

Fuzzy Control in technischen Systemen

Jakob Schomann

- I. Einführung**
- II. Das ‘Fuzziness’ Konzept**
- III. Fuzzy Mengen**
- IV. Geschichte**
- V. Diskursuniversum (Universe of Discourse)**
- VI. Set Operationen**
- VII. Fuzzy Control System**
- VIII. Entwicklung eines Fuzzy Systems**
- IX. Defuzzification**
 - A. Mean of maxima
 - B. Centre of maximums
 - C. Centre of gravity
- X. Kombinatorische Explosion**
- XI. Fuzzy State Machines**
- XII. Typ-2 Fuzzy Logik**
- XIII. Referenzen**

I. Einführung

- *“It (fuzzy logic) can add depth and unpredictability to your game AI...”* Jeff Orkin [1]

Fuzzy Logik ist in der Lage mehr Tiefe und Unvorhersehbarkeit in eine Spiele KI zu bringen.

Bei der Unvorhersehbarkeit ist aber zu beachten das dies nicht bedeutet das nicht vorher gesagt werden kann wie das Control System reagiert, sondern nur das es schwerer ist ohne sich die Regeln im Detail anzusehen zu sagen ab welchem Zeitpunkt das System sein Verhalten ändert.

- Fuzzy = smooth, more ‘humanlike’ transitions [2]

Ein weiterer Vorteil von Fuzzy Logik ist das es flüssigere Übergänge von einem Zustand in einen anderen hat, es hat ein Verhalten das dem Menschen ähnlicher ist als die meisten anderen Kontrollmechanismen.

- Fuzzy rules equivalent to rules-of-thumb we use and make it easier to come up with better rules [3]

Die Regeln eines Fuzzy Control Systems können sehr einfach erstellt werden, man kann mit Daumenregeln also relativ wild aus der Luft gegriffene Regeln anfangen und braucht diese Regeln nur noch anzupassen.

- *“Fuzziness is used as a concept (in game AI), but fuzzy logic is simplified to crisp levels for efficiency”* [4]

Obwohl ein Fuzzy Control System mit Fuzzy Logik als dem Konzept der „Fuzziness“ arbeitet kann es mit der „Crisp Logic“ (Feste Werte, ab einem Level von 20 wird das gemacht) kombiniert werden da der Output eine Feste Zahl, ein bestimmter Zustand oder Verhalten ist welches dann in der „Crisp Logic“ dargestellt und weiterbenutzt werden kann.

II. Das ‘Fuzziness’ Konzept

Im normalen Sprachgebrauch werden oft relativa, also Wörter die nur das verhältniss von etwas zu etwas anderem angeben, benutzt.

Zum Beispiel:

Recht Groß

Sehr Groß

Mittleres Alter

Recht Kompakt

Relativ wenig

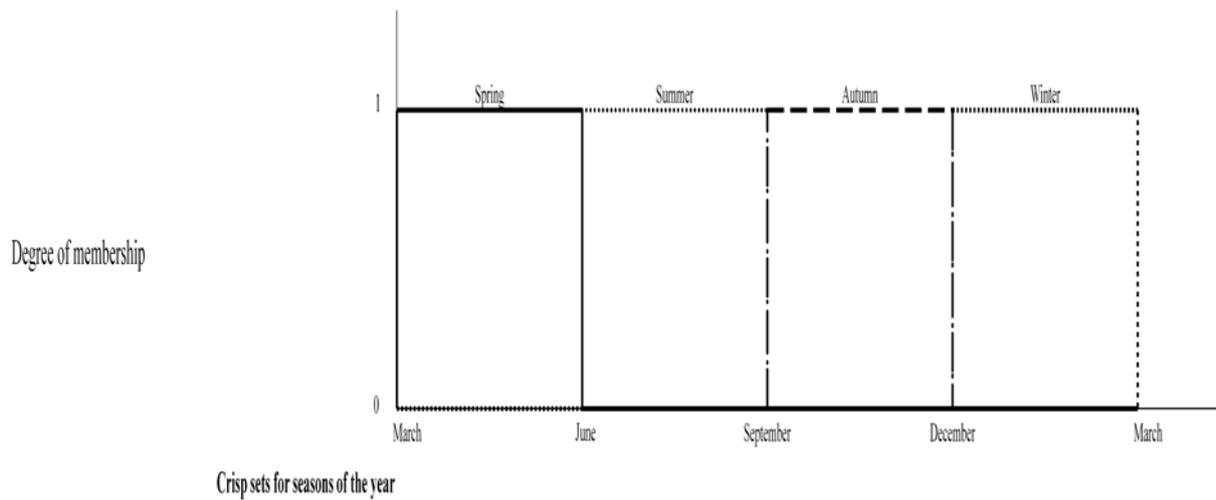
Wie man an diesen Beispielen sehen kann haben sie ohne einen Kontext keine Aussagekraft, denn zu sagen das etwas „Recht Groß“ ist, macht nur Sinn wenn beide Personen wissen wovon gesprochen wird und auch was die durchschnittliche Größen davon ist.

Eine recht große Blume ist immernoch um einiges kleiner als ein kleiner Baum, geschweige den einem großen.

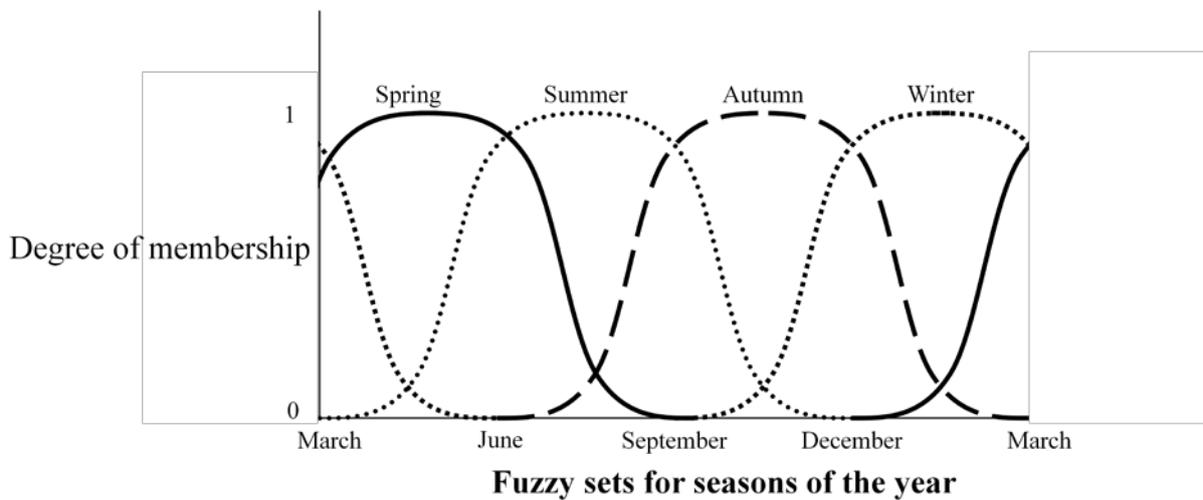
Genauso ist es davon abhängig wer diese Aussage macht denn ein Formel 1 Fahrer hat einen anderen Geschwindigkeitsbereich vor Augen als der durchschnittliche Autofahrer.

