

Speicherprogrammierbare Steuerungen

Teil 7: Fuzzy-Control-Programmierung
(IEC 61131-7:2000) Deutsche Fassung EN 61131-7:2000

DIN**EN 61131-7**

ICS 25.040.40; 35.060; 35.240.50

Programmable controllers – Part 7: Fuzzy control programming
(IEC 61131-7:2000); German version EN 61131-7:2000

Automates programmables – Partie 7: Programmation en logique floue
(CEI 61131-7:2000); Version allemande EN 61131-7:2000

Die Europäische Norm EN 61131-7:2000 hat den Status einer Deutschen Norm.

Beginn der Gültigkeit

Die EN 61131-7 wurde am 2000-11-01 angenommen.

Nationales Vorwort

Für die vorliegende Norm ist das nationale Arbeitsgremium K 962 „SPS“ der DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik im DIN und VDE zuständig.

Der Norm-Inhalt war veröffentlicht als E DIN EN 61131-7:1999-04.

Die enthaltene IEC-Publikation wurde vom SC 65B „Devices“ erarbeitet.

Dieses IEC-Komitee hat entschieden, dass der Inhalt dieser Publikation bis zum Jahr 2005 unverändert bleiben soll. Zu diesem Zeitpunkt wird entsprechend der Entscheidung des Komitees die Publikation

- bestätigt,
- zurückgezogen,
- durch eine Folgeausgabe ersetzt oder
- geändert.

Fortsetzung Seite 2
und 58 Seiten EN

Nationaler Anhang NA (informativ) **Zusammenhang mit Europäischen und Internationalen Normen**

Für den Fall einer undatierten Verweisung im normativen Text (Verweisung auf eine Norm ohne Angabe des Ausgabedatums und ohne Hinweis auf eine Abschnittsnummer, eine Tabelle, ein Bild usw.) bezieht sich die Verweisung auf die jeweils neueste gültige Ausgabe der in Bezug genommenen Norm.

Für den Fall einer datierten Verweisung im normativen Text bezieht sich die Verweisung immer auf die in Bezug genommene Ausgabe der Norm.

Der Zusammenhang der zitierten Normen mit den entsprechenden Deutschen Normen ist nachstehend wiedergegeben. Zum Zeitpunkt der Veröffentlichung dieser Norm waren die angegebenen Ausgaben gültig.

IEC hat 1997 die Benummerung der IEC-Publikationen geändert. Zu den bisher verwendeten Normnummern wird jeweils 60000 addiert. So ist zum Beispiel aus IEC 68 nun IEC 60068 geworden.

Tabelle NA.1

Europäische Norm	Internationale Norm	Deutsche Norm
–	IEC 60050-351:1998	IEV Kapitel 351:1999
EN 61131-3:1993	IEC 61131-3:1993	DIN EN 61131-3:1994-08

Nationaler Anhang NB (informativ) **Literaturhinweise**

IEV Kapitel 351, *Internationales Elektrotechnisches Wörterbuch – Teil 351: Automatisierungstechnik; Identisch mit IEC 60050-351:1998-09.*

DIN EN 61131-3, *Speicherprogrammierbare Steuerungen – Teil 3: Programmiersprachen (IEC 61131-3:1993); Deutsche Fassung EN 61131-3:1993.*

ICS 35.240.50; 25.040.40

Deutsche Fassung

Speicherprogrammierbare Steuerungen

Teil 7: Fuzzy-Control-Programmierung
(IEC 61131-7:2000)

Programmable controllers –
Part 7: Fuzzy control programming
(IEC 61131-7:2000)

Automates programmables –
Partie 7: Programmation en logique floue
(CEI 61131-7:2000)

Diese Europäische Norm wurde von CENELEC am 2000-11-01 angenommen. Die CENELEC-Mitglieder sind gehalten, die CEN/CENELEC-Geschäftsordnung zu erfüllen, in der die Bedingungen festgelegt sind, unter denen dieser Europäischen Norm ohne jede Änderung der Status einer nationalen Norm zu geben ist.

Auf dem letzten Stand befindliche Listen dieser nationalen Normen mit ihren bibliographischen Angaben sind beim Zentralsekretariat oder bei jedem CENELEC-Mitglied auf Anfrage erhältlich.

Diese Europäische Norm besteht in drei offiziellen Fassungen (Deutsch, Englisch, Französisch). Eine Fassung in einer anderen Sprache, die von einem CENELEC-Mitglied in eigener Verantwortung durch Übersetzung in seine Landessprache gemacht und dem Zentralsekretariat mitgeteilt worden ist, hat den gleichen Status wie die offiziellen Fassungen.

CENELEC-Mitglieder sind die nationalen elektrotechnischen Komitees von Belgien, Dänemark, Deutschland, Finnland, Frankreich, Griechenland, Irland, Island, Italien, Luxemburg, Niederlande, Norwegen, Österreich, Portugal, Schweden, Schweiz, Spanien, der Tschechischen Republik und dem Vereinigten Königreich.

CENELEC

Europäisches Komitee für Elektrotechnische Normung
European Committee for Electrotechnical Standardization
Comité Européen de Normalisation Electrotechnique

Zentralsekretariat: rue de Stassart 35, B-1050 Brüssel

Vorwort

Der Text des Schriftstücks 65B/406/FDIS, zukünftige 1. Ausgabe von IEC 61131-7, ausgearbeitet von dem SC 65B „Devices“ des IEC TC 65 „Industrial-process measurement and control“, wurde der IEC-CENELEC Parallelen Abstimmung unterworfen und von CENELEC am 2000-11-01 als EN 61131-7 angenommen.

Nachstehende Daten wurden festgelegt:

- spätestes Datum, zu dem die EN auf nationaler Ebene durch Veröffentlichung einer identischen nationalen Norm oder durch Anerkennung übernommen werden muss (dop) 2001-08-01
- spätestes Datum, zu dem nationale Normen, die der EN entgegenstehen, zurückgezogen werden müssen (dow) 2003-11-01

Anhänge, die als „normativ“ bezeichnet sind, gehören zum Norminhalt.

Anhänge, die als „informativ“ bezeichnet sind, enthalten nur Informationen.

In dieser Norm ist Anhang ZA normativ und sind die Anhänge A, B, C, D und E informativ.

Der Anhang ZA wurde von CENELEC hinzugefügt.

Anerkennungsnotiz

Der Text der Internationalen Norm IEC 61131-7:2000 wurde von CENELEC ohne irgendeine Abänderung als Europäische Norm angenommen.

Inhalt

	Seite
Einleitung	6
1 Anwendungsbereich und Ziel	8
2 Normative Verweisungen	8
3 Begriffe	8
4 Integration in die Speicherprogrammierbare Steuerung	10
5 Fuzzy-Control-Sprache FCL	11
5.1 Austausch von Fuzzy-Control-Programmen	11
5.2 Elemente der Fuzzy-Control-Sprache	12
5.3 FCL-Beispiel	20
5.4 Erzeugungsregeln und Schlüsselwörter der Fuzzy-Control-Sprache (FCL)	21
6 Normerfüllung	25
6.1 Erfüllungsklassen der Fuzzy-Control-Sprache FCL	25
6.2 Datenprüfliste	27
Anhang A (informativ) Theorie	29
A.1 Fuzzy-Logik	29
A.2 Fuzzy-Control	32
A.3 Verhalten von Fuzzy-Control	39
Anhang B (informativ) Beispiele	42
B.1 Vorsteuerung	42
B.2 Parameter-Adaption von herkömmlichen PID-Reglern	42
B.3 Direkte Fuzzy-Regelung	42
Anhang C (informativ) Industriebeispiel – Containerkran	44
Anhang D (informativ) Beispiel für die Verwendung von Variablen im Regelblock	55
Anhang E (informativ) Symbole, Abkürzungen und Synonyme	57
Anhang ZA (normativ) Normative Verweisungen auf internationale Publikationen mit ihren entsprechenden europäischen Publikationen	58
Bild 1 – Beispiel eines Fuzzy-Control-Funktionsbausteins in Funktionsbaustein-Sprachendarstellung (FBS)	11
Bild 2 – Datenaustausch von Programmen in Fuzzy-Control-Sprache (FCL)	12
Bild 3 – Beispiel einer Schnittstellendeklaration in Strukturierterm Text und Funktionsbausteinsprache	13
Bild 4 – Beispiel für Rampen-Terme	14
Bild 5 – Beispiel einer Verwendung von Variablen für Zugehörigkeitsfunktionen	15
Bild 6 – Beispiel von Singletons-Termen	15
Bild 7 – Beispiel für einen Fuzzy-Funktionsbaustein	21
Bild 8 – Ebenen der Normerfüllung	25
Bild A.1 – Zugehörigkeitsfunktionen der Terme „volljährig“ und „erwachsen“	29
Bild A.2 – Beschreibung der linguistischen Variablen „Alter“ durch linguistische Terme und ihre Anordnung auf der Zeitskala (Alter in Jahren)	30

Bild A.3 – Allgemein verwendete Formen von Zugehörigkeitsfunktionen	30
Bild A.4 – Algorithmen zur Implementierung von Operationen zwischen zwei Zugehörigkeitsfunktionen	32
Bild A.5 – Struktur und funktionale Elemente von Fuzzy-Control.....	33
Bild A.6 – Das Prinzip der Fuzzifizierung (am Beispiel)	33
Bild A.7 – Darstellung der Wissensbasis in Form von linguistischen Regeln.....	34
Bild A.8 – Matrix-Darstellung von zwei Variablen.....	34
Bild A.9 – Elemente der Inferenz	35
Bild A.10a – Ein Beispiel, das die Prinzipien der Aggregation zeigt.....	36
Bild A.10b - Prinzipien der Aktivierung (als ein Beispiel).....	37
Bild A.10c – Prinzipien der Akkumulation (als ein Beispiel)	37
Bild A.11a – Methoden der Defuzzifizierung	38
Bild A.11b – Unterschied zwischen linkem Maximum und rechtem Maximum	38
Bild A.11c – Unterschied zwischen Flächenschwerpunkt und Schwerpunkt	39
Bild A.11d – Methoden der Defuzzifizierung	39
Bild A.12 – Beispiele von Kennfeldern von Fuzzy-Control	40
Bild A.13a – Grundstruktur der Fuzzy-basierten Regelung.....	40
Bild A.13b – Beispiel der Fuzzy-basierten Regelung	41
Bild B.1 – Beispiel für eine Vorsteuerung	42
Bild B.2 – Beispiel für eine Parameter-Anpassung.....	42
Bild B.3 – Beispiel für eine direkte Fuzzy-Regelung.....	43
Bild C.1 – Industriebeispiel – Containerkran	44
Bild C.2 – Linguistische Variable „Abstand“ zwischen Krankopf und Zielposition.....	45
Bild C.3 – Linguistische Variable „Winkel“ des Containers zum Krankopf	45
Bild C.4 – Linguistische Variable „Leistung“	46
Bild C.5 – Regelwerk	46
Bild C.6 – Fuzzifizierung der linguistischen Variable „Abstand“	47
Bild C.7 – Fuzzifizierung der linguistischen Variable „Winkel“	47
Bild C.8 – Untermenge der drei Regeln.....	48
Bild C.9 – Elemente der Aggregation	48
Bild C.10 – Prinzipien der Aggregation.....	49
Bild C.11 – Elemente der Aktivierung	49
Bild C.12 – Prinzipien der Aktivierung	50
Bild C.13 – Elemente der Akkumulation	50
Bild C.14 – Prinzipien der Akkumulation.....	51
Bild C.15 – Defuzzifizierung.....	52
Bild C.16 – Beispiel in FCL	53
Bild D.1 – Prinzip des geregelten Systems	55
Bild D.2 – Prinzip der Fuzzy-basierten Ofenregelung	55
Bild D.3 – Regelblock	55
Bild D.4 – Beispiel in FCL	56
Tabelle 1 – Defuzzifizierungsmethoden.....	16
Tabelle 2 – Gleichungen für Defuzzifizierungsmethoden	16
Tabelle 3 – Paarweise Algorithmen	17
Tabelle 4 – Aktivierungsmethoden	18
Tabelle 5 – Akkumulationsmethoden.....	18
Tabelle 6 – Priorität der Operatoren	18

Tabelle 7 – Reservierte Schlüsselwörter in FCL	24
Tabelle 8 – (vorgeschriebene) FCL Sprachelemente der Basis-Ebene.....	26
Tabelle 9 – (optionale) FCL Sprachelemente der Erweiterungsebene	27
Tabelle 10 – Beispiele für eine Liste mit Sprachelementen der offenen Ebene	27
Tabelle 11 – Datenprüfliste.....	28
Tabelle A.1 – Inferenz-Schritte und allgemein verwendete Algorithmen	36
Tabelle C.1 – Inferenzschritte und zugeordnete Operatoren	46
Tabelle E.1 – Symbole und Abkürzungen	57
Tabelle E.2 – Synonyme (Englisch)	57

Einleitung

Die Anwendung der Theorie der Fuzzy-Logik auf dem Gebiet der Leittechnik wird als Fuzzy-Control bezeichnet. Fuzzy-Control verbreitet sich als eine Technologie, die die Möglichkeiten der Industrieautomatisierung verbessern kann, und die für Aufgaben aus der Steuerungsebene geeignet ist, die im Allgemeinen von Speicherprogrammierbaren Steuerungen (SPS) ausgeführt werden.

Fuzzy-Control basiert meist auf praktischem Anwendungswissen, das durch die so genannten linguistischen Regelwerke dargestellt wird, und nicht auf analytischen (empirischen oder theoretischen) Modellen. Fuzzy-Control kann angewendet werden, wenn ein regelbasiertes Fachwissen vorhanden ist, das sich in eine für Fuzzy-Logik geeignete Form überführen lässt.

Dadurch wird es möglich, durch vorhandenes Wissen Prozesse zu verbessern und eine Vielzahl von Aufgaben durchzuführen, wie zum Beispiel

- Regelungen und Steuerungen, Einzel- oder Mehrgrößenregelungen, für lineare oder nichtlineare Systeme,
- Online- oder Offline-Vorgabe der Regelungsparameter,
- Klassifizierungsaufgaben und Mustererkennung,
- Entscheidungsfindung zur Echtzeit (Sende dieses Produkt zur Maschine A oder B?),
- Hilfe für den Bediener beim Treffen von Entscheidungen oder beim Einstellen von Parametern,
- Erkennen und Diagnose von Fehlern in Systemen.

Der breite Anwendungsbereich und das natürliche, auf menschlicher Erfahrung basierende Verfahren machen Fuzzy-Control zu einem Grundwerkzeug, das für den Anwender von SPS als Norm verfügbar gemacht werden sollte.

Fuzzy-Control kann auch direkt mit den klassischen Methoden der Steuerung und Regelung kombiniert werden.

Die Anwendung von Fuzzy-Control kann in solchen Fällen vorteilhaft sein, bei denen kein explizites Prozessmodell verfügbar ist, oder bei denen das analytische Modell zu schwierig auszuwerten ist, oder wenn das Modell zu kompliziert ist, um in Echtzeit damit Auswertungen auszuführen.

Ein weiterer Vorteil von Fuzzy-Control ist, dass menschliche Erfahrung in direkter Weise eingebracht werden kann. Es ist dabei nicht erforderlich, die gesamte Regelung mit Fuzzy-Control zu modellieren: Manchmal interpoliert Fuzzy-Control nur zwischen einer Reihe von lokalen linearen Modellen oder passt dynamisch die Parameter eines „linearen Reglers“ an, wodurch der resultierende Regler nichtlinear wird, oder aber sie „schaltet“ sich einfach auf eine bestimmte Eigenschaft eines vorhandenen Reglers, die zu verbessern ist.

Fuzzy-Control ist eine mehrwertige Regelung, welche die Werte eines Regelungsentwurfs nicht mehr auf die Werte „wahr“ oder „falsch“ einschränkt. Dies macht Fuzzy-Control besonders nützlich, um empirisches Fachwissen zu modellieren, indem man festlegt, welche Steuer/Regel-Aktionen bei einer gegebenen Menge von Eingaben unternommen werden.

Die vorhandene Theorie und die Systeme, die auf dem Gebiet von Fuzzy-Control bereits realisiert sind, unterscheiden sich sehr bezüglich Terminologie (Definitionen), Eigenschaften (Funktionalitäten) und Realisierungen (Werkzeuge).

Fuzzy-Control wird von kleinen und einfachen Anwendungen bis hinauf zu sehr anspruchsvollen und komplexen Projekten verwendet. Um alle diese Anwendungsarten durch diesen Teil der IEC 61131 abzudecken, sind die Eigenschaften eines normgerechten Fuzzy-Control-Systems auf definierte Normerfüllungsklassen abgebildet.

Die Basisklasse definiert eine Mindestmenge von Eigenschaften, die von allen normgerechten Systemen erreicht werden müssen. Dies ermöglicht den Austausch von Fuzzy-Control-Programmen.

Zusätzlich mögliche Normeigenschaften sind in der Erweiterungsklasse definiert. Fuzzy-Control-Programme, die diese Eigenschaften anwenden, können nur zwischen Systemen vollständig portiert werden, die dieselbe Menge von Eigenschaften verwenden, sonst kann nur ein Teilaustausch möglich sein. Diese Norm zwingt nicht alle normgerechten Systeme, alle Eigenschaften in der Erweiterungsklasse zu realisieren, aber sie unterstützt die Möglichkeit der (teilweisen) Portierbarkeit und bewahrt vor der Anwendung von nichtgenormten Eigenschaften. Deswegen sollte ein normgerechtes System keine nichtgenormten Eigenschaften anbieten, die sinnvoll durch Verwendung von Normeigenschaften der Grundklasse und der Erweiterungsklasse realisiert werden können.

Um Systeme, die ihre eigenen, besonders anspruchsvollen Eigenschaften anbieten, nicht vom Erfüllen dieses Teils der IEC 61131 auszuschließen, und um den Fortschritt der zukünftigen Entwicklung nicht zu behindern, erlaubt diese Norm auch zusätzliche nichtgenormte Eigenschaften, die nicht durch die Basisklasse und die Erweiterungsklasse abgedeckt sind. Diese Eigenschaften müssen jedoch in einer genormten Weise angegeben werden, um zu gewährleisten, dass sie leicht als nichtgenormte Eigenschaften zu erkennen sind.

Die Übertragbarkeit von Fuzzy-Control-Anwendungen hängt von den unterschiedlichen Programmiersystemen und auch von den Eigenheiten der Steuerungssysteme ab. Diese Abhängigkeiten sind in der Datenprüfliste behandelt, die vom Hersteller mitzuliefern ist.

Speicherprogrammierbare Steuerungen – Teil 7: Fuzzy-Control-Programmierung

1 Anwendungsbereich und Ziel

Dieser Teil der IEC 61131 legt eine Sprache zur Programmierung von Fuzzy-Control-Anwendungen fest, die von Speicherprogrammierbaren Steuerungen verwendet werden.

Das Ziel dieses Teils der IEC 61131 ist es, den Herstellern und Anwendern ein klar definiertes, gemeinsames Verständnis der grundlegenden Mittel zu bieten, um Anwendungen von Fuzzy-Control in die Sprachen der Speicherprogrammierbaren Steuerungen nach IEC 61131-3 zu integrieren, aber auch die Möglichkeit, übertragbare Fuzzy-Control-Programme zwischen verschiedenen Programmiersystemen auszutauschen.

Um dies zu erreichen, gibt Anhang A eine kurze Einführung in die Theorie von Fuzzy-Logik und Fuzzy-Control, soweit es für das Verständnis dieses Teils der IEC 61131 notwendig ist. Für Leser dieses Teils der IEC 61131, die nicht mit der Fuzzy-Control-Theorie vertraut sind, ist es hilfreich, den Anhang A zuerst zu lesen.

2 Normative Verweisungen

Die folgenden normativen Dokumente enthalten Festlegungen, die durch Verweisungen in diesem Text Bestandteil dieses Teils der IEC 61131 sind. Für den Fall einer datierten Verweisung bezieht sich die Verweisung nicht auf nachfolgende Änderungen oder Überarbeitungen dieser Norm. Jedoch werden Vertragspartner, deren Vereinbarungen auf diesem Teil der IEC 61131 basieren, gebeten, die Möglichkeit zu prüfen, ob die jeweils neuesten Ausgaben der im Folgenden genannten Normen angewendet werden können. Für den Fall einer undatierten Verweisung bezieht sich diese auf die jeweils neueste Ausgabe der in Bezug genommenen Norm. Die Mitglieder von ISO und IEC führen Verzeichnisse der gegenwärtig gültigen Internationalen Normen.

IEC 60050-351:1998, *Internationales Elektrotechnisches Wörterbuch – Teil 351: Automatisierungstechnik*

IEC 61131-3:1993, *Speicherprogrammierbare Steuerungen – Teil 3: Programmiersprachen*

3 Begriffe

Für die Anwendung dieses Teils der IEC 61131 gelten die folgenden Begriffe.

ANMERKUNG Begriffe, die in diesem Abschnitt definiert werden, sind im Text kursiv gesetzt.

3.1

Akkumulation (en: accumulation)

Aggregation des Ergebnisses

Zusammenfassen der Ergebnisse der *linguistischen Regeln* zu einem Gesamtergebnis

3.2

Aggregation (en: aggregation)

Bestimmung des Degree of Firing

Kombination der Zugehörigkeitsgrade aller individuellen Teilbedingungen in einer Regel zur Berechnung des Erfüllungsgrades der *Bedingung* einer Regel

3.3

Aktivierung (en: activation)

Vorgang, durch den der Erfüllungsgrad einer *Bedingung* auf eine Fuzzy-Ausgangsmenge wirkt

3.4

Konklusion (en: conclusion)

Consequent

Ergebnis einer *linguistischen Regel*, z. B. die anzustoßenden Aktionen (der THEN-Teil einer IF..THEN-Fuzzy-Control-Regel)

3.5

Bedingung (en: condition)

Antecedent

Ausdruck, der *Teilbedingungen* umfasst, die mit den *Fuzzy-Operatoren* AND, OR und NOT verknüpft sind

3.6

scharfe Menge (en: crisp set)

Spezialfall der *unscharfen (Fuzzy-)Menge*, in welcher die *Zugehörigkeitsfunktion* nur zwei Werte annimmt, üblicherweise als 0 und 1 definiert

3.7

Defuzzifizierung (en: defuzzification)

Umsetzung einer *unscharfen (Fuzzy-)Menge* in eine numerische Größe

3.8

Zugehörigkeitsgrad (en: degree of membership)

Wert der *Zugehörigkeitsfunktion*

3.9

Fuzzifizierung (en: fuzzification)

Bestimmung der *Zugehörigkeitsgrade* eines scharfen Eingangswertes zu den *linguistischen Termen*, die für jede *linguistische Eingangsvariable* definiert sind

3.10

Fuzzy-Control (en: fuzzy control)

Art der Steuerung und Regelung, bei der der Regelungsalgorithmus auf *Fuzzy-Logik* basiert [IEV 351-07-51, modifiziert]

3.11

Fuzzy-Logik (en: fuzzy logic)

Sammlung mathematischer Theorien, die auf der Anwendung von *unscharfen (Fuzzy-)Mengen* basieren. Die *Fuzzy-Logik* ist ein Teilgebiet der mehrwertigen Logik

3.12

Fuzzy-Operator (en: fuzzy logic operator)

Operator, der in der Theorie der *Fuzzy-Logik* verwendet wird

3.13

unscharfe Menge (en: fuzzy set)

eine *unscharfe Menge* A ist definiert als eine Menge geordneter Zahlenpaare $(x, \mu_A(x))$, wobei x ein Element des Wertebereichs U und $\mu_A(x)$ die *Zugehörigkeitsfunktion* ist, die jedem $x \in U$ eine reelle Zahl $\in [0,1]$ zuordnet, die den *Zugehörigkeitsgrad* des Elementes x zu der Menge A beschreibt

3.14

Inferenz (en: inference)

Anwendung der *linguistischen Regeln* auf Eingangswerte, um Ausgangswerte zu erzeugen

3.15

linguistische Regel (en: linguistic rule)

IF-THEN-Regel mit *Bedingung* und *Konklusion*, von denen mindestens eine linguistisch ist

3.16

linguistischer Term (en: linguistic term)

im Zusammenhang mit Fuzzy-Control werden *linguistische Terme* durch *unscharfe Mengen* definiert

3.17

linguistische Variable (en: linguistic variable)

Variable, deren Werte *linguistische Terme* sind

3.18

Zugehörigkeitsfunktion (en: membership function)

Funktion, die den *Zugehörigkeitsgrad* über dem Wertebereich einer gegebenen *unscharfen Menge* definiert [IEV 351-17-52, modifiziert]

3.19

Singleton (en: singleton)

unscharfe Menge, deren *Zugehörigkeitsfunktion* gleich Eins an einer Stelle und gleich Null an allen anderen Stellen ist

3.20

Teilbedingung (en: subcondition)

Elementarer Ausdruck in der Form einer Variablen oder als Term „*linguistische Variable* IS *linguistischer Term*“¹

3.21

Regelwerk (en: rule base)

Gesamtheit der *linguistischen Regeln*, um bestimmte Ziele zu erreichen

3.22

Wichtungsfaktor (en: weighting factor)

Wert zwischen 0..1, der den Grad der Bedeutung, Glaubwürdigkeit, Vertrauenswürdigkeit einer *linguistischen Regel* angibt

4 Integration in die Speicherprogrammierbare Steuerung

Die Fuzzy-Control-Anwendungen, die in Fuzzy-Control-Sprache (FCL) nach Abschnitt 5 programmiert sind, müssen in Funktionsbausteine (oder Programme) eingeschlossen sein, so wie sie in IEC 61131-3 definiert sind. Das Konzept der Funktionsbaustein-Typen und Funktionsbaustein-Instanzen, das in IEC 61131-3 angegeben ist, gilt für diese Norm.

