

Fachhochschule  
Münster University of  
Applied Sciences



# Regelbasierte Maschinenüberwachung mit Fuzzy-Logik

Schadenserkennung mit vielen Eingängen  
und wenig Regeln

Masterthesis

Alex Besstschastnich  
alex.besst@gmx.de

14. Oktober 2008

Prüfer / Betreuer  
Prof. Dr. rer. nat. Nikolaus Wulff  
Dipl.-Ing. Christian Koers

# Eidesstattliche Erklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit und die zugehörige Implementierung selbstständig verfasst und dabei nur die angegebenen Quellen und Hilfsmittel verwendet habe.

Bielefeld, 14. Oktober 2008

Alex Besstschastnich

# Danksagung

An dieser Stelle möchte ich allen danken, die mich bei der Entstehung dieser Masterarbeit und während meines Studiums in vielfacher Art und Weise unterstützt haben.

Mein besonderer Dank gilt Herrn Prof. Dr. Nikolaus Wulff, der diese Masterarbeit betreute und mir zu jeder Zeit mit hilfreichen Anregungen zur Seite stand.

Weiterhin danke ich Herrn Dipl.-Ing. Christian Koers, der mich in der Firma Prognost Systems GmbH sowohl mit konzeptionellen als auch fachlichen Hilfestellungen immer wieder unterstützt hat.

Außerdem möchte ich Herrn Dipl.-Ing. Thorsten Bojer für seine großzügige Bereitschaft zur fachlichen Diskussion danken, die mir vor allem bei der Implementierung einen schnellen Fortschritt ermöglichte.

Nicht zuletzt möchte ich mich bei meinen Eltern bedanken, die mit ihrer fortwährenden Unterstützung mir ein Studium und somit auch diese Arbeit überhaupt ermöglicht haben.

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>7</b>
<b>2</b>	<b>Fuzzy-Logik</b>	<b>9</b>
2.1	Fuzzy-Sets . . . . .	9
2.2	Zugehörigkeitsfunktionen . . . . .	10
2.2.1	Parametrisierte Darstellung . . . . .	11
2.2.2	Diskrete Darstellung . . . . .	12
2.3	Fuzzy-Elementaroperationen . . . . .	13
2.4	t-Normen und t-Conormen . . . . .	15
2.5	Fuzzy-Relationen . . . . .	18
2.6	Linguistische Variablen und Werte . . . . .	19
<b>3</b>	<b>Mittelnde Operatoren</b>	<b>21</b>
3.1	Kompensatorische Operatoren . . . . .	21
3.2	Ordered Weighted Averaging Operator . . . . .	22
3.2.1	Definition . . . . .	22
3.2.2	Orness und Entropie . . . . .	23
3.2.3	Gewichtsfunktionen mit konstantem Orness-Level . . . . .	24
<b>4</b>	<b>Regelbasierte Fuzzy-Systeme</b>	<b>25</b>
4.1	Mamdani-Controller . . . . .	26
4.1.1	Fuzzyfizierung . . . . .	27
4.1.2	Inferenzeinheit und Regelbasis . . . . .	29
4.1.3	Defuzzyfizierung . . . . .	30
4.2	Takagi-Sugeno-Controller . . . . .	32
<b>5</b>	<b>PROGNOST<sup>®</sup>-NT</b>	<b>33</b>

5.1	Struktur und Arbeitsweise . . . . .	33
5.2	Analysen und Warnschwellen . . . . .	35
5.3	Erfahrungsdatenbank . . . . .	36
5.4	Mustererkennung mit künstlichem neuronalem Netz . . . . .	38
<b>6</b>	<b>Schadenserkennung mit Fuzzy-Logik</b>	<b>39</b>
6.1	Abstraktion von PROGNOST <sup>®</sup> -NT und Vorüberlegung . . . . .	39
6.2	Design der Fuzzy-Variablen in Verbindung mit der Regelbasis . . . . .	40
6.2.1	Anzahl der Fuzzy-Sets pro Fuzzy-Variable und minimale Fuzzy- Regelbasis . . . . .	40
6.2.2	Dynamische Konstruktion der Eingangsvariablen . . . . .	42
6.2.3	Einheitliche Form der Ausgangsvariablen . . . . .	44
6.3	Untersuchung der geeigneten Operatoren . . . . .	44
6.3.1	ODER-Verknüpfung . . . . .	45
6.3.2	UND-Verknüpfung . . . . .	48
6.4	max <sub>cog</sub> -m <sub>a</sub> -Inferenz . . . . .	49
<b>7</b>	<b>Implementierung und Integration der Schadenserkennung</b>	<b>51</b>
7.1	Schnittstelle zu PROGNOST <sup>®</sup> -NT . . . . .	51
7.2	Regelbasis . . . . .	53
7.2.1	Format und Aufbau . . . . .	53
7.2.2	Überführung der XML-Regeln in die internen Strukturen . . . . .	56
7.3	Interne Datenhaltung . . . . .	58
7.4	Fuzzy-Klassen . . . . .	59
<b>8</b>	<b>Vergleich verschiedener Operatoren bei unterschiedlichen Eingangswerten</b>	<b>62</b>
8.1	Szenario und Eingangswerte . . . . .	62
8.2	Operatoren . . . . .	63
8.3	Auswertung . . . . .	64
<b>9</b>	<b>Zusammenfassung</b>	<b>69</b>
<b>A</b>	<b>Monotonie Beweis</b>	<b>71</b>
	<b>Tabellenverzeichnis</b>	<b>73</b>

## *Inhaltsverzeichnis*

---

<b>Abbildungsverzeichnis</b>	<b>74</b>
<b>Literaturverzeichnis</b>	<b>76</b>

# Kapitel 1

## Einleitung

Kolbenmaschinen wie Kolbenverdichter oder Kolbenpumpen werden u.a. in der Öl-, Gas- und chemischen Industrie eingesetzt. Solche Maschinen laufen gewöhnlich über mehrere Jahrzehnte und haben gewisse Anforderungen an die Instandhaltung. Ständiges Überwachen der Maschinen ist die Voraussetzung für eine korrekte und sichere Arbeitsweise. Vermeidung von Schäden am Mensch, Umwelt oder Maschine haben dabei höchste Priorität. Neben der Sicherheit ist es ebenfalls wichtig, maximale Maschinenverfügbarkeit und eine optimierte Arbeitsweise zu erreichen, um kostenoptimierte Produktionsprozesse und -mittel zu realisieren.

Prognost Systems bietet mit PROGNOST<sup>®</sup>-NT ein marktführendes Online-Überwachungs- und Diagnosesystem für Kolbenmaschinen an. Dieses System erfasst Maschinendaten in Echtzeit, analysiert diese und vergleicht sie mittels Mustererkennung mit realen Erfahrungen. Durch die Verschleißüberwachung und frühe Erkennung von Fehlern lassen sich Schäden vermeiden, der Wirkungsgrad optimieren und Kosten für die Instandhaltung einsparen. Die implementierte Mustererkennung arbeitet intern mit einem neuronalen Netz, welches die erfassten Signale auf bekannte Schadensmuster überprüft.

In dieser Arbeit soll ein weiterer Ansatz für die Erkennung von Mustern untersucht werden, und zwar die Mustererkennung basierend auf Fuzzy-Logik. Fuzzy-Logik scheint für diese Aufgabenstellung der Mustererkennung einen Vorteil in der Unschärfe zu haben, da neben „Muster erkannt“ und „Muster nicht erkannt“ eine weitere Klassifizierung des Erkennungsgrades, wie z.B. „Muster schwach erkannt“ oder „Muster fast erkannt“, möglich wären. Es soll untersucht werden, in wieweit eine Fuzzy-logische Verarbeitung der vorhandenen Schadensmuster mit den erfassten Daten und vorhandenem Expertenwissen möglich ist.

Mit den wichtigsten Elementen der Fuzzy-Logik beschäftigt sich das Kapitel 2. Dabei werden nach der Definition einer Fuzzy-Menge die verschiedenen Formen der Zugehörigkeitsfunktionen beschrieben. Dann geht es zu den Operatoren und Relationen zwischen Fuzzy-Mengen. Das Konzept der linguistischen Variable und die Erläuterung eines regelbasierten Fuzzy-Systems schließen den groben Überblick ab.

In Kapitel 3 werden einige mittelnde Operatoren und deren Eigenschaften vorgestellt. Diese werden auf ihre Verknüpfungseigenschaften, bezogen auf die Schadenserkennung, im Verlauf der Arbeit untersucht.

Das System PROGNOST<sup>®</sup>-NT und die aktuelle Mustererkennung werden in Kapitel 5 vorgestellt. Dabei werden die erfassten Signale erläutert und die Kriterien, anhand dieser der Zustand eines Signals ermittelt wird. Weiterhin wird die vorhandene Erfahrungsdatenbank und die Form, in der sie gespeichert ist, beschrieben. Letztendlich schließt die Erläuterung des implementierten Algorithmus zur Mustererkennung die Beschreibung des aktuellen PROGNOST<sup>®</sup>-NT Systems ab.

Anschließend wird die Theorie einer Schadenserkennung mit Fuzzy-Logik in Kapitel 6 ausführlich diskutiert. Nach den Vorüberlegungen und Zielsetzungen werden die einzelnen Komponenten des Fuzzy-Systems zur Schadenserkennung modelliert. Es wird überprüft, in wieweit ein Fuzzy-Controller mit den gegebenen Vorgaben konzipiert werden kann.

Nachdem das optimale Fuzzy-System mit Restriktionen und Erweiterungen beschrieben wurde, erfolgt eine Übersicht der Implementierung. Dabei werden Schnittstellen zu PROGNOST<sup>®</sup>-NT, gemeinsame Daten, Format und Struktur der Regelbasis, sowie die Fuzzy-Klassen beschrieben.

Das letzte Kapitel beschäftigt sich mit dem Vergleich verschiedener Verknüpfungsoperatoren. Anhand einer Beispielregel und ausgewählten Szenarien, werden die Ergebnisse unterschiedlicher Operatoren verglichen. Es soll untersucht werden, in wieweit die Operatoren auf einzelne Ausreißer in einer Regel reagieren und ob es feststellbar ist, ob Ausreißer vorhanden sind.



# Kapitel 2

## Fuzzy-Logik

Die Fuzzy-Logik stellt eine Erweiterung der klassischen binären Logik dar. Sie wurde für die Darstellung von unscharfen Wissen entwickelt und ermöglicht die Angabe über die Zugehörigkeit eines Elements zu einer Menge, anders als in der binären Logik, wo ein Element entweder zu einer Menge gehört, oder nicht. Die Idee der unscharfen Logik, also dass es einen Bereich zwischen wahr und falsch geben muss, hatte Plato (427 – 347 v. Chr.) bereits in der griechischen Antike. Allerdings postulierte sein Schüler Aristoteles (384 – 322 v. Chr.) damals das Gesetz des ausgeschlossenen Dritten, welches die Entwicklung logischer und mathematischer Systeme die nächsten zwei Jahrtausende bestimmte. Erst im 17. Jahrhundert nahmen Philosophen wie G. Hegel (1770 – 1831) und B. Russel (1872 – 1970) die Vermutung von Plato wieder auf. Zur selben Zeit führte J. Lukasiewicz (1878 – 1956) als erster eine systematische Alternative zu Aristoteles zweiwertiger Logik ein. Nachdem J. Lukasiewicz zeigte, dass es Sätze gibt, denen keiner der Wahrheitswerte „wahr“ oder „falsch“ zugeordnet werden kann, schloß er auf die Existenz eines dritten Wahrheitswertes und nannte es „possible“ (1/2). Später entwickelte er eine vier- und fünfwertige Logik und nannte auch eine Möglichkeit einer unendlichwertigen Logik. L. Zadeh (geb. 1921) veröffentlicht schließlich im Jahr 1965 den grundlegenden Artikel „Fuzzy Sets“ [14] in dem er die Mathematik der Fuzzy-Sets-Theorie beschreibt.

### 2.1 Fuzzy-Sets

Es sei  $\Omega$  eine Grundmenge. Zadeh beschreibt in [14] eine Fuzzy-Menge  $A$  einer Menge  $\Omega$  als eine Menge von geordneten Paaren

$$A = \{(x, \mu_A(x)) | x \in \Omega\} \text{ mit } \mu_A(x) : \Omega \rightarrow [0, 1]. \quad (2.1)$$

Die Funktion  $\mu(x)$  gibt den Grad an, zu dem ein Element  $x \in \Omega$  in der Menge  $A$  enthalten ist und wird als Zugehörigkeitsfunktion bezeichnet.

Ist die Grundmenge  $\Omega$  abzählbar und die Fuzzy-Menge  $A$  endlich so schreibt man in der Zadeh-Notation

$$A = \frac{\mu_A(x_1)}{x_1} + \frac{\mu_A(x_2)}{x_2} + \frac{\mu_A(x_n)}{x_n} = \sum_{i=1}^n \frac{\mu_A(x_i)}{x_i} = \sum_{i=1}^n \mu_A(x_i)/x_i. \quad (2.2)$$

Hier sei angemerkt, dass Bruch- und Schrägstrich die Paarzuordnung eines Elements  $x$  der Grundmenge  $\Omega$  mit der Zugehörigkeit  $\mu_A(x)$  zu dieser Menge beschreiben und nicht etwa eine Division, wie es in der klassischen Notation bekannt ist. Weiterhin sind die Pluszeichen als eine Aufzählung zu verstehen und nicht als eine Addition.

Die Zadeh-Notation für nicht abzählbare Fuzzy-Mengen mit einer unendlichen Grundmenge (z.B.  $\Omega = \mathbb{R}$ ) sei wie folgt definiert

$$A = \int_{\Omega} \frac{\mu_A(x)}{x} = \int_{\Omega} \mu_A(x)/x. \quad (2.3)$$

Auch hier gilt im Gegensatz zur klassischen Notation das Integral als Aufzählung unendlich vieler Wertepaare und nicht als Integration.

## 2.2 Zugehörigkeitsfunktionen

In Kapitel 2.1 wurde bereits erwähnt, dass eine Zugehörigkeitsfunktion den Grad der Zugehörigkeit eines scharfen Wertes zu einer Fuzzy-Menge angibt. Sie charakterisiert somit eine Fuzzy-Menge und dient im Allgemeinen als Werkzeug, um eine Grundmenge  $\Omega$  auf das reelle Einheitsintervall abzubilden

$$\mu : \Omega \rightarrow [0, 1]. \quad (2.4)$$

In den folgenden beiden Abschnitten werden einige formale Ansätze zur Repräsentation von Zugehörigkeitsfunktionen behandelt. Im Wesentlichen wird dabei zwischen einer parametrisierten und diskreten Darstellung unterschieden. Welche Darstellung man für die Fuzzy-Sets wählt, hängt letztendlich von dem Anwendungsfall ab.

### 2.2.1 Parametrisierte Darstellung

Bei der parametrisierten Darstellung werden Funktionen benutzt, die durch die Wahl einer oder mehrerer Parameter variiert werden können. Bei der Implementierung hat man den Vorteil, dass nur wenige Parameter gespeichert werden müssen, um trotzdem eine beliebig feine Auflösung zu bekommen. Es lässt sich somit für jedes Argument die exakte Zugehörigkeit ermitteln.

In der Regelungstechnik wird sehr häufig die dreiecksförmige Zugehörigkeitsfunktion eingesetzt, da sie einfach und effizient zu berechnen ist. Durch das vier-Tupel  $(x, m, \alpha, \beta)$  lässt sich diese vollständig festlegen

$$f_{triangle}(x, m, \alpha, \beta) = \begin{cases} 0 & ; (x \leq m - \alpha) \vee (x \geq m + \beta) \\ \frac{x - (m - \alpha)}{\alpha} & ; (m - \alpha) < x \leq m \\ 1 - \frac{x - m}{\beta} & ; m < x < (m + \beta), \end{cases} \quad (2.5)$$

wobei  $m \in \mathbb{R}$  die Mitte und  $\alpha, \beta \in \mathbb{R}^+$  die Breite der linken bzw. der rechten Seite des Dreiecks darstellen, zudem gilt  $\mu(m) = 1.0$  und  $\mu(\alpha) = \mu(\beta) = 0.0$ .

Mit Hilfe von Trapezen ist eine weitere Klasse der (einfachen) Zugehörigkeitsfunktionen konstruierbar.  $f_{trapez}$  lässt sich mit fünf Parametern wie folgt definieren:

$$f_{trapez}(x, m_1, m_2, \alpha, \beta) = \begin{cases} 0 & ; (x \leq m_1 - \alpha) \vee (x \geq m_2 + \beta) \\ \frac{x - (m_1 - \alpha)}{\alpha} & ; (m_1 - \alpha) < x \leq m_1 \\ 1 & ; m_1 \leq x \leq m_2 \\ 1 - \frac{x - m_2}{\beta} & ; m_2 < x < (m_2 + \beta). \end{cases} \quad (2.6)$$

Der Unterschied zu  $f_{triangle}$  ist, dass der Zugehörigkeitsgrad von  $f_{trapez}$  nicht nur in einem Punkt, sondern im Intervall  $[m_1, m_2]$  maximal ist. Abbildung 2.1 veranschaulicht die beiden einfachen linearen Funktionen.

Für manche Anforderungen ist es sinnvoller Funktionen mit einem Kurvenverlauf zu verwenden, um die Übergänge zwischen den Zugehörigkeiten fließend zu halten. Dafür eignen sich s- und z-förmigen Funktionen, oder auch eine Kombination dieser beiden Funktionen. Da allerdings im weiteren Verlauf der Arbeit diese keine Bedeutung ha-

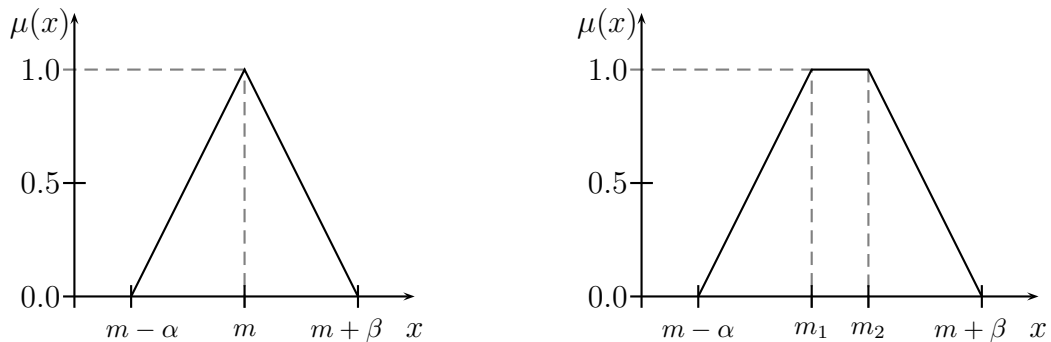


Abbildung 2.1: Zugehörigkeitsfunktionen  $f_{triangle}$  (links) und  $f_{trapez}$  (rechts)

ben, werden sie hier nicht weiter erläutert, trotzdem kann bei Interesse [2] für weitere Definitionen und Beschreibungen herangezogen werden.

## 2.2.2 Diskrete Darstellung

Da mit den parametrisierten Zugehörigkeitsfunktionen sich nicht alle beliebigen Verhaltensweisen modellieren lassen, soll hier eine weitere Möglichkeit zur Darstellung von Fuzzy-Sets beschrieben werden. Bei der diskreten Darstellung werden alle Funktionspunkte angegeben, wo sich die Steigung der Zugehörigkeitsfunktion ändert

$$A := \{(x_1, \mu_1), \dots, (x_n, \mu_n)\}, \quad (2.7)$$

wobei  $\mu_i = \mu(x_i)$  ist. Ferner wird vorausgesetzt, dass  $x_i < x_j$  für alle  $i < j$  gilt, damit gleiche x-Werte vermieden werden. Zwischen zwei benachbarten Punkten wird die Funktion durch eine Gerade approximiert. Die Berechnung eines Zugehörigkeitsgerades  $\mu(x)$  für  $x_i < x < x_{i+1}$  erfolgt mit der Formel

$$\mu(x) = \mu_{i+1} - \frac{(x_{i+1} - x)(\mu_{i+1} - \mu_i)}{x_{i+1} - x_i}, \quad (2.8)$$

unter der Berücksichtigung, dass  $\mu_{i+1} = \mu(x_{i+1})$  und  $i \in \mathbb{N}$  ist. Abbildung 2.2 zeigt ein beispielhaftes Fuzzy-Set  $A = \{(1, 0), (1.5, 0.5), (2, 0.5), (2.25, 0.25), (2.75, 0.5), (3.5, 1), (4, 1), (5, 0)\}$ , welches mit einer stückweise linearen Zugehörigkeitsfunktion beschrieben wird.

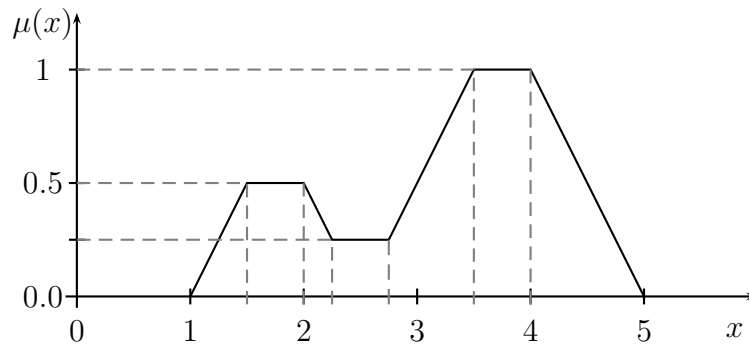


Abbildung 2.2: Beispiel eines Fuzzy-Sets in diskreter Darstellung

## 2.3 Fuzzy-Elementaroperationen

Zadeh definiert in [14] die Operationen Fuzzy-Durchschnitt, Fuzzy-Vereinigung und Fuzzy-Komplement. Diese Fuzzy-Operationen werden als Elementaroperationen bezeichnet, weil sie sich im Kern auf die Verknüpfungen Durchschnitt, Vereinigung und Komplement aus der klassischen Mengenalgebra zurückführen lassen. Die Minimalanforderung an Erweiterungen dieser klassischen Mengenoperationen ist, dass sie in den Randpunkten übereinstimmen. Weiterhin waren die von Zadeh definierten Fuzzy-Operatoren der Ausgangspunkt aller folgenden Fuzzy-Operationen, die in unterschiedlichsten Formen ausgeprägt sind.

