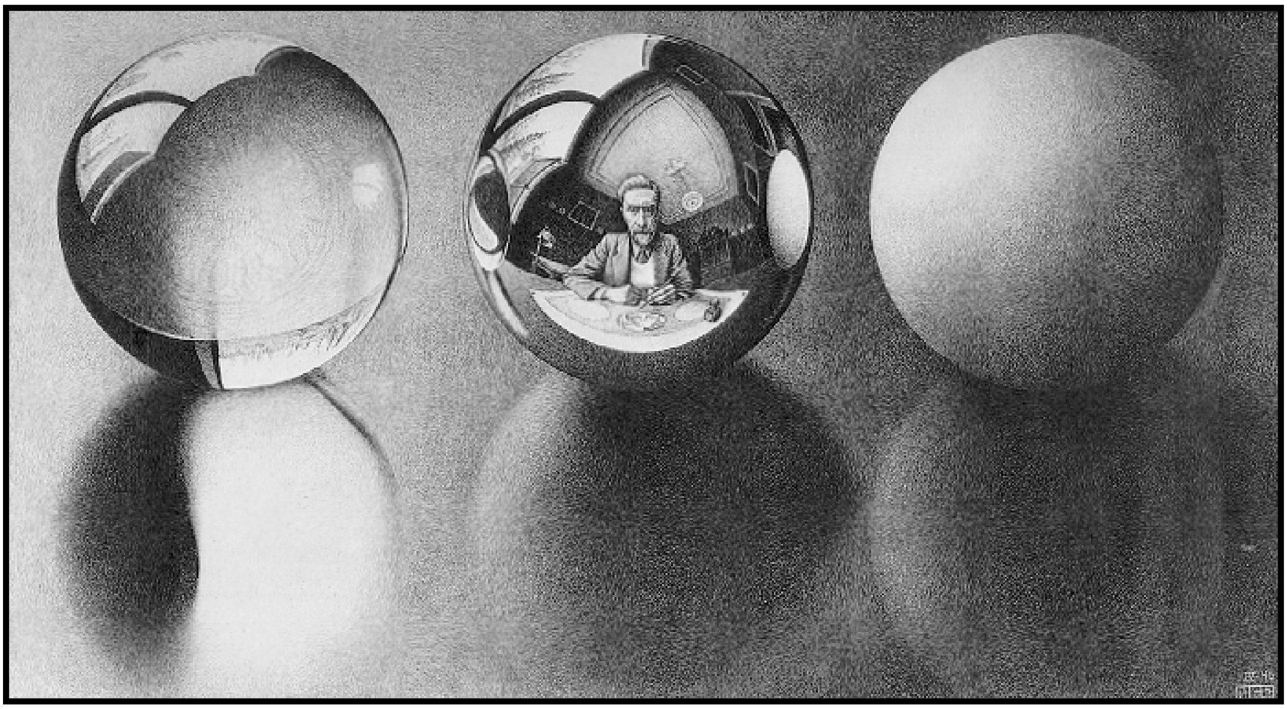
Functie Approximatie
met
Additieve Fuzzy Systemen



Auteur	Dennis Ettes
Examen nr.	139974
Scriptiebegeleider	Jan van den berg

Inhouds opgave

0 Voorwoord

Hier is ze dan, mijn scriptie. Het is niet alleen een afstuiting van een jaar lang zwemmen in de vage logica, maar ook een afsluiting van het leven dat student zijn heet. Een jaar ben ik me nu aan het verdiepen in de fuzzy logica. Een logica die zo vaag is dat iedereen haar vloeiend spreekt, maar niemand haar begrijpt. Dat zal dan ook wel de reden zijn van alle misverstanden in de wereld.
Met deze scriptie ambieer ik echter niet alle problemen uit de wereld te kunnen helpen, maar slechts het oplossen van een simpel probleem als het benaderen van een functie. Doch heb ik me zeer vermaakt met het doen van onderzoek in de theorie der vage systemen en de toepassing hiervan.
Ik wil dit voorwoord verder gebruiken voor het bedanken van iedereen die meegehopen heeft aan het tot stand komen van deze scriptie

- Als eerste uiteraard mijn scriptie begeleider Jan, voor de heftige discussies en de kritische feedback.
- Studiegenoten voor de inhoudelijke discussie.
- Mijn familie en vrienden
- Bløf voor de ondersteunende teksten.
- Iedereen die ik niet heb kunnen uitleggen waar ik mee bezig was, maar toch het geduld op kon brengen om mijn verhaal aan te horen.

Dennis Ettes

Augustus 1998.

1 Inleiding

Deze scriptie gaat over functie approximatie met Additatief Fuzzy Systemen (AFS). Functie approximatie is het bouwen van een functie die een onbekende andere functie benadert. Van de onbekende functie zijn alleen een aantal voorbeeld punten aanwezig. Een additatief fuzzy systeem is een systeem dat op basis van fuzzy sets een functie genereert. Het grote voordeel van het gebruik van een additatief fuzzy systeem is dat de regels, waaruit deze opgebouwd is, onder bepaalde voorwaarden interpreteerbaar zijn. Er is daarom geen sprake is van een black-box model, maar van een white-box model.

Omdat deze scriptie in het Nederlands geschreven is, maar het vakgebied bijna geheel engelstalig is zullen een aantal woorden die in het Engels een speciale betekenis hebben niet vertaald worden. De woorden fuzzy en set zullen bijvoorbeeld niet vertaald worden naar de nederlandse woorden vaag en verzamenling

1.1 Probleemstelling

De probleemstelling in deze scriptie bestaat uit vier vragen

- Hoe wordt er geredeneerd in de fuzzy logica?
- Hoe zien de functies die door een AFS gemaakt zijn eruit?
- Kan een algoritme gemaakt worden dat een AFS genereert dat geen last heeft van de 'curse of dimentionality'?
- Hoe kan een AFS geïnterpreteerd worden en onder welke voorwaarden kan dit?

1.2 Aanpak

Voor het schrijven van deze scriptie zijn de volgende stappen doorlopen:

- Het verkrijgen van informatie over het vakgebied.
- Het maken van de probleemstelling.
- Aanpak van de probleemstellingen
- Schrijven van de scriptie

1.2.1 Het verkrijgen van informatie over het vakgebied.

Informatie over het vakgebied is op twee manieren verkregen. Als eerste door het volgen van het keuzevak 'Fuzzy mathematics' bij de heer Hoogland. Op deze manier is informatie verkregen omtrend de theorie over fuzzy sets en de fuzzy logica. Daarnaast is door het lezen van artikelen en boeken die via mijn scriptiebegeleider verkregen zijn een overzicht van de toepassingen van AFS ontstaan.

1.2.2 Het maken van de probleemstelling

Tijdens het college 'Fuzzy mathmatics' kwam naar boven dat het redeneren dat theoretisch gebruikt wordt bij fuzzy sets niet overeen komt met wat in de praktijk gebruikt wordt. De eerste vraag van de probleemstelling is hoe er geredeneerd wordt in de fuzzy logica. Bij het bekijken van toepassingen van AFS kwam de vraag naar boven hoe functies die gemaakt zijn met een AFS en uitzien. Dit is dan ook de tweede vraag van de probleemstelling. Verder leek het probleem van de 'curse of dimentionality' een probleem dat opgelost diende te worden, waarbij het AFS nog steeds aan de eis van interpretabiliteit voldoet. De derde vraag is dat ook of er een algoritme gemaakt kan worden voor het genereren van een AFS, dat geen last heeft van de ‘curse of dimentionality’ en de vierde vraag is onder welke voorwaarden een AFS geïnterpreteerd kan worden.

1.2.3 Aanpakken van de probleemstelling

Voor het aanpakken van de eerste vraag is in de theorie gedoken en is gekeken hoe de theorie is de praktijk zou werken. Voor het aanpakken van de tweede vraag is onderzocht wat voor invloed de parameters in een AFS hebben op de uiteindelijke functie. Voor het aanpakken van de derde vraag zijn een aantal algoritmen bedacht die fuzzy rules kunnen genereren. Hiervan zijn er twee uitgeprogrammeerd, het fuzzy inductive logic programming algoritme en het fuzzy beslissingsboom algoritme. In eerste instantie is het fuzzy inductive logic programming algoritme uitgewerkt, maar daarmee kon de model complexiteit niet goed gecontroleerd worden. Dus is daarna het fuzzy beslissingsboom algoritme uitgewerkt. De vierde vraag is aangepakt door te proberen de resultaten van het fuzzy beslissingsboom algoritme te interpreteren en te kijken wanneer dit goed of slecht ging.

1.2.4 Schrijven van de scriptie

Tijdens het gehele jaar dat aan deze scriptie is gewerkt zijn de vorderingen in het kort op papier gezet, zodat bij het begin van het schrijven van de scriptie al veel tekst aanwezig was. Het bleek dat de teksten flink uitgebreid dienden te worden om deze leesbaar voor derden te maken. Als goede oefening is samen met Jan van den Berg een artikel[Berg/Ettes] geschreven waarbij we de teksten erg kritisch bekeken hebben. Na alles op een rij gezet te hebben, zijn de ontbrekende delen ingevuld.

1.3 Opbouw scriptie

Deze scriptie is als volgt opgebouwd. Dit is het eerste hoofdstuk met de inleiding. In het tweede hoofdstuk zullen in het kort de belangrijkste eigenschappen van de fuzzy logica en van fuzzy sets behandeld worden. Hierin is geprobeerd om een overzicht te geven van de operatoren die gebruikt worden in de fuzzy logica. Tevens is de basis van het redeneren met fuzzy sets verduidelijkt. In het derde hoofdstuk wordt een bepaalde groep binnen de fuzzy systemen, de additieve fuzzy systemen (AFS) onder de loep genomen. Daarvan is onderzocht wat de representatie mogelijkheden zijn. In het vierde hoofdstuk wordt in gegaan op het leren van Additatieve fuzzy systemen en de problemen die daarbij kunnen ontstaan, zoals de ‘curse of dimentionality’. Het vijfde hoofdstuk gaat over het fuzzy beslissingsboom algoritme dat in het afgelopen jaar ontwikkeld is. Dit is een algoritme dat een fuzzy boom opbouwd en hierbij probeert om het ‘curse of dimentionality’ probleem te voorkomen. In het zesde hoofdstuk worden de resultaten van het fuzzy beslissingsboom algoritme behandeld, waarbij onderandere gekeken is wat de invloed van ruis is op de performance van het algoritme en hoe het resultaat geïnterpreteerd kan worden. In het zevende hoofdstuk staan de conclusies die getrokken worden en de onderwerpen die in de toekomst verder uitgewerkt kunnen worden.

2 Fuzzy logica en fuzzy sets

Fuzzy Logica is de logica van de ‘vage’ of fuzzy begrippen. Deze is ontstaan in de jaren 60 met Lotfi A. Zadeh[Zadeh] als grondlegger. Het is een uitbreiding van de logica, die nodig was omdat bleek dat met de klassieke of crispe logica niet alles beschreven kon worden. In de crispe logica is een bewering altijd of juist of niet juist. In de fuzzy logica kan hiervan afgeweken worden en kunnen beweringen ook voor een deel juist zijn. Dat niet alles met de crispe logica beschreven kan worden komt doordat begrippen in de natuurlijke taal vaak niet absoluut zijn, maar juist vaag of fuzzy. De fuzzy logica verenigt het gebruik van fuzzy begrippen uit de taal, het redenerend vermogen van de logica en tevens het rekenend vermogen van de analyse.
De meeste toepassingen van de fuzzy logica zijn op het gebied van fuzzy control[Kosko]. Dit is het gebruik van eenvoudige fuzzy regels voor het aansturen van apparaten zoals wasmachines, stofzuigers of een Japanse metro. In deze scriptie zal de fuzzy logica gebruikt worden als taal waarmee modellen beschreven worden. Doordat de fuzzy logica dicht tegen de taal aanligt zullen fuzzy modellen doorgaans beter door mensen begrepen kunnen, dan bijvoorbeeld modellen op basis van neurale netwerken[Bishop].
Over de theorie achter de fuzzy logica is heel wat geschreven [Klir],[Hoogland], waarvan de belangrijkste delen voor het begrip van deze scriptie in dit hoofdstuk kort worden uitgelegd. In paragraaf 2.1 zal worden ingegaan op de fuzzy proposities en de operaties die daarop uitgevoerd kunnen worden. In paragraaf 2.2 zullen de fuzzy predicaten en de fuzzy sets worden beschreven. Er wordt geprobeerd om voordat iets in fuzzy wordt uitgelegd eerst de crispe tegenhanger te behandelen. Ook wordt alles dat voor proposities proposities uitgelegd wordt ook voor predicaten of sets uitgelegd, zodat het verband hiertussen goed duidelijk wordt. Niet alle onderdelen waren uitgewerkt in de gebruikte literatuur, dus zijn aanvullingen gezocht.

2.1 Fuzzy proposities

In deze paragraaf zullen de fuzzy proposities uitgelegd worden. Omdat de fuzzy proposities ontstaan zijn door de crispe proposities uit te bereiden, zullen de crispe proposities eerst behandeld worden en daarna de fuzzy proposities. Daarna zullen de operatoren op proposities en het redeneren met proposities aan bod komen.

2.1.1 Crispe proposities

Een crispe propositie is een bewering of uitspraak die of juist of onjuist is. De juistheidswaarde van een crispe propositie wordt aangeduid met de griekse letter τ. $\tau(p)$ is de juistheidwaarde van de crispe propositie p en heeft de waarde 1 als p juist is en de waarde 0 als p onjuist is. Er geldt dus dat $\tau(p) \in \{0,1\}$.Een voorbeeld van een crispe propositie in bijvoorbeeld de uitspraak: ‘Het is warm weer’. Als deze uitpraak waar is geldt τ(‘het is warm weer’)=1 en als deze uitspraak niet waar is geldt τ(‘het is warm weer’)=0.

2.1.2 Fuzzy propositie

Een fuzzy propositie is een uitspraak die niet alleen juist of onjuist is, maar ook een uitspraak die in een zekere mate juist kan zijn. Ook van een fuzzy propositie wordt de juistheidswaarde aangegeven met de griekse letter τ. $\tau(p)$ is de juistheidswaarde van de fuzzy propositie p en heeft een waarde in het interval [0..1]. Er geldt dus dat $\tau(p) \in [0..1]$. Het voorbeeld van een crispe propositie kan ook gezien worden als een fuzzy propositie. In dat geval kan τ('het is warm weer') behalve de waarden 0 en 1 ook waarden er tussenin aannemen. Als het behoorlijk warm weer is dan kan bijvoorbeeld gelden dat τ('het is warm weer')=0.8.