Dieses „Unsicherheit“ kann normalerweise von einem Computer nicht dargestellt werden denn dieser Arbeitet immer mit festen Werten. In der Natur unterliegen die Dinge jedoch einem stetigen Wandel, sie haben nicht einen festen Wert sondern befinden sich in einem ständigen wechseln von einem Zustand in den nächsten.

Beispiel: grafischer Vergleich der beiden Set Arten.



Ein Set mit festen Werten.



Ein Fuzzy Set.

Wie man an dem Beispiel sehen kann stellt das Fuzzy Set die Natur richtig dar. Auch wenn gerade Sommer ist gibt es schon die ersten Anzeichen für den Herbst, es herrscht also nicht genau ein Zustand sondern eine Mischung aus mehreren.

III. Geschichte

Fuzzy Logik wurde in den 60er Jahren von Lotfi Zadeh entwickelt. Die wichtigste Veröffentlichung ist die Abhandlung „Fuzzy Sets“ veröffentlicht in „Information and Control“ im Jahr 1965.

Am Anfang wurde es jedoch nur theoretisch diskutiert und fand keine praktischen Anwendungen, erst in den 80er Jahren wurde es in Japan eingesetzt und erst 10 Jahre später auch in Europa.

Seitdem wird Fuzzy Logik für viele verschiedene Anwendungen benutzt wie zum Beispiel:

- Turbinensteuerung, Antiblockiersystem
- Transmission control, Kontrolle eines Hochofens
- Image/pattern recognition, Planning and scheduling
- Luftkühlung, Waschmaschinen, Videokameras

Dies sind aber nur ein paar Beispiele, denn es gibt noch eine Menge von anderen Anwendungen in denen Fuzzy Logik benutzt wird meistens jedoch in Kontroll- und Ablaufsystemen.

Dabei hat ein Fuzzy System gegenüber den konventionellen Systemen zwei entscheidende Vorteile:

- Es arbeitet sehr nahe an dem wie ein Mensch das System Kontrollieren würde
- Es muss kein detailliertes mathematisches Modell erstellt werden

IV. Fuzzy Mengen

In normalen ('Crisp') Mengen gibt es nur zwei Möglichkeiten, entweder ist ein Element in der Menge oder nicht.

Die Mitgliedschaft in solch einer Menge kann durch die Funktion μ_A dargestellt werden.

$$\mu_A(x) = \left\{ \begin{array}{l} 1 \text{ if } x \text{ is a member of set } A \\ 0 \text{ if } x \text{ is not a member of set } A \end{array} \right\}$$

In einer Fuzzy Menge dagegen wird die Mitgliedschaft eines Elements durch eine **Zugehörigkeitsfunktion** bestimmt, diese kann alle Werte von 0 und 1 annehmen. Der Wert dieser Zugehörigkeitsfunktion gibt den Grad der Mitgliedschaft an.

Beispiel: Vergleich der Mitgliedschafts-Möglichkeiten der beiden Mengen.

In diesem Beispiel geht es um die Größe von Menschen.

Beide Mengen bestehen aus drei Elementen „klein“, „durchschnittlich“ und „groß“.

Die Reichweite der betrachteten Größen geht von 1m bis 2m.

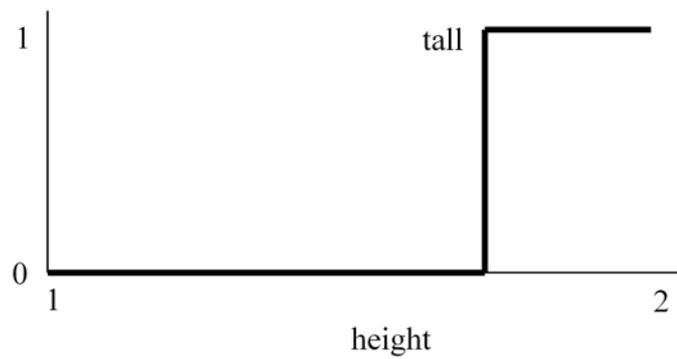
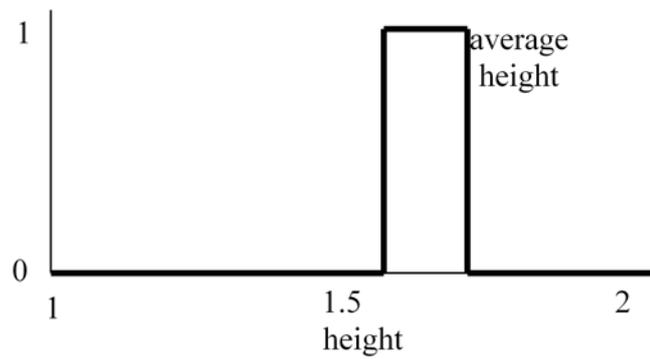
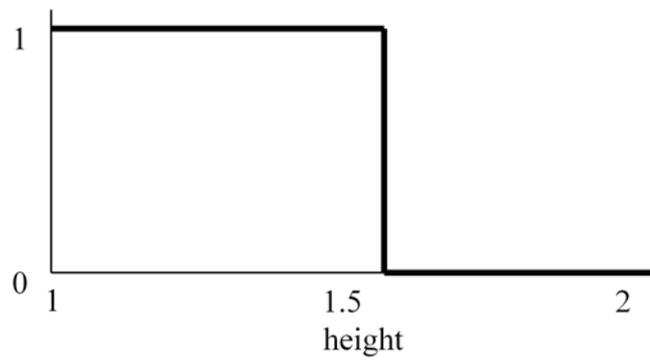
Für eine Menge mit festen Werten könnte man dann bestimmen das:

Jeder der kleiner ist als 1.62m zu „klein“ zählen.

