FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Implementierung und Vergleich von Quantoren-Fuzzifikationsmechanismen

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Logic and Computation

eingereicht von

Cem Okulmus, BSc.
Matrikelnummer 01215428

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Ao.Prof. Dipl.-Ing. Dr.techn. Christian G. Fermüller
Mitwirkung: Mag.rer.nat Dr.techn. Paolo Baldi

Wien, 1. März 2018

Cem Okulmus | Christian G. Fermüller

Implementing and Comparing Quantifier Fuzzification Mechanisms

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Logic and Computation

by

Cem Okulmus, BSc.
Registration Number 01215428

to the Faculty of Informatics

at the TU Wien

Advisor: Ao.Prof. Dipl.-Ing. Dr.techn. Christian G. Fermüller
Assistance: Mag.rer.nat Dr.techn. Paolo Baldi

Vienna, 1st March, 2018

Cem Okulmus   Christian G. Fermüller

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.ac.at

Erklärung zur Verfassung der Arbeit

Cem Okulmus, BSc.
Neustiftgasse 83 / 143, 1070 Wien

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 1. März 2018

Cem Okulmus

Danksagung

Ich danke meinen beiden Betreuern für ihren Rat und ihre Geduld und meiner Familie und Freunden für ihre stete Unterstützung.

Acknowledgements

I thank my two advisers for their guidance and patience, and my family and friends for their constant support.

Kurzfassung

Diese Arbeit dient einer einheitlichen Präsentation der neuesten Modelle von Quantoren-Fuzzifikationsmechanismen (QFMs). Diese sind ein Kernstück eines methodischen Rahmens von Ingo Glöckner, um Fuzzy-Quantoren von Semi-Fuzzy-Quantoren abzuleiten. Es erfolgt eine Analyse dieser Modelle im Bezug auf die axiomatischen Eigenschaften, die Glöckner definiert hat, um die "linguistische Adäquatheit", sowie andere wichtige Eigenschaften von QFMs zu charakterisieren. Zusätzlich wurde als Teil dieser Arbeit ein plattformunabhängiges Programm entwickelt, das es erlaubt Queries mit Fuzzy-Quantoren zu evaluieren und Plots der Wahrheitsfunkionen von Semi-Fuzzy-Quantoren und von QFM abgeleiteten aussagenlogischen Operationen darzustellen.

Abstract

This thesis provides a uniform presentation of recent models for Quantifier Fuzzification Mechanisms (QFMs). The idea of a QFM is at the core of a framework by Ingo Glöckner for modelling fuzzy quantifiers starting from semi-fuzzy quantifiers. It proceeds to analyse QFMs according to the axiomatic properties which Glöckner defined for them in order to characterize "linguistic adequateness" and other important properties. Additionally, as part of this thesis, a multi-platform tool has been developed to evaluate queries using fuzzy quantified expressions and to visualize plots of truth functions of semi-fuzzy quantifiers and of propositional operations induced by various QFMs.

Contents

CHAPTER 1

Introduction

Quantifiers are an important expressive device in natural languages (NL). They allow for statements not just about single individuals or objects, but provide effective means to express something about collections. A statement like “Few women are smokers”, regardless of its veracity, exemplifies this. Especially binary quantifiers, such as the above “few” are used frequently. Other examples of quantifier expressions are “most”, “many”, “about half”, “almost all”, etc. It is notable, that all these expressions exhibit some vagueness in their meaning.

In NL we often encounter vagueness. An example is the property “tall”, where different individuals might disagree on what height is accepted as tall. One choice to formally model such vague notions has been introduced by Lotfi A. Zadeh [Zad65]. He coined the term of *fuzzy set*. In such a set, every individual is a member *to some degree*. This membership degree translates into a degree of truth of an expression like “Tom is tall”, where Tom is an individual in our domain (also known as base set, referential set or universe of discourse). In other words: the fuzzy approach consists in replacing the binary notions of fully accepting or fully rejecting a statement with degrees of truth that a statement can have. Usually the interval $[0, 1]$ is used for these degrees.

Combining both, quantification and fuzzy sets, leads to *fuzzy quantified expressions* [Zad83]. This is also something that Zadeh focused on, providing early models for fuzzy quantifiers that used fuzzy sets as arguments for quantifiers, such as “almost all”, “few”, etc. Ingo Glöckner wrote a pioneering monograph [Glö06], in which he analyses the fuzzy quantification models of Zadeh and others, and finds them to be linguistically inadequate. In his monograph, Glöckner also shows how these early models are unable to model quantification as it occurs in NL . Another motivation for his novel approach is the sheer number of possible truth functions for fuzzy quantifiers. As emphasized in [FR13], even for unary fuzzy quantifiers that can be represented by a truth function of $[0, 1] \to [0, 1]$, there are uncountably many choices. A circumstance Fermüller and Roschger refer to as embarrassment of riches.

Glöckner's approach consists in using restricted quantifiers, called semi-fuzzy quantifiers (SFQs), as the basis. A semi-fuzzy quantifier only takes crisp sets as arguments, i.e., the usual set from mathematics. Glöckner proposes a "lifting" of semi-fuzzy quantifiers to (fully) fuzzy quantifiers, which take fuzzy sets as arguments. Such a lifting is coined a quantifier fuzzification mechanism (QFM). In his monograph [Glö06] he introduces a number of possible models for QFMs. Since there is in principle no restriction as to how a QFM can achieve this lifting, Glöckner also introduces a number of axioms that are meant to guarantee, among other things, linguistic adequateness. A QFM that fulfils these axioms or properties, is then called a determiner fuzzification scheme (DFS). Following his monograph, there has been further research in finding different QFMs for various purposes, such as information retrieval [DRSV14, BF18].

This thesis aims at presenting the existing literature on this topic in a uniform way and analysing which of Glöckner's axioms are fulfilled in various different models. As part of this thesis a tool was developed that allows one to evaluate fuzzy quantified statements using QFMs and SFQs that will be presented in the remainder.

The thesis is structured as follows:

Chapter 2 surveys related work, summarizing early research on fuzzy sets and quantification. *Chapter 3* is intended to provide for a reader with a basic background in Computer Science, all the needed terminology and concepts to understand the rest of the thesis. This is followed by *chapter 4*, which introduces two categories of semi-fuzzy quantifiers, as they form the building block of Glöckner's framework. *Chapter 5* provides a thorough examination of the various quantifier fuzzification mechanisms, that make up the focus of this thesis. This includes, to the best of the author's knowledge, all research that has been published on novel QFM models since Glöckner's pioneering monograph. It is followed by *chapter 6*, a discussion chapter which consists in a comparison of the presented QFMs with respect to whether they fulfil or break the DFS requirements of Glöckner. The principal idea here is to get a better understanding of these models, as there might be good reasons why one might be interested in QFMs that do not satisfy all these properties. The thesis concludes with *chapter 7*, a presentation of the tool which was developed as part of this work. This not only includes some technical descriptions, but also reports on the personal experiences of the author while working on it: what turned out to be surprising, difficult, or just interesting. The thesis closes with a conclusion of the presented results and briefly mentions further issues in fuzzy quantification, trying to go beyond Glöckner's QFM framework.

CHAPTER 2

Related Work

An influential line of research for this thesis originates in the field of linguistics. The "Theory of Generalised Quantifiers" (TGQ) [BC81, PW06] is a rich and complex subject. It tries to interpret noun phrases using generalised quantifiers as they are studied in linguistics. It presents a basic scheme for all possible quantifier expressions, called "determiners". A determiner is part of a larger phrase, see Figure 2.1. A generalised quantifier takes an arbitrary number of (classical) predicates denoting crisp sets as arguments. A crisp set corresponds to a subset of the domain D, leading to the definition of the semantics of a generalised quantifier Q as a function of type $\mathcal{P}(D)^k \to \{0,1\}$.

Vague quantifiers, it should be noted, are not modelled directly in the TGQ. This is not an omission, but rather corresponds to the standard way of modelling vagueness in linguistics. It proceeds in two separate steps. First we assume the choice of an appropriate context for all vague determiners and arguments. Later on, we can treat the whole expression as crisp, meaning it is either accepted or rejected. In other words, vagueness is treated as a *context dependency* and, as explained in [PW06], vagueness is seen as a more general concern and not specific to the modelling of quantifiers. Referring to the example in Figure 2.1, for some speaker, in a specific situation, it is assumed that it is clear what "students" and "are young" means, even if this meaning might change for another person, in another situation. Therefore the meaning of the noun phrase becomes crisp, completely true, or completely false, even if dealing with vagueness. Barwise and Cooper refer to this as the "fixed context assumption" [BC81]. In this thesis, however, we assume that vagueness can be usefully modelled directly as graded notions of truth.

The first to propose fuzzy quantification and provide concrete models was Lotfi A. Zadeh [Zad83]. Prior to this, he established a framework for fuzzy set theory [Zad65], an influential tool for dealing with gradual notions. Zadeh also focuses on the need of modelling expressions such as "many" or "few", applied to arguments that are themselves vague. His work categorizes two types of such quantifiers, "absolute quantifiers" which produce a truth function based on an absolute cardinality (e.g. "around 5"), and proportional

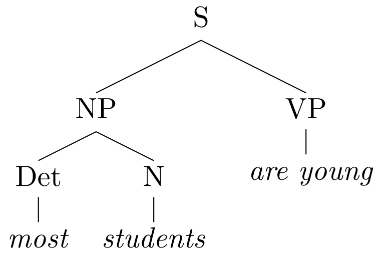


Figure 2.1: A phase-structure grammar of an example sentence (S) with a quantifier expression. The noun-phrase (NP) consists of a noun (N) and determiner (Det), followed by a verb phrase (VP).

(or relative) quantifiers that express a proportionality between two fuzzy sets. Especially this second kind of two-place quantifiers try to model a large class of expressions that we encounter often in natural language. One reason for why they are important, is the semantic role of the two arguments. The first is referred to as *range*, it acts as a restriction that determines which objects are considered, and the second as *scope*, which specifies something about these objects. An example can be seen in Figure 2.1, “most students are young”, where the range is a group of “students” and the clearly vague notion of “are young” is the scope. The quantifier “most” clearly refers to the proportion of young persons among the students.

Since there are no trivial ways of evaluating such quantifiers over fuzzy sets (since there is no obvious, unique way of generalising the notion of cardinality from the crisp to the fuzzy case), a large focus in Zadeh’s research, and of others who follow this framework, consists in providing various “evaluation methods”. The one that Zadeh initially proposes is the Σ-count method [Zad65], others were the “ordered-weighted-average” (OWA) method [Yag98], FG-count and FE-count [Zad83]. An in-depth analysis of these methods is also given in Ingo Glöckner’s monograph [Glö06]: he refers to them as the “traditional modelling framework”.

Glöckner’s monograph on fuzzy quantifiers [Glö06] combines the above, i.e., Zadeh’s fuzzy logic and the TGQ. He begins with an analysis of the fuzzy quantifier models of Zadeh and others who have extended it. Especially the inadequateness in modelling quantifiers from natural language is made clear. Another issue he criticizes is the arbitrary nature of the resulting models. As an example, there are many possible ways to model “about half”, when dealing with fuzzy sets as inputs. It seems unsatisfactory to just settle with an arbitrary one. Therefore his goal is to provide a model for generalised fuzzy quantifiers which also reduces the number of admissible models.

Glöckner’s framework focuses first on semi-fuzzy quantifiers, which range exclusively over crisp sets, and then defines a mechanism to extend them to fuzzy quantifiers, which can accept fuzzy sets as inputs. The latter is called a “quantifier fuzzification mechanism” (QFM). Of course, without further restrictions there are again many ways in which one can specify such a method. Therefore, he also proposes a number of properties that a

QFM should have. A QFM that satisfies them all is then referred to as a "determiner fuzzification scheme" (DFS). He additionally shows that such a DFS can be used to define operations such as the union and intersection on fuzzy sets, which connects his approach to the earlier fuzzy logic of Zadeh. Remarkably, he also shows how the DFS methods he found generalise certain "evaluation methods" from the Zadeh framework. This shows that, while the focus of his work is different from those of those early "pioneers", there is still a deep connection between them.

More recently, there have been attempts to model quantifiers that deal with fuzzy sets, without the two-step approach that Glöckner proposed [Hol08, DH09, Nov08].

Of the presented approaches to modelling fuzzy quantifiers, the Zadeh framework can be seen as the least abstract. Glöckner generalises it by providing classes of quantifiers, as each QFM represents all models of fuzzy quantifier that can be lifted from semi-fuzzy quantifiers. In addition, he also provides axioms that QFMs should satisfy.

The next abstraction step is to forgo concrete models for fuzzy quantifiers, but instead to analyse properties that all of them share. An example of this can be found in [Hol08], where a certain kind of algebraic structure, called residuated lattices, is used to analyse generalised quantifiers. These algebras are interesting since they can be used to model the semantics of Hájek's mathematical fuzzy logics (MFL) [Há06].

CHAPTER 3

Preliminaries

This chapter is meant as a succinct, but self-contained introduction into the related topics which are needed to follow the rest of this thesis. They are organized into a few principal categories. The first is what is referred to as "fuzzy set theory" and introduces and motivates the early works of Zadeh [Zad65]. Next is an introduction into properties of logical quantifiers. This leads directly to the next topic, fuzzy quantification, with an overview of the various types of fuzzy quantifiers and a brief introduction of generalised quantifiers [BC81]. After that Glöckner's unique approach of using semi-fuzzy quantifiers to produce models for fuzzy quantifiers [Glö06], and the motivations for this two-step approach are presented. Thereafter the topic of mathematical fuzzy logic according to Hájek [BC81] is introduced, as it is important for a number of other concepts in the following chapters. This chapter ends with a presentation of a dialogue game designed by Robin Giles [Gil74], originally formulated for reasoning about experiments in physics. The game characterizes ukasiewicz logic — one of the fuzzy logics introduced later in this chapter — and has more recently been used as a framework for defining quantifiers as well as quantifier fuzzification mechanisms.

3.1 Fuzzy Set Theory

An important notion that proved to be very influential in both science and engineering, is fuzziness. It resolves vagueness through the use of *truth or membership degrees*. For a given property, like, say, "tall", "rich", etc., we are unable or unwilling to state determinately for a given element in a set (or figuratively an individual in a group of persons) whether it has this property, or not. Instead, we simply assign a value, usually from the interval $[0,1]$, to each element, which indicates its membership to some degree. A justification for using the fuzzy approach are certain in logical paradoxes, such as the famous Sorites paradox. For more information on this [Hyd14] is recommended.

A *domain* (or base set) D is some non-empty (possibly infinite) collection of elements. A set (also referred to as crisp set) A on D is an element of the powerset $\mathcal{P}(D)$. It can be identified by its membership function $\mu : D \to \{0, 1\}$ (also known as indicator or characteristic function). The latter is defined as:

$$\mu_A(x) = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{otherwise} \end{cases} \tag{3.1}$$

The *fuzzy powerset* $\tilde{\mathcal{P}}(D)$ collects all *fuzzy sets* $\tilde{A}$ on D over a given domain D. Every fuzzy set is specified by its membership function $\mu_{\tilde{A}} : D \to [0, 1]$, i.e.

$$\mu_{\tilde{A}}(x) \in [0, 1] \tag{3.2}$$

Thus $\mu_{\tilde{A}}$ expresses a membership *degree*, where elements of D with membership degree 0 are understood to be absolutely outside the fuzzy set, and those with membership degree 1 are understood to be absolutely contained. We can see that one can consider fuzzy sets as a generalisation of crisp sets. In other words: crisp sets are a special case of fuzzy sets. In cases where it might not be clear whether a set can be fuzzy, the set in question shall be marked with a tilde symbol overhead: $\tilde{A}$ to mark it clearly as a fuzzy set. In cases where its clear from context, we will leave this out.

An obvious and natural mapping from fuzzy sets to crisp ones, are *alpha-cuts*. An alpha-cut $\tilde{A}_\alpha$ is a crisp approximation of a given fuzzy set $\tilde{A}$, for a fixed degree $\alpha \in [0, 1]$. Its membership function is as follows:

$$\mu_{\tilde{A}_\alpha}(x) = \begin{cases} 1 & \text{if } \mu_{\tilde{A}}(x) \geq \alpha \\ 0 & \text{otherwise.} \end{cases} \tag{3.3}$$

A more involved process to acquire crisp approximations of a fuzzy set, called three-cuts, is proposed by Glöckner in his monograph [Glö06]. It involves partitioning the range $[0, 1]$ into three regions, essentially yielding a set with a three-valued membership function. This is then mapped onto a range of possible crisp sets. The reason why Glöckner introduces a new way of producing crisp approximations is that three-cuts are symmetrical with respect to complementation, while alpha-cuts are not. Three-cuts will be introduced in more detail in chapter 5, as it is a central component of Glöckner's framework.

3.2 Logical Quantifiers

The expression *quantifier* is used extensively both in logic and linguistics, if not necessarily with the same meaning. The focus of this thesis is the type of quantifiers used in logic, but with an attempt to logically model certain parts of natural language phrases, called "determiners" in linguistics. The classical logical quantifiers are the existential ($\exists$) and the universal ($\forall$), and, at the first-order level, they quantify over single objects of the domain. One immediate and early generalisation of the existential and universal quantifiers are quantifiers "in-between" these two basic ones, so called "intermediate quantifiers". Before continuing, we introduce a useful convention: throughout this thesis, we shall assume that we have for every element in our domain D a constant c that represents it. We will then use elements from the domain informally as constants in the object language, for ease of notation. Examples for intermediate quantifiers are "many", "few" or "most". As the existential quantifier is the supremum ($\exists x P(x)$ if at least one constant $c \in D$ s.t. $P(c)$) and the universal one the infimum ($\forall x P(x)$ if all $c \in D$ s.t. $P(c)$). Intermediate quantifiers allow for semantics in between these extreme points. It should be noted that *these are not necessarily fuzzy quantifiers.* They do show the need, however, for more varied forms of quantification, already in the classical first-order setting. Additionally, higher-order logic introduces quantification over functions and relations, and relations of relations and so forth. This thesis focuses on first-order quantification. To give an example: "Few bankers are poor", can be written in a more formal notation as $Few\ x\ (Banker(x), Poor(x))$, where Few is a binary quantifier with some specified truth function, and the predicates $Banker$ and $Poor$ can be modelled as fuzzy sets.

The 'Theory of Generalised Quantifiers" (TGQ) [BC81] is a scheme that essentially considers a quantifier to be some mapping from crisp sets to a truth value. In the original concept this is a classical two-valued value. Formally

$$Q : \mathcal{P}(D)^k \to \{0, 1\},$$

where k, called the arity, signifies how many arguments the generalised quantifier Q operates on.

As mentioned earlier, an important pattern that is often encountered in quantified expressions is that of *range* and *scope*. We consider an example expression "*Some Y_1's are Y_2's*". The first argument Y_1 restricts the objects to which the quantifier refers and is called the *range*. The second argument Y_2 asserts something about the individuals specified in Y_1, this is called the *scope*. Glöckner proposes in his framework an extension of this to n-ary quantifiers, by treating the last argument as the scope, while no direct equivalent for the range is given, in general.

A comprehensive survey of quantification in both linguistics and logic is provided by Peters and Westerståhl [PW06]. The authors state that, while there is no complete agreement on what precisely "logical quantifier" should mean, there is widespread agreement that a minimal requirement is that truth values of logically quantified sentences are invariant under permutations of domain elements. More generally the truth value should remain invariant also under cross-domain bijections. We call this property ISOM. To state it more formally we need a few basic definitions:

The property ISOM is not limited to quantifiers, and in fact defined in a more general type-theoretic setting. For a more detailed description, the interested reader is advised to look at chapter 9 of "Quantifiers in Language and Logic", which includes an introduction into the *objects of finite types*.

The definition of the property is done on *universal operators*, these are general functionals O that associate to each domain M a concrete object O_M. This is a more general concept, but clearly also includes quantifiers, which are still defined if one swaps the arguments with sets from a different domain of discourse.

For the definition as it is given in ([PW06], p. 326), a way to generalise bijections of the kind $f : M \to M'$ to bijections on objects of finite type over M to objects of the same type over M' needs to be introduced. This is done in an inductive manner:

- If u is a truth value, $f(u) = u$ (by stipulation)
- If R is a binary relation between individuals in M, $f(R) = \{(f(a), f(b)) \mid (a, b) \in R\}$
- If F is a function from binary relations between individuals in M to sets of individuals in M, $f(F)$ is the corresponding function over M' defined by $f(F)(R') = S'$ iff $F(f^{-1}(R)) = f^{-1}(S')$. (Thinking of F as a relation instead, we have $f(F) = \{(f(R), f(S)) \mid F(R) = S\}$)

Definition 1. ***ISOM for arbitrary operators on domains***
An operator O on domain is ISOM iff for all domains M and all bijections $f : M \to M'$ with domain M:

$$f(O_M) = O_{f(M)}$$

Since a quantifier that is ISOM cannot refer to specific elements in a domain, this restricts their models to operations on the input set itself, like for example union or intersection, but also cardinality (or proportions of cardinalities) to determine the truth value.

