A SPECTRUM OF LOGICS – RANGING FROM BINARY TO FUZZY SYSTEMS

by

GERTRUIDA PETRONELLA VAN WYK

submitted in partial fulfillment of the requirements for the degree of

MASTER IN SCIENCE

in the subject of

MATHEMATICS

at the

RAND AFRIKAANS UNIVERSITY

Supervisor: PROFESSOR I. C. BURGER
Co-Supervisor: PROFESSOR J. HEIDEMA (UNISA)

JUNE 2004
Bedanking

"I am glad you found your way in here, for I am sure there is much that will interest you."

in Bram Stroker’s Dracula.

Aan die einde van hierdie mylpaal kyk ek terug en sien baie mense langs die pad staan. Eerstens wil ek my dank uitspreek teenoor ons Drie-Enige God wat my geseën het met talent, krag en deursettings vermoeë. Thanks to my husband, John, who lovingly assisted me in whatever he could, comforted me in times of doubt and for always encouraging me. Vir my ouers wat my grootgemaak het met liefde, my altyd bystaan en gesorg het dat ek verder kon studeer. My familie en vriende (ek is gelukkig genoeg om te veel te hê om hier te noem) wat my getrou aangemoedig en ondersteun het. Natuurlik sou hierdie skriptie nie die lig gesien het sonder Kinta se menigvuldige ure van hulp, leiding en baie aansporing nie. My dank gaan ook aan die mede-studie leier, Prof. Heidema en eksaminator, Prof. Britz vir hul bydraes en hulp. Met die belangstelling en hulp van almal hier bo genoem, het ek vlerke onder my voete gekry in tye van nood, baie helpende hande wanneer ek gestruikel het en kon ek hierdie skriptie voltooi het met ’n glimlag in my hart.

Baie dankie.
Petro
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1 Propositional and Predicate Logic</td>
<td>3</td>
</tr>
<tr>
<td>1.1 Propositional Logic</td>
<td>4</td>
</tr>
<tr>
<td>1.2 Predicate Logic</td>
<td>14</td>
</tr>
<tr>
<td>2 Modal Logic</td>
<td>19</td>
</tr>
<tr>
<td>2.1 Syntax and Semantics of Modal Logic</td>
<td>21</td>
</tr>
<tr>
<td>2.2 The System S5</td>
<td>25</td>
</tr>
<tr>
<td>3 Nonmonotonic Reasoning</td>
<td>28</td>
</tr>
<tr>
<td>3.1 Basic Notions, Principles and Topics regarding Nonmonotonic Reasoning</td>
<td>31</td>
</tr>
<tr>
<td>3.2 AGM Belief Change</td>
<td>37</td>
</tr>
<tr>
<td>3.3 KLM Nonmonotonic Reasoning</td>
<td>44</td>
</tr>
<tr>
<td>3.4 Epistemic Entrenchment</td>
<td>46</td>
</tr>
<tr>
<td>4 Many-Valued Logic</td>
<td>51</td>
</tr>
<tr>
<td>4.1 Three-Valued Logic of Łukasiewicz</td>
<td>53</td>
</tr>
<tr>
<td>4.2 Three-Valued Logic of Bochvar</td>
<td>56</td>
</tr>
<tr>
<td>4.2 Three-Valued Logic of Kleene</td>
<td>57</td>
</tr>
<tr>
<td>4.4 Many-Valued Generalization of Three-Valued Logic</td>
<td>59</td>
</tr>
<tr>
<td>4.4.1 Łukasiewicz’s Three-Valued Logic Adapated</td>
<td>59</td>
</tr>
<tr>
<td>4.4.2 Kleene’s Four-Valued Logic</td>
<td>61</td>
</tr>
<tr>
<td>4.4.3 General Construction of Many-Valued Logic</td>
<td>65</td>
</tr>
<tr>
<td>5 Fuzzy Logic</td>
<td>69</td>
</tr>
<tr>
<td>5.1 Fuzzy Sets</td>
<td>70</td>
</tr>
<tr>
<td>5.1.1 Operations and Properties of Fuzzy Sets</td>
<td>74</td>
</tr>
<tr>
<td>5.2 Methods in Application of Fuzzy Logic</td>
<td>75</td>
</tr>
<tr>
<td>Conclusion</td>
<td>83</td>
</tr>
<tr>
<td>References</td>
<td>86</td>
</tr>
<tr>
<td>Index</td>
<td>91</td>
</tr>
</tbody>
</table>
Introduction

"There is nothing permanent except change."
Anonymous

The last four decades have brought great social change with the increasing advent of computers into everyday life. However, a great challenge remains in that computers view life's problems as a series of yes and no, on and off, 1 and 0 or black and white answers.

Life, unfortunately (or fortunately) presents itself in a great variety of different shades of color, which can neither be represented as black nor as white. One might see it as different shades of grey which cannot easily be represented as distinct values such as true and false or 1 and 0.

The representation of the grey shaded areas have been a trial for logicians for many decades. The first attempts at resolving the challenge of mathematically depicting reasoning processes gave birth to propositional logic. The propositional language is restricted in the sense that it can only depict sentences as a whole, and only as either true or false.

Since grey areas have always been around, logicians started investigating more modalities than the traditional truth and falsity. As the modalities grew, so did the applications of logic. Today we find ourselves with the knowledge of fuzzy logic, enabling us to make machines almost think like humans.

It is important for the reader to bear in mind that the developments discussed here did not necessarily take place in such a linear order. This dissertation is only a general overview of various logics we found to have profoundly influenced the growth of logic from binary to fuzzy systems. Chapter 1 discusses propositional logic from both a syntactical and semantic point of view.
We also introduce the reader to predicate logic, which one may see as the first development of propositional logic into a more applicable logic, enhancing propositional logic with quantifiers and dividing sentences into components.

Modal logic is the main topic of Chapter 2, focusing on the addition of the modalities 'necessity' and 'possibility'. A discussion on the basic principles and notions (modal model, accessibly relation and standard model) of modal logic is followed by a general overview of the system S5.

Chapter 3 yields an overview of nonmonotonic reasoning, covering matters such as basic principles regarding nonmonotonic reasoning (retraction of beliefs), McDermott and Doyle's modal nonmonotonic reasoning, Moore's autoepistemic reasoning and Reiter's default logic. Another is that of belief change according to the AGM approach with the help of $\alpha$-removal and $\alpha$-revision. We discuss the KLM platform using the consequence relation and conclude the chapter with an inspection of epistemic entrenchment orderings.

Chapter 4 focuses on many-valued logic and specifically on selected many-valued logics. Starting the journey we investigate Łukasiewicz, Bochvar and Kleene's three-valued logic and their respective descriptions of the third truth value. These discussions bring us to the point were we look at some adaptations of Łukasiewicz and Kleene's three-valued logic, including some more truth values. Lastly in this chapter, we look at a general construction of many-valued logics.

Finally, we discuss fuzzy logic in Chapter 5 by looking at fuzzy sets, characterization functions and degrees of membership. Because fuzzy logic was developed for application purposes, we consider the fuzzy logic control and methods pertaining to it, namely fuzzyfication, inference rules and defuzzyfication. With this chapter we conclude our journey through time and development of mathematical logic.
Chapter 1

Propositional and Predicate Logic

"Logic is the hygiene the mathematician practices to keep his ideas healthy and strong."


Richard Hodel [1995] describes logic as the analysis of reasoning. The idea behind logical reasoning began with the Greek philosopher Aristotle in the 4th century B.C. His methods of reasoning did not develop mathematically, but nonetheless it was the first noted systematic study of logic.

Englishman George Boole (1815-1864) developed the first modern mathematical approach to logic in his 1847 paper The Mathematical Analysis of Logic. Boole described an algebra that can be interpreted as either propositional logic (to be discussed in Section 1.1) or as logic of syllogisms (an inference in which one proposition (the conclusion) follows from two others (known as premises) [Daintith and Nelson 1989]).

Another founding father was the German, Friedrich Ludwig Gottlob Frege (1848-1925), who in 1879 published a paper called Begriffsschrift (literally translated as ‘concept writing’) containing the idea of first-order logic (to be discussed in Section 1.2). Frege’s first order logic differs from propositional logic in the inclusion of quantifiers (\(\forall, \exists\)) and in the addition of symbols for constants, relations and functions.
Amongst other well known names associated with the development of mathematical logic is the Austrian-American Kurt Gödel (1906-1978), who in the 1930’s proved the completeness of the first-order calculus [Daintith and Nelson 1989]. Another is the German, David Hilbert (1862-1943), who posed his famous problems (questions) in 1900, greatly influencing development of mathematical logic in the twentieth century. Some of these questions can be found in Hodel [1995].

Bertrand Arthur William Russell (1872-1970) together with Alfred North Whitehead (1861-1947), were two Englishmen who published Principia Mathematica covering three volumes over the period 1910 to 1913. Principia Mathematica was their attempt to derive the whole of mathematics from purely logical principles [Daintith and Nelson 1989]. The logic we study today is strongly influenced still by the same principles.

We will now discuss some aspects of propositional and predicate logic. The latter will be discussed in less detail, as the former forms the foundation of the logics we are about to discuss in this dissertation.

1.1 Propositional Logic

Propositional logic was created mainly to use as a tool in philosophy, but grew into the well known (and well respected) classical propositional calculus, a basic system of two-valued logic. Propositional calculus introduces an unambiguous, symbolic language, which can be used to make deductions with a precision that is associated with the mathematical approach. However, it is not sufficient for all mathematical reasoning processes. In this section we will provide a brief outline of propositional logic, without dwelling on too much detail.

The syntax of a language refers to the arrangement and grammatical relations of the building blocks of the language. Within the scope of formal logic, the syntax of the logic concerns the study of the rules for constructions of expressions by employing the alphabet. These rules are stated wholly in terms of structure, without regard to meaning and truth.

To introduce the syntax of propositional logic, we would like to use an
example. Consider the following situation: An agent is sitting in a tower, looking down at people. Needing to spot an escaped convict in the crowd, he must also compile a written report on his observations as people go by. All the agent knows is that the convict is a man and that the convict's disguise will consist of a hat or glasses or both.

The agent, not wanting to write out 'the man is wearing a hat' or 'the man is wearing glasses' all the time, decides to use letters symbolizing these sentences. He assigns the letter $p$ to 'the man is wearing a hat' and the letter $q$ to 'the man is wearing glasses'. The sentences $p$ and $q$ are known as atomic sentences or atoms, since $p$ and $q$ do not contain any connectives (i.e. formal representations for 'and', 'or', etc.). Due to his need to report in a short hand fashion, the agent also requires some indicators for the possible combinations or connections of sentences that might arise. If a sentence consist of atoms combined with connectives it is called a compound sentence. To depict 'the man is not wearing a hat' he chooses to write $\neg p$, $\neg$ being the notation for the negation connective. If he spots a man wearing both glasses and a hat, he writes $p \land q$, $\land$ denotes the conjunction or meet of the sentences $p$ and $q$. Similarly $\lor$ (inclusively) is denoted by $\lor$ and is known as the connective for disjunction or join. There are also the conditional connective denoted by $\rightarrow$, which represents 'if...then...'. The biconditional connective represents compoundings of the form 'if and only if' and is denoted by $\leftrightarrow$.

According to Alexander Chagrov and Michael Zakharyaschev [1997], the symbols which form the alphabet of the propositional language are:

- the atoms or atomic sentences: $p, q, r,...$
- the logical constant $\bot$, also known as falsum or absurdum.
- the connectives: $\neg$, $\land$, $\lor$, $\rightarrow$, $\leftrightarrow$.
- parentheses: $\left\langle, \right\rangle$.

A proposition is a sentence which may be evaluated as true or false [Chagrov and Zakharyaschev 1997]. The following are examples of propositions:

- Dogs bark.
- The earth is flat.
- 2 plus 2 equals 4.

On the other hand, 'x is an odd number' is not a proposition, since the truth or falsity of this sentence depends on first specifying the value of x.

One can create a proposition – say $\alpha$ – by assigning $\alpha$ only to a atomic sentence or by creating a sentence by using a combination of the above mentioned symbols of the alphabet. For example $\alpha$ may denote the compound sentence $\neg(p \land q)$.

Every formal system has a set of formations rules, specifying construction of grammatically correct expressions. Often abbreviated as wffs (well-formed formulas), a proposition is a wff that is either true or false.

There exist various formal systems for logic. One such system is a Hilbert system, which is syntactically specified by an alphabet of symbols, a set of well formed formulas, a set of axioms and a finite set of inference rules. The last two concepts will be discussed shortly. A formal system is used when referring to a situation where symbols are being used and where the behaviour and properties of the symbols are determined completely by a given set of rules. In a formal system the symbols have no meaning, and in dealing with them one must be careful to assume nothing about their properties other than that specified in the system.

Hamilton [1988] states that axioms are wffs that are stipulated rather than proven to be so through the application of rules of inference. Rules of inference enables one to deduce a wff, say $\alpha$, as a syntactic consequence of a finite sequence of wffs, say $\alpha_1, ..., \alpha_k$, where each $\alpha_i, i = 1, 2, ...k$, is either an axiom or follows as a direct consequence of any two previous members of the sequence by applying the inference rules. The final step in such a sequence is known as a theorem. The axioms and rules of inference jointly provide a basis for proving all theorems. For our purposes we consider a propositional formal system which contains the following axiom schemas and rules of inference as stipulated by Chagrov and Zakharyaschev [1997].
Consider any wffs $\alpha$, $\beta$ and $\gamma$. The axiom schemas are:

(A1) $\alpha \rightarrow (\alpha \rightarrow \beta)$

(A2) $(\alpha \rightarrow (\alpha \rightarrow \gamma)) \rightarrow ((\alpha \rightarrow \beta) \rightarrow (\alpha \rightarrow \gamma))$

(A3) $(\alpha \land \beta) \rightarrow \alpha$

(A4) $(\alpha \land \beta) \rightarrow \beta$

(A5) $\alpha \rightarrow (\beta \rightarrow (\alpha \land \beta))$

(A6) $\alpha \rightarrow (\alpha \lor \beta)$

(A7) $\beta \rightarrow (\alpha \lor \beta)$

(A8) $(\alpha \rightarrow \gamma) \rightarrow ((\beta \rightarrow \gamma) \rightarrow ((\alpha \lor \beta) \rightarrow \gamma))$

(A9) $\bot \rightarrow \alpha$

(A10) $\alpha \lor (\alpha \rightarrow \bot)$

Note that each axiom schema has infinitely many instances, as $\alpha$, $\beta$ and $\gamma$ range over all the wffs in the propositional language.

In this formal system, built on the axioms above, the inference rule of *Modus Ponens* (MP) is applied to deduce the rest of the theorems. MP states that if the two propositions $\alpha$ and $\alpha \rightarrow \beta$ are assumed, then $\beta$ is deduced.

Falsum ($\bot$) represents the set of all logical contradictory sentences. The axiom schema (A9) states that any proposition can be deduced from falsum. In this sense, falsum is the logical "strongest" proposition.

Some authors choose to use a smaller set of connectives in their alphabets, the reason being that the pairs of connectives $\{\neg, \land\}$, $\{\neg, \lor\}$ and $\{\neg, \rightarrow\}$ are *functionally complete (adequate) sets of connectives*, meaning that the other connectives can be expressed in terms of the chosen pair. To illustrate the adequacy of the above mentioned pairs of connectives, we will only consider the pair $\{\neg, \rightarrow\}$. (For a complete proof the reader may refer to Hamilton [1988].) Having $\{\neg, \rightarrow\}$ as the given set of connectives, $\alpha \land \beta$ can be expressed
as \( \neg(\alpha \rightarrow \neg \beta) \) and \( \alpha \lor \beta \) as \( \neg \alpha \rightarrow \beta \). Please note that \( \neg \alpha \) may be expressed as \( \alpha \rightarrow \bot \).

The selection of the set of connectives influence the set of axioms used. Different sets of axioms may generate the same set of theorems and there are many alternative axiomatizations of a propositional formal system. Hamilton [1988] uses only the pair \( \{\neg, \rightarrow\} \) as an adequate set of connectives to formulate the following set of axiom schemas where \( \alpha, \beta \) and \( \gamma \) are any wffs:

(a1) \( \alpha \rightarrow (\beta \rightarrow \alpha) \)

(a2) \( (\alpha \rightarrow (\beta \rightarrow \gamma)) \rightarrow ((\alpha \rightarrow \beta) \rightarrow (\alpha \rightarrow \gamma)) \)

(a3) \( ((\neg \alpha) \rightarrow (\neg \beta)) \rightarrow (\beta \rightarrow \alpha) \).

Hamilton [1988] employs the Modus Ponens rule of inference together with the above axioms to deduce all theorems.

Next, we provide a brief outline of the semantics of propositional logic, that is (plainly put) the assignment of truth values to atoms and compound sentences.

As previously mentioned, a proposition is a sentence which may be evaluated as true or false. The truth value of a proposition may be viewed by using a truth table. We will use the number 1 to portray the assignment of 'true' and 0 for 'false'. Consider the truth table of the compound propositions \( \neg p \), \( p \land q \) and \( p \lor q \):

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>\neg p</th>
<th>p \land q</th>
<th>p \lor q</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Since the values assigned range between two possible values (true or false), one can calculate the number of rows needed by using the number of atoms in the propositional sentence. In the above case, where \( p \) and \( q \) are the atoms, \( 2^2 \) rows are required. When there are \( n \) atoms, one would use \( 2^n \) rows, filled with all possible combination of zero's and one's.
Every row in the truth table assigns a truth value to \( p \) and \( q \) individually. A row, as a unity, may be seen as a valuation. Take for example the case where the man is wearing a hat, but he is not wearing glasses, then 1 would be assigned to \( p \) and 0 to \( q \). This valuation corresponds to the second row in the truth table above. If one wants to look at the instance where both \( p \) and \( q \) are true, it corresponds to the row in which the truth values given to \( p \) and \( q \) both need to be 1 and hence the conjunction \( p \land q \) is true. For the sentence \( p \lor q \), at least one of the values of \( p \) or \( q \) needs to be 1.

Formally, Grzegorz Malinowski [1993] states that valuations are functions applied to the set of propositions, yielding either a 0 or a 1. With \( Z \) being the set of propositions, a valuation can be defined as a function \( v : Z \to \{0, 1\} \) such that for any \( \alpha, \beta \in Z \),

\[
\begin{align*}
v(\neg \alpha) &= 1 \text{ if and only if } v(\alpha) = 0 \\
v(\alpha \lor \beta) &= 0 \text{ if and only if } v(\alpha) = v(\beta) = 0 \\
v(\alpha \land \beta) &= 1 \text{ if and only if } v(\alpha) = v(\beta) = 1 \\
v(\alpha \rightarrow \beta) &= 0 \text{ if and only if } v(\alpha) = 1 \text{ and } v(\beta) = 0 \\
v(\alpha \leftrightarrow \beta) &= 1 \text{ if and only if } v(\alpha) = v(\beta) \\
v(\bot) &= 0.
\end{align*}
\]

We may also express the valuations of \( \neg \alpha, \alpha \lor \beta \) and \( \alpha \land \beta \) as follows:

\[
\begin{align*}
v(\neg \alpha) &= 1 - v(\alpha) \\
v(\alpha \lor \beta) &= \max\{v(\alpha), v(\beta)\} \\
v(\alpha \land \beta) &= \min\{v(\alpha), v(\beta)\}
\end{align*}
\]

Note that the truth value of any compound proposition is determined by the truth values of the atoms it contains. Keeping in mind the definition above, that, for example \( v(p \lor q) = 0 \) if and only if \( v(p) = v(q) = 0 \), one may consider any valuation as a function restricted to the set of atoms. Assigning valuations to propositions can be tedious if one has to write out, for example,
v(p) = 1 and v(q) = 0. Instead, we introduce an abbreviated sequence-notation for valuations, which for the above example would read ‘10’, where the first position in the sequence is associated with v(p) and the second position is associated with v(q). Note that a valuation may also be called a (possible) world, a (possible) state or an interpretation.

