Fuzzy Sets and Signatures✍

Patrik Eklund[a]

[a] *Department of Computing Science, Umeå University, Sweden*

Abstract

This paper intends logically and categorically to explore, and whenever possible, expose the underlying syntax, including signatures, terms and sentences, of possibility theory [1] and possibilistic logic [2]. The categorical machinery is the one adapted for formalization of fuzzy terms [3].

1. The coming age of logic

At the IFSA Congress in Seattle 1989, Lotfi Zadeh spoke about *The Coming Age of Fuzzy Logic*. Did it ever arrive? Is it already here? Was it there already at that time? Some say it was, some say it never came, and some ask the question "Will it ever?".

Lotfi Zadeh's very legitimate promotions of quantifiers and modifiers are seemingly harmless and easy to comprehend, but in a formal logic context quite hard to formulate. Theoreticians and practitioners have tried to solve it during the last decades. Are the solutions those we really want? Do they really bring fuzzy logic towards *the age that came*?

Logic always needs an underlying signature, its "alphabet", as a starting point. Terms are given by formal term constructions, and variables are specific terms. The "object of variables" is an important ingredient for the term construction, and terms in turn are the building blocks in sentences. Sentences appear in "structured conglomerates of sentences", in turn manipulated by entailments, given some particular inference rules making the selected logical machinery work as a whole.

Uncertainty can be identified and studied in all these subareas. Terms can be fuzzy in different ways, e.g., providing the distinction between "fuzzy computing" and "computing with fuzzy". Even the alphabet can be fuzzy. Sentences and their conglomerates can be uncertain. The inference rules can be equipped

✍This is a note on a logical and categorical view concerning an underlying language and formalism needed to resolve some important issues in the discussion on commonality and diversity of respective concepts for 'probability' and 'possibility'.

Email address: `peklund@cs.umu.se` (Patrik Eklund)

URL: `http://www.glioc.com`

with uncertainties and even proof "trees" can be seen in a more generalized context.

Since the beginning of "Fuzzy Sets and Systems", fuzzy logic has not emerged to embrace much of these generalized aspects. The fuzzy community still concentrates to a large extent on "fuzzy sets" and applications, that in particular in engineering are mostly seen as "fuzzy systems". Fuzzy logic was and still is mostly and only derived from algebraic properties of the set of truth values. This note tries to open up some perspectives, given recent developments on formal approaches to fuzzy signatures and terms, and from there on to fuzzy sentences and theoremata.

2. Truth and the set of truth values

How do we explain truth? Where does it come from? When is it, and when isn't it? What about false? Is it kind of symmetric to truth? One is antithetic to the other, but is that being antithetic a symmetric phenomenon, or is one of them in a special role? Of course, these are classical questions and debates, but we should be entitled to ask these questions again, and in light of more formal languages for uncertainty representations in logic. Logicians more than half a century ago did well indeed, but there is no need to overestimate the value of what was written and developed at that time. If we do, we may be blocking further development, and we may unnecessarily constrain ourselves not to ask new and logically innovative questions.

The question about 'right' and 'wrong' is similar. This issue, however, is perhaps more ethical, and aesthetical, than logical, and, clearly, doing things right is preferable over doing things wrong. Another way to say this is that we intuitively prefer to see 'doing wrong' as 'not doing right', and not the other way around. Further, we might prefer to say "if we wouldn't have anything 'wrong', then we wouldn't have anything 'right' either". However, similarly saying "if there wouldn't have anything 'right', then we wouldn't have anything 'wrong' either", does not similarly comply with our understanding of ethics.

Is 'truth' then in a similar role as 'right'? Surely, in particular in decision processes e.g. involving wellness and welfare of humans and humanities. In formal set theory we have $a \in b$ for "a is an element of the set b". In the case of $a \notin b$, we would express it like "a is 'not' an element of the set b", or "no, a is not an element of the set b", or even "it is 'not' so that a is an element of the set b". Note also that it is not clear if we are always allowed to say "no, a is not an element of the set b, but a is an element of ...". If $a \notin b$, there is no *a priori* set c for which $a \in c$. Obviously, if we would do that, the the complement set of b is a candidate, but complement then assumes the existence of some chosen "universe around b". Note that complement is an operator, taking two sets into a new set. Existence of an overall "universe" U is sometimes desirable in set theory, and in category theory, when providing meta statements, but we cannot operate on such sets, similarly as we have to be careful when we deal with "the category of all categories".

Also concerning 'order', there is a fundamental subordination principle. Even if any relation has a symmetric counterpart, when saying "a is 'related' to b" we are really saying that the pair (a, b) is an element of a certain relation as a set of pairs, usually a cartesian product of two sets. In this case a is 'ordinated' with b, and whether we prefer to call it being 'subordinated' or 'superordinated' is not important. What is important is that they are not mutually 'coordinated', but indeed ordinated according to a certain 'order'. In his *Principles of Arithmetic* (1889), Peano writes "we shall generally write signs on a single line", and this is then done 'one-by-one' and in a certain 'order'.

In set theory, the existence of the empty set $\varnothing$ is axiomatized. No other set is axiomatized in this way, even if clearly the existence of certain sets are axiomatized, like those produced by some constructions, e.g., using the axioms of infinity and power set. Given the empty set, we can construct sets like $\{\varnothing\}$, $\{\varnothing, \{\varnothing\}\}$, $\{\varnothing, \{\varnothing\}, \{\varnothing, \{\varnothing\}\}\}$, and so on. In set theory, these sets, including the empty set, and those originating from that empty set, are usually denoted $0, 1, 2, 3$, and so on. Thus we could say that "without 0, we do not have 1", similarly as we may say "without *wrong*, we do not have *right*", or "without *false*, we do not have *true*".

When we provide a "semantic domain of truth values", we usually select a set like $\{false, true\}$, and even if we do not explicitly say which sets[1] $false$ and $true$ denote, their roles and origin are supposed to be different, and they are supposed to be subordinated, one to the other, a subordination that remains the same throughout. It would be no controversy at all to say that $false = \varnothing$ and $true = \{\varnothing\}$, or $false = 0$ and $true = 1$. Whatever we prefer to do, we have to be very honest not to assume anything hidden residing in $false$ and $true$, as 'points', i.e., sets, in that "semantic domain of truth values".

Clearly, we do not want to restrict to having "binary truth" only, so we want to explore the situation having more than two 'elements' in such a "set of truth values". The 'structure' of that set is then more complicated, and a number of choices for that structure indeed become available.

3. Truth and provability

Aristotle's "logic" has been mathematically transformed into what we today call propositional and predicate logic, respectively. There are other logics, and it must be observed that Aristotle never anticipated these ones. Further, Aristotle never cared about types and typing, which perhaps is the greatest weakness of ancient Greek logic.

On syllogism we also have to note that it is related to inference, so $A \vdash B$ is a syllogism, not to be confused with the implication $A \Rightarrow B$, which is a sentence

[1]Note that in formal set theory, e.g., like in ZFC, there are strictly speaking no 'points' or 'elements', as they are all 'sets'. In other formalizations of set theory we may speak of 'sets' and 'classes', but also here there is nothing that is either a set or class. Sets belong to sets, or sets may belong to classes, but classes may not belong to classes.

in propositional or predicate logic, depending on if A and B are propositional constants or predicates. Indeed, since A and B are sentences, then also $A \Rightarrow B$ is a sentence, but the syllogism $A \vdash B$ is not a sentence. Here, of course, many readers think about the "deduction (meta)theorem" in first-order logic, where the temptation is to say "if $A \vdash B$ is true, then $A \Rightarrow B$ is true", but this is not correct since the theorem states "if $A \vdash B$ is true, then $A \Rightarrow B$ is provable", i.e., "$\vdash A \Rightarrow B$" is true. Needless to say, we have to be very careful when using 'true' in natural language.

Aristotle didn't clearly distinguish between truth and provability. Aristotle said "a true conclusion may come through what is false". What is here a "true conclusion"? If B is true, then $\texttt{false} \Rightarrow B$ is also true. Is B then the conclusion, or is "$\texttt{false} \Rightarrow B$ is true" the conclusion, or is it in fact "$\vdash \texttt{false} \Rightarrow B$ is true", i.e., "$\texttt{false} \Rightarrow B$ is provable"? Aristotle also speaks about "the same terms", and then the question is what he means by a "term". Saying "positive terms in positive syllogisms" indicates that terms are sentences, but then "positive" may have at least two different meanings. In his statement "it is impossible that the same thing should be necessitated by the being and by the not-being of the same thing", Aristotle then mixes truth and provability, and trying to make that into a "sentence". Aristotle's statement "just as if it were proved through three terms" clearly reveals how Aristotle becomes intertwined since he does not separate truth from provability.

Obviously we can examine Aristotle's *Prior Analytics* endlessly, trying to figure out what Aristotle really said. One thing is very clear. Had Aristotle been in Göttingen during the 1920's, he would have rewritten his *Prior Analytics* completely.

Mixing truth and provability also takes place in expressions like

$$\Pi(\texttt{CONTRADICTION}) > \Pi(\texttt{NOT}\ P)$$

frequently seen in discussions about possibilistic logic. Obviously, $\texttt{NOT}\ P$ is something for which we can try to establish truth, whereas CONTRADICTION is the results when something wasn't provable, so the same Π shouldn't valuate degrees of truth and provability at the same time, in the same bag.

A well known but equally accepted mixing of truth and provability happens in the foundations for Gödel's numbering. Kleene [4] provides some preparatory ideas and notations about general recursive functions, and then also uses a predicate symbol A, and a predicate $A(x)$, speaking about "$A(x)$ is provable", and using the notation "$\vdash A(x)$". Kleene then proceeds to create a "metamathematical proposition" $\mathfrak{R}(x, Y)$, representing "Y is a proof of $A(x)$", and then proceeds to write

$$(\exists Y)\mathfrak{R}(x, Y) \ \equiv \vdash A(x)$$

but at the same time wondering "What is the nature of the predicate $\mathfrak{R}(x, Y)$?". Kleene then by-passes formalism by saying it must be an "effectively decidable" metamathematical predicate, and that "there must be a decision procedure or algorithm for the question whether $\mathfrak{R}(x, Y)$ holds". Mathematical propositions and metamathematical propositions are thus again allowed to be in the same bag.

Obviously, we may then want to be a bit careful (Gödel wasn't) also about "the set of all proofs ...", and furthermore, about numbering proofs and reasoning about such numberings. Making statements about such numbers is dubious, as is calling these statements "predicates", throwing them into the same bag, where $A(x)$ and $\mathfrak{R}(x, Y)$ already mingle with each other, with $\exists$ playing in the band.

4. Algebraic operation and truth valuation

In logic we need to distinguish between 'term' and 'sentence', even if "elements", like in propositional logic, sometimes may appear simultaneously[2] as both 'term' and 'sentence'. This then invites to thinking that we have sets that are more like "sets of truth values", and other sets that are more like "sets of other types of values than truth values". This is fundamentally important in universal algebra, and we must be very careful not to "mix bags".

Let us still be informal at this point, and suppose A is a "set of other types of values than truth values" and B is a "set of truth values". Then a mapping $f : A \times A \longrightarrow A$ is more like an 'algebraic operation', typically subject to some properties annotated with that operation, and mappings $g : A \longrightarrow B$ and $h : A \times A \longrightarrow B$ are more like 'truth valuation' and 'entailment'. Clearly, $u : B \longrightarrow B$ or $v : B \times B \longrightarrow B$ are something like "algebraic manipulation of or operators on truth values". Note how g can be seen as providing "modes of truth" for 'elements' in A, whereas h can be seen as representing "mode of ordination".

Obviously, the most common selections for B is $\{false, true\}$, or some lattice L, typically complete, focusing on its order related properties, or a quantale Q as a partially ordered algebraic structure, rich enough to embrace desired operations also of more algebraic type. A quantale is a partially ordered structure with a multiplication $* : Q \times Q \longrightarrow Q$ satisfying certain properties. A unital quantale contains a unit element, which again calls for more elaborate interpretations of the 'set of truth values', also for the reason that the 'unit' and the 'top' element may be different. The application domain usually specifies the the scope and properties of such sets of truth values. The unit interval $[0, 1]$ is also frequently used, and the most common view of a fuzzy set is that of being represented by a function $\alpha : A \longrightarrow L$, where A is not necessarily assumed to be more than just an unstructured set, and where L is a lattice with some suitable structure.

[2] In propositional logic, 'term' can be equated with 'sentence', since the sentence functors in propositional logic is defined as the identity functor.

5. Observation and its trustworthiness

Where do numbers and values in general come from, and can we trust them? Are they trustworthy? Are they 'truthworthy'? When we estimate the trustworthiness of an observation we also need to consider the truthworthiness of the observer. If Hans' wife says "Hans ate 2 eggs for breakfast yesterday", then both the observation (2 eggs) and the observer (Hans' wife) are quite trustworthy if the eggs were boiled, and Hans' wife doesn't e.g. suffer from cognitive decline of certain degree. If, on the other hand, Hans was eating scrambled eggs, prepared by himself without his wife being in the kitchen at the time of cooking, then the observation comes with an uncertainty, which is due to the ability of Hans' wife to estimate the number of eggs used in a dish including scrambled eggs. If the observer suffers from cognitive decline, the uncertainty of '2 eggs' comes from other sources.

Observation 5.1. *In data gathering, we should always be as accurate as possible in trying to estimate uncertainty of both observation and observer.*

The following example used in [5] uses the geriatric depression scale `GDS` [6] to illuminate the distinction between uncertainty of the observer and the observation. `GDS` includes 30 questions, where the older person is asked to reply either `yes` or `no` to each question. One of the questions, the first one, is the following:

> *Are you basically satisfied with your life?* (`NO/yes`)

Replying 'no' would score one point. Clearly there is an uncertainty if an older person replies actively, but if a home care social worker is asked to 'observe it', there will be an uncertainty attached also to the 'operation of observing'. See [5] for a detailed discussion, and also for formalism related to underlying signatures.

