

DIE SIMULATION VAGER INFERENZEN ÜBER UNSCHARFEM WISSEN: EINE ANWENDUNG DER MEHRWERTIGEN PROGRAMMIERSPRACHE FUZZY

WOLFGANG WAHLSTER

Germanisches Seminar
Universität Hamburg

SUMMARY. A major problem for the computer simulation of the cognitive processes underlying natural language dialogue comprehension is to handle the fuzziness that is intrinsic to natural language and human thinking. FUZZY, a LISP-embedded PLANNER-type programming language, is a powerful implementation tool in this area, since it provides a number of useful facilities for the efficient representation and manipulation of fuzzy knowledge. It is shown how in the natural language system HAM-RPM, which simulates a dialogue partner conversing about a real-life domain, the FUZZY language is used to represent fuzzy inference rules and the semantics of vague natural language terms. We present in this paper some recent investigations into the use of FUZZY for the implementation of HAM-RPM's inference memory, which maintains an internal trace of the system's inference behaviour. We describe in some detail how the 'procedure demon' mechanism of FUZZY for controlling the global evaluation of fuzzy inference procedures is utilized to store the goal tree; this is useful for subsequent explanations of the reasoning process. 'Why'-questions of the user have in HAM-RPM the effect of traversing the goal tree looking for relevant explanations. The specific organization of HAM-RPM's inference memory makes it possible to answer 'Why'-questions on various levels of detail depending on the conversational state. Explicit comparisons are made with the corresponding capabilities of SHRDLU and MYCIN.

DIE KOMMUNIKATIVE UND KOGNITIVE FUNKTION UNSCHARFER AUSDRÜCKE UND VAGER INFERENZEN

Die Verwendung vager sprachlicher Ausdrücke und vager Inferenzen ist ein typisches Merkmal umgangssprachlicher Kommunikation. Nachdem die Vagheit lange Zeit als Defekt der natürlichen Sprache und des menschlichen Denkens angesehen wurde, ist man heute in den kognitiven Wissenschaften zu einer positiven Auffassung dieses Phänomens gekommen, da man erkannt hat, daß die Vagheit entscheidend zur Nützlichkeit des Kommunikationswerkzeuges 'natürliche Sprache' beiträgt, indem sie es zum Beispiel erlaubt, von irrelevanten Details zu abstrahieren und Information durch vage Prädikationen zu komprimieren (vgl. RIEGER 1976, WAHLSTER 1977).

Dabei spricht man von der Vagheit eines natürlichsprachlichen Ausdrucks, wenn es für bestimmte Anwendungsbereiche und -Situationen keine eindeutigen Konventionen dafür gibt, ob der Ausdruck zur Bezeichnung eines bestimmten Gegenstandes, Sachverhaltes oder Prozesses benutzt werden kann oder ob seine Verwendung ausgeschlossen ist, oder wenn bestimmte, potentiell relevante Aspekte der Aussage, die durch den Ausdruck formuliert wird, offen bleiben (z.B. sind 'groß', 'fast alle', 'nachmittags', und 'Luxusartikel' vage Ausdrücke).

Die Beachtung folgender von GRICE (1975) postulierter allgemeiner Konversationsmaximen führt in natürlichsprachlichen Dialogen zwangsläufig zur Verwendung vager Prädikationen:

- Mache deinen Beitrag nicht informativer als erforderlich.
- Mache deinen Beitrag relevant.

Bei der Entwicklung eines Simulantsystems (vgl. auch den Beitrag von v. HAHN in diesem Band), in dem kommunikative und kognitive Leistungen eines menschlichen Gesprächspartners in einer einfachen, alltäglichen Situation rekonstruiert werden, muß daher die menschliche Fähigkeit, durch plausibles Schließen auch aus vagem Wissen nützliche Aussagen abzuleiten, berücksichtigt werden

(vgl. auch den Beitrag 'Nicht-deterministisches, nicht-probabilistisches Denken und Erkennen: Possibilistische kognitive Prozesse' von UECKERT in diesem Band). Dabei ergibt sich zunächst das Problem einer adäquaten Repräsentation von vagem und unvollständigem Wissen, das als eine Ansammlung von Kenntnissen, Erfahrungen und Problemlösungsmethoden den Hintergrund für komplexe Informationsverarbeitungsprozesse bildet. In WAHLSTER (1977) wurde untersucht, welche Erweiterungen der innerhalb der Künstlichen Intelligenz (KI) bisher vorwiegend für exakte, in sich abgeschlossene Wissensgebiete entwickelten Repräsentationssprachen (u.a. Prädikatenlogik, Produktionensysteme, Semantische Netze, KI-Programmiersprachen, Frame-Systeme) es ermöglichen, auch vages Wissen darzustellen und zu verarbeiten. Im folgenden wird gezeigt, wie die KI-Programmiersprache FUZZY (vgl. LE FAIVRE 1974), die sich für die genannte Aufgabenstellung als besonders gut geeignet erwies, in dem natürlichsprachlichen Dialogsystem HAM-RPM (vgl. WAHLSTER & v. HAHN 1976) zur Darstellung der extensionalen Semantik vager sprachlicher Ausdrücke und zur Simulation von heuristischen Inferenzen bei nur partiell erfüllten Prämissen eingesetzt wird.

Das experimentelle natürlichsprachliche KI-System HAM-RPM antwortet einem menschlichen Dialogpartner auf Fragen über eine vereinbarte Diskurswelt. Obwohl wir mit mehreren Diskurswelten experimentieren, werden sich die folgenden Beispiele auf eine Situation beziehen, in der eine Straßenverkehrsszene besprochen wird (vgl. v. HAHN, HOEPPNER, JAMESON & WAHLSTER 1978).

DIE REPRÄSENTATION DER BEDEUTUNG UNSCHARFER BEGRIFFE IN FUZZY

FUZZY (vgl. LE FAIVRE 1977a) gehört zusammen mit Sprachen wie MIKRO-PLANNER, CONNIVER, QLISP, PLASMA und KRL zur Klasse der 'sehr hohen' Programmiersprachen, die im Rahmen der Forschung zur Künstlichen Intelligenz entwickelt wurden und deren Daten- und Kontrollstrukturen eine flexible Darstellung, modulare Organisation und effiziente Verarbeitung komplexer Wissensstrukturen fördern. Im Gegensatz zu anderen KI-Programmiersprachen orientiert sich FUZZY aber nicht an der zweiwertigen Logik, sondern an der 'fuzzy logic' (Abk.: F-Logik, vgl. GAINES 1976), d.h. an einer auf der Theorie der 'fuzzy sets' (Abk.: F-Mengen, vgl. ZADEH 1977a) basierenden semantischen Interpretation mehrwertiger Logiken. Die listenverarbeitende Sprache LISP (vgl. SIKLOSSY 1976) ist im Sprachumfang von FUZZY enthalten.