2.1.3 Operaties op crispe proposities

Een crispe operatie is een bewerking met een operator op één of meer crispe proposities, waarmee een nieuwe crispe propositie gemaakt wordt. De juistheidswaarde van de zo ontstane propositie hangt af van de juistheidswaarden van de gebruikte proposities. Op crispe propositie zijn er drie belangrijke operatoren, de 'ontkenning', de 'conjunctie' en 'disjunctie'.
De ontkenning is een operator op één propositie. De onkenning van de propopsitie p wordt aangeduid met ¬p (spreek uit 'niet p'). De juistheidswaarde τ(¬p) is 1 als p niet juist is en is 0 als p wel juist is.
De conjunctie is een operator op twee proposities. De conjunctie van de proposities p en q wordt aangeduid met p∧q (spreek uit 'p en q'). De juistheidswaarde τ(p∧q) is 1 als p en q allebei juist zijn en 0 anderzijds.
De disjunctie is ook een operator op twee proposities en de conjuctie van de proposities p en q wordt aangeduid met p∨q (spreek uit 'p of q'). De juistheidswaarde τ(p∨q) is 0 als p en q allebei onjuist zijn en 1 anderzijds.

2.1.4 Operaties op fuzzy proposities

Een fuzzy operatie is een bewerking met een fuzzy operator op één of meer fuzzy proposities, waarmee een nieuwe fuzzy propositie gemaakt wordt. Bij het ontwikkelen van de fuzzy logica is men opzoek gegaan naar fuzzy varianten van de drie belangrijkste crispe operatoren. Deze varianten zouden generalisaties moeten zijn van de crispe operatoren en daarom de crispe operatoren moeten bevatten. Dit betekent dat als een operator op de fuzzy proposities wordt uitgevoerd, die de juistheidswaarden 0 of 1 hebben, dat de juistheidswaarde van de nieuwe propositie gelijk moet zijn aan die van de crispe variant. De juistheidwaarden van operaties op proposities die niet alleen juistheidswaarden 0 of 1 hebben moest dus nog bedacht worden.
Over de fuzzy ontkenning waren de fuzzy logici het snel eens. De juistheidswaarde τ(¬p) is gelijk aan 1-τ(p).
Over de juiste tegenhanger van de con-en disjunctie zijn de fuzzy logici het echter niet eens. Dit komt doordat er meerdere generalisties mogelijk zijn die niet zomaar verworpen kunnen worden. Het probleem wordt dan dat er gezocht kan worden naar een operator die voor elke situatie waarin deze gebruikt wordt overeenkomt met wat 'intuïtief juist' is. Maar er kan niet één operator gevonden worden die voor elke betekenis of semantiek van de proposities ook de juiste is. Daarom zal afhankelijk van de semantiek van de proposities gezocht moeten worden naar een voor die semantiek juiste operator. Verdere uitleg hierover volgt in het onderdeeel semantiek en operatoren.
Hier zullen de meest gebruikte con- en disjuncties gegeven worden.

	Standaard	Algebraïsche	Bounded
Ontkenning τ(¬p)	1-τ(p)	1-τ(p)	1-τ(p)
Conjunctie τ(p∧q)	Min(τ(p),τ(q))	τ(p)*τ(q)	τ(p)*τ(q)
Disjunctie τ(p∨q)	Max(τ(p),τ(q))	τ(p)+τ(q) -τ(p)*τ(q)	Min(τ(p)+τ(q),1)

Tabel 2.1 **Onkenning, con- en disjuncties voor fuzzy proposities.**

Er zijn een aantal eigenschappen die gelden voor operaties op crispe proposities. Deze eigenschappen gelden echter niet voor alle fuzzy operatoren. Hier een overzicht.

Naam eigenschap	Formule eigenschap	Niet geldig voor
Involutie	$\tau(\neg(\neg p))=\tau(p)$	
Commutativiteit	$\tau(p\wedge q)=\tau(q\wedge p)$ $\tau(p\vee q)=\tau(p\vee p)$	
Associatititeit	$\tau((p\wedge q)\wedge r)=\tau(p\wedge(q\wedge r))$ $\tau((p\vee q)\vee r)=\tau(p\vee(q\vee r))$	
Distributiviteit	$\tau(p\wedge(q\vee r))=\tau((p\vee q)\wedge(p\vee r))$ $\tau(p\vee(q\wedge r))=\tau((p\wedge q)\vee(p\wedge r))$	Algebraïsche/Bounded
Idempotentie	$\tau(p\wedge p)=\tau(p)$ $\tau(p\vee p)=\tau(p)$	Algebraïsche/Bounded
Absorptie	$\tau(p\wedge(p\vee q))=\tau(p)$ $\tau(p\vee(p\wedge q))=\tau(p)$	Algebraïsche/Bounded
Identiteit	$\tau(p\wedge \text{true})=\tau(p)$ $\tau(p\vee \text{false})=\tau(p)$	
Dominantie	$\tau(p\wedge \text{false})=0$ $\tau(p\wedge \text{true})=1$	
Complementariteit	$\tau(\neg \text{false})=1$ $\tau(\neg \text{true})=0$	
Wetten van de Morgan	$\tau(\neg(p\wedge q)=\tau((\neg p)\vee(\neg q))$ $\tau(\neg(p\vee q)=\tau((\neg p)\wedge(\neg q))$	Bounded
Law of excluded middle	$\tau(p\vee(\neg p))=1$	Standaard/Algebraïsche/ Bounded
Law of contradiction	$\tau(p\wedge(\neg p))=0$	Standaard/Algebraïsche/ Bounded

Tabel 2.2 **Eigenschappen van de con- en disjuncties**

De laatste twee eigenschappen, de Law of excluded middle en de Law of contradiction, gelden niet voor de Standaard, Algebraïsche en Bounded con- en disjunctie. Voor een andere keuze van de operatoren conjunctie, disjunctie en ontkenning, zouden deze regels wel kunnen gelden.
Doordat de overige eigenschappen wel gelden bij gebruik van de standaard con- en disjunctie worden deze in de theorie het meest gebruikt. In de praktijk wordt er echter ook vaak gekozen voor de algebraïsche con- en disjunctie of de bounded disjunctie.

2.1.5 Semantiek en operatoren

In de literatuur is nog weinig geschreven over de relatie tussen de semantiek en het gebruik van fuzzy operatoren. Daarom wordt hier een aanzet gegeven voor de discussie over de vraag welke operator wanneer gebruikt moet worden.
De toepasbaarheid van een fuzzy operator is afhankelijk van de situatie waarin deze gebruikt wordt en dus van de semantiek van de propositie en operator. De semantiek van een propositie is de betekenis van de propositie. De semantiek van een operator is de betekenis die de operatie heeft die uitgevoerd wordt. In de klassieke logica is de keuze van de operator eenduidig en kan als de vertaling naar de logica eenmaal gemaakt is, verder van de semantiek geabstraheerd worden. In de fuzzy logica kunnen

echter niet eenduidige operatoren gevonden worden die in alle gevallen zullen werken. Daarom hangt de keuze van de operator af van het toepassings domein, ofwel de semantiek van de proposities en de gewenste operatie. Hoe de keuze van de operator afhangt van de semantiek zal in de volgende voorbeelden verduidelijkt worden.
Zij p, q en r fuzzy proposities. p is de uitspraak ‘het is tropisch warm vandaag’, q is de uitspraak ‘het is vandaag volle maan’ en r is de uitspraak ‘het is benauwd vandaag’. Neem nu aan dat geldt $\tau(p) = 0.5$, $\tau(q) = 0.5$ en $\tau(r) = 0.5$. Er wordt gekeken naar de juistheidswaarde van $p \wedge q$ en $p \wedge r$. Er moet nu naar de semantiek van de proposities en de operator gekeken worden. Omdat p en q ongecorreleerd zijn - dat betekent dat dat als p waar is dat niets zegt over de waarheids waarde van q - zal $\tau(p \wedge q)$ kleiner zijn dan 0.5 en bijvoorbeeld $\tau(p \wedge q)=0.25$. p en r zijn wel gecorreleerd, omdat als het tropisch is het meestal ook benauwd is. $\tau(p \wedge r)$ zal nu ongeveer 0.5 zijn, omdat p en r elkaar niet kunnen verzwakken. In het geval van $p \wedge q$ moet dus algebraïsche conjunctie gebruikt worden en in het geval van $p \wedge r$ de standaard conjunctie. Nu word gekeken naar de juistheidswaarde van $p \vee q$ en $p \vee r$, dus of de uitspraak ‘het is tropisch warm vandaag of het is volle maan’ en de uitspraak ‘het is tropische warm of benauwd vandaag’.Omdat p en q ongecorreleerd zijn zullen ze elkaar versterken en zal ongeveer gelden dat $\tau(p \vee q)=0.75$.Omdat p en r positief gecorreleerd zijn, zullen ze elkaar niet versterken en geldt dat $\tau(p \vee r)=0.5$. In het geval van $p \vee q$ moet dus de algebraïsche disjunctie gebruikt worden en in het geval van $p \vee r$ moet de standaard disjunctie gebruikt worden. De bounded disjunctie disjunctie wordt gebruikt als de proposities negatief gecorreleerd zijn. Geconcludeerd kan worden dat bij het kiezen van een operator gekeken moet worden in welke mate gewenst is dat de proposities elkaar versterken of verzwakken.

2.1.6 Modifiers

Andere veel voorkomende operatoren zijn de modifiers. Deze operatoren zorgen voor een versterking of een verzwakking van een propositie. Deze modifiers worden geschreven als p^{α} waarbij $\tau(p^{\alpha})= \tau(p)^{\alpha}$. Als α groter is dan 1 is het een versterking en kan het gebruikt worden als een bijvoegelijk naamwoord ‘zeer’ of ‘erg’. Als α kleiner is dan 1, is het een verzwakking en betekent het b.v. ‘redelijk’ of ‘matig’.

2.1.7 Crispe Implicatie

De implicatie is een operator, op twee proposities en wordt weergegeven als $p \to q$ (spreek uit ‘p impliceert q’). De juistheidswaarde $\tau(p \to q)$ is 1 als q juist is of p onjuist is en 0 anders. Met deze operator worden vaak regels gedefinieerd, waarmee geredeneerd kan worden. In de volgende tabel staat de waarheids tabel.

$\tau(p)$	$\tau(q)$	$\tau(p \to q)$
1	1	1
1	0	0
0	1	1
0	0	1

Tabel 2.3 **Waarheidswaarden van de implicatie**

2.1.8 Fuzzy implicatie

De fuzzy implicatie is een fuzzy operator op twee fuzzy proposities en wordt weergegeven als $p \rightarrow q$. Net als bij de con- en disjunctie zijn de fuzzy logici het ook niet eens over wat de juistheids waarde van de implicatie zou moeten zijn, hoewel de Gödel implicatie theoretisch het meest gebruikt wordt[Hoogland]. Er wordt geprobeerd om de implicatie zo te definiëren dat de crispe implicatie er in besloten zit. De implictie is dus geheel onjuist als $\tau(q)=0$ en $\tau(p)=1$ en geheel juist als $\tau(p)=1$ en $\tau(q)=0$, als $\tau(p)=0$ en $\tau(q)=0$ en als $\tau(p)=1$ en $\tau(q)=1$.

Crispe implicatie	$\tau(p \rightarrow q)= 1$ als $\tau(p)<=\tau(q)$, 0 anderszijds
Gödel implicatie	$\tau(p \rightarrow q)= 1$ als $\tau(p)<=\tau(q)$, $\tau(q)$ anderszijds
Lukasiewics implicatie	$\tau(p \rightarrow q)= \min(1,1-\tau(p)+\tau(q))$
Kleene-Dienes implicatie	$\tau(p \rightarrow q)= \max(1-\tau(p),\tau(q))$

Tabel 2.4 **De verschillende fuzzy implicaties**

2.1.9 Crisp redeneren

Er kan op veel manieren geredeneerd worden, hier zal één methode uitgewerkt worden. Als p juist is en $p \rightarrow q$ juist is, dan wordt afgeleid dat q ook juist is.
In de volgende tabel wordt laten zien wat de juistheidswaarde van q is gegeven de juistheidswaarden van $p \rightarrow q$ en p.

$\tau(p \rightarrow q)$	$\tau(p)$	$\tau(q)$
1	1	1
1	0	0 of 1
0	1	0
0	0	Tegenspraak

Tabel 2.5 **Juistheids waarden**

Uit tabel 2.5 kan gehaald worden dat als $\tau(p)=1$ en $\tau(p \rightarrow q)=1$, dat dan afgeleid kan worden dat $\tau(q)=1$. Deze regel kan bewezen worden met behulp van het volgende bewijs uit het ongereimde.

Neem aan dat geldt $\tau(p \rightarrow q)=1$ en $\tau(p)=1$. Stel nu $\tau(q)=0$. Als $\tau(p)=1$ en $\tau(q)=0$, dan geldt $\tau(p \rightarrow q)=0$. Dit is in tegenspraak met de aannamens, dus geldt niet dat $\tau(q)=0$. Conclusie $\tau(q)=1$.

Op de zelfde manier kan de derde regel bewezen worden. Bij de tweede regel kan niets bewezen worden, dus is $\tau(q)=0$ of 1. Bij de vierde regels zijn de aannamens met elkaar in tegenspraak, dus kan er helemaal niets geconcludeerd worden. Er moet dus gezorgt worden dat voordat begonnen wordt met conclusies trekken dat de aannames niet met elkaar in tegenstrijd zijn. Om te voorkomen dat de aannames om de reden van regel 4 met elkaar in tegenstrijd zijn wordt vaak alleen aangenomen dat $\tau(p \rightarrow q)=1$ en niet dat $\tau(p \rightarrow q)=0$. In dat geval kan eenvoudig geconcludeerd worden dat geldt $\tau(q) \geq \tau(p \wedge (p \rightarrow q))$. In de programeertaal Prolog is zo'n restrictie opgelegd en zijn de clauses altijd in de vorm $q \leftarrow p_1 \wedge p_2 \wedge \ldots \wedge p_n$

2.1.10 Fuzzy redeneren

Zoals er met crispe proposities geredeneerd kan worden, wordt ook geprobeerd te redeneren met fuzzy proposities. Dit levert echter een aantal proplemen op, doordat niet aangenomen kan worden dat geldt dat $\tau(p \rightarrow q)=1$, kunnen de aannamens met elkaar in tegenspraak zijn. In de theorie wordt gebruik gemaakt van een andere methodiek[Klir].

Een regel van de vorm $p \rightarrow q$ is opgebouwd uit de twee proposities p en q, die samen aangeven in welke mate $p \rightarrow q$ juist is. Als de Gödel implicatie gebruikt wordt dan geldt dat $\tau(p \rightarrow q)= 1$ als $\tau(p)<=\tau(q)$ en $\tau(q)$ anderszijds. Daarnaast zijn er de proposities p' en q'. p' is een waarneming van de propositie die p voorstelt. En q' is de conclusie die getrokken wordt over de propositie die door q vertegewoordigd wordt. Op basis van de juistheidswaarden van p' en $p \rightarrow q$ wordt dan q' op de volgende manier afgeleid:

$\tau(q') \geq \min(\tau(p'), \tau(p \rightarrow q))$

Deze redeneer techniek werkt echter niet in de praktijk en daarom wordt meestal gebruik gemaakt van

$\tau(q') \geq \tau(p' \wedge p \wedge q)$ die in praktijk wel blijkt te werken. Waarom dit beter werkt wordt in de volgende paragraaf bij redeneren met fuzzy sets uitgelegd.