6. Syntax and semantics

Which comes first, syntax or semantics? This would be an endless debate, and there is no formal language, meta or otherwise, to deal with this questions. Some prefer to work from semantics "backwards" to syntax, and ensure properties like 'soundness' and 'completeness' along such pathways of scientific exploration. Others start off from syntax, trying to develop a language rich enough to express desired phenomena. A syntax which is too rich may not "communicate well" with a poor semantics. However, the fundamental question here is whether or not a "semantic jacket" is to be allowed to moralize or patronize on choices of syntax, or is richness of syntax, for whatever reason, to be seen as a burden on semantics actually to allow for this richness. It may well be that algebra and universal algebra simply comes short, and the traditional understanding of semantic structures must be revisited.

Syntactically we may have sorts like `s` and `bool`, and syntactic operators like $\omega : \mathtt{s} \times \mathtt{s} \to \mathtt{s}$ and $\mathtt{a} : \mathtt{s} \to \mathtt{bool}$, as part of a certain signature $\Sigma = (S, \Omega)$, where $\mathtt{s}, \mathtt{bool} \in S$ and $\omega \in \Omega$. The algebra of Σ, often called the Σ-algebra, and may informally be denoted $\mathfrak{A}(\Sigma)$ or $\mathfrak{A}_\Sigma$, then describes how to bind a syntactic object with a semantic counterpart. We may desire to bind $\mathtt{s}$ and $\mathtt{bool}$, respectively, with A and B, usually written $\mathfrak{A}(\mathtt{s}) = A$ and $\mathfrak{A}(\mathtt{bool}) = B$, or $\mathfrak{A}_\Sigma(\mathtt{s}) = A$ and $\mathfrak{A}_\Sigma(\mathtt{bool}) = B$, making the underlying signature more visible in context. Similarly, binding $\omega : \mathtt{s} \times \mathtt{s} \to \mathtt{s}$ and $\mathtt{a} : \mathtt{s} \to \mathtt{bool}$, respectively to $f : A \times A \longrightarrow A$ and $g : A \longrightarrow B$, usually written $\mathfrak{A}(\omega) = f$ and $\mathfrak{A}(\mathtt{a}) = g$, so that $\mathfrak{A}(\omega : \mathtt{s} \times \mathtt{s} \to \mathtt{s}) : \mathfrak{A}(\mathtt{s}) \times \mathfrak{A}(\mathtt{s}) \longrightarrow \mathfrak{A}(\mathtt{s})$ is $f : A \times A \longrightarrow A$ and $\mathfrak{A}(\mathtt{a} : \mathtt{s} \to \mathtt{bool}) : \mathfrak{A}(\mathtt{s}) \longrightarrow \mathfrak{A}(\mathtt{bool})$ is $g : A \longrightarrow B$.

All this is well understood and comprehended by readers that are already well versed in universal algebra, and experienced users of many-sorted signatures, in particular. However, those readers that usually prefer intuitively to be "one-sorted", thus not considering the role of sorts and typing, may find it useful to read this "algebraic binding" more carefully.

At this point the understanding of sets and functions as tools in logic is understanding how the semantics of syntactic expressions is written by means of an underlying (formal) set theory. In this presentation we also adopt the fundamental distinction between 'logic' as in "logic for mathematics" and 'logic' in "mathematics for logic". The former is the *fons et origo* logic appearing from nowhere (or from an intuitive understanding of natural numbers), evolving hand in hand with set theory, to become what it is by early 20th century, and commonly understood still today. The latter is the one we build upon the language of category theory, where we make clear distinctions between signature, term, sentence, and so on, and we are (categorically) constructive about each and every bit in this overall structure we call 'logic'.

Now note how we may choose $\mathfrak{A}(\mathtt{bool})$ to be $\{false, true\}$, $[0, 1]$ or L, or something else, depending on the requirements as provided by the application. Also note that a fuzzy set $\alpha : A \longrightarrow L$, logically speaking, is potentially useful in some semantic binding of some syntactic operator $\mathtt{a} : \mathtt{s} \to \mathtt{bool}$, whenever $\mathfrak{A}(\mathtt{s}) = A$ and $\mathfrak{A}(\mathtt{bool}) = L$. In such a case the semantic binding is completed once we set $\mathfrak{A}(\mathtt{a}) = \alpha$. This a really an important observation concerning fuzzy sets, i.e., how they are logically candidates for being the semantic counterparts of corresponding syntactic operators.

The following example was briefly discussed in [3]. Consider distance estimation on a golf course. Low handicappers are fairly good at estimating distances $x \pm 10\%$, and choice of club both for wedges and longer irons require a quite accurate distance estimation to combine with evaluation of hazards of various kind. A higher handicapper facing an iron shot of 150 m may focus more on producing a straight shot rather than ending up on the green with the ball becoming 'pin high'. Golf courses typically have distance markers at 100 m and 150 m, so an estimation of a 125 m distance is rather easy. However, for the purpose of this example, we will ignore these fixed markers, and look at the ability to estimated distance. We may also use units of 10 m, so by $n \in \mathbb{N}$ we intuitively mean the index for "$n * 10$ meters $\pm 10\%$". Clearly, other assignments could also be considered.

Note also that a wedge shot from 60 m is more likely to result in the ball being closer to the flag, thus producing a good birdie chance, as compared to an iron shot from 160 m into the green, which may leave a longer put to finish. Concerning semantic domains, $\mathbb{N}$ understood in this context may then be useful, where clearly numbers e.g. in $\{4, \ldots, 20\}$ are more important for shots from the fairway. Smaller numbers than that typically mean a 'chip', and larger numbers means going for the 'driver'. Syntactically, we may think in terms of having a sort `iron` in the "Golf signature" $\mathtt{GOLF} = (S, \Omega)$, with each iron in the golf bag being a constant of that type. With irons from 'iron 4 (I4)' to the 'pitching wedge (PW)', and other special wedges with loft angles being 52, 56 and 60 degrees, we could write $\mathtt{I4}, \mathtt{I5}, \mathtt{I6}, \mathtt{I7}, \mathtt{I8}, \mathtt{I9}, \mathtt{PW}, \mathtt{W52}, \mathtt{W56}, \mathtt{W60} :\to \mathtt{iron}$ for those constants. Selection of club is then based on each player "internal rule base", in turn based on skill and experience, combined with knowledge about the character of the golf course being played. A typical rule considers the lie of the ball on the fairway (or in the rough), hazards around the green, possibility to 'hold the green' given the effect of backspin depending on how much pitch mark the ball leaves on the surface of the green, and so on. The final decision is selection of club and strategy of shot. All this then goes into execution of that shot, motorized by the golf swing. Various errors and uncertainties then result in ball position away from the flag, or even from the green.

The GOLF-algebra now becomes interesting, and it is immediately clear that simply having $\mathfrak{A}_{\mathtt{GOLF}}(\mathtt{iron}) = \mathbb{N}$ would not make enough sense, since e.g. $\mathfrak{A}_{\mathtt{GOLF}}(\mathtt{I8})$ must mean more than just distance. Note also how syntactic rules may be quite similar for lower and higher handicapper even if their respective algebras $\mathfrak{A}_{\mathtt{GOLF}}^{low-hcp}$ and $\mathfrak{A}_{\mathtt{GOLF}}^{high-hcp}$ may be quite different. Further, uncertainty modelling from syntactic point of view is more about "where and why" shots may not be perfect, where as semantically it means estimations about "how much". This in turn means that we most likely want to consider modelling "$n * d^{low-hcp}$ meters $\pm p^{low-hcp}\%$" differently from "$n * d^{high-hcp}$ meters $\pm p^{high-hcp}\%$". Such considerations might be useful to adapt rules e.g. for "playing scramble" in different forms.

In discussions of which comes first, syntax or semantics, this example clearly speaks well in favour of the importance of syntax.

7. Signatures and terms

For a detailed exposition and strictly formal treatment of 'signatures over categories', we refer to [3]. Here we capture some notation needed for this paper, and also include the "three level" arrangement of signatures using type constructions.

In traditional computer science notation, a many-sorted signature $\Sigma = (S, \Omega)$ consists of a set S of sorts (or types), and a set Ω of operators. For the purpose of this paper we view S as an index set (as residing in ZFC), whereas Ω can be arranged to be an object in a product category[3] $\prod_S \mathtt{C}$, which we denote $\mathtt{C}_S$.

[3] For a category $\mathtt{C}$, we write $\mathtt{C}_S$ for the product category $\prod_S \mathtt{C}$. The objects of $\mathtt{C}_S$ are tuples

If $\mathtt{C} = \mathtt{Set}$, then the signature is 'crisp', i.e., uncertainty is not allowed in the "alphabet". However, if $\mathtt{C}$ is the Goguen category $\mathtt{Set}(\mathfrak{Q})$, where $\mathfrak{Q} = (Q, \vee, \odot)$ may be a quantale, then operators as well as variables always come annotated with uncertainty values. These values may be selected as '1' or '*true*' in Q, but other truth values are now enabled. See [3] for detail.

In the many-sorted case, with $\Omega = (\Omega_{\mathtt{s}})_{\mathtt{s} \in S}$, where $\Omega_{\mathtt{s}}$ is an object in $\mathtt{C}$, we say that $\Sigma = (S, \Omega)$ is a *signature over* $\mathtt{C}$. In the case of $\mathtt{C} = \mathtt{Set}$, '$n$-ary' operators in $\Omega_{\mathtt{s}}$ are syntactically written as $\omega : \mathtt{s}_1 \times \cdots \times \mathtt{s}_n \to \mathtt{s}$. The 0-ary operators $\omega :\to \mathtt{s}$ are the 'constants'. Notice how $\times$ and $\to$ in the syntactic notation for operators actually come without any meaning, until respective Σ-algebras provide $\times$ and $\to$ with semantics, again expectedly in some underlying category.

Intuitively speaking, terms are then produced by signatures such that variables and constants are terms, and if $t_1, \ldots, t_n$ are terms, then also $\omega(t_1, \ldots, t_n)$ is a term, where ω resides in the underlying signature. For the formal term functors construction, see [3]. Note that term functors extendable to being monads are key in order to allow substitutions to be composable. This 'Kleisli property' is very important in generalized contexts. Without that property, logic is a torso.

Example 7.1. A signature $\Sigma_{\mathtt{NAT}}$ (over $\mathtt{Set}$) , or simply written $\mathtt{NAT}$, for the natural numbers, being the signature for the *Peano axioms* as a set of *sentences*, is usually presented as $\mathtt{NAT} = (\{\mathtt{nat}\}, \{\mathtt{0} :\to \mathtt{nat}, \mathtt{succ} : \mathtt{nat} \to \mathtt{nat}\})$. This is a one-sorted signature, and with X as the set of variables, i.e., X as an object of the underlying category $\mathtt{Set}$, the 'set of terms over the signature $\mathtt{NAT}$ and with variables in X' is provided using the term functor $\mathsf{T}_{\mathtt{NAT}} : \mathtt{Set} \longrightarrow \mathtt{Set}$. The 'set of terms' $\mathsf{T}_{\mathtt{NAT}}X$ consists of all variables, all $\mathtt{0}, \mathtt{succ}(\mathtt{0}), \mathtt{succ}(\mathtt{succ}(\mathtt{0})), \ldots$, and all $\mathtt{succ}(x), \mathtt{succ}(\mathtt{succ}(x)), \ldots$, for all $x \in X$.

Example 7.2. The very basic signature $\Sigma_{\mathtt{BOOL}}$, or simply $\mathtt{BOOL}$, for Booleans could be

$$\mathtt{BOOL} = (\{\mathtt{bool}\}, \{\mathtt{false} :\to \mathtt{bool}, \mathtt{true} :\to \mathtt{bool}\}),$$

and additional operators can be introduced to represent the Boolean operators. We could add $\& : \mathtt{bool} \times \mathtt{bool} \to \mathtt{bool}$ as an operator and write e.g. $\mathtt{BOOL}_{\&}$ for the corresponding signature, or we could add $\neg : \mathtt{bool} \to \mathtt{bool}$ and call the signature $\mathtt{BOOL}_{\neg}$. We could add both these operators and call it $\mathtt{BOOL}_{\&,\neg}$, or call it $\Sigma_{\mathtt{PL}}$ as 'the underlying signature for propositional logic'. Note how implication '$\longrightarrow$' is implicit, but it doesn't have to appear as an operator in its own right. It could be included, but then somewhere along the line we would, as usual, need to 'identify' a term like $\longrightarrow(x, y)$ with $\neg(\&(x, \neg(y)))$. The role of implication is

$(X_{\mathtt{s}})_{\mathtt{s} \in S}$ such that $X_{\mathtt{s}} \in \mathrm{Ob}(\mathtt{C})$ for all $\mathtt{s} \in S$. We will also use X_S as a shorthand notation for these tuples. The morphisms between objects $(X_{\mathtt{s}})_{\mathtt{s} \in S}$ and $(Y_{\mathtt{s}})_{\mathtt{s} \in S}$ are tuples $(f_{\mathtt{s}})_{\mathtt{s} \in S}$ such that $f_{\mathtt{s}} \in \mathrm{Hom}_{\mathtt{C}}(X_{\mathtt{s}}, Y_{\mathtt{s}})$ for all $\mathtt{s} \in S$, and similarly we will use f_S as a shorthand notation. The composition of morphisms is defined sortwise (componentwise), i.e., $(g_{\mathtt{s}})_{\mathtt{s} \in S} \circ (f_{\mathtt{s}})_{\mathtt{s} \in S} = (g_{\mathtt{s}} \circ f_{\mathtt{s}})_{\mathtt{s} \in S}$. [3]

also interesting when we move over from propositional to predicate logic. Then it is no longer obvious that implication produces a (boolean) term, but rather appear as a sentence (or statement). This is a crucial point for avoiding the mixed bags between 'terms' and 'sentences'.

Example 7.3. (The *signature for Propositional Logic* [7]) Let $\Sigma_{PL} = (S_{PL}, \Omega_{PL})$ be the underlying *two-valued propositional logic* signature, where $S_{PL} = S$, and $\Omega_{PL} = \{\mathtt{F}, \mathtt{T} :\to \mathtt{bool}, \& : \mathtt{bool} \times \mathtt{bool} \to \mathtt{bool}, \neg : \mathtt{bool} \to \mathtt{bool}\} \cup \{\mathtt{P}_i : \mathtt{s}_{i_1} \times \cdots \times \mathtt{s}_{i_n} \to \mathtt{bool} \mid i \in I, \mathtt{s}_{i_j} \in S\}$. Similarly as bool leading to no additional terms, except for additional variables being terms when using Σ, the sorts in S_{PL}, other than bool, will lead to no additional terms except variables. Adding predicates as operators even if they produce no terms seems superfluous at first sight, but the justification is seen when we compose these term functors with T_Σ. See [7] for more detail.