To give an idea why ISOM is important for logicality, let us consider as an example a quantified statement from natural language: "some of my friends are diligent". If I were to describe a bijection f to another domain, such that "my friends" is mapped to a new set "garden hoes" and "diligent (people)" to "green (garden utensils)", then the new sentence "some garden hoes are green" needs to have the same truth value, after and before the bijection is applied the input sets. In other words, the meaning of the quantifier is independent of the specific domain of discourse, but only depends on the cardinalities of the predicates to which it is applied. If we understand logic as the study of formal reasoning, then it is clear why this is a useful property.

	Crisp Input	Fuzzy Input
Crisp Quantifier	Type I	Type II
Fuzzy Quantifier	Type III	Type IV

Table 3.1: Categorization of fuzzy quantifiers, due to [LK98]

3.3 Fuzzy Quantification

We differentiate a fuzzy quantifier by whether its input or output is two-valued (crisp) or many-valued (fuzzy). This leads to a categorisation of "types" of fuzzy quantifiers, proposed by Liu and Kerre [LK98].

Type I quantifiers therefore amount to exactly the class of quantifiers that was considered by Barwise and Cooper, with classical predicates as argument for a quantifier that only evaluates to a binary semantic value. Furthermore, the classical quantifiers ($\exists$, $\forall$) are also of this type.

Two types of this categorization are of special interest for the framework of Glöckner and are also generally referred in this thesis under the following names. *Semi-fuzzy quantifiers* refers to Type III in this scheme. They quantify over elements of crisp sets and evaluate to a truth value in the range $[0, 1]$. The term "fully" *fuzzy quantifier* refers to Type IV, where one quantifies over fuzzy sets to evaluate, as before, to a truth value in the range $[0, 1]$.

By a "quantifier" as the term is used in this thesis, we understand a function from a (crisp or fuzzy) set (or tuple of such sets) to a truth value, again possibly crisp (two valued) or fuzzy (in the range $[0, 1]$).

A *semi-fuzzy quantifier* (SFQ) is then defined as follows:

$$SFQ : \mathcal{P}(D)^k \to [0, 1] \tag{3.4}$$

We assume that the domain D is always clear from the context. The k arguments are crisp sets over the domain D. A Type IV *fuzzy quantifier* (FQ) is simply defined as:

$$FQ : \tilde{\mathcal{P}}(D)^k \to [0, 1]. \tag{3.5}$$

3.4 Glöckner's axiomatic lifting approach

The eventual goal is to model all four types of quantifiers. Since the fuzzy case can always be seen as a more general form of the crisp one, Type IV quantifiers comprise in some sense all other types of fuzzy quantification.

As mentioned before, Glöckner notes in his monograph [Glö06] that many of the early fuzzy quantifier models following Zadeh's approach are inadequate for modelling natural language expressions, such as *about half.* Another point he stresses, is that it is easy to define a Type IV directly, but difficult to give convincing justifications for it. To use the example “about half”, there are many possible models that come into mind, making it hard to determine which one might be appropriate for any particular use case. This is something that is referred to as “embarrassment of riches” by Fermüller and Roschger [FR13], where they note that even for unary fuzzy quantifiers uncountably many choices arise.

Glöckner addresses this, *in part*, by modelling the desired quantifier as a (Type III) semi-fuzzy one. This reduces the number of choices, and allows one to use operations on crisp sets, such as intersection and cardinality, to define the semantics. Only after this are these semi-fuzzy quantifiers “lifted” to the (Type IV) fuzzy case. An operator that performs such a lifting from a semi-fuzzy quantifier to produce fuzzy quantifiers is called a *quantifier fuzzification mechanism* (QFM). Formally,

Definition 2. *A* quantifier fuzzification mechanism $\mathcal{F}$ *is a function that takes as its input a semi-fuzzy quantifier (defined in 3.4) and returns a fuzzy quantifier (defined in 3.5):*

$$\mathcal{F} : SFQ \to FQ$$

The reason why this only solved the problem *partially* is that it still leaves open the issue of providing models for SFQs. An attempt at solving this can be seen in [FR13], where Fermüller and Roschger use a game-theoretic approach to provide a justification for truth functions of semi-fuzzy quantifiers.

Since there are many different ways in which a quantifier fuzzification mechanism can be defined, a list of desired properties of a QFM were also given in his monograph. These axioms, as he refers to them, focus on producing “reasonable” models of fuzzy quantifiers with respect to the original semi-fuzzy quantifier. A QFM that fulfils all of these is referred to as a *determiner fuzzification scheme* (DFS), where “determiner” is a term from linguistics that is used to describe “quantifier” as it is used in this thesis. The nod to linguistics hints at Glöckner’s goal of producing models of fuzzy quantification that are adequate to model expressions from natural language.

We proceed to give an overview of the DFS axioms ([Glö06], p. 106). All of these are explained in significantly more detail in chapter 6

- **Correct Generalisation** This property ensures that the resulting fuzzy quantifier properly subsumes the corresponding semi-fuzzy one. We first introduce a simple restriction operator that restricts the input of a fuzzy quantifier to be crisp $\mathcal{U}$. If for a QFM $\mathcal{F}$ and for all semi-fuzzy quantifiers Q, it holds that

$$\mathcal{U}(\mathcal{F}(Q)) = Q, \tag{3.6}$$

  then we say that $\mathcal{F}$ satisfies *correct generalisation.*

- **Membership Assessment** A special unary semi-fuzzy quantifier π_e is introduced. It returns 1 iff the input contains $e \in D$. The fuzzy pendant $\tilde{\pi}_e$ is defined over the characteristic function of a given fuzzy set ($\tilde{\pi}_e(F) = \mu_F(e)$). If for a QFM $\mathcal{F}$ and for all $e \in D$, it holds that

$$\mathcal{F}(\pi_e) = \tilde{\pi}_e, \tag{3.7}$$

  then we say that $\mathcal{F}$ satisfies *membership assessment.*

- **Dualisation** This is based on a new operator introduced in chapter 6. For a quantifier Q we can form the *dual* $Q\square$ of a quantifier. This is applicable in both the semi-fuzzy and the fuzzy case (we use $\tilde{\square}$ to signify the latter). If for a QFM $\mathcal{F}$ and for all semi-fuzzy quantifiers Q, it holds that

$$\mathcal{F}(Q\square) = (\mathcal{F}(Q))\tilde{\square}, \tag{3.8}$$

  then we say that $\mathcal{F}$ satisfies *dualisation.*

- **Internal Join** This is based on a union operator for fuzzy sets $\tilde{\cup}$, derived using a QFM. The construction is analogous as for the dual operator. The *internal join* takes an n-ary quantifier Q and returns the $n+1$-ary quantifier $Q\cup$, where an additional argument is *joined* with the last using the union operator. In the fuzzy case this is done using the mentioned fuzzy union operator. If for a QFM $\mathcal{F}$ and for all semi-fuzzy quantifiers Q, we have

$$\mathcal{F}(Q\cup) = (\mathcal{F}(Q))\tilde{\cup}, \tag{3.9}$$

  then we say that $\mathcal{F}$ satisfies *internal join.*

- **Preserving Monotonicity** A semi-fuzzy quantifier $Q : \mathcal{P}(D)^k \to [0,1]$ is said to be nondecreasing (resp. nonincreasing) in its i-th argument, $i \in \{1, \dots, k\}$, if

$$Q(Y_1, \dots, Y_n) \leq Q(Y_1, \dots, Y_{i-1}, Y_i', Y_{i+1}, \dots, Y_n)$$

$$(\text{ resp. } Q(Y_1, \dots, Y_n) \geq Q(Y_1, \dots, Y_{i-1}, Y_i', Y_{i+1}, \dots, Y_n) \ )$$

  whenever the involved arguments $Y_1, \dots, Y_n, Y_i' \in \mathcal{P}(D)$ satisfy $Y_i \subseteq Y_i'$. Glöckner introduces a *fuzzy inclusion operator* and based on this defines nondecreasing (resp. nonincreasing) in the fuzzy case. If for a QFM $\mathcal{F}$ and for all semi-fuzzy quantifiers Q that are nondecreasing (resp. nonincreasing) in their i-th argument, $\mathcal{F}(Q)$ is nondecreasing (resp. nonincreasing) in its i-th argument too, then we say that $\mathcal{F}$ *preserves monotonicity.*

- **Functional application** The final property is concerned with a homomorphism condition between two domains. For any mapping $f : D \to D'$ we have an associated powerset mapping $\widehat{f} : \mathcal{P}(D) \to \mathcal{P}(D)$. An *extension principle* extends f to form a mapping between fuzzy sets, $\mathcal{E}(f) : \tilde{\mathcal{P}}(D) \to \tilde{\mathcal{P}}(D')$. Glöckner introduces a mechanism to use a QFM $\mathcal{F}$ to produce a specific extension principle $\tilde{\mathcal{F}}$, s.t, $\widehat{\mathcal{F}}(f) : \tilde{\mathcal{P}}(D) \to \tilde{\mathcal{P}}(D')$. Before we can state the DFS axiom, we need to introduce some notations. For sake of brevity we write

$$(Q \circ \times_{i=1}^{n} \widehat{f})(Y_1, \ldots, Y_n) = Q(\widehat{f}(Y_1), \ldots, \widehat{f}(Y_n))$$

where $\circ$ is *functional composition* and $\times$ is product mapping, in this case applying the powerset mapping $\widehat{f}$ on all arguments.
If for a DFS $\mathcal{F}$, a semi-fuzzy quantifier $Q : \mathcal{P}(D)^n \to [0,1]$ and the mappings $f_1, \ldots, f_n$, it holds that

$$\mathcal{F}(Q \circ \times_{i=1}^{n} \widehat{f_i}) = \mathcal{F}(Q) \circ \times_{i=1}^{n} \widehat{\mathcal{F}}(f_i), \tag{3.10}$$

then we say that $\mathcal{F}$ satisfies *functional application.*

With these six properties we can then formally define determiner fuzzification schemes:

Definition 3 (Determiner Fuzzification Scheme)**.** *We are given a QFM $\mathcal{F}$. If $\mathcal{F}$ satisfies (3.6), (3.7), (3.8), (3.9), (3.10) and* preserves monotonicity*, then $\mathcal{F}$ is defined to be a determiner fuzzification scheme (DFS).*

3.5 Mathematical Fuzzy Logic

While fuzzy logic as a term was popularized by Zadeh, his approach is not exactly a "logic" in the sense of *mathematical logic.* There have been many ideas for many-valued logics by logicians, predating Zadeh. Jan ukasiewicz and Kurt Gödel are two noteworthy logicians that have worked and contributed to this subject. Zadeh himself notes a distinction between fuzzy logic "in the broad sense" and in the "narrow sense", the latter of which is clearly connected to these multi-valued logics. The first to expand and built on such a connection was Petr Hájek [Há06]. He gives a modern formulation of fuzzy logic in the narrow sense, with a focus on the analysis of properties such as soundness and completeness. For Zadeh, fuzzy logic in the narrow sense was focused on approximate inferences, whereas Hájek is focusing on what he calls "classical logical questions", such as completeness or deductive reasoning. An in-depth analysis on the differences between these two kinds of fuzzy logic can be found in [Běh08].

Hájek's basic fuzzy propositional calculus has five main design goals:

1. The real unit interval [0,1] is fixed as the standard set of truth values. The ordering $\leq$ on real numbers is used to get a "comparative notion of truth". Hájek does not explicitly exclude other structures of truth, such as only partially ordered or finite ones.

2. The logic is *truth functional*, meaning all connectives are interpreted via their respective truth functions. This allows one to interpret expressions in an inductive way, uniquely determined by the truth functions of the connectives used.

3. *Continuous t-norms* (see Definition 4) are taken as possible truth functions of conjunction. This choice is critical as it determines the rest of the logic. Hájek introduces three logics that take different t-norms for the conjunction:

$$x * y = max(0, x + y - 1) \qquad \text{(ukasiewicz t-norm)}$$
$$x *_G y = min(x, y) \qquad \text{(Gödel t-norm)}$$
$$x *_P y = max(0, x + y - 1) \qquad \text{(product t-norm)}$$

The reason why these three t-norms were chosen, beyond just being continuous, can be seen in the *Mostert Shields theorem*, see in [CHN15]. It states that all continuous t-norms can be expressed as ordinal sums of these three above.

4. The truth function of the *implication* is derived from the one for conjunction. The implication ⇒ is defined to be the residuum of the continuous t-norm $*$:

$$x \Rightarrow y = \ max\{z \mid x * z \leq y\}$$

It also holds that $x \Rightarrow y = 1$ iff $x \leq y$. In the case that $x > y$, the residua of the different t-norms are

$$x \Rightarrow y = 1 - x + y \qquad \text{(ukasiewicz)}$$
$$x \Rightarrow y = y \qquad \text{(Gödel)}$$
$$x \Rightarrow y = y/z \qquad \text{(product)}$$

5. The truth function for negation is $(-)x = x \Rightarrow 0$ (x implies falsity).

Definition 4. $\tilde{t} : [0,1] \rightarrow [0,1]$ *is called a* t-norm *if it satisfies*

- $\tilde{t}(x, 0) = 0$ *(0-element)*
- $\tilde{t}(x, 1) = x$ *(1-element)*
- $\tilde{t}(x, y) = \tilde{t}(y, x)$ *(commutativity)*
- *If* $x \leq z$, *then* $\tilde{t}(x, y) \leq \tilde{t}(z, y)$ *(monotonically nondecreasing)*
- $\tilde{t}(\tilde{t}(x, y), z) = \tilde{t}(x, \tilde{t}(y, z))$ *(associativity)*

for all x,y,z $\in [0,1]$.

Hájek proceeds to introduce for these three logics an underlying "basic logic", i.e., a Hilbert-style axiomatic system. Each of the three logics (ukasiewicz, Gödel and product logic) stands "above" this one in the sense that their Hilbert systems are obtained by adding one or more axioms to basic logic. With this as a proof system, he proceeds to analyse soundness and completeness , both in the propositional case and first-order predicate case. For the remainder, we shall refer to each logic of this framework as a mathematical fuzzy logic (MFL).

3.6 Giles' Game for ukasiewicz logic

Robin Giles proposed it both to model experiments in physics and to "provide tangible meaning" for fuzzy truth values [Gil74]. In *Giles' game for* (or $\mathcal{G}$-game for short) there are two players, called for simplicity's sake "I" and "You" and the central objects of the game are multisets of logical formulae. The connection to ukasiewicz logic will become clear in the course of this explanation. Each player can assert a formula, even multiple times. These are referred to as that player's *tenet*. So "You" can bet for a multiset of assertions $\{A_1, \ldots, A_n\}$ and "I" can bet for a corresponding multiset $\{B_1, \ldots, B_m\}$. This means a (possibly final) game state is described by both tenets

$$[A_1, \ldots, A_n \mid B_1, \ldots, B_m]$$

The game starts with an arbitrary formula, asserted as convention by "I", and proceeds by the application of rules that inductively lead to simpler and simpler formulae, until only atomic assertions are left.

What follows is a summary of the rules available in a $\mathcal{G}$-game.

- ($R_\wedge$) If I assert $F \wedge G$ then you attack by pointing either to the left or to the right sub-formula. As corresponding defence, I then have to assert either F or G, according to your choice.
- ($R_\vee$) If I assert $F \vee G$ then I have to assert either F or G at my choice.
- ($R_\rightarrow$) If I assert $F \rightarrow G$ then you may attack by asserting F, which obliges me to defend by asserting G. (Analogously if you assert $F \rightarrow G$)
- ($R_\exists$) If I assert $\exists x F(x)$ then I have to select a constant c and assert $F(c)$.
- ($R_\forall$) If I assert $\forall x F(x)$ then you attack by choosing c, and I have to defend by asserting $F(c)$.

Furthermore, we define $\neg F = F \rightarrow \bot$. The rule $R_\rightarrow$ has a *principle of limited liability.* This means that "You" can choose to not attack $F \rightarrow G$ (since it requires asserting F, which might not be desirable). In that case $F \to G$ is simply removed from the tennet of "I".

At the end of the game, each atomic assertion is seen as a random experiment and the end result is allowed to be "dispersive", meaning that it differs on repetition. For an atomic assertion B, the outcome of the random experiment is denoted by $\langle B \rangle = 1 - \pi(E_B)$ also called its *risk assignment*, where E_B is a binary experiment, and $\pi(E_B)$ signifies the probability of success. Then the final evaluation of a $\mathcal{G}$-game is defined as

$$\langle A_1, \ldots, A_n \mid B_1, \ldots, B_m \rangle = \sum_{1 \leq i \leq m} \langle B_i \rangle - \sum_{1 \leq j \leq n} \langle A_j \rangle \tag{3.11}$$

This final evaluation defines the outcome of the game. At the final game state, either "I" or "You" pays a certain amount to the other player. Since no player can fully anticipate the result, the goal of a $\mathcal{G}$-game is to reduce the *risk*. Note that a risk can be negative, meaning that "I" can expect to receive a net payment from "You".

Giles also provides a remarkable result for this game: the connection of the risk value assignment of $\mathcal{G}$-games with the truth evaluation of a mathematical fuzzy logic, namely ukasiewicz logic. The reason why ukasiewicz logic in particular is due to the fact that it is the only MFL that consists of continuous truth functions for all its connectives. **Note**: we say "$\mathcal{G}$-game for F" if the game in question starts with "I" asserting F. Also, a $\mathcal{G}$-game is y-valued for me (i.e. player "I"), if "I" can enforce a risk of at least y for the outcome of the game.

Theorem 1. *[Gil74]*
A ukasiewicz-sentence F is evaluated as $v_M(F) = x$ in an interpretation M iff every $\mathcal{G}$-game for F under risk assignment $\langle \cdot \rangle_M$ is $(1-x)$-valued for me.

This also shows how Giles' game gives a tangible meaning to fuzzy truth values via game-theoretic semantics. The truth degree of a sentence F corresponds inversely to the associated risk of the $\mathcal{G}$-game for F. This is used as a starting point by Fermüller and Roschger [FR13] to introduce further rules to Giles' games to model semi-fuzzy quantifiers, explained in more detail in Chapter 4. Baldi and Fermüller [BF18] introduce a rule for $\mathcal{G}$-game to extend semi-fuzzy quantifiers (Type III) to fuzzy (Type IV) quantifiers, thereby forming a QFM, explained in detail in Chapter 5.

CHAPTER 4

Semi-Fuzzy (Type III) Quantifiers

This chapter will focus on introducing various schemes found in the literature on fuzzy quantification for specifying semi-fuzzy quantifiers. They represent the needed building block in modelling certain natural language phrases, such as “about half”, “almost all”, “few”, etc., that deal with vague concepts. To briefly repeat the main ideas, in the framework proposed by Glöckner, a two step approach is used: first semi-fuzzy quantifiers must be provided. Then these are lifted to Type IV fuzzy quantifiers, in the Liu and Kerre categorization. This allows for the use of their cardinality as part of the semantics, something that is not available when dealing with fuzzy sets directly.

This “lifting” from semi-fuzzy to fuzzy quantifiers is not without issues and to understand the strength and deficits of the various proposals on how such a lifting can work, we need a broad range of semi-fuzzy quantifiers to test them with. What is presented here is a collection of various categories of such quantifiers and their semantic definitions.

4.1 Absolute and proportional quantifiers

The fuzzy quantification approach of Zadeh does not deal separately with Type III and Type VI quantifiers. Glöckner refers to this as the “traditional framework for fuzzy quantification ” [Glö06]. After the first pioneering work by Zadeh, many others have followed this traditional modelling framework and expanded it in various ways. The Zadeh framework refers to two classes of quantifiers. The first class are called *absolute quantifiers*. They are typically unary quantifiers and refer in some way to the cardinality of the fuzzy argument. Zadeh gives as examples: “*several, few, many, not very many, approximately five, close to ten, much larger than ten, are large number, etc.* ”. Quantifiers of the second class are *proportional quantifiers*, also called relative quantifiers in the literature. They are binary quantifiers, where the range, usually the first in our notation, restricts the scope, as explained earlier. In the context of NL such binary quantifiers play an important role. Examples for proportional quantifiers are given by Zadeh as "*most, many, a large fraction, often, once in a while, much of, etc.*".

While in the Zadeh framework these are effectively Type IV quantifiers, one can also think of them as Type III semi-fuzzy quantifiers by restricting the input to crisp sets. Indeed this has been implemented as part of this thesis, and we will see absolute and proportional quantifiers as semi-fuzzy ones used in chapter 6.

The quantifiers considered by the Zadeh framework all have the property of ISOM, described in the Preliminaries. Therefore only cardinality measures or set operations are used to evaluate the quantifiers and in particular referring to specific elements of the domain is not allowed. Both cardinality measures and operations on sets are completely straightforward in the semi-fuzzy case, working on crisp sets. It should be noted here that much (though not all) of the Zadeh framework is concerned with directly generalising such cardinality measures from the crisp to the fuzzy case and introducing various methods of doing so. We shall briefly mention the Σ-Count and OWA-method here, as such examples, but we will not go into more detail since it is not directly relevant to the topic of this thesis. The Σ-Count is a simple summation of the fuzzy membership degrees, while the OWA (order weighted average), additionally introduces a weighting based on the order.