Der Fuzzy-Durchschnitt zweier unscharfer Mengen  $A$  und  $B$ , geschrieben als  $A \cap B$ , lässt sich anhand der Zugehörigkeitsfunktionen  $\mu_A, \mu_B : \Omega \rightarrow [0, 1]$  definieren, so dass für alle  $x \in \Omega$  gilt:

$$\mu_{A \cap B}(x) = \min[\mu_A(x), \mu_B(x)]. \quad (2.9)$$

Die Abbildung 2.3 veranschaulicht den Fuzzy-Durchschnitt zweier unscharfer Mengen. Mit dem Minimum-Operator wird für alle  $x$  in der Grundmenge  $\Omega$  der kleinste Zugehörigkeitswert  $\mu(x)$  der beiden Fuzzy-Mengen ausgewählt.

Die Fuzzy-Vereinigung zweier Mengen  $A$  und  $B$ , geschrieben als  $A \cup B$  wird anhand der Zugehörigkeitsfunktionen  $\mu_A, \mu_B : \Omega \rightarrow [0, 1]$  definiert als

$$\mu_{A \cup B}(x) = \max[\mu_A(x), \mu_B(x)] \quad (2.10)$$

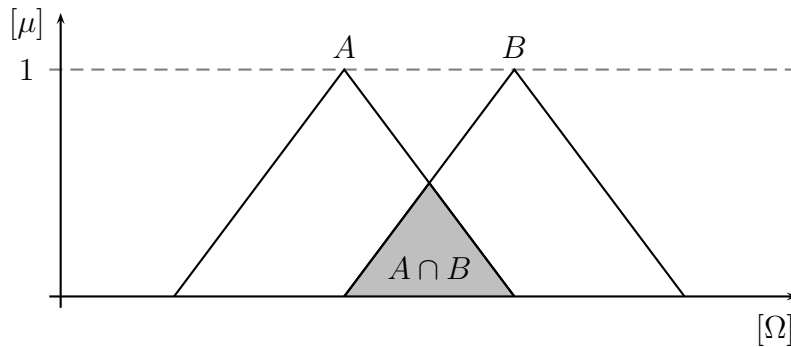


Abbildung 2.3: Darstellung des Fuzzy-Durchschnitts

mit  $x \in \Omega$ . Zur Veranschaulichung der Fuzzy-Vereinigung dient Abbildung 2.4. Im Gegensatz zu dem Fuzzy-Durchschnitt wird hier mit dem Maximum-Operator für alle  $x$  in der Grundmenge  $\Omega$  der größte Zugehörigkeitswert ausgewählt.

Sei  $\mu_A : \Omega \rightarrow [0, 1]$  die Zugehörigkeitsfunktion der unscharfen Menge  $A$ , so ist das Komplement von  $A$  die unscharfe Menge  $\bar{A}$  mit der Zugehörigkeitsfunktion  $\mu_{\bar{A}} : \Omega \rightarrow [0, 1]$ , so dass für  $x \in \Omega$  gilt:

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x). \quad (2.11)$$

Die Abbildung 2.5 zeigt eine unscharfe Menge  $A$  und das Komplement  $\bar{A}$ . Im Gegensatz zur klassischen Mengenalgebra gilt die Komplementarität nicht:

$$\begin{aligned} \text{Klassische Algebra} \quad & A \cap \bar{A} = \emptyset \\ & A \cup \bar{A} = E \end{aligned}$$

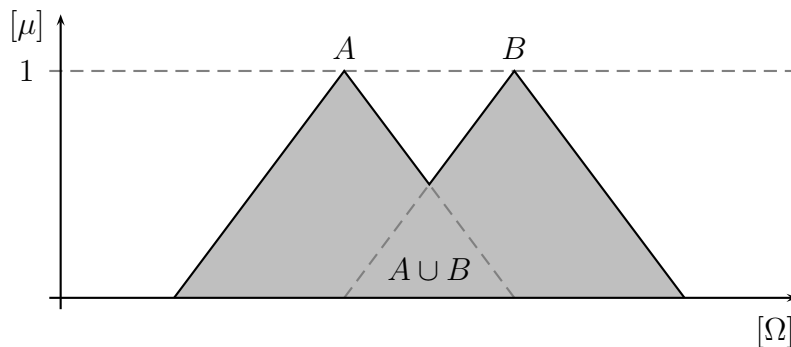


Abbildung 2.4: Darstellung der Fuzzy-Vereinigung

Fuzzy-Algebra  $A \cap \bar{A} \neq \emptyset$   
 $A \cup \bar{A} \neq E$

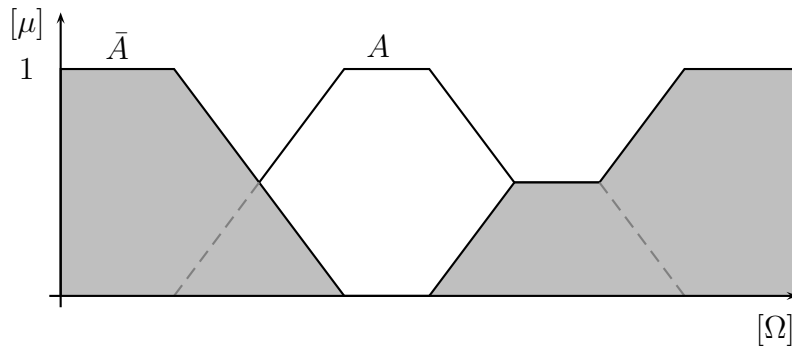


Abbildung 2.5: Darstellung des Fuzzy-Komplements

Die von Zadeh vorgeschlagenen Fuzzy-Elementaroperationen dominieren bis heute die Anwendungen der Fuzzy-Logik. Dies lässt sich zum einen mit den guten algebraischen Eigenschaften und zum anderen mit der leichten Handhabbarkeit begründen. Bis auf das Gesetz vom ausgeschlossenen Dritten (Kontradiktionsgesetz) gelten die Gesetze der klassischen Mengenalgebra<sup>1</sup> auch für die unscharfen Mengenoperatoren  $\cap_{min}$  und  $\cup_{max}$ .

## 2.4 t-Normen und t-Conormen

Neben den Fuzzy-Elementaroperationen Fuzzy-Durchschnitt und Fuzzy-Vereinigung existieren zahlreiche Modifikationen dieser beiden Grundvarianten. Es ist jedoch nicht möglich allgemeingültig zu sagen, welche konkrete UND- bzw. ODER-Variante für einen Anwendungsfall die geeignetste ist, weil die Leistungsfähigkeit eines Fuzzy-Operators allzuoft von der speziellen Problemstellung abhängt. Zum einen wird diese scheinbare Willkür und fehlende mathematische Transparenz von den Fuzzy-Gegnern kritisiert und zum anderen kann man die große Stärke solcher Operatoren sehen, da sie an die Erfahrungen der Anwender und die speziellen Problemstellung angepasst werden können.

<sup>1</sup>Detaillierte Auflistung der algebraischen Eigenschaften von Zadehs elementaren Mengenoperatoren siehe [2] auf Seite 78 Tabelle 3.1.

Nachfolgende werden die t- und s-Normen als erweiterte Fuzzy-Operationen vorgestellt und mit Beispielen erläutert. Vorweg sei erwähnt, dass die Gruppe der kompensatorischen Operatoren in einem separaten Kapitel später noch beschrieben werden.

t-Norm ist die Kurzschreibweise von Dreiecks-Norm (triangular norm) und t-Conorm die Kurzschreibweise von Dreiecks-Conorm (triangular conorm), oder auch als s-Norm bezeichnet. Es ist bekannt, dass eine vollständige Umsetzung der Eigenschaften eines Booleschen Verbands für die Dreiecks-Normen nicht möglich ist. Eine Mindestanforderung soll allerdings die wichtigsten Eigenschaften sicherstellen.

Eine t-Norm ist eine Funktion  $t : [0, 1] \times [0, 1] \rightarrow [0, 1]$ , die für alle  $a, b, c, d \in [0, 1]$  folgende Bedingungen erfüllt:

- |   |                     |
|---|---------------------|
| (1) $t(a, 1) = a$   | (neutrales Element) |
| (2) $a \leq b \wedge c \leq d \Rightarrow t(a, c) \leq t(b, d)$ | (Monotonie)         |
| (3) $t(a, b) = t(b, a)$   | (Kommutativität)    |
| (4) $t(a, t(b, c)) = t(t(a, b), c)$                             | (Assoziativität)    |

Eine t-Conorm (s-Norm) ist eine Funktion  $s : [0, 1] \times [0, 1] \rightarrow [0, 1]$ , die für alle  $a, b, c, d \in [0, 1]$  folgende Bedingungen erfüllt:

- |   |                     |
|---|---------------------|
| (1) $s(a, 0) = a$   | (neutrales Element) |
| (2) $a \leq b \wedge c \leq d \Rightarrow s(a, c) \leq s(b, d)$ | (Monotonie)         |
| (3) $s(a, b) = s(b, a)$   | (Kommutativität)    |
| (4) $s(a, s(b, c)) = s(s(a, b), c)$                             | (Assoziativität)    |

Eine wichtige Eigenschaft der t-s-Normen ist die Assoziativität, denn dadurch lassen sich die t-s-Normen rekursiv auf mehrere Argumente anwenden. Die Tabelle 2.1 veranschaulicht die typischen t- und s-Normen. Die t-Normen sind eine Modellierung der klassischen min-Operation und die s-Normen eine Modellierung der klassischen max-Operation. Das Minimum bildet die Obergrenze der t-Normen und das Maximum die Untergrenze der s-Normen, d.h. für alle  $x, y \in [0, 1]$  gilt:

$$\begin{aligned} \min(x, y) &\geq t(x, y) \\ \max(x, y) &\leq s(x, y). \end{aligned} \tag{2.12}$$



Name	t-Normen ( $\cap$ )	t-Conormen ( $\cup$ )
Zadeh	$\min(x, y)$	$\max(x, y)$
Hamacher	$\frac{xy}{x+y-xy}$ ; falls $x, y \neq 0$ 0 ; sonst	$\frac{x+y-2xy}{1-xy}$ ; falls $x, y \neq 1$ 1 ; sonst
Einstein	$\frac{xy}{1+(1-x)(1-y)}$	$\frac{x+y}{1+xy}$
Algebraisch	$xy$	$x+y-xy$
Beschränkt	$\max(0, x+y-1)$	$\min(1, x+y)$
Drastisch	$\min(x, y)$ ; falls $\max(x, y) = 1$ 0 ; sonst	$\max(x, y)$ ; falls $\min(x, y) = 0$ 1 ; sonst

Tabelle 2.1: Typische nicht-parametrisierte t- und s-Normen

Das drastische Produkt ist dagegen die Untergrenze der t-Normen und die drastische Summe die Obergrenze der s-Normen. Abbildung 2.6 verdeutlicht die Laufbereiche der beschriebenen Fuzzy-Operatoren. Im numerischen Sinne führt jede modifizierte t-Norm zur Verkleinerung des klassischen Durchschnitts und jede modifizierte s-Norm zur Vergrößerung der klassischen Vereinigung.

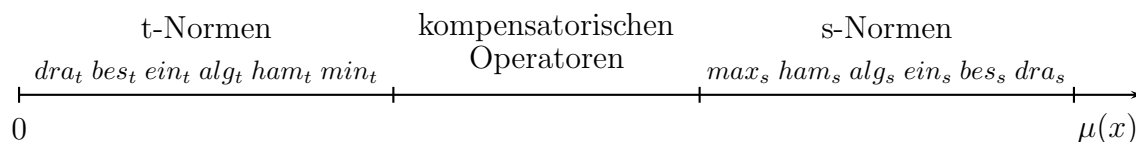


Abbildung 2.6: Laufbereiche der typischen nicht-parametrisierten t- und s-Normen

Eine noch höhere Flexibilität bieten die parametrisierten t- und s-Normen, die die selben algebraischen Eigenschaften aufweisen, wie die nicht parametrisierten t-s-Normen. Diese werden hier jedoch nicht weiter erwähnt<sup>2</sup>.

Alle t- und s-Normen beinhalten die duale Logik als Spezialfall. Werden die Zugehörigkeitsgerade auf die Werte Null und Eins eingeschränkt, so weisen sie das gleiche Verhalten wie die korrespondierenden Operatoren der klassischen Logik auf. Erst

<sup>2</sup>In [2] auf Seite 88 Tabelle 3.4 sind einige bekannte parametrisierte t- und s-Normen aufgelistet.

wenn man die Zugehörigkeitsfunktion  $\mu$  auf das Intervall  $[0, 1]$  erweitert, kommen die spezifischen Eigenschaften der Fuzzy-Operatoren zur Geltung.

## 2.5 Fuzzy-Relationen

In Kapitel 2.4 wurden Beziehungen zwischen Fuzzy-Mengen beschrieben, die auf der selben Grundmenge definiert waren. Dieses Kapitel beschreibt eine Verknüpfung von Fuzzy-Mengen, die auf unterschiedlichen Grundmengen, also unterschiedlichen physikalischen Größen, definiert sind. Eine  $n$ -stellige unscharfe Fuzzy-Relation  $R$  über den Mengen  $A_1, \dots, A_n$  mit  $n \in \mathbb{N}$ , ist eine unscharfe Teilmenge des Kartesischen Produkts der Mengen  $A_1, \dots, A_n$ .  $R$  wird über die Zugehörigkeitsfunktion

$$\mu_R : A_1 \times \dots \times A_n \rightarrow [0, 1]$$

charakterisiert, die jedem  $n$ -Tupel in  $R$  einen Zugehörigkeitsgrad  $\mu_R(a_1, \dots, a_n)$  in  $R$  zuordnet:

$$R = \int_{A_1 \times \dots \times A_n} \mu_R(a_1, \dots, a_n) / (a_1, \dots, a_n). \quad (2.13)$$

Seien nun  $A_1, \dots, A_n$   $n$  Fuzzy-Mengen auf unterschiedlichen Grundmengen  $\Omega_1, \dots, \Omega_n$ , die durch die Zugehörigkeitsfunktionen  $\mu_{A_i}(x)$  charakterisiert werden. Das Kreuzprodukt dieser  $n$  Fuzzy-Mengen wird mit Hilfe einer beliebigen  $t$ -Norm  $t$  (vgl. Kapitel 2.4) realisiert und das Verknüpfungsergebnis ist dann eine  $n$ -stellige Fuzzy-Relation  $\forall (x_1, \dots, x_n) \in \Omega_1 \times \dots \times \Omega_n$  und folgender Zugehörigkeitsfunktion:

$$\mu_R(x_1, \dots, x_n) = \mu_{\Omega_1 \times \dots \times \Omega_n}(x) = t[\mu_{A_1}(x_1), \dots, \mu_{A_n}(x_n)] \quad (2.14)$$

Alle in Kapitel 2.4 beschriebenen Operationen können also bedenkenlos auf Fuzzy-Mengen mit unterschiedlichen Grundmengen angewandt werden. Fuzzy-Relationen eignen sich gut dazu WENN-DANN-Regeln zu beschreiben, wie man später im Kapitel „Inferenzinheit und Regelbasis“ sehen wird.

## 2.6 Linguistische Variablen und Werte

In unserem Alltag werden Informationen sehr häufig in einer unscharfen Form verwendet, weil dies für den Menschen besser verständlich ist. Aussagen wie „Das Auto ist ziemlich schnell“, „Es ist sehr warm heute“ oder „Dieser Mann ist sehr alt“ sind alle unscharf, jedoch ist in vielen Situationen solch eine unscharfe Information für die Meisten aussagekräftiger, als ein konkreter Wert. Mit linguistischen Variablen lässt sich die Unschärfe von Wörtern und Ausdrücken der natürlichen Sprache darstellen. Sie dienen dazu dieses umgangssprachlich ausgedrücktes Wissen in eine formale Sprache zu übersetzen.

Analog zu den klassischen mathematischen Variablen, werden linguistischen Variablen linguistische Werte, oder auch linguistische Termen genannt, zugewiesen. In der Abbildung 2.7 wird die linguistische Variable (LV) *menschliches Alter* über die Grundmenge  $[0,100]$  dargestellt. Die linguistischen Werte (LW) lassen sich auf einer gemeinsamen scharfen Grundmenge  $X$ , hier die Menge der Jahre, definieren. Durch eine geeignete Konstruktion der Fuzzy-Sets auf der Grundmenge  $X$  wird die linguistische Variable *Alter* charakterisiert. Je mehr Fuzzy-Sets definiert werden, desto besser lassen sich die einzelnen Fuzzy-Sets von einander unterscheiden.

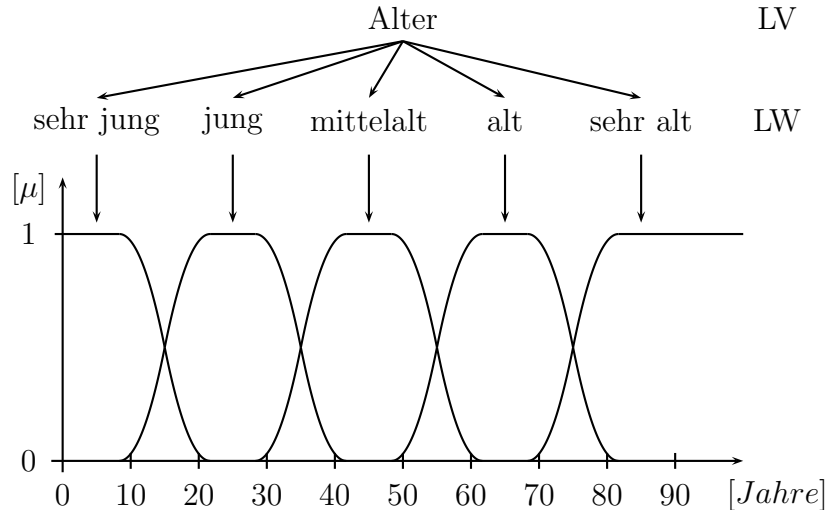


Abbildung 2.7: Linguistische Variable *menschliches Alter*

In unserem Beispiel (Bild 2.7) wird das Alter eines Menschen mit den fünf linguistischen Werten *sehr jung*, *jung*, *mittelalt*, *alt* und *sehr alt* definiert. Diese Definition der lin-

guistischen Werte wurde willkürlich gewählt und dient lediglich zur Veranschaulichung. Im Allgemeinen werden jedoch linguistische Variablen von einem Experten festgelegt, der das nötige Wissen für einen entsprechenden Sachverhalt aufweist.

# Kapitel 3

## Mittelnde Operatoren

### 3.1 Kompensatorische Operatoren

Neben der t- und s-Norm (vgl. Kapitel 2.4) für die UND- bzw. ODER-Verknüpfung gibt es noch die Möglichkeit der kompensatorischen Operatoren, die auch mittelnde Operatoren genannt werden. Diese Operatoren sind Verknüpfungen, die eine t-Norm und eine dazu geeignete s-Norm mit jeweils gleicher Gewichtung gegeneinander kompensieren.

Ein mittelnder Operator (engl. average operator) ist eine Funktion  $m : [0, 1] \times [0, 1] \rightarrow [0, 1]$ ,  $m \notin \{\min, \max\}$ , so dass für alle  $x, y \in [0, 1]$  gilt:

- (1)  $\min(x, y) \leq m(x, y) \leq \max(x, y)$  (Stabilität)
- (2)  $a \leq b \wedge c \leq d \Rightarrow m(a, c) \leq m(b, d)$  (Monotonie)
- (3)  $m(x, y) = m(y, x)$  (Kommutativität)

An der Definition fällt auf, dass die mittelnden Operatoren nicht assoziativ sind und somit die rekursive Aggregation von mehr als zwei Argumenten nicht zulassen. Um dennoch beliebig viele Fuzzy-Mengen mit einem mittelnden Operator verknüpfen zu können, müssen diese als n-stellige Funktionen  $f(x_1, \dots, x_n)$  definiert werden. Bekannte Kompensationsoperatoren, deren Laufbereiche bereits in Abbildung 2.6 veranschaulicht wurden, sind in Tabelle 3.1 aufgelistet.

Wie oben bereits erwähnt, gibt es noch die parametrisierten kompensatorischen Operatoren ( $f_\gamma$ ). Im Vergleich zu den nicht parametrisierten kompensatorischen Operatoren kann man hier den Kompensationsfaktor  $\gamma \in [0, 1]$  angeben, der eine vollständige Justierung zwischen dem Durchschnitts- und Vereinigungsoperator erlaubt. Die Kombination

Bezeichnung	Operator ( $x_i \in [0, 1], n \in \mathbb{N}$ )
Arithmetisches Mittel	$\frac{1}{n} \sum_{i=1}^n x_i$
Geometrisches Mittel	$\prod_{i=1}^n \sqrt[n]{x_i}$
Harmonisches Mittel	$n \left( \prod_{i=1}^n x_i \right) \left( \sum_{i=1}^n x_i \right)^{-1}$
Doppelt geometrisches Mittel	$1 - \prod_{i=1}^n \sqrt[n]{1 - x_i}$
Doppelt harmonisches Mittel	$1 - n \left( \prod_{i=1}^n (1 - x_i) \right) \left( \sum_{i=1}^n (1 - x_i) \right)^{-1}$

Tabelle 3.1: Nicht-parametrisierte kompensatorische Operatoren

erfolgt entweder in einer linearen Form (3.1), oder in Form einer Exponentialfunktion (3.2).

$$f_\gamma(x, y) = \gamma \cdot s(x, y) + (1 - \gamma) \cdot t(x, y) \quad (3.1)$$

$$f_\gamma(x, y) = (s(x, y))^\gamma \cdot (t(x, y))^{1-\gamma} \quad (3.2)$$

## 3.2 Ordered Weighted Averaging Operator

### 3.2.1 Definition

Ein Ordered Weighted Averaging Operator (OWA) ist eine  $n$ -stellige Abbildung  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ , die einen zugehörigen  $n$ -dimensionalen Vektor  $W = [w_1 \ w_2 \ \dots \ w_n]^T$  mit den Bedingungen

- (1)  $w_i \in [0, 1]$
- (2)  $\sum_i w_i = 1$

besitzt. Weiterhin gilt  $f(a_1, \dots, a_n) = \sum_j w_j b_j$  mit  $b_j$  als  $j$ -größtes Element von  $a_i$ . Eine wichtige Eigenschaft eines OWA Operators ist die absteigende Sortierung des Eingangsvektors.

**Beispiel 1:**

Angenommen  $W = [0.4 \ 0.3 \ 0.2 \ 0.1]$ , dann folgt  $f(0.3, 0.7, 1.0, 0.4) = (0.4)(1.0) + (0.3)(0.7) + (0.2)(0.4) + (0.1)(0.3) = 0.72$ .

