2006 IEEE International Conference on Fuzzy Systems
Sheraton Vancouver Wall Centre Hotel, Vancouver, BC, Canada
July 16-21, 2006

Syntax-driven analysis of context-free languages with respect to fuzzy relational semantics

Richard Bergmair, *Member, IEEE Computer Society,* and Ulrich Bodenhofer *Member, IEEE*

Abstract—**A grammatical framework is presented that augments context-free production rules with semantic production rules that rely on fuzzy relations as representations of fuzzy natural language concepts. It is shown how the well-known technique of syntax-driven semantic analysis can be used to infer from an expression in a language defined in such a semantically augmented grammar a weak ordering on the possible worlds it describes. Considering the application of natural language query processing, we show how to order elements in the domain of a relational database scheme according to the degree to which they fulfill the intuition behind a given natural language statement like "Carol lives in a small city near San Francisco".**

I. Introduction & Motivation

It may well be one of the major themes of our modern industrialized society that we have been, and are continuing to be, dependent on ever increasing amounts of information to run our corporations, institutions, and daily lives. When information technology first set out to solve this problem by computerized means, it promised a fountain of wisdom. What has been delivered is a flood of data. Obviously, the major new issue being addressed now in almost every discipline of data processing is the evaluation of data in terms of the intuitive notions of correctness, completeness, conciseness, confidence, quantity, relevance, etc. All of these features are vague in nature. Therefore, as we move on from talking about the truth of a given piece of data in a classical sense, to talking about features like the relevance of a given piece of data, we can no longer organize them in terms of strict binary partitioning. Instead, all of these features impose weak orderings of relevance, quantity, confidence, conciseness, and so on.

A search engine is not successful just because all of the 300 results it produces to a given query are truthful matches of the search expression entered. It is successful only if the first 10 results happen to be the most relevant. This is what PageRank enabled Google to do, and possibly the reason why they have become so successful. Operating system manufacturers have recently recognized the pressing need to provide users with similar levels of access to the floods of data accumulating on their own harddisks nowadays. Apple has already introduced its Spotlight desktop search, and Microsoft will include a similar mechanism with the upcoming system release of Vista.

Richard Bergmair is with the Cambridge University Computer Lab, William Gates Building, 15 JJ Thomson Avenue, Cambridge CB3 0FD, UK (email: rbergmair@acm.org).

Ulrich Bodenhofer is with the Software Competence Center Hagenberg, Hauptstrasse 99, A-4232 Hagenberg, AUSTRIA (email: Ulrich.Bodenhofer@scch.at).

From our point of view there are two major questions to be answered in the construction of such search systems: (a) How do everyday users wish to express what is relevant to them? (b) What model can a computer employ to respond to such a query in terms of an ordered result set? In the opinions of the authors the most straightforward answers are as follows. (a) Users want to express what is relevant to them in the same way they express every abstraction that might be on their minds: by means of natural language. (b) A given model that infers from a query expression a crisp result set can straightforwardly be turned into a model that infers a ranked result set by moving on from bivalent logic to fuzzy logic.

This might perhaps be true for every data organizing system. We, however, will consider the more specific problem of natural language query processing herein, in an attempt to provide an intelligent query facility for database systems. Consider, for example, a fictional database of the California registration office, holding data about its citizens, its cities, the distances between cities and about where citizens reside in a relational form following a model, such as the one defined in section IV.

How do we find a record in such a database, telling us whether or not Carol lives in a small city near San Francisco? [1]. Traditionally this would have been the most trivial application of a standard database. However, with the masses of data that populate today's databases this is becoming seriously non-trivial. What would a traditional interface to this database look like? It would probably allow a user to search for residency records either by name or by city, and for cities by population or by their distances from other cities. Unfortunately none of these query masks is of any use, when there are too many people called "Carol" in California, too many small cities and too many cities near San Francisco.

An SQL query would do slightly better, but still not solve the problem, as it is not straightforward how to translate concepts like "small city" or "near San Francisco" to SQL. Furthermore, experience with modern information retrieval systems has shown that users are not willing to learn special query languages and formulate non-trivial boolean queries. Thus, even something like a fuzzy SQL might still not be a full solution to the problem (see, for example, [2], [3], [4], [5], [6], [7] and many others).

The query language we will propose in section V is a first step towards a precisiated natural language [8], in the sense that expressions in this query language ought to be expressions in English and that the results of any such query ought to match the intuition a human will have about this

English expression. Our approach to the information access problem described is therefore to have a user enter the expression "Carol lives in a small city near San Francisco" in natural language.

In the present work we outline our approach by pointing out the major ideas, making them formally explicit, and providing a proof-of-concept construction. The present paper picks out the highlights of a more detailed exposition [9].

II. History of Prior Work

Although the importance of everyday expressions for the mathematical analysis of vague concepts was recognized quite early, for example in the domain of quantitative research methodology [10], not much has been done in the way of applying fuzzy logic [11] on a broader scale for the analysis of natural language. Although there was an attempt at defining a "fuzzified" formal language theory in the late 60s [12], it did not receive much attention, when fuzzy systems were only just beginning to be successfully applied to simple control tasks and formal language theory was of interest only to fields like compiler construction that naturally had little use for vague concepts.

Linguists, on the other hand, remained widely unaware of fuzzy logic and its potential for the formal treatment of vagueness in natural language, until Lakoff picked up the idea of fuzzy logic in the mid 1970s [13, p. 196]. Unfortunately, although most linguists have had little trouble accepting the idea of vague concepts, its impact on semantic theory has remained only of secondary interest to linguists, following the tradition of Chomsky [13, p. 50].

During the same period computer science saw the rise of artificial intelligence and its historically unparalleled interest in meaning representation. Considerable work on fuzzy meaning representation schemes was carried out by Goguen [14], who also attempted to build a fuzzy SHRDLU, a robot capable of carrying out commands input in natural language in the domain of a fuzzy microworld [15]. Zadeh proposed a fuzzy meaning representation scheme for natural languages as well [16]. However, these representation schemes were mainly concerned with meaning as such, rather than meaning in relation to natural languages. Later, Zadeh presented test-score semantics [17], [18] in an approach to bridge the gap between natural language representation and his fuzzy meaning representation. However his technique was never deployed in an actual grammar. A decade later, Novak presented what is probably the only work really concerned with the nuts and bolts of natural language from the point of view of fuzzy logic [19], [20].