Jeder zwischen 1.62m und 1.78m zum „durschnitt“

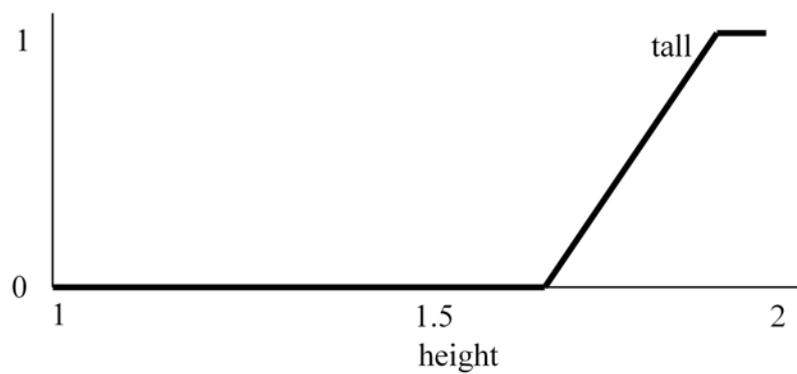
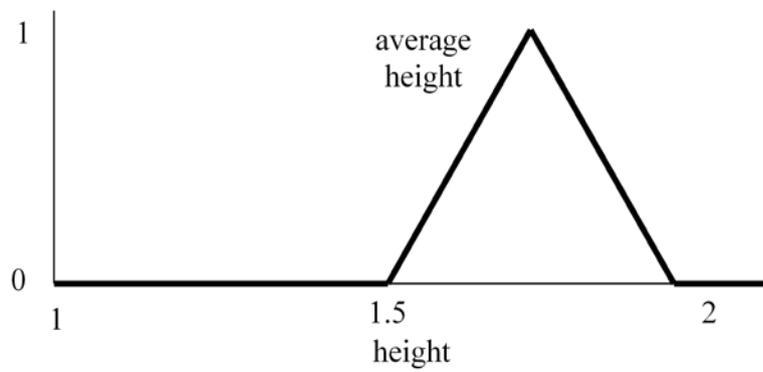
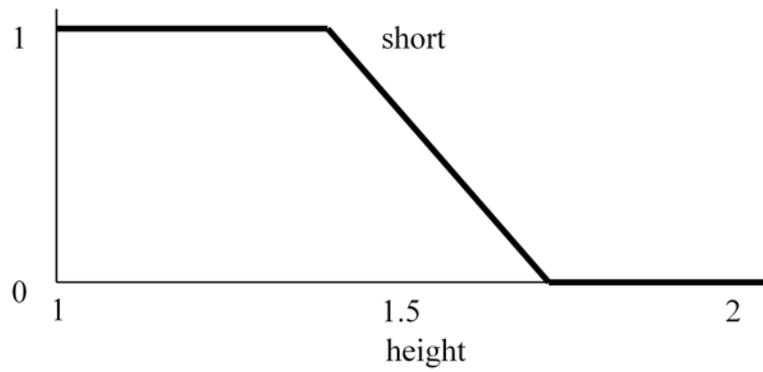
Und jeder der größer ist als 1.78m zu „groß“

Grafisch dargestellt entspräche das:



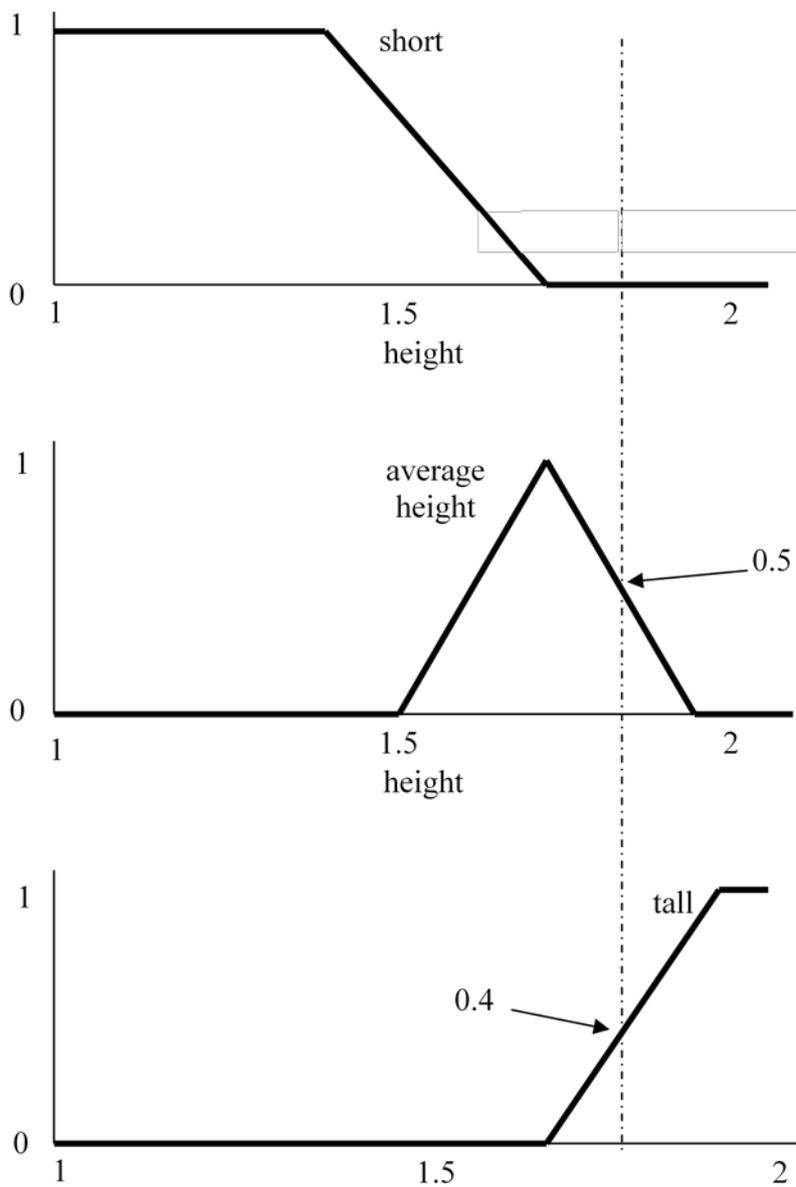
Wie man hier gut sehen kann gibt es drei eigene Bereiche von denen sicher aber keine mit einem anderen überschneidet.

Wenn man jetzt dieselbe Menge als Fuzzy Menge darstellen möchte, überschneiden sich die einzelnen Elemente stark miteinander.



Wie man leicht sehen kann gehören die meisten Menschen jetzt nicht mehr genau einem der drei Elemente an sondern bis zu zwei verschiedenen.

Bei einem Menschen der 1,8m groß ist würde das ganze so aussehen:



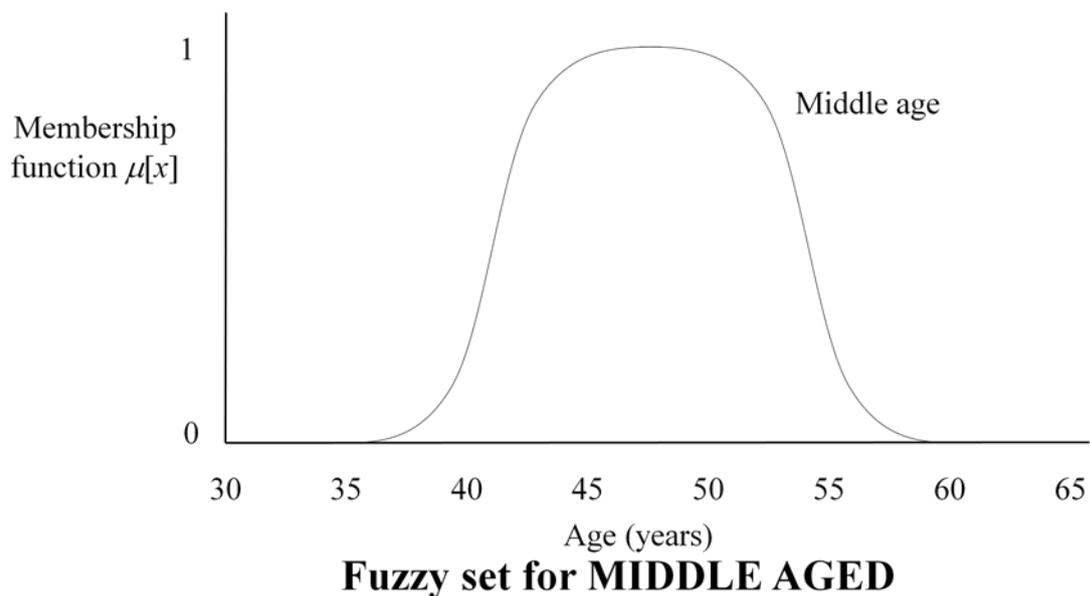
Er hat also genau zwei Mitgliedschaften, zum einen gehört er mit einem Anteil von 0.5 zum Element „durschnitt“ und mit einem Anteil von 0.4 zum Element „groß“.