OWA Operatoren unterscheiden sich nur hinsichtlich ihrer Gewichtungsfunktionen. Yager listet in [12] drei wichtige Spezialfälle der OWA Operatoren auf:

- (1)  $F^* : W = W^* = [1 \ 0 \ \dots \ 0]^T$ ,
- (2)  $F_* : W = W_* = [0 \ 0 \ \dots \ 1]^T$ ,
- (3)  $F_A : W = W_{Ave} = [1/n \ 1/n \ \dots \ 1/n]^T$ .

Weiterhin sind folgende Beziehungen offensichtlich:

$$\begin{aligned}
 F^*(a_1, \dots, a_n) &= \max_i(a_i), \\
 F_*(a_1, \dots, a_n) &= \min_i(a_i), \\
 F_A(a_1, \dots, a_n) &= \frac{1}{n} \sum_i a_i.
 \end{aligned}$$

### 3.2.2 Orness und Entropie

In [12] definiert Yager zwei wichtige Werte, die mit einem OWA Operator verknüpft sind. Der erste Wert ist die Entropie (engl. dispersion, entropy) eines OWA Vektors  $W$  und wird wie folgt definiert:

$$disp(W) = - \sum_i w_i \ln(w_i). \tag{3.3}$$

Die Entropie eines OWA Gewichtsvektors gibt Auskunft darüber, wie stark alle Eingangswerte in die Berechnung einfließen. Dabei gelten folgende Grenzfälle:

- $\exists i : w_i = 1 \Leftrightarrow disp(W) = 0$  (Minimum),
- $w_i = \frac{1}{n} \forall i \Leftrightarrow disp(W) = \ln(n)$  (Maximum).

Die maximale Entropie wird erreicht, wenn alle Eingänge gleich stark berücksichtigt werden. Der umgekehrte Fall tritt ein, wenn die Auswertung sich auf einen Wert beschränkt.

Der zweite Wert ist die Orness eines OWA Gewichtsvektors mit der Definition:

$$\text{orness}(W) = \frac{1}{n-1} \sum_{i=1}^n ((n-i)w_i). \quad (3.4)$$

Diese gibt an, wie stark der Gewichtsvektor des OWA Operators die Eingangswerte ODER-verknüpft. Die Berechnung der andness kann als  $\text{andness}(W) = 1 - \text{orness}(W)$  definiert werden. Folglich kann leicht gezeigt werden, dass folgende Zusammenhänge gelten.

- (1)  $\text{orness}(W^*) = 1$ ,
- (2)  $\text{orness}(W_*) = 0$ ,
- (3)  $\text{orness}(W_{\text{Ave}}) = 0.5$ .

### 3.2.3 Gewichtungsfunktionen mit konstantem Orness-Level

In [13] wurde eine Methode vorgestellt, wie man den Gewichtsvektor an eine vorgegebene Orness numerisch anpasst. Ahn stellte folglich in [1] Gewichtungsfunktionen vor, die eine feste Orness liefern und aber analytisch berechenbar sind. Die Gewichtungsfunktion für die konstante Orness = 3/4 sei wie folgt definiert:

$$w_i = \frac{1}{n} \sum_{j=i}^n \frac{1}{j}. \quad (3.5)$$

Eine konstante Orness von 1/4 lässt sich mit der weiteren analytischen Funktion festlegen:

$$w_i = \frac{1}{n} \sum_{j=1}^i \frac{1}{n-j+1}. \quad (3.6)$$



# Kapitel 4

## Regelbasierte Fuzzy-Systeme

Aus der Regelungstechnik ist bekannt, dass bei einem klassischen Übertragungssystem die Eingänge und die Ausgänge funktional von einander abhängen. Die Menge der Eingänge wird also über scharfe Übertragungsfunktionen auf die Menge der Ausgänge abgebildet. Für die Modellierung von gewünschten Anwendungen, wo sich kein mathematisches Modell aufstellen lässt, oder wo das mathematische Modell zu kompliziert wird, lässt sich ein regelbasiertes Fuzzy-Übertragungssystem sehr gut anwenden. Die beiden Systeme unterscheiden sich nur in der Übertragungseinheit und sind von den äußeren Anschlüssen, also die Eingänge und die Ausgänge, komplett identisch, vergleiche hierzu Abbildung 4.1. Im Gegensatz zum klassischen Übertragungssystem arbeitet das Fuzzy-Übertragungssystem mit unscharfen Mengen und linguistischen Regeln.

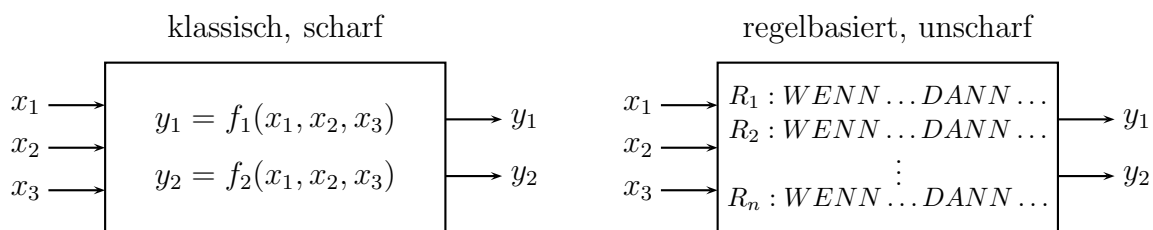


Abbildung 4.1: Komponenten eines klassischen und scharfen bzw. eines regelbasierten und unscharfen Übertragungssystems

Nachdem Zadeh bereits 1972 das Konzept der Fuzzy-Steuerung vorgestellt hatte, waren E. Mamdani und S. Assilian einige Jahre später die Ersten, die dieses Konzept auf die Steuerung einer Dampfmaschine angewandt hatten. Mit der Entwicklung des ersten Fuzzy-Reglers wurde die Möglichkeit für ein breites Anwendungsgebiet geschaffen.

Mittlerweile stößt man in sehr vielen Bereichen, z.B. Fahrzeugtechnik, Medizintechnik, Unterhaltungselektronik, Sprachbeschreibung, Automatisierungstechnik, Betriebswirtschaft und anderen auf einen Fuzzy-Controller. Die Vorstellungen für weitere mögliche Anwendung eines Fuzzy-Controllers kennen keine Grenzen.

## 4.1 Mamdani-Controller

In diesem Kapitel soll die Arbeitsweise eines Fuzzy-Controller nach E. Mamdani [9] (vgl. Abbildung 4.2) näher erläutert werden. Der Mamdani-Controller besteht im Wesentlichen aus den vier Komponenten:

- Fuzzifizierung,
- Regelbasis,
- Inferenzeinheit,
- Defuzzifizierung,

die nachfolgend näher beschrieben werden.

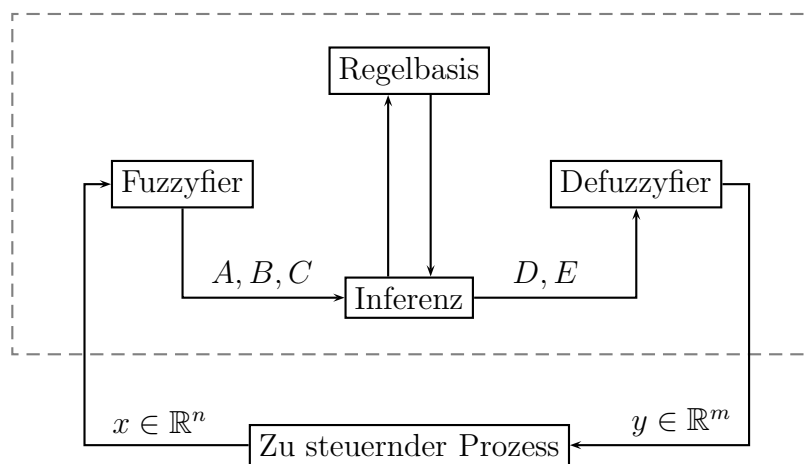


Abbildung 4.2: Fuzzy-Controller nach E. Mamdani und S. Assilian

### 4.1.1 Fuzzyfizierung

In der Fuzzyfizierung geht es darum scharfe Eingangsgrößen durch verschiedene Zuordnungen in das entsprechende Fuzzy-Pendant zu transformieren. Die Anwendung des klassischen Mengenbegriffs (Kapitel 2.1) auf die unscharfen Mengen ist bereits schon als Fuzzyfizierung zu verstehen. In Kapitel 2.4 wurden Fuzzy-Operatoren vorgestellt, die dadurch entstanden, dass man die klassischen Operatoren Vereinigung, Durchschnitt und Komplement fuzzyfizierte, also auf Fuzzy-Mengen transformierte. Umgangssprachliche Ausdrücke führten in Kapitel 2.6 zu linguistischen Variablen mit den entsprechenden linguistischen Werten. Im Allgemeinen kann man immer einen Prozess als Fuzzyfizierung bezeichnen, wenn sich die klassische Logik dieser Elemente bedient.

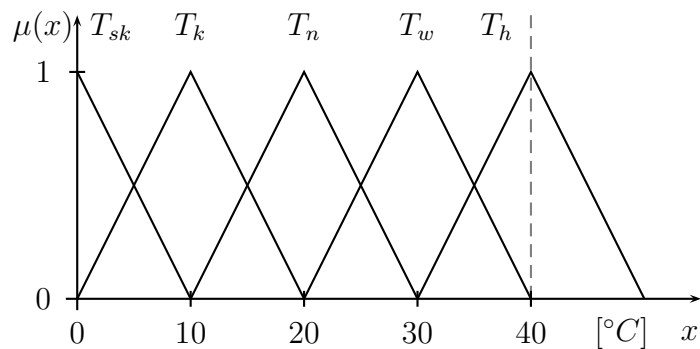


Abbildung 4.3: Linguistische Variable *Temperatur*

Im Bereich des Fuzzy-Controlls hat sich jedoch eine feste Struktur der Fuzzyfizierungseinheit etabliert, die grob in die zwei Bereiche „Definitionen“ und „Berechnungen“ aufgeteilt werden kann. Zum Einen müssen also Vereinbarungen zur Transformation getroffen werden und zum Anderen anhand dieser Vereinbarungen konkrete Werte berechnet werden. Da die Regelbasis mit linguistischen unscharfen Ausdrücken rechnet, werden in der Fuzzyfizierungseinheit hauptsächlich Definitionen bezüglich der linguistischen Variablen (LV) festgehalten, die wie folgt zusammengefasst werden können:

- Wertebereich der LV für jede Eingangs- und Ausgangsgröße
- Skalierung der Wertebereiche der Eingangs- und Ausgangsgrößen
- Anzahl der linguistischen Werte (LW) für jede LV

- Zugehörigkeitsfunktionen der LW über den Wertebereich der LV
- Überlappung der LW

Anhand der vordefinierten linguistischen Variablen werden nun Zugehörigkeitsgerade von scharfen Eingangswerten zu den einzelnen linguistischen Werten ermittelt. Das nachfolgende Beispiel soll die Funktionsweise der Fuzzyfizierung illustrieren.

**Beispiel 2:**

Es soll die Wasserzufuhr eines Gewächshauses in Abhängigkeit von der Temperatur und Feuchtigkeit geregelt werden. Die Temperatur und die relative Feuchtigkeit werden mit Hilfe von Sensoren gemessen und dem Fuzzyfizierer zugefügt. Die Grundmenge der linguistischen Variable *Temperatur* sei  $\Omega = [0, 40]$  in  $^{\circ}C$  und die linguistischen Werte sind *sehr kalt* ( $T_{sk}$ ), *kalt* ( $T_k$ ), *normal* ( $T_n$ ), *warm* ( $T_w$ ) und *heiß* ( $T_h$ ). Die konkreten Ausprägungen der Fuzzy-Sets seien wie folgt festgelegt (für die allgemeine Definition eines Fuzzy-Sets, siehe Gleichung 2.1):

$$\begin{aligned}
 T_{sk} &= \{(x, \mu_{T_{sk}}(x)) | x \in \Omega\} & \text{mit} & \quad \mu_{T_{sk}}(x) = f_{triangle}(x, 0, 10, 10) \\
 T_k &= \{(x, \mu_{T_k}(x)) | x \in \Omega\} & \text{mit} & \quad \mu_{T_k}(x) = f_{triangle}(x, 10, 10, 10) \\
 T_n &= \{(x, \mu_{T_n}(x)) | x \in \Omega\} & \text{mit} & \quad \mu_{T_n}(x) = f_{triangle}(x, 20, 10, 10) \\
 T_w &= \{(x, \mu_{T_w}(x)) | x \in \Omega\} & \text{mit} & \quad \mu_{T_w}(x) = f_{triangle}(x, 30, 10, 10) \\
 T_h &= \{(x, \mu_{T_h}(x)) | x \in \Omega\} & \text{mit} & \quad \mu_{T_h}(x) = f_{triangle}(x, 40, 10, 10)
 \end{aligned}$$

Gleichung 2.5 kann für eine vollständige Beschreibung der Funktion  $f_{triangle}$  herangezogen werden. Das Bild 4.3 demonstriert die komplette linguistische Variable *Temperatur* samt den Zugehörigkeitsfunktionen der linguistischen Werte. Die Aufstellung der Variablen *Feuchtigkeit* und *Bewässerung* erfolgt analog. So könnte die linguistische Variable *Feuchtigkeit* auf der Grundmenge  $\Gamma = [0, 100]$  in % erfasst und mit den linguistischen Werten *sehr trocken*, *trocken*, *normal*, *feucht* und *sehr feucht* definiert werden. Die linguistischen Werte *wenig*, *etwas mehr*, *normal* und *stark* auf der Grundmenge  $\Sigma = [0, 20]$  in der Einheit  $l/m^2$  könnte eine Vereinbarung für die linguistische Ausgangsvariable *Bewässerung* sein.

Das Bild 4.4 demonstriert an zwei scharfen Eingangswerten  $x = 16.7^{\circ}C$  und  $x = 30^{\circ}C$  die Funktionsweise der Fuzzyfizierung. Für  $x = 30^{\circ}C$  wird nur ein Zugehörigkeitsgrad von 1.0 zum linguistischen Wert *warm* ermittelt, die anderen linguistischen Werte haben einen Zugehörigkeitsgrad von 0.0. Beträgt die Temperatur  $16.7^{\circ}C$ , so wird der Grad der

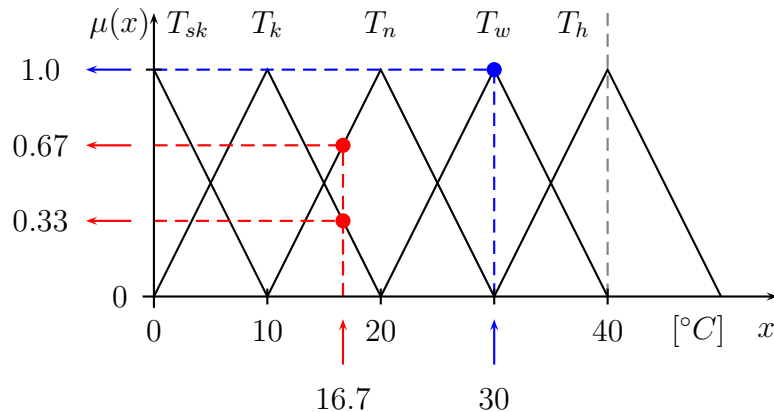


Abbildung 4.4: Fuzzyfizierung zweier beispielhafter Temperaturwerte

Zugehörigkeit 0.67 zu *normal* und 0.33 zu *kalt* ermittelt. Eine typische Schreibweise für dieses Verhalten ist:

$$x \xrightarrow{\text{Fuzzyfizierung}} (\mu_{A_1}(x), \mu_{A_2}(x), \dots, \mu_{A_n}(x)). \quad (4.1)$$

Dabei werden die Zugehörigkeitswerte  $\mu_i(x)$  von n linguistischen Variablen als geordnete Paarmenge zum scharfen Signalwert dargestellt. Auf die Beispiele bezogen heißt das:

$$\begin{aligned} x &\xrightarrow{\text{Fuzzyfizierung}} (\mu_{T_{sk}}(30), \mu_{T_k}(30), \mu_{T_n}(30), \mu_{T_w}(30), \mu_{T_h}(30)) \\ x &\xrightarrow{\text{Fuzzyfizierung}} (0, 0, 0, 1, 0) \\ x &\xrightarrow{\text{Fuzzyfizierung}} (\mu_{T_{sk}}(16.7), \mu_{T_k}(16.7), \mu_{T_n}(16.7), \mu_{T_w}(16.7), \mu_{T_h}(16.7)) \\ x &\xrightarrow{\text{Fuzzyfizierung}} (0, 0.33, 0.67, 0, 0) \end{aligned}$$

### 4.1.2 Inferenzeinheit und Regelbasis

Die Inferenzeinheit ist für die Auswertung der fuzzyfizierten Werte verantwortlich. Die linguistischen Variablen aus der Fuzzyfizierung werden hier anhand WENN-DANN-Regeln, die von einem Experten definiert sind, verknüpft. Ein vollständiger Satz von Regeln bildet die Regelbasis, die dauerhaft in der Inferenzeinheit abgelegt wird. Die Regelbasis repräsentiert also die Wissensbasis in Form der WENN-DANN-Regeln. Ziel der regelbasierten Inferenz ist es nach Mamdani eine unscharfe Ausgangsgröße zu erzeugen, die wiederum an die Defuzzyfizierung weitergeleitet wird.

Eine beliebige Regel verfügt über  $n$  Eingangsvariablen  $x_1, \dots, x_n$  und einer Ausgangsvariable  $y$ . Jede dieser Variablen wird durch eine linguistische Variable repräsentiert, deren Ausprägungen die Fuzzy-Sets  $A_i$  bzw.  $B_k$  sind. Jede beliebige Regel lässt sich mit folgender Form beschreiben:

$$R_j: \text{WENN } x_1 = A_1 \text{ UND } \dots \text{ UND } x_n = A_n \text{ DANN } y_k = B_k.$$

Anstelle der UND-Verknüpfung lassen sich die einzelnen Prämissen auch ODER-verknüpfen. Sei  $G_{P_i} \in [0, 1]$  der Erfüllungsgrad oder auch Zugehörigkeitsgrad einer Prämisse, der bei der Fuzzyifizierung eines scharfen Wertes zu einem linguistischen Wert ermittelt wird. Weiterhin sei  $G_{R_i} \in [0, 1]$  der Erfüllungsgrad einer ganzen Regel. So lässt sich das max-t-Inferenzschema für eine Regelbasis mit  $n$  disjunktiv verknüpften Regeln wie folgt berechnen:

$$\mu_{B'}(y) = \max_{j=1, \dots, n} [t(G_{R_j}, \mu_{B_j}(y))], \quad (4.2)$$

wobei bei  $k$  UND-verknüpften Prämissen  $G_{R_i} = t(G_{P_1}, \dots, G_{P_k})$  und bei  $k$  ODER-verknüpften Prämissen  $G_{R_i} = s(G_{P_1}, \dots, G_{P_k})$  gilt.

### 4.1.3 Defuzzifizierung

Die Einheit Defuzzifizierung sorgt dafür, dass die Fuzzy-Ausgangsmenge aus der Inferenzeinheit in einen scharfen Wert umgewandelt wird. Die bekanntesten Methoden zur Defuzzifizierung sind Maximum (Max-Height), Mean-of-Max (MoM) und Center-of-Gravity (CoG). Nachfolgend werden die drei Defuzzifizierungsmethoden näher behandelt.

#### Maximum-Methode (Max)

Die Maximum-Methode ist wie folgt definiert:

$$\mu(x_s) = \max_{x \in \Omega} (\mu(x)). \quad (4.3)$$

Der scharfe Ausgangswert  $x_s$  entspricht hier der Stelle der Ausgangsmenge  $\Omega$  mit dem größten Zugehörigkeitsgrad. Dieses mathematisch einfache Defuzzifizierungsverfahren lässt sich allerdings nur dann sinnvoll anwenden, wenn man davon ausgehen kann, dass nur ein Maximum im Ausgangs-Fuzzy-Set vorkommt. Sollte trotzdem das Maximum

nicht eindeutig bestimmbar sein, hat man die Möglichkeit das linke oder das rechte Maximum zu wählen.

### Maximum-Mittelwert-Methode (MoM)

Eine Weiterentwicklung der Maximum-Methode ist die Mean-of-Max-Methode. Diese soll den Entscheidungskonflikt, der bei mehreren Maxima unmittelbar auftauchen wird, dadurch umgehen, dass in einem solchen Fall das arithmetische Mittel aller x-Werte mit maximaler Zugehörigkeit gebildet wird

$$x_s = \frac{1}{n} \sum_{i=1}^n x_i \quad \text{mit :} \quad \mu(x_i) = \max_{x \in \Omega} (\mu(x)). \quad (4.4)$$

Wenn die Maximalstelle in Form eines Intervalls vorliegt, so ergibt sich die scharfe Ausgangsgröße  $x_s$  aus dem Mittelwert der beiden Intervallgrenzen

$$x_s = \frac{x_{rechts} - x_{links}}{2} \quad \text{mit :} \quad x_{rechts}, x_{links} \quad \text{Intervallgrenzen.} \quad (4.5)$$

### Flächenschwerpunkt-Methode (CoG)

Eine weitere und am häufigsten angewandte Methode ist die Flächenschwerpunkt-Methode oder auch Center-of-Gravity genannt. Diese Methode basiert auf der Ermittlung des statischen Momentes eines Massepunktes. Durch die Gleichung

$$x_s = \frac{\int_a^b x \mu(x) dx}{\int_a^b \mu(x) dx} \quad (4.6)$$

wird der Flächenschwerpunkt eines Fuzzy-Sets berechnet. Da uns nur der scharfe Ausgangswert auf der Abszisse interessiert, ist die y-Koordinate  $y_s$  des Schwerpunktes in diesem Zusammenhang nicht weiter von Bedeutung. Abbildung 4.5 stellt die drei bekanntesten Defuzzifizierungsmethoden gegenüber.