Today it might be fair to say that, despite the many successes fuzzy logic has had since the 70s, it has failed to live up to the high expectations artificial intelligence enthusiasts once had, when they set out to deploy the technology to make machines *understand* the categories of reasoning that humans use to successfully communicate to each other vague ideas and concepts.

Only recently, Zadeh took up renewed interest in this line of research, addressing the main shortcoming when he observes that "progress has been, and continues to be, slow in those areas where a methodology is needed in which the objects of computation are perceptions—perceptions of time, distance, form, direction, color, shape, truth, likelihood, intent, and other attributes of physical and mental objects" [1]. The key point Zadeh has to make about perceptions is that they are inherently fuzzy, and that humans use natural language representations where machines use numeric measurements. Thus, the paradigm shift that takes Zadeh into his "new direction of artificial intelligence" is one that takes him "from computing with numbers to computing with words" [21]. The representations of fuzzy concepts employed in his computational theory of perceptions are linguistic in nature. They are expressions of a language he refers to as precisiated natural language [8]. Such a language would have to be natural, in the sense that it is a formal language weakly equivalent to a subset of a natural language, and precisiated, in the sense that every such expression can automatically be translated to a form suitable for approximate reasoning.

At this point we would like to highlight one rather questionable assumption underlying the more visionary end of Zadeh's ideas: that a reduction from the problem of "computing with words" to the strong AI problem is straightforward or even possible. For example Zadeh often cites applications such as parking a car, driving in city traffic, playing golf, or cooking a meal [1]—problems that actually do involve perceptions of time, distance, form, direction, color, or shape, and not just perceptions of language as such. His approach therefore presumes that representations of such perceptions in natural languages such as English or German pay justice to the actual objects of cognition, which assumes a flavor of Whorfianism possibly too strong for most contemporary linguists to savor.

Nevertheless, a technology as envisioned by Zadeh, that enables the computational manipulation of linguistic expressions describing fuzzy concepts remains highly desirable. In fact, for the technology described in this paper in particular, we can think at least of two immediate applications: Natural language interfaces to flexible query processing systems [22], [23], [24], and software tools supporting the implementation of fuzzy controllers in a linguistically intuitive way [25], [26], [27]. More remote applications of such technology may possibly include information extraction and retrieval and document classification.

These are, of course, textbook examples for applications of natural language processing, which is why the idea of using fuzzy logic in natural language processing seems obvious. Thus we found it rather surprising that, after an extensive search for the use of fuzzy technology in this field's literature, this work seems to be the first to draw this connection.

III. Ordering-Based Semantics

From the bird's eye's view, the major problem we will concern ourselves with in this paper is that of attributing ordering-based semantics to the expressions of a precisiated natural language. The notion of semantics we employ presumes a model in which a set of data $Dat =$

$\{\mathsf{d}_1, \mathsf{d}_2, \mathsf{d}_3, \ldots, \mathsf{d}_\mathsf{n}\}$ represents a collection of the possible worlds these expressions can denote. Any expression can then be taken to denote a constraint Con on these records. Such a constraint can be represented by an n-ary relation $Con \subseteq Dat^n$ on Dat for some n.

In standard query languages (such as SQL) the semantics of query expressions are taken to be partition based.

Definition 1: The *crisp* or *partition-based* semantics of a query expression, with respect to a set of records Dat is given by a unary relation $Con_c \subseteq Dat$ on Dat.
Such a relation Con_c partitions Dat into a set *True* of records in Dat that fulfill the constraint, and a set *False* of records in Dat that do not fulfill the constraint. This can be seen by letting $True = Con_c$ and $False = Dat \setminus Con_c$ or vice-versa.

In contrast to this crisp approach, we will take a fuzzy approach by taking the semantics of query expressions to be ordering-based:

Definition 2: The *fuzzy* or *ordering-based* semantics of a query expression, with respect to a set of records Dat is given by a binary relation $Con_f \subseteq Dat \times Dat$ on Dat which is reflexive, transitive and complete (i.e. a weak ordering).
Such a relation Con_f can establish a weak ordering on Dat such that $Con_f(\mathsf{d}_1, \mathsf{d}_2)$ if and only if d_1 satisfies the constraint "at least as well as" d_2.

IV. The Semantic Model: A Relational Database

The domain we operate in is defined in terms of a crisp data model that allows us to reason about cities, about distances between cities, about people, and about who of the people lives in which of the cities.

Our scheme involves the following relations:

- $\mathsf{Person}(\underline{p})$: The primary key p refers to a person.
- $\mathsf{Person\_Name}(\underline{p}, x)$: The person referred to by primary key p has name x.
- $\mathsf{Place}(\underline{p})$: The primary key p refers to a place.
- $\mathsf{Place\_Name}(\underline{p}, x)$: The place referred to by primary key p has name x.
- $\mathsf{Place\_Pop}(\underline{p}, x)$: The place referred to by primary key p has population x.
- $\mathsf{Place\_Distance}(\underline{p, q}, x)$: The places referred to by primary keys p and q are at a distance x from each other.
- $\mathsf{Lives\_In}(\underline{x, y})$: The tuple (x, y) is a primary key referring to a person x living in a place y.

In an industrial setup these relations will typically be database tables, or XML-files, but they might just as well be sets of PROLOG facts, or data from a lexicon represented in some special format. Figure 1 shows an entity-relationship diagram.

