

Rekurrente Fuzzy-Systeme

Vom Fachbereich 18
Elektrotechnik und Informationstechnik
der Technischen Universität Darmstadt
zur Erlangung der Würde eines
Doctor rerum naturalium (Dr.rer.nat.)
genehmigte Dissertation

von

Roland Kempf, M.A.

geboren am 28. Mai 1974 in Alzenau in Ufr.

Referent: Prof. Dr.-Ing. J. Adamy
Korreferent: Prof. Dr.-Ing. J. Lunze
Korreferent: Prof. Dr.rer.nat. W. Krabs
Tag der Einreichung: 29. September 2003
Tag der mündlichen Prüfung: 6. Februar 2004

Vorwort

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Institut für Automatisierungstechnik, Fachgebiet Regelungstheorie und Robotik an der Technischen Universität Darmstadt. Das Forschungsprojekt wurde von der Deutschen Forschungsgesellschaft (DFG) im Rahmen des Graduiertenkollegs "Intelligente Systeme für die Informations- und Automatisierungstechnik" sowie durch das Forschungsprojekt "Untersuchung von Fuzzy-Systemen mit chaotischer Dynamik" (AD 174/2-1) finanziell unterstützt.

Mein besonderer Dank geht an Professor Adamy, der mich an das Aufgabenfeld der rekurrenten Fuzzy-Systeme herangeführt hat. Die enge und fruchtbare Zusammenarbeit, aber auch der nötige Freiraum haben nicht unwesentlich zu der vorliegenden Arbeit beigetragen. Die Unterstützung, die ich in anderen Bereichen durch ihn erfahren habe, ermöglichen ein angenehmes Arbeiten.

Mein weiterer Dank gilt Professor Lunze und Professor Krabs für die freundliche Übernahme des Korreferates. Insbesondere danke ich Professor Krabs für sein stets offenes Ohr und seine Begeisterung für mathematische Untersuchungen, die er auf mich übertragen konnte.

Auch meinen Kolleginnen und Kollegen möchte ich danken, im Speziellen Eva Brucherseifer für die Begleitung über den Zeitraum der Promotion und Anna Flemming u.a. für die kritische Durchsicht des Manuskripts. Dem Sekretariat danke ich für die organisatorischen Hilfen und insbesondere Frau Lehmeier, die mit Engelsgeduld alle Grafiken dieser Arbeit erstellt hat.

Des Weiteren haben mir die kritischen Fachgespräche mit Studentinnen und Studenten im Rahmen der Betreuung von zahlreichen Studien- und Diplomarbeiten und Seminaren sehr geholfen, die jeweils vorliegenden Ergebnisse zu präzisieren und zu erweitern. Ihre Ideen haben mir Anregungen zu weiteren Untersuchungen gegeben.

Nicht zuletzt möchte ich meinen Eltern für ihre stetige Unterstützung und Zuwendung danken. Wohl dem, der solche Freunde hat.

Darmstadt, im September 2003

Roland Kempf

Inhaltsverzeichnis

1	Einleitung	1
2	Überblick: Fuzzy-Systeme mit Dynamik	4
2.1	Grundlagen und statische Fuzzy-Systeme	4
2.2	Zeitdiskrete Fuzzy-Systeme	6
2.3	Vergleich zeitdiskreter Fuzzy-Systeme	15
3	Definitionen	18
3.1	Definition rekurrenter Fuzzy-Systeme	18
3.2	Form der Transitionsfunktion	22
4	Automaten und Automatenähnlichkeit	27
4.1	Linguistischer Automat	27
4.2	Automatenähnlichkeit rekurrenter Fuzzy-Systeme	29
5	Chaos	35
5.1	Beispiel eines chaotischen rekurrenten Fuzzy-Systems	35
5.2	Chaos in dynamischen Systemen	38
5.3	Chaoskriterien für rekurrente Fuzzy-Systeme	40
5.4	Grenzen der Chaoskriterien	43
6	Ruhelagen	48
6.1	Standardisierte, rekurrente Fuzzy-Systeme	48
6.2	Ruhelagen und Nichtexpansivität	50
6.3	Konvergenzsätze für standardisierte rekurrente Fuzzy-Systeme	55
6.4	Lokalisierung von Ruhelagen	59
6.5	Asymptotische Stabilität von Ruhelagen	62
7	Erreichbarkeit und Steuerbarkeit	65
7.1	Erreichbarkeit: Definitionen und Grundlagen	65

7.2	Erreichbarkeit in rekurrenten Fuzzy-Systemen	68
7.3	Steuerbarkeit: Definitionen und Grundlagen	78
7.4	Steuerbarkeit in rekurrenten Fuzzy-Systemen	80
8	Anwendung: Sequentielle Mustererkennung	84
8.1	Grundlagen sequentieller Mustererkenner	84
8.2	Sequentielle Mustererkennung mit rekurrenten Fuzzy-Systemen	89
8.3	Anwendungsbeispiel aus der Stahlindustrie	91
8.4	Sequentielle Mustererkennung mit hybriden rekurrenten Fuzzy-Systemen .	97
9	Anwendung: Verkehrssimulation	102
9.1	Verkehrssimulation mit zellulären Automaten	102
9.2	Verkehrssimulation mit rekurrenten Fuzzy-Systemen	111
9.3	Simulationsergebnisse	114
10	Zusammenfassung	118
A	Sammlung spezieller rekurrenter Fuzzy-Systeme	120
B	Lemmata für Stabilitätssätze	125
C	Gültige Sätze bei Min-Max Inferenz	132
D	Sätze über Konvergenz und Ω-Grenzmengen	133
	Literatur	140

1 Einleitung

Zur Beschreibung von Vorgängen, die seine Lebensumstände betreffen, hat der Mensch ein mächtiges Werkzeug geschaffen: die Sprache. Mit ihr kann er unter anderem kausale Zusammenhänge ausdrücken. Oft gelingt es, solche Beschreibungen in ein strukturiertes Schema von "Wenn ..., dann..."-Regeln zu bringen, beispielsweise "Wenn es viel regnet, dann brauche ich nicht zu gießen".

Wendet man sich den Naturwissenschaften bzw. ihrer Nutzung in den Ingenieurwissenschaften zu, so wird schnell deutlich, dass in diesen Bereichen die Alltagssprache oft nicht geeignet ist. Sie ist zu unpräzise. Außerdem kann sie nicht direkt von Maschinen genutzt werden. Deshalb hat sich eine andere Art der Sprache herausgebildet und bewährt: Die Formelsprache der Mathematik.

In vielen Fällen ist es wünschenswert, beide Beschreibungsformen parallel oder ergänzend nutzen zu können. Mit der Entwicklung der Fuzzy-Logik [17, 44] lieferte L. Zadeh [104] einen Brückenschlag zwischen der Alltagssprache und der mathematischen Formelsprache. Dieser Ansatz ermöglicht es, unscharfe Begriffe mathematisch mit Fuzzy-Mengen zu modellieren. In der regelbasierten Fuzzy-Logik, dem approximativen Schließen [17], können obengenannte "Wenn..., dann..."-Regeln in mathematische Funktionen umgewandelt werden. Damit wird umgangssprachliches Wissen technisch nutzbar. Umgekehrt lässt sich die erhaltene mathematische Funktion auch sprachlich interpretieren. Fuzzy-Logik bietet also eine transparente Schnittstelle zwischen Sprache und Mathematik. Dieser Zusammenhang wird in vielen Bereichen [92, 75], beispielsweise in der Diagnose [100], Klassifikation [12], Automatisierung [89, 66] und Modellierung [8], zur Lösung von Problemen ausgenutzt.

In zahlreichen Anwendungen wird mit Fuzzy-Logik ein statischer Zusammenhang nachgebildet. Die betrachteten Größen verändern sich nicht oder ihre Veränderung wird zumindest nicht explizit durch die Regeln beschrieben. Erst in den letzten Jahren wurden vermehrt Ansätze für Fuzzy-Systeme mit Dynamik vorgestellt und untersucht. Die verschiedenen Ansätze haben dabei unterschiedliche Zielsetzungen.

So sind Fuzzy-Automaten [99] wie endliche Automaten aufgebaut, mit dem Unterschied, dass sie statt der klassischen Schaltlogik die Fuzzy-Logik benutzen. Dynamische Fuzzy-Systeme verwenden die regelbasierte Fuzzy-Logik, um iterative Abbildungen von Fuzzy-Mengen auf Fuzzy-Mengen zu beschreiben. Fuzzy dynamical systems bilden ebenfalls Fuzzy-Mengen auf Fuzzy-Mengen ab. Dieser Ansatz ist allerdings allgemein gehalten, um ein möglichst breites Spektrum von Systemen abzudecken. Eine detailliertere Darstellung dieser und weiterer Ansätze ist in Kapitel 2 zu finden.

Ein weiterer Ansatz sind rekurrente Fuzzy-Systeme. Sie betten eine Fuzzy-Funktion, beste-

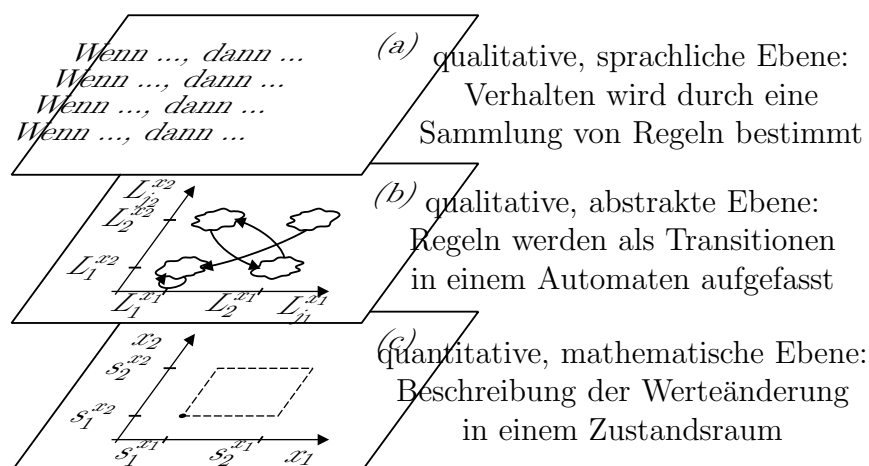


Abbildung 1.1: Beschreibung von rekurrenten Fuzzy-Systemen auf qualitativer und quantitativer Ebene. Die wolkigen Knoten im Zustandsgraphen symbolisieren dabei den unscharfen sprachlichen Charakter der Zustände.

hend aus den Teilschritten Fuzzifizierung, Inferenz und Defuzzifizierung, durch Einführung einer Rückkopplung in eine dynamische Struktur ein. Rekurrente Fuzzy-Systeme wurden aus unterschiedlichen Motivationen 1994 und 1995 eingeführt. In dem einen Fall [1, 2] stand ihre Interpretation als unscharfe Automaten im Vordergrund, im anderen Fall [29] die Nähe zu rekurrenten neuronalen Netzen. Außer der Analyse des Lernens von rekurrenten Fuzzy-Systemen mit Neuro-Fuzzy-Methoden wurden weitere Problembereiche bisher kaum behandelt und eine mathematische Theorie rekurrenter Fuzzy-Systeme fehlte vollständig.

Die qualitative Beschreibung eines rekurrenten Fuzzy-Systems durch einfache Regeln auf einer sprachlichen Ebene suggeriert ein umfassendes Verständnis für die mathematischen Eigenschaften dieses Systems. Diese Vorstellung trifft nicht immer zu. Daher ist im Normalfall eine Untersuchung der zum Teil komplexen mathematischen Struktur des jeweiligen rekurrenten Fuzzy-Systems notwendig. Diese Arbeit will einen Großteil dieser Untersuchungen abdecken und eine mathematische Theorie rekurrenter Fuzzy-Systeme entwickeln.

Ein besonderes Augenmerk wird bei der Analyse rekurrenter Fuzzy-Systeme auf die Übertragung der Ergebnisse der mathematischen Untersuchung auf die sprachliche Ebene der Regeln gelegt. Damit muss sich ein Anwender im Idealfall nicht mit der mathematischen Untersuchung selbst beschäftigen, sondern kann mit Hilfe von Kriterien anhand der Regeln feststellen, ob ein vorliegendes rekurrentes Fuzzy-System bestimmte mathematische Eigenschaften aufweist oder nicht.

Für die Darstellung der Ergebnisse ist es sinnvoll, zwischen der qualitativen, sprachlichen Ebene und der quantitativen, mathematischen Ebene eine zusätzliche qualitative, abstrakte Ebene einzuführen. In dieser Zwischenebene werden die Regeln als Transitionen in einem Automaten aufgefasst. Die drei fiktiven Ebenen sind in Abbildung 1.1 dargestellt und sollen in den nächsten Kapiteln mit Inhalt und Bedeutung ausgefüllt werden.

Die Arbeit gliedert sich wie folgt: Nach einem Überblick über Fuzzy-Systeme mit Dynamik in Kapitel 2 werden in Kapitel 3 rekurrente Fuzzy-Systeme formal definiert, die nötige

Notation eingeführt und einige Grundlagen dieser Systeme behandelt.

Inwieweit das Verhalten von rekurrenten Fuzzy-Systemen die Regeln widerspiegelt, wird in Kapitel 4 dargestellt. Es zeigt sich, dass sich rekurrente Fuzzy-Systeme bei geeigneter Auslegung ähnlich wie endliche Automaten verhalten. Kapitel 5 zeigt, dass rekurrente Fuzzy-Systeme im Gegensatz dazu auch chaotisches Verhalten erzeugen können. Es werden einige Kriterien hergeleitet, mit denen sich alleine anhand der Regeln Chaos in solchen Systemen nachweisen lässt.

Nachdem die Bandbreite der Dynamikarten abgesteckt worden ist, beschäftigt sich Kapitel 6 mit der Untersuchung von Ruhelagen in rekurrenten Fuzzy-Systemen. Wieder werden die Ergebnisse so weit wie möglich mit der Struktur der linguistischen Regeln in Verbindung gebracht. Im darauffolgenden Kapitel 7 wird die Untersuchung auf Erreichbarkeit und Steuerbarkeit ausgedehnt.

Abgerundet wird die Arbeit durch zwei Anwendungsbeispiele: In Kapitel 8 wird ein rekurrentes Fuzzy-System vorgestellt, welches sich zur sequentiellen Mustererkennung eignet. Ausgehend von der sprachlichen Beschreibung des Verhaltens eines sequentiellen Musterkenners wird ein solcher konstruiert. Das zweite Anwendungsbeispiel ist die Modellierung des Verhaltens eines Autofahrers auf einer zweispurigen Autobahn in Kapitel 9. Das Beispiel zeigt, inwieweit rekurrente Fuzzy-Systeme menschliches Entscheidungsverhalten modellieren können, welches sich in "Wenn..., dann..."-Regeln niederschreiben lässt.

2 Überblick: Fuzzy-Systeme mit Dynamik

Seit der Einführung der Fuzzy-Logik im Jahre 1965 durch Zadeh sind eine Vielzahl von Untersuchungen durchgeführt worden. Der Facettenreichtum im Zusammenhang mit Fuzzy-Logik und Fuzzy-Systemen führte in den verschiedenen damit tangierten Wissenschaftsbereichen wie der Mathematik [17], den Ingenieurwissenschaften [38, 61] und den weichen Naturwissenschaften [91] zu einer fast unüberschaubaren Zahl von Anwendungen, Modifikationen und Erweiterungen. Neben dem Einsatz zur Modellierung und Lösung statischer Probleme sind Fuzzy-Methoden häufig auch in Bereichen zu finden, die dynamische Systeme betreffen.

In diesem Kapitel wird eine Einführung in die Grundlagen von statischen Fuzzy-Systemen gegeben. Ausgehend davon werden verschiedene Ansätze von Fuzzy-Systemen mit Dynamik vorgestellt. Diese Zusammenstellung soll einen Überblick über den aktuellen Stand der Forschung geben. Am Ende werden die verschiedenen Ansätze miteinander verglichen.

2.1 Grundlagen und statische Fuzzy-Systeme

Üblicherweise beschreibt man Größen durch einen Wert, zum Beispiel eine Temperatur durch $T = 32^\circ\text{C}$. Diese Angabe ist präzise. Will oder muss man mit unpräzisen Wertangaben arbeiten, so kann man Wertebereiche angeben. Eine hohe Temperatur stammt zum Beispiel aus dem Intervall $[28^\circ\text{C}, 36^\circ\text{C}]$. Diese Menge kann mathematisch durch ihre *charakteristische Funktion* $\chi : \mathbb{R} \rightarrow \{0, 1\}$ festgelegt werden. Es gilt $\chi(T) = 1$, falls $T \in [28^\circ\text{C}, 36^\circ\text{C}]$ und $\chi(T) = 0$, falls $T \notin [28^\circ\text{C}, 36^\circ\text{C}]$.

Mit dieser Art der Modellierung der hohen Temperaturen ergibt sich allerdings folgendes Problem: Die Zuordnung der Temperaturwerte zu einer Menge ist unstetig. Sie entspricht nicht der umgangssprachlichen Vorstellung, denn ein Temperaturwert von 27.99°C wird durch diese Festlegung nicht als hohe Temperatur angesehen, während 28.00°C als hohe Temperatur gilt. Die Aussage " $T = 27.99^\circ\text{C}$ ist eine hohe Temperatur" mag vielleicht nicht vollkommen wahr sein, aber doch bis zu einem gewissen Grad. Mit Hilfe der Fuzzy-Logik kann man derartigen Aussagen einen Wahrheitswert zwischen null und eins zuordnen. Die *Zugehörigkeitsfunktion* $\mu : \mathbb{R} \rightarrow [0, 1]$ beschreibt eine solche Zuordnung auf der Menge der möglichen Temperaturen. Ein Beispiel ist in Abbildung 2.1 gezeigt. Die Zugehörigkeitsfunktion ersetzt die charakteristische Funktion einer klassischen, präzise definierten Menge. Durch sie wird eine nicht präzise Menge, eine *Fuzzy-Menge*, festgelegt.

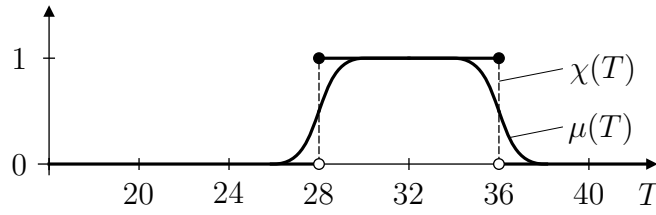


Abbildung 2.1: Charakteristische Funktion χ einer klassischen Menge und Zugehörigkeitsfunktion μ einer Fuzzy-Menge, die beide den Temperaturbereich einer hohen Temperatur beschreiben.

Zugehörigkeitsfunktionen von Fuzzy-Mengen müssen nicht unbedingt auf einer reellen Menge $\mathbb{R}$ definiert sein, sondern können auf beliebigen Mengen definiert werden. Üblicherweise sind diese Mengen endlich oder Teilmengen von reellen Vektorräumen $\mathbb{R}^n$.

Ein funktionaler Zusammenhang zwischen Zahlenwerten wird normalerweise durch eine Gleichung $y = f(x)$, eine Definitionsmenge X und eine Wertemenge Y eindeutig festgelegt. Man spricht von einem deterministischen, statischen System, welches erlaubte Eingangswerte $x \in X$ auf Ausgangswerte $y = f(x) \in Y$ abbildet. Gewöhnlich sind die Ein- und Ausgangsgrößen Zahlenwerte oder Vektoren. Bei Vektoren $\mathbf{x}$ und $\mathbf{y}$ wird entsprechend eine Vektorfunktion $\mathbf{y} = \mathbf{f}(\mathbf{x})$ verwendet.

Funktionale Zusammenhänge zwischen Fuzzy-Mengen lassen sich auf ähnliche Weise beschreiben. Dazu kann das Erweiterungsprinzip von Zadeh [91, 17] verwendet werden. Im einfachsten Fall geht man dabei von einer Abbildung $f : X \rightarrow Y$ einer klassischen Menge X in eine klassische Menge Y aus. Diese Abbildung wird für die Abbildung $\tilde{f}$ zwischen den Fuzzy-Mengen $\tilde{X}$ und $\tilde{Y}$ auf Mengen X und Y wie folgt erweitert: Ausgehend von einer Fuzzy-Menge $\tilde{x} \in \tilde{X}$, die durch ihre Zugehörigkeitsfunktion $\mu^{\tilde{x}}(x)$ mit $\mu^{\tilde{x}} : X \rightarrow [0, 1]$ festgelegt ist, errechnet man die Zugehörigkeitsfunktion $\mu^{\tilde{y}}(y)$ mit $\mu^{\tilde{y}} : Y \rightarrow [0, 1]$ der Fuzzy-Menge $\tilde{y} = \tilde{f}(\tilde{x}) \in \tilde{Y}$ durch

$$\mu^{\tilde{y}}(y) = \sup_{y=f(x)} \mu^{\tilde{x}}(x). \quad (2.1)$$

Die Mengen $\tilde{X}$ und $\tilde{Y}$ sind gewöhnlich Fuzzy-Mengen auf reellwertigen Vektorräumen oder auf endlichen Mengen X und Y . Zusammen mit der Funktion $\tilde{f}$ legen sie ein statisches Fuzzy-System fest.

Eine weitere gebräuchliche Art, funktionale Zusammenhänge zwischen Fuzzy-Mengen zu beschreiben, liefert der Ansatz der regelbasierten Fuzzy-Logik. Ausgangspunkt ist die sprachlich formulierte Beziehung zwischen Größen, die durch Fuzzy-Mengen $\tilde{x}$ und $\tilde{y}$ beschrieben werden. Die Beziehungen sind üblicherweise in Form von "Wenn ..., dann ..." -Regeln gegeben, also "Wenn die Eingangsgröße gleich $\tilde{x}$ ist, dann ist die Ausgangsgröße gleich $\tilde{y}$ ", zum Beispiel "Wenn die Temperatur hoch ist, dann ist der Druck im Kessel hoch". Oft stehen mehrere sprachliche Ausdrücke in der Voraussetzung der Regel, beispielsweise "Wenn die erste Eingangsgröße gleich $\tilde{x}_1$ ist und die zweite Eingangsgröße gleich $\tilde{x}_2$ ist, dann ist die Ausgangsgröße gleich $\tilde{y}$."

Die regelbasierte Fuzzy-Logik wertet solche sprachlichen Regeln aus. Dazu werden die Fuzzy-Mengen wieder durch Zugehörigkeitsfunktionen dargestellt. Die Und-Verknüpfung zwischen Fuzzy-Mengen $\tilde{x}_1$ und $\tilde{x}_2$ wird durch einen mathematischen Operator ersetzt. Beispiele solcher Operatoren sind die Produktbildung, der Minimumsoperator oder allgemein so genannte *T – Normen* [17, 11, 44, 97]. Auch für die Schlussfolgerung steht eine Reihe von mathematischen Operatoren zur Verfügung [17, 11, 44, 97], beispielsweise der Maximumsoperator. Um einen Zusammenhang zwischen Größen durch eine sprachliche Beschreibung vollständig zu charakterisieren ist normalerweise eine Vielzahl von Regeln nötig. Die Gesamtheit der Regeln bildet eine *Regelbasis*. Die einzelnen Regeln der Regelbasis gelten alle parallel je zu einem gewissen Grad. Dies lässt sich durch eine (fuzzy-)logische Oder-Verknüpfung zum Ausdruck bringen. Die Oder-Verknüpfung wird wieder mit Hilfe eines mathematischen Operators ausgewertet. Beispiele dafür sind die Summation, der Maximumsoperator und andere [11, 44, 97]. Damit ist es möglich, die Regelbasis in eine Abbildung von Fuzzy-Mengen auf Fuzzy-Mengen unzuschreiben. Eine solche Abbildung heißt *Inferenz*.

Schließlich gibt es noch Abbildungen von Zahlenwerten auf Fuzzy-Mengen und Fuzzy-Mengen auf Zahlenwerte. Zugehörigkeitsfunktionen stellen die wichtigste Abbildung von Zahlenwerten auf Fuzzy-Mengen dar. Bei dem Einsetzen eines Wertes x_0 in eine Zugehörigkeitsfunktion spricht man von der *Fuzzifizierung* des Wertes x_0 . Umgekehrt wird einer Fuzzy-Menge bei der *Defuzzifizierung* ein Zahlenwert zugeordnet. Für die Defuzzifizierung gibt es auch eine Reihe von Methoden [11, 44, 97], zum Beispiel die gewichtete Mittelwertbildung oder die Wahl eines Zahlenwertes, an dem die Fuzzy-Menge ihren maximalen Wahrheitswert annimmt.

Schaltet man eine Fuzzifizierung, eine Inferenz und eine Defuzzifizierung hintereinander, so erhält man wieder eine gewöhnliche Funktion. Ein so entstandenes System wird hier als *vollständiges Fuzzy-System* bezeichnet. Die Funktion wertet in ihrem Kern einen sprachlich formulierten Zusammenhang aus und erlaubt es dadurch, das sprachlich formulierte Wissen auf gängige Weise in Modellen, Maschinen und Anlagen zu nutzen.

Alle dargestellten statischen Systeme können auch in eine dynamische Struktur eingebettet werden. So erzeugte dynamische Systeme werden in den nächsten Abschnitten behandelt.

2.2 Zeitdiskrete Fuzzy-Systeme

Ein zeitdiskretes System ist durch eine Funktion $\mathbf{f} : X \times U \rightarrow X$, eine Ausgangsfunktion $\mathbf{g} : X \times U \rightarrow Y$ und die Gleichungen

$$\mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k)) \quad (2.2)$$

$$\mathbf{y}(k) = \mathbf{g}(\mathbf{x}(k), \mathbf{u}(k)) \quad (2.3)$$

festgelegt. Ein Blockschaltbild ist in Abbildung 2.2 zu sehen. Das System bildet die Eingangsgrößen $\mathbf{u}(k)$ und Zustandsgrößen $\mathbf{x}(k)$ auf neue Zustandsgrößen $\mathbf{x}(k+1)$ ab. Die Ausgangsfunktion $\mathbf{g}$, die aus den Zustandsgrößen $\mathbf{x}(k)$ und den Eingangsgrößen $\mathbf{u}(k)$ die Ausgangsgrößen $\mathbf{y}(k)$ berechnet, spielt für die Dynamik des Systems keine Rolle. Daher wird sie im Weiteren nicht mehr explizit betrachtet.

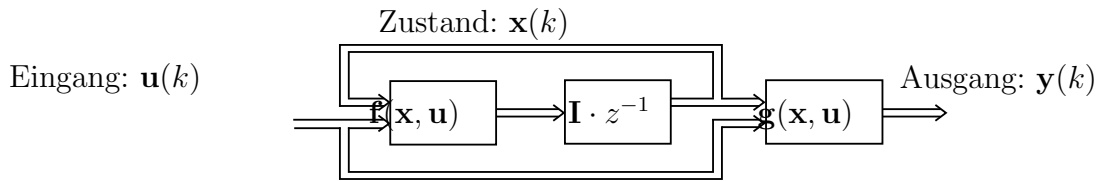


Abbildung 2.2: Blockschaltbild eines allgemeinen zeitdiskreten dynamischen Systems. Mit I wird die Einheitsmatrix bezeichnet.

Je nach Definitionsbereich und Funktionstyp unterscheidet man zwischen verschiedenen zeitdiskreten dynamischen Systemen. Im Allgemeinen sind die Zustandsmenge X und die Eingabemenge U je mit einer Metrik versehen, die sie zu metrischen Räumen macht. Erst dann können stetige Abbildungen f und g behandelt werden.

Häufig sind die Definitionsbereiche X und U für die Zustandsgrößen x und Eingangsgrößen u Teilmengen von reellen Vektorräumen. Allgemein spricht man von gewöhnlichen zeitdiskreten Systemen oder Differenzgleichungen. Bei der Verwendung von linearen Funktionen f und g ergeben sich lineare zeitdiskrete Systeme, bei nichtlinearen Funktionen dementsprechend nichtlineare zeitdiskrete Systeme. Nichtlineare Systeme unterscheidet man weiter nach ihrem Ursprung und ihren Eigenschaften. Werden zum Beispiel neuronale Netze für f und g eingesetzt, so ergeben sich rekurrente neuronale Netze [33].

Sind die Mengen X und U endliche Mengen, so stellt das zeitdiskrete dynamische System einen endlichen Automaten dar. In einem solchen Fall ändert sich der Zustand x des Systems in Abhängigkeit von Eingabewerten u . Die Abbildungen f und g lassen sich durch logische Schaltfunktionen darstellen, die auf einer zweiwertigen oder mehrwertigen Logik mit endlich vielen Wahrheitswerten beruhen.

Für die statische Funktion $f(x, u)$ können auch statische Fuzzy-Systeme eingesetzt werden. So erhält man Fuzzy-Systeme mit Dynamik. In den letzten Jahren ist eine Vielzahl von Ansätzen vorgestellt und untersucht worden. Dadurch ist ein Gewirr aus Bezeichnungen entstanden, in dem sehr ähnliche Ansätze mit unterschiedlichen Bezeichnungen belegt wurden und ähnliche und zum Teil identische Bezeichnungen für ganz unterschiedliche Ansätze verwendet wurden. Im Folgenden werden vier Ansätze von zeitdiskreten Fuzzy-Systemen vorgestellt.

Fuzzy Automaten

Fuzzy Automaten [99] bauen auf endlichen, deterministischen Automaten auf und werden auch unter der Rubrik der "fuzzy finite state machines" [30, 70] behandelt. Wie oben beschrieben ist ein deterministischer Automat ein zeitdiskretes System, dessen Eingangs- menge U , Zustandsmenge X und gegebenenfalls Ausgangsmenge Y endliche Mengen sind. Die Transitionsfunktion $f : X \times U \rightarrow X$ des Automaten beschreibt die Zustandsübergänge von $x(k)$ nach $x(k + 1)$, die durch eine Eingangsgröße $u(k)$ aus U hervorgerufen werden. Die Ausgabefunktion $g : X \times U \rightarrow Y$ bestimmt den Ausgabewert $y(k)$ in Abhängigkeit von $x(k)$ und $u(k)$.

Zählt man die Elemente der endlichen Zustandsmenge X durch, so lässt sich jeder Zustand eines Automaten durch einen Vektor $\chi^x = (\chi_j^x)$ darstellen, dessen Eintrag an der entsprechenden Komponente eins ist und sonst null. Der Vektor entspricht der charakteristischen Funktion der einzelnen Zustandswerte auf dem Zustandsraum X . Für jeden Eingangswert $u(k)$ lässt sich dann die Transitionsfunktion $f(\cdot, u(k))$ durch eine Matrizenmultiplikation mit einer Matrix $(\chi_{w,j}^{f(u(k))})$ realisieren. Die Matrix hat an der Stelle $\chi_{w,j}^{f(u(k))}$ einen Eintrag von eins, falls durch die Transitionsfunktion $f(x, u)$ der j -te Zustandwert auf den w -ten Zustandwert abgebildet wird. Andernfalls ist $\chi_{w,j}^{f(u(k))} = 0$. Der Zahlenwert $\chi_{w,j}^{f(u(k))}$ gibt den Wahrheitswert der Transition vom j -ten auf den w -ten Zustandswert wieder.

Führt beispielsweise eine Eingabe $u(k)$ in einem Automaten mit vier Zuständen eine Transition vom zweiten Zustand, d.h. $\chi^x(k) = (0 \ 1 \ 0 \ 0)^T$, auf den vierten Zustand, d.h. $\chi^x(k+1) = (0 \ 0 \ 0 \ 1)^T$, so ist $\chi_{4,2}^{f(u(k))} = 1$. Die Transitionen von den anderen Zuständen legen die restlichen Einträge von $\chi_{w,j}^{f(u(k))} \neq 0$ fest. So ergibt sich zum Beispiel

$$\chi^x(k+1) = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} = (\chi_{w,j}^{f(u(k))}) \cdot \chi^x(k). \quad (2.4)$$

Die Auswahl der richtigen Matrix $(\chi_{w,j}^{f(u(k))})$ in Abhängigkeit des Eingabewertes $u(k)$ kann durch die gleiche Systematik beschrieben werden. Dazu zählt man alle Elemente der endlichen Eingabemenge U durch und assoziiert jede Eingabe mit einem $|U|$ -dimensionalen Vektor $\chi^u = (\chi_q^u)$, dessen entsprechende Komponente eins ist und sonst null. Zum Beispiel wird für einen zweielementigen Eingangsraum der zweite Eingangswert mit dem Vektor $(0 \ 1)^T$ assoziiert. Analog werden auch die Matrizen $(\chi_{w,j}^{f(u)})$ durchgezählt und in einen Vektor geschrieben. Dieser Vektor aus Matrizen ist ein dreidimensionales Array mit den drei Indizes j, q und w und wird mit $(\chi_{w,q,j}^f)$ bezeichnet. Ein Eintrag $\chi_{w,q,j}^f$ ist genau dann eins, falls bei der Eingabe des q -ten Elements aus U eine Transition vom j -ten Zustandswert auf den w -ten Zustandswert möglich ist. Ist die Transition nicht möglich, so ist $\chi_{w,q,j}^f = 0$. Die Einträge der gesuchten Matrix $(\chi_{w,j}^{f(u(k))})$ erhält man dann durch eine verallgemeinerte Vektormultiplikation aus

$$\chi_{w,j}^{f(u(k))} = \sum_q \chi_{w,q,j}^f \cdot \chi_q^u(k). \quad (2.5)$$

Statt der Matrizenmultiplikation lassen sich auch andere Operatoren verwenden. Bei der Verwendung von Supremum und Minimum liefern die Gleichungen

$$\chi_{w,j}^{f(u(k))} = \sup_q \min(\chi_{w,q,j}^f, \chi_q^u(k)) \quad (2.6)$$

$$\chi_w^x(k+1) = \sup_j \min(\chi_{w,j}^{f(u(k))}, \chi_j^x(k)) \quad (2.7)$$

das gleiche Ergebnis wie die Gleichungen (2.5) und (2.4).

Im Falle eines Fuzzy-Automaten bleiben die Einträge der Vektoren χ^x und χ^u nicht auf die Werte null und eins beschränkt, sondern es werden alle Werte aus dem Intervall $[0, 1]$

zugelassen. Eine Belegung der Vektoren mit Wahrheitswerten aus dem Intervall $[0, 1]$ stellt dann eine Zugehörigkeitsfunktion $\mu^{\tilde{x}}$ einer Fuzzy-Menge $\tilde{x} \in \tilde{X}$ dar, die auf dem endlichen Zustandsraum X definiert ist.

Auch die Transitionsfunktion, genauer deren Darstellung $(\chi_{w,q,j}^f)$, nimmt in einem Fuzzy-Automaten Werte aus dem Intervall $[0, 1]$ an und wird daher als Zugehörigkeitsfunktion $\mu^{\tilde{f}} : X \times U \times X \rightarrow [0, 1]$ einer Fuzzy-Menge $\tilde{f}$ dargestellt. Ein Eintrag $\mu_{w,q,j}^{\tilde{f}}$ beschreibt den Wahrheitswert einer bestimmten Transition. Dieser Wert beschränkt den Wahrheitswert $\mu_w^{\tilde{x}}$, den der neue Zustand $\tilde{x}(k+1)$ aufgrund der Wahrheitswerte $\mu_j^{\tilde{x}}$ des Zustandes $\tilde{x}(k)$ und $\mu_q^{\tilde{u}}$ der Fuzzy-Eingabe $\tilde{u}(k)$ annehmen kann. Üblicherweise verwendet man zur Ermittlung der Wahrheitswertverteilung $\mu^{\tilde{x}}(k+1)$ des neuen Zustandes $\tilde{x}(k+1)$ in Abhängigkeit der Transitionsbeschreibung $\mu^{\tilde{f}}$ und der Wahrheitswertverteilung $\mu^{\tilde{x}}(k)$ des alten Zustandes $\tilde{x}(k)$ die Gleichungen

$$\mu_{w,j}^{\tilde{f}(\tilde{u}(k))} = \sup_q \min(\mu_{w,q,j}^{\tilde{f}}, \mu_q^{\tilde{u}}(k)) \quad (2.8)$$

$$\mu_w^{\tilde{x}}(k+1) = \sup_j \min(\mu_{w,j}^{\tilde{f}(\tilde{u}(k))}, \mu_j^{\tilde{x}}(k)) \quad (2.9)$$

in Anlehnung an die Gleichungen (2.6) und (2.7). Dieses Vorgehen entspricht dem Erweiterungsprinzip von Zadeh [105]. Es sei angemerkt, dass sich der Supremumoperator und der Minimumoperator auch durch andere mathematische Operatoren wie zum Beispiel Summation und Produktbildung ersetzen lassen.

Der Rechenschritt nach den Gleichungen (2.8) und (2.9) wird durch folgende Beispielrechnung illustriert: Ein Fuzzy-Automat sei durch die Darstellung der Fuzzy-Menge $\tilde{f}$ von

$$\mu_{w,1,j}^{\tilde{f}} = \begin{pmatrix} 1 & 0.1 & 0.2 & 0 \\ 0 & 0.9 & 0.1 & 0.6 \\ 0.1 & 0 & 0.3 & 1 \\ 0.8 & 0.3 & 0.9 & 0 \end{pmatrix} \text{ und } \mu_{w,2,j}^{\tilde{f}} = \begin{pmatrix} 0.3 & 0.2 & 0 & 0 \\ 1 & 0.8 & 0.3 & 0.4 \\ 0.2 & 0 & 0.4 & 0.8 \\ 0.4 & 1 & 1 & 0 \end{pmatrix} \quad (2.10)$$

beschrieben. Bei einer Wahrheitswertverteilung $\mu^{\tilde{u}}(k) = (0.6 \ 1)^T$ einer Fuzzy-Menge $\tilde{u}(k)$ ergibt sich nach elementweiser Berechnung nach Gleichung (2.8) die folgende Darstellung der wirkenden Transitionsfunktion:

$$\mu_{w,j}^{\tilde{f}(\tilde{u}(k))} = \begin{pmatrix} 0.6 & 0.2 & 0.2 & 0 \\ 1 & 0.8 & 0.3 & 0.6 \\ 0.2 & 0 & 0.4 & 0.8 \\ 0.6 & 1 & 1 & 0 \end{pmatrix}. \quad (2.11)$$

Ein Fuzzy-Zustand $\tilde{x}(k)$ mit der Wahrheitswertverteilung $\mu^{\tilde{x}}(k) = (0 \ 0.2 \ 1 \ 0.7)^T$ wird dann nach Gleichung (2.9) in einen neuen Fuzzy-Zustand $\tilde{x}(k+1)$ mit der Wahrheitswertverteilung $\mu^{\tilde{x}}(k+1) = (0.2 \ 0.6 \ 0.7 \ 1)^T$ überführt.

Die Transitionen in einem Fuzzy-Automaten kann man auch als Auswertungen der folgenden Regeln interpretieren: "Wenn der Zustand gleich $\tilde{x}(k)$ ist und am Eingang $\tilde{u}(k)$ anliegt und der Übergang zum nächsten Zustand durch $\tilde{f}$ beschrieben wird, dann ergibt sich der neue Zustand $\tilde{x}(k+1)$." Allerdings steckt die gesamte Information dieser Regel schon in

der Nennung der Funktion $\tilde{f}$ und entsteht nicht wie bei der regelbasierten Fuzzy-Logik aus der Mathematisierung der Regel selbst.

Ein Fuzzy-Automat nutzt also das übliche Inferenzverfahren, um Fuzzy-Mengen, die auf einer endlichen Menge X definiert sind, wieder in Fuzzy-Mengen abzubilden. Eine Erweiterung der Menge X auf unendliche Mengen oder Teilbereiche des $\mathbb{R}^n$ sind zwar technisch denkbar, allerdings geht dabei der Grundidee der "Fuzzifizierung" eines endlichen Automaten verloren. Auch eine Ausweitung auf zeitkontinuierliche Systeme ist in diesem Rahmen nicht möglich.

Diese beiden Einschränkungen werden bei dem folgenden Ansatz der fuzzy-dynamischen Systeme aufgebrochen.

Fuzzy-dynamische Systeme

Endliche Automaten sind spezielle zeitdiskrete Systeme. Zeitdiskrete Systeme wiederum lassen sich als Spezialfall von allgemeinen dynamischen Systemen [48] behandeln. Genauso wie endliche Automaten eine Untergruppe allgemeiner dynamischer Systeme sind, so sind auch Fuzzy-Automaten eine Untergruppe von allgemeinen dynamischen Fuzzy-Systemen, die mit dem Begriff "dynamical fuzzy systems" [46] in der Literatur und hier mit *fuzzy-dynamische Systeme* bezeichnet werden. Der Ansatz der fuzzy-dynamischen Systeme deckt sowohl zeitdiskrete als auch zeitkontinuierliche Fuzzy-Systeme ab. Die nachfolgende Darstellung ist auf den Spezialfall von zeitdiskreten fuzzy-dynamischen Systeme umgeschrieben worden.

Die Zustandsmenge $\tilde{X}$ eines fuzzy-dynamischen Systems besteht aus Fuzzy-Mengen, die auf einem metrischen Raum X , z.B. einem reellen Vektorraum, definiert sind. Die Fuzzy-Mengen $\tilde{x} \in \tilde{X}$ werden durch ihre Zugehörigkeitsfunktion $\mu^{\tilde{x}}(x)$ mit $\mu^{\tilde{x}} : X \rightarrow [0, 1]$ festgelegt.

Die Zugehörigkeitsfunktionen sind oberhalbstetig, d.h. die Mengen $\{x \in X | \mu(x) \geq \alpha\}$, auch α -Schnitte genannt, sind für jeden Wert von $\alpha \in [0, 1]$ kompakte Mengen. Auch die *Einflussbreite*, d.h. die Menge $\{x \in X | \mu(x) > 0\}$, ist kompakt. Einfache auf $\mathbb{R}$ definierte Fuzzy-Mengen, die diese Bedingungen erfüllen, sind dreiecksförmige Funktionen, Trapezfunktionen und Singletons, wie sie in Abbildung 2.3 zu sehen sind. Fuzzy-Mengen in der Form einer Gauss'schen Glockenkurve, also $\mu^{\tilde{x}}(x) = e^{-ax^2}$, sind nicht zulässig, da sie überall von null verschieden sind und damit nach der üblichen Metrik in $\mathbb{R}$ keine kompakte Einflussbreite haben. Die Fuzzy-Mengen müssen nicht unbedingt konvex sein, da die α -Schnitte nicht zusammenhängend sein müssen. Die Fuzzy-Mengen können sich zum Beispiel aus mehreren dreiecksförmigen Funktionen zusammensetzen.

Die Funktion $\tilde{f}$, die einen Fuzzy-Zustand auf einen neuen Fuzzy-Zustand abbildet, kann zeitvariant sein. Man kann sich vorstellen, dass die explizite Zeitabhängigkeit durch die Wahl von Fuzzy-Mengen $\tilde{u}(k)$ am Eingang erzeugt wird. Diese Fuzzy-Mengen sind auch wieder durch ihre Zugehörigkeitsfunktion $\mu^{\tilde{u}(k)}(u)$ mit $\mu^{\tilde{u}(k)} : U \rightarrow [0, 1]$ festgelegt.

Die Funktion $\tilde{f}$ ist wie im Falle des Fuzzy-Automaten durch eine Abbildung $\tilde{f} : X \times U \times X \rightarrow X$ als Fuzzy-Funktion höherer Ordnung definiert. Ihre Auswertung erfolgt analog zu

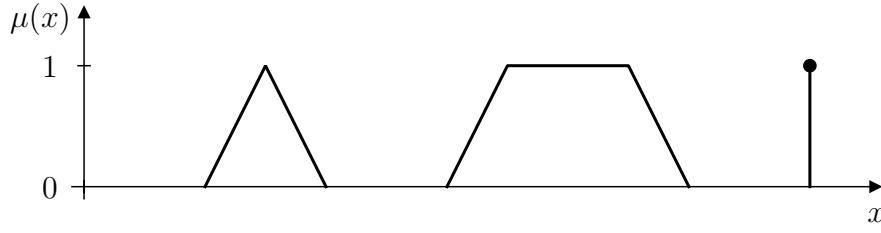


Abbildung 2.3: Einfache Zugehörigkeitsfunktionen in $\mathbb{R}$: Dreiecksfunktion, Trapezfunktion und Singleton.

den Gleichungen (2.8) und (2.9) durch

$$\mu^{\tilde{f}(\tilde{u}(k))}(x(k), x(k+1)) = \sup_{u(k) \in U} \min(\mu^{\tilde{f}}(x(k), u(k), x(k+1)), \mu^{\tilde{u}}(u(k))) \quad (2.12)$$

$$\mu^{\tilde{x}}(x(k+1)) = \sup_{x(k) \in X} \min(\mu^{\tilde{f}(\tilde{u}(k))}(x(k), x(k+1)), \mu^{\tilde{x}}(x(k))), \quad (2.13)$$

wobei die Variablen $x(k) \in X$, $u(k) \in U$ und $x(k+1) \in X$ die Aufgabe der Indizes j , q und w im Falle des Fuzzy-Automaten übernehmen.

Setzt man für $\tilde{x}(k)$ und $\tilde{u}(k)$ Singletons $\tilde{x}(k) = \chi(x_0)$ und $\tilde{u}(k) = \chi(u_0)$ ein, so kann man die Fuzzy-Funktion $\tilde{f}(\chi(x_0), \chi(u_0), x(k+1))$ als Schar der sich ergebenden Zugehörigkeitsfunktion $\mu^{\tilde{x}}(x(k+1))$ mit den Scharparametern x_0 und u_0 auffassen.

Wie im Falle der Fuzzy-Automaten lassen sich die Gleichungen (2.12) und (2.13) durch folgende Regel beschreiben: "Wenn der Zustand gleich $\tilde{x}(k)$ ist und am Eingang $\tilde{u}(k)$ anliegt und der Übergang zum nächsten Zustand durch $\tilde{f}$ beschrieben wird, dann ergibt sich der neue Zustand $\tilde{x}(k+1)$." Wiederum steckt dabei die gesamte Information schon in der Festlegung der Funktion $\tilde{f}$ und nicht in der Mathematisierung der Regel selbst.

Spezialfälle von fuzzy-dynamischen Systemen tauchen auch unter anderen Bezeichnungen in der Fachliteratur auf. Unter der Bezeichnung "iterated fuzzy sets" werden zeitdiskrete fuzzy-dynamische Systeme, die von keiner Eingangsgröße abhängen, auf chaotische Dynamik hin untersucht [45, 21]. Ausgehend davon können Fraktale erzeugt und analysiert werden [19]. Für eine ähnliche Gattung, "dynamic fuzzy systems" genannt, gibt es Konvergenzkriterien [51]. Weiterhin dient der Ansatz zur Darstellung von mehrstufigen Fuzzy Entscheidungsprozessen [44] und zum Auffinden eines Fuzzy-Stoppzeitpunktes in solchen Entscheidungsprozessen [52].

Fuzzy-dynamische Systeme können also verwendet werden, um ein zeitdiskretes dynamisches System auf einer Menge im Raum $\tilde{X}$ der Fuzzy-Mengen darzustellen. Den Fuzzy-Mengen im Raum $\tilde{X}$ können dabei sowohl endliche Mengen X , als auch Teilräume des $\mathbb{R}^n$ oder ganz allgemein metrische Räume zu Grunde liegen. Das System nutzt das übliche Inferenzverfahren, um die Fuzzy-Mengen der neuen Zustände zu berechnen. Durch eine allgemeinere Fassung [46] lassen sich damit auch zeitkontinuierliche dynamische Systeme definieren. Die Transitionsfunktion $\tilde{f}$ ist nicht in erster Linie gedacht, um sprachliche Regeln aus einer Regelbasis auszuwerten.

Dieser letzte Punkt ist für den folgenden Ansatz der dynamischen Fuzzy-Systeme von zentraler Bedeutung.

Dynamische Fuzzy-Systeme

”Dynamische Fuzzy-Systeme” bzw. ”Dynamic fuzzy systems” nach [81, 82] beschreiben auch ein dynamisches System durch eine Abbildung $\tilde{f}$ von Fuzzy-Mengen $\tilde{x}(k)$ und $\tilde{u}(k)$ auf Fuzzy-Mengen $\tilde{x}(k+1)$, die jeweils auf reellen Vektorräumen definiert sind. Die Funktion $\tilde{f}$ soll dabei durch eine geringe Zahl von linguistischen Regeln, die das dynamische System beschreiben, festgelegt sein. Es wird vorausgesetzt, dass die sprachlichen Größen in den Regeln durch einfach interpretierbare Fuzzy-Mengen beschrieben werden. Darunter versteht man in diesem Zusammenhang Fuzzy-Mengen, deren Zugehörigkeitsfunktionen an einer Stelle einen Maximalwert von eins annehmen und links und rechts von dieser Stelle je monoton gegen null gehen. Die Stelle des Maximalwertes ist ein eindeutiger Zahlenwert, sozusagen ein Prototyp der Fuzzy-Menge. Die Unsicherheit, die in diesem Modell dem Zahlenwert zugeordnet ist, wird durch die Flanken der Fuzzy-Menge beschrieben.

Das eingesetzte Inferenzverfahren soll im Wesentlichen zwei Bedingungen genügen: Erstens sollen interpretierbare Mengen wieder auf interpretierbare Mengen abgebildet werden und zweitens soll das Inferenzverfahren die Regelbasis umsetzen. Um die zweite Bedingung zu veranschaulichen wird die Regelbasis mit den folgenden zwei Regeln betrachtet: ”Wenn $\tilde{x}(k) = \tilde{x}_A$, dann ist $\tilde{x}(k+1) = \tilde{f}(\tilde{x}_A)$ ” und ”Wenn $\tilde{x}(k) = \tilde{x}_B$, dann ist $\tilde{x}(k+1) = \tilde{f}(\tilde{x}_B)$.” Beispiele für Zugehörigkeitsfunktionen für $\tilde{x}_A$, $\tilde{x}_B$, $\tilde{f}(\tilde{x}_A)$ und $\tilde{f}(\tilde{x}_B)$ sind in Abbildung 2.4 gezeigt.

Liegt eine Fuzzy-Menge am Eingang an, die der Prämisse einer Regel entspricht, also zum Beispiel $\mu^{\tilde{x}_A}$, so soll sich genau die Fuzzy-Menge der Konklusion der Regel, also $\mu^{\tilde{f}(\tilde{x}_A)}$, aus dem Inferenzverfahren ergeben. Diese Forderung wurde schon durch die Bezeichnung $\tilde{x}(k+1) = \tilde{f}(\tilde{x}_A)$ in der Regelbasis vorweggenommen. Liegt eine Fuzzy-Menge am Eingang näher an der Prämisse einer Regel als eine andere Eingangsgröße, so soll auch das Ergebnis des Inferenzverfahrens für diesen Eingangswert näher an der Schlussfolgerung der Regel als für den anderen Eingangswert liegen. In Abbildung 2.4 ist der Abstand von $\tilde{x}_{IP}$ zu $\tilde{x}_A$ kleiner als der Abstand von $\tilde{x}_B$ zu $\tilde{x}_A$, wenn man dem Abstand durch die Distanz der Maximalwerte der Zugehörigkeitsfunktionen gewinnt. Daher ist auch der Abstand von $\tilde{f}(\tilde{x}_{IP})$ zu $\tilde{f}(\tilde{x}_A)$ kleiner als der Abstand von $\tilde{f}(\tilde{x}_B)$ zu $\tilde{f}(\tilde{x}_A)$. Zusatzinformation, die nicht in der Regelbasis berücksichtigt wird, geht bei der Inferenz verloren. Wenn zum Beispiel eine Fuzzy-Menge $\tilde{x}_{A'}$ eine Teilmenge einer Fuzzy-Menge $\tilde{x}_A$ ist, die in der Prämisse einer Regel steht, so liegt Zusatzinformation vor, da die Größe am Eingang genauer bekannt ist. Trotzdem soll das Inferenzverfahren dann für beide Fälle genau das gleiche Ergebnis $\tilde{f}(\tilde{x}_A)$ liefern, da die Zusatzinformation aufgrund der unscharfen Formulierung der Regel nicht berücksichtigt werden kann. Dies ist auch schematisch in Abbildung 2.4 zu sehen.

Um diese und weitere Bedingungen erfüllen zu können, kann kein Standardinferenzverfahren genutzt werden [82]. Daher wird ein Inferenzverfahren mittels ”interpolierender Regeln” verwendet. Liegt am Eingang eine Fuzzy-Menge $\tilde{x}$ an, so werden die nächstliegenden Fuzzy-Mengen $\tilde{x}_A$ und $\tilde{x}_B$ von linguistischen Werten aus der Regelbasis so gewichtet, dass durch die Interpolation eine Fuzzy-Menge $\tilde{x}_{IP}$ entsteht, deren Maximalstelle mit der von $\tilde{x}$

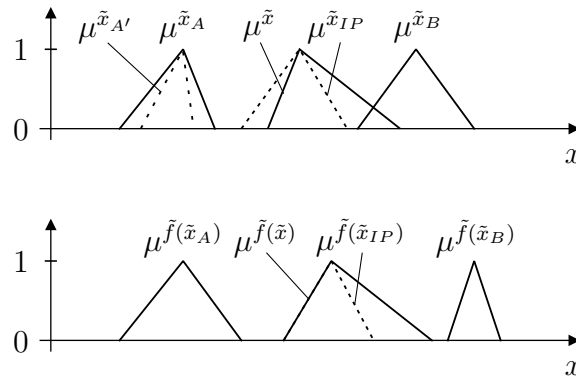


Abbildung 2.4: Zugehörigkeitsfunktionen vor und nach einer geeigneten Abbildung $\tilde{f}$ in einem einfachen dynamischen Fuzzy-System.

übereinstimmt. Die Gewichtungsfaktoren für diese Interpolation werden auch genutzt, um eine Interpolation der Fuzzy-Mengen $\tilde{f}(\tilde{x}_A)$ und $\tilde{f}(\tilde{x}_B)$ in den Konklusionen der beteiligten Regeln durchzuführen. So erhält man eine interpolierte Konklusion $\tilde{f}(\tilde{x}_{IP})$. Abbildung 2.4 zeigt ein Beispiel. Durch die interpolierte Prämisse und interpolierte Konklusion ist die interpolierende Regel festgelegt. Damit ist auch schon die Maximalstelle der Fuzzy-Menge $\tilde{f}(\tilde{x})$ bestimmt, die sich aus dem Inferenzmechanismus ergibt.

In einem weiteren Schritt wird die Unschärfe der Fuzzy-Menge $\tilde{x}$ am Eingang mit der Unschärfe der Prämisse der interpolierenden Regel $\tilde{x}_{IP}$ verglichen. Ist die Unschärfe der Fuzzy-Menge am Eingang größer, so muss die Unschärfe der Konklusion der interpolierenden Regel noch geeignet erhöht werden. Dies geschieht durch ein analoges Interpolationsverfahren. Im Beispiel aus Abbildung 2.4 ist die Fuzzy-Menge $\tilde{x}$ nur an der rechten Flanke unschärfer als $\tilde{x}_{IP}$. Daher wird bei der vorliegenden Regelbasis auch nur die eine Flanke der Menge $\tilde{f}(\tilde{x})$ entsprechend unschärfer als die von $\tilde{f}(\tilde{x}_{IP})$.

Nach der Ermittlung der interpolierenden Regel und der Modifikation ihrer Konklusion $\tilde{f}(\tilde{x}_{IP})$, um der vorhandenen Information gerecht zu werden, liegt die Zugehörigkeitsfunktion $\mu^{\tilde{f}(\tilde{x})}$ des Bildes $\tilde{f}(\tilde{x})$ von $\tilde{x}$ vor.

In [82] wird gezeigt, wie man ausgehend von Messdaten einen dynamischen Prozess mit einem dynamischen Fuzzy-System modellieren kann. Weiter wird behandelt, wie man die Stabilität eines solchen Modells computergestützt nachweisen kann und wie sich Regler für solche Systeme entwerfen lassen. Den praktischen Einsatz dynamischer Fuzzy-Systeme zeigt die Modellierung von Hochtemperatur-Brennstoffzellen [83].

Die "dynamischen Fuzzy-Systeme" nutzen also einen praxisorientierten Ansatz, um aus sprachlichen Regeln zeitdiskrete Fuzzy-Systeme zu erhalten, welches Fuzzy-Mengen auf Fuzzy-Mengen abbilden. Die betrachteten Fuzzy-Mengen sind alle auf reellen Vektorräumen $X = \mathbb{R}^n$ definiert. Eine Erweiterung auf endliche Mengen X ist nicht ohne weiteres möglich. Die Regeln beschreiben den Wechsel eines Zustandes $\tilde{x}(k)$ auf einen neuen Zustand $\tilde{x}(k+1)$ im nächsten Zeitschritt. Da das Verfahren versucht, die Regel eins zu eins in ein Fuzzy-System umzusetzen, macht ihr Einsatz in einem zeitkontinuierlichen Fuzzy-System keinen Sinn. Der Vorteil dieses Ansatzes für ein zeitdiskretes Fuzzy-System ist,

dass es auf transparente Weise Regeln mittels Fuzzy-Logik mathematisch modelliert. Die enge Bindung an die sprachlich formulierten Regeln erfordert, ein sonst unübliches Inferenzverfahren einzusetzen.

Es ist jedoch möglich, Standardverfahren zu nutzen, um sprachliche Regeln eins zu eins in ein zeitdiskretes Fuzzy-System umzusetzen, wie der nachfolgende Ansatz der rekurrenten Fuzzy-Systeme zeigt.

Rekurrente Fuzzy-Systeme

Im Gegensatz zu den vorangegangenen Ansätzen für zeitdiskrete Fuzzy-Systeme beschreiben rekurrente Fuzzy-Systeme nicht eine Abbildung von Fuzzy-Mengen auf Fuzzy-Mengen, sondern von Zahlenwerten auf Zahlenwerte. Dies geschieht, indem sprachliche Regeln mit Hilfe der regelbasierten Fuzzy-Logik ausgewertet werden. Rekurrente Fuzzy-Systeme setzen für die Transitionsfunktion f in einem zeitdiskreten System ein vollständiges Fuzzy-System ein. Dieses besteht aus den drei Teilschritten der Fuzzifizierung, Inferenz und Defuzzifizierung.

Interessanterweise wurde der Ansatz für rekurrente Fuzzy-Systeme unabhängig voneinander von verschiedenen Personen entwickelt. Dabei lagen zwei ganz verschiedene Denkansätze zugrunde.

Die erste Idee [29] resultiert aus der Möglichkeit, bestimmte regelbasierte Fuzzy-Systeme als neuronale Netze, so genannte *Neuro-Fuzzy-Systeme* [65, 33], darstellen zu können. Werden statische neuronale Netze mit einer zeitversetzten Rückführung ihrer Ausgangsgrößen ausgestattet, so nennt man sie auch *rekurrente neuronale Netze* [33]. Dementsprechend macht es Sinn, auch Neuro-Fuzzy-Systeme mit einer solchen Rückführung auszustatten und man erhält rekurrente Neuro-Fuzzy-Systeme oder kurz: rekurrente Fuzzy-Systeme. Ihre Darstellung als rekurrente neuronale Netze erlaubt es, Lernverfahren anzuwenden, die die Genauigkeit des Fuzzy-Modells datengetrieben verbessern kann [29, 40]. Die Tatsache, dass sie zugleich Neuro-Fuzzy-Systeme sind, erlaubt es, eine gute Anfangsinitialisierung für das Lernverfahren anzugeben, da man Vorwissen einbringen kann, welches in Form von sprachlichen Regeln formulierbar ist. Umkehrt lässt sich das Ergebnis des Lernverfahrens auch wieder sprachlich deuten und so auf Plausibilität testen.

Die zweite Idee [1, 2] stellt die Bedeutung der Regelbasis in den Mittelpunkt. Die Regelbasis beschreibt sprachlich ein dynamisches System mit Hilfe von endlich vielen sprachlichen Werten. Würde man diese sprachlichen Werte als voneinander unabhängige Werte auffassen und die Regeln durch klassische Logik auswerten [4], so erhielte man einen endlichen Automaten. Wie im Falle der dynamischen Fuzzy-Systeme sind die sprachlichen Werte aber nur als Beschreibung markanter Zahlenwerte zu verstehen. Durch die regelbasierte Fuzzy-Logik ist es möglich, auch Zahlenwerte zwischen den markanten Werten zu verarbeiten, indem die Regeln, die für die markanten Werte gelten, je nach ihrer Relevanz gewichtet werden. Durch die Defuzzifizierung wird schließlich ein einzelner Zahlenwert gewonnen. Im Gegensatz zu Fuzzy-Automaten, die eine Wahrheitswertverteilung über alle klassischen Zustandswerte berechnen und damit alle Zustandswerte als mehr oder wenig mögliche Ergebnisse ansehen, wird in rekurrenten Fuzzy-Systemen eine eindeutige Entscheidung am Ende der Auswertung der Regeln getroffen.

Rekurrente Fuzzy-Systeme schreiben also sprachliche Regeln mit Hilfe der regelbasierten Fuzzy-Logik in eine mathematische Funktion um. Die sprachliche Interpretation einerseits und die einfache Handhabung andererseits eröffnen rekurrenten Fuzzy-Systemen ein weites Anwendungsfeld [2, 6, 3]. Sie werden auch industriell genutzt [1, 5]. Da die Regeln ein dynamisches System beschreiben, ergibt sich eine einfache Differenzengleichung. Diese kann mit anderen mathematischen Methoden, zum Beispiel mit Lernverfahren, bearbeitet und untersucht werden [4, 41, 42]. Rekurrente Fuzzy-Systeme sind für die Beschreibung von zeitdiskreten Systemen mit einem reellen Vektorraum als Zustandsraum konzipiert. Zur Modellierung von dynamischen Systemen mit einem endlichen Zustandsraum sind sie ohne Modifikation nicht geeignet. Der Einsatz einer Funktion $\mathbf{f}$ in einem zeitkontinuierlichen System ist auch denkbar. Allerdings geht dabei die Bedeutung der Regeln verloren, die die Änderung des Zustandswertes im Laufe eines endlichen Zeitintervalls beschreiben.

Die vier vorgestellten zeitdiskreten Fuzzy-Systeme werden in folgendem Abschnitt miteinander verglichen.

2.3 Vergleich zeitdiskreter Fuzzy-Systeme

In diesem Abschnitt sollen die oben vorgestellten Ansätze von zeitdiskreten Systemen miteinander verglichen werden. Dies soll die Entscheidung für den Einsatz eines dieser Systeme erleichtern. Neben der Möglichkeit, mittels Fuzzy-Methoden eine Funktion $\mathbf{f}$ in einem zeitdiskreten System zu erhalten, darf jedoch nicht außer Acht gelassen werden, dass es eine Reihe von anderen Methoden gibt, Fuzzy-Logik in zeitdiskreten Systemen einzusetzen.

Beispielhaft für weitere Methoden wird hier das erfolgreich eingesetzte Verfahren von Takagi, Sugeno und Kang [89, 88] kurz behandelt. Es gewichtet klassische Differenzengleichungen oder Differentialgleichungen, indem es Regeln auswertet, die die folgende Form haben: "Wenn der Eingangsgrößenvektor $\tilde{\mathbf{u}}_i$ ist, dann nutze die Differenzengleichung $\mathbf{x}(k+1) = \mathbf{f}_i(\mathbf{x}(k), \mathbf{u}(k))$ und $\mathbf{y}(k) = \mathbf{g}_i(\mathbf{x}(k), \mathbf{u}(k))$."

Durch die Fuzzifizierung des Eingangsvektors $\mathbf{u}(k)$ mit Zugehörigkeitsfunktionen für die verschiedenen möglichen Fuzzy-Mengen $\tilde{\mathbf{u}}_i$ wird der Wahrheitswert bestimmt, mit dem die entsprechende Regel gilt. Die Wahrheitswerte bestimmen dann die Gewichtungsfaktoren, mit denen die Differenzengleichungen in der Konklusion der einzelnen Regeln gewichtet werden. Das dynamische Verhalten des Systems ist also nicht sprachlich in den Regeln festgelegt. Vielmehr beschreiben die Differenzengleichungen das Verhalten. Die sprachlichen Regeln beschreiben nur, wie stark welches der verschiedenen mathematischen Modelle Einfluss auf das Gesamtverhalten hat.

Je nach Zielsetzung eignen sich manche Ansätze besser für eine spezielle Problemlösung als andere. Nachfolgend werden einige relevante Eigenschaften der verschiedenen Ansätze diskutiert. Die Ergebnisse sind in der Tabelle 2.1 zusammengefasst.

Will man am Ende der Modellierung eine gewöhnliche Differenzengleichung erhalten, so muss man einen Ansatz wählen, der den Zustand durch einen Zahlenwert oder Vektor beschreibt und den Teilschritt einer Defuzzifizierung enthält, also rekurrente Fuzzy-Systeme oder Takagi-Sugeno-Kang Modelle. Alle anderen Ansätze beschreiben den Zustand der

Tabelle 2.1: Überblick über die Eigenschaften der verschiedenen zeitdiskreten dynamischen Systeme: fuzzy-dynamische Systeme (FDS), Fuzzy-Automaten (FA), dynamische Fuzzy-Systeme (DFS), rekurrente Fuzzy-Systeme (RFS) und Takagi-Sugeno-Kang Modelle (TSK) mit den Bewertungen 0 für nicht zutreffend, 0.5 für nur bedingt zutreffend und 1 für voll zutreffend.

	FDS	FA	DFS	RFS	TSK
Dynamik in Regelbasis	0.5	0.5	1	1	0
dynamische Größe	$\mu(x)$	$\mu(x)$	$\mu(x)$	x	x
Defuzzifizierung	0	0	0	1	1
X : endliche Menge	1	1	0	0	0
$X = \mathbb{R}^n$	1	0	1	1	1
Standard-Inferenz	1	1	0	1	1
zeitkontinuierlich	1	0	0.5	0.5	1

dynamischen Systeme durch Fuzzy-Mengen.

Will man sprachliche Regeln, die einen zeitdiskreten dynamischen Vorgang beschreiben, eins zu eins in ein Fuzzy-Modell umsetzen, so bieten sich dynamische Fuzzy-Systeme oder rekurrente Fuzzy Systeme an. Fuzzy-dynamische Systeme und Fuzzy-Automaten sind nur bedingt dafür geeignet, da sie in erster Linie Funktionen fuzzifizieren und nicht regelbasiert arbeiten. Auch Takagi-Sugeno-Kang Modelle sind in diesem Fall nicht geeignet, da sie das dynamische System explizit durch Differenzgleichungen und nicht durch sprachliche Regeln beschreiben.

Liegt dem Zustandsraum eine endliche Menge X zu Grunde, so können nur Fuzzy-Automaten oder fuzzy-dynamische Systeme eingesetzt werden. Aufgrund der höheren Anschaulichkeit ihres Ansatzes sind Fuzzy-Automaten normalerweise den fuzzy-dynamischen Systemen vorzuziehen. Alle anderen vorgestellten Methoden scheiden unter dieser Prämisse aus.

Umgekehrt eignen sich Fuzzy-Automaten nicht für den Einsatz bei zeitdiskreten dynamischen Systemen, deren Zustandsraum auf einem reellen Vektorraum X aufbaut. Alle anderen Ansätze können ohne Einschränkung damit umgehen.

Alle Ansätze, mit Ausnahme der dynamischen Fuzzy-Systeme, nutzen Standardverfahren für die Inferenz. Das Inferenzverfahren für dynamische Fuzzy-Systeme ist aufwändiger in der Realisierung als bei rekurrenten Fuzzy-Systemen, Fuzzy-Automaten und Takagi-Sugeno-Kang Modellen, da sowohl die Maximalwerte als auch die Fußpunkte der Zugehörigkeitsfunktion des neuen Zustandes bestimmt werden müssen.

Manche der Ansätze sind auch für den Einsatz in zeitkontinuierlichen Systemen geeignet. Am einfachsten ist dies bei Takagi-Sugeno-Kang Modellen möglich, da dort die Differenzgleichungen in den Regeln einfach durch Differentialgleichungen ausgetauscht werden können. Fuzzy-dynamische Systeme lassen sich auch in ihrer erweiterten Form für die Darstellung von zeitkontinuierlichen Fuzzy-Systemen verwenden. Dies geht aber mit der Nutzung von erheblich komplexeren mathematischen Methoden einher. Rekurrente Fuzzy-

Systeme und dynamische Fuzzy-Systeme sind für zeitkontinuierliche Systeme nur bedingt einsetzbar, weil dabei ihre enge Beziehung zur Regelbasis, die ein zeitdiskretes Verhalten beschreibt, verloren geht. Fuzzy-Automaten können als "fuzzifizierte Automaten" nicht für zeitkontinuierliche Systeme verwendet werden.

Nach diesem Überblick über zeitdiskrete Fuzzy-Systeme werden im nächsten Kapitel rekurrente Fuzzy-Systeme formal definiert. Mit dieser und mit weiteren Definitionen kann anschließend die tiefergehende Untersuchung der Systemklasse der rekurrenten Fuzzy-Systeme beginnen.

3 Definitionen

In diesem Kapitel werden rekurrente Fuzzy-Systeme formal definiert. Zur Veranschaulichung der Definitionen wird ein Beispielsystem betrachtet, auf das auch in den nächsten Kapiteln immer wieder zurückgegriffen wird.

Das Beispielsystem modelliert die zeitliche Entwicklung der Größe einer Insektenpopulation [101, 78]. Die Populationsgröße hängt maßgeblich von der zur Verfügung stehenden Nahrungsmenge und der Populationsgröße im vorangegangenen Jahr ab. Die Larven schlüpfen im Frühjahr und entwickeln sich im Laufe des Sommers zu zeugungsfähigen Insekten. Wenn sie bis zum Herbst überlebt haben, legen sie ihre Eier und sterben ab. Die einzelnen Generationen folgen aufeinander, ohne dass sie sich zeitlich überschneiden. Es findet also keine Vermischung der Generationen statt.

Das betrachtete Zeitintervall für das zeitdiskrete Modell beträgt ein Jahr. Die Populationsgröße stellt die Zustandsgröße x und die Nahrungsmenge die Eingangsgröße u dar. Die momentane Populationsgröße $x(k)$ wird durch eine der folgenden drei linguistischen Werte charakterisiert: $L_1^x = \text{klein}$, $L_2^x = \text{mittel}$, oder $L_3^x = \text{groß}$. Das Nahrungsangebot $u(k)$ kann einen der drei linguistischen Werte $L_1^u = \text{gering}$, $L_2^u = \text{normal}$, oder $L_3^u = \text{üppig}$ annehmen.

Die zeitliche Entwicklung der Populationsgröße kann durch wenige einfache Regeln beschrieben werden: Wenn das Nahrungsangebot gering ist, dann wird die Population klein bleiben oder klein werden. Wenn das Nahrungsangebot normal bzw. üppig ist, so wird eine kleine Population auf eine mittlere anwachsen und eine mittlere Population auf eine mittlere bzw. große Population. Eine große Population wird dagegen auf eine kleine bzw. mittlere Population schrumpfen. Dies ist auf intraspezifische Konkurrenz [10] zurückzuführen: Die zur Verfügung stehenden Ressourcen werden durch die große Population zu schnell und zu massiv reduziert, so dass nur wenige Insekten bis zum Eierlegen überleben.

Ziel der Modellierung der Populationsgröße durch ein rekurrentes Fuzzy-System ist, das sprachliche Modell, welches die zeitliche Entwicklung der Populationsgröße beschreibt, in ein mathematisches Modell zu überführen. Dazu wird der folgende allgemeine Ansatz für rekurrente Fuzzy-Systeme gewählt.

3.1 Definition rekurrenter Fuzzy-Systeme

Ein rekurrentes Fuzzy-System ist ein zeitdiskretes dynamisches System, wie es in Abbildung 2.2 im vorangegangenen Kapitel zu sehen ist. Es wird beschrieben durch die Gleichung

chungen

$$\mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k)) \quad (3.1)$$

$$\mathbf{y}(k) = \mathbf{g}(\mathbf{x}(k), \mathbf{u}(k)). \quad (3.2)$$

Sein Zustandsvektor $\mathbf{x} \in X$, Eingabevektor $\mathbf{u} \in U$ und Ausgabevektor $\mathbf{y} \in Y$ sind reellwertige Vektoren. Die Funktionen $\mathbf{f}$ und $\mathbf{g}$ stellen vollständige Fuzzy-Systeme dar, bestehend aus den Teilschritten Fuzzifizierung, Inferenz und Defuzzifizierung.

Im Weiteren wird nur die Funktion $\mathbf{f}$ behandelt. Die Ausgabefunktion $\mathbf{g}$ hat selbst keine Auswirkung auf die Dynamik des Systems und spielt daher für die späteren Untersuchungen nur eine untergeordnete Rolle. Sie ist analog zur Funktion $\mathbf{f}$ aufgebaut.

Für die Beschreibung der Funktion $\mathbf{f}$ werden zuerst die nötigen mathematischen Größen und die Regelbasis eingeführt und anschließend die Teilschritte Fuzzifizierung, Inferenz und Defuzzifizierung behandelt.

Für jede Komponente x_i des Zustandsvektors $\mathbf{x}$ soll es nur endlich viele linguistische Werte $L_{j_i}^{x_i}$ mit $j_i \in \{1, 2, \dots\}$ geben. Gleichmaßen soll es auch linguistische Werte $L_{q_p}^{u_p}$ für jede Komponente u_p des Eingabevektors $\mathbf{u}$ geben.

Im Beispiel der Insektenpopulation ist die Populationsgröße die einzige Zustandsgröße x , so dass der Index i wegfallen kann. Sie wird durch die drei linguistischen Werte $L_1^x = \text{klein}$, $L_2^x = \text{mittel}$ und $L_3^x = \text{groß}$ beschrieben. Die einzige Eingangsgröße u wird in diesem Beispiel durch die drei linguistischen Werte $L_1^u = \text{gering}$, $L_2^u = \text{normal}$ und $L_3^u = \text{üppig}$ charakterisiert.