2.2 Fuzzy predicaten en fuzzy sets

In deze paragraaf worden de crispe predicaten en crispe sets behandeld worden. Waarna de fuzzy predicaten en fuzzy sets bekenen worden. Hiervan zullen de operatoren en het redeneren aan bod komen.

2.2.1 Crispe predicaten

Een predicaat is een uitspraak of een element uit het universum waarop deze gedefinieerd is wel of niet een bepaalde eigenschap heeft. Een predicaat heeft een juistheidsfunktie $\tau_P(x)$, die voor elke x aangeeft of deze een eigenschap P heeft. $\tau_P(x)$ is 1 als x de eigenschap behord bij predicaat P heeft en 0 anderzijds.

2.2.2 Fuzzy predicaten

Een fuzzy predicaat is een uitspraak over de mate waarin een element een bepaalde eigenschap heeft. Het fuzzy predicaat heeft een juistheids funktie $\tau_P(x)$, die voor elke x aangeeft in welke mate deze de eigenschap heeft behorende bij predicaat P. Het bereik van de juistheids functie is [0..1].

2.2.3 Crispe sets

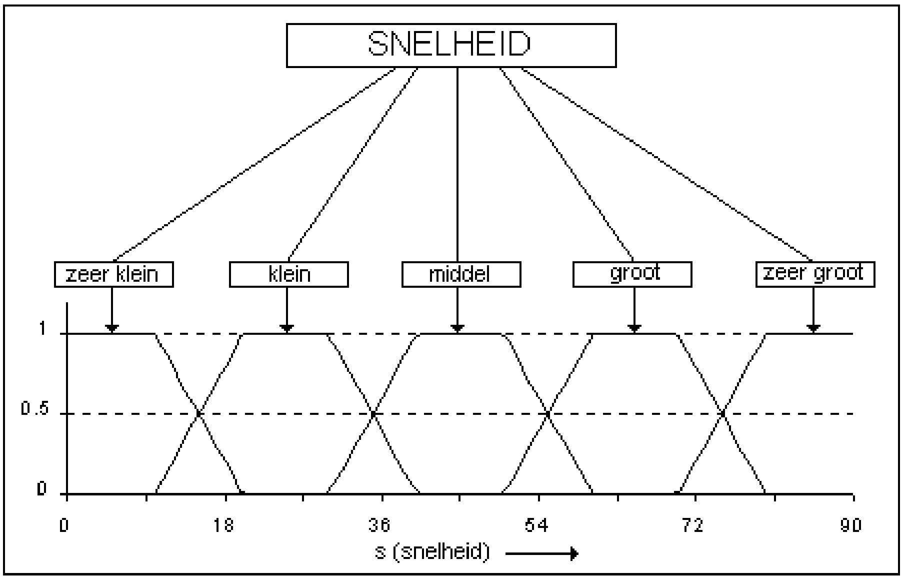
Een crispe set is een verzameling elementen uit het universum. Een set heeft een lidmaatschapsfunktie μ. $\mu_A(x)$ is 1 als het element x in set A zit en 0 anders. Crispe sets en crispe predicaten zijn isomorf, dit betekent dat er een 1 op 1 verband tussen gedefinieerd kan worden. Een crispe set A kan b.v. gedefinieerd worden zodat deze alleen die elementen bevatten met de eigenschap P. En andersom kan een predicaat P gedefinieerd worden met de eigenschap dat de elementen juistheidswaarde 1 hebben als deze in set A zitten. In dit geval geldt voor alle x dat $\tau_P(x) = \mu_A(x)$. Er zijn twee bijzondere crispe sets, namelijk de set X is de set die alle elementen bevat en de set $\emptyset$ is de set die geen enkel element bevat.

2.2.4 Fuzzy sets

De fuzzy set is de bouwsteen van elk fuzzy systeem. Het is een verzameling waarvan elk element van het universum tot een bepaalde hoogte lid van kan zijn. De lidmaatschapswaarde van een element x in set A is de funktie $\mu_A(x)$ en heeft een bereik van [0..1]. $\mu_A(x)$ wordt ook wel als A(x) geschreven. Fuzzy sets en fuzzy predicaten zijn isomorf. Een fuzzy set A kan b.v. alle elementen tot een zelfde hoogte bevatten, als ze de eigenschap P tot een bepaalde hoogte hebben. En andersom, kan predicaat P gedefinieerd zijn dat een element de zelfde mate de eigenschap P heeft als deze lid is van set A. Dan geldt dat $\tau_P(x) = \mu_A(x)$. Omdat fuzzy sets en fuzzy proposities isomorf zijn, gelden alle eigenschappen van fuzzy sets ook voor fuzzy proposities en andersom. Hier zal verder alleen met fuzzy sets gewerkt worden. De tegenhangers van de crispe X en $\varnothing$ zijn de fuzzy X en $\varnothing$, met $\mu_X(x)=1$ en $\mu_0(x)=0$. X is dus de set waar elk element volledig in zit en $\varnothing$ is de set waar alle sets helemaal niet in zitten.

2.2.5 Linguïstische variabele

Een linguïstische variable bestaat uit een aantal linguïstische waarden gedefinieerd op een basis variabele. Deze linguïstische waarden zijn gedefinieerd als fuzzy sets op de basis variabele en hebben een semantische of linguïstische betekenis, zoals bijvoorbeeld klein, middel en groot. Bij fuzzy systemen wordt vaak gebruik gemaakt van linguïstische variabelen, omdat deze begrippen makkelijk in taal om te zetten zijn en daardoor makkelijk zijn te interpreteren. In figuur 2.1 staat een voorbeeld van een linguïstische variabele snelheid, met de linguïstische waarden zeer klein, klein, middel, groot en zeer groot.



Figuur 2.1 **De linguïstische variabele SNELHEID.**

2.2.6 Operaties op crispe sets

Als er operatie op een crispe set wordt uitgevoerd ontstaat er weer een crispe set op het zelfde domein. Er zij drie belangrijke operaties op crispe sets, dat zijn het complement, de doorsnede en de vereniging. Deze operaties zijn de tegenhangers van de ontkenning, de con- en disjunctie bij proposities. Het complement is een operatie op één set, het complement van een set A, $\overline{A}$(spreek uit ‘A complement’) bevat alleen die elementen die de set A niet bevat. De doorsnede is een operatie op twee sets, de doorsnede van set A en B, $A \cap B$ (spreek uit ‘A doorsnede B’) bevat die elementen die zowel in A als in B zitten. De vereniging is ook een operatie op twee sets. De vereniging van set A en B, $A \cup B$ (spreek uit ‘A verenigd B’) bevat de elementen die of in set A of in set B zitten.

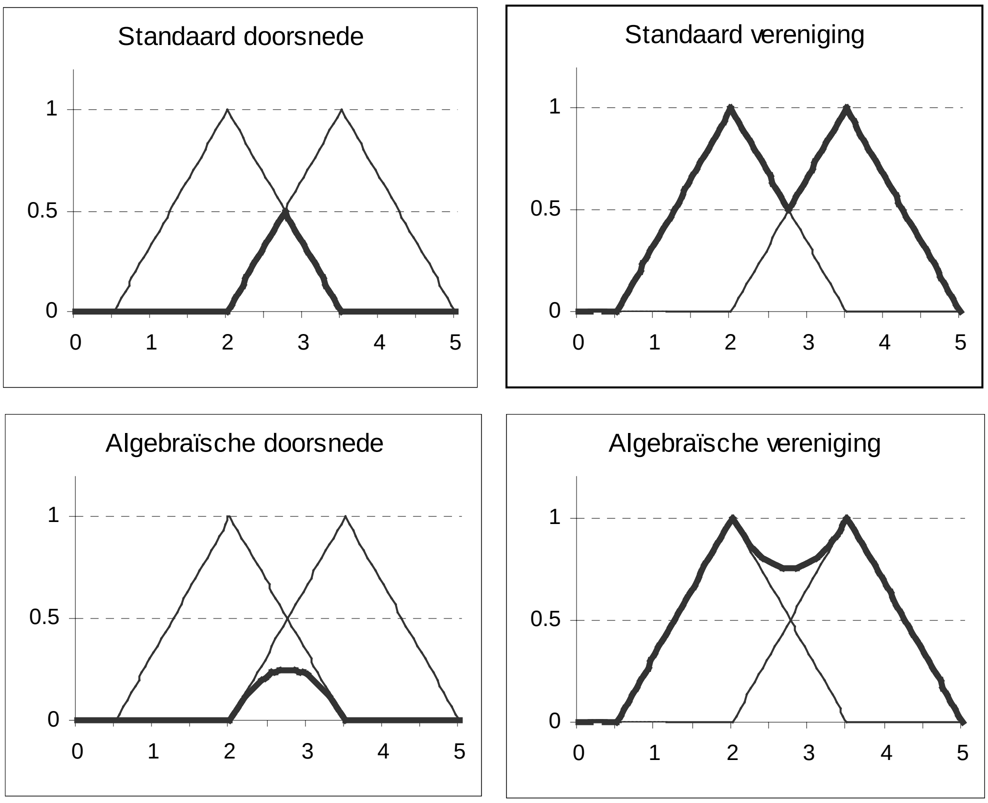
2.2.7 Operaties op fuzzy sets

Net als bij de operaties bij de fuzzy proposities, zijn bij de operatie op fuzzy sets dezelfde problemen. Maar de oplossingen die voor proposities zijn aangedragen, kunnen ook bij de fuzzy sets gebruikt worden. De drie belangrijkste operaties, het complement, de doorsnede en de verzameling worden dan ook analoog gedefinieerd. Ook hier bestaan er drie versies die veel gebruikt worden.

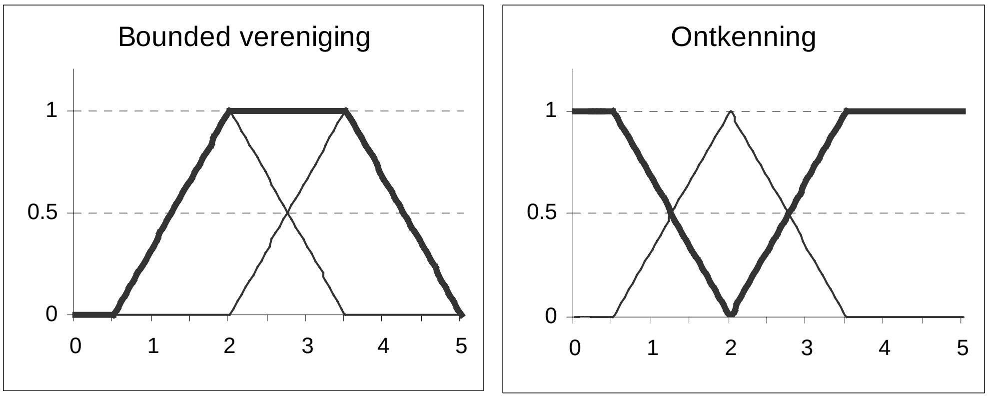
	Standaard	Algebraïsche	Bounded
Complement $\mu_{\bar{A}}(x)$	$1-\mu_A(x)$	$1-\mu_A(x)$	$1-\mu_A(x)$
Doorsnede $\mu_{A \cap B}(x)$	$Min(\mu_A(x), \mu_B(x))$	$\mu_A(x)*\mu_B(x)$	$\mu_A(x)*\mu_B(x)$
Vereniging $\mu_{A \cup B}(x)$	$Max(\mu_A(x),\mu_B(x))$	$\mu_A(x)+\mu_B(x) - \mu_A(x)*\mu_B(x)$	$Min(\mu_A(x)+\mu_B(x),1)$

Tabel 2.5 **Complement, doorsnede en vereniging voor fuzzy sets**

In figuur 2.2 t/m 2/7 staan voorbeelden van de verschillende doorsnedes, verenigingen en de onkennig. De dikke lijn is het resultaat van de operatie.



Figuur 2.2 t/m 2.5 **De verschilende operaties op fuzzy sets.**



Figuur 2.6 t/m 2.7 **De verschilende operaties op fuzzy sets.**

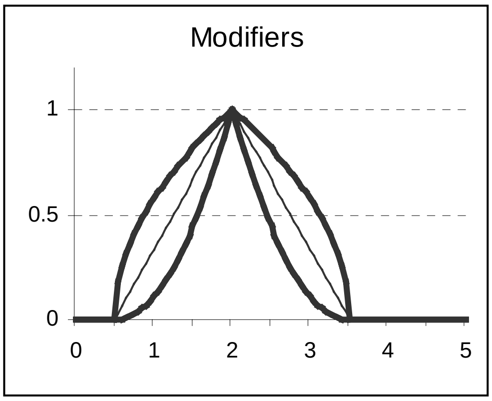
Voor de operaties op crispe set gelden een aantal eigenschappen. Zoals eerder bij de fuzzy proposities gelden deze eigenschappen ook niet voor alle operaties op fuzzy sets. Hier weer een overzicht.

Eigenschap		Niet geldig voor
Involutie	$\overline{\overline{A}} = A$	
Commutativiteit	$A \cap B = B \cap A$ $A \cup B = B \cup A$	
Associatititeit	$(A \cap B) \cap C = A \cap (B \cap C)$ $(A \cup B) \cup C = A \cup (B \cup C)$	
Distributiviteit	$A \cap (B \cup C) = (A \cup B) \cap (A \cup C)$ $A \cup (B \cap C) = (A \cap B) \cup (A \cap C)$	Algebraïsche/Bounded
Idempotentie	$A \cap A = A$ $A \cup A = A$	Algebraïsche/Bounded
Absorptie	$A \cap (A \cup B) = A$ $A \cup (A \cap B) = A$	Algebraïsche/Bounded
Identiteit	$A \cap X = A$ $A \cup \varnothing = A$	
Dominantie	$A \cap \varnothing = \varnothing$ $A \cup X = X$	
Complementariteit	$\overline{\varnothing} = X$ $\overline{X} = \varnothing$	
Wetten van de Morgan	$\overline{A \cap B} = \overline{A} \cup \overline{B}$ $\overline{A \cup B} = \overline{A} \cap \overline{B}$	Bounded
Law of excluded middle	$A \cup \overline{A} = X$	Standaard/Algebraïsche/ Bounded
Law of contradiction	$A \cap \overline{A} = \varnothing$	Standaard/Algebraïsche/ Bounded

Tabel 2.6 **Eigenschappen voor verschillenden operatoren**

2.2.8 Modifiers op fuzzy sets

Andere veel voorkomende operatoren zijn de modifiers. Deze operatoren zorgen voor een versterking of een verzwakking van een fuzzy set. Deze modifiers worden geschreven als A^{α} waarbij $A^{\alpha}(x)= A(x)^{\alpha}$. Als α groter is dan 1 is het een versterking en kan het gebruikt worden als een bijvoeglijk naamwoord ‘zeer’ of ‘erg’. Als α kleiner is dan 1, is het een verzwakking en betekent het b.v. ‘redelijk’ of ‘matig’. In figuur 2.8 is het resultaal te zien van twee modifiers met een α van respectievelijk 0.5 en 2.