Definition 3: A *relational scheme* $Sch = (Dom, Rel)$ consists of

- A finite set Dom which establishes the data domain for all atomic values that appear throughout the database.
- A finite set Rel, of tuples $rel \in Rel$ of the form $rel = (R, n, p)$, where

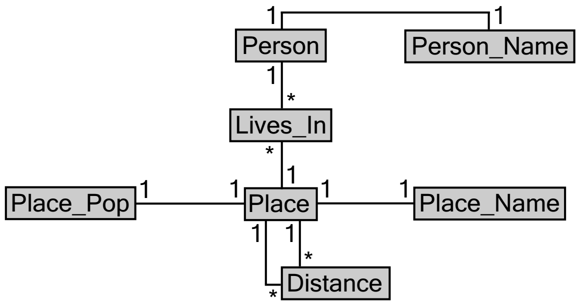

Fig. 1. An entity-relationship diagram of our relational model

- $n \in \mathbb{N} \setminus \{0\}$ is the arity of the relation.
- $R \subseteq Dom^n$ is an n-ary relation on Dom, i.e. R is a set of tuples of the form $(d_1, d_2, \ldots, d_n)$ where each $d_i \in Dom$.
- $p \in \{1, 2, \ldots, n\}$ identifies a primary key, which is unique within a relation, i.e. if $r' \in R$ is of the form $r' = (r'_1, r'_2, \ldots, r'_n)$ and $r'' \in R$ is of the form $r'' = (r''_1, r''_2, \ldots, r''_n)$, then $r'_p = r''_p$ implies that $r' = r''$.

V. The Syntactic Model: A Context-Free Language

Our query language is defined by a context-free grammar. As a point of departure note that we want the following expressions (S) to be in our language:

```
S → Carol lives in a city near SF,
S → Carol lives in the large city near SF,
S → Carol lives in a very small city near SF,
S → Frank lives in San Francisco.
```

We observe that each of these sentences contains a verb (V).

```
V → lives,
```

and some noun-phrases (NP).

```
NP → Carol,
NP → San Francisco,
NP → a city near San Francisco.
```

We can now redefine S by stating that S is of the form

```
S → NP V NP.
```

We observe that our new definition of S now contains a number of expressions that were not in our original definition of S such as `Frank lives in a very small city near San Francisco`. The fact that our new definition now generalizes to other expressions that perfectly match our intuition of what can be a query indicates that we are on the right track. Unfortunately it will also contain a number of expressions that were not in our original definition of S for a good reason, such as *`Frank lives Frank`. However, as we assume that we will use this grammar only for analyzing sentences that are well-formed sentences of English, we can neglect this for the moment.

We could proceed with this kind of analysis, until we arrive at the grammar given in the left column of Figure 2. At this point we have to stress the fact that, from a linguistic point of view, this grammar is far too simplistic to actually describe a substantial fragment of English. However it serves demonstrative purposes quite well, as it is not entirely unreasonable from a linguistic point of view and its workings are readily accessible.

Definition 4: A context-free grammar $G = (\mathtt{V}, \mathtt{T}, P, \mathtt{S})$ consists of

- A finite non-empty set $\mathtt{V}$ of *non-terminal* symbols.
- A finite non-empty set $\mathtt{T}$ of *terminal* symbols.
- A set P of production rules, each of which is of the form $\mathtt{A} \rightarrow \mathtt{a_1 a_2 \dots a_n}$ where $\mathtt{A} \in \mathtt{V}$ is a grammar variable, and each $\mathtt{a}_i$ is a symbol, either terminal or non-terminal, i.e. $\mathtt{a}_i \in (\mathtt{V} \cup \mathtt{T})$.
- A *start-symbol* $\mathtt{S} \in \mathtt{V}$.

Definition 5: Fix a grammar $G = (\mathtt{V}, \mathtt{T}, P, \mathtt{S})$. We say that a non-terminal symbol $\mathtt{X} \in \mathtt{V}$ yields a string $\alpha = \mathtt{t_1 t_2 \dots t_n}$ of terminals $\mathtt{t}_i \in T$ if and only if $\mathtt{X} \rightarrow \mathtt{x_1 x_2 \dots x_m}$ is a production in P where for each $\mathtt{x}_i$

- $\mathtt{x}_i \in \mathtt{V}$ and $\mathtt{x}_i$ yields the string of terminals α_i, or
- $\mathtt{x}_i \in T$, in which case we take α_i to be $\mathtt{x}_i$

and α is the concatenation $\alpha_1 \cdot \alpha_2 \cdot \dots \cdot \alpha_m$.

VI. The Logical Tool: Fuzzy Sets and Relations

Assuming that readers are already familiar with the basics of fuzzy logic, we will only quickly go through some relevant definitions to straighten out the notation. Throughout the paper, let X denote an arbitrary non-empty set. For convenience we will not explicitly distinguish between fuzzy sets and their corresponding membership functions (i.e. $X \rightarrow [0,1]$ mappings). Consequently uppercase letters will be used for both synonymously. Fuzzy relations (i.e. $X_1 \times \dots \times X_n \rightarrow [0,1]$ mappings) are handled in the same way. As usual, we denote the set of all fuzzy sets on X with $\mathcal{F}(X)$.

Definition 6: [28] A function $T : [0,1] \times [0,1] \mapsto [0,1]$ is a triangular norm, if and only if it satisfies:

$$T(x,y) = T(y,x) \qquad \text{(commutativity)} \quad (1)$$

$$T(x, T(y,z)) = T(T(x,y),z) \qquad \text{(associativity)} \quad (2)$$

$$x \leq y \Rightarrow T(x,z) \leq T(y,z) \qquad \text{(non-decreasingness)} \quad (3)$$

$$T(x,1) = x \qquad \text{(neutral element)} \quad (4)$$

Definition 7: A fuzzy set $H \in \mathcal{F}(X)$ is the intersection of two fuzzy sets $F \in \mathcal{F}(X)$ and $G \in \mathcal{F}(X)$ with respect to a triangular norm T, denoted $H = F \cap_T G$, if and only if $H(x) = T(F(x), G(x))$.

Definition 8: If R is an n-ary fuzzy relation on $X_1 \times X_2 \times \dots \times X_n$, and A is a fuzzy subset of X_1, then the image of A with respect to R, denoted $R(A)$, is given by the characteristic function

$$R(A)(x_2, x_3, \dots, x_n) = \sup_{x_1 \in X_1} \{T(A(x), R(x_1, x_2, \dots, x_n)\}$$

for a particular choice T of a triangular norm.