Der Kern der Funktion $\mathbf{f}$ ist die Regelbasis, die die linguistischen Werte der Zustandsgrößen und der Eingangsgrößen auf die linguistischen Werte der neuen Zustandsgrößen abbildet. Sie soll ausschließlich Regeln enthalten, die in die folgende Form gebracht werden können:

$$\begin{array}{llll} \text{Wenn} & x_1(k) = L_{j_1}^{x_1} & \text{und ... und} & x_n(k) = L_{j_n}^{x_n} \\ \text{und} & u_1(k) = L_{q_1}^{u_1} & \text{und ... und} & u_m(k) = L_{q_m}^{u_m}, \\ \text{dann} & x_1(k+1) = L_{w_1}^{x_1} & \text{und ... und} & x_n(k+1) = L_{w_n}^{x_n}. \end{array} \quad (3.3)$$

Diese Regelbasis soll widerspruchsfrei und vollständig sein, d.h. für jede Kombination von linguistischen Werten $L_{j_i}^{x_i}$ und $L_{q_p}^{u_p}$ soll es genau eine Regel für jede Komponente $L_{w_l}^{x_l}$ des neuen Zustandsvektors geben. Des Weiteren sollen in den Schlussfolgerungen der Regeln nur solche linguistischen Werte $L_{w_l}^{x_l}$ erlaubt sein, die auch als linguistische Werte $L_{j_i}^{x_i}$ in der Voraussetzung von bestimmten Regeln in der Regelbasis auftauchen.

Für das Beispiel der Insektenpopulation lassen sich die Bedingungen an die Regelbasis einfach überprüfen, wenn man die Regeln in Form einer Matrix zusammenfasst, wie sie in Abbildung 3.1 zu sehen ist. Die Einträge der Matrix sind die neuen Zustandswerte L_w^x , die die Populationsgröße $x(k+1)$ im Jahr $k+1$ beschreiben.

Liegen mehrere Zustandsgrößen oder mehrere Eingangsgrößen vor, so ist es für eine kompaktere Darstellung der Regelbasis hilfreich, *linguistische Vektoren* $\mathbf{L}_j^x$, $\mathbf{L}_q^u$ und $\mathbf{L}_w^x$ einzuführen, wobei ein solcher linguistischer Vektor eine "und"-Verknüpfung der linguistischen Werte in seinen Komponenten darstellt. So ist z.B. der Ausdruck $\mathbf{x} = \mathbf{L}_j^x = (L_{j_1}^{x_1}, L_{j_2}^{x_2}, L_{j_3}^{x_3})$ eine Kurzschreibweise für " $x_1 = L_{j_1}^{x_1}$ und $x_2 = L_{j_2}^{x_2}$ und $x_3 = L_{j_3}^{x_3}$ ". Mit Hilfe dieser

		aktuelle Populationsgröße L_j^x		
		klein	mittel	groß
Nahrungs- angebot L_q^u	gering	klein	klein	klein
	normal	mittel	mittel	klein
	üppig	mittel	groß	mittel

Abbildung 3.1: Regelbasis eines einfachen Modells der Populationsgröße einer Insektenpopulation in Matrixform.

Schreibweise lassen sich Regeln mit gleicher Prämisse gruppieren und in der folgenden Form darstellen:

$$\text{Wenn } \mathbf{x}(k) = \mathbf{L}_{\mathbf{j}}^{\mathbf{x}} \text{ und } \mathbf{u}(k) = \mathbf{L}_{\mathbf{q}}^{\mathbf{u}}, \text{ dann } \mathbf{x}(k+1) = \mathbf{L}_{\mathbf{w}}^{\mathbf{x}}. \quad (3.4)$$

Jede Regel beschreibt damit auch eine Abbildung der Indexvektoren $\mathbf{j} = (j_1, \dots, j_n)$ und $\mathbf{q} = (q_1, \dots, q_m)$, die die alten linguistischen Vektoren charakterisieren, auf einen Indexvektor $\mathbf{w} = (w_1, \dots, w_n)$, der den neuen linguistischen Zustandsvektor festlegt. Dieser Zusammenhang wird im Weiteren manchmal durch den Ausdruck $\mathbf{L}_{\mathbf{w}(\mathbf{j}, \mathbf{q})}^{\mathbf{x}}$ bzw. $L_{w_l(\mathbf{j}, \mathbf{q})}^{x_l}$ in der Schlussfolgerung von Regeln deutlich gemacht.

Für die Fuzzifizierung der Zustandswerte x_i von $\mathbf{x}$ werden Zugehörigkeitsfunktionen $\mu_{j_i}^{x_i}(x_i)$ für jeden linguistischen Wert $L_{j_i}^{x_i}$ auf Intervallen X_i festgelegt. Sie geben an, bis zu welchem Grad die Eigenschaft $L_{j_i}^{x_i}$ für den speziellen Zahlenwert x_i ausgeprägt ist. Zusätzlich wird je ein Zustandswert $s_{j_i}^{x_i}$ gewählt, an dem die entsprechende Zugehörigkeitsfunktion $\mu_{j_i}^{x_i}(x_i)$ ihren maximalen Wert annimmt und die im Weiteren als *Kernposition* bezeichnet wird. Diese Kernpositionen legen auch die Singletonpositionen fest, durch die die linguistischen Variablen $L_{w_l}^{x_l}$ in der Schlussfolgerung der Regeln beschrieben werden.

Die Zugehörigkeitsfunktionen $\mu_{j_i}^{x_i}(x_i)$ sollen die folgenden Bedingungen erfüllen:

- (B1) Beschränkung: $\mu_{j_i}^{x_i}(x_i) \in [0, 1]$ für alle $x_i \in X_i$.
- (B2) Konvexität: $\begin{cases} \mu_{j_i}^{x_i}(x_i) & \text{monoton steigend für } x_i < s_{j_i}^{x_i} \\ \mu_{j_i}^{x_i}(x_i) & \text{monoton fallend für } x_i > s_{j_i}^{x_i} \end{cases}$
- (B3) Partitionierung: $\sum_{j_i} \mu_{j_i}^{x_i}(x_i) > 0$ für alle $x_i \in X_i$
- (B4) Rückkopplungskorrespondenz: $\mu_{j_i}^{x_i}(s_{j_i}^{x_i}) > 0$ und $\mu_{j_i}^{x_i}(s_{l_i}^{x_i}) = 0$ für $j_i \neq l_i$.

Die Namensgebung von Bedingung (B4) resultiert aus der folgenden Eigenschaft: Hat nach einem Inferenzschritt ausschließlich ein linguistischer Wert $L_{j_i}^{x_i}$ einen positiven Wahrheitswert, so erhält man nach der Defuzzifizierung seiner Zugehörigkeitsfunktion genau den

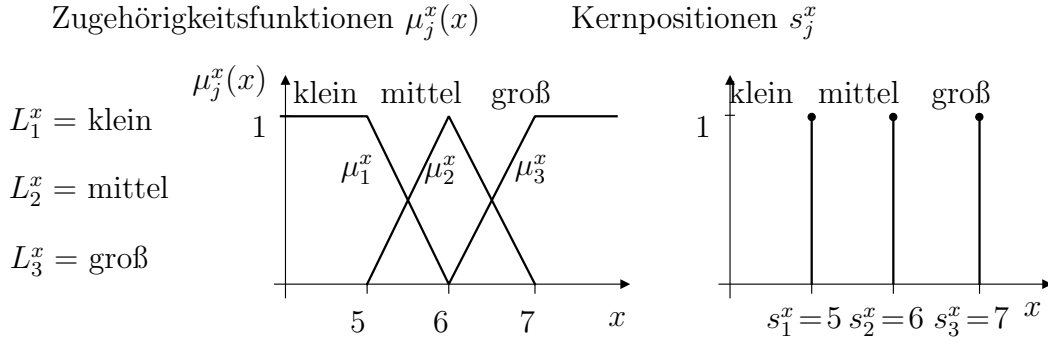


Abbildung 3.2: Beispiel von linguistischen Werten L_j^x , Zugehörigkeitsfunktionen $\mu_j^x(x)$ und Kernpositionen s_j^x für eine einzelne Zustandsvariable x . Da hier keine weiteren Zustandsvariablen betrachtet werden, ist ein zusätzlicher Index i in x_i , $L_{j_i}^{x_i}$, $\mu_{j_i}^{x_i}$ und $s_{j_i}^{x_i}$ überflüssig.

Wert der entsprechenden Singletonposition bzw. Kernposition $s_{j_i}^{x_i}$. Die Rückkopplungskorrespondenzbedingung (B4) sorgt nun dafür, dass bei der anschließenden Fuzzifizierung des Zahlenwertes $s_{j_i}^{x_i}$ nur die Zugehörigkeitsfunktion $\mu_{j_i}^{x_i}(x_i)$ des gleichen linguistischen Wertes $L_{j_i}^{x_i}$ einen Zugehörigkeitswert ungleich null erhält. So bleibt der linguistische Wert über die Teilschritte der Defuzzifizierung und der anschließenden Fuzzifizierung erhalten. Der neue linguistische Wert nach der Fuzzifizierung des Zahlenwertes korrespondiert damit mit dem alten, so rückgekoppelten linguistischen Wert vor der Defuzzifizierung.

Die gleichen Bedingungen, (B1) bis (B4), sollen auch von den Zugehörigkeitsfunktionen $\mu_{q_p}^{u_p}(u_p)$ und ihren Kernpositionen $s_{q_p}^{u_p}$ für jeden linguistischen Wert $L_{q_p}^{u_p}$ erfüllt sein.

Im Beispiel der Insektenpopulation wählt man zuerst Zahlenwerte s_j^x die eindeutig zu einem linguistischen Wert L_j^x der Populationsgröße x zugeordnet werden können – z.B. 10^5 für klein, 10^6 für mittel und 10^7 für groß oder auf einer logarithmischen Skala die Werte $s_1^x = 5$, $s_2^x = 6$ und $s_3^x = 7$. Anschließend kann man die Zugehörigkeitsfunktionen $\mu_1^x(x)$, $\mu_2^x(x)$ und $\mu_3^x(x)$ festlegen, beispielsweise wie in Abbildung 3.2 gezeigt. Dabei ist zu beachten, dass die Bedingungen (B1) bis (B4) eingehalten werden, was in dem betrachteten Beispiel der Fall ist. Analog sind Zugehörigkeitsfunktionen $\mu_1^u(u)$, $\mu_2^u(u)$ und $\mu_3^u(u)$ für die linguistischen Werte der Eingangswerte u zu wählen.

Die Bedingungen (B2) und (B4) zeichnen die Zustandswerte $x_i = s_{j_i}^{x_i}$ und Eingangswerte $u_p = s_{q_p}^{u_p}$, die auf Kernpositionen liegen, aus. Zustandsvektoren $\mathbf{x}$, Eingabevektoren $\mathbf{u}$ und die Kombinationen $(\mathbf{x}, \mathbf{u})$, die ausschließlich Kernpositionen $s_{j_i}^{x_i}$ und $s_{q_p}^{u_p}$ als Einträge in ihren Komponenten haben, sind deshalb auch spezielle Vektoren, die als *Kernpositionsvektoren* $\mathbf{s}_j^x \in X$, $\mathbf{s}_q^u \in U$ und $(\mathbf{s}_j^x, \mathbf{s}_q^u) \in X \times U$ bezeichnet werden. Diese Kernpositionsvektoren bilden eine Art Gitter im Zustandsraum X , Eingaberaum U bzw. im Raum $X \times U$, wie in Abbildung 3.3 angedeutet ist.

Für die Fuzzifizierung, Inferenz und Defuzzifizierung werden Standardverfahren verwendet. Im Inferenzschritt wird die algebraische Multiplikation sowohl als Aggregations- als auch als Implikationsoperator verwendet [11, 97]. Die Akkumulation wird durch die einfache Summenbildung durchgeführt [11, 97]. Zur Defuzzifizierung wird die Schwerpunktmethode

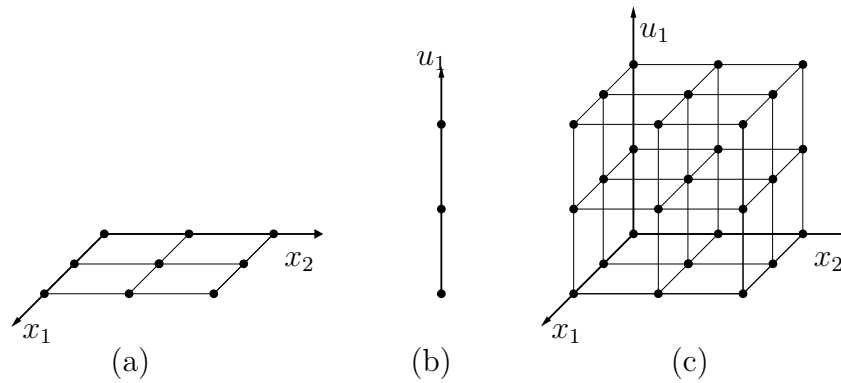


Abbildung 3.3: Die Kernpositionsvektoren bilden ein Gitter in den Vektorräumen (a) X , (b) U , und (c) $X \times U$.

(CoS) als Defuzzifizierungsmethode verwendet [11, 97].¹ Wie sich die Transitionsfunktion $\mathbf{f}$ durch die Fuzzifizierung, Inferenz und Defuzzifizierung ergibt, wird im nächsten Abschnitt detailliert betrachtet.

Damit sind alle Teile der Funktion $\mathbf{f}$ definiert. Die Funktion $\mathbf{g}$ ist analog aufgebaut. Zusammen legen die beiden Funktionen ein rekurrentes Fuzzy-System eindeutig fest.

3.2 Form der Transitionsfunktion

Die Transitionsfunktion $\mathbf{f}$ lässt sich auf eine kompakt Darstellungsform bringen, die sich dann leichter mathematisch untersuchen lässt. Diese Form erhält man durch die schrittweise Ausführung der Operationen in den einzelnen Teilschritten der Fuzzifizierung, Inferenz und Defuzzifizierung, wie sie im letzten Abschnitt festgelegt wurden.

Schritt 1 (Fuzzifizierung): In der Fuzzifizierung der Zustandsgrößen x_i und Eingangsgrößen u_p wird der Wahrheitsgrad der linguistischen Werte $L_{ji}^{x_i}$ und $L_{qp}^{u_p}$ durch entsprechende Zugehörigkeitswerte $\mu_{ji}^{x_i}(x_i)$ und $\mu_{qp}^{u_p}(u_p)$ bestimmt.

Eine logarithmierte Populationsgröße x von 6.2 wäre laut Auswertung der Zugehörigkeitsfunktionen aus Abbildung 3.2 nicht L_1^x =klein (Zugehörigkeitsgrad $\mu_1^x(x) = 0$), größtenteils L_2^x =mittel (Zugehörigkeitsgrad $\mu_2^x(x) = 0.8$) und ein wenig L_3^x =groß (Zugehörigkeitsgrad $\mu_3^x(x) = 0.2$). Entsprechend ist auch die Nahrungsmenge an geeigneten Zugehörigkeitsfunktionen μ_q^u auszuwerten.

Schritt 2 (Aggregation): Im ersten Inferenzschritt wird in der Aggregation die Und-Verknüpfung in der Voraussetzung jeder Regel mathematisch ausgewertet und dadurch der Wahrheitsgrad $\mu_{j,q}^{agg}(\mathbf{x}, \mathbf{u})$ für jede Regel bestimmt. Dazu werden die linguistischen

¹Eine Reihe von Sätzen in der Arbeit sind auch dann gültig, wenn statt der Multiplikation der Minimumsoperator für Aggregation und Implikation und statt der einfachen Summe der Maximumsoperator für die Akkumulation verwendet werden. Eine Aufzählung dieser Sätze sind in Anhang C zu finden. Eine Verallgemeinerung auf die Wahl beliebiger t-Normen und t-co-Normen als Operatoren ist dagegen in der Regel nicht möglich.

Werte $L_{j_i}^{x_i}$ und $L_{q_p}^{u_p}$, die in der Voraussetzung einer Regel stehen, herausgesucht und die entsprechenden Zugehörigkeitswerte $\mu_{j_i}^{x_i}(x_i)$ und $\mu_{q_p}^{u_p}(u_p)$ miteinander multipliziert, also

$$\mu_{\mathbf{j},\mathbf{q}}^{agg}(\mathbf{x}, \mathbf{u}) = \prod_i \mu_{j_i}^{x_i}(x_i) \prod_p \mu_{q_p}^{u_p}(u_p). \quad (3.5)$$

Zur Veranschaulichung der Aggregation wird aus dem Beispiel der Insektenpopulation die Regel "Wenn $x(k)$ =mittel und $u(k)$ =üppig, dann ist $x(k+1)$ =groß." betrachtet. Wenn die aktuelle Nahrungsmenge u zum Beispiel definitiv als üppig zu bewerten ist, d.h. $\mu_3^u(u) = 1$ ist, und eine aktuelle Populationsgröße $x = 6.2$ ist und damit $\mu_2^x(x) = 0.8$ ist, so ergibt sich für die betrachtete Regel der Wahrheitsgrad $\mu_{2,3}^{agg}(x, u) = \mu_2^x(x) \cdot \mu_3^u(u) = 0.8$.

Schritt 3 (Implikation): Im nächsten Inferenzschritt, der Implikation, werden die Zugehörigkeitsfunktionen, die die Folgerungen der Regeln beschreiben, mit dem Wahrheitswert der Regel gewichtet. Dies entspricht der Auswertung der Schlussfolgerungen der Regeln, da der Wahrheitswert der Voraussetzung den Wahrheitswert der Schlussfolgerung bestimmt. Die linguistischen Vektoren $L_{\mathbf{w}(\mathbf{j},\mathbf{q})}^{\mathbf{x}}$ in den Folgerungen der Regeln werden durch Singletons $\mu_{\mathbf{w}(\mathbf{j},\mathbf{q})}^{Singleton}(\mathbf{y})$ beschrieben. Ein Singleton ist eine Zugehörigkeitsfunktion die fast überall null ist. Nur an der Stelle des Singletonpositionsvektors $\mathbf{y} = \mathbf{s}_{\mathbf{w}(\mathbf{j},\mathbf{q})}^{\mathbf{x}}$ nimmt sie den Wert eins an. Damit man erhält:

$$\mu_{\mathbf{w}(\mathbf{j},\mathbf{q})}^{imp}(\mathbf{x}, \mathbf{u}, \mathbf{y}) = \mu_{\mathbf{j},\mathbf{q}}^{agg}(\mathbf{x}, \mathbf{u}) \cdot \mu_{\mathbf{w}(\mathbf{j},\mathbf{q})}^{Singleton}(\mathbf{y}). \quad (3.6)$$

Schritt 4 (Akkumulation): Im letzten Inferenzschritt, der Akkumulation, werden die Folgerungen aller Regeln durch eine fuzzy-logische "oder"-Verknüpfung zusammengefasst. Dies geschieht, indem die gewichteten Singletons aus dem Implikationsschritt zu einer neuen Zugehörigkeitsfunktion addiert werden. So erhält man

$$\mu^{akk}(\mathbf{x}, \mathbf{u}, \mathbf{y}) = \sum_{\mathbf{w}(\mathbf{j},\mathbf{q})} \mu_{\mathbf{j},\mathbf{q}}^{agg}(\mathbf{x}, \mathbf{u}) \cdot \mu_{\mathbf{w}(\mathbf{j},\mathbf{q})}^{Singleton}(\mathbf{y}). \quad (3.7)$$

Schritt 5 (Defuzzifizierung): In der Defuzzifizierung nach der Schwerpunktmethode (CoS) wird der gewichtete Schwerpunkt aller in der Akkumulation addierten Singletons gebildet. Dazu muss man nur die gewichtete Summe aus den Singletonpositionsvektoren $\mathbf{s}_{\mathbf{w}(\mathbf{j},\mathbf{q})}^{\mathbf{x}}$ bilden und die Wahrheitswerte $\mu_{\mathbf{j},\mathbf{q}}^{agg}(\mathbf{x}, \mathbf{u})$ als Gewichte heranziehen. So erhält man den neuen Zustandsvektor durch die Vektorfunktion

$$\mathbf{f}(\mathbf{x}, \mathbf{u}) = \frac{\sum_{\mathbf{j},\mathbf{q}} \mathbf{s}_{\mathbf{w}(\mathbf{j},\mathbf{q})}^{\mathbf{x}} \cdot \mu_{\mathbf{j},\mathbf{q}}^{agg}(\mathbf{x}, \mathbf{u})}{\sum_{\hat{\mathbf{j}},\hat{\mathbf{q}}} \mu_{\mathbf{w}(\hat{\mathbf{j}},\hat{\mathbf{q}})}^{agg}(\mathbf{x}, \mathbf{u})}. \quad (3.8)$$

Gibt es nur eine Ausgangsgröße, wie es in vielen Fuzzy-Systemen der Fall ist, so ist die Funktion $\mathbf{f}$ keine Vektorfunktion, sondern eine skalare Funktion f .

Anstatt obige Schritte einzeln durchzuführen, kann man den Wert der Transitionsfunktion auch durch eine einzige Gleichung bestimmen. Diese erhält man, indem man den Ausdruck $\mu_{\mathbf{w}(\mathbf{j},\mathbf{q})}^{agg}(\mathbf{x}, \mathbf{u})$ aus Gleichung (3.5) in Gleichung (3.8) einsetzt. So ergibt sich schließlich die folgende geschlossene Darstellung der Transitionsfunktion $\mathbf{f}$, die direkt von den Zugehörig-

keitsfunktionen und Singletonpositionen abhängt:

$$\mathbf{f}(\mathbf{x}, \mathbf{u}) = \frac{\sum_{\mathbf{j}, \mathbf{q}} \mathbf{s}_{\mathbf{w}(\mathbf{j}, \mathbf{q})}^{\mathbf{x}} \prod_i \mu_{j_i}^{x_i}(x_i) \prod_p \mu_{q_p}^{u_p}(u_p)}{\sum_{\hat{\mathbf{j}}, \hat{\mathbf{q}}} \prod_i \mu_{\hat{j}_i}^{x_i}(x_i) \prod_p \mu_{\hat{q}_p}^{u_p}(u_p)}. \quad (3.9)$$

Wenn die Zugehörigkeitsfunktionen $\mu_{j_i}^{x_i}(x_i)$ und $\mu_{q_p}^{u_p}(u_p)$ zusätzlich stetig in x_i bzw. u_p sind, so sind die Funktionen $\mathbf{f}$ und $\mathbf{g}$ auch stetig in den Variablen $\mathbf{x}$ und $\mathbf{u}$. In diesem Fall wird das System als *stetiges* rekurrentes Fuzzy-System bezeichnet.

Unter Umständen kann die Form der Transitionsfunktion aus Gleichung (3.9) noch weiter vereinfacht werden, wie im Folgenden gezeigt wird. Dazu formt man den Nenner auf der rechten Seite von Gleichung (3.9) folgendermaßen um:

$$\sum_{\hat{\mathbf{j}}, \hat{\mathbf{q}}} \prod_i \mu_{\hat{j}_i}^{x_i}(x_i) \prod_p \mu_{\hat{q}_p}^{u_p}(u_p) = \prod_i \sum_{\hat{j}_i} \mu_{\hat{j}_i}^{x_i}(x_i) \cdot \prod_p \sum_{\hat{q}_p} \mu_{\hat{q}_p}^{u_p}(u_p). \quad (3.10)$$

Fasst man jeden Faktor $\sum_{\hat{j}_i} \mu_{\hat{j}_i}^{x_i}(x_i)$ mit dem Index i (bzw. p) aus dem Nenner in Gleichung (3.10) mit entsprechenden Faktoren $\mu_{j_i}^{x_i}(x_i)$ im Zähler von Gleichung (3.9) zusammen, so ergibt sich für $\mathbf{f}$:

$$\mathbf{f}(\mathbf{x}, \mathbf{u}) = \sum_{\mathbf{j}, \mathbf{q}} \mathbf{s}_{\mathbf{w}(\mathbf{j}, \mathbf{q})}^{\mathbf{x}} \prod_i \frac{\mu_{j_i}^{x_i}(x_i)}{\sum_{\hat{j}_i} \mu_{\hat{j}_i}^{x_i}(x_i)} \prod_p \frac{\mu_{q_p}^{u_p}(u_p)}{\sum_{\hat{q}_p} \mu_{\hat{q}_p}^{u_p}(u_p)}. \quad (3.11)$$

Wenn sich die Zugehörigkeitsfunktionen $\mu_{j_i}^{x_i}$ (bzw. $\mu_{q_p}^{u_p}$) an jeder Stelle zu eins ergänzen, d.h. wenn

- (B3') Normalisierte Partitionierung: $\sum_{j_i} \mu_{j_i}^{x_i}(x_i) = 1$ für alle $x_i \in X_i$,

als Verschärfung von Bedingung (B3) erfüllt ist, so haben die Nenner aus Gleichung (3.11) den Wert eins und können daher einfach weggelassen werden. Dies ist oft der Fall.

Man kann Bedingung (B3') immer erfüllen, indem man Zugehörigkeitsfunktionen $\tilde{\mu}_{j_i}^{x_i}(x_i)$, welche schon die Bedingungen (B1) bis (B4) genügen, durch neue Zugehörigkeitsfunktionen $\mu_{j_i}^{x_i}(x_i) = \frac{\tilde{\mu}_{j_i}^{x_i}(x_i)}{\sum_{\hat{j}_i} \tilde{\mu}_{\hat{j}_i}^{x_i}(x_i)}$ ersetzt. Dabei ändert sich der Wert der Transitionsfunktion nicht. Die neuen Zugehörigkeitsfunktionen erfüllen dann die Bedingungen (B1) bis (B4), Bedingung (B3') und die folgende Verschärfung der Rückkopplungskorrespondenzbedingung (B4):

- (B4') Normalisierte Rückkopplungskorrespondenz: $\mu_{j_i}^{x_i}(s_{j_i}^{x_i}) = 1$, $\mu_{j_i}^{x_i}(s_{l_i}^{x_i}) = 0$ für $j_i \neq l_i$.

Für den Rest der Arbeit wird immer angenommen, dass die Zugehörigkeitsfunktionen die Normierungsbedingung

$$\sum_{j_i} \mu_{j_i}^{x_i}(x_i) = 1 \quad (3.12)$$

von (B3') erfüllen oder durch solche Zugehörigkeitsfunktionen ersetzt werden.

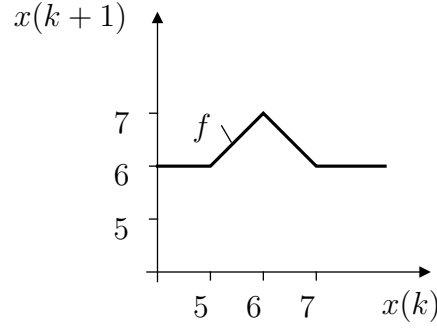


Abbildung 3.4: Transitionsfunktion für ein einfaches Modell einer Insektenpopulation für ein üppiges Nahrungsangebot.

In beiden Fällen fallen also die Nenner von Gleichung (3.11) weg und Gleichung (3.9) vereinfacht sich zu

$$\mathbf{f}(\mathbf{x}, \mathbf{u}) = \sum_{\mathbf{j}, \mathbf{q}} \mathbf{s}_{\mathbf{w}(\mathbf{j}, \mathbf{q})}^{\mathbf{x}} \prod_i \mu_{j_i}^{x_i}(x_i) \prod_p \mu_{q_p}^{u_p}(u_p). \quad (3.13)$$

Im Modell für die Entwicklung der Größe einer Insektenpopulation erfüllen die Zugehörigkeitsfunktionen bereits Bedingung (B3'), wie sich leicht anhand von Abbildung 3.2 überprüfen lässt. Da das System nur eine einzige Zustandsgröße x und eine einzige Eingangsgröße u hat, kann man wiederum die Doppelindizes x_i und u_p durch einfache Indizes x und u ersetzen und erhält so für die Transitionsfunktion:

$$f(x, u) = \sum_{j=1}^3 \sum_{q=1}^3 s_{w(j,q)}^x \cdot \mu_j^x(x) \cdot \mu_q^u(u). \quad (3.14)$$

Die Werte von $s_{w(j,q)}^x$ können einfach aus der Regelbasis in Abbildung 3.1 abgelesen werden. Dabei sind die linguistischen Werte nur durch die entsprechenden Kernpositionswerte zu ersetzen. Damit ist die mathematische Modellierung der zeitlichen Entwicklung der Insektenpopulation durch ein rekurrentes Fuzzy-System abgeschlossen.

Im Falle eines definitiv üppigen Nahrungsangebotes, d.h. für $\mu_1^u(u) = 0$, $\mu_2^u(u) = 0$ und $\mu_3^u(u) = 1$, ergibt sich mit $s_{w(1,3)}^x = s_2^x = 6$, $s_{w(2,3)}^x = s_3^x = 7$ und $s_{w(3,3)}^x = s_2^x = 6$ die Transitionsfunktion f zu:

$$x(k+1) = f(x(k), u(k)) = 6 \cdot \mu_1^x(x(k)) + 7 \cdot \mu_2^x(x(k)) + 6 \cdot \mu_3^x(x(k)). \quad (3.15)$$

Die Funktion ist in Abbildung 3.4 zu sehen.

Behält man den Eingabewert als Parameter in der Transitionsfunktion bei, so lässt sie sich als Kennfeld darstellen, wie in Abbildung 3.5 gezeigt ist. An den Kernpositionsvektoren (s_j^x, s_q^u) sind die linguistischen Werte der Folgewerte aus der Regelbasis vermerkt. Die Funktionswerte zwischen den Kernpositionsvektoren erhält man durch Interpolation mit Hilfe der Zugehörigkeitsfunktionen der linguistischen Werte.

Nachdem die Form der Transitionsfunktion feststeht, lassen sich schließlich noch Rückschlüsse auf eine geeignete Wahl eines Zustandsraumes X ziehen.

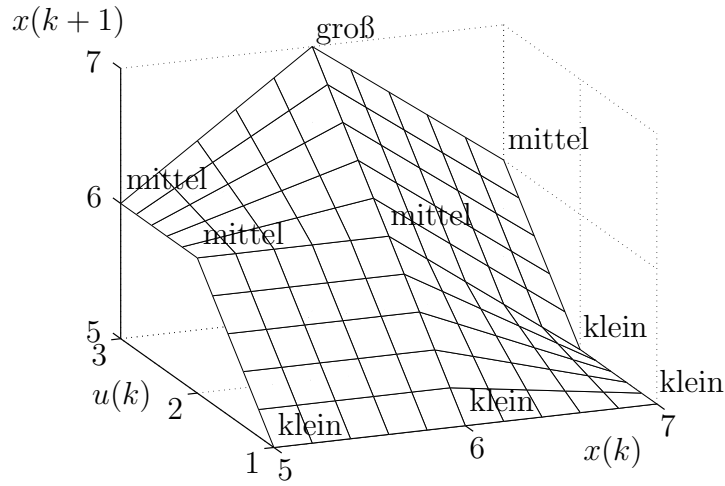


Abbildung 3.5: Transitionsfunktion für ein einfaches Modell einer Insektenpopulation.

Der Zustandsraum X kann beliebig groß sein. Er enthält aber in jedem Fall alle Vektoren $\mathbf{x}$, deren Komponenten x_i für jeden Index i zwischen der jeweils kleinsten Kernposition $s_1^{x_i}$ und größten Kernposition $s_{j_{max}}^{x_i}$ liegen. Diese Vektoren bilden eine kompakte, d.h. beschränkte und abgeschlossene Teilmenge im Zustandsraum. Für die Untersuchung rekurrenter Fuzzy-Systeme kann man sich auf diese Teilmenge beschränken. Dies hat zwei Gründe:

Erstens kommt jeder Zustandvektor nach einem Zeitschritt in der obiger Teilmenge des Zustandsraumes zu liegen, denn die Transitionsfunktion $\mathbf{f}$ realisiert eine Interpolation der Kernpositionsvektoren $\mathbf{s}_w^{\mathbf{x}}$, wie aus Gleichung (3.11) bzw. Gleichung (3.13) zu ersehen ist.

Zweitens werden Zustandsvektoren, die zu Beginn außerhalb obiger Teilmenge liegen, genau so behandelt wie spezielle Vektoren innerhalb dieser Teilmenge. Liegt eine Komponente eines Vektors $\mathbf{x}$ außerhalb der Begrenzung obiger Teilmenge, ist z.B. $x_1 < s_1^{x_1}$, so ist der Zugehörigkeitswert dieser Komponente zu der begrenzenden Zugehörigkeitsfunktion gleich eins, d.h. $\mu_1^{x_1}(x_1) = 1$ in diesem Beispiel. Da die Komponenten x_i nur über die Zugehörigkeitswerte $\mu_{j_i}^{x_i}(x_i)$ in die Berechnung des Folgezustandes eingehen, wird ein solcher Vektor so behandelt, als läge der Wert der entsprechenden Komponenten genau auf der Begrenzung, d.h. $x_1 = s_1^{x_1}$ in diesem Beispiel. Dieser Effekt ist auch in der Transitionsfunktion in Abbildung 3.5 für Zustandwerte $x < 5$ zu erkennen.

Für die Bildmenge $\mathbf{f}(X \times U)$ spielt es also keine Rolle, ob der Zustandsraum X größer als die kompakte Teilmenge ist oder nicht. Für die weitere Behandlung rekurrenter Fuzzy-Systeme wird angenommen, dass der Zustandsraum X schon kompakt ist bzw. mit obiger kompakter Teilmenge übereinstimmt. Dies erleichtert die mathematische Untersuchung rekurrenter Fuzzy-Systeme in den nächsten Kapiteln.

4 Automaten und Automatenähnlichkeit

Nach der Definition von rekurrenten Fuzzy-Systemen wird in diesem Kapitel ihr Verhältnis zu Automaten beleuchtet. Die Untersuchung ist dadurch motiviert, dass Automaten ein dynamisches Verhalten anhand von Transitionen beschreiben, die als logische Schaltfunktionen darstellbar sind. Rekurrente Fuzzy-Systeme beschreiben auch ein dynamisches Verhalten anhand einer Transitionsfunktion, die aus fuzzy-logischen Funktionen aufgebaut ist. Die Verwendung von Fuzzy-Logik als Erweiterung der Booleschen Logik legt nahe, dass rekurrente Fuzzy-Systeme eine Erweiterung von Automaten sind [4]. Die Frage ist: Unter welchen Bedingungen zeigen sich welche Automateneigenschaften in rekurrenten Fuzzy-Systemen?

4.1 Linguistischer Automat

Um die Beziehung von rekurrenten Fuzzy-Systemen zu Automaten zu erhellen, soll zuerst die Regelbasis mit ihren Regeln nach Gleichung (3.4) betrachtet werden. Sie bestimmt das dynamische Verhalten maßgeblich. Die Regeln in der Form

$$\text{”Wenn } \mathbf{x}(k) = \mathbf{L}_j^x \text{ und } \mathbf{u}(k) = \mathbf{L}_q^u, \text{ dann } \mathbf{x}(k+1) = \mathbf{L}_w^x \text{”} \quad (4.1)$$

beschreiben die Abbildung der linguistischen Vektoren $\mathbf{L}_j^x$ des Zustandes zum Zeitpunkt k auf den neuen linguistischen Vektor $\mathbf{L}_w^x$ zum Zeitpunkt $k+1$ in Abhängigkeit des linguistischen Eingabevektors $\mathbf{L}_q^u$. Diese Abbildung erfüllt also auch Gleichung (3.1), wobei die abgebildeten Objekte nicht Zahlenwerte sind, sondern aus einer endlichen Menge von linguistischen Vektoren $\mathbf{L}_j^x$ stammen. Die Regeln aus Gleichung (4.1) mit der Rückkopplung bilden damit einen endlichen Automaten, der hier als *linguistischer Automat* bezeichnet wird. Weitere Informationen zu Automaten finden sich in [18, 23].

Das Verhalten des linguistischen Automaten lässt sich sehr gut anhand eines Zustandsgraphen visualisieren. Dieser kann direkt aus der Regelbasis abgeleitet werden. Die Knoten werden von den linguistischen Vektoren $\mathbf{L}_j^x$ des Zustandsvektors $\mathbf{x}$ gebildet. Die Transitionspfeile geben je nach linguistischen Vektoren $\mathbf{L}_q^u$ des Eingabevektors $\mathbf{u}$ die Übergangsbedingungen von einem Zustand in einen anderen an.

Ein solcher Zustandsgraph ist in Abbildung 4.1 für die Regelbasis des Beispiels der Insektenpopulation aus Abbildung 3.1 aus dem letzten Kapitel zu sehen. Die welligen Knoten symbolisieren dabei die unscharfe Natur der Zustandsvariablen x , hier der Populationsgröße (PG), die die linguistischen Werte klein, mittel und groß annehmen kann. Die einzige

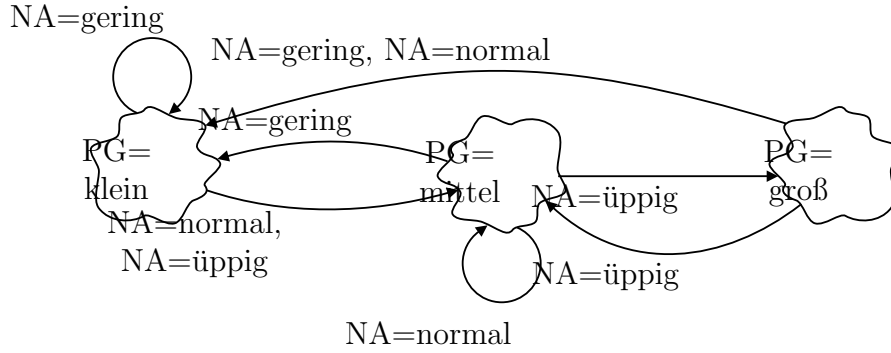


Abbildung 4.1: Zustandsgraph eines einfachen Modells einer Insektenpopulation basierend auf der Populationsgröße (PG) und dem Nahrungsangebot (NA).

Eingangsgröße u , hier das Nahrungsangebot (NA), kann die linguistischen Werte gering, normal und üppig annehmen.

Für einen linguistischen Automaten lässt sich immer aus dessen Regelbasis ein linguistischer Zustandsgraph erstellen [4]. Umgekehrt lassen sich auch aus dieser Darstellung alle linguistischen Werte und die gesamte Regelbasis ablesen. Der Zustandsgraph erlaubt auf einfache Weise, das Verhalten des linguistischen Automaten über mehrere Zeitschritte hinweg zu überblicken. So können auch Eigenschaften wie Erreichbarkeit und Steuerbarkeit von ihm abgelesen werden.

Will man nun nicht nur linguistische Werte, sondern auch Zahlenwerte u_p als Eingangsgrößen und x_i als Zustandsgrößen verarbeiten, so muss man sie mit den linguistischen Werten in Beziehung setzen. Dafür werden die Zahlenwerte in (Fuzzy-)Mengen zusammengefasst, die einem linguistischen Wert entsprechen. Nach weiteren Schritten erhält man so ein rekurrentes Fuzzy-System. Wählt man unstetige Zugehörigkeitsfunktionen, die nur die Wahrheitswerte 0 oder 1 annehmen können, so bilden diese Systeme weiterhin deterministische Automaten, weil immer nur genau eine Regel zur Berechnung des Folgezustandes gilt.

Aber auch bei allgemeinen rekurrenten Fuzzy-Systemen bleibt eine Verbindung zu endlichen Automaten erhalten, wie folgendes Lemma beschreibt:

Lemma 1

Ein rekurrentes Fuzzy-System operiert auf den Kernpositionsvektoren wie der linguistische Automat auf den linguistischen Vektoren.

Beweis:

Wählt man Kernpositionsvektoren $\mathbf{s}_j^x$ und $\mathbf{s}_q^u$ für $\mathbf{x}(k)$ und $\mathbf{u}(k)$, so haben nach der Rückkopplungskorrespondenzbedingung (B4) nur die linguistischen Komponenten $L_{j_i}^{x_i}$ und $L_{q_p}^{u_p}$ der entsprechenden linguistischen Vektoren $\mathbf{L}_j^x$ und $\mathbf{L}_q^u$ positive Zugehörigkeitswerte $\mu_{j_i}^{x_i}(x_i)$ und $\mu_{q_p}^{u_p}(u_p)$. Damit hat nach Gleichung (3.5) ausschließlich die Regel einen positiven Wahrheitswert, die die entsprechenden linguistischen Vektoren $\mathbf{L}_j^x$ und $\mathbf{L}_q^u$ in ihrer Prämisse stehen hat. Da die Kernpositionsvektoren, die den linguistischen Vektoren in den Konklusionen der Regeln entsprechen, für die Berechnung des zukünftigen Zustandsvektors $\mathbf{x}(k+1)$

laut Gleichung (3.13) mit dem Wahrheitswert der entsprechenden Regeln gewichtet werden, ergibt sich $\mathbf{x}(k+1) = \mathbf{f}(\mathbf{s}_j^x, \mathbf{s}_q^u) = \mathbf{s}_{w(j,q)}^x$. Der zukünftige Zustandsvektor $\mathbf{x}(k+1)$ ist also wieder ein Kernpositionsvektor, und zwar der, der dem linguistischen Vektor $\mathbf{L}_{w(j,q)}^x$ entspricht, welcher in der Konklusion der einzigen Regel mit positivem Wahrheitswert steht. $\square$

Die Dynamik eines auf die Kernpositionsvektoren eingeschränkten rekurrenten Fuzzy-Systems kann man also durch den linguistischen Automaten beschreiben und umgekehrt. In Analogie zu Gleichung (4.1) kann man auch

$$\text{''Wenn } \mathbf{x}(k) = \mathbf{s}_j^x \text{ und } \mathbf{u}(k) = \mathbf{s}_q^u, \text{ dann } \mathbf{x}(k+1) = \mathbf{f}(\mathbf{s}_j^x, \mathbf{s}_q^u) = \mathbf{s}_{w(j,q)}^x \text{''} \quad (4.2)$$

schreiben. Dies zeigt die enge Beziehung der linguistischen Vektoren $\mathbf{L}_j^x$ und $\mathbf{L}_q^u$ zu den Kernpositionsvektoren $\mathbf{s}_j^x$ und $\mathbf{s}_q^u$. Diese Eigenschaft zeichnet die Kernpositionen gegenüber anderen Eingangs- und Zustandswerten aus.

Ein rekurrentes Fuzzy-System ist aber nicht nur an den Kernpositionsvektoren definiert. Durch die Zugehörigkeitsfunktionen wird der gesamte Eingangs- und Zustandsraum überdeckt. Bei stetigen Zugehörigkeitsfunktionen gibt es immer einen Überlappungsbereich von benachbarten Zugehörigkeitsfunktionen, in dem beide von null verschieden sind. Dann gilt mehr als eine Regel für die Berechnung des Folgezustandes, die nach dem Wahrheitswert ihrer Prämissen gewichtet werden. Im nächsten Abschnitt wird untersucht, inwieweit der Zustandsgraph die Dynamik eines rekurrenten Fuzzy-Systems beschreiben kann, auch wenn mehrere Transitions Pfeile gleichzeitig wirken.

4.2 Automatenähnlichkeit rekurrenter Fuzzy-Systeme

Ein rekurrentes Fuzzy-System ist auf Kernpositionsvektoren $(\mathbf{s}_j^x, \mathbf{s}_q^u)$ eindeutig durch den Zustandsgraphen eines endlichen Automaten beschreibbar. Für den sinnvollen Einsatz des Zustandsgraphen an beliebigen Stellen $(\mathbf{x}, \mathbf{u})$ im Raum $X \times U$, der durch den Zustandsraum X und den Eingangsraum U aufgespannt wird, ist Folgendes notwendig: Es ist ein in der Nähe von $(\mathbf{x}, \mathbf{u})$ liegender Kernpositionsvektor $(\mathbf{s}_j^x, \mathbf{s}_q^u)$ zu finden, so dass das rekurrente Fuzzy-System für diesen Anfangsvektor annähernd das gleiche Verhalten wie für den ursprünglichen Vektor $(\mathbf{x}, \mathbf{u})$ zeigt. In diesem Fall beschreibt der Zustandsgraph des linguistischen Automaten auch annähernd das dynamische Verhalten für den Vektor $(\mathbf{x}, \mathbf{u})$. Das rekurrente Fuzzy-System verhält sich dann für diesen Vektor $(\mathbf{x}, \mathbf{u})$ automatenähnlich. Um diese Idee zu formalisieren, wird die Nähe von Vektoren $(\mathbf{x}, \mathbf{u})$ zu Kernpositionsvektoren $(\mathbf{s}_j^x, \mathbf{s}_q^u)$ wie folgt charakterisiert:

Definition 1 (Verbindung)

Der Kernpositionsvektor $(\mathbf{s}_j^x, \mathbf{s}_q^u)$ und ein Vektor $(\mathbf{x}, \mathbf{u})$ heißen verbunden, falls in jedem geschlossenen Intervall, das durch die Werte der jeweiligen Komponenten beider Vektoren begrenzt wird, nur eine Kernposition liegt.

Ein Kernpositionsvektor ist mit allen Vektoren verbunden, die in dem maximalen, offenen, achsenparallelen Hyperquader liegen, der nur diesen Kernpositionsvektor als einzigen Kernpositionsvektor enthält. In Abbildung 4.2(a) ist ein solcher Hyperquader gestrichelt

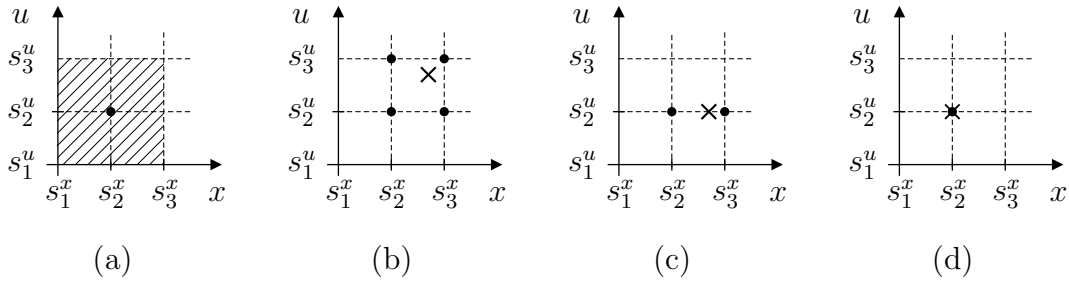


Abbildung 4.2: (a) Der gestichelte Hyperquader enthält alle Vektoren aus $X \times U$, die mit (s_2^x, s_2^u) verbunden sind. Vektoren können mit mehreren Kernpositionsvektoren verbunden sein (siehe (c),(d)) oder mit einem einzigen (siehe (d)).

ingezeichnet. Umgekehrt ist ein Vektor im Allgemeinen mit mehreren Kernpositionsvektoren verbunden. Abbildungen 4.2(b) und 4.2(c) zeigen beispielhaft durch Kreuze markierte Vektoren. Die mit ihnen verbundenen Kernpositionsvektoren sind mit Punkten markiert. Ist der Vektor selbst ein Kernpositionsvektor, so ist er mit keinem anderen Kernpositionsvektor als sich selbst verbunden. Siehe dazu Abbildung 4.2(d). Zwei unterschiedliche Kernpositionsvektoren sind also nie miteinander verbunden. Automatenähnlichkeit ist nun wie folgt definiert:

Definition 2 (Automatenähnlichkeit)

Ein rekurrentes Fuzzy-System heißt automatenähnlich in einer Menge $M \subseteq X \times U$, falls es für jeden Vektor $(\mathbf{x}, \mathbf{u}) \in M$ einen mit ihm verbundenen Kernpositionsvektor (s_j^x, s_q^u) gibt, so dass auch der Kernpositionsvektor $f(s_j^x, s_q^u)$ mit $f(\mathbf{x}, \mathbf{u})$ verbunden ist.

Automatenähnlichkeit überträgt sich sofort auf Vereinigungsmengen solcher Mengen M . Sie ist auch in beliebigen Teilmengen von solchen Mengen M erfüllt.

Automatenähnliche rekurrente Fuzzy-Systeme erlauben es, die Dynamik in der Menge M durch die Dynamik auf den Kernpositionsvektoren anzunähern. Da der Zustandsgraph die Dynamik der Kernpositionsvektoren vollständig beschreibt, kann er also auch zur näherungsweisen Beschreibung der Dynamik des rekurrenten Fuzzy-Systems auf der Menge M genutzt werden. So haben Startpunkte nahe eines bestimmten Kernpositionsvektors auch nach der Abbildung einen geringen Abstand zum Bild dieses Kernpositionsvektors. Während diese Eigenschaft von jeder stetigen Abbildung erfüllt wird, gibt die Automatenähnlichkeit eine Schranke für ein mögliches Anwachsen dieses Abstands an.

Im Speziellen kann ein automatenähnliches rekurrentes Fuzzy-System keinen Kernpositionsvektor als Zustandsvektor annehmen, wenn dieser nicht explizit in den für diesen Rechenschritt benötigten Regeln auftaucht. Mit der Definition der Automatenähnlichkeit folgt aus Lemma 1 direkt

Lemma 2

Jedes rekurrente Fuzzy-System ist automatenähnlich in der Menge seiner Kernpositionsvektoren.

Interessant wäre nun ein Kriterium, mit dem sich für einen Teilraum M von $X \times U$ oder für den gesamten Raum $X \times U$ feststellen ließe, ob ein rekurrentes Fuzzy-System automatenähnlich ist oder nicht. Um ein solches Kriterium formulieren zu können, sind die folgenden Definitionen 3 bis 5 hilfreich:

Definition 3 (Nachbarschaft)

Zwei Kernpositionsvektoren heißen benachbart, wenn sie sich nur in einer Komponente unterscheiden und es zwischen den beiden Werten keine weitere Kernposition gibt.

Zwei linguistische Vektoren $\mathbf{L}_j^x$ (oder $\mathbf{L}_q^u$) heißen benachbart, wenn ihre Kernpositionsvektoren benachbart sind.

Zwei Regeln heißen benachbart, wenn sie in Form von Gleichung (4.1) vorliegen und ihre linguistischen Zustands-/Eingabevektoren ($\mathbf{L}_j^x, \mathbf{L}_q^u$) in der Prämisse benachbart sind.

Definition 4 (Regelstetigkeit)

Eine Regelbasis in der Form von Gleichung (4.1) heißt auf einer bestimmten Menge von Regeln regelstetig, wenn bei je zwei benachbarten Regeln aus dieser Menge auch ihre linguistischen Vektoren im Konklusionsteil gleich oder benachbart sind.

Ist eine Regelbasis auf allen Regeln regelstetig, so heißt die gesamte Regelbasis regelstetig.

Wenn eine Regelbasis regelstetig ist, so überspringt der Konklusionsteil keine linguistischen Werte. Sie verläuft sozusagen linguistisch stetig. Die Bedingungen sind leicht anhand der Matrixdarstellung der Regelbasis zu überprüfen. Senkrecht oder waagerecht benachbarte Matrixfelder müssen gleiche oder benachbarte linguistische Werte enthalten. So lässt sich auch sehr einfach nachweisen, dass die Regelbasis des Insektenpopulationsmodells in Abbildung 3.1 regelstetig ist.

Definition 5 (Elementarer Hyperquader)

Die Menge der mit einem Vektor $(\mathbf{x}, \mathbf{u})$ verbundenen Kernpositionsvektoren bilden die Eckpunkte eines Hyperquaders H . Dieser Hyperquader wird als elementarer Hyperquader des Vektors $(\mathbf{x}, \mathbf{u})$ bezeichnet.

Elementare Hyperquader können eine kleinere Dimension als der Raum $X \times U$ besitzen. So ist z.B. jeder Kernpositionsvektor ein elementarer Hyperquader der Dimension 0. Ein Vektor kann auch gleichzeitig in verschiedenen elementaren Hyperquadern von verschiedenen Vektoren liegen, z.B. wenn er auf gemeinsamen Oberflächen, Kanten oder Eckpunkten dieser Hyperquader liegt.

Die Regeln in kompakter Form von Gleichung (4.1) werden als die *Regeln des Hyperquaders* bezeichnet, wenn deren Prämissen Kernpositionsvektoren entsprechen, die Eckpunkte des elementaren Hyperquaders sind. Liegt ein Zustands-/Eingabevektor $(\mathbf{x}(k), \mathbf{u}(k))$ in einem elementaren Hyperquader, so ist er nur mit den Eckpunkten des Hyperquaders verbunden. Bei der Berechnung des Folgezustandes $\mathbf{x}(k+1)$ werden nur die Regeln des elementaren Hyperquaders benötigt.

Mit diesen Definitionen kann der folgende Satz für *eindimensionale* rekurrente Fuzzy-Systeme, d.h. für rekurrente Fuzzy-Systeme mit nur einer Zustandsgröße, formuliert werden:

Satz 1 (Ähnlichkeitssatz)

- (a) Ein eindimensionales rekurrentes Fuzzy-System ist in einem elementaren Hyperquader $M \subseteq X \times U$ automatenähnlich, wenn die Regeln dieses Hyperquaders regelstetig sind.
- (b) Ein eindimensionales stetiges rekurrentes Fuzzy-System ist genau dann in einem elementaren Hyperquader $M \subseteq X \times U$ automatenähnlich, wenn die Regeln dieses Hyperquaders regelstetig sind.

Wie im Absatz nach der Definition von Automatenähnlichkeit erwähnt, ist ein rekurrentes Fuzzy-System, das in mehreren elementaren Hyperquaden automatenähnlich ist, auch in der Vereinigungsmenge der Hyperquader automatenähnlich. Im Speziellen ist ein rekurrentes Fuzzy-System mit nur einer Zustandsgröße im gesamten Raum $X \times U$ automatenähnlich, wenn seine gesamte Regelbasis regelstetig ist.

Beweis des Ähnlichkeitssatzes:

Da es in einem eindimensionalen rekurrenten Fuzzy-System nur eine Zustandsgröße gibt, sind die Kernpositionen s_j^x in X schon die Kernpositionsvektoren $\mathbf{s}_j^x$ in X .

(a) Angenommen ein rekurrentes Fuzzy-System mit nur einer Zustandsgröße x ist in einem elementaren Hyperquader M regelstetig. Man betrachte einen beliebigen Zustands-/Eingangsvektor $(x, \mathbf{u}) \in M$ und dessen elementaren Hyperquader H , der dann auch in M liegt. Der Vektor $(x, \mathbf{u})$ ist dann genau mit allen Eckpunkten von H verbunden. Die Eckpunkte von H werden nach Lemma 1 durch die Transitionsfunktion f auf eine Menge R von Kernpositionen aus X abgebildet.

Die Menge R enthält aufgrund der Regelstetigkeit alle Kernpositionen in X zwischen $r_{\min} = \min\{R\}$ und $r_{\max} = \max\{R\}$. Um dies zu sehen, betrachte man die Regel von H , die zur Kernposition $r_{\min}$ führt, und ändere schrittweise die linguistischen Werte in den Voraussetzungen der Regeln, bis man zu der Regel von H kommt, die zur Kernposition $r_{\max}$ führt. Aufgrund der Regelstetigkeit hat sich dabei bei jedem Schritt die Kernposition in der Konklusion jeweils um höchstens eine Einheit geändert. Also wurden alle Kernpositionen zwischen $r_{\min}$ und $r_{\max}$ erreicht.

Für die Automatenähnlichkeit ist nur noch zu zeigen, dass eine mit dem Folgevektor $f(x, \mathbf{u})$ verbundene Kernposition im Intervall $[r_{\min}, r_{\max}]$ und damit in R liegt. Da der Folgezustand $f(x, \mathbf{u})$ aufgrund der Rechenvorschrift nach Gleichung (3.13) eine Konvexkombination bzw. nach Gleichung (3.9) ein verallgemeinerter Mittelwert der Kernpositionen aus R ist, gilt: $r_{\min} \leq f(x, \mathbf{u}) \leq r_{\max}$. Alle mit $f(x, \mathbf{u})$ verbundenen Kernpositionen liegen auch in $[r_{\min}, r_{\max}]$, da sonst eine der beiden Kernpositionen $r_{\min}$ oder $r_{\max}$ zwischen $f(x, \mathbf{u})$ und deren verbundenen Kernpositionen liegen würde. Also ist $f(x, \mathbf{u})$ mit einem Kernpositionsvektor aus R verbunden und damit ist das System automatenähnlich.

(b) Wegen Teil (a) reicht es zu zeigen, dass ein stetiges rekurrentes Fuzzy-System in einem elementaren Hyperquader nicht automatenähnlich sein kann, wenn die Regelbasis auf den Regeln dieses elementaren Hyperquaders nicht regelstetig ist.

Angenommen ein rekurrentes Fuzzy-System mit nur einer Zustandsgröße x ist in einem elementaren Hyperquader nicht regelstetig. Im ersten betrachteten Fall sollen bei einem festen linguistischen Eingabevektor $\mathbf{L}_a^u$ zwei benachbarte linguistische Zustände L_a^x und L_b^x zu unterschiedlichen und nicht benachbarten Folgezuständen führen. Dann ergeben sich

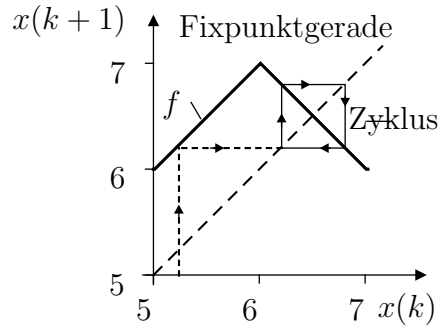


Abbildung 4.3: Transitionsfunktion eines einfachen Modells einer Insektenpopulation.

für den Eingabevektor $\mathbf{u} = \mathbf{s}_q^u$ und die beiden Kernpositionen s_a^x und s_b^x die Funktionswerte $f(s_a^x, \mathbf{s}_q^u)$ und $f(s_b^x, \mathbf{s}_q^u)$, die nach Lemma 1 unterschiedliche und nicht benachbarte Kernpositionen sind. Da die Funktion f stetig in der Variablen x ist, erhält man für einen bestimmten Zwischenwert $x_0 \in [s_a^x, s_b^x]$ einen Funktionswert $f(x_0, \mathbf{s}_q^u)$, der auf einer Kernposition zwischen den Kernpositionen $f(s_a^x, \mathbf{s}_q^u)$ und $f(s_b^x, \mathbf{s}_q^u)$ liegt. Nun ist der Vektor $(x_0, \mathbf{s}_q^u)$ ausschließlich mit den Kernpositionsvektoren $(s_a^x, \mathbf{s}_q^u)$ und $(s_b^x, \mathbf{s}_q^u)$ verbunden. Sein Funktionswert $f(x_0, \mathbf{s}_q^u)$ ist aber mit keiner anderen Kernposition als sich selbst verbunden, also weder mit der Kernposition $f(s_a^x, \mathbf{s}_q^u)$ noch mit der Kernposition $f(s_b^x, \mathbf{s}_q^u)$, wie es für die Erfüllung der Automatenähnlichkeit gefordert wird. Also ist in diesem Fall das rekurrente Fuzzy-System in dem betrachteten elementaren Hyperquader nicht automatenähnlich.

Der zweite Fall, in dem gleiche linguistische Zustandswerte bei benachbarten linguistischen Eingaben zu nicht benachbarten Folgezuständen führen, folgt das Ergebnis analog. $\square$

Es ist also durch ein einfaches Verfahren für eindimensionale rekurrente Fuzzy-Systeme möglich nachzuweisen, dass sie die angenehme Eigenschaft der Automatenähnlichkeit besitzen. Insbesondere bei der Nachbildung von explizit formulierbarem Wissen des Menschen über einen dynamischen Vorgang ist aufgrund der Überschaubarkeit häufig nur eine Zustandsgröße zu verzeichnen und das System ist oft regelstetig und somit auch automatenähnlich. Dann wird dessen Dynamik durch das linguistische Modell approximativ beschrieben. Linguistische Modelle können in einer Vielzahl von Fachrichtungen Verwendung finden wie z.B. in den Ingenieurwissenschaften, der Ökologie, der Ökonomie oder in den Sozialwissenschaften [49, 16, 6]. Da insbesondere die mathematische Differentiation und Integration durch regelstetige linguistische Modelle beschrieben werden können [6], ist zu erwarten, dass es eine Reihe von regelstetigen Modellen für dynamische Prozesse gibt.

Hat ein rekurrentes Fuzzy-System mehr als eine Zustandsgröße, so gilt die Beziehung von Regelstetigkeit und Automatenähnlichkeit im Allgemeinen nicht mehr. Im Anhang A wird dies anhand zweier Gegenbeispiele erläutert.

Zur Illustration des Ähnlichkeitssatzes betrachte man wieder das Modell der Insektenpopulation. Wie oben erwähnt, ist die Regelbasis des rekurrenten Fuzzy-Systems regelstetig. Da das System eindimensional ist, folgt damit aus dem Ähnlichkeitssatz, dass es auch automatenähnlich ist. Die Transitionsfunktion für den Fall eines üppigen Nahrungsangebots ist in Abbildung 4.3 gezeigt. Die eingezeichnete Fixpunktgerade erlaubt es, die zeitliche

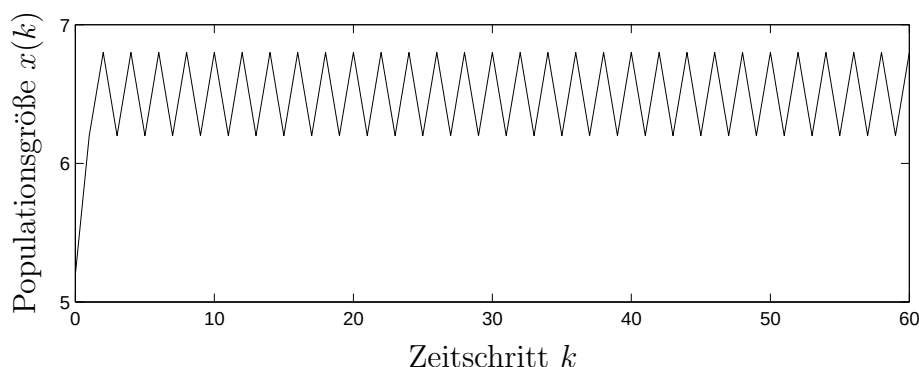


Abbildung 4.4: Zeitreihe der Populationsgröße einer Insektenpopulation nach einem einfachen Modell mit Startwert $x(0) = 5.2$.

Entwicklung der Population aus Abbildung 4.4 graphisch zu bestimmen. Wendet man die Transitionsfunktion f iterativ n -malig an, so zeigt

$$x(k+n) = f^n(x(k)) = f(f(f\dots f(x(k)))) \quad (4.3)$$

Zyklen der Länge 2. Allgemein wird eine Zahlenfolge $a, f(a), \dots, f^{n-1}(a), f^n(a) = a$ mit $f^i(a) \neq a$ für alle $i = 1, \dots, n-1$ als *Zyklus der Länge n* bezeichnet. Die Amplitude dieses Zweierzyklus hängt von dem Startwert ab. Im betrachteten Beispiel liegen die Punkte im Intervall $[6.0, 7.0]$. Das dynamische Verhalten ähnelt dem des dazugehörigen linguistischen Automaten, denn auch der Zustandsgraph des linguistischen Automaten zeigt einen Zyklus der Länge 2 für den Fall eines üppigen Nahrungsangebots, wie in Abbildung 4.1 zu sehen ist.

Wenn das rekurrente Fuzzy-System automatenähnlich ist, so zeigt es ein regelmäßiges Verhalten, wie durch den Ähnlichkeitssatz beschrieben wird. Wenn allerdings eine einzige Regel die Regelstetigkeitsbedingung verletzt, so kann diese Verletzung zu einem gänzlich anderen dynamischen Verhalten führen: Chaos. Dieses Phänomen wird im nächsten Kapitel genauer untersucht.

5 Chaos

Rekurrente Fuzzy-Systeme haben aufgrund ihres Ursprungs eine enge Beziehung zu endlichen Automaten. Deterministische, endliche Automaten können nur endlich viele Zustände annehmen. Bei konstant gehaltener Eingabe wird irgendwann ein Zustand erneut angenommen. Dann läuft der Automat einen Zyklus ab, in dem immer wieder die gleichen Zustandswerte angenommen werden. Rekurrente Fuzzy-Systeme nehmen dagegen Zustände aus einem normalerweise unendlichen Zustandsraum an. Sie sind auch bei konstant gehaltener Eingabe nicht gezwungen, ein zyklisches Verhalten zu zeigen.

In diesem Kapitel wird gezeigt, dass rekurrente Fuzzy-Systeme im Gegensatz zu Automaten chaotisches Verhalten aufweisen können. Zum Einstieg in dieses Problemfeld wird ein kleines Beispiel behandelt. Im Anschluss werden einige Grundlagen aus der Chaostheorie behandelt und hinreichende Kriterien für Chaos hergeleitet. Diese Kriterien werden dann auf rekurrente Fuzzy-Systeme angewandt [4]. Es zeigt sich, dass chaotisches Verhalten sich teilweise schon anhand der Betrachtung der Regeln nachweisen lässt.

5.1 Beispiel eines chaotischen rekurrenten Fuzzy-Systems

Ausgangspunkt für ein chaotisches rekurrentes Fuzzy-System ist wieder das Beispiel der Modellierung einer Insektenpopulation. Gegenüber der Regelbasis im letzten Kapitel wird eine einzige Regel geändert. Die neue Regelbasis ist in Abbildung 5.1 zu sehen. Jetzt

		aktuelle Populationsgröße L_j^x			
		klein	mittel	groß	
Nahrungs- angebot L_q^u	gering	klein	klein	klein	
	normal	mittel	mittel	klein	
	üppig	mittel	groß	klein	← zuvor mittel

Abbildung 5.1: Regelbasis eines chaotischen Modells der Populationsgröße einer Insektenpopulation in Matrixform.

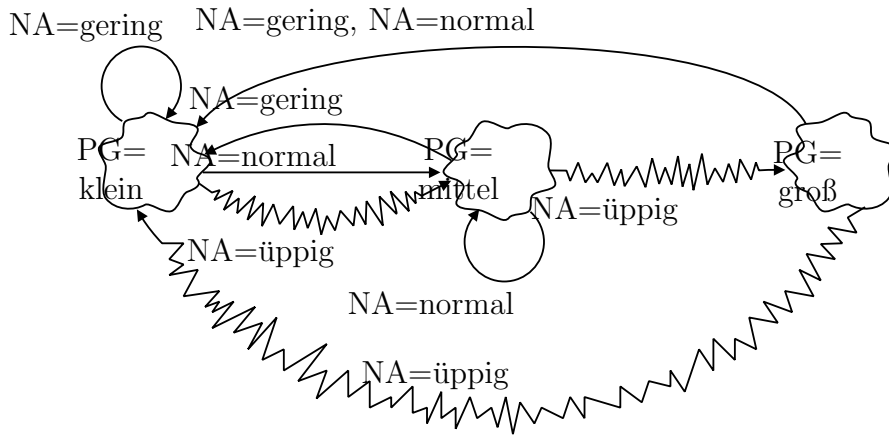


Abbildung 5.2: Zustandsgraph für das chaotische Modell der Größe einer Insektenpopulation mit verschiedenen Populationsgrößen (PG) als Zustand und Nahrungsangebot (NA) als Eingangsgröße.

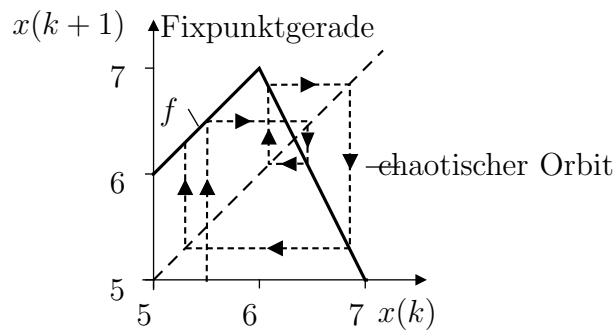


Abbildung 5.3: Transitionsfunktion eines chaotischen Modells einer Insektenpopulation bei üppigem Nahrungsangebot.

soll bei einem üppigen Nahrungsangebot auch bei einer großen Population eine so hohe intraspezifische Konkurrenz herrschen, dass die Populationsgröße auf einen kleinen Bestand im folgenden Jahr fällt. Das dynamische Verhalten ändert sich schlagartig: Chaos tritt auf. Dies wird durch die gezackten Pfeile im Zustandsgraphen in Abbildung 5.2 dargestellt.

Für die neue Transitionsfunktion f ergibt sich in diesem Fall für ein üppiges Nahrungsangebot nach Gleichung (3.13):

$$x(k+1) = f(x(k), u(k)) = 6 \cdot \mu_1^x(x(k)) + 7 \cdot \mu_2^x(x(k)) + 5 \cdot \mu_3^x(x(k)). \quad (5.1)$$

Diese Funktion ist graphisch in Abbildung 5.3 dargestellt. Folgt man dem Startwert bei $x(0) \approx 5.5$, so kann man kein regelmäßiges Verhalten erkennen. Das System hat eine chaotische Dynamik, wie die Simulation für zwei eng beieinanderliegende Startwerte in Abbildung 5.4 zeigt. Das auftretende Chaos ist Chaos im Sinne von Li und Yorke, welches wie folgt definiert ist [58, 48, 7]:

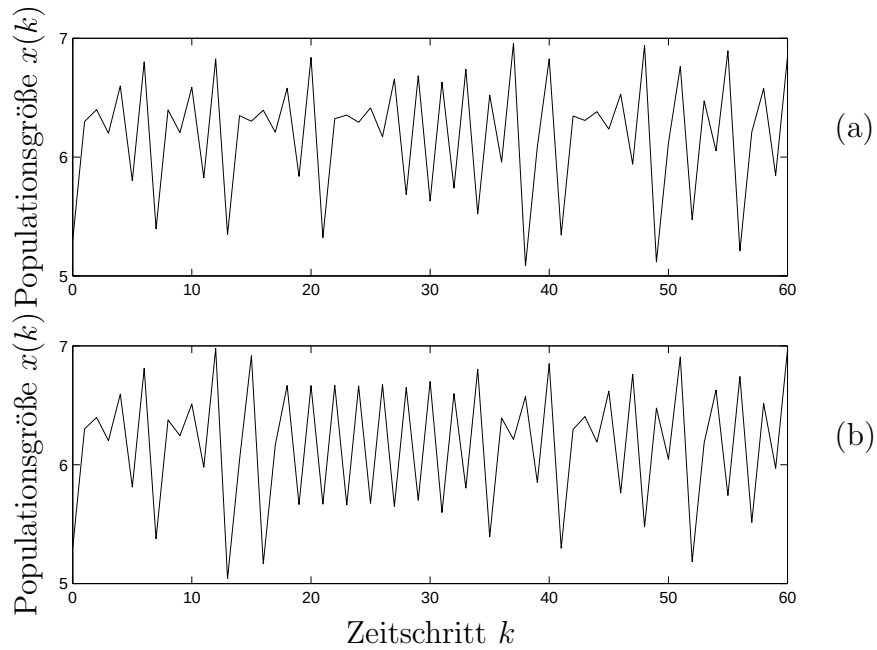


Abbildung 5.4: Zeitreihen der Populationsgröße, welche sich aus dem chaotischen Modell einer Insektenpopulation bei Anfangsgrößen von (a) $x(0) = 5.3001$ bzw. (b) $x(0) = 5.3007$ ergeben.

Definition 6 (Chaos im Sinne von Li und Yorke)

Gegeben sei eine stetige Funktion $f : I \rightarrow I$, wobei I ein endliches abgeschlossenes Intervall ist. Diese Funktion heißt chaotisch im Sinne von Li und Yorke, wenn es eine überabzählbare Menge $S \subseteq I$ gibt, so dass für jedes Paar $x_1, x_2 \in S$ mit $x_1 \neq x_2$ die Bedingungen $\limsup_{n \rightarrow \infty} |f^n(x_1) - f^n(x_2)| > 0$ und $\liminf_{n \rightarrow \infty} |f^n(x_1) - f^n(x_2)| = 0$ erfüllt sind¹.

Anschaulich beschreibt die Definition, dass es Trajektorien von Zuständen gibt, die sich in ihrer zeitlichen Entwicklung immer wieder beliebig nahe kommen, sich allerdings dann immer wieder voneinander trennen. Beliebige nahe beieinanderliegende Startwerte führen dabei zu sehr unterschiedlichen Dynamikausprägungen wie in Abbildung 5.4 zu sehen ist. Zu beachten ist, dass sich in diesem Beispiel die beiden Zeitverläufe nach 40 Iterationen wieder sehr nahe kommen und gleich wieder voneinander entfernen. Eine hinreichende und nach der originalen Definition notwendige Bedingung für Chaos im Sinne von Li und Yorke ist durch folgenden Satz [58] gegeben:

Satz 2 (Li und Yorke)

Sei $f : I \rightarrow I$ eine stetige Funktion auf einem kompakten Intervall I . Wenn es einen Punkt $a \in I$ mit $f^3(a) \leq a < f(a) < f^2(a)$ bzw. $f^3(a) \geq a > f(a) > f^2(a)$ gibt, dann besitzt f einen Dreierzyklus und ist chaotisch im Sinne von Li und Yorke.

¹In der ursprünglichen Definition von Li und Yorke [58] wird zusätzlich gefordert, dass es Zyklen mit jeder Periodenlänge geben muss. In der späteren Fachliteratur verwenden aber die meisten Autoren den Chaosbegriff nach Li und Yorke ohne diese Bedingung [55, 7].

Dieser Satz wird nun auf die Funktion f in Gleichung (5.1) aus dem betrachteten Beispiel angewendet. Wählt man $a = 5$, so gilt $f(a) = f(5) = 6$, $f^2(a) = f(6) = 7$, $f^3(a) = f(7) = 5$, was leicht anhand der Abbildung 5.3 abgelesen und mit Gleichung (5.1) überprüft werden kann. Die Bedingungen von Satz 2 sind damit erfüllt, und f ist chaotisch im Sinne von Li und Yorke. Die Existenz eines Dreierzyklus $a \rightarrow f(a) \rightarrow f^2(a) \rightarrow f^3(a) = a$ und damit die Erfüllung der Voraussetzung von Satz 2 lässt sich schon aus der Regelbasis in Abbildung 5.1 bzw. aus dem Zustandsgraphen in Abbildung 5.2 ersehen.