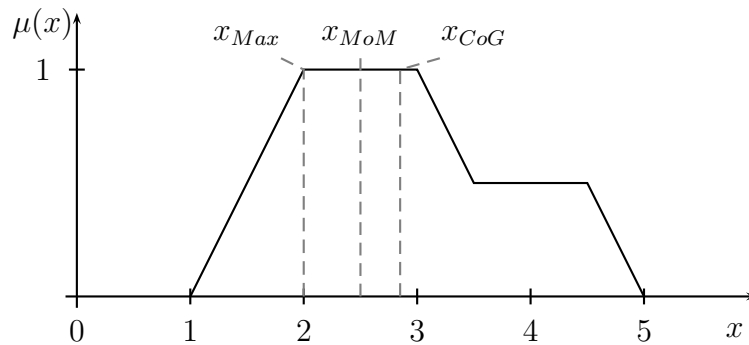


Abbildung 4.5: Defuzzifizierungsmethoden Max, MoM und CoG

## 4.2 Takagi-Sugeno-Controller

Eine weitere Variante für den Aufbau eines regelbasierten Fuzzy-Systems beschreiben Sugeno und Takagi in [8], nachfolgend einfach nur Sugeno-Controller genannt. Dieser Controller ist eine Modifikation des Mamdani-Controllers mit dem Hauptunterschied, dass unter Umständen die Defuzzifizierung wegfällt.

Im DANN-Teil einer Regel nach Sugeno und Takagi wird demnach keine Fuzzy-Menge mehr bestimmt, sondern direkt ein scharfer Wert. Die Regeln haben dann die Form

$$R_j: \text{WENN } x_1 = A_1 \text{ UND } \dots \text{ UND } x_n = A_n \text{ DANN } y_j = f_j(x_1, \dots, x_n).$$

Der entscheidende Unterschied zum Mamdani-Controller an dieser Stelle ist, dass keine linguistischen Regeln verarbeitet werden, sondern funktionale Zusammenhänge der scharfen Eingangsgrößen  $x_1, \dots, x_n$ .

Die Auswertung des WENN-Teils erfolgt genau so wie im Mamdani-Controller. Ferner sei hier ebenfalls mit  $G_{Ri} \in [0, 1]$  der Grad, zu dem die WENN-Bedingung erfüllt ist, bezeichnet. Die scharfe Ausgangsgröße ergibt sich nun als mit den Erfüllungsgraden  $G_{Rj}$  gewichtete Summe der einzelnen Regelausgaben  $f_j$  zu

$$y = \frac{\sum_{j=1}^n G_{Rj} \cdot f_j(x_1, \dots, x_n)}{\sum_{j=1}^n G_{Rj}}. \quad (4.7)$$

Der Nachteil dieses Systems ist die notwendige Festlegung der crisen Funktionen  $f_j$ , was einen Verständnis des Ein- und Ausgabeverhaltens im Gesamtsystem impliziert.



# Kapitel 5

## PROGNOST<sup>®</sup>-NT

PROGNOST<sup>®</sup>-NT [3] ist ein komplexes System, das zur Zustandsüberwachung von Kolbenmaschinen eingesetzt wird. Das System ist zwanzig Jahren im Markt und mittlerweile marktführend. Es beinhaltet sowohl Hard- als auch Softwarekomponenten. Mit PROGNOST<sup>®</sup>-NT werden Signale erfasst, ausgewertet und interpretiert, somit können Kolbenmaschinen sowohl überwacht als auch optimiert werden. Eventuell auftretende Schäden durch Verschleiß, oder sonstige Fehler, lassen sich mit PROGNOST<sup>®</sup>-NT frühzeitig erkennen und beheben, wodurch die Verfügbarkeit der Maschinen enorm erhöht wird. Weiterhin werden durch vermiedene oder geplante Ausfälle hohe Kosten gespart, da die Information von PROGNOST<sup>®</sup>-NT in die Instandhaltung der Maschinen einfließt. Kunden von Prognost sind Weltkonzerne der Öl-, Gas- und Chemieindustrie.

In diesem Kapitel soll ein grober Überblick über das System PROGNOST<sup>®</sup>-NT gegeben werden. Spezielle Bereiche, die für die Mustererkennung von Bedeutung sind, werden in den einzelnen Unterkapiteln näher erläutert.

### 5.1 Struktur und Arbeitsweise

Prognost installiert verschiedene Sensoren auf den Maschinen, um möglichst viele Signale (Temperaturen, Schwingungen, Drücke usw.) zu erfassen. Die Signale werden dann an den Schaltschrank von Prognost übermittelt und digitalisiert. Das PROGNOST<sup>®</sup>-NT Base System analysiert die erfassten Daten und speichert diese dauerhaft ab. Die Prognost Visu ist eine grafische Oberfläche, die als Schnittstelle zwischen der Maschine und dem Maschinenführer fungiert. Wie der Name schon sagt, werden mittels der Prognost Visu (vgl. Bild 5.1) sämtliche Daten der Maschinen visualisiert. Der Maschinenführer

kann jeder Zeit den Status einer Maschine einsehen und wird über alle Störungen, Fehler oder erkannte Schäden informiert.

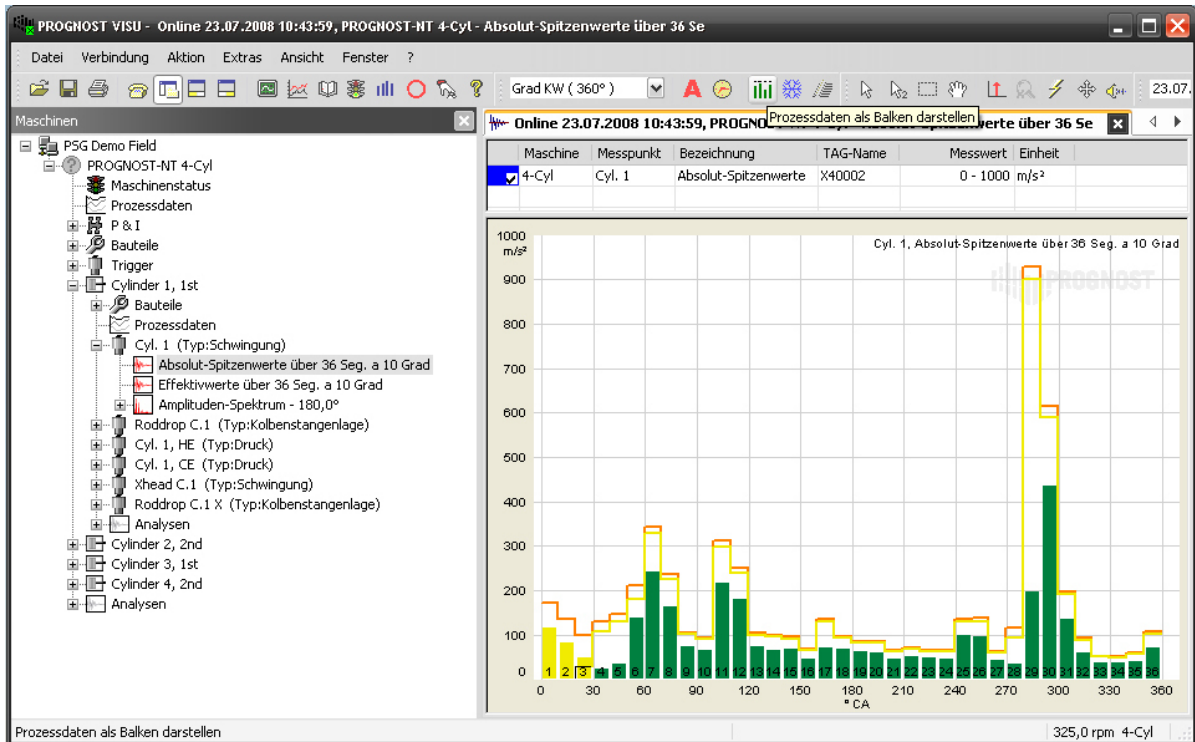


Abbildung 5.1: Visualisierung der überwachten Maschinen mit der PROGNO<sup>®</sup>-NT Visu

Parallel zu der Datenerfassung werden sämtliche Signalmuster der eingehenden Signale mit einer Musterdatenbank verglichen. Wird eine Übereinstimmung erkannt, so erfolgt eine detaillierte Meldung über den erkannten Schadensfall. Die Schadensmuster werden von erfahrenen Experten definiert. Jedes Schadensmuster enthält eine Menge von Analysen und den Verletzungsgrad (vgl. Kapitel 5.2) jeder Analyse. Der Abgleich zwischen den aktuellen Daten und den Schadensmustern erfolgt zur Zeit mit Hilfe eines künstlichen neuronalen Netzes, siehe Kapitel 5.4.

Das System PROGNO<sup>®</sup>-NT bietet viele weitere Features zur Überwachung von Kolbenmaschinen, auf die hier jedoch nicht weiter eingegangen wird, da diese für das Verständnis der Schadenserkennung mit Fuzzy-Logik nicht von Belang sind.

## 5.2 Analysen und Warnschwellen

Prognost verarbeitet sowohl sich statisch ändernde PLS-Daten (Prozessleitsystem [11]) als auch Daten von dynamischen Sensoren, die zusätzlich an die Maschine angebracht werden. Die PLS-Daten werden ca. 1/s aufgefrischt. Die dynamischen Signale werden mit 25kHz abgetastet und in Kurbelwellensegmente zu je 10° Kurbelwinkel aufgeteilt. Für einige Analysen ist die Aufteilung in 8 Segmente zu je 45° Kurbelwinkel oder die Analyse über eine ganze Umdrehung (1 Segment) der Kurbelwelle sinnvoller. Jedes Segment wird individuell analysiert (z.B. RMS, Peak-to-Peak, Mittelwert).

Damit eine Überprüfung auf Verletzung möglich ist, werden alle Segmente mit Warnschwellen versehen. Insgesamt gibt es pro Segment vier Warnschwellen:

- Zweite obere Warnschwelle (2.OWS)
- Erste obere Warnschwelle (1.OWS)
- Erste untere Warnschwelle (1.UWS)
- Zweite untere Warnschwelle (2.UWS)

Weiterhin gibt es pro Segment einen unteren (URP) und oberen Referenzpunkt (ORP). Diese Punkte legen den Bereich fest, in dem sich ein Signal im Gut-Zustand befindet.

Bei der Berechnung der Referenzpunkte wird eine Zeitspanne betrachtet, in der sich das Signal in einem normalen Zustand befindet. In diesem Signaltrend wird der Mittelwert, das absolute Maximum und das absolute Minimum pro Minute gespeichert. Die Dichtefunktion der absoluten Maxima bzw. Minima des Signals, entspricht einer Gauß-Verteilung. Seien nun  $\mu_{max}$  und  $\mu_{min}$  die Erwartungswerte und  $\sigma_{max}$  bzw.  $\sigma_{min}$  die Standardabweichungen der jeweiligen Dichtefunktionen, so werden die Referenzpunkte für  $x \in [1, 3]$  mit

$$\text{ORP} = \mu_{max} + x \cdot \sigma_{max}$$

$$\text{URP} = \mu_{min} - x \cdot \sigma_{min}$$

und die Warnschwellen mit

$$2.OWS = ORP + 2 \cdot x \cdot \sigma_{max}$$

$$1.OWS = ORP + x \cdot \sigma_{max}$$

$$1.UWS = URP - x \cdot \sigma_{min}$$

$$2.UWS = URP - 2 \cdot x \cdot \sigma_{min}$$

berechnet. Abbildung 5.2 visualisiert symbolisch die berechneten Parameter eines Signals. Anhand dieser Parameter lassen sich die Zustände der einzelnen Signale klassifizieren.

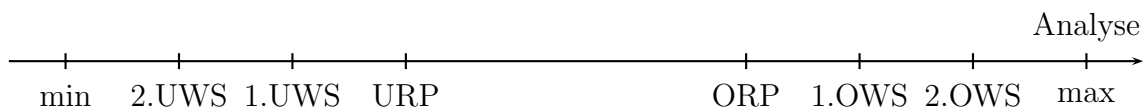


Abbildung 5.2: Parameter einer Analyse

### 5.3 Erfahrungsdatenbank

Die Musterdatenbank ist das Herzstück des PROGNOST<sup>®</sup>-NT Systems. Sie enthält Schadensmuster, die auf konkrete Schadensfälle hinweisen. Die Schadensmuster werden seit Nutzung des Systems kontinuierlich gesammelt und aktualisiert. Mittlerweile wurden über zwanzig Schadensklassen definiert, die sich aus ca. 200 Schadensmustern zusammensetzen.

Ein Schadensmuster beinhaltet eine Menge von Analysen und die zugehörigen Warnschwellen, die verletzt sein müssen. Bei mehrsegmentigen Analysen ist zusätzlich die Angabe der Segmente erforderlich. In der Abbildung 5.3 sind drei unterschiedliche Schadensmuster aus der Datenbank zu sehen. Gelb steht für erste obere, rot für zweite obere, hellblau für erste untere und dunkelblau für zweite untere Warnschwelle.

Ende 2006 wurden von den Experten wieder neue Schadensmuster definiert. Die neuen Schadensmuster sind komplexer und stellen teilweise hohe Anforderung an die Mustererkennung (vgl. Kapitel 5.4). Die wesentlichen neuen Anforderungen sind:



Durchstichpunkt durch den Saugdruck KS:	mind. erste untere WS
<b>UND</b>	
(	
Polytrophenexponent 1 Reexpansion KS:	mind. erste obere WS
<b>ODER</b>	
Polytrophenexponent 2 Reexpansion KS:	mind. erste obere WS
)	

## 5.4 Mustererkennung mit künstlichem neuronalem Netz

Die Überprüfung der aktuellen Signale auf vorhandene Schadensmuster geschieht mit Hilfe eines künstlichen neuronalen Netzes und dem Dynamic Generalized Relevance Learning Vector Quantization (DGRLVQ) Algorithmus. Der DGRLVQ [4] ist eine erweiterte Version des GRLVQ, welcher in [6] auf das Monitoring von technischen Systemen untersucht wurde.

Allgemein lassen sich mit einem LVQ nichtlineare Probleme klassifizieren. Bezogen auf die Mustererkennung in PROGNOT<sup>®</sup>-NT heißt es, dass mehrere Musterklassen existieren, anhand derer die Signale quantisiert werden. Es werden ausschließlich Signale einbezogen, die bereits eine Warnschwelle verletzen. Bevor allerdings der LVQ-Algorithmus auf verletzte Signale angewandt werden kann, muss ein Codebook bestehend aus Codebook-Vektoren kreiert werden. Das Codebook wird erstellt, wenn man das künstliche neuronale Netz mit den vorhandenen Schadensmustern, die von Experten definiert sind, trainiert.

Bezogen auf die neu definierten Schadensmuster sind einige Anforderungen mit der aktuellen Implementierung des neuronalen Netzes garnicht, bzw. umständlich umsetzbar. Diese sind sowohl die Berücksichtigung einer dynamischen Segmentberechnung, als auch die Umsetzung von Schadensmustern, die explizit keine Verletzung einer Warnschwelle als Eingang erwarten. Der erste Punkt lässt sich mit mehr oder weniger Aufwand softwaretechnisch umsetzen, der zweite Punkt ist dagegen sehr schwer, bzw. garnicht umsetzbar, da das neuronale Netz ausschließlich mit Signalen arbeitet, die bereits eine Warschwelle überschritten haben.

Im nachfolgenden Kapitel wird ein System vorgestellt, welches mit Unschärfe arbeitet und somit neue Sichtweiten der Schadenserkennung bei Maschinen öffnet.

# Kapitel 6

## Schadenserkennung mit Fuzzy-Logik

Nachdem die Grundlagen der Fuzzy-Logik in Kapitel 2 beschrieben und das System Prognost-NT in Kapitel 5 vorgestellt wurden, sollen in diesem Kapitel die Komponenten eines unscharfen Inferenzsystems zur Überwachung von Maschinen erläutert werden. Dabei werden die vorhandenen Expertenregeln auf den Erfüllungsgrad untersucht (erfüllt, teilweise erfüllt, nicht erfüllt). Folglich werden unscharfe Variablen modelliert und die Verknüpfungsoperatoren für *UND* und *ODER* analysiert.

### 6.1 Abstraktion von PROGNOST<sup>®</sup>-NT und Vorüberlegung

Um die erarbeiteten Konzepte möglichst allgemein zu halten, soll in den nachfolgenden Kapiteln nur noch ein abstraktes Modell einer Schadenserkennung, welches hier näher erläutert wird, betrachtet werden.

Im Allgemeinen geht es darum, aus einer Vielzahl unterschiedlicher Eingänge und einer regelbasierten Schadensdatenbank, die von Experten definiert worden ist, vorzeitig einen Schaden an einer Maschine zu erkennen, bzw. den Verletzungsgrad eines möglichen Schadens zu ermitteln.

Die mehr als tausend Analysen mit verschiedenen physikalischen Größen werden mit definierten Warnschwellen überwacht. Zusätzlich zu den Warnschwellen gibt es einen oberen und unteren Referenzpunkt, die den Gut-Bereich des Signals markieren. Für detaillierte Berechnungen der Grenzen wird auf das Kapitel 5.2 verwiesen.

Die Regeln aus der Erfahrungsdatenbank beziehen sich auf die gemessenen Signale und eine der vier Warnschwellen. Dabei kann sowohl die Verletzung einer Warnschwelle,

als auch explizit keine Verletzung einer Warnschwelle als Prämisse in einer Regel mit dem zugehörigen Signal stehen. Eine Expertenregel beschreibt genau einen Schaden. Die einzelnen Datensätze innerhalb einer Regel können beliebig *UND* und *ODER* verknüpft werden. Als Ergebnis des gesamten Systems sollen die Verletzungsgrade der einzelnen Schadensmuster ausgegeben werden.

Der entscheidende Unterschied zwischen einer Schadenserkenkung mit einem neuronalen Netz und Fuzzy-Logik ist, dass der Fuzzy-Controller für die Auswertung der Schadensregeln alle Signale in beliebigen Verletzungsgraden berücksichtigt, wobei das neuronale Netz nur Signale in die Berechnung einbezieht, die bereits eine Warnschwellenverletzung haben. D.h. das neuronale Netz gibt Auskunft darüber wie gut das Schadensmuster aus allen verletzten Signalen erkannt worden ist, wobei das Fuzzy-System eine Aussage darüber liefert, wie stark die akumulierte Verletzung genau der im Schadensmuster geforderten Signale ist.

## 6.2 Design der Fuzzy-Variablen in Verbindung mit der Regelbasis

In diesem Kapitel wird untersucht, inwieweit die scharfen Signale und Regeln auf eine unscharfe Inferenz erweiterbar sind und welche Restriktionen dazu notwendig sind. Zuerst wird die mögliche Anzahl der Fuzzy-Sets pro Fuzzy-Variable untersucht, sowie eine unscharfe Regelbasis aufgestellt. Folglich wird beschrieben, wie die Fuzzy-Variablen abhängig von den Warnschwellen und Referenzpunkten dynamisch konstruiert werden. Zuletzt erfolgt eine Beschreibung der Ausgangsvariablen und der Ausprägung der einzelnen Fuzzy-Sets.

### 6.2.1 Anzahl der Fuzzy-Sets pro Fuzzy-Variable und minimale Fuzzy-Regelbasis

Wie bereits in Kapitel 5.3 erwähnt, beziehen sich die vorhandenen Expertenregeln zur Erkennung von Schadensfällen auf scharfe Signale. D.h. eine Prämisse ist demnach entweder erfüllt oder nicht erfüllt. Dies bedeutet wiederum, dass die Anzahl der Fuzzy-Sets bei der Modellierung einer Fuzzy-Variable, minimal gehalten werden muss, da keine definierten Fuzzy-Regeln für Zwischenwerte von den Experten existieren.



Bei der Verknüpfung von zwei Fuzzy-Variablen bestehend aus je  $n$  Fuzzy-Sets gibt es insgesamt  $2^n$  mögliche Regeln. In unserem Fall sind allerdings nur  $n$  Regeln bekannt, nämlich da, wo sich die Fuzzy-Sets  $s_i$  und  $s_j$  der Fuzzy-Variablen  $A$  und  $B$  mit  $i = j$  überlappen. Tabelle 6.1 zeigt die möglichen Regeln in Abhängigkeit von der Anzahl der Fuzzy-Sets.

		A			
		$s_1$	$s_2$	$\dots$	$s_n$
B	$s_1$	×			
	$s_2$		×		
	$\vdots$			$\ddots$	
	$s_n$				×

Tabelle 6.1: Bekannte Regeln in Abhängigkeit von der Anzahl der Fuzzy-Sets  $n$

Da sich die fehlenden Regeln, bei linear steigender Anzahl der Fuzzy-Sets, quadratisch erhöhen, bleibt nur die Möglichkeit die Fuzzy-Variablen mit je zwei Fuzzy-Sets („verletzt“ und „nicht verletzt“) zu konstruieren. Weiterhin sind bei der Verknüpfung zweier Fuzzy-Variablen auch nur zwei Regeln bekannt. Diese Konstrukt entspricht genau dem natural logic controller, der als eine vereinfachte Version des Mamdani-Controllers von Aceves-López und Aguilar-Martin in [7] beschrieben wurde.

Die beiden Fuzzy-Sets, aus denen nun eine Fuzzy-Variable besteht, sollen  $v$  und  $n$  heißen.  $v$  steht für „verletzt“ und repräsentiert die Verletzung einer bestimmten Warnschwelle.  $n$  ist das Komplement zu  $v$  und beschreibt den Normalzustand eines Signals. An dieser Stelle ist zu beachten, dass der Normalzustand abhängig von der Regel zu interpretieren ist, d.h. wenn die Prämisse einer Regel „A keine 1.OWS“ lautet, dann befindet sich der Normalzustand des Signals  $A$  im Bereich der 1.OWS und nicht um den ORP.

Die zwei Fuzzy-Regeln bei einer Verknüpfung von zwei Fuzzy-Variablen lassen sich auch auf beliebig viele Eingänge erweitern. Sei  $m$  die Anzahl der Eingangsvariablen  $A_i$

mit  $i = 1, \dots, m$  und  $B$  die Ausgangsvariable, so lässt sich die Fuzzy-Regelbasis mit den beiden Regeln

$$\begin{aligned} \text{WENN } A_1 = n \text{ UND } A_2 = n \text{ UND } \dots \text{ UND } A_n = n \text{ DANN } B = n \\ \text{WENN } A_1 = v \text{ UND } A_2 = v \text{ UND } \dots \text{ UND } A_n = v \text{ DANN } B = v \end{aligned}$$

beschreiben. Diese beiden Regeln entsprechen der ersten und der letzten Regel einer vollständigen Regelbasis und stellen dabei die beiden Extremfälle dar.

Nachfolgend wird beschrieben, wie die Zugehörigkeitsfunktionen der beiden Fuzzy-Sets abhängig von den Warnschwellen und Referenzpunkten ermittelt werden.

