Mendel, J. M., "An Architecture for Making Judgments Using Computing With Words," *Int. J. Appl. Math. Comput. Sci.*, vol. 12, No. 3, pp. 325-335, 2002.

An Architecture for Making Judgments Using Computing with Words[1]

Jerry M. Mendel

Integrated Media Systems Center

University of Southern California

Los Angeles, CA 90089-2564

mendel@sipi.usc.edu

Telephone: (213) 740-4445 Fax: (213) 740-4651

Abstract

Our thesis is that computing with words needs to account for the uncertainties associated with the meanings of words, and that these uncertainties require using type-2 fuzzy sets. Doing this leads to a proposed architecture for making *judgments* by means of computing with words—a *perceptual computer*—the Per-C. The Per-C includes an encoder, a type-2 rule-based fuzzy logic system, and a decoder. It lets all human-computer interactions be performed using words. In this paper, a quantitative language is established for the Per-C, and many open issues about the perceptual computer are described.

[1] This paper is an expanded version of Mendel (2001b).

1. Introduction

Zadeh (1996, 1999) coined the phrase “computing with words” [see, also, Wang (2001)]; but our thesis is that *words mean different things to different people* and so there is uncertainty associated with words, which means that fuzzy logic (FL) must somehow use this uncertainty when it computes with words [Mendel (1999, 2001a)]. Type-1 FL handles uncertainties about the meanings of words by using *precise* membership functions (MFs) that the user believes captures the uncertainty of the words. Once the type-1 MFs have been chosen, all uncertainty about the words disappears, because a type-1 MF is totally precise. Because of that, type-1 MFs cannot handle the uncertainties about words. We maintain that computing with words requires using type-2 fuzzy sets[2].

Today, computing with words must still be done using numbers, and, therefore, numeric intervals must be associated with words. An earlier paper [Mendel (1999)] reported on an empirical study that was performed to determine how the scale 0–10 can be covered with words (or phrases). In typical engineering applications of FL, we don’t worry about this, because we choose the number of fuzzy sets that will cover an interval arbitrarily, and then choose the names for these sets just as arbitrarily (e.g., *zero*, *small positive*, *medium positive*, and *large positive*). This works fine for many engineering applications, when rules are extracted from data. However, it is questionable practice when rules are extracted from people. Put another way, machines don’t care about words, but people do.

In Mendel (1999), we first established a vocabulary of 16 candidate words or phrases —*terms*—that we thought would let us cover this interval. Those terms are: *none*, *very little*, *a small amount*, *a little bit*, *a bit*, *some*, *a moderate amount*, *a fair amount*, *a good amount*, *a considerable amount*, *a sizeable amount*, *a large amount*, *a substantial amount*, *a lot*, *an extreme amount*, and *a maximum amount*. We then surveyed students and asked them to provide the end-points for intervals on the scale of 0–10 that they associated with each term. The 16 terms were randomized in the survey and we collected 70 useable surveys. We then computed mean and standard deviation values for the two end-points of each of the 16 term’s interval, and plotted the interval for each term. These results are depicted in Figure 1.

[2] A brief introduction to type-2 fuzzy sets is provided in Appendix A.

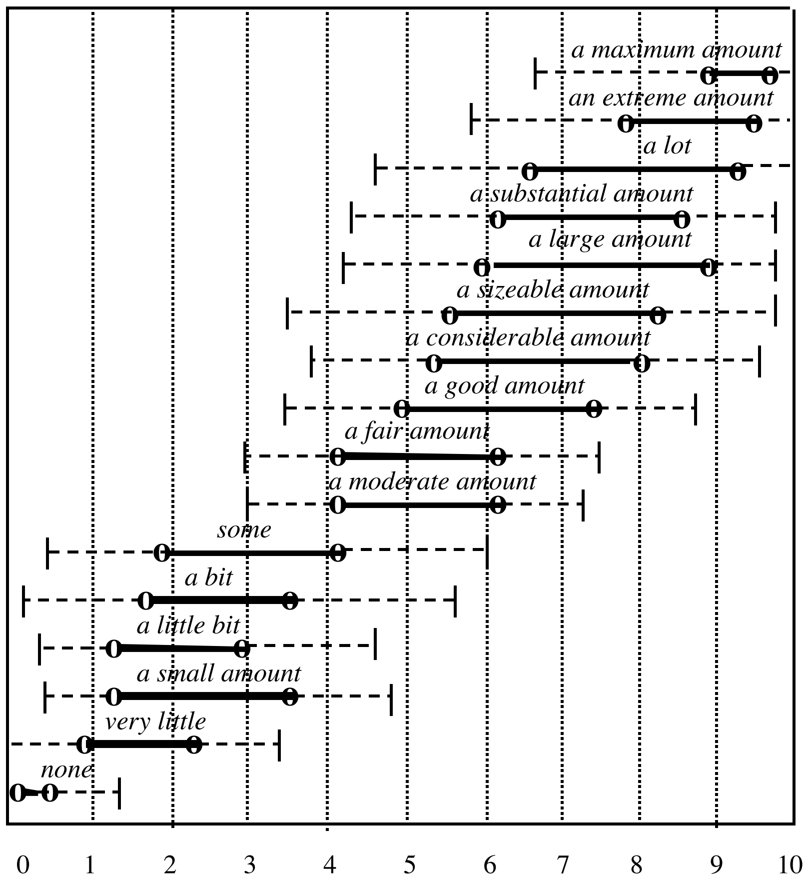

Figure 1: All 16 labels and their intervals and uncertainty bands. Solid lines are drawn between the sample means for the interval end-points and dashed lines are for ±1 standard deviation about each mean end-point.

One of the most striking conclusions drawn from this processed data is: *linguistic uncertainty appears to be useful in that it lets us cover the 0–10 range with a much smaller number of terms than without it*. Figure 2 depicts this for five terms [Figure 2-3 in Mendel (2001a) demonstrates coverage of the 0–10 range for three terms]. Put in the context of firing rules, in a rule-based fuzzy logic system[3] (FLS), *uncertainty can fire rules*. This cannot occur in the framework of a type-1 FLS; but it can occur in the framework of a type-2 FLS. *Uncertainty can therefore be used to control the rule explosion* that is so common in a FLS. If, for example, we ignored uncertainty, and had rules with three antecedents, each of which is described by six terms, it could take 216 rules to completely describe the fuzzy rule base. On the other hand, using three terms for each antecedent requires a rule base with only 27 rules. This is an 87.5% reduction in the size of the rule base.

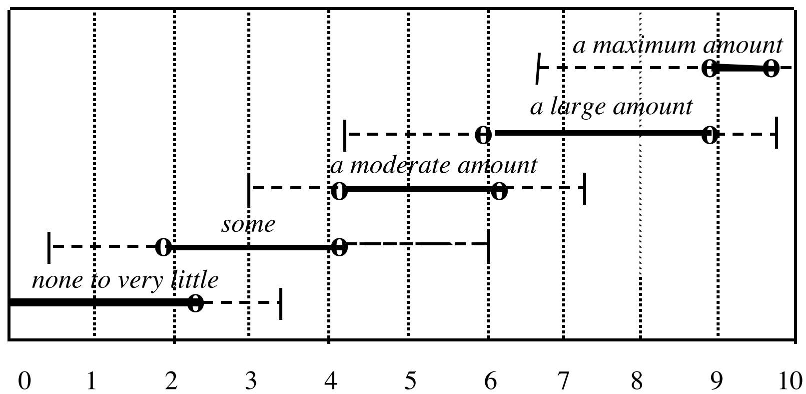

Figure 2: Although five labels cover 0–10, there is not much overlap between some of them. It is when the standard deviation information is used that sufficient overlap is achieved.