Dieses Beispiel begründet die Hoffnung, Kriterien für die Existenz chaotischen Verhaltens zu finden, die alleine anhand der Regelbasis bzw. des Zustandsgraphen auswertbar sind. Bevor diese Kriterien hergeleitet werden, wird im nachfolgenden Kapitel die dafür erforderliche Chaos-Theorie betrachtet.

5.2 Chaos in dynamischen Systemen

Chaos tritt in vielen Modellen und Anwendungsbereichen auf, beispielsweise in der Atomphysik [15], in elektronischen Schwingkreisen [103], ökonomischen Prozessen [76, 24] und Populationsmodellen [62].

Untersuchungen in diesem Feld [43, 25, 26, 48, 77, 7] nutzen dabei neben der Definition von Chaos nach Li und Yorke eine Vielzahl weiterer Definitionen. Geläufig ist auch die Definition von Chaos nach Devaney [20] in zeitdiskreten dynamischen Systemen:

Definition 7 (Chaos im Sinne von Devaney)

Eine stetige Funktion $f : I \rightarrow I$ ist chaotisch auf einer invarianten Menge $M \subseteq I$, wenn die Einschränkung $f|_M$ der Funktion auf die Menge M die folgenden Bedingungen erfüllt:

- (1) Sie ist topologisch transitiv, d.h. für jedes Paar von offenen Mengen $U, V \subseteq M$ gibt es einen Index $k > 0$, so dass $f^k(U) \cap V \neq \{\}$, wobei $f^k(U) = \{f^k(x) | x \in U\}$.
- (2) Ihre periodischen Punkte liegen dicht in M .
- (3) Sie ist sensitiv abhängig von den Anfangsbedingungen, d.h. es gibt einen Abstand $\delta > 0$, so dass es für alle $x \in M$ aus jeder Umgebung N von x einen Wert $y \in N$ gibt, so dass $|f^n(x) - f^n(y)| > \delta$ für einen Wert $n > 0$ erfüllt ist.

Wenn die Menge M nichttrivial ist, d.h. unendlich viele Elemente besitzt, so ist die dritte Bedingung redundant, da sie dann aus den Bedingungen 1 und 2 folgt [48]. Ist die Menge I ein abgeschlossenes Intervall, so ist sogar nur die erste Bedingung nötig [98, 7]. In diesem Fall sind die beiden Definitionen von Chaos fast deckungsgleich. Aus Chaos im Sinne von Devaney folgt dann schon Chaos im Sinne von Li und Yorke [7]. Eine feinere Unterscheidung und eine tiefere Einsicht bietet hier die Existenz bestimmter Periodenlängen von Zyklen. Diese wird durch das Theorem von Sarkovskii [79] angegeben. Dazu betrachte man folgende Ordnung in den natürlichen Zahlen:

$$3 \triangleright 5 \triangleright 7 \triangleright \dots \triangleright 2 \cdot 3 \triangleright 2 \cdot 5 \triangleright \dots \triangleright 2^2 \cdot 3 \triangleright 2^2 \cdot 5 \triangleright \dots \triangleright 2^3 \cdot 3 \triangleright 2^3 \cdot 5 \triangleright \dots \triangleright 2^3 \triangleright 2^2 \triangleright 2 \triangleright 1.$$

In dieser Ordnung treten alle ungeraden Zahlen von 3 bis ∞ auf, gefolgt von den gleichen Zahlen multipliziert mit 2, dann multipliziert mit 2^2 und so weiter. Am Ende erscheinen alle Potenzen von 2 in absteigender Reihenfolge, die mit $2^0 = 1$ enden. Dann gilt:

Satz 3 (Sarkovskii)

Eine stetige Funktion f , welche eine Abbildung eines Intervalls in sich selbst beschreibt und einen Zyklus der Länge k hat, besitzt auch einen Zyklus der Länge l , wenn $k \triangleright l$ ist.

Dieser Satz erlaubt es, stetige Funktionen $f : I \rightarrow I$ auf einem kompakten Intervall I nach ihrer maximalen Zyklenlänge gemäß dieser Ordnung zu klassifizieren. Dabei sind drei Funktionsklassen zu unterscheiden:

- (1) Funktionen f , die einen Zyklus der Länge n haben, die keine 2-er Potenz ist. Sie sind nach [7] sowohl chaotisch im Sinne von Devaney als auch chaotisch im Sinne von Li und Yorke [48, 7].
- (2) Funktionen f , deren Zyklen alle die Länge einer 2-er Potenz haben und unter einer Maximallänge liegen. Sie beschreiben dynamische Systeme, die für jeden Startwert gegen einen Grenzzyklus konvergieren, wie in der Erläuterung von Satz 26 im Anhang D beschrieben wird. Solche Funktionen sind nicht chaotisch.
- (3) Funktionen f , die für jede 2-er Potenz einen Zyklus dieser Länge haben und keine Zyklen besitzen, deren Länge keine 2-er Potenz ist. Diese Funktionen werden auch als vom Typ 2^∞ bezeichnet. Unter ihnen gibt es Funktionen, die chaotisch im Sinne von Li und Yorke sind [86], aber keine, die chaotisch im Sinne von Devaney sind [7]. Für weitere Informationen wird hier auf die Literatur [86, 59, 39, 7] verwiesen.

Im Weiteren werden die Funktionen betrachtet und als *chaotisch* bezeichnet, die chaotisch im Sinne von Devaney sind. Diese Funktionsklasse der eindimensionalen chaotischen Funktionen hat die angenehme Eigenschaft, dass sie eine offene Menge in der Menge der stetigen Funktionen bildet [14, 7]. Dies ist bei der Betrachtung von Funktionen wichtig, die stetig von einem oder mehreren Parametern abhängen. Ist eine solche Funktion für bestimmte Parameter chaotisch, so sind alle Funktionen chaotisch, deren Parameter nahe genug an den ursprünglichen Parametern liegen.

Nach Satz 2 folgen nun einige weitere hinreichende Kriterien für Chaos. So ergibt sich aus [7, Proposition 3.9] zusammen mit [7, Proposition 4.1] der folgende, oben erwähnte Satz:

Satz 4

Sei $f : I \rightarrow I$ eine stetige Funktion auf einem kompakten Intervall I . Wenn f einen Zyklus der Länge n hat, die keine 2-er Potenz ist, d.h. $n \neq 2^k, k \in \mathbb{N}_0$, dann ist f chaotisch.

Satz 2 lässt sich folgendermaßen erweitern:

Satz 5

Sei $f : I \rightarrow I$ eine stetige Funktion auf einem kompakten Intervall I . Wenn es einen Punkt $a \in I$ mit $f^n(a) \leq a < f(a) < f^2(a)$ bzw. mit $f^n(a) \geq a > f(a) > f^2(a)$ für ein $n \geq 3$ gibt, dann ist f chaotisch.

Beweis:

Nach Theorem 9 aus dem zweiten Kapitel von [14] folgt aus der Bedingung $f^n(a) \leq a < f(a)$ entweder, dass f einen periodischen Punkt mit einer ungeraden Periodenlänge hat und damit nach Satz 4 chaotisch ist oder dass $f^k(a) < f^j(a)$ für alle geraden Zahlen k und

ungeraden Zahlen j mit $0 \leq j \leq n$ und $0 \leq k \leq n$ gilt. Der zweite Fall ist aber wegen $f(a) < f^2(a)$ ausgeschlossen. $\square$

Schließlich kann man mit Hilfe von Satz 2 auch Chaoskriterien erhalten, die nicht nur einen einzigen Orbit betrachten. Dazu der folgende Satz:

Satz 6

Sei $f : I \rightarrow I$ eine stetige Funktion auf einem kompakten Intervall I . Wenn es Punkte $a, b \in I$ mit $f(b) \leq a < b < f(a) \leq f^2(a)$ bzw. mit $f(b) \geq a > b > f(a) \geq f^2(a)$ gibt, dann ist f chaotisch.

Beweis:

- 1) Es gelte $f(b) \leq a < b < f(a) \leq f^2(a)$. Der andere Fall kann analog behandelt werden.
- 2) Am Punkt b liegt der Funktionswert $f(b) < b$. Am Punkt $x_0 = f(a) > b$ liegt der Funktionswert $f(x_0) = f^2(a) \geq f(a) = x_0$, d.h. es gilt $f(x_0) \geq x_0$. Also schneidet die Funktion f die identische Abbildung $id(x) = x$ im Intervall $(b, x_0]$ und daher gibt es einen Fixpunkt $p_1 \in (b, x_0] = (b, f(a)]$.
- 3) Nach 2) und 1) gilt $f(a) \geq p_1 > b$ und $b > f(b)$, also $f(a) \geq p_1 > f(b)$. Daher gibt es gemäß dem Zwischenwertsatz im Intervall $[a, b)$ einen Punkt p_2 mit $f(p_2) = p_1$.
- 4) Mit 1), 3) und 2) folgt $f(b) \leq a$, $a \leq p_2 < b$ und $b < p_1 = f(p_1)$, also $f(b) \leq p_2 < f(p_1)$. Gemäß dem Zwischenwertsatz gibt es dann im Intervall $[b, p_1)$ einen Punkt p_3 mit $f(p_3) = p_2$.
- 5) Da $p_2 \in [a, b)$ und $p_3 \in [b, p_1)$, gilt $p_2 < p_3$. Mit $f(p_3) = p_2$ aus 4) ergibt sich dann $f(p_3) < p_3$.
- 6) Nach 5) und 4) gilt $f(p_3) < p_3$ und $p_3 < p_1 = f(p_1)$, also $f(p_3) < p_3 < f(p_1)$. Daher gibt es wiederum gemäß dem Zwischenwertsatz im Intervall (p_3, p_1) einen Punkt p_4 mit $f(p_4) = p_3$.
- 7) Nach 5) und 6) gilt die Ungleichungskette $p_1 > p_4 > p_3 > p_2$. Des Weiteren ergeben sich aus $p_1 = f(p_2)$, $p_2 = f(p_3)$ und $p_3 = f(p_4)$ die Beziehungen $p_1 = f^3(p_4)$, $p_2 = f^2(p_4)$ und $p_3 = f(p_4)$. Setzt man die Beziehungen in obige Ungleichungskette ein, so erhält man $f^3(p_4) > p_4 > f(p_4) > f^2(p_4)$. Damit sind die Voraussetzungen von Satz 2 erfüllt und f ist chaotisch. $\square$

Obige Sätze werden im nächsten Abschnitt auf rekurrente Fuzzy-Systeme angewendet. Das Resultat sind Sätze, anhand derer man alleine aufgrund des Zustandsgraphen bzw. der Regelbasis auf Chaos in einem rekurrenten Fuzzy-System schließen kann.

5.3 Chaoskriterien für rekurrente Fuzzy-Systeme

Wie im letzten Abschnitt dargestellt, resultiert aus der Existenz von bestimmten Zyklen und bestimmten Orbits Chaos in eindimensionalen dynamischen Systemen. Ziel dieses Kapitels ist, diese Ergebnisse auf rekurrente Fuzzy-Systeme zu übertragen und zwar so, dass Chaos in rekurrenten Fuzzy-Systemen allein anhand des Zustandsgraphen bzw. der

Regelbasis feststellbar ist. Um griffige Kriterien für das Auftreten von Chaos formulieren zu können, sind vorab einige Begriffe zu definieren:

Definition 8 (Linguistische Orbits und Kernpositionsorbits)

Für einen festgelegten linguistischen Eingabevektor $\mathbf{L}_q^u$ erzeugt ein linguistischer Automat eine Folge von linguistischen Zustandswerten $L(0), L(1), \dots$, wobei er $L(k)$ auf $L(k+1)$ für $k = 0, 1, 2, \dots$ abbildet. Eine solche Folge von linguistischen Zustandswerten wird als linguistischer Orbit bezeichnet.

Für einen festgelegten Eingabevektor auf einem Kernpositionsvektor $\mathbf{s}_q^u$ erzeugt ein rekurrentes Fuzzy-System eine Folge von Kernpositionen $s(0), s(1), \dots$ durch $s(k+1) = f(s(k), \mathbf{s}_q^u)$ mit $k = 0, 1, 2, \dots$. Eine solche Folge von Kernpositionen wird als Kernpositionsorbit bezeichnet.

Im Beispielsystem des einfachen Modells der Insektenpopulation erzeugt der linguistische Automat bei einer linguistischen Eingabe von $L_2^u = \text{normal}$, ausgehend von dem Zustandswert $L(0) = L_3^x = \text{groß}$, den linguistischen Orbit $L(0) = \text{groß}$, $L(1) = \text{klein}$, $L(2) = \text{mittel}$, $L(3) = \text{mittel}$, $\dots$, wie leicht anhand des Zustandsgraphen in Abbildung 4.1 oder der Regelbasis in Abbildung 3.1 zu sehen ist. Entsprechend erzeugt das dazugehörige rekurrente Fuzzy-System für einen Eingabewert $u = s_2^u$, ausgehend vom Zustandswert $s(0) = s_3^x = 7$, den Kernpositionsorbit $s(0) = 7$, $s(1) = 5$, $s(2) = 6$, $s(3) = 6$, $\dots$

Definition 9 (Linguistischer Zyklus und Kernpositionszyklus)

Ein linguistischer Orbit, $L(0), L(1), \dots$, dessen Elemente $L(k) \neq L(0)$ mit $k = 1, 2, \dots, n-1$ und $L(n) = L(0)$ erfüllen, wird als linguistischer Zyklus der Länge n bezeichnet.

Ein Kernpositionsorbit $s(0), s(1), \dots$, dessen Elemente $s(k) \neq s(0)$ mit $k = 1, 2, \dots, n-1$ und $s(n) = s(0)$ erfüllen, wird als Kernpositionszyklus der Länge n bezeichnet.

Im Fall eines linguistischen Eingabewertes von $L_3^u = \text{üppig}$ erzeugt der linguistische Automat des chaotischen Modells der Insektenpopulation, ausgehend von $L(0) = L_1^x = \text{klein}$, einen linguistischen Zyklus der Länge drei, denn $L(0) = \text{klein}$, $L(1) = \text{mittel}$, $L(2) = \text{groß}$, $L(3) = \text{klein}$. Es sei angemerkt, dass jeder linguistische Orbit in einen linguistischen Zyklus mündet, da der linguistische Automat auf endlich vielen linguistischen Werten operiert.

Definition 10 (Nachbarschaft eines Kernpositionsorbits)

Gegeben sei ein Kernpositionsorbit $s(0), s(1), \dots$ für einen Eingabevektor auf einem Kernpositionsvektor $\mathbf{s}_q^u$. Eine offene Menge $N \subseteq X \times U$ im Produktraum von Zustandsraum X und Eingaberaum U , die alle Vektoren $(s(k), \mathbf{s}_q^u)$ mit $k = 0, 1, 2, \dots$ enthält, wird als Nachbarschaft dieses Kernpositionsorbits bezeichnet.

Definition 11 (Chaos in rekurrenten Fuzzy-Systemen)

Ein rekurrentes Fuzzy-System wird als chaotisch in einer Menge $N \subseteq X \times U$ bezeichnet, wenn für jeden Eingabevektor $\mathbf{u}$ im Falle einer nichtleeren Menge $\tilde{X}(\mathbf{u}) = \{x \in X \mid (x, \mathbf{u}) \in N\}$ die Transitionsfunktion $f(x, \mathbf{u})$ dieses rekurrenten Fuzzy-Systems chaotisch auf einer Teilmenge von $\tilde{X}(\mathbf{u})$ ist.

Lemma 1 besagt, dass jeder linguistische Automat isomorph zu einem rekurrenten Fuzzy-System ist, wenn dieses auf Kernpositionsvektoren für die Zustandsvektoren und Eingabevektoren eingeschränkt ist. Deshalb entspricht auch jedem linguistischen Orbit $L(0)$,

$L(1), \dots$ für einen festen Eingabevektor $\mathbf{L}_q^u$ ein Kernpositionsorbit $s(0), s(1), \dots$ für einen Eingabevektor an einem Kernpositionsvektor $\mathbf{s}_q^u$.

Des Weiteren induzieren die Kernpositionen $s_1^x < s_2^x < s_3^x < \dots$ auch eine Ordnungsrelation auf den entsprechenden linguistischen Werten $L_1^x < L_2^x < L_3^x < \dots$. Damit lassen sich die Sätze 2 bis 6 direkt auf linguistische Werte in der Regelbasis von rekurrenten Fuzzy-Systemen anwenden. So erhält man aus Satz 4 und Satz 5 die folgenden Sätze 7 und 8 für linguistische Zyklen:

Satz 7

Wenn ein eindimensionales, stetiges rekurrentes Fuzzy-System einen linguistischen Zyklus der Länge $l \neq 2^n$ mit $n \in \mathbb{N}_0$ besitzt, dann ist das rekurrente Fuzzy-System in einer Umgebung des entsprechenden Kernpositionszyklus chaotisch.

Beweis:

Wenn der Eingabevektor $\mathbf{u}$ des rekurrenten Fuzzy-Systems auf einem Kernpositionsvektor $\mathbf{s}_q^u$ des linguistischen Eingabevektors $\mathbf{L}_q^u$ liegt, für den er einen linguistischen Zyklus der Länge $l \neq 2^n$ mit $n \in \mathbb{N}_0$ hat, so hat der Kernpositionszyklus, der diesem linguistischen Zyklus entspricht, die gleiche Länge. Dann ist die Transitionsfunktion $f(x, \mathbf{u})$ chaotisch nach Satz 4. Da die Menge der chaotischen Funktionen offen in der Menge der stetigen Funktionen ist und die Funktionen $f(x, \mathbf{u})$ stetig von den Parametern u_p abhängen, gibt es eine Umgebung um den Vektor $\mathbf{u} = \mathbf{s}_q^u$, so dass die Funktionen $f(x, \mathbf{u})$ auch chaotisch für alle Vektoren $\mathbf{u}$ aus dieser Umgebung sind. $\square$

Satz 8

Wenn ein eindimensionales, stetiges rekurrentes Fuzzy-System einen linguistischen Zyklus $L(0), L(1), L(2), \dots$ besitzt, dessen linguistische Werte die Ungleichungskette $L(0) < L(1) < L(2)$ bzw. $L(0) > L(1) > L(2)$ erfüllen, so ist das rekurrente Fuzzy-System in einer Umgebung des entsprechenden Kernpositionszyklus chaotisch.

Der Beweis von Satz 8 verläuft analog zu dem von Satz 7 mit dem Unterschied, dass Satz 5 anstatt Satz 4 benutzt wird.

Im Beispielmmodell der Populationsentwicklung kann mittels beider Sätze im Falle eines üppigen Nahrungsangebots Chaos nachgewiesen werden. Der gezackt eingezeichnete linguistische Zyklus in Abbildung 5.2 hat sowohl eine Länge, die keine Zweierpotenz ist, als auch drei aufsteigende linguistische Zustandswerte. Damit ist die Populationsentwicklung auch dann chaotisch, wenn der Eingabewert des Nahrungsangebots nicht exakt auf dem Kernpositionsvektor s_3^u des linguistischen Wertes groß, sondern nur genügend nahe daran liegt.

Betrachtet man linguistische Orbits, so ergeben sich aus den Sätzen 5 und 6 die nachfolgenden Sätze 9 und 10, deren Beweis auch analog des Beweises von Satz 7 verlaufen:

Satz 9

Wenn es in einem eindimensionalen, stetigen rekurrenten Fuzzy-System einen linguistischen Orbit $L(0), L(1), L(2), \dots, L(n), \dots$ gibt, dessen linguistische Werte die Ungleichungskette $L(n) \leq L(0) < L(1) < L(2)$ bzw. $L(n) \geq L(0) > L(1) > L(2)$ erfüllen, dann ist das rekurrente Fuzzy-System in einer Umgebung des entsprechenden Kernpositionszyklus chaotisch.

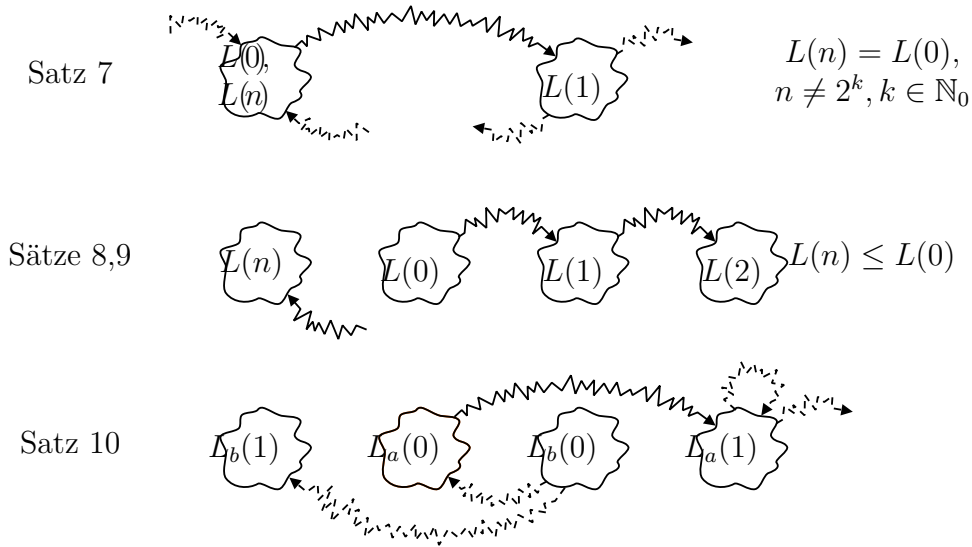


Abbildung 5.5: Graphischer Überblick über die Chaostkriterien. Die gezackten, gestrichelten Linien deuten verschiedene mögliche Transitionen an. Weitere graphische Kriterien erhält man durch Spiegelung der Graphen an einer vertikalen Achse.

Satz 10

Wenn ein eindimensionales, stetiges rekurrentes Fuzzy-System für einen linguistischen Eingabevektor $\mathbf{L}_q^u$ zwei linguistische Orbits besitzt, deren linguistische Werte $L_a(0)$, $L_a(1)$, $L_a(2)$, des ersten Orbits und $L_b(0)$, $L_b(1)$ des zweiten Orbits die Ungleichungskette $L_b(1) \leq L_a(0) < L_b(0) < L_a(1) \leq L_a(2)$ bzw. $L_b(1) \geq L_a(0) > L_b(0) > L_a(1) \geq L_a(2)$ erfüllen, so ist das rekurrente Fuzzy-System in einer Umgebung des entsprechenden Kernpositionszyklus chaotisch.

Damit sind mehrere hinreichende Kriterien für die Untersuchung von Chaos in eindimensionalen rekurrenten Fuzzy-Systemen gegeben. Diese Kriterien können auch graphisch anhand des Zustandsgraphen ausgewertet werden. Dazu ordnet man die linguistischen Zustände in steigender oder fallender Reihenfolge. Zeigt der Zustandsgraph dann Teilmuster, wie sie in Abbildung 5.5 zu sehen sind, so ist das entsprechende rekurrente Fuzzy-System chaotisch.

5.4 Grenzen der Chaostkriterien

Wie gezeigt, ist ein rekurrentes Fuzzy-System, das für Eingabevektoren $\mathbf{u}$ auf Kernpositionen $\mathbf{s}_q^u$ chaotisch ist, auch in einer Umgebung um die Kernposition chaotisch. Wie groß diese Umgebung ist, lässt sich ohne weiteres jedoch nicht feststellen. Dies wird in dem Beispiel in Abbildung 5.6 deutlich. Die Abbildung zeigt Zugehörigkeitsfunktionen (siehe Abb. 5.6 (a)) einer Zustandsgröße x und Transitionsfunktionen f für verschiedene Eingabewerte (siehe Abb. 5.6 (b),(c),(d)). Nach Satz 10 ist das rekurrente Fuzzy-System für Eingaben mit $\mu_1^u(u) = 1$ (siehe Abb. 5.6 (b)) bzw. $\mu_2^u(u) = 1$ (siehe Abb. 5.6 (d)) chaotisch.

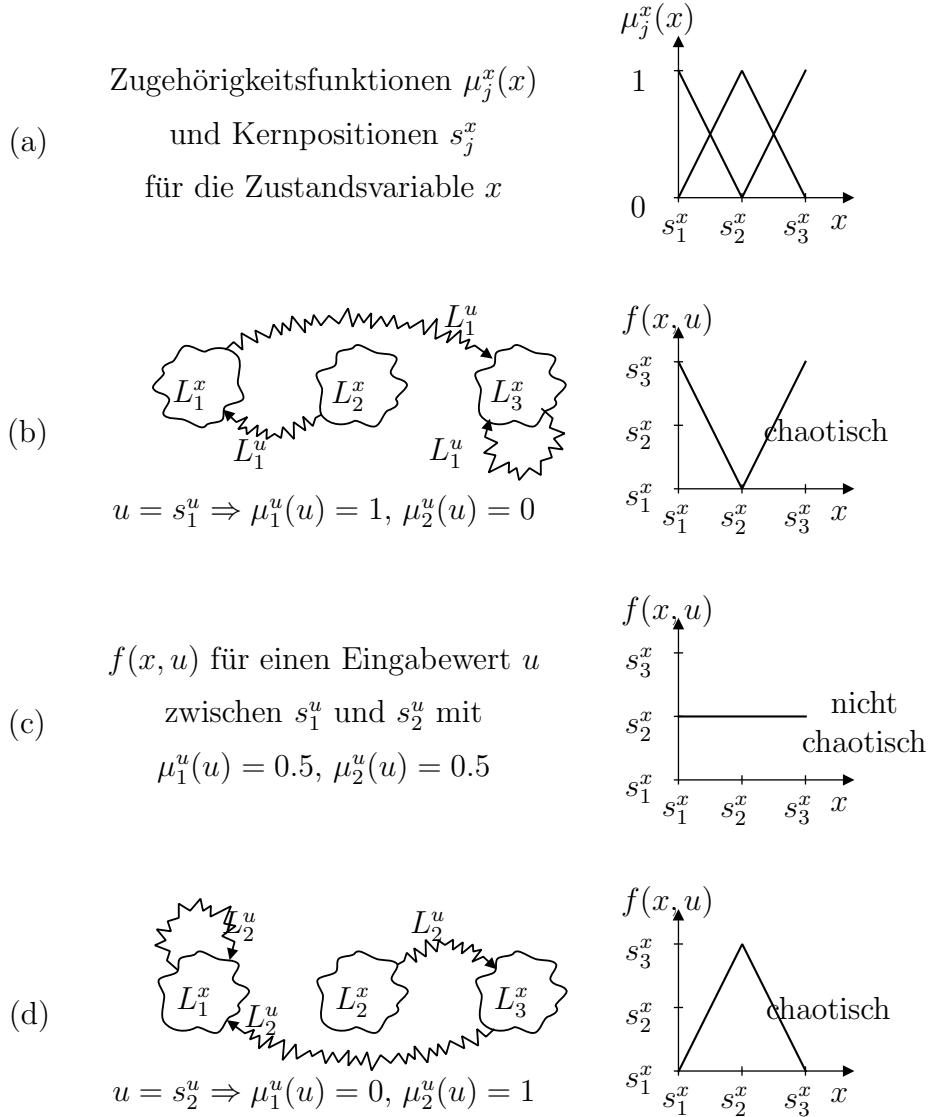


Abbildung 5.6: Ein rekurrentes Fuzzy-System mit den in (a) gezeigten Zugehörigkeitsfunktionen und Kernpositionen für die Zustandsgröße x und einer entsprechenden Regelbasis zeigt, dass die Transitionsfunktion $f(x, u)$, die für Eingabewerte u an Kernpositionen chaotisch ist (siehe (b),(d)), für dazwischenliegende Eingabewerte nicht unbedingt chaotisch sein muss (siehe (c)).

Gelten die Regeln je zur Hälfte, d.h. ist $\mu_1^u(u) = 0.5$ und $\mu_2^u(u) = 0.5$, so ergibt sich eine konstante und damit nicht chaotische Transitionsfunktion (siehe Abb. 5.6 (c)).

Andererseits kann ein rekurrentes Fuzzy-System, das für Eingabevektoren an Kernpositionsvektoren nicht chaotisch ist, trotzdem für andere Eingabewerte chaotisch sein. Dies ist im Beispiel in Abbildung 5.7 zu sehen. Die Abbildung zeigt Zugehörigkeitsfunktionen² (siehe Abb. 5.7 (a)) einer Zustandsgröße x und Transitionsfunktionen f für verschiedene

²Im Beispiel wurden Trapeze als Zugehörigkeitsfunktionen gewählt. Die gleichen Transitionsfunktionen lassen sich auch mit Hilfe von fünf Zuständen und äquidistanten Dreiecksfunktionen erzeugen.

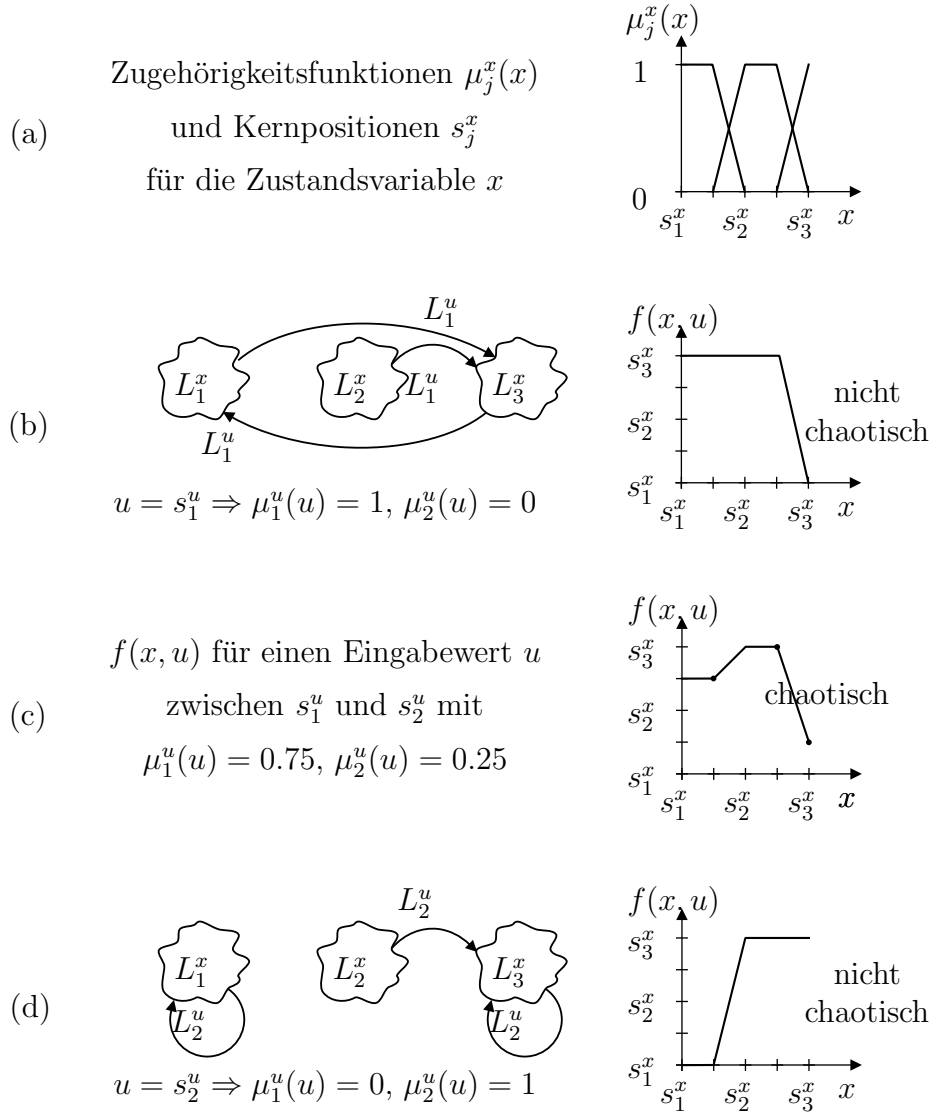


Abbildung 5.7: Ein rekurrentes Fuzzy-System mit den in (a) gezeigten Zugehörigkeitsfunktionen und Kernpositionen für die Zustandsgröße x und einer entsprechenden Regelbasis zeigt, dass die Transitionsfunktion $f(x, u)$, die für Eingabewerte u an Kernpositionen nicht chaotisch ist (siehe (b),(d)), für dazwischenliegende Eingabewerte chaotisch sein kann (siehe (c)).

Eingabewerte (siehe Abb. 5.7 (b),(c),(d)). In den beiden Fällen, in denen die Eingabewerte auf Kernpositionen liegen, d.h. für $\mu_1^u(u) = 1$ (siehe Abb. 5.7 (b)) bzw. $\mu_2^u(u) = 1$ (siehe Abb. 5.7 (d)) ist die Transitionsfunktion monoton. Wie weiter unten gezeigt wird, sind diese Transitionsfunktionen daher nicht chaotisch. Im Übergangsbereich, und im Speziellen für $\mu_1^u(u) = 0.75$ und $\mu_2^u(u) = 0.25$, weist die Transitionsfunktion einen Zyklus der Länge 3 auf (siehe Abb. 5.7 (c)) und ist daher nach Satz 2 chaotisch. Dieses Beispiel illustriert, dass obige Sätze nur hinreichend sind.

Es ist deshalb hilfreich auch Aussagen zu haben, die Chaos für alle Eingabewerte ausschließen. Dazu werden im Folgenden zwei Klassen von eindimensionalen rekurrenten Fuzzy-

Systemen untersucht: zum einen *monotone* Systeme und zum anderen regelstetige bzw. automatenähnliche Systeme.

Eindimensionale, monotone rekurrente Fuzzy-Systeme zeichnen sich dadurch aus, dass bei allen linguistischen Eingabevektoren je zwei linguistische Zustandswerte $L_a(0) < L_b(0)$ auf linguistische Zustandswerte $L_a(1)$ und $L_b(1)$ abgebildet werden, die die Bedingung $L_a(1) \leq L_b(1)$ im Falle monoton steigender und die Bedingung $L_a(1) \geq L_b(1)$ im Falle monoton fallender rekurrenter Fuzzy-Systeme erfüllen. Es gilt

Satz 11

Eindimensionale, monotone rekurrente Fuzzy-Systeme sind für keinen Eingabevektor chaotisch. Jedes eindimensionale, monoton steigende (bzw. fallende) rekurrente Fuzzy-System konvergiert bei festgelegten Eingangs- und Startwerten gegen einen Endwert (bzw. gegen einen Endwert oder einen Grenzyklus der Länge 2).

Beweis:

Nach Gleichung (3.13) ist die Transitionsfunktion f für alle Eingabevektoren monoton steigend oder monoton fallend.

Monoton steigende Funktionen $f(x)$ haben keine Zyklen der Länge 2, da andernfalls zwei Punkte $x_1 = f(x_2)$ und $x_2 = f(x_1)$ eines solchen Zyklus mit $x_1 < x_2$ auch die folgende widersprüchliche Ungleichung erfüllen müssten: $x_2 = f(x_1) \leq f(x_2) = x_1$. Nach [39] konvergiert das System für jeden Startwert gegen einen Endwert und somit ist die Funktion nicht chaotisch.

Bei monoton fallenden Funktionen $f(x)$ ist deren zweifache Iteration, d.h. $f^2(x)$, monoton steigend und hat daher keinen Zyklus der Länge 2. Dann hat eine monoton fallende Funktion $f(x)$ keinen Zyklus der Länge 4, denn ein Punkt aus einem solchen Zyklus wäre zugleich ein Punkt aus einem Zyklus der Länge 2 von $f^2(x)$. Nach [39] konvergiert das System für jeden festgelegten Startwert entweder gegen einen Endwert oder einen Zyklus der Länge 2. Die maximale Zyklenlänge nach der Ordnung von Sarkovskii ist also 2 und damit eine Zweierpotenz. Folglich ist die Transitionsfunktion f für keinen Eingabevektor chaotisch, wie in Abschnitt 5.2 besprochen wurde. $\square$

Beispielsweise ist das chaotische Modell der Populationsentwicklung für ein geringes Nahrungsangebot monoton steigend und konvergiert deshalb nach Satz 11 gegen einen Endwert. Des Weiteren ist das Modell für ein geringes, ein normales und ein dazwischenliegendes Nahrungsangebot monoton fallend. Nach Satz 11 sind für solche Eingabewerte also keine chaotischen Zeitreihen zu erwarten. Tatsächlich konvergiert das Modell in diesen Fällen für jeden Startwert gegen einen Endwert, obwohl es nach Satz 11 auch für manche Eingabewerte gegen Grenzyklen der Länge 2 konvergieren dürfte.

Wie gesehen, zeigt sich bei eindimensionalen, monotonen rekurrenten Fuzzy-Systemen also kein Chaos. Ein ähnliches Resultat könnte man auch für eindimensionale, regelstetige und damit eindimensionale, automatenähnliche rekurrente Fuzzy-Systeme vermuten, da sich aufgrund der Automatenähnlichkeit die Dynamik eines solchen rekurrenten Fuzzy-Systems durch die eines Automaten grob beschreiben lässt und in endlichen Automaten kein Chaos auftreten kann. Umso erstaunlicher ist, dass es auch eindimensionale, chaotische, automatenähnliche rekurrente Fuzzy-Systeme gibt, wie das folgende Beispiel belegt:

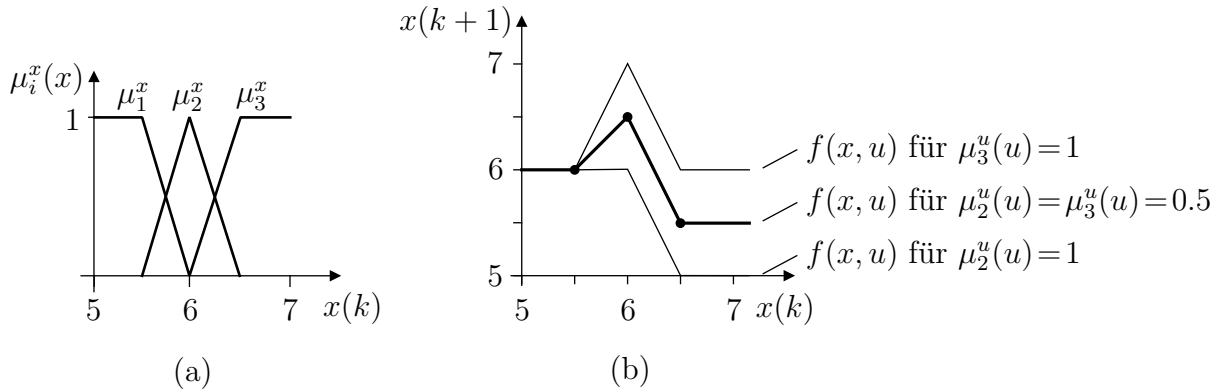


Abbildung 5.8: (a) Trapezförmige Zugehörigkeitsfunktionen, die mit der Regelbasis aus Abbildung 3.1 zu einer (b) chaotischen Transitionsfunktion eines eindimensionalen, chaotischen, automatenähnlichen rekurrenten Fuzzy-Systems führen.

Das einfache Modell der Populationsdynamik ist regelstetig, wie man aus der Regelbasis in Abbildung 3.1 ablesen kann. Nach Satz 1 ist jedes darauf aufbauende rekurrente Fuzzy-System auch automatenähnlich. Wählt man statt Dreiecksfunktionen die in Abbildung 5.8(a) gezeigten trapezförmigen Zugehörigkeitsfunktionen, so ergibt sich für $\mu_2^u(u) = \mu_3^u(u) = 0.5$ die danebenstehende Transitionsfunktion. Die Transitionsfunktion hat einen Dreierzyklus, der in Abbildung 5.8(b) durch kleine Punkte angedeutet wurde. Damit ist nach Satz 2 dieses eindimensionale automatenähnliche rekurrente Fuzzy-System auch chaotisch.

Diesen scheinbaren Widerspruch von Chaos auf der einen Seite und Regelmäßigkeit durch die Automatenähnlichkeit auf der anderen Seite lässt sich wie folgt auflösen: Die Automatenähnlichkeit garantiert, dass die Dynamik auf einer Skala, die durch die Kernpositionen angegeben wird, durch die Dynamik in einem Automaten beschreibbar ist. Chaos dagegen kann auch auf einer sehr kleinen, dicht gedrängten Punktmenge auftreten, die bei der groben Betrachtung nicht mehr aufgelöst wird. Tatsächlich findet die chaotische Dynamik in obigem Beispiel im Intervall $[5.5, 6.5]$ statt und somit sind zum Beispiel auch alle diese auftretenden Zustandswerte mit der Kernposition $s_2^x = 6$ verbunden.

Es lässt sich aber zeigen, dass bei geeigneter Wahl von Zugehörigkeitsfunktionen eindimensionale, automatenähnliche rekurrente Fuzzy-Systeme nicht chaotisch sind. Diese Aussage findet sich in Satz 21 am Ende des nächsten Kapitels, für dessen Beweis noch einige Vorüberlegungen nötig sind.

Offensichtlich spielt die Form der Zugehörigkeitsfunktionen eine entscheidende Rolle für manche dynamischen Eigenschaften des Systems. Dies trifft nicht nur für die Untersuchung nach Chaos, sondern auch für die Untersuchung von Ruhelagen zu, die im nächsten Kapitel betrachtet werden.

6 Ruhelagen

6.1 Standardisierte, rekurrente Fuzzy-Systeme

Bei statischen Fuzzy-Systemen ist die Wahl der Zugehörigkeitsfunktionen im Allgemeinen nicht von großer Bedeutung. Ein Wechsel zu anderen Zugehörigkeitsfunktionen bewirkt in der Regel nur eine kleine Veränderung im statischen Eingabe-/Ausgabeverhalten. Bei dynamischen Systemen und im Speziellen bei rekurrenten Fuzzy-Systemen können diese kleinen Abweichungen allerdings signifikante Folgen für das dynamische Verhalten haben.

Dieser Effekt lässt sich gut an folgendem sehr einfachen rekurrenten Fuzzy-System veranschaulichen. Es modelliert eine "verbale identische Abbildung" durch die folgenden Regeln:

$$\begin{aligned} \text{Wenn } x(k) &= \text{klein,} & \text{dann } x(k+1) &= \text{klein.} \\ \text{Wenn } x(k) &= \text{groß,} & \text{dann } x(k+1) &= \text{groß.} \end{aligned} \quad (6.1)$$

Die Regelbasis legt fest, dass das System seinen linguistischen Zustand nicht ändern soll.

Man könnte erwarten, dass ein rekurrentes Fuzzy-System, basierend auf diesen Regeln, auch eine mathematisch identische Abbildung darstellt. Für die Untersuchung werden die Kernpositionen auf $s_1^x = 0$ für $L_1^x = \text{klein}$ und $s_2^x = 1$ für $L_2^x = \text{groß}$ gesetzt. Als Zugehörigkeitsfunktionen werden nun verschiedene Funktionstypen herangezogen. Dabei reicht die Festlegung von $\mu_2^x(x)$ aus, da sich die andere aufgrund der Normierungsbedingung nach Gleichung (3.12) aus $\mu_1^x(x) = 1 - \mu_2^x(x)$ im Intervall $[0, 1]$ ergibt. Die Transitionsfunktion ist nach Gleichung (3.13) wegen der speziellen Wahl der Kernpositionen durch $f(x) = \mu_2^x(x)$ gegeben.

Abbildung 6.1 zeigt vier Transitionsfunktionen, die sich aus der Flanke einer Dreiecksfunktion ($f(x) = x$), einer Sinus-Kurve ($f(x) = \sin^2(\pi/2 \cdot x)$) und Parabeln ($f(x) = x^2$ und $f(x) = 2x - x^2$) im Intervall $[0, 1]$ ergeben. Darunter ist die Wirkung der Transitionsfunktionen auf die Startwerte x angegeben, wenn die Zahl der Iterationen k der Funktion f nach Gleichung (3.13) gegen unendlich geht.

Die Fixpunkte der Abbildungen lassen sich leicht als Schnittpunkte mit der eingezeichneten Winkelhalbierenden bestimmen, die sich aus der identischen Abbildung ($id(x) = x$) ergibt. Im Falle der Dreiecksfunktionen sind alle Startwerte auch Fixpunkte. Bei der Wahl der Sinuskurve als Zugehörigkeitsfunktion ergibt sich ein instabiler Fixpunkt bei $x = 0.5$ und asymptotisch stabile Fixpunkte bei $x = 0$ und $x = 1$. Besonders im Falle der Parabeln wird deutlich, welchen Effekt die Wahl der Zugehörigkeitsfunktionen haben kann: Im Falle $\mu_2^x(x) = f(x) = x^2$ werden alle Startwerte – mit Ausnahme von $x = 1$ – im Laufe der Zeit

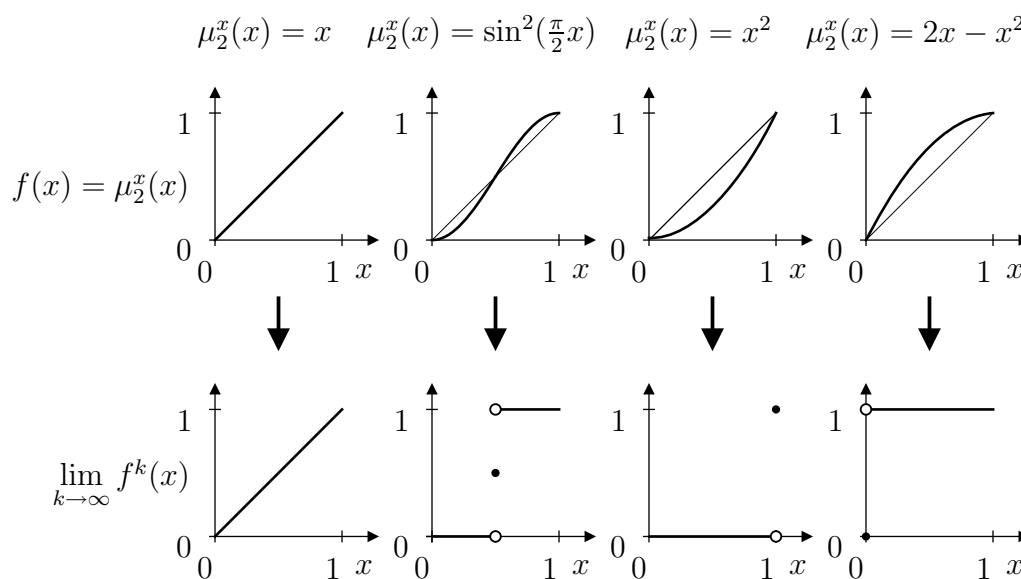


Abbildung 6.1: Verschiedene Transitionsfunktionen, $f(x)$, für die Modellierung der "linguistischen identischen Abbildung" und ihr Verhalten bei einer hohen Zahl von Iterationen, d.h. für $k \rightarrow \infty$.

gegen den Wert $x = 0$ streben, der dem linguistischen Wert "klein" entspricht. Dies gilt aber auch für einen Startwert, z.B. $x = 0.999$, der sehr nahe am Wert $x = 1$ liegt, welcher dem linguistischen Wert "groß" entspricht. Dieses Verhalten ist nicht aus der Regelbasis ersichtlich. Nur bei der Wahl der Dreiecksfunktion als Zugehörigkeitsfunktion wird auch auf quantitativer Ebene die identische Abbildung realisiert, so wie man es gemäß der Regelbasis erwartet. Diese Aussage wird durch Lemma 12 im Anhang B bestätigt.

Wie gesehen, tritt bei manchen Zugehörigkeitsfunktionstypen eine versteckte Dynamik des rekurrenten Fuzzy-Systems auf, die nicht aus der Regelbasis ersichtlich ist. Dies erschwert und behindert im Fall nichtdreiecksförmiger Zugehörigkeitsfunktionen Schlussfolgerungen auf das dynamische Verhalten des rekurrenten Fuzzy-Systems auf der quantitativen Ebene, ausgehend von der Beschreibung des rekurrenten Fuzzy-Systems durch seine Regelbasis auf der qualitativen Ebene. Um eine solche versteckte Dynamik zu vermeiden, wird im Weiteren eine spezielle Klasse der rekurrenten Fuzzy-Systeme behandelt, die standardisierten rekurrenten Fuzzy-Systeme. Sie sind wie folgt definiert:

Definition 12

Ein rekurrentes Fuzzy-System heißt standardisiert, wenn als Zugehörigkeitsfunktionen in der Defuzzifizierung der Zustandsvariablen äquidistante Singletonpositionen und in der Fuzzifizierung der Zustandsvariablen äquidistante Dreiecke verwendet werden.

Die Wahl äquidistanter Kernpositionen in Definition 12 ist keine gravierende Einschränkung. Hat man ein rekurrentes Fuzzy-System, welches mit nichtäquidistanten Kernpositionen ausgestattet ist, so kann man es in ein rekurrentes Fuzzy-System mit äquidistanten Kernpositionen transformieren: Intern wird dann mit äquidistanten Kernpositionen gearbeitet, die durch eine Ausgangsfunktion g auf die ursprünglichen nichtäquidistanten

Kernpositionen abgebildet werden. Strenggenommen ist dies zwar keine Äquivalenzumformung, aber sowohl die Regelbasis als auch die Wertebereiche beider Systeme stimmen dadurch überein.

Wie im nächsten Abschnitt gezeigt wird, erhält man für standardisierte rekurrente Fuzzy-Systeme weitreichende Aussagen über Ruhelagen und deren Stabilität, die im Allgemeinen nicht für rekurrente Fuzzy-Systeme mit anderen Zugehörigkeitsfunktionen getroffen werden können [41].

6.2 Ruhelagen und Nichtexpansivität

In diesem Abschnitt werden die Grundlagen zur Untersuchung von Ruhelagen in rekurrenten Fuzzy-Systemen behandelt. Dazu werden zuerst stetige rekurrente Fuzzy-Systeme betrachtet. Im zweiten Teil wird dann die spezielle Struktur von standardisierten rekurrenten Fuzzy-Systemen ausgenutzt.

Die Transitionsfunktion $\mathbf{f}$ bildet für jeden festen Eingabevektor $\mathbf{u}$ den Zustandsraum X in sich selbst ab. Der Zustandsraum kann als eine kompakte Menge betrachtet werden, wie am Ende des Kapitels 3 erläutert wurde. Bei stetigen rekurrenten Fuzzy-Systemen beschreibt die Transitionsfunktion für jeden festen Eingabevektor $\mathbf{u}$ also eine stetige Abbildung einer kompakten Menge in sich selbst. Damit ist der Fixpunktsatz von Brouwer [85] bzw. der Fixpunktsatz von Schauder [32] anwendbar. Diese Sätze stellen sicher, dass die Transitionsfunktion $\mathbf{f}$ eines stetigen rekurrenten Fuzzy-Systems für jeden Eingabevektor $\mathbf{u}$ mindestens einen Fixpunkt im Zustandsraum X hat.

Ein Fixpunkt $\mathbf{x}$ erfüllt nach seiner Definition die Gleichung $\mathbf{f}(\mathbf{x}, \mathbf{u}) = \mathbf{x}$. Er ist zugleich eine Ruhelage des rekurrenten Fuzzy-Systems, denn es gilt dann: $\mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k)) = \mathbf{x}(k)$. Damit erhält man den folgenden Existenzsatz für Ruhelagen:

Satz 12

Ein stetiges rekurrentes Fuzzy-System besitzt für jeden Eingabevektor $\mathbf{u}$ mindestens eine Ruhelage.

Wünschenswert wäre es zu wissen, wo sich diese Ruhelage oder die Ruhelagen befinden. Einige Ruhelagen lassen sich direkt aus der Regelbasis ablesen: Wenn es für einen linguistischen Eingabevektor $\mathbf{L}_q^u$ einen linguistischen Vektor $\mathbf{L}_j^x$ gibt, der durch die entsprechende Regel aus der Regelbasis auf sich selbst abgebildet wird, so wird dieser linguistische Vektor im Weiteren auch *Endknoten* für diesen Eingabevektor genannt. Da der linguistische Zustandsvektor sich bei der Auswertung der Regel nicht ändert und man die Wirkung der Regeln nach Lemma 1 direkt auf die Kernpositionen übertragen kann, folgt:

Lemma 3

Wenn der linguistische Zustandsvektor $\mathbf{L}_j^x$ ein Endknoten für einen bestimmten linguistischen Eingabevektor $\mathbf{L}_q^u$ ist, so ist der Zustandsvektor $\mathbf{s}_j^x$ eine Ruhelage des rekurrenten Fuzzy-Systems, an dem der Eingabevektor $\mathbf{s}_q^u$ anliegt.

Aussagen über Ruhelagen, die nicht an Kernpositionsvektoren liegen, sind im Allgemeinen schwerer zu treffen, da für die Abbildung solcher Ruhelagen normalerweise mehrere

Regeln verantwortlich sind. Am leichtesten ist dies noch bei regelstetigen Fuzzy-Systemen möglich. Um dies zu sehen wird hier eine alternative Charakterisierung der Regelstetigkeit angegeben: Ein rekurrentes Fuzzy-System ist regelstetig, wenn je zwei benachbarte Regeln in der Regelmatrix gleiche oder benachbarte linguistische Vektoren in ihren Konklusionen aufweisen. D.h., ändert man in regelstetigen Systemen den linguistischen Wert einer linguistischen Variablen in der Prämisse auf den nächst kleineren oder nächst größeren linguistischen Wert, so ändert sich in der Konklusion maximal ein linguistischer Wert und zwar nur entweder auf den nächst kleineren oder nächst größeren.

Bei regelstetigen Systemen vergrößert sich der "Abstand" zweier linguistischer Vektoren durch die Ausführung der Regeln nicht. Um diese Eigenschaft auch auf die quantitative Ebene übertragen zu können, werden im Weiteren regelstetige standardisierte rekurrente Fuzzy-Systeme betrachtet. Für jede Komponente x_i des Zustandsvektors $\mathbf{x}$ haben in diesem Fall benachbarte Kernpositionen einen festen Abstand Δs_i , also $\Delta s_i = |s_{r+1}^{x_i} - s_r^{x_i}|$. Dann wird durch

$$\|\mathbf{x}\| = \sum_i \frac{1}{\Delta s_i} |x_i| \quad (6.2)$$

die *kanonische Vektornorm* im Zustandsraum X festgelegt. Sie ist eine modifizierte Betragssummennorm oder Höldersche Norm (vgl. [53] mit $p = 1$).

Mit Hilfe der kanonischen Vektornorm kann man formulieren, dass sich der Abstand zweier beliebiger Zustandsvektoren $\mathbf{a}, \mathbf{b} \in X$ unter der Transitionsfunktion $\mathbf{f}$ eines regelstetigen standardisierten rekurrenten Fuzzy-Systems nicht vergrößert, d.h. $\|\mathbf{f}(\mathbf{a}, \mathbf{u}) - \mathbf{f}(\mathbf{b}, \mathbf{u})\| \leq \|\mathbf{a} - \mathbf{b}\|$ für alle $\mathbf{a}, \mathbf{b} \in X$ und alle $\mathbf{u} \in U$. Die Transitionsfunktion ist dann *nichtexpandierend* [85] im Zustandsvektor $\mathbf{x}$. Man erhält

Satz 13

Regelstetige standardisierte rekurrente Fuzzy-Systeme sind nichtexpandierend im Zustandsvektor bezüglich ihrer kanonischen Vektornorm.

Beweis:

Zu zeigen ist $\|\mathbf{f}(\mathbf{a}, \mathbf{u}) - \mathbf{f}(\mathbf{b}, \mathbf{u})\| \leq \|\mathbf{a} - \mathbf{b}\|$ für alle $\mathbf{a}, \mathbf{b} \in X$ und alle $\mathbf{u} \in U$. Der Beweis erfolgt in drei Schritten:

1. Schritt: Zuerst wird obige Behauptung für den folgenden Spezialfall gezeigt: $\mathbf{a}$ und $\mathbf{b}$ unterscheiden sich maximal in einer Komponente h , also $a_i = b_i$ für $i \neq h$ und die beiden Werte a_h und b_h liegen in einem Intervall, das durch zwei benachbarte Kernpositionen $s_r^{x_h}$ und $s_{r+1}^{x_h}$ beschränkt wird, also $a_h, b_h \in [s_r^{x_h}, s_{r+1}^{x_h}]$.

Mit $f_l(\mathbf{x}, \mathbf{u}) = \sum_{\mathbf{j}, \mathbf{q}} s_{w_l(\mathbf{j}, \mathbf{q})}^{x_l} \prod_i \mu_{j_i}^{x_i}(x_i) \prod_p \mu_{q_p}^{u_p}(u_p)$ nach Gleichung (3.13) folgt:

$$\begin{aligned} \|\mathbf{f}(\mathbf{a}, \mathbf{u}) - \mathbf{f}(\mathbf{b}, \mathbf{u})\| &= \\ &= \sum_l \frac{1}{\Delta s_l} \left| \sum_{\mathbf{j}, \mathbf{q}} s_{w_l(\mathbf{j}, \mathbf{q})}^{x_l} \prod_i \mu_{j_i}^{x_i}(a_i) \prod_p \mu_{q_p}^{u_p}(u_p) - \sum_{\mathbf{j}, \mathbf{q}} s_{w_l(\mathbf{j}, \mathbf{q})}^{x_l} \prod_i \mu_{j_i}^{x_i}(b_i) \prod_p \mu_{q_p}^{u_p}(u_p) \right|. \end{aligned} \quad (6.3)$$

Da in diesem Fall $a_i = b_i$ für $i \neq h$ ist, folgt:

$$||\mathbf{f}(\mathbf{a}, \mathbf{u}) - \mathbf{f}(\mathbf{b}, \mathbf{u})|| = \sum_l \frac{1}{\Delta s_l} \left| \sum_{\mathbf{j}, \mathbf{q}} s_{w_l(\mathbf{j}, \mathbf{q})}^{x_l} \cdot (\mu_{j_h}^{x_h}(a_h) - \mu_{j_h}^{x_h}(b_h)) \cdot \prod_{i \neq h} \mu_{j_i}^{x_i}(a_i) \prod_p \mu_{q_p}^{u_p}(u_p) \right|. \quad (6.4)$$

Da für die Zustandsgröße x_h die Werte a_h und b_h beide aus dem Intervall $[s_r^{x_h}, s_{r+1}^{x_h}]$ kommen, gilt mit der Festlegung der Hilfsgröße $\Delta\mu = \mu_r^{x_h}(a_h) - \mu_r^{x_h}(b_h)$:

$$\mu_{r+1}^{x_h}(a_h) - \mu_{r+1}^{x_h}(b_h) = (1 - \mu_r^{x_h}(a_h)) - (1 - \mu_r^{x_h}(b_h)) = -(\mu_r^{x_h}(a_h) - \mu_r^{x_h}(b_h)) = -\Delta\mu \quad (6.5)$$

und damit

$$\mu_{j_h}^{x_h}(a_h) - \mu_{j_h}^{x_h}(b_h) = \begin{cases} \Delta\mu & \text{für } j_h = r, \\ -\Delta\mu & \text{für } j_h = r + 1, \\ 0 & \text{für } j_h \notin \{r, r + 1\}. \end{cases} \quad (6.6)$$

Führt man die Summation über j_h in Gleichung (6.4) explizit aus, so erhält man:

$$\begin{aligned} ||\mathbf{f}(\mathbf{a}, \mathbf{u}) - \mathbf{f}(\mathbf{b}, \mathbf{u})|| &= \\ &= \sum_l \frac{1}{\Delta s_l} \left| \sum_{j_i \neq j_h, \mathbf{q}} (s_{w_l(j_1, \dots, r, \dots, j_n, \mathbf{q})}^{x_l} - s_{w_l(j_1, \dots, r+1, \dots, j_n, \mathbf{q})}^{x_l}) \cdot \Delta\mu \cdot \prod_{i \neq h} \mu_{j_i}^{x_i}(a_i) \prod_p \mu_{q_p}^{u_p}(u_p) \right|. \end{aligned} \quad (6.7)$$

Zieht man den gemeinsamen Faktor $\Delta\mu$ aus dem Betrag, so kann man unter Verwendung der Dreiecksungleichung obigen Term abschätzen durch

$$\begin{aligned} ||\mathbf{f}(\mathbf{a}, \mathbf{u}) - \mathbf{f}(\mathbf{b}, \mathbf{u})|| &\leq \\ &\leq |\Delta\mu| \cdot \sum_l \frac{1}{\Delta s_l} \left(\sum_{j_i \neq j_h, \mathbf{q}} |s_{w_l(j_1, \dots, r, \dots, j_n, \mathbf{q})}^{x_l} - s_{w_l(j_1, \dots, r+1, \dots, j_n, \mathbf{q})}^{x_l}| \cdot \prod_{i \neq h} \mu_{j_i}^{x_i}(a_i) \prod_p \mu_{q_p}^{u_p}(u_p) \right). \end{aligned} \quad (6.8)$$

Aufgrund der Regelstetigkeit sind $\mathbf{s}_{\mathbf{w}(j_1, \dots, r, \dots, j_n, \mathbf{q})}^{\mathbf{x}}$ und $\mathbf{s}_{\mathbf{w}(j_1, \dots, r+1, \dots, j_n, \mathbf{q})}^{\mathbf{x}}$ gleiche oder benachbarte Kernpositionsvektoren und daher ist $\sum_l \frac{1}{\Delta s_l} |s_{w_l(j_1, \dots, r, \dots, j_n, \mathbf{q})}^{x_l} - s_{w_l(j_1, \dots, r+1, \dots, j_n, \mathbf{q})}^{x_l}| \leq 1$.

Mit dieser Abschätzung erhält man:

$$||\mathbf{f}(\mathbf{a}, \mathbf{u}) - \mathbf{f}(\mathbf{b}, \mathbf{u})|| \leq |\Delta\mu| \cdot \sum_{j_i \neq j_h, \mathbf{q}} \prod_{i \neq h} \mu_{j_i}^{x_i}(a_i) \prod_p \mu_{q_p}^{u_p}(u_p). \quad (6.9)$$

Mit der Normierungsbedingung aus Gleichung (3.12) folgt nach dem Ausmultiplizieren

$$1 = \prod_{i \neq h} \left(\sum_{j_i} \mu_{j_i}^{x_i}(x_i) \right) \cdot \prod_p \left(\sum_{q_p} \mu_{q_p}^{u_p}(u_p) \right) = \sum_{j_i \neq j_h, \mathbf{q}} \prod_{i \neq h} \mu_{j_i}^{x_i}(a_i) \prod_p \mu_{q_p}^{u_p}(u_p). \quad (6.10)$$

Damit hat die Summe in Gleichung (6.9) den Wert 1 und es gilt:

$$||\mathbf{f}(\mathbf{a}, \mathbf{u}) - \mathbf{f}(\mathbf{b}, \mathbf{u})|| \leq |\Delta\mu|. \quad (6.11)$$

Da standardisierte rekurrente Fuzzy-Systeme dreiecksförmige Zugehörigkeitsfunktionen

nutzen, ist $\mu_r^{x_h}(a_h) = 1 - \frac{1}{\Delta s_h}(a_h - s_r^{x_h})$ und damit lässt sich die linke Seite von Gl.(6.11) wie folgt umformen:

$$|\Delta\mu| = |\mu_r^{x_h}(a_h) - \mu_r^{x_h}(b_h)| = |(1 - \frac{1}{\Delta s_h}(a_h - s_r^{x_h})) - (1 - \frac{1}{\Delta s_h}(b_h - s_r^{x_h}))| = \frac{1}{\Delta s_h}|a_h - b_h|. \quad (6.12)$$

Zusätzlich gilt $\|\mathbf{a} - \mathbf{b}\| = \sum_i \frac{1}{\Delta s_i}|a_i - b_i| = \frac{1}{\Delta s_h}|a_h - b_h|$ und so erhält man zusammen mit Gl.(6.11) und Gl.(6.12):

$$\|\mathbf{f}(\mathbf{a}, \mathbf{u}) - \mathbf{f}(\mathbf{b}, \mathbf{u})\| \leq \|\mathbf{a} - \mathbf{b}\|. \quad (6.13)$$

Damit wurde die Behauptung für den in diesem ersten Schritt betrachteten Spezialfall bewiesen.

2. Schritt: Jetzt wird im Gegensatz zu Schritt 1 nur noch gefordert, dass sich $\mathbf{a}$ und $\mathbf{b}$ maximal in einer Komponente h unterscheiden, also $a_i = b_i$ für $i \neq h$ und nicht mehr zusätzlich $a_h, b_h \in [s_r^{x_h}, s_{r+1}^{x_h}]$.

Ohne Einschränkung der Allgemeinheit wird angenommen, dass $a_h \leq b_h$. Dann liegen zwischen a_h und b_h Kernpositionen $s_{r+1}^{x_h}, s_{r+2}^{x_h}, \dots, s_{r+t}^{x_h}$. Für jeden Index $v = 1, \dots, t$ werden die Komponenten der Hilfsvektoren $\mathbf{d}(v)$ durch $d_i(v) = a_i (= b_i)$ für $i \neq h$ und $d_h(v) = s_{r+v}^{x_h}$ festgelegt. Des Weiteren wird $\mathbf{d}(0) = \mathbf{a}$ und $\mathbf{d}(t+1) = \mathbf{b}$ gesetzt.

Dann gilt:

$$\begin{aligned} \|\mathbf{f}(\mathbf{a}, \mathbf{u}) - \mathbf{f}(\mathbf{b}, \mathbf{u})\| &= \|\mathbf{f}(\mathbf{d}(0), \mathbf{u}) - \mathbf{f}(\mathbf{d}(t+1), \mathbf{u})\| = \\ &= \left\| \sum_{v=0}^t (\mathbf{f}(\mathbf{d}(v), \mathbf{u}) - \mathbf{f}(\mathbf{d}(v+1), \mathbf{u})) \right\| \leq \sum_{v=0}^t \|\mathbf{f}(\mathbf{d}(v), \mathbf{u}) - \mathbf{f}(\mathbf{d}(v+1), \mathbf{u})\| \end{aligned} \quad (6.14)$$

Da zwischen $d_h(v)$ und $d_h(v+1)$ jeweils keine weiteren Kernpositionen liegen, folgt mit Schritt 1:

$$\sum_{v=0}^t \|\mathbf{f}(\mathbf{d}(v), \mathbf{u}) - \mathbf{f}(\mathbf{d}(v+1), \mathbf{u})\| \leq \sum_{v=0}^t \|\mathbf{d}(v) - \mathbf{d}(v+1)\| = \sum_{v=0}^t \frac{1}{\Delta s_h} |d_h(v) - d_h(v+1)|. \quad (6.15)$$

Da $d_h(v)$ mit steigendem v monoton zunimmt, folgt:

$$\sum_{v=0}^t \frac{1}{\Delta s_h} |d_h(v) - d_h(v+1)| = \frac{1}{\Delta s_h} |d_h(0) - d_h(t+1)| = \frac{1}{\Delta s_h} |a_h - b_h| = \|\mathbf{a} - \mathbf{b}\|. \quad (6.16)$$

Aus der Ungleichungskette (6.14), (6.15), (6.16) folgt, dass $\mathbf{f}$ auch für den in diesem Schritt behandelten Spezialfall die Gleichung $\|\mathbf{f}(\mathbf{a}, \mathbf{u}) - \mathbf{f}(\mathbf{b}, \mathbf{u})\| \leq \|\mathbf{a} - \mathbf{b}\|$ erfüllt und damit nichtexpandierend ist.

3. Schritt: Jetzt werden $\mathbf{a}$ und $\mathbf{b}$ beliebig aus X gewählt.

Definiere Vektoren $\mathbf{c}(1)$ bis $\mathbf{c}(n+1)$ aus X durch $c_i(v) = a_i$ für $i \geq v$ und $c_i(v) = b_i$ für $i < v$. Nach dieser Definition ist $\mathbf{c}(1) = \mathbf{a}$ und $\mathbf{c}(n+1) = \mathbf{b}$.

Dann gilt:

$$\begin{aligned} & \|f(\mathbf{a}, \mathbf{u}) - f(\mathbf{b}, \mathbf{u})\| = \|f(\mathbf{c}(1), \mathbf{u}) - f(\mathbf{c}(n+1), \mathbf{u})\| = \\ & = \left\| \sum_{v=1}^n (f(\mathbf{c}(v), \mathbf{u}) - f(\mathbf{c}(v+1), \mathbf{u})) \right\| \leq \sum_{v=1}^n \|f(\mathbf{c}(v), \mathbf{u}) - f(\mathbf{c}(v+1), \mathbf{u})\|. \end{aligned} \quad (6.17)$$

Da sich je zwei Vektoren $\mathbf{c}(v)$ und $\mathbf{c}(v+1)$ nur in ihrer v -ten Komponente unterscheiden, folgt mit Schritt 2:

$$\begin{aligned} & \sum_{v=1}^n \|f(\mathbf{c}(v), \mathbf{u}) - f(\mathbf{c}(v+1), \mathbf{u})\| \leq \sum_{v=1}^n \|\mathbf{c}(v) - \mathbf{c}(v+1)\| = \\ & = \sum_{v=1}^n \frac{1}{\Delta s_v} |c_v(v) - c_v(v+1)| = \sum_{v=1}^n \frac{1}{\Delta s_v} |a_v - b_v| = \|\mathbf{a} - \mathbf{b}\|. \end{aligned} \quad (6.18)$$

Also gilt $\|f(\mathbf{a}, \mathbf{u}) - f(\mathbf{b}, \mathbf{u})\| \leq \|\mathbf{a} - \mathbf{b}\|$ auch für beliebige Vektoren $\mathbf{a}$ und $\mathbf{b}$ aus X und damit ist die Funktion $f(\mathbf{x}, \mathbf{u})$ nichtexpandierend in x . $\square$

Zu beachten ist, dass dieser Satz unabhängig von der speziellen Wahl des Eingabevektors $\mathbf{u}$ ist. Das bedeutet, dass in einem solchen System auch bei Wahl einer willkürlichen Folge $\mathbf{u}(k)$ von Eingabevektoren sich der Abstand zweier Zustandsvektoren im Laufe der Abbildungen nicht vergrößern kann. Des Weiteren wurde im Beweis nur die Regelstetigkeitsbeziehung in den Zustandsgrößen ausgenutzt. Daher ist der Satz auch für rekurrente Fuzzy-Systeme gültig, die in den Zustandsvariablen, aber nicht unbedingt in der Eingabevariablen regelstetig sind.

Die Nichtexpansionseigenschaft aus Satz 13 hat auch direkte Auswirkungen auf die Stabilität von Ruhelagen. Dazu wird wieder vorausgesetzt, dass ein fester Eingabevektor $\mathbf{u}$ anliegt. Es gilt

Satz 14

Jede Ruhelage eines regelstetigen standardisierten rekurrenten Fuzzy-Systems ist stabil im Sinne von Ljapunov¹. Bei gegebener Ruhelage $\mathbf{b}$ ist durch

$$V_{\mathbf{b}}(\mathbf{x}) = \|\mathbf{x} - \mathbf{b}\| = \sum_i \frac{1}{\Delta s_i} |x_i - b_i| \quad (6.19)$$

eine Ljapunovfunktion gegeben.

Beweis:

Es gilt $V_{\mathbf{b}}(\mathbf{b}) = 0$ und $V_{\mathbf{b}}(\mathbf{x}) > 0$ für $\mathbf{x} \neq \mathbf{b}$. Nach Satz 13 gilt:

$$V_{\mathbf{b}}(f(\mathbf{x}, \mathbf{u})) = \|f(\mathbf{x}, \mathbf{u}) - \mathbf{b}\| = \|f(\mathbf{x}, \mathbf{u}) - f(\mathbf{b}, \mathbf{u})\| \leq \|\mathbf{x} - \mathbf{b}\| = V_{\mathbf{b}}(\mathbf{x}). \quad (6.20)$$

Also ist $\Delta V_{\mathbf{b}} = V_{\mathbf{b}}(f(\mathbf{x}, \mathbf{u})) - V_{\mathbf{b}}(\mathbf{x}) \leq 0$. Damit ist $V_{\mathbf{b}}(\mathbf{x})$ eine Ljapunovfunktion und die Ruhelage $\mathbf{b}$ ist stabil im Sinne von Ljapunov. $\square$

¹Definitionen und Sätze über die Stabilitätstheorie von Ljapunov finden sich in [80, 48]

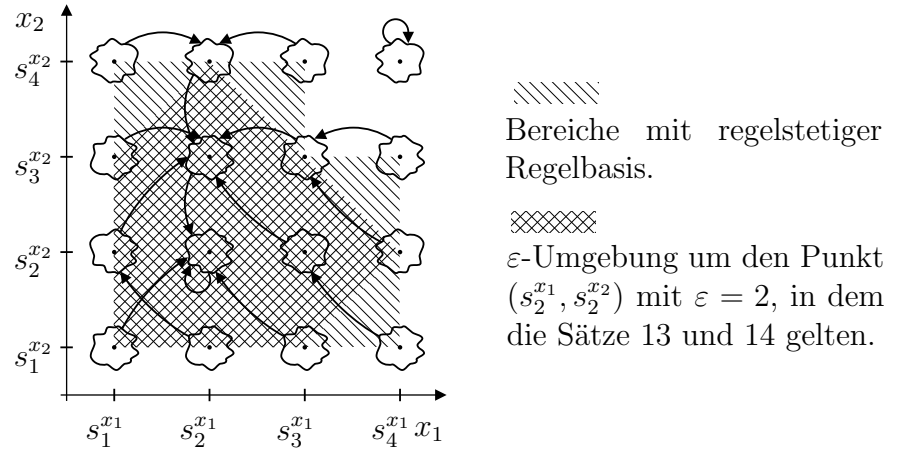


Abbildung 6.2: Sätze 13 und 14 lassen sich auch auf Teilbereiche des Zustandsraumes anwenden, in denen die Voraussetzungen erfüllt sind.