Die Funktionsbaustein-Typen, die in Fuzzy-Control-Sprache (FCL) definiert sind, müssen die Eingangs- und Ausgangsparameter und die für die Fuzzy-Control-Anwendung erforderlichen Regeln und Deklarationen enthalten.

Die zugehörigen Funktionsbaustein-Instanzen müssen die spezifischen Daten der Fuzzy-Control-Anwendungen enthalten.

Funktionsbausteine, die in Fuzzy-Control-Sprache FCL definiert sind, können in Programmen und Funktionsbausteinen verwendet werden, die in einer der Sprachen nach IEC 61131-3 geschrieben sind, z. B. Kontaktplan, Anweisungsliste usw. Die Datentypen der Eingangs- und Ausgangsparameter des in FCL geschriebenen Funktionsbausteins oder Programms müssen mit denen aus der entsprechenden „Aufrufumgebung“ zusammenpassen, so wie es in Bild 1 veranschaulicht ist.

¹ Nationale Fußnote: IS: englischsprachiges Schlüsselwort in der FCL-Sprache

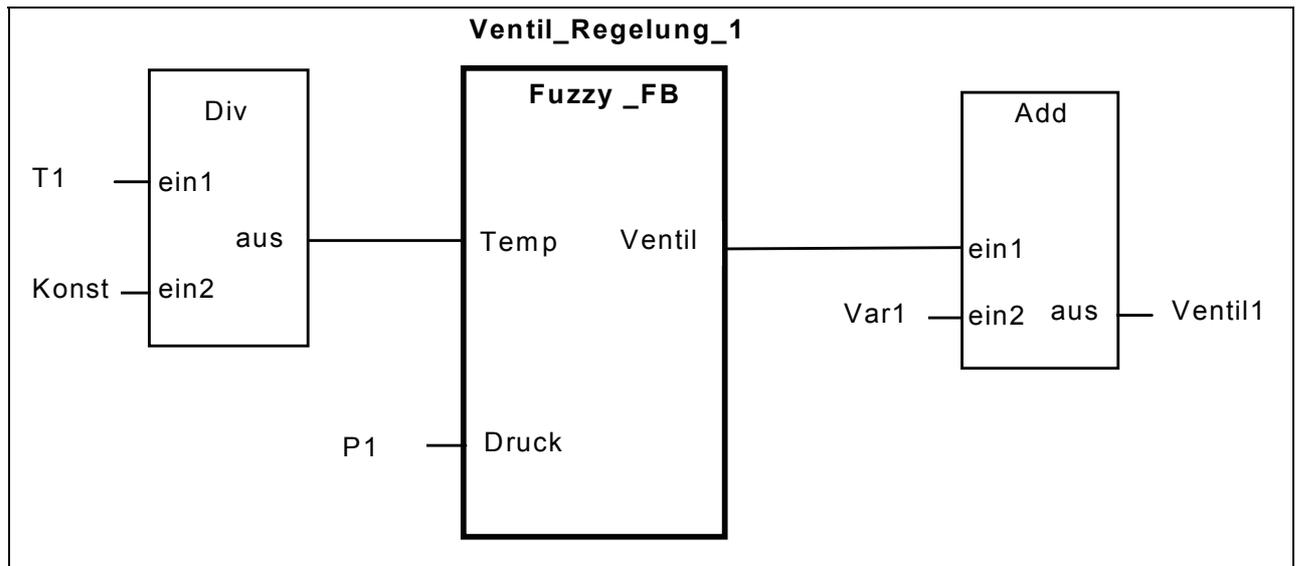


Bild 1 – Beispiel eines Fuzzy-Control-Funktionsbausteins in Funktionsbaustein-Sprachendarstellung (FBS)

In diesem Beispiel ist Ventil_Regelung_1 eine vom Anwender definierte Funktionsbaustein-Instanz des Funktionsbaustein-Typs Fuzzy_FB. Der Funktionsbaustein-Typ Fuzzy_FB kann in Fuzzy-Control-Sprache FCL nach Abschnitt 5 programmiert werden. Der Funktionsbaustein Fuzzy_FB wird hier in einem Programm oder einem Funktionsbaustein verwendet, die in der graphischen Sprache FBS (Funktionsbaustein-Sprache) nach IEC 61131-3 dargestellt sind.

5 Fuzzy-Control-Sprache FCL

5.1 Austausch von Fuzzy-Control-Programmen

Die Definition der Fuzzy-Control-Sprache FCL basiert auf den Definitionen der Programmiersprachen nach IEC 61131-3. Die Interaktion des Fuzzy-Control-Algorithmus mit seiner Programmumgebung geschieht so, dass er vor dem Programm „verborgen“ ist. Der Fuzzy-Control-Algorithmus wird deshalb nach außen als ein Funktionsbaustein nach IEC 61131-3 dargestellt. Die erforderlichen Elemente zum Beschreiben der internen linguistischen Teile des Fuzzy-Control-Funktionsbausteins wie Zugehörigkeitsfunktionen, Regeln, Operatoren und Methoden müssen so festgelegt werden, wie es in diesem Abschnitt definiert ist.

Die Sprachelemente von FCL normen eine gemeinsame Darstellung zum Datenaustausch zwischen Fuzzy-Control-Konfigurierungswerkzeugen verschiedener Hersteller, so wie es in Bild 2 gezeigt ist. Durch Verwendung dieser gemeinsamen Darstellung kann jeder Hersteller von Speicherprogrammierbaren Steuerungen seine Hardware, Software-Editoren und Compiler weiterhin verwenden. Der Hersteller braucht nur die Datenschnittstelle in seinen spezifischen Editor zu implementieren. Der Kunde wird die Fuzzy-Control-Projekte zwischen verschiedenen Herstellern austauschen können.

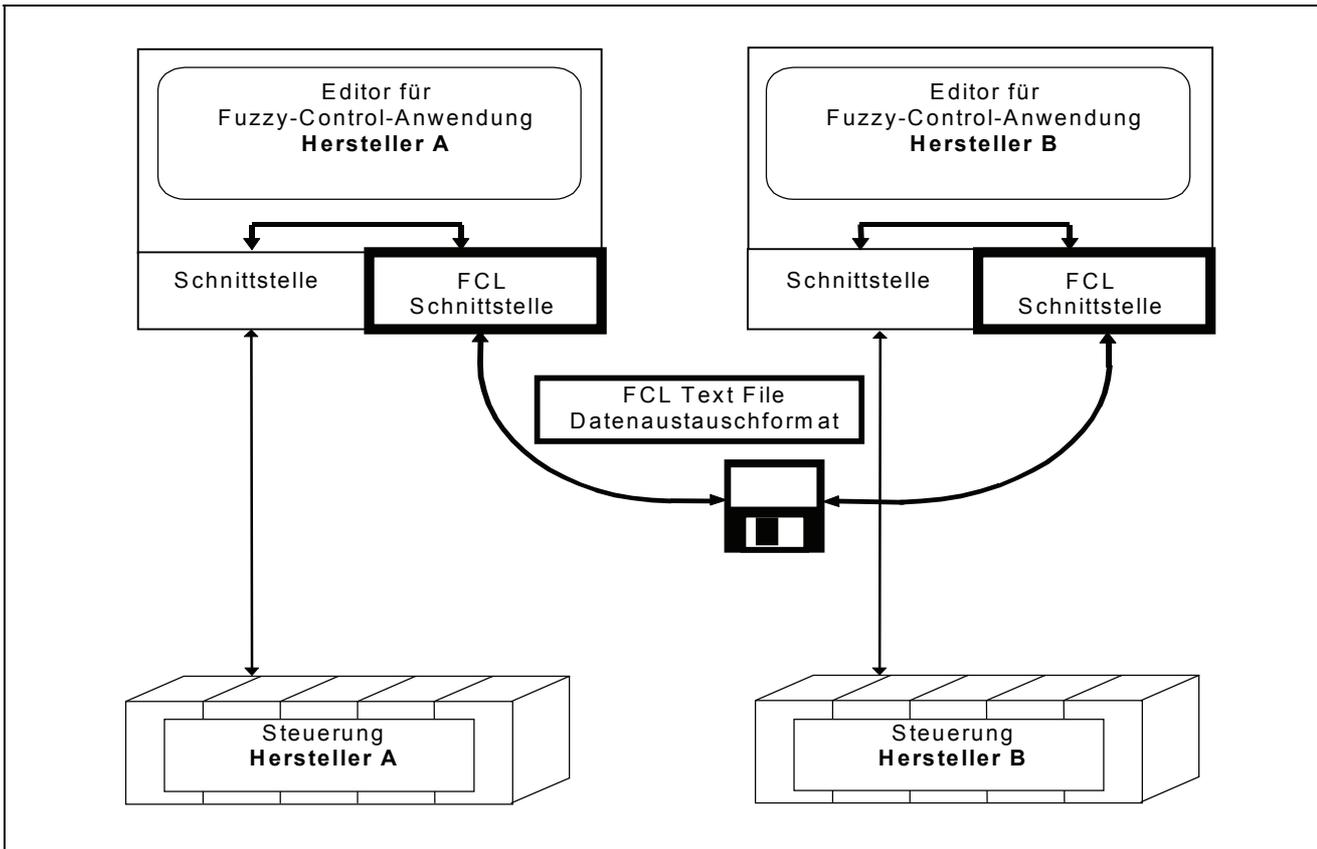


Bild 2 – Datenaustausch von Programmen in Fuzzy-Control-Sprache (FCL)

5.2 Elemente der Fuzzy-Control-Sprache

Die Elemente der Fuzzy-Control-Sprache in diesem Unterabschnitt sind mit Hilfe von Beispielen beschrieben. Die detaillierte Erzeugungsregel ist in 5.4 angegeben.

5.2.1 Funktionsbaustein-Schnittstelle

Nach Abschnitt 4 erfordert die externe Sicht des Fuzzy-Funktionsbausteins, dass die folgenden genormten Sprachelemente der IEC 61131-3 benutzt werden:

FUNCTION_BLOCK <i>Funktionsbaustein_Name</i>	Funktionsbaustein
VAR_INPUT <i>Variablen_Name: Datentyp;</i> END_VAR	Deklaration der Eingangsparameter
VAR_OUTPUT <i>Variablen_Name: Datentyp;</i> END_VAR	Deklaration der Ausgangsparameter
VAR <i>Variablen_Name: Datentyp;</i> END_VAR	Lokale Variablen
END_FUNCTION_BLOCK	

Mit diesen Sprachelementen ist es möglich, eine Funktionsbaustein-Schnittstelle zu beschreiben. Die Funktionsbaustein-Schnittstelle wird mit den Parametern definiert, die in den und aus dem Funktionsbaustein übergeben werden. Die Datentypen dieser Parameter müssen nach IEC 61131-3 definiert werden.

Bild 3 zeigt ein Beispiel für eine Funktionsbaustein-Deklaration in Strukturierterm Text (ST) und Funktionsbaustein-Sprache (FBS).

<pre> FUNCTION_BLOCK Fuzzy-FB VAR_INPUT Temp: REAL; Druck: REAL; END_VAR VAR_OUTPUT Ventil: REAL; END_VAR END_FUNCTION_BLOCK </pre>	
<i>Strukturierter Text (ST)</i>	<i>Funktionsbaustein-Sprache (FBS)</i>

Bild 3 – Beispiel einer Schnittstellendeklaration in Strukturierterm Text und Funktionsbausteinsprache

5.2.2 Fuzzifizierung

Die Werte der Eingangsvariablen müssen in *Zugehörigkeitsgrade* für die *Zugehörigkeitsfunktionen* umgewandelt werden, die für die Variable definiert sind. Diese Umwandlung wird zwischen den Schlüsselwörtern FUZZIFY und END_FUZZIFY beschrieben.

<pre> FUZZIFY <i>Variablen_Name</i> TERM <i>Term_Name</i> := <i>Zugehörigkeits_Funktion</i> ; END_FUZZIFY </pre>

Nach dem Schlüsselwort FUZZIFY muss der Name der Variablen angegeben werden, die für die Fuzzifizierung benutzt wird. Dies ist der Name einer vorher in der VAR_INPUT-Sektion definierten Variablen. Diese *linguistische Variable* muss durch einen oder mehrere *linguistische Terme* beschrieben werden. Die *linguistischen Terme* werden durch das Schlüsselwort TERM eingeleitet und durch *Zugehörigkeitsfunktionen* beschrieben, um die Variable zu fuzzifizieren. Eine *Zugehörigkeitsfunktion* ist eine abschnittsweise lineare Funktion. Sie ist durch eine Liste von Punkten definiert.

Zugehörigkeitsfunktion ::= (Punkt i), (Punkt j), ...

Jeder Punkt ist ein Wertepaar der Variable und des Zugehörigkeitsgrades dieses Wertes, getrennt durch ein Komma. Die Paare sind in Klammern eingeschlossen und durch Kommata getrennt.

Punkt i ::= Wert des Eingangs i | Variablen_Name des Eingangs i , Wert i des Zugehörigkeitsgrades

Mit dieser Definition können alle einfachen Elemente, z. B. Rampe und Dreieck definiert werden. Die Punkte müssen in aufsteigender Reihenfolge des Variablenwertes angegeben werden. Die Zugehörigkeitsfunktion ist zwischen aufeinander folgenden Punkten linear. Der Zugehörigkeitsgrad jedes Terms wird deshalb aus dem scharfen Eingangswert durch lineare Interpolation zwischen den zwei relevanten benachbarten Punkten der Zugehörigkeitsfunktion berechnet.

Die Mindestanzahl von Punkten ist zwei. Die größte Anzahl ist nach den Erfüllungsklassen in Abschnitt 6 begrenzt.

Beispiel einer *Zugehörigkeitsfunktion* mit drei Punkten für den *linguistischen Term* „warm“:

TERM warm := (17.5, 0.0) (20.0, 1.0) (22.5, 0.0);

Wenn der Wert einer *linguistischen Variablen* kleiner als der erste Basispunkt in der Tabelle ist, müssen alle Werte unterhalb des ersten Punkts in der Tabelle denselben *Zugehörigkeitsgrad* haben, wie er für den ersten Punkt definiert ist.

Wenn der Wert einer *linguistischen Variablen* größer als der letzte Basispunkt in der Tabelle ist, müssen alle Werte größer als der letzte Punkt in der Tabelle denselben *Zugehörigkeitsgrad* haben, wie er für den letzten Punkt definiert ist.

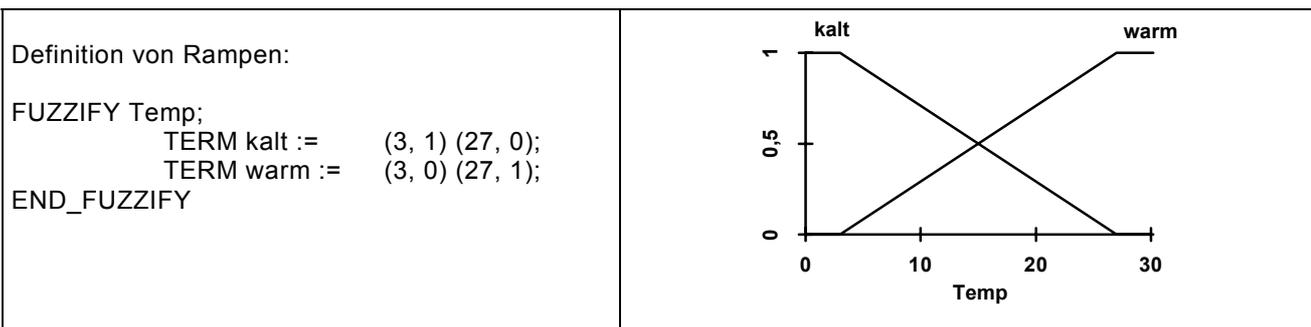


Bild 4 – Beispiel für Rampen-Terme

ANMERKUNG Der Datentyp der Punkte der Zugehörigkeitsfunktionen ist nicht definiert. Der Hersteller muss einen Compiler zu Verfügung stellen, der die notwendige Umwandlung ermöglicht.

Um die Fuzzy-Control-Anwendung on-line anzupassen, können die Basispunkte in den Zugehörigkeitsfunktionen verändert werden. Dies kann durch Verwendung von Variablen geschehen, die Eingänge des Funktionsbausteins sind. Diese Variablen müssen in der VAR_INPUT Sektion des Funktionsbausteins deklariert werden. Ein Beispiel für den Gebrauch von Variablen zur Definition der Punkte der Zugehörigkeitsfunktionen ist in Bild 5 angegeben.

ANMERKUNG Die Werte der Punkte der Zugehörigkeitsfunktionen können zur Ablaufzeit außerhalb der Sequenz liegen.

```

VAR_INPUT
    Temp: REAL;           (* Dieser Eingang muss fuzzifiziert werden *)
    Druck: REAL;          (* Dieser Eingang muss fuzzifiziert werden *)
    bp_warm1, bp_warm2: REAL; (* Diese Eingänge dienen zur on-line-Anpassung *)
END_VAR
FUZZIFY Temp
    TERM warm:= (bp_warm1, 0.0), (21.0, 1.0), (bp_warm2, 0.0);
...
END_FUZZIFY

```

Bild 5 – Beispiel einer Verwendung von Variablen für Zugehörigkeitsfunktionen

5.2.3 Defuzzifizierung

Eine *linguistische Variable* für eine Ausgangsvariable muss in einen numerischen Wert umgewandelt werden. Diese Umwandlung wird zwischen den Schlüsselwörtern DEFUZZIFY und END_DEFUZZIFY beschrieben.

Nach dem Schlüsselwort DEFUZZIFY muss die Variable angegeben werden, die zur *Defuzzifizierung* verwendet wird. Dies ist der Name einer vorher definierten Variablen in der VAR_OUTPUT-Sektion.

```

DEFUZZIFY Variablen_Name
    RANGE(min..max) ;
    TERM Term_Name := Zugehörigkeitsfunktion ;
    Defuzzifizierungsmethode ;
    Voreinstellungswert ;
END_DEFUZZIFY

```

Die Definition von *linguistischen Termen* ist in 5.2.2 angegeben.

Singletons (Einermengen) sind besondere *Zugehörigkeitsfunktionen*, die für Ausgänge benutzt werden, um die *Defuzzifizierung* zu vereinfachen. Sie werden nur durch einen einzelnen Wert für den *linguistischen Term* beschrieben. In Bild 6 sind Beispiele für solche Terme angegeben.

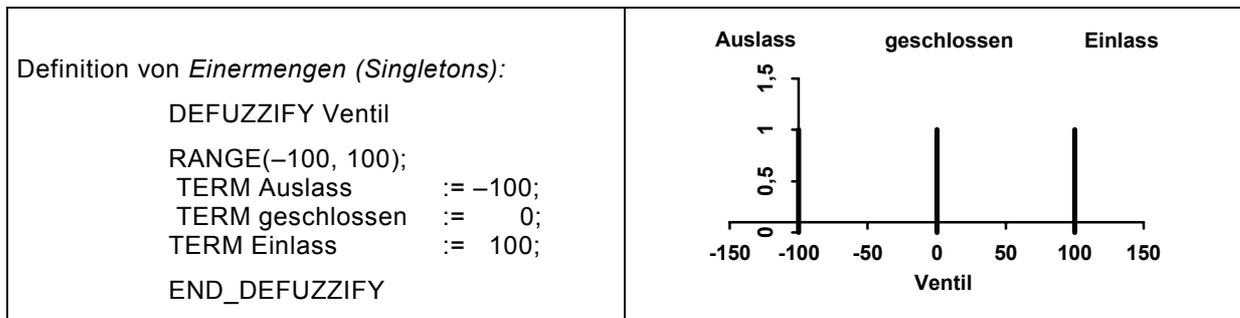


Bild 6 – Beispiel von Singleton-Termen

Die Defuzzifizierungsmethode muss durch das Sprachelement METHOD definiert werden.

```
METHOD : Defuzzifizierungsmethode ;
```

Verwendet werden dürfen die folgenden Defuzzifizierungsmethoden (siehe Tabellen 1 und 2).

Tabelle 1 – Defuzzifizierungsmethoden

Schlüsselwort	Erläuterung
CoG (Centre of Gravity)	Schwerpunkt (Anmerkung 1)
CoGS (Centre of Gravity for Singletons)	Schwerpunkt für Singletons
CoA (Centre of Area)	Flächenschwerpunkt (Anmerkungen 2 und 3)
LM (Left Most Maximum)	Linkes Maximum (Anmerkung 4)
RM (Right Most Maximum)	Rechtes Maximum (Anmerkung 4)
ANMERKUNG 1	Im Englischen ist „Center of Gravity“ gleichbedeutend mit „Centroid of Area“.
ANMERKUNG 2	Im Englischen ist „Center of Area“ gleichbedeutend mit „Bisector of Area“.
ANMERKUNG 3	CoA ist nicht anwendbar, wenn Singletons verwendet werden.
ANMERKUNG 4	LM- und RM-Defuzzifizierungsmethoden sind asymmetrisch bei Null.

Tabelle 2 – Gleichungen für Defuzzifizierungsmethoden

COG	$U = \frac{\int_{\text{Min}}^{\text{Max}} u \hat{\mu}(u) du}{\int_{\text{Min}}^{\text{Max}} \hat{\mu}(u) du}$
COGS	$U = \frac{\sum_{i=1}^p [u_i \hat{\mu}_i]}{\sum_{i=1}^p [\hat{\mu}_i]}$
COA	$U = u', \int_{\text{Min}}^{u'} \hat{\mu}(u) du = \int_{u'}^{\text{Max}} \hat{\mu}(u) du$
RM	$U = \sup(u'), \hat{\mu}(u') = \sup_{u \in [\text{Min}, \text{Max}]} \hat{\mu}(u)$
LM	$U = \inf(u'), \hat{\mu}(u') = \inf_{u \in [\text{Min}, \text{Max}]} \hat{\mu}(u)$

wobei:

- U das Ergebnis der Defuzzifizierung ist;
- u die Ausgangsvariable ist;
- p die Anzahl der Einermengen (Singletons) ist;
- μ die Zugehörigkeitsfunktion der akkumulierten unscharfen Mengen ist;
- i der Index ist
- Min der kleinste Wert für die Defuzzifizierung ist, der in RANGE definiert ist.
Im Fall von Einermengen (Singletons) Min = -unendlich;
- Max der größte Wert für die Defuzzifizierung ist, der in RANGE definiert ist.
Im Fall von Einermengen (Singletons) Max = +unendlich;
- sup der größte Wert ist;
- in der kleinste Wert ist.