Example 7.4. A signature $\mathtt{NATORD} = (\{\mathtt{nat}, \mathtt{bool}\}, \{\mathtt{0} :\to \mathtt{nat}, \mathtt{succ} : \mathtt{nat} \to \mathtt{nat}, \mathtt{false} :\to \mathtt{bool}, \mathtt{true} :\to \mathtt{bool}, \leq : \mathtt{nat} \times \mathtt{nat} \to \mathtt{bool}\})$, could be introduced as a many-sorted situation where the operator $\leq$ needs both nat and bool. Again, this signature can be extended e.g. to NATADDORD by adding the operator $+ : \mathtt{nat} \times \mathtt{nat} \to \mathtt{nat}$ (reserved for addition, but still semantically or equationally undefined) to the set of operators in NATORD. Similarly, logical operators could be included, and so on.

Example 7.5. The set of terms for the signature $\Sigma = (S, \varnothing)$, $S \neq \varnothing$, consists only of variables, and the empty signature $\Sigma = (\varnothing, \varnothing)$, the 'unsorted situation', produces no terms at all, and no logic to play with.

8. Variables

One of the most important messages of this note is that we have to be very cautious about what we mean by a 'variable'. If we have a variable

$$x$$

it will be 'part of' some 'variable set', or more generally, embraced by some 'object of variables'. If our underlying category is the ordinary category of sets and functions, then the 'object of variables' is just a plain set X, and we have

$$x \in X$$

in the sense of membership as prescribed within the underlying formal set theory.

In logic, we use variables as appearing in 'substitution' and 'assignment', and variables are then never 'unsorted'. They must be identified as having a particular sort. It may be one-sorted, and in this case we may avoid explicit mentioning of the particular unique sort, but nevertheless, we must recognize that we do have one underlying sort to which the variables comply. If we claim that a variable can be 'unsorted', then we really say that the so called 'variable set' is actually just a plain set, i.e., there is no way to understand the distinction between 'substitution' and 'assignment'. This then also means we don't care about syntax and semantics. It's just plain set theory, maybe involving the real numbers, and thereby allowing analysis and mainly physical observation related applications.

If we have NAT as our underlying signature over Set, and $X_{\mathtt{nat}}$ is an object of variables (in Set), then a *substitution* is a mapping

$$\rho : X_{\mathtt{nat}} \longrightarrow \mathsf{T}_{\mathtt{NAT}} X_{\mathtt{nat}},$$

and *assignment* of variables in $X_{\mathtt{nat}}$ to values in the semantic domain $\mathfrak{A}_{NAT}(\mathtt{nat})$ is a mapping

$$\mathfrak{v} : X_{\mathtt{nat}} \longrightarrow \mathfrak{A}_{NAT}(\mathtt{nat}).$$

For $x \in X_{\mathtt{nat}}$, we may then have substitutions like

$$\rho(x) = \mathtt{succ}(\mathtt{succ}(0))$$

and if $\mathfrak{A}_{NAT}(\mathtt{nat}) = \mathbb{N}$, we may have an assignment like

$$\mathfrak{v}(x) = 2.$$

However, note that, at this point, expressions or 'equations' like

$$x = \mathtt{succ}(\mathtt{succ}(0))$$

or

$$x = 2$$

make no sense at all. Once the signature is given, the terms can be constructed, but nothing is *a priori* said about sentences. If we adopt a first-order logic, sentences will have a certain syntactic form, and if we adopt more of an equational style of logic, then $x = \mathtt{succ}(\mathtt{succ}(0))$ will be a properly constructed *sentence*. Note also that, if we have an expression like

$$2 + 1$$

and we would like to have an a to 'represent' this value, or give a shorthand notation for it, then we informally write '$a = 2+1$' or something like '$a \triangleq 2+1$', but this does not make a a variable.

An additional important thing to observe in this context is that variables are terms, i.e., variables in logic are always 'syntactic'. The concept and use of variables e.g. in analysis is different. Clearly, we cannot mathematically be in the domain of analysis, and speak of "truth values", without complying with language constructions of logic. 'Truth' and logic belong together within the realm of "mathematics for logic", but 'truth' and analysis is a different story, and in fact part of a more philosophically oriented "(fons et origo) logic for mathematics".

9. Fuzzy variables and one of the fundamental differences between ‘fuzzy’ and ‘probable’

In the previous section, Set is the underlying category, so variables and variable sets are always crisp. Even if it may be intuitively difficult to comprehend or from application point of view hard to motivate that variables and variable sets are fuzzy, it can in fact be formalized, as was shown in [3]. This essentially means that the ‘object of variables’ is a categorical object in the Goguen category $\mathtt{Set}(\mathfrak{Q})$, where $\mathfrak{Q}$ is a suitable lattice or quantale. Such an object then has the form (X, α), where $\alpha : X \to Q$ annotates an uncertainty $\alpha(x)$ with x.

The natural numbers signature can easily be arranged over $\mathtt{Set}(\mathfrak{Q})$, so that $\mathtt{NAT}^{\mathfrak{Q}} = (S, (\Omega, \varsigma))$, where S and Ω are the same as for the NAT signature. In this case ς annotates uncertainties with the operators in Ω. A typical example is the questionable accuracy of a one meter stick, when using it to measure a distance of 100 meters. We may have

$$\varsigma(\mathsf{0}) = 1$$

and

$$\varsigma(\mathtt{succ}) = 0.999$$

which then intuitively means that uncertainties are ‘accumulated’ whenever succ is used inside a calculation, or a term involving succ is used in a sentence for reasoning.

This simple example shows how the underlying Goguen category for representing ‘fuzzy’ can appear already in signatures, and that we in fact can have ‘fuzzy variables’, whatever that intuitively will mean in applications. It is still unknown if the same is valid e.g. for using the category Meas of measurable spaces, to represent ‘probability’ in case we restrict to probability spaces, and to think in terms of ‘probable variables’. The reason is that the formal arrangement of ‘signatures over a category’ requires *monoidal closedness* of that category, and the Goguen category fulfills those conditions for certain lattices and quantales, including the unit interval. If the monoidal operation is the cartesian one, the situation is usually clear, but also other multiplications can be considered. However, Meas may or may not be monoidal closed, and therefore the theoretical pillar, needed for arranging operators and variables over probabilities, is unclear. This is also one of the fundamental differences between ‘fuzzy’ and ‘probable’, namely that ‘fuzzy’ can be logical, since it potentially allows for using underlying signatures.

10. Levels of signatures

Type constructors such as the one producing function types are frequently treated as if they are outside signatures, i.e., that they are not operators in some signature. In order to see it more precisely, let $\mathtt{s}_1$ and $\mathtt{s}_2$ be two sorts in S. The “function sort” involving $\mathtt{s}_1$ and $\mathtt{s}_2$ can be denoted $\mathtt{s}_1 \Rrightarrow \mathtt{s}_2$. Even if we want to view $\mathtt{s}_1 \Rrightarrow \mathtt{s}_2$ as a (constructed) sort, it is usually not seen as part of S.

This creates an awkward meta-level of constructors, and the formalism for treating these constructors is rather lose. This is also the reason why the definition of λ-terms is correspondingly lose, even if it is seen as compact and elegant. However, the traditional so called 'set' of λ-terms is not well-defined even if it is 'well understood'. In fact, without levels of signatures, we cannot produce a strict term functor, or term monad for that matter, for producing λ-terms.

This situation becomes more clear, and lambda term functors can be provided [3], once type constructors are accommodate properly into suitable levels of signatures. The main question is how to expand the signature $\Sigma = (S, \Omega)$, often called the *basic signature*, to a signature $\Sigma' = (S', \Omega')$ so that $\mathtt{s}_1 \Rrightarrow \mathtt{s}_2 \in S'$ whenever $\mathtt{s}_1, \mathtt{s}_2 \in S$, and where Σ' is the underlying signature for the λ-terms. Such an arrangement also enables to keep λ-abstractions, as members of Ω', clearly apart from λ-terms, residing in the set of λ-terms as defined by the λ-term functor.

The three-level arrangement of signatures starts from the basic signature Σ on 'level one', with Σ' being on 'level three'. On 'level two' we have the Σ-*superseding type signature* as a one-sorted signature $\mathtt{S}_\Sigma = (\{\mathtt{type}\}, \mathtt{Q})$, where $\mathtt{Q}$ is a set of *type constructors* satisfying

(i) $\mathtt{s} :\to \mathtt{type}$ is in $\mathtt{Q}$ for all $\mathtt{s} \in S$

(ii) there is a $\Rrightarrow: \mathtt{type} \times \mathtt{type} \to \mathtt{type}$ in $\mathtt{Q}$

If $\mathtt{Q}$ does not contain any other type constructors, apart from those given by (i) and (ii), we say that $\mathtt{S}_\Sigma$ is a Σ-*superseding simple type signature.*

However, under the superceding signature we may consider other type constructor, e.g., for capturing the notion of a 'power type constructor' $\mathtt{pow} : \mathtt{type} \to \mathtt{type}$, or a 'product type constructor' $\boxtimes : \mathtt{type} \times \mathtt{type} \to \mathtt{type}$ that would be more general than the traditional cartesian product.

Example 10.1. In [1], product types are potentially introduced in

$$large = wide \times long$$

where *wide* and *long* could be seen as constants of sort $\mathtt{type}$ on level two, i.e., $\mathtt{wide}, \mathtt{long} :\to \mathtt{type}$, and *large* would be the constructed type $\mathtt{large} = \boxtimes(\mathtt{wide}, \mathtt{long})$, or written as

$$\mathtt{large} = \mathtt{wide} \boxtimes \mathtt{long}$$

using infix notation. This may then be seen as opening up several possibilities for the algebra $\mathfrak{A}_{\mathtt{S}_\Sigma}(\boxtimes)$ of that product type. In [1], the intuition leans on using 'projections', and therefore the algebra chosen is the traditional cartesian product.

Term monads in the many-sorted case are formally constructed in [3]. This construction can then obviously be used also for any Σ-superseding type signature $\mathtt{S}_\Sigma$, and the monad is denoted $\mathbf{T}_{\mathtt{S}_\Sigma}$. Note also that we do not necessarily have to restrict to having monads over $\mathtt{Set}$, but monads can be provided over any monoidal biclosed category, whose objects then intuitively represent the 'object of variables', and on the superceding level indeed intuitively as type variables. Then $\mathsf{T}_{\mathtt{S}_\Sigma} X$, where X is the tuple of objects representing (type) variables, will contain all terms which we call *type terms*. We may write $\mathtt{s} \Rrightarrow \mathtt{t}$ for the type term $\Rrightarrow (\mathtt{s}, \mathtt{t})$.

The signature $\Sigma' = (S', \Omega')$ on level three then is based on $S' = \mathsf{T}_{\mathtt{S}_\Sigma} \varnothing$, i.e., the sorts on level three are those from level one together with the constructed sorts, which on level two appear as terms (the type terms), and then on level three added to those basic sorts coming from level one.

For the operators in Ω' it may sound natural to include all operators from Ω into Ω' so that $\Omega \subseteq \Omega'$, but it is not always desirable. If we consider the $\mathtt{NAT}$ signature on level one we obviously may have both $\mathtt{0} :\to \mathtt{nat}$ and $\mathtt{succ} : \mathtt{nat} \to \mathtt{nat}$ included in the operators for $\mathtt{NAT}'$. However, the unary operator $\mathtt{succ}$, i.e., unary both on level one and level three, can alternatively be (λ-)abstracted to become a constant (0-ary) operator $\lambda_1^{\mathtt{succ}} :\to (\mathtt{nat} \Rrightarrow \mathtt{nat})$ on level three. Clearly, the constant $\mathtt{0} :\to \mathtt{nat}$ converts to $\lambda_0^{\mathtt{0}} :\to \mathtt{nat}$, i.e., a constant on level one remains as a constant on level three. Note also that $\mathtt{nat}$ on level one is not the same as $\mathtt{nat}$ on level three. If we need to be strict, we should use e.g. $\mathtt{nat}'$ for the corresponding sort on level three.

Note, in the mathematical notation for signatures according to [8], that the monoidal arrangement does not embrace any structure related to the Σ-superceding level. That is, in [8] there is no clearly expressed anticipation of applications in type theory.

Church's type constructor [9] is in effect $\Rrightarrow$, so that $(\beta \Rrightarrow \alpha)$ is Church's $(\beta\alpha)$. An interpretation of Church's ι to be $\mathtt{type}$ is clearly less controversial, but for the interpretation of o there are a number of alternative intuitions.

In summary, the three signature levels underlying the production of λ-terms are then following.