Die Sätze 13 und 14 können auch auf Teilbereiche des Zustandsraums angewendet werden, wenn die Regelbasis nur zum Teil regelstetig ist. Abbildung 6.2 zeigt dies beispielhaft. Zu sehen ist ein zweidimensionaler Zustandsraum mit Punkten an den Kernpositionsvektoren. Dem Zustandsraum ist der Zustandsgraph eines linguistischen Automaten überlagert, dargestellt durch die welligen Knoten und die Transitionspfeile. Die Regelbasis ist im schraffierten Bereich regelstetig, wie leicht anhand der Transitionspfeile prüfbar ist. Der karierte Bereich zeigt eine ε -Umgebung mit $\varepsilon = 2$ um die Ruhelage $(s_2^{x_1}, s_2^{x_2})$, der vollständig im schraffierten Bereich liegt und in dem die Transitionsfunktion nichtexpandierend ist. Die Ruhelage $(s_2^{x_1}, s_2^{x_2})$ ist stabil im Sinne von Ljapunov.

Die Stabilität einer Ruhelage stellt kein konvergentes Verhalten in ihrer Umgebung sicher, denn auch Oszillationen sind möglich. Wann man Konvergenz garantieren kann, wird im nächsten Abschnitt behandelt.

6.3 Konvergenzsätze für standardisierte rekurrente Fuzzy-Systeme

Zuerst werden standardisierte rekurrente Fuzzy-Systeme behandelt, die keine Eingabe-größen haben und im Weiteren als *autonome* standardisierte rekurrente Fuzzy-Systeme bezeichnet werden. Ihre Transitionsfunktion wird mit $\mathbf{f}(\mathbf{x})$ bezeichnet. Die Aussagen über diese Systeme sind direkt auf standardisierte rekurrente Fuzzy-Systeme anwendbar, deren Eingabevektor genau auf einem Kernpositionsvektor liegt und sich nicht mit der Zeit ändert. Denn dann gilt auch wie im autonomen Fall nur ein einziger, fester Regelsatz statt mehrerer Regelsätze gleichzeitig.

Unter Konvergenz eines rekurrenten Fuzzy-Systems wird hier die Konvergenz der Folge der Zustandsvektoren $\mathbf{x}(k)$ aus Gleichung (3.13) verstanden. Bei autonomen rekurrenten Fuzzy-Systemen ergibt sich der Vektor $\mathbf{x}(k)$ aus dem Startvektor $\mathbf{x}(0)$ durch k -malige

Anwendung der Transitionsfunktion $\mathbf{f}$, die kompakt mit $\mathbf{f}^k$ bezeichnet wird, also $\mathbf{x}(k) = \mathbf{f}^k(\mathbf{x}(0))$. Analog wird auch in nicht autonomen rekurrenten Fuzzy-Systemen die k -malige Anwendung der Transitionsfunktion $\mathbf{f}$ bei festem Eingabevektor $\mathbf{u}$ mit $\mathbf{f}^k(\mathbf{x}, \mathbf{u})$ bezeichnet.

Damit ein rekurrentes Fuzzy-System für einen beliebigen Startvektor $\mathbf{x}$ konvergiert, dürfen keine *echten Zyklen* auftreten, d.h. Zyklen mit Länge $n > 1$, also Punktfolgen, die periodisch immer wieder die gleichen Werte annehmen und selbst keine Fixpunkte enthalten. Insbesondere darf es keine echten Zyklen in der Regelbasis geben. Daraus folgt, dass jeder linguistische Zustandsvektor durch den linguistischen Automaten irgendwann auf einen Endknoten abgebildet wird, d.h. auf einen linguistischen Vektor, der wieder auf sich selbst abgebildet wird. Ein rekurrentes Fuzzy-System, das diese Bedingung erfüllt, wird als *terminiertes* rekurrentes Fuzzy-System bezeichnet. Es gilt

Satz 15

Autonome, terminierte, regelstetige standardisierte rekurrente Fuzzy-Systeme konvergieren für jeden Anfangszustand.

Beweis:

Da der Zustandsraum X als kompakt vorausgesetzt werden kann, besitzt jede beliebige Folge $\mathbf{a}(k) = \mathbf{f}^k(\mathbf{a})$ eines autonomen, terminierten, regelstetigen standardisierten rekurrenten Fuzzy-Systems einen Häufungspunkt $\mathbf{x}$. Also ist

$$\liminf_{k \rightarrow \infty} \|\mathbf{f}^k(\mathbf{a}) - \mathbf{x}\| = 0. \quad (6.21)$$

Nach Lemma 15 aus Anhang B ist der Vektor $\mathbf{x}$ Fixpunkt und damit folgt:

$$\liminf_{k \rightarrow \infty} \|\mathbf{f}^k(\mathbf{a}) - \mathbf{f}^k(\mathbf{x})\| = \liminf_{k \rightarrow \infty} \|\mathbf{f}^k(\mathbf{a}) - \mathbf{x}\| = 0. \quad (6.22)$$

Nach Satz 13 ist die Funktion $\mathbf{f}$ nichtexpandierend und daher gilt weiter

$$\lim_{k \rightarrow \infty} \|\mathbf{f}^k(\mathbf{a}) - \mathbf{f}^k(\mathbf{x})\| = \liminf_{k \rightarrow \infty} \|\mathbf{f}^k(\mathbf{a}) - \mathbf{f}^k(\mathbf{x})\| = 0. \quad (6.23)$$

Nach Lemma 15 aus Anhang B ist der Vektor $\mathbf{x}$ Fixpunkt und damit gilt schließlich

$$\lim_{k \rightarrow \infty} \|\mathbf{f}^k(\mathbf{a}) - \mathbf{x}\| = \lim_{k \rightarrow \infty} \|\mathbf{f}^k(\mathbf{a}) - \mathbf{f}^k(\mathbf{x})\| = 0, \quad (6.24)$$

und die beliebig gewählte Folge $\mathbf{f}^k(\mathbf{a})$ konvergiert gegen $\mathbf{x}$. □

Liegt ein autonomes, terminiertes, regelstetiges standardisiertes rekurrentes Fuzzy-System vor, so laufen die linguistischen Zustandsvektoren des dazugehörigen linguistischen Automaten in einen Endknoten, sie konvergieren also. Diese Konvergenz überträgt sich nach Satz 15 auf die Zustandsvektoren $\mathbf{x}$ des rekurrenten Fuzzy-Systems. Man kann folglich anhand des Verhaltens des linguistischen Automaten mit Hilfe des Zustandsgraphen Rückschlüsse auf das dynamische Verhalten des rekurrenten Fuzzy-Systems ziehen.

Die Voraussetzungen für Satz 15 lassen sich nicht ohne weiteres lockern. Wie oben erläutert wurde, muss ein konvergentes rekurrentes Fuzzy-System auch terminiert sein. Lässt man die Regelstetigkeit weg, so gibt es auch nicht konvergente rekurrente Fuzzy-Systeme, die sogar chaotisch sein können, wie das Beispiel in Abbildung 5.6 (d) im vorangegangenen

Kapitel zeigt. Schließlich gibt es (nicht autonome) terminierte, regelstetige standardisierte rekurrente Fuzzy-Systeme, die nicht konvergent sind. Ein solches ist als Beispiel 3 im Anhang A zu finden.

Allerdings ist bei einer wichtigen Klasse nicht autonomer, terminierter, regelstetiger standardisierter rekurrenter Fuzzy-Systeme trotzdem Konvergenz gegeben: bei eindimensionalen Systemen, d.h. Systemen, deren Zustandsvektoren nur eine Komponente haben. Dazu der folgende Satz:

Satz 16

Eindimensionale, terminierte, regelstetige standardisierte rekurrente Fuzzy-Systeme sind konvergent.

Beweis:

Angenommen ein eindimensionales, terminiertes, regelstetiges standardisiertes rekurrentes Fuzzy-System ist für einen festen Eingabevektor $\mathbf{u}$ nicht konvergent. Dann hat dessen Transitionsfunktion f nach Satz 26 aus Anhang D einen Zweierzyklus, d.h. es gibt ein $a \in X$ mit $f(a, \mathbf{u}) \neq a$ und $f^2(a, \mathbf{u}) = a$. Nach Satz 12 besitzt die Transitionsfunktion f einen Fixpunkt b , d.h. $f(b, \mathbf{u}) = b$. Da die Funktion f nach Satz 13 nichtexpandierend ist, gilt:

$$\|f^2(a, \mathbf{u}) - f^2(b, \mathbf{u})\| \leq \|f(a, \mathbf{u}) - f(b, \mathbf{u})\| \leq \|a - b\|. \quad (6.25)$$

Aufgrund der Eigenschaften von a und b folgt aus Gleichung (6.25):

$$\|a - b\| \leq \|f(a, \mathbf{u}) - b\| \leq \|a - b\| \quad (6.26)$$

und damit

$$\|f(a, \mathbf{u}) - b\| = \|a - b\|. \quad (6.27)$$

Da die Transitionsfunktion f eindimensional ist, folgt:

$$f(a, \mathbf{u}) - b = \pm(a - b). \quad (6.28)$$

Da a selbst kein Fixpunkt ist, ist $f(a, \mathbf{u}) - b \neq a - b$, und damit folgt:

$$f(a, \mathbf{u}) - b = b - a. \quad (6.29)$$

Wir können ohne Einschränkung im Weiteren annehmen, dass $f(a, \mathbf{u}) < b < a$, da die Punkte a und $f(a, \mathbf{u})$ aus dem gleichen Zweierzyklus stammen. Die Gleichung (6.29) bleibt auch gültig, wenn statt a ein beliebiger Wert $x \in [f(a, \mathbf{u}), a]$ eingesetzt wird, wie im Folgenden gezeigt wird: Mit der Dreiecksungleichung und der Nichtexpansion der Funktion f gilt für alle $x \in [b, a]$:

$$\begin{aligned} \|a - b\| &= \|a - x\| + \|x - b\| \geq \|f(a, \mathbf{u}) - f(x, \mathbf{u})\| + \|f(x, \mathbf{u}) - b\| \geq \\ &\geq \|f^2(a, \mathbf{u}) - f^2(x, \mathbf{u})\| + \|f^2(x, \mathbf{u}) - b\| \geq \|f^2(a, \mathbf{u}) - b\|. \end{aligned} \quad (6.30)$$

Da a der Punkt eines Zweierzyklus ist, ist der erste und der letzte Term in der Ungleichungskette (6.30) gleich. Damit müssen alle Terme der Ungleichungskette gleich sein und

es folgt insbesondere:

$$||a - x|| + ||x - b|| = ||f(a, \mathbf{u}) - f(x, \mathbf{u})|| + ||f(x, \mathbf{u}) - b||. \quad (6.31)$$

Da nach Satz 13 die Transitionsfunktion f nichtexpandierend ist und daher die Ungleichungen $||a - x|| \leq ||f(a, \mathbf{u}) - f(x, \mathbf{u})||$ und $||x - b|| \leq ||f(x, \mathbf{u}) - b||$ gelten, folgt aus Gleichung (6.31) auch $||x - b|| = ||f(x, \mathbf{u}) - b||$ und damit

$$f(x, \mathbf{u}) - b = \pm(x - b). \quad (6.32)$$

Da die Funktion f stetig ist, folgt mit Gleichung (6.29)

$$f(x, \mathbf{u}) = -x + 2b \quad (6.33)$$

für alle $x \in [b, a]$ und analog für alle $x \in [f(a, \mathbf{u}), b]$ und damit für alle $x \in [f(a, \mathbf{u}), a]$.

Der Fixpunkt b liegt zwischen zwei Kernpositionen s_r^x und s_{r+1}^x , also $s_r^x \leq b < s_{r+1}^x$. Dann gilt für alle $x \in [b, s_{r+1}^x] \cap [b, a]$ nach Gleichung (B.2) aus Lemma 11 aus Anhang B:

$$f(x, \mathbf{u}) = \sum_j f(s_j^x, \mathbf{u}) \mu_j^x(x) = f(s_r^x, \mathbf{u}) \mu_r^x(x) + f(s_{r+1}^x, \mathbf{u}) \mu_{r+1}^x(x). \quad (6.34)$$

Wegen der Normierung der Zugehörigkeitsfunktionen nach Gleichung (3.12) gilt $\mu_{r+1}^x(x) = 1 - \mu_r^x(x)$ und für dreiecksförmige Zugehörigkeitsfunktionen gilt $\mu_r^x(x) = \frac{x - s_{r+1}^x}{s_r^x - s_{r+1}^x} = -\frac{x - s_{r+1}^x}{\Delta s}$ mit $\Delta s = s_{r+1}^x - s_r^x$. Dann kann man Gleichung (6.34) umformen in:

$$\begin{aligned} f(x, \mathbf{u}) &= (f(s_r^x, \mathbf{u}) - f(s_{r+1}^x, \mathbf{u})) \mu_r^x(x) + f(s_{r+1}^x, \mathbf{u}) = \\ &= -\frac{1}{\Delta s} (f(s_r^x, \mathbf{u}) - f(s_{r+1}^x, \mathbf{u})) x + \frac{1}{\Delta s} (f(s_r^x, \mathbf{u}) - f(s_{r+1}^x, \mathbf{u})) s_{r+1}^x + f(s_{r+1}^x, \mathbf{u}). \end{aligned} \quad (6.35)$$

Vergleicht man die Koeffizienten von x in Gleichung (6.33) und Gleichung (6.35), so folgt:

$$f(s_r^x, \mathbf{u}) - f(s_{r+1}^x, \mathbf{u}) = \Delta s. \quad (6.36)$$

Wegen der Regelstetigkeit gilt ohnehin $s_{w(r, \mathbf{q})}^x - s_{w(r+1, \mathbf{q})}^x \leq \Delta s$. Würde für einen Indexvektor $\mathbf{q}$ mit $\mu_{q_p}^{u_p} \neq 0$ die Ungleichung $s_{w(r, \mathbf{q})}^x - s_{w(r+1, \mathbf{q})}^x < \Delta s$ gelten, so wäre nach Gleichung (B.2) aus Lemma 11 aus dem Anhang B dagegen $f(s_r^x, \mathbf{u}) - f(s_{r+1}^x, \mathbf{u}) = \sum_{\mathbf{q}} (s_{w(r, \mathbf{q})}^x - s_{w(r+1, \mathbf{q})}^x) \mu_{q_p}^{u_p} < \Delta s$ im Widerspruch zu Gleichung (6.36). Also gilt für alle $\mathbf{q}$ mit $\mu_{q_p}^{u_p} \neq 0$:

$$s_{w(r, \mathbf{q})}^x - s_{w(r+1, \mathbf{q})}^x = \Delta s. \quad (6.37)$$

Es werden nun die beiden Fälle $b = s_r^x$ und $b > s_r^x$ betrachtet:

1. Fall:

Wenn $b = s_r^x$ ist, so gibt es aufgrund der Regelstetigkeit einen Indexvektor $\mathbf{q}$ mit $\prod_p \mu_{q_p}^{u_p} \neq 0$ und $s_{w(r, \mathbf{q})}^x = f(b, \mathbf{u}) = b = s_r^x$. Nach Gleichung (6.37) folgt dann: $s_{w(r+1, \mathbf{q})}^x = s_{w(r, \mathbf{q})}^x - \Delta s = s_r^x - \Delta s = s_{r-1}^x$ und analog $s_{w(r-1, \mathbf{q})}^x = s_{r+1}^x$, also

$$s_{w(r+1, \mathbf{q})}^x = s_{r-1}^x \text{ und } s_{w(r-1, \mathbf{q})}^x = s_{r+1}^x. \quad (6.38)$$

Also weist das rekurrente Fuzzy-System für den linguistischen Eingabevektor $\mathbf{L}_q^u$ mit dem Indexvektor $\mathbf{q}$ einen echten Zyklus auf. Die Regelbasis ist daher nicht terminiert im Widerspruch zur Voraussetzung.

2. Fall:

Es gelte nun $s_r^x < b < s_{r+1}^x$. Die Transitionsfunktion $f(x, \mathbf{u})$ ist zwischen den Kernpositionen s_r^x und s_{r+1}^x monoton in der Zustandsvariablen x und aufgrund von Gleichung (6.36) zwischen s_r^x und s_{r+1}^x monoton fallend. Weil b ein Fixpunkt in $[s_r^x, s_{r+1}^x]$ ist, folgt $f(s_r^x, \mathbf{u}) > b > f(s_{r+1}^x, \mathbf{u})$. Aufgrund der Regelstetigkeit und Gleichung (6.37) gibt es einen Indexvektor $\mathbf{q}$ mit $\prod_p \mu_{q_p}^{u_p}(u_p) \neq 0$ und $s_{w(r, \mathbf{q})}^x > b > s_{w(r+1, \mathbf{q})}^x$. Da nach Gleichung (6.37) zwischen $s_{w(r, \mathbf{q})}^x$ und $s_{w(r+1, \mathbf{q})}^x$ keine weitere Kernposition liegen darf und $s_r^x < b < s_{r+1}^x$, folgt:

$$s_{w(r+1, \mathbf{q})}^x = s_r^x \text{ und } s_{w(r, \mathbf{q})}^x = s_{r+1}^x. \quad (6.39)$$

Das rekurrente Fuzzy-System ist also auch in diesem Fall für den Indexvektor $\mathbf{q}$ nicht terminiert im Widerspruch zur Voraussetzung.

In beiden Fällen kommt es zum Widerspruch. Daher muss die Annahme falsch gewesen sein, und das gegebene rekurrente Fuzzy-System konvergiert. $\square$

Im Beweis stellte sich heraus, dass eindimensionale, terminierte, regelstetige standardisierte rekurrente Fuzzy-Systeme für keinen Eingabevektor $\mathbf{u}$ einen Zweierzyklus besitzen. Daher ist in vielen Fällen Satz 30 aus Anhang D anwendbar, mit dem man auch bei leichter Variation des Eingabevektors noch Konvergenz des Zustandswertes sicherstellen kann.

Mit den Konvergenzsätzen lässt sich untersuchen, ob ein vorliegendes System für alle Startwerte gegen Endwerte konvergiert. Wo diese Endwerte liegen können, wird im nächsten Abschnitt beschrieben.

6.4 Lokalisierung von Ruhelagen

Wie in Lemma 3 im Abschnitt 6.2 beschrieben, kann man Ruhelagen bzw. Fixpunktmen-gen zum Teil direkt aus der Regelbasis ablesen. Im Weiteren werden diese Aussagen auf Punkte ausgeweitet, die keine Kernpositionsvektoren sind. Dazu ist es nötig, beliebige Punkte $\mathbf{x}$ mit Kernpositionsvektoren in Beziehung zu setzen. Dies gelingt mit dem Begriff des elementaren Hyperquaders von $\mathbf{x}$ aus Kapitel 4, der sich für autonome standardisierte rekurrente Fuzzy-Systeme auch wie folgt beschreiben lässt:

Man betrachte dazu die Menge aller Indexvektoren $\mathbf{j}$ für die $\prod_i \mu_{j_i}^{x_i}(x_i) \neq 0$ ist. Die Kernpositionsvektoren $\mathbf{s}_j^x$ mit diesen Indexvektoren $\mathbf{j}$ bilden die Eckpunkte des Hyperquaders von $\mathbf{x}$ im Zustandsraum X . Es gilt

Satz 17

In autonomen, terminierten, regelstetigen standardisierten rekurrenten Fuzzy-Systemen liegt jede Ruhelage in einem elementaren Hyperquader, dessen Eckpunkte Ruhelagen, also Kernpositionsvektoren von Endknoten, sind. Ein solcher elementarer Hyperquader enthält ausschließlich Fixpunkte.

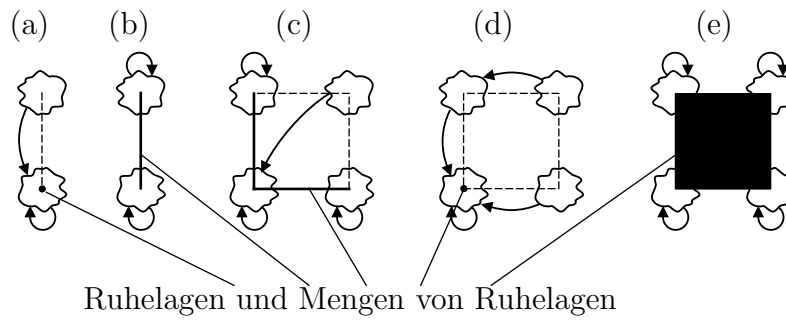


Abbildung 6.3: Beispiele für Mengen von Ruhelagen bei verschiedenen Regelsätzen von autonomen, terminierten, regelstetigen standardisierten rekurrenten Fuzzy-Systemen nach Satz 17. Die gestrichelten Linien stellen die Ränder der Hyperquader dar.

Beweis:

Eine Ruhelage $\mathbf{x}$ ist ein Fixpunkt der Transitionsfunktion. Nach Lemma 15 aus dem Anhang enthält der elementare Hyperquader ausschließlich Fixpunkte. Insbesondere sind auch die Eckpunkte des elementaren Hyperquaders von $\mathbf{x}$ Fixpunkte und damit Kernpositionsvektoren von Endknoten. $\square$

Satz 17 liefert ein Ergebnis, das man von einem rekurrenten Fuzzy-System erwarten würde: Ruhelagen befinden sich nur in solchen Teilbereichen des Zustandsraumes, in dem die Kernpositionen von Endknoten liegen. Abbildung 6.3 zeigt Zustandsgraphen und Ruhelagen im Zustandsraum von einigen einfachen, terminierten, regelstetigen standardisierten rekurrenten Fuzzy-Systemen. Die Ruhelagen in den Teilbildern (a) und (d) werden schon durch Lemma 3 beschrieben, da sie direkt auf Kernpositionen von Endknoten liegen. Die Teilbilder (b) und (e) zeigen, dass auch höherdimensionale elementare Hyperquader aus lauter Ruhelagen bestehen können, wenn deren Eckpunkte Kernpositionen von Endknoten sind. In Teilbild (c) sieht man, dass es auch mehrere elementare Hyperquader in einer Regelbasis geben kann, die aus Ruhelagen bestehen.

Betrachtet man nicht terminierte Systeme, so kann man ein ähnliches Kriterium für deren Ruhelagen nachweisen. Da dann echte Zyklen in der Regelbasis auftreten können, sind die Eckpunkte des elementaren Hyperquaders einer Ruhelage nicht unbedingt Fixpunkte. Trotzdem lassen sich die Ruhelagen in solchen Systemen leicht ausfindig machen, wie folgender Satz zeigt:

Satz 18

In autonomen, regelstetigen standardisierten rekurrenten Fuzzy-Systemen liegt jede Ruhelage in einem elementaren Hyperquader, dessen Eckpunkte durch die Transitionsfunktion permutiert werden.

Beweis:

Sei $\mathbf{x}$ eine Ruhelage in einem autonomen, regelstetigen standardisierten rekurrenten Fuzzy-System. Sei H der elementare Hyperquader von $\mathbf{x}$. Dann gibt es einen Eckpunkt $\hat{\mathbf{s}}$ von

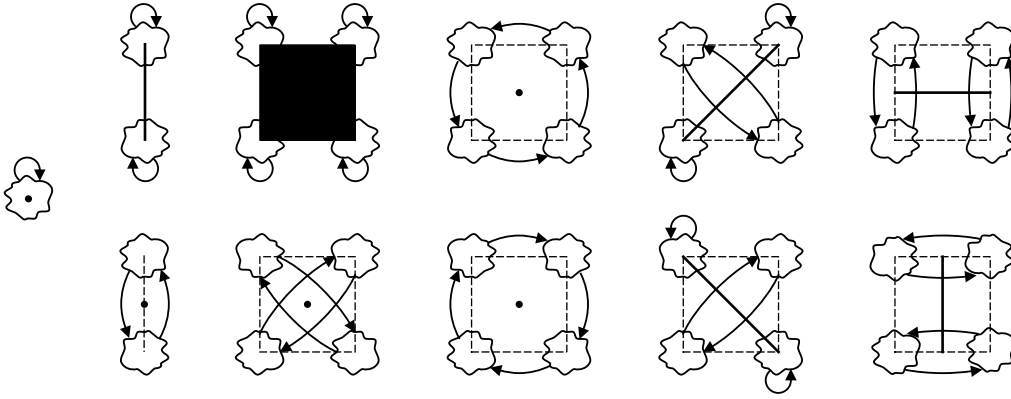


Abbildung 6.4: Schematische Darstellung aller möglichen null-, ein- und zweidimensionalen elementaren Hyperquader, in denen autonome, regelstetige standardisierte rekurrente Fuzzy-Systeme Ruhelagen nach Satz 18 haben können.

H , der den kürzesten Abstand zu $\mathbf{x}$ von allen Kernpositionsvektoren in X hat. Da die Transitionsfunktion $\mathbf{f}$ nach Satz 13 nichtexpandierend ist, gilt $\|\mathbf{f}(\hat{\mathbf{s}}) - \mathbf{f}(\mathbf{x})\| \leq \|\hat{\mathbf{s}} - \mathbf{x}\|$. Da das Bild $\mathbf{f}(\hat{\mathbf{s}})$ auch wieder ein Kernpositionsvektor ist und $\mathbf{x}$ ein Fixpunkt ist, haben die beiden Bildvektoren wieder den gleichen Abstand wie $\hat{\mathbf{s}}$ und $\mathbf{x}$. Außerdem sind die beiden elementaren Hyperquader von $\mathbf{f}(\mathbf{x})$ und $\mathbf{x}$ identisch. Damit sind Lemma 13 und Lemma 14 aus Anhang B in der erweiterten Fassung anwendbar.

Nach Lemma 14 wird die Menge S der Eckpunkte von H in sich selbst abgebildet. Außerdem wird kein anderer Eckpunkt außer $\hat{\mathbf{s}}$ auf $\mathbf{f}(\hat{\mathbf{s}})$ abgebildet, weil dies Lemma 13 widersprechen würde. Jetzt betrachtet man alle Eckpunkte von H , ausgenommen von $\mathbf{e}_1 = \hat{\mathbf{s}}$. Man wählt aus dieser Menge wieder einen Eckpunkt $\mathbf{e}_2$ mit dem kürzestem Abstand zu $\mathbf{x}$. Analog zum Fall von $\mathbf{e}_1 = \hat{\mathbf{s}}$ folgt, dass kein anderer Eckpunkt außer $\mathbf{e}_2$ auf $\mathbf{f}(\mathbf{e}_2)$ abgebildet wird. Per Induktion über alle Punkte aus S folgt dann, dass keine zwei Punkte aus S durch die Transitionsfunktion auf den gleichen Eckpunkt von H abgebildet werden. Da die Menge S der Eckpunkte von H endlich ist, muss die Funktion $\mathbf{f}$ also eine Permutation auf diesen Punkten realisieren. $\square$

Zur Veranschaulichung dieses Satzes sind in Abbildung 6.4 alle möglichen null-, ein- und zweidimensionalen elementaren Hyperquader von Ruhelagen in autonomen, regelstetigen standardisierten rekurrenten Fuzzy-Systemen aufgeführt. Die geltenden Regeln sind durch die Zustandsgraphen angegeben. Die Mengen der Ruhelagen sind Punkte, Linien oder Flächen. Im Falle nicht terminierter Systeme sind die Mengen der Ruhelagen immer nur Teilmengen der Hyperquader. Auffallend ist, dass der Mittelpunkt eines solchen elementaren Hyperquaders immer ein Ruhelage ist, was sich auch allgemein leicht nachweisen lässt.

Nachdem man solch eine Ruhelage gefunden hat, ist deren Stabilität von Interesse. Diese wird im nachfolgenden Abschnitt untersucht.

6.5 Asymptotische Stabilität von Ruhelagen

Bisher wurde betrachtet, ob es Ruhelagen gibt und wie man sie findet. Satz 14 liefert auch eine Aussage über die Stabilität von Ruhelagen in regelstetigen standardisierten rekurrenten Fuzzy-Systemen: Diese sind alle stabil im Sinne von Ljapunov. Nachfolgender Satz 19 beschreibt, welche Auswirkung die Existenz einer asymptotisch stabilen Ruhelage auf die Dynamik des Gesamtsystems hat.

Satz 19

Eine asymptotisch stabile Ruhelage eines regelstetigen standardisierten rekurrenten Fuzzy-Systems ist auch global asymptotisch stabil.

Beweis:

Sei $\mathbf{b}$ eine asymptotisch stabile Ruhelage für einen festen Eingabevektor $\mathbf{u}$. Dann gibt es für einen geeignet kleinen Wert $\varepsilon > 0$ eine Epsilonumgebung $U_\varepsilon(\mathbf{b}) = \{\mathbf{x} \in X \mid \|\mathbf{x} - \mathbf{b}\| < \varepsilon\}$, so dass für alle Punkte $\mathbf{x} \in U_\varepsilon(\mathbf{b})$ die Punkte $\mathbf{f}^k(\mathbf{x}, \mathbf{u})$ gegen $\mathbf{b}$ konvergieren.

Wähle nun ein beliebiges $\mathbf{x} \in X$. Der Punkt $\mathbf{x}$ liegt dann in einer Umgebung $U_{v \cdot \frac{\varepsilon}{2}}(\mathbf{b}) = \{\mathbf{x} \in X \mid \|\mathbf{x} - \mathbf{b}\| < v \cdot \frac{\varepsilon}{2}\}$, wobei $v \in \mathbb{N}$ minimal sein soll. Im Folgenden wird durch vollständige Induktion über v gezeigt, dass auch die Punkte $\mathbf{f}^k(\mathbf{x}, \mathbf{u})$ gegen $\mathbf{b}$ konvergieren:

Für $v = 1$ und $v = 2$ ist die Behauptung nach Voraussetzung erfüllt. Es wird nun angenommen, dass die Behauptung bis zu einem bestimmten Wert v gilt. Es bleibt nur noch zu zeigen, dass die Behauptung auch für $v + 1$ gilt.

Wähle dazu ein beliebiges $\mathbf{x} \in U_{(v+1) \cdot \frac{\varepsilon}{2}}(\mathbf{b})$. Suche ein $\mathbf{a} \in U_{v \cdot \frac{\varepsilon}{2}}(\mathbf{b})$ mit $\|\mathbf{a} - \mathbf{x}\| \leq \frac{\varepsilon}{2}$. Dann gilt nach Induktionsannahme, dass $\mathbf{f}^k(\mathbf{a}, \mathbf{u}) \rightarrow \mathbf{b}$ und damit dass ab einem bestimmten Index N für alle $k > N$ die Punkte $\mathbf{f}^k(\mathbf{a}, \mathbf{u})$ in $U_{\frac{\varepsilon}{2}}(\mathbf{b})$ liegen. Da die Übergangsfunktion $\mathbf{f}$ nach Satz 13 nichtexpandierend bezüglich der kanonischen Norm ist, gilt auch $\|\mathbf{f}^k(\mathbf{a}, \mathbf{u}) - \mathbf{f}^k(\mathbf{x}, \mathbf{u})\| \leq \frac{\varepsilon}{2}$ für alle k . Damit liegen auch die Vektoren $\mathbf{f}^k(\mathbf{x}, \mathbf{u})$ in $U_\varepsilon(\mathbf{b})$ für alle $k > N$ und konvergieren gegen $\mathbf{b}$, da $\mathbf{b}$ asymptotisch stabil ist. $\square$

Für autonome rekurrente Fuzzy-Systeme und für solche, deren Eingabevektor $\mathbf{u}$ direkt auf einem Kernpositionsvektor liegt, kann man asymptotisch stabile Ruhelagen in regelstetigen standardisierten rekurrenten Fuzzy-Systemen sehr einfach finden, wie folgender Satz zeigt:

Satz 20

In autonomen, regelstetigen standardisierten rekurrenten Fuzzy-Systemen liegt jede asymptotisch stabile Ruhelage auf einem Kernpositionsvektor und die folgenden Aussagen sind äquivalent:

- (1) *Es gibt eine asymptotisch stabile Ruhelage.*
- (2) *Es gibt (genau) eine global asymptotisch stabile Ruhelage.*
- (3) *Es gibt genau eine Ruhelage und das rekurrente Fuzzy-System ist konvergent.*
- (4) *Es gibt genau einen Endknoten und das rekurrente Fuzzy-System ist terminiert.*
- (5) *Das rekurrente Fuzzy-System ist terminiert und besitzt einen Endknoten, der keine weiteren Endknoten in der Nachbarschaft hat.*
- (6) *Es gibt einen Endknoten, dessen benachbarte linguistische Vektoren alle gegen diesen (einzigen) Endknoten konvergieren.*

Beweis:

Zuerst wird die Äquivalenz der Aussagen (1) bis (6) mit Hilfe eines Ringschlusses $(1) \Rightarrow (2) \Rightarrow (3) \Rightarrow (4) \Rightarrow (5) \Rightarrow (6) \Rightarrow (1)$ gezeigt. Damit folgt aus jeder der Aussagen jede andere der Aussagen und somit sind alle diese Aussagen äquivalent.

Nach Satz 19 folgt aus (1) die Aussage (2). Aus (2) folgt unmittelbar (3).

Es gelte (3). Da das rekurrente Fuzzy-System konvergent ist, muss es auch terminiert sein. Also gibt es einen Endknoten. Der Kernpositionsvektor eines Endknotens ist zugleich eine Ruhelage. Da es genau eine Ruhelage gibt, kann es auch nur einen Endknoten geben. Also gilt (4).

Es gelte (4). Es gibt daher genau einen Endknoten und damit auch keine weiteren Endknoten in der Nachbarschaft dieses Endknotens. Also gilt (5).

Es gelte (5). Alle linguistischen Vektoren in der Nachbarschaft des in (5) genannten Endknotens können diese Nachbarschaft nicht verlassen, da sie aufgrund der Regelstetigkeit entweder direkt auf den Endknoten oder auf einen zum Endknoten benachbarten linguistischen Vektor abgebildet werden. Des Weiteren konvergieren diese linguistischen Vektoren gegen einen Endknoten, weil das rekurrente Fuzzy-System terminiert ist. Also konvergieren diese linguistischen Vektoren gegen den in (5) genannten Endknoten, da es ja in seiner Nachbarschaft keinen weiteren Endknoten gibt. Also gilt (6).

Es gelte (6). Nach Satz 17 gibt es in einer ε -Umgebung mit $\varepsilon \leq 1$ um den Kernpositionsvektor des Endknotens keine weiteren Ruhelagen, denn eine solche läge in einem elementaren Hyperquader, dessen Eckpunkte Ruhelagen wären und damit gäbe es in der Nachbarschaft des Endknotens einen weiteren Endknoten im Widerspruch zu Aussage (6). Punkte aus der ε -Umgebung des Endknotens können diese Umgebung aufgrund der Nichtexpansion von $\mathbf{f}$ nicht verlassen, müssen aber nach Satz 15 gegen eine Ruhelage konvergieren. Also konvergieren sie alle gegen die einzige Ruhelage in der ε -Umgebung, die damit asymptotisch stabil ist. Also gilt (1).

Damit ist der Ringschluss abgeschlossen und die Aussagen (1) bis (6) sind äquivalent.

Wenn es eine asymptotisch stabile Ruhelage gibt (1), dann gibt es nach Aussage (2) nur eine einzige und diese muss dann ein Kernpositionsvektor eines Endknotens sein, um Aussage (4) erfüllen zu können. Also liegt auch jede asymptotisch stabile Ruhelage auf einem Kernpositionsvektor. $\square$

Mit Hilfe von Satz 20 kann man sehr einfach in solchen Systemen alleine anhand der Regelbasis bzw. des Zustandsgraphen feststellen, ob es eine global asymptotisch stabile Ruhelage gibt und wo diese zu finden ist.

Wenn es eine asymptotische Ruhelage gibt, so muss sie die Kernposition eines Endknotens sein. Hat man einen Endknoten gefunden, so muss man nur nach Kriterium (6) prüfen, ob alle linguistischen Vektoren in der Nachbarschaft gegen diesen Endknoten konvergieren. Zum einen folgt dann, dass die gesamte Regelbasis nach Kriterium (5) terminiert ist. Zum anderen lässt sich feststellen, dass das rekurrente Fuzzy-System nach Kriterium (3) konvergent ist und eine global asymptotisch stabile Ruhelage besitzt, was Kriterium (2) besagt.

Standardisierte rekurrente Fuzzy-Systeme lassen sich nicht nur einfach nach asymptotisch stabilen Ruhelagen untersuchen, sondern haben eine weitere angenehme Eigenschaft, wenn sie regelstetig sind:

Satz 21

Regelstetige standardisierte rekurrente Fuzzy-Systeme sind nicht chaotisch.

Beweis:

Es gibt verschiedene Definitionen von Chaos in zeitdiskreten dynamischen Systemen. Die bekanntesten sind Chaos im Sinne von Li und Yorke [58, 86], Chaos im Sinne von Devaney [20] und Chaos im Sinne von Block und Coppel [14].

Die verschiedenen Definitionen von Chaos oder daraus abgeleitete Eigenschaften haben gemeinsam, dass es einen Abstand $\delta > 0$ gibt und man beliebig nahe beieinanderliegende Startwerte $\mathbf{a}$ und $\mathbf{b}$ finden kann, deren Trajektorien irgendwann einen Abstand voneinander haben, der δ überschreitet. Der Abstand wird dabei aus einer Norm $\|\cdot\|_s$ der Differenzvektoren $\mathbf{f}^k(\mathbf{a}, \mathbf{u})$ und $\mathbf{f}^k(\mathbf{b}, \mathbf{u})$ bestimmt. Da in endlichdimensionalen Vektorräumen Normen äquivalent sind [53], lassen sich die Norm $\|\cdot\|_s$ und die kanonische Norm $\|\cdot\|$ gegenseitig durch Konstanten $0 < m < M$ wie folgt abschätzen: $m\|x\|_s < \|x\| < M\|x\|_s$.

Angenommen, ein regelstetiges standardisiertes rekurrentes Fuzzy-System wäre chaotisch. Dann gäbe es ein $\delta > 0$ und Startvektoren $\mathbf{a}$ und $\mathbf{b}$ mit einem Anfangsabstand $\|\mathbf{a} - \mathbf{b}\|_s < \delta m / M$, deren Bilder nach k Iterationen einen Abstand größer als δ hätten. Dann würde folgende Ungleichungskette gelten:

$$\|\mathbf{a} - \mathbf{b}\| < M\|\mathbf{a} - \mathbf{b}\|_s < m\delta < m\|\mathbf{f}^k(\mathbf{a}, \mathbf{u}) - \mathbf{f}^k(\mathbf{b}, \mathbf{u})\|_s < \|\mathbf{f}^k(\mathbf{a}, \mathbf{u}) - \mathbf{f}^k(\mathbf{b}, \mathbf{u})\|. \quad (6.40)$$

Allerdings widerspricht die resultierende Ungleichung $\|\mathbf{a} - \mathbf{b}\| < \|\mathbf{f}^k(\mathbf{a}, \mathbf{u}) - \mathbf{f}^k(\mathbf{b}, \mathbf{u})\|$ der Ungleichungskette $\|\mathbf{a} - \mathbf{b}\| \geq \|\mathbf{f}(\mathbf{a}, \mathbf{u}) - \mathbf{f}(\mathbf{b}, \mathbf{u})\| \geq \dots \geq \|\mathbf{f}^k(\mathbf{a}, \mathbf{u}) - \mathbf{f}^k(\mathbf{b}, \mathbf{u})\|$, die sich durch die wiederholte Anwendung von Satz 13 ergibt. Also kann das regelstetige standardisierte rekurrente Fuzzy-System nicht chaotisch sein. $\square$

7 Erreichbarkeit und Steuerbarkeit

Bisher wurde immer vorausgesetzt, dass die Eingangsvektoren für die betrachteten rekurrenten Fuzzy-Systeme fest vorgegeben sind. Dann wurde untersucht, wie sich die Eingangsgrößen auf den zeitlichen Verlauf der Zustandsgrößen x_i auswirken. In diesem Kapitel wird ein komplementäres Problem behandelt: Gegeben sind Zustandsvektoren zu zwei unterschiedlichen Zeitschritten. Es werden nun die Folgen von Eingangsvektoren gesucht, mit denen man von dem ersten zum zweiten vorgegebenen Zustandsvektor gelangen kann. Dazu werden die Erreichbarkeit und Steuerbarkeit von Zustandsvektoren untersucht.

Bei linearen Systemen sind Erreichbarkeit und Steuerbarkeit Systemeigenschaften [36]. Bei nichtlinearen Systemen sind sie für lokale Bereiche des Zustandsraumes definiert. Bei der Untersuchung der Erreichbarkeit wird analysiert, in welche Bereiche des Zustandsraums man von einer Startmenge von Zuständen aus vordringen kann. Die Steuerbarkeit befasst sich mit der entgegengesetzten Fragestellung: Von welchen Startzuständen aus kann man zu einem festgelegten Endzustand gelangen? Und wie sind diese Steuergrößen am geeignetsten zu wählen?

Zu Beginn des Kapitels werden die Erreichbarkeitsmengen definiert und einige grundlegende Eigenschaften dieser Mengen vorgestellt. Dann werden die Ergebnisse auf rekurrente Fuzzy-Systeme angewandt. Dabei wird im Speziellen untersucht, welche Aussagen schon bei der Betrachtung des Zustandsgraphen eines rekurrenten Fuzzy-Systems getroffen werden können. Anschließend wird die Steuerbarkeit als duales Problem zur Erreichbarkeit betrachtet.

7.1 Erreichbarkeit: Definitionen und Grundlagen

Die Untersuchung der Erreichbarkeit in einem dynamischen System versucht die folgende Frage zu beantworten: Welche Zustandsvektoren $\mathbf{x}(n)$ kann man von einem Startvektor $\mathbf{x}(0)$ aus in n Schritten erreichen, wenn die Eingangswerte aus einer Menge U_0 gewählt werden können?

Ausgehend von einem Startvektor $\mathbf{x}(0)$ erreicht man bei verschiedenen Eingangsvektoren $\mathbf{u}(0) \in U_0$ normalerweise eine Reihe von möglichen Folgezuständen. Diese Folgezustände werden in der so genannten *Erreichbarkeitsmenge* $E_1(\mathbf{x}(0), U_0)$ zusammengefasst. Sie ist durch die Gleichung

$$E_1(\mathbf{x}(0), U_0) = \bigcup_{\mathbf{u}(0) \in U_0} \{f(\mathbf{x}(0), \mathbf{u}(0))\} \quad (7.1)$$

festgelegt. In der Erreichbarkeitsmenge $E_n(\mathbf{x}(0), U_0)$ werden alle Zustandsvektoren $\mathbf{x}(n)$ zusammengefasst, die man, ausgehend von einem Zustandsvektor $\mathbf{x}(0)$, mit verschiedenen Folgen von Eingangsvektoren $\mathbf{u}(k) \in U_0$ nach n Schritten erreicht.

Die Erreichbarkeitsmengen lassen sich auch für Startmengen X_0 durch

$$E_n(X_0, U_0) = \bigcup_{\mathbf{x}(0) \in X_0} E_n(\mathbf{x}(0), U_0) \quad (7.2)$$

definieren. Sie geben an, welche Zustandsvektoren man bei geeigneter Wahl von Steuervektoren nach n Schritten erreichen kann, wenn man zusätzlich den Startvektor $\mathbf{x}(0)$ aus der Startmenge X_0 frei wählen kann.

Mit diesen Definitionen kann man die folgende iterative Berechnungsvorschrift für Steuerbarkeitsmengen $E_n(X_0, U_0)$ formulieren:

Lemma 4

Es gilt $E_{n+1}(X_0, U_0) = E_1(E_n(X_0, U_0), U_0)$.

Beweis:

Jeden Zustandsvektor $\mathbf{x}(n+1)$ aus der Erreichbarkeitsmenge $E_{n+1}(X_0, U_0)$ erreicht man nach $n+1$ Zeitschritten von einem geeigneten Startvektor $\mathbf{x}(0) \in X_0$ aus unter Verwendung einer geeigneten Folge von Eingangsvektoren $\mathbf{u}(k) \in U_0$ mit $k = 1, \dots, n+1$. Nach n Schritten wird so von $\mathbf{x}(0)$ ein Zustandsvektor $\mathbf{x}(n)$ erreicht, der in $E_n(X_0, U_0)$ liegt und von dem aus man mit einem Eingangsvektor von $\mathbf{u}(n+1) \in U_0$ den Vektor $\mathbf{x}(n+1)$ erreicht. Also ist $E_{n+1}(X_0, U_0) \subset E_1(E_n(X_0, U_0), U_0)$.

Umgekehrt gibt es für jeden Vektor $\mathbf{x}(n+1)$ aus der Menge $E_1(E_n(X_0, U_0), U_0)$ einen Eingangswert $\mathbf{u}(n+1) \in U_0$, mit dem der Punkt $\mathbf{x}(n+1)$ von einem geeigneten Vektor $\mathbf{x}(n)$ aus der Menge $E_n(X_0, U_0)$ erreicht wird. Für den Vektor $\mathbf{x}(n)$ gibt es wiederum eine Folge von Eingangsvektoren $\mathbf{u}(k) \in U_0$ mit $k = 1, \dots, n$, die für das Erreichen eines Vektors $\mathbf{x}(n) \in E_n(X_0, U_0)$, ausgehend von einem geeigneten Startvektor $\mathbf{x}(0) \in X_0$, verwendet wird. Der Vektor $\mathbf{x}(n+1)$ wird also von einem geeigneten Vektor $\mathbf{x}(0) \in X_0$ unter Verwendung einer Eingangsfolge $\mathbf{u}(k) \in U_0$ mit $k = 1, \dots, n+1$ erreicht und liegt daher in $E_{n+1}(X_0, U_0)$. Also ist $E_1(E_n(X_0, U_0), U_0) \subset E_{n+1}(X_0, U_0)$. $\square$

Für die Erreichbarkeitsmengen gilt außerdem die folgende Teilmengenbeziehung:

Lemma 5

Es gilt $E_n(X_1, U_1) \subseteq E_n(X_2, U_2)$ für $X_1 \subseteq X_2, U_1 \subseteq U_2$.

Beweis:

Der Beweis wird durch vollständige Induktion geführt. Für den Induktionsbeginn wird die Aussage des Lemmas für $n = 1$ gezeigt: Mit der Definition der Erreichbarkeitsmengen gilt nach Gleichungen (7.1) und (7.2):

$$E_1(X_1, U_1) = \bigcup_{\mathbf{x} \in X_1} \bigcup_{\mathbf{u} \in U_1} f(\mathbf{x}, \mathbf{u}) \subseteq \bigcup_{\mathbf{x} \in X_2} \bigcup_{\mathbf{u} \in U_2} f(\mathbf{x}, \mathbf{u}) = E_1(X_2, U_2) \quad (7.3)$$

für $X_1 \subseteq X_2$ und $U_1 \subseteq U_2$.

Es wird nun angenommen, dass das Lemma bis zu einem Index k gelte. Dann ist $E_k(X_1, U_1) \subseteq E_k(X_2, U_2)$ und die beiden Mengen erfüllen die gleiche Teilmengenbeziehung wie X_1 und X_2 . Damit gilt Gleichung (7.3) auch, wenn man statt X_1 und X_2 die Mengen $E_k(X_1, U_1)$ und $E_k(X_2, U_2)$ einsetzt. Zusammen mit Lemma 4 folgt dann der Induktionsschritt:

$$E_{k+1}(X_1, U_1) = E_1(E_k(X_1, U_1), U_1) \subseteq E_1(E_k(X_2, U_2), U_2) = E_{k+1}(X_2, U_2). \quad (7.4)$$

Nach dem Prinzip der vollständigen Induktion gilt damit die Aussage des Lemmas für alle $n \in \mathbb{N}$. $\square$

Gelten die Teilmengenbeziehungen innerhalb einer Folge von Erreichbarkeitsmengen, so hat dies Auswirkungen auf das Verhalten für alle weiteren Zeitschritte. Dies wird in nachfolgendem Lemma präzisiert:

Lemma 6

Gilt für ein $n \in \mathbb{N}$ die Mengeninklusion

$$E_{n+1}(X_0, U_0) \subseteq E_n(X_0, U_0) \text{ bzw. } E_{n+1}(X_0, U_0) \supseteq E_n(X_0, U_0), \quad (7.5)$$

so gilt auch für alle $k \geq n$:

$$E_k(X_0, U_0) \subseteq E_n(X_0, U_0) \text{ bzw. } E_k(X_0, U_0) \supseteq E_n(X_0, U_0). \quad (7.6)$$

Ist $E_n(X_0, U_0) = E_{n-1}(X_0, U_0)$, so gilt $E_k(X_0, U_0) = E_n(X_0, U_0)$ für alle $k \geq n$.

Beweis:

Es gelte $E_{n+1}(X_0, U_0) \subseteq E_n(X_0, U_0)$. Setzt man $X_1 = E_{n+1}(X_0, U_0)$ und $X_2 = E_n(X_0, U_0)$, so folgt aus Lemma 4 und Lemma 5, dass $E_{n+2}(X_0, U_0) = E_1(X_1, U_0) \subseteq E_1(X_2, U_0) = E_{n+1}(X_0, U_0)$. Per Induktion folgt für alle $k \geq n$, dass $E_{k+1}(X_0, U_0) \subseteq E_k(X_0, U_0)$. Aufgrund der Inklusionskette $\dots \subseteq E_{n+2}(X_0, U_0) \subseteq E_{n+1}(X_0, U_0) \subseteq E_n(X_0, U_0)$ gilt für alle $k \geq n$, dass $E_k(X_0, U_0) \subseteq E_n(X_0, U_0)$ ist.

Gilt $E_{n+1}(X_0, U_0) \supseteq E_n(X_0, U_0)$, so folgt analog zum Fall $E_{n+1}(X_0, U_0) \subseteq E_n(X_0, U_0)$, dass für alle $k \geq n$ die Mengeninklusion $E_{k+1}(X_0, U_0) \supseteq E_k(X_0, U_0)$ gilt, und auch $E_k(X_0, U_0) \supseteq E_n(X_0, U_0)$ für alle $k \geq n$ ist.

Gilt $E_n(X_0, U_0) = E_{n-1}(X_0, U_0)$, so folgt für alle $k \geq n$, dass einerseits $E_k(X_0, U_0) \subseteq E_n(X_0, U_0)$ und andererseits $E_k(X_0, U_0) \supseteq E_n(X_0, U_0)$, also $E_k(X_0, U_0) = E_n(X_0, U_0)$ für alle $k \geq n$. $\square$

In den weiteren Untersuchungen werden oft die Startmenge X_0 und die Menge U_0 der Eingabevektoren kompakt und zusammenhängend sein und die Transitionsfunktion $\mathbf{f}$ des betrachteten dynamischen Systems wird stetig sein. Dann ist folgendes Lemma anwendbar:

Lemma 7

Bei einer stetigen Transitionsfunktion $\mathbf{f}$ sind die Erreichbarkeitsmengen $E_n(X_0, U_0)$, ausgehend von einer kompakten bzw. zusammenhängenden Startmenge X_0 , unter Verwendung einer kompakten bzw. zusammenhängenden Eingabemenge U_0 selbst kompakt bzw. zusammenhängend.

Beweis:

Nach der Voraussetzung sind X_0 und U_0 kompakt bzw. zusammenhängend. Nach Satz 160.2 aus [32] ist das Bild der zusammenhängenden Menge $X_0 \times U_0$ wieder zusammenhängend. Nach Satz 158.5 aus [32] ist das Bild der kompakten Menge $X_0 \times U_0$ wieder kompakt. Damit ist $E_1(X_0, U_0) = \mathbf{f}(X_0, U_0)$ kompakt bzw. zusammenhängend. Dann ist auch $E_2(X_0, U_0) = \mathbf{f}(E_1(X_0, U_0), U_0)$ kompakt bzw. zusammenhängend, usw... Mittels vollständiger Induktion folgt die Behauptung, dass auch $E_n(X_0, U_0)$ kompakt bzw. zusammenhängend ist. $\square$

Diese allgemeinen Grundlagen werden im nächsten Abschnitt auf rekurrente Fuzzy-Systeme angewendet.

7.2 Erreichbarkeit in rekurrenten Fuzzy-Systemen

Für die Untersuchung der Erreichbarkeit von Zuständen werden zuerst eindimensionale rekurrente Fuzzy-Systeme untersucht. Im Laufe dieses Abschnittes stellt sich heraus, dass sich die Erreichbarkeitsmengen in eindimensionalen, stetigen rekurrenten Fuzzy-Systemen sehr effizient berechnen lassen und dass man eine Abschätzung dieser Erreichbarkeitsmengen direkt aus der Regelbasis ablesen kann. Anschließend wird untersucht, inwieweit sich die Ergebnisse auf mehrdimensionale rekurrente Fuzzy-Systeme übertragen lassen.

Man betrachte die Erreichbarkeitsmengen von eindimensionalen, stetigen rekurrenten Fuzzy-Systemen, die sich für eine kompakte, zusammenhängende Startmenge X_0 und eine kompakte, zusammenhängende Eingabemenge U_0 ergeben. Nach Lemma 7 sind dann alle Erreichbarkeitsmengen selbst kompakt und zusammenhängend. Da zusätzlich der Zustandsraum eindimensional ist, sind die Erreichbarkeitsmengen $E_n(X_0, U_0)$ kompakte Intervalle $E_n(X_0, U_0) = [x_{\min}(n), x_{\max}(n)]$. Für ihre Bestimmung sind also nur ihre Extremwerte $x_{\min}(n)$ und $x_{\max}(n)$ zu ermitteln. Dies ist sehr einfach möglich, denn es gilt:

Lemma 8

Jede Komponente der Transitionsfunktion $\mathbf{f}(\mathbf{x}, \mathbf{u})$ eines rekurrenten Fuzzy-Systems ist innerhalb eines elementaren Hyperquaders monoton in den Variablen x_i und u_p .

Beweis:

Es genügt, Lemma 8 für eine beliebige Komponente f_l von $\mathbf{f}$ zu zeigen. Außerdem kann der Beweis für jede Komponente u_p und x_i gleichermaßen geführt werden. Deshalb reicht es aus zu zeigen, dass eine beliebige Komponente f_l innerhalb eines Hyperquaders H monoton in einer Variablen x_h ist. Die Komponenten u_p und $x_i \neq x_h$ sollen beliebige, aber feste Werte annehmen.

Die Abhängigkeit der Komponente f_l von der Komponente x_h ist nach Gleichung (3.13) durch

$$f_l(x_h) = \sum_{\hat{\mathbf{j}}, \hat{\mathbf{q}}} s_{w_l(\hat{\mathbf{j}}, \hat{\mathbf{q}})}^{x_l} \prod_{i \neq h} \mu_{\hat{\mathbf{j}}_i}^{x_i}(x_i) \prod_p \mu_{\hat{\mathbf{q}}_p}^{u_p}(u_p) \cdot \mu_{\hat{\mathbf{j}}_h}^{x_h}(x_h) \quad (7.7)$$

gegeben, wobei der Faktor $\mu_{\hat{\mathbf{j}}_h}^{x_h}(x_h)$ explizit aus dem großen Produkt herausgezogen wurde.

Liegt x_h speziell auf Kernpositionen $s_{\hat{\mathbf{j}}_h}^{x_h}$, so gilt nach der Rückkopplungskorrespondenzbe-

dingung (B4') $\mu_{\hat{j}_h}^{x_h}(s_{j_h}^{x_h}) = \delta_{\hat{j}_h, j_h}$, wobei $\delta_{\hat{j}_h, j_h}$ das Kroneker-Symbol ist, welches für $\hat{j}_h = j_h$ den Wert eins und für $\hat{j}_h \neq j_h$ den Wert null hat. Damit lässt sich der Funktionswert an der Stelle $x_h = s_{j_h}^{x_h}$ dann wie folgt ausdrücken:

$$f_l(s_{j_h}^{x_h}) = \sum_{\hat{\mathbf{j}}, \hat{\mathbf{q}}} s_{w_l(\hat{\mathbf{j}}, \hat{\mathbf{q}})}^{x_l} \prod_{i \neq h} \mu_{\hat{j}_i}^{x_i}(x_i) \prod_p \mu_{\hat{q}_p}^{u_p}(u_p) \cdot \delta_{\hat{j}_h, j_h}. \quad (7.8)$$

Mit Gleichung (7.8) wird nachfolgend gezeigt, dass sich $f_l(x_h)$ an einer beliebigen Stelle x_h durch eine gewichtete Summe aus den Funktionswerten $f_l(s_{j_h}^{x_h})$ darstellen lässt. Dazu betrachte man den Ausdruck

$$\sum_{j_h} f_l(s_{j_h}^{x_h}) \cdot \mu_{j_h}^{x_h}(x_h) = \sum_{j_h} \sum_{\hat{\mathbf{j}}, \hat{\mathbf{q}}} s_{w_l(\hat{\mathbf{j}}, \hat{\mathbf{q}})}^{x_l} \prod_{i \neq h} \mu_{\hat{j}_i}^{x_i}(x_i) \prod_p \mu_{\hat{q}_p}^{u_p}(u_p) \cdot \delta_{\hat{j}_h, j_h} \cdot \mu_{j_h}^{x_h}(x_h). \quad (7.9)$$

Bei der Summation über den Index j_h auf der rechten Seite von Gleichung (7.9) sind aufgrund des Kroneker-Deltas alle Summanden für Indizes $j_h \neq \hat{j}_h$ gleich null. Es bleiben also genau die Summanden übrig, bei denen $j_h = \hat{j}_h$ ist. Genau diese Summanden stehen auch auf der rechten Seite von Gleichung (7.7). Damit sind die Ausdrücke in Gleichung (7.9) und Gleichung (7.7) identisch und es gilt

$$f_l(x_h) = \sum_{j_h} f_l(s_{j_h}^{x_h}) \cdot \mu_{j_h}^{x_h}(x_h). \quad (7.10)$$

Innerhalb des betrachteten Hyperquaders H liegt die Komponente x_h zwischen zwei Kernpositionen $s_r^{x_h}$ und $s_{r+1}^{x_h}$. Damit sind die Zugehörigkeitsfunktionen $\mu_{j_h}^{x_h}(x_h)$ maximal für die zwei Indizes $j_h = r$ und $j_h = r+1$ von null verschieden und es gilt aufgrund der Normierungsbedingung nach Gleichung (3.12):

$$\mu_{r+1}^{x_h}(x_h) = 1 - \mu_r^{x_h}(x_h). \quad (7.11)$$

Setzt man dies in Gleichung (7.10) ein und führt man die Summation über j_h durch, so erhält man

$$f_l(x_h) = f_l(s_r^{x_h}) \mu_r^{x_h}(x_h) + f_l(s_{r+1}^{x_h}) \mu_{r+1}^{x_h}(x_h) = (f_l(s_r^{x_h}) - f_l(s_{r+1}^{x_h})) \mu_r^{x_h}(x_h) + f_l(s_{r+1}^{x_h}). \quad (7.12)$$

Da $\mu_r^{x_h}(x_h)$ zwischen $s_r^{x_h}$ und $s_{r+1}^{x_h}$ monoton fallend ist, ist $f_l(x_h)$ je nach Vorzeichen des konstanten Wertes $(f_l(s_r^{x_h}) - f_l(s_{r+1}^{x_h}))$ monoton fallend oder monoton steigend. In beiden Fällen ist die Funktion $f_l(x_h)$ monoton in x_h . $\square$

Lemma 8 gilt allgemein für rekurrente Fuzzy-Systeme. Man kann es aber sehr gut für die Berechnung der Extremwerte $x_{\min}(n)$ und $x_{\max}(n)$ von Erreichbarkeitsmengen $E_n(X_0, U_0) = [x_{\min}(n), x_{\max}(n)]$ bei eindimensionalen stetigen rekurrenten Fuzzy-Systemen ausnutzen. Wie dies möglich ist, ist in folgendem Satz zusammengefasst:

Satz 22 (Berechnung von Erreichbarkeitsmengen)

Gegeben ist ein eindimensionales, stetiges rekurrentes Fuzzy-System. Die Startmenge $X_0 = [x_{\min}(0), x_{\max}(0)]$ sei ein kompaktes Intervall des Zustandsraumes X . Die Eingabemenge $U_0 = U_0^{(1)} \times \dots \times U_0^{(n)}$ sei ein direktes Produkt von kompakten Intervallen

$U_0^{(p)} = [u_{p,min}, u_{p,max}]$, also ein achsenparalleler Hyperquader aus dem Eingaberaum U .

Dann ist die Erreichbarkeitsmenge $E_1(X_0, U_0) = f(X_0, U_0) = [x_{min}(1), x_{max}(1)]$ ein kompaktes Intervall mit den Grenzen $x_{min}(1) = \min\{f(\hat{X}_0, \hat{U}_0)\}$ und $x_{max}(1) = \max\{f(\hat{X}_0, \hat{U}_0)\}$, wobei $\hat{U}_0 = \hat{U}_0^{(1)} \times \dots \times \hat{U}_0^{(n)}$ ist und $\hat{X}_0$ und $\hat{U}_0^{(p)}$ endliche Mengen sind, die sich aus den Endpunkten der entsprechenden Intervalle X_0 und $U_0^{(p)}$ und allen dazwischenliegenden Kernpositionen zusammensetzen.

Beweis:

Nach Lemma 7 ist die Erreichbarkeitsmenge eine zusammenhängende, kompakte Menge aus dem Zustandsraum X . Da der Zustandsraum X eindimensional ist, ist die Erreichbarkeitsmenge $E_1(X_0, U_0) = f(X_0, U_0) = [x_{min}(1), x_{max}(1)]$ ein kompaktes Intervall.

Die Intervallgrenzen $x_{min}(1)$ und $x_{max}(1)$ sind die Extremwerte von $f(X_0, U_0)$. Da nach Lemma 8 die Funktion innerhalb eines jeden elementaren Hyperquaders monoton ist, werden die Extremwerte an den Rändern angenommen. So bleiben nur die Punkte $(\hat{x}, \hat{u})$ mit $\hat{x} \in \hat{X}$ und $\hat{u} \in \hat{U}$ als Kandidaten für die Extremstellen übrig. Damit folgen die im Satz angegebenen Gleichungen. $\square$

Als Beispiel wird das chaotische Modell der Insektenpopulationsgröße betrachtet, welches in Kapitel 3 eingeführt wurde. In Abbildung 7.1(a) ist die Regelbasis mit linguistischen Werten L_j^x und L_q^u dargestellt. Die entsprechenden Kernpositionen s_j^x und s_q^u liegen bei $s_1^x = 5$, $s_2^x = 6$, $s_3^x = 7$ und $s_1^u = 1$, $s_2^u = 2$, $s_3^u = 3$. In Abbildung 7.1(b) ist der Raum $X \times U$ gezeigt. Die Kernpositionsvektoren (s_j^x, s_q^u) in $X \times U$ sind durch Punkte "•" markiert. An den Kernpositionsvektoren sind die Funktionswerte angeschrieben, die zugleich die Kernpositionswerte s_w^x der entsprechenden linguistischen Werte L_w^x aus der Regelbasis sind. Gesucht wird nun die Erreichbarkeitsmenge $E_1(X_0, U_0)$ mit dem Startintervall $X_0 = [5, 6.4]$ für Eingangswerte u aus dem Intervall $U_0 = [1, 2.5]$. Der Bereich $X_0 \times U_0$ ist in Abbildung 7.1(b) durch den eingezeichneten Quader gekennzeichnet. Für die Berechnung der Extremwerte von $E_1(X_0, U_0) = [x_{min}(1), x_{max}(1)]$ muss die Funktion aufgrund von Satz 22 nur an den neun Punkten von $\hat{X}_0 \times \hat{U}_0$ ausgewertet werden. Diese Punkte sind in Abbildung 7.1(b) mit einer Raute "◊" markiert. Bei der Verwendung von dreiecksförmigen Zugehörigkeitsfunktionen ergeben sich die angegebenen Funktionswerte. Auf einen Blick erkennt man die Extremwerte und so ergibt sich $E_1(X_0, U_0) = [5, 6.5]$.

Wenn die Intervalle X_0 und $U_0^{(p)}$ aus Satz 22 selbst durch Kernpositionen beschränkt werden, so enthalten die Mengen $\hat{X}_0$ und $\hat{U}_0^{(p)}$ ausschließlich Kernpositionen. Da nach Lemma 1 ein rekurrentes Fuzzy-System, welches auf Kernpositionen beschränkt ist, äquivalent zu einem Automaten ist, sind die Funktionswerte $f(\hat{X}_0, \hat{U}_0)$ selbst wieder Kernpositionen. Im Speziellen sind dann $x_{min}(1) = \min\{f(\hat{X}_0, \hat{U}_0)\}$ und $x_{max}(1) = \max\{f(\hat{X}_0, \hat{U}_0)\}$ auch Kernpositionen. Damit wird die Erreichbarkeitsmenge $E_1(X_0, U_0)$, genau wie die Ausgangsmenge X_0 , durch Kernpositionen beschränkt. Dies setzt sich für alle Erreichbarkeitsmengen $E_n(X_0, U_0)$ fort.

Nach Lemma 1 entsprechen die Kernpositionswerte aus $f(\hat{X}_0, \hat{U}_0)$ linguistischen Werten aus der Regelbasis. Da nur Funktionswerte von Kernpositionsvektoren bei der Auswertung von $f(\hat{X}_0, \hat{U}_0)$ betrachtet werden müssen, können diese alleine anhand der Regelbasis bestimmt werden.

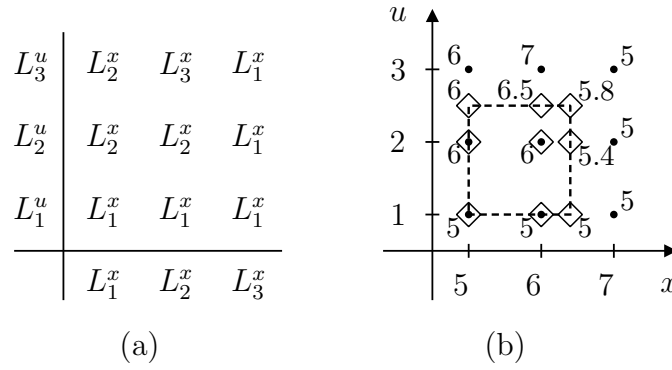
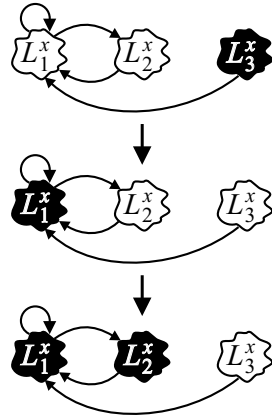


Abbildung 7.1: (a) Regelbasis für ein Insektenpopulationsmodell. (b) Produktraum $X \times U$ des Zustandsraums X und Eingangsraums U mit relevanten Funktionswerten $f(x, u)$ zur Bestimmung einer Erreichbarkeitsmenge $E_1(X_0, U_0)$.

$$U_0 = [s_1^u, s_2^u] \Leftrightarrow \{L_1^u, L_2^u\}$$



$$X_0 = [s_3^x, s_3^x] \Leftrightarrow \{L_3^x\}$$

$$E_1(X_0, U_0) = [s_1^x, s_1^x] \Leftrightarrow \{L_1^x\}$$

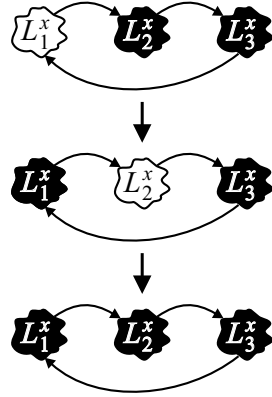
$$E_2(X_0, U_0) = [s_1^x, s_2^x] \Leftrightarrow \{L_1^x, L_2^x\}$$

Abbildung 7.2: Beispiel für die Berechnung von Erreichbarkeitsmengen $E_1(X_0, U_0)$ und $E_2(X_0, U_0)$ anhand eines Zustandsgraphen.

Bei einer vorliegenden Menge $\hat{X}_0 \times \hat{U}_0$ von Kernpositionsvektoren aus $X \times U$ betrachtet man die Menge der Regeln, die die entsprechenden linguistischen Vektoren (L_j^x, L_q^u) in der Prämisse stehen haben. Unter diesen Regeln sucht man sich den kleinsten und größten linguistischen Wert, der in der Konklusion auftaucht und hat damit mit den entsprechenden Kernpositionen die Extremwerte $x_{\min}(1)$ und $x_{\max}(1)$ bestimmt.

Abbildung 7.2 zeigt ein Beispiel für die Bestimmung der Erreichbarkeitsmengen $E_1(X_0, U_0)$ und $E_2(X_0, U_0)$ für die einelementige Startmenge $X_0 = [s_3^x, s_3^x] (\Leftrightarrow \{L_3^x\})$ und die Eingabemenge $U_0 = [s_1^u, s_2^u] (\Leftrightarrow \{L_1^u, L_2^u\})$ anhand eines Zustandsgraphen. Es wurden nur die Transitions Pfeile eingezeichnet, die für L_1^u oder L_2^u gelten. Erreichbare Zustände auf Kernpositionen sind durch geschwärzte Knoten der entsprechenden linguistischen Werte im Zustandsgraphen markiert. Folgt man den eingezeichneten Transitionen, so erreicht man mögliche Folgezustände. Daraus können die Erreichbarkeitsmengen abgelesen werden.

$$U_0 = [s_3^u, s_3^u] \Leftrightarrow \{L_3^u\}$$



$$X_0 = [s_2^x, s_3^x] \Leftrightarrow \{L_2^x, L_3^x\}$$

Zwischenschritt:

Erreichbarkeitsmenge im
linguistischen Automaten:
 $\{L_1^x, L_3^x\}$

$$E_1(X_0, U_0) = [s_1^x, s_3^x] \Leftrightarrow \{L_1^x, L_2^x, L_3^x\}$$

Abbildung 7.3: Beispiel für die Berechnung einer Erreichbarkeitsmenge $E_1(X_0, U_0)$ anhand eines Zustandsgraphen, bei der ein Zwischenschritt nötig ist.

Durch das Folgen der eingezeichneten Transitionen im Zustandsgraphen des linguistischen Automaten erreicht man nicht unbedingt alle linguistischen Zustände, die erreichbaren Kernpositionen im rekurrenten Fuzzy-System entsprechen. Aufgrund der interpolierenden Wirkung von stetigen rekurrenten Fuzzy-Systemen werden auch alle Zwischenwerte angenommen. Diese Zwischenwerte erreicht man gegebenenfalls nur beim Einsatz von Eingabektoren oder Zustandswerten, die nicht auf Kernpositionen liegen, die also nicht direkt linguistischen Werten oder Vektoren entsprechen und die damit im linguistischen Automaten nicht berücksichtigt werden. Bei der graphischen Analyse der Erreichbarkeitsmengen kann man dies einbeziehen: In einem Zwischenschritt werden zuerst die erreichbaren linguistischen Folgezustände im linguistischen Automaten bestimmt. Anschließend werden alle Knoten im Zustandsgraphen, die zwischen dem kleinsten und größten der so erreichten Zustandswerte liegen, durch eine Schwarzfärbung gekennzeichnet und somit verdeutlicht, dass auch die entsprechenden dazwischenliegenden Kernpositionen in einem stetigen rekurrenten Fuzzy-System erreicht werden können. Abbildung 7.3 zeigt dies anhand eines Beispiels. Das Einbeziehen der linguistischen Werte zwischen den linguistischen Extremwerten ist für die Errechnung der Erreichbarkeitsmenge im nächsten Zeitschritt wichtig, weil sich von diesen Zwischenwerten aus eventuell größere oder kleinere linguistische Werte erreichen lassen.

Schließlich kann man aufgrund der Teilmengenbeziehungen aus Lemma 5 beliebige Erreichbarkeitsmengen durch solche abschätzen, die sich für Start- und Eingabemengen ergeben, die durch Kernpositionen begrenzt sind. Anhand der Regelbasis aus Abbildung 7.1(a) wird dies beispielhaft demonstriert. Man betrachte die einelementige Startmenge $X_0 = [s_1^x, s_1^x] \Leftrightarrow \{L_1^x\}$ und die Eingabemenge $U_0 = [1, 2.5]$. Dann ist $\{L_1^u, L_2^u\} \Leftrightarrow [s_1^u, s_2^u] \subset [1, 2.5] \subset [s_1^u, s_3^u] \Leftrightarrow \{L_1^u, L_2^u, L_3^u\}$ und die Erreichbarkeitsmengen $E_n(X_0, U_0)$ können nach Lemma 5 durch $E_n(X_0, [s_1^u, s_2^u]) \subseteq E_n(X_0, U_0) \subseteq E_n(X_0, [s_1^u, s_3^u])$ abgeschätzt werden. In Tabelle 7.2 ist dies bis zur Erreichbarkeitsmenge $E_2(X_0, U_0)$ durchgespielt. Da in den betrachteten Fällen jeweils $E_3(X_0, U_0) = E_2(X_0, U_0)$ gilt, ist nach Lemma 6 auch $E_n(X_0, U_0) = E_2(X_0, U_0)$ für alle $n \geq 2$.

Tabelle 7.1: Durch linguistische Folgezustände (erste und fünfte Zeile) können die Erreichbarkeitsmengen bestimmt werden, für die X_0 und U_0 durch Kernpositionen begrenzt werden (zweite und vierte Zeile), die wiederum zur Abschätzung von allgemeinen Erreichbarkeitsmengen (mittlere Zeile) verwendet werden können.