Falls der *Zugehörigkeitsgrad* für alle *linguistischen Terme* einer Ausgangsvariablen 0 ist, bedeutet das, dass keine Regel für diese Variable aktiv ist. In diesem Fall ist die *Defuzzifizierung* nicht in der Lage, einen gültigen Wert zu erzeugen. Deswegen darf ein Voreinstellungswert für den Ausgang definiert werden. Dieser Voreinstellungswert ist der Wert für die Ausgangsvariable, falls keine Regel aktiv ist.

DEFAULT := Wert | NC;

Nach dem Schlüsselwort DEFAULT (Voreinstellung) muss der Wert festgelegt werden. Ansonsten muss das Schlüsselwort NC (no change, keine Änderung) festgelegt werden, um anzuzeigen, dass der Ausgang gegenüber dem vorherigen Zeitpunkt unverändert bleibt, falls keine Regel aktiv ist.

Das Schlüsselwort RANGE (Bereich) dient der Festlegung eines kleinsten und größten Wertes, die durch zwei Punkte getrennt werden.

RANGE := (Minimalwert .. Maximalwert);

Der Bereich (RANGE) wird zur Begrenzung der jeweiligen Zugehörigkeitsfunktion auf den Bereich der jeweiligen Ausgangsvariable verwendet. Falls Einermengen für Zugehörigkeitsfunktionen des Ausgangs verwendet werden, hat der Bereich (RANGE) keine Wirkung.

Falls kein Bereich (RANGE) definiert ist, muss der voreingestellte Bereich der Bereich des Datentyps der Variablen sein, so wie es in IEC 61131-3 festgelegt ist.

5.2.4 Regelblock

Die *Inferenz* (Bearbeitung) des Fuzzy-Algorithmus muss in einem oder mehreren Regelblöcken definiert werden. Zum richtigen Hantieren und um die Möglichkeiten zu bieten, das Regelwerk in verschiedene Module aufzuteilen, ist die Verwendung von mehreren Regelblöcken erlaubt. Jeder Regelblock muss einen eindeutigen Namen haben.

Die Regeln müssen zwischen den Schlüsselwörtern RULEBLOCK und END_RULEBLOCK definiert werden.

```

RULEBLOCK Regelblock_Name

  Operator_Definition ;
  [Aktivierungsmethode ;]
  Akkumulationsmethode ;
  Regeln ;

END_RULEBLOCK
    
```

Die Fuzzy-Operatoren werden innerhalb des Regelblocks verwendet.

Operator_Definition ::= *Operator* : *Algorithmus*

Zur Erfüllung des de Morgan-Gesetzes müssen Algorithmen für die Operatoren AND und OR paarweise verwendet werden, z. B. muss MAX für OR verwendet werden, wenn MIN für AND verwendet wird.

Tabelle 3 – Paarweise Algorithmen

Operator OR		Operator AND	
Schlüsselwort für Algorithmus	Algorithmus	Schlüsselwort für Algorithmus	Algorithmus
MAX	$\text{MAX}(\mu_1(x), \mu_2(x))$	MIN	$\text{MIN}(\mu_1(x), \mu_2(x))$
ASUM	$\mu_1(x) + \mu_2(x) - \mu_1(x) \mu_2(x)$	PROD	$\mu_1(x) \mu_2(x)$
BSUM	$\text{MIN}(1, \mu_1(x) + \mu_2(x))$	BDIF	$\text{MAX}(0, \mu_1(x) + \mu_2(x) - 1)$

Ein Beispiel für Regelblöcke:

```

RULEBLOCK erster
  AND : MIN;
  ..
END_RULEBLOCK

RULEBLOCK zweiter
  AND : PROD;
  ..
END_RULEBLOCK
    
```

Das folgende Sprachelement definiert die Methode der Aktivierung:

ACT : *Aktivierungsmethode*;

Die folgenden Methoden der Aktivierung können verwendet werden (siehe Tabelle 4):

Tabelle 4 – Aktivierungsmethoden

Name	Schlüsselwort	Algorithmus
Produkt	PROD	$\mu_1(x) \cdot \mu_2(x)$
Minimum	MIN	$\min(\mu_1(x), \mu_2(x))$
ANMERKUNG: Die Aktivierungsmethode ist nicht relevant für Singletons.		

Das folgende Sprachelement definiert die Methode für die Akkumulation:

ACCU : *Akkumulationsmethode*;

Die folgenden Akkumulationsmethoden dürfen verwendet werden (siehe Tabelle 5):

Tabelle 5 – Akkumulationsmethoden

Name	Schlüsselwort	Formel
Maximum	MAX	$\max(\mu_1(x), \mu_2(x))$
Begrenzte Summe	BSUM (bounded sum)	$\min(1, \mu_1(x) + \mu_2(x))$
Normalisierte Summe	NSUM (normalised sum)	$\frac{\hat{i}_1(x) + \hat{i}_2(x)}{\max(1, \max_{x' \in X} (\hat{i}_1(x') + \hat{i}_2(x')))}$

Die Eingänge eines Regelblocks sind *linguistische Variablen* mit einer Menge von *linguistischen Termen*. Jeder Term hat einen zugeordneten *Zugehörigkeitsgrad*.

Innerhalb des Regelblocks werden die Regeln definiert. Jeder Regelblock beginnt mit dem Schlüsselwort RULE, gefolgt von einem Namen für die Regel, und muss mit einem Semikolon abgeschlossen sein. Jede Regel hat eine eindeutige Nummer innerhalb des Regelblocks.

RULE *Nummern* : IF *Bedingung* THEN *Konklusion* [WITH *Wichtungsfaktor*];

Die Regel selbst muss mit dem Schlüsselwort IF beginnen, gefolgt von der *Bedingung*. Nach der *Bedingung* folgt die *Konklusion*, die mit dem Schlüsselwort THEN beginnt.

Es ist erlaubt, mehrere *Teilbedingungen* und Eingangsvariablen in einer Regel zu verknüpfen. Der Grund für die Anwendung von Variablen ist, unscharfe Zugehörigkeitsgrade zu ermöglichen, die in die Fuzzy-Funktionsbausteine importiert werden. Diese müssen alle zwischen den Schlüsselwörtern IF und THEN definiert und durch die Operatoren mit den Schlüsselwörtern AND, OR oder NOT verknüpft werden.

Die Priorität der Operatoren wird gemäß der booleschen Algebra gehandhabt, die in Tabelle 6 angegeben ist.

Tabelle 6 – Priorität der Operatoren

Priorität	Operator
1	() Klammern
2	NOT
3	AND
4	OR

Vereinfachtes Beispiel für eine Regel:

RULE 1 : IF *Teilbedingung 1* AND *Variable1* OR *Variable 2* THEN *Konklusion*;

In der Basisebene der Normerfüllung darf der OR Operator durch Definieren von zwei Regeln realisiert werden:

```
RULE 3 : IF Teilbedingung 1 OR Teilbedingung 2 THEN Konklusion;  
ersetzt durch:  
RULE 3a : IF Bedingung 1 THEN Konklusion;  
RULE 3b : IF Bedingung 2 THEN Konklusion;
```

Die Teilbedingung beginnt mit dem Namen einer *linguistischen Variablen*, gefolgt vom Schlüsselwort IS mit einem optionalen NOT und einem *linguistischen Term* der *linguistischen Variablen*, die in der *Bedingung* verwendet wird.

```
Teilbedingung := linguistische_Variable IS [NOT] linguistischer_Term
```

Die *linguistischen Terme*, die in der *Bedingung* verwendet werden, müssen mit der *linguistischen Variablen* in der gleichen *Bedingung* zusammenpassen. Der verwendete Term muss vorher mit dem Schlüsselwort TERM definiert sein.

Beispiel für Teilbedingungen:

```
Temp IS heiss  
Temp IS NOT heiss
```

Es ist auch erlaubt, das Schlüsselwort NOT vor der *Teilbedingung* zu verwenden. In diesem Fall dürfen Klammern verwendet werden.

```
IF NOT Temp IS heiss THEN ...    oder    IF NOT (Temp IS heiss) THEN ...
```

Die *Konklusion* darf in mehrere Teilkonklusionen und Ausgangsvariable geteilt werden.

Die Teilkonklusionen beginnen mit dem Namen einer *linguistischen Variablen*, gefolgt vom Schlüsselwort IS und einem *linguistischen Term* der angegebenen *linguistischen Variablen*.

```
Teilkonklusion := linguistische_Variable IS linguistischer_Term
```

Beispiele mit mehreren Teilkonklusionen in einer oder mehreren Zeilen:

```
IF Temp IS kalt AND Druck IS niedrig THEN var1, Ventil1 IS Einlass , Ventil2 IS geschlossen;  
oder in mehreren Zeilen:  
IF Temp IS kalt AND Druck IS niedrig  
THEN    var1,  
        Ventil1 IS Einlass,  
        ventil2 IS geschlossen;
```

Zusätzlich ist es erlaubt, jeder Teilkonklusion einen *Wichtungsfaktor* zu geben, der eine Zahl mit einem Wert zwischen 0.0 und 1.0 oder eine Variable ist. Dies muss durch das Schlüsselwort WITH geschehen, gefolgt vom *Wichtungsfaktor*.

Der *Wichtungsfaktor* muss den *Zugehörigkeitsgrad* (*Zugehörigkeitsfunktion*) einer Teilkonklusion durch Multiplikation des Ergebnisses in der Teilkonklusion mit dem *Wichtungsfaktor* verringern.

Um die Parameter der Fuzzy-Control-Anwendung von außen zu beeinflussen, darf der *Wichtungsfaktor* eine Variable sein. In diesem Fall muss die Variable in der VAR_INPUT Sektion deklariert werden. Dies bietet die Möglichkeit, den *Wichtungsfaktor* während der Laufzeit zu verändern, um das Fuzzy-Control-Programm den Prozessanforderungen anzupassen.

Falls es keine WITH-Anweisung gibt, die der Teilbedingung zugewiesen ist, muss ein voreingestellter *Wichtungsfaktor* von 1.0 angenommen werden.

```
IF Bedingung THEN Teilkonklusion [ WITH Wichtungsfaktor ] Teilkonklusion ;
```

Ein Beispiel für einen konstanten *Wichtungsfaktor*:

```
IF Temp IS kalt AND Druck IS niedrig THEN Ventil1 IS Einlass WITH 0.5,  
Ventil2 IS geschlossen;
```

Ein Beispiel für einen variablen *Wichtungsfaktor*:

```
VAR_INPUT  
w_meine_Regel1 : REAL := 0.8;  
END_VAR  
  
RULEBLOCK Temp_Regel  
RULE 1: IF Temp IS kalt AND Druck IS niedrig  
THEN Ventil IS Einlass WITH w_meine_Regel1; ..  
END_RULEBLOCK
```

5.2.5 Optionale (zusätzlich mögliche) Parameter

Für die Implementierung auf verschiedenen Zielsystemen kann es erforderlich sein, dem System zusätzliche Informationen zu geben, um die bestmögliche Umwandlung der Fuzzy-Control-Anwendungen zu gewährleisten.

Solche Informationen dürfen mit einem Sprachelement angefordert werden, das durch OPTIONS und END_OPTIONS eingeschlossen ist.

```
OPTIONS  
anwendungsspezifische_Parameter  
END_OPTIONS
```

Diese Sprachelemente müssen für Eigenschaften in der Normerfüllungsklasse der offenen Ebene nach Abschnitt 6 verwendet werden.

5.3 FCL-Beispiel

Ein Beispiel in Fuzzy-Control-Sprache ist in Bild 7 angegeben.

```
FUNCTION_BLOCK Fuzzy_FB
VAR_INPUT
    Temp : REAL;
    Druck : REAL;
END_VAR
VAR_OUTPUT
    Ventil : REAL;
END_VAR
FUZZIFY Temp
    TERM kalt := (3, 1) (27, 0);
    TERM heiss := (3, 0) (27, 1);
END_FUZZIFY
FUZZIFY Druck
    TERM niedrig := (55, 1) (95, 0);
    TERM hoch := (55, 0) (95, 1);
END_FUZZIFY
DEFUZZIFY Ventil
    TERM Auslass := -100;
    TERM geschlossen := 0;
    TERM Einlass := 100;
    METHOD : CoGS;
    DEFAULT := 0;
END_DEFUZZIFY
RULEBLOCK No1
    AND : MIN;
    ACCU : MAX;
    RULE 1 : IF Temp IS kalt AND Druck IS niedrig THEN Ventil IS Einlass;
    RULE 2 : IF Temp IS kalt AND Druck IS hoch THEN Ventil IS geschlossen WITH 0.8;
    RULE 3 : IF Temp IS heiss AND Druck IS niedrig THEN Ventil IS geschlossen;
    RULE 4 : IF Temp IS heiss AND Druck IS hoch THEN Ventil IS Auslass;
END_RULEBLOCK
END_FUNCTION_BLOCK
```

Bild 7 – Beispiel für einen Fuzzy-Funktionsbaustein

5.4 Erzeugungsregeln und Schlüsselwörter der Fuzzy-Control-Sprache (FCL)

Anhang A der IEC 61131-3 definiert die Spezifikationsmethode für textuelle Sprachen für Speicherprogrammierbare Steuerungen. Diese Spezifikationsmethode wird hier für FCL verwendet.

Anhang B der IEC 61131-3 definiert die formale Festlegung von Sprachelementen für die textuellen Programmiersprachen der IEC 61131-3. Für FCL wird die folgende Untermenge von Sprachelementen des Anhangs B der IEC 61131-3 verwendet:

- B.1.1 Buchstaben, Zahlen, Bezeichner
- B.1.2 Konstante
- B.1.3 Datentypen
- B.1.4 Variablen

5.4.1 Erzeugungsregeln

Zusätzlich zu den oben aufgezählten Sprachelementen der IEC 61131-3 dürfen die folgenden Sprachelemente verwendet werden:

function_block_declaration ::= 'FUNCTION_BLOCK' function_block_name
 {fb_io_var_declarations}
 {other_var_declarations}
 function_block_body
 'END_FUNCTION_BLOCK'

fb_io_var_declarations ::= input_declarations | output_declarations

other_var_declarations ::= var_declarations

function_block_body ::= {fuzzify_block}
 {defuzzify_block}
 {rule_block}
 {option_block}

fuzzify_block ::= 'FUZZIFY' variable_name
 {linguistic_term}
 'END_FUZZIFY'

defuzzify_block ::= 'DEFUZZIFY' f_variable_name
 [range]
 {linguistic_term}
 defuzzification_method
 default_value
 'END_DEFUZZIFY'

rule_block ::= 'RULEBLOCK' rule_block_name
 operator_definition
 [activation_method]
 accumulation_method
 {rule}
 'END_RULEBLOCK'

option_block ::= 'OPTION'
 beliebige herstellenspezifische Parameter
 'END_OPTION'

linguistic_term ::= 'TERM' term_name ':=' membership_function ';'

membership_function ::= singleton | points

ANMERKUNG Zur Verwendung von Einermengen (Singletons) siehe 5.2.3.

singleton ::= numeric_literal | variable_name

points ::= {'(' numeric_literal | variable_name ',' numeric_literal ')'}

ANMERKUNG Siehe 5.2.2 für die zulässige Anzahl von Punkten.

defuzzification_method ::= 'METHOD' ':' 'CoG' | 'CoGS' | 'CoA' | 'LM' | 'RM' ':'

default_value ::= 'DEFAULT' ':=' numeric_literal | 'NC' ':'

range ::= 'RANGE('numeric_literal '..' numeric_literal') ':'

operator_definition ::= [('OR' ':' 'MAX' | 'ASUM' | 'BSUM')]
 [('AND' ':' 'MIN' | 'PROD' | 'BDIF') ':'

ANMERKUNG Siehe Tabelle 3 für paarweise Algorithmen.

activation_method ::= 'ACT' ':' 'PROD' | 'MIN' ';' ;
accumulation_method ::= 'ACCU' ':' 'MAX' | 'BSUM' | 'NSUM' ';' ;
rule ::= 'RULE' integer_literal ':'
'IF' condition 'THEN' conclusion [WITH weighting_factor] ';' ;
condition ::= x {('AND' x)|('OR' x)}
x ::= ['NOT'] (subcondition | ('(' condition ')'))
subcondition ::= variable_name | (variable_name 'IS' ['NOT'] term_name)
conclusion ::= { (variable_name | (variable_name 'IS' term_name)) ';' }
(variable_name | variable_name 'IS' term_name)
weighting_factor ::= variable | numeric_literal
function_block_name ::= identifier
ruleblock_name ::= identifier
term_name ::= identifier
f_variable_name ::= identifier
variable_name ::= identifier
numeric_literal ::= integer_literal | real_literal
input_declarations ::= siehe IEC 61131-3, Anhang B
output_declarations ::= siehe IEC 61131-3, Anhang B
var_declarations ::= siehe IEC 61131-3, Anhang B
identifier ::= siehe IEC 61131-3, Anhang B

5.4.2 Schlüsselwörter

Tabelle 7 – Reservierte Schlüsselwörter in FCL

Schlüsselwörter	Bedeutung	Unterabschnitt
()	Klammern in Bedingung, Term, Bereich	5.2.4
ACCU	Akkumulationsmethode	5.2.4
ACT	Aktivierungsmethode	5.2.4
AND	AND-Operator	5.2.4
ASUM	OR-Operator, Algebraische Summe	5.2.4
BDIF (Bounded difference)	AND-Operator	5.2.4
BSUM (Bounded sum)	Akkumulationsmethode, OR-Operator	5.2.4
CoA (Centre of area)	Flächenschwerpunkt, Defuzzifizierungsmethode	5.2.3
CoG (Centre of gravity)	Schwerpunkt, Defuzzifizierungsmethode	5.2.3
CoGS (Centre of gravity of singletons)	Schwerpunkt, Defuzzifizierung für Singletons	5.2.3
DEFAULT	voreingestellter Ausgangswert, falls keine Regel aktiv war	5.2.3
DEFUZZIFY	Defuzzifizierung der Ausgangsvariablen	5.2.3
END_DEFUZZIFY	Ende der Festlegungen für die Defuzzifizierung	5.2.3
END_FUNCTION_BLOCK	Ende der Festlegungen für den Funktionsbaustein	5.2.1
END_FUZZIFY	Ende der Festlegungen für die Fuzzifizierung	5.2.2
END_OPTIONS	Ende der Festlegungen für die Optionen	5.2.5
END_RULEBLOCK	Ende der Festlegungen für den Regelblock	5.2.4
END_VAR	Ende der Definitionen der lokalen Variablen sowie der Eingangs- und Ausgangsvariablen	5.2.1
FUNCTION_BLOCK	Festlegungen des Funktionsbausteins	5.2.1
FUZZIFY	Fuzzifizierung der Eingangsvariablen	5.2.2
IF	Beginn der Regel, der eine Bedingung folgt	5.2.4
IS	folgt der linguistischen Variablen in Bedingung und Konklusion	5.2.4
LM	Defuzzifizierungsmethode Linkes Maximum	5.2.3
MAX	Maximum-Akkumulationsmethode, OR-Operator	5.2.4
METHOD	Defuzzifizierungsmethode	5.2.3
MIN	Aktivierungsmethode, Minimum als AND-Operator	5.2.4
NC	Keine Änderung der Ausgangsvariablen, falls keine Regel aktiv war.	5.2.3
NOT	NOT-Operator	5.2.4
NSUM	Akkumulationsmethode normalisierte Summe	5.2.4
OPTIONS	Definition der optionalen (zusätzlich möglichen) Parameter	5.2.5
OR	OR-Operator	5.2.4
PROD	Aktivierungsmethode, Produkt als AND-Operator	5.2.4
RANGE	Grenzen der Zugehörigkeitsfunktionen	5.2.3
RM	Defuzzifizierungsmethode Rechtes Maximum	5.2.3
RULE	Beginn der Festlegung der Fuzzy-Regel	5.2.4
RULEBLOCK	Beginn der Festlegung des Regelblocks	5.2.4
TERM	Definition eines linguistischen Terms (Zugehörigkeitsfunktion) für eine linguistische Variable	5.2.2
THEN	Trennt Bedingung von Konklusion	5.2.4
VAR	Definition der lokalen Variable(n)	5.2.1
VAR_INPUT	Definition der Eingangsvariable(n)	5.2.1
VAR_OUTPUT	Definition der Ausgangsvariable(n)	5.2.1
WITH	Definition des Wichtungsfaktors	5.2.4

6 Normerfüllung

6.1 Erfüllungsklassen der Fuzzy-Control-Sprache FCL

Die Ebenen der Normerfüllung für Steuerungssysteme, die die Fuzzy-Control-Sprache (FCL) verwenden, sind in Bild 8 gezeigt. Die Hierarchie besteht aus den folgenden drei Ebenen:

- Die Basisebene, die die Definitionen des Funktionsbausteins und der Datentypen nach IEC 61131-3 einschließt.
- Die Erweiterungsebene mit zusätzlich möglichen (optionalen) Eigenschaften, die in der Tabelle 9 aufgeführt und in dieser Ebene erlaubt sind.
- Die offene Ebene, die zusätzliche Eigenschaften umfasst, die nicht in diesem Teil der IEC 61131, sondern vom Hersteller festgelegt sind.

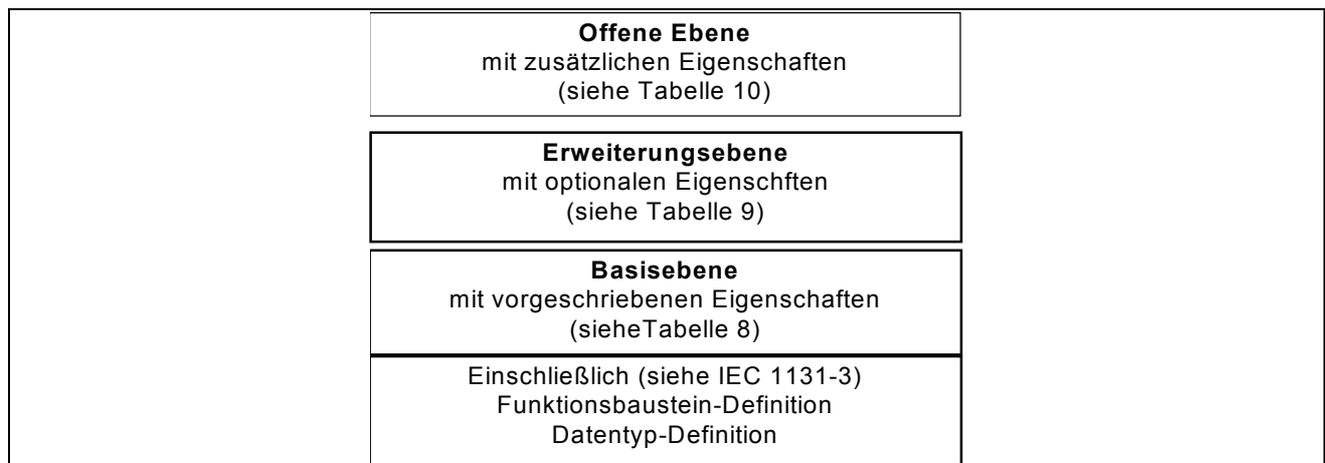


Bild 8 – Ebenen der Normerfüllung

Ein Steuerungssystem, das die Fuzzy-Control-Sprache (FCL) verwendet und den Anspruch erhebt, diesen Teil der IEC 61131 zu erfüllen, muss folgende Regeln einhalten:

- a) Es muss die Eigenschaften der Funktionsbausteine nach IEC 61131-3 verwenden, um die Funktionalität von Fuzzy-Control zu realisieren. Deshalb muss die Definition von Funktionsbausteinen und Datentypen, die für die Eingangs- und Ausgangsparameter von Fuzzy-Control-Funktionsbausteinen erforderlich sind, in Übereinstimmung mit der IEC 61131-3 sein.
- b) Alle Eigenschaften der Fuzzy-Control-Funktionalität, die in Tabelle 8 festgelegt sind, müssen nach den Festlegungen dieses Teils realisiert werden. Diese Tabelle definiert die Menge von Elementen der Basisebene, die für alle Steuerungssysteme gemeinsam gelten müssen, die mit diesem Teil der IEC 61131 übereinstimmen.
- c) Eine Teilmenge der Elemente der Erweiterungsebene, die in Tabelle 9 definiert sind, sind zusätzliche Elemente, die zusätzlich zu den Eigenschaften der Basis-Ebene realisiert werden dürfen. Die Realisierung muss genau nach den Festlegungen dieses Teils erfolgen. Diese Eigenschaften müssen als genormte Erweiterungen gekennzeichnet sein, und es muss eine Liste der realisierten Eigenschaften in Form der Tabelle 9 als Teil der Systemdokumentation vorhanden sein.
- d) Weitere Eigenschaften, die über die Basis-Ebene und die Erweiterungsebene hinausgehen, dürfen realisiert werden, wenn diese Eigenschaften nicht dieselbe oder ähnliche Funktionalität oder Darstellung der genormten Eigenschaften besitzen, um so jede mögliche Verwirrung zu vermeiden. Diese Eigenschaften müssen als Eigenschaften der offenen Ebene gekennzeichnet sein und eine Liste in Form der Tabelle 10 muss Teil der Systemdokumentation sein.
- e) Der Austausch von Anwendungsprogrammen zwischen Fuzzy-Control-Systemen muss in der textuellen Form der Fuzzy-Control-Sprache FCL nach den Festlegungen dieser Norm erfolgen. Dieses Format muss durch Systeme, die normkonform zu diesem Teil der IEC 61131 sind, als Eingabe- und Ausgabe-Format zur Verfügung gestellt werden.