Figuur 2.8 **Voorbeeld van resultaat modifiers**

2.2.9 Carthesische operatoren

De carthesische operatoren worden gebruikt voor het uitvoeren van operaties op fuzzy sets die op verschillende domeinen zijn gedefinieerd. Zij A en B fuzzy sets met $A\subseteq X$ en $B\subseteq Y$. In tabel 2.7 staan een aantal carthesische operatoren.

Carthesische doorsnede	$A \underset{\times}{\cap} B(x,y) = \min(A(x), B(y))$
Carthesisch product	$A \underset{\times}{\cap} B(x,y) = A(x)B(y)$
Carthesische vereniging	$A \underset{\times}{\cup} B(x,y) = \max(A(x), B(y))$
Carthesische som	$A \underset{\times}{\cup} B(x,y) = A(x) + B(y)$

Tabel 2.7 **Verschillende carthesische operatoren.**

2.2.10 Crispe implicatie

Zoals er een implictatie gedefinieerd is op proposities, is er er ook een implicatie gedefinieerd op crispe sets. Zij $A\subseteq X$ en $B\subseteq Y$, dan is $A\rightarrow B$ een set op het domein $X\times Y$, waarin alle paren (x,y) zitten waarvoor geldt dat $y\in B$ of $x\notin A$.

2.2.11 Fuzzy implicatie

Bij de fuzzy implictaie voor sets is er geen standaard die altijd gebruikt wordt. Hier zijn de meest voorkomende.

Crispe implicatie	$(A\rightarrow B)(x,y)$ = 1 als A(x)<=B(y), 0 anderszijds
Gödel implicatie	$(A\rightarrow B)(x,y)$ = 1 als A(x)<=B(y), B(y) anderszijds
Lukasiewics implicatie	$(A\rightarrow B)(x,y) = \min(1,1-A(x)+B(y))$
Kleene-Dienes implicatie	$(A\rightarrow B)(x,y) = \max(1-A(x),B(y))$

Tabel 2.8 **Verschillenden implicaties**

2.2.12 Crisp redenenen

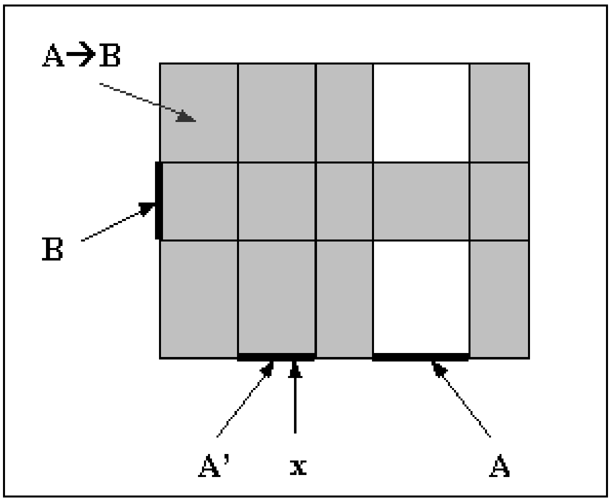
Er wordt niet vaak met crispe sets geredeneerd, maar voor de volledigheid wordt hier een manier vermeld, dat een speciaal geval is van het redeneren met fuzzy sets. Er zijn de sets A, B en A' en de regel A→B. Geprobeerd wordt om op basis hiervan de set B' af te leiden. Er wordt B'=B afgeleid als A en A' elementen gemeen hebben en B' = ∅ als A en A' geen elementen gemeen hebben.

2.2.13 Fuzzy redeneren

Bij het redeneren met fuzzy sets gebeurt op basis van de regel A→B, met A⊆X en B⊆Y fuzzy sets die de lidmaatschapwaarden voor A→B bepalen. Als de Gödel implicatie gebruikt wordt geldt (A→B)(x,y) = 1 als A(x)<=B(y) en B(y) anderszijds. A en B doen dus geen uitspraak over de waarheid van de elementen in A maar over de relatie tussen de elementen in A en B. Daarnaast zijn er de sets A' en B'. A' is een set op het zelfde domein als A en stelt een waarneming op dat domein voor. B' is een set op het zelfde domein als B en stelt de conclusie voor die op basis van A' en A→B getrokken wordt. Voor het afleiden van B' wordt in de theorie gebruik gemaakt van de volgende methodiek[Klir].

$$B'(y) = \sup_{x \in X} \min[A'(x), A \rightarrow B(x,y)]$$

Dat deze methodiek niet werkt zal in het volgende crispe voorbeeld verduidelijkt worden. Stel er is een x waarvoor geldt A'(x)=1 en A(x)=0. In dit geval zal voor elke versie van de implicatie gelden dat A→B(x,y) = 1 ∀y∈Y, dus geldt B' = Y. Alle y uit Y kunnen dus geconcludeerd worden. In figuur 2.9 wordt dit verduidelijkt.



Figuur 2.9 **Voorbeeld ligging sets theoretische methodiek**

Set A, A' en B zijn set met alleen elementen hebben op de zwart streep. x is een punt in A'. Het grijze vlak geeft aan welke elementen in A→B zitten. Dat alle y uit Y geconcludeerd worden is uiteraard niet de bedoeling. Om dit te voorkomen moet gezorgt worden da dat de implicatie A→B alleen gebruikt wordt als A(x) geldt. In de praktijk wordt dan ook de volgende methodiek gebruikt.

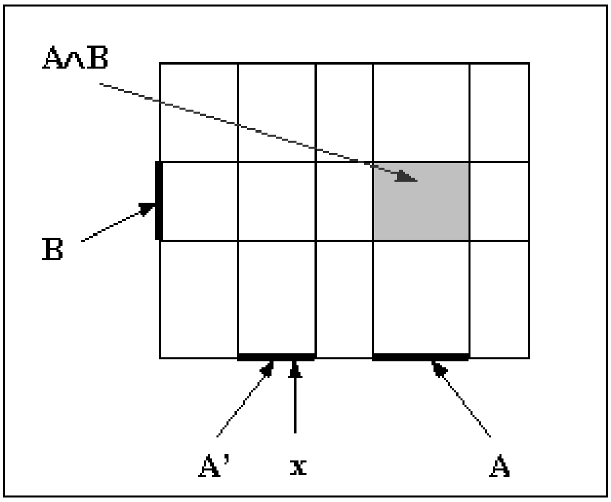
$$B'(y) = \sup_{x \in X}(A'(x) \wedge A(x) \wedge A \rightarrow B(x,y))$$

Voor de combinatie van de Gödel implicatie en de standaard doorsnede geldt dat A∧A→B = A∧B

Daarom wordt vaak $B'(y) = \sup_{x \in X} \min[A'(x), A(x), B(y)]$ en

$B'(y) = \sup_{x \in X}(A'(x)A(x)B(y))$ gebruikt.

Het vorige voorbeeld ziet er dat uit als in figuur 2.10.



Figuur 2.10 **Voorbeeld ligging sets methodiek uit praktijk**

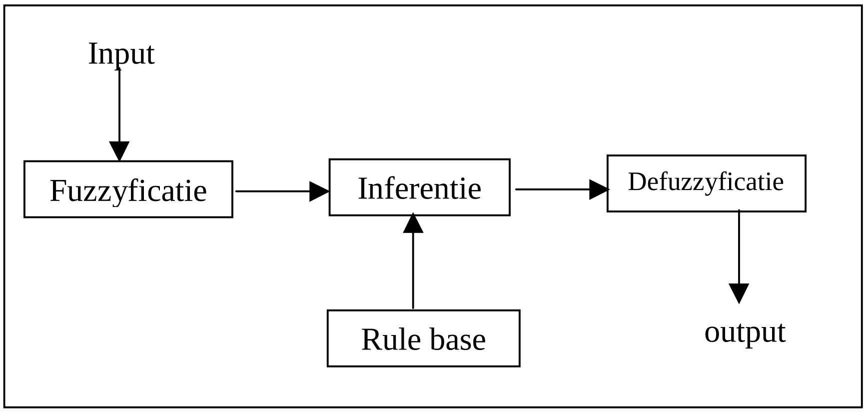
Als er meerdere regels gelden worden de conclusies verenigd. C= $B'_1 \cup B'_2 \cup \ldots \cup B'_n$. Er moet nog wel bepaald worden welke vereniging hier gebruik wordt. Omdat de B'_i afkomstig zijn van verschillende regels is het veelal gewenst dat de regels elkaar versterken. Daarom worden de conclusies meestal opgeteld. In dat geval is er sprake van een Additief Fuzzy Systsem (AFS)

3 Additieve fuzzy systemen.

In dit hoofdstuk gaat het over additieve fuzzy systemen(AFS). Eerst zullen de eigenschappen van fuzzy systemen behandeld worden. Vervolgens zullen de eigenschappen van Additieve fyzzy systemen uitgewerkt worden. De AFS is een speciale groep binnen de Fuzzy systemen, deze systemen worden het meest in de praktijk toegepast.

3.1 Fuzzy system

Een fuzzy systeem is een systeem waarvan één of meer variabelen als fuzzy sets worden weergegeven[Klir]. Een fuzzy systeem ziet er meestal uit als in figuur 3.1



Figuur 3.1 **Een fuzzy systeem**

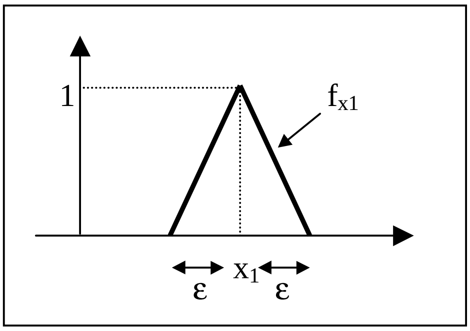
Voordat een input in een fuzzy systeem tot een output komt moeten er drie stappen doorlopen worden. De eerste stap is de fuzzyficatie, bij deze stap worden de crispe input variabelen omgezet in fuzzy sets. De tweede stap is de inferentie, hierbij wordt op basis van de input fuzzy sets en de Rule base een output fuzzy set gemaakt. Bij de derde stap, de defuzzyficatie, wordt de output fuzzy set weer omgezet tot een crispe output, dit gebeurt omdat de buitenwereld niet zo goed om kan gaan met fuzzy sets.

3.1.1 Rule Base

Voordat de drie stappen in het fuzzy systeem uitgelegd worden zal eerst de inhoud van de rule base beschreven worden. De rule base bestaat uit een aantal fuzzy rules van de vorm $A \rightarrow B$, waarbij A een fuzzy set is op het input domein van het fuzzy systeem en B is een fuzzy set op het output domein van het fuzzy systeem. Deze regel kan geïnterpreteerd worden als 'Als A' in A, dan B' in B waarbij A' de gefuzzyficeerde inputvariabelen voorstelt en B' de fuzzy outputvariabele. Het leren van een fuzzy system bestaat uit het bepalen van de fuzzy rules in de rule base. De rule base is dus het belangrijkste deel van een fuzzy system.

3.1.2 Fuzzificatie stap

In de fuzzyficatie stap worden de input variabelen $x = (x_1, x_2, .. ,x_n)$ omgezet tot één fuzzy set A' op het domein van de input variabelen. In figuur 3.2 staat een voorbeeld van een variabele x_1 die omgezet wordt tot een fuzzy set met als functie $f_\varepsilon(x_1)$.



Figuur 3.2 **Fuzzyficatie van een input variabele**

Als x een variabele van meerder variabelen is x = (x_1, x_2, .. ,x_n), dan worden de variabele x_i meestal los gefuzzifiseerd, zoals beschreven in figuur 3.2 en worden ze later samen genomen. Het resultaat van de fuzzyficatie kan er dan b.v. als volgt uitzien.

$$A' = \operatorname{Min}_{i=1}^{n} f_{xi}(x_i) \text{ of } A' = \prod_{i=1}^{n} f_{xi}(x_i)$$

Het is echter niet duidelijk hoe ε geïnterpreteerd moet worden, omdat niet bekend is wat de gevolgen zijn van een grootte of kleine ε op de output van het totale fuzzy systeem. In de literatuur [Klir] wordt ε gezien als een maat van onzekerheid, maar het is onwaarschijnlijk dat in het eindtresultaat van het fuzzy systeem deze onzekerheid goed verwerkt is. In deze scriptie wordt daarom verder geen gebruik gemaakt van een fuzzyficatie en wordt voor de fuzzy set A' die verzameling genomen waarin alleen het element x in het geheel in zit.

3.1.3 Fuzzy Inferentiestap

Tijdens de inferentiestap wordt op basis van de input fuzzy set een output fuzzy set gemaakt. Dit gebeurt op de manier zoals dat beschreven is bij het redeneren met fuzzy sets in het vorige hoofdstuk. Eerst wordt elke regel in de rule base afzonderlijk gebruikt om een conclusie te trekken. Daarna worden deze conclusies samen genomen tot een output fuzzy set.
De input fuzzy set is A' uit de fuzzyficatie stap. De conclusie uit regel i, $A_i \rightarrow B_i$, is B'_i met $B'_i(y) = A'(x) \cap A_i(x) \cap B_i(y)$. Als de standaard doorsnede gebruikt wordt geldt $B'_i(y) = \min(A'(x), A_i(x), B_i(y))$. Als de algebraïsche doorsnede gebruikt wordt geldt $B'_i(y) = A'(x) * A_i(x) * B_i(y)$. In deze scriptie wordt verder hiervoor de algebraïsche doorsnede gebruikt.
Als de conclusies van alle regels bekend zijn moeten deze gecombineerd worden. Dit gebeurt door de vereniging te nemen van alle conclusies B'. Dus conclusie $C = \bigcup_{i=1}^{n} B'_i$.
Ook hier moet het type verenigen gekozen worden. Als de standaard genomen wordt geldt $C(y) = \max_{i=1}^{n}(B'_i(y))$ Als de bounded vereniging genomen wordt worden de conclusies gewoon opgeteld, waarbij gezorgt moet worden dat er niet afgerond word. Als de conclusies opgeteld worden wordt het fuzzy systeem een additief fuzzy systeem genoemd. Er geldt dan $C(y) = \sum_{i=1}^{n}(B'_i(y))$. In deze scriptie wordt verder alleen additieve fuzzy onderzocht en toegepast.