### 6.2.2 Dynamische Konstruktion der Eingangsvariablen

Im Allgemeinen lässt sich der Definitionsbereich einer Analyse in die vier ausschlaggebenden Intervalle  $[2.UWS,URP]$ ,  $[1.UWS,URP]$ ,  $[ORP,1.OWS]$  und  $[ORP,2.OWS]$  einteilen. Abhängig von der Warnschwelle, die mit einem Signal in einer Regel als Prämisse vorkommt, werden die Zugehörigkeitsfunktionen der Fuzzy-Sets  $n$  und  $v$  der zugehörigen Fuzzy-Variable berechnet.

Wird in einer Regel ein Signal auf eine der beiden oberen Warnschwellen überprüft, so lässt sich folgende Definition für die Zugehörigkeitsfunktion von  $v$  angeben:

$$\mu_v(x) = \begin{cases} 0 & ; x \leq ORP \\ (x - ORP)/(OWS - ORP) & ; ORP < x < OWS \\ 1 & ; x \geq OWS \end{cases}$$

$OWS$  ist hierbei entweder erste oder zweite obere Warnschwelle. Die Zugehörigkeitsfunktion von  $n$  ist  $\mu_n(x) = 1 - \mu_v(x)$ , da  $n$  das Komplement von  $v$  ist. Die Berechnung

der Zugehörigkeitsfunktion von  $v$ , bei der Überprüfung auf eine der beiden unteren Warnschwellen, lässt sich mit der linearen Abbildung

$$\mu_v(x) = \begin{cases} 1 & ; x \leq UWS \\ (URP - x)/(URP - UWS) & ; UWS < x < URP \\ 0 & ; x \geq URP \end{cases}$$

festlegen. Analog zur obigen Gleichung, steht  $UWS$  für eine der beiden unteren Warnschwellen. Wird explizit keine Verletzung einer oberen Warnschwelle in einem Schadensmuster erwartet, so lässt sich das Fuzzy-Set  $v$  mit der Zugehörigkeitsfunktion

$$\mu_v(x) = \begin{cases} 1 & ; x \leq ORP \\ (OWS - x)/(OWS - ORP) & ; ORP < x < OWS \\ 0 & ; x \geq OWS \end{cases}$$

beschreiben. Die letzte Zugehörigkeitsfunktion wird angewandt, wenn explizit keine Verletzung einer der beiden unteren Warnschwellen erforderlich ist:

$$\mu_v(x) = \begin{cases} 0 & ; x \leq UWS \\ (x - UWS)/(URP - UWS) & ; UWS < x < URP \\ 1 & ; x \geq URP \end{cases}$$

Abbildung 6.1 veranschaulicht die dynamische Konstruktion einer Fuzzy-Eingangsvariable mit den Fuzzy-Sets  $n$  und  $v$  nach einer oberen Warnschwelle.

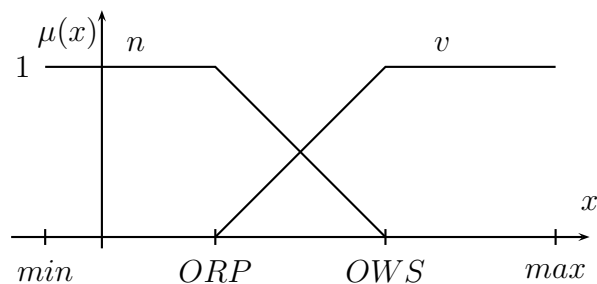


Abbildung 6.1: Dynamische Konstruktion einer Fuzzy-Eingangsvariable nach einer oberen Warnschwelle

### 6.2.3 Einheitliche Form der Ausgangsvariablen

Wie bereits oben erwähnt, soll als Ergebnis einer Regelauswertung der Verletzungsgrad ausgegeben werden. Da die Konklusionen aller Regeln jeweils die selben Eigenschaften haben, wird eine einheitliche Ausgangsvariable für alle Schadensregeln verwendet. Dementsprechend wird die Ausgangsvariable so konstruiert, dass alle Defuzzifizierungswerte im Einheitsintervall liegen. Da weiterhin nur zwei Regeln einer Verknüpfung von beliebig vielen Eingangsvariablen vorhanden sind, besteht die Ausgangsvariable ebenfalls nur aus zwei Fuzzy-Sets  $v$  und  $n$ .  $v$  repräsentiert dabei die Verletzung eines ganzen Schadens und  $n$  beschreibt den Normalzustand eines Schadens.

Die Zugehörigkeitsfunktionen  $\mu_n(y)$  und  $\mu_v(y)$  der Fuzzy-Sets werden mit Hilfe der dreiecksförmigen Funktion  $f_{triangle}$  aus Gleichung 2.5 auf Seite 11 wie folgt dargestellt:

$$\begin{aligned}\mu_n(y) &= f_{triangle}(y, 0, 1, 1) \\ \mu_v(y) &= f_{triangle}(y, 1, 1, 1).\end{aligned}$$

An dieser Stelle ist zu beachten, dass  $n$  nicht das Komplement von  $v$  ist. Abbildung 6.2 veranschaulicht die einheitliche Form der Fuzzy-Ausgangsvariablen.

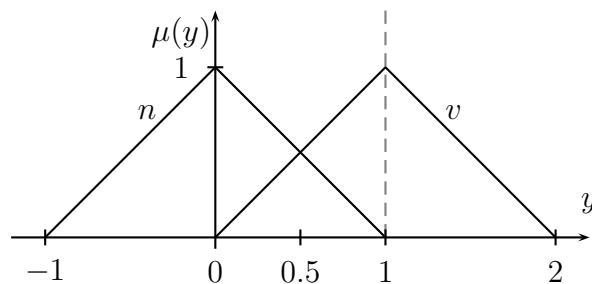


Abbildung 6.2: Einheitliche Form der Fuzzy-Ausgangsvariablen

## 6.3 Untersuchung der geeigneten Operatoren

Nachdem unscharfe Regeln für unscharfe Variablen konzipiert worden sind, geht es in diesem Kapitel um die konkrete Realisierung der UND- und ODER-Verknüpfung. Dabei werden die Nachteile der t-s-Normen in diesem Fuzzy-System genannt und Operatoren vorgestellt, die erwartungsgemäße Ergebnisse liefern.

### 6.3.1 ODER-Verknüpfung

Bevor die eigentliche Realisierung der ODER-Verknüpfung erläutert wird, soll vorerst eine ODER-Verknüpfung im Sinne einer Expertenregel diskutiert werden, die einen Schaden an einer Maschine beschreibt.

In Kapitel 5.3 wurden Expertenregeln vorgestellt, die mit unterschiedlichen UND- und ODER-verknüpften Signalen, einen Schaden an einer Maschine beschreiben. Die Regeln sind zudem in Form eines minimalen logischen Ausdrucks definiert. Überführt man eine Regel  $R$  in die disjunktive Normalform, so bekommt man Disjunktionen von Konjunktionstermen  $R_i$ , womit sich jede Regel als  $R = R_1 \vee \dots \vee R_n$  beschreiben lässt. Die Erwartung der Experten an das Ergebnis disjunktiv verknüpfter Konjunktionsterme, ist der maximal erfüllte Konjunktionsterm. Dies bedeutet, dass die Konjunktionsterme nicht miteinander verrechnet, sondern einzeln ausgewertet werden müssen.

#### Beispiel 3:

Es sei die scharfe Expertenregel  $R$  in Abbildung 6.3 in Form eines Regelbaumes gegeben. Nach dem Auswerten des Regelbaumes und der Überführung in die disjunktive Normalform

$$\begin{aligned} R &= A \wedge (C \vee D) \wedge B \wedge (E \vee F) \\ &= (ABC \vee ABD)(E \vee F) \\ &= ABCE \vee ABCF \vee ABDE \vee ABDF, \end{aligned}$$

hat die Regel die Form:

$$R = R_1 \vee R_2 \vee R_3 \vee R_4$$

Fuzzy-logisch betrachtet, ist jeder Konjunktionsterm eine unscharfe Menge und repräsentiert bereits ein Teilergebnis der gesamten Regel. D.h. das Ergebnis eines Konjunktionsterms ist die Fuzzy-Ausgangsvariable  $A_i$  mit den Erfüllungsgraden der Fuzzy-Sets  $n$  und  $v$ . Üblicherweise bilden jedoch die ODER-verknüpften Teilergebnisse letztendlich das vollständige Ergebnis. In Betracht der Tatsachen, dass im vorliegenden System eine separate Auswertung der Teilergebnisse gefordert ist, wird die Auswertung einer Regel wie folgt definiert:

$$x_{cog} = \max_i[cog(A_i)] = \max_i[x_{cog_i}], \quad (6.1)$$

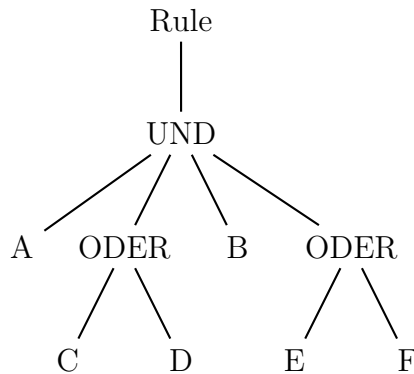


Abbildung 6.3: Beispiel eines Regelbaumes mit sechs Eingängen

mit  $A_i$  i-tes unscharfe Teilergebnis (vgl. Abbildung 6.2) einer Regel und CoG (vgl. Kapitel 4.1.3) als Defuzzifizierungsmethode.

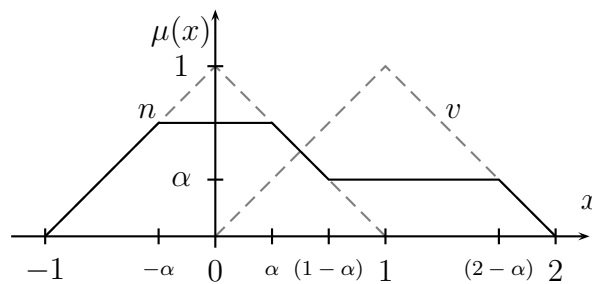


Abbildung 6.4: Defuzzifizierung der unscharfen Ausgangsvariable

Der Flächenschwerpunkt einer unscharfen Mänge lässt sich allgemein mit der Gleichung 4.6 berechnen. Für die Berechnung des CoG der Fuzzy-Variable aus Abbildung 6.4 kann man allerdings eine vereinfachte Form angeben. Hierzu sei  $\alpha$  der Erfüllungsgrad von  $v$  und  $\beta = 1 - \alpha$  der Erfüllungsgrad von  $n$ . Das Integral der Vereinigung (Nenner) der beiden Fuzzy-Sets  $A := A^{<\alpha} \cup A^{<\beta}$  lässt sich somit auf die Form

$$\begin{aligned}
 N(x_{cog}) = & \int_{-1}^{-\alpha} (1+x)dx + (1-\alpha) \int_{-\alpha}^{\alpha} dx + \int_{1-\alpha}^{\alpha} (1-x)dx \\
 & + \alpha \int_{1-\alpha}^{2-\alpha} dx + \int_{2-\alpha}^{\alpha} (2-x)dx = \alpha(1-\alpha) + 1
 \end{aligned} \tag{6.2}$$

vereinfachen. Der nicht normalisierte Erwartungswert (Zähler) ergibt sich durch:

$$\begin{aligned}
 Z(x_{cog}) &= \int_{-1}^{-\alpha} x(1+x)dx + (1-\alpha) \int_{-\alpha}^{\alpha} xdx + \int_{1-\alpha}^{\alpha} x(1-x)dx \\
 &+ \alpha \int_{1-\alpha}^{2-\alpha} xdx + \int_{2-\alpha}^{\alpha} x(2-x)dx = \frac{1}{2}\alpha(3-\alpha).
 \end{aligned} \tag{6.3}$$

Zusammengesetzt lässt sich der defuzzyfizierte Center of Gravity, abhängig von dem Verletzungsgrad  $\alpha$  des Fuzzy-Sets  $v$ , wie folgt analytisch berechnen:

$$x_{cog}(\alpha) = \frac{1}{2} \frac{\alpha(3-\alpha)}{\alpha(1-\alpha) + 1}. \tag{6.4}$$

An dieser Stelle ist zu beachten, dass die Gleichung 6.4 nur aufgrund der speziellen Eigenschaft  $\beta = 1 - \alpha$  abgeleitet werden kann.

Setzt man nun die Funktion  $x_{cog}(\alpha)$  in die Gleichung 6.1 ein, so lässt sich das Ergebnis einer ganzen Regel leicht berechnen. Da  $x_{cog}(\alpha)$  streng monoton steigend<sup>1</sup> ist, wird eine weitere Vereinfachung durchgeführt, indem zuerst das Maximum der einzelnen Verletzungsgrade  $\alpha_i$  ermittelt und dann erst der Center of Gravity berechnet wird. Somit sind die beiden folgenden Funktionen äquivalent:

$$\max_i [x_{cog}(\alpha_i)] \Leftrightarrow x_{cog}(\max_i [\alpha_i]). \tag{6.5}$$

Durch Gleichung 6.5 wird die mehrfache Berechnung der Flächenschwerpunkte der einzelnen Fuzzy-Variablen, abhängig vom Erfüllungsgrad  $\alpha$ , auf ein Mal reduziert. Stattdessen muss nur noch das Maximum der  $\alpha_i$  ermittelt werden.

Durch die obige Konstruktion der Ausgangsvariable lässt sich des Defuzzyfizierungsverhalten mit dem Erfüllungsgrad  $\alpha$  des Sets  $v$  vergleichen, siehe Abbildung 6.5 a). In dieser Abbildung ist zu sehen, dass  $x_{cog}$  nah bei  $\alpha$  liegt und in Abbildung 6.5 b) wird die absolute Differenz  $|x_{cog} - \alpha| \leq 0.041$  veranschaulicht. Dieses s-förmige Verhalten hängt mit dem Fuzzy-Set  $n$  zusammen.

---

<sup>1</sup>Beweis siehe Anhang.

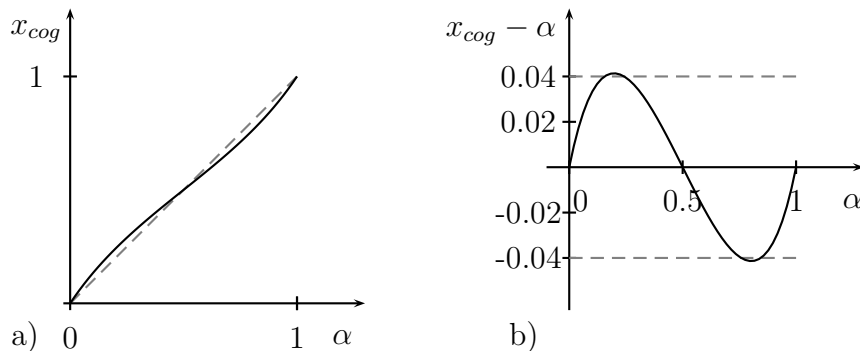


Abbildung 6.5: a) Erfüllungsgrad  $\alpha$  im Vergleich zum resultierenden  $x_{cog}$ , b) Differenz  $x_{cog} - \alpha$

### 6.3.2 UND-Verknüpfung

In üblichen Fuzzy-Controllern werden t-Normen für konjunktiv verknüpfte Prämissen verwendet, allerdings machen diese erst Sinn, wenn die Regelbasis ausreichend gefüllt ist. Wenn die Regelbasis, wie in unserem Fall, nur aus zwei Regeln besteht, können mit einer t-Norm heftige Sprünge in der Inferenz verursacht werden, da für alle t-Normen  $t(a, 0) = 0$  mit  $a \in [0, 1]$  gilt. Bezogen auf die Schadenserkenkung, würde es heißen, dass eine einzige Prämisse ausreicht, um eine ganze Schadensregel als nicht erfüllt zu bezeichnen. Offensichtlich sind t-Normen in diesem Fall nicht für eine weiche Inferenz geeignet.

Durch das Einsetzen eines OWA Operators aus Kapitel 3 können die Sprünge, die durch die mangelnden Fuzzy-Regeln resultieren, aufgeweicht werden. Allerdings wird die Wahl eines optimalen OWA Operators durch die Expertenvorgaben beschränkt. Sei eine Regel aus fünf konjunktiv verknüpften Prämissen gegeben, wo vier von diesen Prämissen eine Verletzung von 100% haben und eine von 0%. Die Experten erwarten in solch einem Fall einen Erkennungsgrad von 80% der gesamten Regel.

Das Beispiel überführt auf eine Verknüpfung mit einem OWA Operator bedeutet, dass an erster Stelle eine maximale Entropie gefordert wird. Dies impliziert jedoch eine Gleichgewichtung der Eingangswerte mit  $\frac{1}{n}$  und einer  $orness(W) = 0.5$ . Durch die gleiche Gewichtung wird zudem die Sortierung der Eingangswerte nach Größe überflüssig, so dass sich die UND-Verknüpfung zu einem arithmetischen Mittel aus Tabelle 3.1 vereinfachen lässt.



Seien  $A, B$  zwei Fuzzy-Variablen auf den Grundmengen  $\Omega_1$  und  $\Omega_2$  mit der in Kapitel 6.2.2 beschriebenen Ausprägungen. Ferner sei der Referenzpunkt bei 0 und die Warnschwelle bei 1. Abbildung 6.6 zeigt in der linken Grafik für  $x_1 \in \Omega_1$  und  $x_2 \in \Omega_2$  die UND-Verknüpfung der beiden Variablen  $A, B$  im Intervall  $[0,1]$  mit einer t-Norm und rechts mit dem arithmetischen Mittel. Die vertikale Achse repräsentiert dabei den Center of Gravity der Ausgangsvariable, die die Form aus Abbildung 6.2 hat.

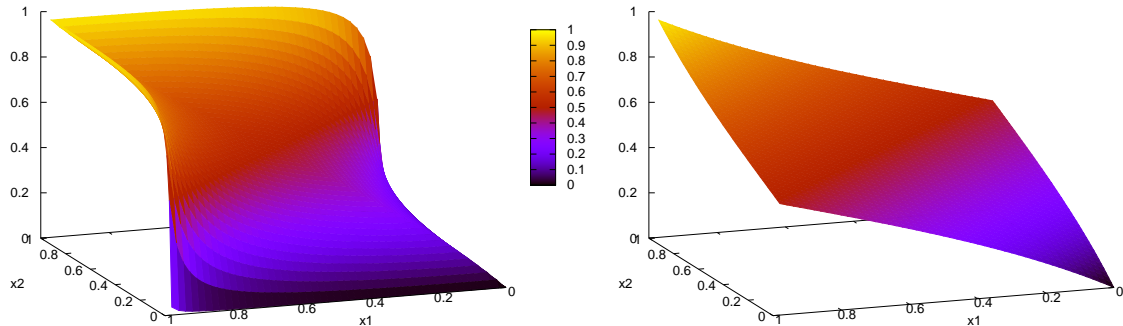


Abbildung 6.6: Zwei Fuzzy-Variablen mit je zwei Sets UND-verknüpft: mit der min-Norm (links) und mit dem arithmetischen Mittel (rechts)

## 6.4 $\max_{\text{cog}}\text{-}m_a\text{-Inferenz}$

Nachdem die optimale Umsetzung einer UND- und ODER-Verknüpfung beschrieben wurden, soll die Verwendung dieser beiden Methoden als  $\max_{\text{cog}}\text{-}m_a\text{-Inferenz}$  bezeichnet werden. Sie wird durch die folgenden zusammengefassten Eigenschaften charakterisiert:

- Überführung der Fuzzy-Regeln in die disjunktive Normalform
- Das Ergebnis eines Konjunktionsterms ist der arithmetische Mittelwert der Erfüllungsgrade der einzelnen Fuzzy-Sets
- Das Ergebnis der gesamten Regel ist der maximale Center of Gravity der einzelnen Konjunktionsterme

In der nachfolgenden Abbildung 6.7 wird die  $\max_{\text{cog}}\text{-}m_a\text{-Inferenz}$  (1) mit der  $\max\text{-}m_a\text{-Inferenz}$  (2) anhand der beiden möglichen Regeln

$$\begin{aligned} &\text{WENN } (A = n \text{ UND } B = n) \text{ ODER } (C = n \text{ UND } D = n) \text{ DANN } E = n \\ &\text{WENN } (A = v \text{ UND } B = v) \text{ ODER } (C = v \text{ UND } D = v) \text{ DANN } E = v \end{aligned}$$

verglichen. Die unscharfen Eingangsvariablen  $A, B, C, D$  und Ausgangsvariable  $E$  haben die in Kapitel 6.2.3 und 6.2.2 beschriebene Form.

$E_1$  und  $E_2$  sind die Konjunktionsterme und repräsentieren bereits die Teilergebnisse. Die Ergebnisse beider Inferenzen spiegeln deutlich die unterschiedliche Arbeitsweise. Da in  $E_1$  das Fuzzy-Set  $v$  und in  $E_2$  das Fuzzy-Set  $n$  einen relativ hohen Erfüllungsgrad haben, werden durch die Maximumbildung hohe Erfüllungsgrade der beiden Sets im Ausgang  $E'$  erreicht, was einen CoG von ca. 0.5 ergibt.

Da in (1) das maximale Teilergebnis gleich das Gesamtergebnis darstellt, in diesem Fall  $E = E_1$ , werden die stark erfüllten Sets der unterschiedlichen Variablen garnicht miteinander verknüpft. Dies führt zu einem höheren CoG und dem entsprechend zu einem höheren Verletzungsgrad der Schadensregel.