V. Diskursuniversum (Universe of Discourse)

Das Diskursuniversum wird durch einen Anfangs und einen Endpunkt bestimmt.

Für diese beiden Punkte können entweder natürliche Maxima benutzt werden, wie zum Beispiel -273,15 Grad Celsius für den absoluten Nullpunkt. Es können aber auch ganz willkürlich Punkte genommen werden die für die bestimmte Menge sinnvoll erscheinen.

In der folgenden Grafik wird das Element „Middle age“ für eine Altersmenge dargestellt.

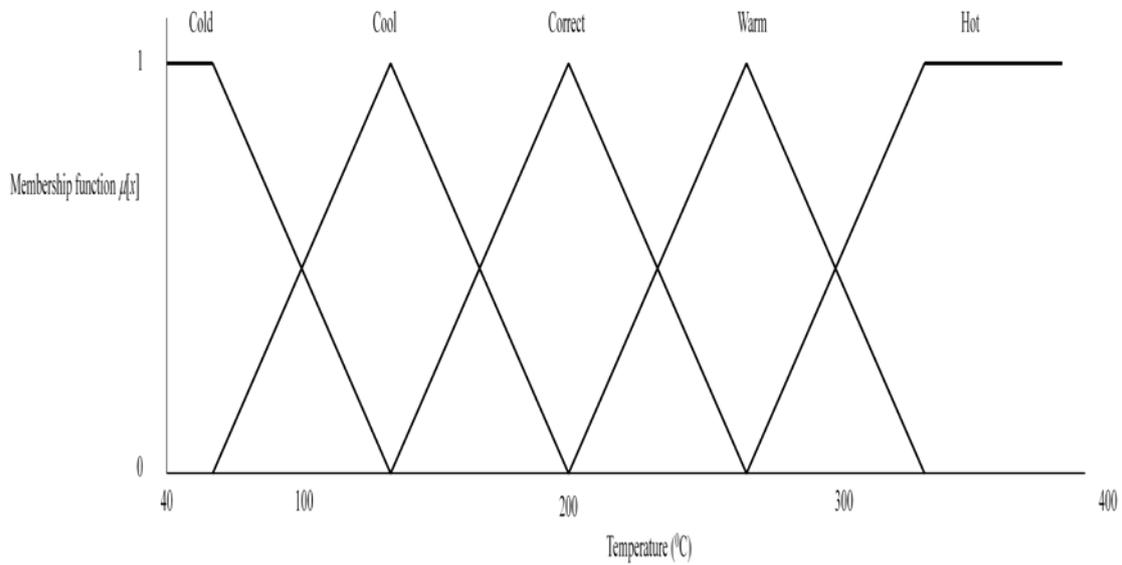


In diesem Beispiel fängt das Diskursuniversum mit 30 Jahren an und geht bis zu 65.

Das Element „Middle age“ dagegen hat die Grenzen 35 bis 60 Jahre.

Ein weiteres Beispiel:

Diesmal hat die Menge fünf verschiedene Elemente und das Diskursuniversum geht von 40° bis 400°.

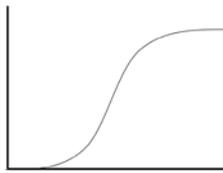


Universe of discourse for TEMPERATURE

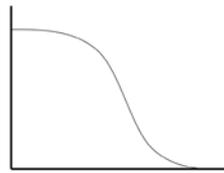
Form der Elemente:

Die Elemente einer Menge können viele verschiedene Formen annehmen. Die Form ist dabei abhängig von der gewählten Zugehörigkeitsfunktion.

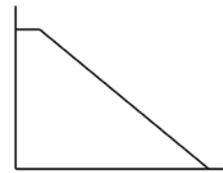
Neun verschiedene Beispiel Formen werden in der folgenden Grafik angezeigt. Diese sind aber nur die am häufigsten benutzten Formen. Theoretisch kann jedoch jede Funktion benutzt werden deren Funktionswerte im Intervall $[0,1]$ liegen.



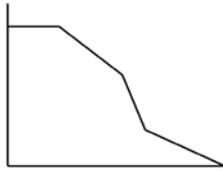
Ascending
S-Curve



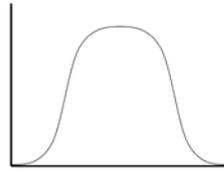
Descending
S-Curve



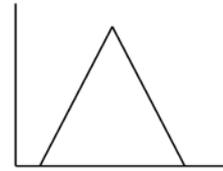
Linear



Piecewise
Linear



Gaussian or
Bell Curve



Triangular



Offset
Triangular



Truncated



Trapezoidal

Fuzzy set shapes

Die gewählte Form hat nur wenig Einfluss da drauf wie sich das System später verhalten wird, deshalb werden die Formen meistens so gewählt das sie schnell und einfach berechnet werden können.

VI. Set Operationen

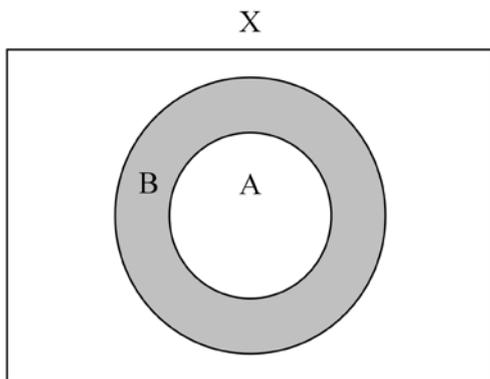
Für die normalen Mengen gibt es seine Reihe von festgelegten Operationen die auf die Menge angewandt werden können.

Die Gleichheit von zwei Mengen kann durch die Funktion μ_A berechnet werden.