- f) Um die Benutzerschnittstelle möglichst komfortabel und passend zu gestalten und den zukünftigen Fortschritt nicht zu behindern, darf das äußere Erscheinungsbild für den Entwurf, die Eingabe, das Testen usw. von Fuzzy-Control-Anwendungsprogrammen mit beliebigen graphischen oder textuellen Mitteln realisiert werden.

Die Elemente in der Tabelle 8 sind die Grundmenge der Eigenschaften, die von allen Fuzzy-Control-Systemen realisiert werden müssen, die mit diesem Teil der IEC 61131 übereinstimmen.

Tabelle 8 – (vorgeschriebene) FCL Sprachelemente der Basis-Ebene

Sprachelement	Schlüsselwort	Einzelheiten
Funktionsbaustein-Deklaration	VAR_INPUT, VAR_OUTPUT	Enthält Eingangs- und Ausgangsvariablen
Zugehörigkeitsfunktion	Eingangsvariable : TERM	Höchstens 3 konstante Punkte (Zugehörigkeitsgrad-Koordinate = 0 oder 1)
	Ausgangsvariable : TERM	Nur konstante Einermengen (Singletons)
bedingte Aggregation	Operator: AND	Algorithmus : MIN
Aktivierung	-	Nicht relevant, weil nur Einermengen verwendet werden.
Akkumulation (Ergebnis Aggregation)	Operator: ACCU	Algorithmus : MAX
Defuzzifizierung	METHOD	Algorithmus : CoGS
Voreinstellungswert	DEFAULT	NC, Wert
Regelblock	RULEBLOCK	Nur ein Regelblock
Bedingung	IF ... IS ...	n Teilbedingungen
Konklusion	THEN	Nur eine Teilkonklusion

Die Elemente in der Tabelle 9 sind die erweiterte Menge von Eigenschaften, die zusätzlich (optional) in einem normenkonformen Fuzzy-Control-System realisiert werden dürfen (z. B. für den AND Operator könnte der Algorithmus PROD oder BDIF oder beide ausgewählt werden). Diese optionalen Eigenschaften bestehen zusätzlich zu allen Eigenschaften der Basisebene.

Tabelle 9 – (optionale) FCL Sprachelemente der Erweiterungsebene

Sprachelement	Schlüsselwort	Einzelheiten
Funktionsbaustein-Deklaration	VAR	Enthält die lokalen Variablen
Zugehörigkeitsfunktion	Eingangsvariable: TERM	Höchstens vier konstante oder variable Punkte (Zugehörigkeitskoordinate = 0 oder 1)
	Ausgangsvariable: TERM	Höchstens vier konstante oder variable Punkte (Zugehörigkeitskoordinate = 0 oder 1)
Bedingte Aggregation	Operator: AND	Algorithmus: PROD , BDIF
	Operator: OR	Algorithmus: ASUM , BSUM
	Operator: NOT	1 - {Argument}
	Klammern	()
Aktivierung	Operator: ACT	Algorithmus: MIN, PROD
Akkumulation	Operator: ACCU	Algorithmus: BSUM , NSUM
Bereich zur Fuzzifizierung	Operator: RANGE	RANGE (Minimalwert..Maximalwert), begrenzt den Bereich der Zugehörigkeitsfunktionen für die Ausgangsvariable.
Defuzzifizierungsmethode	Operator: METHOD	Algorithmus: CoG , CoA , LM , RM
Regelblock	Operator: RULEBLOCK	n Regelblöcke
Bedingung	IF	n Teilbedingungen, n Eingangsvariablen
Konklusion	THEN	n Subkonklusionen, n Ausgangsvariablen
Wichtungsfaktor	WITH	Konstanter Wert oder Wert, der durch Variablen im Deklarationsteil zugewiesen ist. VAR_INPUT.....END_VAR

Die Tabelle 10 zeigt ein Beispiel einer Liste von Sprachelementen in der offenen Ebene. Eine solche Liste muss ein Teil der Systemdokumentation sein.

Tabelle 10 – Beispiele für eine Liste mit Sprachelementen der offenen Ebene

freie Eingangs-/Ausgangszugehörigkeitsfunktionen (z. B. Gauss, exponentiell)
mehr als vier Punkte der Zugehörigkeitsfunktion
Koordinatenwert des Zugehörigkeitsgrads zwischen 0 und 1
variable Zugehörigkeitsgrade

6.2 Datenprüfliste

Diese Datenprüfliste (siehe Tabelle 11) muss in der technischen Dokumentation mitgeliefert werden. In dieser Liste muss ein Hersteller von speicherprogrammierbaren Steuerungen, Fuzzy-Control-Programmierungswerkzeugen und Anwendungssoftware die eigenen Leistungsmerkmale des Fuzzy-Control-Systems beschreiben. Um die Übertragung von Fuzzy-Control-Anwendungen zwischen den Systemen verschiedener Hersteller zu ermöglichen, ist die folgende Datenprüfliste ein Hilfsmittel, um festzustellen, ob eine Programmübertragung möglich ist. Dies ist nicht notwendigerweise eine vollständige Liste, sondern sie dient als ein Beispiel und sie darf durch die Hersteller erweitert werden.

Tabelle 11 – Datenprüfliste

Technische Daten	Herstelleraussage (Beispiele)
Datentypen von Eingängen und Ausgängen von Funktionsbausteinen	<i>REAL, INT</i>
Zeilenkommentar im FCL-Programm	<i>YES, NO</i>
Ausführungszeit (ms)	20, 30
Speicherbedarf (kB)	3, 4
Abbildung der Variablenwerte von Wichtungsfaktoren und Zugehörigkeitsgraden von 0.0 bis 1.0 auf den Bereich der Integer-Werte	0-200, 0-400
Länge von Bezeichnern (z. B. Name von Variablen, Regelblöcken, Termen)	6, 8
Größte Anzahl der Eingangsvariablen zur Fuzzifizierung	6, 8
Größte Anzahl von Termen der Zugehörigkeitsfunktion pro Eingangsvariable	5, 7
Größte Gesamtanzahl von Termen der Zugehörigkeitsfunktion für alle Eingangsvariablen	30, 56
Größte Anzahl von Punkten für die Zugehörigkeitsfunktion, die jedem Term der Eingangsvariable zugeordnet sind	3, 4, 10
Größte Gesamtanzahl von Punkten für die Zugehörigkeitsfunktionen, die allen Termen der Eingangsvariablen zugeordnet sind	90, 224
Größte Anzahl von Ausgangsvariablen zur Defuzzifizierung	6, 8
Größte Anzahl von Termen der Zugehörigkeitsfunktion pro Ausgangsvariable	5, 7
Größte Gesamtanzahl von Termen von Zugehörigkeitsfunktionen für alle Ausgangsvariablen	30, 56
Größte Anzahl von Punkten für die Zugehörigkeitsfunktion, die jedem der Ausgangsvariablen-Terme zugeordnet sind	1, 4, 10
Größte Gesamtanzahl von Punkten für die Zugehörigkeitsfunktionen, die allen Ausgangsvariablen-Termen zugeordnet sind	90, 224
Größte Anzahl von Regelblöcken	1, 10
Größte Anzahl von Regeln pro Block	10
Größte Anzahl von Teilbedingungen pro Regel	4, 10
Größte Anzahl aller Regeln	15
Größte Anzahl von Teilbedingungen pro Regel	4
Schachteltiefe von ()	1, 3

Anhang A (informativ) Theorie

Dieser Anhang ist als eine Erläuterung der Definitionen zu verstehen, die in Abschnitt 3 angegeben sind.

A.1 Fuzzy-Logik

Bei *Fuzzy-Logik* werden linguistische Variablen und Ausdrücke verwendet, um physikalische Variablen zu beschreiben, statt der Namen und Zahlen (gewöhnlich reelle Zahlen), die in konventionellen Steuerungs- und Regelsystemen verwendet werden. Die Terme „niedrig“ oder „ganz offen“ werden als *linguistische Terme* der physikalischen Größen „Temperatur“ oder „Heizungsventilöffnung“ bezeichnet. Wenn ein Eingang durch einen *linguistischen Term* beschrieben wird, so bezeichnet man ihn als einen *linguistischen Wert*.

Jeder *linguistische Term* wird durch eine *unscharfe (Fuzzy-)Menge* M beschrieben. Er ist damit mathematisch eindeutig durch die beiden Angaben Grundmenge G und *Zugehörigkeitsfunktion* μ definiert. Die *Zugehörigkeitsfunktion* gibt die Zugehörigkeit für jedes Element des Wertebereichs G (z. B. Zahlenwerte einer Zeitskala [Alter in Jahren]) in der Menge M (z. B. jung) in Form eines Zahlenwerts zwischen Null und Eins an. Ist die *Zugehörigkeitsfunktion* für einen bestimmten Wert gleich Eins, so trifft die den *linguistischen Term* betreffende Aussage voll zu (z. B. „jung“ für ein Alter von 20 Jahren). Ist sie dagegen Null, dann ist keinerlei Übereinstimmung vorhanden (z. B. „sehr alt“ für ein Alter von 20 Jahren).

Die folgenden Notationen werden verwendet, um *Fuzzy-Mengen* zu beschreiben:

Für endliche Mengen als ungeordnete Paarmengen in aufzählender Form:

$$M = \{(x_1, \mu_M(x_1)), (x_2, \mu_M(x_2)), \dots, (x_n, \mu_M(x_n))\}, \quad x_i \in G, i = 1, 2, \dots, n \quad (\text{A.1})$$

Die $\mu_M(x_i)$ werden als Zahlenwerte angegeben.

Für unendliche Mengen:

$$M = \{x, \mu_M(x)\}, \quad x \in G \quad (\text{A.2})$$

Um den Unterschied zwischen *scharfen* (crisp) und *unscharfen* (fuzzy) Termen klarer zu erläutern, werden die *linguistischen Terme* der *linguistischen Variablen* „Alter“ in Bild A.1 gezeigt. Während „volljährig“ durch Gesetz eindeutig festgelegt ist und damit einen sprungförmigen Übergang bei der Relation zur *Zugehörigkeitsfunktion* aufweist, darf für „erwachsen“ keine scharfe Altersgrenze angegeben werden.

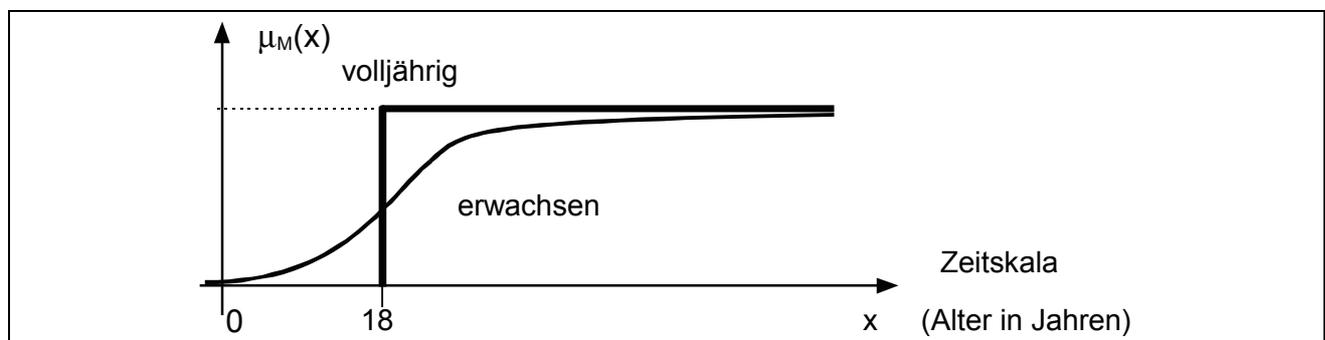


Bild A.1 – Zugehörigkeitsfunktionen der Terme „volljährig“ und „erwachsen“

Als ein Beispiel zeigt Bild A.2 die Beschreibung der *linguistischen Variablen* „Alter“ durch *linguistische Terme* und ihre Anordnung auf der Zeitskala „Alter in Jahren“ mit Hilfe von *Zugehörigkeitsgraden*.

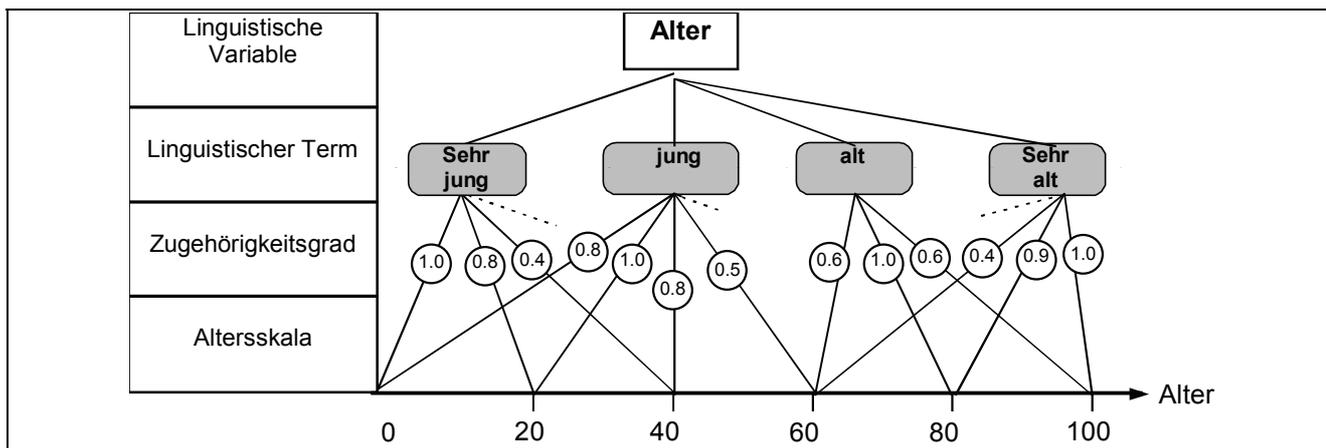


Bild A.2 – Beschreibung der linguistischen Variablen „Alter“ durch linguistische Terme und ihre Anordnung auf der Zeitskala (Alter in Jahren)

Typische Formen von Zugehörigkeitsfunktionen sind in Bild A.3 dargestellt. Zu den besonderen Formen zählen:

- Die Definition über das Rechteck (z. B. Intervall) um Werte zu beschreiben, sowie
- das „Singleton“ zur alternativen Darstellung von Ausgangsvariablen.

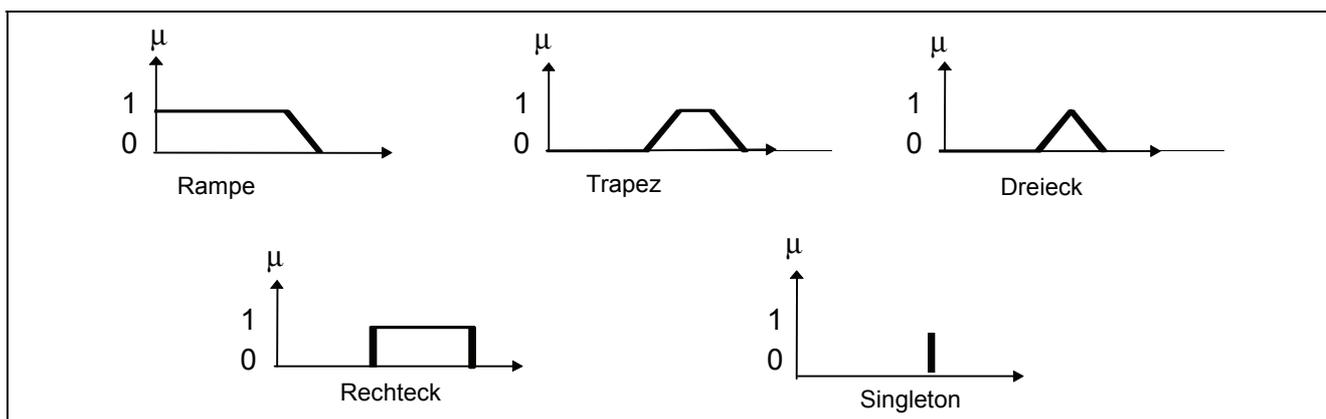


Bild A.3 – Allgemein verwendete Formen von Zugehörigkeitsfunktionen

Ein Ausdruck, in dem *linguistische Variable* mit *linguistischen Termen* verbunden sind, stellt eine *linguistische Aussage* in der *Fuzzy-Logik* dar. Ausdrücke, wie „Temperatur ist hoch“ oder „Temperatur ist niedrig“ mit der einfachen Grundstruktur

$\text{Linguistische Variable} - \text{Vergleichssymbol} - \text{linguistischer Term}$ <p>(Temperatur IS niedrig)</p>	(A.3)
---	-------

werden hier als *linguistische Aussagen* bezeichnet.

Im Gegensatz zur klassischen Logik, in der Aussagen nur einen der Booleschen Zustände „wahr“ oder „falsch“ annehmen, besitzen die *linguistischen Aussagen* in der *Fuzzy-Logik* einen *Zugehörigkeitsgrad*.

Empirisches Wissen kann in *Regeln* definiert werden. Regel R_k hat die folgende Form:

$R_k: \text{ IF Bedingung } P_k \text{ THEN Konklusion } C_k$	(A.4)
---	-------

In diesem Kontext umfasst die *Bedingung* jeder *Regel* eine linguistische Aussage oder eine Kombination von Aussagen durch die Eingangsvariablen, während die *Konklusion* die Ausgangsvariablen im Sinne einer Handlungsanweisung bestimmt.

$P_k = A \text{ AND } B \text{ OR } (\text{NOT } C)$	(A.5)
--	-------

Wenn *Bedingung* und/oder *Konklusion* durch *linguistische Aussagen* bestimmt werden, wird die *Regel* ebenfalls als eine *linguistische Regel* bezeichnet. Ein *Regelwerk* wiederum besteht aus verschiedenen *Regeln*. Im Allgemeinen gelten („sind aktiv“ oder „feuern“) verschiedene *Regeln* zur gleichen Zeit, im Gegensatz zu klassischen regelbasierten Systemen. Deswegen müssen die Ergebnisse der Regeln durch entsprechende mathematische *Operatoren* miteinander verknüpft werden.

Beziehungen zwischen *Fuzzy-Mengen* und *Operationen* mit *Fuzzy-Mengen* sind durch *Zugehörigkeitsfunktionen* definiert.

Es gelten die folgenden Beziehungen zwischen zwei *Fuzzy-Mengen* A und B, deren Elemente x aus der Grundmenge G stammen:

Gleichheit	$A = B,$ ist wahr, wenn $\mu_A(x) = \mu_B(x)$ für alle $x \in G$	(A.6)
Vollständige Inklusion $A \subseteq B,$	ist wahr, wenn $\mu_A(x) \leq \mu_B(x),$ für alle $x \in G$	(A.7)
Teilweise Inklusion $A \subset B,$	ist wahr, wenn $\mu_A(x) \leq \mu_B(x),$ für alle $x \in G$ und $\mu_A(x) < \mu_B(x)$ für mindestens ein $x \in G$	(A.8)

Die folgenden *Operationen* können zwischen zwei *Fuzzy-Mengen* festgelegt werden, deren Elemente x aus der Grundmenge G stammen:

Der <i>Durchschnitt</i> $A \cap B$ ist definiert durch $\mu_{A \cap B}(x) = I(\mu_A(x), \mu_B(x)),$	(A.9)
wobei I der Durchschnittsoperator genannt wird.	
Die <i>Vereinigung</i> $A \cup B$ ist definiert durch $\mu_{A \cup B}(x) = U(\mu_A(x), \mu_B(x)),$	(A.10)
wobei U der Vereinigungsoperator genannt wird.	
Das <i>Komplement</i> ist definiert durch $\bar{\mu}_A(x) = 1 - \mu_A(x)$	(A.11)

Wie bei den klassischen (*scharfen*) Mengen gelten folgende Interpretationen bei den Fuzzy-Mengen:

Der *Durchschnitt* bezieht sich auf den Fuzzy-Operator AND,
die *Vereinigung* bezieht sich auf den Fuzzy-Operator OR,
das *Komplement* bezieht sich auf den Fuzzy-Operator NOT.

Elementare Algorithmen zur mathematischen Interpretation von Durchschnitt, Vereinigung und Komplement sind die folgenden (siehe auch Bild A.4):

Für den <i>Durchschnitt</i> das Minimum $\mu_{A \cap B}(x) = \text{Min} \{ \mu_A(x), \mu_B(x) \},$ $x \in G$	(A.12)
Für die <i>Vereinigung</i> das Maximum $\mu_{A \cup B}(x) = \text{Max} \{ \mu_A(x), \mu_B(x) \},$ $x \in G$	(A.13)
Für das <i>Komplement</i> die Subtraktion von Eins $\bar{\mu}_A(x) = 1 - \mu_A(x)$ $x \in G$	(A.14)

Es gibt eine Vielzahl von möglichen Algorithmen für die Fuzzy-Operatoren AND und OR. Es sollte jedoch hier erwähnt werden, dass AND- und OR-Operatoren nicht willkürlich gewählt werden können.

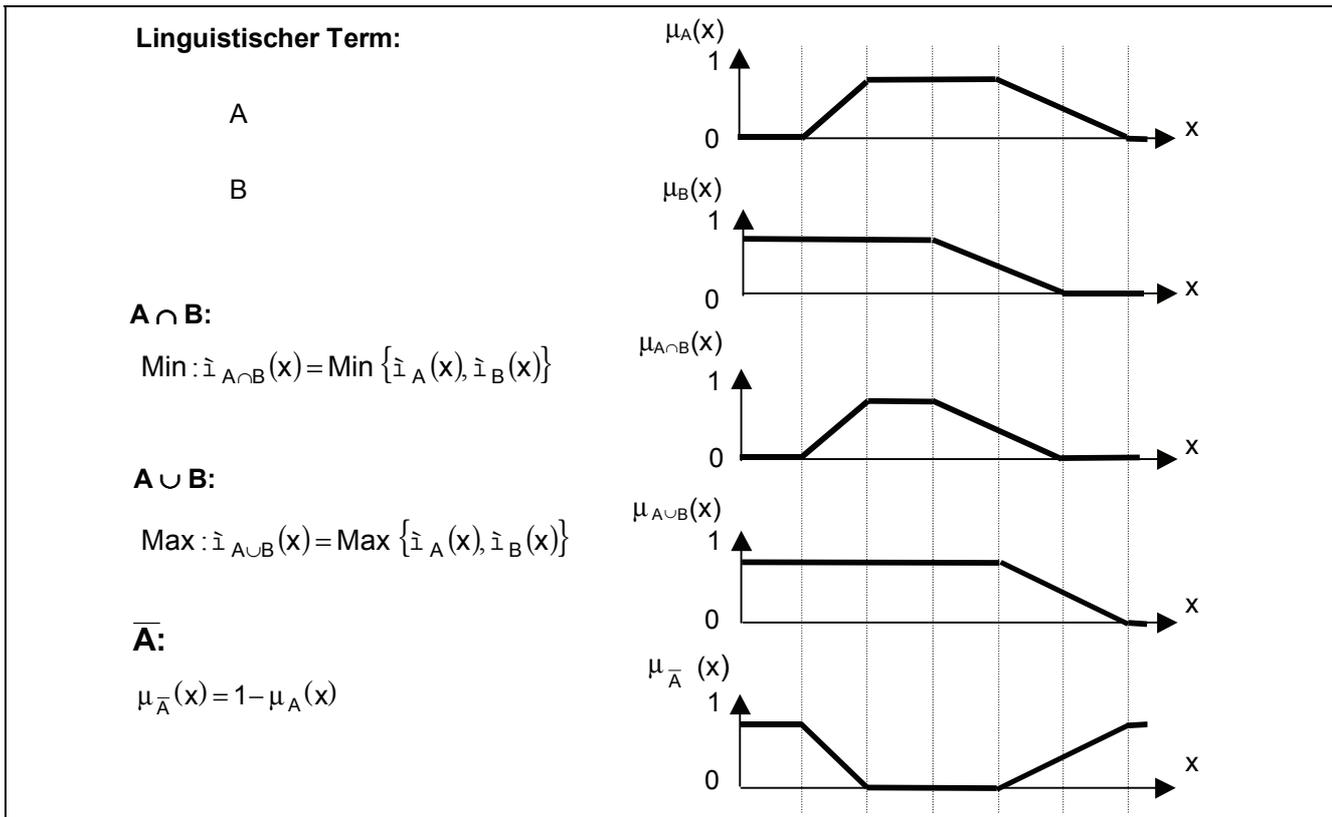


Bild A.4 – Algorithmen zur Implementierung von Operationen zwischen zwei Zugehörigkeitsfunktionen

A.2 Fuzzy-Control

Fuzzy-Control bedeutet das Steuern und Regeln von technischen Prozessen einschließlich der Verarbeitung von gemessenen Werten, das auf der Verwendung von Fuzzy-Regeln basiert, und seine Verarbeitung mit Hilfe der *Fuzzy-Logik*.

