailed in Chapter 3.

1.2 Motivation of the Thesis

This Thesis looks in detail at a previously ignored subset of mass assignment theory. This is the set of mass assignments based upon the Boolean values true and false. In subsequent chapters this Thesis defines the semantics of the universe of discourse on which these mass assignments are defined (called truth-space). Chapter 4 shows that truth-space shares several similarities with \textsc{four} but also has some important differences.

The main reason to look at truth-space mass assignments is be-
cause they allow the knowledge they contain to be manipulated in ways unavailable to normal mass assignments. Current mass assignment operators allow mass assignments to be combined in a way similar to fuzzy logic's intersection and union. The new operators defined in this Thesis extend the set of operators available for combining truth mass assignments to include operators similar to logical conjunction and disjunction.

Through considering the semantics of truth-MAs, parallels are drawn to other reasoning systems, not only mass assignment theory which it is a part of and Belnap's \texttt{FOUR}, but also Boolean logic, probabilistic logic, Kleene's three-valued logic, and interval logic.

The semantics of truth-MAs and the newly operators defined allows Belnap's \texttt{FOUR} to be improved with greater flexibility in defining degrees of certainty and consistency. It also extends mass assignment theory to allow reasoning in a way consistent with existing many-valued logic systems.

**Aims and Objectives**

This thesis aims to do four main things:

1. Define the semantics of the domain that truth-space mass assignments are based upon.
2. Define the semantics of the truth-space mass assignment lattices. This will create a formal algebra for truth-space mass assignments.
3. Examine existing mass assignment operators with a greater focus on the consequences of using incomplete mass assignments.
4. Explore the consequences of using different versions of mass assignment operators.
These aims will allow a deeper understanding of truth-space mass assignments and mass assignment operators in general. Linking the insights gained in this thesis to existing logic systems should allow mass assignment theory and truth-space mass assignments to be used in developing a consistent and flexible reasoning system for imprecise and uncertain logic.

1.3 Overview of the Thesis

The remainder of this chapter gives an overview of the rest of the thesis and how it is broken-down into topics.

Chapter 2: Logic Systems and Imprecise and Uncertain Logic

As mentioned before there are numerous logic systems and in particular there are many which attempt to handle imprecise and uncertain statements. This chapter presents the major works on imprecise and uncertain logic that have influenced this thesis.

First, problems with bivalent logic are discussed. These were originally highlighted (albeit in an informal way) by Aristotle and his contemporary Eubulides of Miletus. Next classical Boolean logic is detailed, the mathematical formalisation of a bivalent logic used as the basis of logic circuits. Then various imprecise and uncertain logic are discussed, in particular Belnap’s four-valued logic, which is one of the bases for the work presented in this Thesis.

Chapter 3: Mass Assignment Theory

The theory of mass assignments is described in this chapter. As implied by the title of the Thesis it is concerned with using the theory of mass assignments to represent imprecise and uncertain
logics. This chapter presents the theory as it is currently described in the literature. In addition, it attempts to clarify those parts of the theory that are unclear.

Chapter 4: Truth-Space

Using Belnap's four-valued logic as an inspiration this chapter defines and explores truth-space, the power set of the Boolean values $\{\top, \bot\}$. Despite similarities between the representation of truth-space and Belnap's $\mathcal{F}_4\mathcal{U} \mathcal{R}$ there are distinct differences in internal semantics. These differences are explained and two lattices are defined that place truth and knowledge partial orderings on the elements of truth-space in a similar way to $\mathcal{F}_4\mathcal{U} \mathcal{R}$. These orderings are then used in later chapters to provide operators for mass assignments based upon truth-space.

Chapter 5: Truth-Space Mass Assignments

Truth-space defined in the previous chapter is used to develop truth-space mass assignments. The basic logic operators of conjunction and disjunction are defined to ensure they have well-formed semantics. These combined with already existing mass assignment operators allow the mass assignments based on truth-space to be combined in many new ways. Truth-space mass assignments and their operators form the bulk of the work comprising this Thesis and later chapters expand on the ideas presented here and explore the consequences.

Chapter 6: Understanding MA Operators

Many mass assignment operators, as presented by Baldwin et al [8, 10], have two main forms. Interval operators which return an upper
CHAPTER 1. INTRODUCTION TO THE THESIS

and lower probability bound that describes the range of uncertainty about the probability of the truth of a result and a point-value operator that makes assumptions to reduce the interval to a single probability value. The two types of operator are presented in different formats making comparison difficult.

This chapter changes the representation of point-value operators to a tableau form similar to that used for interval operators. This allows the various differences present between the types of operators to be highlighted and discussed in isolation from each other.

Chapter 7: Semantic Unification and Inconsistency

Semantic unification is the operator that moves knowledge from mass assignments based on arbitrary domains onto truth-space. This chapter develops the current semantic unification operator in several ways.

It assumes that instead of mapping onto an interval or point-value probability (as it does classically) it instead maps onto a truth-space mass assignment (which can represent interval or point-value probabilities if so desired). It also alters the semantic unification equation to ensure it preserves inconsistency when required so that it works in an intuitive way.

Chapter 8 - Limits of Truth-Space Mass Assignment Operators

The current literature on mass assignment operators mentions that each operator can produce a variety of results. Despite this the majority of applications using mass assignment theory are implemented using only one version (the multiplicative). This chapter shows that
there are extreme results for mass assignments operators that describe the most true, most false, most uncertain, and most certain possible results. Together these extreme points describe an envelope of possible results. The focus of this chapter is on the limits of truth-space operators. The concept can be expanded to mass assignments based upon any domain although the semantics become less clear.

Chapter 9: Conclusion

This final chapter presents the main contributions of the Thesis by highlighting the main additions and alterations to mass assignment theory. Finally it proposes future avenues of research created by the Thesis.
Chapter 2

Logic Systems and Imprecise and Uncertain Logic

2.1 Introduction

Logic as a field of study is an old one. Originally it was the sole purview of philosophy. The ancient Greeks first looked at logic as a way of arguing consistently, with one of the major proponents of logic being Aristotle. The formulation of logic into a subset of mathematics did not take place until George Boole [13, 14] in the mid 19th century. Several paradoxes have been proposed throughout history that highlight limitations of classical two-valued (bivalent) logic. These paradoxes tend to show that bivalent logic cannot handle incomplete or imprecise knowledge very well. Various logic systems have been proposed that attempt to deal with these limitations the first being Lukasiewicz logic [39], the most famous being Zadeh's fuzzy logic [58].

This chapter looks at two-valued logic and highlights its limitations. In particular it examines those limitations due to the insis-
tence that statements must be either true or false. Next it describes the first multi-valued logic. It was conceived by Jan Lukasiewicz in 1917 [37] as a way of handling the unknown truth of statements about the future.

The chapter moves on to look at Belnap's four-valued logic [11, 12] which takes a different view than Lukasiewicz logic as to what the various intermediate logic values mean. Of particular interest in this section is the presentation of bilattices, a mathematical structure created by Ginsberg [25], which is used to describe Belnap's four-valued logic and other systems derived from it.

Next this Thesis presents Fuzzy Sets and Fuzzy Logic. This branch of logic introduced by Lotfi Zadeh allows the representation of vague and imprecise information and provides a system of reasoning that can handle such information.

Finally this chapter introduces the Dempster–Shafer theory of evidence [18, 53]. This theory allows the representation of epistemic knowledge that may have been gathered from imprecise or vague sources. Unlike normal probability theory, the Dempster–Shafer theory of evidence assigned probabilities to sets of events rather than individual events.

2.2 General Comments on Two-valued Logic

'Two-valued logic' covers a range of different logic systems that share one main feature: they require that statements are assigned one of two values (either true or false). Classical logic is the oldest of the logic systems based on and developed from ideas proposed in ancient Greece. Since then the ideas have been refined and formalised whilst still maintaining many of the basic principles laid down at that time.
Fundamental Laws

Common to all two-valued logic systems are the following three laws.

Law 1 (Bivalence) *A statement must be either true or false and cannot take any other value* [38, p. 126].

Law 2 (The excluded middle (LEM)) *'A or not-A' is true (ie it is a tautology). The law states that the combination of A or its negation is always true* [48].

Law 3 (Non-contradiction (LNC)) *'A and not-A' is false. This law states that it is impossible for the combination of A and its negation to be true* [48].

Bivalence (Law 1) is the fundamental law of two-valued logic. If it is accepted as a truth then both of the other laws can be proved as theorems using propositional calculus. However, it is not possible to derive bivalence from either of the other laws.

It is worth pointing out the difference between bivalence and the law of the excluded middle (LEM), because at first inspection they seem similar. The LEM requires that the total of a statement or its negation must be true. However, the LEM places no requirement on the actual values of the statement or its negation, only their combination. The only law that actually states the values must be either true or false is the law of bivalence. The third law, is called the law of non-contradiction (LNC). Stated simply it requires that it is impossible for the total of a proposition and its negation to be true.

Problems with Bivalence

The law of bivalence which underpins all of the two-valued logic systems is also the cause of some of their shortcomings.
CHAPTER 2. LOGIC SYSTEMS

Future Contingents

Consider the following statement, called the contingent sea battle in Aristotle's de Interpretatione (On Interpretation) [3, Part 9]:

'There will be a sea battle tomorrow.'

Clearly the law of the excluded middle holds, either there will be a battle tomorrow or there will not. However, it can be argued that it is not possible to assign the value true or false to the statement today and therefore cannot be expressed in a bivalent logic. This type of statement is called a future contingent and is a problem for all bivalent logics. When Aristotle described this problem, he stated that the truth of future statements is an exception to the normal bivalence of statements.

Insufficient Information

Bivalence also has a problem with insufficient information. This is similar to the problem of future contingents. Bivalent logics claim that a statement is either true or false. However, to assign a truth value to a statement perfect knowledge is assumed. Future contingents are a problem because their truth has not been established yet, i.e., there is insufficient information. Insufficient information can occur in other situations as well, particularly if a source is not totally reliable.

Vagueness

Another problem is that of vagueness. As with future contingents there are statements for which the excluded middle holds but to which an exact value of true or false can not be assigned. Consider the following statement:

'Harold is tall'
Again the excluded middle holds, either Harold is tall or he isn't. However, a problem arises because it is difficult to arrive on a precise definition of tall. It could be decided that anybody over 170cm in height is tall, but this raises the question of why 169.9cm isn't considered tall. These vague problems are called Sorites Paradoxes a class of paradoxes attributed to a contemporary of Aristotle, Eubulides of Miletus. He formulated two of the original vagueness problems, they are presented as a series of related questions:

**The Bald Man:** Would you describe a man with one hair on his head as bald? Yes. Would you describe a man with two hairs on his head as bald? Yes. . . . You must refrain from describing a man with ten thousand hairs on his head as bald, so where do you draw the line?

**The Heap:** Would you describe a single grain of wheat as a heap? No. Would you describe two grains of wheat as a heap? No. . . . You must admit the presence of a heap sooner or later, so where do you draw the line?

The second of these problems gave its name to the class of paradoxes which it typifies; Sorites derives from the Greek *soros* (a heap). There have been several methods to try and deal with the Sorites paradox, such as Zadeh’s Fuzzy Logic [58].

### 2.3 Aristotelian Logic

The first major proponent of logic was Aristotle, one of the great philosophers of ancient Greece. His works on logic are collectively called the *Organon*, which comprises of six main works:

**Categories** [2] This work classifies (categorises) words based upon the object, action or concept that it refers to. *Eg* 'runs' is
an 'acting' word, modern linguistics would call it a verb, and ‘book’ is a substance word. *Categories* also discusses the logical properties of these classifications.

*Prior Analytics* [6] Here Aristotle presented the basic form of argumentation, the 'syllogism'. These arguments are discussed further in the sections below.

*Posterior Analytics* [5] Following up the work in the *Prior Analytics*, Aristotle discusses which syllogisms are valid.

*Topics* [7] In this Aristotle presents his methods for building the valid arguments he discusses in the *Posterior Analytics*.

*On Interpretation* [3] This work discusses propositions (Aristotle called them 'simple sentences') and quantifiers and presents their logical properties.

*On Sophistical Refutations* [4] This deals with logical fallacies, forms of reasoning that are invalid.

These works formed the basis of most of early and medieval reasoning. Aristotle and his followers held that there were two important principles in logic: the excluded middle, and non-contradiction. In addition, Aristotle held that with regards to past or present statements only two truth values apply: either true or false. This makes Aristotelian logic a bivalent logic (see Section 2.2 for a description of these laws).

**Aristotle's Propositions**

Aristotle recognised four different qualified sentences each containing a subject $S$ and a predicate $P$. For historical reasons the sentences are labelled either $A$, $E$, $I$, or $O$:
Universal affirmative (A): Every $S$ is a $P$.

Universal negative (E): No $S$ is a $P$.

Particular affirmative (I): Some $S$ is a $P$.

Particular negative (O): Not every $S$ is a $P$.

Syllogisms

The basic propositions can be combined in groups of three called syllogisms. A syllogism (from the Greek *sullogismos* meaning ‘deduction’) is an inference where one proposition (the conclusion) follows as a necessity from two other propositions (the premises).

There were twenty-four valid syllogisms named by medieval scholars, most of which appear in Aristotle’s *Prior* and *Posterior Analytics*. One example of a valid syllogism is the ‘Barbara’ syllogism:

1. Every $A$ is a $B$ (premise).
2. Every $B$ is a $C$ (premise).
3. Therefore every $A$ is a $C$ (conclusion).

Features and Limitations of the System

One of the problems of Aristotelian logic was highlighted by Aristotle himself. Whilst he held that the truth of past and present statements was either true or false (*i.e.* bivalent) Aristotle did identify a problem with statements about the future. Aristotle realised that statements whose truth is based upon the outcome of future events are exceptions to the normal bivalence rule.

In *On Interpretation* Aristotle claims that each statement must assert or deny the truth of a single predicate on a single subject.
Therefore, several truth assertions cannot be combined using sentential operators such as 'and' or 'or' and then be treated as a single unit. In particular, negations such as the universal negative (\(E\)) and the particular negative (\(O\)) are basic propositions in Aristotelian logic and are not the result of applying a 'not' operator to the universal (\(A\)) or particular (\(I\)) positive propositions. This means that a large number of features common to many modern logic systems such as double negatives, de Morgan's laws, commutativity, associativity, distribution, etc are not present and are in fact impossible in Aristotelian logic.

### 2.4 Boolean Logic

Boolean logic is named after George Boole who first proposed a system of logic that uses algebra to deal with expressions in propositional calculus [13, 14].

Formally, Boolean logic is a lattice [16]. Lattices are mathematical constructs, they can be viewed either as partially ordered sets where every pair of elements has a least upper bound and a greatest lower bound or as an algebra construct that combines a set with two binary operators \(\land\) and \(\lor\), which are called the meet and the join respectively. The meet and the join are chosen (or constructed) so that if \(A\) is a set, \(\langle A, \land, \lor\rangle\) is a lattice and \(a, b, c\) are all in \(A\) then following properties exist:

- **Idempotency** \(a \land a = a\) and \(a \lor a = a\)
- **Commutativity** \(a \land b = b \land a\) and \(a \lor b = b \lor a\)
- **Associativity** \(a \land (b \land c) = (a \land b) \land c\) and \(a \lor (b \lor c) = (a \lor b) \lor c\)
- **Absorption** \(a \land (a \lor b) = a\) and \(a \lor (a \land b) = a\)

Boolean algebras have four additional properties:
CHAPTER 2. LOGIC SYSTEMS

**Lower bound** There exists an element 0 such that for all a in A
\[ a \lor 0 = a \]

**Upper bound** There exists an element 1 such that for all a in A
\[ a \land 1 = a \]

**Distributative** For all elements \( a, b \) and \( c \) in \( A \) then
\[ (a \land b) \lor c = (a \lor b) \land (a \lor c) \] and \[ (a \lor b) \land c = (a \land b) \lor (a \land c) \]

**Existence of complements** For each element \( a \) in \( A \) there exists a complementary element \( \neg a \) so that
\[ a \land \neg a = 0 \] and \[ a \lor \neg a = 1 \]

Any tuple \((A, \land, \lor)\) which fulfils the properties of idempotency, commutativity, associativity and absorption is a lattice. If it also fulfils the other four properties then it is a Boolean Algebra. However, when people talk about Boolean logic they usually refer to the Boolean algebra based on the Boolean set \{true, false\} or \{0, 1\}.

Claude Shannon wrote his Master's thesis on applying Boolean logic to telephone switching circuits [54]. By showing that circuits could be built that would mimic boolean operators Shannon opened up the field of binary computation on which modern computers are based.

### Boolean Operators

The three main operators in Boolean logic are the meet (called conjunction or AND), the join (called disjunction or OR), and the complement (called negation or NOT). The operators can be defined in many ways. The most common way is to define the Boolean set as the numbers \{0, 1\} with zero representing false and one representing true. Then the lattice partial ordering is simply the standard integer ordering and conjunction and disjunction can be defined either arithmetically or comparatively, see Table 2.1.
CHAPTER 2. LOGIC SYSTEMS

<table>
<thead>
<tr>
<th>Name</th>
<th>Arithmetic</th>
<th>Comparative</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conjunction</td>
<td>$a \land b = ab$</td>
<td>$a \land b = \min(a, b)$</td>
</tr>
<tr>
<td>Disjunction</td>
<td>$a \lor b = a + b - (ab)$</td>
<td>$a \lor b = \max(a, b)$</td>
</tr>
<tr>
<td>Negation</td>
<td>$\neg a = 1 - a$</td>
<td></td>
</tr>
</tbody>
</table>

Table 2.1: Boolean Operators

Boolean logic is bivalent and the existence of complements means that Boolean logic follows the laws of the excluded middle and non-contradiction. Because it follows the law of bivalence Boolean logic suffers from the problems of future contingents and sorites paradoxes as detailed previously in Section 2.2.

2.5 Łukasiewicz Logics

The first non-bivalent logic was developed in the 1920s by Jan Łukasiewicz [37, 39]. He was concerned with how to represent the indeterminate truth of a future contingent:

"Therefore the proposition considered at this moment neither true or false and must possess a third value, different from '0' or falsity and '1' or truth. This value we can designate by '1/2'. It represents the possible"

— Łukasiewicz [36]

By adding a third value ('possible') to the classical true and false he created a three-valued logic the first of the many-valued logic systems. Although originally the logic had only three values (referred to as $L_3$) it was subsequently extended to a finitely many-valued logic and to an infinitely valued logic. The finitely many-valued logic ($L_m$)
CHAPTER 2. LOGIC SYSTEMS

<table>
<thead>
<tr>
<th>Name</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strong Conjunction</td>
<td>$a &amp; b = \max(0, a + b - 1)$</td>
</tr>
<tr>
<td>Weak Conjunction</td>
<td>$a \wedge b = \min(a, b)$</td>
</tr>
<tr>
<td>Negation</td>
<td>$\neg a = 1 - a$</td>
</tr>
<tr>
<td>Strong Disjunction</td>
<td>$a | b = \min(1, a + b)$</td>
</tr>
<tr>
<td>Weak Disjunction</td>
<td>$a \lor b = \max(a, b)$</td>
</tr>
<tr>
<td>Implication</td>
<td>$a \rightarrow b = \min(1, 1 - (a + b))$</td>
</tr>
</tbody>
</table>

Table 2.2: Łukasiewicz Logic Operators

has a truth degree set of:

$$W_m = \{k/(m - 1) \mid 0 \leq k \leq m - 1\}, \text{ of rationals in the range } [0, 1]$$

The infinitely valued version ($L_\infty$) has a truth degree set of:

$$W_\infty = \{k \in \mathbb{R} \mid 0 \leq k \leq 1\}$$

In both cases the only value that is considered as true (the so called designated value) is the value 1.

Operators

Łukasiewicz logics have two different conjunction operators, a strong conjunction ($\&$) and a weak conjunction ($\wedge$). These together with a negation operator ($\neg$) allow corresponding strong and weak disjunction operators ($\|\text{ and } \lor\text{ respectively}$) to be derived using de Morgan’s law. A list of these basic logic operators in addition to Łukasiewicz’s implication operator can be seen in Table 2.2.
CHAPTER 2. LOGIC SYSTEMS

Lattice

All of the different Lukasiewicz logics are defined on totally ordered truth degree sets. In $L_3$ the value ‘possible’ lies between true and false and in $L_m$ and $L_{oo}$ all the intermediate values lie between the two extremes of 0 and 1. The truth degree sets of Lukasiewicz logics can be totally ordered.

2.6 Belnap’s Four-Valued Logic

One logic system developed to deal with inconsistent and incomplete information is Belnap’s four-valued logic system [11, 12]. This logic system, commonly called $\text{FOUR}$, adds two extra truth values to the classical true and false, the extra values are ‘underdefined’ (incomplete information, similar to Lukasiewicz’s ‘possible’ value, and ‘overdefined’ (inconsistent information). The notation used to describe these four values varies within the literature, the more common ones are shown in Table 2.3:

<table>
<thead>
<tr>
<th>Truth-value</th>
<th>Symbols Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>{t}</td>
</tr>
<tr>
<td>False</td>
<td>{f}</td>
</tr>
<tr>
<td>Underdefined</td>
<td>$\emptyset$</td>
</tr>
<tr>
<td>Overdefined</td>
<td>{t, f}</td>
</tr>
</tbody>
</table>

Table 2.3: The Truth-Values of $\text{FOUR}$

Of the four notations given the set notation best describes the semantics of the four values in Belnap’s logic. The sets denote whether there is evidence that a statement true, evidence it is false, no evidence either way, or evidence of both (ie contradictory evidence). If these interpretations are followed, the various operators for the lattice have clear meanings.
Like Lukasiewicz logics and other three-valued logics the values in $\text{FOUR}$ can be ordered based on how true they are. However, unlike three-valued logics, $\text{FOUR}$ cannot be totally ordered, instead two different partial orders can be imposed.

**The Truth Ordering**

Scott [52] discusses how the subsets of the set $\{t, f\}$ can be partially ordered according to their relative truth values. The truth ordering for $\text{FOUR}$ ($\preceq_t$) has a lower bound false and an upper bound of true. The two other values of underdefined and overdefined are incomparable based on how true they are. Using the truth ordering $\text{FOUR}$ can be represented as a truth lattice in a similar way to Boolean logic 2.4:

$$\text{FOUR's t-lattice} = \left< \{\emptyset, \{t\}, \{f\}, \{t, f\}\}, \land, \lor \right>$$

The truth meet operator is defined in Figure 2.1 and the truth join operator is defined in Figure 2.2.

**The Knowledge Ordering**

Sandewall [51] developed a non-monotonic logic that ordered its values based on the amount of information they provided and not based upon their amount of truth. Sandewall’s logic is based on intervals of probabilities that a statement is true. The interval $[0,1]$ represents maximum uncertainty in the truth. Narrower intervals represent less uncertainty and there is no uncertainty in the singleton probabilities $\{0\}, \{0.3\}, \{1\}, \text{etc.}$ Under Sandewall’s ordering true $\{1\}$ is incomparable to false $\{0\}$ or any other singleton value. A similar ordering appears in the Dempster-Shafer theory of evidence [18, 53] (see Section 2.8). This second ordering, based on knowledge ($\preceq_k$), can also
CHAPTER 2. LOGIC SYSTEMS

21

\[ A \land B \]

\[
\begin{array}{cccc}
\emptyset & \{t\} & \{t,f\} & \{f\} \\
\emptyset & \emptyset & \{f\} & \{f\} \\
\{t\} & \emptyset & \{t\} & \{f\} \\
\{t,f\} & \{f\} & \{t,f\} & \{f\} \\
\{f\} & \{f\} & \{f\} & \{f\} \\
\end{array}
\]

Figure 2.1: FOLK's Truth Meet Operator (\(\land\))

\[ A \lor B \]

\[
\begin{array}{cccc}
\emptyset & \{t\} & \{t,f\} & \{f\} \\
\emptyset & \emptyset & \{t\} & \emptyset \\
\{t\} & \{t\} & \{t\} & \{t\} \\
\{t,f\} & \{t\} & \{t,f\} & \{t,f\} \\
\{f\} & \emptyset & \{t,f\} & \{f\} \\
\end{array}
\]

Figure 2.2: FOLK's Truth Join Operator (\(\lor\))
be imposed on $\mathcal{FOUR}$'s truth values. The knowledge lattice is based on the same set at the truth lattice but has different meet ($\otimes$) and join ($\oplus$) operators.

$\mathcal{FOUR}$'s k-lattice $= \langle \{\emptyset, \{t\}, \{f\}, \{t,f\}\}, \otimes, \oplus \rangle$

The knowledge meet and join operators are defined in Figures 2.3 and 2.4.

The Bilattice

Ginsberg [25, 24] was the first to combine the truth and knowledge orderings into a single construct, which he called a bilattice. He described several different bilattices the simplest of which was $\mathcal{FOUR}$. In addition to the meet and join operators for each ordering, Ginsberg required bilattices to have a negation operator ($\neg$) defined that reverses the truth ordering but does not alter the knowledge ordering (see Figure 2.5). As pointed out by Fitting in his work on bilattice
logic programming:

'Negations should reverse truth; one expects that. But negations do not change knowledge; one knows as much about \( \neg a \) as one knows about \( a \). Hence a negation operation reverses the \( \leq_t \) ordering but preserves the \( \leq_k \) ordering.' — Fitting [20]

\[ \begin{array}{c|cccc} \emptyset & \{t\} & \{t,f\} & \{f\} \\ \hline \emptyset & \emptyset & \{t\} & \{t,f\} & \{f\} \\ \{t\} & \{t\} & \{t\} & \{t,f\} & \{t,f\} \\ \{t,f\} & \{t,f\} & \{t,f\} & \{t,f\} & \{t,f\} \\ \{f\} & \{f\} & \{t,f\} & \{t,f\} & \{f\} \end{array} \]

Figure 2.4: FOUR's Knowledge Join Operator (\( \oplus \))

\[ \begin{array}{c|cccc} \emptyset & \{t\} & \{t,f\} & \{f\} \\ \hline \emptyset & \emptyset & \{t\} & \{t,f\} & \{f\} \\ \{t\} & \{t\} & \{t\} & \{t,f\} & \{t,f\} \\ \{t,f\} & \{t,f\} & \{t,f\} & \{t,f\} & \{t,f\} \\ \{f\} & \{f\} & \{t,f\} & \{t,f\} & \{f\} \end{array} \]

Figure 2.5: FOUR's Truth Negation Operator (\( \neg \))

FOUR's bilattice is a sextuple combining the set \( 2^{\{t,f\}} \) with all of the previously mentioned operators (Figures 2.1, 2.2, 2.3, 2.4, and 2.5. See Table 2.4).
CHAPTER 2. LOGIC SYSTEMS

<table>
<thead>
<tr>
<th>Operator Name</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>Truth Partial Order</td>
<td>( \leq_t )</td>
</tr>
<tr>
<td>Knowledge Partial Order</td>
<td>( \leq_k )</td>
</tr>
<tr>
<td>Truth Meet</td>
<td>( \land )</td>
</tr>
<tr>
<td>Truth Join</td>
<td>( \lor )</td>
</tr>
<tr>
<td>Truth Negation</td>
<td>( \neg )</td>
</tr>
<tr>
<td>Knowledge Meet</td>
<td>( \otimes )</td>
</tr>
<tr>
<td>Knowledge Join</td>
<td>( \oplus )</td>
</tr>
</tbody>
</table>

Table 2.4: FOUR’s Main Operators andComparators

\[
\text{FOUR's bilattice} = (\{\emptyset, \{t\}, \{f\}, \{t, f\}\}, \land, \lor, \otimes, \oplus, \neg)
\]

Represented on a double Hasse diagram [49] the order of the values in both the truth ordering \((\leq_t)\) and the knowledge ordering \((\leq_k)\) is clear to see. In Figure 2.6 the truth partial-order is on the x-axis and the y-axis shows the knowledge partial-ordering.