Intuitively, the image $R(A)$ of A with respect to an n-ary relation R reduces the arity of R by binding one (in our definition the first) element in the n-tuple by an existential quantifier. Thus, $(x_2, \dots, x_n) \in R(A)$ if and only if $x_2, \dots, x_n$ are R-related to some $x_1 \in A$.

VII. The Linguistic Tool: Syntax-Driven Semantic Analysis

Syntax-driven semantic analysis is the predominant approach to semantic analysis, widely accepted throughout the communities of computational linguistics and compiler construction. Its fundamentals can be traced back to [29] in the domain of natural languages, and [30] in the domain of programming languages. At its heart lies the assumption of a homomorphism between the models capturing a language's syntax and its semantics. More specifically, we could think of any grammatical production rule p as serving two functions: a syntactic, and a semantic one.

Definition 9: A semantic context-free grammar $G = (\mathtt{V}, \mathtt{T}, P, \mathtt{S}, \mathsf{Dom})$ consists of

- A finite non-empty set $\mathtt{V}$ of *non-terminal* symbols.
- A finite non-empty set $\mathtt{T}$ of *terminal* symbols.
- A *start-symbol* $\mathtt{S} \in \mathtt{V}$.
- A finite non-empty set Dom establishing the semantic *domain*.
- A set P of production rules, each of which is of the form $(\mathtt{A} \rightarrow \mathtt{a_1 a_2 \dots a_n}, \mathsf{eval})$ where $\mathtt{A} \in \mathtt{V}$ is a grammar variable, and each $\mathtt{a}_i$ is a symbol, either terminal or non-terminal, i.e. $\mathtt{a}_i \in (\mathtt{V} \cup \mathtt{T})$. Let u be the number of symbols in $\mathtt{a_1 a_2 \dots a_n}$ that are non-terminal, so that $n - u$ is the number of symbols that are terminal. Then $\mathsf{eval} : \mathsf{Dom}^u \mapsto \mathsf{Dom}$ is a mapping from Dom^u to Dom. In the special case where $u = 0$, eval is taken to be a constant value $\mathsf{eval} \in \mathsf{Dom}$.

In section VIII we will use this definition to construct a semantic context-free grammar that deals with fuzzy relational semantics.

Definition 10: Fix a semantic context-free grammar $G = (\mathtt{V}, \mathtt{T}, P, \mathtt{S}, \mathsf{Dom})$. We say that a non-terminal symbol $\mathtt{X} \in \mathtt{V}$ assigns a string $\alpha = \mathtt{t_1 t_2 \dots t_n}$ of terminals $\mathtt{t}_i \in T$ the meaning $\mathsf{X} \in \mathsf{Dom}$ in G if and only if $(\mathtt{X} \rightarrow \mathtt{x_1 x_2 \dots x_m}, \mathsf{eval})$ is a production in P where for each $\mathtt{x}_i$

- $\mathtt{x}_i \in \mathtt{V}$ and $\mathtt{x}_i$ assigns the string of terminals α_i the meaning x_i, or
- $\mathtt{x}_i \in T$, in which case we take α_i to be $\mathtt{x}_i$

and α is the concatenation $\alpha_1 \cdot \alpha_2 \cdot \dots \cdot \alpha_m$ and $\mathsf{X} = \mathsf{eval}(\mathsf{x}'_1, \mathsf{x}'_2, \dots, \mathsf{x}'_n)$, where $\langle \mathsf{x}'_1, \mathsf{x}'_2, \dots, \mathsf{x}'_u \rangle$ is the sequence $\langle \mathsf{x}_1, \mathsf{x}_2, \dots, \mathsf{x}_n \rangle$ with all the elements removed that represent meanings of non-terminals. In the special case where $u = 0$, $\mathsf{eval}()$ is taken to evaluate to a constant value $\mathsf{eval} \in \mathsf{Dom}$, as defined in the production rule.

In section IX we will use this notion of semantic composition to derive a fuzzy set capturing the meaning of `Carol lives in a small city near San Francisco` using our semantic context-free grammar.

VIII. The Syntax/Semantics Interface: Making Ends Meet

Figure 2 gives semantic rules for all syntactic productions of our example grammar to capture their fuzzy semantics with respect to our example data model. The basic idea is that we can view the meaning of any phrase or word in this grammar as a fuzzy relation, and that we can determine such a fuzzy relation for each phrase from fuzzy relations describing its sub-phrases or from data we have in our database.

In the following sections we will go through the most important rules of the resulting semantic grammar. Applying the technique of syntax-driven semantic analysis to this grammar, we can then derive fuzzy sets resembling the meanings of any grammatically correct expression.

A. The Semantics of Nominals

Our definition of the meanings of nominals relies on our database containing data about the names of entities we wish to refer to by nominals. The relation $\mathsf{Person\_Name}$, for example, associates a name to a person. We can then describe the meaning of any nominal by means of a set of entities x, such that the $\mathsf{Nom}(x)$ condition is fulfilled. Recall that $\mathsf{Nom}(x)$ is the characteristic function of a fuzzy set, in this case (rules 16-18), of all entities x in our domain that are called `Carol`, `Frank`, or `San Francisco`.

Note that, in our first naive approach, we have assigned crisp meanings to all nominals. Therefore, in our grammar, it makes sense to state that someone's name is *Carol*, but it does not make sense to state that someone's name is *Carol* to a degree of 0.7.

B. The Semantics of Nouns

As opposed to nominals, which pick out specific entities from the domain by referring to them by name, nouns can be abstractions. Here we chose the simplistic approach of, again, constructing a crisp set of all places in our database, to represent the set of all entities referred to as `city` (rule 15). In a less simplified treatment this might, however, already be a candidate for a fuzzy concept, since one might want to employ certain criteria to decide whether something is a city, possibly based on the density of population and urban infrastructure to distinguish a *city* from, say, a *town* or a *metropolis*.