1. the level of primitive underlying operations, with a usual many-sorted signature $\Sigma = (S, \Omega)$
2. the level of type constructors, with a single-sorted signature $\mathtt{S}_\Sigma = (\{\mathtt{type}\}, \{\mathtt{s} :\to \mathtt{type} \mid \mathtt{s} \in S\} \cup \{\Rrightarrow : \mathtt{type} \times \mathtt{type} \to \mathtt{type}\})$
3. the level including λ-terms based on the signature $\Sigma' = (S', \Omega')$ where $S' = \mathsf{T}_{\mathtt{S}_\Sigma} \varnothing$, $\Omega' = \{\lambda^{\omega}_{i_1,\ldots,i_n} :\to (\mathtt{s}_{i_1} \Rrightarrow \cdots \Rrightarrow (\mathtt{s}_{i_{n-1}} \Rrightarrow (\mathtt{s}_{i_n} \Rrightarrow \mathtt{s})) \mid \omega : \mathtt{s}_1 \times \ldots \times \mathtt{s}_n \to \mathtt{s} \in \Omega\} \cup \{\mathtt{app}_{\mathtt{s},\mathtt{t}} : (\mathtt{s} \Rrightarrow \mathtt{t}) \times \mathtt{s} \to \mathtt{t}\}$

Here $(i_1, \ldots, i_n)$ is a permutation of $(1, \ldots, n)$. Note also that level one operators are always transformed to constants on level three. In traditional notation in λ-calculus, substituting x by $\mathtt{succ}(y)$ in $\lambda y.\mathtt{succ}(x)$ requires a renaming of the bound variable y, e.g., $\lambda z.\mathtt{succ}(\mathtt{succ}(y))$. In our approach we avoid the need for renaming. On level one, and in the case of $\mathtt{NAT}$, we have the substitution (Kleisli morphism) $\sigma_{\mathtt{nat}} : X_{\mathtt{nat}} \longrightarrow \mathsf{T}_{\mathtt{NAT},\mathtt{nat}}(X_{\mathtt{t}})_{\mathtt{t} \in \{\mathtt{nat}\}}$, where $\sigma_{\mathtt{nat}}(x) = \mathtt{succ}(y)$, x being a variable on level one, and the extension of $\sigma_{\mathtt{nat}}$ is $\mu_{X_{\mathtt{nat}}} \circ \mathsf{T}_{\mathtt{NAT},\mathtt{nat}}\, (\sigma_{\mathtt{t}})_{\mathtt{t} \in \{\mathtt{nat}\}} : \mathsf{T}_{\mathtt{NAT},\mathtt{nat}}(X_{\mathtt{t}})_{\mathtt{t} \in \{\mathtt{nat}\}} \longrightarrow \mathsf{T}_{\mathtt{NAT},\mathtt{nat}}(X_{\mathtt{t}})_{\mathtt{t} \in \{\mathtt{nat}\}}$.

On level three we have $\sigma_{\mathtt{nat}'} : X_{\mathtt{nat}'} \longrightarrow \mathsf{T}_{\mathtt{NAT}',\mathtt{nat}'}(X_{\mathtt{t}})_{\mathtt{t}\in S'}$, with $\sigma_{\mathtt{nat}'}(x) = \mathtt{app}_{\mathtt{nat}',\mathtt{nat}'}(\lambda_1^{\mathtt{succ}}, x)$, x being a variable on level three, and requiring no renaming in $\mu_{\mathtt{nat}'} \circ \mathsf{T}_{\mathtt{NAT}',\mathtt{nat}'}\, \sigma_{\mathtt{nat}'}(\mathtt{app}_{\mathtt{nat}',\mathtt{nat}'}(\lambda_1^{\mathtt{succ}}, x))$.

At this point we have the crisp set of λ-terms, given the term functor $\mathsf{T}_{\Sigma'} : \mathtt{Set}_{S'} \longrightarrow \mathtt{Set}_{S'}$. The sets of λ-terms with respect to each end sort $\mathtt{s}' \in S'$ are then represented by respective sets $\mathsf{T}_{\Sigma',\mathtt{s}'}(X_{\mathtt{s}})_{\mathtt{s}\in S'}$.

Note indeed that the λ-term monad may be considered to be over other monoidal biclosed categories [3].

11. Probability

In probability we are dealing with 'event spaces' being σ-algebras[4], where particular events are subsets of the sample space. The 'universe' for that power set is just a set. It has no structure. It is a set of points, and in probability these points cannot be 'dissected' in any way to give them some more meaning or structure. Tossing a coin provides either heads or tails so the set $\{head, tail\}$ of events is just a 'set of points'. We must be careful if we logically really want to say that 'head' is a statement or sentence. The only thing we apparently have is that *head* is an element in the set $\{head, tail\}$.

The underlying 'universe' E of samples is an 'unstructured set of unstructured elements', and σ-algebras $\mathfrak{F}$, as subsets of the power set $\mathsf{P}E$, are used with the probability measure[5] *Prob*, which is a mapping $Prob : \mathfrak{F} \longrightarrow [0, 1]$ satisfying *Kolmogorov's axioms*. For an event $A \in \mathfrak{F}$, $Prob(A)$ is then the "probabilistic valuation" of the event A.

Remark. Some more simple versions of 'probabilistic logic' binds conditionality with implication so that '$Prob(A \mid B)$' would have a logical reading like "the implication $A \leftarrow B$ has an uncertainty degree of $Prob(A \mid B)$". Here A and B are then variables complying with some underlying signature $\mathtt{ProbLog}$ for a suitably tailored propositional style 'probability logic'. Obviously, $\mathfrak{F}$ is a candidate for the semantic domain for these variables, so that $\mathfrak{A}(\mathtt{bool}) = \mathfrak{F}$, and an assignment provides $\mathfrak{v}(A), \mathfrak{v}(B) \in \mathfrak{F}$. However, in probability theory, $\mathfrak{F}$ is not treated as a semantic domain in this way.

The use of 'random variables' is sometimes confusing for logicians, since a random variable is not a variable in the logical sense. In the 'continuous' case, a *real-valued random variable*, often denoted X, is a mapping $X : E \longrightarrow \mathbb{R}$ such that $\{e \in E \mid X(e) \leq r\} \in \mathfrak{F}$ for all $r \in \mathbb{R}$. Note that '$X \leq x$' is just a shorthand notation for the event $\{e \in E \mid X(e) \leq x\}$. In the 'discrete' case, '$X = x$' would be a notation for the event $\{e \in E \mid X(e) = x\}$.

[4] σ-algebras should not to be confused with Σ-algebras, where Σ is a signature. Note that the 'σ' itself in 'σ-algebra' does not have any specific meaning or content.

[5] In this note we prefer to use *Prob* instead of P as a notation for the probability measure, since P will be used for the power set functor, and for the power type constructor.

Conjecture 1. *Events may be seen as "unstructured sets of sentences", where samples can be understood as being constant operators in some underlying signature, and probability theory in general embraces most building blocks of logic, since even entailment can be identified as related to conditional probabilities.*

We can therefore say that the foundations for probability theory may be logically and syntactically seen as based on signatures, terms, sentences, theoremata and entailment. However, when including concepts like distribution functions and expected values, sample sets must be used in their semantic form, and probability theory indeed is never clear about when and how a sample set is used syntactically or semantically.

11.1. Samples and events

Can we now 'detect' or 'invent' some pieces of formal logic in these basic and fundamental constructions, or is probability 'theory' really, intentionally or not, something "very non-logical"[6]? If we try to find some formal logic, we have to start looking at the sample space E. We may choose dice as an example to produce E, and starting with one dice, it seems correct to say that the sample space is

$$E = \{⚀, ⚁, ⚂, ⚃, ⚄, ⚅\}$$

where it is very clear that $⚀, \ldots, ⚅$ are syntactic rather than being semantic values. They are apparently all of the same sort, and let us denote that sort by `dice`. The most obvious and straightforward signature for dice is then $\mathtt{DICE} = (S, \Omega)$, where

$$S = \{\mathtt{dice}\}$$

and

$$\Omega = \{⚀ :\to \mathtt{dice}, ⚁ :\to \mathtt{dice}, ⚂ :\to \mathtt{dice}, ⚃ :\to \mathtt{dice}, ⚄ :\to \mathtt{dice}, ⚅ :\to \mathtt{dice}\}$$

and these constants would typically be expected to have a semantic description according to

$$\mathfrak{A}_{\mathtt{DICE}}(\mathtt{dice}) = \{1, 2, 3, 4, 5, 6\} (\subset \mathbb{N})$$

and

$$\mathfrak{A}_{\mathtt{DICE}}(⚀) = 1, \ldots, \mathfrak{A}_{\mathtt{DICE}}(⚅) = 6.$$

The set of terms over the one-sorted signature `DICE` using an empty set of variables is

$$\mathsf{T}_{\mathtt{DICE}}\varnothing = \{⚀, ⚁, ⚂, ⚃, ⚄, ⚅\}$$

[6]The 'categorization' of probability theory goes back to Lawvere's handouts from 1962. Lawvere was advisor to Giry, who developed the ideas further to become what we today call the "Giry monad". However, in these and related approaches, the sample space E remains as a black box. These approaches didn't involve any term monads, so the possibility e.g. of investigating term monads over the category of measurable spaces was never anticipated. In [10], these involvements were briefly announced as feasible, but detail was not developed.

and here it is very important to understand the distinction between '⚀' as the shorthand notation for the constant operator $\text{⚀} :\to \texttt{dice}$ in Ω, and '⚀' as a term, since "constants are terms". For the latter we may have a notation $\text{⚀} :: \texttt{dice}$ saying that "⚀ is a term" of sort dice. This seems like an unnecessary subtlety, but the distinction is really important to make so that we avoid "mixing bags". A strictly formal and categorical notation, where these distinctions are more clear, uses monoidal closed categories [3], but in these notes it is sufficient to understand that in this example, Ω and $\mathsf{T}_{\texttt{DICE}}\varnothing$, are two different sets, the latter then being

$$\mathsf{T}_{\texttt{DICE}}\varnothing = \{\text{⚀} :: \texttt{dice}, \ldots, \text{⚅} :: \texttt{dice}\}$$

and the sample set must accordingly be seen to be

$$E = \mathsf{T}_{\texttt{DICE}}\varnothing$$

so that "samples are terms".

So far the underlying assumption is that the term functor $\mathsf{T}_{\texttt{DICE}}$ is an endofunctor over the category of sets. What happens when we consider it over the Goguen category, or if we compose the fuzzy power set functor with the term functor? This leads to the distinction between 'fuzzy operation' and 'operation with fuzzy'. In the specific case of DICE the distinction is less clear since DICE consists only of constants. However, the following will show how we invoke uncertainty in different ways.

If, on the one hand, DICE is extended to $\texttt{DICE}^{\Omega}$ in a similar fashion as we did for NAT extended to $\texttt{NAT}^{\Omega}$, we will need to specify

$$\varsigma : \{\text{⚀} :\to \texttt{dice}, \ldots, \text{⚅} :\to \texttt{dice}\} \longrightarrow Q$$

and if the dice is 'traditional', we will use $Q = [0, 1]$, and set

$$\varsigma(\text{⚀} :\to \texttt{dice}) = \frac{1}{6}, \ldots, \varsigma(\text{⚅} :\to \texttt{dice}) = \frac{1}{6}.$$

In this case, ς is the 'fuzzy set', and we have 'fuzzy operation'.

If, on the other hand, we prefer to use the composed monad $\mathbf{L} \bullet \mathbf{T}_{\texttt{DICE}}$ [11, 12] over Set, we will have 'generalized terms' $\alpha \in (\mathsf{Q} \circ \mathsf{T}_{\texttt{DICE}})X_{\texttt{dice}}$, i.e., mappings $\alpha : \mathsf{T}_{\texttt{DICE}}X_{\texttt{dice}} \longrightarrow Q$, and similarly we may set

$$\alpha(\text{⚀} :: \texttt{dice}) = \frac{1}{6}, \ldots, \alpha(\text{⚅} :: \texttt{dice}) = \frac{1}{6}.$$

In this case α is something we operate on "from the outside", thus having 'operation on fuzzy'.

The latter case is more traditional within the fuzzy community, and concepts like 'fuzzy set of samples' or 'fuzzy events' would comply more with the composed monad approach, whereas a concept like 'fuzzy sample' requires the use of signatures over the Goguen category. Clearly, consideration of 'uncertain sample' is very rare in the literature, maybe even non-existing.

Once we have operators beyond constants, it is clear that the "accumulation of uncertainty" arises entirely different in these two cases.

11.2. Random variables

Application oriented books in statistics typically avoid being precise about random variables, and even mix the concept of a random variable with that of a probability distribution. It is very common to write e.g. $\overline{X}$ for the 'mean of observations' without discussing the intuition that the mean as an 'average' is kind of a most appropriate value of some imaginary representative sample being in the 'middle' or 'centre' of a population or a set of samples. The utility of medians has been mostly overlooked in practical examples, maybe due to mean values leading to the notion of 'comparing mean values', and from there onwards to hypothesis testing.

Let us look more closely at the random 'variable' X as a function $X : E \longrightarrow \mathbb{R}$. As said before, X is not a variable in the logical sense. In the case of `DICE`, X will be a 'discrete random variable', and we have X, as a mapping, in the form of

$$X : \{⚀ :: \texttt{dice}, \ldots, ⚅ :: \texttt{dice}\} \longrightarrow \{1, \ldots, 6\} (\subset \mathbb{R}).$$

Now clearly, an expression like

$$X = 5$$

represents the event

$$\{⚄\}$$

and

$$X = \{2, 3\}$$

the event

$$\{⚁, ⚂\}.$$

In these expressions we have omitted ':: `dice`', assuming the sort is known, but being clear about '⚁', '⚂' and '⚄' indeed being terms and not constants. Needless to say, we cannot say that in these examples we have 'substitution' or 'assignment' involving variables, and in fact, we do not have any variables in these expressions. However, X in fact assigns syntactic terms to values in a semantic domain,

$$X : \mathsf{T}_{\texttt{DICE}}\varnothing \longrightarrow \mathfrak{A}_{\texttt{DICE}}(\texttt{dice}),$$

and here it is very clear that the set of variables, in a logical sense, is empty.