Finally, as conjectured in Mendel (2001a) *uncertainty is good in that it lets people make decisions (albeit conservative ones) rapidly*. Perhaps this is why some people can make decisions very quickly and others cannot. The latter may have partitioned their variables into so many fine sets that they get hung up among the resulting enormous number of possibilities. They are the eternal procrastinators. This conjecture is supported by Klir and Wierman (1998) who state: “Uncertainty has a pivotal role in any efforts to maximize the usefulness of systems models.”

Based on these preliminary ideas, in the rest of this paper a specific architecture is proposed for making judgments by computing with words (by *judgment* we mean an assessment of the *level* of a variable of interest). Such a computer will be called a *Perceptual Computer*—**Per-C** for short. We believe that a generic all-purpose methodology can be found for making judgments by computing with words, but that the specific details will be context dependent. So, for example, making judgments by computing with words for diagnostic medicine will have details that are different from those for making judgments by computing with words for accounting. There are many interesting and open issues associated with computing with words, some of which are posed below.

[3] See Appendix B for a high-level description of a FLS.

2. FRAMEWORK FOR A PERCEPTUAL COMPUTER

The architecture for a perceptual computer is depicted in Figure 3. Perceptions (i.e., granulated terms, words) activate the Per-C and are also output by the Per-C; so, it is possible for a human to interact with the Per-C just using a vocabulary—words. The mapping of words into words occurs within the Per-C and is accomplished using the mathematics of type-2 rule-based FLSs. The humans that interact with the Per-C do not have to be concerned with the mathematical details, although the designer of the Per-C must be.

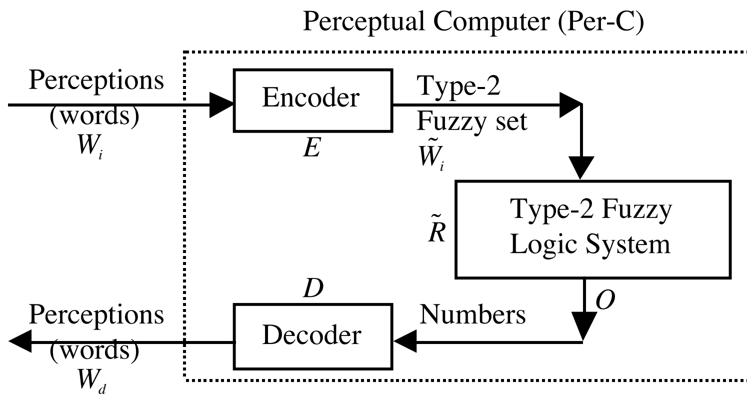

Figure 3: Architecture of a Perceptual Computer (Per-C).

We let W_i denote the *i*th word from a vocabulary, *V*, of N_V words (terms), i.e.,

$$V = \{W_i\}\Big|_{i=1}^{N_V} \tag{1}$$

This vocabulary is context-dependent, and may contain some terms that are common across contexts or that are used as adjectives (e.g. *some*, *a lot of*, *a maximum amount of*). *V* is the vocabulary that a human uses to interface with the Per-C and which the Per-C uses to communicate its findings back to a human (e.g., the 16 terms in Figure 1). Each word in *V* must have a type-2 MF associated with it, which suggests that interval survey information must be obtained for it. New words can be added to *V*, just as a child continues to add words to his or her vocabulary, and interval information about each word in *V* may also change over time.

The *encoder* transforms linguistic perceptions into type-2 fuzzy sets that activate a rule-based type-2 FLS, denoted $\tilde{R}$. We denote the type-2 fuzzy set output of the encoder as $\tilde{W}_i$. Note that the encoder is the same as a fuzzifier; however, it *always* outputs a type-2 fuzzy set, whereas a general-purpose fuzzifier could output singleton, type-1 or type-2 fuzzy sets [e.g., Mendel (2001a)]. This is why we distinguish between an encoder and a fuzzifier.

A type-2 FLS is rule-based, and its rules are context dependent. The rules are IF-THEN rules (which can include a rich variety of rules, e.g., incomplete IF rules, mixed rules, fuzzy statement rules, comparative rules, unless rules, and quantifier rules [Wang (1997)]) whose antecedents and consequent are words that are from antecedent and consequent vocabularies, V_A and V_C, where

$$V_A = \left\{ W_{A_i} \right\}\Big|_{i=1}^{N_A} \subset V \tag{2}$$

and

$$V_C = \left\{ W_{C_i} \right\}\Big|_{i=1}^{N_C} \subset V \tag{3}$$

In this paper, we assume that the words used in V_A and V_C are subsets of the words in *V*, and, that $N_A << N_V$ and $N_C << N_V$ (e.g., *V* might be the 16 terms in Figure 1, whereas V_A and V_C might be the five terms in Figure 2). The latter assumptions mean that there is a *coarser granulation* associated with the antecedents and consequent than in the overall vocabulary. This is one way to control rule explosion. It is important, however, to allow the human to interact with the Per-C using the larger vocabulary so that this interaction is as natural as possible. Because the words in V_A and V_C are subsets of the words in *V*, they will each have a type-2 MF associated with them. We denote the totality of antecedent word type-2 MFs as $\tilde{V}_A$, and the totality of consequent word type-2 MFs as $\tilde{V}_C$.

The output of the type-2 FLS is a number, *O,* which is a result of a sequence of internal operations—inference, type-reduction, and defuzzification—which we shall describe in Section 3. At a very high level, we can describe this output as $O = f[\tilde{W}_i, \tilde{V}_A, \tilde{V}_C]$, where the exact nature of the non-linear function $f[\cdot]$ depends on many specific choices that have to be made within the type-2 FLS (e.g., kind of inference, type-reduction, and defuzzification).

The *decoder*, *D*, maps $O = f[\tilde{W}_i, \tilde{V}_A, \tilde{V}_C]$ into a word W_d, i.e.,

$$W_d = D(O) \subset V \tag{4}$$

How to actually do this is also briefly discussed in Section 3.

3. Some Details of the Perceptual Computer

The main goal of this section is to arrive at an input-output formula for the Figure 3 Per-C. To do this we must provide some details for the elements of the Per-C.