The semantic representation of, for example p ∨ q, can be done by referring to the set of models of p ∨ q, written as Mod(p ∨ q). The set of models of a proposition consists of all those valuations (restricted to the set of atoms) under which the specific sentence is true. The models of a certain proposition can be found by using the truth table and selecting only those rows for which the proposition has the value of 1. Thus, for every proposition α, Mod(α) is a subset of the set of all valuations. (For example, in our p, q-language, Mod(p ∨ q) = {11, 10, 01}, which is a subset of the set {11, 10, 01, 00}.)

Returning to the truth-meaning of the connectives, the truth tables for the conditional and the biconditional connectives are:

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>p → q</th>
<th>p ↔ q</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

The truth table of the conditional connective (→) ‘if...then...’ may be understood if viewed as a “promise-maker”. For example, the executive of your company informs you: If you do this job, then I will increase your salary. The only way for this promise to be broken will be if you do the job and the executive does not increase your salary. Any other combination would not result in a broken promise. The executive might increase your salary out of good will, even if you did not do the job (then everybody would love to work for him!), but this particular action does not imply a breaking of the promise. Thus, the only valuation where the conditional is not true, is when p is assigned the value of 1 and q is designated the value of 0. The set of models for p → q is hence {11, 01, 00}.  

10
The biconditional connective (\(\leftrightarrow\)) ‘if and only if’ brings to mind a mutually binding contract. The contract will be broken if one party commits to his part of the contract, while the other party does not. The only valuations under which the biconditional are true, are those assigning 1 to both \(p\) and \(q\), or 0 to both \(p\) and \(q\). Thus \(\text{Mod}(p \leftrightarrow q) = \{11, 00\}\).

The truth table method can be used for any proposition, varying over any number of atoms and connectives. For example, consider the truth table of the proposition \((p \land r) \leftrightarrow q\):

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>(p \land r)</td>
<td>((p \land r) \leftrightarrow q)</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

To find the valuations that make the proposition \((p \land r) \leftrightarrow q\) true – i.e. the models of \((p \land r) \leftrightarrow q\) – one will only consider the rows where 1 appears under the column \((p \land r) \leftrightarrow q\). These are:

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>(p \land r)</td>
<td>((p \land r) \leftrightarrow q)</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

The set of valuations \(\{111, 100, 001, 000\}\) represents the set of models of the proposition \((p \land r) \leftrightarrow q\).

When all the rows in a truth table of a proposition have the value of one, the proposition is called a tautology, denoted by \(T\). This indicates the sentence
is true in all possible states (i.e. under all valuations) of the system and represents a logical fact [Hodel 1995]. As an example, consider the proposition \( p \to p \) with \( \text{Mod}(p \to p) = \{11, 10, 01, 00\} \). A tautology is true in all possible states, but useless in the sense that it does not really state anything profound. It is like saying 'if the man is wearing a hat, then he is wearing a hat' (you will not find much more truth than that). On the other hand, the truth table of \( \bot \) (which represents all contradictions) consists of only zeros in the main column. An example of a contradiction is the proposition \( p \land \neg p \), stating that 'the man is wearing a hat and he is not wearing a hat'.

Returning to our example of the agent and the convict, assume a possible suspect is standing right beneath the agent's tower. The agent can thus only see the top of the man and he can see the man is wearing a hat. Whether the man is wearing glasses or not is still unknown to the agent, but with what he knows he can at least say \( p \) must be assigned the value 1. Given the information, namely that \( p \) is the case, the set of all possible states (valuations) may thus be shrunk to \( \{11, 10\} \) — i.e. the set of models of \( p \). The more the agent knows, the smaller the set of states representing his knowledge will be. It makes perfect sense if one thinks about 'no knowledge' as the set containing all possible valuations — that is when the agent still has all the options to choose from. As time goes by and more information is gathered, the set of acceptable states shrinks. States that are definitely not possible in the light of the given information are excluded from the set of possible acceptable circumstances. If the man is wearing a hat, it is definitely not the case that he is not wearing a hat. It is therefore senseless to hold on to the states indicating the models of 'the man is not wearing a hat'.

Any set of valuations in a finitely generated language (for example, our language generated by \( p \) and \( q \)) is describable by a proposition. For example the set \( \{11, 00\} \) is describable by \( p \leftrightarrow q \), since \( \{11, 00\} \) is the set of models of \( p \leftrightarrow q \). Any tautology describes the set of all possible states.

To conclude this section we would like to introduce a few last concepts. Consider the states where the following sentences are true; \( p \land q \) being true at state 11 and \( p \to q \) being true at states 11, 01 and 00. The proposition
$p \land q$ entails $p \rightarrow q$, because $p \rightarrow q$ is true at every model of $p \land q$ [Daintith and Nelson 1989]. For any propositions $\alpha$ and $\beta$ (for example $\alpha = p \land q$ and $\beta = p \rightarrow q$), $\alpha$ entails $\beta$, or equivalently $\beta$ is a semantic consequence of $\alpha$, written as $\alpha \models \beta$, if and only if $\text{Mod}(\alpha) \subseteq \text{Mod}(\beta)$. Two sentences $\alpha$ and $\beta$ are logically equivalent (denoted by $\alpha \equiv \beta$) if and only if $\alpha \models \beta$ and $\beta \models \alpha$ (i.e. $\text{Mod}(\alpha) = \text{Mod}(\beta)$). When we write $\models \alpha$, it means $\alpha$ is a tautology.

As an example we consider the Lindenbaum-Tarski algebra generated by two propositional atoms $p$ and $q$ and the usual logical connectives. Under logical equivalence of sentences, this yields the sixteen element diagram depicted in figure 1.1. The class of tautologies is the top element $T$, the class of contradictions is the bottom element $\bot$; and $\alpha \models \beta$ if $\alpha$ lies below $\beta$ in the diagram and $\alpha$ and $\beta$ are connected by a line (either directly or transitively). Note that $p + q$ is equivalent to $(p \land \neg q) \lor (\neg p \land q)$ [Burger and Heidema 2002].

The early philosophers found great use of propositional logic and today it still forms the basis of mathematical reasoning. It is however restricted.
Propositional logic is a logic at the sentential level. The smallest unit one deals with in propositional logic is an atomic sentence representing a certain statement as a whole, without reflecting any internal relationship between objects. However, these relationships are representable in predicate logic – the topic of the next section.

1.2 Predicate Logic

Predicate logic (or calculus) is an extension of propositional logic, since it is a formal system concerned with the representation of logical relations between sentences or propositions as wholes, as well as their internal structures in terms of subjects and predicates. With predicate calculus it is possible to consider the relationship of different parts of a single sentence with each other or with components of entirely different sentences. Since the logical systems discussed in chapters to come are mainly based on propositional logic, we omit a very detailed discussion of predicate logic. The purpose of this section is therefore to only touch upon some aspects of first-order predicate logic, illustrating it as an expansion of propositional logic.

Propositional logic is restricted in the sense that one uses large chunks of mathematical language to assign symbols to those parts that can have a truth value, for example, we may assign $p$ to a whole sentence like ‘the man is wearing a hat’. A simple argument, such as ‘all clowns wear hats’, ‘Coco is a clown’ therefore it follows that ‘Coco is wearing a hat’, cannot be dealt with in propositional logic. Within the propositional language one would be able to assign $p$ to ‘all clowns wear hats’ and $q$ to ‘Coco is a clown’, but how, just by looking at the atoms $p$ and $q$, would one be able to draw the logical conclusion, say $r$, representing the sentence ‘Coco is wearing a hat’? One cannot say $p$, $q$, thus $r$. The moral is that one has to extend the language in such a way as to be able to discuss subjects (objects) and relations. In particular we wish to introduce means to talk about all subjects of a certain domain. Dually, we want a means of expressing ‘there exists a subject such that...’

Predicate logic introduces the “break down” of a sentence into different
and it creates the ability to relate these components with each other or with the components of a different sentence. Hamilton [1988] states that every simple statement in English has a subject and a predicate, each of which may consist of a single word, a short phrase or a whole clause. The statement makes an assertion about the subject and the predicate (the latter refers to a property which the subject has). Predicates in sentences are represented by capital letters, $A, B, C,$..., while subjects are represented by small letters, $a, b, c, x, y, z$. For example the sentence 'a clown always wears a hat' has 'clown' as subject and 'wearing a hat' as predicate; $C(x)$ may denote 'x is a clown' and $H(x)$ may denote 'x wears a hat'. Continuing with this example, we may express 'a clown always wears a hat' as $C(x) \rightarrow H(x)$, which can also be read as 'if x is a clown, then x is wearing a hat'. One may link a property (predicate) to a particular individual (subject), for example 'Coco is a clown' written as $C(c)$, where $c$ denotes Coco.

In combination with the above mentioned components, there are quantifiers to assist in the predicate language. Quantifiers are symbols to denote 'for all' and 'there are some' or 'there exists'. They are symbolically represented respectively by $\forall$, the universal quantifier, and $\exists$, the existential quantifier. For example, the sentence 'all clowns wear hats' can be written as $(\forall x)(C(x) \rightarrow H(x))$. If we consider the expression $C(c)$, it is now possible to conclude $H(c)$ for 'Coco is wearing a hat', since Coco falls under 'for all x' [Mints 1992].

To summarize what is said above, we built our predicate language from symbols for properties and relations (i.e. predicates) and subjects. Furthermore, we add variables to range over subjects and the usual propositional connectives, but now also include the quantifiers. To put it formally a first-order language, denoted by $L$, as described by Hamilton [1988], consists of

- variables: $x_1, x_2, x_3,\ldots$
- individual constants: $a_1, a_2, a_3,\ldots$
- connectives: $\neg, \land, \lor, \rightarrow, \leftrightarrow$.
- the parentheses: $(, )$.  

15
quantifiers: \( \forall, \exists \).

predicates: \( A_1, A_2, \ldots \).

function letters: \( F_1, F_2, \ldots \).

(For the sake of simplicity we do not indicate the arity of the predicates and functions in their formal notation.)

In predicate logic we distinguish between terms and well-formed formulas. A term in \( L \) is defined as follows:

- Variables and individual constants are terms.
- If \( F_i \) is a function letter (of \( n \)-arity) in \( L \), and \( t_1, t_2, \ldots t_n \) are terms in \( L \) then \( F_i(t_1, t_2, \ldots t_n) \) is a term in \( L \).

The set of all terms is generated by the two points above and represents those expressions which can be interpreted as objects in the formal language (i.e. the “things” to which function predicates are applied). Well formed formulas (wffs) are built up from atomic formulas by means of the connectives (refer to section 1.1) as well as the quantifiers. An atomic formula in \( L \) is defined as follows: if \( A_i \) is a predicate letter (of \( n \)-arity) in \( L \) and \( t_1, t_2, \ldots t_n \) are terms in \( L \), then \( A_i(t_1, t_2, \ldots t_n) \) is an atomic formula of \( L \). Atomic formulas are the simplest formulas in the language.

The semantics of predicate logic is much more complicated than that of propositional logic. In the context of propositional logic, ‘interpretation’ refers to the assignment of truth values (zeros and ones). However, in predicate logic an ‘interpretation’ extends to describing a domain and assigning meaning to constants, functions and predicates within the context of the domain. Without dwelling on the details we discuss a simple example which captures some of the notions above.

Consider the following wff, say \( \alpha \),

\[
(\forall x_1)(\forall x_2)(A_1(F_1(x_1, x_2), a_1) \rightarrow A_1(x_1, x_2)).
\]

Within the wff \( \alpha \) certain parts are terms (for example \( F_1(x_1, x_2) \)) and certain parts are atomic formulas (for example \( (A_1(F_1(x_1, x_2), a_1)) \)). The scope of a quantifier in a wff refers to that part of the wff that is governed by the given
quantifier. For example the scope of \( (\forall x_1) \) in \( \alpha \) is \( (\forall x_2)(A_1(F_1(x_1, x_2), a_1) \rightarrow A_1(x_1, x_2)) \). The variables \( x_1 \) and \( x_2 \) occurring within \( \alpha \) in example (1) are said to be *bound* as both fall within the scope of a quantifier. If an occurrence of a variable is not bound it is said to be *free*. If no variable occurs free within a wff, then that particular wff is called *closed* as is the wff \( \alpha \) above.

One may describe many interpretations for the language which is used to build wff \( \alpha \) above. We provide one example: let \( a_1 \) be interpreted as '0', \( A_1 \) as '=' , \( F_1 \) as '+' and let the domain be the set of natural numbers. Within this interpretation \( \alpha \) can be written as \( (\forall x_1)(\forall x_2)((x_1 + x_2 = 0) \rightarrow (x_1 = x_2)) \).

Since \( \alpha \) is a closed wff it can be either true or false in an interpretation, but in this particular interpretation it is false (because \( (x_1 + x_2 = 0) \rightarrow (x_1 = x_2) \) does not hold for *all* natural numbers \( x_1 \) and \( x_2 \); it only holds if \( x_1 = x_2 = 0 \)). If the variables \( x_1 \) and \( x_2 \) are not bound and wff \( \alpha \) is reduced to \( (A_1(F_1(x_1, x_2), a_1) \rightarrow A_1(x_1, x_2)) \) – the wff cannot be determined as true or false in the above interpretation, since the variables \( x_1 \) and \( x_2 \) are free [Van Dalen 1983].

The axioms and rules of inference for a first-order system according to Hamilton [1988] are described as follows: For wffs \( \alpha, \beta \) and \( \gamma \) of \( \mathcal{L} \)

(K1) \( \alpha \rightarrow (\beta \rightarrow \alpha) \).

(K2) \( (\alpha \rightarrow (\beta \rightarrow \gamma)) \rightarrow ((\alpha \rightarrow \beta) \rightarrow (\alpha \rightarrow \gamma)) \).

(K3) \( (\neg \alpha \rightarrow \neg \beta) \rightarrow (\beta \rightarrow \alpha) \).

(K4) \( (\forall x_1)\alpha \rightarrow \alpha, \) if \( x_i \) does not occur free in \( \alpha \).

(K5) \( (\forall x_i)\alpha(x_i) \rightarrow \alpha(t), \) if \( \alpha(x_i) \) is a wff in \( \mathcal{L} \) and \( t \) is a term in \( \mathcal{L} \) which is free for \( x_i \) in \( \alpha(x_i) \).

(K6) \( (\forall x_i)(\alpha \rightarrow \beta) \rightarrow (\alpha \rightarrow (\forall x_i)\beta), \) if \( \alpha \) contains no free occurrence of the variable .

Notice that these are axiom schemas, each with infinitely many instances.
The rules of inference are defined as:

**Modus Ponens** from $\alpha$ and $(\alpha \rightarrow \beta)$ one can deduce $\beta$, where $\alpha$ and $\beta$ are wffs of $\mathcal{L}$

**Generalization** from $\alpha$ deduce $(\forall x_i)\alpha$, where $\alpha$ is a wff of $\mathcal{L}$ and $x_i$ is any variable.

The notion of ‘tautology’ (as was described in the propositional case) can be extended to predicate logic where it is synonymously used with the notion of ‘logically valid’. *Logically valid wffs* are wffs true in any interpretation of $\mathcal{L}$. All the axioms are examples of logically valid wffs. On the other hand, a wff is a *contradiction* if it is false in every interpretation.

The basic ideas introduced in propositional logic is amidst the predicate logic. However, it is clear that predicate logic provides a formal structure enabling one to capture the reasoning process in more detail than what is possible in propositional logic. For the actual practice of mathematics, predicate logic is doubtlessly the perfect tool, since it allows us to handle individuals.

We content ourselves with this short introduction to first-order predicate logic illustrating the expansion of propositional logic into predicate logic. The very fundamental properties of predicate logic, e.g. compactness, completeness, soundness and adequacy are not discussed, but are found in Van Dalen [1983]. For an explication of Gödel’s Incompleteness Theorem we refer the reader to Hamilton [1988].

In the next few chapters, the reader will find some more development in the field of logic, which for example, expands the logic with regards to different modalities (when considering modal logic) or (when considering nonmonotonic reasoning) the expansion and retraction of beliefs an agent holds in his pool of knowledge and consequent beliefs. Throughout all of these further developments however, it will stay clear that the basic building blocks stays the same as stated in this chapter.
Chapter 2

Modal Logic

"Mathematics takes us into the region of absolute necessity, to which not only the actual word, but every possible word, must conform."

Bertrand Russell

The aim of this chapter is to provide a brief overview of selected aspects in modal logic to the extent necessary for a discussion of nonmonotonic logic (Chapter 3). An exhaustive presentation of modal logics can be found in several good monographs on the subject. For example, Chellas [1980], Fitting [1983], and Hughes and Cresswell [1996].

Modal logic concerns the study of concepts such as ‘possibility’, ‘necessity’ ‘contingency’, etc., and of formal systems of which the intended interpretation includes such concepts. In this sense it is an extension of first order predicate (and hence, propositional) logic.

Modal logic is the result of the discovery that there might be more than one possible “surrounding” in which certain events can take place. Keeping this in mind, one can say that some events will definitely be true in a certain environment, while other events only have a possibility to be true in that environment. There are even some possible events that will definitely not be true in the mentioned environment, and these events can thus be excluded from the set of possible events to be considered.

Modal logic dates back to the German philosopher Gottfried Wilhelm Leibniz (1646-1716), who described a proposition as having two possible conditions.
One of the conditions a proposition may have is that it is *necessary* if it holds true in all possible worlds (environments) and the other is that a proposition is *possible* if it holds true in some world or worlds, i.e. in at least one of the worlds [Chellas 1980].

G.E. Hughes and M.J. Cresswell [1996] describe modal logic as the logic of 'may be' and 'must be'. This standard modality has been studied since Aristotle and was fully recognized by the German, Immanual Kant (1724-1804), in his *Kritik der Reinen Vernunft* [Doherty 1996]. Frege dismissed modality from formal logic – using the argument that if a proposition is necessarily true, one can only give an impression of the reason for one's judgement, and according to him it is only the content of a judgement that is logically relevant.

Apart from the most frequent use of modality in mathematical logic – where modality is viewed as 'necessary' and 'possible' – it is used in many other diverse areas like philosophy, computer science and artificial intelligence. Patrick Doherty [1996] refers to the standard interpretation of necessity (using the symbol $\Box$) as *must be*. Some other modal interpretations are in temporal modality using *always* as the form of necessity, deontic modality with *ought to be* or *obligatory* as its form of necessity, default modality with *normally* as an interpretation of necessity, epistemic modality using *known* and doxastic modality with *believed* [Blackburn 1993].

The mathematical build up to certain modal models consists of many building blocks. We will introduce the reader to some of these in this dissertation, but will leave further investigation up to the curious. It is important to note that there are many different notations used by different authors. This dissertation will conform mostly to the notation used by Brian F. Chellas [1980], although some adaptations were made for the sake of compatibility to the notations used in Chapter 1.
2.1 Syntax and Semantics of Modal Logic

Chellas [1980] views the study of modal logic in the context of a language of necessity and possibility where, added to the symbols of the propositional language (of Chapter 1), we have the following primitive symbols: \( T \), \( \Box \) and \( \Diamond \). \( T \) is constant truth. \( \Box \) is the symbol for necessity whereas \( \Diamond \) is the symbol for possibility.