Bei der Auswertung eines Ausdrucks in FUZZY lassen sich die beiden in Abb. 1 dargestellten Fälle unterscheiden. Das bei der Auswertung eines Ausdrucks entstehende Wertepaar besteht aus einem Wert und einem sog. Z-Wert, der den Wert modifiziert.

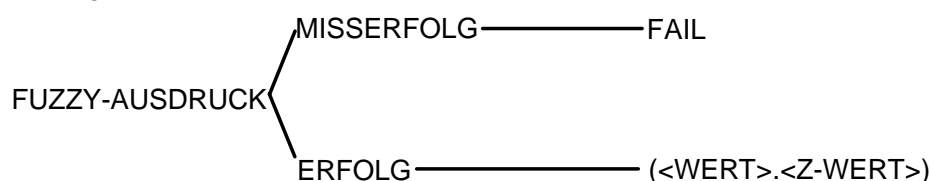


Abb. 1: Die Auswertung eines Ausdrucks in FUZZY

Der numerische Z-Wert kann u.a. als Wahrheitswert im Sinne der F-Logik oder als Zugehörigkeitswert eines Elements zu einer F-Menge interpretiert werden. Der Z-Wert liegt immer im Intervall $[ZLOW, ZHIGH]$, das mit $[0,1] \subset \mathbb{R}$ initialisiert ist und durch entsprechende Elementaranweisungen undefiniert werden kann. Es stehen auch Sprachelemente zur Verfügung, die einen getrennten Zugriff auf die erste (VAL) und zweite Komponente (ZVAL) eines Wertepaares ermöglichen.

FUZZY trägt dazu bei, die Komplexitätsbarrieren, die sich bei der Entwicklung kognitiver Systeme wie HAM-RPM ergeben, abzubauen, indem es nicht nur die für KI-Programmiersprachen typischen Konzepte wie pattern-gesteuerte Suche in einer assoziativen Datenbasis und automatische Deduktionsmechanismen anbietet, sondern dem Programmierer auch die aufwendige Verarbeitung der mit einer F-Logik verbundenen gradierten Wahrheitswerte soweit wie möglich abnimmt, um ihn damit

von der Beschäftigung mit von den eigentlichen Repräsentationsproblemen ablenkender Detailprogrammierung zu befreien.

Vage Prädikationen in Semantischen Netzen

Ein Teil des HAM-RPM zur Verfügung stehenden Wissens ist in Form von Semantischen Netzen codiert, in denen Zahlenwerte aus dem reellen Intervall [0,1] angeben, zu welchem Grad die als Kanten dargestellten semantischen Relationen zwischen den als Knoten repräsentierten semantischen Einheiten erfüllt sind. In den folgenden Beispielen werden wir die in Tabelle 1 zusammengestellte Teilmenge der in HAM-RPM definierten 'epistemologisch primitiven' Kanten verwenden. In HAM-RPM werden Semantische Netze als F-Graphen (vgl. WAHLSTER 1977) dargestellt, in denen die Kanten durch Ausdrücke der Form <Verträglichkeitsgrad >/< Semantische Relation > markiert sind.

Tab. 1: Die Semantik einiger Kanten der Semantischen Netzwerke

$X \xrightarrow{\text{ISA}} Y$	$:\Leftrightarrow X \in Y$	$X \xrightarrow{\text{T}} Y$	$:\Leftrightarrow Y \text{ ist Teil von } X$
$X \xrightarrow{\text{U}} Y$	$:\Leftrightarrow X \subset Y$	$X \xrightarrow{\text{REF}} Y$	$:\Leftrightarrow X \text{ ist durch } Y(X) \text{ charakterisiert}$
$X \xrightarrow{\text{D}} Y$	$:\Leftrightarrow \forall x \in X: Y(x)$	$X \xrightarrow{\text{INST}} Y$	$:\Leftrightarrow X \text{ bezieht sich auf } Y$

Ein Teilgraph wie [1], der bedeutet, daß ein Hut im Gegensatz z.B. zu einer Hose kein prototypisches, sondern eher ein peripheres Element der unscharfen sprachlichen Kategorie 'Kleidung' ist, wird als Assertion in die assoziative Datenbasis von FUZZY durch den Ausdruck [2] eingefügt. Das System hält die assoziative Datenbasis, die aus einer F-Menge von Assertionen besteht, nach abfallenden Z-Werten geordnet, und bei der Suche werden zunächst immer die prototypischen Elemente einer sprachlichen Kategorie gefunden.

$$\begin{array}{l}
 [1] \quad \text{HUT} \xrightarrow{0.6/\text{U}} \text{KLEIDUNG} \\
 [2] \quad (\text{ADD} (\text{U HUT KLEIDUNG}) 0.6)
 \end{array}$$

Eine solche Darstellung ist psychologisch plausibel und wird durch die Ergebnisse empirischer Untersuchungen (vgl. ROSCH & MERVIS 1975) gestützt, die zeigen, daß es einerseits eine bestimmte Menge klarer Fälle und treffender Beispiele und andererseits eine Reihe unklarer Fälle einer sprachlichen Kategorie gibt, welche zu einer Ordnung von besseren zu schlechteren Beispielen einer Kategorie tendieren (vgl. auch KINTSCH 1974, S. 27-31).

Die Assertionen in der Datenbasis können aus beliebig verschachtelten Listen bestehen, so daß das Wissen 'ein Auto hat stets Räder' und 'ein Auto hat meist vier Räder' als [3] repräsentiert werden kann.

$$[3] \quad ((\text{T AUTO RÄDER}). 1.) \quad (((\text{ANZAHL} (\text{T AUTO RÄDER}) 4). 0.7)$$

Auch in der von ZADEH z.Z. entwickelten Repräsentationssprache PRUF (Possibilistic Relational Universal Fuzzy) ist das Konzept einer Datenbasis, die aus F-Relationen besteht, enthalten (vgl. ZADEH 1977b, S.44, ZADEH 1977a, 1978).