Observation 11.1. *The is no* a priori *partial order on the set* $\{⚀, ⚁, ⚂, ⚃, ⚄, ⚅\}$*, but since the partial order in* $\{1, 2, 3, 4, 5, 6\}$ *is acknowledged and used, the fact that* X *is formally a measurable function, means that a partial order on* $\mathsf{T}_{\texttt{DICE}}\varnothing \longrightarrow \mathfrak{A}_{\texttt{DICE}}(\texttt{dice})$ *can be induced by* X*. This is the reason why it is quite frequent not clearly to distinguish between elements in the domain and range of a discrete random variable. Imposing a partial order from the semantic domain 'into' the syntactic side also shows how probability uses measure theory and analysis in general to "overwrite" whatever might happen logically with events and mean, and efforts to distinguish between population and individual, the latter being dependent on having variables and substitutions.*

Observation 11.2. *The nature of random variables indeed reveal that a "bridging" of traditional probability with logical truth is virtually impossible. Probability measures appear in analysis, whereas logic embraces algebraic tools. The mathematical foundations for probability and logic are entirely different. The absence of logical variables in probability theory means that probability theory is essentially 'unsorted', e.g., so that boolean sorts, or sorts representing the typing of truth values, do not exist. If we try very hard, we could say that probability theory is one-sorted in that the sample set E could be seen as a set of terms based on constants, the samples, e.g., of type* `sample`*. However, probability theory moves very quickly to the realm of analysis since the range of a random variable, as a mapping, must be measure-theoretically dressed up. Distribution and density are concepts that are formally defined in the realm of measures and integrals, where the bottom line is supported e.g. by the Radon-Nikodym theorem. Properties of measurability are in the end related to abstractions of what may and may not happen topologically on the real line. Therefore analysis is inherently the mathematical domain of probability theory.*

As a prerequisite for understanding the basic idea of possibility measures, note how the set

$$\{2, 3\}$$

frequently in early fuzzy set theory notation is written as the function

$$\mu_{\{2,3\}} : \{1, 2, 3, 4, 5, 6\} \longrightarrow \{false, true\}$$

where $\mu_{\{2,3\}}(2) = true$ and $\mu_{\{2,3\}}(3) = true$, and $\mu_{\{2,3\}}(a) = false$ for $a = 1, 4, 5, 6$. Since $\{2, 3\}$ and $\mu_{\{2,3\}}$ are "isomorphic" ways of describing crisp subsets, we may write

$$X = \mu_{\{2,3\}}$$

instead of $X = \{2, 3\}$. Similarly, we can define a $\mu_{\{⚁,⚂\}}$ being "the same" as $\{⚁, ⚂\}$, so that '$X = \mu_{\{2,3\}}$' would be abstract notation for the set $\mu_{\{⚁,⚂\}}$. In [1], for single points, this could have been written as

$$X = 2 : \mu_{\{⚁\}}(⚁).$$

Howeer, in [1], there is not a clear distinction between '⚁' and '2', and in fact, they are 'identified'.

11.3. Probability measure

Before looking logically more closely at the function $Prob : \mathfrak{F} \longrightarrow [0, 1]$, we realize that $\mathfrak{F}$ is somehow to be explained using portions of the `DICE` signature, whereas $[0, 1]$ is a semantic domain for syntactic truth values appearing in some `LOGIC` signature, where a sort like `bool` would appear, and we would have $\mathfrak{A}_{\texttt{LOGIC}}(\texttt{bool}) = [0, 1]$. We may then imagine to have some kind of a 'merger signature' of `DICE` and `LOGIC` into a `DICELOGIC` signature, that becomes the underlying signature for providing terms and sentences for eventually developing a reasoning system for a game involving dice. This, however, requires involvement of the level of signatures since we need access to the power type $\mathtt{pow}(\mathtt{dice})$, so that $\mathtt{DICELOGIC}'$ actually resides on level three, and we would desire to define something like

$$\mathfrak{F} = \mathsf{P}(\{⚀, \ldots, ⚅\})$$

i.e., the σ-algebra $\mathfrak{F}$ is the whole power set of the sample set. This in turn requires the observation that we have

$$\mathtt{dice} :\rightarrow \mathtt{type}$$

as a constant in the superceding level signature $\mathsf{S}_{\mathtt{DICELOGIC}}$. If we speak about 'probabilities of events', then the natural choice for a semantic domain of on level two is

$$\mathfrak{A}_{\mathsf{S}_{\mathtt{DICELOGIC}}}(\mathtt{type}) = \mathrm{Ob}(\mathtt{Set}).$$

However, we then have an unexpected "mixed bag" of syntax and semantics since

$$\mathfrak{F} = \mathsf{P}(\{⚀, \ldots, ⚅\}) \neq \mathsf{P}(\{1, \ldots, 6\}) = \mathfrak{A}_{\mathsf{S}_{\mathtt{DICELOGIC}}}(\mathsf{pow})\ \mathfrak{A}_{\mathsf{S}_{\mathtt{DICELOGIC}}}(\mathtt{dice}).$$

Thus the only remaining possible view of $\mathfrak{F}$ is

$$\mathfrak{F} = \mathsf{P}(\mathsf{T}_{\mathtt{DICE}}\varnothing)$$

which brings events in $\mathfrak{F}$ to be closer to "unstructured sets of sentences" rather than more elaborate "structured conglomerates of sentences" with the sentence functor Sen being the identity functor and the theoremata functor Δ being P. See [13] for detail about sentence and theoremata functors.

The probability measure *Prob* is more of a valuation

$$Prob : \mathfrak{F} \longrightarrow [0, 1]$$

of theoremata, and conditional probability $Prob(\_ \mid \_)$, defined as

$$Prob(A \mid B) = \frac{Prob(A \cap B)}{Prob(B)},$$

can be viewed as an entailment

$$\vdash_{Prob} = Prob(\_ \mid \_) : \mathfrak{F} \times \mathfrak{F} \longrightarrow [0, 1]$$

in some fuzzy logic. This can be seen as a "proof" of Conjecture 1.

Remark. Probability theory pays no attention to the situation that '$|$' is actually not a mapping $| : \mathfrak{F} \times \mathfrak{F} \longrightarrow \mathfrak{F}$. This then would strictly mean that we need two separate notations, respectively, for the probability measure and for the conditional probability measure. Syntactically, we have also seen that '$|$' cannot act as a binary "operation" on sentences. It could do that on terms, but that does not make any sense. Therefore the only interpretation we have left for '$|$' is that it indeed corresponds to a many-valued entailment relation.

Open Problem 11.1. *Which properties of entailment does $\vdash$ really fulfill?*

Note also that an extension of this discussion to involve a modelling of 'probabilities of fuzzy events', or similar, the semantic domain of `type` may need to be something closer to the objects of the Goguen category.

11.4. Distribution functions and mean values

The domain of a distribution function must be the range, not the domain, of the related random variable. Since we define distribution functions as

$$f_X(x) = Prob(X \leq x),$$

so that we have

$$f_X : \mathbb{R} \longrightarrow [0,1],$$

and not

$$f_X : E \longrightarrow [0,1],$$

probability theory in fact hides the sample space E inside X which remains just as an index in f_X.

Observation 11.3. *We cannot view x or X to be 'variables' in the sense of logic.*

This is the fundamental turning point of probability theory from potentially being logical to inherently being part of measure theory and analysis. In subsequent developments in probability theory based on distribution functions, the underlying signature of the sample set as a 'set of terms' will never play any role. Moving over to dealing with values only in the range of random variables, and consequently not at all in the domain, makes probability theory fundamentally depart from potential use of logical concepts, since the underlying signature "in the range of random variables" is basically the empty signature.

It is indeed in this interrelation between, on the one hand, the probability measure and, on the other hand, the probability distribution and density, where the concept of 'variable' becomes orphaned. There is a potential use of 'X' and 'x' as variables in the logical sense, but this "shift from E to $\mathbb{R}$", makes probability theory blind for logical notions, and brings the notion of 'variable' simply to be the set theoretic notion of a 'variable or parameter inside a function'. Probability theory as a mathematical discipline can be made formally and even logically more precise, but application development based on probability theory does not always live up to the spirit of those disciplinary notations and structures.

Observation 11.4. *Given the view that Prob could logically be seen as a valuation of "structured conglomerates of sentences" and conditional probability as an entailment, the notion of distribution function closes that window of opportunity that would have extended the understanding of 'chance' as a simple percentage to also embracing 'opportunity' and 'possibility' which are quantified by means of truth values. In traditional probability theory, the unit interval [0,1] is just the 'percentage interval', which disables it to see and manage truth value.*

Note how the *mean*, or the expected value, usually written $E[X]$, for the random variable X connected with dice is

$$E[X] = 1 \cdot \frac{1}{6} + 2 \cdot \frac{1}{6} + 3 \cdot \frac{1}{6} + 4 \cdot \frac{1}{6} + 5 \cdot \frac{1}{6} + 6 \cdot \frac{1}{6}$$

and how it in fact is calculated using values in the range of the random variable. Note also that in this case we do not have any natural *median* since there is no sample being in the 'middle' or 'centre'.

Further, it is important to note that the fundamental difference between 'mean' and 'median' is that 'mean' is value in the range of the random variable, whereas 'median' is to be found in the domain. 'Median' is therefore more syntactic, whereas 'mean' is more semantic. Is there any "consistency" between mean and median? Do we prefer to say that a mean value should be 'sound' or 'correct' with respect to the "median representative", or is a median to be 'sound' or 'correct' with respect to the mean? Which one is master, which one is slave? Similarly, we may look at 'completeness', i.e., given a mean, can we always point at a related median, and given a median, can we predict the mean?

Open Problem 11.2. *Should we allow mean to be semantic, or can we develop a syntactic notion also for the mean, so that some consistency properties for mean and median could be formulated?*

12. Possibility theory

The essential difference between 'probability theory' and 'possibility theory' is that the probability measure

$$Prob : \mathfrak{F} \longrightarrow [0, 1]$$

is defined on a σ-algebra of events being crisp subsets of samples in a sample set E, whereas the possibility measure [1]

$$Poss : \mathfrak{G} \longrightarrow [0, 1]$$

essentially is defined on events that are intuitively fuzzy subsets of samples, or similar, and without necessarily having any "algebraically constraining jacket" requiring fulfillment of conditions like those appearing for σ-algebras involving crisp sets. Similarly as

$$\mathfrak{F} = \mathsf{P}E$$

is a σ-algebra, the traditional selection for $\mathfrak{G}$ is

$$\mathfrak{G} = \mathsf{L}E,$$

where L is the many-valued (fuzzy) power set functor.

However, as possibility theory in [1] rather quickly moves to comparing possibility distributions with probability distribution, there is a by-passing of the discussion about an underlying comparison of '(random) fuzzy variables' as appearing in possibility theory with traditional random variables as appearing in probability theory.

12.1. Fuzzy variables

The 'universe of discourse' U in [1] is not explicitly said essentially to be a 'set of samples' E as used in probability theory. The question is therefore whether U is to be seen more closely related to the domain or range of a traditional random variable.

Whatever this 'universe of discourse' is called in natural language, does not change the fact that the set is just a set of points that are indivisible. If subjected to be described more like a 'sample set' using some underlying signature, points would be constants of the same sort. Similar to the situation in probability theory, also in traditional possibility theory there is no consideration of many-sortedness.

Fuzzy variables as the fuzzy counterpart to random variables is in [1] evidently adopting the principle to neglect the potential added value of analyzing the structure of the sample set. Even if it is not explicitly mention, we should see U as closer to being the range of a random variable than the domain. In fact, in [1], the assignment of a value to an element in the universe of discourse is written as

$$X = u : \mu_F(u)$$

where probability theory in the corresponding expression separates between 'u' in the domain of the random variable from its image in the range.

In [1] it is clearly said that "F is a fuzzy subset of U", so F and μ_F are well-defined. The range of μ_F is not explicitly defined, and it is only said that F is the "fuzzy restriction" on X, semi-formally written

$$R(X) = F,$$

but it is not explained what R formally would be. This is clear, since X is not formally explained. It is more like the informal notion of X is bound to the formal definition of F. Once this has been done, X is used in indexing, but is as an index in its own right somewhat doubtful, since X in fact is "no more" than F.

Observation 12.1. *Possibility theory suffers from the same style of hiding the sample space U inside X which remains just as an index in π_X. Even worse, possibility theory speaks just intuitively about the 'universe of discourse' which only plays the role of the domain for μ_F. Whereas probability through the use of distribution functions has 'closed the window of opportunity to be logical', this window for possibility theory was always closed, since possibility theory has adopted the same view of 'sample sets' and 'distribution', but being less rigorous about the sample set. This indeed makes possibility theory 'simpler' than probability theory, but also less powerful for applications. Having a fuzzy set on that undefined and unstructured universe of discourse may be seen as potentially useful for modelling natural language [14], but unless something structural is provided also for U, it is really hard to be convinced that the richness of natural language structure and phenomenology can be discussed having U as an unstructured set.*

This problem of hiding the underlying structure of the domain of X is very clearly seen e.g. in the statement "X is a small number", where it is evident how to understand "small number" as some F. In this particular case, the domain of X is intuitively the set of numbers, nothing more, and we can then understand "everything inside". However, if we take the statement "John is young" in [1], which is comparable with a statement like "John is capable of remaining in home care", as related to a modelling of assessment scales in [5], we immediately see that "John" in respective statements embraces entirely different information scopes and structures.

In [1], the intuitive notion of "implied attribute" scratches this surface in order to introduce a further discussion on possible 'signatures' and 'terms' content of U, but [1] does not take that discussion any further. The notation is $A(X)$ where A is supposed to be understood as unravelling that structure, but it remains just as a notation. It justifies informal expressions like

$$Age(John)$$

with '*Age*' for A, but if we want to see more structure in an expression like

$$CapableSelfcare(John)$$

then clearly $CapablilitySelfcare$ is a much richer structure than Age, or in other words, $CapablilitySelfcare$ hides much more signature structure than Age.

This may be the reason why [1] prefers, "for simplicity", to assume that

$$A(X) = X.$$

12.2. Possibility distribution functions and possibility measures

In probability theory, distribution functions are defined by the probability measure, but in possibility theory the possibility distribution function comes first, and then the possibility measure is defined based on the possibility distribution function.

Possibility distributions are in [1] "postulated to be equal to" fuzzy subsets, and the formal definition is

$$\pi_X \triangleq \mu_F.$$

The "small number" example clearly shows how possibility differs from probability, and in fact, how the unit interval as the range of the possibility distribution function appears more as a range of truth values, a "range of possibilities", than a "range of change".