In this language a sentence \( \Box \alpha \) is read as 'necessarily \( \alpha \)' and \( \Box \alpha \) is true if and only if \( \alpha \) itself is true at every possible world. On the other hand, \( \Diamond \alpha \) ('possibly \( \alpha \)') is true just in case \( \alpha \) is true at some possible world. To describe this formally we consider the notion of a modal pair.

A modal pair \( M = < W, P > \) consists of \( W \) being a set of possible worlds and \( P \) being an infinite abbreviation of the sequences \( P_1P_2P_3... \) of subsets of \( W \). The intuition is that for each natural number \( n \), the set \( P_n \) collects all the possible worlds at which the corresponding atomic sentence \( p_n \) is true, thus \( p_n \) is true at possible world \( w \) if and only if \( w \) is in the set \( P_n \).

To illustrate the above notions, consider again the familiar escaped convict example. We take \( W \) to be the set of all possible worlds (within our \( p,q \)-language), i.e. \( W = \{11, 10, 01, 00\} \), or also \( W = \{w_1, w_2, w_3, w_4\} \), with \( w_1 = 11 \), \( w_2 = 10 \), \( w_3 = 01 \) and \( w_4 = 00 \). Remember that \( p \) is assigned to 'the man is wearing a hat' and \( q \) is assigned to 'the man is wearing glasses'.

Within this context (with \( p \) and \( q \) being the only atoms, corresponding to \( p_1 \) and \( p_2 \) respectively), an example of a modal pair is \( M_1 = < \{w_1, w_2, w_3, w_4\}, P > \) with \( P = P_1P_2 \), where \( P_1 = \{w_1, w_2\} \) and \( P_2 = \{w_1, w_3\} \).

Let \( \alpha \) be any sentence. In what follows, \( \models_w^M \alpha \) denotes that proposition \( \alpha \) is true at world \( w \) and \( \not\models_w^M \alpha \) that \( \alpha \) is false at \( w \), where \( w \) is an element of \( W \) in an arbitrary modal pair \( M = < W, P > \).

We shall use the notion of modal model of \( \alpha \) to describe the situation when for a particular \( M = < W, P > \), \( \models_w^M \alpha \) for every \( w \in W \).

Before we formally define \( \Box \alpha \) we explain the meaning of \( \Box \) by an example: Consider again the pair \( M_1 \) as described above. \( M_1 \) is not a modal model of \( \Box p \), since \( p \) is not true at every world contained in \( W \). The reason being that \( w_3 \) and \( w_4 \) are included in \( W \) at which \( p \) is false. However \( M_1 \) is a modal model...
of $p \rightarrow p$ and hence also of $\Box(p \rightarrow p)$. In fact, $M_1$ is a modal model of every propositional tautology.

When a sentence $\alpha$ is true at every possible world in every model pair, $\alpha$ is valid. We denote this valid status of $\alpha$ by $\models \alpha$. (In the next section we shall consider a few of the many valid sentences.)

The truth conditions of the sentences of the language, can be depicted in the following symbolic form [Chellas 1980]. For an arbitrary modal pair $M = \langle W, P \rangle$:

(B1) $\models_w p_n$ if and only if $w \in P_n$, $n = 1, 2, 3...$ (An atomic sentence $p_n$ is true at the possible world $w$ if and only if $w$ is a member of the set $P_n$.)

(B2) $\models_w T$. (The truth constant $T$ is always true at $w$.)

(B3) $\not\models_w \bot$. (The falsity constant $\bot$ is always false at $w$.)

(B4) $\models_w \neg \alpha$ if and only if $\not\models_w \alpha$. (The negation $\neg \alpha$ is true at $w$ if and only if its negate $\alpha$ is false at $w$.)

(B5) $\models_w \alpha \land \beta$ if and only if both $\models_w \alpha$ and $\models_w \beta$. (Conjunction $\alpha \land \beta$ is true at $w$ if both its conjuncts are true at $w$.)

(B6) $\models_w \alpha \lor \beta$ if and only if either $\models_w \alpha$ or $\models_w \beta$, or both. (Disjunction $\alpha \lor \beta$ is true at $w$ when at least one of $\alpha$ or $\beta$ is true at $w$.)

(B7) $\models_w \alpha \rightarrow \beta$ if and only if $\models_w \alpha$ then $\models_w \beta$. (Conditional $\alpha \rightarrow \beta$ is true at $w$ if it fails to be the case that the antecedent ($\alpha$) is true at $w$ while the consequent ($\beta$) is false.)

(B8) $\models_w \alpha \leftrightarrow \beta$ if and only if, $\models_w \alpha$ if and only if $\models_w \beta$. (Biconditional $\alpha \leftrightarrow \beta$ is true at $w$ just in case its members $\alpha$ and $\beta$ are either both true, or both false at $w$.)

(B9) $\models_w \Box \alpha$ if and only if for every $w'$ in $M$, $\models_{w'} \alpha$. (Necessitation $\Box \alpha$ is true at $w$ if and only if necessitate, $\alpha$, is true at every possible world $w'$ in $M$.)
(B10) $\models^M_w \Diamond \alpha$ if and only if for some $w'$ in $M$, $\models^M_w \alpha$. ($\Diamond \alpha$ is true at $w$ if there is at least one world $w'$ in $M$, at which $\alpha$ is true.)

The operators $\Box$ and $\Diamond$ are dual in the sense that, for all sentences $\alpha$, $\Diamond \alpha$ is logical equivalent to $\neg \Box \neg \alpha$ with respect to every pair $M = \langle W, P \rangle$. (Duality is the property between two operators when the one operator may be equivalently described in terms of the other operator.) $\Diamond \alpha$ being equivalent to $\neg \Box \neg \alpha$ means that if $\alpha$ is possible, then it is not the case that $\neg \alpha$ is necessary [Chagrov and Zakharyaschev 1997].

To enhance the understanding of modal logic, consider the hat-and-glasses example once again, where the escaped convict will either wear a hat, or glasses, or both a hat and glasses as a disguise. By now we know that one can only claim $\Box p$ – it is necessarily so that the man wears a hat – if $p$ is true at all the possible worlds in $W = \{w_1, w_2, w_3, w_4\}$. But this is not the case. However, as said above, propositional tautologies like $p \rightarrow p$ may be “labelled” as $\Box (p \rightarrow p)$, because they are true at all possible worlds. Considering this information, one might ask where the power of $\Box$ lays? Is the label $\Box$ just another way to convey propositional tautologies? This will be answered when we look at the accessibility relation on possible worlds. For now at least, we can safely claim $\Diamond p$, as it is true that in at least one of the possible worlds the man is wearing a hat. In fact $\Diamond p$ is true at every world in $W$.

If we assume that the agent knows that the man is wearing a hat, we may exclude $w_3$ and $w_4$ and hence we need to consider only two possible worlds, namely $w_1$ and $w_2$. Thus we may shrink the set $W$ of all possible worlds to a new set, say $W'$, containing only the “relevant worlds”, namely $w_1$ and $w_2$. Considering the interpretation of $\Box$ within this new, smaller set $W'$ means that $\Box$ will no longer be used to “label” propositional tautologies. This approach is known as the epistemic approach to $\Box$ since it deals with $\Box$ in the context of what the agent knows.

Within $W'$ the sentence $p$ assumes a “tautologous” status, i.e. $p$ is true at every $w \in W'$. One way of interpreting $\Box$ in a more flexible way (i.e. not only as a label for propositional tautologies) is to change the cardinality of $W$ (for example, shrinking $W$ to $W'$). However, employing a specific binary relation
on \( W \), will also yield the new status of \( \Box \). To formalize what is said above, we introduce the notion of an accessibility relation.

According to Chellas [1980], an accessibility relation \( R \), is a binary relation between possible worlds. A modal pair \( M = \langle W, P \rangle \) is extended to a trio \( M = \langle W, R, P \rangle \) where \( W \) and \( P \) remain as defined previously. One writes \( wRw' \) and it is interpreted as the world \( w' \) is relative to the world \( w \). It can also read \( w' \) is accessible from \( w \). This trio is called the standard model, but it is also known as the Kripke model [Fitting 1983, Hughes and Cresswell 1996].

Within the context of our \( p, q \)-language, let \( R \) be the following accessibility relation on \( W = \{w_1, w_2, w_3, w_4\} \):
\[
R = \{(w_1, w_1), (w_1, w_2), (w_2, w_2), (w_2, w_1), (w_3, w_3), (w_4, w_4), (w_4, w_3)\}
\]
Thus \( R \) is an equivalence relation (i.e., a reflexive, transitive and symmetrical relation), in which \( w_1 \) and \( w_2 \) are equivalent as well as \( w_3 \) and \( w_4 \). Thinking about the hat and glasses example, one may think of the relation \( R \) in the following way: If the agent can see the man is wearing a hat, he can consider himself as being in \( w_1 \). According to the relation \( R \) he can also regard \( w_2 \) as a possibility (as accessible) because \( (w_1, w_2) \in R \).

Chellas [1980] defines the role of \( \Box \) and \( \Diamond \) in light of the standard trio \( M = \langle W, R, P \rangle \), as follows: Let \( \alpha \) be any element of \( W \), then

\[(B11) \models^M w \Box \alpha \text{ if and only if for every } w' \text{ in } M \text{ such that } wRw', \models^M w' \alpha\]

\[(B12) \models^M w \Diamond \alpha \text{ if and only if for some } w' \text{ in } M \text{ such that } wRw', \models^M w' \alpha\]

Note that (B11) and (B12) replace (B9) and (B10) respectively, but (B1) to (B8) retain their definitions.

Consider the modal pair \( M_1 = \langle W, P \rangle \) and the modal trio \( M_2 = \langle W, R, P \rangle \) with \( W = \{11, 10, 01, 00\} = \{w_1, w_2, w_3, w_4\} \), \( P = P_1P_2 \) with \( P_1 = \{11, 10\} \), \( P_2 = \{11, 01\} \) and \( R \) the equivalence relation as given above. Then, \( \not\models^M w \Box p \) for every \( w \in W \), i.e., \( \Box p \) is false at every \( w \in W \). However in \( M_2 \), \( \Box p \) is true at \( w_1 \) and \( w_2 \) (and false at \( w_3 \) and \( w_4 \)). This means that \( M_1 \) and \( M_2 \) are modal models of the sentence \( \Box p \rightarrow p \).

The modal pair \( M_1 = \langle W, P \rangle \) may easily be converted to a trio. In other words, we may add an accessibility relation \( R \) to \( M_1 \) without changing the
truth value of any sentence with regard to any world in $W$. This means a conversion without changing the status of $\Box$ (namely that of necessity) in $M_1$. We just let $R$ be the relation on $W$ consisting of all pairs, i.e. $R = W \times W$. Then every world in $W$ is accessible from every world in $W$. So $\Box \alpha$ is true at $w$ if and only if $\alpha$ is true at every $w \in W$ – just as required within the necessity context.

This concludes the discussion on the general aspects regarding modal logic. In the next section we discuss one of the formal systems developed in modal logic, known as the system $S5$.

### 2.2 The System S5.

Quite a number of formal modal systems were developed over time. Mentioning a few, we will briefly discuss only one of these systems.

Modal logic is based on classical propositional calculus. Using the axioms and rules of inference discussed in Chapter 1 as a basis, we shall now examine the modal system $S5$ (as described in Hughes and Cresswell [1996] and Heidema and Labuschagne [2000]). It is important to note that the use of $\Box$ in the following description is connected with knowledge, i.e. $\Box$ is interpreted as ‘it is known’ (on the other hand $\Diamond$ represents ‘it is believed’). This system is built on the following axiom schemas:

1. $\Box \alpha \rightarrow \alpha$ This schema indicates that whatever is known is so (or the agent only knows what is actually true). This schema also axiomatizes the class of all modal trios whose accessibility relation $R$ is reflexive. The validity of $T$ can be shown by using the truth conditions listed previously: Consider an arbitrary trio $M = \langle W, R, P \rangle$ and suppose $\models^M_w \Box \alpha$ for an arbitrary $\alpha \in W$. Then $\models^M_{w'} \alpha$ for every accessible possible world $w'$ in $M$ by (B11), in particular $w$, thus $\models^M_w \alpha$ and by (B7) $\models^M_w \Box \alpha \rightarrow \alpha$.

4. $\Box \alpha \rightarrow \Box \Box \alpha$ This schema axiomatizes the class of modal trios whose accessibility relation is transitive. Epistemically speaking, schema 4 says that an agent is capable of positive introspection: if an agent knows $\alpha$, then he knows that he knows $\alpha$.  

25
5. $\Diamond \alpha \rightarrow \Box \Diamond \alpha$ This schema says if an agent believes $\alpha$ he must know that he believes $\alpha$. Clarence Irving Lewis (1883-1964) included this axiom in a system called S5 and thus the name 5. It may also be written as: $\neg \Box \neg \alpha \rightarrow \Box \neg \Box \alpha$ due to the duality of $\Box$ and $\Diamond$. This schema represents the claim that agents are capable of negative introspection in the sense of knowing when they don't know something.

Schemas T, 4 and 5 (taken together) axiomatize the class of modal trios whose accessibility relations are equivalence relations. T, 4 and 5 do not hold true in every system of modal logic. A more widely accepted schema is;

K. $\Box (\alpha \rightarrow \beta) \rightarrow (\Box \alpha \rightarrow \Box \beta)$ If an agent knows $\alpha \rightarrow \beta$, then, if he also knows $\alpha$, he will know $\beta$. This schema was named K after Saul Kripke, one of the inventors of possible world semantics. To prove this schema suppose $w$ is a possible world in a trio $M$, in such a way that both $\models^M_{w} \Box (\alpha \rightarrow \beta)$ and $\models^M_{w} \Box \alpha$, then for every accessible possible world $w'$ in $M$, both $\models^M_{w'} \alpha \rightarrow \beta$ and $\models^M_{w'} \alpha$. Hence it follows $\models^M_{w'} \beta$ and thus $\models^M_{w'} \Box \beta$.

A modal system built on schemas K and T is referred to as system T. S4 is the modal system obtained by adding schema 4 to T. Adding schema 5 to S4 yields the system S5.

The inference rules in all these systems are obtained by adding the following inference rule to that of Modus Ponens: From $\alpha$ one can deduce $\Box \alpha$, i.e. if $\models \alpha$ then $\models \Box \alpha$. This means that if $\alpha$ is a valid sentence then $\Box \alpha$ is also valid.

In this section we have focussed on the extension of propositional logic to modal logic. The relationship between propositional logic and modal logic is that modal logic includes propositional logic in the sense that every propositionally valid sentence is modally valid. Predicate logic can, however, also be extended by the addition of modal operators. Although we shall not discuss such extensions in this dissertation, we illustrate the expressibility of such an extension. Within a modal predicate logic we may express that $\exists x \Box \alpha \rightarrow \Box \exists x \alpha$ is valid, but $\Box \exists x \alpha \rightarrow \exists x \Box \alpha$ is not.
The above mentioned approach to S5 is the one suggested by Chellas [1980]. There are, however, other approaches to S5. To view some of these the reader may consult Chagrov and Zakharyaschev [1997] or Fitting [1983].

An understanding of modal logic is particularly valuable in the formal analysis of philosophical arguments, where expressions from the modal family (including logics for belief, for tense and other temporal expressions, for the deontic (moral) expressions such as 'it is obligatory that' and 'it is permitted that', etc.) are both common and confusing. Modal logic also has important applications in computer science.

Modal logic is an important stepping stone for moving towards nonmonotonic reasoning (discussed in the next chapter) where one reasons about knowledge or belief change over time, where retraction of beliefs are possible and necessary to reach the correct conclusion.
Chapter 3

Nonmonotonic Reasoning

"Belief, then, is defeasible knowledge"
Anonymous
in Thomas Meyer's Semantic Belief Change, 1999

Nonmonotonic reasoning, in its broadest sense, is reasoning to conclusions on the basis of incomplete information. Given more information, we are prepared to retract previously drawn inferences (beliefs). Robert C. Moore [1995, p123] states that "Commonsense reasoning is 'nonmonotonic' in the sense that we often draw, on the basis of partial information, conclusions that we later retract when we are given more complete information."

An everyday example of nonmonotonic reasoning can be seen in the behaviour of motorists regarding situations whilst driving. When you are driving and nearing a traffic light – which is green for your traffic flow – you automatically assume that drivers in the perpendicular streets (for whom the traffic lights are red) will stop. Your reasoning is based on traffic rules and general common sense. If one of the motorist to either your left or right skip the red traffic light, you will have to (very quickly) retract your belief that they will stop and react to this new information (probably by doing an emergency stop or swerve around the offending driver). Nonmonotonic reasoning brings us a step closer to this type of human reasoning; and takes us a step further away from (the rigid) propositional (and predicate) logic (whilst keeping it as a foundation).
In propositional logic, if one adds more information to the antecedent of a conditional proposition and one does not change the consequent, it yields a conditional proposition which is "more true" than the original conditional proposition. To illustrate what we mean, consider the two conditional propositions \( p \rightarrow r \) and \( p \land q \rightarrow r \) consisting of three atoms \( p, q \) and \( r \). Viewing the truth table of \( p \rightarrow r \) below, it is easy to see that \( \text{Mod}(p \rightarrow r) = \{111, 101, 011, 010, 001, 000\} \):

\[
\begin{array}{ccc|c}
 p & q & r & p \rightarrow r \\
 1 & 1 & 1 & 1 \\
 1 & 1 & 0 & 0 \\
 1 & 0 & 1 & 1 \\
 1 & 0 & 0 & 0 \\
 0 & 1 & 1 & 1 \\
 0 & 1 & 0 & 1 \\
 0 & 0 & 1 & 1 \\
 0 & 0 & 0 & 1 \\
\end{array}
\]

Consider also the truth table of \( p \land q \rightarrow r \):

\[
\begin{array}{ccc|cc|c}
 p & q & r & p \land q & p \land q \rightarrow r \\
 1 & 1 & 1 & 1 & 1 \\
 1 & 1 & 0 & 1 & 0 \\
 1 & 0 & 1 & 0 & 1 \\
 1 & 0 & 0 & 0 & 1 \\
 0 & 1 & 1 & 0 & 1 \\
 0 & 1 & 0 & 0 & 1 \\
 0 & 0 & 1 & 0 & 1 \\
 0 & 0 & 0 & 0 & 1 \\
\end{array}
\]

The set of models of \( p \rightarrow r \) is contained in the set of models of \( p \land q \rightarrow r \). Thus, the conjunctional adding of atoms to the antecedent of a conditional proposition — in this example we added \( q \) — yields a conditional proposition
with a logically stronger antecedent, but with the same consequent, and a statement which is true at more worlds.

The idea of strengthening the antecedent while retaining the conclusion within a sentence may be extended to the notion of entailment (or deduction). Consider the following deduction within our $p, q$-language: From $p$ and $p \rightarrow q$ we may deduce $q$ (done according to the rule of Modus Ponens). Hence $\{p \rightarrow q, p\}$ |= $q$. Strengthening our set of premises (assumptions, axioms or pieces of information) to include $\neg q$, we obtain the set $\{p \rightarrow q, p, \neg q\}$ of premises which is logically equivalent to $\bot$. Therefore we may deduce any sentence from $\{p \rightarrow q, p, \neg q\}$ - including $q$.