We shall fix a class of proportional quantifiers that are defined for two crisp sets as arguments, and a *proportionality function* $\mu : [0,1] \to [0,1]$. The argument of this proportionality function measures how much of the range is contained in the scope, where 1 means the range is fully contained in the scope, and 0 indicates that their intersection is empty. The formal definition of the class of proportional quantifiers *RQ_mu* is as follows:

Definition 5. *A proportional quantifier* RQ_μ *for a mapping* $\mu : [0,1] \to [0,1]$ *is defined as*

$$RQ_\mu(A,B) = \begin{cases} 1 & \textit{if } A = \emptyset \\ \mu(\frac{|A \cap B|}{|A|}) & \textit{otherwise} \end{cases}$$

Many examples of semi-fuzzy quantifiers that are used in this thesis for the Discussion follow this scheme, restricted to the semi-fuzzy case as explained. To see how various quantifiers can be easily defined using this scheme, we will proceed with showing a few examples and the plots of their corresponding μ functions.

Definition 6. *The following is a collection of semi-fuzzy quantifiers that were defined using the proportional quantifier scheme from Definition 5. Some use the $S_{\alpha,\gamma}$ function of Zadeh, given below.*

$$S_{\alpha,\gamma}(x) = \begin{cases} 0 & \text{if } x \leq \alpha \\ 2\frac{x-\alpha}{\gamma-\alpha}^2 & \text{if } \alpha \leq x \text{ and } x \leq \frac{\alpha+\gamma}{2} \\ 1 - 2\frac{x-\gamma}{\gamma-\alpha}^2 & \text{if } x \leq \gamma \text{ and } \frac{\alpha+\gamma}{2} < x \\ 1 & \text{otherwise} \end{cases}$$

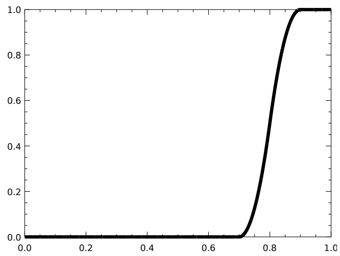
These definition are taken from [Glö06].

- ***almost all*** $= RQ_{\mu_1}$, *where* $\mu_1(x) = S_{0.7,0.9}(x)$
- ***around half*** $= RQ_{\mu_2}$, *where* $\mu_2(x) = S_{0.25,0.4}(x) - S_{0.6,0.75}(x)$
- ***half*** $= RQ_{\mu_3}$, *where* $\mu_3(x) = 2 * max(0, min(x, 1-x))$
- ***few*** $= RQ_{\mu_4}$, *where* $\mu_4(x) = 1 - S_{0.1,0.3}(x)$
- ***equals*** $= RQ_{\mu_5}$, *where* $\mu_5(x) = x$
- ***exists*** $= RQ_{\mu_6}$, *where* $\mu_6(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases}$
- ***all*** $= RQ_{\mu_7}$, *where* $\mu_7(x) = \begin{cases} 1 & \text{if } x = 1 \\ 0 & \text{otherwise} \end{cases}$

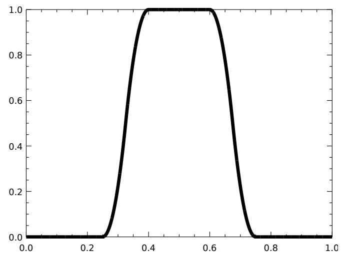
A plot of the corresponding μ mappings for the quantifiers in Definition 6 can be seen in Figure 4.1.

4.2 Blind and Deliberate Choice quantifiers

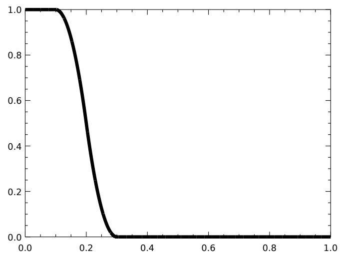
What follows is an attempt at extending the framework of Glöckner, but using a different approach with different underlying foundations. The authors Fermüller and Roschger [FR13] see the idea of starting with (Type III) semi-fuzzy quantifiers and lifting them to (Type IV) fuzzy quantifiers as appealing, but are interested in truth functional models for quantifiers, embedded into an existing MFL, namely ukasiewicz logic. Additionally, they are also interested in general principles that provide semantics for semi-fuzzy quantifiers. The choice of ukasiewicz logic is no accident, it is the only MFL where all truth functions are continuous. To achieve this embedding they utilize Giles' game, which in itself already gives "tangible meaning" to fuzzy truth degrees through its use of dispersive experiments, as introduced in the Preliminaries.



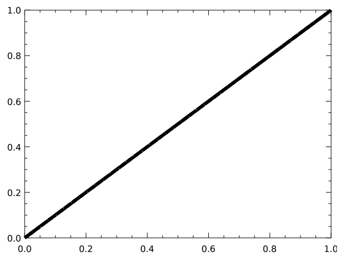
(a) almost all



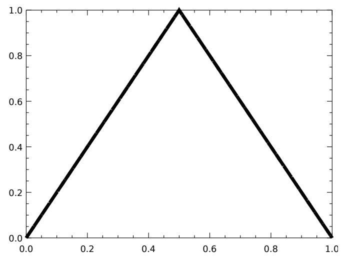
(b) around half



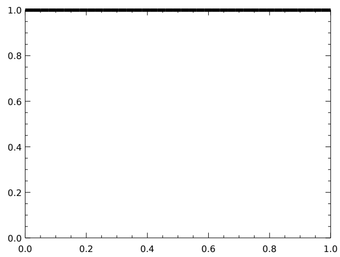
(c) few



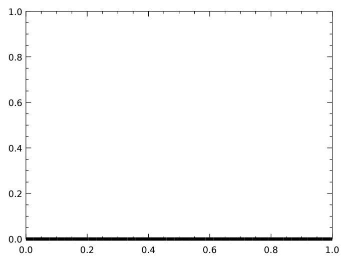
(d) equals



(e) half



(f) exists



(g) all

Figure 4.1: Plots of μ functions of proportional quantifiers

The authors extend the language of ukasiewicz logic with semi-fuzzy quantifiers. The rules of Giles' game are accordingly extended with additional rules that corresponds to these quantifiers. The first kind are called "blind choice" quantifiers, since a number of elements from the domain of quantification must be chosen by the players *before knowing* which specific elements these are. They are acting "blind". In the paper, the authors begin by introducing two such blind choice quantifiers.

We first have to define how to evaluate the proportionality of a fuzzy or crisp concept over a finite domain. This is a critical part of the rules of both blind and deliberate choice quantifiers.

Definition 7. *Let $\tilde{F}(x)$ be a formula and $v_M(\bullet)$ a corresponding evaluation function over the finite domain D. Then*

$$Prop_{\mathcal{M}}(\tilde{F}) = \sum_{c \in D} \frac{v_{\mathcal{M}}(\tilde{F}(c))}{|D|}$$

Note that we are using a slightly different notation here than in the actual paper. The reason is that in chapter 5, where we introduce a QFM based on $\mathcal{G}$-games, we also need $Prop_M$, so instead of defining it twice, with small differences, it is just stated once here and referred to later.

For any $\tilde{F}$, $Prop_{\mathcal{M}}(\tilde{F}) \in [0,1]$ is the expected value of $\tilde{F}(c)$ for a random choice of $c \in D$. We shall see later how this allows us to easily model both blind choice and deliberate choice quantifiers as proportional quantifiers. What follows is the definition of the rules for two blind choice quantifiers, L_m^k and G_m^k:

Definition 8. *($R_{L_m^k}(\tilde{F})$) If player "I" asserts $L_m^k x \tilde{F}(x)$, then player "You" may attack by betting for k random instances of $\tilde{F}(x)$, while I bets against m random instances of $\tilde{F}(x)$.*
($R_{G_m^k}(\tilde{F})$) If player "I" asserts $R_m^k x \tilde{F}(x)$, then player "You" may attack by betting against m random instances of $\tilde{F}(x)$, while I bets for k random instances of $\tilde{F}(x)$.

Before we show the result that links these rules to truth functions in ukasiewicz logic, we introduce and repeat some needed terminology. As before when introducing Giles' games, we say that "a $\mathcal{G}$-game is $(1-x)$-valued for me" to mean that player "I" can limit the expected risk to $(1-x)$. The *risk value assignment* $\langle \cdot \rangle_M$ of a $\mathcal{G}$-game with interpretation M is the assignment of the expected risk for atomic assertions, which was defined in the Preliminaries.

The authors show that the following result holds. The proof is omitted here, as it does not lie in the scope of this work.

Theorem 2. *For any interpretation M, let us extend the evaluation function v_M of ukasiewicz logic by:*

$$v_M(L_m^k x\tilde{F}(x)) = min(1, max(0, 1 + k - (m + k)Prop_{\mathcal{M}}(\tilde{F}))).$$
$$v_M(G_m^k x\tilde{F}(x)) = min(1, max(0, 1 - k + (m + k)Prop_{\mathcal{M}}(\tilde{F}))).$$

A sentence F in the language of ukasiewicz logic extended with L_m^k and G_m^k is evaluated to $v_M(F) = x$ iff every $\mathcal{G}$-game for F augmented by the rules $R_{L_m^k}$ and $R_{G_m^k}$ under risk value assignment $\langle\cdot\rangle_M$ is $(1 - x)$ valued for me.

This shows that blind choice quantifiers, as defined by the given rules, produce a truth functional model. Since ukasiewicz logic consists solely of continuous functions, these quantifiers can be easily and efficiently implemented.

These two quantifiers can be combined to form new ones. For instance H_t^s, which is defined as a combination of L_m^k and R_m^k.

$$v_M(H_t^s x\tilde{F}(x)) = min(v_m(G_{s+t}^{s-t} x\tilde{F}(x)), v_m(L_{s-t}^{s+t} x\tilde{F}(x)))$$

The next category of quantifiers allow the two players to choose the instances *while knowing* precisely which elements are picked from the domain. There is a *deliberate choice.* To understand why this affects the semantics (and leads to different risk evaluations), we consider that each player acts rationally and essentially as a perfect reasoner. The attacker will pick those instances with highest, the defender those with the lowest risk.

For deliberate choice quantifiers, the authors give just one example, though of course based on the game-based approach there are many other possible rules, and therefore concrete quantifiers, that fall in this class.

Definition 9. *($R_{\Pi_m^k}(\tilde{F})$) If player "I" asserts $\Pi_m^k x\tilde{F}(x)$ then, if "You" attacks, $k + m$ constants are randomly chosen and "I" has to pick k of those constants, while betting against the remaining n.*

The authors show the following connection with ukasiewicz logic:

Theorem 3. *For any interpretation M, let us extend the evaluation function v_M of ukasiewicz logic by:*

$$v_M(\Pi_m^k x\tilde{F}(x)) = \binom{k+m}{k}(Prop_{\mathcal{M}}(\tilde{F}))^k(1 - Prop_{\mathcal{M}}(\tilde{F}))^m$$

A sentence F in the language of ukasiewicz logic extended with Π_m^k is evaluated to $v_M(F) = x$ iff every $\mathcal{G}$-game for F augmented by the rule $R_{\Pi_m^k}$ under risk value assignment $\langle\cdot\rangle_M$ is $(1 - x)$ valued for me.

Unfortunately, this alone leads to an unsatisfactory quantifier, since $v_M(\Pi^k_m x\tilde{F}(x))$ is always below 1. This is addressed by introducing a "quantifier modifier", that essentially multiplies the truth value by a constant amount (with a ceiling of 1).

Definition 10. *($W_n(Q)x\tilde{F}(x)$) If player "I" asserts $W_n(Q)x\tilde{F}(x)$ then "You" has to place n bets against $Qx\tilde{F}(x)$, while "I" places just one bet for $Qx\tilde{F}(x)$*

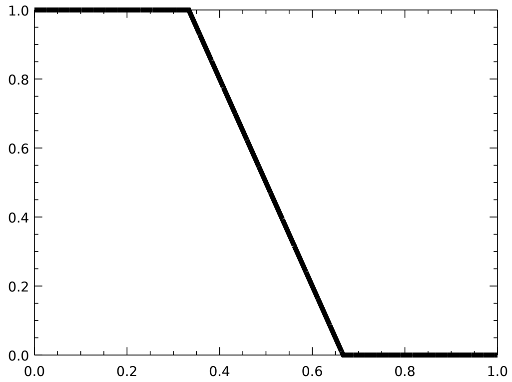
The deliberate choice quantifier Π^k_m is then to be used in combination with W_n. An example that gives a possible semantics for "about half", is $W_3(\Pi^2_2)$. For "about a third", $W_3(\Pi^1_2)$ is proposed and $W_3(\Pi^1_1)$ finally as a possible model for "very roughly half". It is interesting to note that, as opposed to blind choice quantifiers, that can only model piecewise linear functions, deliberate choice quantifiers are not restricted to those and can model non-linear increases and decreases.

Beyond the different foundations of blind and deliberate choice quantifiers, another interesting aspect that sets them apart from other models of semi-fuzzy quantifiers is the fact that they are defined, as was seen, with parameters in their definition. Therefore we can think of each of them as a class of quantifiers and each concrete assignment of k and t (or s and t for H) as a specific semi-fuzzy quantifier. For example, the authors point out how the subclass G^s_{2s}, meaning we pick for some s, $k = s$ and $m = 2s$, corresponds to various forms of "(at least) one third". By the dual definition the subclass L^{2s}_s, for example, corresponds to something along the lines of " (less than) two thirds".

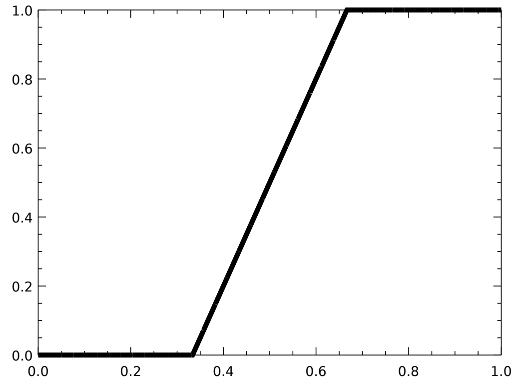
There is a connection between the semantics of both blind and deliberate choice quantifiers that are based on a measure of proportionality of a crisp set (since a classical predicate F can be seen as the set $\{c \mid v_M(F(c)) = 1, c \in D\}$) and the notion of proportional quantifiers in the Zadeh framework, even though one is a unary and the other binary. If we consider the domain not to be implicit, but an explicit crisp set, we can replace $Prop_{\mathcal{M}}(F)$ with a value $x \in [0, 1]$ which represents the proportion $\frac{|D \cap F|}{|D|}$. This way we can use these quantifiers as binary ones, by treating the domain D as the range and F as the scope. This is not a new idea, but is precisely the notion of *relativization* from the TGQ [Wes16]. This tool allows one to extend the arity of a quantifier Q by 1, to form Q_{rel}, where the new argument in Q_{rel} *relativizes* the old ones, meaning only the intersections are considered.

This has been implemented as part of this thesis and some of the examples presented in later chapters denote these quantifiers as taking two arguments. So for example $G^1_2 x(A_1(x), A_2(x))$ is equivalent with considering A_1 as the domain, and evaluating $G^1_2 x A_2(x)$. These binary versions take the truth functions defined here, and follow the scheme of proportional quantifiers given in Definition 5. To give a concrete example, the truth function of L^m_k is stated as

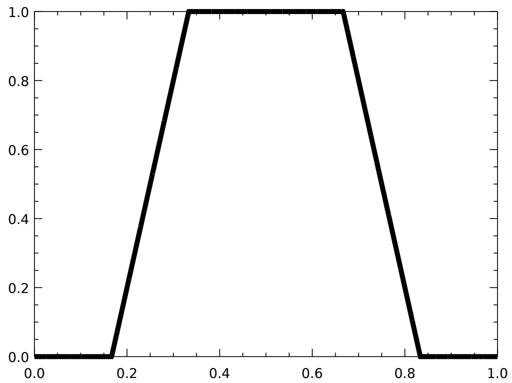
$$v_M(L^k_m x\tilde{F}(x)) = min(1, max(0, 1 + k - (m + k)Prop_{\mathcal{M}}(\tilde{F}))).$$



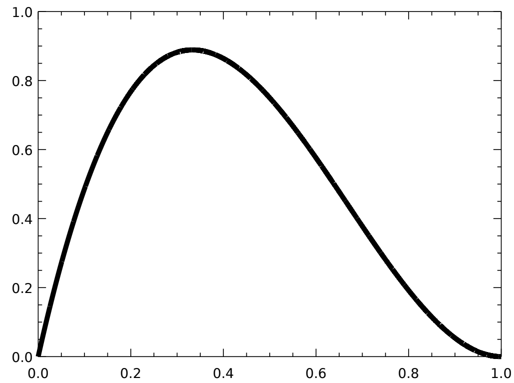
(a) L_2^1



(b) G_1^2



(c) H_1^3



(d) $W_2(\Pi_2^1)$

Figure 4.2: Some example plots of blind and deliberate choice quantifiers.

We can reformulate this as a proportionality function $\mu_{L_m^k} : [0,1] \to [0,1]$, replacing $Prop_{\mathcal{M}}(\tilde{F})$ with an argument $x \in [0,1]$:

$$\mu_{L_m^k}(x) = min(1, max(0, 1 + k - (m+k)x))$$

Then we can define the binary quantifier simply as

$$RQ_{\mu_{L_m^k}}(A,B) = \begin{cases} \mu_{L_m^k}(1) & \text{if } A = \emptyset \\ \mu_{L_m^k}(\frac{|A \cap B|}{|A|}) & \text{otherwise} \end{cases}$$

following the scheme given in Definition 5.

CHAPTER 5

Quantifier Fuzzification Mechanism Models

This chapter introduces various models found in the literature for ways to extend semi-fuzzy quantifiers to the fuzzy case. They are called quantifier fuzzification mechanisms (QFM) [Glö06]. The ultimate goal is to produce models of quantification that lie in Type IV of the Liu and Kerre categorization, so they can take fuzzy arguments as inputs and are evaluated to a fuzzy truth degree. The advantage of being able to define a semi-fuzzy quantifier, for example for the vague concept "about half", that takes crisp sets as arguments and having a mechanism that automatically "lifts" this to a Type IV quantifier should be apparent. Glöckner also limits this lifting process by a number of properties that a "reasonable" QFM should fulfil. We will compare the QFMs listed in this chapter against these properties in the subsequent chapter.

5.1 QFMs by Glöckner

What follows is a simplified account (compared with Glöckner's monograph [Glö06]) that focuses on the basic implementation of three models that Glöckner developed. In his monograph he also gives an axiomatic foundation and first sets out a number of requirements before he shows various mechanisms of extending semi-fuzzy quantifiers to the fuzzy case.

All of the fuzzification mechanisms of Glöckner are based on one central notion of reducing fuzzy sets to crisp sets, which consists of using "three-valued" sets as an intermediate step. We shall fix the notation for the domain as D, as before. A three-valued set is essentially a special case of a fuzzy set, but the membership function is only allowed to map to a set that contains 0, 0.5 and 1, i.e. $\mu_T V : D \to \{0, 0.5, 1\}$. We proceed by defining a way to map truth values in $[0, 1]$ into $\{0, 0.5, 1\}$

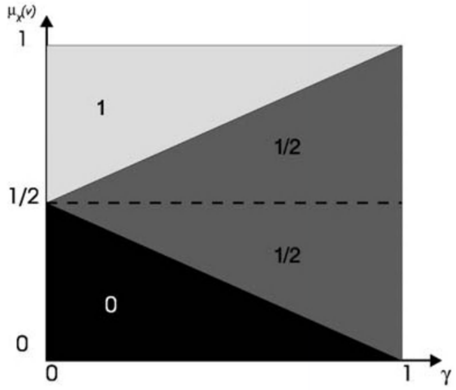


Figure 5.1: Three-valued cut as a function of γ and $\mu_{\tilde{X}}(x)$

$$three\_cut_{\gamma}(x) = \begin{cases} 1 & : x \geq 0.5 + 0.5\gamma \\ 0.5 & : 0.5 - 0.5\gamma < x < 0.5 + 0.5\gamma \\ 0 & : x \leq 0.5 - 0.5\gamma \end{cases} \tag{5.1}$$

In the special case where $\gamma = 0$, it is defined as

$$three\_cut_{0}(x) = \begin{cases} 1 & : x > 0.5 \\ 0.5 & : x = 0.5 \\ 0 & : x < 0.5 \end{cases} \tag{5.2}$$

This is extended to fuzzy sets in a natural way. For a given value of γ, a fuzzy set $\tilde{A}$ and its membership function $\mu_{\tilde{A}}$, we simply apply the $three\_cut_{\gamma}$ function to the output of $\mu_{\tilde{A}}$. So for a fuzzy set $\tilde{A}$, the three-valued cut for γ is defined by $T_{\gamma}(\tilde{A})$, whose membership function is

$$\mu_{T_{\gamma}(\tilde{A})}(c) = three\_cut(\mu_{\tilde{A}}(c)) \text{ for all } c \text{ in the domain} \tag{5.3}$$

For a given three-valued set T, there are two natural crisp approximations. The first is to focus on all elements with membership degree 1, and the second is to be more lenient and also allow for elements with degree 0.5.

$$T_{min} = \{c \in D \mid \mu_T(c) = 1\} \tag{5.4}$$

$$T_{max} = \{c \in D \mid \mu_T(c) \geq 0.5\} \tag{5.5}$$

These are the two extreme ends, so to speak. From these we define a range of possible crisp sets that are in between.

$$\mathcal{R}ange(T) = \{X \mid T_{min} \subseteq X \subseteq T_{max}\} \tag{5.6}$$

For a given three-valued T with n elements of non-zero degree, there are possibly 2^n many crisp sets in $\mathcal{R}ange(T)$ (consider the case that all n elements have membership degree 0.5).