Die Repräsentation der extensionalen Semantik vager sprachlicher Ausdrücke

Die referentielle Bedeutung von vagen Ausdrücken wird in HAM-RPM in Form von F-Mengen definiert. Diese F-Mengen werden extensional durch Einträge in der Datenbasis, z.B. durch die Angabe, auf welche Objekte ein vages Prädikat zu welchem Grad zutrifft, oder implizit im Rahmen einer prozeduralen Semantik repräsentiert, indem die Zugehörigkeitsfunktion der F-Menge in Abhängigkeit von Eigenschaften der Elemente definiert wird. Als Beispiel betrachten wir im folgenden die extensionale Semantik des vagen Adjektivs 'groß', wie sie in HAM-RPM definiert ist,

und beziehen uns dabei auf die Teilmenge [4] der im assoziativen Netz gespeicherten Assertionen.

```
[4] ((REF BAUM1 3 INST HOEHE). 0.9) (REF BAUM1 15 INST HOEHE). 0.8)
      ((REF BAUM3 GROSS). 0.7) (REF BAUM4 KLEIN). 0.25)
```

Neben einer relativ exakten Angabe wie 'Baum1 ist rund 15m hoch' wird in diesem Netz auch sehr vages Wissen gespeichert, z.B. daß Baum3 'ziemlich groß' ist.

In HAM-RPM bewirkt die Analyse der Entscheidungsfrage 'Ist der Baum rechts vorne groß?', daß in der Antwortbasis der Ausdruck [5] ausgewertet wird.

```
[5] (GOAL (REF BAUM3 GROSS) 0.3)
```

Dabei ergibt sich unmittelbar das Wertepaar ((REF BAUM3 GROSS). 0.7), da wie in KI-Sprachen üblich bei der Auswertung einer GOAL-Anweisung zunächst versucht wird, eine passende Assertion in der Datenbasis zu finden. Ersetzt man in [5] BAUM3 durch BAUM1, so wird die im folgenden beschriebene Darstellung der referentiellen Semantik von 'groß', in der auf die Höhenangabe für BAUM1 Bezug genommen wird, durch einen pattern-gesteuerten Prozeduraufruf aktiviert. Dabei ist zunächst zu berücksichtigen, daß mit der Verwendung relativer Adjektive wie 'groß', 'alt', 'teuer', 'eng' stets ein impliziter, kontextabhängiger Vergleich mit einer Erwartungsnorm verbunden ist.

Relative Adjektive. Die Darstellung der referentiellen Bedeutung vager, relativer Adjektive wie 'groß' in Form von implizit definierten F-Mengen muß in den entsprechenden Repräsentationskonstruktionen stets auf eine bestimmte Referenzmenge bezogen werden, so daß in der Regel zur Darstellung der Semantik eines relativen Adjektivs mehrere Repräsentationskonstruktionen abgespeichert werden müssen. Die extensionale Bedeutung von 'groß' in bezug auf die Referenzmenge 'Bäume' wird mit Hilfe einer hier in vereinfachter Form wiedergegebenen DEDUCE-Prozedur definiert (vgl. [6]). DEDUCE-Prozeduren haben in FUZZY die gleiche Funktion wie Konsequenz-Theoreme in MIKRO-PLANNER oder 'if-needed methods' in CONNIVER.

```
[6] (a) (ADD DEDUCE:
      (b) (PROC NAME: M-GROSS
          (c) (REF (*R ?TOKEN (EQ (TYPUS !TOKEN) (QUOTE BAUM))) GROSS)
          (d) (FETCH (REF !TOKEN ?HOEHENANGABE INST HOEHE))
          (e) (SUCCEED (REF !TOKEN GROSS) (SFUNK !HOEHENANGABE 10 20))))
```

Der Teilausdruck in Zeile (a) bewirkt, daß der in (b)-(e) definierte Teilausdruck als prozedurales Wissen in der Wissensbasis abgelegt wird. In Zeile (c) wird das charakteristische Pattern (Invokationspattern) der Prozedur definiert. Einfache wertaufnehmende Variablen werden durch '?' und wertabgebende Variablen durch '!' präfigiert. Das Symbol *R kennzeichnet eine restringierte Variable und bewirkt, daß ein 'semantischer Match' durchgeführt wird, d.h. daß nur diejenigen Bezeichner an die Variable ?TOKEN gebunden werden können, für die eine Auswertung von (EQ (TYPUS !TOKEN) (QUOTE BAUM)) erfolgreich verläuft. TYPUS liefert dabei die sprachliche Kategorie, zu der ein bestimmtes Objekt gehört. In unserem Beispiel wird BAUM1 an !TOKEN gebunden. In (d) erfolgt ein Zugriff auf die assoziative Datenbasis durch die Angabe eines Patterns in einem FETCH-Ausdruck. Es wird im assoziativen Netz nach einer Höhenangabe für BAUM1 gesucht. In (e) wird das Wertepaar, welches das Ergebnis der Prozedur M-GROSS darstellt, konstruiert. Der Z-Wert wird als zweite Komponente durch die Auswertung der LISP-Funktion SFUNK berechnet. SFUNK ist eine standardisierte, über Parameter einrichtbare Funktion mit Definitionsbereich \mathbb{R} und Wertebereich $[0,1] \subset \mathbb{R}$, die eine einfache Darstellung einer Klasse von Zugehörigkeitsfunktionen ermöglicht (vgl. WAHLSTER 1977, S. 59ff). In [6] ergibt sich für die Extremfälle, d.h. für die !HOEHENANGABE ≤ 10 der Wert 0 und für !HOEHENANGABE ≥ 20 der Wert 1, und für !HOEHENANGABE = 15 der Z-Wert 0.5, so daß insgesamt das Wertepaar ((REF BAUM1 GROSS). 0.5) entsteht.

Ersetzt man in [5] BAUM3 durch BAUM2, so ergibt sich nach einem pattern-gesteuerten Prozeduraufruf zunächst intern der Wert ((REF BAUM2 GROSS). 0). Da dieser Z-Wert nicht über dem in der GOAL-Anweisung [5] angegebenen Schwellwert von 0.3 liegt, wird der Ausdruck zu FAIL ausgewertet. Der Z-Wert 0 wird in HAM-RPM als Negation interpretiert und von FAIL unterschieden.

Linguistische Hecken. In der Wissensbasis von HAM-RPM wird auch die Semantik sog. linguistischer Hecken, d.h. von Ausdrücken wie 'sehr', 'mehr oder weniger', 'relativ' formal dargestellt, um auch Fragen wie [7] analysieren und beantworten zu können oder unscharfe Ausdrücke wie 'in etwa rechts neben der Ampel' zur Verbalisierung räumlicher Beziehungen generieren zu können.