Summarizing, adding information in the deduction process can only expand (and never shrink) a set of inferences. This property of classical deduction is known as monotonicity. Formally: If $\alpha |= \gamma$, then $\alpha \land \beta |= \gamma$ (or $\{\alpha, \beta\} |= \gamma$).

A logical system is called nonmonotonic if its provability relation violates the property of monotonicity [Lukaszewicz 1990].

So, the difference between monotonic and nonmonotonic reasoning is that, in the latter, when information is added to the initial pool of (incomplete) information, the outcome of the deduction might change, in the sense that one might need to retract one’s initial deduction and conclude something quite different.

A very famous example of nonmonotonicity involves a bird with the name ‘Tweety’. Assume the agent knows that ‘Tweety is a bird’. Given this definite information (knowledge) and the belief (assumption) that ‘birds (normally) fly’ the agent concludes that ‘Tweety can fly’. But after a while the agent learns something more about Tweety, namely that ‘Tweety has clipped wings’. Consequently the agent will have to retract his initial conclusion that ‘Tweety can fly’, and should now deduce ‘Tweety is a bird that cannot fly’ [Ginsberg 1994].

The question is this: How can we go beyond our definitive (incomplete) knowledge, to derive what is plausible without making blind guesses? Moreover, how do we formalize such a reasoning process? In what follows we consider a few attempts to provide answers to the above, but first we describe a
few of the relevant notions with respect to nonmonotonic reasoning.

### 3.1 Basic Notions, Principles and Topics regarding Nonmonotonic Reasoning

One can consider all the information of an agent to be grouped together in a database or, as Thomas Meyer [1999] views this grouped information in general, as the *epistemic state* of the agent – the knowledge and beliefs of the agent as well as all the information needed for coherent reasoning. We will encounter different representations of an epistemic state in what follows. For example, an epistemic state may consist of a (deductively closed) set of sentences or a default pair (i.e. a pair containing a set of axioms, together with a set of default rules). In the above mentioned example the sentences ‘Tweety is a bird’ and ‘birds (normally) fly’ will be in the epistemic state and consequently the conclusion is that ‘Tweety can fly’. The latter will be retracted upon the additional information that ‘Tweety has clipped wings’. The new conclusion in the epistemic state will be ‘Tweety is a bird that cannot fly’.

Normally an epistemic state is consistent, although one might choose not to retain consistency in a epistemic state (in certain extraordinary circumstances one might prefer an inconsistent epistemic state, leaving no room for arguing a case, since then the epistemic state implicitly contains all information). Consistency, or the lack thereof, will influence the way information is added or removed from an epistemic state. Ginsberg [1994] defines *input* as the addition of data to the epistemic state regardless of the effect it might have on the consistency of the epistemic state, while *updating* an epistemic state adds information, but focuses on keeping the epistemic state consistent. Sometimes priorities can be given to information in the epistemic state. The priority of a sentence (piece of information) will play a role when adding or removing information. Information with a higher priority would be kept during changes, rather than information with a lower priority. (We will discuss the prioritizing of information later, in conjunction with Meyer's [1999] epistemic entrenchment ordering.)
Humans have the ability to naturally exclude the most bizarre notions from their trail of thought. This ability to exclude needs to be considered in the creation of an epistemic state of an agent. For example, if an agent is reasoning about road safety and whether other drivers will violate the traffic regulations, he cannot reasonably contemplate the idea of an aeroplane making an emergency landing on the road he is travelling on. An epistemic state needs to hold viable possibilities without wasting space on bizarre notions. The closed world assumption is another notion pertaining to nonmonotonic reasoning. Ginsberg [1994] discussed the closed world assumption with the help of an example by Reiter. Considering a travel agent’s database, one assumes that if there is no information concerning a flight from Hawaii to Haiti, then no such flights exists. The travel agent’s database contains only information about existing flights, or else the database would have to indicate explicitly, for each pair of cities and possible departure times, whether or not there is a flight between them at the given time. Keeping information in this manner will result in an unmanageable and large database. Thus, the closed world assumption forces a database to contain only the necessary information, resulting in a more manageable database.

An agent can also derive the logically implicit from the information given in the epistemic state [Ginsberg 1994]. As an example, consider the sentence ‘the ball lies on the floor’, implying that ‘the floor is beneath the ball’. This idea ties in with storing the logically strongest information. Say the agent knows sentence $p$ and sentence $q$, he would choose to store this information in the logically strongest form, namely $p \land q$. For a more concrete example, the agent will choose to store information like ‘the ball is blue and round’ instead of ‘the ball is blue’ and ‘the ball is round’. Thinking in terms of computer databases (with a constant restriction on space) it makes sense to store information in the logically strongest form. One can think of a hard drive as a piece of paper where the sentence ‘the ball is blue and round’ consumes much less physical space than the sentences ‘the ball is blue’ and ‘the ball is round’.

There also are some basic principles to keep the reasoning process from falling into endless loops or infinite (even ridiculous) reasoning. The qualify-
cation problem [McCarthy 1977] refers to a problem some situations present. The problem is that the agent may not be sure whether to take a risk and react on the incomplete information available. For example, consider the testing of a power supply. To test whether it is working or not, one would try to switch it on. It might work or it might cause a short, but one cannot try to prove that everything pertaining to the switch is in working order – before switching it on. One should take the leap, switch on the power supply and deal with the consequences (if there are any). Thus the agent should conclude or try something, without considering all possible problems that might arise, and without trying to prove that no such problems exist.

The frame axiom states that things do not change mysteriously when actions are performed [Ginsberg 1994]. This means that the house will not collapse on you if you close the door – given normal circumstances prevail (a risk we take every day). We assume actions change only what they are supposed to change: the door will merely block the specific entrance while it is closed.

In summation, the following rules need to be regarded when handling an epistemic state:

- Regard the epistemic state as a closed world – don’t include unnecessary information.
- Derive the logically implicit from the epistemic state, storing the logically strongest.
- Keep the qualification problem in mind (when is it acceptable to risk a reaction based on incomplete information).
- Build assumptions on the frame axiom which states that things do not change mysteriously.

Before we consider a few formalizations, we first provide a general overview of modal nonmonotonic logic. For the purpose of this dissertation, we restrict the discussion of modal systems built on a propositional language. A modal nonmonotonic logic is defined by adding one of the letters M and L (or both) to the language. If $\alpha$ is an arbitrary proposition, $M\alpha$ and $L\alpha$ are formally handled as atomic sentences in the modal language.
Usually only one letter is defined — either M or L — and added to the language, whilst the other takes on a dual description, for example Lα equates to \neg M\neg \alpha. The interpretation of M and L depends on the circumstances in which one wants to use it. M\alpha may be read as 'it is consistent to believe \alpha', '\alpha is possible' or '\alpha may be believed' (in the absence of definite proof). On the other hand, one can interpret L\alpha as '\alpha is nonmonotonic provable', '\alpha is known' or (depending on the context) '\alpha is believed' (since a certain degree of proof exists). Do note that there is a difference between '\alpha is believed' and 'it is consistent to believe \alpha': The former indicates that the agents holds \alpha as a belief without any doubt — it is to 'know' \alpha, whilst the latter implies that the agent may accept \alpha as true, since there is no proof of the opposite. One can see the link with the modal operators ◊ and □, where M\alpha may be compared with ◊\alpha and L\alpha with □\alpha.

A theory is any set of sentences of the language. Consider now a theory T containing the set of initial premises of an agent. The central concept of modal nonmonotonic logic is that of an extension of a theory T. An extension is the total set of beliefs one may hold, given the set T of initial premises. There are three conditions to be expected from an extension S of T:

- S should be deductively closed, i.e. S must contain all its consequences.
- S ought to include the set of initial premises, that is T \subseteq S.
- S should contain any sentence of the form M\alpha, provided that \neg \alpha is not a consequence of S.

The last condition may be intuitively read as 'in the absence of evidence that \neg \alpha is the case, it is consistent to believe \alpha'.

An extension S of T is therefore defined as the deductive closure of the set, consisting of the union of T with the set of sentences of the form M\alpha given that \alpha is a sentence of the language of T and \neg \alpha \notin S.

D. McDermott and J. Doyle [1980], as explicated in Łukaszewicz [1990], defined a modal nonmonotonic logic by adding the letter M to their language. McDermott and Doyle choose to interpret M\alpha as 'it is consistent to believe \alpha'. L\alpha read '\alpha is nonmonotonic provable' or '\alpha is believed'.

34
A weakness in the system of McDermott and Doyle manifests itself within (amongst others) the following example of the \( p, q \)-language:

Let \( T = \{ M(p \land q), \neg p \} \). It is intuitively difficult to simultaneously accept that \( p \) is false and \( p \land q \) can be consistently believed. However, in McDermott and Doyle’s logic \( T \) is consistent. The theory has one consistent extension including \( M(p \land q) \), but not \( Mp \).

Due to weaknesses – like the above – in the system of McDermott and Doyle, McDermott defined a stronger modal nonmonotonic logic in which he retained a similar description of an extension, but formalized the new system in terms of one of the modal deductive systems described in Chapter 2, namely \( T \), \( S4 \) or \( S5 \). The reader can find more details on the formalization in McDermott and Doyle [1980] and McDermott [1982].

Robert Moore [1995, p125] said “McDermott and Doyle seem to confuse two quite distinct forms of nonmonotonic reasoning, which we will call default reasoning and autoepistemic reasoning. They talk as though their systems were intended to model the former, but they actually seem much better suited to modelling the latter.”

Autoepistemic reasoning embraces reasoning about one’s own knowledge and beliefs. Moore created autoepistemic logic using the same language as McDermott and Doyle as a basis, but choosing to employ \( L \) rather than \( M \). \( L\alpha \) is interpreted as ‘\( \alpha \) is believed’ while \( M\alpha \) reads (as previously) ‘it is consistent to believe \( \alpha \)’. Moore provided both a semantically and syntactical description of an extension of a belief set. Intuitively, such a set is to be viewed as the total collection of beliefs of an agent reasoning about his own beliefs. Those readers who wish to read more about Moore’s criticism towards McDermott and Doyle’s logic and the creation and formalization of autoepistemic reasoning are referred to Moore [1995].

Reiter [1980] initiated default logic. Instead of using rules expressed as formulas of the object language – like in the modal nonmonotonic logics – default logic is represented by special linguistic expressions, referred to as defaults. Default logic is represented by a default pair consisting of a set of axioms (the fixed knowledge) and a set of defaults. The axioms represent log-
ically valid, but generally incomplete information, while the defaults extend this information by sanctioning plausible, although not necessarily true, conclusions. As an example, consider the default pair consisting of the axiom ‘x is a Bird’, portrayed by Bird(x) and the permitted belief ‘Birds Can Fly’, portrayed as CanFly(x), leading to the conclusion that ‘x can fly’ (CanFly(x)). Lukaszewicz [1990] uses the notation

\[ A(x) : B_1, ..., B_n \]

\[ \overline{C(x)} \]

where \( A(x) \) is the prerequisite (which is derivable from the axioms), that is Bird(x) in the above example, while \( B_1, ..., B_n \) represent the justifications (in our example there is only one justification namely CanFly(x)) and lastly \( C(x) \) portrays the consequent of the default, that is CanFly(x) in the example. Thus for the above example the notation will yield;

\[ Bird(x) : CanFly(x) \]

\[ \overline{CanFly(x)} \]

When one receives more information, for example that x is a Penguin (formally expressed as Penguin(x)), one is forced to accept that it is not the case that ‘x can fly’. For a more elaborate discussion on default logic, we refer the reader to Lukaszewicz [1990].

Another topic regarding nonmonotonic reasoning is that of belief change. Thomas Meyer [1999, p1] summarizes belief change as follows: “...a rational agent is sometimes forced to adjust its current beliefs in some appropriate fashion when confronted with new information.” Belief change may (for example) involve the removal of a belief from a set of beliefs, or the expansion of such a set by adding a belief. Some of these processes will be discussed shortly.

First we consider the basic principles on which Meyer [1999] builds his belief change:

- **Minimal Change**: Keep loss and addition to a minimum. Any changes made to the existing epistemic state ought to be only those that have to be made.
• **Conservatism**: Keep the set of beliefs as large as possible. It is seen as ideal to keep the set of beliefs as large as possible and not to lose any information unnecessarily during change.

• **Indifference**: Objects held in equal regard should be treated equally. If one believes equally strong in two different beliefs, none of them should be regarded inferior to the other when a belief change occurs. One should not be willing to change the one belief and leave the other unchanged, since they are regarded as equally important.

• **Preference**: Objects held in higher regard should be afforded a more favorable treatment. When a belief change occurs, one should be willing to give up beliefs less valuable, opposed to those more prized (these are the beliefs that one would be least willing to let go).

The principles of *Indifference* and *Preference* will further be highlighted when we discuss 'epistemic entrenchment' in Section 3.4.

### 3.2 AGM Belief Change

A major development in belief change took place in the late 1970's and early 1980's, namely the AGM approach to belief change. The naming AGM is due to the three developers of the approach, namely Carlos Alchourón, Peter Gärdenfors and David Makinson [Pagnucco and Rott 1999]. This approach is accepted widely and forms the basis for most current research in belief change. The AGM approach considers an agent's belief set as a *deductively closed set of sentences* and explicated three primary types of operations regarding belief change:

- **Removal**: Information is removed from the current set of beliefs of an agent.

- **Revision**: New information is incorporated into the current set of beliefs of an agent in a way that ensures consistency.
• **Expansion**: New information is simply added to the agent’s current set of beliefs to form a new belief set.

Due to the number of rational ways to perform belief change, AGM does not specify a unique definition for revision or removal. However, they do provide a few postulates to guide the removal and revision operations. To provide a general idea of the process we will include some of these revision and removal postulates shortly.

In order to understand the postulates, an understanding of some of the symbols used, is needed. Let $K$ be a fixed belief set contained in the formal object language $\mathcal{L}$, and take $\alpha$ to be the belief which must be removed from or revised to $K$. These processes are respectively denoted by $\alpha$-removal and $\alpha$-revision. $Cn(A)$ consist of all the beliefs that follow logically from the theory (set of sentences) $A$ and is formally defined as $Cn(A) = \{ \alpha \mid A \models \alpha \}$. If $W$ is the set of valuations of $\mathcal{L}$ and $U \subseteq W$ then one can describe the theory determined by $U$ as $Th(U) = \{ \alpha \in L \mid U \subseteq Mod(\alpha) \}$.

The postulates for $\alpha$-removal, as discussed by Meyer [1999], are:

(K-1) $K - \alpha = Cn(K - \alpha)$

(K-2) $K - \alpha \subseteq K$

(K-3) If $\alpha \notin K$ then $K - \alpha = K$

(K-4) If $\not\models \alpha$ then $\alpha \notin K - \alpha$

(K-5) If $\alpha \equiv \beta$ then $K - \alpha = K - \beta$

(K-6) If $\alpha \in K$ then $(K - \alpha) + \alpha = K$

(K-7) $(K - \alpha) \cap (K - \beta) \subseteq K - (\alpha \land \beta)$

(K-8) If $\beta \notin K - (\alpha \land \beta)$ then $K - (\alpha \land \beta) \subseteq K - \beta$

If a removal satisfies the above mentioned postulates (K-1) to (K-6) then it is a basic AGM contraction. If it only satisfies (K-1) to (K-5) then it is called a withdrawal and if it satisfies (K-1) to (K-8) then it is a AGM contraction. The
meaning of the postulates are as follows. (K-1) is the requirement that AGM contractions yield belief sets. (K-2) ensures that the contraction of a belief set results in a contracted belief set, that is the set need still be contained in (at best, equal to) the set prior to the contraction. (K-3) appeals to the Principle of Information Economy, i.e. the principle which states that if a belief is chosen to be removed that was not in the set in the first place, nothing would change. (K-4) ensures that contraction by any non-valid proposition \( \alpha \), is successful (i.e. should not contain \( \alpha \) again). (K-5) just stipulates that the syntax is irrelevant and (K-6) is also known as the Postulate of Recovery and indicates that when a belief set \( K \) is contracted by proposition \( \alpha \), that \( K \) can be reconstructed by a simple \( \alpha \)-expansion of \( K - \alpha \) with \( \alpha \).

For the sake of simplicity of illustration we adapt definitions to the finite case. In such a finitely generated language the belief set \( K \) may be represented by a single sentence. For example, in our \( p, q \)-language we refer the reader to figure 1.1. The belief set \( K = \{ p, p \lor q, q \rightarrow p, \top \} \) consist of one of the parallelograms circumscribed by bold lines. At the bottom of this parallelogram is the sentence \( p \) – the logically strongest sentence in the belief set and the sentence which may be taken as the representative of the belief set. To simplify matters, we may write \( K = p \) (i.e. we use ‘\( = \)’ rather loosely when it comes to a belief set and the sentence logically equivalent to the set). We use the example of a heater-fan scenario to illustrate some of the relevant notions: Suppose an agent is looking at a building from a distance. Due to experiments taking place in the building, it is important to note the state of the heater and the fan in the building. The agent believes that the heater is on (because the heater is basically never off), but he has to “guess” about the status of the fan – whether it is on or off. Let \( p \) represents the sentence ‘the heater is on’ and \( q \) represents ‘the fan is on’. There are four possible combinations, namely:

- The heater is on and the fan is on. This is represented by the world 11.
- The heater is on and the fan is off, represented by 10.
- The heater is off and the fan is on, represented by 01.
- Both the heater and the fan is off, represented by 00.
Meyer [1999] defines a \textit{K-faithful} order \( \preceq \) on the set \( W \) of all possible worlds as any \textit{total preorder} \( \preceq \) which has the properties that (i) \( v \prec w \) for every \( v \in K \) and \( w \not\in K \) and (ii) \( v \not\prec w \) for every \( v, w \in K \). (A total preorder on a set \( X \) is a reflexive and transitive binary relation that is also \textit{connected} (a binary relation \( R \) on any set \( X \) is connected if and only if \( xRx \) or \( yRx \) for every \( x, y \in X \)).) Thus in any \( K \)-faithful order the models of the belief set \( K \) are situated in the bottom class. Let us explain the definition via the heater-fan example. Suppose the agent believes the heater is on, thus he believes his choices of the above four possibilities are restricted to 11 and 10. The agent also believes – given previous experiences – that both the heater and fan being off would most rarely be the case. Remember that the epistemic state of an agent has to be represented in a way that, at the very least, ensures the extraction of the beliefs of the agent, together with the information needed to perform reasoning in a coherent fashion. Meyer [1999] finds it sufficient to semantically represent an epistemic state as an ordered pair \((K, \preceq)\), where \( K \) is a belief set and \( \preceq \) is a \( K \)-faithful total preorder. (Note that Meyer includes the faithful preorder in his discussion of an epistemic state whereas in the general AGM approach, an epistemic state consists of only a belief set.) Regarding the example above, the epistemic state \((K, \preceq)\) is represented as:

\[
\begin{array}{cccc}
00 & 01 & 11 & 10 \\
\end{array}
\]

Meyer [1999] prefers to describe removal and revision in terms of minimal models. An element \( u \in W \) is \textit{minimal} in \( U \subseteq W \) with respect to \( \preceq \) if and only if \( u \in U \) and there is no \( v \in U \) such that \( v \prec u \). The \( \preceq \)-minimal elements of \( \text{Mod}(\alpha) \) is denoted by \( \text{Min}_{\preceq}(\alpha) \). In the above \( K \)-faithful ordering \( \text{Min}_{\preceq}(\neg p) \) is \{01\}.