Die Eingangsinformation umfasst reelle Variablen in der Form von messbaren Prozessvariablen, abgeleiteten Variablen und auch Sollwerten. Die Ausgangsvariablen sind reelle Variablen in der Form von Stellgrößen. Zwischen den Eingangs- und Ausgangsvariablen des Prozesses und der *Fuzzy-Welt* müssen Transformationen ausgeführt werden (*Fuzzifizierung, Defuzzifizierung*). Die Kernkomponente von *Fuzzy-Control* besteht aus den linguistischen Regeln des *Regelwerks* und der *Inferenz*.

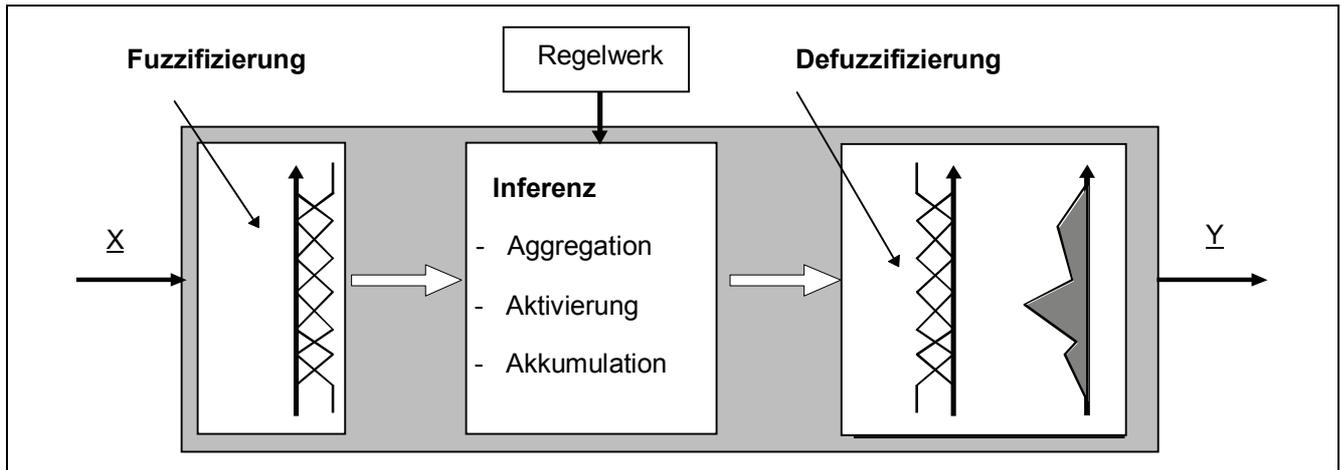


Bild A.5 – Struktur und funktionale Elemente von Fuzzy-Control

Die funktionalen Elemente der oben genannten und in Bild A.5 dargestellten *Fuzzy-Control* werden unten erläutert.

A.2.1 Fuzzifizierung

Die Ermittlung, wie die Eingangsvariablen zu den *linguistischen Termen* passen, wird als *Fuzzifizierung* bezeichnet. Dazu wird der aktuelle Zugehörigkeitsgrad von Eingangsvariablen für jeden *linguistischen Term* der entsprechenden *linguistischen Variablen* bestimmt.

Bild A.6 zeigt ein Beispiel von *Fuzzifizierung*.

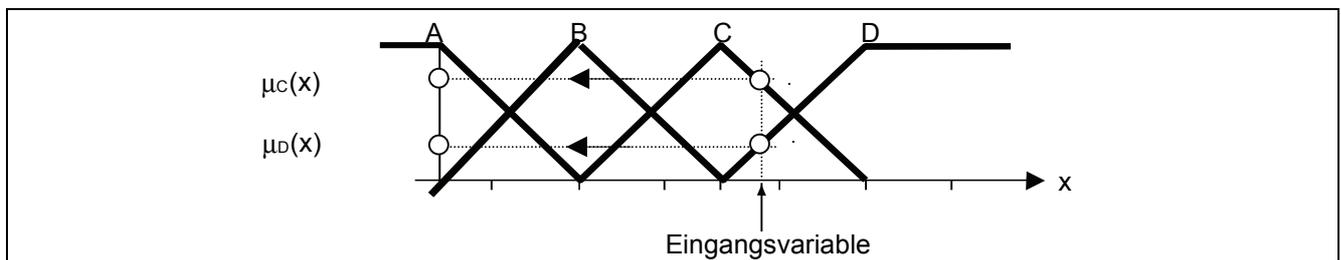


Bild A.6 – Das Prinzip der Fuzzifizierung (am Beispiel)

A.2.2 Regelwerk

Das *Regelwerk* enthält Erfahrungswissen über den Ablauf eines bestimmten Prozesses, der betrachtet wird. Man verwendet *linguistische Regeln*, um das Wissen darzustellen. Unter der Voraussetzung, dass der Fuzzy-Operator AND durch MIN und der Fuzzy-Operator OR durch MAX realisiert ist, kann man leicht zeigen, dass eine Fuzzy-Regel R_j , die auf einer OR-Verknüpfung von m Aussagen beruht, durch m Regeln dargestellt werden kann, deren Aussagen nur durch AND verknüpft sind. Bild A.7 und Bild A.8 sind Beispiele von unterschiedlichen Regel-Darstellungen.

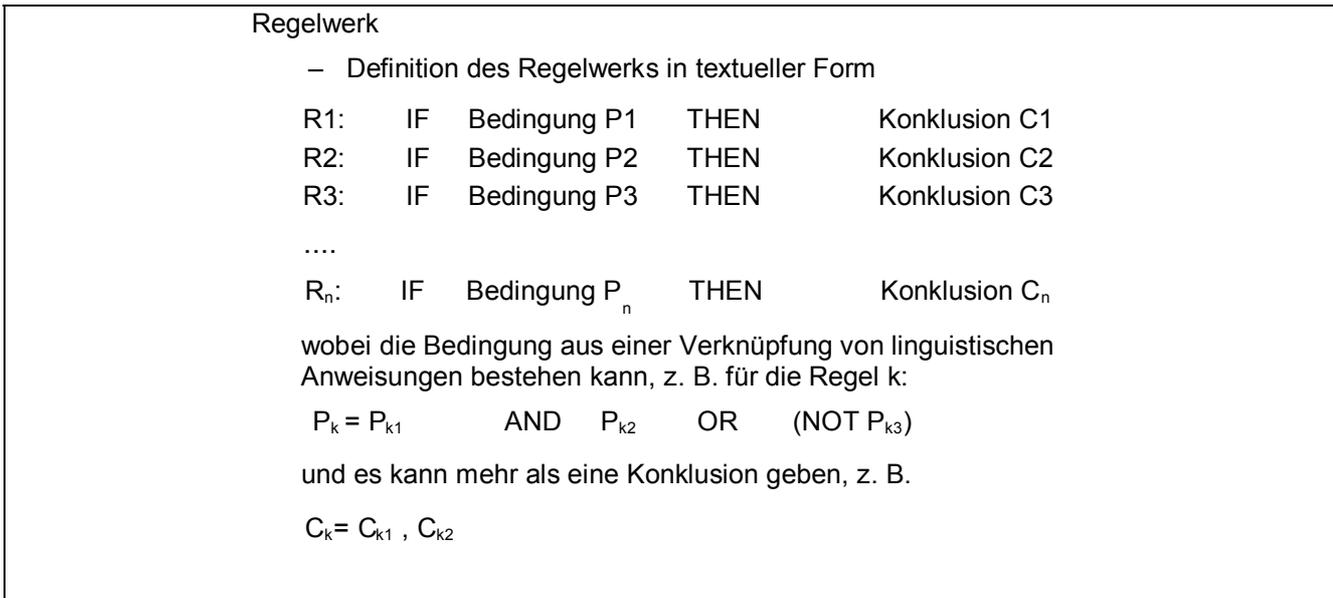


Bild A.7 – Darstellung der Wissensbasis in Form von linguistischen Regeln

Wenn zwei Eingangsvariablen und eine Ausgangsvariable zur Verfügung stehen, und wenn diese zwei Eingangsvariablen nur durch AND verknüpft sind, darf das *Regelwerk* in der Form einer Matrix angegeben werden, in der die Werte der Eingangsvariablen den Spalten und Zeilen zugeordnet sind und die Felder der Matrix die Werte der Ausgangsvariablen enthalten.

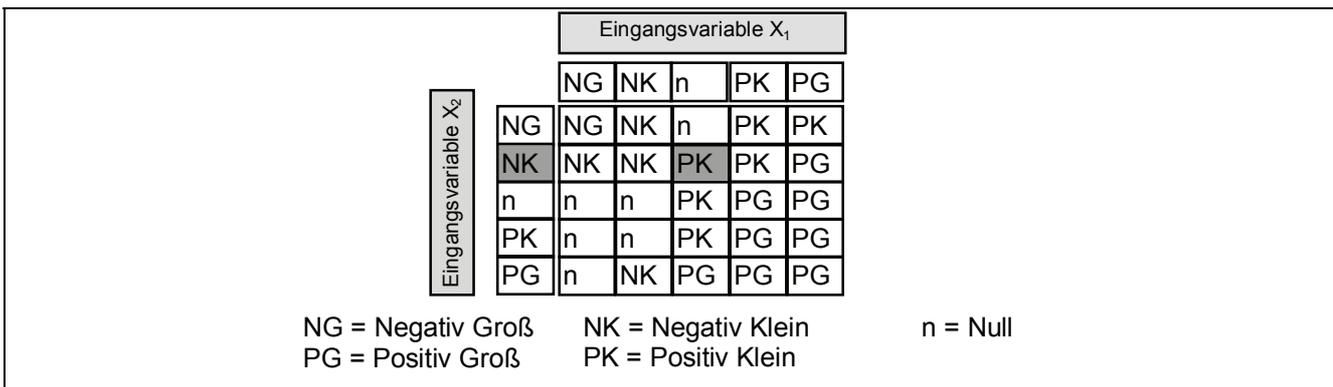


Bild A.8 – Matrix-Darstellung von zwei Variablen

A.2.3 Inferenz

Die hier verwendeten Prinzipien und Definitionen sind für einen vereinfachten Fall von einem unscharfen Regelwerk gültig und vom weit verbreiteten Mamdani-Inferenz-Schema beeinflusst. Kompliziertere Inferenz-Schemata werden hier nicht behandelt.

Die *Inferenz* besteht aus den drei Unterfunktionen *Aggregation*, *Aktivierung* und *Akkumulation*, die in Bild A.9 gezeigt sind. Siehe auch Bild A.10.

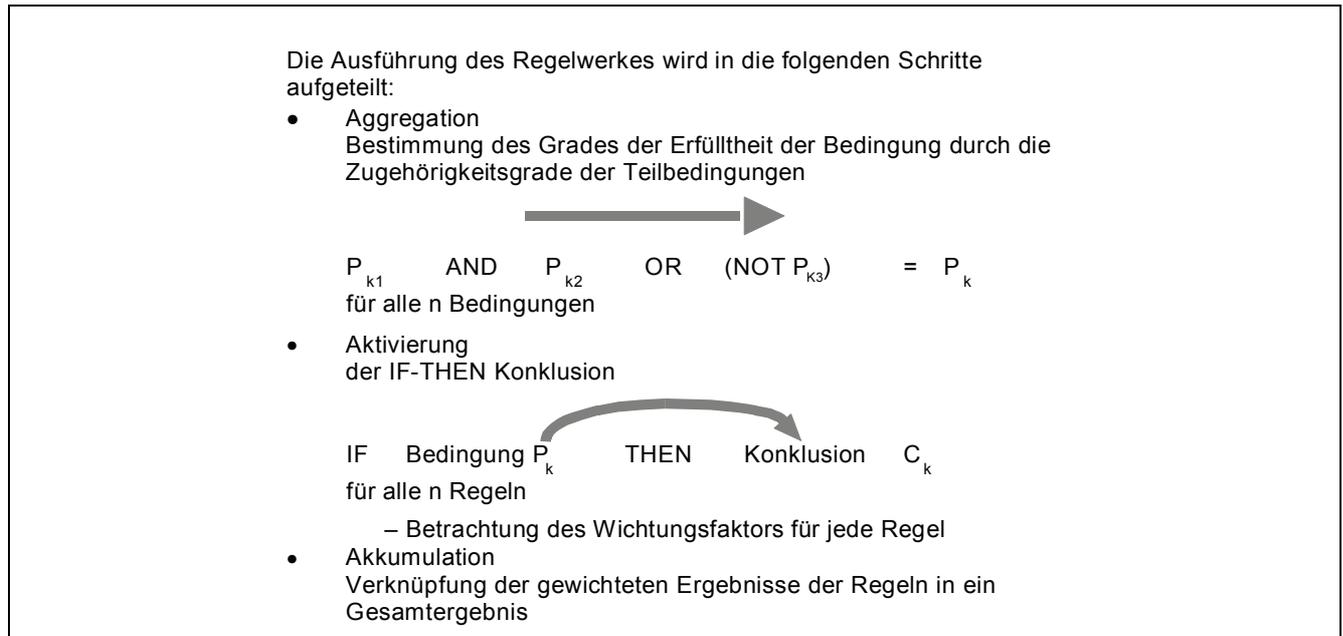


Bild A.9 – Elemente der Inferenz

– *Aggregation:*

Falls die Bedingung aus einer einzelnen Teilbedingung besteht, ist die Gültigkeit der Bedingung identisch mit der der Teilbedingung, d. h. der Grad der Erfüllung der Bedingung entspricht der Teilbedingung. Wenn die Bedingung aber aus einer Verknüpfung verschiedener Teilbedingungen besteht, muss der Grad der Erfüllung durch Aggregation der individuellen Werte bestimmt werden. Wenn eine AND-Verknüpfung der Teilbedingungen vorliegt, wird der Grad der Erfüllung mit Hilfe des Fuzzy-Operators AND berechnet.

– *Aktivierung:*

In der *Konklusion* beziehen sich die *Teilkonklusionen* auf die Ausgangsvariablen. Der Grad der Erfüllung der *Konklusion* wird dann auf der Basis des Grades der Erfüllung der Bedingung bestimmt, die in der Aggregation festgelegt ist (*Konklusion* IF A THEN B). Im Allgemeinen wird MIN oder PROD für die Aktivierung verwendet.

Falls das Regelwerk Regeln mit Wichtungsfaktoren w_k mit $w_k \in [0, 1]$ enthält, darf dies mittels Multiplikation implementiert werden.

$$c_k^* = w_k \times c_k$$

(A.15)

– *Akkumulation:*

Die Ergebnisse der *Regeln* werden verknüpft, um ein Gesamtergebnis zu erhalten. Der *Maximum-Algorithmus* wird gewöhnlich zur *Akkumulation* verwendet. Tabelle A.1 zeigt die *Operatoren*, die im Allgemeinen für die einzelnen *Inferenz-Schritte* verwendet werden.

Abhängig von der Kombination der *Operatoren* in den einzelnen Schritten, erhält man verschiedene *Inferenz-Strategien*. Am besten bekannt sind die sogenannte *MaxMin-Inferenz* und *MaxProd-Inferenz*, die das *Maximum* für die *Akkumulation* und das *Minimum* oder *algebraische Produkt* für die *Aktivierung* benutzen. Im Falle der *MaxMin-Inferenz* sind die *Zugehörigkeitsfunktionen* der *Fuzzy-Mengen* der *Konklusionen* auf den *Grad der Erfüllung* der *Bedingung* begrenzt. Diese *Fuzzy-Mengen* werden dann mit Hilfe der *Maximumbildung* zu einer *Fuzzy-Menge* verknüpft. In der *MaxProd-Inferenz* hingegen werden die *Zugehörigkeitsfunktionen* der *Fuzzy-Mengen* der *Konklusionen* mit dem *Zugehöriggrad* der *Bedingung* gewichtet, d. h. multipliziert und dann verknüpft.

Tabelle A.1 – Inferenz-Schritte und allgemein verwendete Algorithmen

Inferenz-Schritt	Operatoren	Algorithmen
<i>Aggregation</i>		
für AND	<i>Minimum</i>	$a_k = \text{Min}\{a_{k1}(x 1), a_{k2}(x 2)\}$
für OR	<i>Maximum</i>	$a_k = \text{Max}\{a_{k1}(x 1), a_{k2}(x 2)\}$
<i>Aktivierung</i>		
Umwandlung der IF-THEN-Konklusion		
	<i>Minimum</i>	$c_k' = \text{Min}\{a_k, \mu_\infty(u)\}$
Wichtungsfaktor der jeweiligen Regel		
	<i>Multiplikation</i>	$c_k = \text{Mult}\{\omega_k, c_k'\} = \omega_k \times c_k'$
<i>Akkumulation</i>	<i>Maximum</i>	$\mu = \text{Max}\{c_j(u)\}$

$$P_k = \underbrace{P_{k1} \text{ UND } P_{k2}}_{\text{Min}\{a_{k1}, a_{k2}\}} \text{ ODER } \underbrace{(\text{NICHT } P_{k3})}_{(1-a_{k3})}$$

$$a_k = \text{Max}\{\text{Min}\{a_{k1}, a_{k2}\}, (1-a_{k3})\}$$

Bild A.10a – Ein Beispiel, das die Prinzipien der Aggregation zeigt

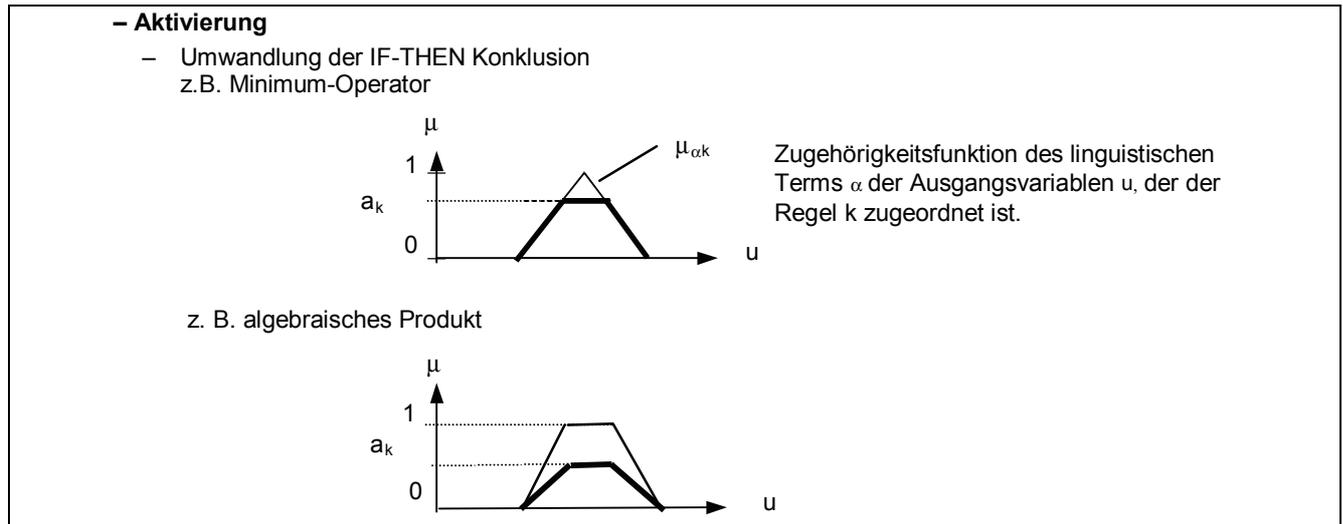


Bild A.10b – Prinzipien der Aktivierung (als ein Beispiel)

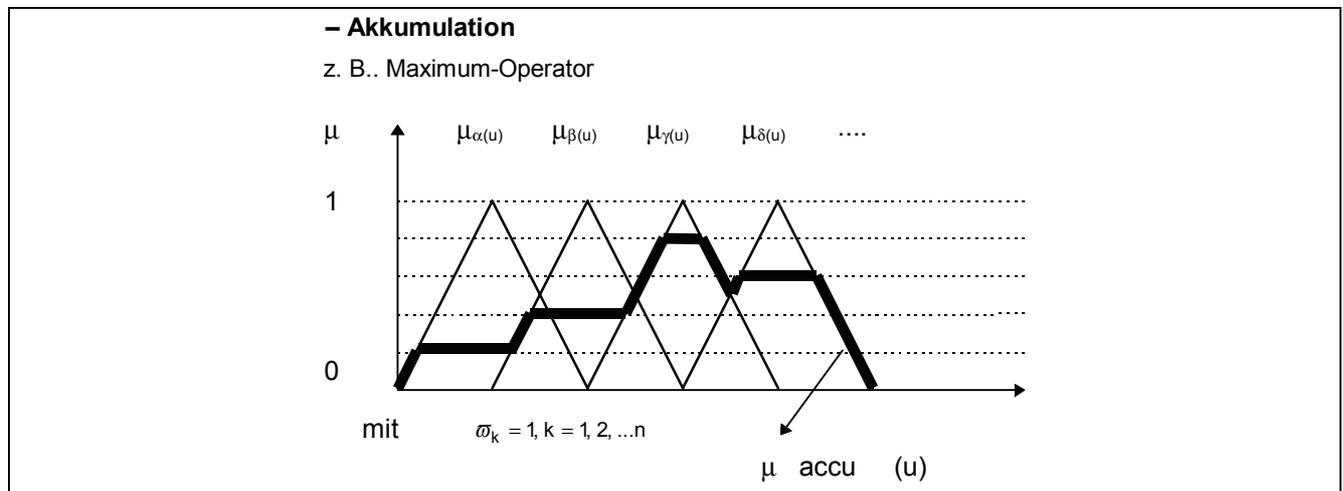


Bild A.10c – Prinzipien der Akkumulation (als ein Beispiel)

A.2.4 Defuzzifizierung (siehe Bild A.11)

Die *Inferenz* liefert als Ergebnis eine *Fuzzy-Menge* oder ihre *Zugehörigkeitsfunktion*. Ein Regelungssystem kann diese unscharfe Information nicht direkt verarbeiten, deshalb muss das Ergebnis des *Inferenz*-Prozesses in *scharfe* numerische Werte umgewandelt werden. Die *scharfe* Zahl, die ermittelt wird (im Allgemeinen eine reelle Zahl), sollte hier eine gute Darstellung der Information liefern, die in der *Fuzzy-Menge* enthalten ist.