This explains how to get crisp approximations for a single three-valued set. For more than one set, say $T_1, \ldots, T_n$, we extend this by first computing the ranges and then forming the Cartesian product $\mathcal{R}ange(T_1) \times \ldots \times \mathcal{R}ange(T_n)$. So we consider, for the crisp case, every possible combination from the elements of the range.

Another important component of this approach is the *generalised* median function. For this we first need the specific median function *med*, which operates on two comparable elements.

$$med(a, b) = \begin{cases} min(a, b) & \text{if } min(a, b) > 0.5 \\ max(a, b) & \text{if } max(a, b) < 0.5 \\ 0.5 & \text{otherwise} \end{cases} \tag{5.7}$$

med is chosen for its properties of being an operator to compute a mean that is associative, idempotent and commutative. It is then generalised to deal with arbitrary subsets $X \subseteq [0, 1]$ of the set of truth values $[0, 1]$. This new generalised operator is also called *med*, in the hope that it is clear from context which version is meant.

Definition 11. *The* generalised median function *med is defined as follows:*

$$med(X) = med(\inf(X), \sup(X))$$

Glöckner introduces a number of *classes* of QFMs. Each of these is a general schema that allows one to get different concrete models of QFMs by using different functions. While the monograph lists more than the two listed below, we pick three models from different classes to represent Glöckner style QFMs. The choice of models follows [DRSV14].

5.1.1 The Class of $\mathcal{M}_\mathcal{B}$-QFMs

The basis of this class of QFMs is the idea of using three-valued cuts to produce ranges of crisp representations, and subsequently using *med* to aggregate the values of the range. The three-cut needs a “cautiousness parameter” γ, so we first assume this parameter is fixed. This produces the following QFM:

Definition 12. *Taken from Def. 7.18 ([Glö06], p. 190)*
Let Q be a semi-fuzzy quantifier , $\gamma \in [0,1]$ and $\tilde{A}$ a fuzzy set. Then $Q_\gamma : \tilde{\mathcal{P}}(D) \to [0,1]$ is defined as:

$$Q_\gamma(\tilde{A}) = med\{Q(B) : B \in \mathcal{R}ange(T_\gamma(\tilde{A}))\},$$

where med, $\mathcal{R}ange$, T_γ are defined as in Definition 11, (5.6), (5.3).

The class $\mathcal{M}_\mathcal{B}$ is defined on a given aggregation method $\mathcal{B}$ which collects values of Q_γ for any possible choice of γ and maps them to a single truth value. For the definition of this class, we also introduce the set $\mathbb{B}$, a collection of all mappings $f : [0,1] \to [0,1]$ that are either nonincreasing, nondecreasing or return constantly $1/2$.

Definition 13. *Let $\mathcal{B} : \mathbb{B} \to [0,1]$ be given. The class of QFMs $\mathcal{M}_\mathcal{B}$ is defined by*

$$\mathcal{M}_\mathcal{B}(Q)(X) = \mathcal{B}((Q_\gamma(X))_{\gamma \in [0,1]}),$$

for all semi-fuzzy quantifiers $Q : \mathcal{P}(D) \to [0,1]$ and $X \in \tilde{\mathcal{P}}(D)$.

The first QFM of Glöckner, which we will introduce, is called simply $\mathcal{M}$. It consists of a simple integral over the range [0,1]:

Definition 14. *Taken from Def. 7.22 ([Glö06], p.192)*
Let $\tilde{A}$ be a fuzzy set and Q a semi-fuzzy quantifier. The QFM $\mathcal{M} : \tilde{\mathcal{P}}(D) \to [0,1]$ of the class $\mathcal{M}_\mathcal{B}$ is defined as

$$\mathcal{M}(Q)(\tilde{A}) = \mathcal{M}_{\int_0^1}(Q)(\tilde{A})$$

Unpacking the definition of $\mathcal{M}_\mathcal{B}$, we can also state it as:

$$\mathcal{M}(Q)(\tilde{A}) = \int_0^1 Q_\gamma(\tilde{A})\, d\gamma,$$

where Q_γ is taken from Definition 12.

It shall be noted here, that it is sufficient to sample over only finite possibilities of values for γ, based on $(\tilde{A})$. Assuming that the domain D is finite, a three-cut can only produce finitely many different crisp ranges based on the truth degrees occurring in the fuzzy set. For every $i \in [0,1]$, there is only one "choice" by the *three_cut* operator, either the value is mapped to 1 or 0.5, if it is above 0.5, or it is mapped to 0.5 or 0, it is below. (The values of 0,0.5 and 1 are preserved, so multiple applications of T_γ on the same set do not produce different outcomes). So for a fuzzy set with at most n different truth degrees we need only sample n values of γ. This finite sampling is explained in more detail in chapter 7.

The next QFM falls into the same class as $\mathcal{M}$, but with a more simplified construction: this simplification allows the QFMs *to be only defined on nonincreasing input.* In other words, in cases where Q_γ is increasing, their behavior is not defined, does not need to be well behaved. We will see in chapter 6 how this plays a role.

The next QFM called $\mathcal{M}_{CX}$ is defined as follows:

Definition 15. *Taken from Def 7.56 ([Glö06], p. 201)*
The QFM $\mathcal{M}_{CX} : SFQ \to FQ$ is defined over an aggregation on the family of quantifiers $Q_\gamma(\tilde{A})_{\gamma\in[0,1]}$.

$$\mathcal{M}_{CX}(f) = \sup\{min(x, f(x)) \mid x \in [0,1]\}$$

where $f := Q_\gamma(\tilde{A})_{\gamma\in[0,1]}$ and f is nonincreasing.
Less abstractly, we are given a fuzzy set $\tilde{A}$ and define $\mathcal{M}_{CX}$ as

$$\mathcal{M}_{CX}(Q)(\tilde{A}) = \sup\{min(\gamma, Q_\gamma(\tilde{A})) \mid \gamma \in [0,1]\}$$

5.1.2 The Class of $\mathcal{F}_\xi$-QFMs

The next class of QFMs that we will cover here, is based on a generalisation of the previous one. Instead of looking on aggregations on the set of mappings from $[0,1] \to [0,1]$ (in particular Q_γ), we rewrite Q_γ as follows:

$$\begin{aligned} Q_\gamma(\tilde{A}) &= med\{Q(B) : B \in \mathcal{R}ange(T_\gamma(\tilde{A}))\} \\ &= med\{\sup\{Q(B) : B \in \mathcal{R}ange(T_\gamma(\tilde{A}))\}, \inf\{Q(B) : B \in \mathcal{R}ange(T_\gamma(\tilde{A}))\}\} \end{aligned}$$

To understand why this rewritten form of $Q_\gamma(\tilde{A})$ is valid, consider the generalised median function: it simply takes the infimum and supremum of its argument and uses the specific median function *med* on ordered pairs.

Essentially we are only interested in two elements of the underlying crisp representation. These are of particular interest to the next class of QFMs, so we shall introduce a shorter notation:

$$\top_{Q,\tilde{A}}(\gamma) = \max_{\gamma\in[0,1]}\{Q_\gamma(\tilde{A})\} \tag{5.8}$$

$$\bot_{Q,\tilde{A}}(\gamma) = \min_{\gamma\in[0,1]}\{Q_\gamma(\tilde{A})\} \tag{5.9}$$

Both of these are functions of the type $[0,1] \to [0,1]$, where $\top_{Q,\tilde{A}}$ is non-decreasing and $\bot_{Q,\tilde{A}}$ is non-increasing. Based on these two, Glöckner defines the class $\mathcal{F}_\xi$. The QFMs in this class operate on the pair $(\top_{Q,\tilde{A}}, \bot_{Q,\tilde{A}})$ and maps this pair to a truth value.

Definition 16. *We are given a fuzzy set* $\tilde{A}$ *and a semi-fuzzy quantifier* Q*. For every mapping* $\xi : (\top_{Q,\tilde{A}}, \bot_{Q,\tilde{A}}) \to [0,1]$*, the QFM* $\mathcal{F}_\xi$ *is defined by*

$$\mathcal{F}_\xi(Q)(\tilde{A}) = \xi(\top_{Q,\tilde{A}}, \bot_{Q,\tilde{A}}),$$

for all semi-fuzzy quantifiers $Q : \mathcal{P}(D) \to [0,1]$ *and all fuzzy sets* $\tilde{A} \in \tilde{\mathcal{P}}(D)$*.*

The example QFM of this class that is mentioned here is $\mathcal{F}_{owa}$. It is named since it is, as shown by Glöckner, a generalisation of the previously mentioned OWA method [Yag98] in the framework of Zadeh.

Definition 17. *Taken from Def 8.13 ([Glö06], p. 226)*
Let $\tilde{A}$ *be a fuzzy set and* Q *a semi-fuzzy quantifier. The QFM* $\mathcal{F}_{owa} : SFQ \to FQ$ *is stated as*

$$\mathcal{F}_{owa}(Q)(\tilde{A}) = \frac{1}{2}\int_0^1 \top_{Q,\tilde{A}}(\gamma)\ d\,\gamma + \frac{1}{2}\int_0^1 \bot_{Q,\tilde{A}}(\gamma)\ d\,\gamma,$$

where $\top_{Q,\tilde{A}}(\gamma)$ *and* $\bot_{Q,\tilde{A}}(\gamma)$ *are defined as in 5.8 and 5.9.*

All the QFMs presented here, $\mathcal{M}$, $\mathcal{M}_{CX}$ and $\mathcal{F}_{owa}$ fulfil a series of properties (or axioms) that Glöckner thought of as necessary to truly capture the use of quantifiers in natural language. As mentioned earlier, Glöckner called such QFMs, determiner fuzzification schemes. While their applicability in all contexts where fuzzy quantifiers might be useful is perhaps not obvious, it nonetheless presents an interesting analysis to see if QFMs that were defined after Glöckner's monograph fulfil or break with some of them.

A final remark on the QFMs shown here: for sake of simplicity, and to hopefully make the (not exactly easy to read) work of Glöckner more approachable for an audience that is new to this topic, all the QFMs were defined using, where applicable, the unary case. So instead of $Q_\gamma(A_1, \ldots, A_n)$ just $Q_\gamma(A)$ was written. This does not mean that any of the mechanisms shown here are restricted to the unary quantifiers. $\mathcal{M}$, $\mathcal{M}_{CX}$ and $\mathcal{F}_{owa}$ take n-ary semi-fuzzy quantifiers and produce n-ary fuzzy quantifiers.

5.2 Probabilistic QFMs by Díaz-Hermida et al.

The framework of Díaz-Hermida et al. [DHBCB04] is interesting in that they first begin by a justification of fuzzy semantics themselves. They propose that for each fuzzy concept (say for example “rich”), we think of a random experiment. We have a set of witnesses, called *voters*, that decide, for a any element in our domain if it fulfils the property or not. Essentially each voter sees the concepts as a crisp one, but they are allowed to differ in their specification of this crisp set. The fuzzy membership degree is then seen as the expected value of this outcome. 0 means that no voter thinks it is part of the fuzzy concept, 0.5 indicates that half of the voters would agree, and half disagree.

While this unique perspective on fuzzy interpretations is not directly related to elevating semi-fuzzy quantifiers to the fuzzy case, it does show a way of extending crisp sets to fuzzy sets. This is essentially determined by the choice of the voters. The idea is then to sample various alpha cuts of fuzzy sets, but under a prespecified probability density function P. It is determined by the relation of the voters "specificity", essentially how strict the voters are in choosing whether an element of the domain belongs to the fuzzy concept or not. This connection between voting behavior and the corresponding probability density function is not explicitly given, for example in a formula,but instead left abstract.

Definition 18. *We are given a semi-fuzzy quantifier Q and fuzzy sets $\tilde{A}_1, \dots, \tilde{A}_n$. Additionally a probability density function P representing the voters specificity respective to a choice of alpha-cuts. We then define the QFM F^P as,*

$$F^P(Q)(\tilde{A}_1, \dots, \tilde{A}_n) = \int_0^1 \cdots \int_0^1 Q(A_{1_{\alpha_1}}, \dots, A_{n_{\alpha_n}}) P(\alpha_1, \dots, \alpha_n) \ d\alpha_1, \dots, \ d\alpha_n.$$

This is the general framework in how all such QFMs can be defined, according to the underlying voting-based model of the fuzzy set. Díaz-Hermida et al. present in their their paper three exemplary choices of this framework.

The critical part here is clearly the definition of the probability density function P. Essentially its function is to give a weighting on the possible alpha-cuts based on the underlying voting behavior. Without any restrictions on this voting behavior of the individual voters, this is rather complex task. In their paper Díaz-Hermida et al. present instead three simplified scenarios that make assumption about the voting process to make it easier to define P.

The first is the *Maximum dependency model* (MD), in which we assume that all voters agree exactly on the choice of alpha-cut, and furthermore where the choice of alpha-cut of one fuzzy set determines all the others. Essentially in this scenario we do not have to produce multiple values for each alpha cut, instead just fixing one for each fuzzy set. Formally this looks as follows:

Definition 19. *We are given a semi-fuzzy quantifier Q and fuzzy sets $\tilde{A}_1, \dots, \tilde{A}_n$. We define the QFM F^{MD} as:*

$$F^{MD}(Q)(\tilde{A}_1, \dots, \tilde{A}_n) = \int_0^1 Q(A_{1_\alpha}, \dots, A_{n_\alpha}) \ d\alpha$$

It is noticeable that in this simplified case, P does not appear, since for any α, we have that $P(\alpha, \dots, \alpha) = 1$. As before for the QFMs from Glöckner, we note that we can replace the integral with a sum over just a finite sample of truth degrees, namely those that appear in all fuzzy sets $(\tilde{A}_1, \dots, \tilde{A}_n)$. This is clearly a finite amount, if we also assume the underlying domain to be finite. The reason why this is equal to the integral over all possible values, is that the alpha cuts used to get crisp sets can, for a finite domain, only produce finitely many crisp approximations. We shall omit these remarks from here on out, and just use the integral definitions to have a consistent notation, while understanding that an efficient implementation should prefer to work with sums.

The next QFM is the *Independence model* (I), where we do not require to have the same specificity for all fuzzy notions. As before, however, we still have that the probability density function P is exactly 1 for each tuple of choices for the alpha cuts. We proceed to define F^I.

Definition 20. *We are given a semi-fuzzy quantifier Q and fuzzy sets $\tilde{A}_1, \dots, \tilde{A}_n$. We define the QFM F^I as:*

$$F^I(Q)(\tilde{A}_1, \dots, \tilde{A}_n) = \int_0^1 \dots \int_0^1 Q(A_{1_{\alpha_1}}, \dots, A_{n_{\alpha_n}}) \ d\alpha_1 \dots \ d\alpha_n$$

The next concrete model presented by Díaz-Hermida et al. is the *Approximate dependence model*. Here we restrict the voters to be "approximately" equally specific for each fuzzy argument. So the choice of one alpha cut, will have an affect on the choice of the others, and this should be expressed by the probability density function. To give a concrete example on how this looks like in the binary case, the authors present the following function.

$$P^{AD}(\alpha_1, \alpha_2) = \begin{cases} h_1 - min(h_1, h_1 \frac{|\alpha_1 - \alpha_2|}{\delta}) & \text{if } 0 \leq \alpha_2 \leq \delta \\ \frac{1}{\delta} - min(\frac{1}{\delta}, \frac{|\alpha_1 - \alpha_2|}{\delta^2}) & \text{if } \delta \leq \alpha_2 \leq 1 - \delta \\ h_2 - min(h_2, h_2 \frac{|\alpha_1 - \alpha_2|}{\delta}) & \text{if } 1 - \delta \leq \alpha_2 \leq 1 \end{cases} \tag{5.10}$$

$$h_1 = - \, (2/(\alpha_2^2 - 2\alpha_2\delta - \delta^2)) \tag{5.11}$$

$$h_2 = (2/(1 - 2\alpha_2\delta + \alpha_2^2 + \delta^2))\delta \tag{5.12}$$

where $0 \leq \delta < 0.5$ is a "parameter of flexibility" in the definition.

Then the definition of F^{AD} in the binary case (consider that P^{AD} is only defined there):

$$F^{AD}(Q)(\tilde{A}_1, \tilde{A}_2) = \int_{a1}^{b1} \int_{a2}^{b2} Q(A_{1_{\alpha_1}}, A_{2_{\alpha_2}}) P^{AD}(\alpha_1, \alpha_2) \ \mathrm{d}\alpha_1 \ \mathrm{d}\alpha_2 \tag{5.13}$$

A somewhat simpler definition of F^{AD} that involves performing an interpolation of F^{MD} and F^I is given by the authors, without a proof. This is also not bounded by argument size and does not require to define a probability density function, making it at least conceptually easier.

Definition 21. *We are given a semi-fuzzy quantifier* Q *and fuzzy sets* $\tilde{A}_1, \ldots, \tilde{A}_n$. *We define the QFM* F^{AD} *as:*

$$F^{AD}(Q)(\tilde{A}_1, \ldots, \tilde{A}_n) = \tau F^{MD}(Q)(\tilde{A}_1, \ldots, \tilde{A}_n) + (1 - \tau)F^{I}(Q)(\tilde{A}_1, \ldots, \tilde{A}_n)$$

where $\tau \in [0, 1]$ *is the parameter that defines the interpolation. We use* $\tau = 1/2$ *for Chapter 6, where we discuss the properties of various QFMs.*

The final QFM of this category is $\mathcal{F}_A$, stated in [DHLBB05]. It is using a different basis of probabilities and in fact is not voting-based. Instead it introduces the notion of how probable it is that a crisp set is a representative of a fuzzy set:

Definition 22. *We assume the domain* D *is given. For a fuzzy set* $\tilde{Y}$ *and and an arbitrary crisp set* X*, we define the* representation probability

$$P(Representative_{\tilde{X}} = Y) = \prod_{e \in Y} \mu_{\tilde{X}}(e) \prod_{e \in D \setminus Y} (1 - \mu_{\tilde{X}}(e))$$

For simplicity, we will use $m_{\tilde{X}}(Y)$ *instead of* $P(Representative_{\tilde{X}} = Y)$.

Under this definition, a fuzzy set can only have a crisp representation with representation probability of 1 or 0, if all of its element have binary membership degrees (1 or 0). The QFM $\mathcal{F}^A$ is then defined formally by checking for each crisp set in the power set, what its representation probability is for the respective fuzzy argument and multiplying that with the result of the semi-fuzzy argument:

Definition 23. *We are given a semi-fuzzy quantifier* Q. *We define the QFM* $\mathcal{F}^A$ *as follows*

$$\mathcal{F}^A(Q)(\tilde{X}_1, \ldots, \tilde{X}_n) = \sum_{Y_1 \in \mathcal{P}(D)} \cdots \sum_{Y_n \in \mathcal{P}(D)} m_{\tilde{X}_1}(Y_1) \ldots m_{\tilde{X}_n}(Y_n) Q(Y_1, \ldots, Y_n)$$

for $\tilde{X}_1, \ldots, \tilde{X}_n \in \tilde{\mathcal{P}}(D)$.

Clearly a naive implementation of this QFM must be exponential in runtime, since there are $2^{|D|}$ crisp sets for a domain D to consider. Díaz-Hermida et al. define an equivalent way to compute $\mathcal{F}^A$ that is computable in polynomial time. It restricts the input to *quantitative* quantifiers. A quantifier is quantitative, if it can be defined using only the cardinalities of its arguments (and their boolean combinations). This property is connected to ISOM, defined earlier, as any quantitative quantifier must be resilient to isomorphisms of the domain. The probabilities in this case can also be defined over cardinalities. This means that it is not necessary to consider all of $\mathcal{P}(D)$, but instead just the cardinalities of the subsets of $\mathcal{P}(D)$. The implementation of $P(\text{Card}_X = j)$ is also given in [DHLBB05], it can be computed in time $|D|^2$. Unfortunately, the authors only state it for the unary case. Already in the binary case, it becomes necessary for a quantitative quantifier to consider boolean combinations (such as intersection) of its arguments. A more fine grained probability partitioning is also required: $P(\text{Card}_{X_1} = j, \text{Card}_{X_1 \cap X_2} = j)$. No implementation for such a probability is given. This restricts the use of $\mathcal{F}^A$ for the Discussion chapter, to unary quantifiers, or very small domains even for the binary case.

5.3 Representation-level based QFM by Sánchez et al.

Sánchez et al. use the notion of *level representation* [SDV11] (LR) to map fuzzy sets to crisp sets. In some sense this can be thought of as generalising the concept of alpha cuts. As with alpha cuts, we think of an assignment from $[0,1]$ to $\mathcal{P}(X)$, subsets of a domain X. We think of each concrete $\alpha \in [0,1]$ as one "level". So in essence, instead of just one alpha-cut we have a set of them. Formally,

Definition 24. *A level representation is a pair (Λ, ρ), where $\Lambda = \{\alpha_1, \ldots, \alpha_m\}$ is a set of levels and ρ is a function. Furthermore $1 = \alpha_1 > \alpha_2 > \cdots > \alpha_m > \alpha_{m+1} = 0$, $m \geq 1$. ρ is defined as*

$$\rho : \Lambda \to \mathcal{P}(X)$$

where X is specified domain.