[7] Ist der Baum links vorne mehr oder weniger klein?

Linguistische Hecken werden in der hier ausschließlich betrachteten extensionalen Semantik als unäre Operatoren dargestellt, welche die Bedeutung der als Operanden fungierenden F-Mengen verändern, indem sie deren Vagheit verstärken oder abschwächen (vgl. ZADEH 1972). Eine solche Repräsentation, die nicht nur als formales linguistisches Modell sondern auch als psycholinguistisches Modell geeignet ist (vgl. HERSH & CARAMAZZA 1976), wird in HAM-RPM ebenfalls durch DEDUCE-Prozeduren realisiert. Die Wirkung der linguistischen Hecke 'mehr oder weniger' aus [7] kann in vereinfachter Form durch die DEDUCE-Prozedur [8] dargestellt werden.

[8] (a) (PROC (?KANTE ?KNOTEN1 (MEHR/ ODER/ WENIGER ?KNOTEN2))
 (b) (GOAL (!KANTE !KNOTEN1 !KNOTEN2))
 (c) (SUCCEED (!KANTE !KNOTEN1 (MEHR/ ODER/ WENIGER !KNOTEN2)) (SQRT (ZVAL))))

Die Funktion SQRT (Quadratwurzel) in Zeile (c) ist nur eine grobe Annäherung der komplexen, oft kontextabhängigen Operation über Zugehörigkeitsfunktionen, die der Semantik von 'mehr oder weniger' entspricht (vgl. WAHLSTER 1977, S. 72ff). ZVAL in (c) liefert als parameterlose Prozedur aufgerufen den Z-Wert, der sich bei der Auswertung von (b) ergibt. Wird in HAM-RPM der von der Tiefenstruktur von [7] abgeleitete FUZZY-Ausdruck (GOAL (REF BAUM4 (MEHR/ ODER/ WENIGER KLEIN)) 0.3) ausgewertet, so ergibt sich das Wertepaar ((REF BAUM4 (MEHR/ ODER/ WENIGER KLEIN)). 0.5).

DIE REPRÄSENTATION UND ANWENDUNG VAGER INFERENZREGELN IN FUZZY

Nachdem wir gezeigt haben, wie die extensionale Semantik vager Konzepte und Operatoren über ihnen in HAM-RPM mit Hilfe von FUZZY repräsentiert werden, gehen wir im folgenden auf die Darstellung vager Inferenzregeln ein (vgl. LE FAIVRE 1977b). Wir diskutieren zunächst ein einfaches, allgemeines Beispiel, um uns danach mit der Anwendung und der verbalen Begründung vager Inferenzen in HAM-RPM zu beschäftigen. Die vagen Propositionen 'Fritz ist Peter sehr gut bekannt', 'Peter kennt Wolfgang relativ gut', 'Udo kennt Hans mehr oder weniger gut' und 'Peter kennt Udo nur flüchtig' seien als [9] im assoziativen Netz gespeichert.

[9] (((IST-BEKANNT FRITZ PETER). 1.) ((IST-BEKANNT PETER WOLFGANG). 0.7) ((IST-BEKANNT UDO HANS). 0.5) ((IST-BEKANNT PETER UDO). 0.3))

Die vage Inferenzregel [10] basiert auf der menschlichen Erfahrung, daß zwei Personen, die sich gut kennen, evtl. auch gemeinsame Bekannte haben. [10] kann in einer F-Logik als vage Implikation [11] dargestellt werden.

[10] Wenn y x bekannt ist und z y bekannt ist, dann ist möglicherweise z auch x bekannt.
 [11] $\text{T}(\forall_{x \in M} \forall_{y \in M} \forall_{z \in M} \text{IST-BEKANNT}(x,y) \wedge \text{IST-BEKANNT}(y,z) \Rightarrow \text{IST-BEKANNT}(x,z)) = 0.7$

Falls der Bekanntheitsgrad unter den auf der linken Seite von [11] in Beziehung gesetzten Personen gering ist, d.h. falls die Prämisse von [11] nur partiell erfüllt ist, muß gemäß unserer Intuition der Wahrheitswert der durch die vage Inferenzregel abgeleiteten Aussage entsprechend stärker eingeschränkt werden.

Besteht die auszuführende Schlußkette aus einer Folge von modus-ponens-Anwendungen, innerhalb derer die vage Inferenzregel mehrmals angewandt wird, so ist die Möglichkeit, daß die abgeleitete Aussage zutreffend ist, mit jeder Anwendung stärker eingeschränkt. Im Gegensatz zur zweiwertigen Logik gibt es in der F-Logik mehrere Möglichkeiten, einen Abtrennungsoperator für den modus ponens zu definieren. Für das vorliegende Beispiel entspricht der multiplikative Abtrennungsoperator, für den $T(B) = T(A) \cdot T(A \Rightarrow B)$ gilt, am besten der eingeschränkten Transitivität von 'IST-BEKANNT', da er bewirkt, daß der Wahrheitswert der Konklusion mit der Länge der Deduktionskette abnimmt.

Bei der Repräsentation vager Inferenzregeln ergeben sich zunächst die beiden folgenden Probleme:

- die Darstellung der 'Stärke' der Implikation
- die Spezifikation eines geeigneten Abtrennungsoperators

Die Simulation approximativer Schlüsse

In FUZZY werden vage Inferenzregeln durch eine allgemeinere Form der DEDUCE-Prozedur als in [6] und [8] angewandt dargestellt, wobei zwei optionale Angaben ZVAL:<X> und DEMON:<Y> im Prozedurkopf zur Spezifikation der Implikationsstärke <X> bzw. des Abtrennungsoperators <Y> dienen. Durch die FUZZY DEDUCE-Prozedur [12] wird die Inferenzregel [11] in einer kompakten Darstellung als prozedurales Wissen in die Datenbasis eingebettet.

```
[12] (a) (ADD DEDUCE:
      (b)      (PROC NAME: TRANS-BEKANNT DEMON: MULT-ABTRENN ZVAL: 0.7
      (c)      (IST-BEKANNT ?PERSON1 ?PERSON3)
      (d)      (FOR FETCH: (IST-BEKANNT !PERSON1 ?PERSON2)
      (e)      (FOR GOAL: (IST-BEKANNT !PERSON2 !PERSON3 (SUCCEED?))))))
```

Im Prozedurrumpf in (d) und (e) wird mit Hilfe des Repetitionsoperators FOR und der Anweisung (SUCCEED?) eine automatische Tiefensuche nach Assertionen codiert, welche die Prämisse von [11] erfüllen.