– Defuzzifizierung

Umwandlung des unscharfen Ergebnisses der Inferenz in eine scharfe Ausgangsvariable U

Schwerpunkt-Methode (CoG, en: center of gravity)

Die scharfe Ausgangsvariable wird als ein Abszissenwert U des Schwerpunkts unter der Zugehörigkeitsfunktion bestimmt:

– Beispiel der Schwerpunkt-Methode

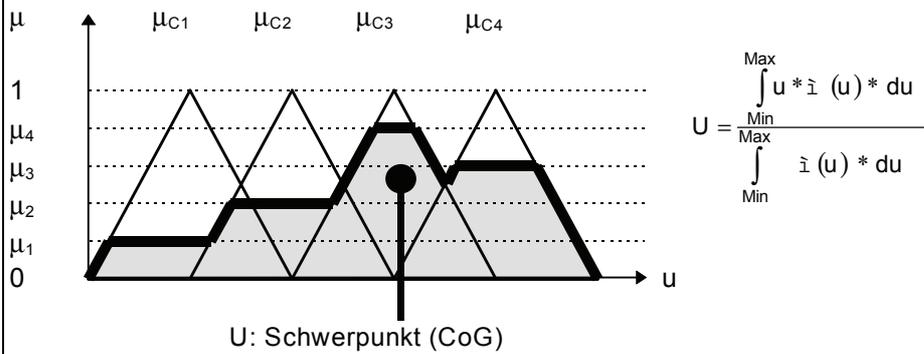


Bild A.11a – Methoden der Defuzzifizierung

Folgende weitere Methoden sind üblich:

Linkes Maximum LM

Es wird der Wert der Ausgangsvariablen bestimmt, für den die *Zugehörigkeitsfunktion* des Ausgangs ihr linkes Maximum erreicht.

Rechtes Maximum RM

Es wird der Wert der Ausgangsvariablen bestimmt, für den die *Zugehörigkeitsfunktion* des Ausgangs ihr rechtes Maximum erreicht.

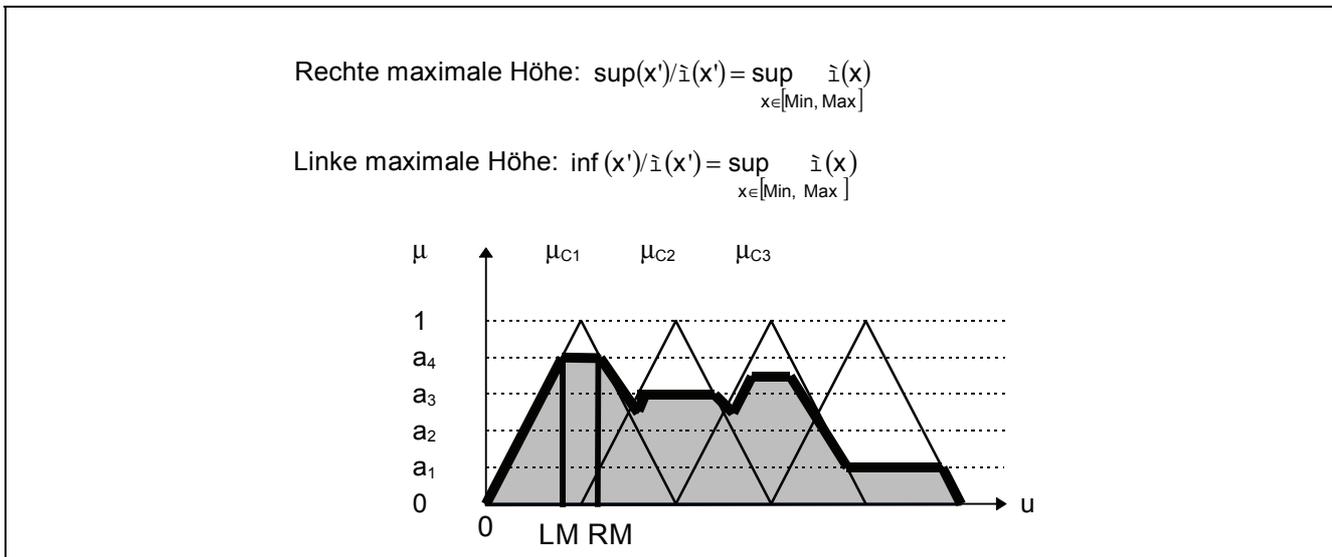


Bild A.11b – Unterschied zwischen linkem Maximum und rechtem Maximum

Flächenschwerpunkt-Methode (en: centre of area)

Hier wird der Ausgangswert als Abszissenwert des Punktes bestimmt, der die Fläche unter der *Zugehörigkeitsfunktion* in zwei gleich große Flächen teilt.

ANMERKUNG 1 Im Englischen ist Centre of Gravity gleichbedeutend mit Centroid of Area.

ANMERKUNG 2 Im Englischen ist Centre of Area gleichbedeutend mit Bisector of Area.

ANMERKUNG 3 CoA ist nicht anwendbar, wenn Singletons verwendet werden.

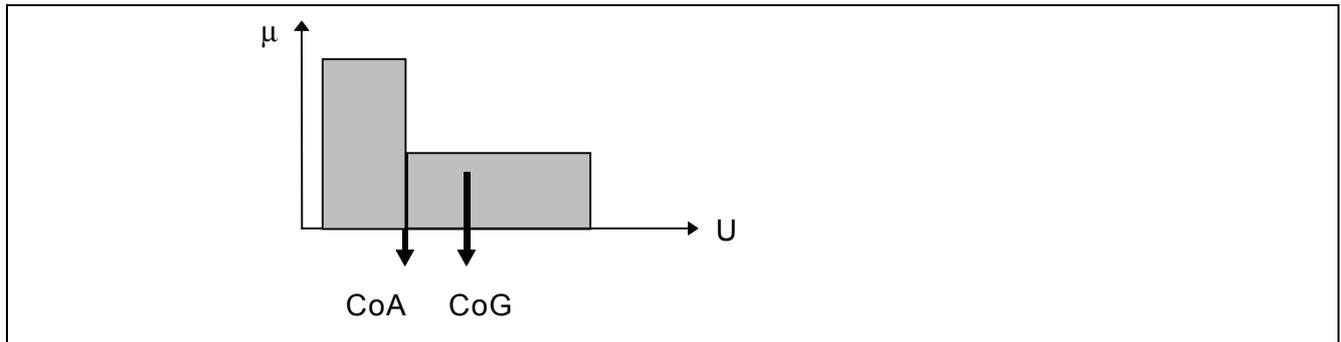


Bild A.11c – Unterschied zwischen Flächenschwerpunkt und Schwerpunkt

ANMERKUNG Formeln der Defuzzifizierungsmethoden CoA und CoG sind in Tabelle 2 angegeben.

Defuzzifizierung

Wenn die Zugehörigkeitsfunktionen der Ausgangswerte Singletons sind, ist die Berechnung durchzuführen mit der

- Schwerpunktmethode für Singletons (COGS)

$$U = \frac{\sum_{i=1}^p [U_i \hat{\mu}_i]}{\sum_{i=1}^p [\hat{\mu}_i]}$$

wobei μ die Ergebnisse der Akkumulation sind.

Im Falle von Singletons ist der Flächenschwerpunkt nicht anwendbar.
Beispiel für die 4 Singletons oben:

$$U = \frac{[U_1 \mu_1 + U_2 \mu_2 + U_3 \mu_4 + U_4 \mu_3]}{[\mu_1 + \mu_2 + \mu_3 + \mu_4]}$$

Bild A.11d – Methoden der Defuzzifizierung

A.3 Verhalten von Fuzzy-Control

Aus der Sicht der Informationstechnologie ist Fuzzy-Control ein regelbasiertes Expertensystem. Aus Sicht der Regelungstechnik handelt es sich um einen Kennfeldregler mit nichtlinearer Charakteristik. Bild A.12 zeigt Beispiele von Kennfeldern von Fuzzy-Control. Die aktuellen Werte der Ausgangsvariablen hängen direkt von den aktuellen Werten der Eingangsvariablen und nicht von vorherigen Werten ab, außer in dem Fall, in dem

keine Regel aktiv ist und kein voreingestellter Wert definiert wurde. Falls der Regler mit dynamischem Verhalten zu realisieren ist, müssen die dynamischen Funktionen außerhalb des Fuzzy-Funktionsbausteins zur Verfügung gestellt werden.

Diese sind üblicherweise differenzierende und integrierende Elemente erster Ordnung. Die Ausgangsvariablen dieser Funktionen sind zusätzliche Eingangsvariable für Fuzzy-Control. Dies gilt auch für die Regelabweichung, die in ähnlicher Weise außerhalb von Fuzzy-Control gebildet werden muss. Umgekehrt können die Ausgangsvariablen von Fuzzy-Control an Rechenfunktionen zur Erzeugung der Stellgrößen übergeben werden, z. B. an eine Integrationseinheit für Geschwindigkeitsalgorithmen, oder sie können an verschiedene Elemente von Regeln verteilt werden.

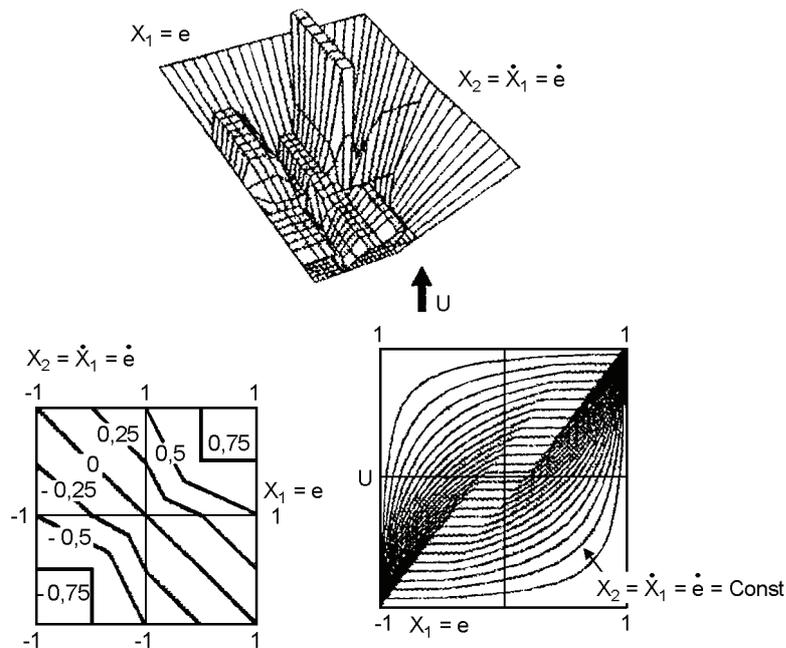


Bild A.12 – Beispiele von Kennfeldern von Fuzzy-Control

Die Grundstruktur einer Fuzzy-Regelung ist in Bild A.13a gezeigt und Bild A.13b gibt dazu ein Beispiel: Die Regeldifferenz (e) wird aus der Differenz zwischen dem Sollwert und dem Istwert gebildet. Diese Differenz, ebenso ihre Ableitung nach der Zeit, werden an Fuzzy-Control als zwei Eingangsvariablen übergeben, die aus Sicht von Fuzzy-Control voneinander unabhängig sind. Die Stellgröße wird aus der Ausgangsvariablen mittels Integration über der Zeit ermittelt. Wenn das Regelwerk so entworfen ist, dass ein Geschwindigkeitsalgorithmus beschrieben wird, zeigt der Fuzzy-basierte Regler ein ähnliches dynamisches Verhalten wie ein PI-Regler.

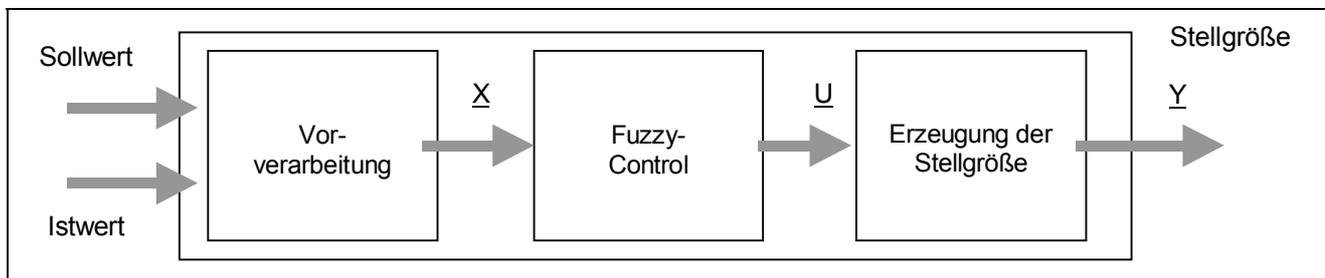


Bild A.13a – Grundstruktur der Fuzzy-basierten Regelung

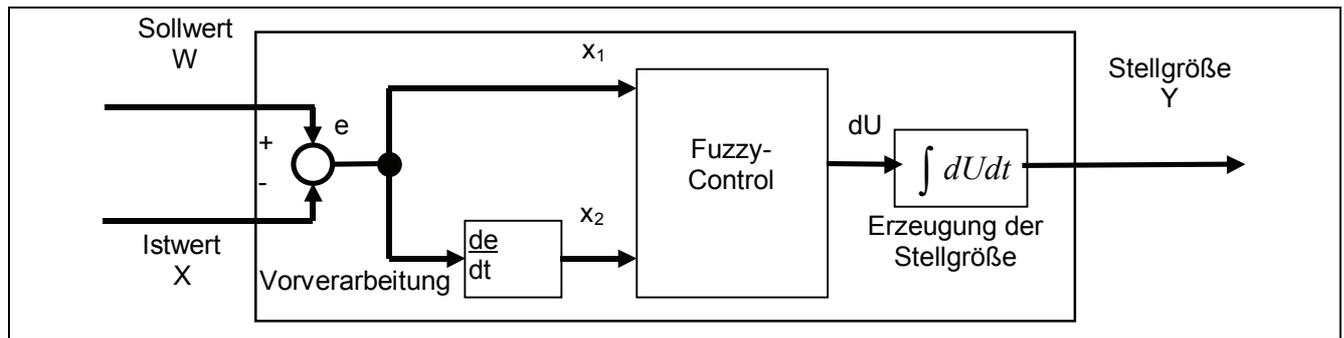


Bild A.13b – Beispiel der Fuzzy-basierten Regelung

Anhang B (informativ) Beispiele

Eines der Hauptanwendungsgebiete im Bereich der Speicherprogrammierbaren Steuerungen ist die Kombination mit herkömmlichen PID-Reglern, um deren Regelqualität zu verbessern. Die folgenden Beispiele zeigen die prinzipiellen Möglichkeiten in allgemeiner Weise, um eine Vorstellung zu vermitteln, wo Fuzzy-Control angewendet werden kann.

B.1 Vorsteuerung

Der Fuzzy-Regler ergänzt den herkömmlichen Regler durch ein zusätzliches Korrektursignal für die Stellgröße (siehe Bild B.1).

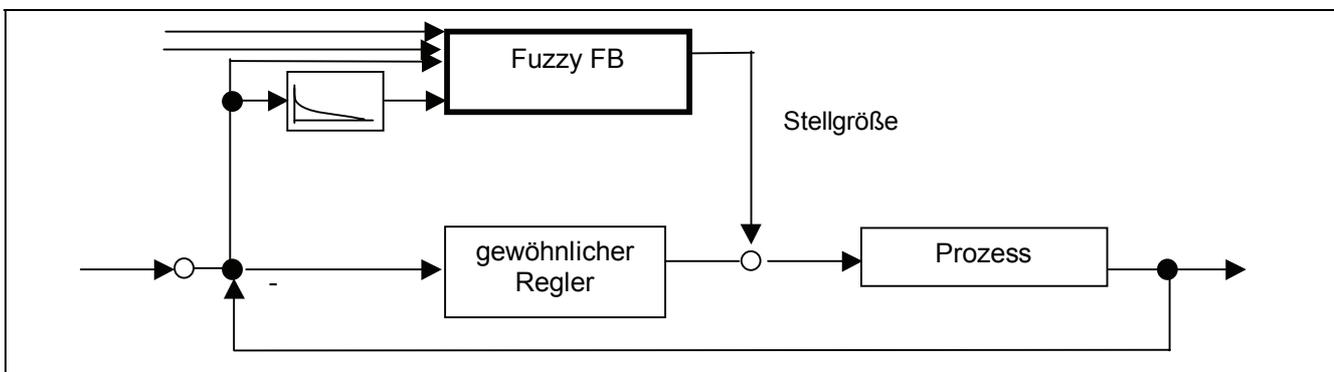


Bild B.1 – Beispiel für eine Vorsteuerung

B.2 Parameter-Adaption von herkömmlichen PID-Reglern

Der Fuzzy-Regler wird verwendet, um die Regelparameter eines PID-Reglers anzupassen (siehe Bild B.2).

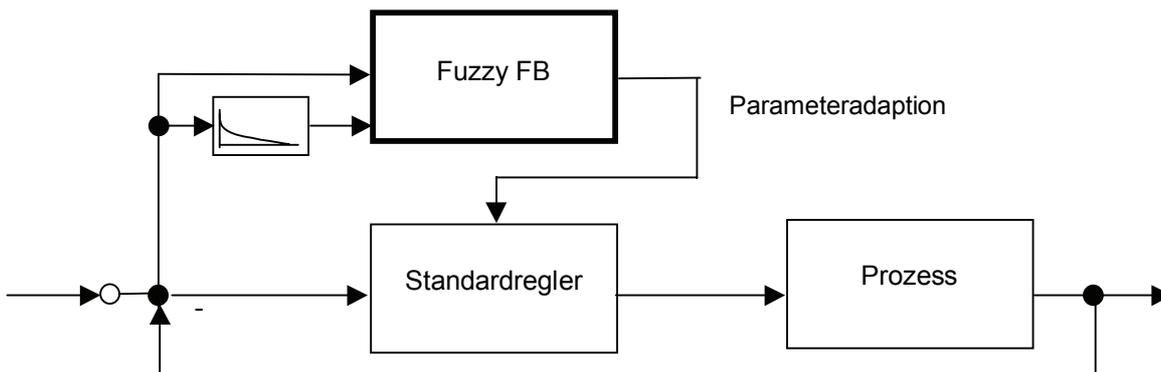


Bild B.2 – Beispiel für eine Parameter-Anpassung

B.3 Direkte Fuzzy-Regelung eines Prozesses

Ein weiteres Anwendungsgebiet besteht darin, empirisches Prozess-Wissen und linguistische Regelungsstrategien direkt in die industrielle Automatisierung einzubringen. Dies ist bei vielen Prozessen anwendbar, bei denen Eingriffe des Bedieners notwendig sind (siehe Bild B.3).

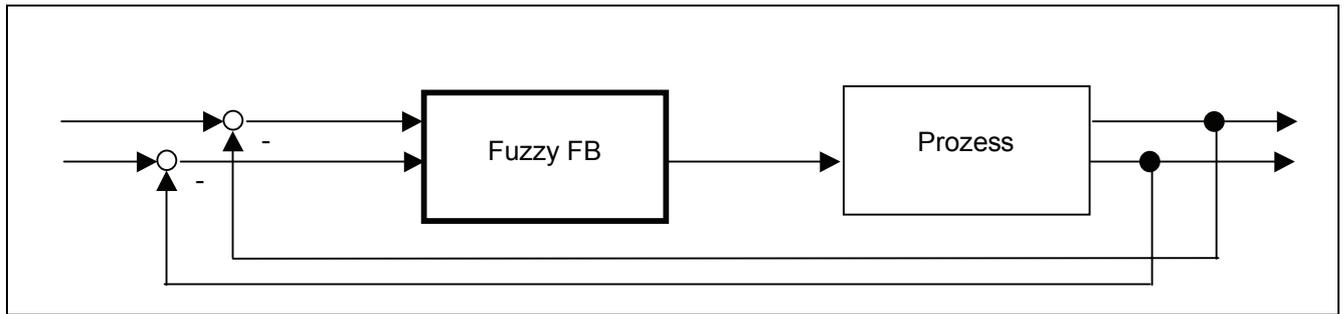


Bild B.3 – Beispiel für eine direkte Fuzzy-Regelung

Anhang C (informativ)

Industriebeispiel – Containerkran

Containerkräne werden in den meisten Häfen zum Be- und Entladen von Containern bei Schiffen verwendet. Sie heben einzelne Container mit Seilen, die am Krankopf befestigt sind. Der Krankopf bewegt sich auf einer horizontalen Schiene. Wenn ein Container angehoben wird und der Krankopf sich zu bewegen beginnt, fängt der Container an zu pendeln, wie es in Bild C.1 gezeigt ist. Beim Transportieren ist das Pendeln kein Problem, ein pendelnder Container kann aber nicht abgeladen werden.

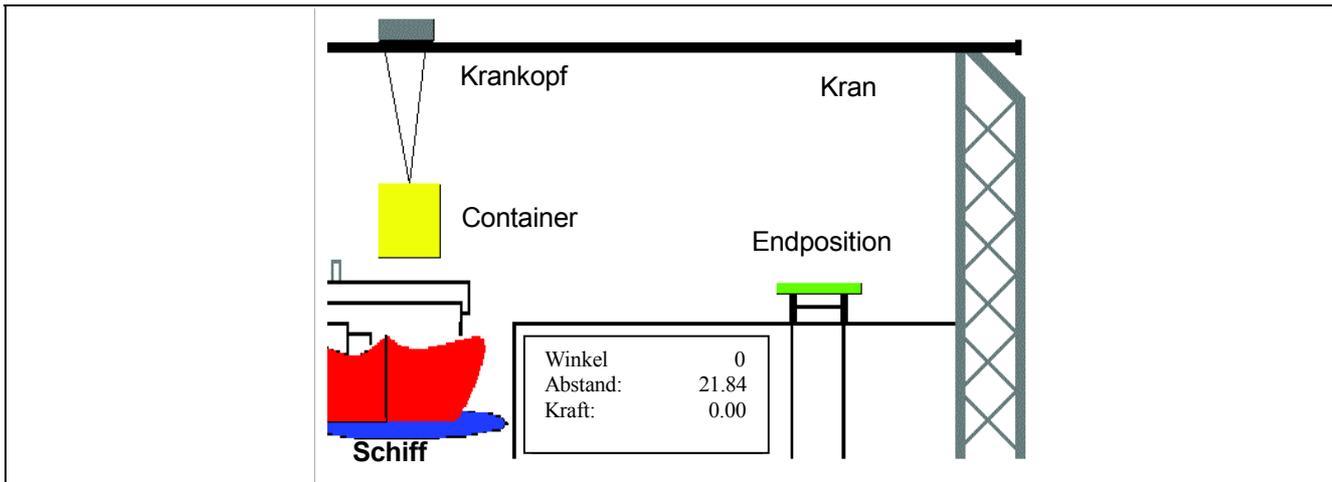


Bild C.1 – Industriebeispiel – Containerkran

Die Analyse der Aktionen eines Bedieners zeigt, dass der Bediener einige „Faustregeln“ anwendet.

- Start mit mittlerer Leistung.
- Wenn man losgefahren und noch weit entfernt vom Ziel ist, wird die Motorleistung so eingestellt, dass der Container ein wenig hinter dem Krankopf hängt.
- Wenn man dem Ziel näher kommt, verringert man die Leistung so, dass der Container ein wenig vor dem Krankopf hängt.
- Sobald der Container nahe am Ziel ist, verwendet man, abhängig von der Richtung des Pendelausschlags, die +/- mittlere Leistung.
- Sobald der Container über dem Ziel ist und der Pendelausschlag null ist, stoppt man den Motor.

Um die Kransteuerung zu automatisieren, werden Sensoren für die Position („Abstand“) des Krankopfes und den Winkel der Pendelbewegung („Winkel“) des Containers eingebaut. Die Motorleistung ist die Ausgangsvariable. Zuerst müssen die linguistischen Variablen für alle Variablen definiert werden. Die linguistischen Variablen Abstand, Winkel und Motorleistung (siehe Bilder C.2, C.3 und C.4) werden in 5 linguistische Terme eingeteilt. Als Zugehörigkeitsfunktionen werden die Formen Rampen, Dreiecke und Singletons verwendet. Die folgenden Bilder zeigen die linguistischen Variablen und die linguistischen Terme.