**Explanation of the Operators**

**Knowledge Operators**

If the truth values are considered as sets of evidence then the knowledge meet and join are clearly defined. The meet of two truth values is simply their set union and their join is the intersection of the truth values. Knowledge meet \((\otimes)\) is a consensus operator. It calculates the largest amount of knowledge that its operands agree on. Conversely, the knowledge join \((\oplus)\) is a gullibility operator. It accepts any information in either operand, even if the information is contradictory.
CHAPTER 2. LOGIC SYSTEMS

Figure 2.6: A Bilattice Representation of \( \mathcal{F} \mathcal{U} \mathcal{R} \)

**Truth Operators**

The classical Boolean operators and the set operations mentioned above can be used to give a meaning to the truth meet and join. When considering the truth ordering, the meet is an internal boolean conjunction on the set elements ie:

\[
\begin{align*}
\{t\} \land \{t\} &= 1 \text{ AND } 1 = 1 = \{t\} \\
\{t\} \land \{f\} &= 1 \text{ AND } 0 = 0 = \{f\} \\
\{f\} \land \{f\} &= 0 \text{ AND } 0 = 0 = \{f\}
\end{align*}
\]

Results involving overdefined operands can be calculated by a set union of the two singleton results. Overdefined \( \{t, f\} \) is a union of the singletons and therefore results involving overdefined are a union
of the singleton results, \textit{ie}:

\[
\{t\} \land \{t, f\} = (\{t\} \land \{t\}) \cup (\{t\} \land \{f\}) \\
= \{t\} \cup \{f\} \\
= \{t, f\}
\]

And

\[
\{f\} \land \{t, f\} = (\{f\} \land \{t\}) \cup (\{f\} \land \{f\}) \\
= \{f\} \cup \{f\} \\
= \{f\}
\]

And finally,

\[
\{t, f\} \land \{t, f\} = (\{t\} \land \{t\}) \cup (\{t\} \land \{f\}) \\
\quad \cup (\{f\} \land \{t\}) \\
\quad \cup (\{f\} \land \{f\}) \\
= \{t\} \cup \{f\} \cup \{f\} \cup \{f\} \\
= \{t, f\}
\]

Similarly, truth-meet results involving underdefined are calculated using the intersection of the singleton results (underdefined is itself an intersection of the singleton values).

\[
\{t\} \land \emptyset = (\{t\} \land \{t\}) \cap (\{t\} \land \{f\}) \\
= \{t\} \cap \{f\} \\
= \emptyset
\]
And

\[
\{f\} \land 0 = (\{f\} \land \{t\}) \cap (\{f\} \land \{f\}) \\
= \{f\} \cap \{f\} \\
= \{f\}
\]

And

\[
\emptyset \land 0 = (\{t\} \land \{t\}) \cap (\{t\} \land \{f\}) \\
\cap (\{f\} \land \{t\}) \cap (\{f\} \land \{f\}) \\
= \{t\} \cap \{f\} \cap \{f\} \cap \{f\} \\
= \emptyset
\]

The only other parts of the meet operator left to explain are the interaction between underdefined and overdefined. This can be explained as either a union of the result of \(\{t\} \land 0\) with the result \(\{f\} \land 0\) or as the intersection of the result of \(\{t\} \land \{t, f\}\) with the result of \(\{f\} \land \{t, f\}\).

\[
(\{t\} \land 0) \cup (\{f\} \land 0) = \emptyset \cup \{f\} \\
= \{f\}
\]

Or

\[
(\{t\} \land \{t, f\}) \cap \\
(\{f\} \land \{t, f\}) = \{t, f\} \cap \{f\} \\
= \{f\}
\]
CHAPTER 2. LOGIC SYSTEMS

The truth join (\( V \)) can be constructed in the same way using Boolean OR and the set union and intersection. The negation operator (\( \neg \)) is simply the classic logic negation operator acting on the focal elements of the sets.

Other works on Bilattices

There are many other works on bilattices such as defining modal operators for them [26], studying the similarities FOUR and bilattices have with two-valued logic and Boolean algebras [1], and [35]. Melvin Fitting shows how Kleene logic (a three-valued logic) [32] is a member of the same family of logics that includes FOUR and all bilattice logics [21]

Some work has also been made in applying bilattices and in particular the four-valued logic. Such as using it to represent structured documents [34].

2.7 Fuzzy Logic

Fuzzy logic was first presented by Lotfi Zadeh in 1965 [58]. The fundamental feature of fuzzy logic is that of fuzzy sets. Unlike classical sets fuzzy sets have no crisp boundary, instead of an element being inside or outside the set an element can have a greater or lesser degree of membership in the fuzzy set [33].

For example, consider the concept 'tall' mentioned on page 11. Crisp sets cannot represent a set of 'tall' heights without running into the Sorites Paradox: if height \( x \) is in the set tall why isn't the arbitrarily shorter height \( x - \epsilon \) also considered tall? Fuzzy set theory allows elements to have a degree of membership between zero and one in a fuzzy set. The higher the number the greater the degree of membership.
Formally, a fuzzy set membership is a function mapping from some domain \( C \) to the unit interval \([0,1]\). Traditionally \( \mu \) is used to denote the membership function; however, each fuzzy set has its own unique membership function so in this Thesis the membership function associated with the fuzzy set \( F \) defined on \( C \) is denoted as:

\[
\mu_F : C \rightarrow [0,1]
\]  

(2.1)

Fuzzy logic is not a bivalent logic (see Page 10). This is because the proposition ‘\( x \) is a member of \( X \)’ may not be either true (1) or false (0); instead it could be somewhere in between (eg 0.7).

Fuzzy sets defined on a discrete domain, where \( C = \{x_1, \ldots, x_n\} \), can be written in the following form:

\[
X = \frac{x_1}{\mu_X(x_1)} + \cdots + \frac{x_n}{\mu_X(x_n)}
\]

However, fuzzy sets defined on continuous domains obviously cannot be written in such a way. Instead this type of fuzzy set is described by a continuous function.

In the example of the fuzzy set ‘tall’ then the domain is the real line of lengths. If in this example it is decided that all heights 1.70m and are definitely tall, and all heights 1.50m and below are definitely not tall, with all heights between the two having a linearly decreasing membership the fuzzy set ‘tall’ can have its membership defined as:

\[
\mu_{\text{tall}}(x) = \begin{cases} 
1 & \text{if } x \geq 1.70m \\
\frac{x-1.50m}{1.70m-1.50m} & \text{if } 1.50m < x < 1.70m \\
0 & \text{if } x \leq 1.50m 
\end{cases}
\]

Figure 2.7 shows the membership function for the concept tall plotted against height in metres.

In the example, the membership function is a linear function be-
between the point 1.70m and 1.50m. However, not only linear functions need be used, a sigmoid function could be used or for other situations a gaussian distribution might be more suited. Most applications of fuzzy logic are not particularly sensitive to the exact function used so simpler shapes are usually used; gaussians tend to be replaced by trapezoids or triangles [33].

**Fuzzy Set Operators**

The operations of complementation, union and intersection are defined for fuzzy sets in a similar way to their crisp counterparts.

**Standard Complementation**

If \( X \) is a fuzzy set then its complement \( X' \) has the following membership function:

\[
\mu_{X'}(x) = 1 - \mu_X(x)
\]
For example, the complement for the set 'tall' is the set 'not tall'. Note that 'not tall' is not necessarily the same as short.

$$
\mu_{\text{not tall}}(x) = \begin{cases} 
0 & \text{if } x \geq 1.70\text{m} \\
1 - \left(\frac{x-1.50\text{m}}{1.70\text{m}-1.50\text{m}}\right) & \text{if } 1.50\text{m} < x < 1.70\text{m} \\
1 & \text{if } x \leq 1.50\text{m}
\end{cases}
$$

Figure 2.8 shows the graph of the complement of the set 'tall' as defined above:

![Figure 2.8: Example Fuzzy Set for the Concept 'Not Tall'](image)

**Standard Fuzzy Set Union**

The fuzzy set union of $A$ and $B$ is denoted by $(A \cup B)$ and its membership function is:

$$
\mu_{(A \cup B)}(x) = \max(\mu_A(x), \mu_B(x))
$$

One thing to notice about the fuzzy set union is that the law of the excluded middle does not hold. This means that if $X$ is a fuzzy set
defined on \( C \) and \( X' \) is its complement. Then the following does not hold:

\[
(\forall x \in C) \quad \mu_{(X\cup X')}(x) = 1.0
\]

\( i.e. \) there are some elements \( x \) for which the statement '\( x \) is definitely a member of \( X \) or \( X'' \) is not true. This can be seen in Figure 2.9, which shows the union of 'tall' and 'not tall'.

![Figure 2.9: Example Fuzzy Set for the Union of 'Tall' and 'Not Tall']

**Standard Fuzzy Set Intersection**

Like crisp sets the intersection of two fuzzy sets \( A \) and \( B \) can be calculated. The result \( (A \cap B) \) has a membership function of:

\[
\mu_{(A\cap B)}(x) = \min(\mu_A(x), \mu_B(x))
\]

From previous subsections it is shown that fuzzy logic is neither bivalent nor does the law of the excluded middle hold. The final law to look at is that of non-contradiction. This is the law that states
the claim ‘$X$ and $X’$’ is false. For fuzzy sets this would be:

$$(\forall x \in C) \, \mu_{(X \cap X')} (x) = 0.0$$

This clearly does not necessarily hold. If $\mu_X (x) = \alpha$ and $0 < \alpha < 1$ then the complement is $\mu_{X'} = 1 - \alpha$. The minimum of $\alpha$ and $1 - \alpha$ is therefore not zero. Figure 2.10 shows the intersection for the example of ‘tall’ and ‘not tall’.

![Figure 2.10: Example Fuzzy Set for the Intersection of ‘Tall’ and ‘Not Tall’](image)

Other Versions of Intersection and Union

The operators mentioned above are often called the ‘standard’ union and intersection, however several other versions of the operators are used. Any t-norm function can be used in place of the min() function in fuzzy-set intersection. Common examples include:

**Product intersection**

$$\mu_{(A \cap B)} (x) = \mu_A (x) \mu_B (x)$$
CHAPTER 2. LOGIC SYSTEMS

Bounded difference intersection

$$\mu_{(A \cap B)}(x) = \max \left( 0, \mu_A(x) + \mu_B(x) - 1 \right)$$

Drastic intersection

$$\mu_{(A \cap B)}(x) = \begin{cases} 
\mu_A(x) & \text{if } \mu_B(x) = 1 \\
\mu_B(x) & \text{if } \mu_A(x) = 1 \\
0 & \text{otherwise}
\end{cases}$$

In a similar way the max() function in fuzzy set union can be replaced by any t-conorm (also called an s-norm) function. Examples include:

Algebraic addition union

$$\mu_{(A \cup B)}(x) = \left( \mu_A(x) + \mu_B(x) \right) - \left( \mu_A(x) \mu_B(x) \right)$$

Bounded sum union

$$\mu_{(A \cup B)}(x) = \min \left( 1, \mu_A(x) + \mu_B(x) \right)$$

Drastic union

$$\mu_{(A \cup B)}(x) = \begin{cases} 
\mu_A(x) & \text{if } \mu_B(x) = 0 \\
\mu_B(x) & \text{if } \mu_A(x) = 0 \\
1 & \text{otherwise}
\end{cases}$$

Applications of Fuzzy Sets

Fuzzy set theory has been applied in many different areas either on its own or in conjunction with other methodologies. One of the successful fields has been that of fuzzy control.
Many fuzzy control systems use fuzzy rules, based upon work by Ebrahim Mamdani in 1974 [40]. Fuzzy control is in itself a broad field, including mobile robotic control [28][50], helicopter autopilots [31], and electro-hydraulic controllers [15]. Many of the mentioned papers blend fuzzy logic with either genetic algorithms, neural networks or other artificial intelligence techniques to learn either the rules or the fuzzy membership functions used in those rules. Other applications include chemical toxicity prediction [42], user profile management [19], the classification of dermatological diseases via image analysis [17], and to aid in stroke diagnosis [43].

This is by no means a complete or comprehensive review of all the work involving fuzzy sets. Instead it simply demonstrates a few of the wide ranging topics that fuzzy sets have been applied to.

### 2.8 The Dempster–Shafer Theory of Evidence

The Dempster–Shafer theory of evidence is a method of representing epistemic knowledge. Arthur Dempster laid the ground work of the theory [18], which was later extended and developed by Glenn Shafer [53].

If $\Theta$ is a set of possibilities; for example, events, diseases, locations etc then $m$ is a basic probability assignment on $\Theta$ and conforms to the following requirements:

\[
m(\emptyset) = 0
\]
\[
m(x) \geq 0 \quad \text{for any } x \subseteq \Theta
\]
\[
\sum_{x \subseteq \Theta} m(x) = 1
\]

A subset $x$ of the set $\Theta$ is called a focal set if $m(x) > 0$. The set...
CHAPTER 2. LOGIC SYSTEMS

of all focal sets is called the core of \( m \) and is denoted \( \kappa(m) \).

\( \Theta \) is called the Universe of Discourse or the Frame of Discernment and its elements are assumed to be a mutually exclusive and exhaustive set of all the possibilities.

The basic probability \( m(x) \) is a measure of the exact amount of evidence that exists which indicates that a subset of \( x \) is in fact the case. The evidence of the set \( \Theta \) is always:

\[
m(\Theta) = 1 - \sum_{x \in \kappa(m) / \Theta} m(x)
\]

Shafer [53] defines two support functions called Belief (\( \text{Bel} \)) and Plausibility (\( \text{Pl}^* \)). They enable measures to be taken on how credible and possible it is that an element of \( x \) is the true cases. The two functions are as follows:

\[
\text{Bel}(S) = \sum_{\forall i \leq S} \sum_{x} m(X_i)
\]

(2.2)

\[
\text{Pl}^*(S) = 1 - \text{Bel}(S')
\]

(2.3)

Although \( m_X(.) \) is not a probability distribution, if all the focal sets are singleton sets then the Belief function (\( \text{Bel} \)) acts like a probability distribution. However, in general there will often be basic probability assigned to sets with a cardinality greater than one. In particular, a method is required that can combine two basic probability assignments that are not actual probability distributions.

**Dempster’s Rule of Combination**

Dempster’s rule of combination is used for updating the probability of one piece of evidence based upon another. Bayesian theory provides an equation to do a similar thing with probabilities (see Equation 2.4).
Because the pieces of evidence in the D–S theory of evidence are not probabilities they cannot use Bayes rule; instead Shafer used Dempster’s rule of combination to combine the evidences.

Consider the functions $m_1$ and $m_2$ and picture them as divisions of the unit interval $[0, 1]$ (Figures 2.11 and 2.12).

$$m_1(A_i)$$

$0$ $1$

Figure 2.11: Dempster-Shafer Assignment 1

$$m_2(B_j)$$

Figure 2.12: Dempster-Shafer Assignment 2

Tabulating the two functions provides a combination of their evidences, Figure 2.13 shows such a tabulation.

The probability mass committed exactly to a subset $C$ of $\Theta$ is
the sum of all rectangles where $A_i \cap B_j = C$.

$$\sum_{A_i \cap B_j = C} m_1(A_i) m_2(B_j)$$

However, this means that some squares will commit probability mass to the empty set. Dempster’s solution is to discard all the rectangles committed to the empty set and normalise all the remaining rectangles so they sum to one. The normalised result $m_3(C)$ is therefore defined to be:

$$m_3(C) = \frac{\sum_{A_i \cap B_j = C} m_1(A_i) m_2(B_j)}{1 - \sum_{A_i \cap B_j = \emptyset} m_1(A_i) m_2(B_j)}$$

**Advances and Alternatives to Dempster–Shafer**

*Advances in the Dempster–Shafer theory of evidence* (1994) [57] is a collection of several papers that look at various aspects of the Dempster–Shafer theory and which present extensions of the theory or present other theories that have similarities with the theory of evidence. Topics include, computational efficiency optimisation, knowledge representation, and improvements of the Dempster rule of combination.

**Dempster–Shafer and Mass Assignment Theory**

Many of the concepts that appear in the Dempster–Shafer theory of evidence also appear in mass assignment theory (Chapter 3). In particular, the concept of basic probability assignment (the ‘mass’ in a mass assignment), the Belief and Plausibility measures.

However, as can be seen in Chapter 3 there are several fundamental differences. The most obvious difference that there is no
requirement for \( m(\emptyset) \) to equal zero. This means that the various measures such as Belief produce different results.

In addition, the rule of combination differs between the two theories. The combination operator in mass assignment theory provides better solutions than Dempster’s solution. The validity of Dempster’s rule of combination is questionable if the two operands are not independent viewpoints.
Chapter 3

Mass Assignment Theory

3.1 Introduction

Mass assignment (MA) theory was introduced in the late 1980s, early 1990s [8] and later implemented in a programming language, FRIL [10], developed by Baldwin et al. It is a rationalisation and extension of the Dempster-Shafer theory of evidence [18, 53] and is firmly based upon the counting of events and intervals of probability. Mass is a precise amount of probability assigned to elements of the power set of a domain (ie sets of events), rather than individual events themselves. Each MA describes a family of probability distributions supported by the MA.

Formally a mass assignment \( m_X \) defined on the domain \( \mathcal{C} \) is a function \( \mathcal{P}(\mathcal{C}) \rightarrow [0,1] \) and written as:

\[
m_X = X_1 : x_1, \ldots, X_n : x_n
\]

Where,

\[
\sum_{i=1}^{n} x_i = 1.0
\]

And, \( X_i \in \mathcal{P}(\mathcal{C}) \)

An element \( X_i \) is a set in the domain \( \mathcal{P}(\mathcal{C}) \) and \( x_i \) is the amount of mass.
assigned to that set. Normally a focal element $X_i$ is only written if $x_i$ is non-zero. Sometimes the mass assignment defined by the function $m_X$ is written simply as $X$.

3.2 The Least Prejudiced Distribution

Each mass assignment describes a family of probability distributions. Sometimes it is desirable to collapse that family into a single probability distribution. The obvious difficulty is which of the possible distributions indicated by the mass assignment should be chosen.

To create a single probability distribution from a mass assignment the mass assigned to focal sets with more than one element needs to be divided between the singleton sets. How this mass is redistributed prejudices the final probability distribution based on some assumption.

The least prejudiced assumption is that the mass should be divided equally between all the elements and no single element is preferred over another. For example, the focal set \{a, b, c\}:0.6 would divide its mass equally so that 0.2 mass is added to the sets \{a\}, \{b\}, and \{c\}; e.g. using the least prejudiced assumption

\[
\{a\} : 0.1 \quad \{a, b\} : 0.2 \quad \{a, c\} : 0.3 \quad \{a, b, c\} : 0.4
\]

Becomes

\[
\{a\} : 0.483 \quad \{b\} : 0.383 \quad \{c\} : 0.13
\]

There are other distributions that can be created from a mass assignment. The least prejudiced distribution is simply the one that does not prefer one element (event, object) over another.
CHAPTER 3. MASS ASSIGNMENT THEORY

3.3 Type–1$_R$ and Type–2$_R$ Restrictions

As stated, a MA describes a family of probability distributions. A restriction of a mass assignment corresponds to a subset of that family. MA theory has two types of restriction, which move mass between focal elements of the MA.

The first, called type–1$_R$, moves mass from a focal element to a subset of that focal element, which corresponds to a gain in knowledge as to which element is correct. For example, the mass assignment \{a, b\} : 1 means that the actual element is a or b. Suppose it is later discovered that the probability that it was 'b' is 0.2; the new mass assignment becomes \{b\} : 0.2 \{a, b\} : 0.8, via a type–1$_R$ restriction. Formally, if $m_X$ is a mass assignment $m_{X'}$ is a type–1$_R$ restriction of $m_X$ made by moving $\alpha$ amount of mass from the set $X_j$ to the set $X_k$ leaving all other sets untouched (see Equation 3.1).

$$m_{X'} = \left\{ \{X_i : x_i \mid i \neq j \land i \neq k\} \cup \{X_j : x_j - \alpha\} \cup \{X_k : x_k + \alpha\} \right\}$$

Where $X_j \supseteq X_k$

$$\alpha \leq x_j$$

The second restriction (type–2$_R$) is less intuitive; it moves an equal quantity of mass from a pair of focal elements to the intersection of the pair and to the union of the pair. The MA results of an operator such as intersection or union, which cannot be obtained by restricting any other result of the operator are called the maximal results. Type–2$_R$ restrictions are formally defined in Equation 3.2, where $m_{X'}$ is the restriction made by moving $\alpha$ mass from sets $X_j$.
and $X_k$ to the sets $X_i$ and $X_m$, leaving all other sets untouched.

$$m_{X'} = \left\{ \begin{array}{l} \{X_i : x_i \mid i \neq j \land i \neq k\} \cup \\
{\{X_j : x_j - \alpha\} \cup \{X_k : x_k - \alpha\} \cup} \\
{\{X_i : x_i + \alpha\} \cup \{X_m : x_m + \alpha\}} \end{array} \right\}$$

Where $X_i = X_j \cap X_k \quad (3.2)$

$X_m = X_j \cup X_k$

$\alpha \leq x_j$

$\alpha \leq x_k$

Type-$2_R$ mass assignments are best illustrated through an example. If $m_X$ is a mass assignment:

$$m = \{a, b\} : 0.3, \{b, c\} : 0.4, \{b\} : 0.3$$

then the mass assignment $m_{X'}$:

$$m_{X'} = \{a, b\} : 0.1, \{b, c\} : 0.2, \{b\} : 0.5, \{a, b, c\} : 0.2$$

is a type-$2_R$ restriction of $m_X$ made by moving 0.2 mass from $\{a, b\}$ and $\{b, c\}$ to their union $\{a, b, c\}$ and their intersection $\{b\}$.

Note that in the current mass assignment literature restrictions are simply called type-1 and type-2 restrictions. The use of the subscript is because in Chapter 5, two more 'restrictions' are defined that move mass in different ways. The restrictions in this section are related to the restrictiveness ordering ($\leq R$) and are denoted by the subscript $R$. The latter two 'restrictions are related to a truth ordering ($\leq T$) and use the subscript $T$. 
3.4 The Restrictiveness Ordering ($\leq_R$)

The restrictiveness ordering was identified by Baldwin and is described in FRIL — Fuzzy and Evidential Reasoning in Artificial Intelligence [10]. This ordering compares whether one mass assignment is a restriction of another. In the current literature it is denoted by a simple $\leq$ symbol, but because this thesis presents several different orderings for mass assignments (which appear in Chapter 5) the ordering described by Baldwin et al will be denoted as $\leq_R$ and referred to as the restrictiveness ordering.

Simply put, a mass assignment $m_A$ is less than mass assignment $m_B$ if $m_A$ is a type-$1_R$ (Equation 3.1) or type-$2_R$ (Equation 3.2) restriction of $m_B$.

$$A \leq_R B = \text{type-}1_R(A, B) \lor \text{type-}2_R(A, B)$$

Where type-$x(A, B)$ means that mass assignment $A$ is a type-$x$ restriction of $B$.

The ordering means that if $m_A \leq_R m_B$ then $m_A$ describes a family of probability distributions that is a subset of the family of probability distributions described by $m_B$ (see Figure 3.1).

![Figure 3.1: Example Probability Families for $m_A \leq_R m_B$](image)

Like all partial ordering relations the restrictiveness ordering is
transitive. Therefore \( m_A \leq_R m_B \) if there is a chain of type-1\( R \) and type-2\( R \) operators that restricts the family of probability distributions \( m_B \) to the family of probability distributions \( m_A \).

Mass assignments \( m_A \) and \( m_B \) are said to be orthogonal if neither one can be obtained from the other by restriction i.e \( \neg(m_A \leq_R m_B) \land \neg(m_B \leq_R m_A) \).

### 3.5 The Complement of a Mass Assignment

The complement of a mass assignment \( m_A \) is denoted \( m'_A \) and has the mass assigned to the set complements of the original.

If \( m = \{ A_i : a_i \} \)

then \( m' = \{ A'_i : a_i \} \)

If \( m_A \) is defined on the powerset \( 2^C \) then the complement of \( \emptyset \) is \( C \) and vice versa. \( A'_i \) is the set complement of \( A_i \) with respect to \( C \).

### 3.6 Mass Assignment Operators

The theory of MA allows two MAs to be combined or compared using operators. If \( A \) and \( B \) are mass assignments then the result of an arbitrary operator \( * \) can be found by tabulating \( A = \{ A_i : a_i \} \) and \( B = \{ B_j : b_j \} \) (see Figure 3.2).

The result is \( C = \{ C_k : c : k \} \), where \( \{ C_k \} = \{ C_{ij} \} \), \( C_{ij} = A_i * B_j \). The mass \( c_{ij} \) is the mass assigned to the cell \( C_{ij} \) and:

\[
c_k = \sum_{C_{ij} = C_k} c_{ij} \quad (3.3)
\]
CHAPTER 3. MASS ASSIGNMENT THEORY

\[
\begin{array}{c|c|c|c}
 & \mathbf{A} & & \\
\hline
\mathbf{A} & A_1 & A_1 & A_m \\
\hline
B_1 : b_1 & C_{11} & C_{i1} & C_{m1} \\
\hline
B_j : b_j & C_{ij} & C_{ij} & C_{mj} \\
\hline
B_n : b_n & C_{1n} & C_{in} & C_{mn} \\
\end{array}
\]

Figure 3.2: Generic Mass Assignment Table

The actual mass for a cell \( (c_{ij}) \) depends on which algorithm is used to assign the mass. Whichever algorithm is used, the cell masses must follow two main constraints.

\[
\begin{align*}
\sum_j c_{ij} &= a_i \quad \forall i & (3.4) \\
\sum_i c_{ij} &= b_j \quad \forall j & (3.5)
\end{align*}
\]

Constraints 3.4 and 3.5 mean that a cell mass \( c_{ij} \) cannot be greater than either its row mass \( a_i \) or its column mass \( b_j \).

Orthogonal Maximal and Minimal Solutions

Of particular interest are the orthogonal maximal and orthogonal minimal solutions to an operator. The set of orthogonal maximal solutions for an operator is the set of results so that all valid results for \( A \ast B \) are restrictions or convex combinations of an orthogonal maximal. In this Thesis the orthogonal maximal results of an operator will be denoted \([A \ast B]_R\).
CHAPTER 3. MASS ASSIGNMENT THEORY

The set orthogonal minimal solutions is the set of orthogonal solutions which cannot be restricted to any smaller valid result. All valid results of the operator are greater than or equal to a convex combination of the orthogonal minimals. The set of orthogonal minimals of an operator is denoted as $[A \times B]_R$.

The $R$ subscript to the maximal and minimal symbols denotes that they are maximal and minimal orthogonal sets with respect to the restrictiveness ordering $\leq_R$. The need for this subscript becomes clear in Chapter 5 onwards where other orderings for mass assignments are introduced.

3.7 Individual Mass Assignment ‘Meet’ and ‘Join’

As noted in the previous section there are an infinite number of results for $A \times B$ which fulfil the constraints 3.3, 3.4 and 3.5. However, a meet or join (in the lattice sense) operator can only have a single unique result. The literature on mass assignment theory calls some operators ‘meets’ or ‘joins’ even when they are not proper lattice meets/joins. Therefore this Thesis will only call an operator a meet or join if it is a lattice operator otherwise the words will be quoted.

Baldwin et al define ‘meet’ and ‘join’ operators for mass assignments. The existing literature uses $\land$ and $\lor$ to denote the operators but these symbols are used later in this Thesis to denote various logic operators on mass assignments. Therefore the mass assignment ‘meet’ and ‘join’ will be represented here by $A \cap B$ and $A \cup B$ respectively. These symbols are used because the underlying operators are actually set union and set intersection.
CHAPTER 3. MASS ASSIGNMENT THEORY

Individual 'Meet' (∩)

Let $A$ and $B$ be two mass assignments defined on $2^C$:

\[ A = \{A_i : a_i\} \]
\[ B = \{B_j : b_j\} \]

The individual 'meet' $A \cap B$ is defined to be a convex combination of orthogonal maximal solutions $[A \cap B]_R$:

\[ A \cap B = \sum_i \alpha_i O_i \]  \hspace{1cm} (3.6)
\[ \text{where } \sum_i \alpha_i = 1 \]

Where $\{O_i\}$ is the set of orthogonal maximal solutions $[A \cap B]_R$. The solutions all conform to constraints 3.3, 3.4 and 3.5. A cell’s focal set is the intersection of its row and column sets ($C_{ij} = A_i \cap B_j$).

Despite its name the individual ‘meet’ is not a true meet because the return type is not the same as the operand type. The operands are individual mass assignments on a domain $C$ and the result is a family of mass assignments.

Individual 'Join' (∪)

The join of two mass assignments is defined in a similar way except using set union to calculate the focal set of a particular cell ($C_{ij} = A_i \cup B_j$). Using the same two mass assignments $A$ and $B$ from the previous section the join $A \cup B$ is defined to be a convex combination
of all the orthogonal minimal solutions \([A \cup B]_R\):

\[
A \cup B = \sum_{i} \alpha_i O_i
\]

where \(\sum_{i} \alpha_i = 1\)

Where \(\{O_i\}\) is the set of all orthogonal minimal solutions \([A \cup B]_R\) and the solutions all conform to constraints 3.3, 3.4 and 3.5.

Again despite its name the individual 'join' is not a true join. It suffers the same problems as the individual 'meet'; the return type is not the same as the operand type.

### 3.8 Families of Mass Assignments

Although the individual 'meet' and 'join' operators do not necessarily produce a unique mass assignment, Baldwin et al. [9, pp.523-525] and [10, pp.76-77] claim that when the operators are extended to apply to parameterised families of mass assignments then the names meet and join, in a lattice theory sense, hold.