	X_0	$E_1(X_0, U_0)$	$E_2(X_0, U_0)$	$E_{n \geq 2}(X_0, U_0)$
$L_q^u \in \{L_1^u, L_2^u\}$	$\{L_1^x\} \rightarrow$	$\{L_1^x, L_2^x\} \rightarrow$	$\{L_1^x, L_2^x\} \rightarrow \dots \rightarrow$	$\{L_1^x, L_2^x\}$
$\updownarrow$	$\updownarrow$	$\updownarrow$	$\updownarrow$	$\updownarrow$
$u \in [1, 2]$	$[5, 5] \rightarrow$	$[5, 6] \rightarrow$	$[5, 6] \rightarrow \dots \rightarrow$	$[5, 6]$
$\subseteq$	$\subseteq$	$\subseteq$	$\subseteq$	$\subseteq$
$u \in [1, 2.5]$	$[5, 5] \rightarrow$	$[5, 6] \rightarrow$	$[5, 6.5] \rightarrow \dots \rightarrow$	$[5, 6.5]$
$\subseteq$	$\subseteq$	$\subseteq$	$\subseteq$	$\subseteq$
$u \in [1, 3]$	$[5, 5] \rightarrow$	$[5, 6] \rightarrow$	$[5, 7] \rightarrow \dots \rightarrow$	$[5, 7]$
$\updownarrow$	$\updownarrow$	$\updownarrow$	$\updownarrow$	$\updownarrow$
$L_q^u \in \{L_1^u, L_2^u, L_3^u\}$	$\{L_1^x\} \rightarrow$	$\{L_1^x, L_2^x\} \rightarrow$	$\{L_1^x, L_2^x, L_3^x\} \rightarrow \dots \rightarrow$	$\{L_1^x, L_2^x, L_3^x\}$

Wie gesehen, ist die Bestimmung der Erreichbarkeitsmengen in eindimensionalen, stetigen rekurrenten Fuzzy-Systemen einfach möglich. Im restlichen Teil dieses Abschnittes sollen die Ergebnisse auf mehrdimensionale rekurrente Fuzzy-Systeme übertragen werden. Die Übertragung des Satzes 22 und seine Auswertung anhand der Regelbasis stehen dabei im Mittelpunkt.

Ausgangspunkt für die Übertragung sind die Grundlagen aus Abschnitt 7.1. Sie sind uneingeschränkt auf mehrdimensionale Systeme anwendbar. Insbesondere können Erreichbarkeitsmengen prinzipiell aufgrund von Lemma 5 abgeschätzt werden und die Mengeninklusionsbeziehung aus Lemma 6 genutzt werden. Des Weiteren sind nach Lemma 7 die Erreichbarkeitsmengen $E_n(X_0, U_0)$ kompakt bzw. zusammenhängend bei (mehrdimensionalen) stetigen rekurrenten Fuzzy-Systemen für kompakte bzw. zusammenhängende Mengen X_0 und U_0 .

Beim Wechsel von eindimensionalen zu mehrdimensionalen Zustandsräumen geht allerdings eine Eigenschaft von zusammenhängenden Mengen verloren, die stillschweigend für den Beweis von Satz 22 verwendet wurde: zusammenhängende Mengen aus eindimensionalen Zustandsräumen sind auch konvex. Die folgenden Grundlagen und weiterführende Eigenschaften konvexer Mengen sind in [96, 56] zu finden. Eine konvexe Menge M zeichnet sich dadurch aus, dass für je zwei beliebige Punkte $\mathbf{a}$ und $\mathbf{b}$ aus der Menge auch ihre Verbindungsline in der Menge liegt, d.h. alle Punkte $\lambda \mathbf{a} + (1 - \lambda) \mathbf{b} \in M$ für alle $\lambda \in [0, 1]$. Mehrdimensionale zusammenhängende Mengen sind im Allgemeinen dagegen nicht konvex. Beispiele sind ein Kreis in der Ebene oder der gedruckte Buchstabe "V".

Der Durchschnitt beliebig vieler konvexer Mengen ist wieder konvex. Bildet man den Durchschnitt aller konvexer Mengen, die eine vorgegebene Menge M enthalten, so erhält man die *konvexe Hülle* dieser Menge M . Die konvexe Hülle einer Menge M ist die kleinste konvexe Menge, die M enthält. Die konvexe Hülle eines Kreises in der Ebene ist die ausgefüllte Kreisscheibe. Die konvexe Hülle einer V-förmigen Menge ist das ausgefüllte

Dreieck, dessen Eckpunkte die Spitze und die Endpunkte des Buchstaben "V" sind. Eine konvexe Menge M ist zugleich ihre konvexe Hülle.

Mit diesen Vorüberlegungen ist es möglich, eine Abschwächung von Satz 22 zu formulieren, die in nachfolgendem Lemma zusammengefasst ist:

Lemma 9

Gegeben ist ein eindimensionales, stetiges rekurrentes Fuzzy-System mit Zustandsraum X und Eingaberaum U . Sei $H_0 = X_0 \times U_0$ ein achsenparalleler Hyperquader aus $X \times U$, der vollständig in einem elementaren Hyperquader liegt. Dann ist die Erreichbarkeitsmenge $E_1(X_0, U_0)$ eine Teilmenge der konvexen Hülle der Bildpunkte der Eckpunkte von H_0 .

Lemma 9 besagt, dass $E_1(X_0, U_0)$ gegebenenfalls nur eine Teilmenge und, nicht wie im Satz 22 gezeigt, identisch mit der konvexen Hülle der Eckpunkte von H_0 ist. Durch diese Abschwächung der Folgerung ist es aber möglich, die Voraussetzungen auch erheblich abzuschwächen. Wie im Weiteren gezeigt wird, gilt Lemma 9 auch für unstetige und mehrdimensionale rekurrente Fuzzy-Systeme. Dies wird in folgendem Satz festgehalten:

Satz 23 (Abschätzung von Erreichbarkeitsmengen)

Gegeben ist ein rekurrentes Fuzzy-System mit Zustandsraum X und Eingaberaum U . Sei $H_0 = X_0 \times U_0$ ein achsenparalleler Hyperquader aus $X \times U$, der vollständig in einem elementaren Hyperquader liegt. Dann ist die Erreichbarkeitsmenge $E_1(X_0, U_0)$ eine Teilmenge der konvexen Hülle der Bildpunkte der Eckpunkte von H_0 .

Beweis:

Man betrachte einen achsenparallelen Hyperquader $H_0 = X_0 \times U_0$ aus $X \times U$, der vollständig in einem elementaren Hyperquader H enthalten ist.

Die Erreichbarkeitsmenge $E_1(X_0, U_0)$ ist genau dann eine Teilmenge der konvexen Hülle der Bildpunkte der Eckpunkte von H_0 , wenn sich jeder Funktionswert $\mathbf{f}(\mathbf{x}, \mathbf{u})$ eines Punktes $(\mathbf{x}, \mathbf{u})$ innerhalb des Hyperquaders H_0 als Konvexkombination der Funktionswerte der Eckpunkte von H_0 darstellen lässt [96]. Eine Konvexkombination ist eine Linearkombination mit Vorfaktoren aus dem Intervall $[0, 1]$, die sich in Summe zu eins ergänzen.

Zuerst wird der Fall behandelt, dass H_0 ein Hyperquader ist, dessen Vektoren in einer Komponente, z.B. in der Komponente x_h , nur Werte aus einem Intervall $[p, s_{r+1}^{x_h}] \subseteq [s_r^{x_h}, s_{r+1}^{x_h}]$ annehmen können. Für solche Vektoren ist $\mu_{j_h}^{x_h}(x_h) = 0$ für $j_h \neq r$ und $j_h \neq r + 1$.

Man betrachte die Eckpunkte $(\mathbf{x}, \mathbf{u})$ des elementaren Hyperquaders H , die für $p > s_r^{x_h}$ nicht in H_0 enthalten sind, die also $x_h = s_{r+1}^{x_h}$, $x_i = s_{j_i}^{x_i}$ für $i \neq h$ und $u_p = s_{q_p}^{u_p}$ erfüllen. Ihre Funktionswerte $\mathbf{f}(\mathbf{x}, \mathbf{u})$ werden mit $\mathbf{a}(\mathbf{j}, \mathbf{q})$ bezeichnet.

Dann betrachte man die restlichen Eckpunkte $(\mathbf{x}, \mathbf{u})$ des elementaren Hyperquaders H , die zugleich Eckpunkte von H_0 sind, die also $x_h = s_{r+1}^{x_h}$, $x_i = s_{j_i}^{x_i}$ für $i \neq h$ und $u_p = s_{q_p}^{u_p}$ erfüllen. Ihre Funktionswerte $\mathbf{f}(\mathbf{x}, \mathbf{u})$ werden mit $\mathbf{b}(\mathbf{j}, \mathbf{q})$ bezeichnet.

Damit kann man die allgemeine Transitionsfunktion nach Gleichung (B.1) aus Anhang B

$$\mathbf{f}(\mathbf{x}, \mathbf{u}) = \sum_{\mathbf{j}, \mathbf{q}} \mathbf{f}(\mathbf{s}_{\mathbf{j}}^{\mathbf{x}}, \mathbf{s}_{\mathbf{q}}^{\mathbf{u}}) \prod_i \mu_{j_i}^{x_i}(x_i) \prod_p \mu_{q_p}^{u_p}(u_p) \quad (7.13)$$

nach expliziter Ausführung der Summation über den Index j_h und der Verwendung der Bezeichnungen $\mathbf{a}(\mathbf{j}, \mathbf{q})$ und $\mathbf{b}(\mathbf{j}, \mathbf{q})$ wie folgt schreiben:

$$\mathbf{f}(\mathbf{x}, \mathbf{u}) = \sum_{j_i \neq j_h, \mathbf{q}} (\mathbf{a}(\mathbf{j}, \mathbf{q}) \mu_r^{x_h}(x_h) + \mathbf{b}(\mathbf{j}, \mathbf{q}) \mu_{r+1}^{x_h}(x_h)) \prod_{i \neq h} \mu_{j_i}^{x_i}(x_i) \prod_p \mu_{q_p}^{u_p}(u_p). \quad (7.14)$$

Jetzt betrachte man die restlichen Eckpunkte $(\mathbf{x}, \mathbf{u})$ des Hyperquaders H_0 , die für $p > s_r^{x_h}$ keine Eckpunkte von H sind, die also $x_h = p$, $x_i = s_{j_i}^{x_i}$ für $i \neq h$ und $u_p = s_{q_p}^{u_p}$ erfüllen. Ihre Funktionswerte $\mathbf{f}(\mathbf{x}, \mathbf{u})$ werden mit $\mathbf{c}(\mathbf{j}, \mathbf{q})$ bezeichnet. Für einen speziellen Vektor $(\hat{\mathbf{x}}, \hat{\mathbf{u}})$ mit $x_h = p$, $x_i = s_{j_i}^{x_i}$ für $i \neq h$ und $u_p = s_{q_p}^{u_p}$ ergibt sich der Funktionswert $\mathbf{c}(\hat{\mathbf{j}}, \hat{\mathbf{q}}) = \mathbf{f}(\hat{\mathbf{x}}, \hat{\mathbf{u}})$. Nach Gleichung (7.14) gilt dann

$$\mathbf{c}(\hat{\mathbf{j}}, \hat{\mathbf{q}}) = \sum_{j_i \neq j_h, \mathbf{q}} (\mathbf{a}(\mathbf{j}, \mathbf{q}) \mu_r^{x_h}(p) + \mathbf{b}(\mathbf{j}, \mathbf{q}) \mu_{r+1}^{x_h}(p)) \prod_{i \neq h} \mu_{j_i}^{x_i}(s_{j_i}^{x_i}) \prod_p \mu_{q_p}^{u_p}(s_{q_p}^{u_p}) \quad (7.15)$$

$$= \mathbf{a}(\hat{\mathbf{j}}, \hat{\mathbf{q}}) \mu_r^{x_h}(p) + \mathbf{b}(\hat{\mathbf{j}}, \hat{\mathbf{q}}) \mu_{r+1}^{x_h}(p). \quad (7.16)$$

Löst man die Gleichungen $\mathbf{c}(\mathbf{j}, \mathbf{q}) = \mathbf{a}(\mathbf{j}, \mathbf{q}) \mu_r^{x_h}(p) + \mathbf{b}(\mathbf{j}, \mathbf{q}) \mu_{r+1}^{x_h}(p)$ nach $\mathbf{a}(\mathbf{j}, \mathbf{q})$ auf und setzt das Ergebnis in Gleichung (7.14) ein, so erhält man bei Verwendung der Beziehung $\mu_{r+1}^{x_h}(x_h) = 1 - \mu_r^{x_h}(x_h)$ für $x_h \in [s_r^{x_h}, s_{r+1}^{x_h}]$ nach ein paar elementaren Umformungen

$$\mathbf{f}(\mathbf{x}, \mathbf{u}) = \sum_{j_i \neq j_h, \mathbf{q}} (\mathbf{c}(\mathbf{j}, \mathbf{q}) \frac{\mu_r^{x_h}(x_h)}{\mu_r^{x_h}(p)} + \mathbf{b}(\mathbf{j}, \mathbf{q}) (1 - \frac{\mu_r^{x_h}(x_h)}{\mu_r^{x_h}(p)})) \prod_{i \neq h} \mu_{j_i}^{x_i}(x_i) \prod_p \mu_{q_p}^{u_p}(u_p) \quad (7.17)$$

für alle Vektoren $(\mathbf{x}, \mathbf{u})$ aus dem Hyperquader H_0 . Da für $x_h \in [p, s_{r+1}^{x_h}]$ zusätzlich noch $\frac{\mu_r^{x_h}(x_h)}{\mu_r^{x_h}(p)} \in [0, 1]$ ist, sind alle Gewichtungsfaktoren für die Funktionswerte $\mathbf{b}(\mathbf{j}, \mathbf{q})$ und $\mathbf{c}(\mathbf{j}, \mathbf{q})$ der Eckpunkte von H_0 in Gleichung (7.17) positiv. Da sie sich außerdem alle zu eins ergänzen, ist $\mathbf{f}(\mathbf{x}, \mathbf{u})$ in diesem Bereich eine Konvexkombination der Eckpunkte $\mathbf{c}(\mathbf{j}, \mathbf{q})$ und $\mathbf{b}(\mathbf{j}, \mathbf{q})$ des Hyperquaders H_0 .

Jetzt kann man den Hyperquader H_0 nacheinander in jeder Komponente beschneiden und erhält analog, dass sich $\mathbf{f}(\mathbf{x}, \mathbf{u})$ in diesen erneut beschnittenen Bereichen als eine Konvexkombination der Eckpunkte des sich neu ergebenden achsenparallelen Hyperquaders ausdrücken lässt. $\square$

Zur Illustration des Satzes 23 wird ein zweidimensionales rekurrentes Fuzzy-System mit dem in Abbildung 7.4(a) gezeigten Zustandsgraphen betrachtet. Dem Zustandsgraphen ist der Zustandsraum unterlegt. Der Anschaulichkeit halber wurde die Eingabemenge U_0 einelementig gewählt. In den Teilbildern 7.4(b), (c) und (d) sind die Erreichbarkeitsmengen der in Abbildung 7.4(a) eingezeichneten Startmengen X_1 , X_2 und X_3 zu sehen. Die Erreichbarkeitsmengen $E_1(X_1, U_0)$ und $E_1(X_3, U_0)$ sind konvex. Daher sind sie identisch mit ihrer konvexen Hülle, die gleichzeitig die konvexe Hülle der Eckpunkte von $X_1 \times U_0$ bzw. $X_3 \times U_0$ ist.

Die in Abbildung 7.4(c) gezeigte Erreichbarkeitsmenge $E_1(X_2, U_0)$ ist dagegen nicht konvex. Es gibt Punkte, deren Verbindungsline nicht in der Menge liegt. Insbesondere liegt der Kernpositionsvektor $(s_2^{x_1}, s_2^{x_2})$ nicht in der Erreichbarkeitsmenge $E_1(X_2, U_0)$, obwohl er auf der Verbindungsline zwischen den Kernpositionsvektoren $(s_3^{x_1}, s_1^{x_2})$ und $(s_1^{x_1}, s_3^{x_2})$

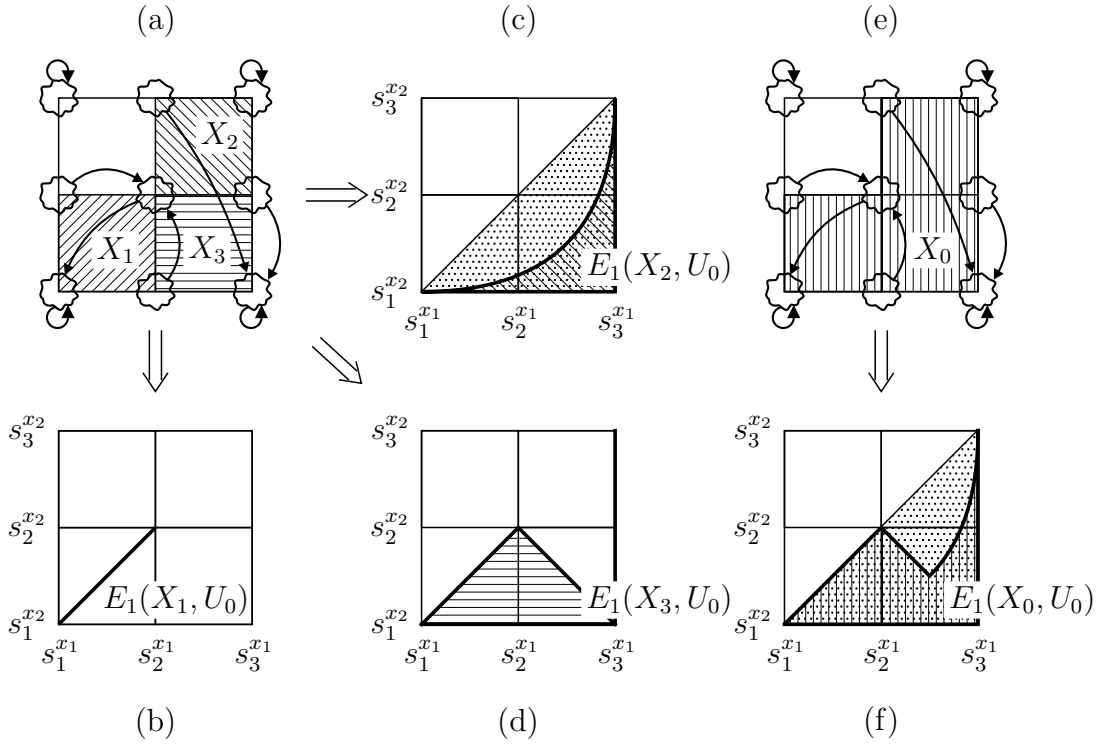


Abbildung 7.4: Beispiel zur Abschätzung von Erreichbarkeitsmengen in einem mehrdimensionalen rekurrenten Fuzzy-System. (a) Zustandsgraph und Ausgangsmengen X_1, X_2 und X_3 im unterlegten Zustandsraum X . (b) Konvexe Erreichbarkeitsmenge $E_1(X_1, U_0)$ ohne Inneres. (c) Nichtkonvexe Erreichbarkeitsmenge $E_1(X_2, U_0)$ und ihre dreiecksförmige konvexe Hülle. (d) Konvexe Erreichbarkeitsmenge $E_1(X_3, U_0)$. (e) Ausgangsmenge $X_0 = X_1 \cup X_2 \cup X_3$, die sich über mehrere elementare Hyperquader erstreckt. (f) Erreichbarkeitsmenge $E_1(X_0, U_0)$ und ihre Abschätzung durch konvexe Hüllen.

aus $E_1(X_2, U_0)$ liegt. Man kann also nicht wie im eindimensionalen Fall in Satz 22 davon ausgehen, dass alle "Zwischenwerte" in mehrdimensionalen rekurrenten Fuzzy-Systemen in der Erreichbarkeitsmenge liegen. Allerdings kann man mit Hilfe des Satzes 23 die Erreichbarkeitsmenge nach oben hin durch die konvexe Hülle einiger weniger Punkte, nämlich der Bilder der Eckpunkte des achsenparallelen Hyperquaders H_0 abschätzen. Im Falle von $H_0 = X_2 \times U_0$ ist die konvexe Hülle der Bilder $\{(s_1^{x_1}, s_1^{x_2}), (s_1^{x_1}, s_3^{x_2}), (s_3^{x_1}, s_1^{x_2})\}$ durch einen gepunkteten Bereich in Abbildung 7.4(c) gekennzeichnet. Nach Lemma 5 ist diese Abschätzung der Erreichbarkeitsmenge durch die konvexe Hülle auch in weiteren Schritten für die Abschätzung von $E_n(X_2, U_0)$ sinnvoll einsetzbar.

Will man die Erreichbarkeitsmenge einer Menge $X_0 \times U_0$ aus $X \times U$, die sich über mehrere Hyperquader erstreckt, berechnen oder abschätzen, so kann man diese Menge durch mehrere achsenparallele Hyperquader, die jeweils in einem elementaren Hyperquader liegen, darstellen oder approximieren. Die Erreichbarkeitsmenge der Menge $X_0 \times U_0$ bzw. deren Abschätzung ergibt sich dann aus der Vereinigung der Erreichbarkeitsmengen der einzelnen Hyperquader bzw. deren Abschätzungen. Dies ist beispielhaft für die Menge $X_0 \times U_0 = (X_1 \cup X_2 \cup X_3) \times U_0$ aus Abbildung 7.4(e) in Abbildung 7.4(f) durchgeführt

worden. Wieder ist die Vereinigung der konvexen Hüllen der einzelnen Hyperquader gepunktet eingezeichnet.

In diesem Beispiel ist sowohl aus der Abschätzung als auch der expliziten Berechnung von $E_1(X_0, U_0)$ zu sehen, dass $E_1(X_0, U_0) \subset X_0$ ist. Nach Lemma 5 folgt auch, dass $E_k(X_0, U_0) \subseteq X_0$ ist und dass somit ausgehend von der Menge X_0 bei Verwendung von Eingaben aus der Menge U_0 keine Zustände außerhalb von X_0 erreichbar sind.

Will man die Erreichbarkeitsmenge $E_1(X_0, U_0)$ explizit bestimmen, so ist im Allgemeinen eine kompliziertere Berechnung erforderlich. Allerdings kann man auch die Startmenge und Eingabemenge durch geeignete achsenparallele Hyperquader, die sich nicht über mehrere elementare Hyperquader erstrecken sollen, darstellen oder zumindest approximieren. Dann berechnet man die Erreichbarkeitsmengen dieser achsenparallelen Hyperquader, wobei für die Berechnung des Randes dieser Erreichbarkeitsmengen aufgrund der Monotoniebedingung aus Lemma 8 keine Funktionswerte im Innern dieser Hyperquader bestimmt werden müssen. Die ursprüngliche Erreichbarkeitsmenge $E_1(X_0, U_0)$ bzw. deren Approximation ergibt sich als Vereinigungsmenge der errechneten Erreichbarkeitsmengen. Dies vereinfacht die exakte Bestimmung von Erreichbarkeitsmengen ein wenig.

Wie auch im eindimensionalen Fall liegen die Bilder der Eckpunkte eines elementaren Hyperquaders H_0 aus $X \times U$ wieder auf Kernpositionsvektoren, da ein rekurrentes Fuzzy-System auf den Kernpositionsvektoren äquivalent zum entsprechenden linguistischen Automaten ist. Deshalb kann die Repräsentation des linguistischen Automaten durch seinen Zustandsgraph auch wieder zur Illustration der Erreichbarkeitsmengen, bzw. deren Abschätzung herangezogen werden.

Dazu wird wieder das Beispielsystem aus Abbildung 7.4(a) betrachtet. Die Menge $H_0 = X_0 \times U_0$ wird zuerst durch geeignete Hyperquader überdeckt. Im betrachteten Beispiel bieten sich bei der Startmenge $X_0 = X_1 \cup X_2 \cup X_3$ und einelementigen Eingabemenge U_0 die Wahl der elementaren Hyperquader $H_1 = X_1 \times U_0$, $H_2 = X_2 \times U_0$ und $H_3 = X_3 \times U_0$ an. Die elementaren Hyperquader im Zustandsraum X werden wie in Abbildung 7.5(a) durch die Schwärzung ihrer linguistischen Vektoren im Zustandsgraphen markiert. In einem Zwischenschritt werden die linguistischen Bildvektoren der Eckpunkte der einzelnen elementaren Hyperquader markiert. Abbildung 7.5(b) zeigt dies durch eine Schwärzung der linguistischen Bildvektoren und durch die Nennung des Index v des jeweiligen elementaren Hyperquaders H_v . In einem letzten Schritt werden alle linguistischen Vektoren markiert, deren Kernpositionen mit Punkten der konvexen Hülle derjenigen Punkte verbunden sind, die den markierten linguistischen Vektoren aus dem Zwischenschritt (b) entsprechen. Bei äquidistanten Kernpositionen können diese konvexen Hüllen direkt in den Zustandsgraphen eingezeichnet werden. Dies ist für die Bilder der Eckpunkte des elementaren Hyperquaders H_2 in Abbildung 7.5(c) zu sehen. Dieser letzte Schritt entspricht der Markierung der "Zwischenwerte" im eindimensionalen Fall, der in Abbildung 7.3(c) betrachtet wurde.

Im betrachteten Beispiel reicht die grobe Abschätzung der Erreichbarkeitsmenge mit Hilfe des Zustandsgraphen aus, um zu erkennen, dass $E_1(X_0, U_0) \subseteq X_0$ und damit $E_k(X_0, U_0) \subseteq X_0$ für alle $k \in \mathbb{N}$ ist und somit ausgehend von X_0 mit Eingaben aus U_0 keine Zustände außerhalb von X_0 erreichbar sind.

Nachdem die Möglichkeiten der Berechnung von Erreichbarkeitsmengen ausgelotet worden sind, beschäftigt sich der nächste Abschnitt mit dem dualen Problem: der Steuerbarkeit.

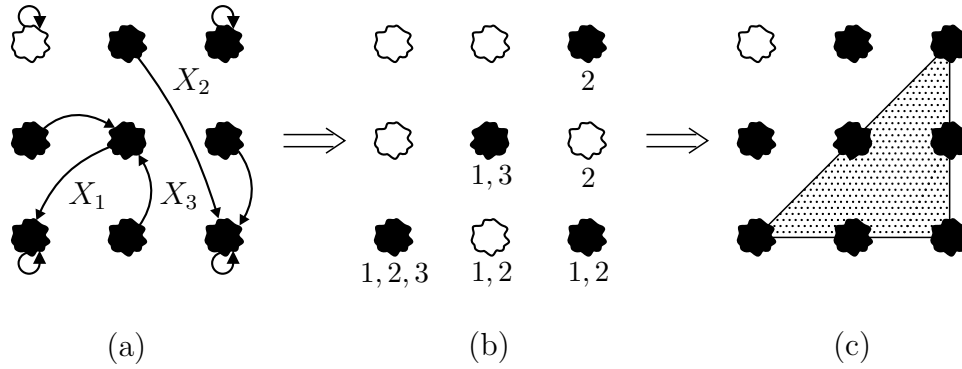


Abbildung 7.5: Abschätzung einer Erreichbarkeitsmenge $E_1(X_0, U_0)$ anhand des Zustandsgraphen eines zweidimensionalen rekurrenten Fuzzy-Systems: Markierung der (a) Startmenge $X_0 = X_1 \cup X_2 \cup X_3$ und geltende Transitionen laut U_0 , (b) Bilder der Eckpunkte der Hyperquader H_1 , H_2 und H_3 und (c) Abdeckung der Erreichbarkeitsmenge $E_1(X_0, U_0)$ und konvexe Hülle der Eckpunkte von H_2 .

7.3 Steuerbarkeit: Definitionen und Grundlagen

Die Untersuchung der Steuerbarkeit in einem dynamischen System versucht die folgenden Fragen zu beantworten: Von welchen Startzuständen aus kann man in n Schritten in einen festgelegten Endzustand $\mathbf{x}(n)$ steuern, wenn die Eingangswerte aus einer Menge U_0 gewählt werden können? Und wie sind diese Steuervektoren $\mathbf{u}(k)$ mit $k = 0, \dots, n-1$ am geeignetsten zu wählen?

Die Vorgehensweise zur Beantwortung der Fragen lehnt sich stark an die der Erreichbarkeit an. Zuerst fasst man alle die Startvektoren, die die obengenannte Eigenschaft haben, in der so genannten *Steuerbarkeitsmenge* $S_n(\mathbf{x}(n), U_0)$ zusammen. Da von einem Startvektor $\mathbf{x}(0) \in S_n(\mathbf{x}(n), U_0)$ der Endzustand auch in n Schritten erreichbar sein muss, gilt die Beziehung

$$\mathbf{x}(n) \in E_n(\mathbf{x}(0), U_0) \Leftrightarrow \mathbf{x}(0) \in S_n(\mathbf{x}(n), U_0). \quad (7.18)$$

Das eigentliche Ziel, die Berechnung von $S_n(\mathbf{x}(n), U_0)$, ist nicht so einfach. Daher wird zuerst vom Spezialfall $n = 1$ und $U_0 = \{\mathbf{u}(0)\}$ ausgegangen. In diesem Fall gilt für alle gesuchten Anfangswerte $\mathbf{x}(0) \in S_1(\mathbf{x}(1), \mathbf{u}(0))$ nach Beziehung (7.18) $\mathbf{x}(1) \in E_1(\mathbf{x}(0), \mathbf{u}(0)) = \mathbf{f}(\mathbf{x}(0), \mathbf{u}(0))$. Da $\mathbf{f}(\mathbf{x}(0), \mathbf{u}(0))$ ein einzelner Wert ist, gilt $\mathbf{x}(1) = \mathbf{f}(\mathbf{x}(0), \mathbf{u}(0))$. Der Ausdruck $\mathbf{f}(\mathbf{x}(0), \mathbf{u}(0))$ kann auch als Funktion in $\mathbf{x}$ an der Stelle $\mathbf{x} = \mathbf{x}(0)$ mit einem Scharparameter $\mathbf{u}$, der auf $\mathbf{u} = \mathbf{u}(0)$ liegt, interpretiert werden. Diese Auffassung führt zu der Darstellung $\mathbf{x}(1) = \mathbf{f}_{(\mathbf{u}(0))}(\mathbf{x}(0))$. Also liegt $\mathbf{x}(0)$ im Urbild von $\mathbf{x}(1)$, d.h. $\mathbf{x}(0) \in \mathbf{f}_{(\mathbf{u}(0))}^{-1}(\mathbf{x}(1))$. Somit ist $S_1(\mathbf{x}(1), \mathbf{u}(0)) = \mathbf{f}_{(\mathbf{u}(0))}^{-1}(\mathbf{x}(1))$.

Über die Vereinigungsmengen erhält man die Steuerbarkeitsmenge

$$S_1(\mathbf{x}(1), U_0) = \bigcup_{\mathbf{u}(0) \in U_0} S_1(\mathbf{x}(1), \mathbf{u}(0)) \quad (7.19)$$

für mehrere mögliche Steuervektoren $\mathbf{u}(0) \in U_0$ und die Steuerbarkeitsmenge

$$S_1(X_1, U_0) = \bigcup_{\mathbf{x}(1) \in X_1} S_1(\mathbf{x}(1), U_0) \quad (7.20)$$

für eine Zielmenge X_1 . Liegt ein Vektor $\mathbf{x}(0)$ in $S_1(X_1, U_0)$, so gibt es mindestens einen Eingabevektor $\mathbf{u}(0) \in U_0$, so dass $\mathbf{x}(0)$ auf einen Vektor $\mathbf{x}(1) \in X_1$ abgebildet wird.

Mit der Definition der Steuerbarkeitsmengen für Zielmengen X_{n+1} gilt analog zu Lemma 4 die folgende Iterationsgleichung zur Bestimmung von $S_{n+1}(X_{n+1}, U_0)$:

Lemma 10

Es gilt $S_{n+1}(X_{n+1}, U_0) = S_1(S_n(X_{n+1}, U_0), U_0)$.

Des Weiteren sind auch die Teilmengenbeziehungen aus Lemma 5 und Lemma 6 von Erreichbarkeitsmengen direkt auf Steuerbarkeitsmengen übertragbar. Im Gegensatz zu Erreichbarkeitsmengen sind Steuerbarkeitsmengen für kompakte bzw. zusammenhängende Ziel- und Eingangsmengen im Allgemeinen nicht kompakt bzw. zusammenhängend, da sich die Steuerbarkeitsmengen aus den Urbildern ergeben. Ein Beispiel für eine nicht-zusammenhängende Steuerbarkeitsmenge wird im nächsten Abschnitt gezeigt. Steuerbarkeitsmengen $S_n(X_n, U_0)$ können auch leer sein, nämlich dann, wenn kein Anfangszustand $\mathbf{x}(0) \in X$ durch Eingabewerte aus der gegebenen Eingangsmenge U_0 in die Zielmenge X_n abgebildet wird.

Für die Lösung eines Steuerungsproblems, d.h. für das Finden einer Steuerfolge von Eingabevektoren $\mathbf{u}(k) \in U_0$, durch die man, ausgehend von einem Startvektor $\mathbf{x}(0)$, den Endzustand $\mathbf{x}(n)$ in n Schritten erreicht, sind sowohl Steuerbarkeits- als auch Erreichbarkeitsmengen notwendig. Für das Auffinden von Steuerbarkeitsmengen sind Umkehrfunktionen $\mathbf{f}_{(\mathbf{u}(0))}^{-1}(\mathbf{x})$ auszuwerten, während für die Berechnung von Erreichbarkeitsmengen die Auswertung von Funktionen $\mathbf{f}_{(\mathbf{u}(0))}(\mathbf{x}) = \mathbf{f}(\mathbf{x}, \mathbf{u}(0))$ ausreicht. Erreichbarkeitsmengen sind also einfacher zu berechnen als Steuerbarkeitsmengen. Daher wird man beim Lösen des Steuerungsproblems möglichst wenige Steuerbarkeitsmengen berechnen wollen. Nachfolgende Überlegungen zeigen, dass die Berechnung einer Reihe von einfachen Steuerbarkeitsmengen $S_1(\mathbf{x}(k), U_0)$ zum Lösen eines Steuerungsproblems ausreichen:

Zuerst muss sichergestellt sein, dass es eine Steuerung gibt, d.h. dass $\mathbf{x}(0) \in S_n(\mathbf{x}(n), U_0)$ ist. Dazu wertet man die äquivalente Bedingung $\mathbf{x}(n) \in E_n(\mathbf{x}(0), U_0)$ aus, indem man rekursiv die Erreichbarkeitsmengen $E_k(\mathbf{x}(0), U_0)$ für $k = 1, \dots, n$ nach Lemma 4 bestimmt.

Dann sucht man Zwischenzustände $\mathbf{x}(k)$ und die Eingabevektoren $\mathbf{u}(k)$ für die Zeitschritte $k = 1, \dots, n$. Dabei geht man wie folgt vor: Ein Zwischenzustand $\mathbf{x}(k)$ muss in $E_k(\mathbf{x}(0), U_0)$ liegen, damit er von $\mathbf{x}(0)$ auch in k Schritten erreichbar ist, und zugleich in $S_{n-k}(\mathbf{x}(n), U_0)$ liegen, damit von ihm aus auch der Endzustand $\mathbf{x}(n)$ angesteuert werden kann. Anstatt mühsam die Steuermengen $S_{n-k}(\mathbf{x}(n), U_0)$ zu berechnen, testet man rekursiv die Bedingung $\mathbf{x}(k) \in E_k(\mathbf{x}(0), U_0) \cap S_1(\mathbf{x}(k+1), U_0)$, beginnend mit $k = n - 1$ bis hinab zu $k = 0$. Da die Erreichbarkeitsmengen $E_k(\mathbf{x}(0), U_0)$ sowieso schon errechnet wurden, ist nur noch die Berechnung von $S_1(\mathbf{x}(k+1), U_0)$ notwendig. In jedem Teilschritt werden nach der Wahl eines geeigneten Zwischenwertes $\mathbf{x}(k)$ zusätzlich die gesuchten Eingabevektoren $\mathbf{u}(k)$ durch Lösen der Gleichung $\mathbf{f}(\mathbf{x}(k), \mathbf{u}(k)) = \mathbf{x}(k+1)$ bestimmt. Am Ende liegen die Steuerfolge

$\mathbf{u}(k)$ für $k = 0, \dots, n - 1$ und alle Zwischenzustände $\mathbf{x}(k)$ vor, und das Steuerungsproblem ist gelöst.

Wie sich diese allgemeinen Eigenschaften von Steuerbarkeitsmengen im Zusammenhang mit rekurrenten Fuzzy-Systemen auswirken und wie man für sie geeignete Steuerfolgen findet, wird im nächsten Abschnitt behandelt.

7.4 Steuerbarkeit in rekurrenten Fuzzy-Systemen

Wie im letzten Abschnitt erwähnt wurde, muss bei der Berechnung von Steuerbarkeitsmengen die Umkehrfunktion $\mathbf{f}_{(\mathbf{u}(0))}^{-1}(\mathbf{x})$ an vielen Stellen $\mathbf{x}$ und für viele Parameter $\mathbf{u}(0)$ ausgewertet werden. Dies ist normalerweise ungleich schwerer als die Funktion $\mathbf{f}(\mathbf{x}, \mathbf{u})$ selbst an verschiedenen Stellen auszuwerten, wie es für die Berechnung von Erreichbarkeitsmengen erforderlich ist. Daher ist zu erwarten, dass man auch für Steuerbarkeitsmengen in rekurrenten Fuzzy-Systemen nicht so viel aussagen kann wie über Erreichbarkeitsmengen in rekurrenten Fuzzy-Systemen.

Um die Problematik der Berechnung von Steuerbarkeitsproblemen in rekurrenten Fuzzy-Systemen abschätzen zu können, werden zuerst zwei Beispielsysteme betrachtet. Anschließend wird aus der Erörterung der Beispiele ein Satz über Steuerbarkeitsmengen in rekurrenten Fuzzy-Systemen motiviert. Abschließend wird anhand eines Beispiels ein Steuerungsproblem gelöst, d.h. eine Steuerfolge berechnet, die in einem vorgegebenen System einen festgelegten Anfangszustand über mehrere Zeitschritte in einen festgelegten Endzustand überführt.

Das erste Beispielsystem soll verdeutlichen, welche Auswirkung die Auswertung der Umkehrfunktionen $f_{(u(0))}^{-1}(x)$ auf die Berechnung der Steuerbarkeitsmenge hat. Abbildung 7.6 zeigt beispielhaft die Bestimmung von Steuerbarkeitsmengen in einem eindimensionalen rekurrenten Fuzzy-System, in dem nur ein einziger Eingangswert $u(0) = s_q^u$ zur Verfügung steht. In einem solchen Fall können Steuerbarkeitsmengen $S_1(X_1, u(0))$ graphisch einfach bestimmt werden: Man zeichnet die Menge X_1 auf der y -Achse ein und bestimmt mit Hilfe der eingezeichneten Transitionsfunktion die Punkte auf der x -Achse, welche auf Punkte in der Menge X_1 abgebildet werden.

Das Beispiel in Abbildung 7.6(a) zeigt, dass eine so bestimmte Steuerbarkeitsmenge nicht unbedingt zusammenhängend sein muss, auch wenn die Menge X_1 selbst zusammenhängend ist. Will man sukzessive die Steuermengen $S_n(X_1, u(0))$ mit Hilfe der Iterationsgleichung $S_{n+1}(X_1, u(0)) = S_1(S_n(X_1, u(0)), u(0))$ nach Lemma 10 bestimmen, so muss man damit rechnen, dass die Zahl der Teilintervalle, die die Mengen $S_{n+1}(X_1, u(0))$ beschreiben, sehr schnell anwachsen kann.

Im Gegensatz zu Erreichbarkeitsmengen lassen sich Steuerbarkeitsmengen in eindimensionalen rekurrenten Fuzzy-Systemen nicht allgemein anhand der Regelbasis bestimmen. Dies zeigt das Beispiel in Abbildung 7.6(b). Für die Zielmenge $X_2 = [s_2^x, s_3^x]$, die durch die Kernpositionen s_2^x und s_3^x begrenzt ist, und eine Eingabemenge, die nur aus der Kernposition s_3^u besteht, erhält man die Steuerbarkeitsmenge $S_1(X_2, s_3^u) = [5, 6.5]$, die nicht mehr durch Kernpositionen begrenzt ist. Während sich Zielmenge und Eingabemenge in diesem Beispiel durch die entsprechenden linguistischen Werte charakterisieren lassen, ist dies für die

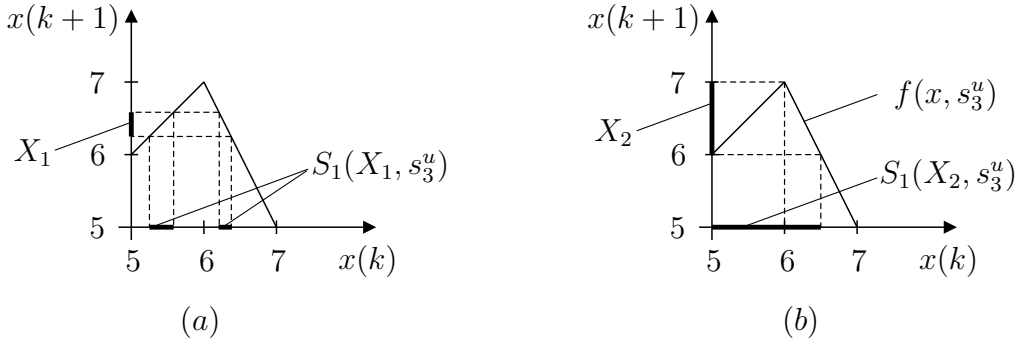


Abbildung 7.6: (a) Steuerbarkeitsmengen von zusammenhängenden Mengen sind auch in rekurrenten Fuzzy-Systemen nicht unbedingt zusammenhängend. (b) Steuerbarkeitsmengen von Zielmengen, die durch Kernpositionen begrenzt sind, sind nicht unbedingt selbst durch Kernpositionen begrenzt.

Steuerbarkeitsmenge $S_1(X_2, s_3^u)$ nicht der Fall. Die Regelbasis kann also im Allgemeinen nicht zur eindeutigen Charakterisierung der Steuerbarkeitsmengen herangezogen werden.

Bei der Lösung eines Steuerungsproblems reicht es aber in vielen Fällen aus, die Steuerbarkeitsmengen durch Teilmengen abzuschätzen. Findet sich nämlich der Ausgangspunkt, der in die Zielmenge abgebildet werden soll, in einer solchen Teilmenge, so ist das Steuerungsproblem auch gelöst.

Satz 23, der eine Abschätzung für Erreichbarkeitsmengen liefert, kann reinterpretiert werden und zur Abschätzung von Steuerbarkeitsmengen herangezogen werden. Der Satz besagt, dass von einem achsenparallelen Hyperquader $H_0 = X_0 \times U_0$ aus $X \times U$, der vollständig in einem elementaren Hyperquader liegt, die Erreichbarkeitsmenge $E_1(X_0, U_0)$ vollständig in der konvexen Hülle der Bildpunkte der Eckpunkte von H_0 liegt. Enthält umgekehrt die Zielmenge X_1 die konvexe Hülle der Bildpunkte eines solchen Quaders, dann liegen alle Punkte dieses Quaders in der Steuerbarkeitsmenge von X_1 . Damit wurde der folgende Satz bewiesen:

Satz 24 (Abschätzung von Steuerbarkeitsmengen)

Gegeben ist ein rekurrentes Fuzzy-System mit Zustandsraum X und Eingaberaum U . Wenn X_1 die konvexe Hülle der Bildpunkte eines achsenparallelen Hyperquaders $H_0 = X_0 \times U_0$ aus $X \times U$ enthält, der vollständig in einem elementaren Hyperquader liegt, dann liegen alle Punkte von H_0 in der Steuerbarkeitsmenge $S_1(X_1, U_0)$, d.h. $H_0 \subseteq S_1(X_1, U_0)$.

Satz 24 soll nicht darüber hinwegtäuschen, dass Erreichbarkeitsmengen leichter als Steuerbarkeitsmengen zu finden sind. Das Problem potenziert sich mit der Berechnung von Steuerbarkeitsmengen S_n über mehrere Zeitschritte hinweg. Im letzten Abschnitt wurde daher eine Vorgehensweise aufgezeigt, durch die sich ein Steuerungsproblem lösen lässt und bei der nur Steuerbarkeitsmengen S_1 bestimmt werden müssen. Wie sich ein solches Vorgehen bei rekurrenten Fuzzy-Systemen bewältigen lässt, wird in dem folgenden, abschließenden Beispiel demonstriert.

Als Beispielsystem dient wieder das Modell der Insektenpopulationsgröße mit der Regelbasis aus Abbildung 7.1(a). Ziel ist es, das System von einem Startwert von $x(0) = 5.1$

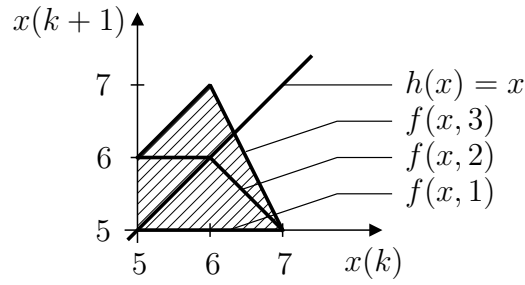


Abbildung 7.7: Eine Schar von Transitionsfunktionen $f(x, u)$, die für einen Scharparameter $u \in [1, 3]$ gelten und Fixpunktgerade $h(x) = x$.

in möglichst wenigen Schritten in den Fixpunkt $x_{fix} = 6.2$ zu steuern, wobei als Eingabemenge $U_0 = [1, 3]$ zur Verfügung steht.

Bevor eine Steuerfolge gesucht wird, wird getestet, ob der Endwert $x_{fix} = 6.2$ auch tatsächlich ein Fixpunkt, also eine Ruhelage des rekurrenten Fuzzy-Systems für eine geeignete Eingabe $u \in U_0$, ist. Diese Frage kann man anhand der graphischen Darstellung der Schar von Transitionsfunktionen $f(x, u)$ mit Scharparameter $u \in [1, 3]$ feststellen. In Abbildung 7.7 sind für die Eingabewerte $u \in \{1, 2, 3\}$ die jeweiligen Transitionsfunktionen eingezeichnet. Die Schar der möglichen Transitionsfunktionen bedeckt den gesamten gestrichelt eingezeichneten Bereich. Außerdem ist die Fixpunktgerade $h(x) = x$ zu sehen. Für $x \in [5, 6.3]$ wird die Fixpunktgerade von einer der Scharcurven $f(x, u)$ geschnitten. Daher sind alle Zustände in $[5, 6.3]$, also auch $x_{fix} = 6.2$, bei geeigneten Eingangswerten u Fixpunkte. Eine weitere Möglichkeit Fixpunkte zu finden ist der Test der Bedingung $x_{fix} \in E(x_{fix}, U_0)$. Dies ist auch wegen $E(6.2, [1, 3]) = [5, 6.6]$ erfüllt. Ein Eingangswert u , bei dem x_{fix} ein Fixpunkt für $f(x, u)$ ist, ergibt sich als Lösung der Gleichung $f(x_{fix}, u_{fix}) = x_{fix}$ zu $u_{fix} = 2.5$.

Für die Berechnung der gesuchten Steuerfolge wird die Vorgehensweise aus dem letzten Abschnitt benutzt: Zuerst wird getestet, ob sich der Zielwert $x_{fix} = 6.2$ auch von dem Startwert $x(0)$ aus erreichen lässt. Dazu werden die Erreichbarkeitsmengen $E_k(x(0), U_0)$ mit Hilfe von Satz 22 bestimmt. Man erhält: $E_0(x(0), U_0) = x(0) = 5.1$, $E_1(x(0), U_0) = [5, 6.1]$, $E_2(x(0), U_0) = [5, 7]$. Da $x_{fix} = 6.2 \in E_2(x(0), U_0)$ ist, kann der Endwert in zwei Schritten erreicht werden.

Jetzt kann man rekursiv die Zwischenzustände berechnen, die bei der Steuerung von $x(0)$ auf $x(2) = x_{fix}$ angenommen werden sollen. Dabei muss der Zwischenzustand $x(1)$ die Bedingung $x(1) \in E_1(x(0), U_0) \cap S_1(x(2), U_0)$ erfüllen. Für den Test der Bedingung muss $S_1(x(2), U_0)$ bestimmt werden. Dazu geht man in Abbildung 7.7 von dem Funktionswert $x(k+1) = 6.2$ aus und zieht gedanklich eine waagrechte Linie durch die Abbildung, so liegen genau die x -Werte in $S_1(x(2), U_0)$, bei denen die waagrechte Linie die gestrichelte Fläche schneidet, welche die Funktionsschar $f(x, u)$ markiert. Man erkennt, dass $S_1(x(2), U_0)$ ein Intervall ist und sich seine Intervallgrenzen durch das Lösen der Gleichung $f(x, 3) = 6.2$ bestimmen lassen. So erhält man $S_1(x(2), U_0) = [5.2, 6.4]$. Der Zwischenwert $x(1)$ muss also die Bedingung $x(1) \in E_1(x(0), U_0) \cap S_1(x(2), U_0) = [5, 6.1] \cap [5.2, 6.4] = [5.2, 6.1]$ erfüllen. Man wähle z.B. $x(1) = 6.1$. Um $x(1)$ auf $x(2)$ zu steuern, sucht man den Eingabewert $u(1)$

durch Lösung der Gleichung $f(x(1), u(1)) = x(2)$, also $f(6.1, u(1)) = 6.2$ und erhält so $u(1) = 2.\bar{3}$.

Der nächste "Zwischenzustand" $x(0)$ muss die Bedingung $x(0) \in E_0(x(0), U_0) \cap S_1(x(1), U_0)$ erfüllen. Man kann analog zu oben $S_1(x(1), U_0) = [5.1, 6.45]$ bestimmen und damit die Bedingung $x(0) \in [5.1, 5.1] \cap [5.1, 6.45] = [5.1, 5.1]$ auswerten. Die Bestimmung von $S_1(x(1), U_0)$ ist allerdings nicht notwendig, da $x(0)$ ja sowieso schon feststeht und aufgrund der Bedingung $x(1) \in E_1(x(0), U_0)$ sichergestellt ist, dass auch $x(0) \in S_1(x(1), U_0)$ ist. Auf jeden Fall ist $x(0) = 5.1$. Für den gesuchten Eingabewert $u(0)$, löst man die Gleichung $f(x(0), u(0)) = x(1)$ und erhält $u(0) = 2$.

Damit ist die Steuerfolge $2, 2.\bar{3}, 2.5, 2.5, 2.5, \dots$ bestimmt, die $x(0) = 5.1$ über $x(1) = 6.1$ in zwei Schritten auf den Endwert $x(2) = x_{fix} = 6.2$ steuert und dort für die weiteren Zeitschritte belässt.

8 Anwendung: Sequentielle Mustererkennung

In diesem Kapitel wird gezeigt, dass sich mit rekurrenten Fuzzy-Systemen allgemeine sequentielle Mustererkenner aufbauen lassen. Nach ein paar einleitenden Worten zu Mustererkennung werden im nächsten Abschnitt einige Grundlagen sequentieller Mustererkenner besprochen, um anschließend einen ersten allgemeinen Ansatz für einen sequentiellen Fuzzy-Mustererkenner entwickeln zu können. Dieser erste Ansatz wird anhand eines industriellen Anwendungsbeispiels illustriert. Ein solches System wird in einem Überwachungssystem für den Stranggießprozess in mehreren Stahlwerken weltweit eingesetzt [1, 2, 42]. Anschließend wird dieser erste allgemeine Ansatz untersucht und das rekurrente Fuzzy-System so erweitert, dass es sich annähernd durch einen endlichen Automaten beschreiben lässt.

Mustererkenner testen, ob sich die Merkmale eines zu untersuchenden Objektes und die Merkmale von speziellen Prototypen ähneln, also ob das Objekt das gleiche Muster wie eines der Prototypen aufweist. Neben konventionellen Mustererkennungsmethoden [22] sind auch Fuzzy-Ansätze erfolgreich [12, 13, 73, 72]. Die Probleme, die mittels einer Mustererkennung gelöst werden sollen, sind dabei meist statisch: Es werden Zugehörigkeitsfunktionen von verschiedenen Prototypen im Merkmalsraum aufgestellt und ausgewertet. Die Zugehörigkeitswerte zu den Prototypen, die sich für die zu untersuchenden Objekte ergeben, stellen dann ein Ähnlichkeitsmaß von den Objekten und Prototypen dar.

Es gibt allerdings auch Mustererkennungsaufgaben aus dem Bereich der syntaktischen Mustererkennung, die nicht durch das Auswerten einer einfachen statischen Funktion gelöst werden können. Auch hier gibt es konventionelle [28, 22] und Fuzzy-Methoden [74, 35, 27] als Lösungsansätze. Eine Gattung der syntaktischen Mustererkennung, die sequentielle Mustererkennung [67, 28], wird hier behandelt. Sie ähnelt string-matching Methoden [22], welche untersuchen, ob eine bestimmte Serie von Merkmalen in einem Objekt, z.B. einer Zeitreihe, auftauchen. Will man solche Probleme mit Fuzzy-Ansätzen lösen, so muss man auf Fuzzy-Systeme mit Dynamik wie rekurrente Fuzzy-Systeme [1, 2, 4, 42] zurückgreifen.

8.1 Grundlagen sequentieller Mustererkenner

Sequentielle Mustererkenner suchen nach einer bestimmten Abfolge von Teilmustern in einer Menge von Daten, z.B. in einem Kurvenverlauf oder einer Symbolkette [67, 28]. Oft entsteht der Kurvenverlauf oder die Symbolkette online und sequentiell, d.h. zeitlich

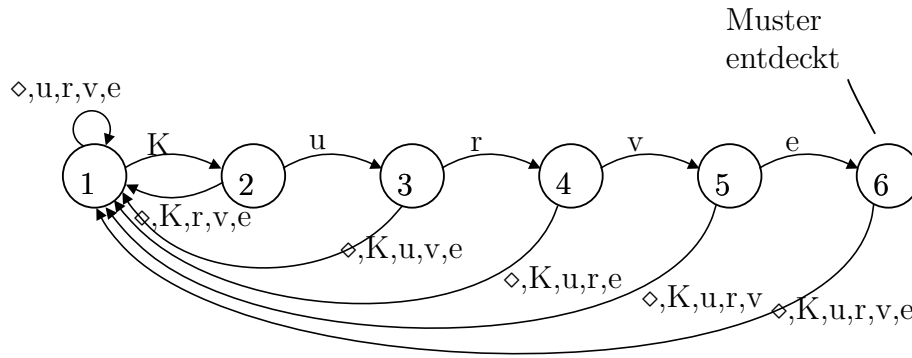


Abbildung 8.1: Zustandsgraph eines sequentiellen Mustererkenners, der auf einem endlichen Automaten basiert. Alle Buchstaben und Satzzeichen außer "K", "u", "r", "v" und "e" werden hier unter dem Zeichen "◊" zusammengefasst.

hintereinander. Für die sequentielle Erkennung solcher Muster verwendet man häufig endliche deterministische Automaten [67, 28]. Das nachfolgende Beispiel zeigt, wie ein solcher Automat Muster in einer Zeichenkette auffindet.

Angenommen, das vorliegende Kapitel soll nach dem Wort "Kurve" durchsucht werden. Der Text stellt die Zeichenkette dar. Das Wort "Kurve" ist das gesuchte Muster, bestehend aus den Teilmustern "K", "u", "r", "v" und "e". Der sequentielle Mustererkenner bekommt als Eingangswerte nacheinander die einzelnen Buchstaben des Textes als Eingaben vorgelegt. Er vergleicht die Eingaben mit einem der Buchstaben "K", "u", "r", "v" oder "e". Seine Zustandsgröße x zeigt ihm an, mit dem wievielten Buchstaben des Wortes "Kurve" er die Buchstaben des Textes am Eingang vergleichen muss. Treffen die Buchstaben direkt nacheinander in der richtigen Reihenfolge ein, so erhöht er den Wert von x jeweils um eine Einheit von anfangs $x = 1$ bis auf $x = 6$ in 5 Schritten. Wird der Wert $x = 6$ erreicht, so wurden die 5 Buchstaben in der richtigen Reihenfolge entdeckt. Taucht während der Suche ein falscher Buchstabe auf, so wird die bis dahin gefundene Zeichenkette verworfen und x sofort auf $x = 1$ zurückgesetzt und eine erneute Suche begonnen.

Das Verhalten des Mustererkenners lässt sich auch einfach von dessen Zustandsgraphen ablesen, der in Abbildung 8.1 gezeigt ist. Die Zustandswerte x sind als Knoten dargestellt. Die Pfeile beschreiben eine Änderung im Zustandswert x aufgrund eines Eingabewertes, der auf den Pfeilen vermerkt ist. Dabei werden unter dem Zeichen "◊" alle Buchstaben und Satzzeichen außer "K", "u", "r", "v" und "e" zusammengefasst.

Man kann die Regeln in die folgenden drei Klassen einteilen: *Übergangsregeln*, die zu einem höheren Zustandswert führen, *Rücksetzregeln*, die die Mustererkennung bei einem falschen Buchstaben unterbrechen und den Automaten auf den Ausgangszustand $x = 1$ zurücksetzen, und die *Endregel*, die nach einem erfolgreichen Erkennen des Musters den Automaten auch in den Ausgangszustand versetzt.

Auf die gleiche Art können Automaten auch für die Suche innerhalb anderer Zeichenketten verwendet werden, wie zum Beispiel Binärfolgen [93] oder Zeichenketten, die von regulären, formalen Sprachen [28, 22] erzeugt werden.

Es können auch Muster in Kurvenverläufen erkannt werden. Dazu wird die Kurve als Serie

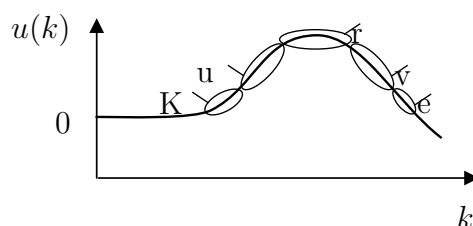


Abbildung 8.2: Beispiel für ein Muster in einer Zeitreihe $u(k)$, welches sich durch eine bestimmte Folge von Teilmustern auszeichnet, die mit Buchstaben "K," "u," "r," "v," und "e" bezeichnet sind.

von Kurvenprimitiva dargestellt. Ein typisches Beispiel ist die Auswertung von Elektrokardiogrammen (ECG) [28, 47]. Dabei werden die Kurvenprimitiva durch Zeichen gekennzeichnet, wie in Abbildung 8.2 beispielhaft gezeigt ist. Das gesuchte Muster kann dann durch eine Zeichenkette oder ein Wort dargestellt werden. Im Beispiel von Abbildung 8.2 das Wort "Kurve."

Nachdem weiteren Kurvenprimitiva, die zwar nicht in dem gesuchten Kurvenabschnitt, aber in anderen Kurvenabschnitten vorkommen können, auch Buchstaben zugeordnet wurden, kann man die zu untersuchende Kurve durch eine Buchstaben- oder Zeichenkette darstellen. Die Suche nach dem Wort "Kurve" in der so gewonnenen Zeichenkette ist dann gleichbedeutend mit der Suche nach dem Muster aus Abbildung 8.2.

Im Nachfolgenden wird beschrieben, wie man die Kurvenmerkmale darstellen kann, um ihnen geeignete Buchstaben zuordnen zu können. In obigem Beispiel entspricht die Gewinnung der Kurvenprimitiva der Merkmalsextraktion in einer klassischen Mustererkennungsaufgabe.

Ausgangspunkt für die Beschreibung der Kurvenprimitiva ist die zur Verfügung stehende Datenbasis. Die Kurven sind üblicherweise als Zeitreihe $u(k)$ gegeben. Die Kurvenprimitiva müssen sich daher durch die Zahlenwerte $u(k)$ und andere davon abgeleitete Größen, wie z.B. deren Änderung $\Delta u(k) = u(k) - u(k-1)$, definieren lassen.

Eine Möglichkeit ist, die Kurvenprimitiva durch Wertebereiche der Größen $u(k)$ und $\Delta u(k)$ zu charakterisieren. Dafür wird der Eingangsraum, der durch die Größen $u(k)$ und $\Delta u(k)$ aufgespannt wird, partitioniert, wie in Abbildung 8.3(a) beispielhaft zu sehen ist. In Abbildung 8.3(b) sind für das obige Beispiel geeignete Bereiche für die Kurvenprimitiva eingezeichnet. Das Teilmuster mit der Bezeichnung "K" kann verbal durch "einen kleinen Anstieg in u bei leicht erhöhten Werten von u ", d.h. durch die Zugehörigkeit eines Eingangsvektors $\mathbf{u}(k) = (u(k), \Delta u(k))$, zu der in Abbildung 8.3 eingezeichneten Menge beschrieben werden. Sobald die Buchstaben vom Eingangssignal durch diese Vorverarbeitung generiert worden sind, arbeitet der sequentielle Mustererkenner genau wie der oben beschriebene Automat.

In der Praxis hat man allerdings in der Regel mit abgetasteten und verrauschten Signalen zu tun. Nachfolgend wird erörtert, welche Probleme dabei entstehen und wie man diese löst.

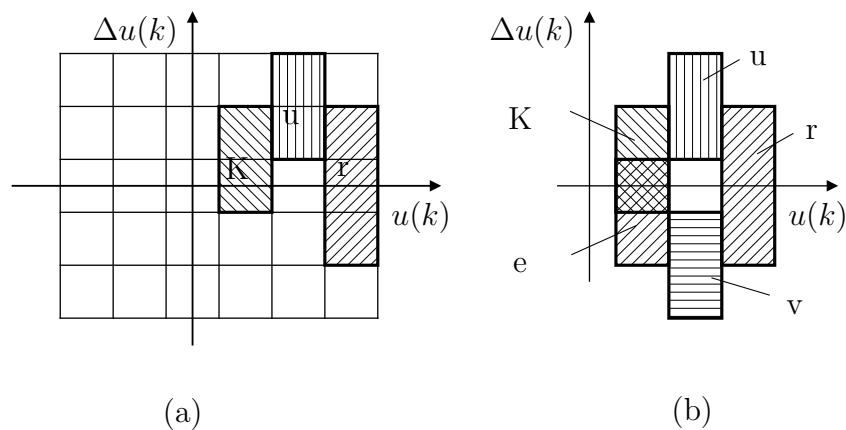


Abbildung 8.3: Der partitionierte Eingangsraum eines sequentiellen Mustererkenners. Die markierten Bereiche entsprechen Kurvenprimitiva.

Im Falle eines abgetasteten Signals gehören mehrere aufeinanderfolgende Eingabewerte normalerweise zu ein und demselben Bereich. Dann liefert die Vorverarbeitung über mehrere Zeitschritte hinweg den gleichen Buchstaben, insbesondere wenn sich das Signal zwischen zwei Abtastzeitpunkten nur geringfügig ändert. Die Wiederholung der Buchstaben führt nach obigem Ansatz normalerweise zu einem Rücksetzen des Automaten. Dieses Problem kann man durch die Einführung von *Halteregeln* umgehen, die dafür sorgen, dass der Automat bei einer Wiederholung eines Zeichens seinen Zustand beibehält.

Hat beispielsweise obiger Mustererkenner schon den Buchstaben "K" gefunden und befindet er sich damit im Zustand $x = 2$, so sucht er nach dem nächsten gesuchten Buchstaben, dem "u". Wenn er statt eines "u" wiederum ein "K" als Eingabe aus der Vorverarbeitung erhält, so bricht er die Suche aufgrund der Halteregel nicht ab, sondern verbleibt im Zustand $x = 2$ und setzt die Suche fort.

Bei verrauschten Signalen tauchen Probleme beim Übergang zwischen den Bereichen auf, die zu verschiedenen Buchstaben führen. Überquert z.B. das Signal die Grenze zwischen den Gebieten "K" und "u", so tritt das Signal aufgrund des Rauschens mit hoher Wahrscheinlichkeit kurzzeitig wieder in den Bereich "K" ein, um dann wieder Werte aus dem Bereich "u" anzunehmen. Anstatt in diesem Fall den Erkennungsprozess zu stoppen, macht es Sinn, den Wert der Zustandsvariablen von $x = 3$ auch kurzzeitig auf $x = 2$ zu verringern, um einen erneuten Anstieg auf $x = 3$ zu ermöglichen. Diese Änderung des Verhaltens wird durch die Einführung von *Rückschrittregeln* erreicht.

Diese neuen Regelarten, Halteregel und Rückschrittregel, lassen sich leicht in den Automaten integrieren. In Abbildung 8.4 ist der Zustandsgraph des so modifizierten Automaten gezeigt. Allerdings stellen diese neuen Regeln auch weitere Bedingungen an den Mustererkenner: Da in jeder denkbaren Situation eindeutig festgelegt sein soll, welche Regelart zum Zuge kommt, dürfen sich im Eingangsraum U die Bereiche, in denen die verschiedenen Regeln gelten, nicht überschneiden.

Der Ansatz eines sequentiellen Mustererkenners wurde anhand eines Beispiels eingeführt. Er lässt sich wie folgt verallgemeinern und kompakt darstellen: Der sequentielle Musterer-

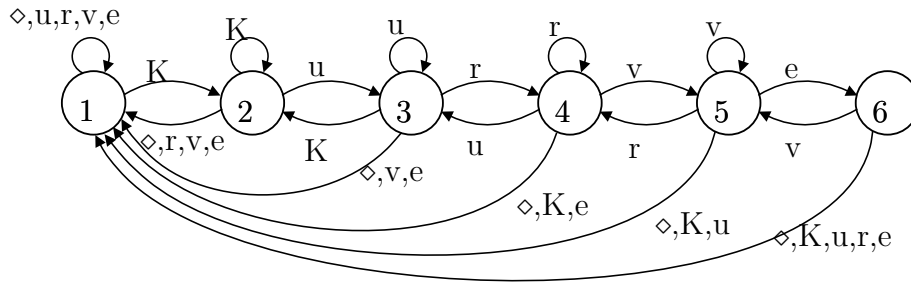


Abbildung 8.4: Zustandsgraph eines sequentiellen Mustererkenners, der auf einem endlichen Automaten basiert und Halte- und Rückschrittregeln enthält.

Tabelle 8.1: Regelbasis eines konventionellen sequentiellen Mustererkenners.

Regelart	$x(k)$	$\mathbf{u}(k)$	$x(k+1)$
Übergangsregel	$L_j^x, j < j_{max}$	$B(L_{j+1}^x)$	L_{j+1}^x
Halteregel	$L_j^x, j < j_{max}$	$B(L_j^x)$	L_j^x
Rückschrittregel	$L_j^x, j > 1$	$B(L_{j-1}^x)$	L_{j-1}^x
Rücksetzregel	L_j^x	$A(L_j^x)$	L_1^x
Endregel	$L_{j_{max}}^x$	$B(L_{j_{max}}^x)$	L_1^x

kenner ist ein endlicher Automat, der ein Muster anhand einer Serie von $j_{max} - 1$ Teilmustern erkennt. Zu diesem Zweck benötigt er eine Zustandsvariable x , die die Werte L_1^x bis $L_{j_{max}}^x$ annehmen kann, z.B. einfach die Werte 1 bis j_{max} . Für jeden Wert $j \in \{2, 3, \dots, j_{max}\}$ wird das $(j - 1)$ -te der insgesamt $j_{max} - 1$ Teilmuster durch Mengen $B(L_j^x)$ im Eingangsraum U beschrieben. Außerdem wird mit $B(L_1^x) = U \setminus B(L_2^x)$ die Menge der Eingabevektoren bezeichnet, die im Initialisierungszustand $x = L_1^x$ anzeigen, dass das erste Teilmuster der Merkmalsreihe noch nicht gefunden wurde. Schließlich werden durch $A(L_j^x) = U \setminus (B(L_{j-1}^x) \cup B(L_j^x) \cup B(L_{j+1}^x))$ die Mengen von Eingabevektoren zusammengefasst, die während der Suche nach dem $(j - 1)$ -ten Teilmuster die Rücksetzung in den Initialisierungszustand auslösen. Durch diese Notationen lassen sich die einzelnen Regeln kompakt zusammenfassen, wie in Tabelle 8.1 zu sehen ist. In jeder Situation gilt exakt eine Regel und somit wird der Folgezustand des Automaten immer eindeutig bestimmt. Damit sind die allgemeine Struktur und die Regeln eines sequentiellen Mustererkenners festgelegt.

Bei dem Entwurf eines speziellen sequentiellen Mustererkenners bleibt allerdings noch eine große Aufgabe zu bewältigen: die Extraktion der Teilmuster aus dem Eingangssignal $u(t)$, d.h. die Gewinnung der Zeichen, die diese Teilmuster beschreiben. In obigem Beispiel wurde das Teilmuster "K" durch eine Menge im Merkmalsraum charakterisiert und verbal durch "eine kleine Erhöhung im Wert u bei leicht erhöhten u -Werten" beschrieben. Das nächste Teilmuster, "u", lässt sich durch "einen leichten bis hin zu einem großen Anstieg bei erhöhten u -Werten" beschreiben. Eine ähnliche Beschreibung ist auch für die anderen Teilmuster möglich.

Statt mit klassischen Mengen lassen sich diese Beschreibungen auch auf natürliche Weise durch Fuzzy-Mengen modellieren. In diesem Fall liefert die Fuzzifizierung auf systematischem Weg die Merkmalsextraktion. Das Ergebnis der Merkmalsextraktion sind dann Zugehörigkeitswerte zu den Fuzzy-Mengen $B(L_j^x)$ im Eingangsraum U . Ein erster Weg, diese Signale zu verarbeiten, wäre, die Regeln aus Tabelle 8.1 als Fuzzy-Regeln und nicht als Schaltregeln in einem Automaten zu interpretieren und die Zustandsvariable x als Fuzzy-Variable zu behandeln. Dann kann die Zustandsvariable auch Zahlenwerte annehmen, die zwischen den Werten für L_j^x liegen. Damit kann das System nicht nur anzeigen, wie viele Teilmuster bisher entdeckt worden sind, sondern auch zu welchem Grad sich Anzeichen für das nächste Teilmuster im Signalverlauf abzeichnen. Wenn sich die Eingabewerte nur langsam im Verhältnis zur Abtastzeit verändern, so ergibt sich ein quasikontinuierlicher Verlauf der Zustandsgröße x , solange keine Rücksetzregel aktiviert wird.

Durch die Ausweitung des Zustandsraumes von einer endlichen Menge auf ein Kontinuum, d.h. $x \in \mathbb{R}$ statt $x \in \mathbb{N}$, ist der sequentielle Mustererkenner dann kein endlicher Automat mehr, sondern ein rekurrentes Fuzzy-System. Wie sich ein solches System entwerfen lässt, wird im nächsten Abschnitt behandelt.

8.2 Sequentielle Mustererkennung mit rekurrenten Fuzzy-Systemen

Der hier beschriebene sequentielle Fuzzy-Mustererkenner wurde in einer ersten Version bereits in [2, 3] eingeführt. Er ist im Prinzip ein rekurrentes Fuzzy-System, welches die Regeln eines sequentiellen Mustererkenners verwendet, die in Tabelle 8.1 im Abschnitt 8.1 zu finden sind. Um ein solches System aufzustellen, müssen alle Bestandteile des benötigten rekurrenten Fuzzy-Systems bestimmt werden.

Zuerst wählt man eine Reihe von linguistischen Werten L_j^x für die Zustandsvariable x , die als Zählvariable im sequentiellen Mustererkenner benutzt wird. Geeignete linguistische Werte sind z.B. $L_1^x = \text{nulltes}$, $L_2^x = \text{erstes}$, ..., und $L_{j_{max}}^x = \text{letztes}$ zum Durchzählen der Teilmuster, oder $L_1^x = \text{null (N)}$, $L_2^x = \text{winzig (W)}$, ..., und $L_{j_{max}}^x = \text{riesig (R)}$, falls die Zustandsgröße x als Erkennungsgrad angesehen wird.

Die linguistischen Werte, welche die Merkmale u_p beschreiben, können durch den inhaltlichen Hintergrund der Eingangsvariablen motiviert sein. Werte wie $L_1^{u_1} = \text{kühl}$, $L_2^{u_1} = \text{warm}$, $L_3^{u_1} = \text{heiß}$ und $L_4^{u_1} = \text{extrem heiß}$ eignen sich z.B. für eine Variable u_1 , welche einen Temperaturwert beschreibt. Es können aber auch generische linguistische Werte wie negativ riesig (NR), negativ groß (NG), negativ klein (NK), null (N), positiv klein (PK), positiv groß (PG) und positiv riesig (PR) verwendet werden.

Um die Regelbasis des sequentiellen Fuzzy-Mustererkenners aufzustellen, nutzt man die Regeln des konventionellen sequentiellen Mustererkenners aus Tabelle 8.1 als Grundlage. Diese Regeln haben die Teilmuster $B(L_j^x)$ in ihrer Prämisse stehen und nicht linguistische Werte $L_{q_p}^{u_p}$, wie für ein rekurrentes Fuzzy-System nach Gleichung (3.3) gefordert wird. Daher werden die Teilmuster $B(L_j^x)$ durch linguistische Werte $L_{q_p}^{u_p}$ der Eingangsvariablen u_p ausgedrückt. In einem ersten Schritt wird dazu eine verbale Beschreibung der Teilmuster herangezogen, z.B.

”Das dritte Teilmuster $B(L_4^x)$, ist charakterisiert durch einen positiven, großen ($= L_5^{u_1}$) Wert im Merkmal u_1 und einen Wert im Merkmal u_2 , der negativ klein ($= L_2^{u_2}$), nahe null ($= L_3^{u_2}$) oder positiv klein ($= L_4^{u_2}$) ist.”