Prozedur-Dämonen. Diese übersichtliche und flexible Darstellung vager Inferenzregeln wird in FUZZY durch das Konzept der Prozedur-Dämonen ermöglicht, welches neben der Mehrwertigkeit das zweite Charakteristikum dieser Programmiersprache ist. Prozedur-Dämonen überwachen und steuern die Auswertung aller Prozeduren, auf die sie 'angesetzt' sind und werden nach jeder Auswertung eines Teilausdrucks der überwachten Prozedur automatisch aktiviert. Die aus anderen KI-Programmiersprachen bekannten Dämon-Prozeduren (vgl. Antezedens-Theoreme in MIKRO-PLANNER und 'if-added methods' in CONNIVER), welche in FUZZY als Prozeduren vom Typ ASSERT und ERASE realisiert werden, überwachen dagegen das Hinzufügen und Löschen im assoziativen Netz (vgl. Abb. 2). Ein Prozedur-Dämon ist eine LISP-Funktion X, die an eine FUZZY-Prozedur Y gekoppelt ist.

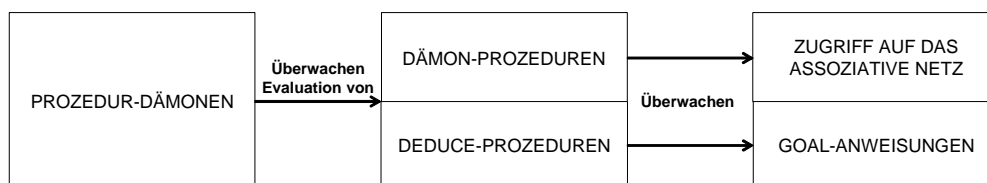


Abb. 2: Prozedur-Dämonen und Dämon-Prozeduren

Nach der Evaluation jedes einzelnen Teilausdrucks von Y wird die Kontrolle an X übergeben. X wird mit folgenden Parametern versorgt:

- Wert des zuletzt evaluierten Teilausdrucks (V)
- durch ZVAL in der überwachten Prozedur gekennzeichnete Wert (ZV)
- akkumulierter Z-Wert, der dynamisch verwaltet wird (AC)

Der *multiplikative Abtrennungsoperator*. Der dem Inferenztyp von [11] angemessene Abtrennungsoperator MULT-ABTRENN kann durch folgenden Prozedur-Dämon definiert werden:

```
[13] (a) (DEFPROP MULT-ABTRENN
      (b)      (LAMBDA (V ZV AC)
      (c)      (COND ((EQ V FAIL) (FAIL))
      (d)      ((EQ V DONE) AC)
      (e)      (T (*MIN AC (*TIMES ZV (ZVAL V)))))) EXPR)
```

Der Wert DONE entsteht nach der Auswertung des letzten Teilausdrucks der überwachten Prozedur. Der Wert der Konjunktion der Prämissen von [11] wird in (e) gemäß der F-Logik als Minimum der Z-Werte der Konjunkte definiert.

[14] (GOAL (IST-BEKANNT FRITZ HANS))

Als Ergebnis der Auswertung des Ausdrucks [14], welcher der Frage 'Kennt Hans Fritz?' entspricht, ergibt sich nach dem in Abb. 3 dargestellten Inferenzprozess der Wert ((IST-BEKANNT FRITZ HANS). 0.147), der z.B. als 'Es ist unter Umständen möglich, daß Hans Fritz bekannt ist' verbalisiert werden kann.

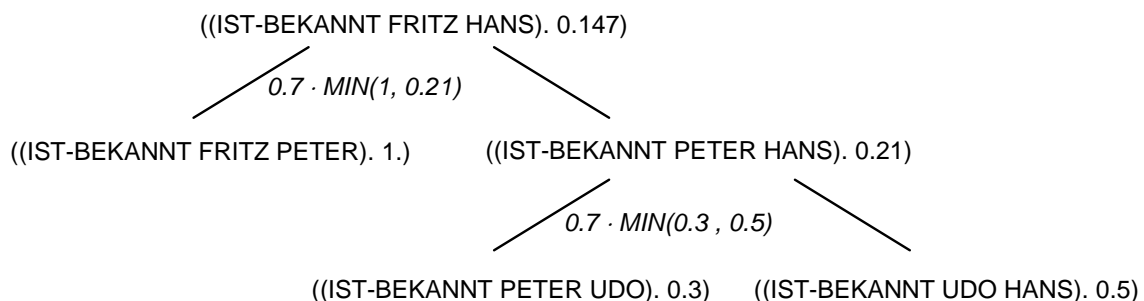


Abb. 3: Zielbaum bei Anwendung eines multiplikativen Abtrennungsoperators

Während der Auswertung von [14], bei der TRANS-BEKANNT rekursiv aufgerufen wird, gerät das System in eine Sackgasse. Da Peter Wolfgang besser kennt als Udo, wird zunächst geprüft, ob Wolfgang Hans kennt. Da dies nicht der Fall ist, muß ein 'backtracking'-Prozeß durchgeführt werden. Ein Problem, auf das in diesem Aufsatz nicht näher eingegangen werden soll, besteht darin, daß die Z-Werte in Abb. 3 auf unterschiedliche Weise interpretiert und verbalisiert werden müssen, da es sich in der Prämisse von [11] um einen Bekanntschaftsgrad, in der Konklusion dagegen um einen subjektiv empfundenen 'Möglichkeitsgrad' (vgl. ZADEH 1978, UECKERT i.d.B.) handelt.

Die Begründung von Schlußfolgerungen mit Hilfe eines Inferenzgedächtnisses

Im Rahmen der in HAM-RPM simulierten Dialogsituation ist es möglich, daß der menschliche Dialogpartner, nachdem eine von ihm gestellte Frage durch den simulierten Redepartner beantwortet wurde, das System durch eine 'Warum'-Frage zu einer Begründung seiner Antwort auffordert. Gerade

dann, wenn Antworten des Systems durch vage Inferenzen und Mutmaßungen zustande kommen, muß ein intelligentes System dazu in der Lage sein, seine Antwort durch die Verbalisierung der vollzogenen Inferenzschritte zu erläutern. Auch für die Weiterentwicklung und die Testbarkeit des Simulationssystems (vgl. auch v. HAHN 1978) sind 'Warum'-Fragen nützlich, da auf diese Weise die Suche nach der Ursache 'falscher' Antworten des Systems erleichtert wird und die Auswirkung von Veränderungen in der Wissensbasis unmittelbar auf der Ebene natürlichsprachlicher Performanz überprüft werden kann (vgl. DAVIS, BUCHANAN & SHORTLIFFE 1977). Nehmen wir an, daß an HAM-RPM die Frage [15] gestellt wird.