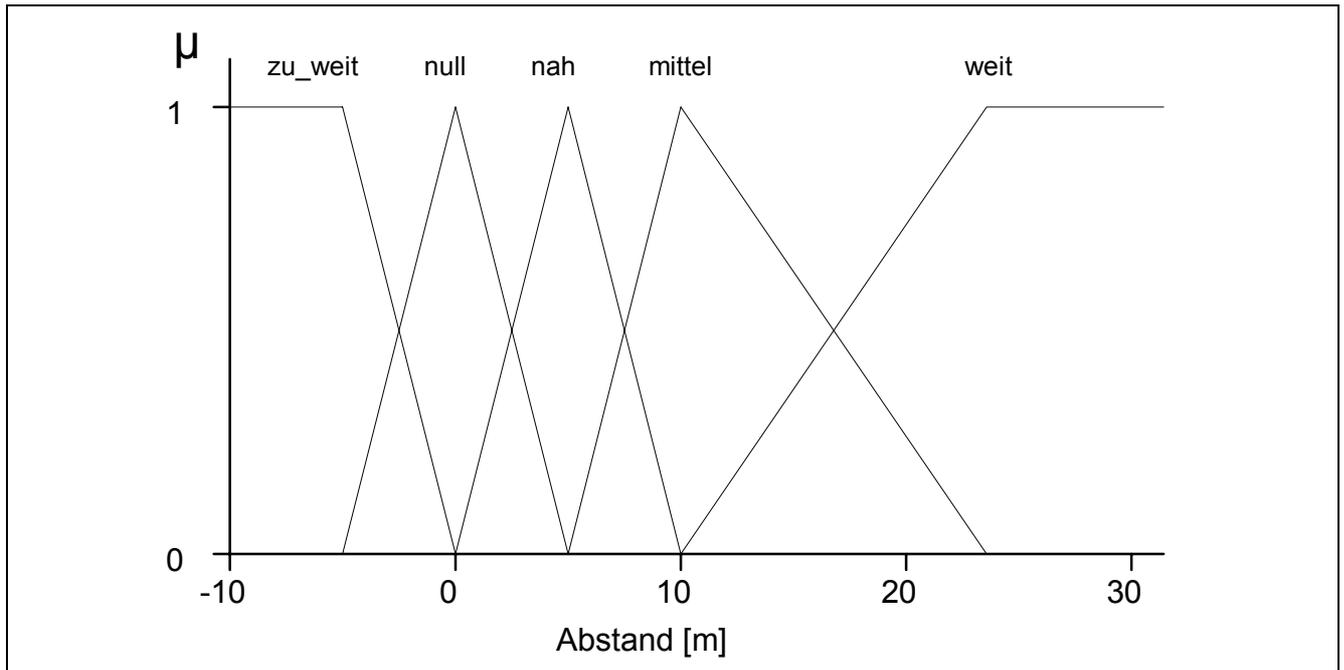


Bild C.2 – Linguistische Variable „Abstand“ zwischen Krankopf und Zielposition

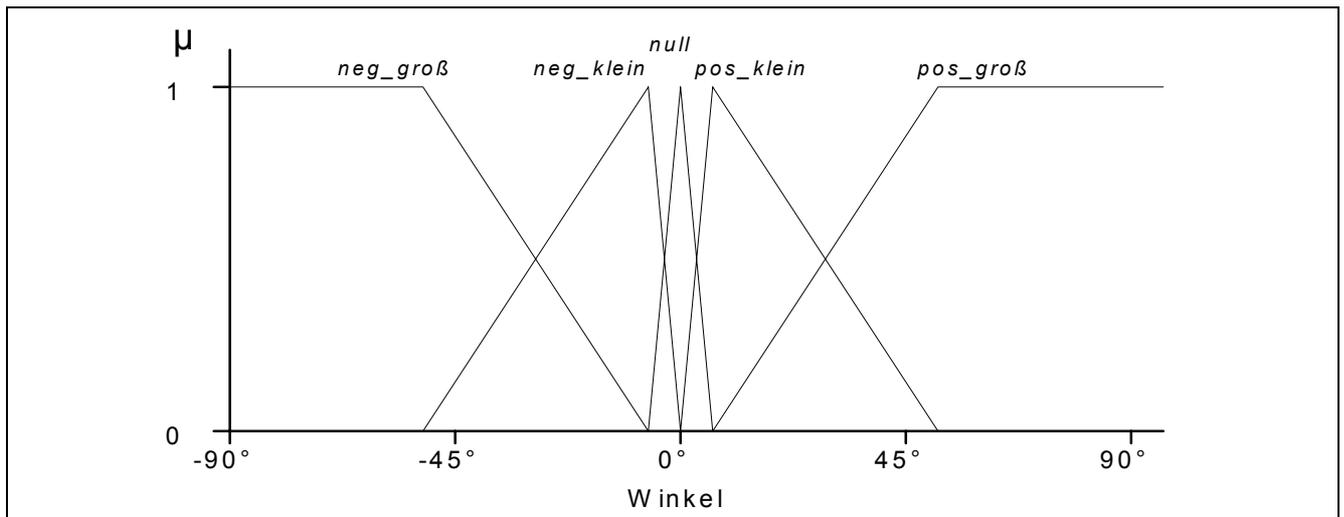


Bild C.3 – Linguistische Variable „Winkel“ des Containers zum Krankopf

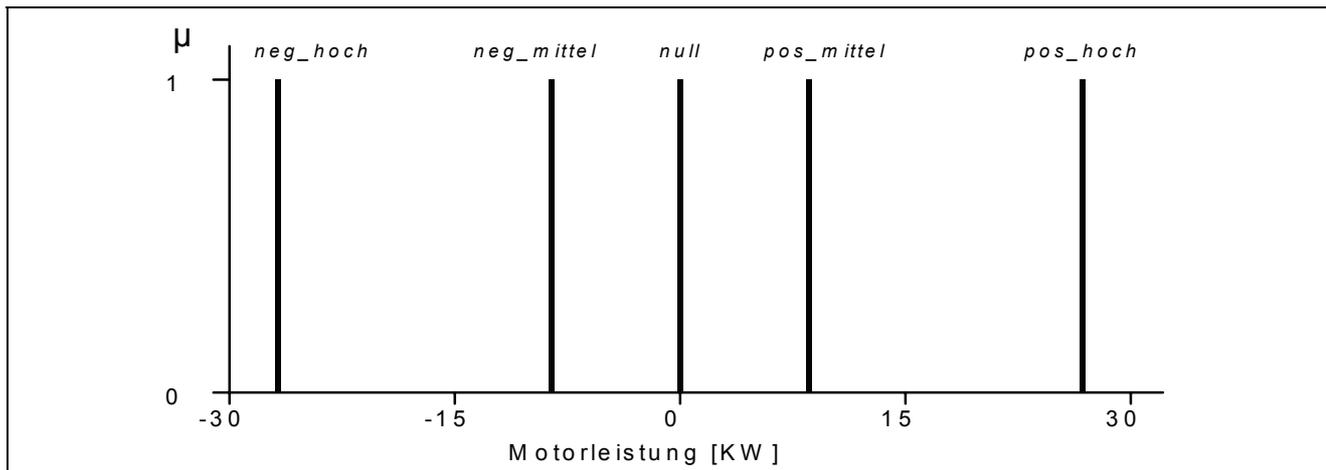


Bild C.4 – Linguistische Variable „Leistung“

Durch Anwendung dieser linguistischen Terme zur Beschreibung des aktuellen Betriebszustands des Krans können die fünf Faustregeln in ein Regelwerk übersetzt werden. Bild C.5 zeigt die Festlegung des Regelwerks in FCL-Schreibweise. Es ist zu beachten, dass Regel 2 in zwei Regeln übersetzt wurde, um in das if-then-Format zu passen. Es ist weiterhin zu beachten, dass einige der definierten Zugehörigkeitsfunktionen hier nicht verwendet werden und das System auf undefinierte Ausgaben geprüft werden sollte, wie Abstand: zu_weit, Winkel: pos_gross.

RULE 1: IF Abstand IS weit	AND Winkel IS null	THEN Leistung IS pos_mittel
RULE 2: IF Abstand IS weit	AND Winkel IS neg_klein	THEN Leistung IS pos_hoch
RULE 3: IF Abstand IS weit	AND Winkel IS neg_gross	THEN Leistung IS pos_mittel
RULE 4: IF Abstand IS mittel	AND Winkel IS neg_klein	THEN Leistung IS neg_mittel
RULE 5: IF Abstand IS nah	AND Winkel IS pos_klein	THEN Leistung IS pos_mittel
RULE 6: IF Abstand IS null	AND Winkel IS null	THEN Leistung IS null

Bild C.5 – Regelwerk

Tabelle C.1 zeigt die Inferenzschritte bzw. die angewendeten Operatoren.

Tabelle C.1 – Inferenzschritte und zugeordnete Operatoren

Inferenzschritt	Operatoren
Aggregation	
AND	Minimum
Aktivierung	
Umwandlung der IF-THEN-Konklusion	
	Minimum
Akkumulation	Maximum

Betrachtet sei eine aktuelle Situation des Krans, in der der Abstand des Krankopfs zur Zielposition 12 m und der Winkel des Containers +4° beträgt. Zur Erläuterung wird eine Untermenge von drei Regeln angenommen.

Die Bilder C.6 und C.7 zeigen, wie die *Fuzzifizierung* in diesem Fall berechnet wird.

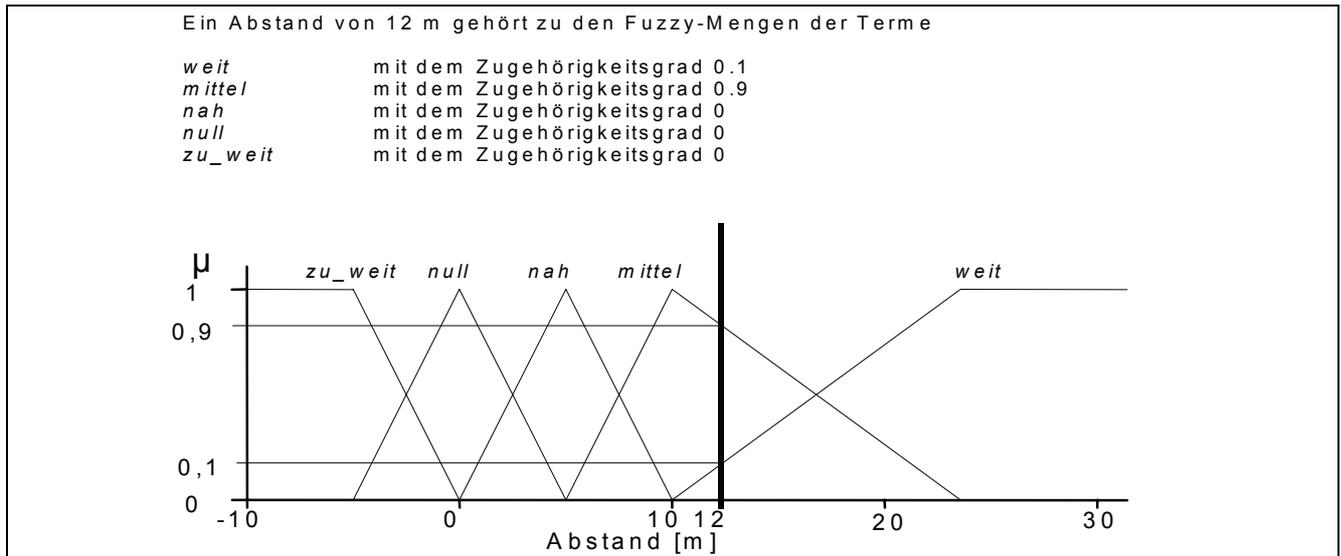


Bild C.6 – Fuzzifizierung der linguistischen Variable „Abstand“

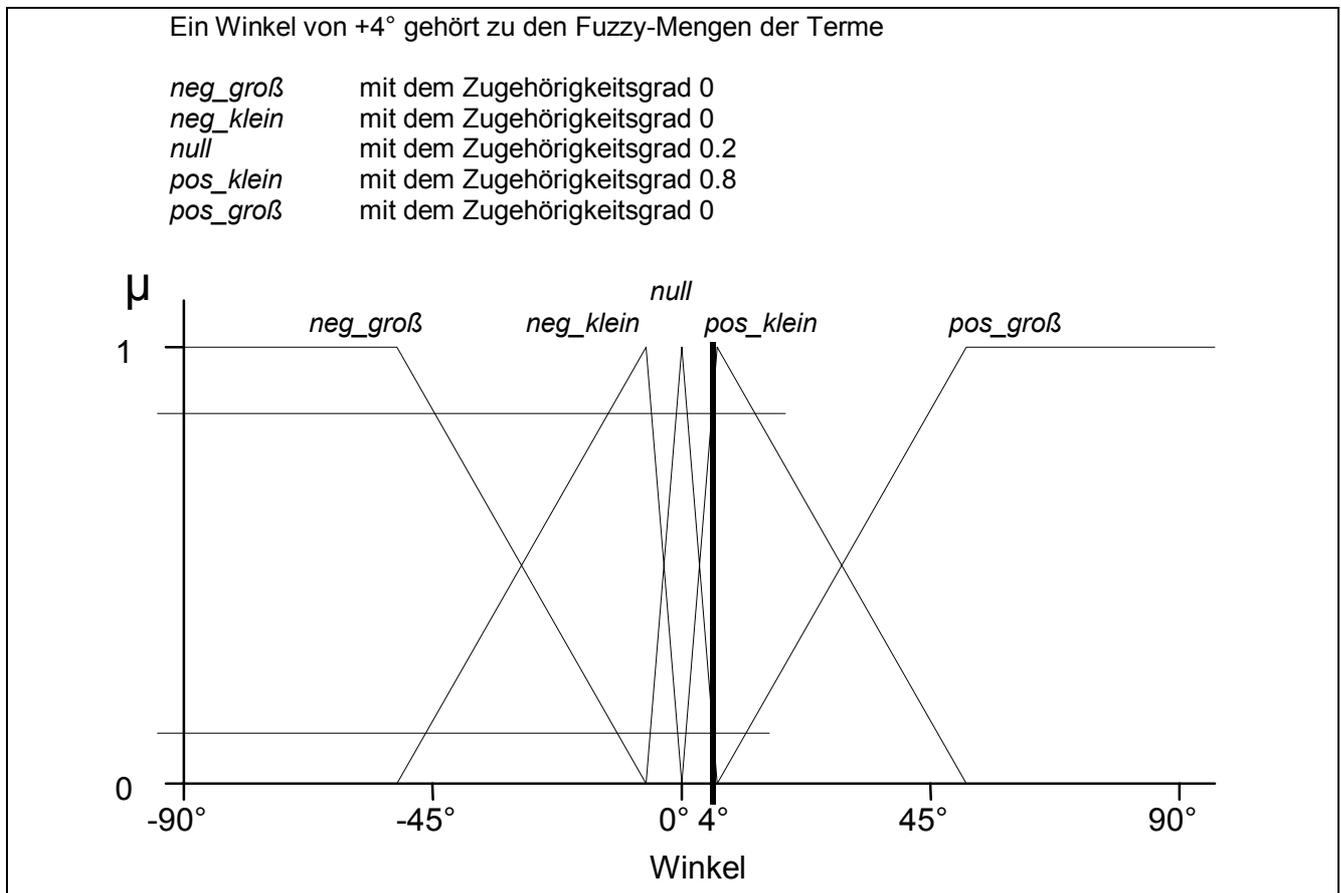


Bild C.7 – Fuzzifizierung der linguistischen Variable „Winkel“

Der Abstand von 12 m wird in den Wert {0.1, 0.9, 0, 0, 0} der linguistischen Variablen überführt und kann als „noch mittel, aber schon etwas weit“ betrachtet werden. Der Winkel von +4° wird in den Wert {0, 0, 0.2, 0.8, 0} der linguistischen Variablen überführt und kann als „positiv klein und fast null“ betrachtet werden.

– Inferenz

Nachdem nun alle Eingangsvariablen in Werte von linguistischen Variablen umgewandelt worden sind, kommt der Schritt der Fuzzy-Inferenz zur Ermittlung der Regeln, die für die aktuelle Situation

anzuwenden sind und die Werte der linguistischen Ausgangsvariablen berechnen, zur Anwendung. Bild C.8 zeigt zur Erläuterung eine Untermenge der drei Regeln:

Regel 1: IF Abstand IS mittel	AND Winkel IS pos_klein	THEN Leistung IS pos_mittel
Regel 2: IF Abstand IS mittel	AND Winkel IS null	THEN Leistung IS null
Regel 3: IF Abstand IS weit	AND Winkel IS null	THEN Leistung IS pos_mittel

Bild C.8 – Untermenge der drei Regeln

Die Inferenz besteht aus den drei Unterfunktionen Aggregation, Aktivierung und Akkumulation.

– Aggregation (siehe Bild C.10).

Sie dient zur Bestimmung des Zugehörigkeitsgrades der Bedingung aus den Zugehörigkeitsgraden der elementaren Aussagen.

Regel 1:	Abstand = mittel	AND	Winkel = pos_klein	=	P_1
Regel 2:	Abstand = mittel	AND	Winkel = null	=	P_2
Regel 3:	Abstand = weit	AND	Winkel = null	=	P_3

für alle Regeln

Bild C.9 – Elemente der Aggregation

Der Min-Operator entspricht der AND-Aggregation. Bild C.9 zeigt, wie die *Aggregation* in diesem Fall berechnet wird.

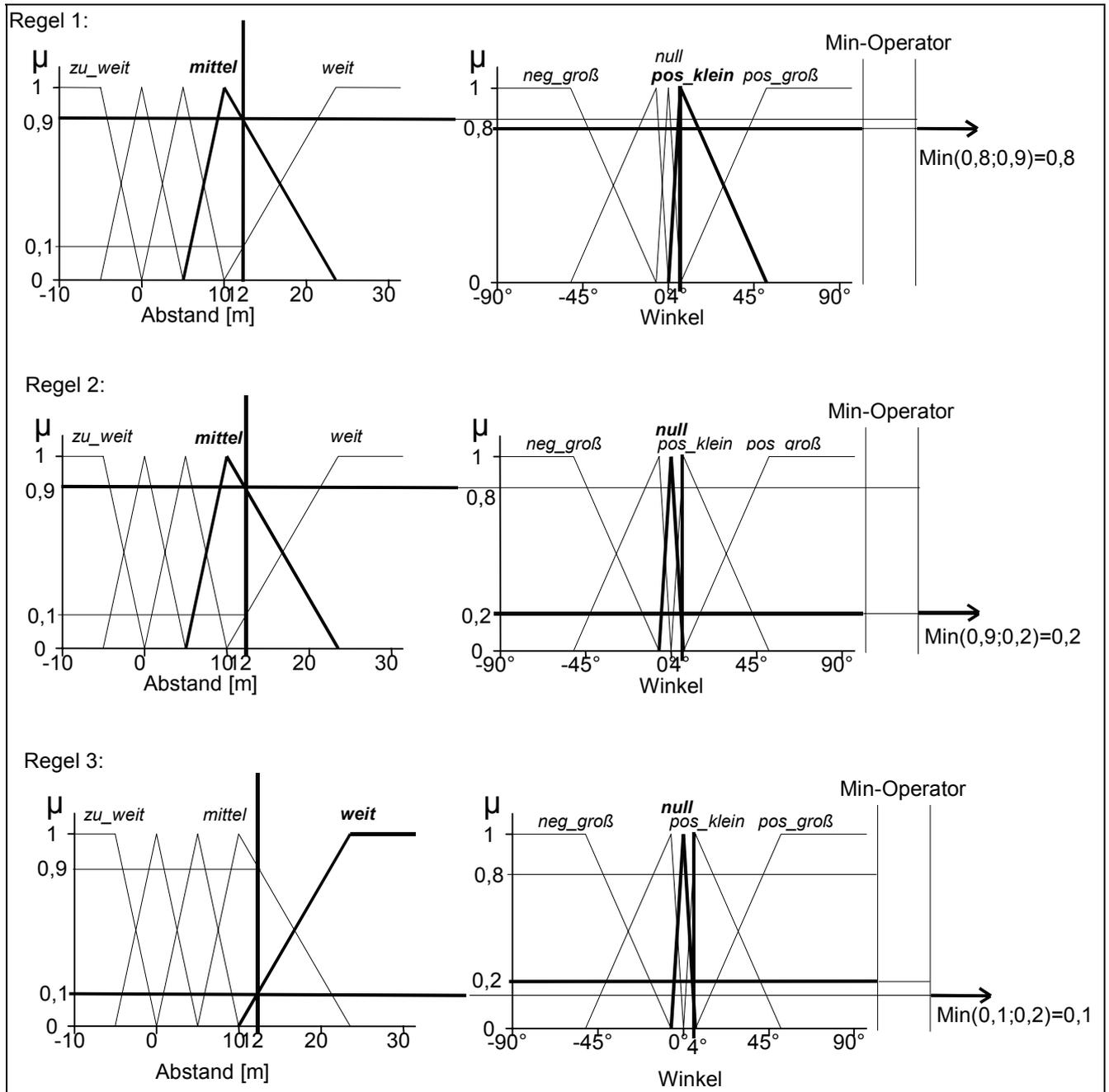


Bild C.10 – Prinzipien der Aggregation

- Aktivierung (siehe Bild C.12)
Sie dient zur Umwandlung der IF-THEN Konklusion.

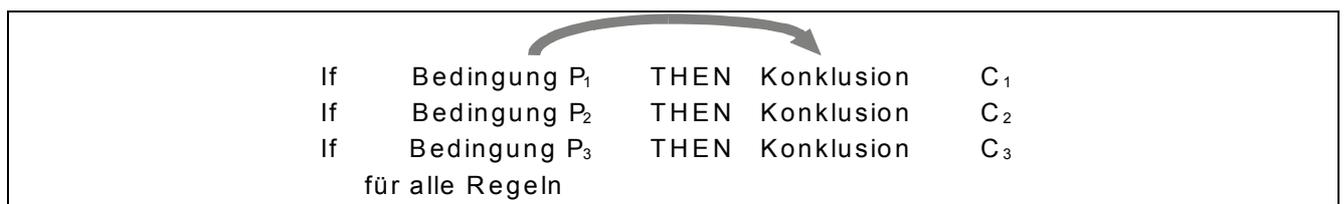


Bild C.11 – Elemente der Aktivierung

Bild C.11 zeigt, wie die Aktivierung in diesem Fall berechnet wird. Auf der linken Seite wird das Ergebnis der Aggregation, auf der rechten Seite das Ergebnis der Aktivierung beschrieben.