#### The Meet and Join of Families of Mass Assignments

If \(\{O_i, \ldots, O_n\}\) is a set of orthogonal mass assignments defined on a domain \(\mathbb{C}\), then the convex combination \(A\) is a family of mass assignments parameterised over \(\alpha\).

\[
A = \sum_{i=1}^{n} \alpha_i O_i
\]

where \(\sum_{i=1}^{n} \alpha_i = 1\)
Let \( A \) and \( B \) be two such families of mass assignments over \( \mathbb{C} \).

\[
A = \sum_{i=1}^{n} \alpha_i O_i \quad \sum_{i=1}^{n} \alpha_i = 1
\]

\[
B = \sum_{j=1}^{m} \beta_j P_j \quad \sum_{j=1}^{m} \beta_j = 1
\]

One family of mass assignments \( A \) is considered less than another \( B \) in the restrictiveness ordering mentioned before if all of the members of \( A \) are less than all the members of \( B \). Formally,

\[
A \leq_R B \text{ iff } (\forall O_i \in O, \forall P_j \in P)(O_i \leq_R P_j)
\]

The meet \((A \cap B)\) and join \((A \cup B)\) for two families of mass assignments are defined as:

\[
A \cap B = \sum_{i=1}^{n} \sum_{j=1}^{m} \phi_{ij}(O_i \cap P_j) \quad \sum_{i=1}^{n} \sum_{j=1}^{m} \phi_{ij} = 1 \quad (3.8)
\]

where for any \( i, j \) \( \phi_{ij} = 0 \) if \((O_i \cap P_j) \leq_R (O_r \cap P_s)\) for any \( r, s \).

\[
A \cup B = \sum_{i=1}^{n} \sum_{j=1}^{m} \phi_{ij}(O_i \cup P_j) \quad \sum_{i=1}^{n} \sum_{j=1}^{m} \phi_{ij} = 1 \quad (3.9)
\]

where for any \( i, j \) \( \phi_{ij} = 0 \) if \((O_i \cup P_j) \leq_R (O_r \cup P_s)\) for any \( r, s \).

These definitions are taken from *A Calculus for Mass Assignments in Evidential Reasoning* [9] because the definition in FRIL — *Fuzzy and Evidential Reasoning in Artificial Intelligence* [10] appears to have a typographic error: the definitions in the later book have identical 'where' clauses for the two operators.
The Where Clauses
As described the operators are not clear. The first thing to note is that the definition of $\phi$ is impossible for both operators. Consider the meet operator, for any $i,j$ there is always an $r$ and $s$ so that $(O_i \cap P_j) \leq_R (O_r \cap P_s)$, namely when $r = i$ and $s = j$. This means that $\phi_{ij}$ is always zero and is inconsistent with the claim that all $\phi$ sum to one. Similarly for the join $(\cup)$.

The simplest solution is to redefine the 'where' clauses of each operator to use a strict less than comparator ($<_R$) rather than the less-than-or-equal-to comparator ($\leq_R$) that is there currently. With this correction the claim that the meet and the join operators follow the properties of lattice operators is better founded.

The Results
Note that the results are families of mass assignments and that mass assignments are families of probability distributions (or incomplete probability distributions if mass is assigned to the empty set).

Two families of mass assignments are equal if they describe the same set of probability distributions. For example, the family $F_1$:

$$F_1 = [\emptyset : 0.6 \{a,b\} : 0.4]$$

Is equal to the family $F_2$:

$$F_2 = \alpha [\emptyset : 0.6 \{a\} : 0.4]$$
$$+ (1 - \alpha) [\emptyset : 0.6 \{b\} : 0.4]$$

Because they describe the same partial probability distributions. $F_1$ describes the probability distributions:

$$F_1 = \emptyset : 1 - x - y \{a\} : x \{b\} : y \quad \text{where} \quad 0 \leq x + y \leq 0.4$$
CHAPTER 3. MASS ASSIGNMENT THEORY

And $F_2$ describes:

$$F_2 = \emptyset : \alpha(1 - x') + (1 - \alpha)(1 - y')$$

\begin{align*}
\{a\} &: \alpha(x') \\
\{b\} &: (1 - \alpha)(y') \\
= \emptyset &: \alpha - \alpha(x') + 1 - \alpha - (y' - \alpha(y')) \\
\{a\} &: \alpha(x') \\
\{b\} &: (1 - \alpha)(y') \\
= \emptyset &: 1 - \alpha(x') - (1 - \alpha)(y') \\
\{a\} &: \alpha(x') \\
\{b\} &: (1 - \alpha)(y') \\
\end{align*}

where $0 \leq x' + y' \leq 0.4$

with

\begin{align*}
x &= \alpha(x') \\
y &= (1 - \alpha)(y')
\end{align*}

means

$$F_2 = \emptyset : 1 - x - y$$

\begin{align*}
\{a\} &: x \\
\{b\} &: y
\end{align*}

The partial probability distributions are equal. Hence the families are equal $F_1 = F_2$. 
3.9 Fril’s Multiplicative Knowledge

‘Meet’ and ‘Join’

Any algorithm that follows the constraints placed on mass assignment tableaux (Equations 3.4 and 3.5) can be used to calculate the mass placed in a cell \( c_{ij} \). Once such method is using the product of the row and column masses.

\[
c_{ij} = a_i b_j
\]

It is used in the programming language FRIL as the basis of all its operators on mass assignments. In FRIL — Fuzzy and Evidential Reasoning in Artificial Intelligence[10] Baldwin et al. claim that if the individual operator \((O_i \circ P_j)\) is replaced by the multiplicative operator \((O_i \times P_j)\) then the algebra and lattice still hold.

Despite this claim, it is simple to construct an example that follows the FRIL multiplicative ‘meet’ equation and does not fulfil the axioms required to be a proper lattice meet. Firstly, the multiplicative ‘meet’ returns a single result instead of a family of orthogonal mass assignments like the standard meet.

If \( A \) is a mass assignment then \( A \) is the family consisting of just that one mass assignment. For example,

\[
A = \{a\} : 0.3 \quad \{a,b\} : 0.7
\]

\[
A = \sum_i \alpha_i A_i
\]

\[
= \alpha A
\]

\[
= 1(\{a\} : 0.3 \quad \{a,b\} : 0.7)
\]

The individual multiplicative meet of the mass assignment \( A \) and
itself \((A \cap A)\) is:

\[
A \cap A = \{a\} : 0.51 \ \{a, b\} : 0.49
\]

See Figure 3.3 for the tableau for the multiplicative mass assignment meet. This is the only result for the multiplicative meet and is therefore a family of one.

\[
\begin{array}{c|c|c}
\{a\} & \{a, b\} & \\
0.3 & 0.7 & \\
\{a\} & \{a\} & 0.09 \ 0.21 \\
{a, b} & \{a, b\} & 0.21 \ 0.49 \\
\end{array}
\]

Figure 3.3: Example Multiplicative Mass Assignment Meet

The multiplicative meet of the mass assignment family \(A\) and itself is:

\[
A \cap A = \sum_i \sum_j \phi_{ij}(A_i \cap A_j)
\]

\[
= \phi(A \cap A)
\]

\[
= 1(\{a\} : 0.51 \ \{a, b\} : 0.49)
\]

The family of probability distributions described by \(A \cap A\) is:

\[
A \cap A = \{a\} : 1 - x \ \{b\} : x \quad \text{where } 0 \leq x \leq 0.49
\]

However the family of probabilities described by \(A\) is:

\[
A = \{a\} : 1 - x' \ \{b\} : x' \quad \text{where } 0 \leq x' \leq 0.7
\]
\( A \cap A \) does not equal \( A \) because it describes a narrower family of probability distributions. The multiplicative meet is therefore not idempotent and does not fulfil all the requirements of a lattice meet. Similarly for the multiplicative join.

3.10 Discussion on MA Operators

This section recaps the contents of the previous few sections and makes some general comments. The main points were:

1. A mass assignment operator based upon a single focal cell function \((\star)\) has an infinite number of ways of assigning the mass to the cells of a tableau. Many of these assignments result in different solutions to the operator.

2. Some of these results belong to the set of orthogonal maximal \((\lceil A \star B \rceil_R)\) or orthogonal minimal \((\lfloor A \star B \rfloor_R)\) results.

3. Operators on individual mass assignments are not candidates for meet and join operators. Their return type is not the same as their operand type.

4. Operators based upon the maximal intersection and minimal union are a meet and a join when applied to families of mass assignments.

5. The mass assignment family operators based upon the multiplicative union and multiplicative intersection are not a meet and a join. They are not idempotent.

Despite not necessarily being lattice meets and joins any operator that conforms to the constraints of Equations 3.4 and 3.5 is a valid mass assignment operator.
3.11 Support Functions

Because a mass assignment describes a family of probability distributions it is possible to define the necessary belief and plausibility measures of a particular element of the mass assignment. If $S$ is a set from the universe of discourse $\mathcal{C}$ and $X$ is a mass assignment defined on $2^\mathcal{C}$ then the belief and plausibility of $S$ can be calculated.

Firstly, the mass assigned to a particular set can be found using:

$$m_X(X_i) = x_i$$  \hspace{1cm} (3.10)

Note that $m_X$ is simply a synonym for the mass assignment function $X$.

The Dempster–Shafer theory of evidence defines these functions, but does so under the assumption of $m(\emptyset) = 0$. As mentioned in Section 2.8 Shafer [53] defines Belief (Bel) and Plausibility ($P^*$) to be as follows:

$$\text{Bel}(S) = \sum_{X_i \subseteq S} m(X_i)$$

$$P^*_X(S) = 1 - \text{Bel}(S^c)$$

Shafer’s theory of evidence does not allow probability mass to be assigned to the empty set; therefore, $\text{Bel}_X(S) + P^*_X(S^c) = 1$.

Works that allow mass in the empty set differ as to whether the mass should be counted as part of the belief and plausibility. For example, mass assignment theory[9][10] counts $m(\emptyset)$ in its belief measure, the Transferable Belief Model (TBM) [55][56], on the other hand, does not count the empty set mass when calculating belief.

This Thesis differentiates between the two belief functions by denoting them as $\text{Bel}^+(\cdot)$ if it includes the empty set and $\text{Bel}^-(\cdot)$ if it
CHAPTER 3. MASS ASSIGNMENT THEORY

does not and defines them as:

\begin{align}
\text{Bel}^+(X_i) &= \sum_{Y_j \subseteq X_i} m(Y_j) \tag{3.12} \\
\text{Bel}^-(X_i) &= \sum_{\emptyset \neq Y_j \subseteq X_i} m(Y_j) \tag{3.13}
\end{align}

\text{Bel}^-(\cdot) is the TBM definition of belief and \text{Bel}^+(\cdot) is the mass assignment definition of belief. It should be noted that the TBM has a measure called \textit{implicability} that is identical to the mass assignment belief function \text{Bel}^+(\cdot). In this Thesis the difference in name is simply to clarify the relationship between the two belief functions.

The plausibility function for both the TBM and mass assignment theory does not count the empty set in its calculation. This Thesis therefore denotes it \text{Pl}^-() to correspond with the belief notation and it is defined as:

\begin{equation}
\text{Pl}^-(X_i) = \sum_{Y_j \cap X_i \neq \emptyset} m(Y_j) \tag{3.14}
\end{equation}

\text{Pl}^-() is the inverse of the \text{Bel}^+(\cdot) function. The inverse of the \text{Bel}^-(\cdot) function is denoted as \text{Pl}^+(\cdot) because it is identical to the \text{Pl}^-() function except it also considers the empty set:

\begin{equation}
\text{Pl}^+(X_i) = \sum_{Y_j \cap X_i \neq \emptyset \land Y_j \neq \emptyset} m(Y_j) \tag{3.15}
\end{equation}

The function \text{Pl}^+(\cdot) does not appear in either the literature on TBM or mass assignment theory. However, it is a valid measure of the basic probability mass. \text{Pl}^+(\cdot) measures the plausibility that one of the subsets of \(X_i\) is actually the real case. Without the \text{Pl}^+(\cdot) measure there is no way of making the statement ‘it is plausible that \(X\) is inconsistent.’
Together the two belief and two plausibility functions make pairings. $\text{Bel}^+(\cdot)$ and $\text{Pl}^+(\cdot)$ both consider inconsistency, $\text{Bel}^-(\cdot)$ and $\text{Pl}^-(\cdot)$ do not. $\text{Pl}^+(X) = 1 - \text{Bel}^-(X')$ and $\text{Pl}^-(X) = 1 - \text{Bel}^+(X')$ make complementary pairings.

### 3.12 Semantic Unification

Semantic unification (SU) is a method of comparing how well one mass assignment supports another. In the current literature on mass assignment theory the unification operator maps from two MAs to a probability interval $[L, U]$, where $L$ is the lower probability that the MAs are a match and $U$ is the upper probability. In effect the semantic unification is a function that maps mass assignments on arbitrary domains to mass assignments defined on truth-space (the domain $\{\top, \bot\}$). See Chapter 4 for more details about truth space.

If $X = \{X_i : x_i\}$ and $Y = \{Y_j : y_j\}$ are two mass assignments then their semantic unification $R$ can be defined as:

$$(X | Y) = R = \{R_{ij} : r_{ij}\}$$

Where

$$R_{ij} = \begin{cases} 
\{\top\} & \text{if, } (Y_j \subseteq X_i) \wedge (Y_j \neq \emptyset) \\
\{\bot\} & \text{if, } (X_i \cap Y_j = \emptyset) \wedge (Y_j \neq \emptyset) \\
\{\top, \bot\} & \text{otherwise}
\end{cases} \quad (3.16)$$

$$r_{ij} = x_i y_j \quad \text{(multiplicative semantic unification)}$$

This definition is taken from Baldwin et al. [10, p.81]. It is the standard semantic unification operator extended to handle inconsistency by assigning mass to the unknown set ($\{\top, \bot\}$) if $Y_j$ is inconsistent. It can be read as saying:

1. The truth of $X$ is $\textbf{True}$ if $Y$ supports $X$. 
2. The truth of $X$ is **False** if $Y$ denies $X$.

3. The truth of $X$ is **Unknown** if $Y$ partially supports $X$.

4. The truth of $X$ is **Unknown** if $Y$ is inconsistent.

The first three interpretations are easy to justify and the fourth (that inconsistent $Y$ tells you nothing about the truth of $X$) seems reasonable at face value, but closer inspection shows it causes problems. Chapter 7 studies the semantic unification operator in more detail and proposes alterations to it to ensure it acts in a more intuitive way.

Note that the above definition is of the multiplicative semantic unification operator. Like any mass assignment operator, semantic unification can have many different results that conform to the constraints of Equations 3.4 and 3.5.

**Point-value Semantic Unification**

The book *FRIL — Fuzzy and Evidential Reasoning in AI* gives the following definition of a multiplicative point-value semantic unification [10, p. 80]:

\[
\tau_{ij} = \left\{ \frac{\text{card}(X_i \cap Y_j)}{\text{card } Y_j} \right\} x_i y_j
\]

The probability of $\Pr(X \mid Y)$ is then given by:

\[
\Pr(X \mid Y) = \sum_{i,j} \tau_{ij}
\]

If $Y_k$ is the empty set, *ie* the mass assignment is not complete, then the masses $\tau_{ik}$ are half the product of the row and column
masses $x_i$ and $y_k$.

$$r_{ij} = \frac{(x_i y_j)}{2} \quad \text{if } y_j = \emptyset$$

As implied by its name, point-valued semantic unification provides a single probability value that represents how well one mass assignment supports the other. This point probability lies within the interval of probabilities produced by Equation 3.16.

**Semantic unification and truth-space**

Interval semantic unification is the operator that links arbitrary domains to that of the Boolean values ($\{T, \bot\}$). Mass assignments defined on the power set of Boolean values are called truth-space mass assignments. The remaining content of this Thesis is primarily concerned with defining and exploring the semantics of truth-space, truth-space mass assignments and their operators.
Chapter 4

Truth-Space

4.1 The Domain of Truth-Space Mass Assignments

Like all mass assignments, truth-space MAs describe a family of probability distributions. In particular they are distributions over the power set of the truth domain:

\[
\text{Truth-Space} = \{\emptyset, \{T\}, \{\bot\}, \{T, \bot\}\}
\]

The four elements in the power set are: the empty set (\(\emptyset\) or \(0\)), denoting inconsistency; the true set (\(\{T\}\) or \(T\)); the false set (\(\{\bot\}\) or \(F\)); and the true-false set (\(\{T, \bot\}\) or \(U\)), which denotes uncertainty.

4.2 Truth-Space and Belnap Domains

There is a lot of similarity between the domain of truth-space mass assignments and that of the set notation for Belnap’s \textit{FOUR} (Section 2.6, Table 2.5). However, there are a couple of crucial differences; so to help with differentiation between the domains another
CHAPTER 4. TRUTH-SPACE

notation is used.

The Basic Truth-Space Elements

The two elements that truth-MAs are based on are T and \( \bot \). *Verum* (T) is an indication that the proposition is true and only true (*cf* FOURR’s true \( t \), which merely indicates evidence of truth) and *falsum* (\( \bot \)) indicates that the proposition is false and only false. These two concepts are mutually exclusive. If it is stated that a proposition is verum this totally excludes the possibility that the proposition will be proven false by some means. Again, compare to FOURR’s domain, where (t) does not preclude the existence of (f).

Truth-MAs cannot actually make the assertion that a proposition is T or \( \bot \). What they can do is assign all the mass to the set \{T\} or the set \{\( \bot \}\}. It is important to notice \{T\}:1.0 and T are not the same, they indicate different things. By looking at FOURR’s sets and mass assignment truth-space, the semantics of truth-mass assignments can be better understood.

The Implicit Internal Set Connectives

Belnap uses the sets to collect evidence of the truth or falsity of a proposition. The empty set in FOURR has no evidence of either true or false and the set \{t, f\} has evidence of both; therefore FOURR has an implicit ‘and’ connective between set elements. Conversely, truth-space MAs have an implicit ‘or’ connective between set elements. The set \{T, \( \bot \}\} states that the actual truth could be either true, or false, or implicitly neither. FOURR’s \{t, f\} reads as evidence of true and evidence of false and nothing, truth-MA’s \{T, \( \bot \}\} reads only truth (verum) or only false (falsum) or neither.

The use of an ‘or’ connective within MA sets has deep implications for the interpretation of truth-MAs. For example, it is impossible using truth-MAs to assert a proposition is true and only
true, the closest possible assertion is that the proposition is either true and only true or it is neither true nor false (ie inconsistent).

Following this logic the following semantics can be given to the focal sets of truth-MAs:

- **Inconsistent (Ø)** Neither true nor false.
- **True (∋T]** True or neither true nor false.
- **False (∋⊥]** False or neither true nor false.
- **Uncertain (∋T, ⊥]** True or false or neither true nor false.

The similarities between truth-space and Σ extend beyond the set on which they are defined. Like Σ the focal elements of a truth-MA can be partially ordered on the knowledge they contain and the truth they indicate.

### 4.3 The Knowledge Ordering of the Truth Domain (≤ₖ)

The knowledge ordering (≤ₖ) is one of excluding possibilities from the set {T, ⊥}. This is a consequence of the 'or' connective that exist within the sets. As knowledge increases it restricts the possible elements in the sets. Therefore, Ø is the lower bound of the sets. The knowledge it implies has excluded both T and ⊥ ending up with neither (ie inconsistency). The upper bound is {T, ⊥}; with no knowledge no possibilities can be excluded. {T} and {⊥} being incomparable elements in between the other two sets; they have both excluded one possibility from the set.

The meet of the knowledge lattice (∘) is set union, see Figure 4.1. The knowledge join (⊗) is the set intersection, see Figure 4.2. This is the inverse of the knowledge meet and the join in Σ (see Section 2.6).
Figure 4.1: The Knowledge Meet of the Truth Domain ($\otimes$)

<table>
<thead>
<tr>
<th>$A \otimes B$</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>\emptyset</td>
<td>{T} {T, \bot} {\bot}</td>
</tr>
<tr>
<td>{T}</td>
<td>{T} {T, \bot} {T, \bot}</td>
</tr>
<tr>
<td>{T, \bot}</td>
<td>{T, \bot} {T, \bot} {T, \bot}</td>
</tr>
<tr>
<td>{\bot}</td>
<td>{\bot} {T, \bot} {T, \bot} {\bot}</td>
</tr>
</tbody>
</table>

Figure 4.2: The Knowledge Join of the Truth Domain ($\oplus$)

<table>
<thead>
<tr>
<th>$A \oplus B$</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>\emptyset</td>
<td>{T} {T, \bot} {\bot}</td>
</tr>
<tr>
<td>{T}</td>
<td>{T} {T} {T} \emptyset</td>
</tr>
<tr>
<td>{T, \bot}</td>
<td>{T} {T, \bot} {T, \bot} {\bot}</td>
</tr>
<tr>
<td>{\bot}</td>
<td>\emptyset \emptyset {\bot} {\bot}</td>
</tr>
</tbody>
</table>
4.4 The Truth Ordering of the Truth Domain ($\leq_T$)

The truth ordering of the truth-MA domain is identical to that used by $\text{FOUR}$. $\{T\}$ is obviously the upper bound, denoting the existence of truth and no falsity. The lower bound is $\{\bot\}$, which shows the maximum falsity and no truth. $\emptyset$ and $\{T, \bot\}$ are incomparable elements between, $\{T, \bot\}$ contains both $T$ and $\bot$ balancing the truth it contains. $\emptyset$ contains neither and so is balanced as well. Because the truth orderings for $\text{FOUR}$ and the truth domain are the same (barring differences in representation), the truth meet ($\land$) and the truth join ($\lor$) are the same. See Figures 4.3 and 4.4.

\begin{figure}[h]
\centering
\begin{tabular}{|c|c|c|c|c|c|}
\hline

$A \land B$ & $\emptyset$ & $\{T\}$ & $\{T, \bot\}$ & $\{\bot\}$ & $\{\bot\}$ \\
\hline
$\emptyset$ & $\emptyset$ & $\emptyset$ & $\{\bot\}$ & $\{\bot\}$ & $\{\bot\}$ \\
\hline
$\{T\}$ & $\emptyset$ & $\{T\}$ & $\{T, \bot\}$ & $\{\bot\}$ & $\{\bot\}$ \\
\hline
$\{T, \bot\}$ & $\{\bot\}$ & $\{T, \bot\}$ & $\{T, \bot\}$ & $\{\bot\}$ & $\{\bot\}$ \\
\hline
$\{\bot\}$ & $\{\bot\}$ & $\{\bot\}$ & $\{\bot\}$ & $\{\bot\}$ & $\{\bot\}$ \\
\hline
\end{tabular}
\caption{The Truth Meet of the Truth Domain ($\land$)}
\end{figure}

Remember that these are the knowledge and truth orderings for the domain that truth-MAAs are defined on and not truth and knowledge orderings for truth mass assignments themselves. These mass assignment orderings are discussed in Sections 5.3 and 5.4 in the next chapter.
4.5 The Bilattice of the Truth-MA Domain

The two partial orderings described above have a least upper bound and a greatest lower bound for each pair of elements. These comprise meets and joins on two lattices.

In addition, a negation operator for the truth ordering can be defined that is identical to that used by \texttt{FOUR} (see Figure 4.5). Internally it uses Boolean negation on the set elements.

![Figure 4.4: The Truth Join of the Truth Domain (\(\lor\))](image)

![Figure 4.5: The Truth Domain's Truth Negation Operator (\(\neg\))](image)

Taking the knowledge and truth lattices with the negation oper-
ator on the truth ordering means the two lattices can be combined into a bilattice (see Figure 5.1).

Figure 4.6: The Truth-Space Bilattice

This lattice looks similar to that used by \textsc{Four}. A superficial examination makes it seem that the truth-space lattice is simply the \textsc{Four} lattice with different symbols and flipped in the knowledge ordering. However, considering the underlying difference in semantics of the truth-space and \textsc{Four} sets the two knowledge lattices are quite different.

4.6 Discussion on Truth-Space

As has been shown in this chapter truth-space is the power set of Boolean values. Usually these values are called \textit{true} and \textit{false}; however, in order to prevent the reader making assumptions about the semantics of these values different symbols have been used.
Truth-space consists of two related lattices, which order the values by truth and knowledge respectively. The orderings are similar to those used in Belnap's FOUR with one main difference. FOUR's values uses an implicit 'and' connective within the sets truth-space uses an implicit 'or' connective. The difference in these implicit connectives means there is a distinct difference in the semantics of truth-space compared to FOUR. These semantics underlie the semantics of any mass assignment that is based upon truth-space.
Chapter 5

Truth-Space Mass Assignments

5.1 Overview

Truth-space mass assignments are mass assignments that are defined on the power set of \{T, \bot\}. Discussion and analysis of the semantics of truth-space mass assignments (t-MAs) comprises the bulk of the contribution of this Thesis.

Whilst they are simply a subset of general mass assignment theory they have several features that make them worth special discussion, which has so far been neglected in the literature. The main features are:

1. The size of the power set they are defined on is small, making many problems and calculations that use them tractable.

2. There is only one valid type-$2_R$ restriction. This makes the analysis of the results simpler.

3. The elements of the sets t-MAs are defined on have distinct semantics based on classical logic (Chapter 4). This allows the
definition of logic mass assignment operators like conjunction and disjunction in addition to the normal mass assignment operators.

**Tractability**

The cardinality of truth-space is four. This means that any operator that has truth mass assignments as operators has a tableau of sixteen cells. This has implications for the work presented in Chapter 8, where this Thesis describes the limits to the range of possible results for a truth operators. The small size of the tableaux makes the analysis of the limits simpler and their semantics easier to define. Although the results of that chapter can be generalised to mass assignments defined on domains other than truth-space the semantics of the limits becomes less clear.

**The Type-2\(_R\) Restriction**

There is only one type-2\(_R\) restriction (Equation 3.2) for truth-space mass assignments that produces any change in how the mass is distributed between the focal sets. If \(A\) is a truth mass assignment then \(A'\) is a type-2\(_R\) restriction of \(A\) that moves \(\delta\) mass from \(\{T\}\) and \(\{\bot\}\) to \(\emptyset\) and \(\{T, \bot\}\).

\[
A = \emptyset : i \{T\} : t \{\bot\} : f \{T, \bot\} : u \\
A' = \emptyset : i + \delta \{T\} : t - \delta \{\bot\} : f - \delta \{T, \bot\} : u + \delta
\]

Where, \(\delta \leq t \land \delta \leq f\)

**The Semantics**

Unlike most mass assignments the domain of truth-space mass assignments is based ultimately on elements that have clear logical
semantics. As discussed in Chapter 4 these elements can be ordered on knowledge in the same way that generic mass assignment can, but also in a truth order. Later in this chapter it will be shown that this truth order can be used to create logical conjunction and disjunction operators for truth-space mass assignments. This in turn allows truth-mass assignments to interact in ways that generic mass assignments cannot.

The key to understanding the semantics of truth-space mass assignments is defining partial orderings of mass assignments. This allows mass assignments to be compared based on the amount of knowledge used to create them and also the amount of truth in the statement they represent.

5.2 The Partial Order Limits

Finding partial orderings for truth mass assignments is more complex than finding the orderings for the truth domain they are based upon. This is because in addition to the domain elements there is also the factor of the mass assigned to those elements to be considered. Therefore, the upper and lower bounds of partial orderings of truth-MAs are not subsets of the domain \{T, \bot\}, but are instead mass assignments.

First the potential limit points of any partial orderings are identified. To mimic the truth ordering the most true and most false mass assignments are identified, respectively they are MAs where all the mass is assigned to \{T\} or all to \{\bot\}. Similarly the mass assignments containing the most and least knowledge are identified. They either assign all mass to \emptyset or \{T, \bot\}.

These points can be placed on a double Hasse diagram [49] (Figure 5.1) in a similar way to the truth domain Hasse diagram. The limit points are:
CHAPTER 5. TRUTH-SPACE MASS ASSIGNMENTS

Most inconsistent: \( \emptyset:1 \{T\}:0 \{\bot\}:0 \{T, \bot\}:0 \)

Most true: \( \emptyset:0 \{T\}:1 \{\bot\}:0 \{T, \bot\}:0 \)

Most false: \( \emptyset:0 \{T\}:0 \{\bot\}:1 \{T, \bot\}:0 \)

Most uncertain: \( \emptyset:0 \{T\}:0 \{\bot\}:0 \{T, \bot\}:1 \)

Figure 5.1: Double Hasse Diagram of the Truth-MA Limits

Unlike the Hasse diagrams for \( \text{FOUR} \) and for the truth domain, the truth mass assignment orderings contain an infinite number of mass assignments making drawing a complete Hasse diagram an impossible task. The four labelled points in Figure 5.1 are simply the limits of the orderings. The following sections describe the truth and knowledge orderings for truth-space mass assignments. The orderings take into account both the focal elements and the mass assigned to them.
5.3 The Truth-MA Knowledge Ordering($\leq_K$)

The knowledge ordering of the limits shows that the knowledge ordering for truth-space mass assignments is the reverse of the restrictiveness ordering.

The most inconsistent MA is a type-$1_R$ restriction of the most true and a type-$1_R$ restriction of the most false. Similarly, the most true and most false MAs are type-$1_R$ restrictions of the most uncertain MA.