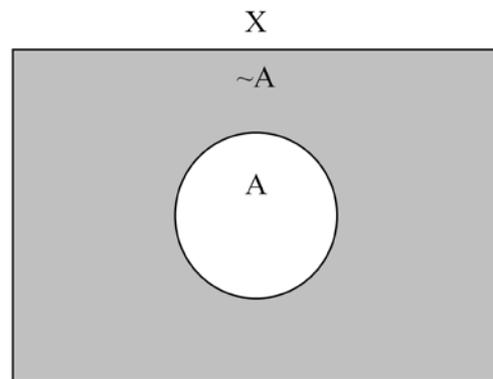
Die wichtigsten Operationen und eine grafische Darstellung sind in den folgenden Grafiken aufgeführt:

$$\mu_A(x) = \begin{cases} 0 & \text{if } x \notin A \\ 1 & \text{if } x \in A \end{cases} \quad (2)$$

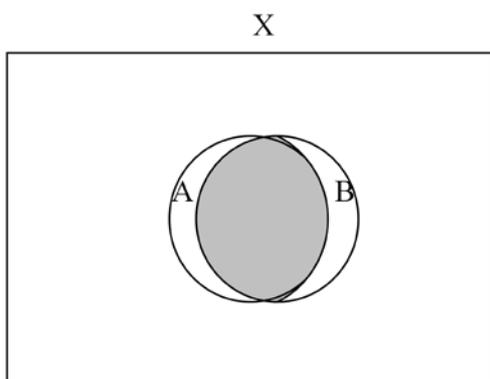
- \in Member of set
- \notin Not a member
- \cap Set AND or INTERSECTION
- \cup Set OR or UNION
- \subset INCLUSION
- \supset IMPLICATION
- \sim Set NOT, COMPLEMENT or INVERSION
- \emptyset NULL or EMPTY set



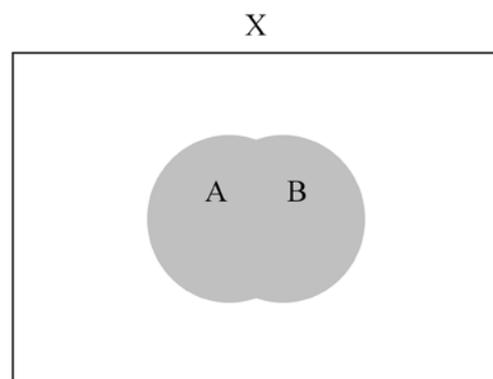
$A \subset B$
Inclusion of set A within set B



$\sim A$
Set A & Compliment $\sim A$



$A \cap B$
Intersection of sets A & B



$A \cup B$
Union if sets A & B

Crisp set operations

Für eine genauere Erklärung dieser und weiterer Operationen auf Mengen sei auf die Diskrete Mathematik Vorlesung von Prof. Dr. Iwanowski verwiesen.

Bei den normalen Mengen kann man nur dazu gehören oder nicht, es gibt keine Mehrdeutigkeit wie bei einer Fuzzy Logik Menge. Wegen dieser Mehrdeutigkeit lassen sich die logischen Operatoren nicht direkt übertragen sie müssen so abgeändert werden das sie mit dieser Mehrdeutigkeit arbeiten. Diese abgeänderten Operatoren existieren für die Vereinigungs- (**Union**), Schnittmengen- (**Intersection**) und Negationsoperatoren (**Complement**).

$$\text{Intersection} \quad A \cap B = \min(\mu_A[x], \mu_B[y]) \quad (3)$$

$$\text{Union} \quad A \cup B = \max(\mu_A[x], \mu_B[y]) \quad (4)$$

$$\text{Complement} \quad \sim A = 1 - \mu_A[x] \quad (5)$$

- Der **Schnittmengenoperator** wurde so geändert dass er eine neue Menge erstellt, diese enthält für jedes Element $\mu_N(X)$ den kleineren Zugehörigkeitswert der Mengen A und B.
- Der **Vereinigungsoperator** ist das genaue Gegenteil des Schnittmengenoperators anstatt des kleineren Zugehörigkeitswertes wird der größere der beiden Mengen A und B genommen.
- Weil die Zugehörigkeit eines Elements zu einer Fuzzy-Menge höchstens 1 betragen und mindestens 0 ist, wird die **Negation** jetzt mit $1 - \mu_A$ berechnet.
- Die anderen Logischen Operationen auf Mengen sind für Fuzzy Mengen nicht definiert.

VI. Fuzzy Control System

Vereinfacht gesagt ist ein Fuzzy Control System dasselbe wie ein regelbasiertes Experten System.

Zum einen gibt es eine Wissensdatenbank, in dieser stehen das Expertenwissen und die Regeln. Zum anderen gibt es eine sogenannte Inferenz Maschine, die ist eine Komponente die in der Lage ist aus dem Expertenwissen und den Regeln Schlussfolgerungen zu ziehen.

Der einzige Unterschied zwischen einem Fuzzy Control System und einem Experten System ist das die Regeln aus „Fuzzy propositions“ aufgebaut ist. Also die Regeln arbeiten nichtmehr mit festen Werten sondern mit Fuzzy Mengen.

Eines der Hauptprobleme mit Experten Systemen ist die Größe ihrer Wissensdatenbank. Von ihrer Größe hängt die Laufzeit des Experten Systems ab und diese kann bei komplexen Experten Systemen sehr groß werden.

Um dies zu verhindern kann man die Regeln in einzelne Module unterteilen. Diese haben dann eine Größe die noch bewältigt werden kann, also keine enorme Größe die einen tatsächlichen Einsatz des Systems verhindern würde.

VII. Entwicklung eines Fuzzy Systems

Am Anfang der Entwicklung eines Fuzzy Control Systems müssen viele Parameter festgelegt werden. Um sagen zu können wie welcher Parameter gewählt werden sollte ist viel Erfahrung in der Entwicklung eines Fuzzy Control Systems notwendig. Für einfachere Anwendungen reicht es jedoch meistens aus mit Schätzwerte zu beginnen und bei Bedarf das System nachträglich anzupassen.