Diese sprachliche Beschreibung kann dann in einen logischen Ausdruck von linguistischen Werten der linguistischen Eingangsvariablen überführt werden, z.B.

$$B(L_4^x) = L_5^{u_1} \wedge (L_2^{u_2} \vee L_3^{u_2} \vee L_4^{u_2}). \quad (8.1)$$

Diese logischen Ausdrücke werden anschließend in ihre disjunktive Normalform gebracht. Für das betrachtete Beispiel erhält man so den Ausdruck

$$B(L_4^x) = (L_5^{u_1} \wedge L_2^{u_2}) \vee (L_5^{u_1} \wedge L_3^{u_2}) \vee (L_5^{u_1} \wedge L_4^{u_2}). \quad (8.2)$$

In einem letzten Schritt werden die Ausdrücke, die sich für die Teilmuster $B(L_j^x)$ ergeben, in die Regeln des sequentiellen Mustererkenner eingesetzt und das Resultat in mehrere Fuzzy-Regeln aufgespalten. So erhält man z.B. für den Ausdruck $B(L_4^x) = (L_5^{u_1} \wedge L_2^{u_2}) \vee (L_5^{u_1} \wedge L_3^{u_2}) \vee (L_5^{u_1} \wedge L_4^{u_2})$ und die Übergangsregel ”Wenn $x(k) = L_3^x$ und $\mathbf{u}(k) = B(L_4^x)$, dann $x(k+1) = L_4^x$ ” die folgenden Fuzzy-Regeln:

$$\begin{aligned} \text{Wenn } x(k) = L_3^x \text{ und } u_1(k) = L_5^{u_1} \text{ und } u_2(k) = L_2^{u_2}, \text{ dann } x(k+1) &= L_4^x. \\ \text{Wenn } x(k) = L_3^x \text{ und } u_1(k) = L_5^{u_1} \text{ und } u_2(k) = L_3^{u_2}, \text{ dann } x(k+1) &= L_4^x. \\ \text{Wenn } x(k) = L_3^x \text{ und } u_1(k) = L_5^{u_1} \text{ und } u_2(k) = L_4^{u_2}, \text{ dann } x(k+1) &= L_4^x. \end{aligned} \quad (8.3)$$

Da die Ergebnisse der Regeln bei der Auswertung der Regelbasis durch einen Oder-Operator verknüpft werden, findet sich der Oder-Operator in der Beschreibung der Teilmuster $B(L_j^x)$ implizit in der Auflistung der Regeln in der Regelbasis wieder.

Wenn diese Schritte für jede Regel durchgeführt worden sind, so erhält man eine vollständige und widerspruchsfreie Regelbasis, wie es von einem rekurrenten Fuzzy-System gefordert wird.

Auch in diesem Fall kann die Regelbasis wieder anhand eines Zustandsgraphen des dazugehörigen linguistischen Automaten graphisch dargestellt werden. In Abbildung 8.5 ist ein Beispiel für einen sequentiellen Fuzzy-Mustererkenner mit sechs Zustandsvariablen zu sehen, der sich für das Erkennen eines Musters, bestehend aus fünf Teilmustern, eignet. Die welligen Knoten sind mit den linguistischen Werten null (N), winzig (W), klein (K), mittel (M), groß (G) und riesig (R) der Zustandsgröße markiert. An den Transitions Pfeilen stehen die Bedingungen für die Eingangswerte, damit die jeweilige Transition wirkt. Da das rekurrente Fuzzy-System und der konventionelle Mustererkenner die gleichen Regeln benutzen, ist es nicht verwunderlich, dass ihre Zustandsgraphen auch gleich sind, wie man bei einem Vergleich von Abbildung 8.5 und Abbildung 8.4 sieht.

Für die Definition der Transitionsfunktion f müssen noch die Zugehörigkeitsfunktionen und Kernpositionen für die einzelnen linguistischen Werte gewählt werden. Für die Zählvariable x wählt man geeigneterweise äquidistante Dreiecke als Zugehörigkeitsfunktionen μ_j^x , um eine versteckte Dynamik auszuschließen, wie in Abschnitt 6.1 beleuchtet wurde. Für die linguistischen Werte der Merkmalsvariablen u_p bieten sich trapezförmige Zugehörig-

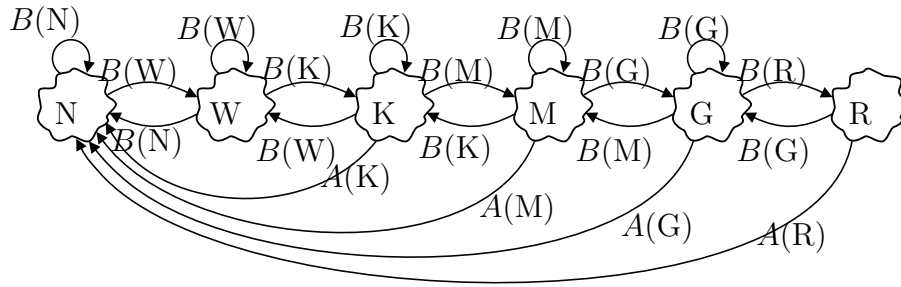


Abbildung 8.5: Zustandsgraph eines sequentiellen Mustererkenners, der auf einem rekurrenten Fuzzy-System basiert.

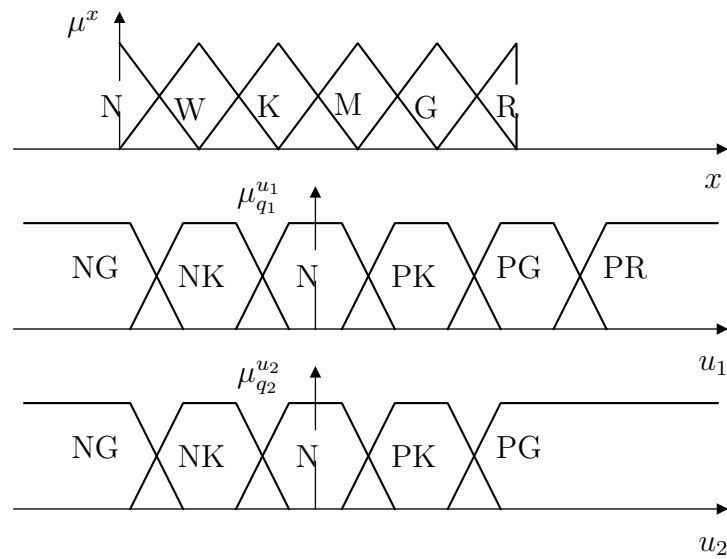


Abbildung 8.6: Gängige Zugehörigkeitsfunktionen für die Zustandsgröße x und die Eingabegrößen u_p eines rekurrenten Fuzzy-Systems.

keitsfunktionen an. Ein Beispiel ist in Abbildung 8.6 zu sehen.

Dann sind alle Teile des rekurrenten Fuzzy-Systems und damit des sequentiellen Fuzzy-Mustererkenners festgelegt. Aus Gleichung (3.13) erhält man schließlich den Ausdruck $x(k+1) = f(x(k), \mathbf{u}(k))$, der die zeitliche Entwicklung des Erkennungsgrades x beschreibt. Der prinzipielle Aufbau eines solchen sequentiellen Fuzzy-Mustererkenners wird im nächsten Abschnitt anhand eines Beispiels illustriert.

8.3 Anwendungsbeispiel aus der Stahlindustrie

In diesem Abschnitt wird ein sequentieller Fuzzy-Mustererkenner [1, 2] beschrieben, welcher seit einigen Jahren erfolgreich zur Durchbruchfrüherkennung in Stranggießanlagen von Stahlwerken in Südafrika, Indien und Deutschland eingesetzt wird (siehe auch [75]).

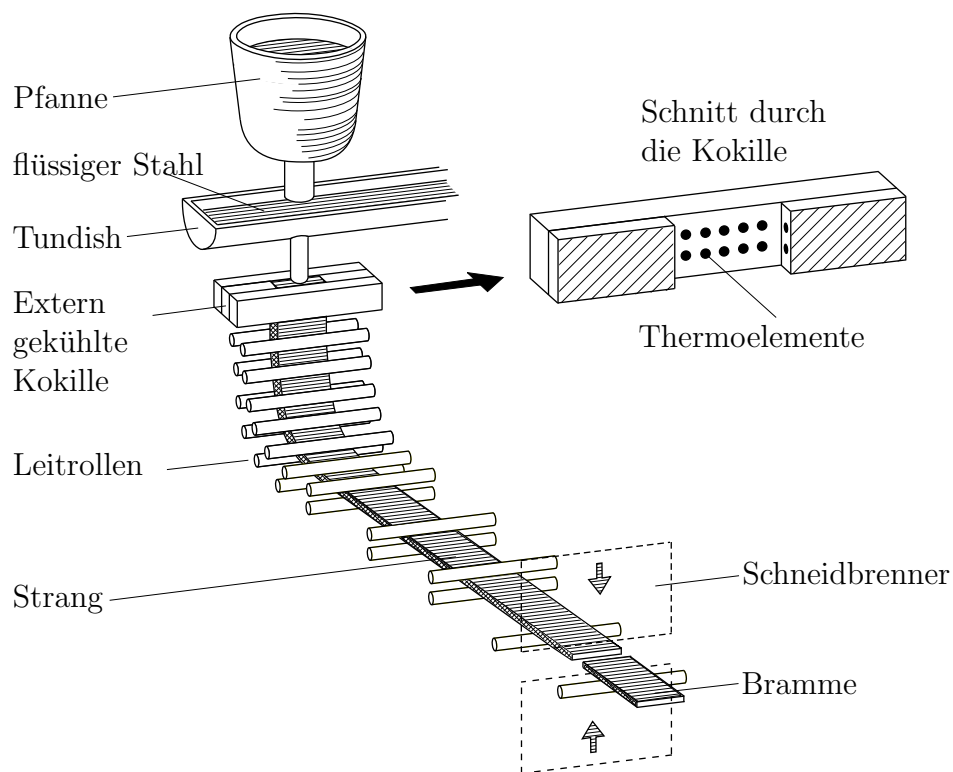


Abbildung 8.7: Schematischer Aufbau des Stranggießprozesses.

Das System ist Teil des Prozessleitsystems und wird unter der Bezeichnung "B.O.P.S." ("breakout-prediction system") vermarktet.

Im Stranggießprozess wird flüssiger Stahl in der so genannten Pfanne zu dem Tundish gebracht, der den Stahl auf mehrere Stranggießlinien verteilt. Der Stahl ergießt sich in die extern gekühlte Gießform, die so genannte Kokille. Dabei wird der Gießpiegel in der Kokille durch eine Regelung [50, 68, 69] konstant gehalten, da Variationen im Gießpiegel sich negativ auf die Stahlqualität auswirken und zu Durchbrüchen führen können. Während der flüssige Stahl durch die gekühlte Kokille wandert, beginnt er sich an den Kontaktflächen zu verfestigen. Bis zum Austritt aus der Kokille hat sich dann eine Schale um den noch flüssigen Stahlkern ausgebildet. Diese Schale verhindert, dass der flüssige Stahl im Inneren ausbricht, während sie aus der Kokille herausgezogen wird. Der noch flüssige Kern verfestigt sich im weiteren Verlauf, sodass der so entstandene Stahlstrang anschließend in Brammen zerschnitten werden kann. In Abbildung 8.7 ist der Ablauf schematisch dargestellt.

Das Ausbilden der Schale in der Kokille ist ein kritischer Punkt. So können bei fehlerhaftem Wachstum der Schale Schwachstellen entstehen. Als Folge dieser Schwachstellen bricht möglicherweise die Schale nach dem Austritt aus der Kokille auf. Bei einem solchen Durchbruch kann der flüssige Stahl austreten. Dies führt unweigerlich zu einer Beschädigung und einem Ausfall der Anlage und zu kostspieligen Reparaturen.

Wachstumsfehler können verschiedene Ursachen haben [95]. Die häufigste Ursache sind so

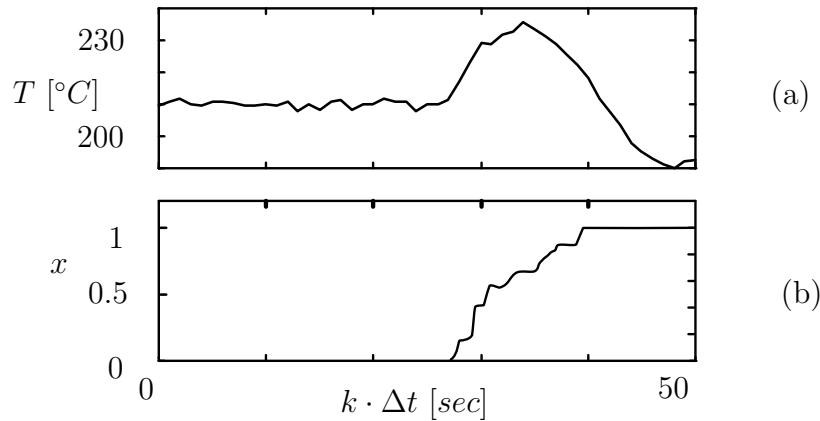


Abbildung 8.8: (a) Temperaturkurve und (b) subjektive Wahrscheinlichkeitseinschätzung eines Durchbruchs.

genannte *Kleber* [95, 63, 60]: Aufgrund eines Mangels an Gießpulver, welches die Reibung zwischen Stahlschale und Kokille verringert, haftet die Schale an dieser Stelle stärker an der Kokillenwand. Die Geschwindigkeit wird daher dort lokal verringert, was zu Spannungen und dem Aufbrechen der sich ausbildenden Stahlschale in der Kokille führt. Flüssiger Stahl, der aus dem Riss austritt, erreicht die Kokillenwand und erhöht die Temperatur der Kokille an dieser Stelle. Dem Temperaturanstieg folgt ein Abfallen der Temperatur unter den Normalwert, sobald das Material oberhalb des Risses die Stelle erreicht, da das Material aufgrund der niedrigeren Fließgeschwindigkeit eine ungewöhnlich dicke Schicht ausbilden konnte. Der Temperaturverlauf wird durch Thermoelemente in der Kokille gemessen. Ein typischer Temperaturverlauf, der durch einen Kleber verursacht wird, ist in Abbildung 8.8(a) zu sehen.

Sobald ein Kleber detektiert wird, verringert man die Geschwindigkeit, mit der der Strang aus der Kokille gezogen wird. Damit verbleibt die dünne Schale länger in der Kokille und kann sich weiter ausbilden, was einen anschließenden Durchbruch verhindert [71]. Deshalb ist ein Überwachungssystem nötig, welches die charakteristischen Merkmale eines Klebers im laufenden Betrieb aus den Temperaturkurven herausliest. Für diese Aufgabe gibt es eine Reihe von Lösungsansätzen. Einige arbeiten direkt mit der gemessenen Temperatur und deren zeitlichen Änderung [9, 95, 54], während andere Mustererkenner nutzen, die auf neuronalen Netzen [37, 90, 34] oder anderen Methoden [87] basieren.

Das hier beschriebene System [1, 2] nutzt rekurrente Fuzzy-Systeme für die sequentielle Mustererkennung der Kleber¹. Es zeichnet sich durch einen geringen Aufwand in Entwurf und Implementierung und ein sehr zuverlässiges Verhalten aus.

Für die Charakterisierung des Musters im Temperaturverlauf werden die Temperatur $T(k)$ und ihre zeitliche Änderung $\Delta T(k) = T(k) - T(k - 1)$ verwendet. Der aktuelle Temperaturwert $T(k)$ stellt den ersten Eingangswert u_1 des sequentiellen Fuzzy-Mustererkenners dar. Er wird durch die linguistischen Werte $L_1^{u_1} = \text{negativ groß (NG)}$, $L_2^{u_1} = \text{negativ klein (NK)}$, $L_3^{u_1} = \text{null (N)}$, $L_4^{u_1} = \text{positiv klein (PK)}$, $L_5^{u_1} = \text{positiv groß (PG)}$, und $L_6^{u_1} = \text{positiv sehr groß (PSG)}$ beschrieben.

¹Das kommerzielle Produkt [1, 2] findet auch *Cracks* [95], eine weitere Ursache für Durchbrüche.

tiv riesig (PR) beschrieben. Die Temperaturänderung $\Delta T(k)$ ist die zweite Eingangsgröße u_2 , die durch die linguistischen Werte $L_1^{u_2}$ = negativ groß (NG) bis hin zu $L_5^{u_2}$ = positiv groß (PG) charakterisiert wird.

Die Zustandsvariable des sequentiellen Fuzzy-Mustererkenners ist die Zählvariable x , die eine subjektive, d.h. keine mathematische, Wahrscheinlichkeitseinschätzung für einen Durchbruch angibt. Sie kann die linguistischen Werte L_1^x = null (N), L_2^x = winzig (W), L_3^x = klein (K), L_4^x = mittel (M), L_5^x = groß (G), und L_6^x = riesig (R) annehmen.

Das erste Teilmuster, das eine Änderung der Zählvariablen x von N auf W bewirken soll, kann durch "einen leichten Anstieg in der Temperatur bei leicht erhöhten Temperaturwerten" charakterisiert werden. Der Bereich $B(W)$ im Eingangsraum wird daher durch die Fuzzy-Menge $(T = PK) \wedge (\Delta T = (N \vee PK))$ festgelegt. Analog ergibt sich aus der sprachlichen Beschreibung für das zweite Teilmerkmal K die Fuzzy-Menge $B(K) = (T = PG) \wedge (\Delta T = (PK \vee PG))$.

Die verbalen Beschreibungen der anderen Teilmerkmale, die für die Erreichung der Erkennungszustände M, G und R aufgefunden werden müssen, liefern weitere Fuzzy-Mengen $B(M)$, $B(G)$ und $B(R)$ und zusätzlich $B(N)$. Zusammen erhält man:

$$\begin{aligned} B(N) &= U \setminus B(W) \\ B(W) &= (u_1 = PK) \wedge (u_2 = (N \vee PK)) \\ B(K) &= (u_1 = PG) \wedge (u_2 = (PK \vee PG)) \\ B(M) &= (u_1 = PR) \wedge (u_2 = (NK \vee N \vee PK)) \\ B(G) &= (u_1 = PG) \wedge (u_2 = (NG \vee NK)) \\ B(R) &= (u_1 = PK) \wedge (u_2 = (NK \vee N)) \end{aligned}$$

Mit der Festlegung dieser Fuzzy-Mengen für die Bereiche $B(L_j^x)$ lässt sich die gesamte Regelbasis des sequentiellen Fuzzy-Mustererkenners aufstellen. Die Regeln müssen dann noch in die Form von Gleichung (3.3) gebracht werden, um im rekurrenten Fuzzy-System verwendet werden zu können. Dazu wird die folgende Übergangsregel beispielhaft behandelt:

$$\text{Wenn } x(k) = N \text{ und } u(k) = B(W), \text{ dann } x(k+1) = W. \quad (8.4)$$

Im ersten Schritt wird die Beschreibung des Bereichs $B(W)$ in seine disjunktive Normalform $B(W) = ((u_1 = PK) \wedge (u_2 = N)) \vee ((u_1 = PK) \wedge (u_2 = PK))$ überführt. Die disjunktive Normalform wird in obige Übergangsregel eingesetzt und die resultierende Regel in die folgenden beiden Regeln aufgespalten:

$$\begin{aligned} \text{Wenn } x(k) = N \text{ und } u_1(k) = PK \text{ und } u_2(k) = N, \quad & \text{dann } x(k+1) = W. \\ \text{Wenn } x(k) = N \text{ und } u_1(k) = PK \text{ und } u_2(k) = PK, \quad & \text{dann } x(k+1) = W. \end{aligned} \quad (8.5)$$

Der Oder-Operator in der Beschreibung des Bereichs $B(W)$ wird in der Akkumulation im Fuzzy-System berücksichtigt, welches die Ergebnisse der einzelnen Regeln durch einen Fuzzy-Oder-Operator verknüpft. Wendet man obige Prozedur auf alle Mengen $B(L_j^x)$ und $A(L_j^x)$ in allen Regeln an, so erhält man eine vollständige und widerspruchsfreie Regelbasis mit Regeln in der Form von Gleichung (3.3).

Im nächsten Schritt müssen die linguistischen Werte L_j^x , $L_{q_1}^{u_1}$ und $L_{q_2}^{u_2}$ durch Zugehörig-

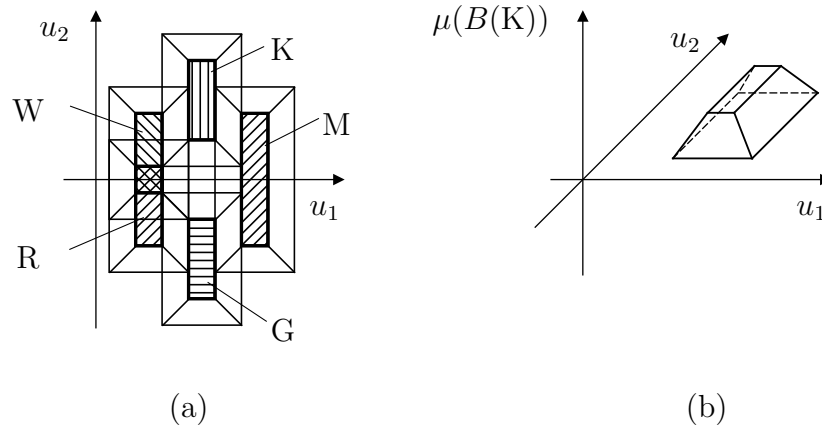


Abbildung 8.9: (a) Draufsicht auf die Zugehörigkeitsfunktionen der Fuzzy-Mengen $B(L_j^x)$, die die Teilmuster im Merkmalsraum U charakterisieren. (b) Die Zugehörigkeitsfunktion der Fuzzy-Menge $B(L_3^x) = B(K)$.

keitsfunktionen μ_j^x , $\mu_{q_1}^{u_1}$ und $\mu_{q_2}^{u_2}$ beschrieben werden. Eine typische Wahl solcher Zugehörigkeitsfunktionen findet sich in Abbildung 8.6. Nach der Wahl der Zugehörigkeitsfunktionen ist auch die Form der Fuzzy-Mengen festgelegt, welche die Bereiche $B(L_j^x)$ im Eingangsraum beschreiben. Abbildung 8.9(a) zeigt die Draufsicht auf die Fuzzy-Mengen $B(L_j^x)$, deren Kernbereiche jeweils schraffiert sind. In Abbildung 8.9(b) ist die Fuzzy-Menge $B(L_3^x) = B(K)$ dargestellt.

Damit sind alle Teile des sequentiellen Fuzzy-Mustererkenners definiert. In Abbildung 8.8(a) ist der charakteristische Temperaturverlauf eines Klebers gezeigt. Darunter ist in Abbildung 8.8(b) die zeitliche Entwicklung der Zustandsgröße x des Mustererkenners zu sehen. Der Zustandswert erreicht alle Kernpositionen s_j^x , die zu den einzelnen linguistischen Zustandswerten $L_1^x = \text{null (N)}$ bis $L_6^x = \text{riesig (R)}$ gehören. Daher wird das Muster in diesem Beispiel erkannt. Des Weiteren zeigt der quasikontinuierliche Verlauf des Erkennungsgrades x , dass der Zustand des Mustererkenners auch zwischen den verbal definierten Zustandswerten N, W, K, M, G und R liegen kann. Wie erwähnt, zeigen solche Zwischenwerte an, dass es zwar schon Anzeichen für das nächste erwartete Teilmuster gibt, diese Anzeichen allerdings für sein vollwertiges Erkennen noch nicht ausreichen.

Wie in Kapitel 4 erörtert wurde, ist der so entstandene sequentielle Mustererkenner kein Automat mehr. Da er jedoch nur eine Zustandsgröße besitzt, kann einfach mit Hilfe des Ähnlichkeitssatzes (Satz 1) festgestellt werden, ob der Mustererkenner automatenähnlich ist. Dazu reicht es aus zu überprüfen, ob die Regelbasis aus Abschnitt 8.2 regelstetig ist.

In Abbildung 8.10 ist ein Ausschnitt aus der Regelbasis des Beispielsystems für den Zustandswert $x = L_3^x$ gezeigt. Die Regelbasis zeigt regelstetige, aber auch nicht regelstetige Bereiche: Bei einem linguistischen Eingabevektor $(u_1, u_2) = (L_5^{u_1}, L_5^{u_2})$ ergibt sich als linguistischer Folgezustand L_2^x . Ändert man den Wert der linguistischen Eingabevariablen u_1 auf den nächst kleineren Wert, d.h. auf $u_1 = L_4^{u_1}$, so ist der linguistische Folgezustand L_1^x und nicht L_2^x . Da L_1^x aber der nächst kleinere linguistische Wert in Bezug auf den ursprünglichen linguistischen Folgezustandswert L_2^x ist, ist die Regelbasis in diesem Teilbereich regelstetig. Betrachtet man den linguistischen Eingabevektor $(u_1, u_2) = (L_6^{u_1}, L_3^{u_2})$,

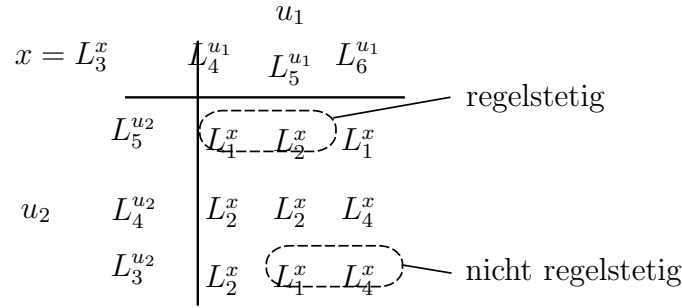


Abbildung 8.10: Auszug aus der Regelbasis eines sequentiellen Fuzzy-Mustererkenners für einen linguistischen Zustandswert $x = L_3^x$.

so ergibt sich L_4^x als linguistischer Folgezustand. Wählt man auch hier den nächst kleineren linguistischen Eingabewert für u_1 , d.h. $u_1 = L_5^x$, so ist der neue linguistische Folgezustand L_1^x und damit weder der nächst kleinere, nächst größere noch gleiche Wert wie der ursprüngliche linguistische Zustandswert L_4^x .

Die Regelbasis ist also nicht überall regelstetig und damit ist das rekurrente Fuzzy-System, welches den sequentiellen Fuzzy-Mustererkenner beschreibt, nach dem Ähnlichkeitssatz (Satz 1) nicht überall automatenähnlich.

Diese Situation tritt bei allen nichttrivialen sequentiellen Fuzzy-Mustererkennern auf, die nach diesem Ansatz erstellt wurden. Im Normalfall gibt es für linguistische Zustandswerte L_j^x mit $j \geq 3$ in der Regelbasis Rücksetz-, Übergangs-, Halte- und Rückschrittregeln. Damit stehen zwangsläufig die linguistischen Folgezustände L_1^x von Rücksetz- bzw. Endregeln und L_{j-1}^x , L_j^x bzw. L_{j+1}^x von Rückschritt-, Halte- bzw. Übergangsregeln direkt nebeneinander in der Regelbasis. Für $j \geq 4$ unterscheiden sich diese zwei Gruppen von Folgezuständen, L_1^x einerseits und L_{j-1}^x , L_j^x bzw. L_{j+1}^x andererseits, um mehr als eine Einheit. Damit ist ein solches rekurrentes Fuzzy-System nicht regelstetig und damit auch nicht automatenähnlich.

Was kann also passieren, wenn der sequentielle Fuzzy-Mustererkenner in einem nicht automatenähnlichen Bereich arbeitet? Durch die Gewichtung von Rücksetz- bzw. Endregeln einerseits und Rückschritt-, Halte- bzw. Übergangsregeln andererseits wird der sequentielle Fuzzy-Mustererkenner nicht vollständig zurückgesetzt und sein Zustandswert wird sich auf einem nicht genauer definierbaren Wert zwischen L_1^x und L_{j+1}^x befinden. Im Falle der Durchbruchfrüherkennung passen die dann anliegenden Eingangswerte nicht mehr zu dem neuen Zustandswert und so wird mit hoher Wahrscheinlichkeit das System im nächsten Zeitschritt auf den Initialisierungszustand L_1^x zurückgesetzt. Da der Mustererkenner in der Anwendung zusätzlich redundant ausgelegt ist, hat dieses Problem in der Praxis keine gravierenden Folgen.

Für einen allgemeinen Ansatz eines sequentiellen Fuzzy-Mustererkenners ist das nicht automatenähnliche Verhalten allerdings nicht wünschenswert. Man müsste sich bei jedem Entwurf Gedanken darüber machen, welche Auswirkungen das nicht automatenähnliche Verhalten haben kann und wie man in dem speziellen Fall damit umgehen soll. Daher wird im nächsten Abschnitt das Problem allgemein untersucht und ein neuer Ansatz vor-

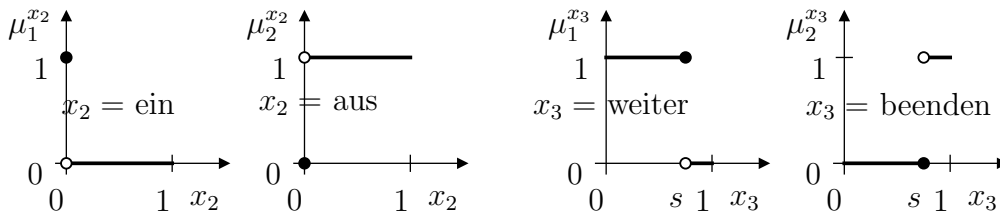


Abbildung 8.11: Zugehörigkeitsfunktionen für die Schaltvariablen x_2 bzw. x_3 .

gestellt, der sequentielle Fuzzy-Mustererkenner liefert, welche oben geschilderte Probleme nicht aufweisen.

8.4 Sequentielle Mustererkennung mit hybriden rekurrenten Fuzzy-Systemen

Im letzten Abschnitt wurde gezeigt, dass sich mit dem ersten vorgestellten Ansatz normalerweise keine automatenähnlichen rekurrenten Fuzzy-Systeme erstellen lassen, die als sequentielle Mustererkenner eingesetzt werden können. Die Ursache dafür ist, dass die Folgezustände von Rücksetz- und Endregeln auf der einen Seite und die Folgezustände von Rückschritt-, Halte- und Übergangsregeln auf der anderen Seite in der erzeugten Regelmatrix direkt nebeneinander liegen. Daraus resultiert eine nicht regelstetige Regelbasis und damit ein nicht automatenähnliches rekurrentes Fuzzy-System.

Um einen mit Automaten vergleichbaren sequentiellen Fuzzy-Mustererkenner konzipieren zu können, muss die Verwendung von Fuzzy-Logik in diesem Ansatz kritisch hinterfragt werden. Die Teilmerkmale im Eingangsraum gehen durch die unscharfe Beschreibung langsam ineinander über. Rückschritt-, Halte- und Übergangsregeln können mit diesem graduellen Übergang umgehen. So wird z.B. ein Übergang zum nächsten Erkennungsgrad s_j^x nur teilweise durchgeführt, wenn nur erste Anzeichen für das nächste Teilmuster vorliegen. Bei der Verarbeitung von Rücksetz- und Endregeln ist dies anders. Die Entscheidung über einen solchen Vorgang muss hart, d.h. crisp, getroffen werden, denn es macht keinen Sinn, den Mustererkenner nur teilweise in den Initialisierungszustand zu versetzen.

Um diese harten Entscheidungen in einem rekurrenten Fuzzy-System durchführen zu können, werden im Folgenden Schaltvariablen eingeführt. Schaltvariablen werden durch zwei linguistische Werte wie zum Beispiel ein/aus oder Fehler/OK beschrieben. Ihre Zugehörigkeitsfunktionen für den Fuzzifizierungsschritt sind unstetig und nehmen nur die Werte 0 oder 1 an. Ein Beispiel ist in Abbildung 8.11 zu sehen. Durch die Wahl der sprungförmigen Zugehörigkeitsfunktionen hat die so beschriebene Variable immer nur zu einem linguistischen Wert einen positiven Zugehörigkeitsgrad. Wie im Falle von stetigen Zugehörigkeitsfunktionen werden auch Singletons in der Defuzzifizierung verwendet. Durch den Einsatz der unstetigen Zugehörigkeitsfunktionen erhält das rekurrente Fuzzy-System einen hybriden Charakter, da es sowohl Schaltvariablen als auch stetige Zustandsvariablen verwendet.

Tabelle 8.2: Regelbasis des hybriden rekurrenten Fuzzy-Systems, welches zur sequentiellen Mustererkennung verwendet wird.

Regelart	$x_1(k)$	$x_2(k)$	$x_3(k)$	$\mathbf{u}(k)$	$x_1(k+1)$	$x_2(k+1)$	$x_3(k+1)$
Übergangsregel	$L_j^{x_1}, j < j_{max}$	ein	weiter	$B(L_{j+1}^{x_1})$	$L_{j+1}^{x_1}$	ein	weiter
Halteregel	$L_j^{x_1}, j < j_{max}$	ein	weiter	$B(L_j^{x_1})$	$L_j^{x_1}$	ein	weiter
Rückschrittregel	$L_j^{x_1}, j > 1$	ein	weiter	$B(L_{j-1}^{x_1})$	$L_{j-1}^{x_1}$	ein	weiter
Rücksetzregel I	$L_j^{x_1}$	ein	weiter	$A(L_j^{x_1})$	$L_1^{x_1}$	aus	weiter
Rücksetzregel II	beliebig	aus	weiter	U	$L_1^{x_1}$	ein	weiter
Endregel I	$L_{j_{max}}^{x_1}$	ein	weiter	$B(L_{j_{max}}^{x_1})$	$L_{j_{max}}^{x_1}$	ein	beenden
Endregel II	beliebig	beliebig	beenden	U	$L_1^{x_1}$	ein	weiter

Der Zustandsvektor des sequentiellen Fuzzy-Mustererkenners wird um zwei Schaltvariablen erweitert. Erstens x_2 , das die linguistischen Werte $L_1^{x_2} = \text{ein}$ und $L_2^{x_2} = \text{aus}$ annehmen kann und angibt, ob die Mustererkennung nun an ist oder rückgesetzt werden soll. Zweitens x_3 , das die linguistischen Werte $L_1^{x_3} = \text{weiter}$ und $L_2^{x_3} = \text{beenden}$ annehmen kann und angibt, ob die Erkennung weiter laufen soll, weil das Muster noch nicht vollständig entdeckt wurde, oder ob die Erkennung beendet werden soll, da das Muster erkannt wurde.

Die entsprechenden Zugehörigkeitsfunktionen sind in Abbildung 8.11 dargestellt. Der Wert, an dem der Sprung in den Zugehörigkeitsfunktionen stattfindet, wurde bei der Rücksetzvariablen x_2 zu null gewählt. Ein Rücksetzen erfolgt also, sobald auch nur das kleinste Anzeichen dafür vorhanden ist. Im Falle der Endvariablen x_3 zeigt der Wert s der Sprungstelle den Wahrheitswert an, den der oberste linguistische Erkennungszustand $x_1 = L_{j_{max}}^{x_1}$ erreichen muss, bevor die Erkennung als abgeschlossen bezeichnet wird. Die Singletonpositionen für die Defuzzifizierung werden jeweils an die Stellen $s_1^{x_2/3} = 0$ und $s_2^{x_2/3} = 1$ gesetzt.

Die Regeln für den neuen sequentiellen Fuzzy-Mustererkenner erhält man durch die Modifikation der Regeln des ursprünglichen Ansatzes von Tabelle 8.1. Während des normalen Betriebs, d.h. für die Belegung $x_2 = \text{ein}$ und $x_3 = \text{weiter}$ bleiben die Rückschritt-, Halte- und Übergangsregeln unverändert.

Die ursprüngliche Rücksetzregel wird in zwei neue Regeln aufgespalten: Wenn eine Rücksetzbedingung während des Betriebs auftaucht, so setzt die Rücksetzregel I die Rücksetzvariable x_2 auf den Wert $x_2 = \text{aus}$ und den Erkennungsgrad auf einen beliebig wählbaren Wert, z.B. $L_1^{x_1}$. Im zweiten Schritt wird durch Rücksetzregel II der Mustererkenner auf den Initialisierungszustand, d.h. $x_1 = L_1^{x_1}$, $x_2 = \text{ein}$, $x_3 = \text{weiter}$, zurückgesetzt.

Die ursprüngliche Endregel wird auch in zwei neue Regeln aufgespalten: Sobald der Erkennungsgrad x_1 den Wert $x_1 = L_{j_{max}}^{x_1}$ während des normalen Betriebs erreicht, wird Endregel I aktiviert. Endregel I ähnelt einer Halteregel mit dem Unterschied, dass die Endvariable x_3 auf den Wert $x_3 = \text{beenden}$ gesetzt wird. Im zweiten Schritt sorgt Endregel II dafür, dass der Initialisierungszustand wieder erreicht wird und eine erneute Mustererkennung gestartet werden kann.

Die neuen Regeln für den hybriden sequentiellen Fuzzy-Mustererkenner sind in Tabelle 8.2 zusammengefasst.

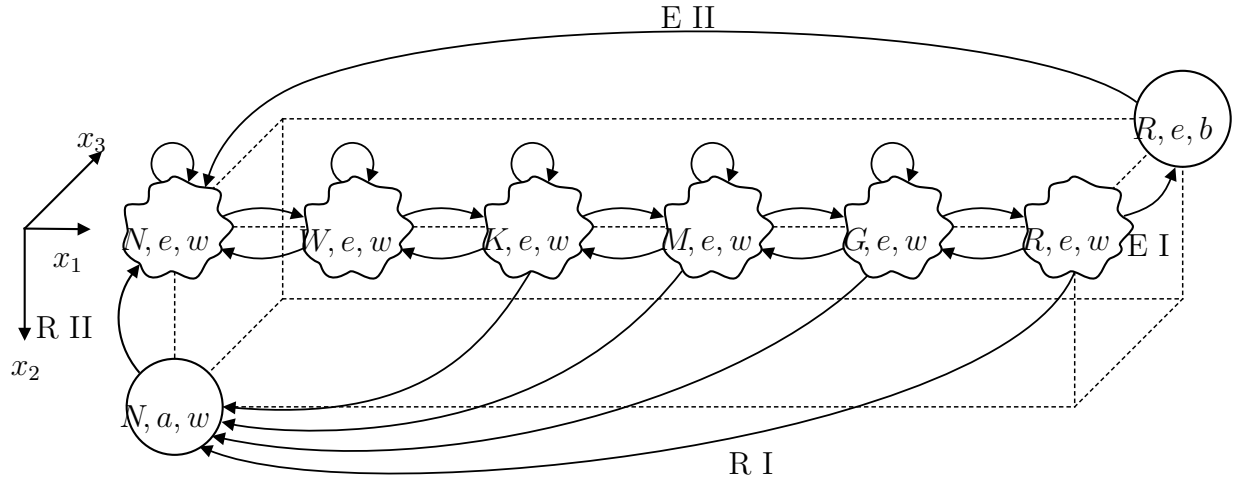


Abbildung 8.12: Zustandsgraph eines hybriden rekurrenten Fuzzy-Systems, basierend auf der Regelbasis aus Tabelle 8.2. Zur besseren Übersicht wurden nur die Transitions Pfeile der neuen Regelarten, d.h. Endregel I (E I), Endregel II (E II), Rücksetzregel I (R I), und Rücksetzregel II (R II), markiert.

Wiederum kann ein Zustandsgraph für das rekurrente Fuzzy-System erstellt werden, der in Abbildung 8.12 zu sehen ist. Dem Zustandsgraphen ist ein Quader hinterlegt, der den dreidimensionalen Zustandsraum X des rekurrenten Fuzzy-Systems andeuten soll. Die Zustände, die während des laufenden Betriebs, d.h. für $x_2 = \text{ein}$ und $x_3 = \text{weiter}$, angenommen werden, sind durch wellige Knoten dargestellt, um die unscharfe Natur der Zustände zu symbolisieren. Die anderen eingezeichneten Zustände werden durch runde Knoten dargestellt, was deren scharfer Natur entspricht. Einige Zustände, z.B. (W, a, w) oder (K, a, b) , die von keinem Transitions Pfeil erreicht werden, wurden nicht eingezeichnet, um die Übersichtlichkeit der Darstellung zu erhöhen.

Für einen sequentiellen Mustererkenner, der auf diesem Ansatz aufbaut gilt:

Satz 25

Der hybride sequentielle Fuzzy-Mustererkenner arbeitet wie ein Automat oder ist automatenähnlich, wenn die Regelbasis in der Komponente x_1 in den Bereichen der Rückschritt-, Halte- und Übergangsregeln und der Endregel I regelstetig ist.

Beweis:

Im Beweis werden nacheinander die verschiedenen Regelarten einzeln behandelt.

Endregel II ist nach Tabelle 8.2 die einzige Regelart, die linguistische Eingaben bearbeitet, bei denen $x_3 = \text{beenden}$ ist. Der Wahrheitswert für Endregel II ist entweder 0 oder 1, da die Bedingungen für x_1 , x_2 und $\mathbf{u}$ in der Prämisse immer voll erfüllt sind und die Bedingung für x_3 entweder ganz oder gar nicht erfüllt ist. Tritt also diese Regel in Kraft, so ist sie die einzige aktive Regel. In diesem Fall verhält sich das System genau wie ein Automat, da dieser auch nur jeweils eine einzelne Regel verarbeitet. Die Entscheidung über das Ende der Mustererkennung wird also scharf getroffen.

Rücksetzregel II kann analog zu Endregel II behandelt werden. Außer Endregel II ist Rücksetzregel II die einzige Regelart, die linguistische Eingabewerte mit $x_2 = \text{aus}$ behandelt. Wie bei Endregel II ist der Wahrheitswert der Rücksetzregel II entweder 0 oder 1. Tritt sie in Kraft, so ist sie auch die einzige aktive Regel. Dann wird der Zustand vollständig auf den Initialisierungszustand, (L_1^x, e, w) , rückgesetzt und das System verhält sich auch in diesem Fall wie ein Automat.

Rücksetzregel I kann im Gegensatz zu Endregel II und Rücksetzregel II gleichzeitig mit anderen Regeln gelten. Ihr Wahrheitswert kann daher auch Werte zwischen 0 und 1 annehmen. Ist Rücksetzregel I zu einem gewissen Grad aktiv, so wird der neue Wert von x_2 im nächsten Zeitschritt $k + 1$ die Kernposition $s_1^{x_2} = 0$ überschreiten. Aufgrund der sprungförmigen Zugehörigkeitsfunktion von x_2 aus Abbildung 8.11 wird im Schritt $k + 1$ entweder Rücksetzregel II oder in speziellen Fällen Endregel II gelten und zwar jeweils als die einzige geltende Regel, wie oben gezeigt wurde. Dann wird der Folgezustand $\mathbf{x}(k + 2) = (L_1^{x_1}, e, w)$ unabhängig von der Belegung von $x_1(k + 1)$ erreicht und der Rücksetzvorgang wird erfolgreich abgeschlossen. Selbst wenn bei der Berechnung des Wertes x_1 beim Auswerten der Rücksetzregel I die Regelstetigkeitsbedingung verletzt wird, wird die Funktionstüchtigkeit des Mustererkenners dadurch nicht beeinflusst. Auch in einem solchen Fall verhält sich das rekurrente Fuzzy-System wie eine Automat, bei dem die Zustandsvariable x_1 zeitweise mit einem beliebigen Wert belegt ist, der als "don't care"-Zustand angesehen werden kann.

Übergangs-, Halte- oder Rückschrittregeln treten nur im laufenden Betrieb, also für $x_2 = \text{ein}$ und $x_3 = \text{weiter}$, ein. Sie verändern auch nur die erste Komponente des Zustandsvektors. Deshalb ist in diesem Fall x_1 die einzige dynamische Größe. Daher kann für die Untersuchung der Automatenähnlichkeit der Zustandsvektor auf die erste Komponente x_1 reduziert werden. Nachdem die Regelbasis nach Voraussetzung in den Bereichen der Rückschritt-, Halte- und Übergangsregeln regelstetig bezüglich der Komponente x_1 ist, folgt mit dem Ähnlichkeitssatz, dass das System in diesen Bereichen auch automatenähnlich ist.

Endregel I ist die letzte zu behandelnde Regelart. Wenn sie als einzige Regel aktiv ist, so verhält sich das System wie ein Automat und ist auch automatenähnlich. In allen anderen Fällen kann sie für die Untersuchung nach Automatenähnlichkeit wie eine gedachte Halteregel für den Zustand $L_{j_{max}}^{x_1}$ behandelt werden, wie im Weiteren gezeigt wird:

Angenommen man würde in dem System statt der Endregel I eine Halteregel für den Zustand $L_{j_{max}}^{x_1}$ einsetzen. Ein solches System ist beim Eintreten von einer Kombination von Übergangs-, Halte- oder Rückschrittregeln automatenähnlich, wie oben gezeigt wurde. Zu den beiden Zustandsvektoren $\tilde{\mathbf{x}}(k)$ und $\tilde{\mathbf{x}}(k + 1) = \mathbf{f}(\tilde{\mathbf{x}}(k), \mathbf{u}(k))$ vor und nach der Abbildung gibt es daher jeweils einen mit ihnen verbundenen Kernpositionsvektor $\mathbf{s}_j^{\mathbf{x}}$ und $\mathbf{s}_w^{\mathbf{x}} = \mathbf{f}(\mathbf{s}_j^{\mathbf{x}}, \mathbf{s}_q^{\mathbf{u}})$ mit einem Kernpositionsvektor $\mathbf{s}_q^{\mathbf{u}}$, der mit $\mathbf{u}(k)$ verbunden ist. Da in diesem Fall nur Übergangs-, Rückschritt- und Halteregeln aktiv sind, liegt die dritte Komponente von $\tilde{\mathbf{x}}(k + 1)$ auf der Kernposition $s_1^{x_3}$ des linguistischen Wertes $L_1^{x_3} = \text{weiter}$, also $\tilde{x}_3(k + 1) = s_1^{x_3}$. Deshalb hat auch der mit $\tilde{\mathbf{x}}(k + 1)$ verbundene Kernpositionsvektor $\mathbf{s}_w^{\mathbf{x}}$ eine dritte Komponente $s_{w_3}^{x_3} = s_1^{x_3}$.

Die gleichen Kernpositionsvektoren $\mathbf{s}_j^{\mathbf{x}}$ und $\mathbf{s}_w^{\mathbf{x}}$ können auch für den Nachweis der Automatenähnlichkeit im Falle der Verwendung der Endregel I eingesetzt werden. Dazu betrachtet

man den Zustandsvektor $\mathbf{x}(k) = \tilde{\mathbf{x}}(k)$ und seinen Bildvektor $\mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k))$. Der Vektor $\mathbf{x}(k)$ ist wie schon $\tilde{\mathbf{x}}(k)$ mit dem Kernpositionsvektor $\mathbf{s}_j^x$ verbunden. Der neue Zustandsvektor $\mathbf{x}(k+1)$ unterscheidet sich von $\tilde{\mathbf{x}}(k+1)$ nur in der dritten Komponente, da sich Endregel I und die Halteregel im Konklusionsteil auch nur in der dritten Komponente unterscheiden. Es gilt also $\tilde{x}_1(k+1) = x_1(k+1)$ und $\tilde{x}_2(k+1) = x_2(k+1)$. Weiterhin liegt $x_3(k+1)$ zwischen den Kernpositionen $s_1^{x_3}$ und $s_2^{x_3}$ und kann bei der Suche nach einem mit ihm verbundenen Kernpositionsvektor immer auf $s_{w_3}^{x_3} = s_1^{x_3}$ abgerundet werden. Damit ist $\mathbf{s}_w^x$ auch mit $\mathbf{x}(k+1)$ verbunden. Endregel I führt wie auch die Halteregel zu automatenähnlichem Verhalten. $\square$

Im Normalfall ist die Regelbasis, eingeschränkt auf Übergangs-, Halte- und Rücksprungregeln, regelstetig. Es sind zwei seltene Situationen möglich, in denen die Regelstetigkeit verletzt werden kann. Erstens könnten für einen festen linguistischen Zustandswert L_j^x Rückschrittregeln und Übergangsregeln in der Regelmatrix direkt nebeneinander liegen. Zweitens könnte es sein, dass für einen festen linguistischen Eingangsvektor, $\mathbf{L}_q^u$ sowohl eine Rückschrittregel für einen linguistischen Zustandswert L_j^x als auch eine Übergangsregel für den linguistischen Zustandswert L_{j+1}^x gilt, d.h. es gibt ein $\mathbf{L}_q^u \in B(L_{j-1}^x) \cap B(L_{j+2}^x) \neq \emptyset$. In beiden Fällen sollte es jedoch möglich sein, die betroffenen Regeln in der Regelbasis zu trennen und eine regelstetige Regelbasis zu erhalten, indem man einen weiteren linguistischen Zustandswert zwischen L_j^x und L_{j+1}^x einführt. Dann ist das rekurrente Fuzzy-System auch in diesen Bereichen regelstetig und automatenähnlich.

Verwendet man im Beispiel der Durchbruchfrüherkennung die Zugehörigkeitsfunktionen aus Abbildung 8.9, so tritt keiner der oben angeführten Ausnahmefälle auf. Für einen solchen hybriden sequentiellen Fuzzy-Mustererkenner ist Satz 25 direkt anwendbar.

In den Bereichen, in denen ausschließlich Rückschritt-, Halte- und Übergangsregeln gelten und damit x_1 die einzige dynamische Zustandsgröße ist, gleicht das System einem eindimensionalen, terminierten, regelstetigen standardisierten rekurrenten Fuzzy-System. Daher ist auch nach Satz 16 sichergestellt, dass der Zustandswert gegen einen festen Endwert konvergieren wird, solange nur Rückschritt-, Halte- und Übergangsregeln gelten und der Eingangswert konstant gehalten wird. Solange keine Rücksetzregel gilt, ist das rekurrente Fuzzy-System zusätzlich monoton in der ersten Komponente. Damit sind nach Satz 11 auch chaotische Dynamiken in obigem Fall ausgeschlossen.

Hier werden nochmals die Vorteile eines sequentiellen Fuzzy-Mustererkenners mit einem hybriden rekurrenten Fuzzy-System zusammengefasst: Erstens baut das rekurrente Fuzzy-System auf den Regeln auf, die auch bei herkömmlichen sequentiellen Mustererkennern verwendet werden können. Die Regeln sind daher unmittelbar interpretierbar. Zweitens ergibt sich die Extrahierung der Merkmale aus einem Kurvenverlauf systematisch aus deren linguistischer Beschreibung und ihrer Fuzzifizierung. Drittens liefert der Fuzzy-Zustand nicht nur eine Angabe über die Zahl der erkannten Teilmuster wie im Fall des klassischen Automaten, sondern auch darüber, ob und bis zu welchem Grad sich das nächste Merkmal in den Eingangsgrößen abzeichnet, da sein Zustandswert auch Zwischenwerte annehmen kann. Viertens ist in vielen Fällen sichergestellt, dass das System sich wie ein Automat verhält oder automatenähnlich ist.

9 Anwendung: Verkehrssimulation

Rekurrente Fuzzy-Systeme eignen sich auch für die Modellierung von menschlichem Entscheidungsverhalten, wenn sich dieses adäquat durch Wenn-dann-Regeln beschreiben lässt. Dies wird in diesem Kapitel anhand einer Modellierung des Entscheidungsverhaltens von Autofahrern demonstriert. Die Autofahrer befinden sich mit vielen anderen auf einer zweispurigen Autobahn. Jeder Fahrer trifft in jedem Zeitintervall eine Entscheidung, mit welcher Geschwindigkeit er im nächsten Zeitintervall fahren soll und ob er die Spur wechseln will und wechseln kann. In die Entscheidungen gehen seine eigene Situation und die der unmittelbaren Verkehrsnachbarn ein.

Ausgangspunkt ist ein Modell, basierend auf einem zellulären Automaten. Die Regeln eines solchen Modells können, wie anschließend gezeigt wird, auch mit Fuzzy-Logik ausgewertet werden. Dabei ist es sinnvoll, einige Größen als unscharfe Fuzzy-Größen und andere Größen als binäre Größen zu behandeln. Das entstandene Fuzzy-System mit Rückführung entspricht in seiner Form zwar nicht der eines rekurrenten Fuzzy-Systems, kann aber in ein äquivalentes rekurrentes Fuzzy-System transformiert werden.

Mit Hilfe von Simulationsergebnissen wird die Plausibilität und Brauchbarkeit des Modells belegt.

9.1 Verkehrssimulation mit zellulären Automaten

Es gibt eine Reihe von Ansätzen für mikroskopische Verkehrsmodelle, d.h. Verkehrsmodelle, in denen einzelne Autos modelliert werden. Diese basieren auf gewöhnlichen Differentialgleichungen [94] oder zellulären Automaten [64, 102, 84]. Modelle mit zellulären Automaten können eine Vielzahl von Verkehrsteilnehmern berücksichtigen.

Ein zellulärer Automat besteht aus einer Menge Zellen, die in einer regelmäßigen Struktur, normalerweise einem Gitter, angeordnet sind. Alle Zellen sind identisch hinsichtlich ihrer Umgebung und der Regeln, die die Veränderung ihres Zustandswertes beschreiben. Die Regeln ziehen zu einem Zeitpunkt k den eigenen Zustandswert der Zelle und die Zustandswerte von Zellen in deren Umgebung in Betracht, um den neuen Zustandswert für den Zeitpunkt $k + 1$ zu bestimmen. Die Zellen verändern ihren Zustand zeitgleich in diskreten Zeitschritten.

Der hier eingesetzte zelluläre Automat modelliert eine zweispurige Straße. Die Zellen beschreiben diskrete Positionen auf der Straße. Sie können leer oder mit einem Auto besetzt sein, wie in Abbildung 9.1 dargestellt ist. In jedem Simulationsschritt verlassen die Autos

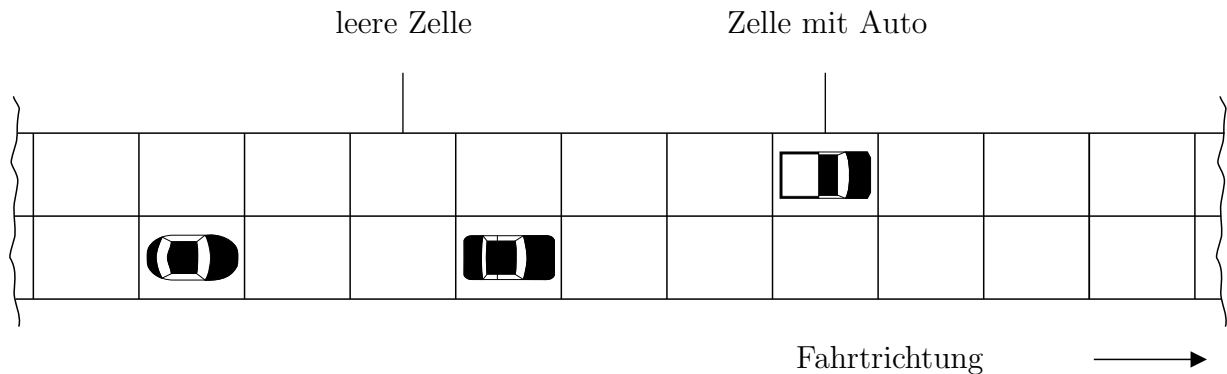


Abbildung 9.1: Modellierung einer zweispurigen Autobahn als zellulärer Automat

im Normalfall eine bestimmte Zelle und wechseln zu einer neuen Zelle. Überschreitet dabei ein Auto das Ende der Straße, so betritt es einen neuen Straßenabschnitt.

Im hier betrachteten Beispielsystem wird eine Ringstraße simuliert, die für die bessere Darstellung an einem fiktiven Punkt aufgeschnitten wurde. Daher werden die Autos, die das fiktive Ende der Ringstraße überschreiten, an den fiktiven Anfang der Straße zurückgesetzt. Auf der Ringstraße bleibt auch die Anzahl der Zellen, in denen sich Autos befinden, gleich.

Während der Simulation gibt es keine Beeinflussung durch externe Größen. Der zeitliche Verlauf hängt nur von der Anfangsbelegung der Zellen und den weiteren internen Zuständen der besetzten Zellen, d.h. der Autos, ab. Eine besetzte Zelle wird durch die folgenden Zustandsgrößen vollständig beschrieben:

- *Position (Pos)*
Die Position beschreibt die ganzzahlige x-Koordinate der vom Auto belegten Zelle. Der Wertebereich erstreckt sich von 1 bis zur Zahl der Zellen auf der Straße.
- *Spur (Sp)*
Die Spur beschreibt die y-Koordinate des Autos. Im betrachteten Fall gibt es zwei Spuren (rechts und links).
- *Mögliche Geschwindigkeiten (v_l, v_r)*
Die Geschwindigkeiten v_l bzw. v_r sind die maximal möglichen Geschwindigkeiten, mit denen der Fahrer auf der linken bzw. rechten Spur fahren kann. Sie sind in vier Geschwindigkeitsklassen eingeteilt: Stillstand, langsam, mittel oder schnell. Sie beschreiben die Bewegung eines Autos um 0, 1, 2 oder 3 Zellen pro Zeitschritt.
- *Blinker (Bl)*
Der Blinker kann an oder aus sein. Er ermöglicht eine realitätsnahe Simulation des Spurwechselverhaltens ohne Unfälle.
- *Charakter (Ch)*
Die Autofahrer können verschiedene Fahrcharaktere haben, die während der Simulation unverändert bleiben. Sportliche Fahrer ($Ch = \text{sportlich}$) wechseln die Spur

schon mit minimalem Sicherheitsabstand, d.h. der Abstand reicht gerade aus, um Unfälle zu vermeiden. Ihre Höchstgeschwindigkeit ist schnell ($v_{max} = 3$). Normale Fahrer ($Ch = \text{normal}$) wechseln die Spur eher zögerlich, d.h. mit einem erhöhten Sicherheitsabstand nach hinten. Ihre Höchstgeschwindigkeit ist auch schnell ($v_{max} = 3$). Defensive Fahrer ($Ch = \text{defensiv}$), wechseln die Spur auch nur zögerlich. Ihre Höchstgeschwindigkeit ist nur mittel ($v_{max} = 2$).

Die Veränderung der Zellbesetzungen und deren zusätzlichen internen Zustandswerte können durch einen Regelsatz beschrieben werden. Prinzipiell sind diese Regeln auf alle Zellen anzuwenden. Allerdings wäre es ein enormer Aufwand, für jede leere Zelle zu überprüfen, ob sie tatsächlich im nächsten Zeitschritt durch ein Auto besetzt wird. Stattdessen werden nur die durch Autos besetzten Zellen betrachtet. Für jedes dieser Autos wird die Änderung der internen Zustände bestimmt und errechnet, in welcher Zelle es zu liegen kommt. Damit sind auch alle leeren Zellen berücksichtigt, die im nächsten Zeitschritt ein Auto enthalten.

Da die Autos innerhalb des Modells erhalten bleiben und jedes Auto seine Zustände zeitlich ändert, kann man jedes Auto als eigenständiges dynamisches System auffassen, welches die Zustände der benachbarten (besetzten) Zellen beachtet. Im Idealfall geben die Regeln, die die Änderung der Zustandsgrößen beschreiben, das menschliche Entscheidungsverhalten des Autofahrers transparent wieder. Die Entscheidungen werden nicht alle gleichzeitig getroffen, sondern erfolgen in mehreren Einzelschritten. Der hier gewählte Ansatz beschreibt das Entscheidungsverhalten in den folgenden Schritten:

- (1) Der Fahrer stellt seine Fahrspur und Geschwindigkeit fest.
- (2) Der Fahrer schätzt ab, wie schnell er im nächsten Zeitschritt auf der linken bzw. rechten Spur fahren kann.
- (3) Der Fahrer überlegt, ob er die Spur wechseln möchte und ob ein Spurwechsel ohne Gefahr möglich ist.
- (4) Der Fahrer bestimmt seine zukünftige Fahrspur und ob er den Blinker im nächsten Zeitschritt setzen will.

Diese Schritte werden im Folgenden einzeln erläutert.

Schritt 1: Der Fahrer stellt seine Fahrspur und Geschwindigkeit fest.

Die Fahrspur ist durch die Zustandsgröße Sp festgelegt. Die *Geschwindigkeit* v des Autos ergibt sich aus der Spur und den möglichen Geschwindigkeiten v_l und v_r : bei $Sp = \text{rechts}$ zu $v = v_r$, und bei $Sp = \text{links}$ zu $v = v_l$. Die Geschwindigkeit v ist in die gleichen Klassen unterteilt wie die Geschwindigkeiten v_l und v_r , also in Stillstand, langsam, mittel und schnell, bzw. 0, 1, 2 oder 3 Zellen pro Zeiteinheit.

Schritt 2: Der Fahrer schätzt ab, wie schnell er im nächsten Zeitschritt auf der linken bzw. rechten Spur fahren kann.

Zuerst muss der Fahrer die Verkehrsteilnehmer vor sich in Betracht ziehen. Autos die weiter als acht Zellen vor dem betrachteten Auto fahren, spielen für die Entscheidung des Fahrers keine Rolle. Relevant sind außerdem nur der direkte Vordermann auf derselben Spur und der Vordermann auf der anderen Spur. Anstatt alle Fahrzeuge in den nächsten acht Zellen

		Δ_{VMl} bzw. Δ_{VMr}											Δ_{VMl} bzw. Δ_{VMr}											Δ_{VMl} bzw. Δ_{VMr}								
		0	1	2	3	4	5	6	7	8			0	1	2	3	4	5	6	7	8			0	1	2	3	4	5	6	7	8
v_{VMl}	0	0	0	0	0	1	1	2	2	2	0	0	0	0	1	1	2	2	2	3	0	0	0	1	1	2	2	2	3	3		
bzw.	1	0	0	0	0	1	1	2	2	2	1	0	0	0	1	1	2	2	2	3	1	0	0	1	1	2	2	2	3	3		
v_{VMr}	2	0	0	0	1	1	2	2	2	2	2	0	0	1	1	2	2	2	3	3	2	0	1	1	2	2	2	3	3	3		
	3	0	0	1	2	2	2	2	2	2	3	0	1	2	2	2	3	3	3	3	3	1	2	2	2	3	3	3	3	3		
		$Ch = \text{defensiv}$											$Ch = \text{normal}$											$Ch = \text{sportlich}$								

Abbildung 9.2: Regelbasis zur Ermittlung der vorläufigen möglichen Geschwindigkeiten (v_r^*, v_l^*)

zu berücksichtigen, werden durch eine Vorverarbeitung diese speziellen Verkehrsteilnehmer gesucht. In die Überlegung des Fahrers gehen dabei die folgenden Eingangsgrößen ein:

- *Geschwindigkeiten der Vordermänner* (v_{VMl}, v_{VMr})
Die Geschwindigkeiten v_{VMl} des linken Vordermanns und v_{VMr} des rechten Vordermanns sind in die gleichen Klassen unterteilt wie die Geschwindigkeiten v_l und v_r .
- *Abstände zu den Vordermännern* ($\Delta_{VMl}, \Delta_{VMr}$)
Die Abstände Δ_{VMl} zum linken Vordermann und Δ_{VMr} zum rechten Vordermann liegen zwischen 0 und dem maximal betrachteten Abstand von 8 Zellen.

Da die Entscheidungen der anderen Verkehrsteilnehmer unbekannt sind, muss der Fahrer vom kritischsten Fall ausgehen: einer Vollbremsung des Vordermanns. In dieser Situation bremst der Vordermann in jedem kommenden Zeitschritt um eine Geschwindigkeitseinheit, bis er zum Stehen kommt. Eine Erhöhung der eigenen Geschwindigkeit ist nur dann erlaubt, wenn bei einer Vollbremsung des Vordermanns eine eigene anschließende Vollbremsung ausreicht, um das eigene Fahrzeug hinter dem Vordermann anzuhalten.

Je nach Fahrcharakter geht der Fahrer unterschiedlich mit diesem Problem um. Für den sportlichen Fahrer sind die Regeln so ausgelegt, dass ein Auffahrunfall gerade so verhindert wird, d.h., dass er in einer solchen Extremsituation genau hinter dem Vordermann zum Stehen kommt. Ein sportlicher Fahrer hält also den minimalen Abstand zum Vordermann. Normale Fahrer halten eine Zelle mehr Sicherheitsabstand. Defensive Fahrer halten zwei Zellen mehr Sicherheitsabstand und haben eine Maximalgeschwindigkeit von $v_{max} = 2$.

Lässt man die eigene Geschwindigkeit unberücksichtigt, so lassen sich die vorläufigen möglichen Geschwindigkeiten (v_l^* und v_r^*) aus den Tabellen in Abbildung 9.2 ablesen. Beispielsweise ergibt sich so für $Ch = \text{sportlich}$, $v_{VMr} = 1$ und $\Delta_{VMr} = 5$ eine vorläufige mögliche Geschwindigkeit $v_r^* = 2$.

Die Entscheidung über die vorläufigen möglichen Geschwindigkeiten (v_l^* und v_r^*) müssen zuweilen allerdings revidiert werden. Wegen der beschränkten Beschleunigungsmöglichkeiten kann der Fahrer die Geschwindigkeit seines Fahrzeugs in einem Zeitschritt maximal um eine Geschwindigkeitsklasse erhöhen oder verringern. Außerdem sollte ein Fahrer auf der rechten Spur einen Vordermann auf der linken Spur nicht überholen; er sollte also nach

		v_r^*						v_r^*						v_r^*						v_r^*			
		0	1	2	3			0	1	2	3			0	1	2	3			0	1	2	3
v	0	0	0	0	0	0	0	0	1	1	1	0	0	0	1	1	1	0	0	0	1	1	1
	1	0	0	0	0	1	0	0	1	1	1	1	0	0	1	2	2	1	0	0	1	2	2
	2	1	1	1	1	2	1	1	1	1	1	2	1	1	1	2	2	2	1	1	1	2	3
	3	2	2	2	2	3	2	2	2	2	2	3	2	2	2	2	2	3	2	2	2	2	3
		$v_{VMl} = 0$						$v_{VMl} = 1$						$v_{VMl} = 2$						$v_{VMl} = 3$			

Abbildung 9.3: Regelbasis zur Ermittlung der möglichen Geschwindigkeiten (v_l, v_r)

		v_r						v_r						v_r			
		0	1	2	3			0	1	2	3			0	1	2	3
v_l	0	nein	nein	ja	ja	0	nein	nein	nein	ja	ja	0	nein	nein	nein	nein	nein
	1	nein	nein	ja	ja	1	nein	nein	nein	ja	ja	1	ja	nein	nein	nein	nein
	2	nein	nein	ja	ja	2	nein	nein	nein	ja	ja	2	ja	ja	nein	nein	nein
	3	nein	nein	ja	ja	3	nein	nein	nein	ja	ja	3	ja	ja	ja	ja	nein
		$Sp = \text{links}, v_{max} = 2$						$Sp = \text{links}, v_{max} = 3$						$Sp = \text{rechts}, v_{max} = \text{beliebig}$			

Abbildung 9.4: Regelbasis zur Ermittlung des Spurwechselwunsches (Sw)

Möglichkeit seine Geschwindigkeit v_r nicht höher als die seines linken Vordermanns (v_{VMl}) wählen. Aus diesen Überlegungen ergeben sich weitere Regeltabellen in Abbildung 9.3, aus denen sich mit der vorläufigen möglichen Geschwindigkeit v_r^* die tatsächlich mögliche Geschwindigkeit v_r für die rechte Spur direkt ablesen lässt. Für zum Beispiel $v_r^* = 2$, $v = 2$ und $v_{VMl} = 1$ ergibt sich so $v_r = 1$. Für die Bestimmung der tatsächlich möglichen Geschwindigkeit v_l für die linke Spur aus der vorläufig möglichen Geschwindigkeit v_l^* muss kein Rechtsüberholverbot beachtet werden. Es gelten also analoge Regeln wie für die Bestimmung von v_r im Fall $v_{VMl} = 3$, d.h. in dem Fall, in dem der linke Vordermann so schnell fährt, dass er nicht beachtet werden muss. Beispielsweise für $v_l^* = 3$, $v = 2$ und beliebiges v_{VMl} erhält man die mögliche Geschwindigkeit $v_l = 3$.

Schritt 3: Der Fahrer überlegt, ob er die Spur wechseln möchte und ob ein Spurwechsel ohne Gefahr möglich ist.

Ein Fahrer auf der rechten Spur wünscht nur auf die linke Spur zu wechseln, falls er dort schneller fahren kann als rechts, also falls $v_l > v_r$ ist. Ein Fahrer auf der linken Spur entschließt sich auf die rechte Spur zu wechseln, falls er rechts mit Maximalgeschwindigkeit fahren kann. Defensive Fahrer mit $v_{max} = 2$ wollen also schon bei $v_r = 2$ nach rechts einscheren, normale und sportliche Fahrer mit $v_{max} = 3$ erst bei $v_r = 3$.

In die Bestimmung des Wunsches zum Spurwechsel gehen also die möglichen Geschwindigkeiten auf beiden Spuren (v_r und v_l), die eigene Spur (Sp) und die eigene Höchstgeschwindigkeit (v_{max}) ein. In Abbildung 9.4 ist zum einen die Regelbasis für die linke Spur bei einer möglichen Maximalgeschwindigkeit $v_{max} = 2$ bzw. $v_{max} = 3$ und zum anderen die Regelbasis für die rechte Spur unabhängig von der möglichen Maximalgeschwindigkeit zu

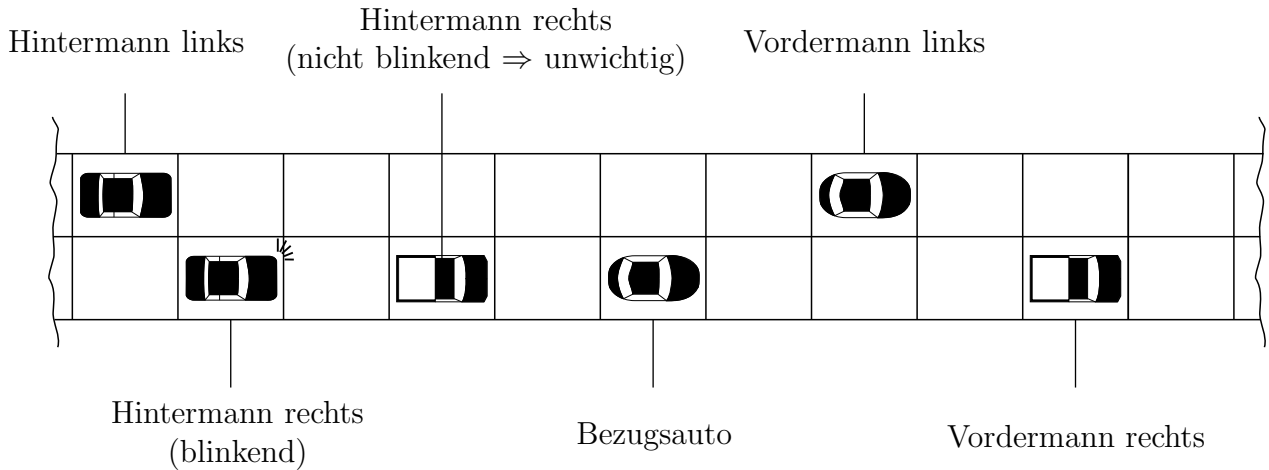


Abbildung 9.5: Wichtige Verkehrsteilnehmer, die der Fahrer im Bezugsauto beachten muss.

sehen. Daraus kann der *Spurwechselwunsch* (Sw_w) direkt abgelesen werden.

Wenn der Fahrer die Spur wechseln will, so darf er nicht einfach ausscheren. Ein von hinten kommendes, schnelles Fahrzeug könnte eventuell dann einen Unfall nicht mehr vermeiden. Daher muss sich der Fahrer bei einem Spurwechsel im nächsten Entscheidungsschritt über die *Spurwechselerlaubnis* (Swe) Gedanken machen.

Bei einem Spurwechsel muss der Fahrer seine Hintermänner beachten, insbesondere seinen direkten Hintermann auf der anderen Spur und den ersten Hintermann auf der eigenen Spur, der einen Blinker gesetzt hat und damit auch eventuell die Spur wechseln wird. Ein Beispiel ist in Abbildung 9.5 zu sehen.

Wie schon bei den Vordermännern werden durch eine Vorverarbeitung die relevanten Hintermänner ausgesucht und der Abstand Δ_{HML} zum linken Hintermann und dessen Geschwindigkeit v_{HML} bzw. der Abstand Δ_{HMr} zum rechten Hintermann und dessen Geschwindigkeit v_{HMr} bestimmt.

Die Erlaubnis zum Spurwechsel hängt auch von der zukünftigen eigenen Geschwindigkeit auf der anderen Spur ab, also v_l bei $Sp = \text{rechts}$ und v_r bei $Sp = \text{links}$. Es muss gewährleistet sein, dass bei einem Spurwechsel mit anschließender Vollbremsung des eigenen Fahrzeugs alle Hintermänner rechtzeitig zum Stehen kommen können. In Bezug auf beide Hintermänner wird die Spurwechselerlaubnis getrennt bestimmt. Nur wenn in beiden Fällen ein Spurwechsel erlaubt ist, darf er vollzogen werden.

Fahrer mit sportlichem Charakter verhalten sich so, dass bei einem Spurwechsel und anschließender Vollbremsung im Extremfall der Hintermann gerade noch rechtzeitig zum Stehen kommt. Aus der Tabelle in Abbildung 9.6 lässt sich die Spurwechselerlaubnis wie folgt ablesen: Ist der Wert der möglichen Geschwindigkeit v_l bzw. v_r größer als der Wert, der sich aus der Tabelle ergibt, so ist ein Spurwechsel noch erlaubt ($Swe = \text{ja}$), andernfalls nicht mehr ($Swe = \text{nein}$). Beispielsweise muss bei einem beabsichtigten Spurwechsel von rechts nach links der Wert 1, der sich für $\Delta_{HML} = 4$ und $v_{HML} = 3$ ergibt, durch v_l übertroffen werden. Für mögliche Geschwindigkeiten v_l von 0 oder 1 ist der Spurwechsel

		Δ_{HMI} bzw. Δ_{HMr}											Δ_{HMI} bzw. Δ_{HMr}											Δ_{HMI} bzw. Δ_{HMr}								
		0	1	2	3	4	5	6	7	8			0	1	2	3	4	5	6	7	8			0	1	2	3	4	5	6	7	8
v_{HMI}	0	3	3	0	0	0	0	0	0	0	0	3	3	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0
bzw.	1	3	3	1	1	0	0	0	0	0	1	3	3	1	1	0	0	0	0	0	0	1	3	1	1	0	0	0	0	0	0	0
v_{HMr}	2	3	3	2	1	1	1	1	0	0	2	3	3	2	2	2	1	1	0	0	0	2	3	2	2	2	1	1	0	0	0	0
	3	3	3	2	2	2	1	1	0	0	3	3	3	2	2	2	1	1	0	0	0	3	3	2	2	2	1	1	0	0	0	0
$Ch = \text{defensiv}$											$Ch = \text{normal}$											$Ch = \text{sportlich}$										

Abbildung 9.6: Tabelle zur Errechnung der Mindestgeschwindigkeit für verschiedene Fahrcharaktere (Ch) in Abhängigkeit von dem Abstand zum Hintermann (Δ_{HMI} bzw. Δ_{HMr}) und dessen Geschwindigkeit (v_{HMI} bzw. v_{HMr}), aus der man nach dem Vergleich mit der tatsächlichen Geschwindigkeit die Spurwechselerlaubnis (Swe) erhält.

nicht erlaubt, wohl aber für Werte von 2 oder 3.