One can define removal in terms of a \( K \)-faithful preorder \( \preceq \) by:

\[
K - \alpha = \text{Th}(\text{Mod}(K) \cup \text{Min}_{\preceq}(\neg \alpha)).
\]

Let us consider an example with \( K = p \), \( \alpha = p \lor q \) and \( \neg \alpha = \neg p \land \neg q \), based
on the faithful preorder as defined above, namely:

| 00 | 01 | 11 | 10 |

Removal of $\alpha$ in terms of the above $K$-faithful preorder yields the following:

$$K - \alpha = Th(\text{Mod}(K) \cup \text{Min}_{\leq}(-\alpha))$$

$$= Th(\text{Mod}(K) \cup \text{Min}_{\leq}(-p \land -q))$$

$$= Th(\{11, 10\} \cup \{00\})$$

$$= q \rightarrow p$$

(We shall show – after the discussion on $\alpha$-revision – how removal and revision are interdefinable.)

The postulates for revision, as explicated in Heidema et al. [2000], are:

(K*1) $K * \alpha = Cn(K * \alpha)$

(K*2) $K * \alpha \subseteq K + \alpha$

(K*3) If $-\alpha \notin K$ then $K * \alpha = K + \alpha$

(K*4) $\alpha \in K * \alpha$

(K*5) If $\alpha \equiv \beta$ then $K * \alpha = K * \beta$

(K*6) $\bot \in K * \alpha$ if and only if $\models -\alpha$

(K*7) $K * (\alpha \land \beta) \subseteq (K * \alpha) + \beta$

(K*8) If $-\beta \notin K * \alpha$ then $(K * \alpha) + \beta \subseteq K * (\alpha \land \beta)$

If a revision satisfies (K*1) to (K*6) then it is a basic AGM revision, if it satisfies (K*1) to (K*8) then it is a ACM revision [Meyer 1999]. Postulate (K*1) is a requirement that revision of a belief set with a proposition $\alpha$ will result in a belief set again. (K*2) places an appropriate upper bound on the belief set obtained from revision. Plainly put, (K*2) makes sure that a revision is still within bounds of an expansion of the same information. (K*3) invokes the Principle of Minimal Change for the case where the proposition with which
to revise is consistent with the current belief set. (K*4) ensures that revision is always successful (i.e. contains the proposition with which the belief set was revised). (K*5) highlights that syntax is irrelevant. (K*6) explains that one will find a belief set to be inconsistent only if one revises the belief set with a belief of which its negation is regarded as tautologous.

Revision is also definable in terms of a $K$-faithful preorder $\preceq$ by

$$K \ast \alpha = Th(Min_{\preceq}(\alpha)).$$

Let us consider the same example as was used to illustrate removal, i.e. $K = p$, $\alpha = p \lor q$ and $\neg \alpha = \neg p \land \neg q$, and $\preceq$ is the following $K$-faithful preorder:

\begin{tabular}{|c|c|}
\hline
00 & 01 \\
\hline
11 & 10 \\
\hline
\end{tabular}

Then,

\begin{align*}
K \ast \alpha &= Th(Min_{\preceq}(\alpha)) \\
&= Th(Min_{\preceq}(p \lor q)) \\
&= Th(Min_{\preceq}([11, 10, 01])) \\
&= Th([11, 10]) \\
&= p
\end{align*}

We shall now consider the interdefinability of removal and revision by considering the Levi and Harper Identities.

When comparing the AGM postulates of removal and revision there seems some structural similarities in the postulates. These similarities are used as a basis to interdefine removal and revision which are structurally similar, but different in principle.

Levi defined revision in terms of removal with the following identity as explicated in Gärdenfors and Rott [1995]

$$K \ast \alpha = (K - \neg \alpha) + \alpha,$$

whilst Harper defined removal in terms of revision by

$$K - \alpha = (K \ast \neg \alpha) \cap K.$$
Sticking to the simple example used previously, let $K = p$, $\alpha = p \lor q$ and $\neg \alpha = \neg p \land \neg q$, with $\leq$ the following $K$-faithful preorder:

<table>
<thead>
<tr>
<th>00</th>
<th>01</th>
<th>10</th>
</tr>
</thead>
</table>

Now one can examine the result of revision and removal with the help of Levi and Harper’s identities:

\[
K \star \alpha = (K - \neg \alpha) + \alpha \\
= Th(Mod(K) \cup Min_{\leq}(\alpha)) + \alpha \\
= Th\{\{11, 10\} \cup \{11, 10\}\} + \{p \lor q\} \\
= \{p\} + \{p \lor q\} \\
= \{p, p \lor q\} \\
= p
\]

It is clear that the Levi identity yields the same answer as the definition of revision in terms of the $K$-faithful preorder namely $K \star \alpha = Th(Min_{\leq}(\alpha))$.

In the finite case the Harper identity boils down to $K - \alpha = (K \star \neg \alpha) \lor K$ where $K$ is now seen as a single sentence.

\[
K - \alpha = (K \star \neg \alpha) \lor K \\
= Th(Min_{\leq}(\neg \alpha)) \lor K \\
= Th(Min_{\leq}(-p \land \neg q)) \lor K \\
= Th\{\{00\}\} \lor K \\
= (-p \land \neg q) \lor p \\
= q \rightarrow p
\]

Again, the Harper identity yields the same answer as the definition for removal in terms of the $K$-faithful preorder, $K - \alpha = Th(Mod(K) \cup Min_{\leq}(\neg \alpha))$.

We will now consider the KLM system, which strives to create a comparison platform over the various nonmonotonic reasoning processes.
3.3 KLM Nonmonotonic Reasoning

Due to the various types of nonmonotonic reasoning around and despite the fact that each of these systems has something interesting to offer (for example McDermotte and Doyle's modal systems, Moore's autoepistemic logic and Reiter's default logic), there was a lack of general framework that made comparisons and evaluations between these systems rather difficult. The group of people whom succeeded best in generalizing the nonmonotonic setting was Sarit Kraus, Daniel Lehmann and Menachem Magidor. The KLM approach is named after Kraus, Lehmann and Magidor.

Kraus et al. established a binary relation on $\mathcal{L}$ that can be seen as a nonmonotonic consequence relation denoted by $\models \sim$. For propositions $\alpha, \beta$ of $\mathcal{L}$, $\alpha \models \sim \beta$ should be read as ' $\beta$ is a plausible consequence of $\alpha$' or 'if $\alpha$ holds true then the agent is willing to (defeasibly) jump to the conclusion that $\beta$ holds', thus $\alpha$ is the available evidence from which the plausible $\beta$ is drawn.

The nonmonotonic consequence relation is a binary relation on $\mathcal{L}$ that a rational agent uses in reasoning. Because of the rationality of the agent, the nonmonotonic consequence relation must possess certain properties to comply with common sense reasoning. These properties are stipulated by the following postulates [Gärdenfors and Rott 1995]:

- **Reflexivity**: For every $\alpha \in \mathcal{L}$, $\alpha \models \sim \alpha$. Reflexivity ensures that $\alpha$ is a plausible consequence of $\alpha$.

- **Right Weakening**: If $\beta \models \gamma$ and $\alpha \models \sim \beta$ then $\alpha \models \sim \gamma$. Thus, $\gamma$, being logically weaker than $\beta$, must be a plausible consequence of $\alpha$, if $\beta$ is a plausible consequence of $\alpha$.

- **Left Logical Equivalence**: If $\alpha \equiv \beta$ and $\alpha \models \sim \gamma$ then $\beta \models \sim \gamma$. If data are logically equivalent (but possibly different), they should yield the same plausible consequences.

- **And**: If $\alpha \models \sim \beta$ and $\alpha \models \sim \gamma$ then $\alpha \models \sim \beta \wedge \gamma$. If $\alpha$ yields $\beta$ and $\gamma$ respectively as plausible consequences, it should yield the conjunction of $\beta$ and $\gamma$ as a plausible consequence.
• Or: If $\alpha \models \sim \gamma$ and $\beta \models \sim \gamma$ then $\alpha \lor \beta \models \sim \gamma$. If two different pieces of data yield the same plausible consequence respectively, a rational agent would suspect that their disjunction would yield the same plausible consequence again.

• Cautious Monotonicity: If $\alpha \models \sim \beta$ and $\alpha \models \sim \gamma$ then $\alpha \land \beta \models \sim \gamma$. As we know from nonmonotonicity, one cannot add any piece of information in conjunction to the initial information and be certain that the consequence would stay the same. Cautious monotonicity, however, ensures that as long as one adds information ($\beta$) which is a plausible consequence of the original information ($\alpha$) then this conjunction will still yield the same plausible consequence ($\gamma$).

• Cut: If $\alpha \land \beta \models \sim \gamma$ and $\alpha \models \sim \beta$ then $\alpha \models \sim \gamma$. An agent cannot say, when confronted with a conjunction of $\alpha$ and $\beta$, whether any of the pieces of data alone would yield the same consequence, namely $\gamma$, as they do in their conjuncted form. But if the additional data shows that one of the pieces of information is a plausible consequence of the other ($\alpha \models \sim \beta$), then the agent can assume that $\alpha \models \sim \gamma$.

• Conditionalization: If $\alpha \land \beta \models \sim \gamma$ then $\alpha \models \sim \beta \rightarrow \gamma$

• Supraclasiclality: If $\alpha \models \beta$ then $\alpha \models \sim \beta$. If $\beta$ is logically weaker than $\alpha$, then $\beta$ must be a plausible consequence from $\alpha$. (In other words, defeasible deduction is an extension of normal deduction.)

KLM combines the AGM method (theory revision) with nonmonotonic reasoning. It takes a belief set $K$ and a $\alpha$-revision and then defines a nonmonotonic consequence relation (|~): the nonmonotonic consequence relation $|~$ is defined by letting the set of plausible consequences of $\alpha$ coincide with the new belief set obtained from an $\alpha$-revision of $K$. On the other hand, a nonmonotonic consequence relation $|\sim$ together with an appropriate belief set $K$, induce an $\alpha$-revision of $K$: the resulting belief set $K \ast \alpha$ is equal to the set of plausible consequences of $\alpha$ [Meyer 1999].
Thus, formally KLM defines the consequence relation in terms of revision by
\[ \alpha \models \beta \text{ if and only if } \beta \in K \ast \alpha, \]
and revision in terms of the consequence relation by
\[ K \ast \alpha = \{ \beta | \alpha \models \beta \}. \]

With the help of Harper's definition and the definition of the consequence relation in terms of the \( K \)-faithful preorder, namely \( \alpha \models \beta \) if and only if \( \text{Min}_K(\alpha) \subseteq \text{Mod}(\beta) \), one can also express removal in terms of the consequence relation and vice versa.

For a broader understanding regarding the links between KLM theory change and the AGM approach we refer the reader to Meyer [1999].

Being more familiar with removal and revision in various forms, we would like to introduce the last associated section regarding belief change, namely that of epistemic entrenchment. We will also show how epistemic entrenchments and \( K \)-faithful preorders are related.

### 3.4 Epistemic Entrenchment

Since there are beliefs in an agent's set \( K \) of beliefs that he would prefer to keep, rather than others, it is sensible to consider an ordering on the beliefs of an agent. The agent would prefer to keep the beliefs which are more entrenched when confronted with changing his belief set. Thus, belief change will be done in terms of an entrenchment ordering on all sentences of the language.

Peter Gärdenfors and David Makinson [1988] compiled the best known version of *epistemic entrenchment orderings* (EE-orderings) where \( \alpha \sqsubseteq_{EE} \beta \) portrays \( \beta \) being more (or equally) entrenched than \( \alpha \) (i.e. higher up in the ordering \( \sqsubseteq_{EE} \)). The ordering \( \sqsubseteq_{EE} \) is an ordering on (the equivalent classes of) the sentences of the language \( \mathcal{L} \) which induces partitions on \( \mathcal{L} \) with a linear ordering on the partition classes.
Formally, the postulates for Gärdenfors and Makinson's [1988] epistemic entrenchment ordering are as follows:

**(EE1)** $\subseteq_{EE}$ is transitive.

**(EE2)** If $\alpha \vdash \beta$ then $\alpha \subseteq_{EE} \beta$.

**(EE3)** For all $\alpha, \beta \in K$, $\alpha \subseteq_{EE} \alpha \land \beta$ or $\beta \subseteq_{EE} \alpha \land \beta$.

**(EE4)** If $K \neq Cn(\bot)$ then $\alpha \notin K$ if and only if $\alpha \subseteq_{EE} \beta$ for all $\beta$.

**(EE5)** If $\alpha \subseteq_{EE} \beta$ for all $\beta$ then $\vdash \beta$.

A binary relation $\subseteq_{EE}$ is an epistemic entrenchment ordering with respect to a belief set $K$ if and only if it satisfies (EE1) to (EE5).

(EE1) is a reasonable condition to require from a ordering relation – if belief $\alpha$ is less entrenched than belief $\beta$ and belief $\beta$ is less entrenched than belief $\gamma$, then belief $\alpha$ is less entrenched than belief $\gamma$. Formally: If $\alpha \subseteq_{EE} \beta$ and $\beta \subseteq_{EE} \gamma$ then $\alpha \subseteq_{EE} \gamma$. (EE2) requires that logically weaker propositions be more entrenched than logically stronger propositions. This does sound strange, but thinking about it, one realizes that it is impossible to remove a proposition from a belief set without removing all the logically stronger propositions as well (or else the proposition will be logically deductable again). (EE3) is the cornerstone of the controversial property that every EE-ordering is a total preorder. (EE4) is a minimality condition declaring that all the propositions not in $K$ are equally entrenched, but less entrenched than the propositions in $K$ and (EE5) is a maximality condition, stating that logically valid propositions are equally entrenched, but more entrenched than all other propositions.

In this section it is necessary to distinguish between the notations for a belief set (deductively closed set of beliefs) and the notation for the single sentence which represents the whole belief set. We shall write $K$ to indicate the whole belief set and $K$ to denote the single sentence that represents the belief set $K$. With referral to the Lindenbaum-Tarski algebra if $K = p$, $K = \{p, p \lor q, p \lor \neg q, \top\}$ – the set of sentences (including $K$) lying above $K = p$ in figure 1.1. For our example, let $B$ portray the set of all sentences of the
p, q-language. We use the notation $B - K$ to indicate the compliment of $K$ in $B$ (i.e. all the sentences in the chosen language without the set of sentences $K$). The set $B - K$ is the set (strictly) beneath the proposition $K$ in the Lindenbaum-Tarski algebra. $B - K$ is the bottom class in $\cap_{EE}$ according to (EE4) and $\top$ (class of tautologies) is the highest class in $\subseteq_{EE}$.

To reach an epistemic entrenchment ordering, one needs to partition $K - \{\top\}$ into classes such that these classes will be situated between the class $B - K$ at the bottom and $\top$ at the top. From (EE2) we deduce that the proposition $K$ must be contained in the partition class second from the bottom (just above $B - K$). Every partition class $P$ must be a convex set, i.e. for every $x, y, z \in B$, if $x, z \in P$ and $x \models y$ and $y \models z$ then $y \in P$: If $x \models y$ then $x \subseteq_{EE} y$ and if $y \models z$ then $y \subseteq_{EE} z$, thus $x \subseteq_{EE} z$ from (EE1). But $x, z \in P$ thus $z \subseteq_{EE} x$, and hence, $y \subseteq_{EE} x$. Thus $x$ and $y$ (and $z$) must be in the same partition class.

Heidema (in an unpublished report) describes an algorithm for obtaining an epistemic entrenchment ordering for a given belief set $K$. The algorithm is built on the partitioning of infatoms. Meyer [1999] defines infatoms (also known as content elements) as the basic units of an epistemic state. Content elements are propositions of $L$ and the basic independent pieces from which the beliefs of an agent are built up. In the $p, q$-language the content elements are \{p \lor q, p \lor \neg q, \neg p \lor q, \neg p \lor \neg q\}. The more content elements (in the belief set), the more information in the set of beliefs. The set of all content elements contains too much information – in such a case an agent will include all the propositions in his belief set, i.e. the belief set contains contradictory information. Every content element represents a ‘world being excluded’, for instance $p \lor q \equiv \neg (\neg p \land \neg q)$ which may be read as not the world 00. The content elements of $p$, for example, are $p \lor q$ and $p \lor \neg q$.

It can be shown that each partition class between $B - K$ and $\top$ contains at least one content element of $K$, implying that an epistemic entrenchment induced by $K$ cannot have more partition classes than there are content elements of $K$. 

48
Application of Heidema’s algorithm, yields the following example of an epistemic entrenchment ordering on a belief set $K = p$:

<table>
<thead>
<tr>
<th>$\top$</th>
<th>$p \lor q$</th>
<th>$p, p \lor \neg q$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$B = {p, p \lor q, p \lor \neg q, \top}$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Propositions lower down in the entrenchment are the ones the agent is less attached to and more willing to let go. In this example, each class between $B - K$ and $\top$ contains one content element of $p$.

Let us look at how belief change take place with an EE-ordering in place. An AGM contraction of proposition $\alpha$ from a belief set $K$ (or $K$, when seen as a single sentence) under an EE-ordering can be defined as follows:

$$K - \alpha = \begin{cases} K \cap \{\beta | \alpha \sqsubseteq_{EE} \alpha \lor \beta\} & \text{if } \alpha \in K, \text{ and } \not\models \alpha \\ K & \text{otherwise} \end{cases}$$

Let us consider an example. Let $K = p$ (as used above) and $\alpha = p \lor q$. Thus, $K = \{p, p \lor q, p \lor \neg q, \top\}$, $\alpha \in K$ and $\not\models \alpha$.

Then,

$$K - \alpha = K \cap \{\beta | \alpha \sqsubseteq_{EE} \alpha \lor \beta\}$$

$$K - (p \lor q) = K \cap \{\beta | (p \lor q) \sqsubseteq_{EE} (p \lor q) \lor \beta\}$$

$$= \{p, p \lor q, p \lor \neg q, \top\} \cap \{p \lor \neg q, p \leftrightarrow q, \neg q, p \land \neg q, \top, \neg p \lor q, \neg p \lor \neg q, \neg p\}$$

$$= \{p \lor \neg q, \top\}$$

which is logically equivalent to the single sentence $p \lor \neg q$ (i.e. $q \rightarrow p$). Thus, for this particular example contraction via an EE-ordering yields the same result as our previously discussed contraction via the definition of contraction (see Section 3.2). With the help of the Levi identity one can also express revision in terms of an EE-ordering.

As promised, we will now look at the connection between $K$-faithful preorders and epistemic entrenchments. In an epistemic entrenchment ordering...
propositions lower down are less entrenched and should be given up more easily, while in a \( K \)-faithful preorder the models of \( K \) are situated at the bottom of the ordering, representing the belief set of an agent.

Considering the content elements in an epistemic entrenchment, one needs to find the corresponding ‘world excluded by the content element’. By portraying this world for every corresponding content element, one would conduct a \( K \)-faithful preorder. Consider again the following epistemic entrenchment on a belief set \( K = p: \)

\[
\begin{array}{c}
T \\
p \lor q \\
p, p \lor \neg q \\
B = \{p, p \lor q, p \lor \neg q, T\}
\end{array}
\]

The ‘excluded world’ corresponding to \( p \lor q \) is 00, while \( p \lor \neg q \equiv \neg(\neg p \land q) \) excludes the world 01. The bottom class contains the other two content elements \( \neg p \lor q \) and \( \neg p \lor \neg q \) – corresponding to ‘excluded worlds’ 10 and 11 respectively (the top class, \( T \), does not contain any content elements). The result, our familiar \( K \)-faithful preorder:

\[
\begin{array}{c}
00 \\
01 \\
11 10
\end{array}
\]

One may start with a \( K \)-faithful preorder and use the reversed method to get to an epistemic entrenchment ordering. These methods get more interesting when working in more complex languages.