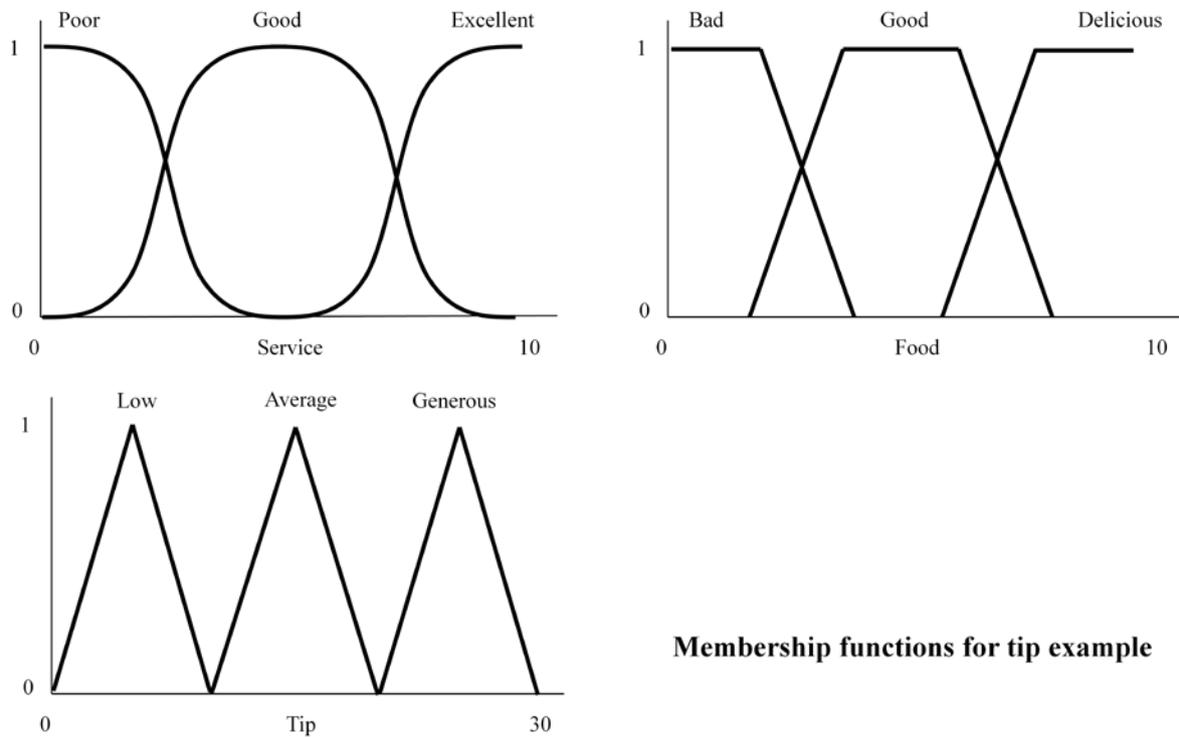
- Als erstes muss die Problemstellung analysiert und überlegt werden wie sie als Fuzzy Menge dargestellt werden kann.
- Dann muss entschieden werden welche Zugehörigkeitsfunktion benutzt werden soll.
- Außerdem muss eine Defuzzification Methode gewählt werden.
- Danach kann das System schon mit Regeln gefüllt werden.

Beispiel

In diesem Beispiel werden im System drei Regeln gespeichert:

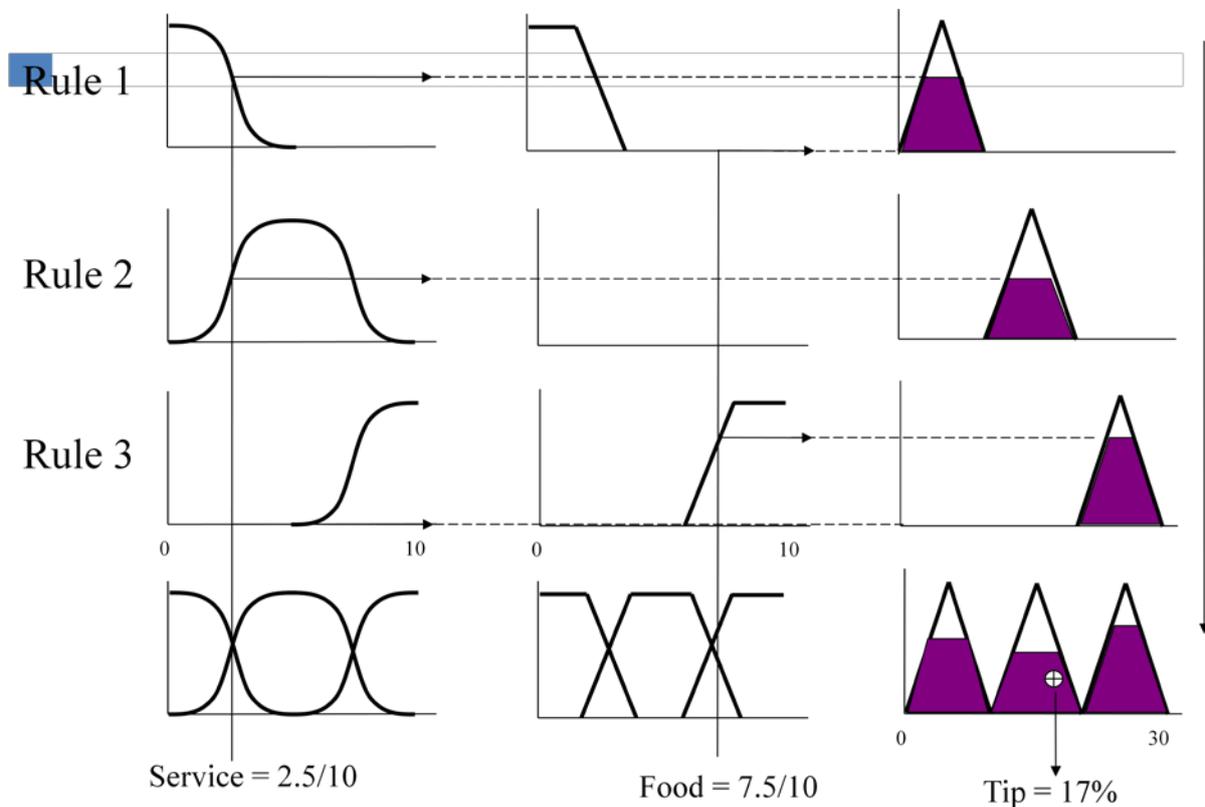
- **Rule 1: IF 'service is poor' OR 'food is bad' THEN 'tip is low'**
- **Rule 2: IF 'service is good' THEN 'tip is average'**
- **Rule 3: IF 'service is excellent' OR 'food is delicious' THEN 'tip is generous'**

Und für die drei Variablen (Service und Essen (Food) als Input, und Trinkgeld (Tip) als Output) werden die im Folgenden dargestellten Mengen benutzt.



Membership functions for tip example

Dieses Beispiel System hat bei einem Input von 2.5 für Service und 7.5 fürs Essen einen Output von 17%. Zuerst wird jede Regel einzeln getestet und wenn vorhanden der Output der Output Menge hinzugefügt. Diese Menge muss dann noch „Defuzzifiziert“ werden.