C. The Semantics of Adjectives

As our simple example grammar follows the approach of intersective semantics the meaning of an adjective like `small` will be the set of all small things, in much the same way as the meaning of the noun phrase `small thing` would be the set of all small things.

The adjective `small` serves as a good example for a fuzzy concept. What does it mean, as in this particular domain for a city, to be `small`? A fuzzy set could easily be set up that takes into account several measures of "smallness" such as population, area or density of infrastructure, etc. In our simple approach we only use population as a measurement of the size of a city and define a fuzzy decision boundary on this measure. Obviously smallness is a non-increasing function, and we chose to use a piecewise linear membership function as an additional simplification.

Let x be a city with population p. Then rule 14 assigns a degree of fulfillment $\mathsf{Adj}(x) = 1.0$ to all cities x whose population $p \leq 10000$, a degree of fulfillment 0.0 to all cities whose population $p \geq 20000$. For cities whose population is between 10000 and 20000 we interpolate linearly between those points.

D. The Semantics of Adjectival Phrases

In our sample grammar, adjectival phrases can consist of nothing but an adjective. In this case it is quite straightforward to assume that the meaning of the adjectival phrase will be exactly the same as for the only adjective it contains, which is what rule 9 does.

To exemplify the case where the phrase consists of an adverb followed by an adjectival phrase, we chose *very* which serves as a classic example in the fuzzy logic literature of what is called a linguistic hedge there. Rule 10 implements the most simplistic technique to construct the meaning for a fuzzy set resembling `very AP` from the meaning of a fuzzy set resembling `AP`, which is to square the degrees of fulfillment. We stick with this solution once proposed by Zadeh for its simplicity. However, the reader should note that a much more adequate treatment of the famous hedges is given by [31], [32].

E. The Semantics of Prepositional Phrases

In consequence to our usage of intersective semantics, the meaning of a prepositional phrase like `in San Francisco` will be the set of all things that are `in San Francisco`, and thus the same as the noun phrases `thing in San Francisco` or `San Francisco` on its own (this is implemented in rule 11).

Things get more interesting considering the preposition `near`. The approach we chose was to use geographic distance as a measurement of how close two cities are to each other, and to define a fuzzy decision boundary on this measure as before.

Let city x be a distance d away from city y. Then rule 12 assigns a degree of fulfillment $\mathsf{PP}(x) = 1.0$ to all cities x whose geographic distance d from y satisfies $d \leq 20km$, and a degree of fulfillment 0.0 to all cities with $d \geq 50km$. Again we interpolate linearly between those points.

F. The Semantics of Noun Phrases

Our grammar allows for noun phrases that either rewrite to a nominal (in which case the meaning of the nominal is simply preserved in the noun phrase as defined in rule 4) or to a determiner followed by a category we called `N'`. Here we take the simplistic approach that a determiner doesn't contribute to the meaning of a noun phrase, and pass up the meaning of the `N'` to the noun phrase (rule 5). An `N'`, in turn, can rewrite to a noun (in which case, again, we simply pass up the noun's meaning in rule 6), or be pre-modified by an

	syntax	semantics
(1)	S → NP VP	$\mathsf{S}(x) := \sup_y\{T\big(\mathsf{NP}(y), \mathsf{VP}(x,y)\big)\}$
(2)	VP → V PP	$\mathsf{VP}(x,\lambda_y) := \sup_z\{T\big(\mathsf{V}(x,\lambda_y,z), \mathsf{PP}(z)\big)\}$
(3)	VP → V NP	$\mathsf{VP}(x,\lambda_y) := \sup_z\{T\big(\mathsf{V}(x,\lambda_y,z), \mathsf{NP}(z)\big)\}$
(4)	NP → Nom	$\mathsf{NP}(x) := \mathsf{Nom}(x)$
(5)	NP → Det N'	$\mathsf{NP}(x) := \mathsf{N'}(x)$
(6)	N' → N	$\mathsf{N'}(x) := \mathsf{N}(x)$
(7)	N' → AP N	$\mathsf{N'}(x) := T\big(\mathsf{AP}(x), \mathsf{N}(x)\big)$
(8)	N' → N' PP	$\mathsf{N'}(x) := T\big(\mathsf{N'}(x), \mathsf{PP}(x)\big)$
(9)	AP → Adj	$\mathsf{AP}(x) := \mathsf{Adj}(x)$
(10)	AP → very AP	$\mathsf{AP}(x) := \big(\mathsf{AP}(x)\big)^2$
(11)	PP → in NP	$\mathsf{PP}(x) := \mathsf{NP}(x)$
(12)	PP → near NP	$\mathsf{PP}(x) := \sup_y\{T\big(\mathsf{NP}(y), \max(\min(\frac{50km-d}{50km-20km},1),0\big) \mid \mathsf{Place\_Distance}(x,y,d)\}$
(13)	V → lives	$\mathsf{V}(x,\lambda_y,\lambda_z) := 1.0$ if $\mathsf{Lives\_In}(x,\lambda_y,\lambda_z)$, 0.0 otherwise
(14)	Adj → small	$\mathsf{Adj}(x) := \max\big(\min(\frac{20000-p}{20000-10000},1),0\big) \mid \mathsf{Place\_Population}(x,p)$
(15)	N → city	$\mathsf{N}(x) := 1.0$ if $\mathsf{Place}(x)$, 0.0 otherwise
(16)	Nom → Carol	$\mathsf{Nom}(x) := 1.0$ if $\mathsf{Person\_Name}(x, \mathsf{carol})$, 0.0 otherwise
(17)	Nom → Frank	$\mathsf{Nom}(x) := 1.0$ if $\mathsf{Person\_Name}(x, \mathsf{frank})$, 0.0 otherwise
(18)	Nom → San Fr.	$\mathsf{Nom}(x) := 1.0$ if $\mathsf{Place\_Name}(x, \mathsf{san\ francisco})$, 0.0 otherwise
(19)	Det → a	$\mathsf{Det}(x) := 0$

Fig. 2. Our Example Semantic Grammar

adjectival phrase (rule 7) or post-modified by a prepositional phrase (rule 8).