A. Vocabulary

Content experts need to be part of the process of establishing a meaningful vocabulary for specific judgments. This vocabulary should be as large as possible in order to provide a human with as much flexibility as possible. After a behavior of interest is identified for which judgments will be made (e.g., flirtation), the indicators of that behavior must be established (e.g., eye contact, touching, acting witty, smiling, complementing, primping). A small subset of the indicators needs to be established, because rules will be established for that small subset. This can be done by means of an associated survey in which indicators are rank-ordered. Scales then need to be established for each indicator and the behavior of interest[4]. Names and interval information need to be established for each of the indicator's fuzzy sets and behavior of interest's fuzzy sets. Doing this leads to the vocabulary *V*, and the antecedent and consequent vocabularies V_A and V_C. It also leads to the type-2 MFs that are used for the words in all vocabularies. Although the interval surveys described in Section 1 contain uncertainty information about each term, how to use that information to derive an associated type-2 MF is an open issue. Mendel (2001a) advocates transforming the interval uncertainties into a *footprint of uncertainty* (FOU) for each term, but this requires making an a priori choice for the shape of the primary MF (e.g., a Gaussian primary MF whose mean and/or standard deviation are uncertain). Choosing a FOU to model linguistic uncertainty is analogous to choosing a probability density function to model random uncertainty. There is no unique choice for a FOU, just as there is no unique choice for a type-1 MF. Hopefully a Per-C will be robust to the choice of MF shapes, just as, e.g., a type-1 fuzzy logic controller is. Recent results (Mendel, 2001d) about FOUs have demonstrated that a similar looking FOU can be obtained for a triangular primary MF (where there can be uncertainties about all three vertices), a trapezoidal primary MF (where there can be uncertainties about all four vertices), and a Gaussian primary MF (where both the mean and standard deviation are uncertain). It appears, therefore, that granulation of a type-1 fuzzy set to a type-2 fuzzy set reduces the problem of determining the type-1 MFs.

Because the type-2 FLS operates on numbers, scales must be established for each indicator and the behavior of interest[5]. Commonly used scales are 1 through 5, 0 through 5, 0 through 10, etc. The survey described in Section 1 for the 16 terms was performed in a context-free situation. An open issue is whether or not such context-independent results can be applied when the terms are used within a specific context, e.g. if *some*, in a context-independent situation, is located (on a scale 0–10) in the interval [0.5, 6], then are *some touching* and *some primping* located over that same interval? There may also be situations where a natural scale already exists for an indicator (e.g., pressure, temperature). Is the interval associated with, e.g. *low pressure* proportionately the same as the interval associated with *low temperature*? If not, then is there a way to achieve a scale-invariance for words so that context-independent-intervals can be used in context-dependent situations? Humans seem to understand certain terms in a context-free situation, and are able to apply them in context-dependent ones, and also understand terms in a context and are able to adapt them to other contexts; so, we conjecture that it ought to be possible for computers to do likewise.

The FOUs for the words in V are pre-computed using the word surveys described above. Another open issue is how much uncertainty one should associate with the interval end-points. Let σ_L and

[4] For some judgments (e.g., wealth), the indicators and judgments will have natural scales, whereas for others (e.g., flirtation) no natural scales exist. Instead of using scales, it may be possible to use a line of arbitrary length and percentages that are associated with that length.
[5] See footnote 4 for an alternative approach.

σ_R denote the standard deviation for the left and right end-points of a word's interval, respectively, and ρ denote a *fraction of uncertainty*, i.e. $0 \leq \rho \leq 1$; then, when we use $\rho\sigma_L$ or $\rho\sigma_R$ for interval end-point uncertainties, what is ρ? If ρ is chosen too large then all MFs will overlap too much. A theory is needed to guide us in how much uncertainty should be used to characterize the uncertainty of words.

B. Encoder

The encoder transforms a word into a type-2 MF μ_{W_i}. A type-2 MF is three-dimensional. Each element of a FOU has a possibility value—a secondary MF—assigned to it. The union of all possibility values defined over the FOU constitutes the type-2 MF. The FOU for each of the words in *V* is pre-computed, as described in Section 3.A. The remaining issue is what to choose for the secondary MFs. Mendel (2001a, 2001b) advocates using interval sets because then all remaining operations within the type-2 FLS are manageable. Additionally, even for a type-1 FLS there is no one best choice for the shape for a MF, so why should we compound this by trying to choose secondary MFs as arbitrary shapes?

Another very important reason for using interval type-2 fuzzy sets is that all set-theoretic laws (e.g., DeMorgan's, distributive, associative, etc.) are satisfied regardless of which t-norm or t-conorm are used. This is not the case when arbitrary type-2 fuzzy sets are used [see Appendix B in Mendel (2001a)], which could cause serious problems in a rule-based system.

C. Type-2 FLS

As described in Appendix B, a type-2 FLS consists of a fuzzifier and an inference engine followed by type-reduction and defuzzification. The fact that all words are encoded using type-2 fuzzy sets means that the encoder already accomplishes fuzzification.

Rules, which establish the detailed architecture of the FLS, are the heart of any FLS. They remain the same regardless of whether we use type-2 or type-1 fuzzy sets in the inference engine. What changes is the way in which we model the antecedent and consequent fuzzy sets. To begin, a specific architecture must be chosen for the FLS. This will depend in part on how many indicators of a judgment are considered to be significant. More than two indicators cause a problem, because people do not like to answer many questions and have great difficulty in correlating more than two things at a time (e.g., a two-antecedent rule can be interpreted as providing the *correlation* between the two antecedents). So, if more than two indicators are required, then some new architecture(s) will be needed for the type-2 FLS, e.g. a parallel interconnection of sub-advisors, or a hierarchical architecture. To-date, no such architectures have been published for type-2 FLSs, so this is another open issue.

Choices must be made to implement the (compositional) *inference engine*, namely the implication ($\rightarrow$) and t-norm ($\bigstar$) to be used. Although Mamdani implications and product or minimum t-norms are widely used in many engineering applications of a FLS, for the Per-C the kind of implication and t-norm to use are open issues. In fact, any choices for implication or t-norm different from those just mentioned will require new results be developed for the extended sup-star composition of a type-2 FLS, which is yet another open issue. Yager and Filev (1994), for example, propose that Mamdani implication and logic implication ($\sim p \vee q$) are the lower and upper bound for the inference engine, and establish a procedure—*compromise fuzzy reasoning*—in which a linear combination (i.e., a parametric model) of these two extremes is used. Further discussions about the many *choices* that have to be made for the inference engine as compared to rule-related uncertainties are given in Mendel and Wu (2002).

Rules are type-2 compositions, and in Figure 3 are denoted as $\tilde{R}$, i.e.

$$\tilde{R} = \tilde{R}(\tilde{V}_A, \tilde{V}_C) = \tilde{R}(\tilde{V}_A, \tilde{V}_C \big| \rightarrow) \tag{5}$$

Inferencing in a type-2 FLS is done using the extended sup-star composition [Mendel (2001a)] which transforms $\tilde{W}_i$ and $\tilde{R}$ into another type-2 fuzzy set, $\tilde{I}$, and involves a t-norm operation, i.e.,

$$\tilde{I} = \tilde{I}(\tilde{W}_i, \tilde{R}) = \tilde{I}(\tilde{W}_i, \tilde{V}_A, \tilde{V}_C) = \tilde{I}(\tilde{W}_i, \tilde{V}_A, \tilde{V}_C \big| \rightarrow, \bigstar) \tag{6}$$

Inferencing is very straightforward for an interval type-2 FLS[6].