It is clear that one has much more freedom in reasoning using nonmonotonic logic. Throughout this chapter quite a few methods of reasoning were highlighted with propositional logic as foundation. In terms of programming machines to think like humans, nonmonotonic reasoning is an enlightening course to follow.
Chapter 4

Many-Valued Logic

"Contradiction is not a sign of falsity, nor the lack of contradiction a sign of truth."
Blaise Pascal


Many-valued logic goes as far back as the 4th century B.C. when Aristotle considered the future contingent sentences within a modal framework. Grzegorz Malinowski [1993] refers to chapter 9 of Aristotle's treatise De Interpretatione where he introduced the famous sentence representing the future contingent: 'there will be a sea-battle tomorrow'. Future contingent sentences are neither actually true nor actually false, but potentially either true or false. Prior to the future event a third, indeterminate logical value exists. Although the above description is linked more to philosophy, the basic idea is that there might be a third truth value, apart from the orthodox 'true' and 'false'. The existence of a third truth value forms the central concept of many-valued logic.

Nicholas Resher [1969] refers to the early history of many-valued logic as having three founding fathers, namely Scotsman Hugh MacColl (1837-1909), American Charles Sanders Pierce (1939-1914), and Russian Nikolai A. Vasil'ev (1880-1940). MacColl created a system of propositional logic in which propositions can take on several distinct truth values, not only limited to the traditional values of 'true' and 'false'. A certain proposition is one which is always and necessarily true, an impossible proposition is always and necessarily false, while a variable proposition is sometimes true and sometimes false. An exam-
A simple example of a certain proposition is '2 = 2', an impossible proposition is something like '2 = 3' and a variable proposition is viewed as 'x = 2'.

Pierce, in the early 1900's, extended the truth table method to three-valued logic by creating a third value intermediate between determinate truth and determinate falsity as a truth status. He saw it as a limit between the two extremes of truth and falsity as explicated in a letter he wrote to William James in 1909: “The recognition (of the third value) does not involve any denial of existing logic, but it involves a great addition to it” [Resher 1969, p5]. He considered three-valued connectives which would be reinvented later by others like Łukasiewicz and Bochvar. (Pierce's truth tables will not be included here, instead we will discuss the truth tables of Łukasiewicz, Bochvar and Kleene later on.)

Vasil’ev developed his definition of three-valued logic based on the laws of Contradiction, Excluded Middle and No-Self Contradiction. The first law of Contradiction states that no object can have a property contradicting itself. The law of Excluded Middle defines an object as either having a certain property or not. Lastly, the law of No-Self Contradiction, states that one and the same object cannot be simultaneously true and false [Resher 1969].

Development of many-valued logic peaked in the early 1920’s with the work done by the Pole Jan Łukasiewicz and American Emil L. Post. Łukasiewicz enriched the set of the classical logic values with an intermediate value and laid down the principles of a three-valued propositional calculus. Post defined (finite) many-valued logical algebras in 1921, but unlike Łukasiewicz, never furthered his work on this subject [Malinowski 1993, Rescher 1969]. Other famous developments are those of Stephen C. Kleene and the Russian D.A. Bochvar around 1938. They are the founders of the original three-valued constructions motivated by the indeterminacy or absurdity of some propositions at a certain stage of investigation [Malinowski 1993].

We will discuss the three-valued logic of Łukasiewicz, Bochvar and Kleene. The basic difference between these logicians’ views on three-valued logic is the description of the third truth value (which influences truth tables, axiomatizations and generalizations). Łukasiewicz took the third truth value to be
an intermediate value, joining truth with falsity. Bochvar described the third value as labelling the undecidable, while Kleene used the third truth value to represent the unknown.

Following the discussion of the three-valued logic of Łukasiewicz, Bochvar and Kleene, we will highlight some many-valued extensions of the systems of Łukasiewicz and Kleene. Lastly we will discuss the general construction of a many-valued logic.

4.1 Three-Valued Logic of Łukasiewicz

Łukasiewicz first publicized his three-valued system of logic in a lecture presented to the Polish Philosophical Society in Lwów in 1920. He based his ideas on modality and a third value he described as "...neither true nor false...different from 0 (or, false) and 1 (or, true). This value we can designate by $\frac{1}{2}$. It may be read as 'possible' and joins 'truth' and 'false' as a third value. The three-value system of propositional logic owes its origin to this line of thought." [Rescher 1969, p23]. Łukasiewicz entertained the idea of Aristotle and his 'future contingent' matters that have a truth status of neither true nor false (at least at the current moment). In fact, according to Malinowski [1993], Łukasiewicz was a fierce follower of indeterminism. Łukasiewicz's indeterminism found its expression, amongst others, in the introduction of the third logical value to be assigned to non-determined propositions, specifically to propositions describing future events (the future contingent matters).

Łukasiewicz defined the normal propositional connectives in terms of three truth values. The truth values are the two classical values of 'true' (1) and 'false' (0) and the third being 'indeterminate' ($\frac{1}{2}$). Do take note that one may also use the letters T, F and I instead of 1, 0 and $\frac{1}{2}$ to denote the truth values. Łukasiewicz's three-valued system will be abbreviated with $L_3$ (the subscript indicates the number of truth values involved). The truth tables he put together are:
Lukasiewicz defined the above mentioned operators initially just with the two connectives \( \neg \) and \( \rightarrow \), and then proceeded by defining the rest by employing already defined connectives: \( p \lor q \) as \( (p \rightarrow q) \rightarrow q \); \( p \land q \) as \( \neg(\neg p \lor \neg q) \); and lastly \( p \leftrightarrow q \) as \( (p \rightarrow q) \land (q \rightarrow p) \) [Resher 1969].

Let \( v \) be a valuation (function) with domain \( \{p, q\} \) and range \( \{0, \frac{1}{2}, 1\} \). Investigation into the first truth tables leads to the discovery that \( v(\neg p) = 1 - v(p) \). When looking at the section in the truth table indicating \( p \land q \) one can find the intersection of the first column, where \( p = \frac{1}{2} \) and the second row, where \( q = 0 \) and see that the conjunction between \( \frac{1}{2} \) and \( 0 \) yields \( 0 \). In general, conjunction may be expressed by \( v(p \land q) = \min\{v(p), v(q)\} \). The same process applies to the section indicating the disjunction, \( p \lor q \), where one finds that \( v(p \lor q) = \max\{v(p), v(q)\} \). Thus Lukasiewicz’s negation as well as conjunction and disjunction are similarly defined as in the propositional case.

However, \( L_3 \) does differ from the normal propositional calculus (abbreviate with \( C_2 \) in the sense that some tautologies in \( C_2 \) are not tautologies in \( L_3 \). An example of a tautology in \( C_2 \), which is not a tautology in \( L_3 \) is the Law of Excluded Middle, \( p \lor \neg p \), or otherwise known as the Principle of Contradiction, namely \( \neg(p \land \neg p) \). Another difference between \( L_3 \) and \( C_2 \) is that some classical contradictory formulas in \( C_2 \) turn out to be non-contradictions in \( L_3 \). An example is \( p \land \neg p \) [Malinowski 1993].

Lukasiewicz introduced the modal operators of necessity (\( \Box \)) and possibility
(◊) into his three-valued logic. He described □ and ◊ with the help of a truth table, as depicted below (□ can be defined in terms of ¬ and ◊ as □p = ¬◊¬p):

<table>
<thead>
<tr>
<th>p</th>
<th>□p</th>
<th>◊p</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>½</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

From the above truth table one can see that neither the necessity operator (□) nor the possibility operator (◊) allow an intermediate status. This indicates that the modal operators label a proposition as either true or false. In the instance of the necessity operator, only a proposition that is true will be necessarily true. The possibility operation is less strict and labels both propositions that are true and intermediate as possibly true, while a false proposition stays false (with no possibility to become true).

Resher [1969] stated that Mordchaj Wajsberg succeeded to axiomatize the three-valued logic of Łukasiewicz in 1931 by the following axiom schemas together with the inference rule of Modus Ponens:

1. \( \alpha \rightarrow (\beta \rightarrow \alpha) \)
2. \( (\alpha \rightarrow \beta) \rightarrow ((\beta \rightarrow \gamma) \rightarrow (\alpha \rightarrow \gamma)) \)
3. \( (\neg\alpha \rightarrow \neg\beta) \rightarrow (\beta \rightarrow \alpha) \)
4. \( ((\alpha \rightarrow \neg\alpha) \rightarrow \alpha) \rightarrow \alpha \)

Łukasiewicz was a follower of indeterminism and described the third truth value as 'indeterminant', a value between 'true' and 'false'. In ordering his truth values in terms of desirability (normally it is better to know a proposition is true than to know it is false) one finds that 'indeterminant' is a more coveted outcome than the value 'false', whilst the value 'true' still seems to be the most welcomed outcome.


4.2 Three-Valued Logic of Bochvar

Bochvar built his three-value system on the idea of dividing propositions into sensible and senseless. A proposition is sensible (meaningful) provided it is either true or false, while any other proposition, falling outside this class, is senseless (or paradoxical). In 1939 he published a paper on three-valued logic in which he viewed the value $\frac{1}{2}$ (or I) as 'undecidable'. The truth value $\frac{1}{2}$ is not so much an 'intermediate value between truth and falsity, but rather the value indicating a sentence is senseless (or paradoxical). An example of such a sentence is 'this sentence is false' depicted by the letter $p$. The sentence $p$ can either be true or false. If $p$ is true, this self referred sentence will imply that 'this sentence is false' is true, and due to the referral of itself, $p$ is implied to be false (and vice versa). The above sentence is a known paradox and probably a tricky one to understand. It does, however, illustrate the use of the value $\frac{1}{2}$ in Bochvar's three-valued logic (one would assign the value $\frac{1}{2}$ to this paradoxical sentence $p$). We will refer to Bochvar's three-valued system as $B_3$ [Malinowski 1993].

Due to the fact that $\frac{1}{2}$ represent the value for a meaningless sentence, Bochvar's definition of the connectives turns out to have the common rule that says: If one of the input values of a compound sentence is meaningless (or undecidable) the outcome will also be meaningless (undecidable), otherwise the propositional rules pertaining to the connectives apply. The following truth tables show how Bochvar interpreted the connectives:

<table>
<thead>
<tr>
<th>$p$</th>
<th>$\neg p$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
Due to Bochvar's description of the third truth value as being the 'undecidable', each row or column headed by \( \frac{1}{2} \) contains only \( \frac{1}{2} \)'s inside the truth table. The result is that the truth value \( \frac{1}{2} \) weighs more than both 1 and 0.

### 4.3 Three-Valued Logic of Kleene

Kleene introduced a system of three-valued logic in 1938. He chose to use the third value \( \frac{1}{2} \) as an indicator of the unknown or the undeterminable in a knowledge-related way [Recher 1969]. One can consider 'knowledge-related' by means of an example of a very long and tedious algorithm running on a computer. In the end the program is supposed to give the answer of the algorithm as either 'true' or 'false'. However, sometimes this program can run for such a length of time, that it stops processing after a certain period of time (this time limit is often predefined), with an 'unknown' answer to the algorithm, that is, neither 'true' nor 'false'.

The connectives Kleene defined, presented in a truth table, are as follows:

<table>
<thead>
<tr>
<th>( p \text{ &amp; } q )</th>
<th>( p \lor q )</th>
<th>( p \rightarrow q )</th>
<th>( p \leftrightarrow q )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0.5</td>
<td>0.5</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( p )</th>
<th>( \neg p )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
Kleene based these truth tables on mathematical application such as described by Malinowski [1993]: Take for example the predicate $P$ of variable $x$ ranging over domain $D$, where $P(x)$ is defined for only a part of the domain, $P(x)$ if and only if $1 \leq x \leq 2$, thus $P(x)$ is

- true when $\frac{1}{2} \leq x \leq 1$
- undefined if $x = 0$
- false for all other cases, i.e. $x \neq 0, x < \frac{1}{2}, x > 1$.

We shall refer to Kleene's three-valued logic as $K_3$. The connectives discussed above portrays Kleene's strong connectives. He also created a set of weak connectives. The weak connectives coincide with Bochvar's set of connectives and yields the output value of $\frac{1}{2}$, whenever $\frac{1}{2}$ is any of the input values. Kleene created this weak family of connectives, trying to describe a mechanism that is incapable of processing indeterminate statements, represented by $\frac{1}{2}$. For instance, in Kleene's strong connectives, $\frac{1}{2}$'s conjunction with 0 yields 0, meaning that when one has to find the conjunction between two sentences, one being undecidable and the other false, one would accept the outcome to be false. On the other hand, when taking the connective as weak, $\frac{1}{2}$'s conjunction with 0 will yield $\frac{1}{2}$. The undecidability of the one sentence causes one to be unsure about the outcome of the conjunction, and thus one has to accept the outcome as $\frac{1}{2}$.

With a knowledge of the three developments above, we next discuss the adaptation of the three-valued logics of Łukasiewicz and Kleene in order to incorporate even more truth values.
4.4 Many-Valued Generalization of Three-Valued Logic

The development of three-valued logic opened the door to incorporation of more than three values in the basic idea of many-valued logic. We will discuss some adaptations to the three-valued logics of Lukasiewicz and Kleene.

4.4.1 Lukasiewicz’s Three-Valued Logic Adapted

In the early 1930’s Lukasiewicz generalized his three-valued logic to include more truth values, even an infinite number of values. As a reminder, note that \( \mathbb{L}_3 \) can be portrayed by the truth values 0, \( \frac{1}{2} \) and 1 or F, I and T respectively. Lukasiewicz divided the interval between 0 and 1 by inserting evenly spaced division points to created a generalization of his three-valued logic. (In fact, the points 0, \( \frac{1}{2} \) and 1 can be considered as a 3 point division.) A reference to the table indicating the division of \( n \) points \( (n \geq 2) \) is given by Resher [1969]:

<table>
<thead>
<tr>
<th>( n )</th>
<th>Division Points</th>
<th>Truth Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0, ( \frac{1}{2} ), 1</td>
<td>0, 1</td>
</tr>
<tr>
<td>3</td>
<td>0, ( \frac{1}{2} ), ( \frac{2}{3} )</td>
<td>0, ( \frac{1}{2} ), 1</td>
</tr>
<tr>
<td>4</td>
<td>0, ( \frac{1}{3} ), ( \frac{2}{3} ), ( \frac{3}{3} )</td>
<td>0, ( \frac{1}{3} ), ( \frac{2}{3} ), 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( n )</td>
<td>0, ( \frac{1}{n-1} ), ( \frac{2}{n-1} ), ..., ( \frac{n-1}{n-1} )</td>
<td>0, ( \frac{1}{n-1} ), ..., 1</td>
</tr>
</tbody>
</table>

From the above it is easy to see that \( \mathbb{L}_2 \) corresponds with \( C_2 \).

Lukasiewicz’s generalization can be applied to create infinite-valued systems by applying the following guidelines:

**The system** \( \mathbb{L}_{\infty} \) Take 0 and 1 with all the *rational numbers* between 0 and 1 as truth values. This set of truth values are denumerably infinite and has the same cardinality as the set of integers (or natural numbers).
The system \( L_{x_1} \). Take all the real numbers in the interval 0 to 1 as truth values. This set of truth values is not denumerably infinite and has the cardinality of the continuum [Resher 1969]. (The reader will see the connection between \( L_{x_1} \) and fuzzy logic in Chapter 5.)

To shed some light on the idea of Łukasiewicz's many-valued system, consider the truth tables for \( L_4 \) — created by taking 0 and 1 as well as two evenly spaced points between 0 and 1, namely \( \frac{1}{3} \) and \( \frac{2}{3} \):

<table>
<thead>
<tr>
<th></th>
<th>( \neg p )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>( \frac{2}{3} )</td>
<td>( \frac{1}{3} )</td>
</tr>
<tr>
<td>( \frac{1}{3} )</td>
<td>( \frac{2}{3} )</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

One may consider the negation of \( p \) as \( v(\neg p) = 1 - v(p) \), for example if \( v(p) = \frac{1}{3} \) then \( v(\neg p) = 1 - \frac{1}{3} = \frac{2}{3} \).

In the above table, \( v(p \land q) \) equates to \( \min\{v(p), v(q)\} \), while \( v(p \lor q) = \max\{v(p), v(q)\} \) — corresponding to \( L_3 \). For example, let \( v(p) = \frac{2}{3} \) and \( v(q) = \frac{1}{3} \), then \( v(p \land q) = \min\{\frac{2}{3}, \frac{1}{3}\} = \frac{1}{3} \) (as can be seen in the second row and fourth column) and \( v(p \lor q) = \max\{\frac{2}{3}, \frac{1}{3}\} = \frac{2}{3} \). It is interesting to note that the logical equivalence that applies in \( C_2 \), namely \( p \rightarrow q = \neg p \lor q \) does not apply in \( L_4 \) as \( v(p \rightarrow q) \) with \( v(p) = \frac{2}{3} \) and \( v(q) = \frac{1}{3} \) equates to \( \frac{1}{3} \), whilst \( v(\neg p \lor q) = \frac{2}{3} \).

In 1958 A. Rose and J.B. Rosser managed to create a successful axiomatization of Łukasiewicz's denumerably many-valued system, \( L_{x_0} \), by using the
inference rule Modus Ponens and the axiom schemas (1) to (3) of $L_3$ together with the following axiom schemas:

(4) $((\alpha \to \beta) \to \alpha) 	o ((\beta \to \alpha) \to \alpha)$

(5) $((\alpha \to \beta) \to (\beta \to \alpha)) \to (\beta \to \alpha)$

Lukasiewicz's generalizations corresponds with his idea of describing the third truth value as 'intermediate' (a value between 'true' and 'false'). He took the idea further and created new values by evenly spacing a number of $n$-values in an interval. The possibility to create a many-valued logic with the cardinality of the continuum is an exciting step towards further, more advanced, logics (as the reader will see in Chapter 5).

We will now see how Kleene approached many-valuedness. He used quite a different approach to that of Lukasiewicz, but the results are just as valuable.

### 4.4.2 Kleene's Four-Valued Logic

Kleene focused on the use of orderings to expand his three-valued logic to many-valuedness. As the reader will recall, Kleene's logic is knowledge-related. It is possible to consider two orderings on Kleene's logic namely $\leq_k$ and $\leq_t$. The first, $\leq_k$, is an ordering expressing the degree of knowledge. It is important to note that this ordering does not give a quantity, but rather an indication of the type of knowledge an agent has (he either knows a sentence is true or false, or, on the other hand, he does not know whether a sentence is true or false, meaning the sentence has an 'unknown' value linked to it). (We will discuss this ordering in more detail shortly.) The other ordering, $\leq_t$, expresses the degree of truth. Again, this is not a quantitative value, but indicates that truth ranks higher than falsity. The agent would prefer a sentence to be true, rather than false. Even a sentence with a truth value of 'unknown' is better than a sentence being false.