The authors point out two unique properties of their model. First, the levels do not have to be nested. So for $\alpha_i, \alpha_{i+1} \in \Lambda$, such that $\alpha_i < \alpha_{i+1}$ we cannot assume that $\rho(\alpha_i) \subseteq \rho(\alpha_{i+1})$, which holds for simple alpha cuts. The second is, that LR gives a direct generalisation of crisp operations to fuzzy set, such as union, intersection, etc, by directly applying them for each level independently.

Before we can introduce the QFM, we need some further notation and concepts.

For a given fuzzy concept F, represented by (Λ_F, ρ_F), we define the set of crisp approximations of F as

$$\Omega_F = \{\rho_F(\alpha) \mid \alpha \in \Lambda_F\}$$

We also want ρ to give us a result for all $\alpha \in [0,1]$, even those outside of Λ. To accomplish this, ρ should simply "round down" to the nearest value in Λ. Let $\alpha \in [0,1]$, $\alpha \notin \Lambda$ and $\alpha_i, \alpha_{i+1} \in \Lambda$ such that $\alpha_i \leq \alpha < \alpha_{i+1}$. Then

$$\rho(\alpha) = \rho(\alpha_i)$$

Sánchez et al. proceed to define for a LR a probability distribution that assigns to a crisp set, how likely it is to occur in it.

$$m(Y) = \sum_{\alpha_i \mid Y = \rho(\alpha_i)} \alpha_i - \alpha_{i+1}$$

The QFM based on LR proceeds by two steps. For a given n-ary semi-fuzzy quantifier we evaluate it *levelwise*, meaning we collect the crisp evaluations, and the numerically sum them up.

First we define the levelwise evaluation.

Definition 25. *Let Q be a n-ary semi-fuzzy quantifier on domain X. Let $A_1, \dots, A_n$ be fuzzy sets represented by levels as $(\Lambda_{A_1}, \rho_{A_1}), \dots, (\Lambda_{A_n}, \rho_{A_n})$.*
The "evaluation" $E \equiv Q(A_1, \dots, A_n)$ is defined by the level representation (Λ_E, ρ_E), where

$$\Lambda_E = \bigcup_{i=1}^{n} \Lambda_{A_i}$$

and $\forall \alpha \in \Lambda_E$,

$$\rho_E(\alpha) = P(\rho_{A_1}(\alpha), \dots, \rho_{A_n}(\alpha))$$

Finally we have to compute a "numerical summary" $S(E)$ for the evaluation (Λ_E, ρ_E).

Definition 26. *We are given a semi-fuzzy quantifier Q and fuzzy sets $\tilde{A}_1, \dots \tilde{A}_n$*

$$S(Q)(\tilde{A}_1, \dots \tilde{A}_n) = S(E) = \sum_{\beta \in \Omega_E} m_E(\beta) \cdot \beta$$

where E is the evaluation from Definition 25.

5.4 Closeness-based QFM by Baldi and Fermüller

Like the blind and deliberate choice quantifiers, this proposed QFM [BF18] is also based on the game-based approach of *Giles' game for ukasiewicz logic*. A brief account of how this form of evaluation game works was given earlier and shall not be repeated here. It should be noted though, that a remarkable feature of this game-based approach is that is directly embeddable in ukasiewicz logic, which gives direct access to the model theoretic and proof theoretic tools of MFL.

An interesting note here is that Baldi and Fermüller omit non-unary quantifiers. This is their initial presentation of their work on QFMs and they presented it the unary case first, to avoid intensional range and scope dependency problems which already arise in the binary case. See section 8.1.1 for more on this issue. To have a clearer basis of comparison for the chapter 6, this thesis proposes a generalisation of their QFM to also cover arbitrary n-ary quantifiers. We explain this generalisation at the end of this chapter.

This QFM needs a way of approximating fuzzy predicates as "crisp" ones by using the following *precisification*: for a fuzzy predicate F we pick a $c \in D$ acting as threshold, then $F^c(x) = \Delta(F(c) \to F(x))$ is the precisification with threshold c. Δ is a special function that performs the following:

$$v_{\mathcal{M}}(\Delta(\varphi)) = \begin{cases} 1 & \text{if } v_{\mathcal{M}}(\varphi) = 1 \\ 0 & \text{otherwise} \end{cases}$$

Therefore, $v_{\mathcal{M}}(\Delta(F(c) \to F(x))) = 1$ iff $v_{\mathcal{M}}(F(x)) \geq v_{\mathcal{M}}(F(c))$, and $v_{\mathcal{M}}(\Delta(F(c) \to F(x))) = 0$ otherwise. This corresponds to performing an alpha-cut with $\alpha = v_M(F(c))$. Additionally $F^{\top}(x) = \Delta(F(x))$, where $F^{\top}$ means using $\top$ as the threshold.

As mentioned before, in Giles' game we have two players, referred to as "I" and "You". The basic idea is stated by the authors as this: we assume that at some point during the game, player "I" has in its tenet the fuzzy quantified sentence $\tilde{Q}xF(x)$, where $\tilde{Q}$ is based on a semi-fuzzy quantifier Q. If this is attacked by "You", then, informally speaking, "I" has the following two-step defence

1. "I" picks an element $c \in D \cup \{\top\}$ anged adds $QxF^c(x)$ to his tenet
2. "I" has to state that the precisification F^c is "close" to the original F.

So player "I" has two goals: choose a precisification that maximizes the truth value of the semi-fuzzy quantified sentence, but also is as "close" as possible to the original fuzzy predicate. The closeness measure is defined by the authors as first picking a random element c from the domain (extended by $\top$) and to be expressed as the ukasiewicz sentence $F(x) \leftrightarrow F^c(x)$. The expected value of this sentence corresponds to $Prop_{\mathcal{M}}(F \leftrightarrow F^c)$. The last point is the reason they named this approach "closeness-based".

We proceed to state the game-based rule for the QFM.

Definition 27. *($R^{Cl}_{\tilde{Q}}$) If "I" asserts $\tilde{Q}xF(x)$ and "You" attacks, "I" adds $\tilde{Q}xF^c(x)$ to his tenet, where $c \in D \cup \{\top\}$. An element d is then randomly chosen, and "You" can then choose between the following:*

1. *"You" adds $F^c(d)$ to his tenet, thereby forcing "I" to add $F(d)$ to his tenet*
2. *"You" adds $F(d)$ to his tenet, thereby forcing "I" to add $F^c(d)$ to his tenet*

For this rule, the authors have shown the following result.

Theorem 4. *For any interpretation M, let us extend the evaluation function v_M of ukasiewicz logic by:*

$$v_{\mathcal{M}}(Cl(QxF(x))) = \sup_{c \in D \cup \{\top\}} (max\{0, Prop_{\mathcal{M}}(F^c \leftrightarrow F) + v_{\mathcal{M}}(QxF^c(x)) - 1\})$$

A sentence F in the language of ukasiewicz logic extended with Q is evaluated to $v_M(F) = x$ iff every $\mathcal{G}$-game for F augmented by the rule $R^{Cl}_{\tilde{Q}}$ under risk value assignment $\langle \cdot \rangle_M$ is $(1 - x)$ valued for me.

The need to explicitly include $\top$ stems from the fact that none of our domain elements $c \in D$ might evaluate to $v_{\mathcal{M}}(F(c)) = 1$, but by definition $v_{\mathcal{M}}(F(\top)) = 1$. The alpha cut at 1 is needed to cover every possible precisification. We will see in chapter 6 that this is indeed needed to satisfy one of the DFS axioms.

In addition to this general version of Cl, there are two special versions that are also presented. These are given to better model quantifiers that are nonincreasing or nondecreasing. For these monotonic quantifiers, the $\leftrightarrow$ is replaced with $\rightarrow$ (in the increasing case) and $\leftarrow$ (in the decreasing case).

This is the form in which Cl is presented by its authors and clearly restricted to unary quantifiers, as we only perform one alpha cut on the one possible fuzzy argument. This makes it stand out from all other mechanisms so far, and while for some of the comparisons in the chapter 6 we will consider exactly such unary quantifiers, it was deemed too restrictive for the purposes of this thesis. Therefore a slightly altered version is presented here, that coincides with the original in the unary case, but extends it to deal with arbitrary arguments. We also use a notation that is more in line with previous QFMs.

For this generalisation to deal with n-ary arguments, we note that the critical aspect is dealing with $Prop_{\mathcal{M}}(\tilde{F}^c \circ \tilde{F})$, where $\circ$ depends on the monotonicity. Instead of just one fuzzy set to consider, it must be applied to many. The first method for this is to compute $Prop_{\mathcal{M}}(\tilde{F_i}^{c_i} \circ \tilde{F_i})$ for each fuzzy set independently and aggregating the result *externally* over a t-norm. The other method is to form, in essence, an average of all $\tilde{F}^c \circ \tilde{F}$ pairs by using a t-norm and computing $Prop_{\mathcal{M}}$ just once for this average fuzzy set. This is the *internal* one. Additionally, in both cases we can choose from one of the three t-norms of MFL, defined in the Preliminaries. This leads to six alternatives altogether.

We proceed to define Cl_{ext}:

Definition 28. *We are given a semi-fuzzy quantifier* $Q : \mathcal{P}(D)^n \rightarrow [0,1]$*, and fuzzy arguments* $\tilde{A}_1, \ldots, \tilde{A}_n \in \tilde{\mathcal{P}}(D)$*. The QFM* Cl_{ext} *is defined as*

$$\begin{aligned} &Cl_{ext}(Q)(\tilde{A}_1, \ldots, \tilde{A}_n) = \\ &\quad \sup_{c_1 \in D \cup \{\top\}} \ldots \sup_{c_n \in D \cup \{\top\}} (max\{0, (Prop_{\mathcal{M}}(\tilde{A_1}^{c_1} \circ_1 \tilde{A}_1) * \cdots * Prop_{\mathcal{M}}(\tilde{A_n}^{c_n} \circ_n \tilde{A}_n)) \\ &\qquad\qquad + v_{\mathcal{M}}(Qx\tilde{A_1}^{c_1}, \ldots, \tilde{A_n}^{c_n}) - 1\}) \end{aligned}$$

where $*$ *is either* $*$, $*_G$ *or* $*_P$ *and* $\circ_i$ *is either* $\rightarrow$, $\leftarrow$ *or* $\leftrightarrow$ *depending on the monotonicity of the i-th argument. To make it clear which t-norm is being used, the symbol , G or P for, respectively, ukasiewicz, Gödel or Product is to be used (eg.* Cl_{ext}*)*

Before we can state the definition of the next mechanism, we explain how we define the combination of two fuzzy sets F_1,F_2 via a t-norm $*$.

Definition 29. *For two fuzzy sets F_1, F_2 in the domain D and a given t-norm $*$, the* combination under $*$ $(F_1 * F_2)$ *is*

$$(F_1 * F_2)(c) = F_1(c) * F_2(c) \quad \forall c \in D$$

The *internal* method Cl_{int} is stated below.

Definition 30. *We are given a semi-fuzzy quantifier $Q : \mathcal{P}(D)^n \to [0,1]$, and fuzzy arguments $\tilde{A}_1, \ldots, \tilde{A}_n \in \tilde{\mathcal{P}}(D)$. The QFM Cl_{int}*

$$Cl_{int}(Q)(\tilde{A}_1, \ldots, \tilde{A}_n) = \sup_{c_1 \in D \cup \{\top\}} \ldots \sup_{c_n \in D \cup \{\top\}} (max\{0, (Prop_{\mathcal{M}}((\tilde{A_1}^{c_1} \circ_1 \tilde{A}_1) * \cdots * (\tilde{A_n}^{c_n} \circ_n \tilde{A}_n)) + v_{\mathcal{M}}(Qx\tilde{A_1}^{c_1}, \ldots, \tilde{A_n}^{c_n}) - 1\})$$

where $*$ is either $*$, $*_G$ or $*_P$ and $\circ_i$ is either $\to$, $\leftarrow$ or $\leftrightarrow$ depending on the monotonicity of the i-th argument. To make it clear which t-norm is being used, the symbol , G or P for, respectively, ukasiewicz, Gödel or Product is to be used (eg. Cl_{int})

CHAPTER 6

Discussion

The aim of this chapter is to characterize the various proposed QFMs in the literature. Each of the presented models follows the approach of lifting semi-fuzzy quantifiers to fuzzy quantifiers from Glöckner, but the various authors had different motivations and goals in mind. Glöckner established a series of properties that he saw as necessary for a "reasonable" mechanism of fuzzification. Because of these differences in motivation, some of the proposed QFMs will conflict with his proposed properties. Moreover, some of the newer works focus on dealing with issues that Glöckner did not address in his monograph, and also with reducing computational complexity, at the cost of losing some of the DFS properties. This analysis is focused in looking at the strengths and "weaknesses" of the various QFMs that hint at their possible use in modelling various aspects of natural languages.

Table 6.1 explains the symbols we use to denote each of the QFMs in the example results.

6.1 Glöckner's "Induced Propositional Logic"

In the TGQ there are methods to operate directly on quantifiers, producing new quantifiers by using existing ones. For example, there is an operator to negate a quantifier or to produce an antonym, and so forth. All of these operations, however, were defined originally for the classical two-valued setting.

Glöckner is interested in generalising this to the fuzzy setting in a systemic way. As a preliminary step, Glöckner uses the QFM framework to define a *induced* fuzzy truth function $\tilde{f} : [0,1]^n \to [0,1]$ which is lifted from a simpler semi-fuzzy truth function $f : \{0,1\}^n \to [0,1]$.

It is notable here, that the *Induced Propositional Logic* (IPL) is considering the classical truth functions for conjunction and negation as a starting point (with disjunction and

Quantifier Fuzzification Mechanism	Symbol used	Definition
Glöckner's Integral based DFS	$\mathcal{M}$	14
Glöckner's CX model	$\mathcal{M}_{CX}$	15
Glöckner's OWA generalisation	$\mathcal{F}_{OWA}$	17
Maximum dependence model	$\mathcal{F}_{MD}$	19
Independence model	$\mathcal{F}_{I}$	20
Approximate dependency model	$\mathcal{F}_{AD}$	21
Probabilistic model	$\mathcal{F}^{A}$	23
Representation-level based QFM	S	26
Closeness-based QFM using ukasiewicz t-norm	$Cl_{\text{int}}, Cl_{\text{ext}}$	(30,28)
Closeness-based QFM using Gödel t-norm	$Cl_{\text{int}_G}, Cl_{\text{ext}_G}$	(30,28)
Closeness-based QFM using Product t-norm	$Cl_{\text{int}_P}, Cl_{\text{ext}_P}$	(30,28)

Table 6.1: The abbreviations used to denote QFMs

implication being derivable from these, in his framework). We will state their definitions later in this section.

First some basic notational definitions. We fix the domain as $D = \{1, \ldots, n\}$.

A bijection η is defined between $\mathbf{2}^n = \{(x_1, \ldots, x_n) \mid x_i \in \{0, 1\}\}$ and the power set $\mathcal{P}(D)$.

$$\eta(x_1, \ldots, x_n) = \{c \in D \mid x_c = 1\} \tag{6.1}$$

$$\eta^{-1}(D) = (x_1, \ldots, x_n), \text{ where } x_c = 1 \text{ iff } c \in D \tag{6.2}$$

In the fuzzy case, there exists an analogous bijection $\tilde{\eta} : [0,1]^n \to \tilde{\mathcal{P}}(D)$, where $[0,1]^n = \{(x_1, \ldots, x_n) \mid x_i \in [0,1]\}$

$$\mu_{\tilde{\eta}(x_1,\ldots,x_n)}(c) = x_c \ \forall c \in D \tag{6.3}$$

$$\tilde{\eta}^{-1}(\tilde{F}) = (\mu_{\tilde{F}}(1), \ldots, \mu_{\tilde{F}}(n)) \tag{6.4}$$

These bijections are used to link semi-fuzzy truth functions $f : \mathbf{2}^n \to [0,1]$ to unary semi-fuzzy quantifiers $Q_f : \mathcal{P}(D) \to [0,1]$. Analogously in the fuzzy case, for the truth function $\tilde{f} : [0,1]^n \to [0,1]$ we have the quantifier $Q_{\tilde{f}} : \tilde{\mathcal{P}}(D) \to [0,1]$. To give a formal definition of this:

Definition 31. *Let us assume we have a QFM $\mathcal{F} : SFQ \to FQ$, a semi-fuzzy truth function $f : \mathbf{2}^n \to [0,1]$ and $\eta^{-1} : \mathcal{P}(D) \to \mathbf{2}^n$, stated in 6.2. The semi-fuzzy quantifier $Q_f : \mathcal{P}(D) \to [0,1]$ is defined as*

$$Q_f(Y) = f(\eta^{-1}(Y)) \text{ for all } Y \in \mathcal{P}(D).$$

The induced fuzzy truth function $\tilde{\mathcal{F}}(f) : [0,1]^n \to [0,1]$ *is then defined using $\tilde{\eta} : [0,1]^n \to \tilde{\mathcal{P}}(D)$ (from 6.3):*

$$\tilde{\mathcal{F}}(f)(x_1, \ldots, x_n) = \mathcal{F}(Q_f)(\tilde{\eta}(x_1, \ldots, x_n)) \text{ for all } x_i \in [0,1].$$

In the remainder of this chapter, therefore, if we have a QFM $\mathcal{F}$ and a semi-fuzzy truth function f, then the induced fuzzy truth function will be simply denoted by $\tilde{\mathcal{F}}(f)$.

We shall now proceed to state "reasonable" definitions for semi-fuzzy truth functions for negation, conjunction and disjunction (as they are used by Glöckner) and then state the requirements that should hold for a QFM, as they are part of Glöckner's DFS scheme.

$$\neg(x) = \begin{cases} 0 & \text{if } x = 1 \\ 1 & \text{otherwise} \end{cases} \tag{6.5}$$

$$\wedge(x, y) = \begin{cases} 1 & \text{if } x = y = 1 \\ 0 & \text{otherwise} \end{cases} \tag{6.6}$$

$$\vee(x, y) = \begin{cases} 1 & \text{if } x = 1 \text{ or } y = 1 \\ 0 & \text{otherwise ``} \end{cases} \tag{6.7}$$

With Definition 31 and these semi-fuzzy truth functions we have all that is needed to induce the fuzzy truth functions via a given QFM. Since there are many ways to define a QFM, there is no guarantee of which properties the induced truth functions will have. To restrict this, Glöckner expects a DFS to fulfil certain requirements on its IPL. First some definitions that are used as part of these DFS requirements.

Definition 32. $\tilde{\neg} : [0, 1] \to [0, 1]$ *is called a* strong negation operator *if it satisfies*

- $\tilde{\neg}0 = 1$ *(boundary condition)*
- $\tilde{\neg}x_1 \geq \tilde{\neg}x_2$ *for all* $x_1, x_2 \in [0, 1]$ *s.t.*$x_1 \leq x_2$ *(monotonically non-increasing)*
- $\tilde{\neg} \circ \tilde{\neg}$ *is the identity function (involutive)*

Definition 33. $\tilde{s} : [0, 1] \to [0, 1]$ *is called a* s-norm *(or t-conorm) if it satisfies*

- $\tilde{s}(x, 1) = 1$ *(absorbing element)*
- $\tilde{s}(x, 0) = x$ *(neutral element)*
- $\tilde{s}(x, y) = \tilde{s}(y, x)$ *(commutativity)*
- *If* $x \leq z$*, then* $\tilde{s}(x, y) \leq \tilde{s}(z, y)$ *(monotonically nondecreasing)*
- $\tilde{s}(\tilde{s}(x, y), z) = \tilde{s}(x, \tilde{s}(y, z))$ *(associativity)*

for all x,y,z $\in [0, 1]$.

For the induced operators, the following should hold under a DFS $\mathcal{F}$:

1. $\tilde{\neg} = \tilde{\mathcal{F}}(\neg)$ is a strong negation operator, see Definition 32

2. $\tilde{\wedge} = \tilde{\mathcal{F}}(\wedge)$ is a t-norm, see Definition 4

3. $\tilde{\vee}(x_1, x_2) = \tilde{\neg}(\tilde{\wedge}(\tilde{\neg}x_1, \tilde{\neg}x_2))$ is the dual s-norm (see Definition 33) of $\tilde{\wedge}$ under $\tilde{\neg}$ (i.e. $\tilde{\vee}(x_1, x_2)$ as it is defined, fulfils the s-norm properties)

4. $\tilde{\Rightarrow}(x_1, x_2) = \tilde{\vee}(\tilde{\neg}x_1, x_2)$

It is perhaps appropriate to draw a comparison here between the approach based on QFMs and the framework from Fermüller and Roschger that is starting from ukasiewicz logic and extends it with fuzzy quantifiers. Whereas IPL is a top-down approach, the other is essentially the bottom-up counterpart.