As we have explained before, intersective semantics treats any kind of modification as an intersection. Therefore the meaning of a noun phrase like `very small city` is simply the set AP of all things that are `very small` intersected with the set N of all things that are cities, and the meaning of a noun phrase like `a city near San Francisco` is simply the set of all things that are a `city` and the set of all things that are `near San Francisco`.

We have explained before why we can use triangular norms to capture intersections of fuzzy constraints in a straightforward and intuitive way, and this is exactly what rules 7–8 make use of.

G. The Semantics of Verbs

So far we have only dealt with sets of entities. We have represented the semantics of phrases headed by nouns, adjectives and prepositions by means of fuzzy sets and combined them by means of intersection. In order to capture the semantics of verbs, we will need to make use of fuzzy relations. For example the meaning of `likes` in our grammar, would be the relation $\mathsf{V}(x, y, z)$ that holds between three elements from our domain x, y, z if, and only if, x is an eventuality involving some person y liking some person z. (This is what rule 13 does for the verb `lives`).

Here we have, again, assumed that the eventualities that these verbs denote are not actually fuzzy concepts. One can, however, imagine situations in which verbs refer to fuzzy concepts. For example if we had a data model that contains facts about moving people, we might use the velocity with which people are moving to determine whether someone `strolls`, `walks`, or `runs`.

H. The Semantics of Verb Phrases

Now that we know what it means for some eventuality x to involve some person λ_y liking some other person λ_z (i.e. $\mathsf{V}(x, \lambda_y, \lambda_z)$), what does it mean for x involve some person λ_y to like Carol? Clearly this can be the case if, and only if, *there exists* some person z such that x involves λ_y liking z (i.e. $\mathsf{V}(x, \lambda_y, z)$) *and* z refers to something described by `Carol` (i.e. $\mathsf{NP}(z)$).

As mentioned earlier, we can translate an existential quantifier to the fuzzy domain, by means of a supremum, and the conjunction by means of a triangular norm. Thus the meaning VP of a `VP` given the meaning V of a `V` and the meaning NP of a `NP` will amount to the fuzzy image V(NP) of NP with respect to V (rule 3) The meaning of verb phrases consisting of a verb and a prepositional phrase, can be defined by V(PP) in analogy (rule 2).

I. The Semantics of Sentences

Now that we know what it means for some eventuality x to involve some person λ_y liking Carol (i.e. $\mathsf{VP}(x, \lambda_y)$), what does it mean for x to refer to the eventuality described by a full sentence like `Frank likes Carol`? In analogy to our definition of the semantics of verb phrases, we go about this by stating that Frank likes Carol if and only if *there exists* some person y such that x is the eventuality of y liking Carol (i.e. $\mathsf{VP}(x, y)$) *and* y refers to something described by the noun phrase `Frank` (i.e. $\mathsf{NP}(y)$).

Thus the meaning S of an $\mathtt{S}$ given the meaning NP of a $\mathtt{NP}$ and the meaning VP of a $\mathtt{VP}$ will amount to the fuzzy image $\mathsf{VP}(\mathsf{NP})$ of NP with respect to VP.

J. Fuzzy Relational Semantics of Context-Free Languages

Definition 11: A context-free grammar with fuzzy relational semantics $G = (\mathtt{V}, \mathtt{T}, P, \mathtt{S}, \mathsf{SDom}, \mathsf{Rel})$ is defined by

- a semantic context-free grammar $(\mathtt{V}, \mathtt{T}, P, \mathtt{S}, \mathit{GDom})$,
- a relational scheme $(\mathit{SDom}, \mathit{Rel})$

where $\mathit{GDom} = \bigcup_i \mathcal{F}(\mathit{SDom}^i)$ over all i between zero and the maximal number of non-terminals that appear in the body of any production rule in P.

Definition 12: Fix a context-free grammar with fuzzy relational semantics $G = (\mathtt{V}, \mathtt{T}, P, \mathtt{S}, \mathsf{SDom}, \mathsf{Rel})$. We say that $\mathtt{X}$ assigns a string α the meaning X in G if and only if $\mathtt{X}$ assigns α the meaning X in $(\mathtt{V}, \mathtt{T}, P, \mathtt{S}, \mathit{GDom})$ with GDom defined by SDom as above.

Definition 13: Fix a context-free grammar with fuzzy relational semantics $G = (\mathtt{V}, \mathtt{T}, P, \mathtt{S}, \mathsf{SDom}, \mathsf{Rel})$. The fuzzy semantics of a string $\alpha = \mathtt{t}_1\mathtt{t}_2 \ldots \mathtt{t}_n$ of terminals $\mathtt{t}_i \in T$ with respect to G is given by the binary relation $\mathsf{S}(x) \leq \mathsf{S}(y)$ on $\mathsf{SDom} \times \mathsf{SDom}$ where $\mathtt{S}$ assigns α the meaning S in G. Note that, since the relation $\leq$ on the unit-interval is a weak ordering, so is the relation $\mathsf{S}(x) \leq \mathsf{S}(y)$ on $\mathsf{SDom} \times \mathsf{SDom}$.

IX. The Carol Example

In order to demonstrate how exactly, one can arrive at a fuzzy set denoting a natural language sentence by fully automated means, we will show how to determine the denotation of `Carol lives in a small city near San Francisco`, by successively computing fuzzy sets denoting the phrases subsumed by the nodes $N_1 ... N_{16}$ in the syntax tree shown in Figure 3. Here we denote fuzzy sets by crisp sets of tuples $(x, \mu(x))$.