VIII. Defuzzification

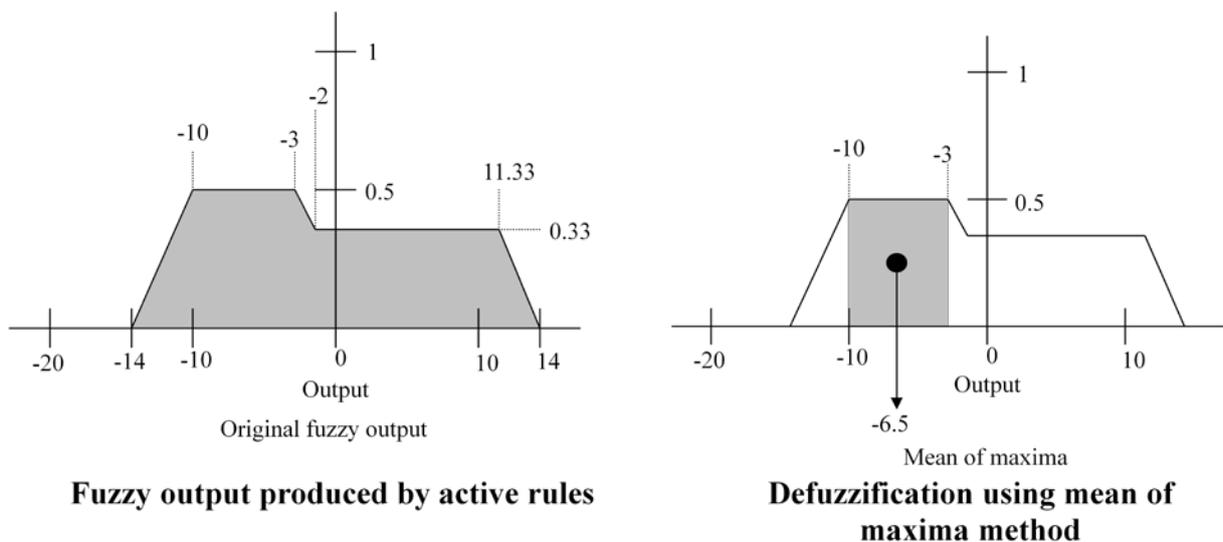
Die Defuzzification sorgt dafür dass der Output des Systems auch weiterverwendet werden kann. Das Fuzzy Control System hat als Output ein Fuzzy Set, also eine Menge, diese ist für die Weiterverwendung aber nicht zu gebrauchen solange diese nicht in ein weiteres Fuzzy System weitergeleitet wird. Deshalb muss diese Menge erst mal Defuzzifiziert werden also in einen konkreten Wert verwandelt werden.

Dafür gibt es drei Standardmethoden:

- Mean of maxima
- Centre of maximums
- Centre of gravity(centroid)

A. Mean of maxima

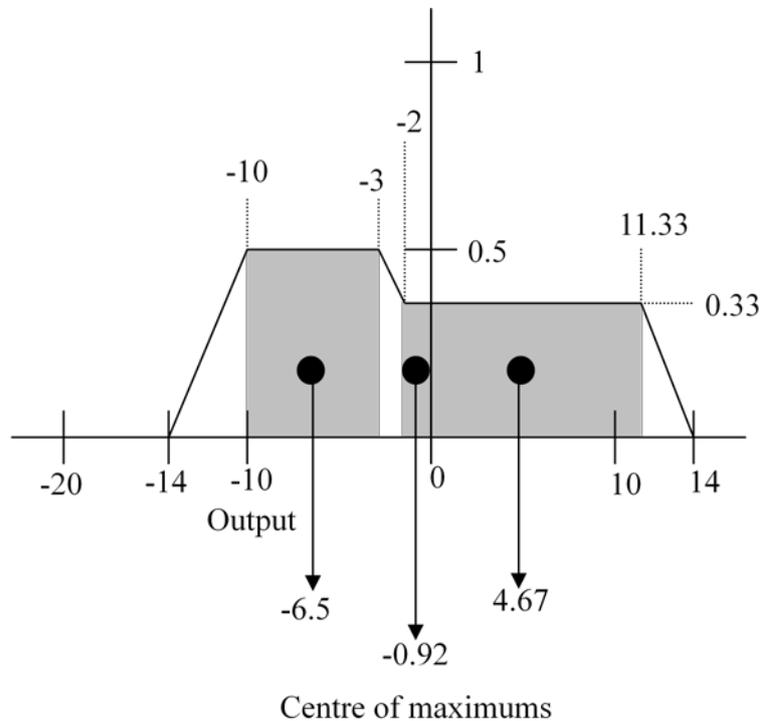
Beim mean of maxima wird zuerst der höchste Wert des Fuzzy Menge bestimmt. Von diesem wird dann der Mittelpunkt berechnet, dazu wird einfach nur der linke und rechte Rand addiert und das ganze durch 2 geteilt. Das Ergebnis ist dann der Mittelpunkt des größten Wertes.



$$\text{Crisp output} = \frac{S_{\max} + S_{\min}}{2} = \frac{-3 + (-10)}{2} = -6.5$$

B. Centre of maximums

Das Centre of maximums ist fast das gleich wie das Mean of maxima der einzige Unterschied ist das es zweimal angewandt wird, einmal auf den höchsten Bereich und einmal auf den zweit höchsten. Von diesen beiden wird dann der Mittelpunkt bestimmt.



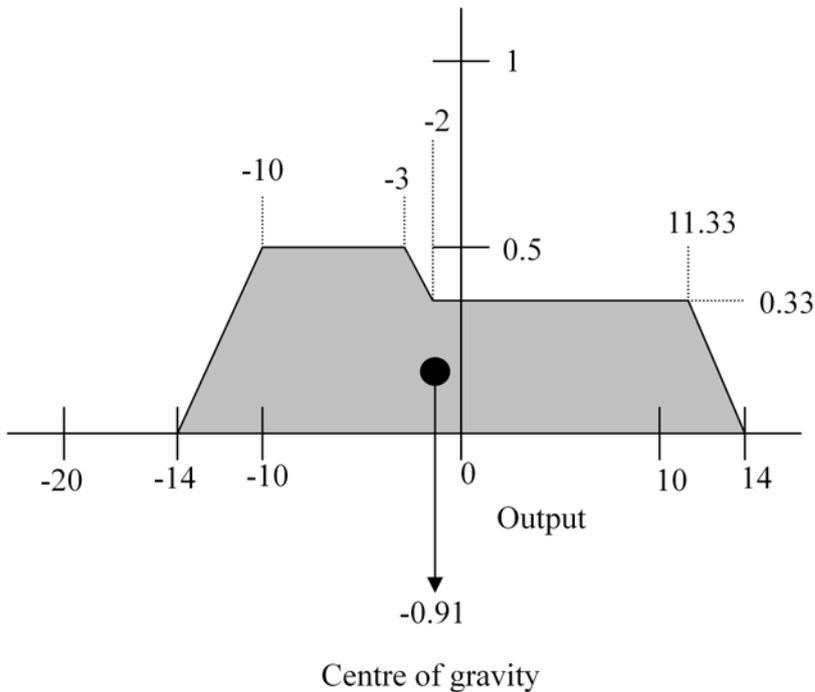
Defuzzification using centre of maximums method

$$\text{Crisp output} = \frac{4.67 + (-6.5)}{2} = -0.92$$

C. Centre of gravity

Bei der Centre of gravity Methode wird, wie der Name schon sagt, der Masseschwerpunkt berechnet. Wie man diesen am besten berechnet ist davon abhängig welche Zugehörigkeitsfunktion gewählt wurde. Den gegebenen Output kann man relativ einfach in Drei und Rechtecke unterteilen. Da dieses einfache Geometrische Körper sind kann man dann mit der Formel μ den Masseschwerpunkt berechnen.

$$\mu = \frac{\sum Ci * Ai}{\sum Ai}$$



Defuzzification using centre of gravity method

$$\text{Crisp output} = \frac{\sum (\text{Moments of individual areas})}{\text{Total area under curve}} = -0.91$$

IX. Kombinatorische Explosion

Wie bei alle Experten Systemen ist die Größe der Wissensdatenbank ein Problem. Die Anzahl der Regeln steigt Exponentiell an wenn für jede Inputvariable alle Regelkombinationen in der Datenbank stehen.