6.2 Glöckner's DFS properties for different QFM proposals

As mentioned in section 3.4, Glöckner argued for the need of fuzzy quantifiers to be linguistic adequateness and introduced six axioms that a QFM should satisfy to capture this notion. The definitions are taken from his monograph [Glö06]. For sake of completeness, we will also showcase Glöckner's own QFM models in this chapter, even though these are already known to satisfy the DFS axioms. Especially with the $\mathcal{M}_{CX}$ model there are some surprising (on first glance) results.

The majority of the others QFMs do not have the same motivations and will unsurprisingly "fail" at some of these properties. As stated before, this is not necessarily a failure as much as it stems from a difference in focus and application. Nonetheless it is useful to understand which of the aforementioned QFMs violate which principle and to understand why this is so.

6.2.1 Correct generalisation

This property is concerned with the relation of the fuzzy quantifiers created by a QFM and how (and if) they subsume the original semi-fuzzy quantifiers. In other words, once we apply a QFM $\mathcal{F}$ to a given semi-fuzzy quantifier Q, we want to know if $\mathcal{F}(Q)$ behaves as Q under crisp sets. If we call this restriction step $\mathcal{U}$ this can be formally stated as follows:

Glöckner's DFS Property 1.
We are given a QFM $\mathcal{F}$ and any semi-fuzzy quantifier Q. If it holds that

$$\mathcal{U}[\mathcal{F}(Q)] = Q,$$

then $\mathcal{F}$ satisfies Correct Generalisation.

QFM	$\mathcal{U}[\mathcal{F}(\textbf{almost all})](C_1, C_2)$	$\textbf{almost all}(C_1, C_2)$
$\mathcal{M}$	1/8	1/8
$\mathcal{M}_{CX}$	1/8	1/8
$\mathcal{F}_{OWA}$	1/8	1/8
$\mathcal{F}_{MD}$	1/8	1/8
$\mathcal{F}_{I}$	1/8	1/8
$\mathcal{F}_{AD}$	1/8	1/8
$\mathcal{F}_{A}$	1/8	1/8
S	1/8	1/8
Cl_{int}	**3/4**	1/8
Cl_{ext}	**3/4**	1/8
Cl_{int_G}	**3/4**	1/8
Cl_{ext_G}	**3/4**	1/8
Cl_{int_P}	**3/4**	1/8
Cl_{ext_P}	**3/4**	1/8

Table 6.2: Results of Correct generalisation example.

We proceed to give an example that shows how the various QFMs satisfy or break this property in the given case:

Example 1. *Consider the semi-fuzzy quantifier* ***almost all****, defined in Definition 6. We will use for simplicity sake the two crisp sets* $C_1 = \{A, B, C, D\}$*,* $C_2 = \{A, B, C\}$*.*

We compute first the expression **almost all**(C_1, C_2)*, and then compare it for each QFM* $\mathcal{F}$ *with* $\mathcal{U}[\mathcal{F}(\textbf{almost all})](C_1, C_2)$*, where* $\mathcal{U}$ *is the restriction operator stated earlier.*

We can see the results in Table 6.2.

To understand why most QFMs presented here satisfy this property, we simply have to look at how alpha cuts, or three cuts deal with such a fuzzy-set, which is essentially crisp. Clearly a fuzzy set with just membership degrees 1 and 0 will always lead to one unique crisp approximation. No matter how a QFM is aggregating the results from multiple possible crisp approximations, in the case of just one, they all collapse to the underlying semi-fuzzy model.

In case of the closeness-based *Cl* QFMs, we are allowed to consider all choices of domain elements as precisification, and then consider the supremum. In Example 1, we can pick elements that lead to an alpha-cut with $\alpha = 0$, which means we produce a crisp approximation that contains the entire domain. Since **almost all** is increasing on its scope, evaluating it on this crisp approximation will lead to a larger truth value than in the semi-fuzzy case, namely 1. At the same time the closeness measure will be lower, than if we would just choose the original crisp set. In sum we obtain 3/4: a higher truth overall than the 1/8 in the crisp case, leading to a diverging result. Therefore, the *Cl* QFMs do not generalise the semi-fuzzy quantifiers correctly, as Glöckner requires.

6.2.2 Membership Assessment

We call a (unary) quantifier a *membership assessment* if it is true iff a given element e of the domain is contained in its argument. In the semi-fuzzy case, this will be a two-valued quantifier, since we can determine for a crisp set whether it absolutely contains e or not. On a fuzzy set we expect the membership degree of e to be the output. To fix the notation, we shall refer to the semi-fuzzy membership assessment as π_e and for the fuzzy case we shall use $\tilde{\pi}_e$.

To make this property clearer, we shall formally define the needed types of quantifiers:

$$\pi_e(A) = \mu_A(e) = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{otherwise} \end{cases} \tag{6.8}$$

where we recall that μ_A is the membership function, or characteristic function of the set A, and e is an element of the domain D. The analogous definition holds for $\tilde{\pi}_e$, except we use a fuzzy set $\tilde{A}$ as input and its membership function $\mu_{\tilde{A}}$, ie.:

$$\tilde{\pi}_e(\tilde{A}) = \mu_{\tilde{A}}(e) \tag{6.9}$$

This special class of quantifiers are dependant on the inclusion of an individual element of the domain in a crisp or fuzzy set. Glöckner requires that a "plausible model" of quantifiers fuzzification should lift the quantifier π_e to its fuzzy counterpart $\tilde{\pi}_e$.

Glöckner's DFS Property 2.
We are given a QFM $\mathcal{F}$. If we have

$$\mathcal{F}(\pi_e) = \tilde{\pi}_e \ \forall e \in D,$$

where π_e and $\tilde{\pi}_e$ are stated in 6.8 and 6.9, then F satisfies Membership Assessment.

Two of Glöckner's QFM models fulfil this as it is part of the DFS requirements. All voting-based QFM models of Díaz-Hermida et. al and the representation level approach of Sánchez fulfil the membership assessment property. To understand why, we consider a simplified view that describes how they both work. Each method collects truth values over various alpha cuts, in the range from 0 to 1, and then aggregates them into an integral or a finite sum. This sum corresponds to a proportion over this range, between the parts in the range $[0, 1]$ where π_e is true, and the parts where it is false.

Interestingly enough, the QFM $\mathcal{M}_{CX}$ breaks this property. The explanation found for this stems from the definition of the quantification method in Glöckner's monograph, as given in chapter 5. $\mathcal{M}_{CX}$ is defined on quantifiers that produce truth functions which are nonincreasing. Therefore Glöckner here is clearly limiting the possible models of

QFM	$\mathcal{F}(\pi_{\text{Sue}})(\mathbf{tall})$
$\mathcal{M}$	3/4
$\mathcal{M}_{CX}$	**1/2**
$\mathcal{F}_{OWA}$	3/4
$\mathcal{F}_{MD}$	3/4
$\mathcal{F}_{I}$	3/4
$\mathcal{F}_{AD}$	3/4
$\mathcal{F}_{A}$	3/4
S	3/4
Cl_{int}	**11/15**
Cl_{ext}	**11/15**
Cl_{int_G}	**11/15**
Cl_{ext_G}	**11/15**
Cl_{int_P}	**11/15**
Cl_{ext_P}	**11/15**

Table 6.3: Results of Membership Assessment example.

semi-fuzzy quantifiers for which $\mathcal{M}_{CX}$ is defined. As the author notes "no models of interest are lost", as far as he was concerned. The semi-fuzzy quantifier π_e is increasing, based on the element e and the crisp set supplied to it. By its definition, $\mathcal{M}_{CX}$ is therefore exempt from this property, but we shall still, for completeness sake, use it in the example below.

The proposed QFM *Cl* from Baldi and Fermüller does not fulfil Glöckner's DFS Property 2. To see why this is the case, we shall recall the underlying semantics to their approach: The main problem is that the closeness measure is considered globally for all elements in the domain, not restricted to e from π_e. This will not, in general, lead to a correct proportion on how π_e behaves over all possible alpha cuts. This is not so much a weakness of *Cl* as it is simply a characterization of its underlying semantics.

Example 2. *Consider a domain $D = \{Tom, Sue, Marcus, Michael, Mary\}$. Then a fuzzy set on D **tall** $= \{(Tom, 2/3), (Sue, 3/4), (Marcus, 1), (Michael, 1/2), (Mary, 1/4)\}$. We consider the semi-fuzzy quantifier π_e as defined in 6.8.*

The fuzzy quantifier $\tilde{\pi}_{Sue}(\mathbf{tall})$, as defined in 6.9, evaluates to 3/4.

The results for the various fuzzifications of π_{Sue} are listed in Table 6.3.

6.2.3 Dualisation

This property is based on the IPL, as introduced earlier in this chapter. To repeat the basic idea: One first establishes a bijection between the sets $\mathbf{2}^n$ and $\mathcal{P}(\{1,\ldots,n\})$, where n is the size of the domain. In the fuzzy case we have the sets $[0,1]^n$ and $\tilde{\mathcal{P}}(\{1,\ldots,n\})$, where $[0,1]$ is an ordered range of truth values, and $\tilde{\mathcal{P}}(\{1,\ldots,n\})$ is the set of all fuzzy sets on the domain. For a semi-fuzzy truth function $f : \mathbf{2}^n \to [0,1]$ we produce a translation into a "fully" fuzzy truth function $\tilde{f} : [0,1]^n \to [0,1]$.

The focus of the Dualisation property is on the truth function for negation $\tilde{\neg} : [0,1] \to [0,1]$. We use the following definition.

Definition 34. *We are given a QFM $\mathcal{F} : SFQ \to FQ$ and the semi-fuzzy truth function $\neg : \mathbf{2} \to [0,1]$, stated in 6.5. The* fuzzy complement operator *is defined as*

$$\mu_{\tilde{\neg} X}(e) = \tilde{\mathcal{F}}(\neg)\mu_X(e)$$

where $\tilde{\mathcal{F}}(f)$ for a semi-fuzzy truth function f is given in Definition 31. Furthermore, we shall denote $\tilde{\neg} = \tilde{\mathcal{F}}(\neg)$ for the induced fuzzy truth function for negation.

This way we get a fuzzy complement operation from a simple truth function, using a QFM (therefore the "induced" part). Similarly, one can also define a fuzzy union and intersection, from the semi-fuzzy truth functions $\wedge$ and $\vee$, though they will not be needed for this property.

Glöckner defines the following operations on (both semi-fuzzy and fuzzy) quantifiers:

Definition 35. *We are given the induced fuzzy truth function for negation $\tilde{\neg} : [0,1] \to [0,1]$. The* external negation $\tilde{\neg} Q$ *of a semi-fuzzy quantifier Q is defined as:*

$$(\tilde{\neg} Q)(Y_1, \ldots, Y_n) = \tilde{\neg}(Q(Y_1, \ldots, Y_n))$$

External negation is the first form of quantifier negation and consists of applying the induced truth function for negation to the final result. As an example for the models produced by external negation, Glöckner shows that **no** is the external negation of **some**:

$$\mathbf{no}(\mathbf{birds}, \mathbf{fish}) = \tilde{\neg}\mathbf{some}(\mathbf{birds}, \mathbf{fish})$$

Definition 36. *We are given an induced fuzzy complement operator $\tilde{\neg} : \tilde{\mathcal{P}}(\{1, \ldots, n\}) \to \tilde{\mathcal{P}}(\{1, \ldots, n\})$. The* antonym (or internal negation) $Q\tilde{\neg}$ *of a semi-fuzzy quantifier Q is defined as:*

$$Q\tilde{\neg}(Y_1, \ldots, Y_n) = Q(Y_1, \ldots, Y_{n-1}, \tilde{\neg} Y_n)$$

The antonym consists of applying the induced fuzzy complement operator to the last argument of the quantifier. This might seem arbitrary at first, however, by convention this last arguments represents the scope. Therefore, we "flip" the scope (relative to a given domain) to get the antonym. To see how the antonym is used in natural language, Glöckner shows a few equivalences, following Def. 3.9 [Glö06]. For example **no** is the antonym of **all**, leading to

$$\mathbf{no}(\mathbf{birds}, \mathbf{fish}) = \mathbf{all}(\mathbf{birds}, \neg\mathbf{fish})$$

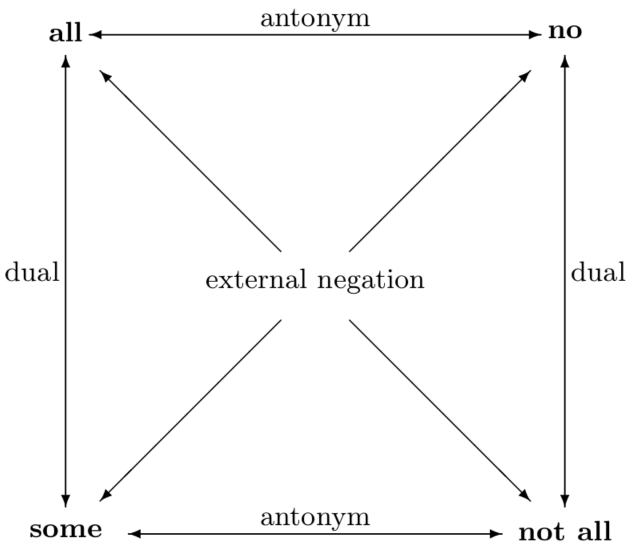


Figure 6.1: A square of opposition, for "all", "some" and "no", from Glöckner

Definition 37. *We are given both an induced fuzzy complement operator* $\tilde{\neg} : \tilde{\mathcal{P}}(\{1, \dots, n\}) \to \tilde{\mathcal{P}}(\{1, \dots, n\})$ *and the induced fuzzy truth function for negation* $\tilde{\neg} : [0,1] \to [0,1]$*. The* dual $Q\tilde{\square}$ *of a semi-fuzzy quantifier* Q *is defined as:*

$$Q\tilde{\square}(Y_1, \dots, Y_n) = \tilde{\neg} Q(Y_1, \dots, Y_{n-1}, \tilde{\neg} Y_n)$$

Finally, the dual of a quantifier is obtained by performing *both* external and internal negation simultaneously. As before, Glöckner gives example of a few quantifiers and shows how they form a *square of opposition*, following the classical model of Aristotle, seen in Figure 6.1.

Glöckner states for each form of quantifier negation a corresponding requirement that a QFM should satisfy to be a DFS. The idea behind them is to have a "reasonable" behavior, that connects the semi-fuzzy case with the lifted fuzzy case. For brevity's sake, it is enough to simply list the one for the dual operator, as it incorporates the previous two forms of negation, and the other requirements can be derived from it [Glö06].

For a given DFS $\mathcal{F}$, a semi-fuzzy quantifier Q, the following must hold:

Glöckner's DFS Property 3.
Let Q *be any semi-fuzzy quantifier and* $\mathcal{F}$ *a QFM. If it holds that*

$$\mathcal{F}(Q\tilde{\square}) = \mathcal{F}(Q)\tilde{\square},$$

where $\tilde{\square}$ *is given in Definition 37, then* $\mathcal{F}$ *satisfies* Dualisation*.*

Note that we use the same symbol $\tilde{\square}$ for both the semi-fuzzy and fuzzy case. This is done for ease of notation, though it should be noted here that both of these are defined by combining internal and external negation, but they necessarily use different forms of complementation for the internal negation. In the semi-fuzzy case this uses the usual complement operation from set theory. The fuzzy complement is as defined in Definition 34.

This property holds for all QFMs defined by Glöckner, and this is also shown in the examples given here. It is interesting to note that even though $\mathcal{M}_{CX}$ fails to induce a "strong negation", since it is only defined for non-increasing quantifiers, it actually does not violate Glöckner's DFS Property 3. This seems like an oversight, one might wish to strengthen this property by requiring for a DFS $\mathcal{F}$, that $\tilde{\mathcal{F}}(\neg)$ is a strong negation operator.

Of the three voting-based QFM methods from Delgado et al. only the independence method fulfils Glöckner's DFS Property 3, while the maximum-dependence model breaks it. This follows from their own paper, and is confirmed in all examples shown here. Since approximate-dependency method can be defined as a combination of maximum-dependency and the independence method, it also breaks this property, as noted by Delgado et al.

The generalised *Cl* QFMs do not fulfil this property, as will be shown by a counterexample. To see why this is so, we look at the two truth functions that are to be induced by the QFM and the requirements on these induced truth functions. *Cl* induces a *strong negation operator* namely standard negation (for all variants), so external negation corresponds to the DFS requirement, but the induced complement operator does not conform with complementation in the crisp case.

Example 3. *We assume a domain $D = \{Tom, Sue, Marcus, Michael, Mary, James, Sarah, Katherine, Martin, Lucas, John, Patrick, Lisa\}$.*
On this we define two fuzzy sets, **Young** *and* **Dilligent***, as follows:*

$$\begin{aligned}\textbf{Young} = \{&(Sue, 2/7), (Marcus, 3/7), (Michael, 4/7), (Mary, 5/7), (James, 6/7),\\ &(Sarah, 1), (Katherine, 6/7), (Martin, 5/7), (Lucas, 4/7), (John, 3/7), (Patrick, 2/7), \}\\ \textbf{Dilligent} = \{&(Sue, 6/7), (Marcus, 5/7), (Michael, 4/7), (Mary, 3/7), (James, 2/7),\\ &(Sarah, 1/7), (Katherine, 2/7), (Martin, 3/7), (Lucas, 4/7), (John, 5/7), (Patrick, 6/7), \}\end{aligned}$$

As a quantifier we choose for this example **few***, defined in Chapter 4. To see whether the various QFMs fulfil the Duality principle, we first compute the left hand side of Glöckner's DFS Property 3, which entails computing the dual of the semi-fuzzy quantifier, and then lifting it to the fuzzy case. Then the right hand side, were the order is the opposite: the lifting is performed first, and then the dual of the new fuzzy quantifier is applied to the argument. The results for the considered QFMs can be seen in Table 6.4. As explained $\mathcal{F}_{MD}$ fails the property Glöckner's DFS Property 3, leading to different results. This also shows that the representation-level approach S from Sánchez et al. does not satisfy this property.*

QFM	$\mathcal{F}(\mathbf{few}\tilde{\Box})(\mathbf{Young}, \mathbf{Dilligent})$	$\mathcal{F}(\mathbf{few})\tilde{\Box}(\mathbf{Young}, \mathbf{Dilligent})$
$\mathcal{M}$	5/7	5/7
$\mathcal{M}_{CX}$	1/2	1/2
$\mathcal{F}_{OWA}$	5/7	5/7
$\mathcal{F}_{MD}$	**5/7**	**6/7**
$\mathcal{F}_{I}$	36041371/47064402	36041371/47064402
$\mathcal{F}_{AD}$	**69658801/94128804**	**76382287/94128804**
$\mathcal{F}_{A}$		
S	**5/7**	**6/7**
Cl_{int}	**32/77**	**36/77**
Cl_{ext}	**32/77**	**36/77**
Cl_{int_G}	**50/77**	**5585/12474**
Cl_{ext_G}	**54/77**	**30/77**
Cl_{int_P}	**274/539**	**36/77**
Cl_{ext_P}	**270/539**	**36/77**

Table 6.4: Results of the Dualisation example.

6.2.4 Internal Joins

As with Dualisation before, this property is build on the IPL, introduced earlier in this chapter. The focus here is on the union operator and the truth function for $\vee$. For crisp sets we have the standard union operation $\cup$ from set theory. For fuzzy sets the induced operation is defined analogously to the induced complement in Definition 34:

Definition 38. *We are given a QFM $\mathcal{F} : SFQ \to FQ$ and the semi-fuzzy truth function $\vee : \mathbf{2} \to [0,1]$, stated in 6.7. The* fuzzy union operator $\tilde{\cup}$ *is defined as*

$$\mu_{Y\tilde{\cup}X}(e) = \tilde{\mathcal{F}}(\vee)(\mu_Y(e), \mu_X(e))$$

where $\tilde{\mathcal{F}}(f)$ for a semi-fuzzy truth function f is given in Definition 31. Furthermore, we shall denote $\tilde{\vee} = \tilde{\mathcal{F}}(\vee)$ for the induced fuzzy truth function for negation.

Next Glöckner states an operation on both semi-fuzzy and fuzzy quantifiers. This operations extends the arity of the quantifier by taking an additional argument (a crisp or fuzzy set) and "joining" it with the last argument using a crisp of fuzzy union operator.

Definition 39. *Let a semi-fuzzy quantifier $Q : \mathcal{P}(X)^n \to [0,1]$ be given. We define the semi-fuzzy quantifier $Q\cup : \mathcal{P}(X)^{n+1} \to [0,1]$ by*

$$Q\cup(Y_1, \ldots, Y_{n+1}) = Q(Y_1, \ldots, Y_n \cup Y_{n+1})$$

for all $Y_1, \ldots Y_{n+1} \in \mathcal{P}(X)$.

This is defined analogously for fuzzy quantifiers, but $\tilde{Q}\tilde{\cup}$ is defined using an induced fuzzy union operator $\tilde{\cup}$

Finally for a DFS, it is required that performing such an internal join before the lifting on the semi-fuzzy quantifier, should produce the same value as lifting it first to the fuzzy case and performing the internal join on the fuzzy case thereafter.

Glöckner's DFS Property 4.
Let Q be any semi-fuzzy quantifier and a QFM $\mathcal{F}$. If we have

$$\mathcal{F}(Q\cup) = \mathcal{F}(Q)\tilde{\cup},$$

where $\cup$ and $\tilde{\cup}$ are given in Definition 39, then $\mathcal{F}$ satisfies Internal Join*.*

From the voting-based QFMs, this axiom is only satisfied by $\mathcal{F}_{MD}$, but not by $\mathcal{F}_I$ or $\mathcal{F}_{owa}$. It is also fulfilled by the level-representation based QFM S.