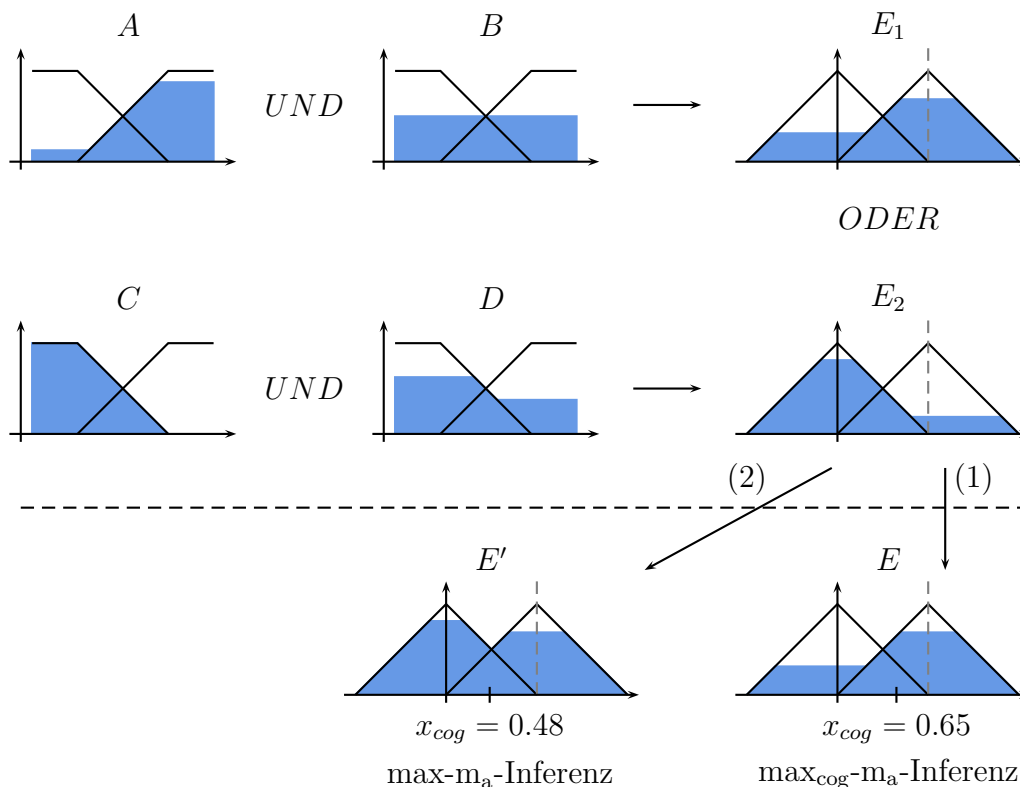


Abbildung 6.7: Vergleich max- $m_a$ -Inferenz mit der max<sub>cog</sub>- $m_a$ -Inferenz

# Kapitel 7

## Implementierung und Integration der Schadenserkennung

In diesem Kapitel wird ein kurzer Überblick über die Schnittstelle zu PROGNOST<sup>®</sup>-NT und die Implementierung der Schadenserkennung mit Fuzzy-Logik gegeben. Dabei wird zuerst auf die gemeinsame Schnittstelle eingegangen und ferner die Realisierung und das Format der Regelbasis erläutert. Danach folgt die Beschreibung der internen Datenhaltung, der Fuzzy-Klassen sowie des Fuzzy-Controllers. Bei den nachfolgenden Klassendiagrammen ist zu beachten, dass der Einfachheit halber nicht alle Funktionen und Attribute dargestellt werden.

### 7.1 Schnittstelle zu PROGNOST<sup>®</sup>-NT

Die Schnittstelle zu PROGNOST<sup>®</sup>-NT wird insgesamt durch die drei Klassen *IPatternFuzzy*, *IPatternFuzzyResult* und *IPatternFuzzyClass* beschrieben. Diese drei Klassen repräsentieren jeweils ein Interface, wie im Klassendiagramm der Abbildung 7.1 zu sehen.

*IPatternFuzzy* beschreibt die eigentliche Schnittstelle zu PROGNOST<sup>®</sup>-NT, welche sowohl für den Eingang der Daten als auch für die Ausgabe der Schäden samt Verletzungsgrad zuständig ist. Mit der Funktion *Init()* werden dem Fuzzy-System alle benötigten Daten über die vorhandenen Analysen der überwachten Maschine übergeben. Diese Daten beinhalten folgende Informationen einer Analyse:

- Ort
- Segmentierung

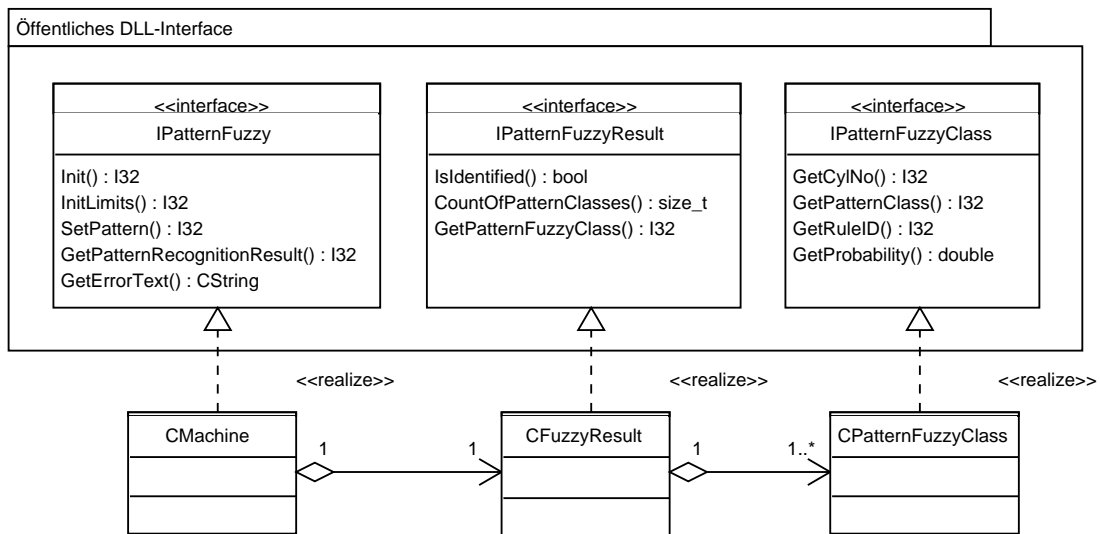


Abbildung 7.1: Klassendiagramm der Schnittstelle zu PROGNOST<sup>®</sup>-NT

- Wertebereich
- Sensortyp
- Speciestyp

Mit der Methode *InitLimits()* werden nachfolgend die eingestellten Warnschwellen jeder Analyse an das Fuzzy-System übergeben. Die Trennung der Analyseinformationen von den Warnschwellen hat den Grund, dass bei einer Änderung der Warnschwellen nicht gleich die ganzen Informationen neu geparkt werden müssen. Dies spart Zeit und kostbare Operationen.

Nachdem die komplette Initialisierung erfolgreich war, wird *SetPattern()* aufgerufen, um der Fuzzy-Schadenserkenkung die aktuell gemessenen Werte der Analysen mitzuteilen. Zusätzlich zu den Werten ist die Übergabe eines minimalen Erkennungsgrads möglich, der die Schwelle darstellt, ab welcher Verletzung die Ausgabe einer Schadenserkenkung wirksam wird. Nachdem die aktuellen Messwerte übergeben wurden, beginnt in der selben Funktion die Auswertung der Schadensmuster.

Mit *GetPatternRecognitionResult()* werden die Ergebnisse des aktuellen Zyklus abgefragt. Der Rückgabewert ist eine Instanz der Schnittstelle *IPatternFuzzyResult*. *IPatternFuzzyResult* ist ein Container für Instanzen der Klassen *IPatternFuzzyClass*. Diese wiederum beschreibt einen identifizierten Schaden mit einem Erkennungsgrad und einer eindeutigen RuleID.

Der Datentyp *I32* im obigen Diagramm repräsentiert einen Integer, der die feste Größe von 4 Byte hat, da es vorkommen kann, dass die Größe einer *int*-Variable von System zu System abweicht. Um also auf allen Systemen einheitliche Integergrößen zu realisieren, werden *I32* und *I16* verwendet.

## 7.2 Regelbasis

### 7.2.1 Format und Aufbau

Die komplette Regelbasis, mit den bis dato bekannten Schadensmustern, wird in einer XML-Datei gespeichert. Zu dieser XML-Datei existiert ein XML-Schema (XSD), welches die Struktur und die Syntax der Regelbasis beschreibt. Weiterhin wird die Regelbasis vor der Verwendung anhand des XML-Schemas validiert, um eventuell vorhandene Unstimmigkeiten vorzeitig zu erkennen.

Das XML-Dokument besteht aus einem *RuleBase*-Element, welches die Wurzel darstellt. Es muss mindestens ein Kindelement *Rule* vorhanden sein, welches eine Regel repräsentiert. Jedes *Rule*-Element wird mit den Attributen aus Tabelle 7.1 parametrisiert.

Attribut	Beschreibung
<i>name</i>	Regelname
<i>patternClass</i>	Schadensklasse der Regel, da es möglich ist, dass mehrere Regeln die selbe Schadensklasse beschreiben
<i>type</i>	Regelgranularität (pro Zylinder, pro Kurbel, pro Maschine)
<i>validityThreshold</i>	Individueller Verletzungsgrad der Regel
<i>ruleId</i>	Eindeutige Regelnummer
<i>validFor</i>	Maschinentyp oder -gruppe, für die diese Regel gilt

Tabelle 7.1: Attribute des *Rule*-Elements in XML

Weiterhin lässt sich die Regelgültigkeit mit den vier optionalen Elementen *dependsOnOneOf*, *dependsOnAllOf*, *deniedByOneOf* und *deniedByAllOf*, die direkt unterhalb des *Rule*-Elements platziert werden, aufgrund vorhandener oder nicht vorhandener Analysen

einschränken. Somit lassen sich Regeln modellieren, die z.B. speziell bei abgehobenen Ventilen aktiviert oder deaktiviert werden können.

Nach der Festlegung der Eigenschaften beginnt die eigentliche Struktur einer Regeln. Diese lässt sich mit den Elementen *and*, *or* und *DataSpecies* kreieren. Unter dem *Rule*-Element kann nur eins von diesen Elementen als Kind vorkommen. Jedes *and*-Element kann beliebig viele *or*- und *DataSpecies*-Elemente als Kinder haben, allerdings keine weiteren *and*-Elemente. Analog dazu können unter einem *or*-Element beliebig viele *and*- und *DataSpecies*-Elemente auftreten, jedoch keine weiteren *or*-Elemente. Somit lässt sich der Regelbaum bequem in dem XML-Dokument abbilden.

Das Element *DataSpecies* definiert mit den Attributen *type* und *where* den Typ und die Position einer Analyse. Mit *Position* ist ein Sensor an der Maschine gemeint. Die unterschiedlichen Typen können u.a. Schwingungssignale, Drucksignale oder PLS-Werte sein. Weiterhin wird die Analyse bei dem gesetzten optionalen Attribut *ignoreable* ignoriert, falls diese Analyse an der überwachten Maschine nicht vorhanden ist. Ist *ignoreable* nicht gesetzt und die Analyse fehlt, so erfolgt eine Löschung der Regel oder wenn möglich, nur eines Teilbaumes der Regel.

Unterhalb eines *DataSpecies*-Elements wird die Analyse spezifiziert. Die möglichen Unterelemente von *DataSpecies* sind in Tabelle 7.2 aufgelistet. Jedes dieser Elemente enthält die Attribute *threshold* und *weight*. Mit *threshold* wird die Warnschwelle festgelegt, nach der eine Analyse ausgewertet werden soll und *weight* lässt sich ein individuelles Gewicht zu einer Analyse definieren, welches dann bei der Regelauswertung berücksichtigt wird.

<b>Kindelement</b>	<b>Beschreibung</b>
<i>Threshold</i>	Einsegmentige Analyse
<i>Segment8</i>	Achtsegmentige Analyse
<i>Segment36</i>	36-segmentige Analyse
<i>Segment8Offset</i>	Achtsegmentige Analyse, Segmentangabe erfolgt relativ zu einer anderen Analyse
<i>Segment36Offset</i>	36-segmentige Analyse, Segmentangabe erfolgt relativ zu einer anderen Analyse

Tabelle 7.2: Kindelemente des *DataSpecies*-Elements

Die Elemente *Segment8* und *Segment36* haben zusätzlich ein *no*-Attribut, welches die Nummer des Segments angibt. Mit den drei Attributen *offset*, *relativeToType* und *relativeToPosition* werden in den Elementen *Segment8Offset* und *Segment36Offset* Angaben bezüglich der relativen Segmentberechnung gemacht.

*relativeToType* ist dem *type*-Attribut und *relativeToPosition* dem *where*-Attribut im *DataSpecies*-Element gleichzusetzen. Sie bestimmen eindeutig die Analyse, anhand derer die relative Segmentberechnung erfolgen soll. Die Einheit der relativen Analyse sollte Gradkurbelwinkel sein, damit dieser einem Segment zugeordnet werden kann, auf welches sich die relative Berechnung mit dem Offset stützt, anderenfalls ist eine Offsetberechnung nicht möglich.

An dieser Stelle ist zu beachten, dass die Werte der Attribute *validFor*, *type*, *where*, *relativeToType*, *relativeToPosition* und die Werte der Elemente *dependsOnOneOf*, *dependsOnAllOf*, *deniedByOneOf*, *deniedByAllOf* exakt mit den Makros aus den Header-Dateien *Prognost-Defines.h* und *Prognost-Datatypes.h* übereinstimmen müssen, damit eine korrekte Arbeitsweise gewährleistet ist. Alle anderen möglichen Werte der verschiedenen Attribute sind im XML-Schema definiert (XSD).

Das nachfolgende Beispiel veranschaulicht mit den oben erklärten Elementen und Attributen die Struktur einer beispielhaften Regel im XML-Dokument.

#### Beispiel 4:

```
<RuleBase>
  <Rule name=" Beispielregel"
    patternClass="1"
    type=" perCylinder"
    ValidityThreshold="0.8"
    ruleId="10"
    validFor="KOLBENMASCHINEN">
    <and>
      <DataSpecies type="DT.ENDDRUCK_KOMPRESSION" where="SENS.DRUCK.DECKELSEITE">
        <Threshold threshold="1.lower" />
      </DataSpecies>
      <or>
        <DataSpecies type="DT.POLYTROPENEXPONENTL_KOMP" where="SENS.DRUCK.DECKELSEITE">
          <Threshold threshold="1.upper" />
        </DataSpecies>
        <DataSpecies type="DT.36.RMS" where="SENS.VIBRATION_KREUZKOPFBAHN">
          <Segment36Offset relativeToType="DT.RODLOAD_REVERSAL_START"
            relativeToPosition="SENS.INVALID"
            offset="-2" threshold="1.upper" weight="1" />
        </DataSpecies>
      </or>
    </DataSpecies type="DT.36.ZEROPEAK" where="SENS.VIBRATION_ZYLINDER">
```

```
<Segment36Offset offset="1" relativeToType="DT.DURCHSTICHSPUNKT.KOMP.END"
    relativeToPosition="SENS.DRUCK.KURBELSEITE"
    threshold="1.upper" />
</DataSpecies>
<DataSpecies type="DT.36.ZEROPEAK" where="SENS.VIBRATION.ZYLINDER">
  <Segment36Offset offset="2" relativeToType="DT.DURCHSTICHSPUNKT.KOMP.END"
    relativeToPosition="SENS.DRUCK.KURBELSEITE"
    threshold="1.upper" />
</DataSpecies>
</and>
</Rule>
</RuleBase>
```

## 7.2.2 Überführung der XML-Regeln in die internen Strukturen

Die vollständig definierte Regelbasis in XML wird mit Hilfe von MSXML 4.0 (Microsoft XML Core Services [5]) eingelesen und geparkt. Die Klasse *CXMLParser* stellt dabei Wrapperfunktionen für die Verwendung der wichtigsten Methoden aus der MSXML Bibliothek bereit, damit die COM-Objekte zentral gehalten werden und die Übersicht des Programmcodes erhöht wird. Das Klassendiagramm in Bild 7.2 veranschaulicht unter anderem die Benutzung der MSXML-Bibliothek über die von Microsoft definierte COM-Schnittstelle (Component Object Model [10]).

Nachdem die Regeln aus dem XML-Dokument ausgelesen wurden, stehen diese in Form eines DOM-Baumes dem Programm zu Verfügung. Anschließend werden alle vorhandenen Regeln durch die Klasse *CRuleBase* an die zu überwachte Maschine angepasst. Bei der Anpassung geht es in erster Linie darum, inkompatible Regeln zu verwerfen und andere wiederum zu manipulieren. Kriterien für das Löschen einer Regel können ein falscher Maschinentyp oder die Existenz bzw. Inexistenz einer Analyse in den Attributen *dependsOnOneOf*, *dependsOnAllOf*, *deniedByOneOf* und *deniedByAllOf* sein.

Hat eine Regel den richtigen Kontext für die Maschine, so wird überprüft, ob die Maschine die nötigen Analysen aufweist. Sollte eine Analyse fehlen, so wird zuerst versucht einen Teil der Regel zu entfernen, ohne dabei den Rest der Regel zu verfälschen. Das Löschen eines Teilbaumes kann man sich leichter vorstellen, wenn man die Regel in die disjunktive Normalform überführt. Und zwar wird bei einer fehlender Analyse ein ganzer Konjunktionsterm (vgl. Kapitel 6.3.1) gelöscht. Ist dies nicht möglich, wird ebenfalls die ganze Regel eliminiert. Sind Regeln z.B. *perCylinder* definiert und eine Maschine besteht aus vier Zylindern, so wird diese Regel für alle vorhandenen Zylinder dupliziert.



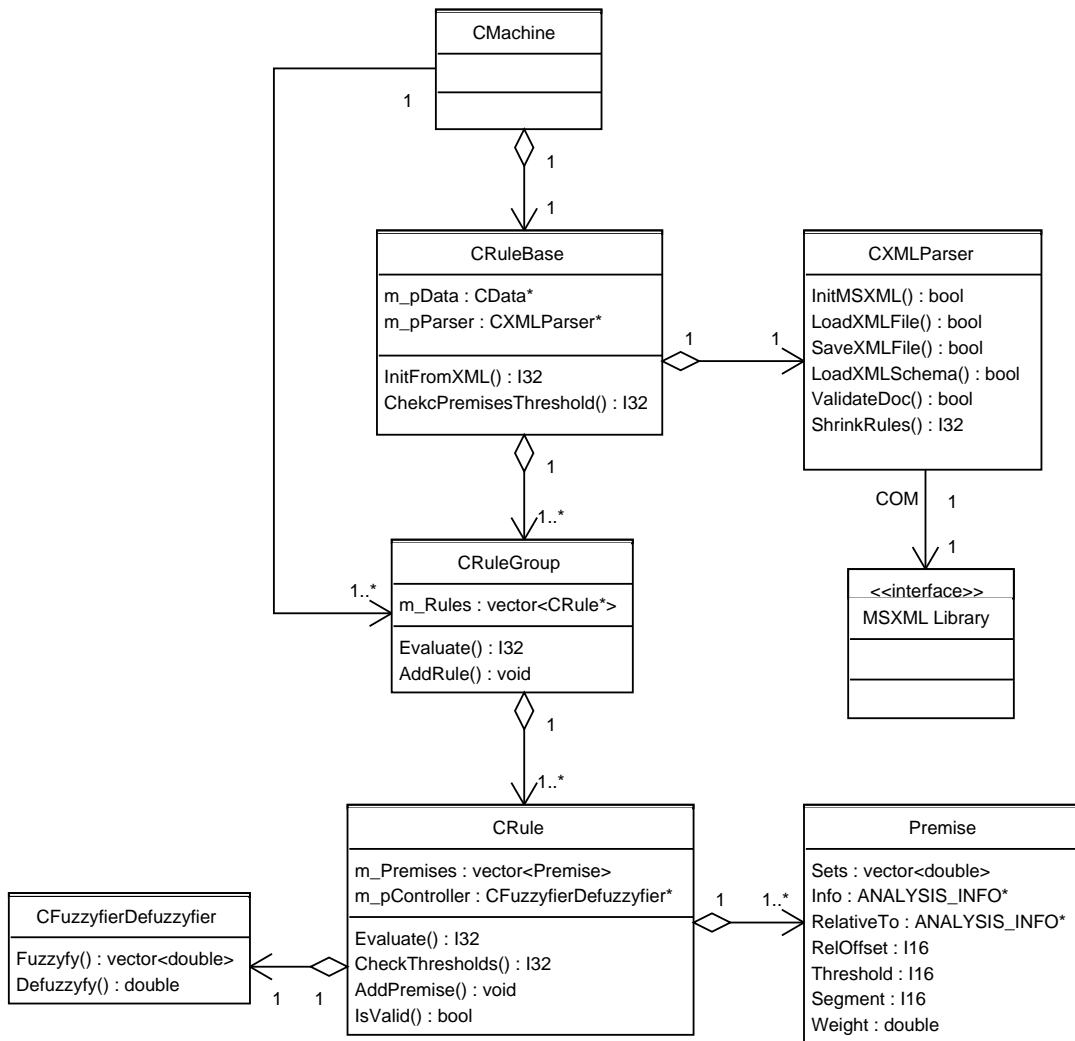


Abbildung 7.2: Klassendiagramm von *CRuleGroup* und *CRule*

Folglich sind nur noch gültige Regeln für die zu überwachte Maschine übrig geblieben, allerdings sind diese vorerst im DOM-Baum gespeichert und müssen über die COM-Schnittstelle angesprochen werden. Um den Regelbaum in die disjunktive Normalform zu transformieren (vgl. Kapitel 6.3.1), wird der DOM-Baum in mehrere flache Listen überführt, so dass eine Liste einen Konjunktionsterm der Regel darstellt. Ein Konjunktionsterm wird in der Klasse *CRule* abgelegt und die kumulierten Konjunktionsterme einer Regel werden in der Klasse *CRuleGroup* gespeichert. D.h. *CRuleGroup* repräsentiert eine einzige Regel aus dem XML-Dokument, die vollständig an eine Maschine angepasst und

in die DNF überführt ist. Das Klassendiagramm der Regeln veranschaulicht Abbildung 7.2.

Jede Regel besteht aus beliebig vielen Prämissen und verwendet einen *CFuzzyfierDefuzzyfier*. Jede Prämisse enthält eine Referenz auf eine Analyse. Zusätzlich werden in einer Prämisse nur die Zugehörigkeitsgrade der beiden Fuzzy-Sets  $n$  und  $v$  gespeichert und nicht die Ausprägungen der Fuzzy-Sets. Der *CFuzzyfierDefuzzyfier* dient dazu, um Analysen zu fuzzyfizieren (vgl. Kapitel 6.2.2) und um das unscharfe Ergebnis wieder zu defuzzyfizieren (vgl. Kapitel 6.3.1).

Die Überführung des Regelbaumes in flache Listen hat bei der Implementierung zusätzlich den Vorteil, dass Regeln schneller ausgewertet werden, da Zugriffe über COM und Rekursionen beim Iterieren des Regelbaumes wegfallen.