[15] Ist die Parkzone links hinten geteert?

Falls die Parkzone für den simulierten Dialogpartner z.Z. aufgrund einer Verdeckung durch parkende Autos nicht sichtbar ist und im referentiellen Netz (vgl. WAHLSTER & v. HAHN 1976) keine Angaben zur Beschaffenheit der Parkzone gespeichert sind, versucht das System aufgrund von allgemeinem Wissen trotz unvollständiger Information eine Antwort durch vage Inferenzen zu finden. Um eine vom Dialogpartner geforderte Begründung für seine Antwort liefern zu können, muß das System über die Fähigkeit verfügen

- ein Inferenzgedächtnis aufzubauen und
- relevante Teile des Inferenzgedächtnisses zu verbalisieren.

Im Zusammenhang mit den oben angeführten Konversationsmaximen ergibt sich dabei das Problem, den für eine kommunikativ adäquate Begründung erforderlichen Detaillierungsgrad und das richtige Argumentationsniveau zu finden. 'Triviale' Schlüsse, wie die Anwendung von Definitionen oder elementarer logischer Schlüsse, sind bei der Generierung einer Begründung zunächst zu übergehen, und erst aufgrund weiterer Nachfragen des Benutzers wird schließlich die gesamte Inferenzkette ausgegeben. Im folgenden wird gezeigt, wie die Fähigkeit, sein eigenes Inferenzverhalten zu erklären, in HAM-RPM mit Hilfe von FUZZY simuliert wird.

Inferenz-Präferenzen. In dem assoziativen Netz seien u.a. folgende Assertionen gespeichert: (((T STRASSE PARKZONE). 0.9) ((U VERKEHRSWEG STRASSE). 1.) ((D VERKEHRSWEG GETEERT). 0.5)). Außerdem seien die Inferenzregeln [16] und [17] in der Wissensbasis gespeichert.

```
[16] (a) (PROC NAME: REGEL1 DEMON: MIN-TRACE-DEMON
      (b)      ZVAL: (QUOTE (0.3 0.5))          (D ?X ?Y)
      (c)      (FOR FETCH: (U ?Z !X)
      (d)      (FOR GOAL: (D !Z !Y)          (SUCCEED?))))

[17] (a) (PROC NAME: REGEL2 DEMON: MIN-TRACE-DEMON
      (b)      ZVAL: (QUOTE (0.4 0.8))          (D ?X ?Y)
      (c)      (FOR FETCH: (T ?Z !X)
      (d)      (FOR GOAL: (D !Z !Y)          (SUCCEED?))))
```

REGEL1 entspricht einem Schluß von der Eigenschaft einer F-Menge auf Eigenschaften ihrer Elemente. In REGEL2 wird von der Eigenschaft eines zusammengesetzten Objekts auf die Eigenschaft seiner Teile geschlossen. In Zeile (b) des Prozedurkopfes werden in beiden Inferenzregeln zwei Werte als ZVAL spezifiziert. Der erste wird als Schwellwert interpretiert, der von keinem Glied der Inferenzkette unterschritten werden darf. Der zweite Wert gibt an, wie trivial, bzw. wie erwähnenswert, die entsprechende Inferenz ist und wird im folgenden als 'Triftigkeit' bezeichnet (vgl. auch die I-tags in CARBONELL & COLLINS 1973).

Die Analyse von Frage [15] führt zur Auswertung des Ausdrucks [18].

[18] (GOAL (D PARKZONE GETEERT))

Da sowohl das charakteristische Pattern von REGEL1 (vgl. [16] (b)) als auch von REGEL2 (vgl. [17] (b)) zu der in [18] angegebenen Assertion paßt, muß aufgrund einer Inferenzstrategie entschieden werden, welche Regel präferiert wird. Zur Zeit wird von HAM-RPM die Strategie verfolgt, daß zunächst immer spezielle Inferenzen, die durch eine hohe Triftigkeit gekennzeichnet sind, angewendet werden und daß erst, falls diese nicht zum Ziel führen, auf allgemeine, aber weniger erwähnenswerte Inferenzregeln zurückgegriffen wird. Bei der Anwendung dieser Strategie, die nur eine von vielen denkbaren kognitiven Hypothesen modelliert, auf Ausdruck [18] wird zunächst REGEL2 ausgewertet. Nach einem weiteren rekursiven Aufruf von REGEL2, der nicht zum Ziel führt, wird die Inferenzkette durch Anwendung von REGEL1 erfolgreich abgeschlossen (vgl. Abb. 4).

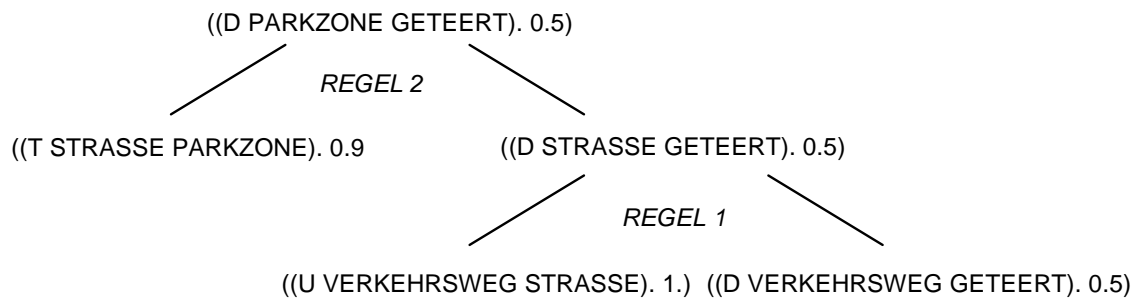


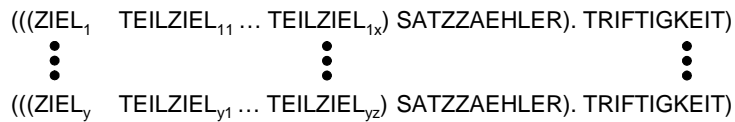
Abb. 4: Im Inferenzgedächtnis gespeicherter Zielbaum