Number of Variables	No of mf's Per Variable	Number of Rules
2	5	$5^2 = 25$
3	5	$5^3 = 125$
4	5	$5^4 = 625$
5	5	$5^5 = 3\ 125$
6	5	$5^6 = 15\ 625$
7	5	$5^7 = 78\ 125$
8	5	$5^8 = 390\ 625$
9	5	$5^9 = 1\ 953\ 125$
10	5	$5^{10} = 9\ 765\ 625$

Bei einem solchen exponentiellen Anstieg wird die Datenbank schnell so groß das sie nicht mehr eingesetzt werden kann. Es gibt jedoch ein paar Möglichkeiten um dem entgegen zu wirken.

Zum einen dadurch dass die Sets einfach kleiner gemacht werden indem Elemente die nicht benötigt werden rausgenommen werden. Dies sollte schon am Anfang gemacht werden weil es dann am einfachsten ist noch etwas so Grundsätzliches es kann sich aber trotzdem lohnen auch hinterher noch einmal zu überprüfen ob jedes Element wirklich benötigt wird um die Datenbank zu verkleinern.

Eine weitere Methode ist die „Comb Methode“. Diese kann aus dem exponentiellen Anstieg einen linearen machen.

Dafür wird die folgende Logische Gleichheit ausgenutzt:

$$((p \wedge q) \Rightarrow r) \iff ((p \Rightarrow r) \vee (q \Rightarrow r))$$

Für eine genauere Erklärung der Comb Method sei auf das Buch „*The Fuzzy Systems Handbook*“ [5] verwiesen.

Im Buch „*Game Programming Gems 2*“ gibt es ein Spiele bezogenes Anwendungsbeispiel von Michael Zarozinski [6].

X. Fuzzy State Machines

„Fuzzy State Machine“ ist eine Erweiterung der normalen State Machines. Diese normalen State Machines wechseln ab einem bestimmten Punkt von einem Zustand in einen anderen, dies führt vor allem bei Spielen zu sehr einfach vorherzusehendem Verhalten. Wenn für sowohl die Inputs als auch die Outputs statt der normalerweise benutzten festen Werte Fuzzy Werte benutzt werden, kann statt der sprunghaften Wechsels von einem Zustand in einen anderen ein fließender und schwieriger vorherzusehender Wechsel erreicht werden.

“Advantage of smooth control and reasoning with degrees of truth which often proves more humanlike” [2]

Zusätzlich zu den In und Outputs können auch die verschiedenen Zustände als Fuzzy-Variablen definiert werden. Dies ist jedoch nur für kleinere State Machines sinnvoll denn es führt sehr schnell zur kombinatorischen Explosion.

XI. Typ-2 Fuzzy Logik

In einer Typ 1 Fuzzy Logik gibt es nur wenige Einstellungsmöglichkeiten.

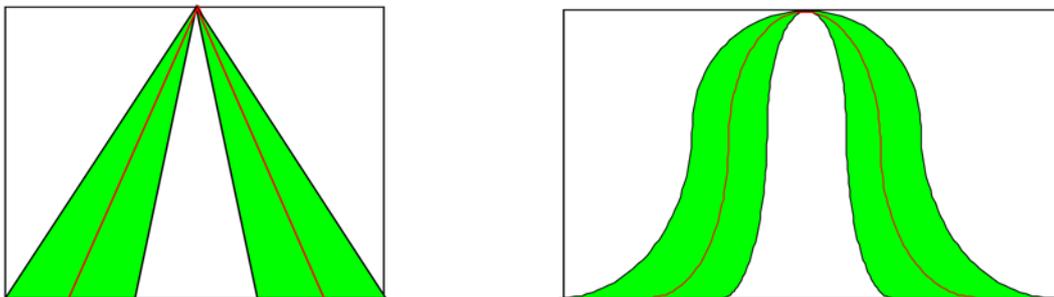
Die meisten Einstellungen werden direkt am Anfang vorgenommen und können später nur schwer wieder geändert werden. Deshalb ist die einzige Möglichkeit den Output des Systems zu ändern das hinzufügen oder entfernen von Regeln.

In der Typ 2 Fuzzy Logik können deshalb zusätzlich noch die Position und die Breite einer Zugehörigkeitsfunktion variieren. Wie sie variieren wird dabei durch weitere Regeln bestimmt.

Zum Beispiel:

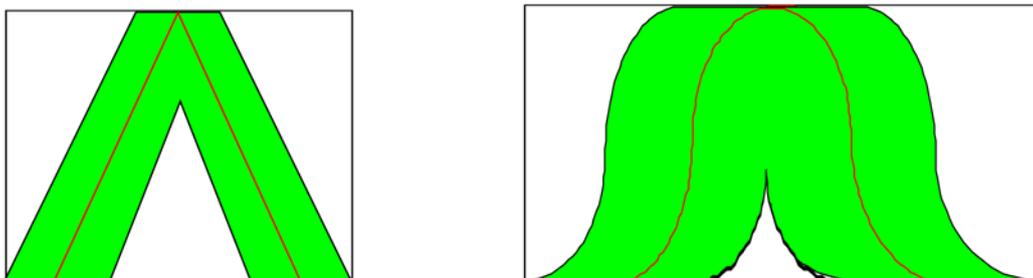
- Rule 1. IF 'x is a' THEN 'Width is Narrow'
- Rule 2. IF 'x is b' THEN 'Width is Medium'
- Rule 3. IF 'x is c' THEN 'Width is Wide'

- Variable Breite



Footprint of uncertainty – varying width

- Position



Footprint of uncertainty – varying position

- Oder aber beides auf einmal (Grafisch leider nur schwer darzustellen)

XII. Referenzen

- 1) Jeff Orkin, *AI Game Programming Wisdom – Vol.2*, Charles River Media, USA, 2003, p101.
 - 2) Alex Champanard, *AI Game Development – Synthetic Creatures with Learning and Reactive Behaviours*, New Riders Publishing, USA, 2004, p541.
 - 3) William van der Sterren, *AI Game Programming Wisdom – Vol.2*, Charles River Media, USA, 2003, pp251-252.
 - 4) Alex Champanard, *AI Game Development – Synthetic Creatures with Learning and Reactive Behaviours*, New Riders Publishing, USA, 2004, p401.
 - 5) William E Combs, *The Fuzzy Systems Handbook 2nd. Ed.*, Academic Press, 1999.
 - 6) Michael Zarozinski, *Game Programming Gems 2*, Charles River Media, USA, pp342-350.
 - 7) type2fuzzylogic.org
-
- Bradley *et al*, *Mechatronics and the design of intelligent machines and systems* — Ch.6.10-6.14
 - J.M. Mendel, *Uncertain Rule-Based Fuzzy Logic: Introduction and New Directions*, Prentice-Hall, 2000. (For information on type-2 fuzzy logic)
 - Matlab Fuzzy Logic Toolbox manual