The knowledge ordering ranks mass assignments based upon how much knowledge they contain. The more knowledge (less uncertainty) a mass assignment contains the smaller the family of probabilities that it describes. The restrictiveness ordering states $A \leq_R B$ if $A$ is a restriction (smaller family) of $B$. Therefore, $A$ less than $B$ in the restrictiveness ordering means that $B$ is greater than $A$ in the knowledge ordering (Equation 5.1).

$$A \leq_K B \equiv B \leq_R A$$

$$A \leq_K B = \text{type-}1_R(B, A) \lor \text{type-}2_R(B, A)$$

**Examples:**

For example, consider the two mass assignments $A$ and $B$:

$$A = \emptyset : 0.2 \{ \top \} : 0.3 \{ \bot \} : 0.3 \{ \top, \bot \} : 0.2$$

$$B = \emptyset : 0.2 \{ \top \} : 0.2 \{ \bot \} : 0.3 \{ \top, \bot \} : 0.3$$
\[ B \leq^K A \text{ because } A \text{ is a type-1}_R \text{ restriction of } B \text{ moving } 0.1 \text{ mass} \]
\[ \text{from } \{T, \bot\} \text{ to } \{T\}. \]

\[ C = \emptyset : 0.1 \{T\} : 0.1 \{\bot\} : 0.3 \{T, \bot\} : 0.5 \]
\[ D = \emptyset : 0.1 \{T\} : 0.2 \{\bot\} : 0.2 \{T, \bot\} : 0.5 \]

The two mass assignments \( C \) and \( D \) are incomparable because neither \( C \) nor \( D \) are restrictions of the other, therefore neither contains more knowledge than the other.

\[ E = \emptyset : 0.4 \{T\} : 0.2 \{\bot\} : 0.2 \{T, \bot\} : 0.2 \]
\[ F = \emptyset : 0.4 \{T\} : 0.2 \{\bot\} : 0.2 \{T, \bot\} : 0.2 \]

\( E \) and \( F \) are equal as the masses assigned to every focal set are equal.

\[ G = \emptyset : 0.1 \{T\} : 0.1 \{\bot\} : 0.3 \{T, \bot\} : 0.5 \]
\[ H = \emptyset : 0.2 \{T\} : 0.0 \{\bot\} : 0.2 \{T, \bot\} : 0.6 \]

\( H \) is a type-2\(_R\) restriction of \( G \) therefore the family of probability distributions of \( H \) is narrower than the family associated with \( G \), hence \( G \leq^K H \).

### 5.4 The Truth-MA Truth Ordering \((\leq_T)\)

The truth ordering can be constructed in a similar way. The knowledge (and restrictiveness) ordering are based upon two restrictions. Similar restrictions need to be constructed for the truth ordering before the ordering itself can be made.
CHAPTER 5. TRUTH-SPACE MASS ASSIGNMENTS

A \land B and A \lor B means the focal set meet or focal set join of the sets A and B for the truth ordering of truth-space (Figures 4.3 and 4.4). Although Equation 5.3 is written in general terms there is only actually one type of 'restriction' that alters any masses in a truth-space mass assignment. This 'restriction' is when \( X_j = \emptyset \) and \( X_k = \{T, \perp\} \) or vice versa. As stated before this moves mass \( \alpha \) to \( \{T\} \ (\emptyset \lor \{T, \perp\}) \) and \( \{\perp\} \ (\emptyset \land \{T, \perp\}) \).

Such a 'restriction' is only valid for truth-space mass assignments as shall be called a type-2T 'restriction'.

The Truth Ordering

Together the truth type-1T and truth type-2T 'restrictions' define the truth ordering. If mass assignment A is a type-1T or type-2T 'restriction of B then the family of (partial) probability distributions defined by A must have at least as much mass assigned to \( \{T\} \) than the family of distributions defined by B. Therefore:

\[
A \leq_T B \equiv \text{type-1}_T(B, A) \lor \text{type-2}_T(B, A) \quad (5.4)
\]

Where type-\( x \)(B, A) means that B is a type-\( x \) 'restriction' of A.

Examples:

For example, consider the two mass assignments A and B:

\[
A = \emptyset : 0.50 \{T\} : 0.2 \{\perp\} : 0.2 \{T, \perp\} : 0.10
\]

\[
B = \emptyset : 0.25 \{T\} : 0.2 \{\perp\} : 0.2 \{T, \perp\} : 0.35
\]
The MAs $A$ and $B$ are incomparable on the truth-ordering because neither is a restriction of the other.

$$C = \emptyset : 0.25 \{T\} : 0.20 \{\bot\} : 0.20 \{T, \bot\} : 0.35$$

$$D = \emptyset : 0.25 \{T\} : 0.20 \{\bot\} : 0.20 \{T, \bot\} : 0.35$$

The mass assignments $C$ and $D$ are equal as all masses are identical.

$$E = \emptyset : 0.10 \{T\} : 0.20 \{\bot\} : 0.50 \{T, \bot\} : 0.20$$

$$F = \emptyset : 0.30 \{T\} : 0.30 \{\bot\} : 0.30 \{T, \bot\} : 0.10$$

Here $E \leq_T F$ because $F$ is a 'restriction' of $E$ moving 0.1 mass from $\emptyset$ and $\{T, \bot\}$ to $\{T\}$ and $\{\bot\}$ via a type-$2_T$ 'restriction' and then moving 0.3 mass from $\{\bot\}$ to $\emptyset$ via a type-$1_T$ 'restriction'.

### 5.5 Truth Mass Assignment Operators

There are various operators that work on truth space mass assignments. The restrictiveness 'meet' and 'join' operators that are valid for general mass assignments (see Section 3.7) are also defined for truth-space mass assignments. As mentioned previously in this Thesis and by Baldwin et al. [9][10] they are lattice meet and joins when applied to families of mass assignments.

The other two sets of operators are related to the two other partial orderings already mentioned. The knowledge operators, which are related to the partial ordering $\leq_K$ and the truth operators related to the partial ordering $\leq_T$.

#### Knowledge Operators

The knowledge ordering for truth-space mass assignments is the inverse of the restrictiveness ordering. In addition, the knowledge
meet operator for truth-space is set union and the knowledge meet for truth-space is set intersection. This is the exact opposite of the operators used to construct the restrictiveness meet and join.

The simplest way to define the knowledge meet and join is to defined them as the restrictiveness join and meet respectively. If \( A \) and \( B \) are families of mass assignments where each member of the family is orthogonal with respect to the knowledge ordering then the knowledge meet \( \otimes \) and join \( \oplus \) of those families are:

\[
\begin{align*}
A \otimes B &= \sum_{i=1}^{n} \sum_{j=1}^{m} \phi_{ij}(O_i \otimes P_j) \quad \sum_{i=1}^{n} \sum_{j=1}^{m} \phi_{ij} = 1 \\
A \oplus B &= \sum_{i=1}^{n} \sum_{j=1}^{m} \phi_{ij}(O_i \oplus P_j) \quad \sum_{i=1}^{n} \sum_{j=1}^{m} \phi_{ij} = 1
\end{align*}
\]

(5.5)

where for any \( i, j \) \( \phi_{ij} = 0 \) if \( (O_i \otimes P_j) \leq_K (O_r \otimes P_s) \) for any \( r, s \).

\[
\begin{align*}
A \otimes B &= \sum_{i=1}^{n} \sum_{j=1}^{m} \phi_{ij}(O_i \oplus P_j) \quad \sum_{i=1}^{n} \sum_{j=1}^{m} \phi_{ij} = 1 \\
A \oplus B &= \sum_{i=1}^{n} \sum_{j=1}^{m} \phi_{ij}(O_i \oplus P_j) \quad \sum_{i=1}^{n} \sum_{j=1}^{m} \phi_{ij} = 1
\end{align*}
\]

(5.6)

where for any \( i, j \) \( \phi_{ij} = 0 \) if \( (O_r \oplus P_s) \leq_K (O_i \oplus P_j) \) for any \( r, s \).

**Individual Knowledge Operators**

The individual knowledge 'meet' and join are defined in the same way as the individual restrictiveness operators.

\[
A \otimes B = \sum_{i} \alpha_i O_i
\]

(5.7)

where \( \sum_{i} \alpha_i = 1 \)

Where \( \{O_i\} \) is the set of orthogonal maximal solutions \( [A \otimes B]_K \).

The solutions all conform to constraints 3.3, 3.4 and 3.5. A cell's
focal set is the union of its row and column sets \( (C_{ij} = A_i \cup B_j) \).

\[
A \oplus B = \sum_i \alpha_i O_i
\]

where \( \sum_i \alpha_i = 1 \)

Where \( \{O_i\} \) is the set of all orthogonal minimal solutions \( |A \oplus B|_K \) and the solutions all conform to constraints 3.3, 3.4 and 3.5. A cell's focal set is the intersection of its row and column sets \( (C_{ij} = A_i \cap B_j) \).

Remember that \( \leq_K \) is the inverse of \( \leq_R \), \( \otimes \equiv \cup \) and \( \oplus \equiv \cap \).

Therefore the orthogonal maximals in the knowledge ordering \( |A \ast B|_K \) are the orthogonal minimals in the restrictiveness ordering \( |A \ast B|_R \).

**Truth Operators**

The truth operators are similar to the knowledge and restrictiveness operators except they use \( \land \), \( \lor \) and \( \leq_T \) instead of the set union, set intersection and \( \leq_K \) or \( \leq_R \).

If \( A \) and \( B \) are families of mass assignments where each member of the family is orthogonal with respect to the truth ordering then the truth meet \( (\land) \) and join \( (\lor) \) of those families are:

\[
A \land B = \sum_{i=1}^{n} \sum_{j=1}^{m} \phi_{ij}(O_i \land P_j) \quad \sum_{i=1}^{n} \sum_{j=1}^{m} \phi_{ij} = 1
\]

where for any \( i, j \) \( \phi_{ij} = 0 \) if \( (O_i \land P_j) \leq_T (O_r \land P_s) \) for any \( r, s \).

\[
A \lor B = \sum_{i=1}^{n} \sum_{j=1}^{m} \phi_{ij}(O_i \lor P_j) \quad \sum_{i=1}^{n} \sum_{j=1}^{m} \phi_{ij} = 1
\]

where for any \( i, j \) \( \phi_{ij} = 0 \) if \( (O_i \lor P_j) \leq_T (O_r \lor P_s) \) for any \( r, s \).
Individual Truth Operators

The individual truth operators that the mass assignment family operators are defined upon are similar to the knowledge and restrictiveness operators. This time however they return a different set of orthogonal maximals and minimals. Rather than being orthogonal in the restrictiveness ordering they are orthogonal in the truth ordering (Equation 5.4).

\[ A \land B = \sum_i \alpha_i O_i \]  
\[ \text{where } \sum_i \alpha_i = 1 \]

Where \( \{O_i\} \) is the set of orthogonal maximal solutions \([A \land B]_T\).

The solutions all conform to constraints 3.3, 3.4 and 3.5. A cell's focal set is the conjunction of its row and column sets \( (C_{ij} = A_i \land B_j) \).

\[ A \lor B = \sum_i \alpha_i O_i \]  
\[ \text{where } \sum_i \alpha_i = 1 \]

Where \( \{O_i\} \) is the set of all orthogonal minimal solutions \([A \lor B]_T\) and the solutions all conform to constraints 3.3, 3.4 and 3.5. A cell's focal set is the disjunction of its row and column sets \( (C_{ij} = A_i \lor B_j) \).

5.6 Truth Negation

Section 3.5 defines the complement of an arbitrary mass assignment. It is based upon the complement of the focal sets in the original mass assignment. When acting upon truth-space mass assignments the operation of complementation alters the knowledge orderings. This is because it swaps the mass assigned to the focal sets \( \emptyset \) and
\{T, \bot\} as well as swapping the mass assigned to \{T\} and \{\bot\}.

**Proof that complementation alters** \( \leq_K \)

Consider the two truth mass assignments \( A \) and \( B \) such that \( A \leq_K B \):

\[
A = \emptyset : 0.0 \{T\} : 0.3 \{\bot\} : 0.5 \{T, \bot\} : 0.2 \\
B = \emptyset : 0.4 \{T\} : 0.5 \{\bot\} : 0.1 \{T, \bot\} : 0.0
\]

Their complements of each mass assignment according to Section 3.5 are:

\[
A' = \emptyset : 0.2 \{T\} : 0.5 \{\bot\} : 0.3 \{T, \bot\} : 0.0 \\
B' = \emptyset : 0.0 \{T\} : 0.1 \{\bot\} : 0.5 \{T, \bot\} : 0.4
\]

Prior to the complementation \( A \leq_K B \) because there was a type-1_R restriction moving 0.2 mass from \( \emptyset \) to \( \{T\} \) and a type-1_R restriction moving 0.4 mass from \( \{\bot\} \) to \( \emptyset \). Therefore, \( B \) is a restriction of \( A, B \leq_R A \) and hence \( A \leq_K B \).

After complementation \( B' \leq_K A' \) because there is a type-1_R restriction moving 0.4 mass from \( \{T, \bot\} \) to \( \{T\} \) and another type-1_R restriction that moves 0.2 mass from \( \{\bot\} \) to \( \emptyset \). Therefore, \( A' \) is a restriction of \( B' \), \( A \leq_R B \) and hence \( B \leq_K A \). The knowledge/restrictiveness has been reversed and therefore complementation affects knowledge.

**Negation and knowledge**

The comment made by Fitting [20] and quoted in Section 2.6 is just as valid for truth-space mass assignments as it is for the bilattices which he was originally referring to. It is desirable for a negation
operator to not change the knowledge ordering and reverse the truth ordering. Mass assignment complementation is obviously unsuitable so a different operator is needed.

This Thesis proposes Equation 5.13 as a negation operator. It simply swaps the mass assigned to the sets \{T\} and \{\bot\}, leaving the mass assigned to sets \emptyset and \{T, \bot\}.

\[
\neg A = \left\{ \begin{array}{c}
\emptyset : m_\emptyset \\
\{T\} : m_{\{\bot\}} \\
\{\bot\} : m_{\{T\}} \\
\{T, \bot\} : m_{\{T, \bot\}}
\end{array} \right. 
\]  

(5.13)

The underlying operator of this function is Boolean negation acting on the focal set elements. This follows the same basis as the truth meet (which uses Boolean conjunction internally) and the truth join (which uses Boolean disjunction internally).

The negation operator does not alter the knowledge ordering; it does not alter the \emptyset and \{T, \bot\} masses and although the \{T\} and \{\bot\} masses are altered they are incomparable under the knowledge ordering and so they do not alter the ordering when they are swapped.

**Proof that negation does not change knowledge**

Since the \emptyset and \{T, \bot\} masses are unchanged \( A \leq_K B \) means \( \neg A \leq_K \neg B \). If there is a type-1\(_R\) restriction making \( A \leq_K B \) then \( A \) has more mass assigned to \{T, \bot\} or less mass assigned to \emptyset. If there is a type-2\(_R\) restriction then both \emptyset and \{T, \bot\} are greater in \( B \) than in \( A \). Because negation does not alter the \emptyset and \{T, \bot\} masses, negation does not alter the knowledge ordering of the mass assignments.
Negation and truth

The negation operator is not a true complementation operator as although it does not change the knowledge ordering it is not completely order reversing for the truth ordering.

Proof that negation is not completely order reversing with respect to truth

Assume there are two mass assignments such that:

\[ A = \emptyset : a \{T\} : b \{\bot\} : c \{T, \bot\} : d \]
\[ B = \emptyset : \alpha \{T\} : \beta \{\bot\} : \gamma \{T, \bot\} : \delta \]

and

\[ \neg A = \emptyset : a \{T\} : c \{\bot\} : b \{T, \bot\} : d \]
\[ \neg B = \emptyset : \alpha \{T\} : \gamma \{\bot\} : \beta \{T, \bot\} : \delta \]

If \( A \leq_T B \) because \( B \) is a type-1\(_T\) restriction of \( A \) then \( b \geq \beta \) or \( c \leq \gamma \). This means that \( \neg A \) is a type-1\(_T\) restriction of \( \neg B \), because:

\[ m_B(\{\bot\}) \leq m_A(\{\bot\}) \quad \text{or} \quad m_B(\{T\}) \geq m_A(\{T\}) \]

hence

\[ m_{\neg B}(\{\bot\}) \geq m_{\neg A}(\{\bot\}) \quad \text{or} \quad m_{\neg B}(\{T\}) \leq m_{\neg A}(\{T\}) \]

This means that \( A \leq_T B \) implies \( \neg B \leq_T \neg A \) if type-1\(_T\) \((B, A)\). Therefore, negation is order reversing for type-1\(_T\) restricted mass assignments.

However if \( A \leq_T B \) due to a type-2\(_T\) 'restriction' then \( b \leq \beta \) and
c ≤ γ. This means:

\[ m_B(\{\bot\}) \leq m_A(\{\bot\}) \quad \text{and} \quad m_B(\{\top\}) \leq m_A(\{\top\}) \]

hence

\[ m_{\sim B}(\{\bot\}) \leq m_{\sim A}(\{\bot\}) \quad \text{and} \quad m_{\sim B}(\{\top\}) \leq m_{\sim A}(\{\top\}) \]

This means \( A \leq_T B \) implies \( \neg A \leq_T \neg B \) if \( \text{type-}2 \neg (B, A) \). Therefore, negation is not order reversing for type-2 restricted mass assignments.

Negation and the Bilattice

Because the negation operator only partially reverses the truth order it cannot be used to combine the truth and knowledge into a proper bilattice as described by Ginsberg. They can however be combined into a pseudo-bilattice \((t-\text{MA}, \wedge, \vee, \otimes, \otimes, \sim)\) where \( \sim \) is only partially order reversing w.r.t. truth and \( t-\text{MA} \) is the set of all truth mass assignments.

Negation and the support measures

Using the support measures defined in section 3.11 a link can be made between the various versions of the belief and plausibility of \( X \) in \( \sim A \) and \( A \).

\[
\begin{align*}
\text{Bel}^+(X) & = \text{Bel}^+(\sim X) = 1 - \text{Pl}^- (X) = 1 - \text{Pl}^- (\sim X) \\
\text{Pl}^+(X) & = \text{Pl}^+(\sim X) = 1 - \text{Bel}^- (X) = 1 - \text{Bel}^- (\sim X) \\
\text{Bel}^-(X) & = \text{Bel}^-(\sim X) = 1 - \text{Pl}^+ (X) = 1 - \text{Pl}^+ (\sim X) \\
\text{Pl}^-(X) & = \text{Pl}^-(\sim X) = 1 - \text{Bel}^+ (X) = 1 - \text{Bel}^+ (\sim X)
\end{align*}
\]
If there is no mass assigned to $\emptyset$ or $\{T, \bot\}$ then the negation acts in a similar way to probability negations: $\Pr_{-A}(X) = 1 - \Pr_A(X)$.

5.7 Discussion on Truth-MAs

This chapter has presented mass assignments defined upon truth-space — the power set of Boolean values. Parallels have been drawn with Belnap’s four-valued logic $\mathbb{F}OU\mathbb{R}$ [11][12] and using the bilattice theories proposed by Ginsberg [24] these parallels have been used to define truth and knowledge orderings on truth-space mass assignments.

The knowledge ordering is simply the inverse of the restrictiveness ordering already described in mass assignment literature by Baldwin et al and is therefore valid for all mass assignments. The truth ordering takes its inspiration from the knowledge ordering but is only valid for mass assignments defined on truth-space.

The orderings on mass assignments and their corresponding orderings of truth-space have been used to define ‘meet’ and ‘join’ operators for those orderings and a proper lattice meet and join for each ordering has been defined for families of mass assignments.

The construction of these truth operators allows mass assignments defined on truth-space to be combined in ways that are unavailable to mass assignments defined on other domains. These truth operators are underpinned by the Boolean operators acting on the focal set elements and are similar to their counterparts in $\mathbb{F}OU\mathbb{R}$.

The final operator needed to combine two lattices into a single bilattice is one of negation. This should reverse the truth order but preserve the knowledge order. This chapter showed that the mass assignment complementation operator described in current mass assignment literature is not suitable. It also presents a negation operator based upon Boolean negation that is does not alter knowledge
and whilst not totally order-reversing for truth is at least partially
order-reversing. Because a true bilattice cannot be formed Equa-
tion 5.14 is a pseudo-bilattice algebra.

\[ T = (t-MA, \wedge, \vee, \otimes, \oplus, \neg) \]  \hspace{1cm} (5.14)

All the truth-space operators have been defined using the meets,
joins and negation of truth-space sets and they derive their in-
spiration from the work by Ginsberg [23][24][25][26] and Fitting
[20][21][22] on Belnap’s four-valued logic FOUR.

Truth-MAs and other multi-valued logic

Truth-space mass assignments have many similarities with other
logic systems. The most obvious one is that if mass is not allowed
to be assigned to the empty set then the truth mass assignment:

\[ \emptyset : 0.0 \hspace{0.5cm} \{ \top \} : \alpha \hspace{0.5cm} \{ \bot \} : \beta \hspace{0.5cm} \{ \top, \bot \} : \gamma \]

is actually a support pair \([\alpha, 1-\beta]\). The programming language FRIL
normalises its mass assignments to ensure that no mass is assigned
to the empty set. In addition, its definition of semantic unification
(Equation 3.16) does not assign mass to the empty set and means
that a support pair is always produced.

Restricting truth-space mass assignments in other ways allows
t-MAs to mimic other logic systems. Consider how truth-MAs op-
erators act if their operands are restricted to having all the mass
assigned to \(\{ \top \}\) or all the mass assigned to \(\{ \bot \}\), ie the only valid
assignments are:

\[ T = \emptyset : 0.0 \hspace{0.5cm} \{ \top \} : 1.0 \hspace{0.5cm} \{ \bot \} : 0.0 \hspace{0.5cm} \{ \top, \bot \} : 0.0 \]
\[ F = \emptyset : 0.0 \hspace{0.5cm} \{ \top \} : 0.0 \hspace{0.5cm} \{ \bot \} : 1.0 \hspace{0.5cm} \{ \top, \bot \} : 0.0 \]
The truth meet and join operators in this case will act exactly like the classical logic conjunction and disjunction:

\[
\begin{align*}
T \land T &= T \\
T \land F &= F \\
F \land T &= F \\
F \land F &= F
\end{align*}
\]

and

\[
\begin{align*}
T \lor T &= T \\
T \lor F &= T \\
F \lor T &= T \\
F \lor F &= F
\end{align*}
\]

If all the mass is also allowed to be assigned to \( \{\top, \bot\} \) or all to \( \emptyset \) then the logic acts as Belnap's four-valued logic \( \forall \forall \forall \forall \). This can clearly be seen from comparing the operator tableaux Figures 2.1, 2.2, 4.3 and 4.4. Fitting [21][22] showed that Kleene's strong three-valued logic [32] is a restriction of Belnap's \( \forall \forall \forall \forall \). Therefore, by not allowing all the mass to be assigned to \( \emptyset \) and limiting the only valid assignments to:

\[
\begin{align*}
T &= \emptyset : 0.0 \quad \{\top\} : 1.0 \quad \{\bot\} : 0.0 \quad \{\top, \bot\} : 0.0 \\
F &= \emptyset : 0.0 \quad \{\top\} : 0.0 \quad \{\bot\} : 1.0 \quad \{\top, \bot\} : 0.0 \\
U &= \emptyset : 0.0 \quad \{\top\} : 0.0 \quad \{\bot\} : 0.0 \quad \{\top, \bot\} : 1.0
\end{align*}
\]

Then truth-space mass assignments act as Kleene's strong three-valued logic. Kleene's weak three-valued logic can also be used as a basis for truth-space mass assignment operators as long as the t-MAs
are limited to $T, F$ and $U$.

Truth-space mass assignments can also be restricted to act as a probabilistic logic. This is done by not allowing mass to be assigned to the focal sets $\emptyset$ or $\{T, \bot\}$ but allowing $\{T\}$ and $\{\bot\}$ to vary freely (as long as they still sum to one), e.g.

$$X = \emptyset : 0.0 \quad \{T\} : \alpha \quad \{\bot\} : 1 - \alpha \quad \{T, \bot\} : 0.0$$

This basically ensures that the family of probability distributions described by $X$ has no uncertainty and is complete. The probability of $X$ being true is:

$$Pr(X) = Bel_X^{\pm}(\{T\}) = E_X^{\pm}(\{T\}) = m_X(\{T\})$$

If there are two mass assignments $X$ and $Y$ that are both limited in the way described above then if they are assumed to be independent then:

$$Pr(X \land Y) = Pr(X) Pr(Y)$$

This is because where independence is assumed it is possible to use multiplicative versions of the truth-MA truth meet. This uses the same tableaux as Figure 4.3 but the cell masses are set to the multiplication of their row and column masses. Therefore,

$$X \land Y = \emptyset : 0.0$$

$$\{T\} : m_X(\{T\})m_Y(\{T\})$$

$$\{\bot\} : 1 - m_{X \land Y}(\{T\})$$

$$\{T, \bot\} : 0.0$$
And

\[ m_X(\{T\}) = \Pr(X) \]
\[ m_Y(\{T\}) = \Pr(Y) \]

Therefore

\[ \Pr(X \land Y) = \Pr(X) \Pr(Y) \]

In addition, the conditional probability equation can be reproduced for the restricted mass assignments.

\[ X \mid Y = \frac{X \land Y}{Y} \]
\[ \Pr_X(\{T\}) = \frac{\Pr_{(X \land Y)}(\{T\})}{\Pr_Y(\{T\})} \]

Where \( X \land Y \) is any single valid result of the truth 'meet' of the two mass assignments. It does not have to be one of the orthogonal maximals \([X \land Y]_T\), it simply has to conform to the constraints: 3.3, 3.4 and 3.5. If the multiplicative truth-'meet' is used then the conditional probability is equal to the independence equation.

**Final comments on Truth-MAs**

Truth-space mass assignments are a very flexible method of representing and reasoning with imprecise and incomplete knowledge. Unrestricted it provides a method of combining mass assignments based on the knowledge they provide (a method also available to generic mass assignments) and also based on the amount of truth they contain (a method not available to general mass assignments).

If various restrictions are placed upon the way mass can be assigned in the operands then truth-space mass assignments are able
to mimic several different logic systems such as support logic, classical logic, probabilistic logic, Kleene's three-valued logic and Belnap's FOUR.
Chapter 6

Understanding MA Operators

6.1 Introduction

Two different types of mass assignment operator have been presented in this Thesis and previously by other mass assignment literature. The most obvious example of these two different types of operator are the interval and point-value semantic unifications. These two types of operator either produce probability distributions in the case of the point-value version or a family of probabilities in the case of the 'interval' version. Other 'interval' operators already presented are the meets and joins for the various orderings of truth-space mass assignments.

Note that the name 'interval' is a slight misnomer if the operator is something like the truth-meet then some mass might be assigned to the empty set. This means that any resulting mass assignment might not be complete and does not truly describe an interval. However, the name is sufficient to distinguish them from the point-value operators.

Current literature usually presents point-value and interval oper-
CHAPTER 6. UNDERSTANDING MA OPS.

Ators as separate. This chapter examines mass assignment operators in a new way that shows the relation between the two different type of operator.

6.2 The Tableau

Interval mass assignment operators tend to be represented as an operator tableau. This Thesis takes the interval mass assignment operators as its starting point and shows how point-value operators are simply the result of applying several functions to the interval result.

The tableau is made up of cells. Current literature describes each cell assigning mass to a single focal element of the result. The total mass assignment is found by summing all the individual mass contributions made by each cell. This chapter proposes a slightly different semantics for the tableau.

Under these new semantics, each cell can be considered as returning a partial mass assignment rather than assigning mass to a single set. For example, if an imaginary operator $X \star Y$ would assign mass to the set $\{\bot\}$ then the new semantics mean it creates a partial mass assignment:

$$\emptyset : 0 \quad \{\top\} : 0 \quad \{\bot\} : f \quad \{\top, \bot\} : 0$$

Where $0 < f \leq 1$

The mass assignments are partial because their total mass (including that assigned to $\emptyset$) can be less than one. The final result of a mass assignment operator is found by summing the partial mass assignments associated with each cell. For this to produce a valid result the constraints already mentioned in Section 3.6 (Equations 3.3, 3.4 and 3.5) must still hold.
This seems a small change but this now allows a cell to assign some of its mass to one focal set and some to another, eg

\[ \emptyset : i \{ \top \} : 0 \{ \bot \} : f \{ \top, \bot \} : 0 \]

Where \( 0 < i, f \leq 1 \land i + f \leq 1 \)

For the interval operator this does not alter the result in any way at all. It is just a different view of how the operator works. Using this view it is possible to define two functions the contribution function and the distribution function.

### 6.3 The Contribution Function

The contribution function determines how much mass is assigned to each cell. It does not say where the mass goes just how much. The most obvious contribution function is the multiplicative one. Other contribution functions already mentioned in this Thesis and in other literature are the orthogonal maximal and minimal for the truth, knowledge and restrictiveness orderings. These contribution functions are all based around the min function in the same way that the multiplicative contribution function is based around multiplication.

All contribution functions must return the mass contributed by cell\(_{ij}\) it therefore needs to conform to the constraints already mentioned in Section 3.6 (Equations 3.3, 3.4 and 3.5). Given a row mass \( x_i \) and a column mass \( y_j \) the contribution function returns the mass contributed by cell\(_{ij}\)

\[ \text{Cont} : [0, 1] \times [0, 1] \to [0, 1] \]
CHAPTER 6. UNDERSTANDING MA OPS.