- The syntactic token `Carol` was found in the input, by rule 16. By the associated semantic rule, we get. $\mathsf{N}_1 = \{(\mathsf{p}_1, 1)\}$ where p_1 denotes `Carol`.
- Rule 4 produces $\mathsf{N}_2 = \{(\mathsf{p}_1, 1)\}$.
- The syntactic token `lives` was found in the input, by rule 13, which produces the set $\mathsf{N}_3 = \{(((\mathsf{p}_1, \mathsf{c}_{17}), \mathsf{p}_1, \mathsf{c}_{17}), 1), ...\}$, where p_1 lives in c_{17}.
- ...
- The syntactic token `small` was found in the input, by rule 13, which produces the set $\mathsf{N}_5 = \{(\mathsf{c}_{17}, 0.7792), ...,\}$, where c_{17} is the city Half Moon Bay, which has population 12208.
- ...
- Rule 8 is applied to construct a fuzzy set N_{12} denoting `small city near San Francisco` from the fuzzy sets N_8 denoting `small city` and N_{11} denoting `near San Francisco`. This would produce, $\mathsf{N}_{12} = \{(\mathsf{c}_{17}, 0.566), ...\}$, where $\mathsf{N}_{11}(\mathsf{c}_{17}) = 0.779$, $\mathsf{N}_8(\mathsf{c}_{17}) = 0.566$, using T_{min}.
- ...
- Rule 2 is applied to construct a fuzzy set N_{15} denoting `lives in a small city near San Francisco` from the fuzzy sets N_3 denoting `lives` and N_{14} denoting `in a small...` producing $\mathsf{N}_{15} = \{(((\mathsf{p}_1, \mathsf{c}_{17}), \mathsf{p}_1), 0.566), ...\}$. This is due to the fact that we get a supremum of $T\big(\mathsf{V}(x, \lambda_y, z), \mathsf{PP}(z)\big)$ for $z = c_{17}$.
- Rule 1 is applied in the same way to get $\mathsf{N}_{16} = \{((\mathsf{p}_1, \mathsf{c}_{17}), 0.566), ...\}$.

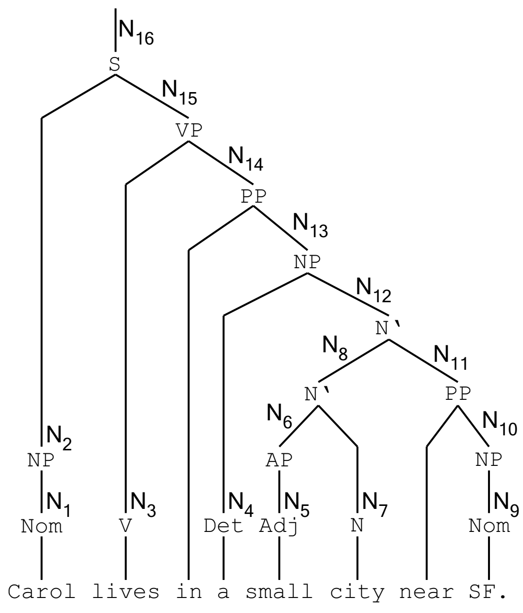

Fig. 3. Derivation for the Carol example

We can then order the records by their degrees of fulfillment which would give us a ranking that denotes the semantics of this sentence.

X. Concluding Remarks & Future Directions

In this paper a grammatical framework was shown that augments context-free production rules with semantic production rules that rely on fuzzy relations as representations of fuzzy natural language concepts. Furthermore, it was shown how the well-known technique of syntax-driven semantic analysis can be used to infer from such a semantically augmented grammar the semantics of a given expression, where the semantics of expressions are taken to be orderings on the possible worlds they describe.

More specifically we were considering the application of natural language query processing to motivate our studies. We assumed that we were given a relational scheme on a certain domain, and showed how we could arrive at an ordering of that domain according to the degree to which its elements satisfy a constraint specified by means of a natural language statement. We considered the specific example of a relational database on a domain of people and places containing information about which cities people live in, populations of cities, and distances between cities, and showed how exactly our technique could be applied to arrive at an ordered sequence of records satisfying the natural language query statement "Carol lives in a small city near San Francisco". In a technical report, we describe a Prolog prototype that does the same [9].

The primary aim of this paper was to demonstrate the overall approach by particular examples and to make a first attempt at defining what exactly a fuzzy semantic grammar may look like, and how it is to be interpreted. However, it raises a number of questions that were not covered herein. Most importantly: Will the approach scale to cover the full complexity of the semantic microdomains that are of interest today, such as typical industrial or scientific knowledge bases, and the full expressive power of natural languages?

On the semantic side this clearly raises questions about computational complexity and about how inference mechanisms can best be incorporated into the system. On the syntactic side this certainly requires more sophisticated linguistic formalisms to be investigated in the context of fuzzy semantics than plain context-free grammars. It seems promising to define the fuzzy semantics of a natural language proposition in terms of a tectogrammatical analysis derived from a system that is more state-of-the-art than the toy grammar we have used.

Considering the work that lies ahead, we can perhaps only claim to have contributed a small first step towards the actual inception of a precisiated natural language as a basis of a computational theory of perceptions. However, we also made clear why, in the light of modern applications, the pursuit of this line of research is probably more rewarding today than ever before.

Acknowledgments

We would like to thank Ann Copestake and Daniel Osherson for reading early manuscripts of this paper, as well as Ted Briscoe for his continued support of the work. The second author gratefully acknowledges partial support by the Austrian Government, the State of Upper Austria, and the Johannes Kepler University Linz in the framework of the K*plus* Competence Center Program. The first author holds a studentship funded by the EPSRC and a bursary funded by the Cambridge European Trust and would like to thank the benefactors who made this possible.

References

[1] L. A. Zadeh, "A new direction in AI: Toward a computational theory of perceptions," *AI Magazine*, vol. 22, pp. 73–84, 2001.

[2] P. Bosc, B. Buckles, F. Petry, and O. Pivert, "Fuzzy databases: Theory and models," in *Fuzzy Sets in Approximate Reasoning and Information Systems*, ser. The Handbooks of Fuzzy Sets, J. Bezdek, D. Dubois, and H. Prade, Eds. Boston: Kluwer Academic Publishers, 1999, vol. 5, pp. 403–468.

[3] P. Bosc and O. Pivert, "SQLf: A relational database language for fuzzy querying," *IEEE Trans. Fuzzy Systems*, vol. 3, pp. 1–17, 1995.