3.1.4 Defuzzyficatie stap

Bij de defuzzificatie stap wordt de conclusie weer omgezet naar een crisp getal, zodat deze makkelijk gebruikt kan worden. Er zijn een aantal methode hiervoor ontwikkeld. De meest gebruikte is de center of gravity methode, deze bepaalt het punt onder de grafiek zodat het oppervlak onder de fuzzy sets rechts en links even groot is. Er geldt dan dat

$$(3.1) \qquad f(x) = \frac{\int_{y=-\infty}^{\infty} C(y) y dy}{\int_{y=-\infty}^{\infty} C(y) dy}$$

Andere methodes zijn de mean of maxima en center of maxima methode. Omdat deze niet eenvoudig te berekenen zijn, worden deze in de praktijk niet vaak gebruikt.
Het gebruikt van de center of gravity methode is eenvoudig als de conclusies uit de verschillende regels opgeteld worden. Als de fuzzy inferentiestap en de defuzzyficatie stap samen genomen worden kan de conclusie als volgt opgeschreven worden.

$$(3.2) \qquad f(x) = \frac{\sum_{i=1}^{n} c_i A_i(x) V_i}{\sum_{i=1}^{n} A_i(x) V_i}$$

V_i is hier het volume van de set B_i.
c_i is het centrum van set B_i, bepaald met de center of grafity methode.

Omdat c_i en V_i van te voren bepaald kunnen worden kost het uitrekenen van de conclusie c niet veel tijd meer.

3.2 Additieve fuzzy systemen

In de vorige paragraaf werden fuzzy systemen behandeld. Hier zal een speciaal soort fuzzy systeem bekeken worden die in de praktijk het meest gebruikt wordt[Kosko], het additieve fuzzy systeem(AFS). Een AFS is additief, omdat de conclusies uit de verschillende regels opgeteld worden. Er zijn heel veel manieren waarop een AFS gebouwd kan worden, omdat er veel manieren zijn voor fuzzyficatie, inferentie en defuzzyficatie, maar vooral omdat de rule base op veel manieren kan worden opgebouwd. In deze scriptie zullen we verder één type systeem gebruiken en zal geconcentreerd worden op het bepalen en interpreteren van de fuzzy rules.
De fuzzyficatie, fuzzy inferentie en de defuzzyficatie stappen liggen dus als volgt vast. Er zal geen gebruik gemaakt worden van fuzzyficatie, tijdens de inferentie wordt gebruik gemaakt van de algebraïsche doorsnede, de conclusie sets worden opgeteld en voor de defuzzyficatie wordt gebruik gemaakt van de center of gravity methode.
De eigenschappen die hier bekeken worden liggen vooral op het vlak van de representatie capaciteiten. Hierbij wordt gekeken welke mogelijke vormen de functie van een AFS kan aannemen.
De functie die door het AFS gegenereerd wordt heeft de volgende vorm:

$$(3.3) \qquad f(x) = \frac{\sum_{i=1}^{n} c_i A_i(x) w_i V_i}{\sum_{i=1}^{n} A_i(x) w_i V_i}$$

V_i is hier het volume van de set B_i.
c_i is het centrum van set B_i, bepaald met de center of grafity methode.
w_i is de wegingsfactor behorend bij rule i.

Omdat w_i en V_i beide het zelfde effect hebben wordt verder in deze scriptie alleen de wegings factor gebruikt en wordt het volume van de output fuzzy set B_i gelijk aan 1 verondersteld.

3.2.1 Basis sets

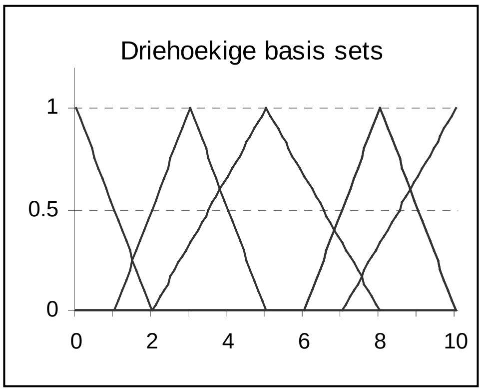
Het input domein van fuzzy rules bestaat vaak uit meerdere variablen. Om gestructureerd sets te maken voor een input domein van meerde variabelen wordt er vaak gebruik gemaakt van sets die gedefinieerd zijn op één van de variabele. Deze set noemen we basis sets. Op elke input variabele worden basis sets gedefinieerd. Om de sets te maken voor de rule base wordt er een carthesische doorsnede of carthesische produkt genomen van de basis sets van de verschillende variabelen.
Zij $A_{i,j}$ de basis set i op variabele j. Er zijn m variabelen en elke variabele heeft n sets
De fuzzy sets die gebruikt worden zijn dan:

$$A_{i_1,..,i_m} = \bigcap_{j=1}^{m} A_{i_j,j} \text{ met } i_j \in [1..n]$$

Er zijn dus totaal n^m fuzzy rules. Als de algebraïsche doorsnede gebruikt wordt worden de sets produkt ontbindbaar genoemd. In deze scriptie zal hiervoor verder de algebraïsche doorsnede gebruikt worden.
Er zijn twee soorten basis sets die vaak gebruikt worden, dit zijn de driehoekige basis sets en de gausische basis sets.

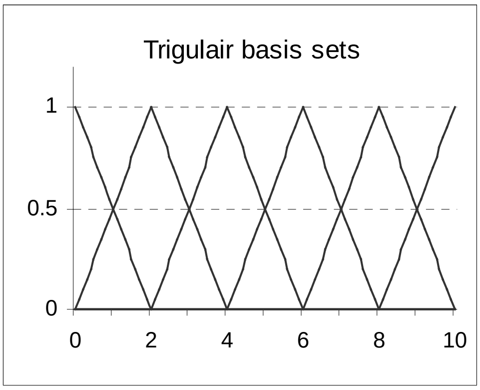
3.2.2 Driehoekige basis sets

In deze paragraaf zullen een aantal eigenschappen van de driehoekige basis sets bespoken worden, uiteraard de vorm van de basis sets en het resultaat van het AFS bij gebruik van deze basis sets.
Driehoekigge basis sets zien eruit als in figuur 3.3



Figuur 3.3 **Driehoekige basis sets.**

Elke basis set bestaat uit een driehoek, die maximaal de waarde 1 aanneemt, de basis sets kunnen elkaar echter willekeurig overlappen.
Een speciale groep van basis sets binnen de driehoekige basis sets is de groep van trigular basis sets. Trigular komt van de samentrekking van het woord triangular en regular. Regular betekend wetmatig, dus de trigular fuzzy basis sets liggen volgens een bepaalde regelmaat naast elkaar. Een voorbeeld van trigular basis sets is te zien in figuur 3.4



Figuur 3.4 **Trigular basis sets.**

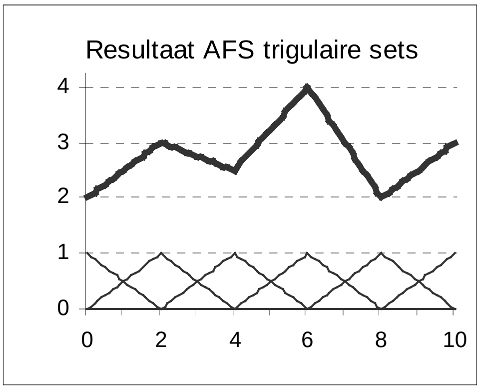
Het is duidelijk te zien dat bij trigular basis sets de waarde van de naastgelegen sets de waarde 0 aannemen, als de set zelf de waarde 1 aanneemt. De trigular fuzzy sets zijn gedefinieerd op basis van een verdeling van het domein. Elke input variabele x^l is verdeeld in n^l delen. De delen zien er als volgt uit: $e^l_j = [x_{l,j-1}, x_{l,j}]$ voor j= 1,2,…,n^l.
Voor elke twee aansluitende delen wordt een fuzzy basis set a^l_j gedefinieerd.

(3.4) $$A^l{}_j(x_l) = \begin{cases} \dfrac{x_l - x_{l,j-1}}{x_{l,j} - x_{l,j-1}} & \text{als } x_l \in e^l{}_j \\ \dfrac{x_{l,j+1} - x_l}{x_{l,j+1} - x_{l,j}} & \text{als } x_l \in e^l{}_{j+1} \\ 0 & \text{anders} \end{cases}$$

Merk op dat de basis sets opgetellen tot 1, aangezien er op elk deel steeds twee fuzzy sets actief zijn. Als er meerdere variabelen gebruikt worden wordt voor het genereren van de fuzzy sets het carthesisch product telken één set uit elke basis set genomen. Dan zijn er voor elk punt 2^m sets tegelijk actief, met m het aantal variabelen.
Als het resultaat van het AFS uit (3.3) bekeken wordt met trigular basis sets en met V_i en w_i gelijk aan 1, ziet het er een stuk eenvoudiger uit, dan in (3.3)

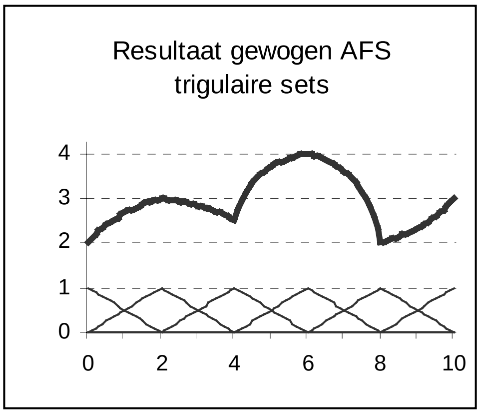
(3.5) $$f(x) = \sum_{i=1}^{n} c_i A_i(x)$$

In figuur 3.5 is het resultaat te zien van een AFS met trigular basis sets, waarbij de outputwaarden van de sets van links naar rechts respectievelijk 2, 3, 2.5, 4, 2 en 3 zijn.



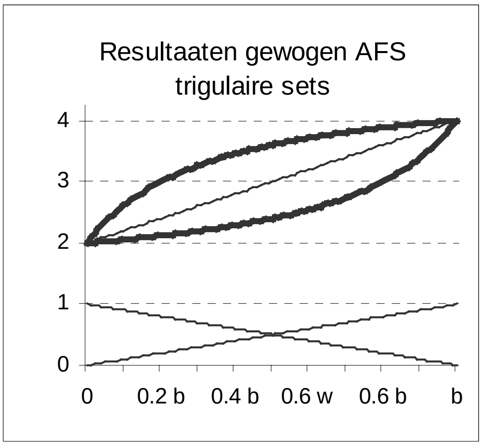
Figuur 3.5 **Resultaat AFS met trigular basis sets**

Omdat in figuur 3.5 slechts één inputvariabele gebruikt wordt is het resultaat van het AFS op een deel telkens lineair. Dit komt doordat alle $A_i(x)$ daar lineair zijn. Als er meerdere input variabelen gebruikt worden is de functie niet meer lineair, maar polynominaal. Dit komt doordat $A_i(x)$ dan een product is van m lineaire delen.
Het is goed te zien in figuur 3.5 dat de functie die het AFS beschrijft telkens ook de waarde c_i aanneemt. Dit komt doordat op de grens tussen de delen er één sets actief is en de functie dus daar de waarde c_i krijgt. Bij meerdere variabelen is er op het hoogtepunt van een fuzzy set ook geen ander fuzzy set actief en zullen dan ook alle c_i waarden aangenomen worden.
Als er wel wegings factoren gebruikt worden is het resultaat van het AFS niet meer in lineaire stukken op de delen en ziet die eruit als in figuur 3.6. De zelfde outputwaarden zijn gebruikt als in fiuguur 3.5 en de wegingen voor de sets zijn van links naar rechts respectievelijk 1, 2, 1, 4, 1 en 0.5.



Figuur 3.6 **Resultaat AFS met trigular basis sets en wegings factoren.**

Het valt op dat in figuur 3.6 in vergelijking met figuur 3.5 de functie vlakker loop in de buurt van het centrum van een fuzzy set als er een zwaare wegingsfactor hangt aan die set ten opzichte van de sets ernaast en juist steiler als er een lichtere wegingfactor aanhangt dan de sets ernaast.
Om verder de eigenschappen van dit AFS te onderzoeken wordt nu gekeken naar een AFS met twee sets met wegingen w1 en w2 en outputwaarden c1 en c2. Het centrum van de linker set ligt bij 0 en het centrum van de rechter set ligt bij b.
In figuur 3.7 staat hiervan een voorbeeld met c1=2 en c2=4 en voor (w1, w2) respectievelijk de combinaties (1,2) en (2,1).



Figuur 3.7 **Resultaten van een met gewogen AFS met trigular sets.**

De functie op het interval tussen 0 en b kan als volgt beschreven worden:

$$f(x) = \frac{w_1c_1 + \frac{(w_2c_2 - w_1c_1)}{b}x}{w_1 + \frac{(w_2 - w_1)}{b}x}$$

De afgeleide op dit interval is: $f'(x) = \dfrac{w_1 \frac{(w_2c_2 - w_2c_1)}{b}}{(w_1 + \frac{(w_2 - w_1)}{b}x)^2}$

De rechter afgeleide in het punt 0 is dan: $f'_{\leftarrow}(0) = \dfrac{w_2}{w_1}\dfrac{(c_2 - c_1)}{b}$

De linker afgeleide in het punt b is dan: $f'_{\rightarrow}(b) = \dfrac{w_1}{w_2}\dfrac{(c_2 - c_1)}{b}$

De verhouding tussen de afgeleidde in het punt 0 en de afgeleide in het punt b is:

$$\frac{f'_{\leftarrow}(0)}{f'_{\leftarrow}(b)} = \frac{w_2^{\,2}}{w_1^{\,2}}$$

De verhouding tussen het de afgeleide bij de maxima van de fuzzy sets is dus omgekeert evenredig aan de verhouding tussen het kwadraat van de wegingen van de fuzzy sets. Bij het bouwen van een AFS zou hier meer rekening gehouden kunnen worden bij het kiezen van de wegingsfactoren.

3.2.3 Gausische basis sets

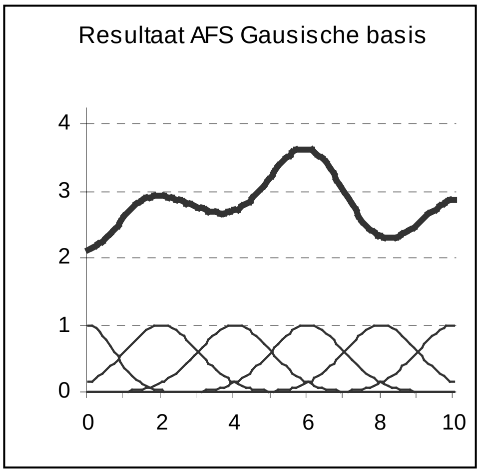
Behalve driehoekige basis sets worden ook vaak Gausische basis sets gebruikt, die de vorm hebben van een gausische kromme.De gausische basis sets zijn van de volgende vorm.

(3.6) $$a^l{}_j(x_l) = e^{-(\frac{x_l - \mu_{l,j}}{\sigma_{l,j}})^2}$$

$\mu_{l,j}$ geeft het maximum van de basis sets aan.