Wie für den sportlichen Fahrer kann man die Spurwechselerlaubnis für den normalen Fahrer aus der passenden Tabelle aus Abbildung 9.6 ermitteln. Im Gegensatz zum sportlichen Fahrer hält ein normaler Fahrer einen zusätzlichen Sicherheitsabstand zum Hintermann. Dies wird erreicht, indem alle Einträge in der Tabelle für den sportlichen Fahrer um eine Spalte nach rechts verschoben werden, um die Einträge für den normalen Fahrer zu erhalten.

Defensive Fahrer halten denselben Sicherheitsabstand wie normale Fahrer, wie man aus dem Vergleich der entsprechenden Tabellen in Abbildung 9.6 ablesen kann. Sie haben aber im Fall eines mit $v_{HMI} = 2$ bzw. $v_{HMr} = 2$ fahrenden Hintermanns und einer eigenen möglichen Geschwindigkeit von $v_l = 2 (= v_{max})$ zwei Zellen Sicherheitsabstand weniger als der normale Fahrer. Diese Änderung erzeugt aus folgenden Gründen keine Unfälle: Wechselt der defensive Fahrer auf die rechte Spur, besteht keine Gefahr für den Hintermann, da dieser, wie im letzten Entscheidungsschritt erklärt wird, nicht schneller als der defensive Fahrer fahren darf. Wechselt der defensive Fahrer auf die linke Spur, hat er, inklusive Bremsweg des Vordermanns, mindestens 6 Zellen Freiraum vor sich. Er hat somit in den nächsten beiden Zeitschritten die Gelegenheit, seine Maximalgeschwindigkeit von $v_{max} = 2$ beizubehalten. Dies lässt den Hintermännern in dieser Situation ausreichend Zeit einen Auffahrunfall durch Abbremsen zu verhindern¹.

Schritt 4: Der Fahrer bestimmt seine zukünftige Fahrspur und ob er den Blinker im nächsten Zeitschritt setzen will.

Der Fahrer wechselt nur dann die Spur, wenn er im vorherigen Zeitschritt den Blinker

¹Ohne diese Änderung kommt es vor, dass ein links fahrender, defensiver Fahrer von einem rechts fahrenden sportlichen Fahrer begleitet wird. Dabei wagt der defensive Fahrer wegen eines geringen Sicherheitsabstandes nach hinten nicht auf die rechte Spur zu wechseln. Der sportliche Fahrer dagegen würde nicht nach links wechseln, da er dort auch nicht schneller vorankommt, und darf auch nicht rechts überholen. Es kommt zu einer Pattsituation. Durch die Änderung des Spurwechselverhaltens des defensiven Fahrers wird diese Situation aufgelöst, indem der defensive Fahrer eben doch die Spur wechselt.

		<i>Sww</i>				<i>Sww</i>	
		nein	ja			nein	ja
<i>Swe</i>	nein	links	links	<i>Swe</i>	nein	links	links
	ja	links	links		ja	links	rechts
<i>Bl</i> = aus				<i>Bl</i> = an			

Abbildung 9.7: Regelbasis zur Berechnung der neuen Spur für ein Fahrzeug auf der linken Spur

		<i>Sww</i>				<i>Sww</i>	
		nein	ja			nein	ja
<i>Swe</i>	nein	aus	an	<i>Swe</i>	nein	aus	an
	ja	aus	an		ja	aus	aus
<i>Bl</i> = aus				<i>Bl</i> = an			

Abbildung 9.8: Regelbasis zur Errechnung des neuen Blinkerzustandes

gesetzt hat, um die anderen Verkehrsteilnehmer vorzuwarnen, und sowohl der Spurwechselwunsch als auch die Spurwechselerlaubnis vorliegt. Die Regeln sind für den Wechsel von links nach rechts in Abbildung 9.7 zu sehen. Für den Wechsel von rechts nach links sind alle Einträge in den abgebildeten Tabellen durch den jeweils anderen Wert auszutauschen, also rechts $\leftrightarrow$ links.

Der Blinker wird dann angeschaltet, falls der Wunsch für einen Spurwechsel besteht und kein Spurwechsel in diesem Zeitschritt stattfindet. Die Regeln hängen von *Sww*, *Swe* und *Bl* ab und sind in Abbildung 9.8 zu sehen.

Nach diesen vier Entscheidungsschritten sind die Geschwindigkeiten v_l und v_r , die Spur *Sp* und der Blinkerzustand *Bl* des Autos des Fahrers für den nächsten Zeitschritt festgelegt. Der Fahrcharakter bleibt unverändert. Mit der Spur lässt sich aus den Geschwindigkeiten v_l und v_r dann die tatsächliche Geschwindigkeit v auswählen. So ergibt sich die neue Position aus der alten Position plus der Geschwindigkeit v . Damit sind alle neuen Zustandsgrößen dieses Autos bekannt und können in einer Simulation berechnet werden. Die Rechenschritte sind in Abbildung 9.9 in Form eines Blockschaltbildes zusammengefasst.

Während die Simulation mit dem Ansatz des zellulären Automaten sehr schnell vor sich geht, nimmt man bei diesem Ansatz einige Nachteile in Kauf. So können die Autos nur diskrete Positionen aufgrund der Zellen annehmen. Auch die Geschwindigkeiten sind auf wenige feste Werte beschränkt.

Diese Beschränkungen kann man durch den Einsatz von Fuzzy-Logik aufweichen. Im nächsten Kapitel wird gezeigt, dass mit Fuzzy-Systemen auch kontinuierliche Positionen und Ge-

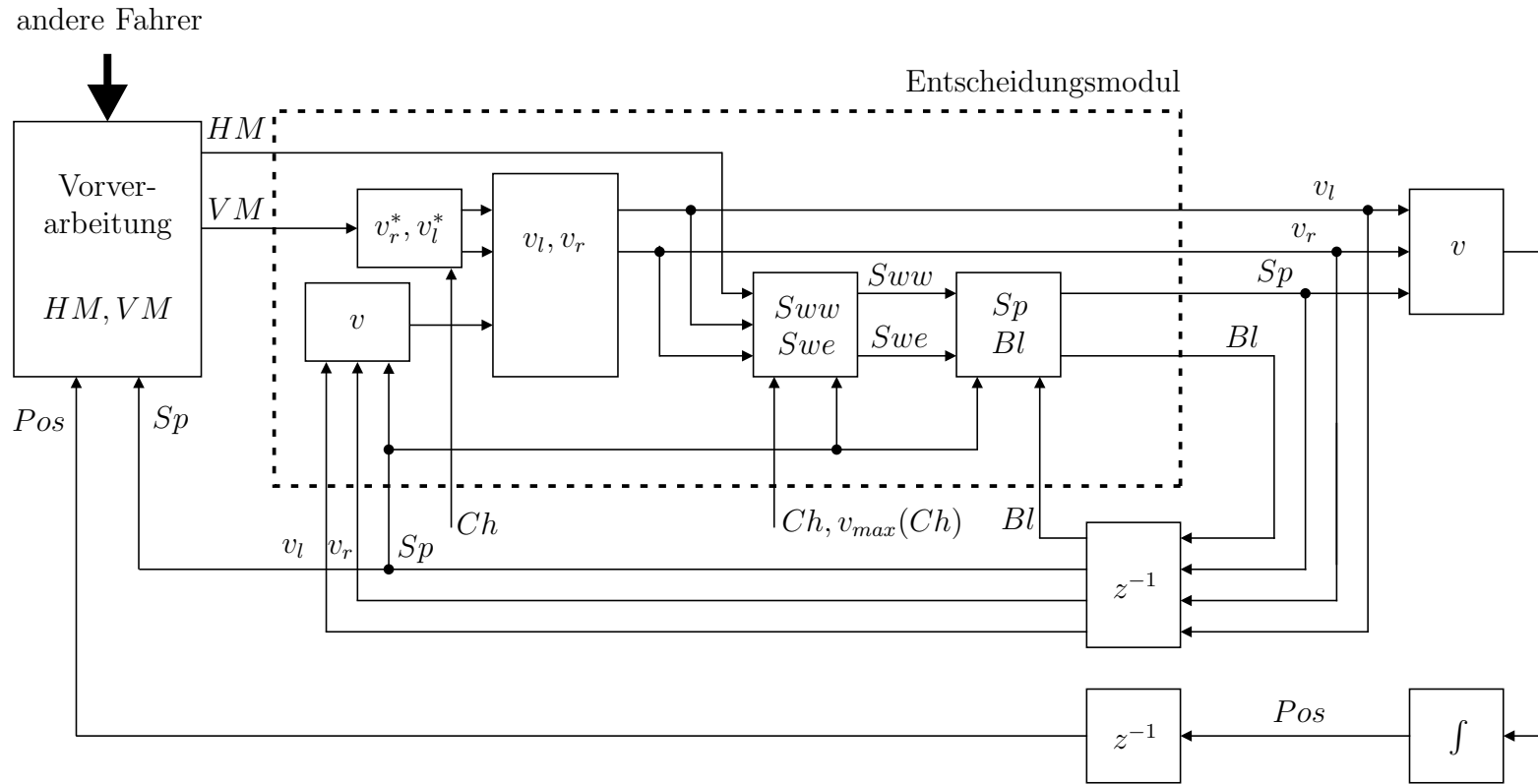


Abbildung 9.9: Modularer Aufbau des Automaten, der die Entscheidung eines Verkehrsteilnehmers modelliert. Die Wechselwirkung mit den anderen Fahrern, d.h. dessen Vordermänner (VM) und Hintermänner (HM) wird über die Vorverarbeitung realisiert. Unter HM sind die Geschwindigkeiten v_{HMI} , v_{HM_r} und Abstände Δ_{HMI} , Δ_{HM_r} zu den beiden Hintermännern zusammengefasst. Analoges gilt für die Größen der Vordermänner in VM .

schwindigkeiten verarbeitet werden können. Außerdem ermöglicht es die Simulation von Fahrern mit differenzierteren Charaktereigenschaften, wie zu Beispiel Kleinbusfahrern, die als Mischung aus defensivem und normalem Fahrer modelliert werden können.

9.2 Verkehrssimulation mit rekurrenten Fuzzy-Systemen

Im letzten Abschnitt wurde das Entscheidungsverhalten der Autofahrer in mehrere Teilschritte zerlegt. Für jeden Block im Modell der Entscheidungsfindung aus Abbildung 9.9 wurde eine Reihe von Regeln aufgestellt. Diese Regeln lassen sich entweder direkt aus den angegebenen Tabellen ablesen oder nach einer Anleitung im Text erstellen.

Die aufgestellten Regeln können auch als unscharfe Fuzzy-Regeln aufgefasst werden. Bei der Modellierung der Teilblöcke aus dem Modell aus Abbildung 9.9 mit Fuzzy-Logik dienen diese Tabellen bzw. die daraus abgeleiteten Tabellen dann als vollständige und widerspruchsfreie Regelbasen der Teilsysteme. Es stellt sich nun die Frage, wie diese geeignet ausgewertet werden können und ob das resultierende Modell ein rekurrentes Fuzzy-System darstellt.

Es wird der folgende Weg eingeschlagen: Alle Größen werden beim Eintritt in den Entscheidungsblock, der in Abbildung 9.9 gestrichelt eingetragen ist, mit geeigneten Zugehörigkeitsfunktionen fuzzifiziert. Innerhalb des Entscheidungsblocks wird nur mit Fuzzy-Größen gerechnet, d.h. es findet innerhalb des Entscheidungsblocks keine Fuzzifizierung und keine Defuzzifizierung statt. Als Inferenz-Operatoren kommen Multiplikationsoperatoren für die Aggregation und Implikation und der Summationsoperator für die Akkumulation zum Einsatz. Erst beim Austritt der Größen aus dem Entscheidungsblock werden diese dann mit der Center of Singleton-Methode [11, 97] defuzzifiziert.

Für die Zwischengrößen v_l^* , v_r^* , S_{ww} und S_{we} müssen keine Zugehörigkeitsfunktionen festgelegt werden, da sie innerhalb des Entscheidungsblocks weder fuzzifiziert noch defuzzifiziert werden.

Die Abstände zu den Verkehrsnachbarn ($\Delta_{V_{ML}}$, $\Delta_{V_{Mr}}$, $\Delta_{H_{ML}}$, $\Delta_{H_{Mr}}$) werden mit äquidistanten Dreiecksfunktionen fuzzifiziert, die ihre Maxima an den Positionen 0, 1, ..., 8 haben. Der Abstand zwischen benachbarten Positionen entspricht der ursprünglichen Zellenlänge. Auch für die Fuzzifizierung der Fahrcharaktere werden äquidistante Dreiecksfunktionen verwendet, deren Maxima bei eins für den defensiven, bei zwei für den normalen und bei drei für den sportlichen Fahrcharakter liegen. Für die Geschwindigkeiten (v_l , v_r , $v_{V_{ML}}$, $v_{V_{Mr}}$, $v_{H_{ML}}$, $v_{H_{Mr}}$) bieten sich für ihre Fuzzifizierung auch äquidistante Dreiecksfunktionen an, die an den Stellen 0, 1, 2 und 3 ihren maximalen Wert annehmen. An diesen Stellen liegen auch die Singletons, die für die Defuzzifizierung von v_l und v_r verwendet werden.

Spur und Blinker bilden keine unscharfe Größen: Entweder ist der Blinker an oder aus. Entweder wird links oder rechts gefahren. Dies wird erreicht, indem diese Größen mit sprungförmigen Zugehörigkeitsfunktionen fuzzifiziert werden. Die Sprungstelle soll in der Mitte der beiden Singletonpositionen liegen, die für die Defuzzifizierung genutzt werden, z.B. bei 0.5, wenn 0 für einen ausgeschalteten Blinker und 1 für einen angeschalteten Blinker verwendet wird.

Wenn die Teilsysteme wie oben beschrieben ausgewertet werden, so liegt ein modulares Fuzzy-System mit Rückführung vor. Die Form entspricht allerdings nicht der eines rekurrenten Fuzzy-Systems. Wie im Nachfolgenden gezeigt wird, ist das modulare Fuzzy-System mit Rückkopplung trotzdem mathematisch äquivalent mit einem rekurrenten Fuzzy-System mit den Zustandsgrößen v_l , v_r , Sp und Bl .

Der Charakter Ch ändert sich nicht und kann daher auch als Eingangsgröße betrachtet werden. Zusätzlich wird die Position Pos nicht berücksichtigt, da sie außerhalb des Entscheidungsmoduls bestimmt wird.

Um die gewohnte Schreibweise nutzen zu können, setze man $x_1 = v_l$, $x_2 = v_r$, $x_3 = Sp$, $x_4 = Bl$, $u_1 = \Delta_{VMI}$, $u_2 = v_{VMI}$, $u_3 = \Delta_{VMr}$, $u_4 = v_{VMr}$, $u_5 = Ch$, $\hat{x}_5 = v$, $\hat{x}_6 = v_l^*$, $\hat{x}_7 = v_r^*$, $\hat{x}_8 = Sww$, $\hat{x}_9 = Swe$. Die Zwischenzustände $\hat{x}_5$ bis $\hat{x}_9$ sind durch ein Dach über dem x gekennzeichnet.

Zuerst wird die Äquivalenz anhand der Regeln plausibel gemacht. Beispielhaft werden dazu die hintereinanderfolgenden Berechnungen der Zwischenwerte $v(k) = \hat{x}_5(k)$ und $v_l^*(k) = \hat{x}_6(k)$ und die anschließende Berechnung von $v_l(k+1) = x_1(k+1)$ betrachtet.

Die Zwischengröße $v(k) = \hat{x}_5(k)$ hängt nur von $v_l(k) = x_1(k)$, $v_r(k) = x_2(k)$ und $Sp(k) = x_3(k)$ ab. Daher werden folgende Regeln ausgewertet:

$$\text{Wenn } x_1(k) = L_{j_1}^{x_1} \text{ und } x_2(k) = L_{j_2}^{x_2} \text{ und } x_3(k) = L_{j_3}^{x_3}, \text{ dann } \hat{x}_5(k) = L_{j_5(j_1, j_2, j_3)}^{\hat{x}_5}. \quad (9.1)$$

Analog erhält man für die Berechnung von $v_l^*(k) = \hat{x}_6(k)$ in Abhängigkeit von $\Delta_{VMI}(k) = u_1(k)$, $v_{VMI}(k) = u_2(k)$ und $Ch = u_5(k)$ die folgenden Regeln:

$$\text{Wenn } u_1(k) = L_{q_1}^{u_1} \text{ und } u_2(k) = L_{q_2}^{u_2} \text{ und } u_5(k) = L_{q_5}^{u_5}, \text{ dann } \hat{x}_6(k) = L_{j_6(q_1, q_2, q_5)}^{\hat{x}_6}. \quad (9.2)$$

Da $v_l(k+1) = x_1(k+1)$ nur von $v(k) = \hat{x}_5(k)$ und $v_l^* = \hat{x}_6$ abhängt, erhält man:

$$\text{Wenn } \hat{x}_5(k) = L_{j_5}^{\hat{x}_5} \text{ und } \hat{x}_6(k) = L_{j_6}^{\hat{x}_6}, \text{ dann } x_1(k+1) = L_{w_1(j_5, j_6)}^{\hat{x}_1}. \quad (9.3)$$

Setzt man die ersten beiden Regelarten in die letzte ein, so erhält man Regeln der Form:

$$\begin{aligned} \text{Wenn } & x_1(k) = L_{j_1}^{x_1} \text{ und } x_2(k) = L_{j_2}^{x_2} \text{ und } x_3(k) = L_{j_3}^{x_3} \\ \text{und } & u_1(k) = L_{q_1}^{u_1} \text{ und } u_2(k) = L_{q_2}^{u_2} \text{ und } u_5(k) = L_{q_5}^{u_5}, \\ \text{dann } & x_1(k+1) = L_{w_1(j_5(j_1, j_2, j_3), j_6(q_1, q_2, q_5))}^{\hat{x}_1}. \end{aligned} \quad (9.4)$$

Diese Ersetzung der Zwischenwerte kann man auch auf der mathematischen Ebene durchführen. Dazu bestimmt man zuerst die Wahrheitswerte der Regeln des ersten Regelsatzes. Man erhält durch die Aggregation wie im normalen rekurrenten Fuzzy-System den Wahrheitswert

$$\mu_{j_1}^{x_1}(x_1) \cdot \mu_{j_2}^{x_2}(x_2) \cdot \mu_{j_3}^{x_3}(x_3) \quad (9.5)$$

für die Prämisse einer speziellen Regel. Durch die Addition der Wahrheitswerte aller Regeln, die den l -ten linguistischen Wert $L_l^{\hat{x}_5}$ in der Konklusion haben, erhält man den Wahr-

heitswert $\mu_l^{\hat{x}_5}$ dieses linguistischen Wertes für $\hat{x}_5 = v(k)$. Es folgt

$$\mu_l^{\hat{x}_5} = \sum_{j_5(j_1, j_2, j_3)=l} \mu_{j_1}^{x_1}(x_1) \cdot \mu_{j_2}^{x_2}(x_2) \cdot \mu_{j_3}^{x_3}(x_3). \quad (9.6)$$

Für den Wahrheitswert $\mu_h^{\hat{x}_6}$ des h -ten linguistischen Wertes für $\hat{x}_6 = v_l^*(k)$ ergibt sich analog

$$\mu_h^{\hat{x}_6} = \sum_{j_6(q_1, q_2, q_5)=h} \mu_{q_1}^{u_1}(u_1) \cdot \mu_{q_2}^{u_2}(u_2) \cdot \mu_{q_5}^{u_5}(u_5). \quad (9.7)$$

Genauso kann man den dritten Regelsatz für die Berechnung des Wahrheitswertes $\mu_r^{x_1}$ des r -ten linguistischen Wertes für $v_l(k+1) = x_1(k+1)$ heranziehen und man erhält

$$\mu_r^{x_1} = \sum_{w_1(l, h)=r} \mu_l^{\hat{x}_5} \cdot \mu_h^{\hat{x}_6}. \quad (9.8)$$

Setzt man die Gleichungen (9.6) und (9.7) in Gleichung (9.8) ein, so ergibt sich

$$\mu_r^{x_1} = \sum_{w_1(j_5(j_1, j_2, j_3), j_6(q_1, q_2, q_5))=r} (\mu_{j_1}^{x_1}(x_1) \cdot \mu_{j_2}^{x_2}(x_2) \cdot \mu_{j_3}^{x_3}(x_3)) \cdot (\mu_{q_1}^{u_1}(u_1) \cdot \mu_{q_2}^{u_2}(u_2) \cdot \mu_{q_5}^{u_5}(u_5)). \quad (9.9)$$

Dies ist genau die Summe aller Wahrheitswerte, die sich aus den zusammengesetzten Regeln in der Form von 9.4 für die Konklusion "dann $v(k+1) = L_r^{x_1}$ " ergeben. Durch das Ersetzen der Zwischenwerte auf der sprachlichen und auf der mathematischen Ebene bleibt die Beziehung zwischen den Regeln und den Gleichungen bestehen.

Bei der Defuzzifizierung der Größe $x_1(k+1) = v(k+1)$ werden die einzelnen Wahrheitswerte $\mu_r^{x_1}$ mit den Singletonpositionen $s_r^{x_1}$ gewichtet und addiert. So erhält man den neuen Zahlenwert von

$$\begin{aligned} x_1(k+1) &= \sum_r s_r^{x_1} \sum_{w_1(j_6(j_1, j_2, j_3), j_7(q_1, q_2, q_5))=r} \mu_{j_1}^{x_1}(x_1) \cdot \mu_{j_2}^{x_2}(x_2) \cdot \mu_{j_3}^{x_3}(x_3) \cdot \mu_{q_1}^{u_1}(u_1) \cdot \mu_{q_2}^{u_2}(u_2) = \\ &= \sum_{\check{w}_1(j_1, j_2, j_3, q_1, q_2, q_5)} s_{w_1}^{x_1} \cdot \mu_{j_1}^{x_1}(x_1) \cdot \mu_{j_2}^{x_2}(x_2) \cdot \mu_{j_3}^{x_3}(x_3) \cdot \mu_{q_1}^{u_1}(u_1) \cdot \mu_{q_2}^{u_2}(u_2), \end{aligned} \quad (9.10)$$

wobei $\check{w}_1(j_1, j_2, j_3, q_1, q_2, q_5) = w_1(j_6(j_1, j_2, j_3), j_7(q_1, q_2, q_5))$ gesetzt wurde. Die Zuordnung $\check{w}_1(j_1, j_2, j_3, q_1, q_2, q_5)$ der Indizes $j_1, j_2, j_3, q_1, q_2, q_5$ zu den Indizes $\check{w}_1$ entspricht der Zuordnung der linguistischen Werte durch die Regelbasis (9.4).

Gleichung (9.10) hat die gleiche Form wie die erste Komponente der Transitionsfunktion eines rekurrenten Fuzzy-Systems. Analog lassen sich auch für die anderen Zustandsgrößen solche Gleichungen aufstellen. Damit steht fest, dass das Modell die mathematische Struktur eines rekurrenten Fuzzy-Systems hat.

Im praktischen Einsatz wird man eine solche Transformation in ein rekurrentes Fuzzy-System in dieser Form vermeiden. Dies hat mehrere Gründe. Der triftigste Grund ist der exponentielle Anstieg der Anzahl der Regeln, der mit der Transformation einhergeht. In den Regelsätzen (9.1), (9.2) und (9.3) stehen insgesamt $4 \cdot 4 \cdot 2 + 9 \cdot 4 \cdot 3 + 4 \cdot 4 = 156$ Regeln. Im Regelsatz (9.4) stehen dagegen $4 \cdot 4 \cdot 2 \cdot 9 \cdot 4 \cdot 3 = 3456$ Regeln. Mit der Zahl

der Regeln steigt auch der Speicherbedarf und die benötigte Rechenzeit an.

Des Weiteren werden durch die Transformation die Zwischengrößen eliminiert und die Anzahl der Größen in der Prämisse der betrachteten Regelsätze nimmt zu. Dies erschwert die inhaltliche Überprüfung der sprachlichen Regelbasis. Die kleinen Teilsysteme, die mit bedeutungsvollen Zwischengrößen arbeiten, sind dagegen wesentlich einfacher zu begreifen und lassen sich unabhängig voneinander überprüfen. Auch tauchen die Zwischengrößen im Entscheidungsvorgang des Fahrers auf. Also entspricht das Modell aus mehreren Teilsystemen eher dem Entscheidungsablauf, dem der modellierte Fahrer folgt.

Auf der anderen Seite ist eine Transformation für die Untersuchung der Dynamik des Gesamtsystems sinnvoll. Unter Umständen lassen sich auf das transformierte System einige der Sätze für rekurrente Fuzzy-Systeme anwenden. Im Folgenden werden zwei Fälle betrachtet.

Im ersten Fall wechselt der Fahrer auf die linke Spur und verbleibt dort. Dann kann man die Zustandsgröße der Spur $x_3 = Sp$ als Konstante bzw. als konstante Eingabegröße behandeln. Damit hängt die Zwischengröße $v(k) = \hat{x}_5(k)$ und auch die Zustandsgröße $v_l(k+1) = x_1(k+1)$ aufgrund der speziellen Form der Regelbasis nicht von $x_2(k) = v_r(k)$ ab. Also ist $v_l(k) = x_1(k)$ die einzige relevante dynamische Größe und Gleichung (9.10) beschreibt ein eindimensionales rekurrentes Fuzzy-System.

Die Zwischengröße $v(k)$ ist monoton steigend mit $v_l(k)$ und $v_l(k+1)$ ist wiederum monoton steigend in $v(k)$. Also ist das rekurrente Fuzzy-System in dem betrachteten Fall monoton steigend. Aufgrund der Wahl der Zugehörigkeitsfunktionen $\mu_{j1}^{x_1}(x_1)$ ist es außerdem standardisiert. Da die Regelbasen (9.1) (9.2) und (9.3) jeweils regelstetig sind, kann man sich vergewissern, dass auch die zusammengesetzte Regelbasis regelstetig ist. Weil das System regelstetig und eindimensional ist, ist es nach Satz 1 auch automatenähnlich. Da es monoton steigend ist, konvergiert es nach Satz 11 und hat daher keine Zyklen und muss daher auch terminiert sein.

Im zweiten Fall wechselt der Fahrer auf die rechte Spur und verbleibt dort. Eine analoge Analyse des Modells führt auf ein eindimensionales rekurrentes Fuzzy-System mit der Zustandsvariablen v_r bei ansonsten gleichen Eigenschaften.

In beiden Fällen lässt sich die Dynamik der relevanten Zustandsgröße also durch ein eindimensionales, regelstetiges, automatenähnliches, monoton steigendes, terminiertes standardisiertes rekurrentes Fuzzy-System beschreiben. Nahezu alle Sätze und Lemmata der Arbeit lassen sich anwenden. Damit ist in diesem Fall Chaos ausgeschlossen, die Automatenähnlichkeit nachgewiesen und bei konstanten Randbedingungen die Konvergenz des Zustandes gesichert.

Nach der theoretischen Betrachtung und Analyse des Fahrermodells werden im nächsten Kapitel noch Ergebnisse der Simulation gezeigt.

9.3 Simulationsergebnisse

Da das rekurrente Fuzzy-System ein nichtlineares System ist, können die Simulationen zum Teil sehr unterschiedlich auf veränderte Startbedingungen reagieren. Als Gesamtsystem ist

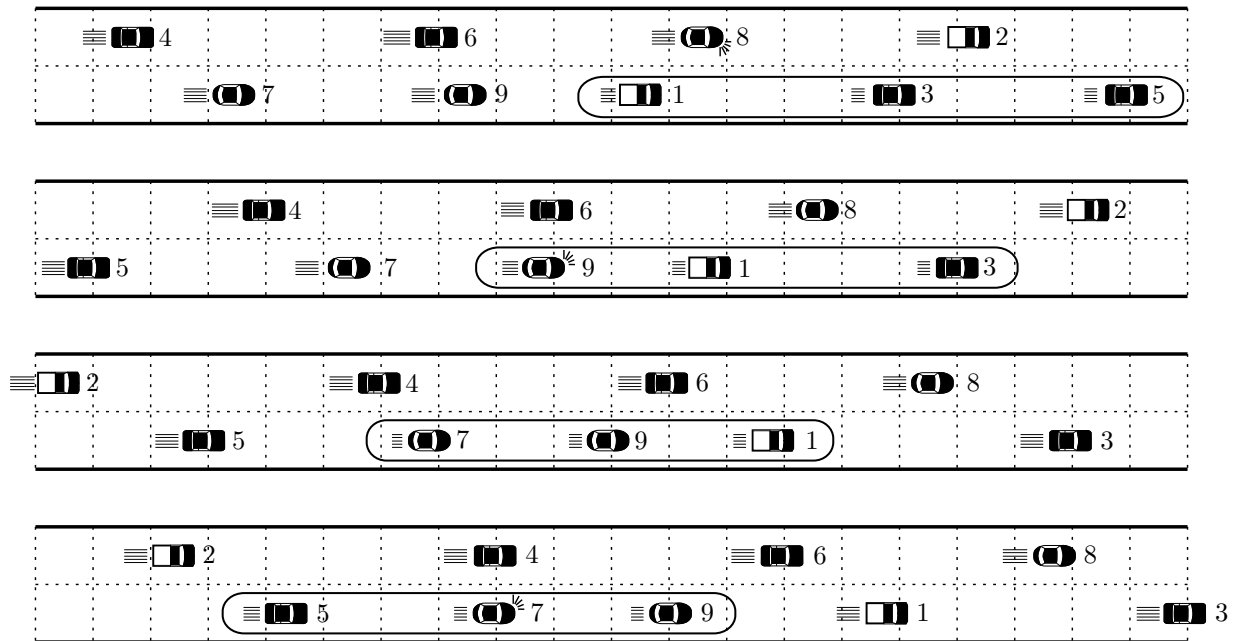


Abbildung 9.10: Vier Simulationsschritte, die das Auftreten einer lokalen Reduktion der Fahrgeschwindigkeit aufgrund einer hohen Verkehrsdichte belegen.

das Modell allerdings deterministisch und nicht von äußeren Einflussgrößen abhängig.

Im Folgenden werden zwei Effekte dargestellt, die in verschiedenen Konstellationen in mehreren Simulationen auftreten. Zuerst wird ein Straßenabschnitt betrachtet, auf dem ein verstärktes Verkehrsaufkommen eine lokale Reduzierung der Fahrgeschwindigkeit erzeugt. Dann wird das Modell für das Umfahren eines stehenden Hindernisses getestet.

Im ersten Beispiel wird eine Ringstraße mit einer Länge von 20 Zellen gewählt. Auf der Straße befinden sich Autofahrer mit unterschiedlichen Fahrcharakteren. Liegt nach einer Rundung der Wert des Fahrcharakters bei eins, so wird das Fahrzeug als LKW (□□) dargestellt, bei einem Ergebnis von zwei als PKW (□■) und bei einem Ergebnis von drei als Sportwagen (■■). Zur Identifizierung wird jedes Fahrzeug mit einer Nummer versehen. Die Straße ist für vier aufeinanderfolgende Simulationsschritte in Abbildung 9.10 zu sehen. Die Autos fahren von links nach rechts. Die vier Linien hinter jedem Fahrzeug deuten den Fahrtwind an. Ihre Länge ist proportional zur jeweiligen Geschwindigkeit. In diesem Beispiel liegen die Geschwindigkeiten zwischen einer und zwei Zellen pro Zeitschritt. Zu beachten ist, dass die Fahrzeuge nicht wie beim zellulären Automaten die angedeuteten Zellen besetzen müssen, sondern auch dazwischen liegen können. Ragt ein Auto über das Ende der Straße hinaus, so liegt es zwischen der zwanzigsten und der ersten Zelle.

Auf der linken Spur befinden sich nur vier Fahrzeuge. Diese haben ausreichend Platz und fahren nahezu konstant mit einer Geschwindigkeit von circa zwei Zellen pro Zeitschritt, die maßgeblich durch die Höchstgeschwindigkeit des einen LKW auf der Spur begrenzt wird. Auf der rechten Spur sind fünf Fahrzeuge. Für sie reicht der Abstand nicht aus, um stets mit einer Geschwindigkeit von circa zwei Zellen pro Zeitschritt fahren zu können. Es gibt immer eine Zone, in der sie nur mit einer geringeren Geschwindigkeit vorankommen. Diese Zone

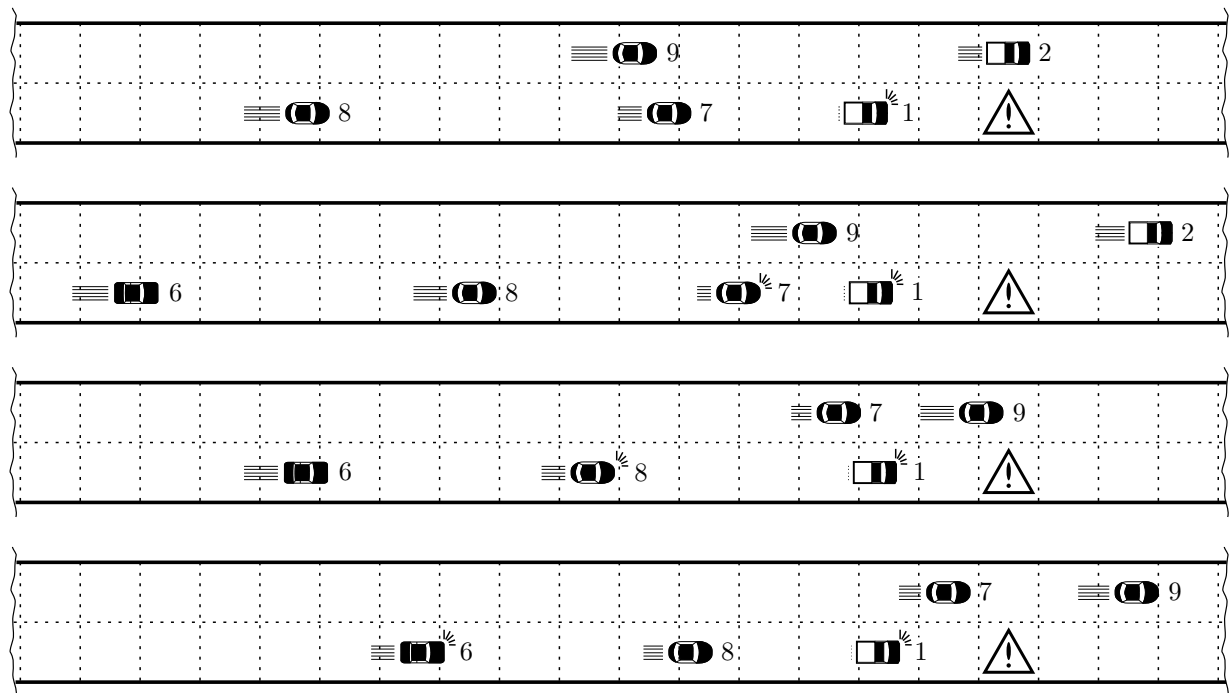


Abbildung 9.11: Vier Simulationsschritte, die sich beim Umfahren eines Hindernisses ergeben.

ist durch die abgerundeten Kästen in Abbildung 9.10 angedeutet. Nähert sich ein Auto der Zone von hinten, so muss es leicht abbremesen. Ein Auto am anderen Ende der Zone kann dagegen wieder schneller fahren. Damit wandert die Zone reduzierter Geschwindigkeit ähnlich wie eine Stauwelle gegen die Fahrtrichtung nach hinten. Bei Simulationen auf längeren Fahrbahnabschnitten können auch mehrere solcher Zonen auftreten.

Aufgrund der hohen Verkehrsdichte kommt es im gezeigten Beispiel zu keinem Spurwechsel. Zum Beispiel blinkt im zweiten Zeitschritt Fahrer 9, weil er aufgrund des vorausfahrenden LKWs zum Zeitpunkt davor auf der rechten Spur nicht so schnell fahren konnte, wie er auf der linken Spur hätte fahren können. Allerdings wird der Fahrer links von einem PKW überholt, so dass ein Wechsel auf die linke Spur nicht möglich ist. Da er im weiteren Verlauf rechts schneller fahren kann als links, stellt der Fahrer den Blinker wieder ab. Erst ein Spurwechsel würde das System aus seinem quasistationären Zustand bringen. Dies wird im zweiten Beispiel erreicht.

Im zweiten Beispiel wird an einer bestimmten Stelle ein Hindernis auf der rechten Spur eingebracht. Es ist durch ein Warndreieck dargestellt. Das Hindernis wird im Fuzzy-Modell von den anderen Verkehrsteilnehmern wie ein Auto mit einer Geschwindigkeit von null behandelt. Um das Hindernis zu umfahren, muss ein Auto auf der rechten Spur nach links wechseln. Damit treten in diesem Beispiel viele Spurwechselvorgänge auf.

Abbildung 9.11 zeigt einen Ausschnitt von 20 Zellen aus einer 50 Zellen langen Ringstraße. Auf der Straße befinden sich neben dem Hindernis neun Fahrzeuge, die allerdings nicht alle in dem gezeigten Ausschnitt liegen. Die Geschwindigkeiten der Fahrer liegen zwischen

drei bei Fahrer 9 und nahe null bei LKW 1. Der LKW kommt vor dem Hindernis fast zum Stehen. Er würde gerne die Spur wechseln, da er links schneller voran kommen könnte, und blinkt deshalb. Allerdings kann er die Spur nicht wechseln, da er in jedem Zeitschritt auf einen Hintermann auf der linken Spur Rücksicht nehmen muss.

Fahrzeug 7 schafft es, elegant an dem Hindernis vorbeizufahren. Im ersten Zeitschritt entscheidet der Fahrer zu blinken, weil er links schneller vorankommen könnte als rechts. Im zweiten Zeitschritt will er immer noch die Spur wechseln. Er hat zwar zu diesem Zeitpunkt einen geringeren Abstand zu seinen Vordermann auf der linken Spur (Fahrer 9) als zu den Vordermann auf der rechten Spur (Fahrer 1), kann aber trotzdem links schneller fahren, weil Fahrer 9 wesentlich schneller ist als Fahrer 1. Da Fahrer 7 zusätzlich schon den Blinker gesetzt hat und ihm kein Hintermann im Weg steht, kann er die Spur wechseln und so am Hindernis vorbeifahren.

Die beiden Beispiele demonstrieren, dass das Fahrermodell, welches ausschließlich auf der sprachlichen Beschreibung des Entscheidungsverhaltens der Fahrer aufbaut, eine realistische Simulation eines Verkehrsflusses erlaubt. Sie zeigen, dass sich mit rekurrenten Fuzzy-Systemen auch komplexere Zusammenhänge, welche durch sprachliche Regeln charakterisiert werden können, sinnvoll und erfolgreich in ein mathematisches Modell umsetzen lassen.

10 Zusammenfassung

Rekurrente Fuzzy-Systeme entstehen durch die Umsetzung einer Regelbasis, die einen zeitdiskreten Vorgang beschreibt, in ein Fuzzy-System. Die Zeitabhängigkeit in dem Modell wird durch die Rückführung früherer Ergebnisse des Fuzzy-Systems an den Ausgang beschrieben. Rekurrente Fuzzy-Systeme werden bereits in industriellen Anlagen eingesetzt. Ihre dynamischen Eigenschaften wurden dagegen noch nicht umfassend untersucht. Diese Lücke will diese Arbeit zum Teil schließen. Gleichzeitig legt sie den Grundstein für eine Theorie rekurrenter Fuzzy-Systeme.

Rekurrente Fuzzy-Systeme reihen sich in die Menge der zeitdiskreten Fuzzy-Systeme ein, deren weitere wichtige Vertreter Fuzzy-Automaten, dynamical fuzzy systems und dynamische Fuzzy-Systeme sind. Im Gegensatz zu den anderen Ansätzen wird bei rekurrenten Fuzzy-Systemen die Regelbasis durch ein vollständiges statisches Fuzzy-System, bestehend aus den Teilschritten Fuzzifizierung, Inferenz und Defuzzifizierung, in eine gewöhnliche Funktion überführt. Zusammen mit der Rückführung ergibt sich ein Differenzengleichungssystem.

Ziel der Arbeit war es, die dynamischen Eigenschaften eines rekurrenten Fuzzy-Systems in Abhängigkeit der Struktur der Regelbasis zu analysieren. Dies soll eine Bewertung der Dynamik eines solchen Systems auf der Ebene der sprachlich formulierten Regeln erlauben, ohne die meist detailreiche Ausführung des unterliegenden mathematischen Systems betrachten zu müssen.

Interpretiert man die Regelbasis als wage Beschreibung der Abbildung eines Fuzzy-Zustandes auf einen neuen Fuzzy-Zustand, so liegt die Vermutung nahe, dass die Regelbasis einen Automaten beschreibt und dass das darauf aufbauende rekurrente Fuzzy-System Ähnlichkeiten zu einem Automaten aufweist. Diese Idee wird formalisiert und es zeigt sich, dass sich rekurrente Fuzzy-Systeme nur unter bestimmten Umständen im gesamten Zustandsraum ähnlich wie Automaten verhalten.

Die Nähe zu Automaten erlaubt es jedoch, Methoden aus der Automatentheorie zu entlehnen. Insbesondere ist es möglich, die gesamte Regelbasis eines Fuzzy-Systems als Zustandsgraphen eines endlichen Automaten darzustellen. Anhand dieser Darstellung können sowohl die sprachliche Regelbasis als auch dynamische Eigenschaften rekurrenter Fuzzy-Systeme abgelesen werden. Daher kann diese Darstellungsform als Bindeglied zwischen der sprachlichen und der mathematischen Beschreibung eines rekurrenten Fuzzy-Systems genutzt werden.

Da sich ein rekurrentes Fuzzy-System nicht immer ähnlich wie ein Automat verhält, muss man es im Allgemeinen als nichtlineares zeitdiskretes System betrachten. Es zeigt sich, dass

schon einfache rekurrente Fuzzy-Systeme ein komplexes dynamisches Verhalten aufweisen können. Dies wird besonders bei der Behandlung von chaotischen Dynamiken deutlich. In der Arbeit werden hinreichende Kriterien hergeleitet, mit denen man chaotische Dynamiken nachweisen kann. Weitere hinreichende Kriterien werden für den Ausschluss von Chaos angegeben. Alle diese Kriterien lassen sich auch am Zustandsgraphen, der die Regelbasis beschreibt, graphisch auswerten.

Des Weiteren werden hinreichende Kriterien hergeleitet, die Aussagen über das Konvergenzverhalten und die Verteilung von Ruhelagen im Zustandsraum eines rekurrenten Fuzzy-Systems erlauben. Wieder dient der Zustandsgraph zur Darstellung und späteren Auswertung der Ergebnisse.

Auch Erreichbarkeit und Steuerbarkeit verschiedener Zustandsvektoren werden in rekurrenten Fuzzy-Systemen untersucht. Mit graphischen Methoden, die sich am Zustandsgraphen anlehnen, können Erreichbarkeitsmengen und Steuerbarkeitsmengen abgeschätzt und zum Teil exakt bestimmt werden.

Der Einsatz rekurrenter Fuzzy-Systeme wird anhand zweier Anwendungsbeispiele demonstriert. Im ersten Beispiel wird die Modifikation eines sequentiellen Mustererkenners vorgestellt, welcher im industriellen Einsatz betrieben wird. Das zweite Beispielsystem modelliert das Entscheidungsverhalten von Autofahrern auf Basis von sprachlich formulierten Entscheidungsregeln.

Mit den hier gezeigten Untersuchungen sind rekurrente Fuzzy-Systeme noch lange nicht vollständig in all ihren dynamischen Eigenschaften verstanden. Interessante Themenfelder wären zum Beispiel die Fragen nach der Beobachtbarkeit von Zuständen oder das Verhalten bei zeitlich variierenden, verrauschten oder unsicheren Eingangsgrößen.

Insbesondere konkrete Anwendungsfälle dürften die weiteren Untersuchungen von rekurrenten Fuzzy-Systemen stimulieren. Denn letztendlich bestimmen die Anwendungen, in welchen Bereichen weitere Kriterien notwendig sind, um testen zu können, ob sich die Vorstellung über das Verhalten eines sprachlichen Modells tatsächlich in dessen mathematischer Darstellung in Form eines rekurrenten Fuzzy-Systems wiederfindet.

A Sammlung spezieller rekurrenter Fuzzy-Systeme

In diesem Teil des Anhangs sind einige rekurrente Fuzzy-Systeme aufgeführt. Ihre Existenz und Analyse zeigen, dass sich bestimmte Sätze nicht ohne weiteres verallgemeinern lassen.

Beispiel 1

Es gibt (autonome) automatenähnliche (standardisierte) rekurrente Fuzzy-Systeme mit zwei Zustandsgrößen, die nicht regelstetig sind (vgl. Satz 1).

Der Zustandsgraph des betrachteten Beispiels ist in Abbildung A.1 links zu sehen, aus dem die Regelbasis abgelesen werden kann. Die Regelbasis ist nicht regelstetig, da die benachbarten Kernpositionsvektoren $(0, 0)$ und $(0, 1)$ auf nicht benachbarte Kernpositionen $(0, 0)$ und $(1, 1)$ abgebildet werden.

Bei dreiecksförmigen Zugehörigkeitsfunktionen liegt ein standardisiertes rekurrentes Fuzzy-System vor. Die Transitionsfunktion für ein solches rekurrentes Fuzzy-System mit der gezeigten Regelbasis ist durch $\mathbf{f}((x_1, x_2)) = (x_2, x_2)$ gegeben. Das Bild des Zustandsraumes $X = [0, 1] \times [0, 1]$ ist in Abbildung A.1 rechts gezeigt. Ein Vektor $\mathbf{x}$ mit $x_2 > 0$ ist mit dem Kernpositionsvektor $(0, 1)$ und/oder dem Kernpositionsvektor $(1, 1)$ verbunden. Des Weiteren ist sein Bild $\mathbf{f}(\mathbf{x})$ mit dem Bild der Kernpositionsvektoren $(0, 1)$ und $(1, 1)$, d.h. dem Kernpositionsvektor $(1, 1)$, verbunden. Also ist das rekurrente Fuzzy-System für alle $\mathbf{x}$ mit $x_2 > 0$ automatenähnlich. Aufgrund des symmetrischen Aufbaus der Regelbasis folgt analog die Automatenähnlichkeit für alle $\mathbf{x}$ mit $x_2 < 1$.

Das betrachtete System ist demnach automatenähnlich, aber nicht regelstetig.



Abbildung A.1: Zustandsgraph und Bild des Zustandsraums X eines automatenähnlichen, aber nicht regelstetigen rekurrenten Fuzzy-Systems.

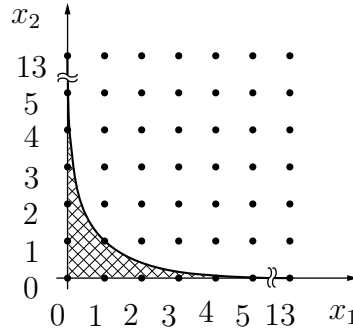


Abbildung A.2: Bild eines Hyperquaders eines regelstetigen, aber nicht automatenähnlichen rekurrenten Fuzzy-Systems.

Beispiel 2

Es gibt regelstetige rekurrente (standardisierte) Fuzzy-Systeme mit mehr als einer Zustandsgröße, die nicht automatenähnlich sind (vgl. Satz 1).

Das hier aufgeführte Beispiel hat zwei Zustandsgrößen und $n - 2$ Eingangsgrößen. Die Zahl n soll gerade sein und wird später genauer spezifiziert. Zusätzlich sind die Abstände der Kernpositionen in jeder Komponente des Zustandsvektors gleich.

Man betrachte einen n -dimensionalen elementaren Hyperquader $H_0 = X_0 \times U_0$ im Raum $X \times U$. Ein Eckpunkt des Hyperquaders H_0 soll auf $(\frac{n}{2}, 0) \in X$ abgebildet werden. Er wird mit $\mathbf{z}_0$ bezeichnet. Alle Eckpunkte $\mathbf{z}$ im Hyperquader H_0 mit einem Abstand $\|\mathbf{z}_0 - \mathbf{z}\| = v \leq \frac{n}{2}$ von $\mathbf{z}_0$ werden auf $(\frac{n}{2} - v, 0)$ abgebildet und alle mit einem Abstand $\|\mathbf{z}_0 - \mathbf{z}\| = v > \frac{n}{2}$ von $\mathbf{z}_0$ werden auf $(0, v - \frac{n}{2})$ abgebildet. Damit ist die Wirkung aller Regeln im Hyperquader H_0 festgelegt. Offensichtlich ist die Regelbasis regelstetig.

Sei $\mathbf{p}$ der Mittelpunkt des Hyperquaders H_0 . Bei geeigneter Wahl der Zugehörigkeitsfunktionen, z.B. bei der Wahl von Dreiecken, hat für diesen Punkt jede Regel im Hyperquader H_0 den Wahrheitswert $(\frac{1}{2})^n$. Für die Berechnung der ersten Komponente $f_1(\mathbf{p})$ des Bildvektors $\mathbf{f}(\mathbf{p})$ von $\mathbf{p}$ reicht es aus, alle Regeln von H_0 zu betrachten, die zu einem Wert ungleich null für diese Komponente führen, also alle Eckpunkte $\mathbf{z}$ von H_0 mit $\|\mathbf{z}_0 - \mathbf{z}\| = v \leq \frac{n}{2}$. Für jeden Abstand v gibt es genau $\frac{n!}{(n-v)!v!}$ solcher Punkte.

Damit ergibt sich:

$$f_1(\mathbf{p}) = \left(\frac{1}{2}\right)^n \cdot \sum_{v=0}^{\frac{n}{2}} \binom{n}{2-v} \frac{n!}{(n-v)!v!}. \quad (\text{A.1})$$

Aufgrund des symmetrischen Aufbaus der Regelbasis ist auch $f_1(\mathbf{p}) = f_2(\mathbf{p})$. Ab $n = 26$ ist $f_1(\mathbf{p}) = f_2(\mathbf{p}) > 1$. Abbildung A.2 skizziert das Bild des Hyperquaders H_0 für $n = 26$. Man sieht, dass es einen Vektor $\mathbf{x} \in X_0$ gibt, der auf den Kernpositionsvektor $(1, 1)$ abgebildet wird. Dieser Punkt ist ausschließlich mit Eckpunkten des Hyperquaders H_0 verbunden. Keiner der Funktionswerte der Eckpunkte des Hyperquaders H_0 ist mit dem Kernpositionsvektor $(1, 1)$ verbunden, welcher zugleich der Funktionswert von $\mathbf{x}$ ist. Daher ist ein solches System nicht automatenähnlich, obwohl die Regelbasis regelstetig ist.

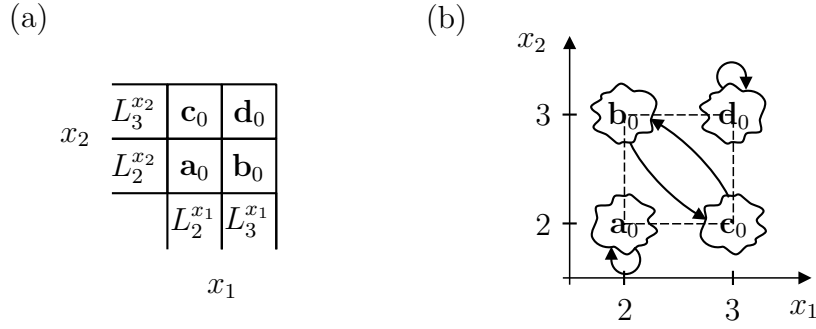


Abbildung A.3: (a) Regelbasis und (b) Zustandsgraph eines einfachen, nicht terminierten, regelstetigen rekurrenten Fuzzy-Systems.

Anmerkung: Es gibt auch regelstetige rekurrente Fuzzy-Systeme mit nicht-äquidistanten Kernpositionen, die nicht automatenähnlich sind und weniger als $n = 26$ Größen haben.

Beispiel 3

Es gibt terminierte, regelstetige standardisierte rekurrente Fuzzy-Systeme, die nicht für jeden Anfangszustand konvergieren (vgl. Satz 15).

Für die Konstruktion des Beispiels betrachte man ein erstes rekurrentes Fuzzy-System mit den linguistischen Vektoren $\mathbf{a}_0 = (L_2^{x_1}, L_2^{x_2})$, $\mathbf{b}_0 = (L_2^{x_1}, L_3^{x_2})$, $\mathbf{c}_0 = (L_3^{x_1}, L_2^{x_2})$ und $\mathbf{d}_0 = (L_3^{x_1}, L_3^{x_2})$ und mit der Regelbasis aus Abbildung A.3(a). Aus der regelstetigen Regelbasis lässt sich der Zustandsgraph aus Abbildung A.3(b) erstellen. Dem Zustandsgraph ist der Zustandsraum unterlegt, wobei die linguistischen Vektoren genau an den Stellen der entsprechenden Kernpositionsvektoren eingezeichnet sind. Da der Zustandsgraph einen linguistischen Zyklus der Länge zwei enthält, konvergiert das Fuzzy-System nicht für jeden Startwert.

Ausgehend von der Regelbasis dieses Systems wird die Regelbasis eines terminierten, regelstetigen standardisierten rekurrenten Fuzzy-Systems erzeugt, welches auf einem Teilraum des Zustandsraums die gleiche Dynamik wie das obige System hat und damit auch für manche Startwerte nicht konvergiert.

Die Regelbasis des zu konstruierenden rekurrenten Fuzzy-Systems entsteht durch eine Erweiterung der Regelbasis aus Abbildung A.3(a) auf mehr Zustandswerte pro Zustandsgröße, wie in Abbildung A.4(a) zu sehen ist. In dem Ausschnitt für linguistische Vektoren $(L_2^{x_1}, L_2^{x_2})$, $(L_2^{x_1}, L_3^{x_2})$, $(L_3^{x_1}, L_2^{x_2})$ und $(L_3^{x_1}, L_3^{x_2})$ stehen anstatt der linguistischen Zustandsvektoren $\mathbf{a}_0, \mathbf{b}_0, \mathbf{c}_0, \mathbf{d}_0$, die den Eckpunkten des ursprünglichen elementaren Hyperquaders $X_0 = [2, 3] \times [2, 3]$ entsprechen, neue linguistische Vektoren $\mathbf{a}_w, \mathbf{b}_w, \mathbf{c}_w, \mathbf{d}_w$. Die neuen linguistischen Vektoren entsprechen Eckpunkten von neuen Hyperquadrern X_w mit $w = 1, \dots, 8$, die in Abbildung A.4(b) zu sehen sind. Analog zur Abbildung A.3(b) gelten zum Beispiel für den Hyperquader X_1 die Beziehungen $\mathbf{a}_1 \rightarrow (1, 3)$, $\mathbf{b}_1 \rightarrow (1, 4)$, $\mathbf{c}_1 \rightarrow (2, 3)$, $\mathbf{d}_1 \rightarrow (2, 4)$. Die Wahl der neuen linguistischen Vektoren hat eine Verschiebung der Bildvektoren aus dem Hyperquader X_0 in den jeweiligen neuen Hyperquader X_w zur Folge.

Letztendlich wird über den Eingabevektor entschieden, welcher Index w gewählt wird und damit welche linguistischen Vektoren $\mathbf{a}_w, \mathbf{b}_w, \mathbf{c}_w, \mathbf{d}_w$ in der Schlussfolgerung einer Regel

(a)

$L_4^{x_2}$	c_w	d_w	d_w	d_w
$L_3^{x_2}$	a_w	c_w	d_w	d_w
$L_2^{x_2}$	a_w	a_w	b_w	d_w
$L_1^{x_2}$	a_w	a_w	a_w	b_w
	$L_1^{x_1}$	$L_2^{x_1}$	$L_3^{x_1}$	$L_4^{x_1}$

x_1

(b)

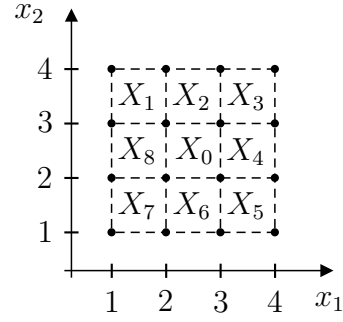


Abbildung A.4: (a) Regelbasis eines nicht konvergenten, terminierten, regelstetigen rekurrenten Fuzzy-Systems, dessen Einträge durch (b) die Lage elementarer Hyperquader X_w festgelegt sind.

stehen. Dementsprechend werden die linguistischen Eingabevektoren $\mathbf{L}_q^u$ weiter unten in acht Kategorien Q_w mit $w = 1, \dots, 8$ eingeteilt.

Anhand der Regelbasis lässt sich nachweisen, dass für jeden konstanten linguistischen Eingabevektor ein beliebiger linguistischer Zustandsvektor nach mehreren Zeitschritten auf einem Endknoten zu liegen kommt. Also ist das rekurrente Fuzzy-System terminiert. Solange benachbarte Eingabevektoren aus den gleichen oder benachbarten Kategorien Q_w stammen, ist die Regelbasis zusätzlich regelstetig. Bei der Verwendung von dreiecksförmigen Zugehörigkeitsfunktionen liegt dann ein terminiertes, regelstetiges standardisiertes rekurrentes Fuzzy-System vor.

In den folgenden Schritten wird eine geeignete Zuweisung der Eingangsvektoren zu den Kategorien Q_w konstruiert, die zu einer regelstetigen Regelbasis führt und zugleich gewährleistet, dass sich bei geeigneter Wahl eines Eingabevektors $\mathbf{u}$ die Verschiebungen von X_0 auf die jeweiligen Hyperquader X_w in ihrer gewichteten Summe gerade aufheben.

Für die Zuweisung der linguistischen Eingabevektoren $\mathbf{L}_q^u$ zu den Kategorien Q_w betrachtet man einen elementaren Hyperquader H im Eingangsraum U mit geeignet hoher Dimension n . Man wähle einen Eckpunkt $\mathbf{z}_0$ des Hyperquaders H in U . Anschließend fasst man alle linguistischen Eingabevektoren $\mathbf{L}_q^u$ dieses Hyperquaders H , deren Kernpositionsvektoren einen Abstand von $v = \|\mathbf{z}_0 - \mathbf{z}\|$ zu $\mathbf{z}_0$ haben, in Mengen D_v mit $v = 0, \dots, n$ zusammen. Jede Menge D_v enthält dann genau $|D_v| = \frac{n!}{(n-v)!v!}$ Elemente und benachbarte linguistische Eingabevektoren liegen in benachbarten Mengen D_v und D_{v+1} .

In der Reihenfolge $D_1, D_2, D_3, \dots$ weist man die Elemente aus D_v der Kategorie Q_1 zu, bis $\frac{1}{8}$ aller linguistischen Vektoren des Hyperquaders H , d.h. insgesamt $\frac{1}{8} \cdot 2^n$ linguistische Vektoren, der Kategorie Q_1 zugeordnet wurden. Die nächsten $\frac{1}{8} \cdot 2^n$ Elemente aus den Mengen D_v werden dann der Kategorie Q_2 zugeordnet, und so weiter.

Bei geeignet großer Dimension n kann man erreichen, dass beliebige Elemente aus zwei benachbarten Mengen D_v und D_{v+1} in der gleichen Kategorie Q_w oder in benachbarten Kategorien Q_w und Q_{w+1} liegen. Dies ist gewährleistet, wenn jede Menge Q_w mehr Ele-

mente enthält als die maximale Summe von benachbarten Mengen D_v und D_{v+1} . Dies ist wiederum sichergestellt, wenn die Zahl $|Q_w| = \frac{1}{8} \cdot 2^n$ der Elemente in jeder Kategorie Q_w mehr als doppelt so groß ist wie die maximale Zahl $\max_v |D_v|$ der Elemente der Mengen D_v , wenn also $\frac{1}{8} \cdot 2^n > 2 \max_v |D_v|$ ist. Für geradzahlig Dimensionen n ist $\max_v |D_v| = \frac{n!}{(n/2)!(n/2)!}$. Mit der Stirlingschen Näherungsformel $n! \approx (\frac{n}{e})^n \sqrt{2\pi n}$ ergibt sich für geradzahlig Dimensionen n eine geschätzte Schranke für die Dimension von $\frac{32^2}{2\pi} \approx 163$. Die Wahl von $n = 164$ sollte daher ausreichen. Ab diesen Dimensionen des Eingangsraumes liegen Elemente aus zwei benachbarten Mengen D_v und D_{v+1} in der gleichen Kategorie Q_w oder in benachbarten Kategorien Q_w und Q_{w+1} .

Benachbarte linguistische Eingabevektoren gehören also zu zwei benachbarten Mengen D_v und D_{v+1} und deshalb bei geeignet großer Dimension des Eingangsraumes entweder zur gleichen Kategorie Q_w oder zu benachbarten Kategorien Q_w und Q_{w+1} . Damit ist sichergestellt, dass die Regelbasis regelstetig ist. Eine solche Regelbasis wird im Weiteren verwendet.

Der Eingabevektor $\mathbf{u}$ liege nun auf dem Mittelpunkt des Hyperquaders H in U . Aufgrund der Wahl von dreiecksförmigen Zugehörigkeitsfunktionen wird jede Regel bezüglich des Eingabevektors $\mathbf{u}$ mit dem gleichen Wahrheitswert von $\prod_p \mu_{q_p}^{u_p}(\mathbf{u}) = (\frac{1}{2})^n$ gewichtet. Für Zustandsvektoren aus $X_0 = [2, 3] \times [2, 3]$ haben je $|Q_w| = \frac{1}{8} \cdot 2^n$ Regelsätze die gleiche Wirkung wie die Regelbasis aus Abbildung A.3 mit einer anschließenden Verschiebung des Hyperquaders X_0 in verschiedene Hyperquader X_w . Da alle geltenden Regelsätze bezüglich $\mathbf{u}$ gleich stark mit dem Wahrheitswert $(\frac{1}{2})^n$ gewichtet werden, heben sich die Verschiebungen in gegenüberliegende Hyperquader, zum Beispiel in X_1 und X_5 oder X_4 und X_8 , genau auf. Das rekurrente Fuzzy-System realisiert dann für diesen Eingabevektor $\mathbf{u}$ im Bereich $X_0 = [2, 3] \times [2, 3]$ die gleiche Abbildung wie die des nicht konvergenten rekurrenten Fuzzy-Systems, welches durch die Regelbasis aus Abbildung A.3 charakterisiert ist.

Damit ist die Konstruktion eines terminierten, regelstetigen standardisierten rekurrenten Fuzzy-Systems, welches nicht für alle Startwerte konvergiert, gelungen.

B Lemmata für Stabilitätssätze

In diesem Anhang sind einige Hilfssätze zu finden, die in den Beweisen in Kapitel 6 verwendet werden. Lemma 11 gibt mehrere implizite Gleichungen für die Transitionsfunktion an.

Lemma 11

Die Transitionsfunktion $\mathbf{f}$ eines rekurrenten Fuzzy-Systems ist durch die Funktionswerte $\mathbf{f}(\mathbf{s}_j^x, \mathbf{s}_q^u)$ an den Kernpositionsvektoren, die Zugehörigkeitsfunktionen $\mu_{ji}^{x_i}(x_i)$ und $\mu_{qp}^{u_p}(u_p)$ und durch folgende Gleichung bestimmt:

$$\mathbf{f}(\mathbf{x}, \mathbf{u}) = \sum_{\mathbf{j}, \mathbf{q}} \mathbf{f}(\mathbf{s}_j^x, \mathbf{s}_q^u) \prod_i \mu_{ji}^{x_i}(x_i) \prod_p \mu_{qp}^{u_p}(u_p). \quad (\text{B.1})$$

Die Transitionsfunktion $\mathbf{f}$ erfüllt außerdem die folgenden Gleichungen:

$$\mathbf{f}(\mathbf{x}, \mathbf{u}) = \sum_{\mathbf{j}} \mathbf{f}(\mathbf{s}_j^x, \mathbf{u}) \prod_i \mu_{ji}^{x_i}(x_i), \quad (\text{B.2})$$

$$\mathbf{f}(\mathbf{x}, \mathbf{u}) = \sum_{\mathbf{q}} \mathbf{f}(\mathbf{x}, \mathbf{s}_q^u) \prod_p \mu_{qp}^{u_p}(u_p). \quad (\text{B.3})$$

Beweis:

Setzt man $\mathbf{s}_{\mathbf{w}(\mathbf{j}, \mathbf{q})}^x = \mathbf{f}(\mathbf{s}_j^x, \mathbf{s}_q^u)$ aus Gleichung (4.2) in Gleichung (3.13) ein, so erhält man Gleichung (B.1). Speziell für $\mathbf{x} = \mathbf{s}_j^x$ liefert Gleichung (B.1):

$$\mathbf{f}(\mathbf{s}_j^x, \mathbf{u}) = \sum_{\mathbf{q}} \mathbf{f}(\mathbf{s}_j^x, \mathbf{s}_q^u) \prod_p \mu_{qp}^{u_p}(u_p). \quad (\text{B.4})$$

Aus Gleichung (B.2) erhält man durch eine Äquivalenzumformung, nämlich das Einsetzen von Gleichung (B.4), wiederum Gleichung (B.1). Damit gilt auch Gleichung (B.2). Analog lässt sich Gleichung (B.3) herleiten. $\square$

In Lemma 12 wird eine Sonderstellung der dreiecksförmigen Zugehörigkeitsfunktionen deutlich, die auch bei standardisierten rekurrenten Fuzzy-Systemen verwendet werden.

Lemma 12

Folgende Gleichung gilt für rekurrente Fuzzy-Systeme dann und nur dann, wenn Dreiecke

als Zugehörigkeitsfunktionen für den Zustand gewählt werden:

$$\mathbf{x} = \sum_{\mathbf{j}} \mathbf{s}_{\mathbf{j}}^{\mathbf{x}} \prod_i \mu_{j_i}^{x_i}(x_i) = \sum_{\mathbf{j}, \mathbf{q}} \mathbf{s}_{\mathbf{j}}^{\mathbf{x}} \prod_i \mu_{j_i}^{x_i}(x_i) \prod_p \mu_{q_p}^{u_p}(u_p). \quad (\text{B.5})$$

Beweis:

Für jeden Index q_p kann man den Term $\sum_{q_p} \mu_{q_p}^{u_p}(u_p)$ aus der linken Seite von Gleichung (B.5) ausklammern, da jedes Produkt $\mathbf{s}_{\mathbf{j}}^{\mathbf{x}} \prod_i \mu_{j_i}^{x_i}(x_i) \prod_p \mu_{q_p}^{u_p}(u_p)$ nur jeweils genau einen Faktor, nämlich $\mu_{q_p}^{u_p}(u_p)$, enthält, der von q_p abhängt, und dieser Faktor selbst von keinem anderen Index abhängt. Nach der Normierungsbedingung für rekurrente Fuzzy-Systeme nach Gleichung (3.12) ist für jeden Index q_p der Term $\sum_{q_p} \mu_{q_p}^{u_p}(u_p)$ gleich eins und kann daher weggelassen werden. Damit gilt generell das rechte Gleichheitszeichen in Gleichung (B.5).

Es gelte nun Gleichung (B.5). Im Folgenden wird gezeigt, dass die Zugehörigkeitsfunktionen $\mu_{j_i}^{x_i}$ in diesem Fall dreiecksförmig sein müssen. Man betrachte dazu eine beliebige Komponente x_h von $\mathbf{x}$. Aus den gleichen Überlegungen für die Indizes $j_i \neq j_h$ wie oben für die Indizes q_p folgt aus Gleichung (B.5):

$$x_h = \sum_{\mathbf{j}} s_{j_h}^{x_h} \prod_i \mu_{j_i}^{x_i}(x_i) = \sum_{j_h} s_{j_h}^{x_h} \mu_{j_h}^{x_h}(x_h). \quad (\text{B.6})$$

Der Punkt x_h liegt zwischen zwei Kernpositionen $s_r^{x_h}$ und $s_{r+1}^{x_h}$. Dann sind die Zugehörigkeitsfunktionen $\mu_{j_h}^{x_h}(x_h)$ maximal für die zwei Indizes $j_h = r$ und $j_h = r+1$ von null verschieden und es gilt nach Gleichung (3.12):

$$\mu_{r+1}^{x_h}(x_h) = 1 - \mu_r^{x_h}(x_h). \quad (\text{B.7})$$

Setzt man dies in Gleichung (B.6) ein, so erhält man:

$$x_h = \sum_{j_h} s_{j_h}^{x_h} \mu_{j_h}^{x_h}(x_h) = s_r^{x_h} \mu_r^{x_h}(x_h) + s_{r+1}^{x_h} \mu_{r+1}^{x_h}(x_h) = (s_r^{x_h} - s_{r+1}^{x_h}) \mu_r^{x_h}(x_h) + s_{r+1}^{x_h}. \quad (\text{B.8})$$

Löst man die Gleichung nach $\mu_r^{x_h}(x_h)$ auf, so ergibt sich als Zugehörigkeitsfunktion im Intervall $[s_r^{x_h}, s_{r+1}^{x_h}]$ eine Geradengleichung:

$$\mu_r^{x_h}(x_h) = \frac{s_{r+1}^{x_h} - x_h}{s_{r+1}^{x_h} - s_r^{x_h}}. \quad (\text{B.9})$$

Diese Geradengleichung beschreibt eine Flanke einer dreiecksförmigen Zugehörigkeitsfunktion. Die andere Flanke im Intervall $[s_{r-1}^{x_h}, s_r^{x_h}]$ erhält man aus:

$$\mu_r^{x_h}(x_h) = 1 - \mu_{r-1}^{x_h}(x_h) = 1 - \frac{s_r^{x_h} - x_h}{s_r^{x_h} - s_{r-1}^{x_h}}. \quad (\text{B.10})$$

Gleichung (B.5) impliziert also, dass die Zugehörigkeitsfunktionen $\mu_{j_i}^{x_i}$ dreiecksförmig sein müssen.

Hat man umgekehrt Dreiecke als Zugehörigkeitsfunktionen gewählt, so gilt Gleichung (B.9), die in Gleichung (B.6) überführt werden kann, die wiederum äquivalent zu Gleichung (B.5) ist. $\square$

Die linke Seite der Gleichung (B.5) stellt die Transitionsfunktion $\mathbf{f}$ der "verbalen identischen Abbildung" dar, wie durch Einsetzen von $\mathbf{f}(\mathbf{s}_j^x, \mathbf{s}_q^u) = \mathbf{s}_j^x$ in Gleichung (3.4) zu sehen ist. Lemma 12 zeigt, dass die "verbale identische Abbildung" nur bei dreiecksförmigen Zugehörigkeitsfunktionen zu einer identischen Abbildung, d.h. zu einer Transitionsfunktion $\mathbf{f}(\mathbf{x}, \mathbf{u}) = \mathbf{x}$, führt.

Die folgenden Lemmata 13, 14 und 15 gelten für standardisierte rekurrente Fuzzy-Systeme. In diesen Systemen kann ein elementarer Hyperquader eines Vektors $\mathbf{x}$ auch wie folgt charakterisiert werden: Man betrachte die Menge aller Indexvektoren $\mathbf{j}$, für die $\prod_i \mu_{j_i}^{x_i}(x_i) \neq 0$ ist. Die Kernpositionsvektoren $\mathbf{s}_j^x$ mit diesen Indexvektoren sind dann die Eckpunkte des elementaren Hyperquaders von $\mathbf{x}$. Es gilt:

Lemma 13

Erfüllen ein Zustandsvektor $\mathbf{x}$ und ein Kernpositionsvektor $\hat{\mathbf{s}} \in X$ in einem autonomen, regelstetigen standardisierten rekurrenten Fuzzy-System die Gleichung $\|\mathbf{f}(\mathbf{x}) - \mathbf{f}(\hat{\mathbf{s}})\| = \|\mathbf{x} - \hat{\mathbf{s}}\|$, so gilt

$$\|\mathbf{f}(\mathbf{s}_j^x) - \mathbf{f}(\hat{\mathbf{s}})\| = \|\mathbf{s}_j^x - \hat{\mathbf{s}}\| \quad (\text{B.11})$$

für alle Kernpositionsvektoren $\mathbf{s}_j^x$, die Eckpunkte des elementaren Hyperquaders von $\mathbf{x}$ sind.