Observation 12.2. *Probability and possibility distribution functions have the same domain and range, but the assignment of probabilities through assignment by the probability measure, which is based on observation of frequency of samples, is different than the assignment of possibility through assignment of fuzzy membership values. This is illuminated by the "Hans eating eggs" example. It is* *even easy to create examples where the probability of a statement is '0', whereas, for the same statement, the possibility is '1'. In [1], the* possibility/probability consistency principle *provides a quantification for "consistency", but it is also said that "it is not a precise law that is intrinsic in the concepts of possibility and probability".*

Discussions in [1] about expressions like "slightly possible" clearly show that `slightly` must be seen as a logical operator, but it is unclear if such an operator is on signature levels one or three as

$$\texttt{slightly} : \texttt{bool} \to \texttt{bool}$$

or if it in fact is a type constructor on level two, as

$$\texttt{slightly} : \texttt{type} \to \texttt{type}$$

so that

$$\texttt{slightly}(\texttt{bool})$$

can be used as a sort in its own right. Similarly, if

$$\texttt{age} :\to \texttt{type}$$

is constant on level two so that

$$\texttt{age}$$

is a sort on level three, then also

$$\texttt{slightly}(\texttt{age})$$

is available as a sort on level three. Within the fuzzy community, such type theoretic arrangements have never been consider. In [3], the "fruitbasket example", involving sorts like

$$\texttt{FruitBasket}, \texttt{Fruit}, \texttt{Apple}, \texttt{IngridMarie}$$

which can be seen as prerequisite for developing a knowledge base for cooking, enables expressions like "*if you have Ingrid Marie apples, give me a fruit basket with more apples than pears, others more pears than apples*" or "*give me a fruit basket with some pears, but mostly apples, and at least a few Ingrid Marie apples*".

This also clearly shows how, not just

$$\texttt{mean}, \texttt{median}$$

but also e.g.

$$\texttt{low}, \texttt{high}, \texttt{tall}, \texttt{rich}$$

as operators can be developed as useful tools in possibility theory and possibility logic, and then the three level arrangement of signatures is really required to enable a more transparent formalization of type constructors.

Open Problem 12.1. *Try to develop examples like "Swedes are tall", "Robert is rich" or "snow is white" in this three level signatures context. Comparing humans and citizens with fruit and types of fruit, is "Swede" to be sortwise compared e.g. with* `FruitBasket`*,* `Fruit` *or* `Apple`*?*

Once the possibility distribution function π_X is given, the possibility measure of an 'event' A, as a subset of the universe of discourse U, is written as

$$\pi_X(A)$$

or

$$Poss\{X \in A\}$$

and defined as the supremum

$$\bigvee_{u \in A} \pi_X(u).$$

In case we would have 'fuzzy events' A represented by membership functions μ_A, the possibility measure can be extended as

$$Poss\{X \ is \ A\} = \bigvee_{u \in A} \mu_A(u) \wedge \pi_X(u).$$

12.3. Fuzzy sentences

In discussions on "possibility and information", [1] comes to the dilemma concerning the "implied attribute" 'A' in '$A(X)$'. This leads to introducing the mapping I in order to assign membership values as

$$I(X \ is \ F)$$

and even if the domain of I is not explicitly spelled out, it is

$$I : \mathfrak{G} \longrightarrow [0, 1]$$

where $\mathfrak{G}$ is the set of all expressions like '$X \ is \ F$', i.e., basically all 'fuzzy events', but $\mathfrak{G}$ is not explicitly assumed to have similar properties like the σ-algebra $\mathfrak{F}$ for probability theory. Furthermore, I is not far from $Poss$. In $Poss$, the identification $A(X) = X$ seems still to be valid, whereas I allows A to appear. However, A as the "implied attribute" is still not explained, but remains as a black box.

Logically this can be seen as A is hiding the underlying signature Σ needed to open up the "inner structures" of samples, and that A is actually providing the indexing of I,

$$I_A$$

in the same way as

$$\mathsf{Sen}_\Sigma$$

provides the sentence functor related to the underlying signature Σ, and is needed when the theoremata functor Δ composes with Sen_Σ.

Observation 12.3. *Poss and I can therefore, similarly as Prob, be seen as "possibilistic valuations" of theoremata. 'Conditional possibility' is also introduced in [1], and is somewhat comparable with notation and criteria for 'conditional probability', but these concepts are different.*

Open Problem 12.2. *Possibilistic conditionality and conditional possibility can be investigated as entailments in the sense of 'generalized general logics' [13], and several options for such entailments should be tried out in connection with corresponding* Δ*'s.*

Is the "proposition"

if it is impossible that p, then it is possible that p

logically correct? The question cannot have an answer, since in this expression the "if ... then ..." construction cannot be a logical implication, since $Poss(p)$ is not a sentence. This expression "mixes bags".

In [14], there is a broad discussion on how translation rules between, on the one hand, possibility theory and its "hidden logic", and, on the other hand, natural language, pertain, respectively, to modification, composition, quantification and qualification. All these are then fundamentally related to logic language constructions, where the role of modification and composition is more about unary and binary operations on terms, whereas quantification may, as we have shown, need to involve type constructors, if quantification restricts to being over sets of terms. Quantification over sentences is 'operating' on sentences, and these 'operations' cannot be seen as part of any signature, but are closer to being natural transformations between sentence functors. Universal and existential quantifiers are special examples, and in this case it is even unclear if some general 'quantifier symbol' can quantify terms and sentences in general, or if term and sentence constructors "own their quantifications". In [3], this was discussed in the case of 'λ' in lambda-calculus, where the three level signature shows how each operator in fact own their capacity to abstract themselves, and that in fact it is not possible to have a general-purpose 'abstractor' λ. In traditional lambda-calculus the definition of the 'set of lambda terms' is seen as elegant, but is actually not well-defined without particular rules to fix problems that occur if variables are not renamed every now and then. The existential quantifier $\exists$ may well need to be examined in a similar fashion.

13. Possibilistic logic

A similar exposition of possibilistic logic will now follow using concepts and notations as adopted in [15].

Possibility measures and possibility distributions in [15] are treated using the same notation, and are based on the dual concept of a necessity measure N, being a mapping from "the set of logical formulas of a language" to a "totally ordered bounded scale", which for the purpose of these notes can be selected as the unit interval $[0, 1]$. This makes comparisons with possibility as treated in [1] and probability more transparent.

An important question at start is what we here really mean by "language". Possibilistic logic in [15] is almost exclusively propositional, even if the universal quantifier appears in connection with "particularization" in connection with a brief discussion on "classical resolution" and inference rules. The use of "classical" clearly indicates that [15] has no ambition to allow for uncertainties in underlying signatures, if and when such a treatment of possibilistic logic would move towards "first-order" aspects. However, it is stated that "first-order possibilistic logic formula is essentially a pair made of a classical first order logic formula and a weight expressing certainty or priority", which clearly collapses the framework to a fuzzy logic only involving uncertainty as related to algebraic manipulation of truth values.

Indeed, [15] is propositional, and the "set of logical formulas" is then very much comparable to the realm as provided by the sample set E and the σ-algebra $\mathfrak{F}$ in probability theory, and the 'universe of discourse' U in possibility theory [1]. This then also means that the underlying signature for possibilistic logic is basically one-sorted, with propositional constants, and with required logical operators. Whereas "the set of statements" in probability theory as well as in possibility theory are closer to being theoremata, "logical formulas" in possibilistic logic are not sentences, but actually only terms. They can be viewed as sentences as well if using the identity sentence functor to provide a term with such a double role.

13.1. Terms

Indeed, elements from that "totally ordered bounded scale" can be dressed up syntactically as logical constants in an imaginary signature $\texttt{PossLog} = (S, \Omega)$, where S consists of a single sort, say $\texttt{bool}$, and Ω contains at least

$$\top, \bot :\to \texttt{bool}, \neg : \texttt{bool} \to \texttt{bool}, \wedge : \texttt{bool} \times \texttt{bool} \to \texttt{bool}.$$

Additionally it is tempting to add

$$\equiv : \texttt{bool} \times \texttt{bool} \to \texttt{bool}$$

which potentially produces terms, i.e., "logical formulas" like $p{\equiv}q$, but the necessity measure N seems not naturally applicable as $N(p{\equiv}q)$, even if, by the time of discussions related to conditionality and reasoning, "possibility of a material implication" is said significantly to differ from "possibilistic conditioning". Indeed, the "syntax-independence axiom"

$$p{\equiv}q \Rightarrow N(p) = N(q)$$

is said to assume $\equiv$ to be the "equivalence in classical logic", so the axioms of possibility logic kind of merges axioms that are comparable with Kolmogorov's axioms for probability theory with this "syntax-independence axiom", where the $\Rightarrow$ then also appears to be some kind of "external logic operation".

Given this "merger" it is unclear if we should keep $\equiv$ in Ω or if it is outside. If, for the time being, it is not part of Ω, then, given a set $X_{\mathtt{bool}}$ of propositional variables, the "set of all formulas" is the set of terms

$$\mathsf{T}_{\mathtt{PossLog}} X_{\mathtt{bool}}$$

where indeed it remains somewhat unclear how to consider $p{\equiv}q$ as an operator in PossLog.

The necessity measure now in fact possesses all properties of a PossLog-algebra $\mathfrak{A}_{\mathtt{PossLog}}$, where

$$\mathfrak{A}_{\mathtt{PossLog}}(\top) = 1, \mathfrak{A}_{\mathtt{PossLog}}(\bot) = 0,$$

$$\mathfrak{A}_{\mathtt{PossLog}}(\mathtt{neg})(\mathfrak{A}_{\mathtt{PossLog}}(p)) = 1 - \mathfrak{A}_{\mathtt{PossLog}}(p),$$

and

$$\mathfrak{A}_{\mathtt{PossLog}}(\wedge) = min,$$

even if it is never considered to view N as actually operational given a $\mathfrak{A}_{\mathtt{PossLog}}$. The possibility measure

$$\Pi(p)$$

is then just a shorthand notation for

$$1 - N(\neg\, p).$$

In such a presentation of possibilistic logic according to [15], there is not a deliberate and clear distinction about syntactic and semantic objects. Indeed, since "min" is selected as the semantic model of $\wedge$, and "one minus" as the semantics of $\neg$, the treatment in [15] wouldn't in fact suffer severly if N would be seen as a mapping

$$N : [0,1] \longrightarrow [0,1].$$

This in the end then shows how possibilistic logic also hides "attributes", and even more drastically as compared to how possibility theory hides the [1] "implied attributes".

Note also that syntactic constants, other than $\top$ and $\bot$, are not included e.g. like

$$\mathtt{a} :\to \mathtt{bool}$$

with

$$\mathfrak{A}_{\mathtt{PossLog}}(\mathtt{bool}) = [0,1]$$

so that

$$\mathfrak{A}_{\mathtt{PossLog}}(\mathtt{a}) \in [0,1].$$

Either there is then one a for each $a \in [0,1]$, so that $\mathfrak{A}_{\mathtt{PossLog}}(\mathtt{a}) = a$, or there is just a selection of them. Some preservation condition may then be desirable. However, these syntactic matters are not discussed in [15].

Later on a conditional operator

$$| : \mathtt{bool} \times \mathtt{bool} \to \mathtt{bool}$$

is also introduced, where the definition of a "qualitative" $N(p \mid q)$ is said to "make sense in a finite (language) setting only".

13.2. Semantics and satisfaction

Having done syntactic representations, in [15] there is then kind of a "universe of discourse" of models or interpretation, which clearly implies that possibilistic logic really is propositional. There is at start no notation for this universe of models, so for the time being we may use U for that universe, and, following notations in [15], having $\omega \in U$ as particular models. A notation

$$\omega \models p$$

is introduced for "ω is a model for p", and a possibility measure π, as defined in [1], is introduced to express uncertainty $\pi(p)$ of models p. However, properties of that satisfaction relation $\models$ is not discussed or presented.

Entailment, together with soundness and completeness aspects is also covered in [15], but not embracing any scope of signatures and terms that would require consideration of substitution and assignment.

14. EXAMPLES

This section provides some preliminary examples, or just indications of examples, some of which appear in the above. While the exposition above complied with traditional notation in probability theory and possibility theory, in developing these examples we primarily comply with the logic language and categorical framework underlying descriptions in [3], aiming at presenting the expressive power and depth of this framework. Signatures will be important and in many case also the three level arrangement of signatures.

These examples intend to show, that uncertainty and uncertainty representation is not so much about

`which number we find`

but

`where we actually annotate it.`

If we are unsorted and work in analysis, there is no place for annotation. From database point of view, we might say we are able to be more flexible about attributes in an `XML File` as compared to being constrained by typing as provided in `XML Schema`. In `XML File`, the application developer may implement a signature of choice, whereas in `XML Schema`, a database management systems provides the "typing jacket". In data analytics, the problem is often having no database and no signature, and in public repositories and information systems, the problem is sometimes having 'too much database' but still without a proper underlying signature, meaning that authorities and stakeholders of various kind do not speak the same language.

14.1. 'Small' integer

This example is related to the "X is a small integer" example in [1], and is an ingredient in more extensive efforts to provide a formal logic based 'fuzzification of arithmetic', as initiated in [16].

```
 Integers are for counting or scoring, where small numbers
and their internal relations are important.  'Small' clearly
depends on the application context, but usually we would say
'4' is smaller than '7' more distinctly as saying '4' is smaller
than '5', and often '1004' being smaller than '1007' is not
very meaningful.  Again, the application context is important.
Referring to the golf example, we should also note that when
we speak of 80 and 110 meters, semantically, then symbolically
and syntactically we are closer to using '8' and '11'.  'Small'
appears mostly to be a unary operator, i.e., it operates on
a term (or on a set of terms simultaneously), and it may state
``well, yes, this is (or they are) rather small''.  Apart
from 'small', as being a unary operator, 'smaller than' as
a binary operator is more complex.  We may additionally be
```

```
interested in 'how much smaller', so such binary operations
may connect with a truth value or some other type of value.
Clearly, there are many ways of looking at such applications,
and there is no canonic signature supporting all kinds of
sentence developments.
```

For comparison, in [1] the approach and solution is based on

$$X = u : \mu_F(u)$$

where u is a natural number, i.e., $u \in \mathbb{N}$, and μ_F is a given fuzzy set on natural numbers, i.e., $\mu_F : \mathbb{N} \longrightarrow [0,1]$. In [1], the specifically mentioned μ_F is denoted

$$\textit{small integer} = 1/1 + 1/2 + 0.8/3 + 0.6/4 + 0.4/5 + 0.2/6$$

so that e.g. $\mu_F(5) = 0.4$ and $\mu_F(8) = 0$. This $\mu_F(5)$ can then be used in various ways, and the reader is referred to [1] for following the route of possibility theory.

In our notation [3], we will show how the signature for 'small numbers' could be arranged, respectively, over Set and Set($\mathfrak{Q}$), and, respectively, using a traditional single level signature the three level signatures framework.