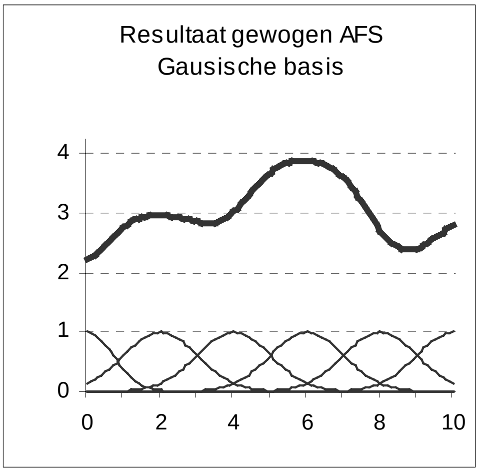
$\sigma_{l,j}$ geeft de breedte van de basis set weer.

De basis sets en het resultaat van de basis sets zijn, zonder gebruik van wegingsfactoren zijn te zien in figuur 3.8. De outputwaarden zijn de zelfde als in figuur 3.5.



Figuur 3.8 **Gausische basis sets en het resultaat**

In figuur 3.8 is te zien dat de functie gladder is dan bij gebruik van trigular sets. De functie neemt echter niet meer de waarden c_i aan op het maximum van de fuzzy sets. Dit komt doordat de overige fuzzy sets in de buurt daar niet de waarde 0 aannemen. Dit is een reden dat AFS op basis van gausische basis sets minder goed te interpreteren zijn. Het is zelfs zo dat alle regels altijd tegelijkertijd actief zijn.
Als er wegings factoren gebruikt worden ziet het resultaat van het AFS er uit als in figuur 3.9.



Figuur 3.9 **Resultaat AFS met gausische basis sets en wegings factoren.**

Net als bij de trigulare basis set loopt de functie vlakker als in de buurt van het centrum van de fuzzy set als de buur sets een kleinere weging hebben. En de functie loopt stijler als de buur sets een zwaardere weging hebben. Verder is te zien dat een zwaardere wegings factor bij een fuzzy set de functie dichter naar de c_i waarden van die set doet lopen.

4 Fuzzy learning

In het vorige hoofdstuk zijn een aantal eigenschappen, zoals het representatie vermogen, van Additieve Fuzzy Systemen uitgewerkt. In dit hoofdstuk zullen we verder gaan met de leer mogelijkheden van AFS. Eerst zal bekeken worden wat voor problemen we aan willen pakken, daarna zullen drie bestaande leermethodes uitgewerkt worden. Afsluitend zullen de problemen die hierbij spelen behandeld worden.

4.1 Functie approximatie

Er zijn verschillende problemen die opgelost kunnen worden met AFS[Klir]. In deze scriptie wordt geprobeerd een approximatie probleem op te lossen. Een functie wordt benaderd door een AFS zo in te stellen , dat het resultaat van het AFS zo veel mogelijk op de functie lijkt. Deze functie is zelf onbekend, maar er zijn wel een ‘groot’ aantal voorbeeld punten aanwezig. Deze voorbeeld punten hoeven niet zuiver te zijn: er kan dus ruis in de voorbeelden zitten.
Omdat er ruis in de voorbeelden kan zitten moet er niet gezocht worden naar een AFS die de voorbeelden 100% benaderd, maar naar een AFS die de globale structuur van de voorbeelden weergeeft en een goede benadering geeft voor voorbeelden die nog niet gezien zijn. Het is dus het oplossen van een regressie probleem en niet van een interpolatie probleem. Gezochtmoet worden naar een AFS dat goed kan generaliseren.
Een bekend theorema over AFS is het unverciële approximatie theorema, dis thoerema zegt dat elke bounded en continue functie benaderd kan worden met een AFS tot een willekeurige naukeurigheid.
Verder willen we graag dat de regels in het AFS geen black box zijn, maar regels zijn die goed geïnterpreteerd kunnen worden. Dit in tegenstelling tot bijvoorbeeld neurale netwerken[Bishop] die bijna nooit geïnterpreteerd kunnen worden.

4.2 Wang-Mendel

In deze paragraaf zal in het kort worden uitgelegd hoe het Wang-Mendel leer algoritme[WangMendel] er uit ziet.
Als eerste worden er op de input variabelen trigular basis sets gedefinieerd, de basis sets op één variabele hebben allen de zelfde breedte. Het aantal basis sets per variabele moet vooraf gekozen worden, maar dit kan per variabele verschillend zijn.
Als de basis sets gekozen zijn worden liggen de input fuzzy sets van de fuzzy regels vast. Deze worden namelijk gemaakt door de doorsnede te nemen van de basis sets.
Op het output domein worden ook een aantal regular sets gedefinieerd, die dan moeten functioneren als output fuzzy sets.
Het leren van het AFS komt nu op neer dat er voor elke input fuzzy set de keuze gemaakt moet worden welke output fuzzy set er gekozen worden. Er kan ook gekozen worden om geen ouput set aan een input set te koppelen, in dat geval wordt er van die input set geen regel gemaakt.
Het leren gaat als volgt. Voor elke voorbeeld wordt gekeken in welke input set deze het meest valt. Als dit ook het voorbeeld is dat van alle voorbeelden het meest in die input set valt, wordt er een regel aangemaakt met waarin die input set gekoppeld is aan de die output set waarin de output waarde van het voorbeeld het meeste valt.
Op basis van deze regels wordt snel een eenvoudig AFS samengesteld.

4.3 Kleinste kwadraten

Bij het algoritme van Wang-Mendel worden er slechts een beperkt aantal output fuzzy gebruikt, zodat de keuze van de output fuzzy sets eenvoudig is. Als er echter geen beperking is op het aantal en de positie van de verschillende output fuzzy sets kan voor het bepalen van de output waarden c_i de kleinste kwadraten methode gebruikt worden. Bij de kleinste kwadraten methode worden de c_i zo bepaald dat het kwadratisch verschil tussen de ouput waarden van het AFS en de ouput waarden van het voorbeeld minimaal is. De functie die dan geminimaliseerd wordt ziet er als volgt uit.

(3.1) $$\sum_{j=1}^{m} (f(x_j) - y_j)^2$$

(3.2) $$f(x_j) = \sum_{i=1}^{n} c_i \frac{A_i(x_j)}{\sum_{k=1}^{n} A_k(x_j)}$$

x_j is de inputwaarde van het j-de voorbeeld.
y_j is de outputwaarde van het j-de voorbeeld.
A_i is de input fuzzy set van de i-de regel.

Het optimaal bepalen van de c_i kan eenvoudig met behulp van lineaire regressie[Fraleigh/Beauregard] Voor het uitvoeren van de lineaire regressie wordt een $m \times 1$ vector y gevuld met de de uitkomst waarden van de voorbeelden. Een $m \times n$ matrix A wordt gemaakt en deze wordt op de volgende manier gevuld.

$$a_{i,j} = \frac{A_i(x_j)}{\sum_{k=1}^{n} A_k(x_j)}$$

De vector met de optimale c_i wordt dan als volgt bepaald:

$$c = (A^T A)^{-1} A^T y$$

Het kleinste kwadraten algoritme kan in het algemeen toegepast worden voor het bepalen van de ouputwaarden van de fuzzy rules, als de inputfuzzy sets bekend zijn.

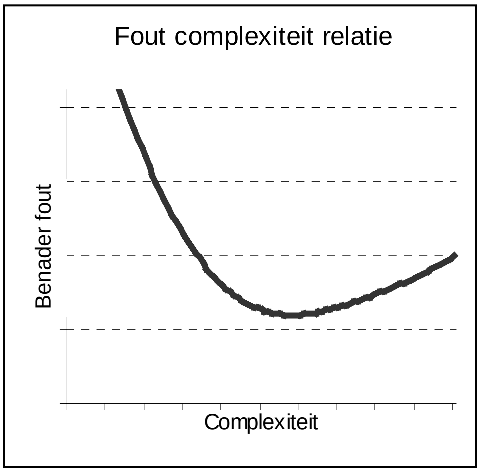
4.4 Gradient descent

De gradient descent methode is een greedy niet-lineaire optimalisatie methode die een functie probeert te minimaliseren. Deze methode garandeert echter niet dat het minimum ook werkelijk gevonden wordt, er bestaat een kans dat de methode stop in een lokaal minimum. Deze methode lijkt op back-propegation bij neurale netwerken.
De kleinste kwadraten methode wordt toegepast voor het bepalen van de centra van de output sets van de fuzzy rules. Met de gradient descent methode kunnen ook de parameters van de input fuzzy sets bepaald worden. Deze methode wordt toegepast door Roger Jang in ANFIS[Jang].
In het kort komt de gradient descent methode op het volgende neer. Het doel is net als bij de Kleinste kwadraten methode om de functie (3.1) te minimaliseren. Er wordt een beginwaarde voor alle parameters in het AFS gekozen. Daarna wordt de partitiële afgeleide van de doelstellings functie bepaald voor elke parameter. Nu is bekend in welke richting de parameters moeten veranderen om de doelstellings functie te verkleinen. Er wordt nu een stap genomen, waarbij alle parameters aangepast worden in die richting. Dit wordt herhaald totdat alle partitiële afgeleiden gelijk aan 0 zijn en er dus in een lokaal minimum aangekomen is.

Omdat de partitiële afgeleide genomen moeten worden is het van belang dat de doelstellings functie differentieerbaar is. Zonder kunstgrepen kan er in dit geval dus geen gebruik gemaakt worden van driehoekige basis sets. In ANFIS wordt er daarom gebruik gemaakt van gausische basis sets.
Bij het gebruik van de gradient descent methode worden de input fuzzy sets zo gevormd dat het resultaat van het AFS zoveel mogelijk lijkt op de te benaderen functie. Daarom moet er rekening gehouden worden dat hierdoor aan de eis, dat het systeem interpreteerbaar moet zijn, misschien niet meer voldaan wordt.

4.5 Model Complexiteit

Bij het modeleren van een probleem is er een belangrijk punt en dat is de model complexiteit[Bishop] van het probleem. Model complexiteit is een soort voor hoe ingewikkeld een model is. Het begrip complexiteit is niet makkelijk ergens in uit te drukken. Voor elk model moet een andere maat vast gesteld worden. Neem als voorbeeld een model dat met een polynoom een functie benaderd op basis van een aantal voorbeeldpunten. Voor de model complexiteit kan hier de graad van het polynoom dat gebruikt wordt genomen worden. Hoe lager de graad hoe lager de complexiteit.
De keuze van de complexiteit is van belang voor de generalisatie capaciteiten van het model. Als in het voorbeeld van de polynomen de graad te klein wordt genomen, zal er niet tot een goed model gekomen kunnen worden. Want een niet lineaire functie kan bijvoorbeeld niet goed met een lineaire functie benaderd worden. Als de graad te groot genomen wordt zullen de punten wel steeds beter benaderd worden, maar als er ruis in de voorbeelden zit zal het polynoom sterk gaan zwabberen, om er voor te zorgen dat alle punten benaderd worden. De algemene struktuur achter de voorbeelden, zal dan niet gevonden worden en het model zal voorbeelden die nog niet gezien zijn slecht benaderen.
De model complexiteit moet dus niet te klein en niet te groot zijn, er moet een optimale complexiteit zijn voor een optimale benadering. In figuur 4.1 staat weer gegeven hoe een typische relatie tussen de model complexitiet en de fout in de benadering er uitziet. Bij het bouwen van een model, moet dus de model complexiteit zorgvuldig gekozen worden. De keuze van de complexiteit wordt vaak gemaakt met behulp van validatie.



Figuur 4.1 **De complexiteit benaderfout relatie**

4.5.1 validatie

Validatie is een manier om de generalisatie capaciteiten van een model te bepalen. Dit kan met behulp van de split sample methode. Deze methode wordt vaak gebruikt om de optimale model complexiteit van een model te bepalen, zodat deze de beste generalisatie capaciteiten heeft. Het idee achter split sample is dat een model gemaakt wordt op basis van een trainings set van voorbeelden en het model daarna gevalideerd wordt op basis van de validatie set. Het model dat het beste de validatie set kan benaderen heeft de beste generalisatie eigenschappen en de complexiteit van dat model moet dus gebruikt worden voor het uiteindelijke model. Het uiteindelijke model wordt gemaakt op basis van de voorbeelden in zowel de trainings set als de validatie set. Om de uiteindelijke generalisatie capaciteiten van het model te vinden wordt een derde, de verificatie set gebruikt.

De keuze van de validatie set is uiteraard zeer belangrijk en daar moet ook goed over nagedacht worden. De validatie set moet niet te groot zijn, omdat de trainings set dan te klein is, maar deze moet wel een representatief deel zijn van de voorbeelden. Het process kan ook een aantal keer herhaald worden met verschillende kleine validatie sets, die totaal alle voorbeelden bevatten, zodat het totaal zeker representatief is.

4.6 Curse of dimensionality

De 'curse of dimensionality' is een probleem dat speelt bij het maken van een fuzzy model op basis van veel variabelen. De curse of dimensionality is niet alleen een probleem bij fuzzy systemen, maar speelt bijvoorbeeld ook bij hoog dimensionale functie benaderingen met polynomen. Het probleem is dat de model complexiteit snel toeneemt met het aantal variabelen dat gebruikt wordt.

Bij fuzzy systemen neemt het aantal fuzzysets exponenciëel toe met het aantal dimensies van de inputruimte. De fuzzy sets worden namelijk gemaakt door de doorsnede te nemen van elke fuzzy basis set met elke basis set van elke andere variabele. Als elke variable zeg n basis sets heeft, dan is het aantal fuzzysets dat dat gebruikt wordt n^m, waar m het aantal variabelen is. Het aantal fuzzy sets is dus exponenciëel afhankelijk van het aantal variabelen.

De model complexiteit van een fuzzy systeem zit voornamelijk in het aantal fuzzy sets en de vormen die een fuzzy set mag aannemen. Bij een hoogdimensionele input ruimte kan de model complexiteit dus niet meer zorgvuldig gekozen worden en zal de kans groter zijn dat het model een slechte generalisatie cappaciteit heeft. Boverndien is een AFS met veel fuzzy sets niet meer interpreteerbaar.

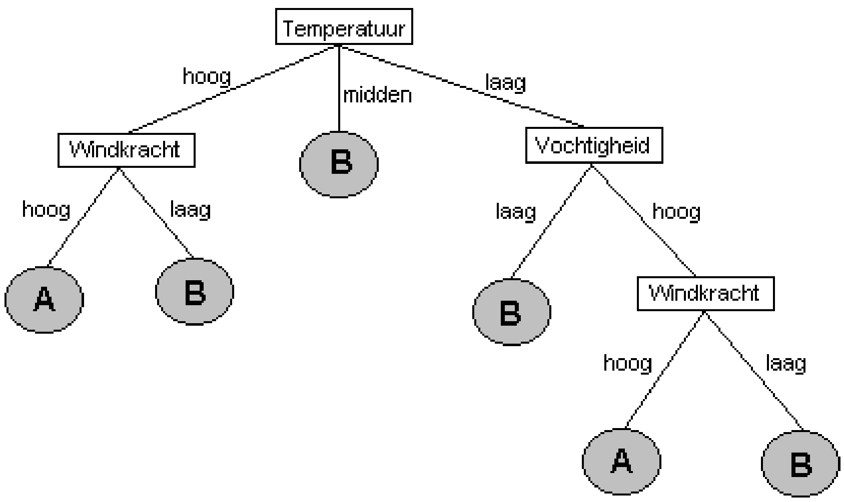
Als fuzzy systemen wel gebruikt gaan worden voor een probleem met een hoogdimensionale inputruimte, dan moet dit probleem opgelost worden. Om dit probleem op te lossen moet het aantal fuzzy sets niet afhankelijk zijn van het aantal input variabelen, maar deze moet flexibel gekozen kunnen worden.