94

The Multiplicative Contribution Function

The multiplicative contribution function is one of the easiest to define. Given two mass assignments $X = \{X_i : x_i\}$ and $Y = \{Y_j : y_j\}$ the mass contributed by cell$_{ij}$ is:

$$\text{Cont}(x_i, y_j) = x_i y_j \quad (6.1)$$

The Minimum Contribution Function

The various orthogonal maximal and minimal operator results are based upon the min contribution function.

$$\text{Cont}(x_i, y_j) = \min \left[ x_i - a_i, y_j - b_j \right] \quad (6.2)$$

The min contribution function is more complex than the multiplicative contribution function. The numbers $a_j$ and $b_j$ are the masses that have already been assigned to cells on the same row and column as cell$_{ij}$. After calling Cont for cell$_{ij}$ the amount of mass assigned in that row/column has obviously changed. Therefore, after Cont is called on the cell, $a_i$ and $b_j$ need to be updated (designated as $a'_i$ and $b'_j$):

$$a'_i = a_i + \min \left[ x_i - a_i, y_j - b_j \right]$$
$$b'_j = b_j + \min \left[ x_i - a_i, y_j - b_j \right]$$

Because of the minimisation function, after $a_i$ and $b_j$ are updated either $a_i = x_i$ or $b_j = y_j$. This effectively causes all remaining cells on that row or column to be assigned a mass of zero.

For example, if Cont is called for a cell$_{ij}$ and no cells in the same row or column have yet been assigned then the variables are
as follows:

$$\text{Cont}(x_i, y_j) = \min \left[ x_i - a_i, y_j - b_j \right]$$

where

$$a_i = 0$$
$$b_j = 0$$

If $x_i \leq y_j$ then

$$a'_i = a_i + x_i = x_i$$
$$b'_j = b_j + x_i = x_i$$

Any subsequent call to a cell sharing the same row (cellik) will result in a mass of zero being assigned because:

$$\text{Cont}(x_i, y_k) = \min \left[ x_i - a'_i, y_k - b_k \right]$$
$$= \min \left[ x_i - x_i, y_k - b_k \right]$$
$$= \min \left[ 0, y_k - b_k \right]$$
$$= 0$$

A similar situation occurs for cells on the same column as cellij when $x_i > y_j$.

Each call to Cont changes the potential mass assigned to other (currently unassigned) cells. This makes the result returned by any semantic unification based upon the min contribution function dependent on the order in which the cells are processed. For a row by column semantic unification tableau there are $(row \times column)!$ different orderings possible. Some of these orderings will ultimately
produce the same result, but many produce distinctly different answers. Chapter 8 looks more closely at the effects of calling the min contribution function in different orders.

### 6.4 The Distribution Function

If the contribution function determines how much mass is assigned by each cell then the distribution function determines where the mass is assigned. The distribution function takes the sets $X_i$ and $Y_j$ and calculates where the mass is to be assigned.

$$\text{Dist} : 2^A \times 2^B \rightarrow (2^C \rightarrow \{0, 1\})$$

Where $(\forall X_i \in 2^A)(\forall Y_j \in 2^B) \sum_{\forall Z_k \in 2^C} \text{Dist}(X_i, Y_j)(Z_k) = 1$

Where $X_i \in 2^A$, $Y_j \in 2^B$ and $(X_i \ast Y_j) \in 2^C$. For truth-space mass assignment operators the distribution function would be:

$$\text{Dist} : 2^{\{T, \bot\}} \times 2^{\{T, \bot\}} \rightarrow (2^{\{T, \bot\}} \rightarrow \{0, 1\})$$

The result is a mapping from the focal sets of the result to a weighting that dictates where the mass is distributed. The distribution function return a function $(2^{\{T, \bot\}} \rightarrow \{0, 1\})$ rather than a simple tuple $(2^{\{T, \bot\}} \times \{0, 1\})$ because each set in $2^{\{T, \bot\}}$ maps to either 0 or 1 never both, it is a many-to-one function.
CHAPTER 6. UNDERSTANDING MA OPS.

Interval Knowledge Join

One example distribution function is the knowledge-join distribution function, defined as:

\[
\text{Dist}(X_i, Y_j)(Z_k) = \begin{cases} 
1 & \text{if } Z_k = X_i \cap Y_j \\
0 & \text{otherwise}
\end{cases}
\]  

(6.3)

And with some example sets:

\[
\begin{align*}
\text{Dist}({a}, \{a, b\})(\{\}) &= 0 \\
\text{Dist}({a}, \{a, b\})(\{a\}) &= 1 \\
\text{Dist}({a}, \{a, b\})(\{b\}) &= 0 \\
\text{Dist}({a}, \{a, b\})(\{a, b\}) &= 0
\end{align*}
\]

Interval SU

Interval semantic unification has a more complex definition:

\[
\text{Dist}(X_i, Y_j) = \begin{cases} 
\text{d}_1 & \text{if, } (Y_j \subseteq X_i) \land (Y_j \neq \emptyset) \\
\text{d}_2 & \text{if, } (X_i \cap Y_j = \emptyset) \land (Y_j \neq \emptyset) \\
\text{d}_3 & \text{otherwise}
\end{cases}
\]  

(6.4)
Where

\[
\begin{align*}
d_1(Z_k) &= \begin{cases} 
1 & \text{if } Z_k = \{T\} \\
0 & \text{otherwise}
\end{cases} \\
d_2(Z_k) &= \begin{cases} 
1 & \text{if } Z_k = \{⊥\} \\
0 & \text{otherwise}
\end{cases} \\
d_3(Z_k) &= \begin{cases} 
1 & \text{if } Z_k = \{T, ⊥\} \\
0 & \text{otherwise}
\end{cases}
\end{align*}
\]

6.5 Combining the Contribution and Distribution Functions

The contribution function determines how much mass a cell assigns and the distribution function determines where the mass goes. Therefore, the partial mass assignment associated with a cell_{ij} is:

\[
\text{cell}_{ij} = \text{Cont}(x_i, y_j) \times \text{Dist}(X_i, Y_j)
\]

This simply assigns the contributed mass to the focal set indicated by the distribution function. For example, if \(X_i : x_i = \{a\} : 0.7\) and \(Y_j : y_j = \{a, b\} : 0.3\) then multiplicative interval semantic unification would result in:

\[
\begin{align*}
\text{Cont}(0.7, 0.3) \times \text{Dist}(\{a\}, \{a, b\}) \\
&= 0.21 \times \text{Dist}(\{a\}, \{a, b\}) \\
&= 0 : 0.0 \text{ } \{T\} : 0.0 \text{ } \{⊥\} : 0.0 \text{ } \{T, ⊥\} : 0.21
\end{align*}
\]

ie \(\text{cell}_{ij}\) assigns 0.21 mass to \(\{T, ⊥\}\).

The final result of a mass assignment operator is simple the sum of the partial mass assignments associate with each cell. Of course
this means that to produce valid results the contribution and distribution functions need to be designed so that none of the constraints (Equations 3.3, 3.4 or 3.5) placed on mass assignment operators are broken.

So far this new look at how mass assignment operators work has not provided any new insight and seemingly makes the operator definitions more complex. However, now that the groundwork of the previous few sections has been laid it is now possible to define a further two functions that alter the distribution function and provide a link between point-value operators and interval operators.

### 6.6 The Consistency Function

The first of these functions is the consistency function. Occasionally mass assignment operators assign mass to the empty set thereby making the mass assignment incomplete and inconsistent. However, it is often desirable to remove this inconsistency from the result via some form of assumption. This function ‘Cons()’ is called the consistency function because it makes the mass assignment consistent. The consistency function takes the result distribution function and returns a modified version that redistributes the mass assigned to ∅ to other sets.

\[
\text{Cons} : (2^A \rightarrow [0, 1]) \rightarrow (2^A \rightarrow [0, 1])
\]

Where

\[
\text{Cons}(f)(\emptyset) = 0
\]

And

\[
\sum_{\forall Z_k \in 2^A \atop Z_k \neq \emptyset} \text{Cons}(f)(Z_k) = \sum_{\forall Z_k \in 2^A} f(Z_k)
\]
Note that unlike the distribution function the modified function can distribute mass between more than one focal set in the result. The classic consistency function is one which divides the mass that had been assigned to \( \emptyset \) equally between all the other focal sets. In effect a normalising function.

**Consistent Knowledge Join**

Take for example the knowledge join defined earlier in Equation 6.3. It distributes its mass to the empty set if \( X_i \cap Y_j = \emptyset \). If the normalisation consistency function is applied then:

\[
\text{Dist}(X_i, Y_j)(Z_k) = \begin{cases} 
1 & \text{if } Z_k = X_i \cap Y_j \\
0 & \text{otherwise}
\end{cases}
\]

Becomes

\[
\text{Cons} (\text{Dist}(X_i, Y_j))(Z_k) = 
\begin{cases} 
0 & \text{if } Z_k = \emptyset \\
\frac{1}{|X_i| - 1} & \text{if } (X_i \cap Y_j = \emptyset) \land (Z_k \neq \emptyset) \\
\text{Dist}(X_i, Y_j)(Z_k) & \text{if } (X_i \cap Y_j \neq \emptyset) \land (Z_k \neq \emptyset)
\end{cases}
\]

Different consistency functions can produce different 'normalised' mass assignment results. For example, it is possible for truth-space mass assignments to work under the assumption that inconsistency can be assumed true, or that inconsistency should be assumed false or any assumption in between. The nature of the consistency function means that it could be applied either to the distribution function of each cell to ensure there is no inconsistency in the result or it can be applied to the result itself to remove inconsistency.
6.7 The Certainty Function

The certainty function performs a similar role to the consistency function but removes uncertainty instead of consistency from the distribution function. It is a mapping from a distribution function to a new distribution function.

\[ \text{Cert} : (2^A \to [0,1]) \to (2^A \to [0,1]) \]

Where

\[ \text{Cert}(f)(Z_k) = 0 \text{ if } |Z_k| > 1 \]

And

\[ \sum_{\forall Z_k \in 2^c \atop |Z_k| \leq 1} \text{Cert}(f)(Z_k) = \sum_{\forall Z_k \in 2^c} f(Z_k) \]

The certainty function is any function that removes mass from sets with a cardinality greater than one (i.e., representing uncertainty) and distributes it between the singleton sets. This means the cell will return a partial probability distribution. The main certainty function seen so far is that used by the point-value semantic unification.

Point-Value SU

The interval semantic unification (Equation 6.4, page 97) becomes:

\[
\text{Cert}(\text{Dist}(X_i, Y_j)) = \begin{cases} 
  d_1 & \text{if, } (Y_j \subseteq X_i) \land (Y_j \neq \emptyset) \\
  d_2 & \text{if, } (X_i \cap Y_j = \emptyset) \land (Y_j \neq \emptyset) \\
  d_3 & \text{if, } Y_j = \emptyset \\
  d_4 & \text{if, } (Y_j \not\subseteq X_i) \land \\
  & (X_i \cap Y_j \neq \emptyset) \land (Y_j \neq \emptyset)
\end{cases} \quad (6.5)
\]
CHAPTER 6. UNDERSTANDING MA OPS.

Where

\[
\begin{align*}
d_1(Z_k) &= \begin{cases} 
1 & \text{if } Z_k = \{T\} \\
0 & \text{otherwise}
\end{cases} \\
d_2(Z_k) &= \begin{cases} 
1 & \text{if } Z_k = \{\bot\} \\
0 & \text{otherwise}
\end{cases} \\
d_3(Z_k) &= \begin{cases} 
\frac{1}{2} & \text{if } (Z_k = \{T\}) \lor (Z_k = \{\bot\}) \\
0 & \text{otherwise}
\end{cases} \\
d_4(Z_k) &= \begin{cases} 
\frac{|X_i \cap Y_j|}{|Y_j|} & \text{if } Z_k = \{T\} \\
1 - \frac{|X_i \cap Y_j|}{|Y_j|} & \text{if } Z_k = \{\bot\} \\
0 & \text{otherwise}
\end{cases}
\end{align*}
\]

This is not the most compact version of the point-value semantic unification using this notation, but it does show that point-value unification is simply interval unification with the mass that had been assigned to \{T, \bot\} divided between \{T\} and \{\bot\} based on the set overlaps and with a special case for \(Y_j = \emptyset\).

Least Prejudiced SU

Many different certainty functions can be defined. The overlap measure used in point-value unification can be replaced with a certainty function that divides mass that had been assigned to \{T, \bot\} equally between the sets \{T\} and \{\bot\} eg:

\[
\text{Cert}(\text{Dist}(X_i, Y_j)) = \begin{cases} 
d_1 & \text{if } (Y_j \subseteq X_i) \land (Y_j \neq \emptyset) \\
d_2 & \text{if } (X_i \cap Y_j = \emptyset) \land (Y_j \neq \emptyset) \\
d_3 & \text{otherwise}
\end{cases} \quad (6.6)
\]
CHAPTER 6. UNDERSTANDING MA OPS.

Where

\[ d_1(Z_k) = \begin{cases} 
1 & \text{if } Z_k = \{T\} \\
0 & \text{otherwise}
\end{cases} \]

\[ d_2(Z_k) = \begin{cases} 
1 & \text{if } Z_k = \{\bot\} \\
0 & \text{otherwise}
\end{cases} \]

\[ d_3(Z_k) = \begin{cases} 
\frac{1}{2} & \text{if } (Z_k = \{T\}) \lor (Z_k = \{\bot\}) \\
0 & \text{otherwise}
\end{cases} \]

Unlike the certainty function that creates the 'normal' point-value semantic unification this certainty function does not consider how the sets \(X_i\) and \(Y_j\) overlap. In effect it is the least prejudiced assumption (Section 3.2) applied to the interval semantic unification. Like the consistency function, the certainty function can be applied to either the cell or the overall result depending on the exact definition of the function.

6.8 Discussion

The representation of mass assignment operators used in current literature and in Equations 3.16 and 3.17 are certainly sufficient to describe the results they produce. However, the relationship between the interval and point-value semantic unification is obscured. It is also unclear how point-value versions of other operators can be constructed.

This chapter has shown that internally the interval mass assignment operators can be considered as the combination of a distribution function and a contribution function for determining a partial mass assignment associated with each cell in the tableau. The final result is a summation of all these partial mass assignments.
CHAPTER 6. UNDERSTANDING MA OPS.

Using the interval operator as a base it is possible to define point-value and consistent versions of the operator using two functions that alter the distribution function but retaining the same contribution function.

This has made the actual definition of mass assignment operators more complex and are unnecessary for most applications. Simpler representations of the operators, such as those used in the current literature and for the majority of this Thesis, can be used as long as it is understood that the underlying semantics of the operators are the application of a distribution function modified by a consistency and certainty function and the contribution function.
Chapter 7

Semantic Unification and Inconsistency

7.1 Overview

This chapter looks at the mass assignment operation of semantic unification in more detail. The operator provides the ability to move knowledge from an arbitrary domain to a truth-space mass assignment. Consequently, it acts as a valuation function that determines the truth of one mass assignment given another. Chapter 5 discusses the truth meet and join operators that can combine truth-space mass assignments in ways unavailable to general mass assignments. The ability to move the knowledge in a mass assignment to truth-space therefore adds flexibility to the reasoning process.

The next section looks at interval semantic unification as defined in current MA literature and revises the definition to work more intuitively. In particular it alters the definition to allow the measures $\text{Bel}^\pm$ and $\text{Pl}^\pm$ to be preserved through the valuation process. Section 7.13 then looks at point-value semantic unification. This variation of semantic unification removes uncertainty from the result and produces a 'most likely' probability that the valuation is
true. Again, changes are made to allow the unification process to operate intuitively. Finally, Section 7.14 offers some conclusions.

The majority of work in this chapter involves making alterations to the distribution function. However, to simplify the presentation of the changes the standard representation of semantic unification is used. If the reader wishes they can read \( \{Z\} \) as representing a function that distributes all the mass from that cell to the focal set \( \{Z\} \), ie:

\[
\{Z\} = d(x) = \begin{cases} 1 & \text{if } x = \{Z\} \\ 0 & \text{otherwise} \end{cases}
\]

### 7.2 Notation

A brief comment on notation. In this chapter \( X \) and \( Y \) represent generic mass assignments and \( R \) is the result of a generic semantic unification. \( C \) is a generic domain on which sets are defined. \( B, E, G, I, M, \) and \( N \) all represent specific mass assignments and are defined in the relevant sections.

### 7.3 Interval Semantic Unification

As stated before in Chapter 3, semantic unification is an operator that measures how well one mass assignment supports another. It is also called a conditional mass assignment. Restating the definition of the operator made previously gives:

\[
(X \mid Y) = R = \{R_{ij} : r_{ij}\}
\]

Where \( R_{ij} = \begin{cases} \{T\} & \text{if, } (Y_j \subseteq X_i) \land (Y_j \neq \emptyset) \\ \{1\} & \text{if, } (X_i \cap Y_j = \emptyset) \land (Y_j \neq \emptyset) \\ \{T, 1\} & \text{otherwise} \end{cases} \ (7.1) \]
\( r_{ij} = x_i y_j \) (multiplicative semantic unification)

First Equation 7.1 can be rewritten to replace the 'otherwise' case with a more exact definition.

\[(X | Y) = R = \{ R_{ij} : r_{ij} \} \]

Where \( R_{ij} = \begin{cases} 
\{ \top \} & \text{if, } (Y_j \subseteq X_i) \land (Y_j \neq \emptyset) \\
\{ \bot \} & \text{if, } (X_i \cap Y_j = \emptyset) \land (Y_j \neq \emptyset) \\
\{ \top, \bot \} & \text{if, } (X_i \cap Y_j \neq \emptyset) \land (Y_j \neq \emptyset) \\
\{ \top, \bot \} & \text{if, } (Y_j = \emptyset) 
\end{cases} \]

(7.2)

\( r_{ij} = x_i y_j \) (multiplicative semantic unification)

This rewrite does not alter the functionality of the operator. The cases are also complete and mutually exclusive. Either \( Y_j \) is the empty set (case \( \{ \top, \bot \} \)) or it is not. If it isn't then sets \( X_i \) and \( Y_j \) are either disjoint (case \( \{ \bot \} \)) or conjoint. If they are conjoint then \( Y_j \) is either a subset of \( X_i \) (case \( \{ \top \} \)) or it is not (case \( \{ \top, \bot \} \)).

This is not the most concise way of writing the operator because it contains redundant clauses. However, it does explicitly show how the operator works.

To illustrate further how semantic unification works, Figure 7.1 is a tableau showing where Equation 7.1 assigns mass depending on the sets \( X_i \) and \( Y_j \). The first thing to consider is the various ways that \( X_i \) and \( Y_j \) can interact. The situations where \( X_i \) or \( Y_j \) are the empty set are taken as special cases because they represent inconsistency. These special cases are:

1. \( X_i \) is the empty set, but \( Y_j \) is not

2. \( X_i \) is not empty set, but \( Y_j \) is
3. $X_i$ and $Y_j$ are both the empty set

If neither $X_i$ nor $Y_j$ are the empty set there are three additional ways they can interact.

1. $Y_j \subseteq X_i$
2. $X_i \cap Y_j \neq \emptyset \land Y_j \not\subseteq X_i$
3. $X_i \cap Y_j = \emptyset$

These six cases are all illustrated in Figure 7.1.

Figure 7.1: A Tabulation of the Original Semantic Unification Operator

Figure 7.2 is an example result of applying the semantic unification operator to arbitrary mass assignments. The mass assigned to each focal set in the operands, therefore to the result, is irrelevant for the purposes of this illustration.

7.4 Mass Assignments as Claims

A claim is a mass assignment that purports to state a truth about the world. Although claims may be true, the only way to discover if they
are actually true is to evaluate the claim based upon some evidence. The function that maps the claim onto the truth domain is called a valuation function. Semantic unification as defined in Equation 7.1 seems to be a good valuation function, mapping a claim \( (X) \) onto the truth domain \( \{\{T, \bot\}\} \) given some evidence \( (Y) \).

In the first part of this chapter all claims are what shall be called unitary mass assignments; mass assignments where all the mass is assigned to a single focal set. ie

\[
X = \{X_i : 1.0\} \cup \\
\{(\forall X_j \in C : X_j \neq X_i) \land X_j : 0.0\}
\]

### 7.5 Semantic Unification as a Valuation

For semantic unification to work properly it should operate in an intuitive way. Consider a claim \( (G) \), absolutely any claim can be made

<table>
<thead>
<tr>
<th>Y</th>
<th>{a}</th>
<th>{a,c}</th>
</tr>
</thead>
<tbody>
<tr>
<td>{}</td>
<td>{T, \bot}</td>
<td>{\bot}</td>
</tr>
<tr>
<td>{a}</td>
<td>{T, \bot}</td>
<td>{T, \bot}</td>
</tr>
<tr>
<td>{b}</td>
<td>{T, \bot}</td>
<td>{\bot}</td>
</tr>
<tr>
<td>{a,b}</td>
<td>{T, \bot}</td>
<td>{T, \bot}</td>
</tr>
</tbody>
</table>

Figure 7.2: An Example of the Original Semantic Unification Operator
and so $G$ is defined on the infinite set $U$ representing everything.

$$G : 2^U \rightarrow [0, 1]$$

If $G$ is a claim stating 'The grass colour is green', this can be represented as a mass assignment as so:

$$G = \text{'The grass colour is green'} = \{\text{green}\} : 1.0$$

Next assume there is some evidence $(E)$ which represents the actual state of the world. Obviously the colour of grass must actually be a colour, so therefore the evidence is based on the set of all colours. In this example the set has been simplified to $\{\text{green, red, blue}\}$ for clarity. In general the evidence can be any mass assignment and is not restricted to unitary mass assignments in the same way that the claims are. In this example case, the evidence has been collected from a relatively poor source and contains inconsistency (denoted by the assignment to $\emptyset$), green results, red results, and green/red results but no blue.

$$E = \emptyset : 0.1, \{\text{green}\} : 0.6, \{\text{red}\} : 0.1, \{\text{green, red}\} : 0.2$$

To evaluate the truth of the claim $G$ given the evidence $E$, all that need be done is to calculate $(G \mid E)$ using Equation 7.1.

$$(G \mid E) = \emptyset : 0.0, \{T\} : 0.6, \{\bot\} : 0.1, \{T, \bot\} : 0.3$$

Again this seems reasonable, but if a closer look is made it can be seen that this method does not work intuitively.
CHAPTER 7. SEMANTIC UNIFICATION

7.6 Support Functions as Questions

There are several questions that can be asked of both the evidence and the truth-valuation of the claim. These questions are all concerned with measuring the probabilities of certain sets using measures such as Bel±(), m(), and Pl±(), which return the necessary belief, probability mass and plausibility respectively.

1. Is the grass colour necessarily green?
2. Is the grass colour plausibly green?
3. Is the grass colour necessarily not green?
4. Is the grass colour plausibly not green?
5. Is the statement 'the grass colour is green' inconsistent?

For semantic unification to be an intuitive valuation function these questions should give the same answers whether they are asked of the evidence or the truth-valuation of a claim based on the evidence.

7.7 General Unitary Case for the Original SU

In general, if $X = X_i : 1.0$ is an unitary claim, with all mass assigned to a single focal element then a general piece of evidence $(Y)$ consists of four main parts when compared to the generic claim $X$, as seen in Figure 7.1. They are:

1. The empty set $\emptyset$
2. Focal Sets $S = \{S_p\}$ where $(\forall S_p \in S): S_p \subseteq X_i$
3. Focal Sets $D = \{D_q\}$ where $(\forall D_q \in D): D_q \cap X_i = 0$
4. Focal Sets $C = \{C_r\}$ where $(\forall C_r \in C) : C_r \cap X_i \neq 0 \land C_r \neq \emptyset \land C_r \nsubseteq X_i$

Together these four sets describe all the possible focal elements of $Y$ with respect to how they relate to $X_i$. And their union is the evidence $Y$.

- The empty set.
- $S$ is all the focal elements that are subsets of $X_i$.
- $D$ is all the disjoint focal elements with respect to $X_i$.
- $C$ is all the focal elements conjoint with $X_i$ but not subsets of $X_i$.

$$Y = \{\emptyset : k_0\} \cup \{S_p : l_p\} \cup \{D_q : m_q\} \cup \{C_r : n_r\}$$

Using the definition of semantic unification in Equation 7.2 the unification of the general mass assignment claim $X$ and the general evidence $Y$ is:

$$\emptyset : 0.0, \quad \{\top\} : \sum_{S_p \in S} m_X(X_i) m_Y(S_p), \quad \{\bot\} : \sum_{D_q \in D} m_X(X_i) m_Y(D_q), \quad \{\top, \bot\} : m_X(X_i) m_Y(\emptyset) + \sum_{C_r \in C} m_X(X_i) m_Y(C_r)$$
CHAPTER 7. SEMANTIC UNIFICATION

Because \( m_{X}(X_{i}) = 1 \) the result of \((X \mid Y)\) is actually:

\[
\begin{align*}
\emptyset & : 0.0, \\
\{\top\} & : \sum_{S_{p} \in S} m_{Y}(S_{p}), \\
\{\bot\} & : \sum_{D_{q} \in D} m_{Y}(D_{q}), \\
\{\top, \bot\} & : m_{Y}(\emptyset) + \sum_{C_{r} \in C} m_{Y}(C_{r})
\end{align*}
\]

The definition of \( \text{Bel}^{+}_{Y}(X_{i}) \) (Equation 3.12) means the result consists of the mass assigned to the empty set plus the mass assigned to the focal sets \( S_{p} \):

\[
\text{Bel}^{+}_{Y}(X_{i}) = m_{Y}(\emptyset) + \sum_{S_{p} \in S} m_{Y}(S_{p})
\]

However, when the same question is asked of the evidence the equation used is:

\[
\text{Bel}^{+}_{X|Y}(\{\top\}) = 0.0 + \sum_{S_{p} \in S} m_{Y}(S_{p})
\]

If \( m_{Y}(\emptyset) \) is greater than zero then the two questions are not equivalent, the measure is not preserved through the unification process. The general plausibility question for the evidence evaluates as:

\[
\text{Pl}^{+}_{Y}(X_{i}) = m_{Y}(\emptyset) + \sum_{S_{p} \in S} m_{Y}(S_{p}) + \sum_{C_{r} \in C} m_{Y}(C_{r})
\]

and the equivalent question for \((X \mid Y)\) is:

\[
\text{Pl}^{+}_{X|Y}(\{\top\}) = m_{Y}(\emptyset) + \sum_{C_{r} \in C} m_{Y}(C_{r}) + \sum_{S_{p} \in S} m_{Y}(S_{p})
\]

In the case of \( \text{Pl}^{+}(\cdot) \) there is no difference in the result of evaluating the evidence or the claim given the evidence. If the measures \( \text{Bel}^{-}(\cdot) \)
and \( \Pi^-() \) are used then they evaluate as:

\[
\begin{align*}
\Ber^-_{Y}(X) &= \sum_{s_p \in S_Y} m(S_p) \\
\Ber^-_{X|Y}(\top) &= \sum_{s_p \in S_Y} m(S_p) \\
\Pi^-_{Y}(X) &= \sum_{s_p \in S_Y} m(S_p) + \sum_{C_r \in C_Y} m(C_r) \\
\Pi^-_{X|Y}(\top) &= m(\emptyset) + \sum_{C_r \in C_Y} m(C_r) + \sum_{s_p \in S_Y} m(S_p)
\end{align*}
\]

In this case the belief is transferred correctly but the plausibility is not. Finally, the result of evaluating the claim given the evidence is more consistent than the evidence it is based upon. This seems counter intuitive.

\[
\begin{align*}
m_Y(\emptyset) &= m_Y(\emptyset) \\
\mu_{X|Y}(\emptyset) &= 0.0
\end{align*}
\]

**Example**

\[
G = \text{The grass colour is green'} = \{\text{green}\} : 1.0
\]
\[
E = \emptyset : 0.1, \{\text{green}\} : 0.6, \{\text{red}\} : 0.1, \{\text{green, red}\} : 0.2
\]
\[
(G \mid E) = \emptyset : 0.0, \top : 0.6, \bot : 0.1, \top, \bot : 0.3
\]

Returning to the first question 'Is the grass colour necessarily green?' If this question is asked of the evidence then it is equivalent to asking the evidence's belief in 'green':

\[
\Ber^+_{E}(\{\text{green}\}) = \emptyset : 0.1 + \{\text{green}\} : 0.6 = 0.7
\]

This question is also the same as asking what is the belief that the
claim ‘The grass colour is green’ is true, ie

\[ \text{Bel}^+_{G|E}(\{T\}) = \emptyset : 0.0 + \{T\} : 0.6 = 0.6 \]

As can be seen there has been a loss of 0.1 probability mass, during the valuation function. Although both equations ask the same question they give different answers. Another question that can be asked of both the evidence and the claim given the evidence is ‘What is the necessary support that the grass colour is not green’, written as:

\[ \text{Bel}^+_{G}(\{\text{green}'\}) = \text{Bel}^+_{G}(\{\text{red, blue}'\}) = \emptyset : 0.1 + \{\text{red}\} : 0.1 = 0.2 \]

And,

\[ \text{Bel}^+_{G|E}(\{T\}'\}) = \text{Bel}^+_{G|E}(\{\bot\}) = \emptyset : 0.0 + \{\bot\} : 0.1 = 0.1 \]

Again 0.1 mass has gone ‘missing’.