Kleene took the third value to be the unknown value (above referred to as $\frac{1}{2}$, but we will now comply with Fitting's [1994] notation and assign $\bot$ to denote unknown), being neither true nor false, but which can be swayed to either truth or falsity upon an increase in knowledge. The ordering $\leq_k$ applied
to the three truth values yields truth (T) and falsity (F) on the same level, above \( \bot \), indicating that it is better to know something – even if it is that the sentence is false – than not knowing whether a sentence is true or false. Thus \( \bot \leq_k F \) and \( \bot \leq_k T \). (Refer to figure 4.1)

The degree of truth ordering focuses on the fact that it is best to have a true sentence. Being unsure of whether the sentence is true or false, thus having truth value \( \bot \), is better than being certain that the sentence is false. Degree of truth yields a linear ordering with falsity at the bottom, the unknown truth value in the middle and truth at the top, that is \( F \leq_t \bot \leq_t T \). (See figure 4.1.)

The ordering \( \leq_t \) yields a complete lattice – a partially ordered set in which any two elements have a least upper bound and a greatest lower bound [Daintith and Nelson 1989]. Conjunction (\( \land \)) and disjunction (\( \lor \)) coincide with the \( K_3 \) conjunction and disjunction respectively. The knowledge ordering, \( \leq_k \), does not yield a complete lattice since the truth values F and T do not have any least upper bound [Fitting 1991]. Adding the truth value over-defined, denoted by \( T \), to \( K_3 \), a four-valued logic is created and both orderings \( \leq_k \) and \( \leq_t \) yield complete lattices. (See figure 4.2.)
To avoid possible confusion, we would like to bring to the reader’s attention the difference in notation. Where we previously referred to $\bot$ as the set of contradictions and $\top$ as the set of tautologies, these notations do not portray contradiction and tautology in this section. In the four-valued logic, one can compare $\bot$, the unknown, with the empty set ($\emptyset$), implicating a lack of knowledge, while over-defined, $\top$, can be compared with the pair $\{F,T\}$ containing conflicting information, claiming a sentence to be both false and true (in this case $\{F,T\}$ represents the knowledge that a sentence is both false and true). We refrain from using numerical values in the four-valued logic of Kleene, as it could possibly create an unwanted link between the numerical values and their normal linear ordering as rational numbers (this might only confuse the matter unnecessarily).

The ordering $\leq_k$ can now be described by figure 4.2 where $\bot$ lies at the bottom of the ordering, with both $F$ and $T$ above it (whilst $F$ and $T$ are not comparable). At the top of the ordering, one finds $\top$, being the least upper bound of both $F$ and $T$. The degree of truth ordering, $\leq_t$, is the binary relation where $F$ is at the bottom, below both $\bot$ and $T$. The truth value $T$ is at the top.
of the $\leq_t$ ordering, above both $\bot$ and $\top$. (See figure 4.2.) The truth values of Kleene's four-valued logic coincide with the truth values of Dunn and Belnap's four-valued logic system, published in 1976.

A generalization of Kleene's four-valued logic developed into bilattices. Bilattices are multi-valued logics with two orderings (and consequently, two sets of connectives), with certain relationships postulated between the orderings [Fitting 1991]. A pre-bilattice is a structure $(B, \leq_t, \leq_k)$, where $B$ is a non-empty set of truth values, $\leq_t$ and $\leq_k$ are partial orderings which provide $B$ with the structure of a complete lattice. The four operators are $\land, \lor$ for $\leq_t$ and $\otimes, \oplus$ for $\leq_k$. Consider $\land$ and $\lor$ as meet and join respectively under the $\leq_t$ ordering and $\otimes$ and $\oplus$ as meet and join under the $\leq_k$ ordering. Since there are four operators, there are twelve possible distributive laws. A distributive bilattice is a pre-bilattice in which all distributive laws hold. Kleene's four-valued logic is the simplest example of a distributive bilattice. The above definitions will soon become more clear in the light of the discussion.

Let us consider a more specific extension. For this structural example, instead of assigning classical truth values, assign probability estimates, that is values within the unit interval $[0, 1]$. Suppose one can only limit oneself to a sub-interval $[a, b] \subseteq [0, 1]$, then the truth values lies in the closed set $B = \{[a,b]|0 \leq a \leq b \leq 1\}$.

To form a pre-bilattice, one needs two orderings. The two suggestions for these orderings are given by:

- E. Sandewall [1985] who suggested an knowledge ordering by set inclusion, i.e. $[c,d] \subseteq [a,b]$ implies the knowledge inherited by $[c,d]$ is greater than that in $[a,b]$, because $[c,d]$ is narrower. Thus $[a,b] \leq_k [c,d]$ if $[c,d] \subseteq [a,b]$

- D. Scott [1982] suggested $[a,b] \leq_t [c,d]$ if $a \leq c$ and $b \leq d$ as a partial ordering expressing the degree of truth.

Considering the orderings in terms of lattices, bearing the above suggestions in mind, one finds that the degree of truth ordering, $\leq_t$, yields a complete lattice with $[a,b] \land [c,d] = [\min\{a,c\}, \min\{b,d\}]$ and $[a,b] \lor [c,d] = \max\{a,c\}, \max\{b,d\}]$. 


Here \([0, 0]\) is the least value, corresponding with the truth value 'false' and \([1, 1]\) is the greatest value, corresponding with the truth value 'true' (see figure 4.3).

The \(\leq_k\) ordering does not give a complete lattice. Meet (or conjunction) is always defined as \([a, b] \otimes [c, d] = [\text{min}\{a, c\}, \text{max}\{b, d\}]\), but join or disjunction is not always meaningful in context when defined as \([a, b] \oplus [c, d] = [\text{max}\{a, c\}, \text{min}\{b, d\}]\) (depending on the situation or values that one uses, disjunction may not always present unique outcomes). Here the least value in \(\leq_k\) is \([0, 1]\), but there is no greatest value in \(\leq_k\). Unfortunately one cannot create a bilattice, but this is still an acceptable structure for certain use.

Consider an example pertaining to the structure described above, only taking into account the values 0 and 1. This example yields three intervals namely \([0, 0] = \{0\}\); \([0, 1] = \{0, 1\}\) and \([1, 1] = \{1\}\) and collapses to \(K_3\).

Now take 0, \(\frac{1}{2}\) and 1 as values. This yields six subsequent truth values if one takes all possible combinations consisting of two numbers from the given values (see figure 4.3). Considering the \(\leq_k\) ordering, one finds three intervals that are maximal, namely \([0, 0]\) as the 'false' value; \([\frac{1}{2}, \frac{1}{2}]\); and \([1, 1]\) as the 'truth' value. The smallest in the ordering is \([0, 1]\). Lastly, there are the two intermediate values \([0, \frac{1}{2}]\) and \([\frac{1}{2}, 1]\) (refer to figure 4.3).

By choosing any discrete subset of the unit interval \([0, 1]\) (including 0 and 1), one can construct any generalization of Kleene’s three-valued logic. For example \(\{0, \frac{1}{3}, \frac{2}{3}, 1\}\) will yield a ten-valued logic.

4.4.3 General Construction of Many-Valued Logic

This discussion will portray the fundamental basics for creating a general many-valued logic, allowing a more general ordering than the presently discussed many-valued logic based on a linear ordering. In the previous section we discussed a six-valued logic where the six values are the six possible intervals within the linear ordered set \(\{0, \frac{1}{2}, 1\}\). In what follows we describe the construction of a many-valued logic where the values are intervals in any complete lattice \(L\) with ordering \(\leq_L\).

Describing an interval formally, let \(a, b \in L\) with \(a \leq_L b\). The interval
determined by \( a \) and \( b \), denoted by \([a, b]\), is \( \{x \in L | a \leq_L x \leq_L b\} \).

As described by Fitting [1991], the basic results of a generalization of a multi-valued logic, using \( \wedge, \vee \) as conjunction and disjunction under the \( \leq_t \) ordering and \( \otimes \) and \( \oplus \) as conjunction and disjunction under the \( \leq_k \) ordering, are given by the following summarization:

Let \([a, b], [c, d]\) and \([e, f]\) be intervals in \( K(L) \), then

- \([a, b] \leq_t [c, d]\) if and only if \( a \leq_L c \) and \( b \leq_L d \)
- \([a, b] \leq_k [c, d]\) if and only if \( a \leq_L c \) and \( d \leq_L b \)
- \([a, b] \wedge [c, d] = [a \wedge c, b \wedge d]\)
- \([a, b] \vee [c, d] = [a \vee c, b \vee d]\)
- \([a, b] \otimes [c, d] = [a \wedge c, b \vee d]\)
- \([a, b] \oplus [c, d] = [a \wedge c, b \wedge d]\)
- if \([a, b] \leq_k [c, d]\) then \([a, b] \wedge [e, f] \leq_k [c, d] \wedge [e, f]\)
- if \([a, b] \leq_k [c, d]\) then \([a, b] \vee [e, f] \leq_k [c, d] \vee [e, f]\)

Figure 4.3: A Six-Valued Logic
Consider the following example. Let $L$ be an underlying lattice, having four points 0, $a$, $b$ and 1, with $\mathcal{K}(L)$ consisting of nine points. $L$ interchanges 0 and 1, while leaving $a$ and $b$ as is. (See figure 4.4.)

Consider two agents A and B being asked ‘yes’ or ‘no’ questions. The four points have the following meaning:

$a$: Agent A says ‘yes’, while agent B says ‘no’.

$b$: Agent A says ‘no’, while agent B says ‘yes’.

$0$: Both agents A and B say ‘no’.

$1$: Both agents A and B say ‘yes’.

Intervals may now be constructed, by using the four possibilities above. The information contained in this interval must be interpreted as a combination of
the contained possibilities. For example, \([0, a]\) means agent B says 'no' while agent A's answer is not known in the light that both 0 and a mean that agent B says 'no', where as 0 together with a give contradictory information regarding the answer of agent A. the interval \([0, 0]\) means both agents A and B answer 'no'. \([a, a]\) means A says 'yes' and B says 'no' and lastly \([b, 1]\) indicates agent A's answer is unknown and agent B answered 'yes'.

With the above generalization we conclude this chapter on many-valued logic. Although many-valued logic was developed initially for application to philosophical problems, many other uses were found in the interim. The most important direct application of many-valued logic is the mathematical theory of fuzzy sets and the mathematical analysis of 'fuzzy' (which will be discussed in Chapter 5).

Many-valued logic was born due to a need in philosophy, but the power of a logic – able of handling more than the conservative two truth values – soon developed into an interesting field of Mathematics. Due to these developments it is possible to concentrate on the development of fuzzy logic, which is very applicable to the current day and age.
Chapter 5

Fuzzy Logic

“Vagueness, clearly, is a matter of degree.”
Bertrand Russell
in Thomas Meyer’s Semantic Belief Change, 1999

Normal mathematical modelling needs certain measures to create a model, but problems arise where measuring scales do not exist, as in the case of an offensive smell, touching, taste and so on. Another problem is the use of “vague” words to describe an object or situation, words such as many, little, very, low, etc. Fuzzy logic is a logic which generalizes a theory from crisp, discreet logic (like the logics discussed up to now) to a more continuous or fuzzy form. The construction of a more continuous theory creates the ability to take decision making and problem solving tasks that are too complex to understand quantitatively, and apply an approximation or less precise decision making.

Fuzzy logic is the logic of inexact, imprecise and vague propositions and gives a comparative notion of truth. Kiyoji Asai, Toshiro Terano and Michio Sugeno [1994, p9] state “From the word ‘fuzz’ comes the idea of a hazy outline, something not seen clearly, so ‘fuzzy’ means ‘unclear’ or ambiguous”.

Fuzzy logic was inspired by many-valued logic and was initiated in 1965 by Lotfi A. Zadeh, a Professor at the University of California. Petr Hájek [2002] feels that the idea of fuzzy logic has led to a revival of interest in many-valued logics.
In the first subsection we introduce the mathematically based syntax of the logic. In Section 5.2 we consider the methods used during applications of fuzzy logic.

5.1 Fuzzy Sets

A **crisp** set \( M \) contains elements \( a_1, a_2, \ldots, a_n \), i.e. \( M = \{a_1, a_2, \ldots, a_n\} \) or \( M = \{a_i|1 \leq i \leq n\} \). \( M \) may be a subset of a bigger set \( X \) [Bandemer and Gottwald 1995]. \( X \) is known as the *universe of discourse*. The universe of discourse has to be chosen to the specific situation. If, for example, one is considering the sentence 'some roads are slippery', the universe of discourse will be \( \{\text{All Roads}\} \). In most cases the universe of discourse is determined by the context and is not mentioned separately. One can think of the elements in \( X \) as being marked to indicate their *membership* to the set \( M \). This "marking" will be done with the help of a function, namely \( \mu_M : X \to \{0, 1\} \), such that for all \( x \in X \),

\[
\mu_M = \begin{cases} 
1 & \text{if } x \in M \\
0 & \text{otherwise.}
\end{cases}
\]

The function above will mark all \( x \in X \) which belong to \( M \) with a 1 and is called the *indicator function* or the *characteristic function* \( \mu_M \) [Bezdek 1993]. (See figure 5.1.)

Fuzzy logic was created as a result of the need for a more "flowing" logic, since every day situations are sometimes not crisp or clear enough to apply normal mathematical modelling. Many-valued logic helped in adapting applications to handle the more complicated cases, but the most complex cases were still an issue (those with input and output ranging over a continuum of possibilities). Although Łukasiewicz’s many-valued system \( L_{\lambda_1} \) touched upon the idea of a logic ranging over the continuum (see Chapter 4), it was not ideal (\( L_{\lambda_1} \) requires the specification of each truth value, whereas in fuzzy logic truth values are automatically generated by specified models – the reader will obtain a better understanding as this chapter unfolds). There was a need for some system able to handle the *gradual transition* between elements being members
Bandemer and Gottwald [1993] used the following example to highlight the difference between a crisp set and a fuzzy set (see figure 5.2). If one looks at the set of positive real numbers greater than or equal to 18 ($\geq 18$), there is a "natural" jump at the point $x = 18$ (i.e. $M$ is the set of positive real numbers greater than or equal to 18, whilst the universe of discourse, $X$, is the set of all real numbers. Thus, any real number greater than or equal to 18 is a complete member of the set $M$). Similarly, a natural jump occurs when one looks at the set $M_1$ of all ages of peoples that have reached adulthood. The naturalness of the jump is lost, though, if one intents to understand $M_1$ as the set of all ages at which a human being is (in a biological sense) full-grown.

The problem is not that the jump is placed at an incorrect point (for instance at $x = 18$), but rather the jump itself. If the jump is a crisp jump then a person goes from a non-member (of the set 'full-grown') at any age before 18, to a member of the set 'fully grown' at any age on or after 18. Realistically we know this is not a property that changes "in a moment", there is a gentle move from not being full-grown to being full-grown.
Figure 5.2: Difference between crisp jumps and gradual transition (broken line)

There are many more examples portraying the same effect of not having the ability to do a clear jump at a certain point (or even having a clear measuring scale). Some of these examples of sets are: 'some roads are slippery', 'the smell of the curry is too strong', 'the lights are very bright' and 'the baby is fairly small'. Intuitively one knows there is a gentle move from being a complete non-member to being a complete member in the above examples, and that the usual crisp sets cannot represent this flowing move. Fuzzy sets are designed to realize the gradual transition from non-membership to membership.

Fuzzy sets are created on the concept of assigning membership degrees to variables. Membership degrees are real numbers between 0 and 1, assigned to variables indicating a measure of their membership to a certain set. The number 0 indicates complete non-membership, while 1 indicates complete membership. Using an advanced characteristic function, one can assign a degree of membership to a certain variable ranging from 0 to 1. Malinowski [1993] gives the definition of a fuzzy set $A$ of a given domain $X$ as an abstract object, characterized by a generalized characteristic function $\mu_A$ with values in the real set $[0,1]$. The values of $\mu_A$ are interpreted as degrees of membership of the
elements of $X$ to the fuzzy set $A$. The function $\mu_A$ is called the membership function of $A$ over the universe of discourse $X$. One may refer to $A$ as a fuzzy set over $X$ or a fuzzy subset of $X$.

Fuzzy logic is a super set of conventional Boolean logic, meaning that it is possible to collapse fuzzy logic into Boolean logic. The collapsing is very easily done by disregarding the in-between values, i.e. values in the interval $(0, 1)$. One then takes into account the two most extreme values only, namely 0 and 1. Formally known as the Extension Principle, classical results of Boolean logic can be recovered from fuzzy logic operations when all fuzzy membership degrees are restricted to the traditional set $\{0, 1\}$. Thus, all crisp or traditional subsets are fuzzy subsets of a special case. There is consequently no conflict between fuzzy and crisp methods.

As an example of a fuzzy set, consider the set indicating a person’s youth. Some people are still “young” at an age of 55, the question is what is the definition of the notion young? This question will certainly raise different answers, as a 13 year old teenager will be convinced that a 41 year old person is about to die of old age, while a 60 year old person will probably tell you that a 41 year old person is still a spring chicken! Hung T. Nguyen and Elbert A. Walker [1997] give two possible models of the fuzzy concept ‘young’:

As a teenager might see it:

$$Y(x) = \begin{cases} 
1 & \text{if } x < 25 \\
\frac{40-x}{15} & \text{if } 25 \leq x \leq 40 \\
0 & \text{if } 40 < x 
\end{cases}$$

Taking the 13 year old’s view of a 41 year old person in this model, one sees that $41 > 40$ thus $Y(41) = 0$, and a person of age 41 does not fall into the set ‘young’ at all (i.e. 41 is a complete non-member of the set ‘young’ according to the model built from the point of view of a teenager).

An older person might model it differently, with the function:

$$Z(x) = \begin{cases} 
1 & \text{if } x < 40 \\
\frac{80-x}{40} & \text{if } 40 \leq x \leq 60 \\
\frac{70-x}{20} & \text{if } 60 < x \leq 70 \\
0 & \text{if } 70 < x 
\end{cases}$$
$Y(x)$ and $Z(x)$ are the membership functions which maps the value of $x$ to the real unit interval $I = [0, 1] = \{x \in \mathbb{R} | 0 \leq x \leq 1\}$. Using the above model, and the 60 year old person's view of a 41 year old person, one finds that with $x = 41$, $Z(41) = \frac{80-41}{40} = 0.975$, which is a very high degree of membership of the fuzzy set 'young'.

From the above example, one can perceive just how important it is to build a good model. Models can easily be influenced by a certain point of view (for example, the point of view of a teenager or the point of view of someone older). The building of models is currently human driven, and one needs substantial information to build an acceptable model. The more variables involved, the more complex the model. One may see the above two models as forming one set, with the two variables being 'age' and 'point of view'.

5.1.1 Operations and Properties of Fuzzy Sets

Fuzzy sets $A$ and $B$ are equal if they have the same membership functions, i.e. $A = B$ if and only if $\mu_A(x) = \mu_B(x)$ for all $x \in X$. Zadeh [1965] described a relation of inclusion on the family of fuzzy (sub)sets of a given domain $X$ by $A \subseteq B$ if and only if $\mu_A(x) \leq \mu_B(x)$ for any $x \in X$.