5 Fuzzy beslissingsboom

Voor het oplossen van de ‘curse of dimensionality’ moet gezocht worden naar een andere en betere manier voor het kiezen van de fuzzy sets. Bij standaard fuzzy systems is een fuzzy set automatisch afhankelijk van alle variabelen, zodat het aantal fuzzy sets exponentieel afhankelijk is van het aantal variabelen. Voor het oplossen van de curse of dimensionality moet de eis dat een fuzzy sets afhankelijk is van alle variabelen losgelaten worden en kan een fuzzy set dus slechts afhankelijk zijn van een beperkt aantal variablen. Tevens moet de model complexiteit goed gekozen kunnen worden. Omdat een beslissingsboom er voor zorgt dat bladeren afhankelijk zijn van een beperkt aantal variabelen en per blad de aantal variabelen kan verschillen ligt het gebruik van een fuzzy versie van een beslissingsboom voor de hand. Daarnaast is het handig dat een beslissingsboom stapgewijs opgebouwd wordt, zodat de model complexiteit in de hand gehouden kan worden. Eerst zullen de eigenschappen van een crispe beslissingsboom uiteengezet worden, waarna de fuzzy beslissingsboom behandeld wordt.

5.1 Crispe beslissingsboom

Een cripse beslissingsboom wordt gebruikt om classificatie problemen op te lossen. Als een element geclassificeerd moet worden geeft de boom een stappenplan voor het bepalen van de klasse. Er wordt begonnen bij de wortel knoop. Een knoop stelt een vraag over een bepaalde variabele van het element en geeft afhankelijk van het antwoord aan naar welke knoop doorgegaan moet worden. Dit gaat door totdat er bij een blad uitgekomen wordt die aangeeft in welke klasse het element valt.



Figuur 5.1 **Een beslissingsboom**

5.1.1 Construeren van de boom

Bij het construeren van een boom is het de bedoeling dat de boom zo klein mogelijk wordt, maar toch alle voorbeelden goed classificeerd. Een algoritme dat hiervoor veel gebruikt wordt is ID3 [Quinlan 1985]. De zaken waarmee rekening gehouden moet worden zal hier behandeld worden.

Een beslissingsboom wordt meestal top-down gebouwd, waarbij er stap voor stap een knoop wordt toegevoegd. Het toevoegen van knopen gaat door totdat alle voorbeelden goed geclassificeerd zijn. Voor het bepalen welke variabele gebruikt wordt in de knoop wordt gebruik gemaakt van een informatie functie[BoekeLubbe 1988]. Die informatie functie geeft aan hoeveel bits er minimaal nodig zijn om de voorbeelden in de klasse te classificeren. Als er twee klassen, positief en negatief zijn ziet een informatie functie er als volgt uit.

$$I(p,n) = -\frac{p}{p+n}\log_2\frac{p}{p+n} - \frac{n}{p+n}\log_2\frac{n}{p+n}$$

p is het aantal positieve voorbeelden in de knoop.
n is het aantal negatieve voorbeelden in de knoop.

De variabele die gebruikt wordt voor de knoop is die waarvan de gewogen som van de informatie functie over zijn takken het kleinst is. Dit is dus een greedy algoritme, dat niet kan garanderen dat de kleinste boom gevonden wordt, maar meestal wel een goede oplossing vindt.

5.1.2 Ruis in de voorbeelden

Als ruis in de voorbeelden zit levert dat problemen op bij het construeren van de boom. De boom wordt snel te groot. Om dit probleem op te lossen, zijn er twee verschillende methodes.
De eerste methode is door niet op basis van de hele data set een boom te construeren, maar slechts een kleine trainings set te gebruiken. Als de trainings set representatief is voor de data set, zal dit een boom opleveren met goede generaliserende eigenschappen. De trainings set kan ook iteratief worden opgesteld, door nadat een boom gemaakt is de voorbeelden die niet goed geclassificeerd zijn aan de trainings set toe te voegen. Ook als er geen ruis in de voorbeelden zit kan deze methode het construeren van een boom doen versnellen.
De tweede methode is het beperken van de modelcomplexiteit door het vervroegd stoppen met toevoegen van knopen, als er slechts een klein deel fout geclassificeerd wordt. Het bepalen wanneer een deling geen zin meer heeft kan met behulp van een chi-kwadraat test. Er wordt dan gekeken of de voorbeelden in de takken niet allemaal tot de zelfde verdeling kunnen behoren. Als dit namelijk het geval is heeft het geen zin om deze knoop te splitsen.

5.2 Fuzzy beslissingsboom

In het vorige hoofdstuk kwam naar voren dat de fuzzy systemen die hun fuzzy sets kiezen door een doorsnede te nemen van alle een basis sets van alle variabelen problemen krijgt van de ‘curse of dimensionality’. Dit komt doordat het aantal fuzzy sets exponencieel groeit met het aantal input variabelen. In het fuzzy beslissings boom algoritme dat in deze scriptie gebruikt wordt, wordt geprobeerd het aantal fuzzy sets en dus de complexiteit van het AFS zo klein mogelijk te houden. Dit gebeurt door net als in het ID3 algoritme voor crispe beslissings bomen te beginnen met een systeem dat met één variabele werkt en dan langzaam het aantal gebruikte variabelen toe te laten nemen, door knopen toe te voegen. In de volgende paragraaf zal een minimalisatie probleem geponeerd worden. Het fuzzy beslissings boom algoritme zal proberen het minimaliserings probleem op te lossen. Dit uiteraard onder de voorwaarde dat het AFS interpretabel blijft.

5.3 Minimaliserings probleem

Bij het bouwen van een fuzzy systeem is het de bedoeling om een model te maken dat zo goed mogelijke generalisatie capaciteiten heeft en niet een model dat alleen de trainings set zo goed mogelijk benadert. Een manier om de generalisatie cappaciteiten te meten is door gebruik te maken van een validatie set. Er mag bij het construeren van het model echter geen kennis zijn omtrend de inhoud van de validatie set. Daarom moet het minimaliserings probleem opgesplits worden in twee delen. Het binnenste probleem maakt een model dat zo goed mogelijk de trainings set benaderd met een beperkte complexiteit. Hierbij mogen de fuzzy sets en de output waarden van de fuzzy sets gekozen worden.

Het buitenste probleem kiest het model, dat de validatieset zo goed mogelijk benadert, door de juiste model complexiteit te kiezen.

(5.1) $$\min_{u}\Big(\sum_{j\in V}(f_j - y_j)^2\Big)$$

st.

$$\min_{U,c_i}\Big(\sum_{j\in T}(f_j - y_j)^2\Big)$$

st.

$$f_j = \frac{\sum_{i\in U} A_i(x_j)c_i}{\sum_{i\in U} A_i(x_j)} \qquad \forall j \in V \cup T$$

$$U \subseteq S$$

$$\#U < u$$

U is de verzameling van fuzzy sets die gebruikt worden.
S is de verzameling van fuzzy sets die toegestaan zijn.
T is de verzameling voorbeelden in de trainings set.
V is de verzameling voorbeelden in de validatie set.
A_i is de i-de fuzzysets uit de set U.
c_i is de outputwaarde van de fuzzyset A_i.
u is het maximaal aantal fuzzy sets in U.
x_j is de inputwaarde van voorbeeld j.
y_j is de outputwaarde van voorbeeld j.

De maat voor de modelcomplexiteit is het aantal fuzzy sets. Bij de keuze van de fuzzy sets wordt geprobeerd om de trainings set zo goed mogelijk te benaderen. De structuur wordt zo gekozen dat de trainings set benaderd wordt en niet zodat het systeem goed generaliseert. Daarom kan de keuze van de fuzzy sets ook in het buitenste probleem gemaakt worden en wordt dan alleen nog de keuze van de outputwaarden in het binneste probleem bepaald. Op deze manier worden die sets gekozen die de structuur van het probleem het beste weergeven.

(5.2) $$\min_{U}\Big(\sum_{j\in V}(f_j - y_j)^2\Big)$$

st.

$$U \subseteq S$$
$$\underset{c_i}{Min}\left(\sum_{j \in T} (f_j - y_j)^2\right)$$
st.
$$f_j = \frac{\sum_{i \in U} A_i(x_j) c_i}{\sum_{i \in U} A_i(x_j)} \qquad \forall j \in V \cup T$$

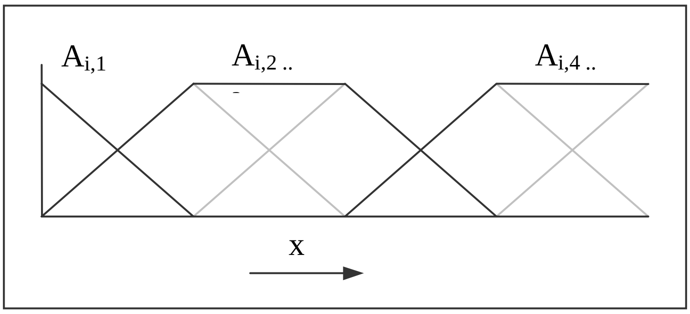
Dit kan uiteraard alleen als er een sterk genoege restrictie op de sets in S ligt die er voor zorgt dat de ruis niet gefit wordt. Het minimaliserings probleem in 5.2 wordt in deze scriptie niet gebruikt, maar zou in de toekomst verder uitgewerkt kunnen worden.

5.4 Fuzzy beslissingsboom algoritme

In deze paragraaf, zal geprobeerd worden het minimaliserings probleem beschreven in (5.1) op te lossen voor een bepaalde verzameling S. Omdat het bepalen van de optimale waarden voor c_i veel rekentijd vergt wordt ook gebruik gemaakt van een andere minimalisatie functie, de fuzzy varantie.

5.4.1 Klasse van Fuzzy sets.

Het beslissingsboom algoritme dat in deze scriptie gebruikt wordt, maakt slechts gebruik van een beperkt classe van fuzzy sets. Deze klasse wordt in het minimalisatie probleem als de set S beschreven. Bij de constructie van de klasse is rekening gehouden dat de fuzzy sets ook werkelijk interpretabel zijn.
Eerst moeten alle inputvariabelen die gebruikt worden afzonderlijk gefuzzyfiseert worden in trigular basis sets. Hoe dit gebeurt en het aantal basis sets dat per variabelen gebruikt wordt moet door een expert bepaald worden. Als refinement kan na afloop van het algoritme op de keuzes teruggekomen worden.
Er worden dus fuzzy basis sets $A_{i,j}$ gedefinieerd waarbij i de variabele weergeeft en j het nummer van de fuzzyset. Deze worden allen uitbereid naar de hele inputruimte, zodat $A_{i,j}(x_1,x_2,\ldots) = A_{i,j}(x_i)$.
Voor elke variable kan een aaneengestoten groepset gedefinieerd worden, door de fuzzy basis sets van naast elkaar liggende sets op te tellen. De volgende vormen ontstaan dan:
$A_{i,j1..j2}(x_1,x_2,\ldots) = A_{i,j1}(x_1,x_2,\ldots) + \ldots + A_{i,j2}(x_1,x_2,\ldots)$. Deze vormen zijn tezien in de figuur 5.2



Figuur 5.2 **De vorm van de groepsets.**

De uiteindelijke klasse van fuzzy sets die gebruikt worden bestaan uit de doorsnede(produkt) van één fuzzy groepset van elk van de variabelen. Als er vier variabelen die allen in 7 delen gefuzzifiseerd zijn dan is het volgende bijvoorbeeld een goede fuzzyset.

$A_{1,1..3} \cap A_{2,1..7} \cap A_{3,3..5} \cap A_{4,1..7}$

In dit voorbeeld worden van de tweede en de vierde variabelen alle sets opgeteld daarom kunnen deze termen ook weggelaten worden en kan deze set ook in de volgnde vorm geschreven worden:

$A_{1,1..3} \cap A_{3,3..5}$.

Als de basis sets gedefinieerd zijn ligt er dus een soort raster over de inputruimte waar verder niet meer van afgeweken kan worden. Als een fuzzy set in twee delen gesneden moet worden kan dit slechts op een beperkt aantal manieren.

5.4.2 Fuzzy variantie

Er moet een groep fuzzysets gekozen worden zodat de square error zoals in (4.1) gedefinieerd is minimaal is. Helaas is dat niet eenvoudig, omdat voor elke mogelijkheid die nagegaan wordt de optimale waarden van c bepaald moeten worden en dit kost echter veel rekentijd. Tevens is het voor het algoritme zinvol om een minimalisatiefunctie te hebben die gesplits kan worden naar fuzzyset en niet naar voorbeelden. Daarom zal tijdens het algoritme behalve de bestaande minimalisatiefunctie ook de fuzzy variantie als minimalisatiefunctie worden gebruikt.

$$MIN\left(\sum_{i\in T} \frac{\sum_{j\in E} A_i(x_j)(\overline{c_i} - y_j)^2}{\sum_{j\in E} A_i(x_j)}\right)$$

De fuzzy verwachting van een fuzzy set i wordt hier als volgt gedefinieerd:

$$\overline{c_i} = \frac{\sum_{j\in E} A(x_j) y_j}{\sum_{j\in E} A(x_j)}$$

De fuzzy variantie geeft de gewogen square error aan wanneer de punten onder de fuzzy set benaderd worden met de constante $\overline{c_i}$. Deze is dus laag als de functie onder de fuzzy set vlak loopt en hoog als deze stijl loopt of zwabbert. Als de fuzzy variantie geminimaliseerd wordt worden de fuzzy sets zo gekozen dat de functie onder de set zo vlak mogelijk liggen. Dit lijkt een goede heuristiek voor het bepalen hoe een fuzzy set in twee delen gesneden moet worden. Het is echter maar de vraag of de fuzzy variantie ook een goede heuristiek is voor het bepalen welke fuzzy sets als eerste in twee stukken verdeeeld moet worden. Betere heuristieken hiervoor moeten bestaan.

5.4.3 Het algoritme

Dit algoritme bepaald een deelverzameling T uit S door steeds een fuzzyset op te delen in twee kleinere fuzzysets totdat het maximum aantal fuzzysets is bereikt. S is hier de verzameling van fuzzysets die in de klasse zit zoals in par 3.2.1 gedefinieerd. In het begin bestaat T alleen uit de 0-set A_0, die bestaat uit de som van alle fuzzy basis sets. Deze wordt dan bij elke stap in twee delen gesplitst.