Example 4. *For this example we take the fuzzy sets from a previous example and add as a third one* **Tall***, as the $\cup$ operator increases the arity of a given quantifier. Furthermore, for performance reasons, we have reduced the size of the domain and we are leaving out completely $\mathcal{F}^A$ as its exponential runtime, in the naive implementation, make it infeasible for ternary quantifiers.*

$$\begin{aligned}
\textbf{Young} &= \{(Marcus, {}^3/_7), (Michael, {}^4/_7), (Mary, {}^5/_7), (James, {}^6/_7),\\
&\quad (Sarah, 1), (Katherine, {}^6/_7), (Lucas, {}^4/_7), (John, {}^3/_7), \}\\
\textbf{Dilligent} &= \{(Marcus, {}^5/_7), (Michael, {}^4/_7), (Mary, {}^3/_7), (James, {}^2/_7),\\
&\quad (Sarah, {}^1/_7), (Katherine, {}^2/_7), (Lucas, {}^4/_7), (John, {}^5/_7), \}\\
\textbf{Tall} &= \{(Marcus, {}^1/_2), (Michael, {}^1/_2), (Mary, {}^1/_2), (James, {}^1/_2),\\
&\quad (Sarah, {}^1/_2), (Katherine, {}^1/_2), (Lucas, {}^1/_2), (John, {}^1/_2), \}
\end{aligned}$$

We abbreviate **Young***,* **Dilligent** *and* **Tall***, respectively, as* **Y***,* **D** *and* **T** *to fit them into the table.*

As a semi-fuzzy quantifier we choose the deliberate-choice quantifier $W_2(\Pi^1_2)$.

The results can be seen in Table 6.5. It is broken by all versions of the ClQFM, as their induced union operators $\tilde{\cup}$ fail to adhere to Glöckner's requirements. This leads to diverging results between the first and second case. The same is true for $\mathcal{F}^I$ and $\mathcal{F}^{AD}$, as stated by Delgado et al in their survey [DRSV14]. While left out for this example for performance reasons, $\mathcal{F}_A$ satisfies this DFS axiom, according to [DHLBB05], so we would expect the same value on both sides.

6.2.5 Preserving Monotonicity

To define the monotonicity property on fuzzy quantifiers, we will first need to define a *fuzzy inclusion relation.*

QFM	$\mathcal{F}(W_2(\Pi_2^1)\cup)(Y,D,T)$	$\mathcal{F}(W_2(\Pi_2^1))\tilde{\cup}(Y,D,T)$
$\mathcal{M}$	1/2	1/2
$\mathcal{M}_{CX}$	1/2	1/2
$\mathcal{F}_{OWA}$	593/1344	593/1344
$\mathcal{F}_{MD}$	9/320	9/320
$\mathcal{F}_{I}$	**14849/141120**	**13607/282240**
$\mathcal{F}_{AD}$	**9409/141120**	**13607/282240**
$\mathcal{F}_{A}$		
S	9/320	9/320
Cl_{int}	**0**	**583/3360**
Cl_{ext}	**0**	**583/3360**
Cl_{int_G}	**23/63**	**1343/3360**
Cl_{ext_G}	**7/18**	**1121/2520**
Cl_{int_P}	**281/2520**	**4937/17640**
Cl_{ext_P}	**409/3528**	**1009/3528**

Table 6.5: Results of the Internal Joins example.

Definition 40. *Taken from Def. 3.14 ([Glö06], p. 98)*
Assume a domain D and $\tilde{X}_1, \tilde{X}_2 \in \tilde{\mathcal{P}}(D)$ are fuzzy sets on D. We say that $\tilde{X}_1$ is contained in $\tilde{X}_2$ ($\tilde{X}_1 \subseteq \tilde{X}_2$) if

$$\mu_{\tilde{X}_1}(e) \leq \mu_{\tilde{X}_2}(e) \quad \forall e \in D$$

Based on this, we can proceed to define what it means for a semi-fuzzy or fuzzy quantifier to be monotonic on one of its arguments.

Definition 41. *Taken from Def. 3.15 ([Glö06], p. 98)*
A semi-fuzzy quantifier $Q : \mathcal{P}(D) \to [0,1]$ is said to be nondecreasing *in its i-th argument, $i \in \{1, \ldots, n\}$, if*

$$Q(Y_1, \ldots, Y_n) \leq Q(Y_1, \ldots, Y_{i-1}, Y_i', Y_{i+1}, \ldots, Y_n)$$

whenever the involved argument $Y_1, \ldots, Y_n, Y_i' \in \mathcal{P}(D)$ satisfy $Y_i \subseteq Y_i'$. Q is said to be nonincreasing *in the i-th argument, if $Y_i \subseteq Y_i'$, it always holds that*

$$Q(Y_1, \ldots, Y_n) \geq Q(Y_1, \ldots, Y_{i-1}, Y_i', Y_{i+1}, \ldots, Y_n)$$

The corresponding definitions for fuzzy quantifiers are entirely analogous. The arguments in this case range over $\tilde{\mathcal{P}}(D)$ and the fuzzy inclusion relation is used instead.

Examples for semi-fuzzy quantifiers with monotonic behaviour in one of their arguments are *almost all*, which is nonincreasing in the first and nondecreasing in the second, and

few, where it is the other way around, nondecreasing for the first, nonincreasing for the second.

The DFS property has to preserve these behaviours in the newly produced fuzzy quantifiers. Formally, we have:

Glöckner's DFS Property 5. *Taken from Def. 3.17 ([Glö06]. p. 100)*
We are given a QFM $\mathcal{F}$. If it holds that all semi-fuzzy quantifiers $Q : \mathcal{P}(D) \to [0,1]$ which are nondecreasing (resp. nonincreasing) in their i-th argument, $i \in \{1, \dots, n\}$, are mapped to fuzzy quantifiers $\mathcal{F}(Q)$ which are also nondecreasing (resp. nonincreasing) in their i-th argument, then we say that $\mathcal{F}$ preserves monotonicity in the arguments

All QFMs of Glöckner which are looked at here fulfil this property. Of the voting-based QFMs, both F_{MD} and F_I fulfil this, and therefore also F_{AD} since it was defined as an interpolation of the previous two. As was stated in [DHLBB05], $\mathcal{F}^A$ is also preserving monotonicity of the arguments.

Finally, all the variations Cl for generalised quantifiers were resistant against attempts to break monotonicity. Of course, a simple test is not enough to formally prove that there is no counter-example to be found. However, this suggests that the extension to the n-ary case presented here might have this property.

Example 5. *We will use two fuzzy sets from a previous example.*

$$\begin{aligned}\textbf{Young} = \{&(Sue, 2/7), (Marcus, 3/7), (Michael, 4/7), (Mary, 5/7), (James, 6/7),\\ &(Sarah, 1), (Katherine, 6/7), (Martin, 5/7), (Lucas, 4/7), (John, 3/7), (Patrick, 2/7), \}\\ \textbf{Dilligent} = \{&(Sue, 6/7), (Marcus, 5/7), (Michael, 4/7), (Mary, 3/7), (James, 2/7),\\ &(Sarah, 1/7), (Katherine, 2/7), (Martin, 3/7), (Lucas, 4/7), (John, 5/7), (Patrick, 6/7), \}\end{aligned}$$

As a semi-fuzzy quantifier we select ***almost all****. It is increasing in its second argument, so to test whether the produced fuzzy quantifiers preserve this we will define another fuzzy set* **Dilligent′***, s.t.* **Dilligent** $\subseteq$ **Dilligent′***, where $\subseteq$ follows Definition 40.*

$$\begin{aligned}\textbf{Dilligent}' = \{&(Sue, 1), (Marcus, 5/7), (Michael, 4/7), (Mary, 3/7), (James, 3/7),\\ &(Sarah, 6/7), (Katherine, 4/7), (Martin, 3/7), (Lucas, 1), (John, 1), (Patrick, 6/7), \}\end{aligned}$$

Since no truth degree in **Dilligent′** *is smaller than in* **Dilligent***,* **Dilligent** $\subseteq$ **Dilligent′** *holds.*

The results of the monotonicity test can be seen in Table 6.6.

We can see in the example that all QFMs produce in the second case values that are equal to or higher than those in the the first case. The monotonicity of **almost all** is preserved.

QFM	$\mathcal{F}(\textbf{almost all})(\textbf{Young}, \textbf{Dilligent})$	$\mathcal{F}(\textbf{almost all})(\textbf{Young}, \textbf{Dilligent}')$
$\mathcal{M}$	$2/7$	$3/7$
$\mathcal{M}_{CX}$	$1/2$	$1/2$
$\mathcal{F}_{OWA}$	$2/7$	$3/7$
$\mathcal{F}_{MD}$	$2/7$	$3/7$
$\mathcal{F}_{I}$	$11023031/47064402$	$2913/5929$
$\mathcal{F}_{AD}$	$24470003/94128804$	$2727/5929$
$\mathcal{F}_{A}$		
S	$2/7$	$3/7$
Cl_{int}	$47/77$	$5/7$
Cl_{ext}	$47/77$	$5/7$
Cl_{int_G}	$47/77$	$5/7$
Cl_{ext_G}	$47/77$	$5/7$
Cl_{int_P}	$47/77$	$5/7$
Cl_{ext_P}	$47/77$	$5/7$

Table 6.6: Results of the Preserving Monotonicity example.

6.2.6 Functional application

The final property that determines a DFS, is concerned with resilience against bijections from one domain to another. It is therefore essentially a homomorphism condition, which ensures that expressions are evaluated the same, independently of whether the bijection is applied before or after the semi-fuzzy quantifier is lifted.

In the next step we use cross-domain bijections to define a mapping from sets of one domain to another. First the straightforward *powerset mapping* for crisp sets.

Definition 42. *Taken from Def. 3.19 ([Glö06], p. 101)*
To each mapping $f : D \to D'$, we associate a mapping $\widehat{f} : \mathcal{P}(D) \to \mathcal{P}(D')$ (the powerset mapping *of f) which is defined by*

$$\widehat{f}(Y) = \{f(e) \mid e \in Y\}$$

for all $Y \in \mathcal{P}(D)$.

In the fuzzy case, such a mapping is called an *extension principle*.

Definition 43. *Taken from Def. 3.21 ([Glö06], p. 102)*
An extension principle *$\mathcal{E}$ assigns to each mapping $f : D \to D'$ a corresponding mapping $\mathcal{E}(f) : \tilde{\mathcal{P}}(D) \to \tilde{\mathcal{P}}(D')$. For convenience, we shall assume that $D, D' \neq \emptyset$.*

Glöckner proceeds to state how to define such an extension principle using a QFM and a given mapping between domains.

Definition 44. *Taken from Def. 3.25 ([Glö06], p. 103)*
Every QFM $\mathcal{F}$ induces an extension principle $\widehat{\mathcal{F}}$ which to each $f : D \to D'$ (where $D, D' \neq \emptyset$) assigns the mapping $\widehat{\mathcal{F}}(f) : \tilde{\mathcal{P}}(D) \to \tilde{\mathcal{P}}(D')$ defined by

$$\mu_{\widehat{\mathcal{F}}(f)(X)}(e') = \mathcal{F}(\pi_{e'})(\widehat{f}(X)),$$

where $\pi_{e'}$ has been defined earlier for membership assignment.

For a succinct formulation of the final property, Glöckner proceeds to introduce some shorthand notations. We can construct a semi-fuzzy quantifier $Q' : \mathcal{P}(D')^n \to [0,1]$ by composing a quantifier $Q : \mathcal{P}(D)^n \to [0,1]$ with a given collection of powerset mappings $\widehat{f}_i, \dots, \widehat{f}_n$

$$Q'(Y_1, \dots, Y_n) = Q(\widehat{f}_1(Y_1), \dots, \widehat{f}_n(Y_n))$$

To express this more compactly, Glöckner utilizes *product mapping* and *functional decomposition*

$$(Q \circ \bigtimes_{i=1}^{n} \widehat{f}_i)(Y_1, \dots, Y_n) = Q(\widehat{f}_1(Y_1), \dots, \widehat{f}_n(Y_n))$$

A similar construction on fuzzy quantifiers is produced by using the induced extension principle $\widehat{\mathcal{F}}$ of a QFM. This is a composition of $\tilde{Q} : \mathcal{P}(\tilde{D})^n \to [0,1]$ with $\widehat{\mathcal{F}}(f_1), \dots, \widehat{\mathcal{F}}(f_n)$ to form, analogously to the above, the quantifier $\tilde{Q} \circ \bigtimes_{i=1}^{n} \widehat{\mathcal{F}}(f_i) : \mathcal{P}(\tilde{D'})^n \to [0,1]$, defined as

$$(\tilde{Q} \circ \bigtimes_{i=1}^{n} \widehat{\mathcal{F}}(f_i))(X_1, \dots, X_n) = \tilde{Q}(\widehat{\mathcal{F}}(f_1)(X_1), \dots, \widehat{\mathcal{F}}(f_n)(X_n).$$

The last DFS property can now be simple defined as follows:

Glöckner's DFS Property 6. *Taken from Def. 3.26 ([Glö06], p. 104)*
Let $\mathcal{F}$ be a given QFM. If it holds that

$$\mathcal{F}(Q \circ \bigtimes_{i=1}^{n} \widehat{f}_i) = \mathcal{F}(Q) \circ \bigtimes_{i=1}^{n} \widehat{\mathcal{F}}(f_i),$$

then $\mathcal{F}$ is compatible with functional application*.*

As with the properties before, these are fulfilled by the QFMs from Glöckner. From the voting-based ones, there is no answer given by the survey from Delgado et al. [DRSV14]. Interestingly enough, in all experiments with the implemented voting-based QFMs, they do seem to comply with this property. Of course, the absence of a counterexample is hardly a proof, but it suggests that a deeper formal analysis might be called for to

QFM	$\mathcal{F}(\textbf{around half} \circ \bigtimes_{i=1}^{n} \widehat{h})(\mathbf{X_1}, \mathbf{X_2})$	$\mathcal{F}(\textbf{around half}) \circ \bigtimes_{i=1}^{n} \widehat{\mathcal{F}}(h)(\mathbf{X_1}, \mathbf{X_2})$
$\mathcal{M}$	1/2	1/2
$\mathcal{M}_{CX}$	1/2	1/2
$\mathcal{F}_{OWA}$	1/2	1/2
$\mathcal{F}_{MD}$	7/10	7/10
$\mathcal{F}_{I}$	61/100	61/100
$\mathcal{F}_{AD}$	131/200	131/200
$\mathcal{F}_{A}$	151232/253125	151232/253125
S	7/10	7/10
Cl_{int}	**8/15**	**23/36**
Cl_{ext}	**8/15**	**23/36**
Cl_{int_G}	**19/30**	**133/180**
Cl_{ext_G}	**11/15**	**139/180**
Cl_{int_P}	**7/12**	**599/900**
Cl_{ext_P}	**44/75**	**1807/2700**

Table 6.7: Results of the Functional Application example.

clearly determine if these three QFMs are compatible with functional applications. A clear answer is lacking also for S, in the absence of counterexamples only a formal proof can positively answer the question.

Example 6. *We begin by defining two fuzzy sets again, this time on properties of various kinds of trees.*

$$\textbf{Tall trees} = \mathbf{X_1} = \{(Pine, 1)(Ash, 4/5)(Maple, 1/2)(Pear, 0)(Larch, 2/5)(Oak, 1/2)\}$$
$$\textbf{Timber hardness} = \mathbf{X_2} = \{(Pine, 9/10)(Ash, 0)(Maple, 3/5)(Pear, 1/2)(Larch, 0)(Oak, 4/5)\}$$

As we will define an isomorphism on domains, we make the domain of the fuzzy sets explicit.

$$D_1 = \{Pine, Ash, Maple, Peak, Larch, Oak\}$$

The domain onto which this is mapped is D_2, on colors.

$$D_2 = \{Red, Blue, Green, Orange, Violet, Turquoise\}$$

The isomorphism $h : D_1 \to D_2$ will map every element in D_1 to on in D_2, based on the order we have written down the sets (e.g. "Pine" to "Red", and "Oak" to "Turquoise").

The result of the functional application test can be seen in Table 6.7 .

To see why no variation of Cl fulfils this, we note that the induced extension principle needs the Membership Assessment property, as otherwise it will not correctly preserve the truth degrees. This is the reason why it differs on the left and right column of the table: $\widehat{h}$ is a different extension principle than $\widehat{\mathcal{F}}(h)$, if $\mathcal{F}$ is one of the Cl QFMs. Therefore, none of the Cl QFMs preserves functional application.

CHAPTER 7

QFMtool

As part of the thesis, a tool was developed which implements the presented semi-fuzzy quantifiers and QFMs models. It allows the user to declare fuzzy or crisp sets to represent vague or crisp concepts and then to state a fuzzy quantified sentence using these sets. This input functions as a query, the answer is the truth value of the input sentence. The tool will compute the truth value using the models presented in this thesis.

The statement at the end of such a query can be of the form

$$\mathcal{F}[Q](A_1, \dots, A_n),$$

where $\mathcal{F}$ is a QFM, Q is a semi-fuzzy quantifier, $A_1, \dots, A_n$ are fuzzy arguments. Another possible form for the statement at the end is

$$Q(B_1, \dots, B_n),$$

where Q is a semi-fuzzy quantifier, as before, but $B_1, \dots, B_n$ are crisp sets. The declarations of the argument sets must precede the final statement.

The semi-fuzzy quantifiers and QFMs are pre-defined, and can be selected via a drop-down menu, as seen in Figure 7.1. This will insert them at the current cursor position, and can also replace selected text.

The queries can be loaded from a selection of text files to present to the user a number of different expressions. This is meant to encourage the user to vary the QFMs to see how they differ on the same output, or vice-versa, how the same QFM produces different fuzzy quantifiers on various semi-fuzzy quantifiers. New queries can also be saved as text files.

Another utility built into QFMtool is to see various "plots", i.e. 2D or 3D representations of the truth functions of the used semi-fuzzy quantifiers, and the induced t-norms for

Figure 7.1: Screenshot of QFMtool, running on Ubuntu 17.10

the various QFMs. The former is useful to better understand the results of the fuzzy quantified expressions, the latter is an example on how the QFM models differ, as some of the different induced t-norms are used in some of the DFS axioms.

7.1 Technical description with implementation details

The tool and underlying algorithms were implemented in the functional programming language Objective Caml (OCaml for short), developed by the INRIA research institution in Rocquencourt, France [LDF+14]. It is an extension of the existing language Caml, which itself is based on Standard ML. So while OCaml was first released in 1996, it has been influenced by languages that date back further. As the name suggests, OCaml added object-oriented programming paradigms, such as introducing classes into the type system and multiple inheritance. OCaml is statically typed, so the type of all expressions is inferred at compile-time, making explicit type declarations unnecessary (though useful, as stricter type definitions can help detecting errors faster).

The implementation of the described models of QFMs and SFQs is based directly on the original source, in many cases using the mathematical descriptions as they were given. This was aided by the fact that purely functional code somewhat resembles mathematical notation, an example being the use of equality as an expression of (boolean) truth, rather than value assignments.

7.1.1 Finite computation of integrals

One technical detail, which is given in [DHBCB04], is the transformation of integrals to finite sums, preserving the actual value. This works only for finite domains, as in a finite domain fuzzy sets can only have finitely many crisp representations. To go into more detail, we look at the QFM $\mathcal{M}$ by Glöckner, stated in Definition 14.

$$\mathcal{M}(Q)(\tilde{A}) = \int_0^1 Q_\gamma(\tilde{A})\mathrm{d}\gamma$$

Based on$\tilde{A}$ we can then define a finite set $\mathcal{I} \subseteq [0,1]$ of truth values for which the three cuts will produce different crisp representations. Considering more values does not change the final value. We can then redefine $\mathcal{M}$ as:

$$\mathcal{M}(Q)(\tilde{A}) = \sum_{\gamma \in \mathcal{I}} Q_\gamma(\tilde{A}) * m(\gamma)$$

where $m(\gamma) = \gamma - \gamma'$ and γ' is the immediate predecessor of γ in $\mathcal{I}$ (or 0 if γ is the smallest already).