Following the inference engine is *type-reduction* (TR)[7]. TR maps a type-2 fuzzy set into a type-1 fuzzy set. Just as there are many defuzzification methods for a type-1 FLS, there are many TR methods for a type-2 FLS. Whether or not one TR method is best for a Per-C is yet another open issue. Whether fired output sets should be combined prior to or as part of TR is yet another open issue (as it is for defuzzification). Without going into the details of how to perform TR, we can simply view it as a non-linear operator on $\tilde{I}$, i.e., for an interval type-2 FLS,

$$TR = f_{TR}\left[\tilde{I}(\tilde{W}_i, \tilde{V}_A, \tilde{V}_C \,\middle|\, \rightarrow, \bigstar)\right] = [y_l, y_r] \tag{7}$$

where $[y_l, y_r]$ is an interval type-1 fuzzy set, and

$$y_l = y_l\left[\tilde{I}(\tilde{W}_i, \tilde{V}_A, \tilde{V}_C \,\middle|\, \rightarrow, \bigstar, f_{TR})\right] \tag{8}$$

and

$$y_r = y_r\left[\tilde{I}(\tilde{W}_i, \tilde{V}_A, \tilde{V}_C \,\middle|\, \rightarrow, \bigstar, f_{TR})\right] \tag{9}$$

The final operation within the type-2 FLS is *defuzzification*, which is the mapping of the type-1 fuzzy set *TR* into a number, *O*, a type-0 fuzzy set. The obvious choice for defuzzification is the center of gravity (COG) of type-1 set *TR*, i.e.

$$\begin{aligned} O = COG(TR) &= COG\left\{f_{TR}\left[\tilde{I}(\tilde{W}_i, \tilde{V}_A, \tilde{V}_C \,\middle|\, \rightarrow, \bigstar)\right]\right\} \\ &= \frac{1}{2} y_l\left[\tilde{I}(\tilde{W}_i, \tilde{V}_A, \tilde{V}_C \,\middle|\, \rightarrow, \bigstar, f_{TR})\right] + \frac{1}{2} y_r\left[\tilde{I}(\tilde{W}_i, \tilde{V}_A, \tilde{V}_C \,\middle|\, \rightarrow, \bigstar, f_{TR})\right] \end{aligned} \tag{10}$$

[6] See Appendix D for inferencing results for singleton interval type-2 FLSs. More general results for non-singleton interval type-2 FLSs (in which measurements that activate the FLS are modeled either as type-1 or type-2 fuzzy sets) can be found in Mendel (2001a, Chapters 11 and 12).
[7] See Appendix C for a brief discussion about type-reduction methods.

Equation (10) is the input-output formula for the type-2 FLS within the Per-C.

D. Decoder

The decoder, *D*, operates on *O* to provide a word W_d as in (4), i.e.,

$$\begin{aligned} W_d &= D(O) = D[COG(TR)] \\ &= D\Big[COG\Big\{f_{TR}\Big[\tilde{I}(\tilde{W}_i, \tilde{V}_A, \tilde{V}_C \Big| \rightarrow, \bigstar)\Big]\Big\}\Big] \subset V \end{aligned} \tag{11}$$

Observe that the output of the decoder is a word from the vocabulary *V*.

How to go from MF numerical values for a variable to a linguistic description of that variable for type-1 fuzzy sets is well known; however, how to do this for type-2 fuzzy sets is not so well known.

Consider, for example, the type-1 situation depicted in Figure 4 at $x = x'$. This value of *x* only generates a non-zero membership value in the fuzzy set F_4 = *Medium Positive*; hence, $x = x'$ can be described linguistically, without any ambiguity, as "Medium Positive." The situation at $x = x''$ is different, because this value of *x* generates a non-zero membership value in two fuzzy sets F_4 = *Medium Positive* and F_5 = *Very Positive*. It would be very awkward to speak of x'' as "being Medium Positive to degree $\mu_{F_4}(x'')$ and Very Positive to degree $\mu_{F_5}(x'')$." People just don't communicate this way. Instead, we usually compare $\mu_{F_4}(x'')$ and $\mu_{F_5}(x'')$ to see which is larger[8], and then assign x'' to the set associated with the larger value; hence, in this example, we would speak of x'' as "being Medium Positive."

We can formally describe what we have just explained, as follows. Given *P* fuzzy sets F_i with MFs $\mu_{F_i}(x)$, i = 1, …, *P*. When $x = x'$, evaluate all *P* MFs at this point, and then compute $\max\left[\mu_{F_1}(x'), \mu_{F_2}(x'), \ldots, \mu_{F_P}(x')\right] \equiv \mu_{F_m}(x')$. Let $L(x')$ denote the linguistic label associated with x'. Then, $L(x') \equiv F_m$, i.e.,

[8] There is a literature that deals with other ways for doing this; however, all other ways are more complicated than the present way, and usually rely on the availability of "truth" data. Such data is usually not available when computing with words.

$$L(x') = \arg\ \max_{\forall F_i} \left[\mu_{F_1}(x'), \mu_{F_2}(x'), \ldots, \mu_{F_P}(x') \right] \tag{12}$$

Consider the type-2 situation depicted in Figure 5 at $x = x'$. This value of x only generates a non-zero membership in the type-2 fuzzy set $\tilde{W}_4$ = *Medium Positive*; hence, $x = x'$ can be described linguistically, without any ambiguity, as "Medium Positive." The situation at $x = x''$ is quite different, because this value of x generates a range of non-zero secondary MF values in the two type-2 fuzzy sets $\tilde{W}_4$ = *Medium Positive* and $\tilde{W}_5$ = *Very Positive*. It would be extraordinarily difficult to communicate this linguistically. An approach (not necessarily an optimal one, but one that generalizes its type-1 counterpart to type-2 fuzzy sets), which we describe next, is to first convert the intersection of the vertical line at $x = x''$ with the FOUs into a collection of numbers, after which we can choose the linguistic label at $x = x''$ using the algorithm described below.

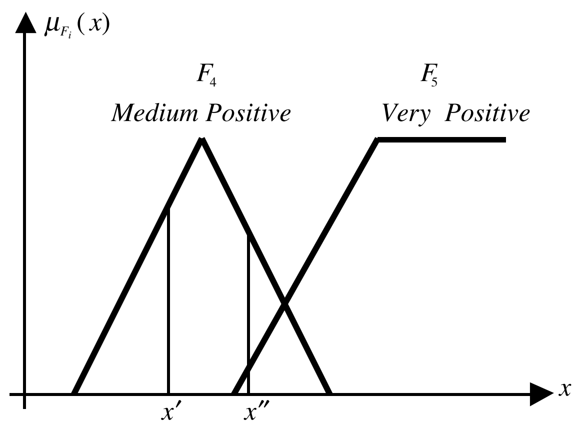

Figure 4: Returning to a linguistic label for type-1 fuzzy sets.