Firstly, we will look at a typical single level signature over Set trying to capture the intuition in this problem formulation. We could initially define

$$\Sigma_{\texttt{SMALL}} = (S, \Omega)$$

over Set, to be 2-sorted, with one sort for the integers, and another sort for the truth values, i.e.,

$$S = \{\texttt{nat}, \texttt{bool}\}.$$

In this case the algebra $\mathfrak{A}_{\Sigma_{\texttt{SMALL}}}$ would typically define

$$\mathfrak{A}_{\Sigma_{\texttt{SMALL}}}(\texttt{nat}) = \mathbb{N}, \mathfrak{A}_{\Sigma_{\texttt{SMALL}}}(\texttt{bool}) = [0,1].$$

Operators in Ω are at least

$$\texttt{0} :\to \texttt{nat}, \texttt{succ} : \texttt{nat} \to \texttt{nat}$$

and

$$\texttt{false}, \texttt{true} :\to \texttt{bool}, \neg : \texttt{bool} \to \texttt{bool}, \& : \texttt{bool} \times \texttt{bool} \to \texttt{bool}.$$

Additionally, we may desire to have a set of constants

$$\texttt{a} :\to \texttt{bool}$$

with $\mathfrak{A}_{\Sigma_{\texttt{SMALL}}}(\texttt{a}) \in [0,1]$, maybe even so that for each $a \in [0,1]$ we have an $\texttt{a} :\to \texttt{bool}$ satisfying $\mathfrak{A}_{\Sigma_{\texttt{SMALL}}}(\texttt{a}) \in [0,1]$. We don't have to do that, but if we don't, and we somewhere along the line prefer to have a variable $p :: \texttt{bool}$, then substitution with p can only involve false or true, or some other variables $q :: \texttt{bool}$. In such a case, the term set is not so rich, and all algebraic manipulations of truth values take place only on the semantic side.

As far as '$\mathtt{small}$' is concerned, it appears to be reasonable to model it as an operator like

$$\mathtt{small} : \mathtt{nat} \to \mathtt{bool}$$

so that e.g.

$$\mathtt{small}(\mathtt{succ}(\mathtt{succ}(\mathtt{succ}(0)))) :: \mathtt{bool}$$

is a term. The most obvious semantics for $\mathtt{0}$ and $\mathtt{succ}$ being $\mathfrak{A}_{\Sigma_{\mathtt{SMALL}}}(\mathtt{0}) = 0 \in \mathbb{N}$ and $\mathfrak{A}_{\Sigma_{\mathtt{SMALL}}}(\mathtt{succ}) : \mathfrak{A}_{\Sigma_{\mathtt{nat}}}(\mathtt{nat}) \longrightarrow \mathfrak{A}_{\Sigma_{\mathtt{nat}}}(\mathtt{nat})$ as defined by $\mathfrak{A}_{\Sigma_{\mathtt{SMALL}}}(\mathtt{succ})(x) = 1+x$, would enable a semantics for $\mathfrak{A}_{\Sigma_{\mathtt{SMALL}}}(\mathtt{small}) : \mathfrak{A}_{\Sigma_{\mathtt{SMALL}}}(\mathtt{nat}) \longrightarrow \mathfrak{A}_{\Sigma_{\mathtt{SMALL}}}(\mathtt{bool})$, i.e., $\mathfrak{A}_{\Sigma_{\mathtt{SMALL}}}(\mathtt{small}) : \mathbb{N} \longrightarrow [0,1]$ that could e.g. fulfill

$$\mathfrak{A}_{\Sigma_{\mathtt{SMALL}}}(\mathtt{small})(\mathfrak{A}_{\Sigma_{\mathtt{SMALL}}}(\mathtt{succ})(\mathfrak{A}_{\Sigma_{\mathtt{SMALL}}}(\mathtt{succ})(\mathfrak{A}_{\Sigma_{\mathtt{SMALL}}}(\mathtt{succ})(\mathfrak{A}_{\Sigma_{\mathtt{SMALL}}}(\mathtt{0}))))) = 0.8,$$

i.e.,

$$\mathfrak{A}_{\Sigma_{\mathtt{SMALL}}}(((0+1)+1)+1) = 0.8,$$

thus being in a semantic correlation with the possibility theory approach $X = u : \mu_F(u)$, where $X = 3 : 0.8$.

Similarly as we allow ourselves to speak about 'small integer', we might also desire to speak about 'moderately true' and 'certainly true'. These are then obviously modelled as operators

$$\mathtt{mild}, \mathtt{moderate} : \mathtt{bool} \to \mathtt{bool}$$

with some suitable algebraic interpretations

$$\mathfrak{A}_{\Sigma_{\mathtt{SMALL}}}(\mathtt{mild}), \mathfrak{A}_{\Sigma_{\mathtt{SMALL}}}(\mathtt{moderate}) : [0,1] \longrightarrow [0,1]$$

so that we would be able to use terms like

$$\mathtt{mild}(\mathtt{small}(\mathtt{succ}(\mathtt{succ}(0))))$$

and

$$\mathtt{moderate}(\mathtt{small}(\mathtt{succ}(\mathtt{succ}(0)))).$$

Secondly, we have the option to view a single level signature over $\mathtt{Set}(\mathfrak{Q})$, so that modelling something like "the more you add, the worse it gets", could mean annotating $\mathtt{succ} \in \Omega$ with an uncertainty, using $\varsigma : \Omega \longrightarrow [0,1]$. Further, $\mathtt{small}$ is also annotated with a value $\varsigma(\mathtt{small})$, which may be '1' in case some 'external (human) observation' of something being 'small' is a certain or fully believable observation, but can also be less than '1', even as self-evaluated by the observer. This is what happened in the case of depression and observing values into the GDS assessment scale [5].

Using $\mathtt{Set}$ or $\mathtt{Set}(\mathfrak{Q})$ is indeed fundamentally different, as being over $\mathtt{Set}$ means we are delivering a "(crisp) observation of a fuzzy observable", whereas being over $\mathtt{Set}(\mathfrak{Q})$ we have a truly "fuzzy observation", which can be applied both for a crisp as well as a fuzzy observable. In the case of the depression scale, the observable 'no/yes' is crisp, but the observation of it is self-reportedly fuzzy.

Thirdly, we have the three level signature arrangement over $\mathtt{Set}$, which brings in an entirely new apparatus for arranging sorts and operations. Note how in the single level case we do have a sort $\mathtt{nat}$ for 'integer', or if we use apparently obsolete brackets, we could write '⟨integer⟩'. However, up to that point, we never suggested to define a separate sort for 'fuzzy integer'. Indeed, 'fuzzy' is usually not seen as a syntactic component, but as a pure semantic arrangement for fuzzy sets and membership functions.

```
How can we have truly 'fuzzy logic',

if 'fuzzy' is outside the syntactic scope?
```

In order to discuss the alternatives for a typing of 'fuzzy integer', we could informally, now using brackets, write '⟨fuzzy number⟩' as distinct from 'fuzzy ⟨number⟩'. Intuitively, these are two entirely different concepts, i.e., different underlying sorts, where '⟨fuzzy number⟩' would be typed on level one by a separate sort, call it $\mathtt{fuznat}$, and 'fuzzy ⟨number⟩' would require a type constructor $\mathtt{fuz} : \mathtt{type} \to \mathtt{type}$ on level two, so that eventually the term $\mathtt{fuz}(\mathtt{nat}) :: \mathtt{type}$ on level two becomes a sort $\mathtt{fuz}(\mathtt{nat}) \in S'$ on level three.

This is also how the previously discussed

$$\mathtt{large} = \mathtt{wide} \boxtimes \mathtt{long}$$

can enter an underlying signature for a specific application. Inside a house or apartment, we often prefer to be cartesian and having 90 degree angles at least every now and then, but in a garden or in nature this is not always the case. Nevertheless, 'long', 'wide' and 'large' do play some roles, and being 'monoidal' is somehow important for scope and extent in the garden, but being exclusively 'cartesian' may seem too restrictive.

Finally, the three level signature arrangement over $\mathtt{Set}(\mathfrak{Q})$ is then a similar extension as for going from $\mathtt{Set}$ to $\mathtt{Set}(\mathfrak{Q})$ on level one only. Note how this means we may want to have $\mathtt{Set}(\mathfrak{Q})$ on level one also if we decide to have it on level three, since some semantic preservations may be needed between level one and level three. For level two, having $\mathtt{Set}(\mathfrak{Q})$ as the underlying category, means in effect that sorts and type constructors come equipped with uncertainty, which requires an even further extension of this logic machinery, since then also sorts become uncertain. Computer science indeed speaks about "polymorphic types", but this is an entirely different style of typing as compared to sorts themselves being uncertain. However, doesn't this actually appear in natural language, namely, that statements are allowed to change form and context on-the-fly?

14.2. 'Rich' and 'wealthy'

The operator for 'small' in the previous example was more "uncomposed" as an operator, and as such, operating on less complex terms. In all, the signature $\Sigma_{\mathtt{SMALL}}$ is rather trivial, and there are not so many real-world applications to be developed exclusively based on such a signature. However, it provided an example for

```
where we actually annotate it.
```

'Rich' is obviously more complicated, in particular if underlying data is supposed to go far beyond just a number on a bank account. If it would be simply all about that number on a bank account, then we are facing a simple issue of 'large', in signature complexity comparable to $\Sigma_{\texttt{SMALL}}$, only with other operators and suitable algebras. So we assume it is more than that number on a bank account.

We may want to consider more complex situations and conditions, like those for 'wealthy' and 'healthy'.

```
 Whether we are 'wealthy' or 'healthy', we try to maintain
it, and even increase and improve it, and if it diminishes,
we try to intervene or prevent, in order to stop or avoid
that ''unfavourable progrediation''.  This then brings us
to understand the difference between 'iatrogenic' intervention
and 'salutogenic' prevention.  Note also that, concerning
'wealth' and 'health', it is not always clear what is intervention
and what is prevention.  Investment may appear to be a preventive
action, whereas furlough is intervention, similar as increasing
or maintaining 'muscle strength' using physiotherapy is often
seen as a preventive action, whereas a hip fraction is a case
for surgical intervention.
```

See [17] for detail concerning a logic and ontology of assessment of conditions in older people. Such an information modelling is part also of on-going activities concerning European cooperation in the area of *Active and Healthy Ageing*, which is an 'Innovation Partnership', announced as one of the *Key Initiatives* by the *European Commission*. Other key initiatives fall under *Knowledge*, *Good ideas to market*, *Regional and social benefits* and SCIENCE as expected to be executed within *International Cooperation*.

APPENDIX — Category theory notations

Basic concepts and notations

In a category $\mathtt{C}$ with objects A and B, morphisms f from A to B are typically denoted by $f : A \longrightarrow B$ or $A \xrightarrow{f} B$. The (A-)identity morphism is denoted $A \xrightarrow{\mathrm{id}_A} A$ and morphism composition uses $\circ$. The set of $\mathtt{C}$-morphisms from A to B is written as $\mathrm{Hom}_{\mathtt{C}}(A, B)$ or $\mathrm{Hom}(A, B)$.

The category of sets, $\mathtt{Set}$, is the most typical example of a category, and consists of sets as objects and functions (in ZFC) as morphisms together with the ordinary composition and identity. Other categories may be defined, for example, using $\mathtt{Set}$ as a basis: a structure, defined by the given metalanguage, is added on $\mathtt{Set}$-objects, and then morphisms are defined as $\mathtt{Set}$-morphisms preserving these structures. A typical example is to add uncertainty, modelled by a quantale $\mathfrak{Q}$, on $\mathtt{Set}$-objects: The objects of the Goguen category $\mathtt{Set}(\mathfrak{Q})$ are pairs (X, α), where X is an object of $\mathtt{Set}$ and $\alpha\colon X \longrightarrow Q$ is a function (in ZFC). The morphisms $(X, \alpha) \xrightarrow{f} (Y, \beta)$ are $\mathtt{Set}$-morphisms $X \xrightarrow{f} Y$ satisfying $\alpha \leq \beta \circ f$. The composition of morphisms is defined as composition of $\mathtt{Set}$-morphisms. Originally, Goguen considered a completely distributive lattice as the underlying lattice in [18] and further properties for Goguen categories can be found in [19].

A (covariant) *functor* $\mathsf{F} : \mathtt{C} \longrightarrow \mathtt{D}$ between categories is a mapping that assigns each $\mathtt{C}$-object A to a $\mathtt{D}$-object $\mathsf{F}(A)$ and each $\mathtt{C}$-morphism $A \xrightarrow{f} B$ to a $\mathtt{D}$-morphism $\mathsf{F}(A) \xrightarrow{\mathsf{F}(f)} \mathsf{F}(B)$, such that $\mathsf{F}(f \circ g) = \mathsf{F}(f) \circ \mathsf{F}(g)$ and $\mathsf{F}(\mathrm{id}_A) = \mathrm{id}_{\mathsf{F}(A)}$. Composition of functors is denoted $\mathsf{G} \circ \mathsf{F} : \mathtt{C} \longrightarrow \mathtt{E}$ and the identity functor is written $\mathrm{id}_{\mathtt{C}} : \mathtt{C} \longrightarrow \mathtt{C}$. The (covariant) powerset functor $\mathsf{P} : \mathtt{Set} \longrightarrow \mathtt{Set}$ is the typical example of a functor, and is defined by $\mathsf{P}A$ being the powerset of A, i.e., the set of subsets of A, and $\mathsf{P}f(X)$, for $X \subseteq A$, being the image of X under f, i.e., $\mathsf{P}f(X) = \{f(x) \mid x \in X\}$. A contravariant functor $\mathsf{F} : \mathtt{C} \longrightarrow \mathtt{D}$ maps to each $\mathtt{C}$-morphism $A \xrightarrow{f} B$ a $\mathtt{D}$-morphism $\mathsf{F}(B) \xrightarrow{\mathsf{F}(f)} \mathsf{F}(A)$, and for the contravariant powerset functor $\overline{\mathsf{P}} : \mathtt{Set} \longrightarrow \mathtt{Set}$ we have $\overline{\mathsf{P}}A = \mathsf{P}A$ and $\overline{\mathsf{P}}f(Y) = \{x \in X \mid \exists y \in Y : f(x) = y\}$.