This problem does not occur when calculating plausibility, at least when using \( \text{Pl}^+ \).

\[ \text{Pl}^+_{E}(\{\text{green}\}) = \emptyset : 0.1 + \{\text{green}\} : 0.6 + \{\text{green, red}\} : 0.2 = 0.9 \]

\[ \text{Pl}^+_{G|E}(\{T\}) = \{T\} : 0.6 + \{T, \bot\} : 0.3 = 0.9 \]

And,

\[ \text{Pl}^+_{G}(\{\text{red, blue}\}) = \emptyset : 0.1 + \{\text{red}\} : 0.1 + \{\text{green, red}\} : 0.2 = 0.4 \]

\[ \text{Pl}^+_{G|E}(\{\bot\}) = \{\bot\} : 0.1 + \{T, \bot\} : 0.3 = 0.4 \]
For $\text{Bel}_{E}^{-}(\cdot)$ and $\text{Pl}_{E}^{-}(\cdot)$ the results are:

$$
\begin{align*}
\text{Bel}_{E}^{-}(\{\text{green}\}) &= 0.6 \\
\text{Bel}_{E}^{-}(\{\top\}) &= 0.6 \\
\text{Bel}_{E}^{-}(\{\text{red, blue}\}) &= 0.1 \\
\text{Bel}_{E}^{-}(\{\bot\}) &= 0.1 \\
\text{Pl}_{E}^{-}(\{\text{green}\}) &= 0.8 \\
\text{Pl}_{E}^{-}(\{\top\}) &= 0.9 \\
\text{Pl}_{E}^{-}(\{\text{red, blue}\}) &= 0.3 \\
\text{Pl}_{E}^{-}(\{\bot\}) &= 0.4
\end{align*}
$$

And finally the example claim given the evidence is more consistent than the evidence it is based upon.

$$
\begin{align*}
\text{m}_{E}(\emptyset) &= 0.1 \\
\text{m}_{E}(\emptyset) &= 0.0
\end{align*}
$$

As stated before, this discrepancy in belief, and plausibility seems counter-intuitive. In fact the lack of inconsistency preservation is linked to the mistakes in the calculation of belief and plausibility.

### 7.8 Updated Interval Semantic Unification

One way to solve the problem is to redefine semantic unification so that it preserves inconsistency and allows belief and plausibility to
be transferred correctly.

$$ (X | Y) = R = \{ R_{ij} : r_{ij} \} $$

$$ R_{ij} = \begin{cases} 
\emptyset & \text{if, } Y_j = \emptyset \\
\{ \top \} & \text{if, } (Y_j \subseteq X_i) \land (Y_j \neq \emptyset) \\
\{ \bot \} & \text{if, } (X_i \cap Y_j = \emptyset) \land (Y_j \neq \emptyset) \\
\{ \top, \bot \} & \text{if, } (Y_j \not\subseteq X_i) \land (Y_j \cap X_i \neq \emptyset) \land (Y_j \neq \emptyset) 
\end{cases} \quad (7.3) $$

As before the equation can also be represented as a tableau (Figure 7.3). The only change from Figure 7.1 is the assignment of mass when $Y_j = \emptyset$.

![Figure 7.3: A Tabulation of the Updated Semantic Unification Operator](image)

If inconsistency implies everything why does the updated unification operator assign mass to $\emptyset$ and not to $\{ \top, \bot \}$ when $Y_j = \emptyset$? As can be seen from the sub-section 7.7 assigning to both does not preserve belief or plausibility, depending on which measure is used. From a more philosophical point of view, inconsistency ($\emptyset$) does not just imply $\top$ or $\bot$, instead it implies both $\top$ and $\bot$ at the same
time *ie* inconsistency. Altering the operator to assign mass to the empty set in this case was the sole outcome of this update of the unification operator. As can be seen in the sub-section below the alteration 'fixes' the belief and plausibility measures.

**General Unitary Case**

Going back to the general case the unification of $X$ and $Y$ is now:

$$
\emptyset : m(\emptyset),
\{T\} : \sum_{S_p \in S} m(S_p),
\{\bot\} : \sum_{D_q \in D} m(D_q),
\{T, \bot\} : \sum_{C_r \in C} m(C_r)
$$

The belief in the general claim for the evidence has not changed:

$$
\text{Bel}^+_Y(X_i) = m(\emptyset) + \sum_{S_p \in S} m(S_p)
$$

However, now the result of evaluating $\text{Bel}^+_{X|Y}(\{T\})$ has changed to preserve its value.

$$
\text{Bel}^+_{X|Y}(\{T\}) = m(\emptyset) + \sum_{S_p \in S} m(S_p)
$$

The plausibility measures stay the same:

$$
\text{Pl}^+_Y(X_i) = \sum_{S_p \in S} m(S_p) + \sum_{C_r \in C} m(C_r)
$$

$$
\text{Pl}^+_{X|Y}(\{T\}) = \sum_{C_r \in C} m(C_r) + \sum_{S_p \in S} m(S_p)
$$
CHAPTER 7. SEMANTIC UNIFICATION

The changes made in Equation 7.3 means that now the equations for Bel±() and Pl± now all preserve their values through the unification process.

\[
\begin{align*}
\text{Bel}_Y^+(X_i) &= \sum_{S_p \in S} m_Y(S_p) \\
\text{Pl}_Y^+(X_i) &= \sum_{S_p \in S} m_Y(S_p) + \sum_{C_r \in C} m_Y(C_r) \\
\text{Bel}_{X|Y}^+\{\top\} &= \sum_{S_p \in S} m_Y(S_p) \\
\text{Pl}_{X|Y}^+\{\top\} &= \sum_{C_r \in C} m_Y(C_r) + \sum_{S_p \in S} m_Y(S_p)
\end{align*}
\]

In addition, now the claim given the evidence is no longer more consistent that the evidence it is based upon.

\[m_{X|Y}(\emptyset) = m_Y(\emptyset)\]

Example

Using this new definition of semantic unification (Equation 7.3) as a valuation function the new resulting truth-MA is:

\[G = \{\text{green}\} : 1.0\]
\[E = \emptyset : 0.1, \{\text{green}\} : 0.6, \{\text{red}\} : 0.1, \{\text{green, red}\} : 0.2\]
\[(G \mid E) = \emptyset : 0.1, \{\top\} : 0.6, \{\bot\} : 0.1, \{\top, \bot\} : 0.2\]

Note that this time the semantic unification is not more consistent than its evidence. If the same questions given in Section 7.6 are asked again on this new valuation the following results are obtained:

\[\begin{align*}
\text{Bel}_E^+(\{\text{green}\}) &= 0.1 + 0.6 = 0.7 \\
\text{Pl}_E^+(\{\text{green}\}) &= 0.9 \\
\text{Bel}_{E|G}^+(\{\top\}) &= 0.1 + 0.6 = 0.7 \\
\text{Pl}_{E|G}^+(\{\top\}) &= 0.9 \\
\text{Bel}_E^+(\{\text{red, blue}\}) &= 0.1 + 0.1 = 0.2 \\
\text{Pl}_E^+(\{\text{red, blue}\}) &= 0.4 \\
\text{Bel}_{E|G}^+(\{\bot\}) &= 0.1 + 0.1 = 0.2 \\
\text{Pl}_{E|G}^+(\{\bot\}) &= 0.4
\end{align*}\]
And,

\[
\begin{align*}
\text{Bel}_E(\{\text{green}\}) &= 0.6 & \text{Pl}_E(\{\text{green}\}) &= 0.8 \\
\text{Bel}_E(\{T\}) &= 0.6 & \text{Pl}_E(\{T\}) &= 0.8 \\
\text{Bel}_E(\{\text{red, blue}\}) &= 0.1 & \text{Pl}_E(\{\text{red, blue}\}) &= 0.3 \\
\text{Bel}_E(\{\bot\}) &= 0.1 & \text{Pl}_E(\{\bot\}) &= 0.3
\end{align*}
\]

Also,

\[
\begin{align*}
\text{m}_E(\emptyset) &= 0.1 \\
\text{m}_{G|E}(\emptyset) &= 0.1
\end{align*}
\]

Both belief and plausibility are now calculated consistently. In addition, the truth-valuation of the claim \(G\) is no more, nor less, consistent than the evidence \(E\).

### 7.9 Discussion

The difference between the two definitions of semantic unification is in the way they handle inconsistent evidence. The function proposed by Baldwin et al. (Equation 7.1) states that inconsistent evidence means that the truth of the claim is unknown (\(\{T, \bot\}\)). The new definition in this chapter (Equation 7.3) states that inconsistent evidence means that the truth of the claim given the evidence is inconsistent. This second interpretation appears much more intuitive and also allows inconsistency, mass, belief and plausibility to be preserved through the valuation function.
7.10 More Claims

One final claim that could be tested is 'The grass colour is inconsistent' denoted as claim 1. This is different from 'Is the claim "the grass colour is green" inconsistent?'. The first is a claim (ie a MA), the second is a question (ie a probability function such as Bel+()). The claim that 'the grass colour is inconsistent' could be used to check whether the evidence is consistent. Although the same information can be found by asking Bel+(θ) of the evidence (as done in Section 7.6), the claim is a valid one and should therefore work in an intuitive way. It can be written as:

\[ X = \emptyset : 1.0 \]

Using the updated semantic unification (Equation 7.3) produces the result:

\[ \emptyset : m(\emptyset), \]
\[ \{\top\} : 0.0 \]
\[ \{\bot\} : \sum_{S_p \in S} m(S_p) + \sum_{D_q \in D} m(D_q) + \sum_{C_r \in C} m(C_r), \]
\[ \{\top, \bot\} : 0.0 \]
Assessing the measure Bel+() and Pl+() gives:

\[ \text{Bel}^+ (\emptyset) = Y_Y \]
\[ \text{Bel}^+ (\{T\}) = Y_Y \]
\[ \text{Pl}^+ (\emptyset) = m(0) + \sum_{S_p \in S} m(S_p) + \sum_{D_q \in D} m(D_q) + \sum_{C_r \in C} m(C_r) = 1.0 \]
\[ \text{Pl}^+ (\{T\}) = Y_Y \]

Plausibility in the empty set is not preserved. Note that the measures Bel−() and Pl−() do not consider the empty set in their calculation. Using the updated semantic unification the two measures preserve their values.

\[ \text{Bel}^− (\emptyset) = \text{Bel}^− (\{T\}) = \text{Pl}^− (\emptyset) = \text{Pl}^− (\{T\}) = 0.0 \]

**Example**

Using the same evidence as before, Equation 7.3 produces the result:

\[ I = \emptyset : 1.0 \]
\[ E = \emptyset : 0.1, \{\text{green}\} : 0.6, \{\text{red}\} : 0.1, \{\text{green, red}\} : 0.2 \]
\[ (I \mid E) = \emptyset : 0.1, \{\top\} : 0.0, \{\bot\} : 0.9, \{\top, \bot\} : 0.0 \]

The evidence shows that the necessary support for the grass colour being inconsistent is:

\[ \text{Bel}^+ (\emptyset) = 0.1 \]

Consequently, it is reasonable to suppose the necessary support for the claim being true would be the same.

\[ \text{Bel}^+ (\{T\}) = 0.1 \]
Similarly with plausibility:

\[ P_I^+(\emptyset) = 1.0 \]

\[ P_I^+\{T\} = 0.1 \]

Again there is a problem with the definition of semantic unification, plausibility is not being preserved.

### 7.11 The Final Version of Interval Semantic Unification

This final revision of semantic unification modifies the operator again to allow Bel^+() and Pl^+() to preserve their values through the unification process when \( X_i = \emptyset \). The semantic unification of \((\emptyset | \emptyset)\) should be true, because if the evidence is inconsistent then the claim is correct. Also \((\emptyset | Y)\) should give \{\(T, \bot\}\} if \(Y\) is not the empty set. This requires a further change to the definition of semantic unification.

\[
(X | Y) = R = \{R_{ij} : r_{ij}\}
\]

Where

\[
R_{ij} = \begin{cases} 
\emptyset & \text{if, } (Y_j = \emptyset) \land (X_i \neq \emptyset) \\
\{T\} & \text{if, } (Y_j = X_i = \emptyset) \\
\{T\} & \text{if, } (Y_j \subseteq X_i) \land (X_i \neq \emptyset) \\
& \quad \land (Y_j \neq \emptyset) \\
\{\bot\} & \text{if, } (X_i \cap Y_j = \emptyset) \land (X_i \neq \emptyset) \\
& \quad \land (Y_j \neq \emptyset) \\
\{T, \bot\} & \text{if, } (Y_j \cap X_i \neq \emptyset) \land (Y_j \not\subseteq X_i) \\
\{T, \bot\} & \text{if, } (X_i = \emptyset) \land (Y_j \neq \emptyset) 
\end{cases}
\]

(7.4)
The new definition of semantic unification may seem more complex than the one given in Equation 7.3, but it simply adds the additional assertion that \((\emptyset \mid Y) = \{\top\}\) if \(Y = \emptyset\) otherwise \((\emptyset \mid Y) = \{\top, \bot\}\). This can be seen in Figure 7.4. This is not the most concise way of representing the operator, but it does show how the cases are exclusive and collectively exhaustive.

![Figure 7.4: A Tabulation of the Final Semantic Unification Operator](image)

**General Unitary Case**

Using the new definition of semantic unification (Equation 7.4) gives the result of unifying \(X = \emptyset : 1.0\) and a generic evidence \(Y\) as:

\[
\begin{align*}
\emptyset : 0.0, \\
\{\top\} : m(\emptyset), \\
\{\bot\} : 0.0, \\
\{\top, \bot\} : \sum_{S_p \in S} m(S_p) + \sum_{D_q \in D} m(D_q) + \sum_{C_r \in C} m(C_r)
\end{align*}
\]
CHAPTER 7. SEMANTIC UNIFICATION

The makes the belief and plausibility:

\[
\text{Bel}_Y^+(\emptyset) = m(\emptyset)
\]

\[
\text{Bel}_X^+(\{T\}) = m(\emptyset)
\]

\[
\text{Pl}_Y^+(\emptyset) = m(\emptyset) + \sum_{S_p \in S} m(S_p) + \sum_{D_q \in D} m(D_q) + \sum_{C_r \in C} m(C_r) = 1.0
\]

\[
\text{Pl}_X^+(\{T\}) = m(\emptyset) + \sum_{S_p \in S} m(S_p) + \sum_{D_q \in D} m(D_q) + \sum_{C_r \in C} m(C_r) = 1.0
\]

Using Bel\(^-\)(\(\emptyset\)) and Pl\(^-\)(\(\emptyset\)) results in:

\[
\text{Bel}_Y^-(\emptyset) = 0.0
\]

\[
\text{Bel}_X^-(\{T\}) = m(\emptyset)
\]

\[
\text{Pl}_Y^-(\emptyset) = 0.0
\]

\[
\text{Pl}_X^-(\{T\}) = m(\emptyset) + \sum_{S_p \in S} m(S_p) + \sum_{D_q \in D} m(D_q) + \sum_{C_r \in C} m(C_r) = 1.0
\]

Compared to the values of Bel\(^-\)(\(\emptyset\)) and Pl\(^-\)(\(\emptyset\)) using the previous version of semantic unification it appears that the alteration made to allow Bel\(^+\)(\(\emptyset\)) and Pl\(^+\)(\(\emptyset\)) to preserve their values has caused the other measures to 'break'.

Consider what the claim and the measures are saying. Both '\(-\)ve' versions of the measure explicitly do not count the mass assigned to the empty set. Therefore, asking the question 'What is the necessary belief that \(Y\) is inconsistent?' using the Bel\(^-\)(\(\emptyset\)) always returns zero as does Pl\(^-\)(\(\emptyset\)).

However, if it is claimed that the world is inconsistent then either the evidence bears this claim out or it does not. The value of Bel\(^-\)(\(X|Y\))\((\{T\})\) should therefore provide a measure of where the claim is true or not. Similarly, for the plausibility measure Pl\(^-\)(\(\{T\}\)).

Looking at the consistency measure it is clear that inconsistency
CHAPTER 7. SEMANTIC UNIFICATION

is not preserved through the operation:

\[ m(\emptyset) = m(\emptyset) \]
\[ m(\emptyset) = 0.0 \]

Again, it may seem strange therefore that the final version of semantic unification (Equation 7.4) appears not to maintain it in all situations. Closer inspection shows that the only time when semantic unification produces a result more consistent than the evidence is with the claim ‘The evidence is inconsistent’. This is intuitively what should occur; either the evidence is inconsistent or it is not. It is possible to be uncertain as to whether the claim ‘The evidence is inconsistent’ is true or not, but the claim cannot itself be inconsistent.

Example

Recalculating \((I | E)\) using the new definition gives:

\[ I = \emptyset : 1.0 \]
\[ E = \emptyset : 0.1, \{\text{green}\} : 0.6, \{\text{red}\} : 0.1, \{\text{green, red}\} : 0.2 \]
\[(I | E) = \emptyset : 0.0, \{\top\} : 0.1, \{\bot\} : 0.0, \{\top, \bot\} : 0.9 \]

Again by testing it can be shown that the belief using the evidence has not changed and neither has the belief in the claim given the evidence:

\[ \text{Bel}^{+}_E(\emptyset) = 0.1 \]
\[ \text{Bel}^{+}_{I|E}(\{\top\}) = 0.1 \]
However, now plausibility is preserved:

\[
\begin{align*}
\text{Pl}^+(\emptyset) &= 1.0 \\
\text{Pl}^+(\{T\}) &= 1.0
\end{align*}
\]

And,

\[
\begin{align*}
\text{Bel}^-(\emptyset) &= 0.0 \\
\text{Bel}^-(\{T\}) &= 0.1 \\
\text{Pl}^-(\emptyset) &= 0.0 \\
\text{Pl}^-(\{T\}) &= 1.0
\end{align*}
\]

7.12 Extending the Claims

In prior sections the claims have been limited to unitary mass assignments, where all the mass is assigned to a single focal set. A more general type of claim would be one consisting of an arbitrary mass assignment:

\[
X = \{X_i : x_i\}, \quad \sum_{x_i \in X} x_i = 1.0
\]

The semantic unification operator works with the more general claim without modification. However, evaluating the operator is more difficult. Consider the following mass assignments; although they are relatively simple the point they make stands for any mass assignment.

\[
\begin{align*}
X &= \{a\} : 0.5 \quad \{b\} : 0.5 \\
Y &= \{b\} : 0.25 \quad \{c\} : 0.75 \\
X | Y &= \{T\} : 0.125 \quad \{\bot\} : 0.875
\end{align*}
\]
The multiplicative unification \( X \mid Y \), which is the unification operator used in the previous sections, results in the tableau shown in Figure 7.5.

\[
\begin{array}{cc}
(a):0.5 & (b):0.5 \\
\hline
\{\bot\} & \{\bot\} \\
0.125 & 0.375 \\
\{\top\} & \{\bot\} \\
0.125 & 0.375 \\
\end{array}
\]

Figure 7.5: Multiplicative Mass Assignment Example

Analysing the tableau in detail shows that the multiplicative semantic unification results in the following probability masses.

\[
m_{X\mid Y}(\{a\} \mid \{b\}) = 0.125 \\
m_{X\mid Y}(\{a\} \mid \{c\}) = 0.375 \\
m_{X\mid Y}(\{b\} \mid \{b\}) = 0.125 \\
m_{X\mid Y}(\{b\} \mid \{c\}) = 0.375
\]

The multiplicative mass assignment makes an assumption of independence between \( X \) and \( Y \). Under such an assumption the mass of each cell \( (X_i \mid Y_j) \) is the multiplication of the row and column mass. Given a general claim and a general piece of evidence the belief in a claim being true is the sum of all cells that equal \( \{\top\} \):

\[
\text{Bel}^\Gamma(\{\top\}) = \sum_{i=1}^{m} \sum_{j=1}^{n} m_{X\mid Y}(X_i \mid Y_j) \quad \text{Where,} \quad (X_i \mid Y_j) = \{\top\}
\]
Separating out the case where the claim and the evidence are inconsistent gives (from Equation 7.4):

\[
\begin{align*}
&= m_{X|Y}(\emptyset | \emptyset) + \sum_{i=1}^{m} \sum_{\emptyset \neq Y_j \subseteq X_i} m(X_i) m(Y_j) \\
&= m_X(\emptyset) m_Y(\emptyset) + \sum_{i=1}^{m} m(X_i) \text{Bel}^{-}(X_i) \\
&= m_X(\emptyset) m_Y(\emptyset) + \sum_{i=1}^{m} m(X_i) \text{Bel}^{-}(X_i)
\end{align*}
\]

(see definition of \( \text{Bel}^{-}(\cdot) \) in Equation 3.13)

Note that the \( m_{X|Y}(\emptyset | \emptyset) \) is not the same as \( m_{X|Y}(\emptyset) \). The first is part of \( m_{X|Y}(\{T\}) \) the second is not.

### 7.13 Point-Value Semantic Unification

So far this chapter has looked at interval semantic unification. The updated version of the operator does not strictly return a probability interval any more because of the introduction of the empty set; however, many similarities still exist.

As mentioned in Chapter 6 on MA operators applying a certainty function to the semantic unification distribution function allows the creation of a point-value version of semantic unification. Applying the certainty function to the original semantic unification operator gives the point-value semantic unification given in Equation 6.5 repeated here for clarity.
CHAPTER 7. SEMANTIC UNIFICATION

\[
\text{Cert}(\text{Dist}(X_i, Y_j)) = \begin{cases} 
  d_1 & \text{if, } (Y_j \subseteq X_i) \land (Y_j \neq \emptyset) \\
  d_2 & \text{if, } (X_i \cap Y_j = \emptyset) \land (Y_j \neq \emptyset) \\
  d_3 & \text{if, } Y_j = \emptyset \\
  d_4 & \text{if, } (Y_j \nsubseteq X_i) \land (X_i \cap Y_j \neq \emptyset) \land (Y_j \neq \emptyset)
\end{cases}
\]

Where

\[
\begin{align*}
  d_1(Z_k) &= \begin{cases} 
    1 & \text{if } Z_k = \{\top\} \\
    0 & \text{otherwise}
  \end{cases} \\
  d_2(Z_k) &= \begin{cases} 
    1 & \text{if } Z_k = \{\bot\} \\
    0 & \text{otherwise}
  \end{cases} \\
  d_3(Z_k) &= \begin{cases} 
    \frac{1}{2} & \text{if } (Z_k = \{\top\}) \lor (Z_k = \{\bot\}) \\
    0 & \text{otherwise}
  \end{cases} \\
  d_4(Z_k) &= \begin{cases} 
    \frac{|X_i \cap Y_j|}{|Y_j|} & \text{if } Z_k = \{\top\} \\
    1 - \frac{|X_i \cap Y_j|}{|Y_j|} & \text{if } Z_k = \{\bot\} \\
    0 & \text{otherwise}
  \end{cases}
\end{align*}
\]

Point-values and Inconsistency

The previous work in Sections 7.3–7.11 shows that it is not generally intuitive to have any valuation that is more consistent than the evidence used to test it. The FRIL point-value unification does not preserve inconsistency, nor does it handle inconsistent claims correctly. Following the reasoning given in Section 7.8, changes need to be made to point-value semantic unification for it to handle inconsistency correctly.
The original point-value unification is simply the original interval unification with the mass that would be assigned to \{T, \bot\} divided between \{T\} and \{\bot\} based on the overlap measure. It therefore seems reasonable that the updated point-value unification is based upon the updated interval unification.

Rewriting the updated semantic unification to show its distribution function more explicitly gives:

\[
\text{Dist}(X_i, Y_j) = \begin{cases} 
  d_1 & \text{if, } (Y_j = \emptyset) \land (X_i \neq \emptyset) \\
  d_2 & \text{if, } (Y_j = X_i = \emptyset) \\
  d_2 & \text{if, } (Y_j \subseteq X_i) \land (X_i \neq \emptyset) \\
  & \land (Y_j \neq \emptyset) \\
  d_3 & \text{if, } (X_i \cap Y_j = \emptyset) \land (X_i \neq \emptyset) \\
  & \land (Y_j \neq \emptyset) \\
  d_4 & \text{if, } (Y_j \not\subseteq X_i) \\
  d_4 & \text{if, } (X_i = \emptyset) \land (Y_j \neq \emptyset) \\
\end{cases}
\]

Where

\[
\begin{align*}
  d_1(Z_k) &= \begin{cases} 
    1 & \text{if } Z_k = \emptyset \\
    0 & \text{otherwise} \\
  \end{cases} \\
  d_2(Z_k) &= \begin{cases} 
    1 & \text{if } Z_k = \{T\} \\
    0 & \text{otherwise} \\
  \end{cases} \\
  d_3(Z_k) &= \begin{cases} 
    1 & \text{if } Z_k = \{\bot\} \\
    0 & \text{otherwise} \\
  \end{cases} \\
  d_4(Z_k) &= \begin{cases} 
    1 & \text{if } Z_k = \{T, \bot\} \\
    0 & \text{otherwise} \\
  \end{cases}
\end{align*}
\]
Applying a similar certainty function to the one that turns the original semantic unification into a point-value unification gives:

\[
\text{Cert}(\text{Dist}(X_i, Y_j)) = \begin{cases} 
  d_1 & \text{if, } (Y_j = \emptyset) \land (X_i \neq \emptyset) \\
  d_2 & \text{if, } (Y_j = X_i = \emptyset) \\
  d_2 & \text{if, } (Y_j \subseteq X_i) \land (X_i \neq \emptyset) \\
  & \land (Y_j \neq \emptyset) \\
  d_3 & \text{if, } (X_i \cap Y_j = \emptyset) \land (X_i \neq \emptyset) \\
  & \land (Y_j \neq \emptyset) \\
  d_4 & \text{if, } (Y_j \cap X_i \neq \emptyset) \land (Y_j \not\subseteq X_i) \\
  & \land (Y_j \neq \emptyset) \\
  d_4 & \text{if, } (X_i = \emptyset) \land (Y_j \neq \emptyset) 
\end{cases}
\]

Where

\[
\begin{align*}
  d_1(Z_k) & = \begin{cases} 
  1 & \text{if } Z_k = \emptyset \\
  0 & \text{otherwise}
\end{cases} \\
  d_2(Z_k) & = \begin{cases} 
  1 & \text{if } Z_k = \{\top\} \\
  0 & \text{otherwise}
\end{cases} \\
  d_3(Z_k) & = \begin{cases} 
  1 & \text{if } Z_k = \{\bot\} \\
  0 & \text{otherwise}
\end{cases} \\
  d_4(Z_k) & = \begin{cases} 
  \left(\frac{|X_i \cap Y_j|}{|Y_j|}\right) & \text{if } Z_k = \{\top\} \\
  \left(1 - \frac{|X_i \cap Y_j|}{|Y_j|}\right) & \text{if } Z_k = \{\bot\} \\
  0 & \text{otherwise}
\end{cases}
\end{align*}
\]

Note that the only actual change is that mass that had been assigned to the set \(\{\top, \bot\}\) is now divided between \(\{\top\}\) and \(\{\bot\}\) \((d_4)\). Unlike the original point-value unification mass is not divided
between \( \{ \top \} \) and \( \{ \bot \} \) if \( Y_j = \emptyset \). Due to the changes made in this chapter this mass is assigned to the empty set or to \( \{ \top \} \) depending the value of \( X_i \).

If no inconsistency is desired in the result of either the interval or the point-value unification then a consistency function can be applied to the distribution functions. This would in effect change the function \( d_1 \). This makes the definition of consistent point-value semantic unification:

\[
\text{Cons}(\text{Cert}(\text{Dist}(X_i, Y_j))) = \begin{cases} 
  d_1 & \text{if, } (Y_j = \emptyset) \land (X_i \neq \emptyset) \\
  d_2 & \text{if, } (Y_j = X_i = \emptyset) \\
  d_2 & \text{if, } (Y_j \subseteq X_i) \land (X_i \neq \emptyset) \\
 & \quad \land (Y_j \neq \emptyset) \\
  d_3 & \text{if, } (X_i \cap Y_j = \emptyset) \land (X_i \neq \emptyset) \land (Y_j \neq \emptyset) \\
  d_4 & \text{if, } (Y_j \cap X_i \neq \emptyset) \land (Y_j \not\subseteq X_i) \\
 & \quad \land (Y_j \neq \emptyset) \\
  d_4 & \text{if, } (X_i = \emptyset) \land (Y_j \neq \emptyset) 
\end{cases} \quad (7.5)
\]
Where

\[
\begin{align*}
    d_1(Z_k) &= \begin{cases} 
    \frac{1}{2} & \text{if } (Z_k = \{T\}) \lor (Z_k = \{\bot\}) \\
    0 & \text{otherwise}
    \end{cases} \\
    d_2(Z_k) &= \begin{cases} 
    1 & \text{if } Z_k = \{T\} \\
    0 & \text{otherwise}
    \end{cases} \\
    d_3(Z_k) &= \begin{cases} 
    1 & \text{if } Z_k = \{\bot\} \\
    0 & \text{otherwise}
    \end{cases} \\
    d_4(Z_k) &= \begin{cases} 
    \left( \frac{|X \cap Y|}{|Y|} \right) & \text{if } Z_k = \{T\} \\
    \left( 1 - \frac{|X \cap Y|}{|Y|} \right) & \text{if } Z_k = \{\bot\} \\
    0 & \text{otherwise}
    \end{cases}
\end{align*}
\]

### 7.14 Conclusions

This chapter has looked at the versions of semantic unification presented by Baldwin et al. [10], with a view to using them as valuation functions. Section 7.7 showed that when mass is moved into truthspace, it is done in a non-intuitive way. In particular, inconsistency, belief and plausibility are not preserved correctly.