The type-2 MFs $\mu_{W_4}(x,u)$ and $\mu_{W_5}(x,u)$ are characterized by their (shaded) FOUs, $FOU(\tilde{W}_4)$ and $FOU(\tilde{W}_5)$, respectively. The upper and lower MFs for $\tilde{W}_4$ are $\bar{\mu}_{\tilde{W}_4}(x)$ and $\underline{\mu}_{\tilde{W}_4}(x)$, whereas the comparable quantities for $\tilde{W}_5$ are $\bar{\mu}_{\tilde{W}_5}(x)$ and $\underline{\mu}_{\tilde{W}_5}(x)$. Consider, for example, the vertical line at $x = x''$, and its intersections with the FOU for $\tilde{W}_4$ (see Figure 5). Associated with the interval $\left[\underline{\mu}_{\tilde{W}_4}(x''), \bar{\mu}_{\tilde{W}_4}(x'')\right]$ is the secondary MF $f_{x''}(u)$, $u \in \left[\underline{\mu}_{\tilde{W}_4}(x''), \bar{\mu}_{\tilde{W}_4}(x'')\right]$. Let the center of gravity of

$f_{x''}(u)$, $\forall u \in \left[\underline{\mu}_{\tilde{W}_4}(x''), \bar{\mu}_{\tilde{W}_4}(x'')\right]$ be denoted $f_{x''}^{cg}(\tilde{W}_4)$. In a similar manner, we can compute $f_{x''}^{cg}(\tilde{W}_5)$. We can then compare $f_{x''}^{cg}(\tilde{W}_4)$ and $f_{x''}^{cg}(\tilde{W}_5)$. If $f_{x''}^{cg}(\tilde{W}_4) > f_{x''}^{cg}(\tilde{W}_5)$, then we would speak of x'' as "being Medium Positive"; otherwise we would speak of x'' as "being Very Positive."

We can formally describe what we have just explained as follows. Given N_V type-2 fuzzy sets $\tilde{W}_i$ with MFs $\mu_{\tilde{W}_i}(x,u)$, $i = 1, \ldots, N_V$. These MFs are characterized by their FOUs, $FOU(\tilde{W}_i)$, whose upper and lower MFs are $\bar{\mu}_{\tilde{W}_i}(x)$ and $\underline{\mu}_{\tilde{W}_i}(x)$ ($i = 1, \ldots, N_V$), respectively. Consider an arbitrary value of x, say $x = x'$, and compute $\max\left[f_{x'}^{cg}(\tilde{W}_1), f_{x'}^{cg}(\tilde{W}_2), \ldots, f_{x'}^{cg}(\tilde{W}_{N_V})\right] \equiv f_{x'}^{cg}(\tilde{W}_m)$ where $f_{x'}^{cg}(\tilde{W}_i)$ is the center of gravity of the secondary MF $f_{x'}(u), \forall u \in \left[\underline{\mu}_{\tilde{W}_i}(x'), \bar{\mu}_{\tilde{W}_i}(x')\right]$. Let $L(x')$ denote the linguistic label associated with x'. Then, $L(x') \equiv \tilde{W}_m$, i.e.,

$$L(x') = \arg\ \max_{\forall \tilde{W}_i} \left[f_{x'}^{cg}(\tilde{W}_1), f_{x'}^{cg}(\tilde{W}_2), \ldots, f_{x'}^{cg}(\tilde{W}_{N_V})\right] \tag{13}$$

For interval secondary MFs, it is easy to compute $f_{x'}^{cg}(\tilde{W}_i)$, as

$$f_{x'}^{cg}(\tilde{W}_i) = \tfrac{1}{2}\left[\bar{\mu}_{\tilde{W}_i}(x') + \underline{\mu}_{\tilde{W}_i}(x')\right] \tag{13}$$

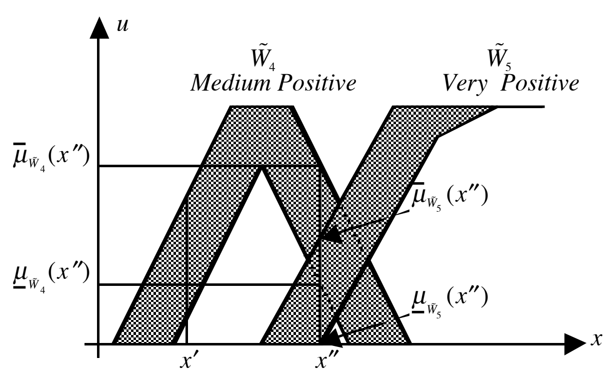

Figure 5: Returning to a linguistic label for type-2 fuzzy sets. The shaded regions are the FOUs for the two type-2 fuzzy sets.

5. CONCLUSIONS

We have presented a specific architecture for making judgments by computing with words—the perceptual computer (Per-C)—and have argued that type-2 fuzzy sets must be used for computing with words. The following issues (including others) must be resolved before the Per-C can be implemented:

1. How should survey interval information about words be used to derive an associated type-2 MF?
2. Can context-independent results be applied when terms are used within a specific context? Is there a way to achieve scale-invariance for words so context-independent intervals can be used in context-dependent situations? Or, is there a way to map from context-independent intervals to context-dependent intervals?
3. How much uncertainty should be associated with the interval end-points. A methodology for choosing or designing the uncertainty factor needs to be developed.
4. What are new architectures for type-2 FLSs that can be used in a Per-C? Can a specific architecture be validated?
5. Which implications and t-norms are most appropriate for a Per-C? What are the associated extended sup-star composition results for them? It seems that, at the very least, we must account for the uncertainties present in all rule-words, including connector words. This can be accomplished by using type-2 fuzzy sets for antecedent and consequent words and parametric operators for connector words. It may then be necessary to account for the uncertainties associated with implication and combining of rules. Whether or not it is necessary to parameterize the enormous numbers of choices that are available for the operator models of implication, union, intersection, complement, t-norm and t-conorm, as in Yager and Filev (1994) remains to be explored. See Wu and Mendel (2001) for more discussions about this.
6. How are fired rule outputs combined by people? The engineering literature on FLSs has no adequate answer to this question. Yager and Filev (1994) introduce the notion of *soft rule aggregation* in which rules are combined using either the SOWA-OR or the SOWA-AND. This approach remains to be examined in the context of type-2 fuzzy sets.
7. Which type-reduction method is best for a Per-C?

8. Is there a best way to return to a linguistic label for a type-2 fuzzy set, i.e., an optimal decoder?

In the future, a computer-architecture may come into being that will let us actually compute with words. In the meantime, if we are to "compute with words," it must be done within the framework of existing computer architectures, all of which compute with numbers. The Per-C lets us make judgments[9] by computing with words using existing computer architectures.

APPENDIX A: Background Knowledge about Type-2 Fuzzy Sets

In this appendix we collect some important definitions about type-2 fuzzy sets. For more details about such fuzzy sets, as well as many examples that illustrate the definitions, see Mendel (2001a).

Definition A-1: A *type-2 fuzzy set*, denoted $\tilde{A}$, is characterized by a (three-dimensional) *type-2 membership function* $\mu_{\tilde{A}}(x,u)$, i.e.,

$$\tilde{A} = \int_{x \in X} \int_{u \in J_x} \mu_{\tilde{A}}(x,u) \,/\, (x,u) \qquad J_x \subseteq [0,1] \tag{A-1}$$

where $\int\int$ denotes union over all admissible x and u, and $0 \le \mu_{\tilde{A}}(x,u) \le 1$. At each fixed value of $x \in X$, J_x is the *primary membership* of x and x is called the *primary variable*. ■