The critical thing is then of course the definition of $\mathcal{I}$. We differentiate between QFMs that use the three cut function to get crisp approximations, and QFMs that use the much more common alpha cut. For three cut based QFMs this set, which we denote

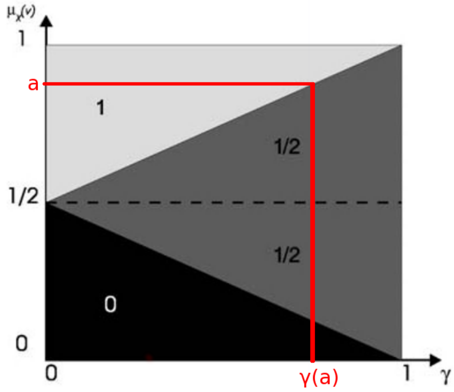


Figure 7.2: Relation of a to $\gamma(a)$, compared with the underlying *three-cut* function

by $\mathcal{I}_{tc}$, is based on the actual three cut function, given in 5.1. For all truth degrees $a \in \{\mu_{\tilde{A}}(c) \mid c \in D\}$, $\mathcal{I}_{tc}$ must contain the threshold $\gamma(a) \in [0, 1]$, that determines if the three valued set uses the *lower* or *higher* choice. To explain this in more detail: if one studies the three-cut function it becomes clear that values above $1/2$ can only be mapped to 1 or $1/2$, and values below $1/2$ will be mapped to 0 or $1/2$. The threshold $\gamma(a)$ is the critical point where a will be mapped to the corresponding lower choice. In QFMtool, this threshold can be defined with the following function:

$$\gamma(a) = \begin{cases} |2(a - 1/2)| - \epsilon & \text{if } a \notin \{0, 1/2, 1\} \\ a & \text{otherwise} \end{cases}$$

ϵ is some value that is "small enough". Currently that is 10^{-6}. The restriction here is that no two truth values, occurring in the fuzzy sets as a membership degree, should be closer than this chosen ϵ.

A visual presentation of $\gamma(a)$ can be seen in Figure 7.2. The definition of $\mathcal{I}_{tc}$ for the three cut based QFMs is then:

$$\mathcal{I}_{tc} = \{\gamma(\mu_{\tilde{A}}(c)) \mid c \in D\}$$

For alpha-cut based QFMs, this is considerably simpler. The truth values themselves *are* the thresholds that are needed. So in these QFMs, we have the following:

$$\mathcal{I}_{\alpha} = \{\mu_{\tilde{A}}(c) \mid c \in D\}$$

So we only need to collect the individual truth values occurring in the fuzzy arguments, and can replace all integrals by a sum over them.

7.1.2 Proportional quantifiers

Proportional semi-fuzzy quantifiers are defined using the scheme given in Definition 5. This general scheme needs only a simpler function $\mu : [0,1] \to [0,1]$ to produce a truth function for a semi-fuzzy quantifier. The implementation in OCaml follows the definition almost exactly, the only deviation is due to the fact that the type of semi-fuzzy quantifiers is assumed to be general, so no restriction on arity is given. Since proportional quantifiers of this kind are binary, an exception is thrown if the argument list does not have length 2. The implementation uses as the type of a crisp set a list of strings, so each single element in the domain is represented by a unique string.

Algorithm 7.1: The function used to define all binary SFQs in QFMtool

```
let relative_quantor_2 (mu: num -> num) = function
    | [[];b]  -> Int 1  (* if range is empty *)
    | [a ;b]  -> mu (Int (length (inter a b)) // Int (length a))
    | l       -> failure $ argument_error 2 (length l)
```

7.1.3 Using arbitrary precision numerals

The principal operation of QFMtool lies in the manipulation of real numbers. The properties on QFMs, such as those for DFS, are sensitive to small changes in the truth values of fuzzy quantified expressions. Unfortunately, the standard way of representing real numbers on computers, IEEE 754, is susceptible for rounding errors. While there are methods to compensate for these, QFMtool eschews floating-point numbers and instead uses *arbitrary precision numerals*. These work by representing real numbers by two integers, a denominator and a numerator. If necessary, the numerator will be increased. Therefore, this representation does not give a memory bound. The larger the number (higher denominator) and the more precise (higher numerator), the more memory it will consume. Furthermore, these numbers are not supported by special hardware. In essence, they are both slower and need more memory.

The problems with bounding errors show the importance of having QFMs that satisfy an additional property not discussed so far: continuity, or better *arg-continuity* [Glö06]. Roughly speaking, if a QFM is arg-continuous, the fuzzy quantifiers obtained are such that small changes in the input only determine small changes in the output. Continuous fuzzy quantifiers would thus be robust against bounding errors and any kind of noisy inputs.

For a performance critical application it is likely a better idea to use floating-point numbers instead, and look for methods to correct or at least mitigate rounding errors, such as arg-continuity mentioned above. Since the focus of QFMtool is not to be used in a performance sensitive environment, the need for precision seemed more urgent and therefore the numerous technical downsides justified. It shall be noted that replacing

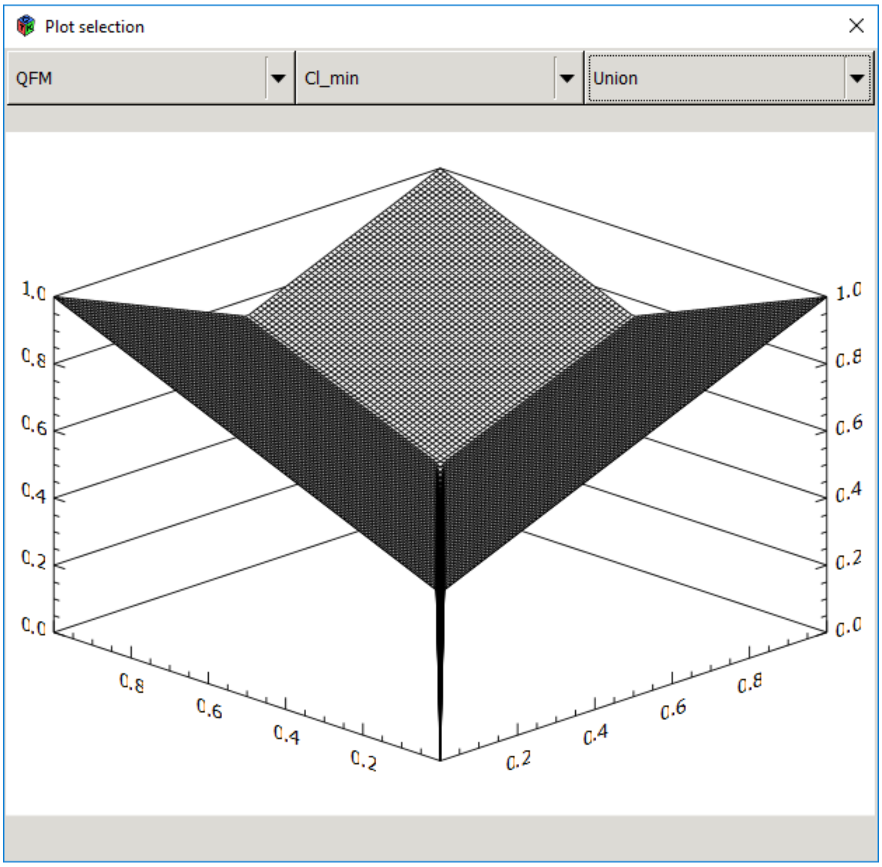


Figure 7.3: The plot viewer of QFMtool, running on Windows 10

the use of numerals with floating point numbers is fairly easy, only the used operators need to be replaced, with all other code staying the same.

7.2 Personal report on writing QFMtool

My experience was generally positive. I already had some experience working with OCaml, and I very quickly had a working prototype of Glöckner's QFMs running. This early version did not yet use the finite sums approach explained above but instead relied on a Montecarlo sampling over the range $[0, 1]$ and it also used the inbuilt floating point numbers for truth values.

I was a bit surprised how quickly I could make progress in implementing the QFMs and getting basic prototypes working. Of course, these often proved to have numerous bugs and unforeseen issues. Especially the finite sums for integrals took quite a few iterations to get working well, though most bugs turned out to be fairly small things (such as the weighting function m in the finite sums having to consider the ϵ value, for the Glöckner QFMs)

The idea to shift from floating point numbers to arbitrary precision numerals came while I was working on the Discussion chapter. Some properties were sensitive to very small changes in truth values, for example monotonicity. So it seemed infeasible to keep working with a representation that was inherently susceptible to rounding errors. I quickly read about arbitrary precision representations and how I could use them in OCaml. For this end, I used the *num* package which is part of the OCaml standard library before version 4.06 and is continued as a separate library after it. The change involved replacing operators like $+, -$ and $/$ with special versions in the *num* package, such as $+/$, $-/$ and $//$.

This was the first time I developed a graphical user interface in OCaml, and I expected it to take quite a while to get running and looking well. Again, I was positively surprised how easy this turned out to be. OCaml provides bindings to GTK+, a cross-platform widget toolkit for graphical user interfaces. The entire code for the user interfaces turned out to have less than 200 lines of code, far less than I had experienced for imperative languages, such as C/C++ or Java. It took a bit work to get an OCaml environment running under Windows. But once it was set up, it also produced a native application that runs, and has been tested for Windows 10. Unfortunately, the window decorations look a bit outdated in the Windows version, but this seems to be due to issues with GTK itself, I was unable to find a way to improve or fix them.

For the plotting functionality, I settled on the cross-platform PLPlot library. It provided support for both 2D and 3D plots, allowing me to render functions of type $[0,1] \to [0,1]$ with 2D plots and functions of type $[0,1]^2 \to [0,1]$ with 3D plots. The former are the truth functions for the proportional semi-fuzzy quantifiers and the latter are the induced truth functions for conjunction and disjunction that are produced by the QFM, as explained in section 6.1. I decided to include a small widget in QFMtool that allows the user to see these plots, and to select the involved SFQs and QFMs. For performance reasons, I render all the plots before hand, and the tool simply has accesses them in runtime.

CHAPTER 8

Conclusion

As part of this thesis a presentation of various recently published QFM models has been provided. These models have been developed after Glöckner's monograph introduced the concept of and argued for the need for a mechanism lifting semi-fuzzy quantifiers to fuzzy quantifiers. Besides three chosen QFMs from Glöckner's monograph [Glö06], four probabilistic QFM models have been developed by Díaz-Hermida et al. [DHBCB04, DHLBB05], one model using a novel level-representation for fuzzy sets from Sánchez et. al [SDV11] and the closeness-based QFM from Baldi and Fermüller [BF18], which uses Giles' game as a foundation for its semantics and aims to embed fuzzy quantifiers into ukasiewicz logic.

A separate chapter has been devoted to the analysis of the QFM models listed above with respect to the DFS axioms, which Glöckner considered critical for any "plausible interpretation of fuzzy quantifiers". These six criteria form together the basis of many other properties that are derivable from them. Since many QFM models are motivated by different goals than the ones Glöckner had in mind, it is not surprising that these more recent approaches might prefer other properties. Of the presented models, the only one for which it is known that it is compatible with the DFS axioms, and which was not developed by Glöckner himself, is $\mathcal{F}_A$ from Díaz-Hermida, which was also presented as part of his doctoral thesis. For all the other ones, conflicts can be found, and have been presented in at least one of the examples in chapter 6.

In addition to the previous two points, a tool, called simply QFMtool, has been developed. It implements all presented QFM models on a semantic level, and, was also of help during the writing of chapter 6. QFMtool supports the evaluation of fuzzy quantified expressions, for a number of predefined semi-fuzzy quantifiers, and the presented QFMs. It also allows one to easily choose from a number of examples of quantified statements and the user is encouraged to change the QFMs and semi-fuzzy quantifiers involved, to see how the results vary. QFMtool has been written in the functional programming language OCaml [LDF+14], and the tool has been built for the Linux distribution Ubuntu, version 17.10, and for Microsoft Windows 10. The tool, including the source code and compiled executables for Linux and Windows, can be accessed at `www.github.com/cem-okulmus/QFMtool`.

8.1 Future work

This section is meant to highlight selected issues that go beyond the framework of Glöckner and is intended to give the reader an outlook into the future work that could be done in this area.

8.1.1 Issues with linguistic adequateness and modelling intensionality

An important point to keep in mind when modelling fuzzy quantification is to try to be "linguistically adequate". This term is unfortunately a bit vague, but is generally understood to apply to formal systems that can be useful for modelling natural languages in a way that is compatible with the basic assumptions made in scientific studies of natural language. As noted by Fermüller [Fer15], however, linguists mostly eschew working with fuzzy models and prefer working with binary notions (such as "accepted" or "not accepted"). There are of course many reasons why one might wish to avoid fuzzy notions, such as assuming that the vagueness disappears if a more precise context is found. This thesis will not go further in this direction. Another interesting reason presented in [Fer15], however, is that there is one relevant notion explored in linguistics that is supported neither in Zadeh's nor in Glöckner's model: namely *intensionality.*

As the word suggests, intensionality is about the intended target of given arguments. In this context, intensionality refers to meaning-related dependencies between the range and scope predicates. For this reason, this can be essentially ignored for unary quantifiers, such as absolute quantifiers from the Zadeh framework. It is only with binary quantifiers, such as proportional quantifiers, that this problem appears. This is best understood by an example.

Example 7. *This example follows one from [Fer15], adjusted here to use semi-fuzzy quantifiers and QFMs already presented.*

We assume two fuzzy concepts ***child*** *and* ***poor****. As the quantifier we choose* ***almost all*** *and we use the $\mathcal{M}$ QFM from Glöckner. The two concepts shall be represented by the following fuzzy sets:*

$\boldsymbol{child} = \{(Tom, \frac{1}{2}), (Sue, \frac{1}{2}), (Marcus, \frac{1}{2}), (Michael, \frac{1}{2}), (Mary, \frac{1}{2})\}$
$\boldsymbol{poor} = \{(Tom, \frac{1}{2}), (Sue, \frac{1}{2}), (Marcus, \frac{1}{2}), (Michael, \frac{1}{2}), (Mary, \frac{1}{2})\}$

The evaluations are:
$\mathcal{M}[\boldsymbol{almost\ all}](child, poor) = \frac{1}{2}$
$\mathcal{M}[\boldsymbol{almost\ all}](child, child) = \frac{1}{2}$

Clearly, as it is used in natural language, we expect the expression "almost all children are children" to be treated differently than "almost all children are poor", regardless of *how the set "children" and "poor" are represented. To respect intensionality, we have to take into account that different precisifications of the range and scope predicates may result in different evaluations, even if the predicates coincide if presented as fuzzy sets.*

Fermüller notes that it is not possible to capture intensionality if one only considers fuzzy sets for fuzzy concepts (or predicates), as is done in both the Zadeh and Glöckner approaches. In other words, the *extension* (the sets used to model the arguments) is not enough to capture the intention behind them. It should also be mentioned here that this is essentially a different way of representing and thinking about vague concepts and that the reason why the prior methods cannot address is more due to having different paradigms in mind. Furthermore, the example shows that there is a *modal aspect* in modelling binary (or generally speaking, more than unary) quantification, which cannot be captured in a purely truth functional approach.

8.1.2 Combining the QFM framework with MFL

As part of his QFM framework, Glöckner also develops what he refers to as the "Induced Propositional Logic", explained in detail in chapter 6. The IPL provides truth functions for conjunction, disjunction, negation and implication. The implication is defined directly via the negation and disjunction as in classical logic:

$$\tilde{\Rightarrow}(x_1, x_2) = \tilde{\vee}(\tilde{\neg} x_1, x_2).$$

On the other hand, one of Hájek's stated design goals for MFL was, as we recall, that the truth function for implication should be derived from the truth function of conjunction $*$, as its residuum:

$$x \Rightarrow y = max\{z \mid x * z \leq y\}.$$

Glöckner himself notes this discrepancy in his monograph ([Glö06], p. 156):

> Some readers might prefer a different choice of the implication operator, namely $x_1 \tilde{\Rightarrow} x_2 = min(1, 1 - x_1 + x_2)$. However, it is clear that every QFM with the highly desirably property of preserving Aristotelian squares will also preserve the interdefinability of connectives, and therefore differ from ukasiewicz logic.

With IPL he provides a truth functional fuzzy logic to be used, presumably, with his framework for fuzzy quantifiers. The IPL, however, is quite different from the MFL approach. Since Hájek first introduced it, there has been a plethora of research for MFL in model theory, proof theory and other fields [CHN15]. This makes it desirable to model fuzzy quantifiers within MFL: one possible scenario might be to develop a syntactic systems that can reason with fuzzy quantifiers, proving that some fuzzy quantified expression must hold for a given premise, for example.

Fermüller and Roschger [FR13] and Baldi and Fermüller [BF18], showed how to model fuzzy quantifiers in MFL, through extensions of the Giles' game for ukasiewicz logic, which we discussed in section 3.6. In [FR13] the authors extended Giles' games with semi-fuzzy (Type III) quantifiers, see Section 4.2. This was further extended in [BF18], where in the same setting the authors introduced the closeness-based QFM, to deal with fully fuzzy (Type IV) quantifiers.

List of Figures

List of Tables

List of Algorithms

Bibliography

[BC81] Jon Barwise and Robin Cooper. Generalized Quantifiers and Natural Language. *Linguistics and Philosophy*, 4(2):159–219, 1981.

[Běh08] Libor Běhounek. On the difference between traditional and deductive fuzzy logic. *Fuzzy Sets and Systems*, 159(10):1153–1164, 2008.

[BF18] Paolo Baldi and Christian G. Fermüller. From Semi-fuzzy to Fuzzy Quantifiers via Łukasiewicz Logic and Games. In Janusz Kacprzyk, Eulalia Szmidt, Sławomir Zadrożny, Krassimir T. Atanassov, and Maciej Krawczak, editors, *Advances in Fuzzy Logic and Technology 2017*, Advances in Intelligent Systems and Computing, pages 112–124. Springer, 2018.

[CHN15] Petr Cintula, Petr Hájek, and Carles Noguera. *Handbook of Mathematical Fuzzy Logic, Volume 3*, volume 58 of *Studies in Logic, Mathematical Logic and Foundations*. College Publications, 2015.

[DH09] Antonín Dvořák and Michal Holčapek. L-fuzzy quantifiers of type $< 1 >$ determined by fuzzy measures. *Fuzzy Sets and Systems*, 160(23):3425–3452, 2009.

[DHBCB04] Félix Díaz-Hermida, Alberto Bugarín, Purificación Cariñena, and Senén Barro. Voting-model based evaluation of fuzzy quantified sentences: a general framework. *Fuzzy Sets and Systems*, 146(1):97–120, 2004.

[DHLBB05] Félix Díaz-Hermida, David E Losada, Alberto Bugarín, and Senén Barro. A probabilistic quantifier fuzzification mechanism: The model and its evaluation for information retrieval. *IEEE Transactions on Fuzzy Systems*, 13(5):688–700, 2005.

[DRSV14] Miguel Delgado, M Dolores Ruiz, Daniel Sánchez, and María-Amparo Vila. Fuzzy quantification: a state of the art. *Fuzzy Sets and Systems*, 242:1–30, 2014.

[Fer15] Christian G Fermüller. Combining Fuzziness and Context Sensitivity in Game Based Models of Vague Quantification. In *International Symposium on Integrated Uncertainty in Knowledge Modelling and Decision Making*, Lecture Notes in Computer Science, pages 19–31. Springer, 2015.

[FR13] Christian G Fermüller and Christoph Roschger. Randomized game semantics for semi-fuzzy quantifiers. *Logic Journal of IGPL*, 22(3):413–439, 2013.

[Gil74] Robin Giles. A non-classical logic for physics. *Studia Logica*, 33(4):397–415, 1974.

[Glö06] Ingo Glöckner. *Fuzzy Quantifiers: A Computational Theory*, volume 193 of *Studies in Fuzziness and Soft Computing*. Springer, 2006.

[Hol08] Michal Holčapek. Monadic l-fuzzy quantifiers of the type $< 1n, 1 >$. *Fuzzy Sets and Systems*, 159(14):1811–1835, 2008.

[Hyd14] Dominic Hyde. Sorites Paradox. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Winter 2014 edition, 2014. URL `https://plato.stanford.edu/archives/win2014/entries/sorites-paradox/`.

[Há06] Petr Hájek. *A Companion to Philosophical Logic*, chapter Why Fuzzy Logic?, pages 595–605. Blackwell Publishing Ltd, 2006.

[LDF+14] Xavier Leroy, Damien Doligez, Alain Frisch, Jacques Garrigue, Didier Rémy, and Jérôme Vouillon. The OCaml system release 4.02. *Institut National de Recherche en Informatique et en Automatique*, 54, 2014.

[LK98] Yaxin Liu and Etienne E Kerre. An overview of fuzzy quantifiers. (I). Interpretations. *Fuzzy Sets and Systems*, 95(1):1–21, 1998.

[Nov08] Vilém Novák. A formal theory of intermediate quantifiers. *Fuzzy Sets and Systems*, 159(10):1229–1246, 2008.

[PW06] Stanley Peters and Dag Westerståhl. *Quantifiers in Language and Logic.* Oxford University Press, New York, 2006.

[SDV11] Daniel Sánchez, Miguel Delgado, and María-Amparo Vila. An Approach to General Quantification Using Representation by Levels. In *Proc: WILF 2011*, volume 6857 of *Lecture Notes in Artificial Intelligence*, pages 50–57. Springer, 2011.

[Wes16] Dag Westerståhl. Generalized Quantifiers. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Winter 2016 edition, 2016. URL `https://plato.stanford.edu/archives/win2016/entries/generalized-quantifiers/`.

[Yag98] Ronald R Yager. On Ordered Weighted Averaging Aggregation Operators in Multicriteria Decisionmaking. *IEEE Transactions on Systems, Man, and Cybernetics*, 18(1):183–190, 1998.

[Zad65] Lotfi A Zadeh. Fuzzy sets. *Information and Control*, 8(3):338–353, 1965.

[Zad83] Lotfi A Zadeh. A computational approach to fuzzy quantifiers in natural languages. *Computers and Mathematics with Applications*, 9(1):149–184, 1983.