Section 7.11 presented a modified form of the interval semantic unification, which preserves belief and plausibility from the evidence to the truth-space result. This allows equivalent queries to be asked of the evidence and the claim, and for those queries to have the same results. In addition, the updated semantic unification function can handle claims that are inconsistent with the evidence as well as claims about the inconsistency itself.

Point-value semantic unification was then dealt with in the same way. FRIL's point-value semantic unification is identical to its interval semantic unification except in the way it handles uncertainty in
the result. A new consistent point-value function has been defined which mimics the updated interval versions handling of inconsistency (see Equation 7.5).

For any reasoning system based on mass assignment theory to be useful in practice a mechanism is needed to move facts and hypotheses from the 'real' world and into truth-space. The work presented in this chapter allows knowledge to be moved into truth-space without losing belief, plausibility or inconsistency. Without this preservation of information any truth-valuation of a mass assignment would not be consistent with the 'real' world, and therefore would make any inferences made from the valuation difficult to justify. This chapter solves that problem.
Chapter 8

Limits of Truth-Space Mass Assignment Operators

8.1 Introduction

As has been mentioned several times before in this Thesis there are an infinite number of results for a single mass assignment operator. These results are obtained by varying the contribution function (Section 6.3) used to calculate the mass assigned to each cell. Two main contribution functions have been identified the multiplicative contribution function and the minimum contribution function. The minimum contribution function can itself produce several different results depending on the order in which cells are processed.

Given that except in the most trivial of cases the number of possible results is infinite it is desirable to define the limits of valid results which an operator can make. Some of these limit results have already been defined. They are the orthogonal maximals and minimals for the restrictiveness (knowledge) and truth partial orderings.
8.2 The Limits

Truth-space mass assignments have four focal elements. Depending on the contribution function used the mass assigned to each set in the result varies. Obviously there will be a maximum value that each focal set can take. For example, the maximum amount of mass that could be placed in the set \{T\} by any result might be 0.3; therefore no valid result will have more than 0.3 mass assigned to \{T\}. Therefore, there are at least four limits for truth-space mass assignment operators.

1. The results when \(\emptyset\) is maximised
2. The results when \{\top\} is maximised
3. The results when \{\bot\} is maximised
4. The results when \{\top, \bot\} is maximised

Of course by maximising a focal set \(X\) the values that the other focal sets can take are restricted accordingly. If the maximum valued of \{\top, \bot\} is 0.4 then the sum of the other focal sets when \{\top, \bot\} is maximised is 0.6. This means that given that one focal set is at its maximum all other focal sets have another potential maximum.

Therefore, the calculation to find the limits of truth-space mass assignments consists of maximising the focal sets according to some particular order. For example, one limit will be when \{\top\} is maximised, then given that fact \{\top, \bot\} is maximised, then \{\bot\} and finally \(\emptyset\). There are four focal sets and therefore \(4! = 24\) different permutations to maximise the focal sets in. These are shown in Table 8.1.

Each entry in Table 8.1 describes a different permutation to maximise the focal sets in and therefore a limit to the operator. The example ordering which maximises \{\top\} then \{\top, \bot\} then \{\bot\} and
CHAPTER 8. LIMITS OF TRUTH-SPACE MA OPS.

Table 8.1: Permutations for Maximising Truth-Space MAs

<table>
<thead>
<tr>
<th>UTF0</th>
<th>TUF0</th>
<th>FUT0</th>
<th>0UTF</th>
</tr>
</thead>
<tbody>
<tr>
<td>UFT0</td>
<td>TFU0</td>
<td>FTU0</td>
<td>0TUF</td>
</tr>
<tr>
<td>UT0F</td>
<td>TU0F</td>
<td>FU0T</td>
<td>0UFT</td>
</tr>
<tr>
<td>UF0T</td>
<td>TF0U</td>
<td>FT0U</td>
<td>0TFU</td>
</tr>
<tr>
<td>U0TF</td>
<td>T0UF</td>
<td>F0UT</td>
<td>0FUT</td>
</tr>
<tr>
<td>U0FT</td>
<td>T0FU</td>
<td>F0TU</td>
<td>0FTU</td>
</tr>
</tbody>
</table>

finally 0 is written TUF0 and is highlighted in bold text in the table.

Note that there will probably not be twenty-four unique limits as several permutations will produce an identical result. Therefore, depending on the operator and the operands there may be several ways of calculating the same limit.

8.3 Limits and Maximals

The orthogonal maximals are the set of orthogonal results that are greater than or equal to any other result based on some ordering. For example the set \([X \ast Y]_T\) is the set of results that are greater than all other results in the truth ordering. Other than the maximals of the truth-ordering, the other maximals for truth-space mass assignments mentioned in previous chapters are those for the knowledge ordering \([X \ast Y]_K\) and those related to its inverse: the restrictiveness ordering \([X \ast Y]_R\).

The limits describe various maximums that the focal set masses can take. There is obviously a link between the orthogonal maximals for each ordering and the limits. The question is which orthogonals are related to which limit.
8.4 Limits and the Truth Maximals

The truth-ordering is:

\[ A \leq_T B \equiv \text{type-1}_T(B, A) \lor \text{type-2}_T(B, A) \]

Where \( \text{type-1}_T(B, A) \) means that mass assignment \( B \) is a type-1 restriction of mass assignment \( A \) (Equation 5.2) and \( \text{type-2}_T(B, A) \) means that \( B \) is a type-2 restriction of \( A \) (Equation 5.3) This means that \( A \geq_T B \) if:

\[ A \geq_T B \equiv \text{type-1}_T(A, B) \lor \text{type-2}_T(A, B) \]

The Limits and \( \{T\} \)

The mass assignment results with the most mass assigned to \( \{T\} \) are candidates for the orthogonal maximals. These are:

\[ \text{TFU0, TUF0, TU0F, TF0U, T0FU, T0UF} \]

Consider the result \( A \) of a mass assignment operator which has the maximum possible mass assigned to \( \{T\} \). For another mass assignment, \( B \), to be greater than \( A \) in the truth ordering it would need to be either a type-1 or type-2 restriction of \( A \). However, there are no type-1 or type-2 restrictions of \( A \) that are valid results. Any restriction would increase the mass assigned to \( \{T\} \) making the restriction invalid.

Note that there may be a valid type-1 restriction of \( A \) that has the same amount of mass assigned to \( \{T\} \) as \( A \). This type-1 would move mass from \( \{\perp\} \) to either \( \emptyset \) or \( \{T, \perp\} \). Therefore, although all the orthogonal truth maximals \( [X * Y]_T \) have the maximum mass assigned to \( \{T\} \) not all results with maximum mass assigned to \( \{T\} \) are orthogonal maximals.
The Limits and \( \{\bot\} \)

The next question is, therefore, which of the six orderings that max­imise truth produce the orthogonal truth maximals? If \( A \) is a result of \( X \ast Y \) that was produced from one of the six orderings then it has maximum mass assigned to \( \{\top\} \). In addition, from the previ­ous section it is known that there are no mass assignments greater than \( A \) due to a type–2\( _T \) restriction. However, there may be a mass assignment that has the same mass assigned to \( \{\top\} \) but is still a type–1\( _T \) restriction of \( A \).

This occurs when the restriction \( B \) moves mass from \( \{\bot\} \) to either \( \emptyset \) or \( \{\top, \bot\} \). Remember, it cannot move the mass from \( \{\bot\} \), \( \{\top, \bot\} \), or \( \emptyset \) to \( \{\top\} \) because it would make the result invalid. This means that the truth maximal results will have the least possible mass assigned to \( \{\bot\} \) given that \( \{\top\} \) has already been maximised. This means in effect that the orderings that produce the orthogonal truth maximals are shown in Table 8.2:

<table>
<thead>
<tr>
<th>Table 8.2: Limits that Create the Truth Maximals</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{TUØF} )</td>
</tr>
</tbody>
</table>

Depending on the operands and the operator in question these two orderings for the min contribution function may produce different results. However, these results will be orthogonal maximals because no type–1\( _T \) or type–2\( _T \) restriction will convert one to the other. In addition, all other orthogonal truth maximal results will be linear combinations of \( \text{TUØF} \) and \( \text{TØUF} \). This because all the orthogonal maximals must have the same \( \{\top\} \) and \( \{\bot\} \) masses otherwise there will be a type–1\( _T \) or type–2\( _T \) restriction between them making them non-maximal.
The Truth Limits and Truth Operators

The truth meet operators discussed in Section 5.5 return a convex combination of the truth maximals. Because the truth maximals can be created by using the min contribution function on the focal sets in the ordering $T0UF$ and $TU0F$ the truth meet can be calculated using the same method.

8.5 Limits and the Knowledge Maximals

The knowledge ordering is:

$$A \leq_k B \equiv \text{type-}1_R(B, A) \lor \text{type-}2_R(B, A)$$

Where type-1$_R$ $(B, A)$ means that mass assignment $B$ is a type-1$_R$ restriction of mass assignment $A$ (Equation 3.1) and type-2$_R$ $(B, A)$ means that $B$ is a type-2$_R$ restriction of $A$ (Equation 3.2)

The Limits and \(\emptyset\)

This is practically identical to the previous section of the truth maximals and limits. However, this time the restrictions used are type-1$_R$ and type-2$_R$ restrictions. This means that all the potential knowledge maximals have maximum mass assigned to $\emptyset$, they are:

$$\emptyset TFU \ \emptyset FTU \ \emptyset FUT \ \emptyset TUF \ \emptyset UTF \ \emptyset UFT$$

This is because both type-1$_R$ and type-2$_R$ restrictions move mass into $\emptyset$. Therefore, if $A$ has maximum mass assigned to $\emptyset$ then all restrictions of them would have more mass assigned to $\emptyset$ making them invalid.
CHAPTER 8. LIMITS OF TRUTH-SPACE MA OPS.

The Limits and \( \{T, \perp\} \)

Again in a similar way to the truth limits the results which minimise the mass assigned to \( \{T, \perp\} \) are the extremes of the knowledge maximals \([X \ast Y]_K\). These are shown in Table 8.3:

Table 8.3: Limits that Create the Knowledge Maximals

<table>
<thead>
<tr>
<th>( \emptyset \text{TFU} )</th>
<th>( \emptyset \text{FTU} )</th>
</tr>
</thead>
</table>

This is because the type-1\( R \) restriction can move mass from \( \{T, \perp\} \) to its subsets. This means that if two mass assignments have the same mass assigned to \( \emptyset \) and \( A \) has less mass assigned to \( \{T, \perp\} \) than \( B \) then \( A \) is a restriction of \( B \) and therefore greater in the knowledge ordering. The actual set of orthogonal maximals for the truth ordering is the set of all results that are linear combinations of \( \emptyset \text{TFU} \) and \( \emptyset \text{FTU} \).

The Knowledge Limits and Knowledge Operators

The knowledge meet operator returns a convex combination of the knowledge maximals. Because the truth maximals can be created by using the min contribution function on the focal sets in the ordering \( \emptyset \text{TFU} \) and \( \emptyset \text{FTU} \) the knowledge meet can be calculated using the same method.
8.6 Limits and the Restrictiveness Maximals

The restrictiveness ordering is the inverse of the knowledge ordering:

\[ A \leq_R B \equiv \text{type}^{-1}_R(A, B) \lor \text{type}^{-2}_R(A, B) \]

This is simply the inverse of the knowledge ordering. \( A \leq_R B \) if \( B \leq_K A \). Because of this the potential limits that generate the maximals of the restrictiveness ordering are the inverse of those that potentially generate the knowledge maximals. They are shown in Table 8.4:

Table 8.4: Limits that Potentially Create the Restrictiveness Maximals

<table>
<thead>
<tr>
<th>UFT0</th>
<th>UTF0</th>
<th>TUF0</th>
<th>FUT0</th>
<th>FTU0</th>
<th>TFU0</th>
</tr>
</thead>
</table>

Narrowing down the potential maximals from these limit results is not as easy as it was for the knowledge limits and maximals. It might be tempting to limit them to:

<table>
<thead>
<tr>
<th>UTF0</th>
<th>UFT0</th>
</tr>
</thead>
</table>

The reason for this is that these two orderings are the reverse of the orderings that produce the knowledge maximals. However, the knowledge limits maximise \( \emptyset \) and then minimise \( \{\top, \bot\} \), the inverse of this is minimising \( \emptyset \) and then maximising \( \{\top, \bot\} \). This is not the same as maximising \( \{\top, \bot\} \) and then minimising \( \emptyset \). Consider the following three tableaux.
Case 1

In Figure 8.1 both the UTF₀ and the UFT₀ limit orderings for the first tableau produces the result \( \emptyset : 0.5 \{ \top, \bot \} : 0.5 \). However, any other limit ordering listed in Table 8.4 produces \( \{ \top \} : 0.5 \{ \bot \} : 0.5 \).

\[
\text{type-}2_R([\emptyset : 0.5 \{ \top, \bot \} : 0.5], [\{ \top \} : 0.5 \{ \bot \} : 0.5])
\]

So therefore,

\[
[\emptyset : 0.5 \{ \top, \bot \} : 0.5] \leq_R [\{ \top \} : 0.5 \{ \bot \} : 0.5]
\]

Hence in this case neither the UTF₀ or UFT₀ limits produce a maximal result, the other limit orderings are maximals for the restrictiveness ordering.

Case 2

Taking Figure 8.2 the UTF₀ an UFT₀ limits produce

\[
\{ \top \} : 0.5 \{ \top, \bot \} : 0.5
\]
The other orderings produce:

\[
\begin{align*}
\text{UTF0} & = \emptyset : 0.0 \quad \{\top\} : 0.5 \quad \{\bot\} : 0.0 \quad \{\top, \bot\} : 0.5 \\
\text{UFT0} & = \emptyset : 0.0 \quad \{\top\} : 0.5 \quad \{\bot\} : 0.0 \quad \{\top, \bot\} : 0.5 \\
\text{TUF0} & = \emptyset : 0.0 \quad \{\top\} : 0.5 \quad \{\bot\} : 0.0 \quad \{\top, \bot\} : 0.5 \\
\text{TFU0} & = \emptyset : 0.0 \quad \{\top\} : 0.5 \quad \{\bot\} : 0.0 \quad \{\top, \bot\} : 0.5 \\
\text{FTU0} & = \emptyset : 0.5 \quad \{\top\} : 0.0 \quad \{\bot\} : 0.5 \quad \{\top, \bot\} : 0.0 \\
\text{FUT0} & = \emptyset : 0.5 \quad \{\top\} : 0.0 \quad \{\bot\} : 0.5 \quad \{\top, \bot\} : 0.0
\end{align*}
\]

\[
[\emptyset : 0.5 \quad \{\bot\} : 0.5] \leq_R [\{\top\} : 0.5 \quad \{\top, \bot\} : 0.5]
\]

Because there is a series of type-1\(_R\) restrictions between the two sets. This means that all but the last two limits produce a maximal result.
Case 3

Taking Figure 8.3 the limit orderings produce:

\[
\begin{align*}
\text{UTF}_0 & = \emptyset : 0.0 \ \{\top\} : 0.0 \ \{\bot\} : 0.0 \ \{\top, \bot\} : 0.5 \\
\text{UFT}_0 & = \emptyset : 0.0 \ \{\top\} : 0.0 \ \{\bot\} : 0.0 \ \{\top, \bot\} : 0.5 \\
\text{TUF}_0 & = \emptyset : 0.5 \ \{\top\} : 0.5 \ \{\bot\} : 0.0 \ \{\top, \bot\} : 0.0 \\
\text{TFU}_0 & = \emptyset : 0.5 \ \{\top\} : 0.5 \ \{\bot\} : 0.0 \ \{\top, \bot\} : 0.0 \\
\text{FTU}_0 & = \emptyset : 0.0 \ \{\top\} : 0.0 \ \{\bot\} : 0.5 \ \{\top, \bot\} : 0.5 \\
\text{FUT}_0 & = \emptyset : 0.0 \ \{\top\} : 0.0 \ \{\bot\} : 0.5 \ \{\top, \bot\} : 0.5
\end{align*}
\]

\[
[\emptyset : 0.5 \ \{\top\} : 0.5] \leq_R [\{\bot\} : 0.5 \ \{\top, \bot\} : 0.5]
\]

Because there is a series of type-1\textsubscript{R} restrictions between the two sets. This means that all but the middle two limits produce a maximal result.

The Restrictiveness Limits and Restrictiveness Operators

This means that the exact limit results that produce the restrictiveness maximals depends on the operator and the operands being
used. All the maximals have one thing in common, they have the least mass assigned to \( \emptyset \).

Whichever set of limits produce the maximals for the restrictiveness orderings the same set of orderings produces the minimals for the knowledge ordering. This is because the two orderings are the inverse of each other.

### 8.7 Limits and the Falsity Maximals

In the same way that the restrictiveness ordering is the inverse of the knowledge ordering it is possible to define an inverse of the truth ordering. This will be called the falsity ordering and denoted \( \leq_F \). It is defined as:

\[
A \leq_F B \equiv \text{type-}1_T(A, B) \lor \text{type-}2_T(A, B) \quad (8.1)
\]

The reasoning that relates the restrictiveness limits and maximals to the knowledge limits and maximals also applies to the truth and falsity limits. The truth limits maximise \( \{ \top \} \) and then minimise \( \{ \bot \} \) so the falsity limits minimise \( \{ \top \} \) and then maximise \( \{ \bot \} \). These limits are shown in Table 8.5.

Table 8.5: Limits that Potentially Create the Falsity Maximals

<table>
<thead>
<tr>
<th>FØUT</th>
<th>FUØT</th>
<th>UØFT</th>
<th>UFØT</th>
<th>ØUFT</th>
<th>ØFUT</th>
</tr>
</thead>
</table>

The list cannot be narrowed further because of similar reasons to the restrictiveness limits: the limits that give maximals are dependent on the operator and operands used.
The Falsity Limits and Truth Operators

Again the maximals of the falsity ordering are the minimals of the truth ordering and vice versa. This means that the truth join, which requires the truth minimal results, can be calculated using the min contribution function and the falsity limits.

\[[X \star Y]_F = [X \star Y]_T\]

8.8 The Envelope of Valid Results

Truth-space mass assignments have four focal sets; this means that they can be plotted as a point in \(\mathbb{R}^4\) with each axis corresponding to one of the four focal sets. The unique results generated from the twenty-four limit permutations are in effect a convex hull containing all valid results.

Four-dimensional space can be difficult to visualise. However, the masses in a MA must sum to one, this means that any truth-space mass assignment can be plotted as a vector in \(\mathbb{R}^3\). Each axis corresponds to one of the four focal sets, the focal set mass that is unassociated with an axis can be calculated from the other three. For example, the axis could be \(m(\{\top\})\), \(m(\{\bot\})\), and \(m(\{\top, \bot\})\) with \(m(\emptyset) = 1 - [m(\{\top\}) + m(\{\bot\}) + m(\{\top, \bot\})]\).

The twenty-four limit results for any operator the results in a truth mass assignment are shown in Table 8.1. If these limits are plotted in \(\mathbb{R}^3\) then they produce a volume that describes an envelope in which all valid results for that operation lie. For example, the envelope for a hypothetical operation might look like the one in Figure 8.4.

The corners of the polyhedron are the mass assignment results generated by the limit orderings. The edges are results that are the linear combination of any two of the limit results. The other points...
such as the surfaces and the points on the inside of the volume are all convex combinations of several of the limit results.

### 8.9 Limits of Point-Value Operators

Varying the contribution function can produce a range of different results for the interval version of a mass assignment operator. The limits of the envelope of results can be found by maximising the focal set in a particular order as presented in the previous sections.

Each mass assignment in this envelope can be converted into a point-value by applying a consistency and certainty function to the distribution algorithm (Sections 6.6 and 6.7). Consider the following interval mass assignment:

\[
\emptyset : 0.25 \{T\} : 0.25 \{\bot\} : 0.25 \{T, \bot\} : 0.25
\]
CHAPTER 8. LIMITS OF TRUTH-SPACE MA OPS.

Applying a consistency function to this assignment will move mass from $\emptyset$ to one or many of the other focal sets. However, there are limits to how much mass can be moved, i.e., no more than 0.25 mass can be moved. These limits are found by moving all the mass assigned to $\emptyset$ to either $\{T\}$, $\{\bot\}$ or $\{T, \bot\}$. This means that the consistent mass assignments that can be created from the example mass assignment are limited by the MAs:

- Most True = $\emptyset : 0.0$  $\{T\} : 0.50$  $\{\bot\} : 0.25$  $\{T, \bot\} : 0.25$
- Most False = $\emptyset : 0.0$  $\{T\} : 0.25$  $\{\bot\} : 0.50$  $\{T, \bot\} : 0.25$
- Most Uncertain = $\emptyset : 0.0$  $\{T\} : 0.25$  $\{\bot\} : 0.25$  $\{T, \bot\} : 0.50$

This consistency envelope can be seen in Figure 8.5. The black point indicates the original mass assignment, the black triangle indicates the envelope of consistent mass assignments that can be made from that point.

Using the same original point it is possible to apply a certainty function that move mass from $\{T, \bot\}$ to the other three focal sets. Again there is a limit to how much mass can be moved. This produces the limit MAs:

- Most Inconsistent = $\emptyset : 0.50$  $\{T\} : 0.25$  $\{\bot\} : 0.25$  $\{T, \bot\} : 0.00$
- Most True = $\emptyset : 0.25$  $\{T\} : 0.50$  $\{\bot\} : 0.25$  $\{T, \bot\} : 0.00$
- Most Uncertain = $\emptyset : 0.25$  $\{T\} : 0.25$  $\{\bot\} : 0.50$  $\{T, \bot\} : 0.00$

These points and the resulting envelope can be seen in Figure 8.6. Again the black point indicates the original MA and the black triangle shows the envelope of certain results that can be created from it.

Applying both a certainty and consistency function to the original mass assignment creates a envelope of consistent point-value
CHAPTER 8. LIMITS OF TRUTH-SPACE MA OPS.

Figure 8.5: A Consistency Envelope

Figure 8.6: A Certainty Envelope
mass assignments. This envelope can be seen in Figure 8.7. It is depicted as the solid black line that lies along the grey line \( m(\{T\}) + m(\{\bot\}) = 1 \).

![Figure 8.7: A Consistency and Certainty Envelope](image)

### 8.10 Example of Limits for Semantic Unification

Consider the semantic unification of two example mass assignments \( A \) and \( B \). The actual mass assignments used in this example come from FRIL — *Fuzzy and Evidential Reasoning in Artificial Intelli-
There is no mass assigned to the empty set in either $A$ or $B$ so there will be no mass assigned to the empty set in any result (see Equation 7.4). This means that there will only be cells assigning to \{T\}, {⊥} or \{T, ⊥\} in the tableau. Therefore, there will be at most six (3! = 6) unique limit results for this particular tableau, these are:

<table>
<thead>
<tr>
<th>UTF</th>
<th>TUF</th>
<th>FUT</th>
<th>UFT</th>
<th>TFU</th>
<th>FTU</th>
</tr>
</thead>
</table>

Of these limits UTF and UFT are the orthogonal maximals for the restrictiveness ordering $[A * B]_R$. Note that UTF$\emptyset$ and UFT$\emptyset$ are both listed as potential restrictiveness maximals in Table 8.4, they are:

\[
(UTF) = \emptyset : 0.0 \quad \{T\} : 0.2 \quad \{⊥\} : 0.1 \quad \{T, ⊥\} 0.7
\]

\[
(UFT) = \emptyset : 0.0 \quad \{T\} : 0.1 \quad \{⊥\} : 0.2 \quad \{T, ⊥\} 0.7
\]

This is the same result given in FRIL — *Fuzzy and Evidential Reasoning in Artificial Intelligence*. However, the book stops there; the work presented in this chapter allows further calculations to find the other limit results. In full, including the UTF and UFT limits, these limits are as shown in Table 8.6:

The last two limits, TFU and FTU, are identical so of the potential six limits there are only five unique limits. Including the orthogonal restrictiveness maximals which have already been men-
Table 8.6: Limits for Example $A \mid B$

<table>
<thead>
<tr>
<th>Order</th>
<th>$\emptyset$</th>
<th>${T}$</th>
<th>${\bot}$</th>
<th>${T, \bot}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>UTF</td>
<td>0.0</td>
<td>0.2</td>
<td>0.1</td>
<td>0.7</td>
</tr>
<tr>
<td>UFT</td>
<td>0.0</td>
<td>0.1</td>
<td>0.2</td>
<td>0.7</td>
</tr>
<tr>
<td>TUF</td>
<td>0.0</td>
<td>0.5</td>
<td>0.1</td>
<td>0.4</td>
</tr>
<tr>
<td>FUT</td>
<td>0.0</td>
<td>0.1</td>
<td>0.3</td>
<td>0.6</td>
</tr>
<tr>
<td>TFU</td>
<td>0.0</td>
<td>0.5</td>
<td>0.3</td>
<td>0.2</td>
</tr>
<tr>
<td>FTU</td>
<td>0.0</td>
<td>0.5</td>
<td>0.3</td>
<td>0.2</td>
</tr>
</tbody>
</table>

mentioned the maximals and their associated minimals are:

$$[A \ast B]_T = \{TU\emptyset F, T\emptyset UF\} = \{TUF\}$$

$$[A \ast B]_F = \{FU\emptyset T, F\emptyset UT\} = \{FUT\}$$

$$[A \ast B]_R = \{UTF\emptyset, UFT\emptyset\} = \{UTF, UFT\}$$

$$[A \ast B]_K = \{\emptyset TFU, \emptyset FTU\} = \{TFU, FTU\}$$

Note that because the mass assigned to $\emptyset$ is always zero the maximals for truth and falsity consist of only one limit each.

**Plotting the Limits**

The six limit results for the example unification can be plotted on the plane $m(\{\top\}) + m(\{\bot\}) + m(\{\top, \bot\}) = 1$ and is depicted by the grey polygon in Figure 8.8. For comparison the multiplicative result: $\{\top\}:0.33 \{\bot\}:0.24 \{\top, \bot\}:0.43$ is plotted as the black point.
As can clearly be seen from the figure the limits TFU and FTU are the most restrictive (most knowledge) because the single result that represents them is the closest point to the line $m(\{T\}) + m(\{\perp\}) = 1$. Conversely the limits UTF and UFT are furthest from the line $m(\{T\}) + m(\{\perp\}) = 1$ and represent the least restrictive results.

The Limits of the Point-value Results for the Example

Taking the envelope of the interval results it is possible to find the consistent and certain results for each point in that envelope. The extremes of these point-value results are when all the mass assigned to $\emptyset$ and $\{T, \perp\}$ is assigned to either $\{T\}$ or $\{\perp\}$. This makes the
most true point value:

$$\emptyset : 0.0 \{\top\} : 0.9 \{\bot\} : 0.1 \{\top, \bot\} : 0.0$$

This can be constructed by taking the UTF limit and moving all the mass assigned to \{\top, \bot\} to \{\top\}. The most false point-value result is:

$$\emptyset : 0.0 \{\top\} : 0.1 \{\bot\} : 0.9 \{\top, \bot\} : 0.0$$

This is constructed by taking the UFT limit and moving all the mass from \{\top, \bot\} to \{\bot\}.

The two extreme point-value results and all the point-value results in between are depicted in Figure 8.9 by the thick black line. The classic point-value is also shown on the graph by a black point at:

$$\emptyset : 0.0 \{\top\} : 0.54583 \{\bot\} : 0.45416 \{\top, \bot\} : 0.0$$

Figure 8.9: Point-Value Limits for the Example SU (A | B)
8.11 Racing Pundit Example

Consider two racing pundits (X and Y). Based on previous form they are both accurate one third of the time. This can be modelled as mass assignments representing $m_X = 'agent X is correct' $ and $m_Y = 'agent Y is correct' $.

\[ m_X = \{ T \} : 1/3 \ \{ \bot \} : 2/3 \]

\[ m_Y = \{ T \} : 1/3 \ \{ \bot \} : 2/3 \]

To find the probability that both the pundits will agree the conjunction of the two mass assignments is used $(m_X \land m_Y)$. There are many ways of interpreting this operator though. If independence is assumed between the two then:

\[ m_X \land m_Y = \{ T \} : 1/9 \ \{ \bot \} : 8/9 \]

This means that based on known information the probability that both pundits will be correct on the same call is 1/9. This is what would be expected from normal probability theory. If $Pr_X$ is the probability that $X$ is correct (equivalent to $m_X$) and similarly $Pr_Y$ is the probability that $Y$ is correct then:

\[
Pr_{(X \text{ and } Y)} = Pr_X Pr_Y \\
= 1/3 \times 1/3 \\
= 1/9
\]

However, if the restrictiveness meet (set union) would produce:

\[ m_X \setminus m_Y = \emptyset : 4/9 \ \{ T \} : 1/9 \ \{ \bot \} : 4/9 \]

Here inconsistency has been introduced into the mass assignment.
Whilst the probability that they will both be correct (assuming independence) has not changed the probability that they would not both be correct is lower (4/9 instead of 8/9). The probability that either $X$ or $Y$ will be wrong should be:

$$Pr_{\neg (X \text{ and } Y)} = 1 - (Pr_X Pr_Y)$$

$$= 1 - (1/3 \cdot 1/3)$$

$$= 8/9$$

Using mass assignment union would not be correct in this case and the conjunction should be used. The restrictiveness meet is actually calculating the probability that both pundits will be incorrect.