Definition A-2: At each value of x, say $x = x'$, the 2D plane whose axes are u and $\mu_{\tilde{A}}(x',u)$ is called a *vertical slice* of $\mu_{\tilde{A}}(x,u)$. A *secondary membership function* is a vertical slice of $\mu_{\tilde{A}}(x,u)$. It is $\mu_{\tilde{A}}(x = x',u)$ for $x' \in X$ and $\forall u \in J_{x'} \subseteq [0,1]$, i.e.,

$$\mu_{\tilde{A}}(x = x',u) \equiv \mu_{\tilde{A}}(x') = \int_{u \in J_{x'}} f_{x'}(u) / u \qquad J_{x'} \subseteq [0,1] \tag{A-2}$$

in which $0 \le f_{x'}(u) \le 1$. Because $\forall x' \in X$, we drop the prime notation on $\mu_{\tilde{A}}(x')$, and refer to $\mu_{\tilde{A}}(x)$ as a secondary membership function; it is a type-1 fuzzy set, which we also refer to as a *secondary set*. ■

Based on the concept of secondary sets, we can reinterpret a type-2 fuzzy set as the union of all secondary sets, i.e., using (A-2), *we can re-express $\tilde{A}$ in a vertical-slice manner*, as:

$$\tilde{A} = \{(x, \mu_{\tilde{A}}(x)) \mid \forall x \in X\} \tag{A-3}$$

or as

$$\tilde{A} = \int_{x \in X} \mu_{\tilde{A}}(x) \Big/ x = \int_{x \in X} \left[\int_{u \in J_x} f_x(u) / u \right] \Big/ x \qquad J_x \subseteq [0,1] \tag{A-4}$$

Definition A-3: The *domain* of a secondary membership function is called the *primary membership* of *x*. In (A-4), J_x is the primary membership of *x*, where $J_x \subseteq [0,1]$ for $\forall x \in X$. ■

Definition A-4: The *amplitude* of a secondary membership function is called a *secondary grade*. In (A-2) and (A-4), $f_x(u)$ is a secondary grade. ■

Definition A-5: An *interval type-2 fuzzy* set is a type-2 fuzzy set all of whose secondary membership functions are type-1 interval sets, i.e., $f_x(u) = 1, \forall u \in J_x \subseteq [0,1]$, $\forall x \in X$. ■

Interval secondary membership functions reflect a uniform uncertainty at the primary memberships of x, and are the ones most commonly used in type-2 FLSs. Note that an interval set can be represented just by its domain interval, which can be expressed in terms of its left and right end-points as [*l, r*], or by its center and spread as [*c* - *s*, *c* +*s*], where $c = (l + r)/2$ and $s = (r - l)/2$.

[9] We claim nothing else for the Per-C, although it's architecture may also be useful for other kinds of computing with words.

Definition A-6: Assume that each of the secondary membership functions of a type-2 fuzzy set has only one secondary grade that equals 1. A *principal membership function* is the union of all such points at which this occurs, i.e.,

$$\mu_{principal}(x) = \int_{x \in X} u/x \text{ where } f_x(u) = 1 \tag{A-5}$$

and is associated with a type-1 fuzzy set. ■

For interval secondary membership functions, we define the principal membership function as occurring at the union of all primary membership *midpoints*. Note that when all membership function uncertainties disappear, a type-2 membership function reduces to its principal membership function.

Definition A-7: Uncertainty in the primary memberships of a type-2 fuzzy set, $\tilde{A}$, consists of a bounded region that we call the *footprint of uncertainty* (FOU). It is the union of all primary memberships, i.e.,

$$FOU(\tilde{A}) = \bigcup_{x \in X} J_x \quad ■ \tag{A-6}$$

The term FOU is very useful, because it not only focuses our attention on the uncertainties inherent in a specific type-2 membership function, whose shape is a direct consequence of the nature of these uncertainties, but it also provides a very convenient verbal description of the entire domain of support for all the secondary grades of a type-2 membership function.

Definition A-8: Consider a family of type-1 membership functions $\mu_A(x \mid p_1, p_2, ..., p_v)$ where $p_1, p_2, ..., p_v$ are parameters, some or all of which vary over some range of values, i.e., $p_i \in P_i$ ($i = 1$, …, v). A *primary membership function (MF)* is any one of these type-1 membership functions, e.g., $\mu_A(x \mid p_1 = p_{1'}, p_2 = p_{2'}, ..., p_v = p_{v'})$. For short, we use $\mu_A(x)$ to denote a primary membership function. It will be subject to some restrictions on its parameters. The family of all primary membership functions creates a FOU. ■

Two examples of very useful primary MFs are: Gaussian MF with uncertain mean and certain standard deviation, and Gaussian MF with certain mean and uncertain standard deviation.

Definition A-9: An *upper membership function* and a *lower membership function* are two type-1 membership functions that are bounds for the FOU of a type-2 fuzzy set $\tilde{A}$. The upper membership function is associated with the upper bound of $FOU(\tilde{A})$, and is denoted $\overline{\mu}_{\tilde{A}}(x)$, $\forall x \in X$. The lower membership function is associated with the lower bound of $FOU(\tilde{A})$, and is denoted $\underline{\mu}_{\tilde{A}}(x)$, $\forall x \in X$, i.e.,

$$\overline{\mu}_{\tilde{A}}(x) \equiv \overline{FOU(\tilde{A})} \quad \forall x \in X \tag{A-7}$$

and

$$\underline{\mu}_{\tilde{A}}(x) \equiv \underline{FOU(\tilde{A})} \quad \forall x \in X \tag{A-8}$$

Because the domain of a secondary membership function has been constrained in Definition A-1 to be contained in [0, 1], lower and upper membership functions always exist. ■

APPENDIX B. Rule-Based Fuzzy Logic Systems

A rule-based FLS contains four components—rules, fuzzifier, inference engine, and output processor—that are inter-connected, as shown in Figure B-1. Once the rules have been established, a FLS can be viewed as a mapping from inputs to outputs (the solid path in Figure B-1, from “Crisp inputs” to “Crisp outputs”), and this mapping can be expressed quantitatively as $y = f(\mathbf{x})$. This kind of FLS is very widely used in many engineering applications of FL, such as in FL controllers and signal processors, and is also known as a *fuzzy controller*, *fuzzy system*, *fuzzy expert system*, or *fuzzy model.*

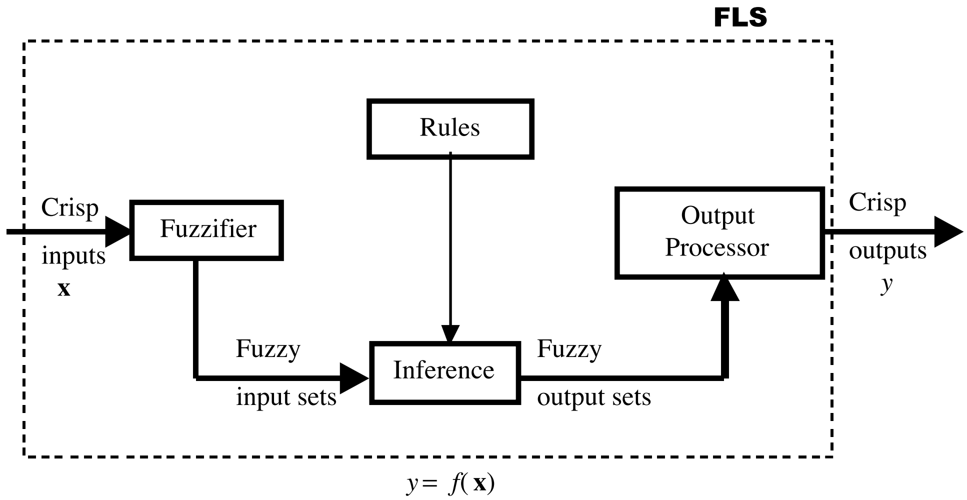

Figure B-1: Fuzzy logic system.