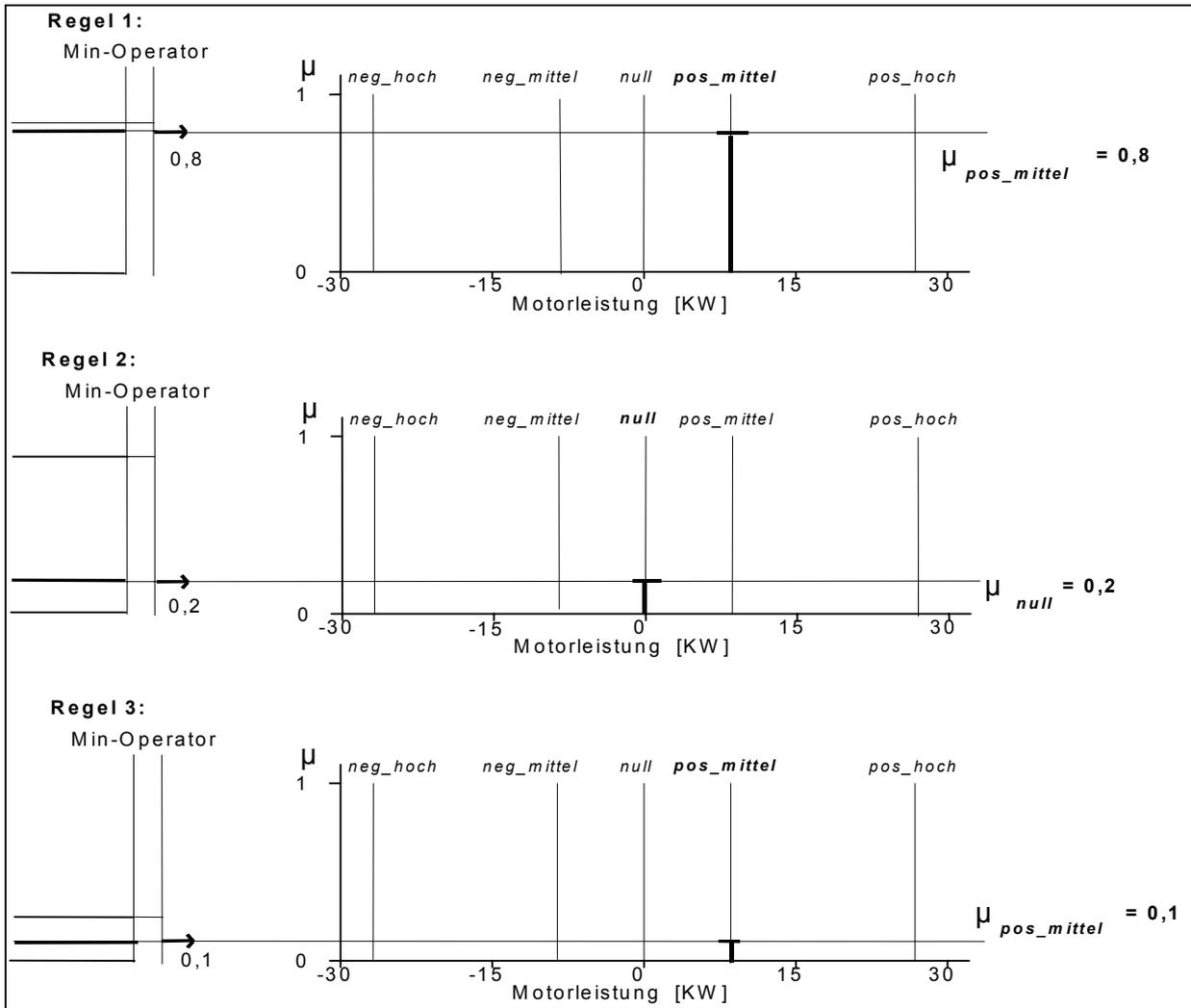


Bild C.12 – Prinzipien der Aktivierung

– Akkumulation (siehe Bild 14).

Sie dient zur Verknüpfung der gewichteten Ergebnisse der Regeln zu einem Gesamtergebnis.

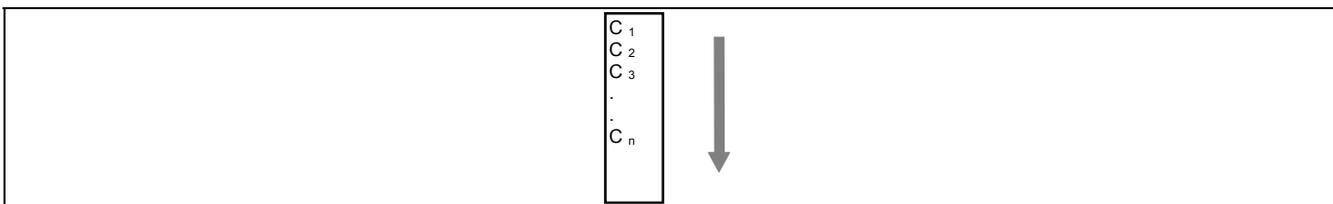


Bild C.13 – Elemente der Akkumulation

Das Ergebnis der Akkumulation der Regeln 1 bis 3 ist im unteren Teil von Bild C.13 gezeigt. Das Ergebnis beispielsweise des Singletons (Einermenge) *pos_mittel* ist $\text{Max}(0,8; 0,1) = 0,8$.

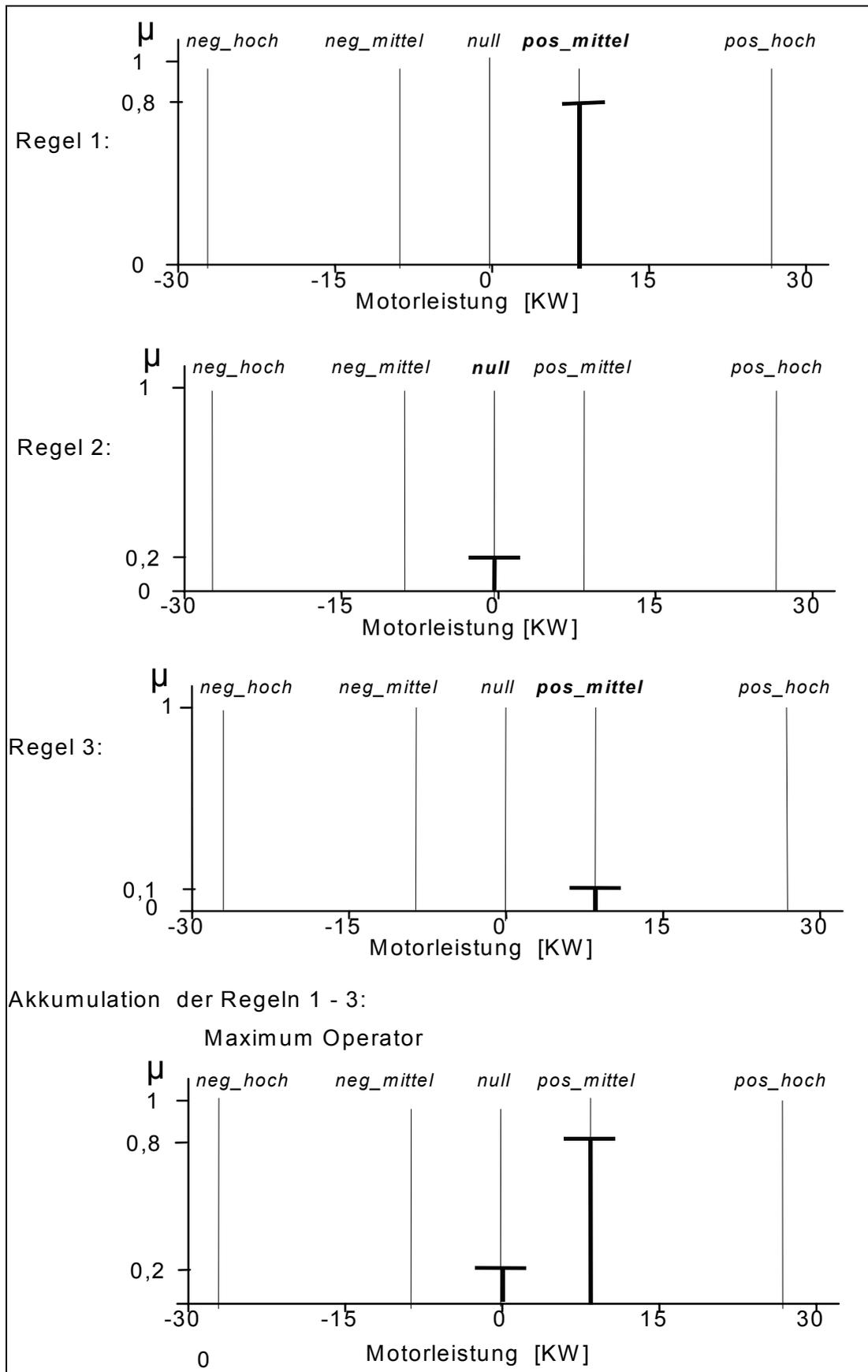


Bild C.14 – Prinzipien der Akkumulation

- Defuzzifizierung (siehe Bild C.15).

Die *Inferenz* liefert als Ergebnis eine *Fuzzy-Menge* oder ihre *Zugehörigkeitsfunktion*. Um es zum Einstellen der Motorleistung zu verwenden, muss es in einen scharfen Zahlenwert umgewandelt werden. Hier sollte der zu bestimmende *Wert* (immer eine reelle Zahl) die bestmögliche Darstellung der Information bieten, die in der vorliegenden *Fuzzy-Menge* enthalten ist. Bei Anwendung *Schwerpunkt-Methode* für Singletons (CoGS) wird die scharfe Ausgangsvariable Motorleistung wie folgt berechnet:

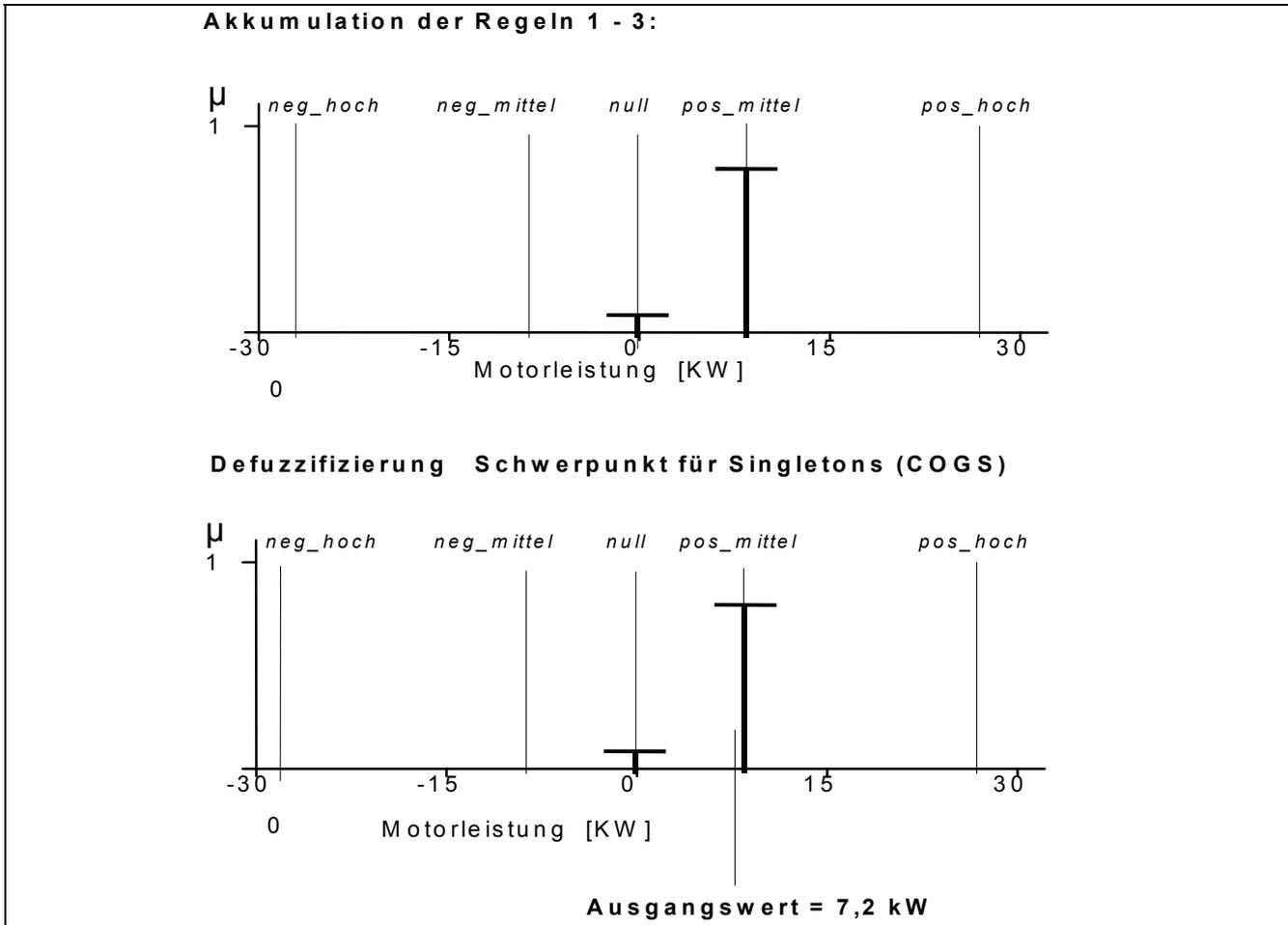


Bild C.15 – Defuzzifizierung

- Implementierung des Beispiels Containerkran in FCL (siehe Bild C.16):

```

FUNCTION_BLOCK Container_Kran

VAR_INPUT
    Abstand: REAL;
    Winkel: REAL;
END_VAR

VAR_OUTPUT
    Leistung: REAL;
END_VAR

FUZZIFY Abstand
    TERM zu_weit := (-5, 1) ( 0, 0);
    TERM null := (-5, 0) ( 0, 1) ( 5, 0);
    TERM nah := ( 0, 0) ( 5, 1) (10, 0);
    TERM mittel := ( 5, 0) (10, 1) (22, 0);
    TERM weit := (10, 0) (22, 1);
END_FUZZIFY

FUZZIFY Winkel
    TERM neg_gross := (-50, 1) (-5, 0);
    TERM neg_klein := (-50, 0) (-5, 1) ( 0, 0);
    TERM null := ( -5, 0) ( 0, 1) ( 5, 0);
    TERM pos_klein := ( 0, 0) ( 5, 1) (50, 0);
    TERM pos_gross := ( 5, 0) (50, 1);
END_FUZZIFY

DEFUZZIFY Leistung
    TERM neg_hoch := -27;
    TERM neg_mittel := -9;
    TERM null := 0;
    TERM pos_mittel := 9;
    TERM pos_hoch := 27;
    METHOD : CoGS;
    DEFAULT := 0;
END_DEFUZZIFY

RULEBLOCK No1
    AND : MIN;
    ACCU : MAX;
    RULE 1: IF Abstand IS weit AND Winkel IS null THEN Leistung IS pos_mittel;
    RULE 2: IF Abstand IS weit AND Winkel IS neg_klein THEN Leistung IS pos_gross;
    RULE 3: IF Abstand IS weit AND Winkel IS neg_gross THEN Leistung IS pos_mittel;
    RULE 4: IF Abstand IS mittel AND Winkel IS neg_klein THEN Leistung IS neg_mittel;
    RULE 5: IF Abstand IS nah AND Winkel IS pos_klein THEN Leistung IS pos_mittel;
    RULE 6: IF Abstand IS null AND Winkel IS null THEN Leistung IS null;
END_RULEBLOCK

END_FUNCTION_BLOCK

```

Bild C.16 – Beispiel in FCL

Der FCL-Funktionsbaustein muss in der gleichen Weise, wie es nach IEC 61131-3 für Programme gilt, aufgerufen werden. Der Grund dafür ist, dass der Fuzzy-Block von außen gesehen seine Schnittstelle nur über seine Variablen anbietet.

Der Aufruf für das oben gezeigte Beispiel könnte nach IEC 61131-3 lauten:

```
Container_Kran (Abstand:= EING_ABSTAND, Winkel:= EING_WINKEL);  
A:= Container_Kran.Leistung;
```

Die Variablen EING_ABSTAND und EING_WINKEL könnten direkt mit Eingangsvariablen der Steuerung verbunden oder aus anderen Werten berechnet sein.

Anhang D (informativ)

Beispiel für die Verwendung von Variablen im Regelblock

In einem Tunnelofen werden Kekse gebacken. Die Farbe der Kekse wird von einem Farbsensor am Ende des Tunnels gemessen. Es wird eine Fuzzy-Klassifikationsmethode verwendet, bei der die Farbe eine dreidimensionale Messung darstellt, um die Zugehörigkeit der gemessenen Farbe zu den drei folgenden Klassen zu bestimmen: Braun, hell, dunkel. Die Feuchte im Ofen wird ebenfalls gemessen. Der Ofen kann mit zwei Temperaturregelkreisen geregelt werden: einer für die erste Hälfte und einer für die zweite Hälfte des Ofens.

Das Prinzip der Regelung ist in den Bildern D.1 und D.2 angegeben.

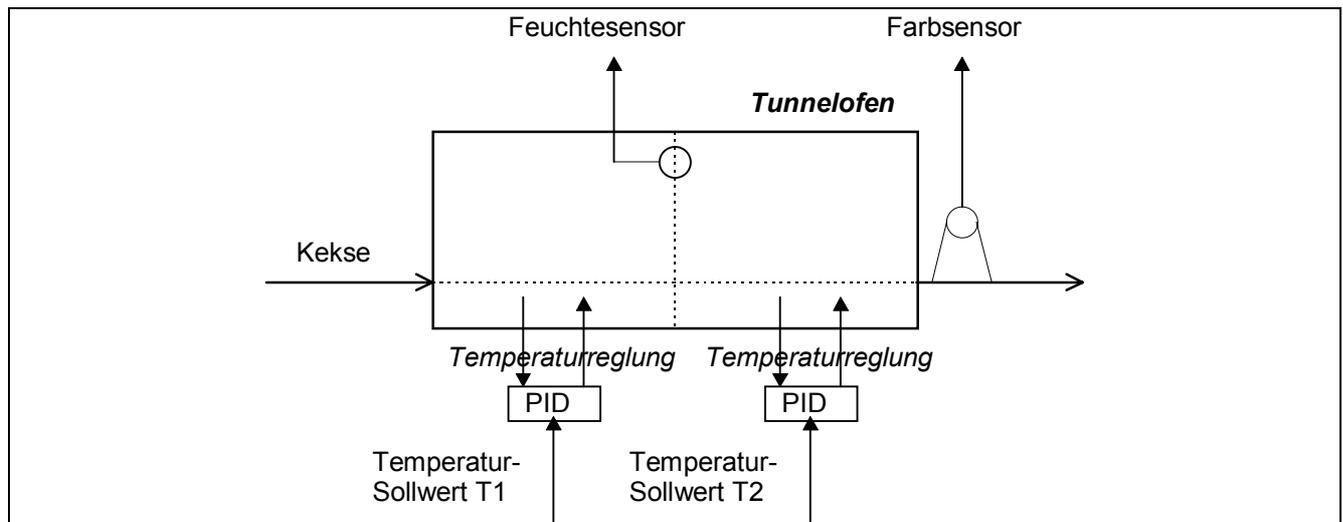


Bild D.1 – Prinzip des geregelten Systems

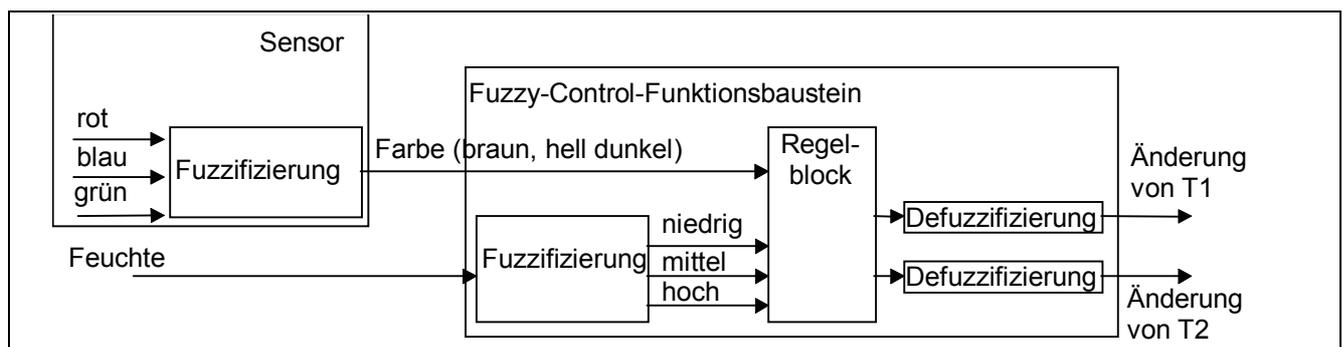


Bild D.2 – Prinzip der Fuzzy-basierten Ofenregelung

Der Regelblock enthält die folgenden 5 Regeln (siehe Bild D.3):

IF Feuchte	IS mittel AND Farbe IS braun	THEN dT1 IS null AND dT2 IS null
IF Feuchte	IS hoch	THEN dT1 IS positiv
IF Feuchte	IS niedrig	THEN dT1 IS negativ
IF Feuchte	IS mittel AND Farbe IS hell	THEN dT2 IS positiv
IF Feuchte	IS mittel AND Farbe IS dunkel	THEN dT2 IS negativ

Bild D.3 – Regelblock

Die FCL-Syntax für dieses Beispiel ist in Bild D.4 angegeben.

ANMERKUNG Wie unten gezeigt ist, kann statt der Verwendung der 3 Variablen hell, braun und dunkel nur 1 Variable vom Typ Structure verwendet werden. In diesem Beispiel ist dT2 undefiniert, wenn die Feuchte hoch oder niedrig ist.

```
TYPE
STRUCT Farbtyp
    braun : REAL;
    hell : REAL;
    dunkel : REAL;
END_STRUCT
END_TYPE

FUNCTION_BLOCK Ofenregelung
VAR_INPUT
    Feuchte : REAL;
    Farbe : Farbtyp;
END_VAR

VAR_OUTPUT
    dT1 : REAL;
    dT2 : REAL;
END_VAR

FUZZIFY Feuchte
    TERM niedrig := (30, 1) (50, 0);
    TERM mittel := (30, 0) (50, 1) (70, 1) (80, 0);
    TERM hoch := (70, 0) (80, 1);
END_FUZZIFY

DEFUZZIFY dT1
    TERM negativ := -5;
    TERM null := 0;
    TERM positiv := 5;
    METHOD : CoGS;
    DEFAULT := 0;
END_DEFUZZIFY

DEFUZZIFY dT2
    TERM negativ := -3;
    TERM null := 0;
    TERM positiv := 3;
    METHOD : CoGS;
    DEFAULT := 0;
END_DEFUZZIFY

RULEBLOCK Inferenz
    AND : MIN;
    ACCU : MAX;
    RULE 1 : IF Feuchte IS mittel AND Farbe = braun THEN dT1 IS null AND dT2 IS null;
    RULE 2 : IF Feuchte IS hoch THEN dT1 IS positiv;
    RULE 3 : IF Feuchte IS niedrig THEN dT1 IS negativ;
    RULE 4 : IF Feuchte IS mittel AND Farbe = hell THEN dT2 IS positiv;
    RULE 5 : IF Feuchte IS mittel AND Farbe = dunkel THEN dT2 IS negativ;
END_RULEBLOCK
END_FUNCTION_BLOCK
```

Bild D.4 – Beispiel in FCL

Anhang E (informativ)

Symbole, Abkürzungen und Synonyme

Tabelle E.1 – Symbole und Abkürzungen

CoA	Centre of Area	Flächenschwerpunkt
CoG	Centre of Gravity	Schwerpunkt
FB	Function Block	Funktionsbaustein
FBD	Function Block Diagram	FBS: Funktionsbaustein-Sprache
FCL	Fuzzy-Control Language	Fuzzy-Control-Sprache
IL	Instruction List	AWL: Anweisungsliste
MAX	Maximum operator	Maximum-Operator
MIN	Minimum operator	Minimum-Operator
PROD	Product operator	Produkt-Operator
ST	Structured Text	Strukturierter Text
μ	Degree of Membership	Zugehörigkeitsgrad
ω	Weight factor	Wichtungsfaktor

Tabelle E.2 – Synonyme (Englisch)

conclusion	consequent
accumulation	result aggregation
activation	composition
condition	antecedent
Centre of Gravity	Centroid of Area
Centre of Area	Bisector of Area

Anhang ZA (normativ)

Normative Verweisungen auf internationale Publikationen mit ihren entsprechenden europäischen Publikationen

Diese Europäische Norm enthält durch datierte oder undatierte Verweisungen Festlegungen aus anderen Publikationen. Diese normativen Verweisungen sind an den jeweiligen Stellen im Text zitiert, und die Publikationen sind nachstehend aufgeführt. Bei datierten Verweisungen gehören spätere Änderungen oder Überarbeitungen dieser Publikationen zu dieser Europäischen Norm nur, falls sie durch Änderung oder Überarbeitung eingearbeitet sind. Bei undatierten Verweisungen gilt die letzte Ausgabe der in Bezug genommenen Publikation (einschließlich Änderungen).

ANMERKUNG Ist eine internationale Publikation durch gemeinsame Abweichungen modifiziert worden, gekennzeichnet durch (mod.), dann gilt die entsprechenden EN/HD.

Publikation	Jahr	Titel	EN/HD	Jahr
IEC 60050-351	1998	International Electrotechnical Vocabulary – Part 351: Automatic control	–	–
IEC 61131-3	1993	Programmable controllers – Part 3: Programming languages	EN 61131-3	1993