Assumptions other than independence can be made. Both pundits will be basing their decision on similar or identical information and their decision making system is likely to be similar as well. An assumption that their decision making process is completely identical then the result would be found using the TUF ordering:

$$m_\land_2 m = \{\top\} : 1/3 \ \{\bot\} : 2/3$$

Note: that this is not stating the probability that they will agree (that is assumed to be one), it is the probability that they will both be correct on the same call. To compare, if the mass assignment union is used then the result is:

$$m_\lor_2 m = \{\top\} : 1/3 \ \{\bot\} : 2/3$$

There appears to be no difference between the two operator in this case. The pundits are both making the same call and they are both going to be correct 1/3 of the time. This is what would be expected given the data and the assumptions.
However, if some uncertainty is introduced as to how accurate the pundits are (say that their accuracy is unknown 1/3 of the time) then the mass assignments look like:

\[
\begin{align*}
\mathbf{m}_X &= \{T\} : 1/3 \ \{T, \bot\} : 1/3 \ \{\bot\} : 1/3 \\
\mathbf{m}_Y &= \{T\} : 1/3 \ \{T, \bot\} : 1/3 \ \{\bot\} : 1/3 \\
\mathbf{m}_X \wedge \mathbf{m}_Y &= \{T\} : 1/3 \ \{T, \bot\} : 1/3 \ \{\bot\} : 2/3 \\
\mathbf{m}_X \cap \mathbf{m}_Y &= \{T\} : 2/3 \ \{T, \bot\} : 0 \ \{\bot\} : 1/3
\end{align*}
\]

Here the optimistic assumption (TUF) produces different results. The conjunction is, in effect, idempotent. Because both \(X\) and \(Y\) are assumed to always make the same call then they should be considered as a single source and it is expected that \(X \wedge X = X\). The probability that they will both be correct is unchanged and no assumption is made as to whether the unknown calls would have been accurate or not. However, the restrictiveness meet has altered. It assumes that the unknown cases would have resulted in the pundits being correct.

This clearly demonstrates that not only the type of operator, eg truth or restrictiveness, but also the assumption used matters greatly in the calculation of the result.

### 8.12 Discussion on Limit Results

This chapter concentrated on finding the extremes to which a truth-space mass assignment operator can assign mass to either \(\{T\}, \{\bot\}, \emptyset\) or \(\{T, \bot\}\). These extremes provide the limits of an envelope of all valid results for that operator. Every valid result will be a convex combination of the limits results.

These limits are created by maximising the mass assigned to a single focal set and then maximising the mass assigned to the other
sets in turn. Because there are four focal sets in a truth space mass assignment there are 24 permutations of these points and therefore 24 potential limits. However, only some of these limits will actually produce unique results.

This chapter also showed which limits are related to which of the maximals for the truth, knowledge, restrictiveness and falsity partial orderings. In the case of the truth and knowledge orders the limits can be restricted to only two limits. For the other two orderings (which are the inverse of the truth and knowledge orders) the set of limits related to the maximals can only be restricted to six limits for each order.

The envelope described by these limits can be plotted against the mass assigned to each focal set. This volume helps visualise the range of different results an operator can give. An example semantic unification was given and plotted. Although the example did not contain an assignment to \( \emptyset \), and therefore produced a plane and not a volume, it could easily be extended to an example which produces a volume.

Chapter 6 describes how interval results (i.e., those lying inside the envelope of valid results) can be transformed into consistent and certain results such as those returned by the point-value operators. These transformations have their own extremes to which they can assign mass. For example, even the most \( \{\top\} \) most consistent results may need to assign a minimum amount of mass to \( \{\perp\} \). The limits of the consistency and certainty functions can be plotted in a similar way to the envelope of interval results.

All the operators that result in a truth-space mass assignment can have their envelopes of valid results plotted as well the extremes of their point-value results. This provides a new visualisation method for mass assignment operators. In addition a MA operator may be described as 'narrow' if the variation in valid results is small (a small
envelope) or ‘wide’ if there is a large amount of variation in the valid results (a large envelope). The size of the envelope (its length if it is a line, its area if it is a plane or its volume if it is a volume) describes how much the result of the operator can be varied using assumptions about how the operands interact.

Of all the results in the envelope the limits are the most important. As well as defining the extents of the envelope they also are potential candidates for the maximals andimals of the truth and restrictiveness orderings. These results are key to calculating the meets and joins of the lattices related to those orderings.

Currently there is no algorithm proven to take two mass assignments, an operator and a limit ordering such as UTF0 and calculate the limits associated with that ordering. Appendix A at the end of this Thesis discusses some of the constraints that any such algorithm would need to obey if it is to ensure the focal sets are maximised. The design of an algorithm that maximises focal sets in order is a topic for future research.
Chapter 9

Conclusion

The main points presented in this work have been discussed at the end of the relevant chapter. This chapter provides a summary of the main points of the work and how they relate to each other and to current and future work by other researchers.

9.1 Contribution of Thesis

The main contributions of this Thesis to current and future works can be divided into five main parts:

1. The Definition of Truth-Space. The chapter on truth-space (Chapter 4) draws heavily on work on bilattices by Ginsberg and Fitting and also on Belnap’s $\mathcal{FUR}$. However, despite superficial similarities, there are distinct differences in the semantics of $\mathcal{FUR}$ and truth-space which have particular implications for mass assignments defined upon it. The major differences are the implicit ‘or’ connective between elements of the sets and the semantics of the set elements; $\mathcal{FUR}$ uses $t$ to indicate the presence of evidence of truth, truth-space uses $T$ to indicate ‘true and only true’.
2. **Truth-Space Mass Assignments.** T-MAs are fundamentally a type of mass assignment but they also have several special features. Chapter 5 uses the semantics of truth-space and the concept of bilattices to create a partial bilattice that provides a new partial ordering for t-MAs. Combining this new ordering with the existing restrictiveness ordering allows t-MAs to be compared using the amount of truth they contain. In addition, new truth operators allow t-MAs to be combined in ways closer to T\textsc{our}'s conjunction and disjunction. This provides a link between mass assignment theory and other logic systems like T\textsc{our} and Kleene's three-valued logic.

3. **Examining the Internals of Mass Assignment Operators.** Chapter 6 looks at how interval and point-value operators are related. Prior to the work in this chapter there were two different types of semantic unification operator, one that returned a mass assignment and one which returned a point-value probability. In addition, there are an infinite number of ways of solving a mass assignment tableau. There were also functions that turned mass assignments into probability distributions and normalised mass assignments that had mass in the empty set.

The chapter showed how all these elements are related. Operators that return mass assignments (or intervals) consist of two parts a contribution and a distribution algorithm. The distribution algorithm defines what exactly the operator is. All versions of an operator will have exactly the same distribution algorithm. Point-value operators, normalising functions and functions that produce distributions like the least-prejudiced distribution use certainty and consistency functions to alter the mass assignment operator to produce different results.
4. Updating Semantic Unification. Once truth-space mass assignments had been defined and discussed the Thesis then looked at the mechanism used to compare the similarity between two mass assignments in an arbitrary domain. The existing semantic unification operator returned a probability interval (or a point-probability) and in effect was a valuation function that calculated the truth of a mass assignment given some evidence represented as a second mass assignment. However, detailed analysis of the operator showed that it did not preserve inconsistency, belief or possibility through the operator. This lack of intuitive operation occurred when either the claim or the evidence were inconsistent (i.e., had mass assigned to the empty set).

The work in Chapter 7 alters the semantic unification operator to make it preserve information in this particular circumstance. The key to this was allowing the semantic unification operator to assign mass to the empty set in the result. This turned the result from an interval \([m(\{T\}), 1 - m(\{\bot\})]\) to a full truth-space mass assignment. The updated operator is the link between mass assignments defined on an arbitrary domain and those defined on truth-space.

5. Defining the Limits of a MA Operator. With so many different versions of a mass assignment operator it is difficult to highlight particular ones which are of use. Existing literature highlighted the maximal and minimal results of the restrictiveness operators used as part of the solution of the meet and join operators for the restrictiveness ordering. Chapter 5 extended this set of special results to include the maximal and minimal results of the truth-ordering used to create the meet and join of that partial ordering.
These sets of maximal and minimal results have one thing in common: they are all limits to an envelope of valid results for that operation. No valid result can be outside this envelope. These limits can be calculated by ordering the focal sets and then maximising the mass assigned to each focal set in order. Different permutations potentially produce different limits.

The chapter showed how this envelope can be plotted as a volume in three dimensional space. In addition, the limits to which contribution and certainty functions can alter the result can also be plotted. This provides the extents of a range of valid point-value interpretations for that operation. The size of the envelope and point-value range indicates how great a range of assumptions can be made as to how the operands interact.

9.2 Applicability of the Thesis

The work presented in the Thesis is applicable in many different areas. Most applications that use FRIL or mass assignment theory use the semantic unification operator. The changes made to that operator in this Thesis therefore will impact on any current or future application that wishes to unify mass assignments with mass associated with the empty set.

The truth mass assignments themselves offer a great amount of flexibility when combining and reasoning with information. Placing restrictions on them allows them to represent Boolean logic, probabilities, intervals, three-values logic, and four-valued logic. Mass assignments on other domains can represent fuzzy sets, probability families and probability distributions both partial and complete.
CHAPTER 9. CONCLUSION

Applications

Once implemented into a reasoning system, truth-space mass assignments can be used for a wide variety of practical applications. The changes to semantic unification are applicable to all existing applications that use it. Some such applications as listed in FRIL — Fuzzy and Evidential Reasoning in Artificial Intelligence [10] are: helicopter crew behaviour modelling, developing intelligent manuals and an orthodontic expert system.

Any domain which features possible inconsistencies from data sources or from the reasoning process rules could also benefit from an application that implements truth-space mass assignments. One such application is an expert system for analysing statements given to police. In these situations it is often necessary to keep a record of inconsistencies and where information may be vague or unreliable.

9.3 Future Work

This Thesis has opened up a great number of possible research topics and work for future consideration. Some of these new topics involve applying the work in this Thesis to practical domains and others involve expanding on the work in this Thesis.

There are many possible applications for truth-mass assignments. They provide the basis of a logic system that can handle truth, falsity, inconsistency and certainty in a much more flexible way to Belnap's FOUR. This is because t-MAs can represent degrees of belief in the truth and knowledge about a statement.

In addition, the changes to semantic unification can be applied to applications that use mass assignment theory. The changes made mean that such applications can be extended to handle inconsistency in an intuitive way.
The work presented in this Thesis can be expanded upon and developed in many ways. The most obvious area of development is to see how altering the contribution function used by an operator alters the results of a reasoning process. The use of the multiplicative contribution function to assume independence between operands and the use of the maximal and minimal results to produce meet and join operators is known. However, there are several other limits and an infinite number of valid results in between, the assumptions they represent and the consequences of choosing one of them over another certainly warrants investigation.

Also missing is the creation of an algorithm that can calculate the result of a mass assignment operator limit. Such an algorithm would take an operator, its operands and a limit ordering such as UTF0 and guarantees that its result has the maximum mass assigned to \{\top, \bot\}, then \{\top\} and so on as implied by the ordering. Appendix A is a short piece of work that looks at the constraints that any such algorithm must obey to ensure that a focal set reaches its potential maximum. However, it only considers one focal set at a time, extending the work to consider subsequent focal sets and how they can be maximised is a topic for future investigation.

Finally, it would be interesting to study whether the size of the envelope produced by limit results can provide any information into the operands, the information they contain and their relationship.

9.4 Final Comments

This piece of work has shown that truth-space mass assignments appear to be an extremely useful subset of mass assignments. It has defined the basic semantics of the mass assignments, their operators and the limits of their flexability. They provide a link between mass assignments and other popular many-valued logic systems and could
be applied to many and varied fields.
Appendix A

Calculating the Limits of Truth-Space Mass Assignments

A.1 Limit Mass Assignments

Chapter 8 discussed the extremes to which a mass assignment operator can assign mass to each of the focal sets. In addition, it stated that these limits can be found by maximising the mass assigned to a focal set and then maximising the mass placed to a second focal set and so on until all sets have been maximised according to some ordering.

The question that arises is how to maximise the mass assigned to a focal set? This appendix looks at the constraints placed on any algorithm that attempts to maximise a focal set. If all these constraints are fulfilled the focal set will be contain the maximum possible mass for that particular operation. First a single cell in an operator tableau is considered.
A.2 Maximising a Single Cell

Consider two mass assignment operands $A = \{A_i : m_A(A_i)\}$ and $B = \{B_j : m_B(B_j)\}$. According to mass assignment theory the mass assigned to a single cell in a mass assignment table cannot exceed its row mass, nor can it exceed its column mass. Therefore the maximum mass that can be assigned to a focal element cell ($C_{ij} = A_i \times B_j$) is the minimum of the cell’s row and column masses.

$$\text{MAXIMUM } m(C_{ij}) = \min \left( m(A_i), m(B_j) \right)$$ (A.1)

Note that this is just the maximum mass for a single cell in a tableau. The actual mass assigned to a focal set $C_k$ is the sum of all cells that assign to that focal set:

$$m(C_k) = \sum_i \sum_j m(C_{ij}) \quad \text{where } C_k = C_{ij}$$

A.3 Maximising the Current Focal Set

This section presents a comparison operator that maximises the mass assigned to the current focal set (CFS). It ignores the effect the ordering will have on subsequent focal elements.

If there is only a single cell that assigns mass to the current focal set then to maximise the focal set we assign maximum mass to those cells (see section A.2).

Definitions

Cells $C_{ij}$ and $C_{xy}$ are co-dependent (↔) if they share either a row or a column.

$$C_{ij} \leftrightarrow C_{xy} \text{ if } (i = x \lor j = y)$$ (A.2)
APPENDIX A. CALCULATING THE LIMITS

Cells in the CFS which are not co-dependent are said to be independent (∥).

\[ C_{ij} \parallel C_{xy} \text{ if } \neg(C_{ij} \leftrightarrow C_{xy}) \quad (A.3) \]

Groupings

A group is a set of cells that can be processed in any order without affecting the amount of mass assigned to the group as a whole.

• A single cell is a group.

• Group \( \alpha \) is **totally dependent** (→) on group \( \beta \) if every cell in group \( \alpha \) is co-dependent with at least one cell in group \( \beta \). (Figure A.1)

\[ \alpha \rightarrow \beta \text{ if } \left( \forall C_{ij} \in \alpha \ \exists C_{xy} \in \beta \mid i = x \lor j = y \right) \quad (A.4) \]

• Groups \( \alpha \) and \( \beta \) are **co-dependent** (↔) if \( \alpha \) is totally dependent on \( \beta \) and \( \beta \) is totally dependent on \( \alpha \). (Figure A.2)

\[ \alpha \leftrightarrow \beta \text{ if } \alpha \rightarrow \beta \land \beta \rightarrow \alpha \quad (A.5) \]

• Groups \( \alpha \) and \( \beta \) are **independent** (∥) if no cell in group \( \alpha \) is co-dependent with any cell in group \( \beta \) and vice-versa. (Figure A.3)

\[ \alpha \parallel \beta \text{ if } \left( \forall C_{ij} \in \alpha \ \forall C_{xy} \in \beta \mid \neg(C_{ij} \leftrightarrow C_{xy}) \right) \quad (A.6) \]

Although, figures A.1, A.2, and A.3 show the cells adjacent to each other, this is not necessary. It is the shared row and columns that is important not proximity.
Co-Dependence

**Law 4** Co-dependent groups can be coalesced without affecting the overall mass.

\[
\max(\alpha \beta) \equiv \max(\alpha) + \max(\beta | \alpha) \equiv \max(\beta) + \max(\alpha | \beta)
\]

If \( \alpha \leftrightarrow \beta \)

**Proof 4.1** That \( \max(\alpha \beta) \equiv \max(\alpha) + \max(\beta | \alpha) \) if \( \alpha \leftrightarrow \beta \)

\[
\max(\alpha \beta) = \min(x + w, \ y) \\
\max(\alpha) = \min(x, \ y) \\
\max(\beta | \alpha) = \min\left(w, y - \min(x, \ y)\right)
\]

If \( \min(x, \ y) = x \)

\[
\max(\alpha) + \max(\beta | \alpha) = x + \min(w, \ y - x) \\
= \min(x + w, \ y) \quad (A.7) \\
= \max(\alpha \beta)
\]

Else if \( \min(x, \ y) = y \)

\[
\max(\alpha) + \max(\beta | \alpha) = y + \min(w, \ y - y) \\
= y + \min(w, \ 0) \quad (A.8) \\
= y
\]
And,

\[ \max(\alpha; \beta) = \min(x + w, \ y) \]
\[ = y \] \hspace{1cm} (A.9)

**Proof 4.2** That \( \max(\alpha; \beta) = \max(\beta) + \max(\alpha \mid \beta) \) if \( \alpha \leftrightarrow \beta \)

\[ \max(\beta) = \min(w, \ y) \]
\[ \max(\alpha \mid \beta) = \min\left(x, y - \min(w, \ y)\right) \]

Similar to Proof 4.1 with masses \( w \) and \( x \) exchanged.

**Independence**

**Law 5** Independent groups can be coalesced without affecting the overall mass.

\[ \max(\alpha; \beta) \equiv \max(\alpha) + \max(\beta \mid \alpha) \equiv \max(\beta) + \max(\alpha \mid \beta) \]
\[ \text{If } \alpha \parallel \beta \]

**Proof 5.1** That Law 5 holds true.

Groups \( \alpha \) and \( \beta \) share neither rows nor columns (see Figure A.3). If \( \alpha \) is processed first, it restricts the masses \( x \) and \( y \), but not the masses \( w \) and \( z \). Similarly for \( \beta \) before \( \alpha \) does not restrict \( \alpha \). Therefore the cells in groups \( \alpha \) and \( \beta \) can be processed in any order.

**Partial and Indirect Dependence**

**Law 6** If \( \alpha \) is totally dependent on \( \beta \), but \( \beta \) is not totally dependent on \( \alpha \) then \( \alpha \) and \( \beta \) cannot be coalesced and \( \alpha \) must be processed first to maximise the total mass.
APPENDIX A. CALCULATING THE LIMITS

\[ \max(\alpha) + \max(\beta | \alpha) \geq \max(\beta) + \max(\alpha | \beta) \]

If \((\alpha \rightarrow \beta) \land \neg(\beta \rightarrow \alpha)\)

In this case Group \(\beta\) can be split into subgroups \(\gamma\) and \(\delta\) so that \(\alpha \leftrightarrow \gamma, \gamma \leftrightarrow \delta, \text{and } \alpha \parallel \delta\) (see Figure A.4). In this case \(\alpha\) is said to be indirectly dependent on \(\delta\).

\[ \begin{array}{ccc}
    y & & z \\
    x & \alpha & \\
    w & \gamma & \delta
\end{array} \]

Figure A.4: \(\alpha \leftrightarrow \gamma \land \gamma \leftrightarrow \delta\)

Once one of the three groups (\(\alpha, \gamma, \text{or } \delta\)) has been maximised then the other two groups can be processed in either order (see Laws 4 and 5). This means that there are three situations to consider.

1. \(\max(\alpha) + \max(\gamma \delta | \alpha)\)
2. \(\max(\gamma) + \max(\alpha \delta | \gamma)\)
3. \(\max(\delta) + \max(\alpha \gamma | \delta)\)

However, the second order (\(\gamma\) then \(\alpha \delta\)) reduces the overall mass assigned to the trio of cells.

**Proof 6.1** That \(\max(\alpha) + \max(\gamma \delta | \alpha) \geq \max(\gamma) + \max(\alpha \delta | \gamma)\).

\[
\begin{align*}
\max(\alpha) &= \min(x, \ y) \\
\max(\gamma \delta | \alpha) &= \min\left(w, y + z - \max(\alpha)\right) \\
\max(\gamma) &= \min(w, \ y) \\
\max(\alpha \delta | \gamma) &= \min\left(x, y - \min(\gamma)\right) + \min\left(w - \min(\gamma), z\right)
\end{align*}
\]
The possible answers to \( \max(\alpha) + \max(\gamma \delta | \alpha) \), given the various mass inequalities, are:

\[
\begin{align*}
(x + w) & \quad \text{If } (x \leq y) \land (w \leq y + z - x) \quad (A.10) \\
(y + z) & \quad \text{If } (x \leq y) \land (y + z - x \leq w) \quad (A.11) \\
(y + w) & \quad \text{If } (y \leq x) \land (w \leq z) \quad (A.12) \\
(y + z) & \quad \text{If } (y \leq x) \land (z \leq w) \quad (A.13)
\end{align*}
\]

And for \( \max(\gamma) + \max(\alpha \delta | \gamma) \) the possibilities are:

\[
\begin{align*}
(w + x) & \quad \text{If } (w \leq y) \land (x \leq y - w) \quad (A.14) \\
(y) & \quad \text{If } (w \leq y) \land (y - w \leq x) \quad (A.15) \\
(w) & \quad \text{If } (y \leq w) \land (w - y \leq z) \quad (A.16) \\
(y + z) & \quad \text{If } (y \leq w) \land (z \leq w - y) \quad (A.17)
\end{align*}
\]

Using Equation A.10:

\[
\begin{align*}
(x + w) &= (w + x) \\
(x + w) &\geq (y) \quad \text{Because, } (y - w \leq x) \equiv (y \leq x + w) \quad \text{from Equation A.15} \\
(x + w) &> (w) \\
(x + w) &\geq (y + z) \quad \text{Because, } (z \leq w - y) \equiv (z + y \leq w) \quad \text{from Equation A.17}
\end{align*}
\]
Using Equ A.11:

\((y + z) > (w + x)\)  Because, \((x \leq y - w) \iff (w + x \leq y)\)

from Equation A.14

\((y + z) > (y)\)

\((y + z) \geq (w)\)  Because, \((w - y \leq z) \iff (w \leq z + y)\)

from Equation A.16

\((y + z) = (y + z)\)

Using Equation A.12:

\((y + w) \geq (w + x)\)  Because, \((x \leq y - w) \iff (w + x \leq y)\)

from Equation A.14

And \((y \leq x) \land (w \geq 0)\)

from Equation A.12

\((y + w) > (y)\)

\((y + w) > (w)\)

\((y + w) > (y + z)\)  Because, \((z \leq w - y) \iff (z + y \leq w)\)

from Equation A.17

Equation A.13 is the same as Equation A.11.

Proof 6.2  That \(\max(\alpha) + \max(\gamma \delta | \alpha) = \max(\delta) + \max(\alpha \gamma | \delta)\).

\[
\begin{align*}
\max(\alpha) &= \min(x, y) \\
\max(\gamma \delta | \alpha) &= \min(w, y + z - \max(\alpha)) \\
\max(\delta) &= \min(w, z) \\
\max(\alpha \gamma | \delta) &= \min(y, x + w - \max(\delta))
\end{align*}
\]
The answers for $\max(\alpha) + \max(\gamma \delta \mid \alpha)$ are given in Proof 6.1. For $\max(\delta) + \max(\alpha \gamma \mid \delta)$ the possibilities are:

\begin{align*}
(w + y) & \quad \text{If } (w \leq z) \land (y \leq x) \quad (A.18) \\
(w + x) & \quad \text{If } (w \leq z) \land (x \leq y) \quad (A.19) \\
(z + y) & \quad \text{If } (z \leq w) \land (y \leq x + w - z) \quad (A.20) \\
(x + w) & \quad \text{If } (z \leq w) \land (x + w - z \leq y) \quad (A.21)
\end{align*}

Using Equation A.10:

\begin{align*}
(x + w) &= (w + y) \quad \text{Because, } (x = y) \text{ from Equ A.10 and A.18} \\
(x + w) &= (w + x) \\
(x + w) &= (y + z) \quad \text{Because, } (y \leq x + w - z) \equiv (y + z \leq x + w) \\
&\quad \text{from Equ A.20} \\
&\quad \text{And } (w \leq y + z - x) \equiv (w + x \leq y + z) \\
&\quad \text{from Equ A.10} \\
(x + w) &= (x + w)
\end{align*}
Using Equation A.11:

\[(y + z) = (w + y)\]  Because, \((w \leq z) \land (y \leq x)\) from Equ A.18

\[\therefore (w + y \leq x + z)\]

\[(y + z - x \leq w) \equiv (y + z \leq w + x)\]

from Equ A.11

And \((x = y)\) from Equs A.11 and A.18

\[(y + z) = (w + x)\]  Because, \((y + z - x \leq w) \equiv (y + z \leq w + x)\)

from Equ A.11

And \((w \leq z) \land (x \leq y)\) from Equ A.19

\[\therefore (w + x \leq +y)\]

\[(y + z) = (y + z)\]

\[(y + z) = (x + w)\]  Because, \((y + z \leq w + x)\) from Equ A.13

\[(x + w - z \leq y) \equiv (w + x \leq y + z)\]

from Equ A.21

Using Equation A.12:

\[(y + w) = (w + y)\]

\[(y + w) = (w + x)\]  Because, \((y = x)\) from Equs A.12 and A.19

\[(y + w) = (z + y)\]  Because, \((w = z)\) from Equs A.12 and A.20

\[(y + w) = (x + w)\]  Because, \((x + w - z \leq y) \equiv (x + w \leq y + z)\)

from Equ A.12

\[(w = z)\) from Equs A.12 and A.21

\[\therefore (x \leq y)\]

And \((y \leq x)\) from Equ A.12
Using Equation A.13:

\[(y + z) = (w + y)\quad \text{Because, } (w \leq z) \land (y \leq x) \text{ from Equ A.18}\]
\[\therefore (w + y \leq x + z)\]
\[(z \leq w) \land (y \leq x) \text{ from Equ A.13}\]
\[\therefore (z + y \leq x + w)\]
And \((y = x)\) from Equs A.13 and A.18

\[(y + z) = (w + x)\quad \text{Because, } (y \leq x) \land (z \leq w) \text{ from Equ A.13}\]
\[\therefore (y + z \leq w + x)\]
And \((w \leq z) \land (x \leq y)\) from Equ A.19
\[\therefore (w + x \leq z + y)\]

\[(y + z) = (y + z)\]
\[(y + z) = (x + w)\quad \text{Because, } (y + z \leq w + x) \text{ from Equ A.13}\]
And \((x + w - z \leq y) \equiv (w + x \leq y + z)\)

from Equ A.21

\section*{A.4 Considering the Other Focal Elements}

Only one of the three laws actually constrains the order in which the current focal element cells can be ordered without reducing the overall mass assigned to the focal element. Law 6 states that if a group \((\gamma)\) shares a column with a group \((\alpha)\) and a row with a group \((\delta)\) then \(\gamma\) must be done after either \(\alpha\) or \(\delta\). Cell maximisation orderings that follow this constraint allow the current focal element to be maximised. However, looking at Chapter 8 shows that the limit results that define the envelope of valid results for an operation are produced by maximising the focal sets in a particular order.

This means that there will be additional constraints on how a
focal set's cells should be ordered to not only ensure that it is maximised but also that the subsequent focal set in the ordering can be maximised. For example, in Section 8.10 there are two focal set orderings that maximise \( \{T, \perp\} \) yet still produce different results because one ordering maximises \( \{T\} \) next and the other maximises \( \{\perp\} \). Although the maximum mass assigned to \( \{T, \perp\} \) can be calculated by solving the constraints presented in this appendix there will be several different solutions. One solution may allow the subsequent focal set to attain a greater mass than another solution. From the example in Section 8.10:

\[
(UTF) = \emptyset : 0.0 \{T\} : 0.1 \{\perp\} : 0.7 \\
(UFT) = \emptyset : 0.0 \{T\} : 0.2 \{\perp\} : 0.7
\]

The maximum mass for \( \{T, \perp\} \) is 0.7 and can be found by solving the constraints presented in this appendix. However, different solutions to those constraints allow \( \{T\} \) and \( \{\perp\} \) to attain different 'maximums'. There are obviously more constraints than just the one implied by Law 6 acting on how the cells of \( \{T, \perp\} \) are processed. Study of these additional constraints provide an interesting topic of further research.
Appendix B

Publications

This is a list of all publications and pending publications written or co-written by the author during the research for this Thesis. Some, such as those on semantic unification, formed the basis of work that has been presented in the Thesis. Others are on topics related to mass assignment theory and its refinement.


APPENDIX B. PUBLICATIONS


Bibliography