Rules are the heart of a FLS, and may be provided by experts or can be extracted from numerical data. In either case, the rules that we are interested in can be expressed as a collection of IF–THEN statements.

Fuzzy sets are associated with terms that appear in the antecedents or consequents of rules, and with the inputs to and output of the FLS. Membership functions are used to describe these fuzzy sets. Two kinds of fuzzy sets can be used in a FLS, type-1 and type-2. Type-1 fuzzy sets are described by membership functions that are totally certain, whereas type-2 fuzzy sets are described by membership functions that are themselves fuzzy. The latter let us quantify different kinds of uncertainties that can occur in a FLS.

A FLS that is described completely in terms of type-1 fuzzy sets is called a *type-1 FLS*, whereas a FLS that is described using at least one type-2 fuzzy set is called a *type-2 FLS.* A type-2 FLS whose MFs are interval type-2 fuzzy sets is called an *interval type-2 FLS*. In this paper we assume the use of interval type-2 FLSs.

Returning to the Figure B-1 FLS, the fuzzifier maps crisp numbers into fuzzy sets. It is needed to activate rules that are in terms of linguistic variables, which have fuzzy sets associated with them. The inputs to the FLS prior to fuzzification may be certain (e.g., perfect measurements) or uncertain (e.g., noisy measurements). Type-1 or type-2 fuzzy sets can be used to model the latter measurements.

The inference engine of the Figure B-1 FLS maps fuzzy sets into fuzzy sets. It handles the way in which rules are activated and combined. Just as we humans use many different types of inferential procedures to help us understand things or to make decisions, there are many different FL inferential procedures.

In many applications of a FLS, crisp numbers must be obtained at its output. This is accomplished by the output processor, and is known as *defuzzification*. The output processor for a type-1 FLS consists only of a defuzzifier; however, the output processor of a type-2 FLS contains two components: the first maps a type-2 fuzzy set into a type-1 fuzzy set and is called *type-reduction*, and the second performs defuzzification on the latter set. Type-reduction is overviewed in Appendix C.

APPENDIX C: Type-Reduction

The *type-reduced set* provides an interval of uncertainty for the output of a type-2 FLS, in much the same way that a confidence interval provides an interval of uncertainty for a probabilistic system. The more uncertainties that occur in a type-2 FLS, which translate into more uncertainties about its MFs, the larger will be the type-reduced set, and vice-versa.

Five different type-reduction methods are described in Mendel (2001a). Each is inspired by what we do in a type-1 FLS [when we defuzzify the (combined) output of the inference engine using a variety of defuzzification methods that all do some sort of centroid calculation] and are based on computing the *centroid of a type-2 fuzzy set*. Using the Extension Principle, Karnik and Mendel (2001) defined the centroid of a type-2 fuzzy set; it is a type-1 fuzzy set. Computing the centroid of a general type-2 fuzzy set can be very intensive; however, for an interval type-2 fuzzy set, an exact iterative method for computing its centroid has been developed by Karnik and Mendel (2001). This was possible because *the centroid of an interval type-2 fuzzy set is an interval type-1 fuzzy set*, and such sets are completely characterized by their left- and right-end points; hence, computing the centroid of an interval type-2 fuzzy set only requires computing those two end-points.

Center-of-sets, centroid, center-of-sums, and height type-reduction can all be expressed as

$$Y_{TR}(\mathbf{x}) = [y_l, y_r] = \int_{y^1 \in [y_l^1, y_r^1]} \cdots \int_{y^M \in [y_l^M, y_r^M]} \int_{f^1 \in [\underline{f}^1, \overline{f}^1]} \cdots \int_{f^M \in [\underline{f}^M, \overline{f}^M]} 1 \Big/ \frac{\sum_{i=1}^M f^i y^i}{\sum_{i=1}^M f^i} \tag{C-1}$$

For the different type-reduction methods y_l^i, y_r^i, $\underline{f}^i$, $\overline{f}^i$ and M have different meanings, as summarized in Table C-1.

The Karnik-Mendel iterative procedure for computing y_r is:

1. Without loss of generality, assume that the pre-computed y_r^i are arranged in *ascending order*; i.e., $y_r^1 \le y_r^2 \le \cdots \le y_r^M$.
2. Compute y_r as $y_r = \sum_{i=1}^M f_r^i y_r^i \Big/ \sum_{i=1}^M f_r^i$ by initially setting $f_r^i = (\underline{f}^i + \overline{f}^i)/2$ for i = 1, …, M, where $\underline{f}^i$ and $\overline{f}^i$ have been previously computed using the equations given in Appendix D, respectively, and let $y_r' \equiv y_r$.

3. Find R ($1 \le R \le M-1$) such that $y_r^R \le y_r' \le y_r^{R+1}$.

4. Compute $y_r = \sum_{i=1}^M f_r^i y_r^i \Big/ \sum_{i=1}^M f_r^i$ with $f_r^i = \underline{f}^i$ for $i \le R$ and $f_r^i = \overline{f}^i$ for $i > R$, and let $y_r'' \equiv y_r$.

5. If $y_r'' \ne y_r'$, then go to Step 6. If $y_r'' = y_r'$, then stop and set $y_r'' \equiv y_r$.

6. Set y_r' equal to y_r'', and return to Step 3.

The procedure for computing y_l is very similar to the one just given for y_r. Just replace y_r^i by y_l^i, and, in step 3 find L ($1 \le L \le M-1$) such that $y_l^L \le y_l' \le y_l^{L+1}$. Additionally, in step 2 we now compute y_l as $y_l = \sum_{i=1}^{M} f_l^i y_l^i \big/ \sum_{i=1}^{M} f_l^i$ by initially setting $f_r^i = (\underline{f}^i + \overline{f}^i)\big/2$ for i = 1, …, M, and, in step 4 we compute y_l as $y_l = \sum_{i=1}^{M} f_l^i y_l^i \big/ \sum_{i=1}^{M} f_l^i$ with $f_l^i = \overline{f}^i$ for $i \le L$ and $f_l^i = \underline{f}^i$ for $i > L$.

These two four-step iterative-procedures (Steps 1 and 2 are initialization-steps) have been proven by Karnik and Mendel (2001) to converge to the exact solutions in no more than M iterations.

Table C-1: Meanings of y_l^i, y_r^i, $\underline{f}^i$, $\overline{f}^i$ and M in (C-1) for different type-reduction methods[a].

Type-reduction method	y_l^i and y_r^i defined	$\underline{f}^i$ and $\overline{f}^i$ defined[d]	M defined
center-of sets	left and right end points of the centroid of the consequent of the ith rule	lower and upper firing degrees of the ith rule	number of rules
centroid[b]	$y_l^i = y_r^i = y^i$, the ith point in the sampled universe of discourse of the FLS's output	lower and upper membership grades of the ith sampled output of the FLS's output	number of sampled points
center-of-sums[c]	$y_l^i = y_r^i = y^i$, the ith point in the sampled universe of discourse of the FLS's output	*sums* of lower and upper membership grades for the ith sampled point of all rule outputs	number of sampled points
height	$y_l^i = y_r^i = y^i$, a single point in the consequent domain of the ith rule, usually chosen to be the point having the highest primary membership in the principal MF of the output set	lower and upper firing degrees of the ith rule	number of rules

a. Comparable results for modified height type-reduction can be found in Mendel (2001a), Section 9.5.4.

b. Prior to calculating the centroid type-reduced set, the fired type-2 fuzzy sets are unioned.

c. Prior to calculating the center-of-sums type-reduced set, the membership functions of the fired type-2 fuzzy sets are added (or a linear combination of them is formed).

d. See Appendix D for formulas for $\underline{f}^i$ and $\overline{f}^i$, as well as for rule outputs and unioned rule outputs.