Der Aufbau eines Inferenzgedächtnisses. Der speziell zum Aufbau eines Inferenzgedächtnisses entwickelte Prozedur-Dämon MIN-TRACE-DEMON (vgl. [19]) baut, während er die Auswertung von Inferenzregeln überwacht, den entstehenden Zielbaum in einem separaten assoziativen Netz auf. Außerdem bewirkt MIN-TRACE-DEMON, daß gemäß dem Inferenztyp von REGEL1 und REGEL2 das Minimum als Abtrennungsoperator benutzt wird.

```
[19] (a) (DEFPROP MIN-TRACE-DEMON
      (b) (LAMBDA (V ZV AC)
          (c) (SETQ AC (LISTIFY AC))
          (d) (COND ((EQ V DONE) (IN (QUOTE INFERENZGEDAECHTNIS)
              (e) (ADD & (LIST (CONS (LIST (ZINSTR (CAR (GET ZNAME (QUOTE PDEF))))
              (f) (CAR AC)) (REVERSE (CDR AC))) SATZZAEHLER) (CADR ZV)))
              (g) (CAR AC))
              (h) ((EQ V FAIL) (FAIL))
              (i) ((*LESS (ZVAL V) (CAR ZV)) (FAIL))
              (j) (T (CONS (*MIN (CAR AC) (ZVAL V))
              (k) (CONS (LIST (VAL V) (ZVAL V)) (CDR AC)))))) EXPR)
```

Falls ein Teilausdruck der überwachten Prozedur FAIL liefert (h) oder einen Wert, der unter dem in der überwachten Prozedur als Schwellwert angegebenen Z-Wert (i) liegt, liefert MIN-TRACE-DEMON als Ergebnis (FAIL). In (j) wird, falls keine der Bedingungen (d), (h) oder (i) zutrifft, ein neuer Wert von AC konstruiert, der aus dem Minimum des bisher ermittelten Z-Wertes (CAR AC) und des Z-Wertes, der sich für den gerade ausgewerteten Teilausdruck ergab, und aus einer Liste besteht, welche neben dem Wert des gerade ausgewerteten Teilausdrucks (LIST (VAL V) (ZVAL V)) auch die Ergebnisse der bisher innerhalb der überwachten Inferenzregel ausgewerteten Teilausdrücke (CDR AC) enthält. Wird die Inferenzregel nach erfolgreicher Auswertung verlassen (d), so erfolgt mit Hilfe der ADD-Anweisung (vgl. (e)-(f)) ein Eintrag in das Inferenzgedächtnis.

Das Inferenzgedächtnis wird als FUZZY CONTEXT realisiert. Jeder FUZZY CONTEXT stellt eine separate assoziative Datenbasis dar und bietet so die Möglichkeit, unterschiedliche Teile eines Gedächtnisses zu modellieren. Durch die Anweisung (IN<CONTEXTNAME> <AUSDRUCK>) wird der Ausdruck in Bezug auf das als <CONTEXTNAME> bezeichnete assoziative Netz ausgewertet. Nach der Evaluation der Anweisung beziehen sich alle Operationen wieder auf das assoziative Netz, das vor der IN-Anweisung bearbeitet wurde. Die Einträge in den CONTEXT INFERENZGEDAECHTNIS haben folgende allgemeine Struktur:



Durch ZINSTR wird das charakteristische Pattern (CAR (GET ZNAME (QUOTE PDEF))) der überwachten Prozedur reinstantiiert. Die einzelnen Assertionen werden im Inferenzgedächtnis nach absteigenden Werten der Triftigkeit (CADR ZV) angeordnet. Jede Assertion enthält zu einem Knoten des Zielbaums alle von ihm dominierten Teilzielknoten.

Die beschriebene Lösung hat gegenüber von Systemen, in denen keine Prozedur-Dämonen zur Verfügung stehen, wie z.B. dem SHRDLU-System (vgl. WINOGRAD 1972), welches in MIKRO-PLANNER implementiert ist, den Vorteil, daß die Anweisungen zum Aufbau des Inferenzgedächtnisses (MEMORY und MEMOREND in SHRDLU, vgl. WINOGRAD 1972, S. 123/124) nicht in den einzelnen Inferenzregeln enthalten sind, sondern von einem allgemein einsetzbaren Prozedur-Dämon vorgenommen werden. Dadurch konnte die Klarheit und Übersichtlichkeit der Inferenzregeln wesentlich verbessert werden. Sich bei der Anwendung von Inferenzregeln stets wiederholende lokale Operationen werden in HAM-RPM in einen höheren Prozeß verlagert, der durch den Prozedur-Dämon ausgelöst und gesteuert wird.

Die Generierung von Begründungen. Die Beantwortung von 'Warum'-Fragen besteht in HAM-RPM aus Suchprozessen über dem im Inferenzgedächtnis gespeicherten Zielbaum, der während der Generierung der Antwort, auf die sich die 'Warum'-Frage des Benutzers bezieht, aufgebaut wurde, und einer anschließenden Transformation von Teilen des Zielbaums in eine natürlichsprachliche Oberflächenstruktur.

[20] (((D PARKZONE GETEERT) 0.5) ((T STRASSE PARKZONE) 0.9) ((D STRASSE GETEERT) 0.5)) 1). 0.8)

Nach einer entsprechenden Aufforderung des Benutzers wird z.B. die Assertion, die als Teil des Zielbaumes (Abb. 4) im Inferenzgedächtnis gespeichert ist, in die Oberflächenstruktur 'Eine Parkzone ist ein Teil einer Strasse und eine Strasse ist oft geteert' transformiert und als vorläufige Begründung für die positive Beantwortung von Frage [15] ausgegeben.

In HAM-RPM werden zu einem in Frage gestellten Zielknoten die von ihm unmittelbar dominierten Teilzielknoten, in SHRDLU und MYCIN (vgl. DAVIS, BUCHANAN & SHORTLIFFE 1977) dagegen die ihn dominierenden Teilzielknoten des Zielbaums verbalisiert. Dies ist darauf zurückzuführen, daß in HAM-RPM 'Warum'-Fragen als 'Wie begründest du diese Antwort?' in SHRDLU und MYCIN dagegen als 'Mit welchem Ziel hast du dies getan' bzw. 'Mit welchem Ziel hast du diese Frage gestellt?' interpretiert werden.