Die *CMachine* hat einen Direkten Zugriff auf die *CRuleGroups* und kann mit der Methode *Evaluate()* die einzelnen Regeln direkt ausgewertet. Als Ergebnis wird der Center of Gravity per Reference zurückgegeben. Der Center of Gravity, die eindeutige RuleID und die Musterklasse bilden zusammen die Konklusion einer Regel, die in der Klasse *CPatternFuzzyResult* abgespeichert wird (vgl. Diagramm 7.1).

## 7.3 Interne Datenhaltung

Als zentrale Datenhaltung dient die Klasse *CData*. Sie ermöglicht einen schnellen Zugriff auf die Analysen und deren Informationen. Die einzelnen Informationen werden bei der Initialisierung in verschachtelte HashMaps gespeichert. Die HashMaps repräsentieren die topologische Struktur der Maschine. Von der Kurbeln zu den Zylindern, dann zu den Messpunkten, weiter zu den Analysen und zu letzt zu den Statusinformationen der einzelnen Segmente.

Alle Knoten werden durch positive Zahlen adressiert. Jedoch gibt es in den äußeren drei Ebenen den möglichen Schlüssel -1, um anzudeuten, dass die darunter liegenden Informationen direkt neben der jeweiligen Ebene vorhanden sind. Eine -1 als Kurbelnummer würde bedeuten, dass alle darunter befindlichen Analysen sich auf dem Maschinenniveau befinden. Weiterhin würde ein -1 als Zylinder Nummer bedeuten, dass die darunter gespeicherten Analysen sich auf die Kurbel beziehen.

Die Klasse *CData* bietet weiterhin Methoden, um die Makros aus den Header-Dateien *Prognost-Defines.h* und *Prognost-Datatypes.h* einlesen zu können. Dies wird für die Zu-

ordnung zwischen den sprachlich verständlichen Daten aus dem XML-Dokument und den in PROGNOST<sup>®</sup>-NT verwendeten Datentypen in numerischer Form benötigt.

Das nachfolgende Klassendiagramm verdeutlicht die Verwendung der Klasse *CData*. Sie wird beim Initialisieren der Schadenserkenkung instantiiert und mit allen nötigen Daten gefüllt. *CRuleBase* greift dann auf die Daten aus *CData* zu, um die geparsten Regeln zu überprüfen.

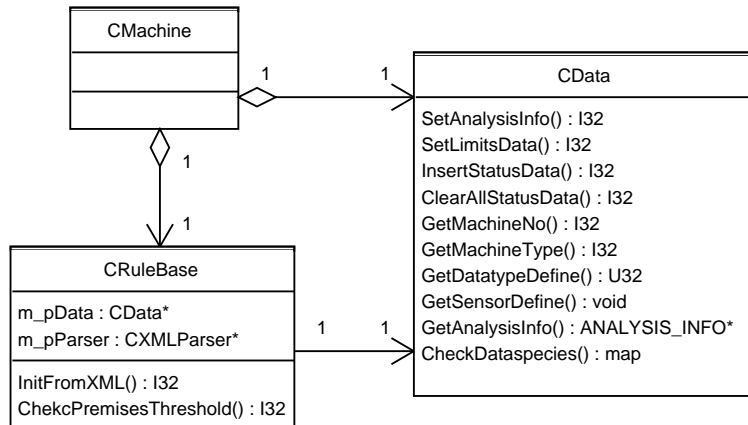


Abbildung 7.3: Klassendiagramm vom *CData* Kontext

Die angepassten Regeln in den Klassen *CRuleGroup* und *CRule*, deren Zusammenhang im vorherigen Kapitel erläutert wurde, werden ebenfalls dauerhaft gespeichert.

## 7.4 Fuzzy-Klassen

In diesem Kapitel werden die implementierten Fuzzy-Klassen für die nötigen Fuzzy-Sets sowie Fuzzy-Operationen erläutert. Abbildung 7.4 zeigt ein vereinfachtes Klassendiagramm der Fuzzy-Klassen.

Das oberste Interface lautet *IFuzzySet*, welches alle wesentlichen Funktion eines Fuzzy-Sets definiert. Die abstrakte Klasse *CFuzzyObject* implementiert einige allgemeine Funktionen des Interfaces, wie z.B. getter und setter für die Definitionsgrenzen. Konkrete Fuzzy-Sets, wie Polygone oder Dreiecke, implementieren weitere Methoden wie *CenterOfGravity()* oder *Containment()*. Das System ist jederzeit um weitere Fuzzy-Sets mit anderen Ausprägungen wie s- oder z-Form erweiterbar.

Die *CFuzzyUnion* leitet sich von der abstrakten Klasse *CFuzzyOperation* ab und berechnet die Vereinigung zweier Fuzzy-Sets auf dem selben Definitionsbereich. Der Fuzzy-

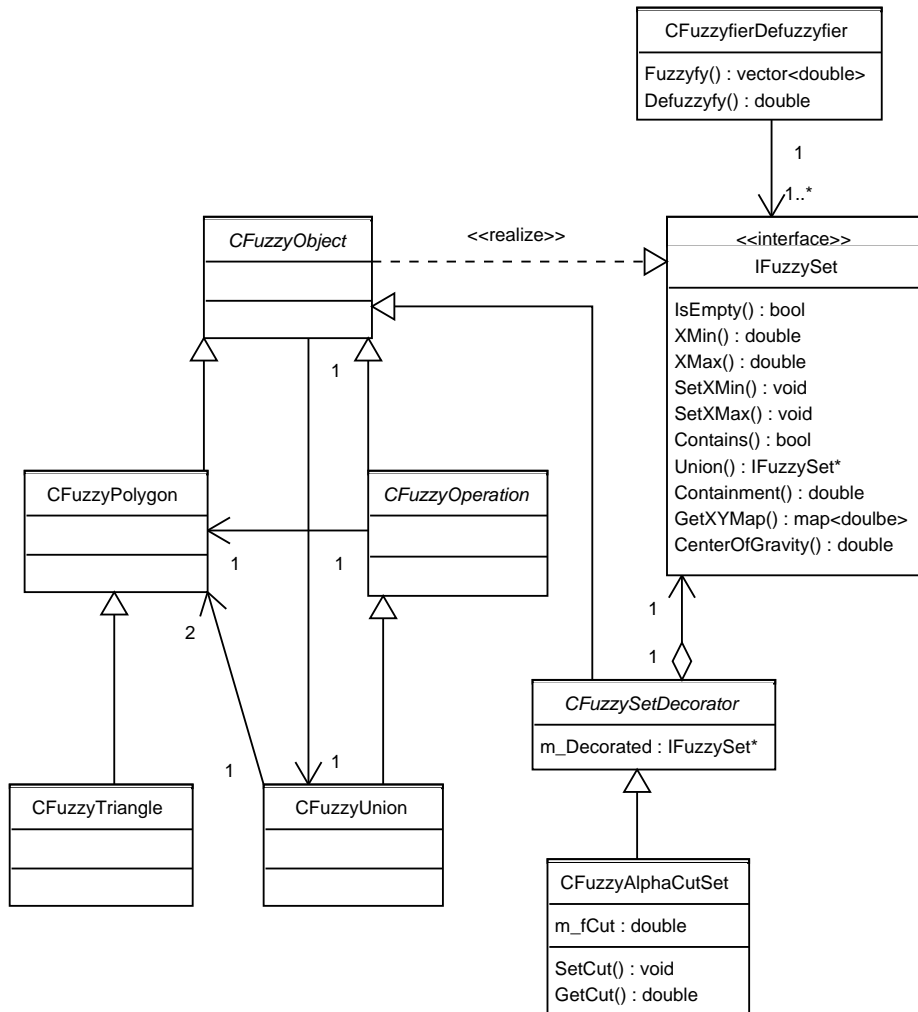


Abbildung 7.4: Klassendiagramm der Fuzzy-Klassen

Durchschnitt und das Fuzzy-Komplement werden aufgrund des in Kapitel 6 vorgestellten Systems nicht benötigt.

Weiterhin existiert ein abstrakter *CFuzzySetDecorator*, der ein Fuzzy-Set dekoriert. Die konkrete Klasse *FuzzyAlphaCutSet* dekoriert ein Fuzzy-Set, indem alle Werte der Zugehörigkeitsfunktion, die größer als das  $\alpha$  sind, auf den Wert  $\alpha$  normiert werden. In den oberen Kapiteln wurde  $\alpha$  auch als Erfüllungsgrad bezeichnet.

Wie bereits oben erwähnt, werden in *CRule* nur die Zugehörigkeitsgrade der Fuzzy-Sets gespeichert. Da alle Prämissen und alle Ausgangsvariablen jeweils die gleiche Form haben (vgl. Kapitel 6.2.2 und 6.2.3), werden diese zentral im *CFuzzyfierDefuzzyfier*

abgelegt. Der *CFuzzyfierDefuzzyfier* mit dem *IFuzzySet* bilden den einfachen Fuzzy-Controller, der wiederum jeder Regel bekannt ist.

# Kapitel 8

## Vergleich verschiedener Operatoren bei unterschiedlichen Eingangswerten

Im letzten Kapitel wird das Verhalten der Schadenserkennung mit unterschiedlichen Operatoren untersucht. Die Untersuchung beschränkt sich auf eine einzige Regel, die nachfolgend vorgestellt wird, deren Struktur eine durchschnittliche Schadensregel repräsentiert. Es soll untersucht werden, in wieweit die Operatoren auf einzelne Ausreißer in einer Regel reagieren und ob es feststellbar ist, ob Ausreißer vorhanden sind. Folglich werden in Kapitel 8.2 einige Operatoren ausgewählt und deren Eigenschaften festgelegt. Zuletzt werden die unterschiedlichen Ergebnisse der Schadenserkennung ausgewertet.

### 8.1 Szenario und Eingangswerte

Für die folgenden Analysen soll die Regel aus Abbildung 8.1 verwendet werden. Diese Regel hat in der disjunktiven Normalform folgende Ausprägung:

$$z = abce \vee abc f \vee abde \vee abdf.$$

Sei nun  $x$  ein Vektor, der die scharfen Eingangswerte  $a, b, \dots, f$  beinhaltet, der Einfachheit halber gilt  $a, b, \dots, f \in [0, 1]$ . Weiterhin seien  $A, B, \dots, F$  jeweils die Fuzzy-Eingänge, die mit  $a, b, \dots, f$  fuzzyfiziert werden und  $z$  der CoG des unscharfen Ausgangs  $Z$ . Der Vektor  $x$  dient dazu die aktuellen Eingangswerte für alle Prämissen der Regel zu veranschaulichen. Da die gleichzeitige Darstellung der Verhältnisse aller Eingänge extrem unübersichtlich sein wird, wurden nur bestimmte Kombinationen ausgewählt, wo sich nur eine einzige Prämisse ändert und alle anderen konstant bleiben. Diese Prämisse,

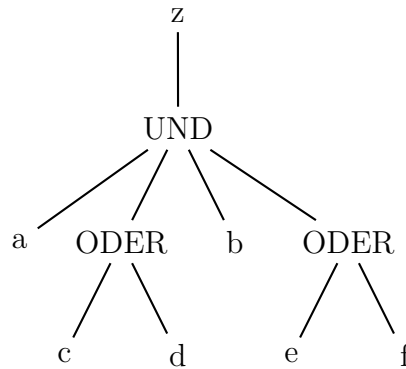


Abbildung 8.1: Szenario mit einer durchschnittlichen Schadensregel

die variabel ist, wird im Vektor  $x$  mit ihren Namen gekennzeichnet und die konstanten Prämissen haben dem entsprechend einen festen Wert.

**Beispiel 5:**

Der Vektor  $x = [ a \ b \ c \ d \ e \ f ] = [ a \ 0.1 \ 0.4 \ 0.9 \ 0.2 \ 0.55 ]$  ist so zu interpretieren, dass bei den konstanten Prämissen  $b, c, d, e, f$  das Verhalten der ganzen Regel in Abhängigkeit von  $a$  mit unterschiedlichen Operatoren untersucht werden soll.

## 8.2 Operatoren

Es werden insgesamt sechs Operatoren mit unterschiedlichen Eigenschaften für die UND-Verknüpfung verwendet. Als ODER-Verknüpfung wird, wie in Kapitel 6.3.1, der maximale Center of Gravity der einzelnen Konjunktionsterme verwendet. Diese sechs Operatoren sind:

- $(GO_0)$  Linearer  $\gamma$ -Operator aus Gleichung 3.1 mit  $\gamma = 0$ . Als t-Norm kommt die min-Norm und als s-Norm die max-Norm aus Tabelle 2.1 zum Einsatz. Die Orness ist in diesem Fall gleich 0.
- $(OWA_{1/4})$  OWA mit konstanter Orness = 0.25 (vgl. Gleichung 3.5).
- $(AM)$  OWA Operator mit konstanten Gewichten, oder auch arithmetisches Mittel aus Tabelle 3.1. Das arithmetische Mittel dient als Referenzwert, da es laut Experten das „richtige“ Ergebnis liefert.

- (GM) Geometrisches Mittel ebenfalls aus Tabelle 3.1.
- (OWA<sub>3/4</sub>) OWA mit konstanter Orness = 0.75 (vgl. Gleichung 3.6).
- (GO<sub>1</sub>) Linearer  $\gamma$ -Operator aus Gleichung 3.1 mit  $\gamma = 1$  und selber Normkombination wie oben, allerdings mit Orness = 1.

### 8.3 Auswertung

Zuerst soll eine Kombination untersucht werden, wo alle Prämissen außer  $a$  eine Verletzung nahe Null haben,  $x = [ a \ 0.1 \ 0.05 \ 0.2 \ 0.16 \ 0.23 ]$  sei hierzu gegeben. Abbildung 8.2 veranschaulicht die Ergebnisse der unterschiedlichen Operatoren. An dieser Stelle ist zu beachten, dass die ODER-verknüpften Konjunktionsterme nach dem Prinzip des maximalen Center of Gravity, wie in Kapitel 6.3.1 beschrieben, ausgewertet werden.

Das AM ist in rot eingezeichnet und repräsentiert das implementierte Verhalten der Schadenserkenkung, welches in Kapitel 6 beschrieben wurde. Auffällig bei dem AM ist, dass das Ergebnis der Schadensregel durchgehend konstant steigt, auch wenn der Ausreißer  $a$  zunehmend stärker wird.

Ein auffälliges Verhalten in den Grenzwerten weist das GM auf. Sobald das Fuzzy-Set  $n$  der Fuzzy-Variable  $A$  voll erfüllt ist sinkt das Ergebnis rapide auf 0, da ein voll erfülltes Set  $n_A$  auch gleichzeitig ein nicht erfülltes Set  $v_A$  impliziert. Die Multiplikation mit 0 zwingt letztendlich die ganze Regel auf einen Verletzungsgrad von 0. Analog verhält sich die Regel bei einem Erfüllungsgrad von 1 des Sets  $v_A$ .

Die beiden OWA Operatoren OWA<sub>1/4</sub> und OWA<sub>3/4</sub> liegen wie erwartet leicht über bzw. unter dem AM. Der GO<sub>1</sub> wird für  $a = 1$  nur leicht höher als 0.5, was daran liegt, dass der Erfüllungsgrad von  $n_Z$  nur etwas kleiner ist als  $v_Z$ . Der GO<sub>0</sub> verhält sich ähnlich zum GM, allerdings mit weicherer Steigung. Für  $a > 0.7$  fängt die Kurve an extrem zu steigen, bis sie schließlich das Maximum erreicht, was damit zusammenhängt, dass der Erfüllungsgrad von  $n_Z$  gegen 0 läuft und  $v_Z$  stattdessen einen minimalen Erfüllungsgrad größer 0 behält.

Wenn man in diesem Fall den Wert des GO<sub>0</sub> zusätzlich zum AM heranzieht, lassen sich in diesem Fall Ausreißer nach oben hin erkennen, was bei der Bewertung des Ergebnisses helfen könnte.



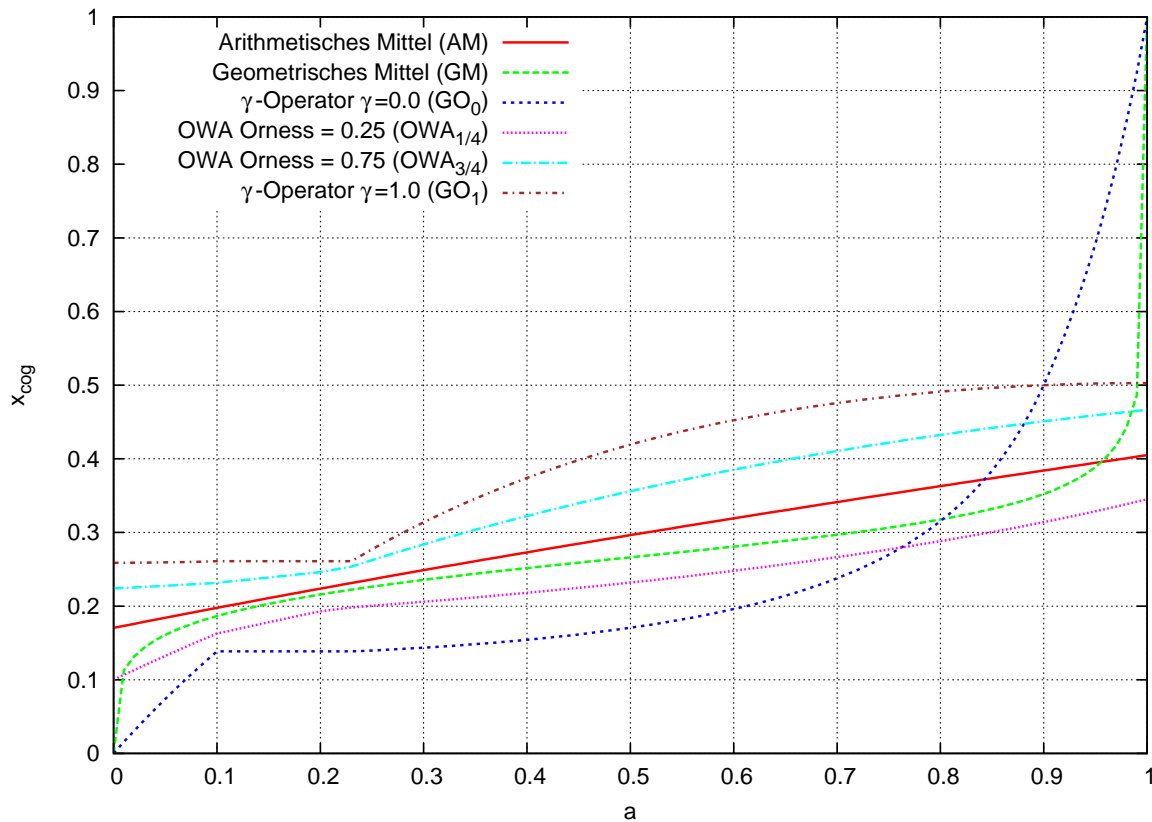


Abbildung 8.2: Eingangsvektor  $x = [a \ 0.1 \ 0.05 \ 0.2 \ 0.16 \ 0.23]$

Erweitert man den ersten Wertevektor soweit, dass eine weitere stark verletzte Prämisse dazu kommt, so lassen sich ebenfalls mit dem  $GO_0$  starke Abweichungen in den Eingangswerten erkennen. Für  $x = [a \ 0.2 \ 0.15 \ 0.3 \ 0.05 \ 0.96]$  sind die Ergebnisse in Abbildung 8.3 dargestellt.

Der schnelle Anstieg der blau-gestrichelten Kurve hängt damit zusammen, dass für  $a = 0$  die Erfüllungsgrade von  $n_Z$  und  $v_Z$  gleich 0.04 und 0.0 sind. Da der Erfüllungsgrad von  $n_Z$  sehr gering ist, reicht eine leichte Erhöhung des Erfüllungsgrades von  $v_Z$  aus, um im Verhältnis zum Erfüllungsgrad von  $n_Z$  viel größer zu sein. Für  $a = 0.1$  stellen sich die Erfüllungsgrade auf 0.04 (von  $n_Z$ ) und 0.2 (von  $v_Z$ ) ein. Die nächste Steigung erfolgt dann, wenn  $a > 0.96$  wird, da somit zwar der Erfüllungsgrad von  $v_Z$  bei 0.2 bleibt, aber der Erfüllungsgrad von  $n_Z$  zunehmend kleiner wird und gegen Null läuft.

Weiterhin ist zu sehen, dass die beiden letzten Operatoren  $OWA_{3/4}$  und  $GO_1$  mit einer starken Orness nahe zu konstante Ergebnisse liefern, was daran liegt, dass fast

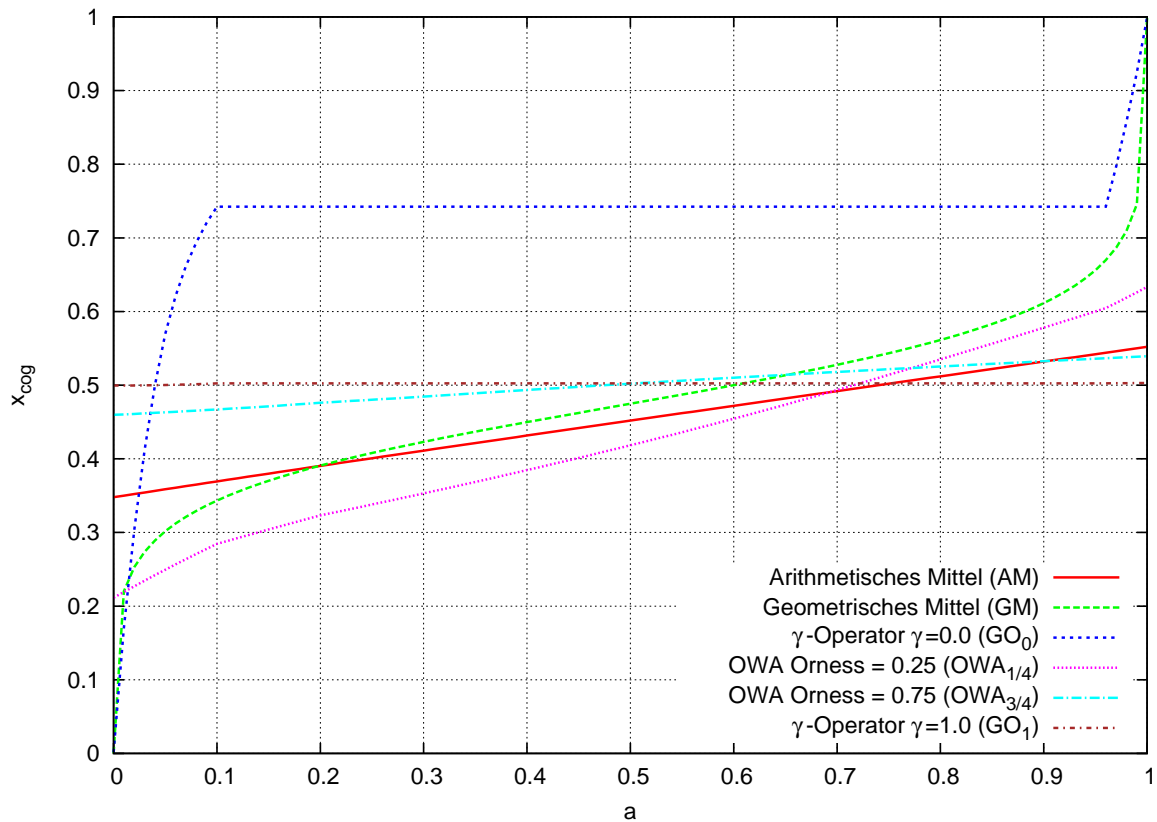


Abbildung 8.3: Eingangsvektor  $x = [a \ 0.2 \ 0.15 \ 0.3 \ 0.05 \ 0.96]$

durchgehend die Erfüllungsgrade von  $n_Z$  und  $v_Z$  auf gleicher Höhe sind. Die Ausbrüche in den Grenzen beim GM entstehen auf gleicher Weise wie oben beschrieben. Beim  $OWA_{1/4}$  sind an den selben Stellen, wie beim  $GO_0$ , ansatzweise Sprünge erkennbar.