A *natural transformation* $\tau : \mathsf{F} \longrightarrow \mathsf{G}$ between functors assigns to each $\mathtt{C}$-object A a $\mathtt{D}$-morphism $\tau_A : \mathsf{F}A \longrightarrow \mathsf{G}A$ such that $\mathsf{G}f \circ \tau_A = \tau_B \circ \mathsf{F}f$, for any $f : A \longrightarrow B$. The identity natural transformation $\mathsf{F} \xrightarrow{\mathrm{id}_{\mathsf{F}}} \mathsf{F}$ is defined by $(\mathrm{id}_{\mathsf{F}})_A = \mathrm{id}_{\mathsf{F}A}$. If all τ_A are isomorphisms, τ is called a *natural isomorphism*, or *natural equivalence.* For functors F and natural transformations τ we often write $\mathsf{F}\tau$ and $\tau\mathsf{F}$ to mean $(\mathsf{F}\tau)_A = \mathsf{F}\tau_A$ and $(\tau\mathsf{F})_A = \tau_{\mathsf{F}A}$, respectively. It is easy to see that $\eta : \mathrm{id}_{\mathtt{Set}} \longrightarrow P$ given by $\eta_X(x) = \{x\}$, and $\mu : P \circ P \longrightarrow P$ given by $\mu_X(\mathcal{B}) = \bigcup \mathcal{B} (= \bigcup_{B \in \mathcal{B}} B)$ are natural transformations. The (vertical) composition $\sigma \circ \tau : \mathsf{F} \longrightarrow \mathsf{H}$ of natural transformations is defined by $(\sigma \circ \tau)_A = \sigma_A \circ \tau_A$, for all $\mathtt{D}$-objects A.

Whereas morphisms are typically seen as 'mappings' between objects in a category, functors are 'mappings' between categories, i.e., morphisms in (quasi-

)categories of categories, and natural transformations are 'mappings' between functors, i.e., morphisms in functor categories. These notions clearly lead to views on hierarchies of sets, classes and conglomerates, where foundational issues enter the scene, and our approach roughly follows Grothendieck's [20] and Gähler's [21] views of set-theoretic foundations for category theory.

A *monad* (or triple, or algebraic theory) over a category $\mathtt{C}$ is written as $\mathbf{F} = (\mathsf{F}, \eta, \mu)$, where $\mathsf{F} : \mathtt{C} \longrightarrow \mathtt{C}$ is a (covariant) functor, and $\eta : \mathrm{id} \longrightarrow \mathsf{F}$ and $\mu : \mathsf{F} \circ \mathsf{F} \longrightarrow \mathsf{F}$ are natural transformations for which $\mu \circ \mathsf{F}\mu = \mu \circ \mu\mathsf{F}$ and $\mu \circ \mathsf{F}\eta = \mu \circ \eta\mathsf{F} = \mathrm{id}_{\mathsf{F}}$ hold. A Kleisli category $\mathtt{C}_{\mathbf{F}}$ for a monad $\mathbf{F}$ over a category $\mathtt{C}$ is defined as follows: Objects in $\mathtt{C}_{\mathbf{F}}$ are the same as in $\mathtt{C}$, and the morphisms are defined as $\mathrm{Hom}_{\mathtt{C}_{\mathbf{F}}}(X, Y) = \mathrm{Hom}_{\mathtt{C}}(X, \mathsf{F}Y)$, that is morphisms $f : X \rightharpoonup Y$ in $\mathtt{C}_{\mathbf{F}}$ are simply morphisms $f : X \longrightarrow \mathsf{F}Y$ in $\mathtt{C}$, with $\eta_X : X \longrightarrow \mathsf{F}X$ being the identity morphism on X. Composition of morphisms is defined as

$$(X \xrightarrow{f} Y) \diamond (Y \xrightarrow{g} Z) = X \xrightarrow{\mu_Z \circ \mathsf{F}g \circ f} \mathsf{F}Z.$$

The category $\mathtt{Rel}$ with sets as objects and binary relations as morphisms, is isomorphic with the Kleisli category of the powerset monad over $\mathtt{Set}$. This invites to viewing Kleisli morphisms as a general notion for relations in the sense of intuitively being "substitutions".

Powerset monads and their many-valued extensions are in close connection to fuzzification and are good candidates to represent situations with incomplete or imprecise information. The many-valued covariant powerset functor L for a completely distributive lattice $\mathfrak{L} = (L, \vee, \wedge)$ is obtained by $\mathsf{L}X = L^X$, i.e. the set of functions (or $\mathfrak{L}$-sets) $\alpha : X \longrightarrow L$, and following [18], for a morphism $f : X \longrightarrow Y$ in $\mathtt{Set}$, by defining $\mathsf{L}f(\alpha)(y) = \bigvee_{f(x)=y} \alpha(x)$. Further, if we define $\eta_X : X \longrightarrow \mathsf{L}X$ by

$$\eta_X(x)(x') = \begin{cases} \top & \text{if } x = x' \\ \bot & \text{otherwise} \end{cases}$$

and $\mu : \mathsf{L} \circ \mathsf{L} \longrightarrow \mathsf{L}$ by

$$\mu_X(\mathcal{M})(x) = \bigvee_{\alpha \in \mathsf{L}X} A(x) \wedge \mathcal{M}(\alpha)$$

then $\mathbf{L} = (\mathsf{L}, \eta, \mu)$ is a monad.

Sorted categories

In the one-sorted (and crisp) case for signatures we typically work in $\mathtt{Set}$, but in the many-sorted (and crisp) case we need the "sorted category of sets" for the many-sorted term functor. We start this section by a more general view by considering "a sorted category of objects".

Let S be an index set (in ZFC), the indices are called *sorts* (or types), and we do not assume any order on S. For a category $\mathtt{C}$, we write $\mathtt{C}_S$ for the product category $\prod_S \mathtt{C}$. The objects of $\mathtt{C}_S$ are tuples $(X_{\mathtt{s}})_{\mathtt{s} \in S}$ such that $X_{\mathtt{s}} \in \mathrm{Ob}(\mathtt{C})$ for all $\mathtt{s} \in S$. We will also use X_S as a shorthand notation for these tuples. The morphisms between objects $(X_\mathtt{s})_{\mathtt{s}\in S}$ and $(Y_\mathtt{s})_{\mathtt{s}\in S}$ are tuples $(f_\mathtt{s})_{\mathtt{s}\in S}$ such that $f_\mathtt{s} \in \mathrm{Hom}_\mathtt{C}(X_\mathtt{s}, Y_\mathtt{s})$ for all $\mathtt{s} \in S$, and similarly we will use f_S as a shorthand notation. The composition of morphisms is defined sortwise (componentwise), i.e., $(g_\mathtt{s})_{\mathtt{s}\in S} \circ (f_\mathtt{s})_{\mathtt{s}\in S} = (g_\mathtt{s} \circ f_\mathtt{s})_{\mathtt{s}\in S}$.

Functors $\mathsf{F}_\mathtt{s} \colon \mathtt{C} \longrightarrow \mathtt{D}$ are lifted to functors $\mathsf{F} = (\mathsf{F}_\mathtt{s})_{\mathtt{s}\in S}$ from $\mathtt{C}_S$ to $\mathtt{D}_S$. so that e.g. the regular powerset functor $\mathsf{P}_S = (\mathsf{P})_{\mathtt{s}\in S}$ and the regular many-valued powerset functor $\mathsf{L}_S = (\mathsf{L})_{\mathtt{s}\in S}$, both are lifted to functors on $\mathtt{Set}_S$.

Products and coproducts, $\prod$ and $\coprod$, are handled sortwise. We also have a "subobject relation", thus, $(X_\mathtt{s})_{\mathtt{s}\in S} \subseteq (Y_\mathtt{s})_{\mathtt{s}\in S}$ if and only if $X_\mathtt{s} \subseteq Y_\mathtt{s}$ for all $\mathtt{s} \in S$. It is clear that all limits and colimits exist in $\mathtt{Set}_S$, because operations on $\mathtt{Set}_S$-objects are defined sortwise for sets. Further, the product $\prod_{i\in I} \mathsf{F}_i$ and coproduct $\coprod_{i\in I} \mathsf{F}_i$ of covariant functors F_i over $\mathtt{Set}_S$ are defined as

$$(\prod_{i\in I} \mathsf{F}_i)(X_\mathtt{s})_{\mathtt{s}\in S} = \prod_{i\in I} \mathsf{F}_i(X_\mathtt{s})_{\mathtt{s}\in S}$$

and

$$(\coprod_{i\in I} \mathsf{F}_i)(X_\mathtt{s})_{\mathtt{s}\in S} = \coprod_{i\in I} \mathsf{F}_i(X_\mathtt{s})_{\mathtt{s}\in S}$$

with morphisms being handled accordingly.

The category $\mathtt{Set}(\mathfrak{Q})_S$ is called the *many-sorted Goguen category.* Objects in this category are tuples of pairs $((X_\mathtt{s}, \alpha_\mathtt{s}))_{\mathtt{s}\in S}$ as objects, where for each $\mathtt{s} \in S$, $\alpha_\mathtt{s} \colon X_\mathtt{s} \longrightarrow Q$ is a function (in ZFC). So, fixing $\mathtt{s} \in S$ we consider pairs $(X_\mathtt{s}, \alpha_\mathtt{s})$ as objects in $\mathtt{Set}(\mathfrak{Q})$. Now, the $\mathtt{Set}(\mathfrak{Q})$-morphisms $(X_\mathtt{s}, \alpha_\mathtt{s}) \xrightarrow{f_\mathtt{s}} (Y_\mathtt{s}, \beta_\mathtt{s})$ form morphisms $((X_\mathtt{s}, \alpha_\mathtt{s}))_{\mathtt{s}\in S} \xrightarrow{(f_\mathtt{s})_{\mathtt{s}\in S}} ((Y_\mathtt{s}, \beta_\mathtt{s}))_{\mathtt{s}\in S}$.

```
@misc{   FuzzySetsAndSignatures,
  author = {Eklund, Patrik},
  title     = {Fuzzy Sets and Signatures},
  howpublished = {GLIOC Notes},
  month     = {October},
  year      = {2013},
  url       = {http://www.glioc.com}
}
```

References

[1] L. Zadeh, Fuzzy sets as a basis for a theory of possibility, Fuzzy Sets and Systems 1 (1978) 3–28.

[2] H. Prade, Modèles mathématiques de l'imprécis et de l'incertain en vue d'applications au raisonnement naturel, Ph.D. thesis, Université Paul Sabatiere (1982).

[3] P. Eklund, M. Galán, R. Helgesson, J. Kortelainen, Fuzzy terms, Fuzzy Sets and Systems, in press.

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

[5] P. Eklund, Signatures for assessment, diagnosis and decision-making in ageing, in: E. Hüllermeier, R. Kruse, F. Hoffmann (Eds.), IPMU (2), Vol. 81 of Communications in Computer and Information Science, Springer, 2010, pp. 271–279.

[6] J. Yesavage, Development and validation of geriatric depression screening scale: A preliminary report, Journal of Psychiatry 17 (1983) 37–499.

[7] P. Eklund, M. Galán, R. Helgesson, J. Kortelainen, G. Moreno, C. Vasquez, Towards categorical fuzzy logic programming, in: F. Masulli, G. Pasi, R. Yager (Eds.), WILF 2013 - 10th International Workshop on Fuzzy Logic and Applications, Vol. 8256 of Lecture Notes in Computer Science, Springer, 2013, 109-121.

[8] J. Bénabou, Structures algébriques dans les catégories, Cahiers de Topologie et Géométrie Différentielle Catégoriques 10 (1) (1968) 1–126.

[9] A. Church, A formulation of the simple theory of types, The journal of symbolic logic 5 (2) (1940) 56–68.

[10] P. Eklund, J. Kortelainen, L. N. Stout, Adding fuzziness to terms and powerobjects using a monadic approach, Fuzzy Sets and Systems 192 (2012) 104–122.

[11] P. Eklund, M. A. Galán, M. Ojeda-Aciego, A. Valverde, Set functors and generalised terms, Proc. IPMU 2000, 8th Information Processing and Management of Uncertainty in Knowledge-Based Systems Conference III (2000) 1595–1599.

[12] M. Galán, Categorical unification, Ph.D. thesis, Umeå University, Department of Computing Science (2004).

[13] R. Helgesson, Generalized general logics, Ph.D. thesis, Umeå University, Department of Computing Science (2013).

[14] L. Zadeh, PRUF - a meaning representation language for natural languages, Int. J. Man-Machine Studies 10 (1978) 395–460.

[15] D. Dubois, H. Prade, Possibilistic logic: a retrospective and prospective view, Fuzzy Sets and Systems 144 (2004) 3–23.

[16] P. Eklund, M. Galán, J. Kortelainen, On fuzzy successors, in: L. Magdalena, M. Ojeda-Aciego, J. Verdegay (Eds.), Proceedings of IPMU'08, 21st Int. Conf. Information Processing and Management of Uncertainty in Knowledge-based Systems, 2008, pp. 124–128.

[17] P. Eklund, in: P. W. Guo, P. (Ed.), Human-Centric Decision-Making Models for Social Sciences, Studies in Computational Intelligence, Springer, 2014, p. in print.

[18] J. Goguen, L-fuzzy sets, Journal of Mathematical Analysis and Applications 18 (1967) 145–174.

[19] L. N. Stout, The logic of unbalanced subobjects in a category with two closed structures, in: S. Rodabaugh, E. Klement, U. Höhle (Eds.), Applications of Category Theory to Fuzzy Subsets, Kluwer Academic Publishers, 1992, pp. 73–106.

[20] N. Bourbaki, Séminaire de Géométrie Algébrique du Bois Marie - 1963–64 - Théorie des topos et cohomologie étale des schémas - (SGA 4) - vol. 1, Vol. 269 of Lecture notes in mathematics, Springer-Verlag, 1972.

[21] W. Gähler, Axioms of structures and functor power series, in: J. Novák, W. Gähler, H. Herrlich, J. Mikisiński (Eds.), Proc. Conference on Convergence Structures, Bechyně 1984, Akademie-Verlag, Berlin, 1984, pp. 138–152.