Operations like the complement ($\neg$), union ($\cup$) and intersection ($\cap$) on fuzzy sets may be expressed in terms of respective characteristic functions as follows:

$-A = X - A = \overline{A}$ by $\mu_{-A}(x) = 1 - \mu_A(x)$

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

$A \cap B$ by $\mu_{A\cap B}(x) = \min\{\mu_A(x), \mu_B(x)\}$

The usual set theoretical properties apply to fuzzy sets, for example idempotence ($A \cap A = A$ and $A \cup A = A$); commutativity ($A \cap B = B \cap A$ and $A \cup B = B \cup A$); associativity ($(A \cap B) \cap C = A \cap (B \cap C)$ and $(A \cup B) \cup C = A \cup (B \cup C)$); distributivity ($A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$ and $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$) and De Morgan's Law ($\overline{A \cap B} = \overline{A} \cup \overline{B}$).
Relations occurring in fuzzy logic are known as *fuzzy relations*. Gideon Schepers [1998] states that fuzzy relations are more or less vague relationships between some fixed number of objects and can be formally treated like fuzzy sets. For simplicity, only binary relations are considered. Consider the set whose members are ordered pairs of the universe of discourse $X$, where $X = U_1 \times U_2 = \{(u_1, u_2) | u_1 \in U_1 \text{ and } u_2 \in U_2\}$. A fuzzy relation $R$ on $X$ is nothing else than a fuzzy subset of $X$. The membership degree $\mu_R(a, b)$ in this case is interpreted as the degree to which the fuzzy relation $R$ holds true for the objects $(a, b)$. Bearing this in mind, $\mu_R(a, b) = 1$ indicates that the fuzzy relation $R$ for $(a, b)$ completely holds, while $\mu_R(a, b) = 0$ reads that the fuzzy relation $R$ for $(a, b)$ does not hold at all. For more information on the fuzzy relation the reader may refer to Nguyen and Walker [1997], Bandemer and Gottwald [1995], and Bodjadziev and Bodjadziev [1995].

### 5.2 Methods in Application of Fuzzy Logic

Fuzzy logic is used in various every day applications. It is therefore important to look at the structure, building blocks and methods used during a fuzzy process, from an application point of view. The first example that jumps to mind, is a washing machine. Some washing machines use fuzzy logic to assess the intensity of the washing process needed for a specific bundle of clothing. If the clothes are very dirty, a longer, more intense cycle is needed. If the wash consists of a few delicate garments, the washing machine will switch to a very gentle cycle. The washing machine will assess the quantity of water needed, based on the weight of the clothes and soap powder necessary according to how dirty the clothes are. Washing machines that use fuzzy logic are much more complicated than this example portrays, but the reader can get the gist of it.

We now consider the fuzzy logic reasoning process or, as it is known in the industry, the *fuzzy logic control*.

The process commences with a challenge to solve a problem, stated in normal English words. The problem consists of *linguistic variables*. Linguistic
variables provide a means of describing complex systems that cannot be represented by discrete mathematics (i.e. many-valued logic) [Cartwright and Calvo 1998]. Examples of linguistic variables are 'heavy', 'hot', 'full', etc. Linguistic variables are used in combination with linguistic modifiers — that is words like 'very', 'more or less', 'little', etc.

As the complexity of a system or problem increases, it becomes more difficult and eventually impossible to make a precise statement about its behaviour. A system may arrive at a point of complexity where the fuzzy logic method, born in humans, is the only way to "understand" the problem. This is known as the Principle of Incompatibility.

The goal is to create an algorithm to solve the problem. An algorithm is a step by step procedure of which computer programs are very good examples of. A fuzzy algorithm is a procedure made up of statements relating to linguistic variables. An example of a sentence in such a fuzzy algorithm will be something like: 'If the crate is rather heavy then apply force \( x \) to move it.' Algorithms consist mainly of \textit{IF-THEN} statements [Asai, et al. 1994].

Humans summarize information automatically. We take numerous inputs into account and summarize the situation. Conclusions are based on fuzzy assessments, fuzzy truths, fuzzy inferences and so on, resulting in an average summarized output. The output is given as a precise number or decision value which can be acted upon. \textit{Summarizing information} is an eminent concept in fuzzy logic.

It is important to bear in mind that there might be numerous fuzzy inputs and fuzzy rules to be applied in a single system. For the sake of simplicity, we will normally just refer to single inputs (regarding one condition only), together with limited rules.

To explain the method, we introduce an example: Say a factory uses a machine to move crates from point A to B, and the machine needs to have the ability to distinguish between 'light' and 'heavy' weights of the crates. By standards of the factory it never gets crates lighter than 10 kg and none heavier than 1000 kg. The machine knows, via built in or programmed information, how much force to use when moving a 10 kg crate (as a minimum setting)
and the maximum force needed to move a 1000 kg crate. With the help of
the fuzzy control it will calculate the force needed to move various crates with
weights between 10 and 1000 kg.

In reality the above mentioned machine would need to know the contents of
the crates too, for it will need to move a crate very delicately if the crate is filled
with porcelain, but if the crates is filled with rocks, there will be no need for
special care. Maybe there are different types of crates as well, and adjustment
to the machines clamps are required. There are many more possible rules to
be followed, but hopefully this creates an idea of the variety of fuzzy inputs a
single system might have.

There are 3 steps in a fuzzy logic control. In short, these 3 steps are
discussed by G. Bodjadziev and M. Bodjadziev [1995] and can be given by:

1. Fuzzification of the terms that appear in the conditions of rules.

2. Inference from fuzzy rules.

3. Defuzzification of the fuzzy terms that appear in the conclusions of the
   rules.

The first step consists of fuzzification. Some problems, for example moving
a crate with a machine, call for algorithms as solutions. The problem is not
easily stated in clear steps (clear steps would lead to the easy creation of an
algorithm), thus one is dealing with fuzzy perceptions or input conditions.

The fuzzy procedure allows a fuzzy condition in a sentence – such as ‘the
crate is rather heavy’ – to determine the extent to which the crate’s weight
belong to the fuzzy set, based upon the vague or fuzzy description ‘rather
heavy’. This is the assignment of a degree of membership to an element in a
fuzzy set. The assignment of a degree of membership is done according to the
linguistic description of the element’s status. Schepers [1998] more formally
described fuzzification as the process of mapping from observed inputs to fuzzy
sets in the various input universes of discourse.

For example, the crate’s weight is our fuzzy input variable and, if it’s ‘rather
heavy’, we may assign a 0.85 degree of membership to it (fuzzyfying the terms
in the condition by using predefined models). It is the gradual change of the
membership value (related to the condition 'heavy') that gives fuzzy logic its strength.

There are a number of linguistic modifiers (descriptions) to describe fuzzy variables, for example 'rather', 'a bit', 'mostly', etc. Typically, fuzzy concepts have an odd number of descriptions. By an odd number of descriptions we refer to the number of various classes between (and including) complete membership and complete non-membership. Normally in the middle of complete membership and complete non-membership one finds a medium class (indicating the middle class). Thus, there are three descriptions already, for example, light, medium and heavy. Then there are a split between the non-membership and the half-membership and on the other hand, between the half-membership and full membership. We have five descriptions or classes now. For example, we can extend the values: light, medium and heavy by adding very light and very heavy. The process of inserting descriptions can continue over quite a number of descriptions, but not ad infinitum, because our natural language is restricted. One of the powerful concepts regarding fuzzy logic is the fact that a single description, for example very light, labels a certain membership class (containing more than one value). Thus, one is able to use a small number of fuzzy descriptions (for example light, medium weight and heavy) covering the continuum between complete membership and complete non-membership, opposed to many-valued logic where a label for each input value would be necessary. Fuzzy logic ranges over a continuum of possible membership values, given only a restricted number of fuzzy descriptions of the fuzzy variable. Less descriptions reduce the number of rules required to capture the knowledge relating to a fuzzy concept.

Let us look at another example. Consider the fuzzy variable regarding 'tallness'. According to figure 5.3 the model specifies that anybody as tall as 150cm and shorter can be seen as 'short', i.e. a complete member of the fuzzy set 'short' and a complete non-member of the fuzzy set 'tall'. The gradual move between 'short' and 'tall' is between 150cm and 210cm, in which interval a person is of medium tallness of various degrees, whilst anybody of 210cm and taller is seen as a complete member of the fuzzy set 'tall'.
Figure 5.3: Values of the fuzzy variable 'Tallness'

The second step is that of the inference of fuzzy rules. Inference rules consist mainly of IF-THEN statements. These are statements made in typical English words [Bandemer and Gottwald 1995].

First the fuzzyfication of the variables in the conditions of the rules were done (in the first step). Now one must compute the truth value for the premise (assumption or condition part) as a whole, of each rule. In this inference step, one will use the logic operators: AND, OR and NOT [Schepers 1998]. The truth value must then be applied to the conclusion (or operating part) of each rule. (An example follows shortly.)

Certain rules, among the fuzzy rules, can carry even more weight than other rules. If this is the case the membership values assigned to the outcome of these rules will be multiplied by a value between 0 and 1, thus assigning a higher value (carrying more weight) to some rules. By default each rule's weight is set to 1.

For an example of a rule, consider the following: IF 'crate A is heavy' AND 'crate A contains dinner plates' THEN 'apply a greater force with a lot of care'. For this rule the degree of membership for 'crate A is heavy' (given the weight of the crate) and 'crate A contains dinner plates' (given the contents of the crate) needs to be found. Taking the logic operator AND into consideration one must assign the minimum value of the two to the outcome: 'apply a greater force with a lot of care'. That is, if 'crate A is heavy' has a
degree of membership of 0.92 and 'crate A contains dinner plates' has a 0.57
degree of membership, the chosen membership degree for the condition part
will be 0.57, because the AND operator indicated the choice of the minimum
between the original two values.

A fuzzy logic control requires the development of a *knowledge base* that
is the totality of all the fuzzy sets, over all variables in all circumstances,
making possible the stipulation of the IF-THEN rules by using fuzzy sets.
Important here is the role that experience and knowledge of a human expert
plays [Bodjadziev and Bodjadziev 1995]. For example, in the crate moving
scenario, someone needs to tell the machine what type of force is needed to
move what type of crate, the machine merely interprets given models.

The last step is that of *defuzzyfication*. Fuzzy rules generates fuzzy out-
comes, which need to be transformed back into objective terms, i.e. crisp
numbers, before using the output. Because one is working with fuzzy inputs
(for example, 'the crate is heavy'), many fuzzy control answers (outcomes) will
initially be a fuzzy value again (for example, 'move the crate with a maximum
force'). For the machinery or other components involved, the outcome needs to
be a crisp number (for example 'use force setting 9.7 to move the crate'), thus
the outcome needs to be transformed from a fuzzy value to a crisp number.

Nguyen and Walker [1997] state that after obtaining an overall fuzzy out-
come as a fuzzy subset over the output space, it needs to be summarized into a
single value and used as the actual control action. There are various methods
of defuzzyfication and no unique way to perform this action. The two more
commonly used techniques are the: centroid and mean of maximum
methods. The reader can find more of these methods in Nguyen and Walker [1997], Asai
et al. [1994] and Schepers [1998].

The defuzzification stage is not required if the outcomes are crisp concepts
such as 'the reason why the document did not print is a faulty printer'. In these
cases, fuzzy inference results in assigning confidence factors (or probabilities)
to the various outcomes. The reasoning is called *probabilistic reasoning* where
the rules of the calculus of probabilities are applied to derive confidence in
possible decisions [Nguyen and Walker 1997]. Do note that the calculus of
probabilities is a whole other field in mathematics and not part of fuzzy logic.

There are a wide variety of applications of fuzzy logic in the industry. For example, automatic train operations and applications in expert systems such as bus scheduling. Others are speech recognition in security, medical diagnoses, decision-making support and applications in business like modelling of large-scale systems. More detail on all of the above can be found in Asai et al. [1994], Mlynuk and Patyra [1996] and Bandemer and Gottwald [1995]. Another very interesting article can be found on fuzzy logic being used in breast cancer diagnosis (refer to Peña-Reyes and Sipper [1998]).

If we look at cameras and camcorders for example, fuzzy logic links image data to various lens settings. The first camera that used fuzzy rules was the Canon hand-held H800, introduced in 1990, which adjusted the auto-focus based on 13 fuzzy rules. Sensors measured the clarity of images in six areas. The rules take up about a kilobyte of memory and convert the sensor data to new lens settings. Panasonic relies on more rules to cancel the image jitter that a shaking hand causes in its small camcorders. The fuzzy rules infer where the image will shift. The rules heed local and global changes in the image and then compensate for them. In contrast, camcorder controllers based on mathematical models can compensate for no more than a few types of image jitter.

These days 'fuzzy logic' is a common household term. It is unusual to see a washing machine without fuzzy logic control. To take a peek at the "behind-the-scenes" of the buttons on a washing machine with fuzzy logic control, the reader may refer to G. Bodjadziev and M. Bodjadziev [1995].

Another very interesting application can be find in Mlynuk and Patyra [1996] on a fuzzy logic approach to handwriting recognition, where letters are broken up into dots for fuzzy recognition.

Fuzzy sets and logic must be viewed as a formal mathematical theory for the representation of uncertainty. Uncertainty is crucial for the management of real systems. In this sense, fuzzy logic is a great leap forward, compared to propositional (and predicate) logic. By being able to give vague descriptions to fuzzy machines while receiving clear answers, humans are saved a lot of time.
and energy.

There is an increasing market demand for fuzzy logic based products, however, some current fuzzy logic systems appear to be more burdensome than helpful. Although fuzzy logic systems can boast with advantages such as flexibility, programmability and compatibility with current systems, there are systems that perform better using the already existing methods (even if they are ‘old-fashioned’).
Limits. Everyday we are confronted with limits. Limitations occur in various resources, such as time, money, space, ideas and so on. In the past, mathematical logic was also extremely limited. Although it is an extremely effective problem solving method, one can only reason to the far extremes of total truth or total falsity, restricting mathematical reasoning to simple problems. When confronted with complex cases, the true-or-false type of reasoning does not suffice. Logicians overcame this confinement, little by little, and today we can call on powerful logic systems such as nonmonotonic and fuzzy systems.

Man's ability to create, is based on his visions. One could hardly expect Aristotle to imagine computers as we know them today, neither their databases nor algorithms. Was it not for his ideas though (way back then), we would probably not have computers today.

An overview of the process of mathematical logic's growth is depicted in this dissertation. Man began at the very beginning, distinguishing only between truth and falsity (a huge leap in those days, and definitely one in the right direction). Like a sound “abc”, propositional and predicate logic were developed to be the basis for other mathematical logics. One needs to crawl first, before one walks. Given this frame of reference, humans could let their imaginations roam free.

The thought of being limited by using only truth and falsity, was not a foreign concept during the beginning of mathematical developments in logic. It did not, therefore, take very long for the first expansions of propositional
and predicate logic.

As time progressed, so did thoughts, visions and ideas. Soon mathematicians were developing more enhanced logics, such as modal, many-valued and nonmonotonic logics. In fact, modal logic (or the idea behind it) was considered by Aristotle himself. New developments encouraged mathematicians (and people in other fields – for example computer scientists) to broaden their thinking and produce new ideas.

Relatively speaking, fuzzy logic is one of the most recent developments. A very powerful logic, given current computer strengths. Fuzzy logic is a system not without drawbacks, even with powerful computers driving fuzzy logic systems. For example, as the number of inputs in a certain system increase, so does the level of complexity, rendering even enormous computers incapable of coping. Currently the success of any fuzzy logic system depends on the model driving it. These models are built by humans, based on a variety of information gathered over time. If, for instance, in medical diagnoses, the reference base says the symptom of a sore throat is associated with an appendix, the diagnoses will certainly be wrong and you might lose your appendix while only suffering a cold.

In this day and age we are standing on the verge of a totally computerized environment. The fridge will soon tell you that you are out of milk and that your brand of milk is currently on a special at a specific shop. It will be able to order and pay for new milk. Keeping an open mind, you might be able to envisage a little robot collecting the milk from your front gate, programmed with the ability to judge whether the milk is fresh (by referring to the sell-by-date and the smell and colour of the milk). The robot might even be able to tell you to increase your intake of fresh fruit and vegetables as your pale skin color, dark rings under your eyes, your level of fatigue and current intake of these produce indicates a lack thereof. In the case that you did not sleep at all during the night before (due, for instance, to a deadline that needed to be met), the robot can decide that this is more than enough reason for your physical appearance and fatigue. You can let your imagination run wild amongst the multitude of possibilities.
The fact is that we are now equipped with sound reasoning tools and a method to use algorithms, established with limited rules, to calculate outputs based on loads of input. Not even 'the sky is the limit' applies anymore. We already broke that barrier.
References


Gärdenfors and Makinson 1988 — P. Gärdenfors and D. Makinson. Re- 
vision of Knowledge Systems using Epistemic Entrenchment in [Vardi 
1988], pages 83-95.

Gärdenfors and Rott 1995 — P. Gärdenfors and H Rott. Belief Revision 
in [Gabbay et al. 1995], pages 35-119.

Ginsberg 1994 — M.L. Gingsberg. AI and Nonmonotonic Reasoning in [Gab-
bay et al. 1994], pages 1-31.

Hájek 2002 — P. Hájek. ‘Fuzzy Logic’ in The Stanford Encyclopedia of Phi-
losophy (Fall 2002 Edition). Edward N. Zalta (ed.), 


Heidema and Labuschagne 2000 — J. Heidema and W.A. Labuschagne. 

‘Refined Epistemic Entrenchment’ in Journal of Logic, Language and 
Information 9(2), pages 237-259.

Hodel 1995 — R.E. Hodel. An Introduction to Mathematical Logic. PWS, 
Boston.


Łukaszewicz 1990 — W. Łukaszewicz. Non-Monotonic Reasoning. Ellis Hor-
wood, West Sussex.

Oxford.

McCarthy 1977 — J. McCarthy. ‘Epistemological Problems of Artificial In-
telligence’ in Proceedings of the Fifth International Joint Conference on 
Artificial Intelligence, pages 1038-1044. MA, Cambridge


Index

AGM, 37
Alchourón, 37
Aristotle, 3, 51
atomic formula, 16
atomic sentences, 5
atoms, 5
Autoepistemic, 35
axioms, 6
belief change, 36
bilattices, 64
Bochvar, 52, 56
Boole, 3
characteristic function, 70
closed world assumption, 32
complete lattice, 62
connectives, 5
content elements, 48
contradictions, 12
crisp, 70
default logic, 35
defuzzification, 80
dual, 23
EE-orderings, 46
entails, 12
epistemic entrenchment orderings, 46
epistemic state, 31
expansion, 38
extension of a theory, 34
first-order language, 15
frame axiom, 33
Frege, 3, 20
fuzzification, 77
fuzzy relations, 75
fuzzy set, 72
Gärdenfors, 37
Gödel, 4
Harper, 42
Hilbert, 4
infatoms, 48
inference of fuzzy rules, 79
interpretation, 10
K-faithful, 40
Kant, 20
Kleene, 52, 57
Kleene's
degree of knowledge, 61
degree of truth, 61
KLM, 44
knowledge base, 80
Kraus, 44
Kripke model, 24

Lehmann, 44
Leibniz, 19
Levi, 42
Lindenbaum-Tarski, 13
logically implicit, 32
logically strongest, 32
Lukasiewicz, 52

MacColl, 51
Magidor, 44
Makinson, 37
membership, 70
membership degrees, 72
membership function, 73
modal nonmonotonic logic, 33
modal pair, 21
Modus Ponens, 7
monotonicity, 30

nonmonotonic, 30
nonmonotonic consequence relation, 44

Pierce, 51
Post, 52
pre-bilattice, 64
predicate
wffs, 16

qualification problem, 33
quantifiers, 15
removal, 37, 38
revision, 37, 41
Rules of inference, 6
Russell, 4

semantics, 8
set of models, 10
standard model, 24
syntax, 4
system S5, 25

tautology, 11
theorem, 6
truth value, 8

universe of discourse, 70
valuation, 8
Vasil’ev, 51
wffs, 6
Whitehead, 4
withdrawal, 38
Zadeh, 69

92