Beweis:

Die Menge $A = \{\mathbf{j} \mid \prod_i \mu_{j_i}^{x_i}(x_i) \neq 0\}$ besteht aus den Indizes aller Eckpunkte des elementaren Hyperquaders von $\mathbf{x}$. Mit Lemma 11 ist

$$\|\mathbf{f}(\mathbf{x}) - \mathbf{f}(\hat{\mathbf{s}})\| = \left\| \sum_{\mathbf{j} \in A} \mathbf{f}(\mathbf{s}_j^x) \prod_i \mu_{j_i}^{x_i}(x_i) - \mathbf{f}(\hat{\mathbf{s}}) \right\|. \quad (\text{B.12})$$

Weiter gilt $\sum_{\mathbf{j} \in A} \prod_i \mu_{j_i}^{x_i}(x_i) = \prod_i \sum_{j_i \in A} \mu_{j_i}^{x_i}(x_i) = 1$ aufgrund der Normierung der Zugehörigkeitsfunktionen $\mathbf{f}$, und somit folgt:

$$\begin{aligned} \left\| \sum_{\mathbf{j} \in A} \mathbf{f}(\mathbf{s}_j^x) \prod_i \mu_{j_i}^{x_i}(x_i) - \mathbf{f}(\hat{\mathbf{s}}) \right\| &= \left\| \sum_{\mathbf{j} \in A} \mathbf{f}(\mathbf{s}_j^x) \prod_i \mu_{j_i}^{x_i}(x_i) - \mathbf{f}(\hat{\mathbf{s}}) \prod_i \mu_{j_i}^{x_i}(x_i) \right\| = \\ &= \left\| \sum_{\mathbf{j} \in A} (\mathbf{f}(\mathbf{s}_j^x) - \mathbf{f}(\hat{\mathbf{s}})) \prod_i \mu_{j_i}^{x_i}(x_i) \right\| \leq \sum_{\mathbf{j} \in A} \|\mathbf{f}(\mathbf{s}_j^x) - \mathbf{f}(\hat{\mathbf{s}})\| \prod_i \mu_{j_i}^{x_i}(x_i). \end{aligned} \quad (\text{B.13})$$

Aufgrund der Nichtexpansion von $\mathbf{f}$ nach Satz 13 folgt $\|\mathbf{f}(\mathbf{s}_j^x) - \mathbf{f}(\hat{\mathbf{s}})\| \leq \|\mathbf{s}_j^x - \hat{\mathbf{s}}\|$ und damit:

$$\begin{aligned} \sum_{\mathbf{j} \in A} \|\mathbf{f}(\mathbf{s}_j^x) - \mathbf{f}(\hat{\mathbf{s}})\| \prod_i \mu_{j_i}^{x_i}(x_i) &\leq \\ &\leq \sum_{\mathbf{j} \in A} \|\mathbf{s}_j^x - \hat{\mathbf{s}}\| \prod_i \mu_{j_i}^{x_i}(x_i) = \sum_{\mathbf{j} \in A} \sum_l \frac{1}{\Delta s_l} |s_{j_l}^{x_l} - \hat{s}_l| \prod_i \mu_{j_i}^{x_i}(x_i) \end{aligned} \quad (\text{B.14})$$

Da $\frac{1}{\Delta s_l}(s_{jl}^{x_l} - \hat{s}_l)$ ganzzahlig ist und sich für alle $\mathbf{j} \in A$ die Werte $\frac{1}{\Delta s_l}(s_{jl}^{x_l} - \hat{s}_l)$ maximal um 1 unterscheiden, folgt für jeden einzelnen Index l entweder $\frac{1}{\Delta s_l}(s_{jl}^{x_l} - \hat{s}_l) \leq 0$ für alle $\mathbf{j} \in A$ oder $\frac{1}{\Delta s_l}(s_{jl}^{x_l} - \hat{s}_l) \geq 0$ für alle $\mathbf{j} \in A$. Also gilt auch

$$\sum_{\mathbf{j} \in A} \sum_l \frac{1}{\Delta s_l} |s_{jl}^{x_l} - \hat{s}_l| \prod_i \mu_{j_i}^{x_i}(x_i) = \sum_l \frac{1}{\Delta s_l} \left| \sum_{\mathbf{j} \in A} (s_{jl}^{x_l} - \hat{s}_l) \prod_i \mu_{j_i}^{x_i}(x_i) \right|. \quad (\text{B.15})$$

Mit Lemma 12 folgt:

$$\begin{aligned} \sum_l \frac{1}{\Delta s_l} \left| \sum_{\mathbf{j} \in A} (s_{jl}^{x_l} - \hat{s}_l) \prod_i \mu_{j_i}^{x_i}(x_i) \right| &= \sum_l \frac{1}{\Delta s_l} \left| \sum_{\mathbf{j} \in A} s_{jl}^{x_l} \prod_i \mu_{j_i}^{x_i}(x_i) - \hat{s}_l \right| = \\ &= \left\| \sum_{\mathbf{j} \in A} \mathbf{s}_{\mathbf{j}}^{\mathbf{x}} \prod_i \mu_{j_i}^{x_i}(x_i) - \hat{\mathbf{s}} \right\| = \|\mathbf{x} - \hat{\mathbf{s}}\|. \end{aligned} \quad (\text{B.16})$$

Aufgrund der Gleichungen (B.12), (B.13), (B.14), (B.15), (B.16) gilt $\|\mathbf{f}(\mathbf{x}) - \mathbf{f}(\hat{\mathbf{s}})\| \leq \|\mathbf{x} - \hat{\mathbf{s}}\|$ und nach Voraussetzung gilt $\|\mathbf{f}(\mathbf{x}) - \mathbf{f}(\hat{\mathbf{s}})\| = \|\mathbf{x} - \hat{\mathbf{s}}\|$. Folglich sind alle Terme in den Gleichungen (B.12), (B.13), (B.14), (B.15), (B.16) gleich. Damit gilt insbesondere

$$\sum_{\mathbf{j} \in A} \|\mathbf{f}(\mathbf{s}_{\mathbf{j}}^{\mathbf{x}}) - \mathbf{f}(\hat{\mathbf{s}})\| \prod_i \mu_{j_i}^{x_i}(x_i) = \sum_{\mathbf{j} \in A} \|\mathbf{s}_{\mathbf{j}}^{\mathbf{x}} - \hat{\mathbf{s}}\| \prod_i \mu_{j_i}^{x_i}(x_i). \quad (\text{B.17})$$

Außerdem ist $\mathbf{f}$ nichtexpandierend, also gilt $\|\mathbf{f}(\mathbf{s}_{\mathbf{j}}^{\mathbf{x}}) - \mathbf{f}(\hat{\mathbf{s}})\| \leq \|\mathbf{s}_{\mathbf{j}}^{\mathbf{x}} - \hat{\mathbf{s}}\|$. Da $\prod_i \mu_{j_i}^{x_i}(x_i) \neq 0$ für alle $\mathbf{j} \in A$ ist, gilt

$$\|\mathbf{f}(\mathbf{s}_{\mathbf{j}}^{\mathbf{x}}) - \mathbf{f}(\hat{\mathbf{s}})\| = \|\mathbf{s}_{\mathbf{j}}^{\mathbf{x}} - \hat{\mathbf{s}}\| \quad (\text{B.18})$$

für alle $\mathbf{j} \in A$, weil andernfalls Gleichung (B.17) nicht erfüllt wäre. $\square$

Lemma 14

Erfüllen ein Zustandsvektor $\mathbf{x}$ und ein Kernpositionsvektor $\hat{\mathbf{s}} \in X$ in einem autonomen, regelstetigen standardisierten rekurrenten Fuzzy-System die Gleichung $\|\mathbf{f}(\mathbf{x}) - \mathbf{f}(\hat{\mathbf{s}})\| = \|\mathbf{x} - \hat{\mathbf{s}}\|$, so ist die Dimension des elementaren Hyperquaders von $\mathbf{f}(\mathbf{x})$ kleiner oder gleich der Dimension des elementaren Hyperquaders von $\mathbf{x}$. Wenn die Dimensionen gleich sind, werden die Eckpunkte des elementaren Hyperquaders von $\mathbf{x}$ auf Eckpunkte des elementaren Hyperquaders von $\mathbf{f}(\mathbf{x})$ abgebildet.

Beweis:

Die Menge $A = \{\mathbf{j} \mid \prod_i \mu_{j_i}^{x_i}(x_i) \neq 0\}$ besteht aus den Indizes aller Eckpunkte des elementaren Hyperquaders von $\mathbf{x}$. Die Menge $S = \{\mathbf{s}_{\mathbf{j}}^{\mathbf{x}} \mid \mathbf{j} \in A\}$ besteht aus den Eckpunkten des elementaren Hyperquaders von $\mathbf{x}$, dessen Dimension mit d bezeichnet wird.

Man betrachte die beiden Kernpositionsvektoren $\mathbf{M}$ und $\mathbf{m}$ aus S mit $\|\mathbf{M} - \hat{\mathbf{s}}\| = \max_{\mathbf{j} \in A} \|\mathbf{s}_{\mathbf{j}}^{\mathbf{x}} - \hat{\mathbf{s}}\|$ und $\|\mathbf{m} - \hat{\mathbf{s}}\| = \min_{\mathbf{j} \in A} \|\mathbf{s}_{\mathbf{j}}^{\mathbf{x}} - \hat{\mathbf{s}}\|$. $\mathbf{M}$ und $\mathbf{m}$ sind zwei sich diagonal gegenüberliegende Eckpunkte im d -dimensionalen Hyperquader, weil $\mathbf{m}$ in jeder Komponente die Kernposition m_i annimmt, die möglichst nahe an der von $\hat{\mathbf{s}}$ liegt und $\mathbf{M}$ in jeder Komponente die Kernposition M_i annimmt, die möglichst weit von der von $\hat{\mathbf{s}}$ entfernt liegt.

Nun wählt man einen beliebigen Eckpunkt $\mathbf{z} \in S$ des Hyperquaders. Ersetzt man, ausgehend von $\mathbf{m}$, nacheinander die Komponenten von $\mathbf{m}$ durch die von $\mathbf{M}$, in denen sich $\mathbf{m}$ und

$\mathbf{M}$ unterscheiden, so hat man insgesamt d Ersetzungen durchgeführt. Diese Ersetzungen kann man so vornehmen, dass man nach einer dieser d Ersetzungen den Vektor $\mathbf{z}$ erhält. Die Zahl der Abweichungen in den Komponenten von $\mathbf{z}$ zu denen von $\mathbf{m}$ und $\mathbf{M}$ ist in Summe also genau gleich d . Dies gilt auch für die Abstände zwischen $\mathbf{z}$ und $\mathbf{m}$ und $\mathbf{z}$ und $\mathbf{M}$, da sich deren einzelnen Komponenten maximal um 1 unterscheiden können. Also folgt zusammen mit der Nichtexpansivität von $\mathbf{f}$ und der Dreiecksungleichung:

$$d = \|\mathbf{M} - \mathbf{z}\| + \|\mathbf{z} - \mathbf{m}\| \geq \|\mathbf{f}(\mathbf{M}) - \mathbf{f}(\mathbf{z})\| + \|\mathbf{f}(\mathbf{z}) - \mathbf{f}(\mathbf{m})\| \geq \|\mathbf{f}(\mathbf{M}) - \mathbf{f}(\mathbf{m})\|. \quad (\text{B.19})$$

Wiederum mit der Dreiecksungleichung und Lemma 13 folgt:

$$\|\mathbf{f}(\mathbf{M}) - \mathbf{f}(\mathbf{m})\| \geq \|\mathbf{f}(\mathbf{M}) - \mathbf{f}(\hat{\mathbf{s}})\| - \|\mathbf{f}(\mathbf{m}) - \mathbf{f}(\hat{\mathbf{s}})\| = \|\mathbf{M} - \hat{\mathbf{s}}\| - \|\mathbf{m} - \hat{\mathbf{s}}\|. \quad (\text{B.20})$$

Da $\mathbf{M}$ und $\mathbf{m}$ gegenüberliegende Eckpunkte im d -dimensionalen Hyperquader sind, folgt:

$$\|\mathbf{M} - \hat{\mathbf{s}}\| - \|\mathbf{m} - \hat{\mathbf{s}}\| = \max_{\mathbf{j} \in A} \|\mathbf{s}_j^x - \hat{\mathbf{s}}\| - \min_{\mathbf{j} \in A} \|\mathbf{s}_j^x - \hat{\mathbf{s}}\| = d. \quad (\text{B.21})$$

Damit sind alle Terme in der obigen Ungleichungskette (B.19), (B.20), (B.21) gleich und es gilt insbesondere:

$$\|\mathbf{f}(\mathbf{M}) - \mathbf{f}(\mathbf{z})\| + \|\mathbf{f}(\mathbf{z}) - \mathbf{f}(\mathbf{m})\| = \|\mathbf{f}(\mathbf{M}) - \mathbf{f}(\mathbf{m})\| = d. \quad (\text{B.22})$$

Schreibt man in Gleichung (B.22) die Definition der Norm aus, so erhält man:

$$\sum_l \frac{1}{\Delta s_l} |f_l(\mathbf{M}) - f_l(\mathbf{z})| + \sum_l \frac{1}{\Delta s_l} |f_l(\mathbf{z}) - f_l(\mathbf{m})| = \sum_l \frac{1}{\Delta s_l} |f_l(\mathbf{M}) - f_l(\mathbf{m})|. \quad (\text{B.23})$$

Nach der Dreiecksungleichung gilt sowieso

$$|f_l(\mathbf{M}) - f_l(\mathbf{z})| + |f_l(\mathbf{z}) - f_l(\mathbf{m})| \geq |f_l(\mathbf{M}) - f_l(\mathbf{m})|. \quad (\text{B.24})$$

Damit auch die Gleichheit in Gleichung (B.23) erfüllt werden kann, muss auch Folgendes für jeden Index l gelten:

$$|f_l(\mathbf{M}) - f_l(\mathbf{z})| + |f_l(\mathbf{z}) - f_l(\mathbf{m})| = |f_l(\mathbf{M}) - f_l(\mathbf{m})|. \quad (\text{B.25})$$

Daraus ergibt sich wiederum für alle Indizes l :

$$f_l(\mathbf{m}) \leq f_l(\mathbf{z}) \leq f_l(\mathbf{M}) \text{ oder } f_l(\mathbf{m}) \geq f_l(\mathbf{z}) \geq f_l(\mathbf{M}). \quad (\text{B.26})$$

Die Vektoren $\mathbf{f}(\mathbf{z})$ unterscheiden sich also nur in den Komponenten, in denen sich auch $\mathbf{M}$ und $\mathbf{m}$ unterscheiden, also maximal in d Komponenten. Da $\mathbf{f}(\mathbf{x})$ aus einer Konvexkombination, d.h. einer Linearkombination mit Vorfaktoren, die zwischen 0 und 1 liegen und sich in Summe zu 1 ergänzen, dieser Vektoren $\mathbf{f}(\mathbf{z})$ berechnet wird, ist der elementare Hyperquader von $\mathbf{f}(\mathbf{x})$ maximal d -dimensional. Damit ist der erste Teil gezeigt.

Wenn der elementare Hyperquader des Bildpunktes $\mathbf{f}(\mathbf{x})$ wieder d -dimensional ist, so müssen sich $\mathbf{f}(\mathbf{M})$ und $\mathbf{f}(\mathbf{m})$ auf der einen Seite in mindestens d Komponenten unterscheiden. Auf der anderen Seite können sie sich nur in maximal d Komponenten unterscheiden,

weil nach Gleichung (B.22) $\|\mathbf{f}(\mathbf{M}) - \mathbf{f}(\mathbf{m})\| = d$ gilt. Aus diesen beiden Überlegungen folgt, dass sie sich genau in d Komponenten unterscheiden und zwar um jeweils genau eine Einheit, um die Beschränkung des Abstandes nach Gleichung (B.22) zu erfüllen. Sie bilden wiederum zwei gegenüberliegende Eckpunkte eines d -dimensionalen elementaren Hyperquaders H .

Nach Gleichung (B.26) sind auch alle Vektoren $\mathbf{f}(\mathbf{z})$ für $\mathbf{z} \in S$ Eckpunkte des Hyperquaders H . Da $\mathbf{f}(\mathbf{x})$ eine Konvexkombination aus diesen Vektoren darstellt und nach Voraussetzung in einem d -dimensionalen elementaren Hyperquader liegt, liegt er auch in H . Damit ist der zweite Teil gezeigt. $\square$

Lemma 15

Jeder Häufungspunkt $\mathbf{x}$ einer Trajektorie im Zustandsraum X eines autonomen, terminierten, regelstetigen standardisierten rekurrenten Fuzzy-Systems ist ein Fixpunkt der Transitionsfunktion und sein elementarer Hyperquader besteht aus Fixpunkten.

Beweis:

Sei $\mathbf{x} \in X$ ein Häufungspunkt einer Trajektorie im Zustandsraum X . Für beliebig kleine $\varepsilon > 0$ gibt es einen Vektor $\mathbf{a}$ in dieser Trajektorie, so dass für beliebig hohe Indizes k_0 sowohl $\mathbf{a}$ als auch $\mathbf{f}^{k_0}(\mathbf{a})$ zum Vektor $\mathbf{x}$ einen Abstand kleiner als $\varepsilon/2$ haben. Da die Funktion $\mathbf{f}$ nichtexpandierend ist, gilt

$$\|\mathbf{f}^{k_0}(\mathbf{x}) - \mathbf{x}\| \leq \|\mathbf{f}^{k_0}(\mathbf{x}) - \mathbf{f}^{k_0}(\mathbf{a})\| + \|\mathbf{f}^{k_0}(\mathbf{a}) - \mathbf{x}\| \leq \|\mathbf{x} - \mathbf{a}\| + \|\mathbf{f}^{k_0}(\mathbf{a}) - \mathbf{x}\| < \varepsilon. \quad (\text{B.27})$$

Daher kommt auch die neue Trajektorie $\mathbf{f}^k(\mathbf{x})$ immer wieder beliebig nahe an $\mathbf{x}$ heran.

Da das rekurrente Fuzzy-System terminiert ist, besitzt es einen Endknoten, dessen Kernpositionsvektor $\hat{\mathbf{s}} \in X$ ein Fixpunkt von $\mathbf{f}$ ist. Also ist $\|\mathbf{f}^k(\mathbf{x}) - \hat{\mathbf{s}}\| = \|\mathbf{f}^k(\mathbf{x}) - \mathbf{f}^k(\hat{\mathbf{s}})\| \leq \|\mathbf{x} - \hat{\mathbf{s}}\|$. Die echte Ungleichheit $\|\mathbf{f}^k(\mathbf{x}) - \hat{\mathbf{s}}\| = \|\mathbf{f}^k(\mathbf{x}) - \mathbf{f}^k(\hat{\mathbf{s}})\| < \|\mathbf{x} - \hat{\mathbf{s}}\|$ für ein $k \in \mathbb{N}$ würde aber zu $\varepsilon < \|\mathbf{x} - \hat{\mathbf{s}}\| - \|\mathbf{f}^k(\mathbf{x}) - \hat{\mathbf{s}}\| \leq \|\mathbf{x} - \hat{\mathbf{s}}\| - \|\mathbf{f}^{k_0}(\mathbf{x}) - \hat{\mathbf{s}}\| \leq \|\mathbf{f}^{k_0}(\mathbf{x}) - \mathbf{x}\|$ für ein $\varepsilon > 0$ und für alle $k_0 > k \in \mathbb{N}$ führen, was allerdings Gleichung (B.27) widerspricht. Also ist $\|\mathbf{f}^k(\mathbf{x}) - \hat{\mathbf{s}}\| = \|\mathbf{f}^k(\mathbf{x}) - \mathbf{f}^k(\hat{\mathbf{s}})\| = \|\mathbf{x} - \hat{\mathbf{s}}\|$ für alle k .

Insbesondere für $k = 1$ erhält man die Gleichung $\|\mathbf{f}(\mathbf{x}) - \mathbf{f}(\hat{\mathbf{s}})\| = \|\mathbf{x} - \hat{\mathbf{s}}\|$ und daher sind die Voraussetzungen für Lemma 13 und 14 erfüllt. Nach Lemma 14 kann die Dimension der elementaren Hyperquader von $\mathbf{f}^k(\mathbf{x})$ mit steigendem Index k nicht zunehmen. Die Dimension kann aber auch nicht abnehmen, da die Trajektorie $\mathbf{f}^k(\mathbf{x})$ dem Vektor $\mathbf{x}$ immer wieder beliebig nahe kommt. Daher sind Lemma 13 und 14 in ihrer erweiterten Version anwendbar.

Wie im Beweis von Lemma 14 werden wieder die Mengen A , S und die Kernpositionsvektoren $\mathbf{M}$ und $\mathbf{m}$ eingeführt: Die Menge $A = \{\mathbf{j} \mid \prod_i \mu_{j_i}^{x_i}(x_i) \neq 0\}$ besteht aus den Indizes aller Eckpunkte des elementaren Hyperquaders von $\mathbf{x}$. Die Menge $S = \{\mathbf{s}_j^x \mid \mathbf{j} \in A\}$ besteht aus deren Eckpunkten. Die beiden Kernpositionsvektoren $\mathbf{M}$ und $\mathbf{m}$ aus S sind durch $\|\mathbf{M} - \hat{\mathbf{s}}\| = \max_{\mathbf{j} \in A} \|\mathbf{s}_j^x - \hat{\mathbf{s}}\|$ und $\|\mathbf{m} - \hat{\mathbf{s}}\| = \min_{\mathbf{j} \in A} \|\mathbf{s}_j^x - \hat{\mathbf{s}}\|$ ausgezeichnet.

Man betrachte nun die Folge $\mathbf{f}^k(\mathbf{M})$. Da das rekurrente Fuzzy-System terminiert ist und $\mathbf{M}$ ein Kernpositionsvektor ist, erreicht die Folge nach endlich vielen Schritten den Kernpositionsvektor $\mathbf{M}_\infty$ eines Endknotens und verbleibt dort. Zugleich ist für jedes $k \in \mathbb{N}$ der Punkt $\mathbf{f}^k(\mathbf{M})$ auch der Eckpunkt des elementaren Hyperquaders des Punktes $\mathbf{f}^k(\mathbf{x})$

(Lemma 14), der am weitesten von $\hat{\mathbf{s}}$ entfernt ist (Lemma 13). Da die Folge $\mathbf{f}^k(\mathbf{x})$ wieder beliebig nahe an $\mathbf{x}$ herankommt, nimmt die Folge $\mathbf{f}^k(\mathbf{M})$ den Kernpositionsvektor $\mathbf{M}$ immer wieder an. Also ist $\mathbf{M}$ ein Häufungspunkt der Folge $\mathbf{f}^k(\mathbf{M})$ und damit identisch mit deren Grenzwert $\mathbf{M}_\infty$. Also ist $\mathbf{M}$ ein Fixpunkt. Analog folgt, dass auch $\mathbf{m}$ ein Fixpunkt ist. Daher sind nach Lemma 14 für alle k die elementaren Hyperquader der Vektoren $\mathbf{f}^k(\mathbf{x})$ gleich und die Funktion $\mathbf{f}$ bildet die Menge S in sich selbst ab.

Angenommen es gibt einen Kernpositionsvektor aus S , der kein Fixpunkt ist. Dann gibt es einen Kernpositionsvektor $\mathbf{z}$ in S , dessen Bild $\mathbf{f}(\mathbf{z}) \neq \mathbf{z}$ der Kernpositionsvektor eines Endknotens ist, weil die Transitionsfunktion die Menge S in sich selbst abbildet und das rekurrente Fuzzy-System terminiert ist. Da $\hat{\mathbf{s}}$ der Kernpositionsvektor eines beliebigen Endknotens ist, kann man auch $\hat{\mathbf{s}} = \mathbf{f}(\mathbf{z})$ wählen. Dann gilt einerseits $0 = \|\mathbf{f}(\mathbf{z}) - \mathbf{f}(\hat{\mathbf{s}})\|$ und andererseits nach Lemma 13: $0 = \|\mathbf{f}(\mathbf{z}) - \mathbf{f}(\hat{\mathbf{s}})\| = \|\mathbf{z} - \hat{\mathbf{s}}\| \neq 0$. Also war die Annahme falsch und alle Kernpositionsvektoren aus S sind Fixpunkte, d.h. es gilt $\mathbf{f}(\mathbf{s}_j^x) = \mathbf{s}_j^x$ für alle $\mathbf{j} \in A$.

Damit gilt für jeden beliebigen Vektor $\tilde{\mathbf{x}}$ aus dem elementaren Hyperquader von $\mathbf{x}$ nach Lemma 11 und 12:

$$\mathbf{f}(\tilde{\mathbf{x}}) = \sum_{\mathbf{j} \in A} \mathbf{f}(\mathbf{s}_j^x) \prod_i \mu_{j_i}^{x_i}(\tilde{x}_i) = \sum_{\mathbf{j} \in A} \mathbf{s}_j^x \prod_i \mu_{j_i}^{x_i}(\tilde{x}_i) = \tilde{\mathbf{x}} \quad (\text{B.28})$$

und daher besteht der elementare Hyperquader von $\mathbf{x}$ aus Fixpunkten. Insbesondere ist auch der Häufungspunkt $\mathbf{x}$ ein Fixpunkt. $\square$

C Gültige Sätze bei Min-Max Inferenz

Modifiziert man rekurrente Fuzzy-Systeme dahingehend, dass statt des Produkt-Operators in Aggregation und Implikation der Minimumsoperator und statt des Summationsoperators in der Akkumulation der Maximumsoperator verwendet werden, so bleiben einige Lemmata und Sätze aus der Arbeit gültig.

Insbesondere bleibt dabei die Äquivalenz von Automaten und den so modifizierten rekurrenten Fuzzy-Systeme bei Einschränkung auf Kernpositionswerte erhalten (Lemma 1). Außerdem sind die so modifizierten rekurrenten Fuzzy-Systeme weiterhin stetig bei stetigen Zugehörigkeitsfunktionen und monoton zwischen Kernpositionsvektoren. Die nachfolgende Tabelle listet alle die Lemmata und Sätze auf, die nur auf den Äquivalenz-, Stetigkeits- und Monotonieüberlegungen beruhen und daher auch für die so modifizierten rekurrenten Fuzzy-Systeme gelten.

Lemma/Satz	Seite
Lemma 1	28
Lemma 2	30
Satz 1	32
Satz 7	42
Satz 8	42
Satz 9	42
Satz 10	43
Satz 11	46
Satz 12	50
Lemma 3	50

D Sätze über Konvergenz und Ω -Grenzmengen

In diesem Teil des Anhangs werden das Konvergenzverhalten und die Ω -Grenzmengen von zeitdiskreten Systemen untersucht. Siehe dazu auch [39]. Vorrangig werden zeitdiskrete Systeme (f_k, I) behandelt, die durch Selbstabbildungen f_k eines kompakten Intervalls I und einer Iterationsgleichung der Form

$$x(k+1) = f_{k+1}(x(k)) \text{ mit } k \geq 0 \quad (\text{D.1})$$

gegeben sind. Beispielsweise ergeben sich die Funktionen $f_{k+1}(x)$ bei eindimensionalen, rekurrenten Fuzzy-Systemen durch $f_{k+1}(x(k)) = f(x(k), \mathbf{u}(k))$. Die Hintereinanderschaltung der Funktionen f_k bildet eine Funktionsfolge

$$\{f_k^k\}_{k \in \mathbb{N}_0} \text{ mit } f_k^k = f_k \circ \dots \circ f_2 \circ f_1 \text{ für } k \geq 1 \text{ und } f_0^0 = id, \quad (\text{D.2})$$

den *Fluss* des dynamischen Systems.

Ausgehend von einem Startwert $x(0)$ generiert das dynamische System, genauer dessen Fluss, eine Folge

$$\gamma = \{x(k)\}_{k \in \mathbb{N}_0} = \{f_k^k(x(0))\}_{k \in \mathbb{N}_0} \quad (\text{D.3})$$

von Zustandswerten, den *Orbit* von $x(0)$. Für die Untersuchung des Verhaltens für $k \rightarrow \infty$ ist die Menge der Häufungspunkte von γ von Interesse. Sie ist bei dynamischen Systemen mit stetigen Funktionen f_k identisch mit der Ω -Grenzmenge des Systems [48], die durch

$$\Omega = \bigcap_{n \in \mathbb{N}} \overline{\bigcup_{k \geq n} f_k^k(x(0))} \quad (\text{D.4})$$

gegeben ist. Aus dieser Definition ist ersichtlich, dass eine Ω -Grenzmenge als Schnitt von unendliche vielen, abgeschlossenen Mengen selbst abgeschlossen ist [31].

In autonomen Systemen¹ sind die Funktionen f_k unabhängig vom Parameter k . In diesem Fall wird der Index k ausgelassen. Der Fluss eines autonomen Systems wird folgendermaßen

¹Zu beachten ist, dass der hier verwendete Begriff eines autonomen Systems allgemeiner gefasst ist, als dies bei autonomen rekurrenten Fuzzy-Systemen der Fall ist. Auch bei einem nicht-autonomen rekurrenten Fuzzy-System mit konstantem Eingabevektor $\mathbf{u}(k) = \mathbf{u}_0$ hängt die Transitionsfunktion $f_k(x) = f(x, \mathbf{u}(k)) = f(x, \mathbf{u}_0)$ als Funktion von x nicht von der Zeit k ab. Daher ist ein solches System autonom in dem hier verwendeten Sinne.

ausgedrückt:

$$f^k = \underbrace{f \circ \dots \circ f \circ f}_{k\text{-fach}} \text{ für } k \geq 1 \text{ und } f^0 = id. \quad (\text{D.5})$$

In autonomen Systemen mit stetiger Funktion f ist jede Ω -Grenzmenge invariant gegenüber f , d.h. es gilt² $f(\Omega) = \Omega$. Wird ein Punkt x durch ein autonomes System nach n Schritten das erste Mal wieder auf sich selbst abgebildet, d.h. $f^n(x) = x$ für das kleinste $n \in \mathbb{N}$, so wird x als Punkt eines Zyklus der Länge n bezeichnet. Ein Punkt, der $f(x) = x$ erfüllt, heißt auch Fixpunkt von f .

Für autonome Systeme (f, I) auf einem kompakten Intervall I ergeben sich ausschließlich einelementige Ω -Grenzmengen, falls die Funktion f stetig ist und keinen 2-er Zyklus besitzt, wie folgender Satz zeigt:

Satz 26

Für ein autonomes zeitdiskretes System (f, I) auf einem reellen, kompakten Intervall $I = [a, b]$ mit einer stetigen Funktion f sind die folgenden Aussagen äquivalent:

- (1) Der Orbit $\gamma = \{f^k(x(0))\}_{k \in \mathbb{N}_0}$ konvergiert für jeden Anfangszustand $x(0) \in [a, b]$.
- (2) Die Funktion f besitzt keine 2-er Zyklen.
- (3) Alle periodischen Punkte von f sind Fixpunkte, d.h. aus $f^k(x_0) = x_0$ folgt schon $f(x_0) = x_0$.
- (4) Falls $f(x_0) > x_0$ ist, so ist auch $f^k(x_0) > x_0$ für alle $k \in \mathbb{N}$; falls $f(x_0) < x_0$ ist, so ist auch $f^k(x_0) < x_0$ für alle $k \in \mathbb{N}$.
- (5) Falls $f(x_0) > x_0$ ist, folgt $f(x) > x_0$ für alle $x \in [x_0, f(x_0)]$; falls $f(x_0) < x_0$ ist, folgt $f(x) < x_0$ für alle $x \in [f(x_0), x_0]$.

Beweis:

Zuerst wird die Äquivalenz von den Aussagen (2) bis (5) durch folgende Implikationskette gezeigt: (2) $\Rightarrow$ (3) $\Rightarrow$ (4) $\Rightarrow$ (5) $\Rightarrow$ (2).

Die Implikation (2) $\Rightarrow$ (3) folgt aus Satz 3, dem Theorem von Sarkovskii [79].

Der nächste Schritt – (3) $\Rightarrow$ (4) – wird durch einen Widerspruchsbeweis gezeigt. Dabei reicht es aus, den Fall mit $f(x_0) > x_0$ zu behandeln. Der andere Fall folgt analog. Die Annahme, dass $f^k(x_0) = x_0$ für ein $k \in \mathbb{N}$ gilt, führt sofort zu einem Widerspruch mit (3). Deshalb muss nur noch der Fall $f^k(x_0) < x_0$ mit einem $k \in \mathbb{N}$ ausgeschlossen werden. In einem solchen Fall erhält man für den kleinsten Wert a aus dem Intervall I : $f^k(a) \geq a$. Nach dem Mittelwertsatz gibt es dann ein $x_1 \in [a, x_0)$ mit $f^k(x_1) = x_1$, welches nach (3) auch $f(x_1) = x_1$ erfüllt. Daher ist x_1 ein Fixpunkt in $[a, x_0)$. Ohne Einschränkung der Allgemeinheit sei x_1 der größte Fixpunkt in dem Intervall $[a, x_0)$. Also erfüllen alle $x \in (x_1, x_0)$ die Ungleichung $f(x) > x$. Jetzt lässt sich ein $x_2 \in (x_1, x_0)$ nahe x_1 finden, so dass $\{f(x_2), f^2(x_2), \dots, f^k(x_2)\} \subset (x_1, x_0)$. Dann folgt aus $f^k(x_2) > x_2$ und der Annahme $f^k(x_0) < x_0$, dass es einen weiteren Punkt $x_3 \in (x_2, x_0)$ mit $f^k(x_3) = x_3$ gibt. Wiederum muss x_3 nach (3) ein Fixpunkt sein. Allerdings widerspricht x_3 der Maximalitätseigenschaft von x_1 . Die Annahme, dass $f^k(x_0) < x_0$ ist, führt also auch zu einem Widerspruch mit (3). Deshalb ist $f^k(x_0) > x_0$ für alle $k \in \mathbb{N}$.

²Der Beweis ist einfach, bzw. die Aussage folgt als Spezialfall aus Satz 27, der im Anschluss gezeigt wird.

Die Implikation $(4) \Rightarrow (5)$ wird ebenfalls durch einen Widerspruchsbeweis gezeigt: Es wird wiederum nur der Fall $f(x_0) > x_0$ betrachtet. Angenommen es gäbe ein $x_1 \in [x_0, f(x_0)]$ mit $f(x_1) \leq x_0$. Da $f(x_0) > x_0$ ist, gäbe es ein $x_2 \in (x_0, x_1]$ mit $f(x_2) = x_0$. Also wäre $f(x_2) = x_0 < x_2$ und $f^2(x_2) = f(x_0) \geq x_1 \geq x_2$, also $f(x_2) < x_2$ und $f^2(x_2) \geq x_2$, im Widerspruch zu (4).

Für die Implikation $(5) \Rightarrow (2)$ muss nur $x = f(x_0)$ in Aussage (5) eingesetzt werden.

An dieser Stelle ist die Implikationskette gezeigt und die Aussagen (2) bis (5) sind äquivalent. Nun wird Aussage (1) behandelt:

Die Implikation $(1) \Rightarrow (2)$ ist trivial. Umgekehrt werden nun die äquivalenten Aussagen (2) und (4) verwendet, um (1) herzuleiten. Zuerst wird angenommen, dass $\gamma = \{f^k(x(0))\}_{k \in \mathbb{N}_0}$ selbst für große Zeitschritte k nicht monoton ist, da sonst die Folge sowieso konvergieren würde. Jetzt lässt sich γ in zwei Teilfolgen γ_a und γ_b aufteilen: γ_a enthält alle Folgelemente $x(k)$ von γ mit $f(x(k)) > x(k)$ und γ_b alle Folgelemente $x(k)$ von γ mit $f(x(k)) < x(k)$.

Da γ niemals monoton wird, enthalten beide Teilfolgen unendlich viele Elemente. Nach Aussage (4) ist γ_a (bzw. γ_b) streng monoton steigend (bzw. fallend) und konvergiert daher gegen einen Wert a_∞ (bzw. b_∞).

Die Funktion f bildet immer wieder Elemente von γ_a auf Elemente von γ_b ab und umgekehrt, da sich γ ja aus den beiden Teilfolgen γ_a und γ_b zusammensetzt. Da f stetig ist, gilt für die Grenzwerte der Teilfolgen: $f(a_\infty) = b_\infty$ und $f(b_\infty) = a_\infty$. Nach (2) ist also a_∞ ein Fixpunkt von f und daher ist $a_\infty = b_\infty$. Da die Folge γ aus den beiden Teilfolgen γ_a und γ_b besteht, konvergiert sie gegen a_∞ . $\square$

Der Beweis kann auch elementar geführt werden, ohne das Theorem von Sarkovskii zu bemühen: Aussage (1) impliziert (3) und (3) impliziert (2). Setzt man $k = 2$ in Aussage (4) so kann die Äquivalenz von (2), (5) und der so modifizierten Aussage (4) auf die gleiche Art gezeigt werden wie sie im Beweis oben angegeben wurde. Im letzten Schritt – (2) und modifizierte Aussage (4) $\Rightarrow$ (1) – ist es nur nötig, zusätzlich nachzuweisen, dass die Teilfolgen γ_a und γ_b monoton sind. Dies kann durch Induktion geschehen. So lässt sich mit elementaren Mitteln zeigen, dass (2) und (3) äquivalent sind und die Einschränkung von $k = 2$ kann damit fallen gelassen werden.

Satz 26 lässt sich auch auf stetige Funktionen anwenden, wenn sich diese zu einer Funktion auf einem geschlossenen Intervall oder auf $\mathbb{R} \cup \{-\infty, \infty\}$ stetig ergänzen lassen. Im zweiten Fall ist eine Konvergenz gegen $\pm\infty$ als bestimmte Divergenz zu bewerten.

Außerdem ist die Behandlung von Funktionen mit Zyklen möglich. Betrachtet werden Funktionen $f : [a, b] \rightarrow [a, b]$, die ausschließlich Zyklen der Länge 2^n für verschiedene $n \leq N \in \mathbb{N}_0$ haben. In diesem Fall hat die Funktion f^{2^N} keinen Zweierzyklus und daher konvergiert nach Satz 26 ein Orbit unter f^{2^N} für jeden Startwert $x(0)$. Da die Funktion f stetig ist, konvergieren auch alle Orbits unter f gegen einen Grenzzyklus, dessen Länge eine Zweierpotenz ist, welche kleiner oder gleich 2^N ist. Funktionen mit anderen Zyklenlängen lassen sich nicht so einfach behandeln, wie im Kapitel 5 im Anschluss an Satz 3 angedeutet wird.

Für nicht-autonome Systeme ist Satz 26 nicht übertragbar. Im Allgemeinen lassen sich

für solche Systeme auch keine Zyklen definieren. Schränkt man das Verhalten für nicht-autonome Systeme allerdings geeignet ein, so kann man zu ähnlichen Ergebnissen kommen. Dazu ermutigt der nachfolgende Satz:

Satz 27

Sei (f_k, I) ein nicht-autonomes zeitdiskretes System auf einem kompakten Intervall I . Wenn die Funktionen f_k gleichmäßig gegen eine stetige Grenzfunktion $f_\infty : I \rightarrow I$ konvergieren, so ist die Ω -Grenzmenge Ω eines jeden Orbits invariant unter f_∞ , d.h. es gilt $f_\infty(\Omega) = \Omega$.

Beweis:

Im ersten Schritt wird gezeigt, dass $x_\Omega \in \Omega \Rightarrow f_\infty(x_\Omega) \in \Omega$, also $f_\infty(\Omega) \subset \Omega$.

Man wähle ein $x_\Omega \in \Omega$ und ein $\varepsilon > 0$. Da f_∞ stetig ist, gibt es ein $\delta > 0$, so dass für alle x in einer δ -Umgebung $U_\delta(x_\Omega)$ von x_Ω auch $f_\infty(x) \in U_\varepsilon(f_\infty(x_\Omega))$ ist. Da f_k gleichmäßig gegen f_∞ konvergiert, gibt es einen Index $N \in \mathbb{N}$, ab dem für alle $k > N$ und alle $x \in U_\delta(x_\Omega)$ Folgendes gilt: $f_k(x) \in U_\varepsilon(f_\infty(x)) \subset U_{2\varepsilon}(f_\infty(x_\Omega))$. Die Folge γ hat also für jedes $\delta > 0$ eine unendliche Zahl von Folgegliedern in $U_\delta(x_\Omega)$. Deshalb liegen auch unendlich viele Folgeglieder von γ in $U_{2\varepsilon}(f_\infty(x_\Omega))$. Da ε beliebig klein gewählt werden kann, ist $f_\infty(x_\Omega)$ ein Häufungspunkt von γ und damit ist $f_\infty(x_\Omega) \in \Omega$.

Im zweiten Schritt ist zu zeigen, dass es für jedes $x_\Omega \in \Omega$ ein $x_0 \in \Omega$ gibt, mit $f_\infty(x_0) = x_\Omega$, also dass $f_\infty(\Omega) \supset \Omega$ ist.

Man wähle ein $x_\Omega \in \Omega$ und ein $\varepsilon > 0$. Mit $\tilde{x}(k)$ werden nun die Folgeglieder $x(k)$ von γ bezeichnet, die zusätzlich $f_{k+1}(x(k)) \in U_\varepsilon(x_\Omega)$ erfüllen. Diese Folgeglieder bilden eine Teilfolge von γ und haben einen Häufungspunkt $x_\varepsilon \in \Omega$. Da f_∞ stetig ist und die Funktionen f_k gleichmäßig gegen f_∞ konvergieren, folgt für genügend große Indizes k :

$$\|f_\infty(x_\varepsilon) - x_\Omega\| \leq \underbrace{\|f_\infty(x_\varepsilon) - f_\infty(\tilde{x}(k))\|}_{< \varepsilon} + \underbrace{\|f_\infty(\tilde{x}(k)) - f_{k+1}(\tilde{x}(k))\|}_{< \varepsilon} + \underbrace{\|f_{k+1}(\tilde{x}(k)) - x_\Omega\|}_{< \varepsilon}$$

Daher ist $x_\varepsilon \in \Omega \cap f_\infty^{-1}(U_{3\varepsilon}(x_\Omega))$. Wählt man nun ein Folge $\{\varepsilon_n\}_{n \in \mathbb{N}_0} \rightarrow 0$, so lässt sich eine Folge $\{x_{\varepsilon_n}\}_{n \in \mathbb{N}_0}$ von Häufungspunkten in $\Omega \cap f_\infty^{-1}(U_{3\varepsilon_n}(x_\Omega))$ konstruieren. Diese Folge $\{x_{\varepsilon_n}\}_{n \in \mathbb{N}_0}$ besitzt wiederum einen Häufungspunkt x_0 , der auch in Ω liegt, da Ω eine abgeschlossene Menge ist. Weil f_∞ stetig ist, ist $f_\infty(x_0)$ ein Element aus dem Abschluss $\overline{U_{3\varepsilon_n}(x_\Omega)}$ für jeden noch so kleinen Wert von ε_n . Deshalb ist $f_\infty(x_0) = x_\Omega$. $\square$

Im Beweis wurde an keiner Stelle die Eindimensionalität des Intervalls I ausgenutzt. Deshalb ist Satz 27 auch auf dynamische, zeitdiskrete Systeme anwendbar, die nicht auf einem kompakten Intervall, sondern allgemein auf einem beschränkten und abgeschlossenen metrischen Raum operieren.

Des Weiteren sei angemerkt, dass bei einer Teilfolge von $\{f_k\}_{k \in \mathbb{N}_0}$, bestehend aus stetigen Funktionen f_k , mit der gleichmäßigen Konvergenz schon die Stetigkeit der Grenzfunktion f_∞ gewährleistet ist [31].

Die nachfolgenden Beispiele zeigen, dass man die Voraussetzungen für Satz 27 nicht ohne weiteres lockern kann.

Im ersten Gegenbeispiel ist die Grenzfunktion f_∞ unstetig. Die Wahl von

$$f_k(x) = f_\infty(x) = \begin{cases} \frac{1}{2}x & \text{für } x \neq 0, \\ 1 & \text{für } x = 0 \end{cases} \quad (\text{D.6})$$

ergibt $\Omega = \{0\}$ für alle x , aber $f_\infty(\Omega) \neq \Omega$.

Im zweiten Gegenbeispiel ist statt der gleichmäßigen Konvergenz nur punktweise Konvergenz der Funktionen f^k gegen eine Grenzfunktion f_∞ gegeben. Bei einer Wahl von

$$f_k(x) = \begin{cases} 1 - \frac{1}{k+1} & \text{für } x = \frac{1}{k} \text{ und geraden Index } k, \\ \frac{1}{k+1} & \text{für } x = 1 - \frac{1}{k} \text{ und ungeraden Index } k, \\ 0 & \text{sonst} \end{cases} \quad (\text{D.7})$$

ist f_∞ identisch 0. Speziell für den Startwert $x(0) = 0$ ergibt sich aber $\Omega(0) = \{0, 1\}$ und damit $f_\infty(\Omega(0)) = \{0\} \neq \Omega(0)$. Ändert man die Definition der Funktionen f_k in einer ε -Umgebung um $\frac{1}{k}$ bzw. $1 - \frac{1}{k}$ mit $\varepsilon = \frac{1}{k}$, so können selbst stetige Funktionen f_k gefunden werden, die zum gleichen Widerspruch führen.

Satz 27 zeigte die Invarianz von Ω -Grenzmengen unter der Grenzfunktion f_∞ . Hat eine solche Grenzfunktion keinen 2-er Zyklus, so ergibt sich die folgende Aussage, die der von Satz 26 ähnelt:

Satz 28

Sei (f_k, I) ein nicht-autonomes, zeitdiskretes dynamisches System auf einem reellen, kompakten Intervall I . Die Funktionen f_k sollen gleichmäßig gegen eine stetige Grenzfunktion $f_\infty : I \rightarrow I$ konvergieren, die keinen Zyklus der Länge 2 hat. Dann ist für jeden Anfangszustand dessen Ω -Grenzmenge ein – möglicherweise degeneriertes – abgeschlossenes Intervall, das ausschließlich Fixpunkte von f_∞ enthält.

Für den Beweis von Satz 28 sind einige Vorüberlegungen zu treffen und dabei die nachfolgenden drei Hilfssätze zu beweisen.

Da Ω im betrachteten Fall beschränkt und generell abgeschlossen ist, sind $v = \min\{x \in \Omega\}$ und $w = \max\{x \in \Omega\}$ wohldefiniert.

Hilfssatz 1

Zu jedem $c \in [v, w]$ gibt es ein $x \in [c, w]$ mit $f_\infty(x) \leq c$.

Beweis:

Es wird vom Gegenteil ausgegangen: Alle $x \in [c, w]$ erfüllen $f_\infty(x) > c$. Man wähle dann ein $0 < \varepsilon < \min_{x \in [c, w]} (f_\infty(x) - c)$. Da f_∞ stetig ist, lässt sich ein $\delta > 0$ finden, so dass für alle $x \in [c, w + \delta]$ gilt: $f_\infty(x) - c > \varepsilon > 0$. Da die Funktionen f_k gleichmäßig gegen f_∞ konvergieren, gibt es einen Index $N \in \mathbb{N}$, ab dem für alle $k > N$ und alle $x \in [c, w + \delta]$ die Ungleichung $|f_k(x) - f_\infty(x)| < \frac{\varepsilon}{2}$ erfüllt ist. Eine Kombination der beiden Ungleichungen ergibt dann $f_k(x) > c + \frac{\varepsilon}{2}$ für alle $x \in [c, w + \delta]$. Für genügend große Indizes k befinden sich alle Folgenglieder von γ in $[v - \delta, w + \delta]$. Erreicht die Folge das Intervall $[c + \frac{\varepsilon}{2}, w + \delta]$

wird sie es nicht mehr verlassen. Dies widerspricht aber der Voraussetzung, dass $v \leq c$ ein Häufungspunkt von γ ist. $\square$

Hilfssatz 2

Alle Punkte in $[v, w]$ sind Fixpunkte von f_∞ , d.h. alle $x \in [v, w]$ erfüllen $f_\infty(x) = x$.

Beweis:

Durch $A = \{x \in [v, w] \mid |f_\infty(x) - x| = \max_{x_0 \in [v, w]} |f_\infty(x_0) - x_0|\}$ wird eine nichtleere Teilmenge von $[v, w]$ festgelegt.

Angenommen $A_> = A \cap \{x \mid f_\infty(x) > x\}$ ist nicht leer. In diesem Fall lässt sich $c = \max\{x \in A_>\}$ finden. Alle $x \in [c, f_\infty(c)]$ erfüllen dann $f_\infty(x) > c$ laut Aussage (5) aus Satz 26. Dies gilt auch für alle $x \in (f_\infty(c), w]$, denn nach der Definition von $A_>$ gilt $-f_\infty(x) + x \leq |f_\infty(x) - x| \leq |f_\infty(c) - c| = f_\infty(c) - c$ und daher $f_\infty(x) \geq x - f_\infty(c) + c > c$. Also erfüllen alle $x \in [c, w]$ die Ungleichung $f_\infty(x) > c$ im Widerspruch zu Hilfssatz 1. Deshalb ist die Annahme falsch und die Menge $A_>$ ist die leere Menge.

Analog lässt sich folgern, dass $A_< = A \cap \{x \mid f_\infty(x) < x\}$ auch leer ist.

Also ist $\max_{x_0 \in [v, w]} |f_\infty(x_0) - x_0| = 0$ und daher $f_\infty(x) = x$ für alle $x \in [v, w]$. $\square$

Hilfssatz 3

$\Omega = [v, w]$.

Beweis:

Angenommen es gibt ein $x \in [v, w] \setminus \Omega$. Da $[v, w] \setminus \Omega$ offen ist, liegt darin eine ganze Umgebung $U(x)$ von x . Aus dieser Umgebung lässt sich ein echtes Intervall aus $U(x)$ wählen. Da außerdem nur endlich viele Elemente von γ in diesem Intervall liegen, gibt es darin wiederum ein echtes Intervall $[c, d]$ ohne Elemente von γ .

Aus dem Hilfssatz 2 folgt, dass für alle $x \in [c, w] \cap \gamma = [d, w] \cap \gamma$ gilt: $f_\infty(x) = x \geq d > c$. Wendet man die gleichen Argumente wie die aus dem Beweis von Hilfssatz 1 an, wobei man jedoch jedes dort erwähnte Intervall durch dieses Intervall geschnitten mit γ ersetzt, so erhält man schließlich auch einen Widerspruch. Also ist $\Omega = [v, w]$. $\square$

Beweis von Satz 28:

Nach Hilfssatz 3 ist Ω ein abgeschlossenes Intervall, welches nach Hilfssatz 2 ausschließlich Fixpunkte von f_∞ enthält. $\square$

Mit weiteren Bedingungen an die Funktionsfolge kann auch die Konvergenz jedes Orbits sichergestellt werden. In einem solchen Fall ist also jede Ω -Grenzmenge einelementig wie in Satz 26. Die Bedingungen werden im nachfolgenden Satz angegeben:

Satz 29

Sei (f_k, I) ein nicht-autonomes, zeitdiskretes dynamisches System auf einem reellen, kompakten Intervall I . Die Funktionen f_k sollen gleichmäßig gegen eine stetige Grenzfunktion $f_\infty : I \rightarrow I$ konvergieren, welche keinen Zyklus der Länge 2 besitzt. Ist zusätzlich $\sum_k \sup_{x \in [a, b]} |f_{k+1}(x) - f_\infty(x)| < \infty$, so besteht jede Ω -Grenzmenge aus einem Fixpunkt von f_∞ .

Beweis:

Laut Satz 28 ist die Ω -Grenzmenge eines jeden Orbits γ ein – eventuell degeneriertes – Intervall, welches nur Fixpunkte von f_∞ enthält. Angenommen $\Omega = [v, w]$ ist kein degeneriertes Intervall, d.h. $v < w$. In diesem Fall kann man ein ε mit $0 < \varepsilon < \frac{w-v}{5}$ wählen.

Für einen genügend großen Index $N \in \mathbb{N}$ gilt, dass $\sum_{k=N}^{\infty} \sup_{x \in [v, w]} |f_{k+1}(x) - f_\infty(x)| < \varepsilon$ und zugleich $x(N) \in [v + 2\varepsilon, w - 2\varepsilon]$.

Jetzt wird ein Index M so gewählt, dass für alle $k \in \{N, N+1, \dots, M\}$: $x(k) \in \Omega$. Die Wahl von $M = N$ ist immer möglich. Dann folgt $|x(M+1) - x(N)| \leq \sum_{k=N}^M |x(k+1) - x(k)| =$

$$\sum_{k=N}^M |f_{k+1}(x(k)) - f_\infty(x(k))| \leq \sum_{k=N}^{\infty} \sup_{x \in [v, w]} |f_{k+1}(x) - f_\infty(x)| < \varepsilon. \text{ Also ist } x(M+1) \in \Omega.$$

Mittels vollständiger Induktion folgt $|x(M+1) - x(N)| < \varepsilon$ für alle $M \geq N$. Folglich ist $|w - x(M+1)| \geq |w - x(N)| - |x(N) - x(M+1)| > 2\varepsilon - \varepsilon = \varepsilon$ für alle $M \geq N$ im Widerspruch zur Voraussetzung, dass w ein Häufungspunkt von γ ist. $\square$

Zu guter Letzt lässt sich Satz 29 auch wie folgt ausdrücken:

Satz 30

Gegeben sei eine Funktion $f(x, \mathbf{u}) : I \times U \rightarrow I$ mit einem reellen, geschlossenen Intervall I und $U \subset \mathbb{R}^n$. f sei Lipschitz-stetig in $\mathbf{u}_\infty \in U$. Sei $\{\mathbf{u}(k)\}_{k \in \mathbb{N}_0}$ eine Folge, die gegen $\mathbf{u}_\infty$ konvergiert und $\sum_k |\mathbf{u}(k) - \mathbf{u}_\infty| < \infty$ erfüllt. Außerdem sei die Funktion $f_\infty(x) = f(x, \mathbf{u}_\infty)$ stetig in der Variablen x und habe keine Zyklen der Länge 2.

Dann konvergiert die Funktionenfolge $\{f_k^k(x)\}_{k \in \mathbb{N}}$, die durch $f_k^k = f_k \circ f_{k-1} \circ \dots \circ f_2 \circ f_1$ und $f_k(x) = f(x, \mathbf{u}(k))$ definiert ist, punktweise gegen eine Grenzfunktion $f_\infty^\infty(x)$ und der Orbit eines jeden Startwertes $x(0)$ konvergiert gegen den Grenzwert $f_\infty^\infty(x(0))$.

Beweis:

Aufgrund der Lipschitzbedingung gilt: $|f(x, \mathbf{u}) - f(x, \mathbf{u}_\infty)| \leq l \cdot |\mathbf{u} - \mathbf{u}_\infty|$ mit einem $l > 0$. Damit gilt auch $\sup_{x \in [a, b]} |f_{k+1}(x) - f_\infty(x)| \leq l \cdot |\mathbf{u} - \mathbf{u}_\infty|$ unabhängig von x . Also konvergiert

die Funktionenfolge $\{f_k(x)\}_{k \in \mathbb{N}}$ gleichmäßig gegen $f_\infty(x) = f(x, \mathbf{u}_\infty)$. Außerdem folgt mit der Voraussetzung $\sum_k |\mathbf{u}(k) - \mathbf{u}_\infty| < \infty$ die Abschätzung

$$\sum_k \sup_{x \in [a, b]} |f_{k+1}(x) - f_\infty(x)| \leq l \sum_k |\mathbf{u}(k) - \mathbf{u}_\infty| < \infty.$$

Damit ist Satz 29 anwendbar und somit konvergiert die Punktfolge $\{f_k^k(x(0))\}_{k \in \mathbb{N}}$ für jeden Startwert $x(0) \in I$. Also konvergiert die Funktionenfolge $\{f_k^k(x)\}_{k \in \mathbb{N}}$ punktweise gegen eine Grenzfunktion $f_\infty^\infty(x)$, die die Abbildung eines jeden Startwertes $x(0)$ der Folge $\{f_k^k(x(0))\}_{k \in \mathbb{N}}$ auf dessen Grenzwert $f_\infty^\infty(x(0))$ beschreibt. $\square$

Literaturverzeichnis

- [1] J. Adamy, Breakout Prediction for Continuous Casting by Fuzzy Mealy Automata, Proc. of EUFIT'95, Aachen (1995) 754-759
- [2] J. Adamy, Device for early detection of runout in continuous casting, EP Priority Date 03.04.1995, European Patent EP 0 819 033 B1 (1998), US Patent 5,904,202 (1999)
- [3] J. Adamy, J. Freitag, S. Lorenz, Process for parametering a fuzzy automaton that compares a measurement system to a pattern signal, European Patent EP0941504B1 (2001), US Patent US 6,345,206 B1 (2002)
- [4] J. Adamy, R. Kempf, Regularity and chaos in recurrent fuzzy systems, Fuzzy Sets and Systems, 140 (2) (2003) 259-284.
- [5] J. Adamy, W. Tautz, M. Niemann, H.-J. Nitsche, J. Göpel, Automatisierungsverfahren mit Fuzzy-Systemen und neuronalen Netzen für Stranggießanlagen und Stahlbehandlungslinien, Proc. GMA-Conference on 'Computational Intelligence', Berlin, VDI-Report 1381 (1998) 69-86
- [6] J. Adamy, A. Yousif, Controller with dynamic discrete-time fuzzy-systems, German Patent DE19734711C1 (1999), European Patent EP1002260B1 (2002).
- [7] B. Aulbach, B. Kieninger, On three definitions of chaos, Nonlinear Dynamics and Systems Theory 1 (2) (2001) 23-37
- [8] R. Babuska, Fuzzy Modelling for Control, Kluwer, Boston, 1998
- [9] P. Bardet, A. Leclercq, M. Mangin, B. Sarter, Contrôle du processus de coulée continue de brames à la Sollac, Revue de Metallurgie - CIT, 80 (1983) 303-312
- [10] M.E. Begon, J.L. Harper, C.R. Townsend, Ecology: individuals, populations and communities, Blackwell Science, Oxford, 1996
- [11] T. Bertram, F. Svaricek, T. Bindel, R. Böhm, H. Kiendl, B.-M. Pfeiffer, M. Weber, Fuzzy Control, Zusammenstellung und Beschreibung wichtiger Begriffe, Automatisierungstechnik 42 (7) (1994) 322-326
- [12] J. C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum Press, New York, 1981
- [13] J. C. Bezdek, S. K. Pal, Fuzzy Models for Pattern Recognition, IEEE Press, Los Alamitos, 1992

- [14] L.S. Block, W.A. Coppel, Dynamics in one dimension, Springer Lecture Notes, 1513, Berlin, 1992
- [15] R. Blümel, W.P. Reinhardt, Chaos in atomic physics, Cambridge University Press, Cambridge (U.K.), 1997
- [16] W. Bock, A. Salski, A fuzzy knowledge-based model of population dynamics of the Yellow-necked mouse (*Apodemus flavicollis*) in a beach forest, *Ecol. Modelling* 108 (1-3) (1998) 155-161
- [17] G. Böhme, Fuzzy-Logik, Springer, Berlin, 1993
- [18] T. Booth, Sequential Machines and Automata Theory, John Wiley and Sons, New York, 1967
- [19] C.A. Cabrelli et al., Iterated Fuzzy Sets Systems: A New Approach to the Inverse Problem for Fractals and Other Sets, *J. Math. Anal. Applications* 171 (1) (1992) 79-100
- [20] R.L. Devaney, An introduction to chaotic dynamical systems, Addison-Wesley, Redwood City, CA, 1989
- [21] P. Diamond, Chaos in Iterated Fuzzy Systems, *J. Math. Anal. Applications* 184 (3) (1994) 472-484
- [22] R. O. Duda, P. E. Hart, G. E. Stork, Pattern Classification, Wiley, New York, 2001
- [23] H. Ehrig, Universal Theory of Automata, Teubner, Stuttgart, 1974
- [24] G. Feichtinger, C.M. Hommes, A. Milik, Complex Dynamics in a Threshold Advertising Model, *OR-Spektrum* 16 (1994) 101-111
- [25] A.L. Fradkov, A.Y. Pogromsky, Introduction to control of oscillations and chaos, World Scientific, Singapore, 1998
- [26] P. M. Frank, Komplexe Systeme - Nichtlineare Rückkopplungssysteme jenseits der Stabilität, *Automatisierungstechnik* 46 (4) (1998) 166-179
- [27] J. R. Garitagoitia, J. R. Gonzalez de Mendivil, J. Echanobe, J.J. Astrain, F. Farina, Deformed Fuzzy Automata for Correcting Imperfect Strings of Fuzzy Symbols, *IEEE Transactions on Fuzzy Systems* 11 (3) (2003) 299-310
- [28] R. Gonzalez, M. Thomason, Syntactic pattern recognition, Addison-Wesley, Reading MC (1978)
- [29] V. Gorrini, H. Bersini, Recurrent fuzzy systems, *Proc. 3rd IEEE Conference on Fuzzy Systems*, Orlando (1994) 193-198
- [30] J. Grantner, M. Patyra, Synthesis and analysis of fuzzy logic finite state machine models, *Proc. IEEE Int. Conf. Fuzzy Systems*, Orlando (1994) 205-210
- [31] H. Heuser, Lehrbuch der Analysis, Teil 1, Teubner, Stuttgart, 1988

- [32] H. Heuser, Lehrbuch der Analysis, Teil 2, Teubner, Stuttgart, 1998
- [33] S. Haykin, Neural networks: a comprehensive foundation, Prentice-Hall, London, 1999
- [34] M. Hormel, M. Schwartz, Trainable Process Monitoring System Applied to Sticker Detection in Continuous Casting, Proc. of EUFIT'98, Aachen (1998) 1458-1462
- [35] S. Y. Hwang, H.-S. Lee, J. J. Lee, General fuzzy acceptors for syntactic pattern recognition, Fuzzy Sets and Systems 80 (3) (1996) 397-401
- [36] R. Isermann, Digitale Regelsysteme, Springer, Berlin, 1988
- [37] S. Itoyama, et al., Process of and Apparatus for Continuous Casting with Detection of Possibility of Break Out, US Patent 4,949,777 (1990)
- [38] J. Kahlert, H. Frank, Fuzzy-Logik und Fuzzy-Control, Vieweg, Braunschweig, 1994
- [39] R. Kempf, On Ω -limit sets of discrete-time dynamical systems, Journal of Difference Equations and Applications 8 (12) (2002) 1121-1131
- [40] R. Kempf, H. Staab, Funktionswurzeln und Lernalgorithmen für abgetastete Systeme, Proc. 12. Workshop Fuzzy-Systeme des GMA Fachausschuß 5.22, Dortmund (2002) 90-99
- [41] R. Kempf, J. Adamy, Equilibria of recurrent fuzzy systems, Fuzzy Sets and Systems 140 (2) (2003) 231-257.
- [42] R. Kempf, J. Adamy, Sequential pattern recognition employing recurrent fuzzy systems, Fuzzy Sets and Systems, im Druck.
- [43] L.D. Kiel, Chaos theory in the social sciences: foundations and applications, Univ. of Michigan Press, Ann Arbor, 1997
- [44] G. J. Klir, B. Yuan, Fuzzy sets and fuzzy logic: theory and applicaions, Prentice Hall, London, 1995
- [45] P.E. Kloeden, Chaotic iterations of fuzzy sets, Fuzzy Sets and Systems 42 (1) (1991) 37-42
- [46] P.E. Kloeden, Fuzzy dynamical systems, Fuzzy Sets and Systems 7 (3) (1982) 275-296
- [47] A. Koski, M. Juhola and M. Meriste, Syntactic Recognition of ECG Signals by Attributed Finite Automata, Pattern Recognition, 28 (12) (1995) 1927-1940
- [48] W. Krabs, Dynamische Systeme: Steuerbarkeit und chaotisches Verhalten, Teubner, Stuttgart, 1998
- [49] V. Krebs, E. Schäfers, Dynamische Fuzzy Systeme zur qualitativen Prozeßmodellierung, GMA-Tagung 'Computational Intelligence', Berlin, VDI-Report 1381 (1998) 115-135

- [50] S. Kubota, A. Koyama, Y. Sasabe and H. Miki, A Real-time Expert System Applied to the Mold Bath Level Control of a Continuous Caster, *The Sumitomo Search* 50 (1992) 51-58
- [51] M. Kurano, M. Yasuda, J. Nakagami, Y. Yoshida, A limit theorem in some dynamic fuzzy systems, *Fuzzy Sets and Systems* 51 (1) (1992) 83-88
- [52] M. Kurano, M. Yasuda, J. Nakagami, Y. Yoshida, An approach to stopping problems of a dynamic fuzzy system, *Fuzzy Sets and Systems* 131 (2) (2002) 225-233
- [53] P. Lancaster, M. Tismenetsky, *The Theory of Matrices*, Academic Press, London, 1985.
- [54] M. Langer, M. Arzberger, New approaches to breakout prediction, *Steel Times International* 26 (10) (2002) 23-26
- [55] H.-S. Lee, Minimizing Fuzzy Finite Automata, *The Ninth IEEE International Conference on Fuzzy Systems* (2000) 65 -70
- [56] *Lexikon der Mathematik*, Red: Guido Walz, Spektrum, Heidelberg, 2001
- [57] S. Li, ω -chaos and topological entropy, *Trans. Amer. Math. Soc.* 339 (1993) 243-249
- [58] T.-J. Li, J.A. Yorke, Period three implies chaos, *AMM* 82 (1975) 985-992
- [59] V.J. Lopez, L. Snoha, There are no piecewise linear maps of type 2^∞ , *Trans. Amer. Math. Soc.* 349 (4) (1997) 1377-1387
- [60] J.M. Lu, K.J. Lin, C.H. Kou, W.C. Chien, Sticker Breakout Theory and its Prediction in Slab Continuous Casting, *Proc. of 76th Steelmaking Conference*, Dallas (1993) 343-353
- [61] J. Lunze, *Künstliche Intelligenz für Ingenieure*, Oldenbourg, München, 1994
- [62] R.M. May, Simple mathematical models with very complicated dynamics, *Nature* 261 (10) (1976) 459-467
- [63] K. C. Mills, T. J. H. Billany, A. S. Normaton, B. Walker, P. Grieveson, Causes of sticker breakout during continuous casting, *Ironmaking and Steelmaking*, 18 (4) (1991) 253-265
- [64] K. Nagel, D. E. Wolf, P. Wagner, P. Simon, Two-lane traffic rules for cellular automata: A systematic approach, *Physical Review E* 58 (2) (1998), 1425-1437
- [65] D. Nauck, F. Klawonn, R. Kruse, *Neuronale Netze und Fuzzy-Systeme*, Vieweg, Braunschweig, 1996
- [66] O. Nelles, LOLIMOT - Lokale, lineare Modelle zur Identifikation nichtlinearer, dynamischer Systeme, *Automatisierungstechnik* 45 (4) (1997) 163-174
- [67] H. Niemann, *Methoden der Mustererkennung*, Akademische Verlagsgesellschaft, Frankfurt a. M., 1974

- [68] M. Niemann, J. Adamy, H.-J. Nitsche, Modular mould level control for continuous casting plants, *Metallurgical Plant and Technology International* 20 (5) (1997) 92-98
- [69] M. Niemann, J. Adamy, H.-J. Nitsche, Method and Device for Casting a Strand from Liquid Metal, German Patent DE 196 33 738 C (2002), Italian Patent IT 0001293830 B (2002), US Patent US 5,915,456 (1999)
- [70] C. Omlin, K. Thornber, C. Gilies, Representation of fuzzy finite-state automata in continuous recurrent neural networks, *Proc. IEEE Int. Conf. Fuzzy Systems* (1996) 1023-1027
- [71] M. A. Patten, A. Klein, M. M. Wolf, Advances Breakout Prevention System Custom-Made to Product Mix, *Proc. of the 74th Steelmaking Conference*, Washington D.C., vol. 74 (1991) 553-560
- [72] N. R. Pal, *Pattern Recognition in Soft Computing Paradigm*, World Scientific Pub. Co, Singapur, 2001
- [73] W. Pedrycz, Fuzzy sets in pattern recognition: Accomplishments and challenges, *Fuzzy Sets and Systems* 90 (2) (1997) 171-176
- [74] K. Peeva, Fuzzy acceptors for syntactic pattern recognition, *Internat. J. Approximate Reasoning* 5 (3) (1991) 291-306
- [75] B.-M. Pfeiffer, J. Jäkel, A. Kroll, C. Kuhn, H.-B. Kuntze, U. Lehmann, T. Slavinski, V. Tews, Erfolgreiche Anwendungen von Fuzzy Logik und Fuzzy Control, *Automatisierungstechnik* 50 (10) (2002) 461-471 und (11) (2002) 511-521
- [76] T. Puu, A chaotic process with slow feed back: the case of business cycles, in: G. Haag, *Economic evolution and demographic change*, *Lecture Note in Economics and Mathematical Systems* 395, Springer, London, 1992
- [77] H. Richter, Chaotisches Verhalten dynamischer Systeme und seine regelungstechnische Behandlung, *Automatisierungstechnik* 48 (10) (2000) 471-477
- [78] J. Roughgarden, *Primer of ecological theory*, Prentice Hall, New Jersey, 1998
- [79] A.N. Sarkovskii, Co-existence of Cycles of Continuous Maps of the Line into Itself, *Ukrain. Mat. Z.* 16 (1964) 61-71 (russisch)
- [80] S. Sastry, *Nonlinear Systems*, Springer, New York, 1991.
- [81] E. Schäfers, V. Krebs, Dynamic Fuzzy Systems for Qualitative Process Modeling: Principles of a New System Theory, *Automatisierungstechnik* 47 (8) (1999) 382-389
- [82] E. Schäfers, *Dynamische Fuzzy-Systeme zur qualitativen Prozeßmodellierung: Eine neue Systemtheorie*, Doktorarbeit, Fortschritt-Berichte VDI, Reihe 8, Nr. 745, VDI Verlag, Düsseldorf, 1999
- [83] K. Schmid, V. Krebs, E. Ivers-Tiffée, Qualitative dynamische Modelle für die Werkstoffentwicklung von Hochtemperatur-Brennstoffzellen, *Automatisierungstechnische Praxis* 44 (7) (2002) 55-60

- [84] M. Schreckenberg, J. Wahle, Towards a Multi-Agent System for On-Line Simulations Based on Real-World Traffic Data, *Automatisierungstechnik* 49 (11) (2001) 485-492
- [85] D.R. Smart, Fixed point theorems, Cambridge Univ. Press, Cambridge, 1974.
- [86] J. Smital, Chaotic functions with zero topological entropy, *Trans. Amer. Math. Soc.* 297 (1986) 269-282
- [87] E. Sowka, P. Schulze-Diekhoff, J. Harder, F. Munscher, G. Beirer, Breakout avoidance system, BASYS, for continuous slab casting, *Iron and Steel Engineer* 76 (5) (1999) 30-34
- [88] M. Sugeno, K. Kang, Fuzzy modeling and control of multilayer incinerator, *Fuzzy Sets and Systems* 18 (3) (1986) 329-346
- [89] H. Takagi, M. Sugeno, Fuzzy Identification of Systems and its Application to Modeling and Control, *IEEE Trans. on Systems, Man and Cybernetics* 15 (1) (1985) 116-132
- [90] T. Tanaka, H. Endo, N. Kamada, S. Naito, H. Kominami, Trouble Forecasting System by Multi-Neural Network on Continuous Casting Process of Steel Productions, in T. Kohonen et al. (Ed.), *Artificial Neural Networks*, Elsevier Science Publishers B.V. (1991) 835-840
- [91] T. Terano, K. Asai, M. Sugeno, Fuzzy systems theory and its applications, Academic Press, San Diego, 1992
- [92] T. Terano, K. Asai, M. Sugeno, Applied fuzzy systems, Academic Press, London, 1994
- [93] J.T. Tou, R.C. Gonzalez, Pattern Recognition Principles, Addison-Wesley, Reading, MA, 1974
- [94] M. Treiber, D. Helbing, Microsimulations of Freeway Traffic Including Control Measures, *Automatisierungstechnik* 49 (11) (2001), 478-484
- [95] A. Tsuneoka, W. Ohasi, S. Ishitobi, T. Kataoka, M. Tenma, Measurement and Control System of Solidification in Continuous Casting Mold, *Proc. ISS-AIME Steelmaking Conference*, Detroit (1985) 3-10
- [96] F. A. Valentine, *Convex Sets*, Krieger, Huntington NY, 1976
- [97] Fuzzy logic and fuzzy control - Terms and definitions, VDI/VDE-Richtlinien, VDI/VDE 3550 Blatt 2, Beuth Verlag GmbH, Berlin, 2000
- [98] M.H. Vellekoop and R. Berglund, On Intervals, Transitivity = Chaos, *AMM* 101 (1994) 353-354
- [99] W.G. Wee, K.S. Fu, A Formulation of Fuzzy Automata and its Application as a Model of Learning Systems, *IEEE Transactions on System Science and Cybernetics* 5 (3) (1969) 439-447
- [100] M. Willimowski, D. Füssel, R. Isermann, Diagnose von Verbrennungsaussetzern in Ottomotoren durch Messung des Abgasdrucks, *mtz- Motortrtechnische Zeitschrift*, 60 (10) (1999) 654-663

- [101] Ch. Wissel, Theoretische Ökologie, Springer, Berlin, 1989
- [102] D.E. Wolf, Cellular automata for traffic simulations, Physica A 263 (1-4) (1999), 438-451
- [103] Y.C. Wang, S.G. Adams, J.S. Thorp, N.C. MacDonald, P. Hartwell, F. Bertsch, Chaos in MEMS, parameter estimation and its potential application, IEEE Transactions on Circuits and Systems 45 (10) (1998) 1013-1020
- [104] L. Zadeh, Fuzzy Sets, Information and Control 8 (3) (1965), 338-353
- [105] L. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning I, II, III", Information Science 8 (3) (1975), 199-251, 8 (4) (1975), 301-357 und 9 (1) (1976) 43-80

Curriculum Vitae

Roland Kempf wurde am 28.05.1974 in Alzenau i. Ufr. in Deutschland geboren. Er studierte von 1994-1997 Physik und Mathematik an der Universität Würzburg. Er beendete sein Studium nach einem Jahr Aufenthalt an der University of Texas at Austin mit dem Master of Arts in Physik im Jahr 1998. Von 1998 bis 2001 war er Promotionsstipendiat im Graduiertenkolleg "Intelligente Systeme für die Informations- und Automatisierungstechnik" der Deutschen Forschungsgemeinschaft am Fachgebiet Regelungstheorie und Robotik im Fachbereich Elektrotechnik an der Technischen Universität Darmstadt. In diesem Zeitraum begann er mit seiner Doktorarbeit. Seit 2001 ist er am selben Fachgebiet als wissenschaftlicher Mitarbeiter im Rahmen des DFG-Drittmittelprojektes "Untersuchung von Fuzzy-Systemen mit chaotischer Dynamik" (AD 174/2-1) tätig.