Auch in diesem Fall ist der Abstand zwischen dem AM und dem  $GO_0$  bis zu 0.4 groß, was auf stark auseinanderliegende Verletzungsgrade deutet. Für  $a \rightarrow 1$  wird dieser Abstand dann geringer, da dann schon zwei von vier Prämissen sehr stark verletzt sind und auch keine Ausreißer mehr darstellen.

Der Vollständigkeit halber soll auch der umgekehrte Fall betrachtet werden, wo mit  $x = [a \ 0.93 \ 0.99 \ 0.96 \ 0.97 \ 0.95]$  alle Prämissen bis auf  $a$  fast voll verletzt sind, vgl. Bild 8.4.

Im Gegensatz zur ersten Konstellationen klettert in diesem Fall das Ergebnis mit dem  $GO_0$  schon ziemlich am Anfang von 0 bis fast auf das Maximum hinauf. Das Anfangs-

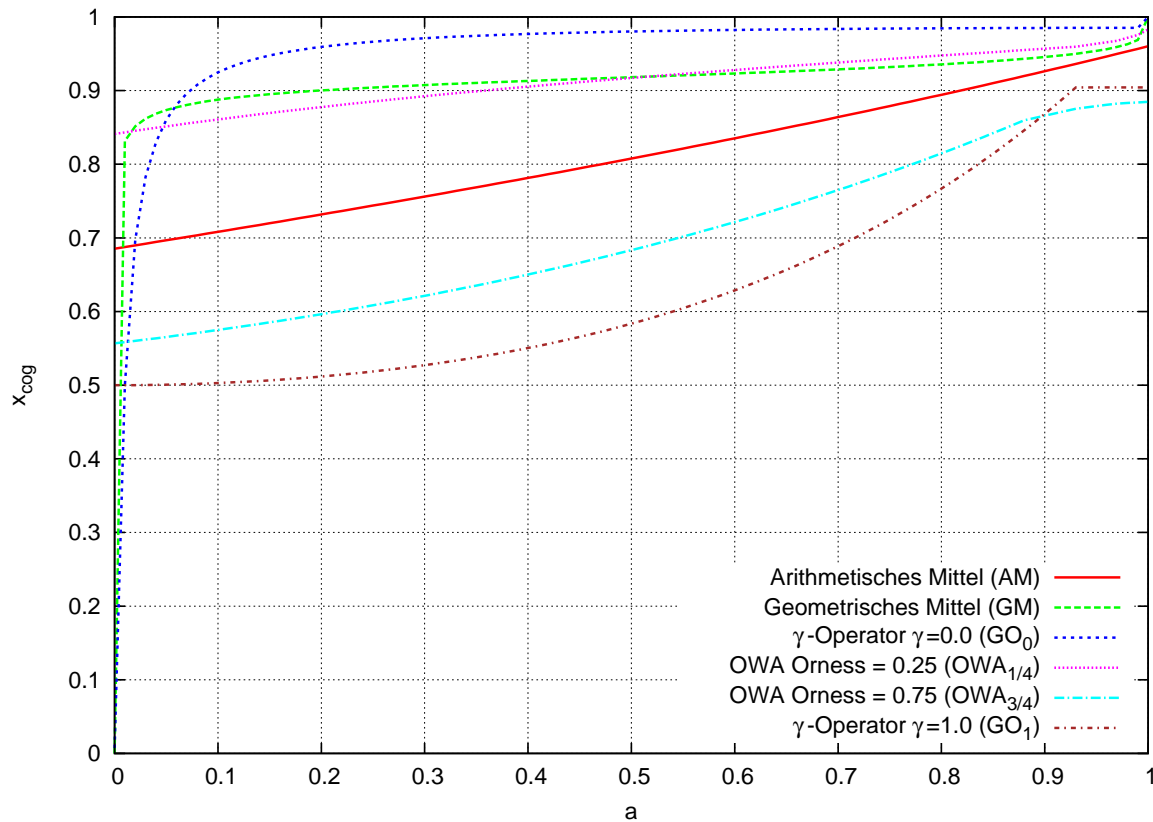


Abbildung 8.4: Eingangsvektor  $x = [a \ 0.93 \ 0.99 \ 0.96 \ 0.97 \ 0.95]$

verhalten von  $GO_0$  ist vergleichbar mit dem vom GM, d.h. Ausreißer können nur im Bereich  $a < 0.02$  bemerkt werden.

Würde man in diesem Fall die Schwelle für den Verletzungsgrad eines Schadens auf 0.8 setzen, so würde die Regel mit  $OWA_{3/4}$  unabhängig von  $a$  sofort als „verletzt“ ausgegeben. Der GM und der  $GO_0$  verhalten sich ähnlich. Mit dem AM wird eine Verletzung der Regel oberhalb 0.8 erst erreicht, wenn  $a$  bei 0.5 liegt. Die beiden Operatoren  $GO_1$  und  $OWA_{3/4}$  würden den Verletzungsgrad 0.8 erst für  $a > 0.83$  bzw.  $a > 0.77$  überschreiten.

Für die letzte Einstellung sei  $x = [a \ 0.45 \ 0.56 \ 0.6 \ 0.51 \ 0.39]$ , wo alle Prämissen einen Verletzungsgrad von ca. 0.5 haben, vgl. Abbildung 8.5. Bis auf die Operatoren  $GO_0$  und GM verhalten sich die anderen relativ ähnlich, um so höher die Orness desto flacher der Graph.

Am GM ist auch in diesem Fall zu sehen, dass die Grenzwerte 0 und 1 starke Sprünge verursachen, d.h. bis auf die Grenzwerte verhält sich das GM ebenfalls ähnlich zu den restlichen.

Anhand des  $GO_0$  lässt sich in diesem Fall der Ausreißer sehr deutlich erkennen. Für  $a \approx 0.5$  liefert der  $GO_0$  die gleichen Werte, wie das AM und andere Operatoren. Je weiter sich jedoch  $a$  von 0.5 entfernt, desto stärker sinkt, bzw. steigt der Verletzungsgrad der Regel. Für  $a = 0.1$  und  $a = 0.9$  beträgt der Abstand zum AM ca. 0.25 und in den Grenzen sogar bis zu 0.4.

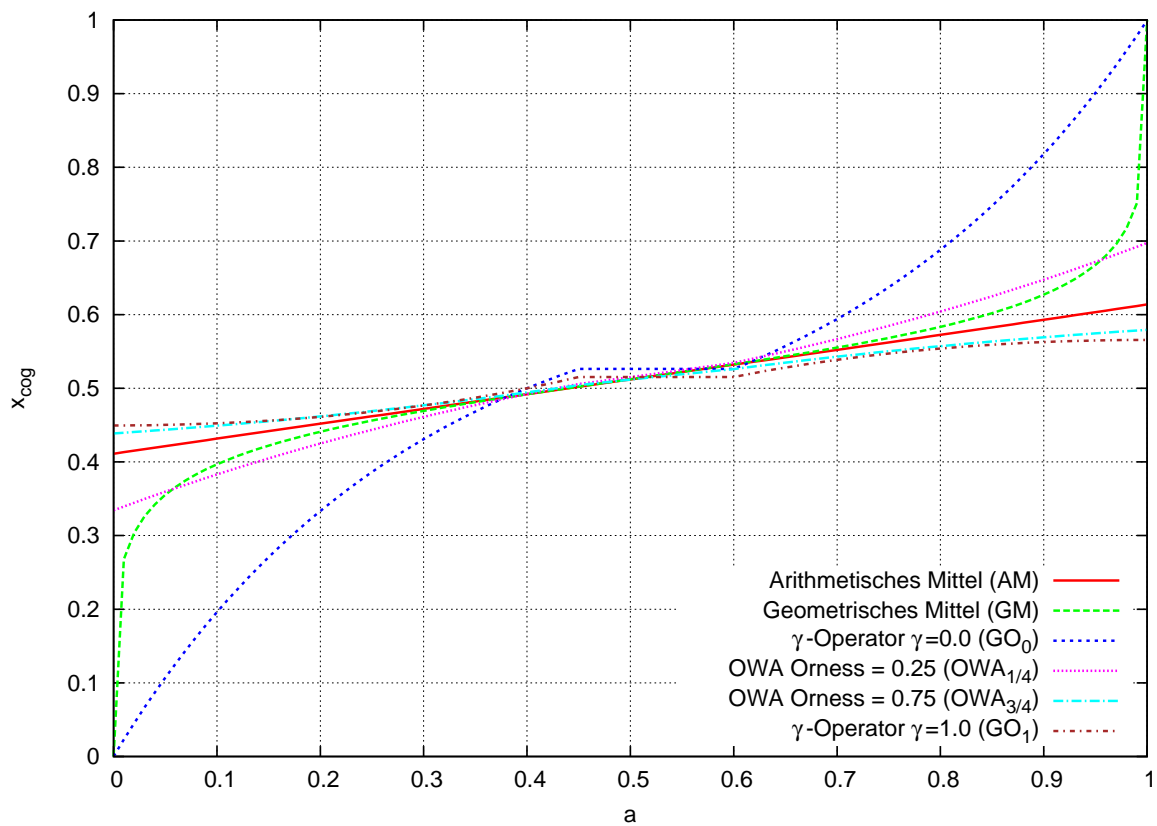


Abbildung 8.5: Eingangsvektor  $x = [a \ 0.45 \ 0.56 \ 0.6 \ 0.51 \ 0.39]$

# Kapitel 9

## Zusammenfassung

Ziel dieser Arbeit war es, ein Fuzzy-System zu entwickeln, welches mit vielen Eingängen und wenig Regeln trotzdem erfolgreich Schadensmuster erkennt. Die scharfen Schadensregeln wurden auf eine minimale Fuzzy-Regelbasis überführt, woraus sich schließlich auch die Anzahl der Fuzzy-Sets pro Variable ergab. Mit der dynamischen Konstruktion der Fuzzy-Eingangsvariablen wurde die Handhabung einer beliebigen Warnschwelle ermöglicht. Ferner diente das arithmetische Mittel dazu, die Sprünge, die durch das Fehlen der Fuzzy-Regeln unmittelbar entstanden, aufzuweichen. Mit dem maximalen Center of Gravity der Konjunktionsterme wurde eine optimale ODER-Verknüpfung für die Schadenserkenkung realisiert. Weiterhin ließ sich die Berechnung des CoG aufgrund der speziellen Konstruktion der Fuzzy-Ausgangsvariable und der gewählten Inferenz soweit vereinfachen, dass diese nur in Abhängigkeit vom Erfüllungsgrad  $\alpha_v$  des Sets  $v$  durchführbar ist.

Der grundlegende Unterschied zwischen dem Fuzzy-System und neuronalem Netz ist, dass das Fuzzy-System den Verletzungsgrad einer Schadensregel aus den nötigen Analysen ermittelt, und das neuronale Netz den Erkennungsgrad des Schadensmusters aus allen verletzten Analysen berechnet. Somit erfolgt mit dem Fuzzy-System bereits eine Ausgabe, auch wenn keine einzige Prämisse eine Warnschwelle verletzt. Diese Eigenschaft ermöglicht nicht nur das vorzeitige Erkennen von Schäden, sondern auch die Analyse der Tendenz eines Schadens und wie sich diese entwickelt.

Weiterhin ist es nun mit dem Fuzzy-System möglich, relative Segmentangaben, die von einer anderen Analyse abhängen, zu berechnen. Ebenfalls kann das Fuzzy-System Schadensregeln behandeln, die explizit keine Verletzung einer Warnschwelle erfordern. Durch die Speicherung der Regeln in einem XML-Dokument, können Experten die Re-

geln unabhängig bearbeiten und zudem lässt sich der Regelbaum in der XML-Struktur wiedererkennen.

Der Vergleich der Operatoren hat gezeigt, dass das arithmetischen Mittel eine stabile und sprunglose Inferenz ermöglicht und mit der maximalen Entropie den Erwartungen der Experten genügt. Zusätzlich ist es möglich mit dem  $\gamma$ -Operator und  $\gamma = 0$  so wie der min-max-Norm, Ausreißer in den Eingängen zu detektieren, indem die Differenz der Ergebnisse der beiden Operatoren betrachtet wird. Um so größer die Differenz desto größer der Ausreißer, allerdings muss es nicht heißen, dass bei einer kleinen Differenz kein Ausreißer vorhanden ist.

Das Fuzzy-System wurde viel anhand theoretischer Modelle getestet und lieferte die gewünschte Ergebnisse. Es bleibt allerdings abzuwarten, wie sich das Fuzzy-System in der Praxis im Vergleich zum neuronalen Netz bewährt. Weiterhin bleibt zu überprüfen, in wie weit die Entwicklung eines Schadens möglich bzw. hilfreich ist.

# Anhang A

## Monotonie Beweis

Um zu zeigen, dass die Äquivalenz in 6.5 gilt, reicht zu beweisen, dass  $x_{cog}(\alpha)$  monoton steigend ist. Es seien  $a, b \in [0, 1]$  gegeben, es ist zu zeigen, dass  $\forall a, b : a < b$  die Ungleichung  $x_{cog}(a) < x_{cog}(b)$  gilt. Die nachfolgende Äquivalenzumformung führt zu dem richtigen Ergebnis.

**Beweis:**

$$\begin{aligned} & x_{cog}(a) < x_{cog}(b) \\ \Leftrightarrow & \frac{1}{2} \frac{a(3-a)}{a(1-a)+1} < \frac{1}{2} \frac{b(3-b)}{b(1-b)+1} \\ \Leftrightarrow & (3a-a^2)((b-b^2)+1) < (3b-b^2)((a-a^2)+1) \\ \Leftrightarrow & 3ab-3ab^2+3a-a^2b+a^2b^2-a^2 < 3ab-3a^2b+3b-ab^2+a^2b^2-b^2 \\ \Leftrightarrow & -3ab^2+3a-a^2b-a^2 < -3a^2b+3b-ab^2-b^2 \\ \Leftrightarrow & -2ab^2+3a-a^2 < -2a^2b+3b-b^2 \end{aligned}$$

Addiert man nun auf der linken Seite der Ungleichung  $a^2$  und auf der rechten Seite  $b^2$ , so bleibt die Ungleichung wegen der Annahme  $a < b$  immer noch gültig.

$$\begin{aligned} \Leftrightarrow & -2ab^2+3a < -2a^2b+3b \\ \Leftrightarrow & 2ab^2-3a > 2a^2b-3b \end{aligned}$$

Nochmals eine Addition des Terms  $3a$  auf der linken und  $3b$  auf der rechten Seite.

$$\Leftrightarrow 2ab^2 > 2a^2b$$

$$\Leftrightarrow b > a$$

□



# Tabellenverzeichnis

2.1	Typische nicht-parametrisierte t- und s-Normen . . . . .	17
3.1	Nicht-parametrisierte kompensatorische Operatoren . . . . .	22
6.1	Bekannte Regeln in Abhängigkeit von der Anzahl der Fuzzy-Sets $n$ . . . . .	41
7.1	Attribute des <i>Rule</i> -Elements in XML . . . . .	53
7.2	Kindelemente des <i>DataSpecies</i> -Elements . . . . .	54

# Abbildungsverzeichnis

2.1	Zugehörigkeitsfunktionen $f_{triangle}$ (links) und $f_{trapez}$ (rechts) . . . . .	12
2.2	Beispiel eines Fuzzy-Sets in diskreter Darstellung . . . . .	13
2.3	Darstellung des Fuzzy-Durchschnitts . . . . .	14
2.4	Darstellung der Fuzzy-Vereinigung . . . . .	14
2.5	Darstellung des Fuzzy-Komplements . . . . .	15
2.6	Laufbereiche der typischen nicht-parametrisierten t- und s-Normen . . . . .	17
2.7	Linguistische Variable <i>menschliches Alter</i> . . . . .	19
4.1	Komponenten eines klassischen und scharfen bzw. eines regelbasierten und unscharfen Übertragungssystems . . . . .	25
4.2	Fuzzy-Controller nach E. Mamdani und S. Assilian . . . . .	26
4.3	Linguistische Variable <i>Temperatur</i> . . . . .	27
4.4	Fuzzyifizierung zweier beispielhafter Temperaturwerte . . . . .	29
4.5	Defuzzyifizierungsmethoden Max, MoM und CoG . . . . .	32
5.1	Visualisierung der überwachten Maschinen mit der PROGNOST <sup>®</sup> -NT Visu	34
5.2	Parameter einer Analyse . . . . .	36
5.3	Drei verschiedene Schadensmuster aus der Datenbank . . . . .	37
6.1	Dynamische Konstruktion einer Fuzzy-Eingangsvariable nach einer oberen Warnschwelle . . . . .	43
6.2	Einheitliche Form der Fuzzy-Ausgangsvariablen . . . . .	44
6.3	Beispiel eines Regelbaumes mit sechs Eingängen . . . . .	46
6.4	Defuzzyifizierung der unscharfen Ausgangsvariable . . . . .	46
6.5	a) Erfüllungsgrad $\alpha$ im Vergleich zum resultierenden $x_{cog}$ , b) Differenz $x_{cog} - \alpha$ . . . . .	48

6.6	Zwei Fuzzy-Variablen mit je zwei Sets UND-verknüpft: mit der min-Norm (links) und mit dem arithmetischen Mittel (rechts) . . . . .	49
6.7	Vergleich max- $m_a$ -Inferenz mit der max <sub>cog</sub> - $m_a$ -Inferenz . . . . .	50
7.1	Klassendiagramm der Schnittstelle zu PROGNOST <sup>®</sup> -NT . . . . .	52
7.2	Klassendiagramm von <i>CRuleGroup</i> und <i>CRule</i> . . . . .	57
7.3	Klassendiagramm vom <i>CData</i> Kontext . . . . .	59
7.4	Klassendiagramm der Fuzzy-Klassen . . . . .	60
8.1	Szenario mit einer durchschnittlichen Schadensregel . . . . .	63
8.2	Eingangsvektor $x = [a \ 0.1 \ 0.05 \ 0.2 \ 0.16 \ 0.23]$ . . . . .	65
8.3	Eingangsvektor $x = [a \ 0.2 \ 0.15 \ 0.3 \ 0.05 \ 0.96]$ . . . . .	66
8.4	Eingangsvektor $x = [a \ 0.93 \ 0.99 \ 0.96 \ 0.97 \ 0.95]$ . . . . .	67
8.5	Eingangsvektor $x = [a \ 0.45 \ 0.56 \ 0.6 \ 0.51 \ 0.39]$ . . . . .	68

# Literaturverzeichnis

- [1] B. S. Ahn. On the Properties of OWA Operator Weights Functions with Constant Level of Orness. *IEEE Transactions on Fuzzy Systems*, 14:511–515, 2006.
- [2] Benno Biewer. *Fuzzy-Methoden: praxisrelevante Rechenmodelle und Fuzzy-Programmiersprachen*. Springer, 1997.
- [3] Prognost Systems GmbH. PROGNOST<sup>®</sup>-NT . <http://www.prognost.com>.
- [4] B. Hammer M. Strickert, T. Bojer. Generalized relevance LVQ for time series. *in: G.Dorffner, H.Bischof, K.Hornik (eds.), Artificial Neural Networks - ICANN'2001, Springer*, pages 677–683, 2001.
- [5] Microsoft. Microsoft XML Core Services. [http://msdn.microsoft.com/en-us/library/ms763742\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms763742(VS.85).aspx), März 2008.
- [6] C. Koers T. Bojer, B. Hammer. Monitoring technical systems with prototype based clustering. *in: M.Verleysen (ed.), European Symposium on Artificial Neural Networks'2003, D-side publications*, pages 433–439, 2003.
- [7] A. Aceves-Lopez und J. Aguilar-Martin. A Simplified Version of Mamdani's Fuzzy Controller: The Natural Logic Controller. *IEEE Transactions on Fuzzy Systems*, 14:16–29, 2006.
- [8] T. Takagi und M. Sugeno. Fuzzy identification of systems and its applications to modeling and control. *IEEE Trans. on Systems, Man and Cybernetics*, 15:116–132, 1985.
- [9] E.H. Mamdani und S. Assilian. An Experiment in Linguistic Synthesis with Fuzzy Logic Controller. *International Journal of Man-Machine Studies*, 7:1–13, 1975.

- [10] Wikipedia. Component Object Model. [http://de.wikipedia.org/wiki/Component\\_Object\\_Model](http://de.wikipedia.org/wiki/Component_Object_Model), Juli 2008.
- [11] Wikipedia. Prozessleitsystem. <http://de.wikipedia.org/wiki/Prozessleitsystem>, Juli 2008.
- [12] R. R. Yager. On ordered weighted averaging aggregation operators in multi-criteria decision making. *IEEE Trans. on Systems, Man and Cybernetics*, 18:183–190, 1988.
- [13] R. R. Yager. Families of OWA operators. *Fuzzy Sets and Systems*, 59:125–148, 1993.
- [14] L. A. Zadeh. Fuzzy Sets. *Information and Control*, 8:338–353, 1965.