Wird nach einer Begründung für ein Teilziel gefragt, das mit einem Eintrag im Semantischen Netz identisch ist, so wird die Standardantwort 'Ist doch klar' ausgegeben. In HAM-RPM werden diejenigen Teilbäume des Zielbaums, die im Inferenzgedächtnis einen hohen Z-Wert haben, zuerst verbalisiert, so daß zunächst stets nur die besonders erwähnenswerten Inferenzen als Begründung genannt werden. Durch die Vorgabe eines Z-Wertes als untere Grenze der Triftigkeit noch aufzuführender Teilbegründungen wird es möglich, den Detaillierungsgrad der Begründung abhängig vom Dialogzustand zu variieren. Aus der vorgestellten Lösung ergibt sich für HAM-RPM eine größere Nähe zu natürlichem Dialogverhalten als für das SHRDLU-System, welches keine unterschiedlichen Detaillierungsgrade von Begründungen vorsieht, und das MYCIN-System, in dem der Benutzer durch eine optionale numerische Angabe (z.B. WARUM 2) lediglich die Schrittweite variieren kann, nach welcher beim Traversieren des Zielbaumes von unten nach oben ein Teilziel verbalisiert wird.

SCHLUSSBEMERKUNGEN

Neben der Beantwortung von 'Warum'-Fragen eröffnet das in HAM-RPM implementierte Inferenzgedächtnis weitere Möglichkeiten der Simulation einfacher Formen der 'Selbstreflexion'.

Beispielsweise können Meta-Inferenzen über den im assoziativen Netz gespeicherten Zielen einer Deduktion operieren und den Inferenzprozess so zu steuern versuchen, daß eine kombinatorische Explosion vermieden wird (vgl. auch KLEER, DOYLE, STEELE & SUSSMAN 1977).

Abschließend stellen wir fest, daß FUZZY den Programmierer u.a. von der Verarbeitung der Z-Werte, der Verwaltung einer durch CONTEXTE gegliederten assoziativen Datenbasis und expliziten Prozeduraufrufen weitgehend entlastet, so daß er sich stärker auf das 'Was' als auf das 'Wie' konzentrieren kann und eher zu aufwendigen Experimenten mit unterschiedlichen Repräsentationskonstruktionen und Kontrollstrukturen bereit ist.

Auf diese Weise wird es möglich, die Vagheit der natürlichen Sprache und des menschlichen Denkens, die wegen der Komplexität der damit verbundenen Informationsverarbeitungsprozesse bisher meist unberücksichtigt blieb, bei der Simulation kommunikativer und kognitiver Prozesse zu berücksichtigen.

Den Herren H. Boley, W. v. Hahn, W. Hoepfner, A. Jameson und H. Ueckert sei an dieser Stelle für die kritische Durchsicht des Manuskripts zu diesem Aufsatz gedankt. Die Arbeiten an HAM-RPM werden von der Deutschen Forschungsgemeinschaft (DFG) unterstützt.

LITERATUR:

- DAVIS, R., BUCHANAN, B. & SHORTLIFFE, E. (1977): Production rules as a representation for a knowledge-based consultation program. *Artificial Intelligence* 8, 15-45.
- GAINES, B. R. (1976): Foundations of fuzzy reasoning. *International Journal of Man-Machine Studies* 8, 623-668.
- GRICE, H. P. (1975): Logic and conversation. In P. Cole & J. L. Morgan (Eds.): *Syntax and semantics*. Vol. 3: Speech acts. New York: Seminar Press. 41-58.
- HERSH, H. M. & CARAMAZZA, A. (1976): A fuzzy set approach to modifiers and vagueness in natural languages. *Journal of Experimental Psychology: General* 105, 254-276.
- KINTSCH, W. (1974): *The representation of meaning in memory*. Hillsdale: Erlbaum.
- KLEER, J., DOYLE, J., STEELE, G. L. & SUSSMAN, G. J. (1977): AMORD. Explicit control of reasoning. In *Proceedings of the Symposium on Artificial Intelligence and Programming Languages*, Rochester. 116-125.
- LE FAIVRE, R. (1974): Fuzzy problem solving. Technical Report 37, Computer Science Department. Madison: University of Wisconsin.
- LE FAIVRE, R. (1977a): FUZZY reference manual. Computer Science Department. New Brunswick: Rutgers University.
- RIEGER, B. (1976): Unschärfe Semantik natürlicher Sprache. Zum Problem der Repräsentation und Analyse vager Bedeutungen. Beitrag zum Symposium 'Naturwissenschaftliche Linguistik' der Deutschen Akademie der Naturforscher Leopoldina in Halle (Erscheint in *Nova Acta Leopoldina* 1978.).
- ROSCH, E. & MERVIS, C. B. (1975): Family resemblance. *Studies in the internal structure of categories*. *Cognitive Psychology* 7, 573-605.
- SIKLÓSSY, L. (1976): *Let's talk LISP*. Englewood Cliffs: Prentice-Hall.
- VON HAHN, W., HOEPPNER, W., JAMESON, A. & WAHLSTER, W. (1978): HAM-RPM. Natural dialogues with an artificial partner. In *Proceedings of the AISB/GI Conference on Artificial Intelligence in Hamburg*, 122-131.
- WAHLSTER, W. (1977): Die Repräsentation von vagem Wissen in natürlichsprachlichen Systemen der Künstlichen Intelligenz. *Institutsbericht* 38, Institut für Informatik. Hamburg: Universität Hamburg.
- WAHLSTER, W. & VON HAHN, W. (1976): Einige Erweiterungen des natürlichsprachlichen AI-Systems HAM-RPM. In J. H. Laubsch & H. J. Schneider (Hg.): *Dialoge in natürlicher Sprache und Darstellung von Wissen*. Workshop der GI-Fachgruppe Künstliche Intelligenz. 204-225.

- WINOGRAD, T. (1972): Understanding natural language. New York: Academic Press.
- ZADEH, L. A. (1972): A fuzzy-set-theoretic interpretation of linguistic hedges. *Journal of Cybernetics* 2, 2-34.
- ZADEH, L. A. (1977a): A theory of approximate reasoning (AR). Memorandum UCB/ERL M77/58, Electronics Research Laboratory. Berkeley: University of California.
- ZADEH, L. A. (1977b): PRUF. A meaning representation language for natural languages. Memorandum UCB/ERL M77/61, Electronics Research Laboratory. Berkeley: University of California.