APPENDIX D: Fuzzy Inference Engine Results for Interval Type-2 Fuzzy Sets

Consider a type-2 FLS having p inputs $x_1 \in X_1,\ldots,x_p \in X_p$ and one output $y \in Y$. We assume there are M rules where the lth rule has the form

$$R^l\text{: IF } x_1 \text{ is } \tilde{F}_1^l \text{ and } \cdots \text{ and } x_p \text{ is } \tilde{F}_p^l, \text{ THEN } y \text{ is } \tilde{G}^l \quad l=1,\ldots,M \tag{D-1}$$

This rule represents a type-2 relation between the input space $X_1 \times \cdots \times X_p$, and the output space, Y, of the type-2 FLS. Associated with the p antecedent type-2 fuzzy sets, $\tilde{F}_i^l$, are the type-2 MFs $\mu_{\tilde{F}_i^l}(x_i)$ (i = 1, …, p), and associated with the consequent type-2 fuzzy set $\tilde{G}^l$ is its type-2 MF $\mu_{\tilde{G}^l}(y)$.

The major result for an interval singleton type-2 FLS is summarized in the following:

Theorem D-1: [Liang and Mendel (2000), Mendel (2001a)] In an interval singleton type-2 FLS using product or minimum t-norm, for input $\mathbf{x}=\mathbf{x}'$: (a) The result of the input and antecedent operations, is an interval type-1 set, called the *firing set*, i.e.,

$$F^l(\mathbf{x}') = [\underline{f}^l(\mathbf{x}'), \overline{f}^l(\mathbf{x}')] \equiv [\underline{f}^l, \overline{f}^l] \tag{D-2}$$

where

$$\underline{f}^l(\mathbf{x}') = \underline{\mu}_{\tilde{F}_1^l}(x_1') \bigstar \cdots \bigstar \underline{\mu}_{\tilde{F}_p^l}(x_p') \tag{D-3}$$

and

$$\overline{f}^l(\mathbf{x}') = \overline{\mu}_{\tilde{F}_1^l}(x_1') \bigstar \cdots \bigstar \overline{\mu}_{\tilde{F}_p^l}(x_p'); \tag{D-4}$$

(b) The rule R^l fired output consequent set, $\mu_{B^l}(y)$, is the type-1 interval fuzzy set

$$\mu_{\tilde{B}^l}(y) = \int_{b^l \in \left[\underline{f}^l \star \underline{\mu}_{\tilde{G}^l}(y),\, \overline{f}^l \star \overline{\mu}_{\tilde{G}^l}(y)\right]} 1 \big/ b^l, \quad y \in Y \qquad \text{(D-5)}$$

where $\underline{\mu}_{\tilde{G}^l}(y)$ and $\overline{\mu}_{\tilde{G}^l}(y)$ are the lower and upper membership grades of $\mu_{\tilde{G}^l}(y)$.

(c) Suppose that *N* of the *M* rules in the FLS fire, where $N \leq M$, and the combined output type-1 fuzzy set, $\mu_{\tilde{B}}(y)$, is obtained by combining the fired output consequent sets by taking the union of the rule R^l fired output consequent sets[10]; then,

$$\mu_{\tilde{B}}(y) = \int_{b \in \left[\left[\underline{f}^1 \star \underline{\mu}_{\tilde{G}^1}(y)\right] \vee \cdots \vee \left[\underline{f}^N \star \underline{\mu}_{\tilde{G}^N}(y)\right],\, \left[\overline{f}^1 \star \overline{\mu}_{\tilde{G}^1}(y)\right] \vee \cdots \vee \left[\overline{f}^N \star \overline{\mu}_{\tilde{G}^N}(y)\right]\right]} 1 \big/ b, \quad y \in Y \ \blacksquare \qquad \text{(D-6)}$$

A complete proof of this theorem can be found in Liang and Mendel (2000) and in Mendel (2001a). Generalizations of this theorem to the very important case when the input to the type-2 FLS is a type-2 fuzzy set—which would be the case when the words that activate the Per-C are modeled as type-2 fuzzy sets—are also given in those references.

[10] We do not necessarily advocate taking the union of these sets. Part c merely illustrates the calculations if one chooses to do this.

References

Karnik N. N. and Mendel J. M. (2001): *Centroid of a type-2 fuzzy set*. — Information Sciences, Vol. 132, pp. 195-220.

Klir G. J. and Wierman M. J. (1998): *Uncertainty-Based Information*. — Heidelberg, Germany: Physica-Verlag.

Liang Q. and Mendel J. M. (2000): *Interval type-2 fuzzy logic systems: theory and design*. — IEEE Trans. on Fuzzy Systems, Vol. 8, pp. 535-550.

Mendel J. M. (1999): *Computing with words, when words can mean different things to different people*. — Proc. of Third International ICSC Symposium on Fuzzy Logic and Applications, Rochester Univ., Rochester, NY, June, pp. 158-164.

Mendel J. M. (2001a): *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions*. — Upper Saddle River, NJ: Prentice-Hall.

Mendel J. M. (2001b): *On the importance of interval sets in type-2 fuzzy logic systems*. — Proc. of IFSA/NAFIPS Conference, Vancouver, Canada, July, pp. 1647-1652.

Mendel J. M. (2001c): *The Perceptual Computer: an architecture for computing with words*. — Proc. IEEE Conference on Fuzzy Systems, Melbourne, Australia, December, pp. 35-38.

Mendel J. M. (2002): *Uncertainty, type-2 fuzzy sets, and footprints of uncertainty*. — The 9th Int'l. Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, Annecy, France, July, pp. 325-331.

Mendel J. M. and Wu H. (2002): Uncertainty *versus* choice *in rule-based fuzzy logic systems*. — IEEE WCCI, FUZZ Conference, Honolulu, HI, May, pp. 1336-1341.

Wang L.-X. (1997): *A Course in Fuzzy Systems and Control*. — Upper Saddle River, NJ: Prentice-Hall.

Wang P. P. (Ed.) (2001): *Computing With Words*. — New York: John Wiley & Sons, Inc.

Yager R. R. and Filev D. P. (1994): *Essentials of Fuzzy Modeling and Control*. — New Jersey: John Wiley & Sons, Inc.

Zadeh L. A. (1996): *Fuzzy logic = computing with words*. — IEEE Trans. on Fuzzy Systems, Vol. 4, pp. 103-111.

Zadeh L. A. (1999): *From computing with numbers to computing with words—from manipulation of* *measurements to manipulation of perceptions.* — IEEE Trans. on Circuits and Systems–I: Fundamental Theory and Applications, Vol. 4, pp. 105-119.