[4] F. E. Petry and P. Bosc, *Fuzzy Databases: Principles and Applications*, ser. International Series in Intelligent Technologies. Boston: Kluwer Academic Publishers, 1996.

[5] A. Motro, "VAGUE: A user interface to relational databases that permits vague queries," *ACM Trans. Off. Inf. Syst.*, vol. 6, no. 3, pp. 187–214, 1988.

[6] J. Küng and J. Palkoska, "VQS—a vague query system prototype," in *Proc. 8th Int. Workshop on Database and Expert Systems Applications*. Los Alamitos, CA: IEEE Computer Society Press, 1997, pp. 614–618.

[7] U. Bodenhofer and J. Küng, "Fuzzy orderings in flexible query answering systems," *Soft Computing*, vol. 8, no. 7, pp. 512–522, 2004.

[8] L. A. Zadeh, "Precisiated natural language," *AI Magazine*, vol. 25, 2004.

[9] R. Bergmair, "Syntax-driven analysis of context-free languages with respect to fuzzy relational semantics," University of Cambridge, Computer Laboratory, Tech. Rep. UCAM-CL-TR-663, Mar. 2006. [Online]. Available: http://www.cl.cam.ac.uk/TechReports/UCAM-CL-TR-663.pdf

[10] D. Sheppard, "The adequacy of everyday quantitative expressions as measurements of qualities," *Brit. J. Psychol.*, vol. 45, pp. 40–50, 1954.

[11] L. A. Zadeh, "Fuzzy sets," *Inf. Control*, vol. 8, pp. 338–353, 1965.

[12] E. T. Lee and L. A. Zadeh, "Note on fuzzy languages," *Inform. Sci.*, vol. 1, pp. 421–434, 1969.

[13] H. Parret, *Discussing Language*, ser. Janua Linguarum Series Maior. Mouton, 1974, vol. 93.

[14] J. A. Goguen, "Concept representation in natural and artificial languages: Axioms, extensions and applications for fuzzy sets," *Int. J. Man-Mach. Stud.*, vol. 6, pp. 513–561, 1974.

[15] ——, "On fuzzy robot planning," in *Fuzzy Sets and their Applications to Cognitive and Decision Processes*, L. A. Zadeh, K. S. Fu, K. Tanaka, and M. Shimura, Eds. Academic Press, 1975, pp. 429–447.

[16] L. A. Zadeh, "PRUF — a meaning representation language for natural languages," *Int. J. Man-Mach. Stud.*, vol. 10, pp. 395–460, 1978.

[17] ——, "Test-score semantics for natural languages and meaning representation via PRUF," in *Empirical Semantics: A collection of new approaches in the field*, B. B. Rieger, Ed. Studienverlag Brockmeyer, 1981, pp. 281–349.

[18] ——, "Test-score semantics for natural languages," in *Proc. 9th Conf. on Computational Linguistics*. Academia Praha, 1982, pp. 425–430.

[19] V. Novák, *The Alternative Mathematical Model of Linguistic Semantics and Pragmatics*, ser. IFSR International Series on Systems Science and Engineering. Plenum Press, 1992, vol. 8.

[20] ——, "Fuzzy logic, fuzzy sets, and natural languages," *Int. J. General Systems*, vol. 20, pp. 83–97, 1991.

[21] L. A. Zadeh, "From computing with numbers to computing with words. from manipulation of measurements to manipulation of perceptions," *IEEE Trans. Circuits Syst. I*, vol. 46, pp. 105–119, 1999.

[22] ——, "From search engines to question-answering systems — the need for new tools," in *Advances in Web Intelligence*, ser. Lecture Notes in Computer Science. Springer, 2003, vol. 2663, pp. 15–17.

[23] ——, "A note on web intelligence, world knowledge, and fuzzy logic," *Data & Knowledge Engineering*, vol. 50, pp. 291–304, 2004.

[24] A. Dvorák and V. Novák, "On the extraction of linguistic knowledge in databases using fuzzy logic," in *Flexible Query Answering Systems: Recent Advances*, H. L. Larsen, J. Kacprzyk, S. Zadrozny, T. Andreasen, and H. Christiansen, Eds. Physica-Verlag, 2000, pp. 445–454.

[25] V. Novák, "Linguistically oriented fuzzy logic controller and its design," *Internat. J. Approx. Reason.*, vol. 12, no. 3–4, pp. 263–277, 1995.

[26] ——, "Evaluating linguistic expressions and their role in the design of the fuzzy control strategy," in *Proc. 2nd Int. ICSC Symp. on Fuzzy Logic and Applications*. ICSC Academic Press, 1997, pp. 89–94.

[27] U. Bodenhofer and P. Bauer, "A formal model of interpretability of linguistic variables," in *Interpretability Issues in Fuzzy Modeling*, ser. Studies in Fuzziness and Soft Computing, J. Casillas, O. Cordón, F. Herrera, and L. Magdalena, Eds. Heidelberg: Springer-Verlag, 2003, vol. 128, pp. 524–545.

[28] E. P. Klement, R. Mesiar, and E. Pap, *Triangular Norms*, ser. Trends in Logic. Dordrecht: Kluwer Academic Publishers, 2000, vol. 8.

[29] R. M. Montague, "The proper treatment of quantification in ordinary English," in *Approaches to Natural Language*, ser. Synthese Library, J. Hintikka, P. Suppes, and J. M. E. Moravcsik, Eds., 1973, vol. 49, pp. 221–242.

[30] D. E. Knuth, "Semantics of context-free languages," *Math. Syst. Theory*, vol. 2, no. 2, pp. 127–145, 1968.

[31] M. De Cock, U. Bodenhofer, and E. E. Kerre, "Modelling linguistic expressions using fuzzy relations," in *Proc. 6th Int. Conf. on Soft Computing*, Iizuka, October 2000, pp. 353–360.

[32] M. De Cock and E. E. Kerre, "A context-based approach to linguistic hedges," *Int. J. Appl. Math. Comput. Sci.*, vol. 12, no. 3, pp. 371–382, 2002.