$T = \{A_0\}$

do {

Haal een set D uit T.

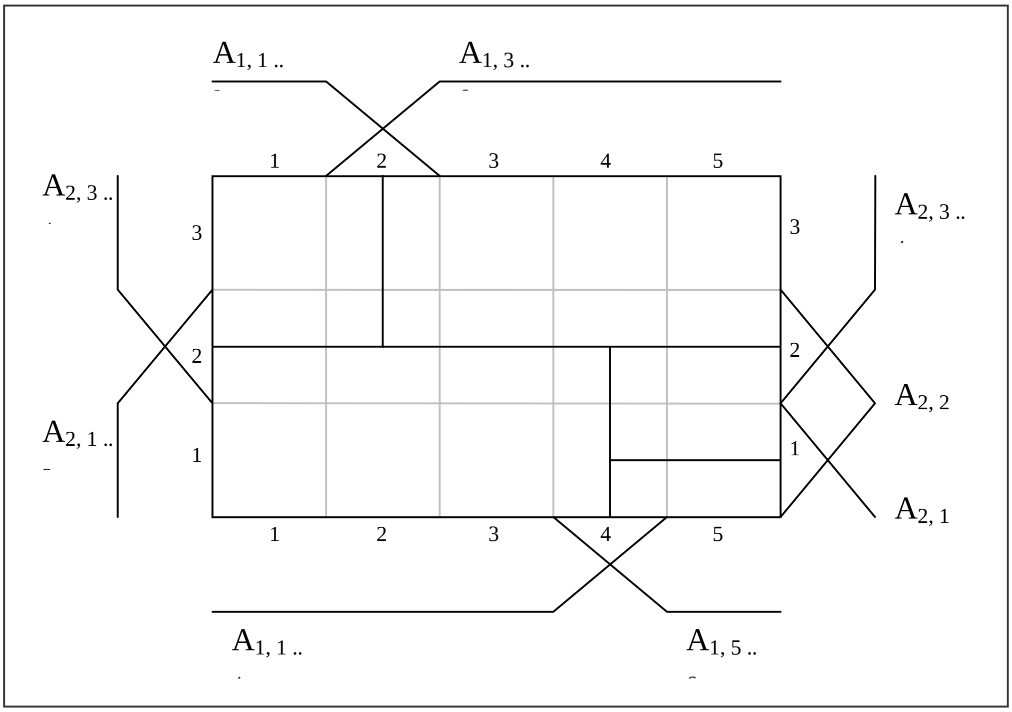
Bepaal een manier van verdelen van D.

Voeg de delen van D toe aan T.
Bepaal optimale waarden voor c_i.
} while (#T<t)
Bepaal welke waarde voor #T de beste resultaten levert op de validatie set.

De constante t geeft het maximaal aantal fuzzy sets weer. Dit hoeft echter niet het aantal fuzzy sets te zijn in het uiteindelijk model.
In het algoritme is alleen nog niet gespecifiseerd welke set D er gekozen gaat worden en op welke wijze deze gesplitst gaat worden.
Het beste lijkt te zijn, door die D en die manier van splitsen te kiezen, zodat de minimalisatie term zoveel mogelijk daalt. Dit is de steepest desent benadering. Omdat er heel veel manieren zzijn om de set D te splitsen kan hier niet square error minimalisatie functie gebruikt worden, maar wordt de fuzzy variantie als minimalisatie functie gebruikt.
In het algoritme gebeurt het kiezen van de manier van snijden door voor elke variabele te kijken op welke manieren de groepset in twee delen gesplitst kan worden en hoeveel winst hiermee gehaald wordt. De splitsing die het meest opleverd zal dan gekozen worden. Als set D wordt uiteraard die set gekozen die bij splitsing het meest opleverd. Hierbij wordt duidelijk waarom het handig is dat de fuzzy variantie naar fuzzy sets gesplitst kan worden. Dit is zodat de winst die gehaald wordt door een fuzzyset te splitsen onafhankelijk is van de vorm van de overige fuzzy sets. De winst hoeft per set slechts eenmaal berekend hoeft te worden en niet telkens opnieuw als een buurset gesplitst wordt.
Het algoritme ziet er dan als volgt uit.

$T = \{A_0\}$
do {
Haal set D uit T met de meeste winst.
Deel D in twee delen.
Bepaal van de delen wat de maximale winst zou zijn bij splitsing.
Voeg de delen van D toe aan T.
Bepaal optimale waarden voor c_i.
} while (#T<t)
Bepaal welke waarde voor #T de beste resultaten levert op de validatie set.

In figuur 5.3 staat aangegeven hoe de fuzzy sets er uit kunnen zien als er twee input variabelen gebruikt worden.



Figuur 5.3 **Voorbeeld van uitkomst van het fuzzy beslissingsboom algoritme**

6 Experimenten

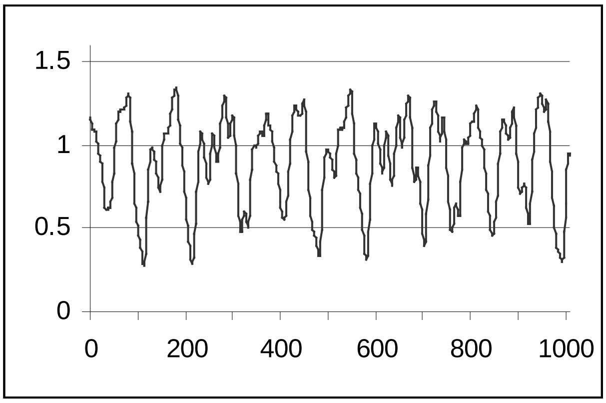
In dit hoofd stuk zullen de experimenten die uitgevoerd zijn doorgenomen worden. Alle experimenten zijn uitgevoerd op één data set. Deze dataset is een tijdreeks van het Mackey-Glass type. Deze zal behandeld worden in de eerste paragraaf.

6.1 Tijdreeks

De tijdreeks die als dataset gebruikt is van het type Mackey-Glass. Dit is een tijdreeks die chaotisch gedrag vertoond. Dit betekend dat een kleine verandering van de begin waarden van de tijdreeks een onvoorspelbare verandering tot gevolg heeft. Onder chaotische systemen bevinden zich bijvoorbeeld beurskoersen en het weer, maar er zijn ook wiskundige modellen die chaotische eigenschappen hebben. De Mackey-Glass tijdreeksen zijn gemaakt op basis van de volgende differentiaal vergelijking.

$$\frac{dx(t)}{dt} = \frac{0.2x(t-\tau)}{1+x^{10}(t-\tau)} - 0.1x(t)$$

De tijdreeks die in deze scriptie gebruikt is, is gemaakt met de parameters τ=30 en dt = 1. De begin waarden zijn willekeurig genomen en zijn eerst 1000 maal geïtereerd, waarna de gebruikte set gegenereerd is. De gebruikte tijdreeks is te zien in figuur 6.1



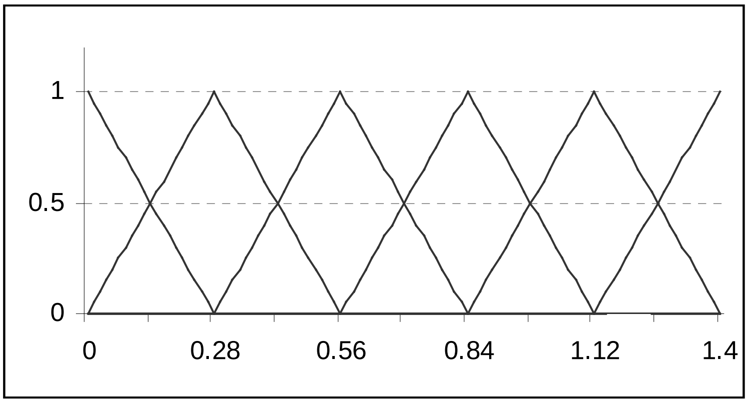
Figuur 6.1 **De gebruikte tijdreeks**

6.2 Doelstelling

De bedoeling is om een systeem te maken die een goede voorspelling kan geven voor een waarde van de tijdreeks in de toekomst op basis van informatie over het gedrag van de tijdreeks in het verleden. Voor het gedrag van de tijdreeks in het verleden worden de negen afgelopen waarden van de tijdreeks gebruikt. Er wordt dus geprobeerd x_t te voorspellen op basis van de waarden x_{t-1}, x_{t-2}, … , x_{t-9}. Op basis van 1010 waarden zijn 1000 voorbeelden gemaakt die gebruikt gaan worden voor het trainen van de fuzzy beslissingsboom.

6.3 Gebruikte basis sets

Voordat met het fuzzy beslissigsboom algoritme gestart wordt moet eerst de set S gekozen worden. Dit betekent dat bepaald moet worden hoe de inputvariabelen gefuzzyfiseerd worden. Omdat alle inputvariabelen allemaal een waarde van de tijdreeks voorstellen is de keuze gemaakt om alle variabelen op de zelfde manier te fuzzyfiseren waarbij de basis sets op het interval van [0 .. 1.4] liggen. Er is gekozen om alle sets even breed te nemen, zodat er alleen het aantal basis sets per variabele als parameter over blijft. Hoe de basis sets er uitzien met zes basis sets is te zien in figuur 6.2

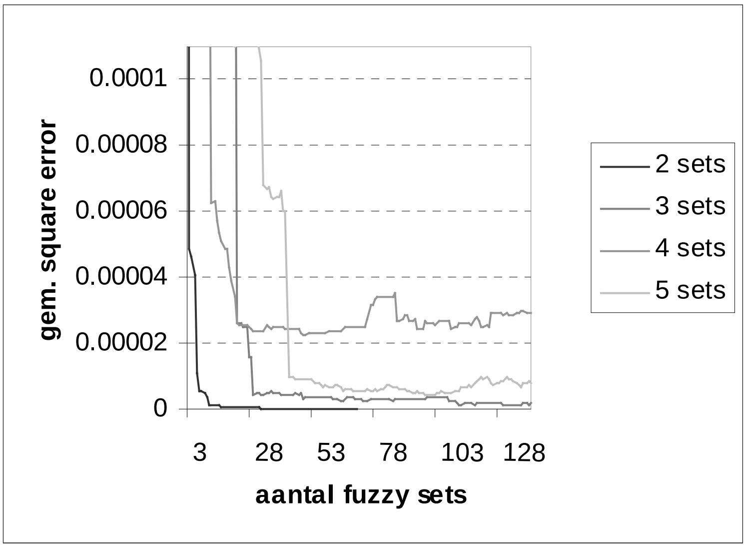


Figuur 6.2 **De gebruikte basis sets**

6.4 Resultaten

In deze paragraaf staan de resultaten van het fuzzy beslissgingsboom algoritme zoals beschreven in het vorige hoofdstuk. Er wordt geleerd op een trainings set van 900 voorbeelden en de generalisatie capaciteiten worden geschat op basis van de validatie set van 100 voorbeelden.

In figuur 6.3 is het resultaat te zien van het fuzzy beslissings boom algoritme bij een verschillende aantal basis sets. Op de x-as staat het aantal fuzzy sets dat in het AFS zit, op de y-as staat de gemiddelde square error op de validatie set. Voor 3, 4 en 5 basis sets is tot 140 fuzzy sets doorgerekend. Voor 2 basis sets is tot 80 doorgerekend, omdat er na dat punt numerieke problemen kwamen met het optimaal bepalen van de output waarden voor de fuzzy sets.



Figuur 6.3 **Resultaat bij 2, 3, 4 en 5 basis sets.**

Als naar figuur 6.3 gekeken wordt valt op dat het gebruik van 2 basis sets het beste resultaat levert en dat de overige daar sterk bij achter blijven. Opvallend is ook dat bij gebruik van 3 en 5 basis sets de beste waarden beter zijn, dan bij gebruik van 4 basis sets. Waarschijnlijk komt dit doordat het gebruik van 4 basis sets de variablen minder goed fuzzyfiseert. Dat meer basis sets ook een slechtere aproximatie met zich mee brengt is te verwachten, omdat de model complexiteit dan toeneemt.

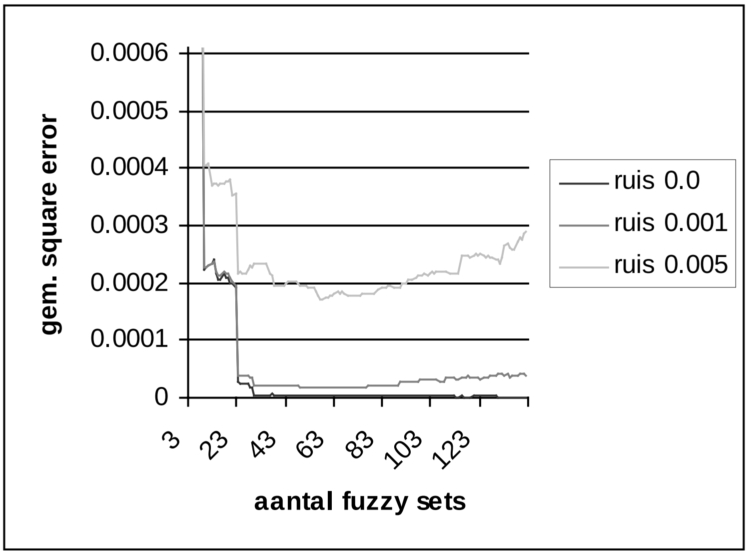
Aantal basis sets	Minimale error	Aantal fuzzy sets
2	$1.16*10^{-7}$	75
3	$1.25*10^{-6}$	134
4	2.26*10-5	51
5	$4.28*10^{-6}$	101

Tabel 6.1 **Beste waarden voor 2, 3, 4 en 5 basis sets**

In tabel 6.1 staan de beste waarden uit figuur 6.3. Hieruit kun je concluderen dat het aantal basis sets dat gebruikt wordt een belangrijke parameter is voor de prestaties van het fuzzy systeem. Als het aantal fuzzy sets niet goed gekozen wordt kan dit snel een factor 10 tot 100 op de square error schelen.

6.4.1 Ruis in de voorbeelden

Om te kijken wat hoe het fuzzy beslissingsboom algoritme reageert op voorbeelden die ruis bevatten wordt aan de waarden van de tijdreeks witte ruis toegevoegd. Het resultaat hiervan is te zien in figuur 6.4.



Figuur 6.4 **Resultaat met 3 basis sets en ruis met standaard deviatie 0.0, 0.001, en 0.005**

In figuur 6.4 is te zien dat als er ruis in zit de voorbeelden zit de error eerst afneemt, maar daarna weer toeneemt. Dit komt doordat de model complexiteit toeneemt met het aantal fuzzy sets en daardoor er meer ruis gefit wordt.

SD van de ruis	Minimale error	Aantal fuzzy sets
0.0	$1.25*10^{-6}$	134
0.001	$1.65*10^{-5}$	68
0.005	$1.72*10^{-4}$	57

Tabel 6.2 **Beste waarden voor ruis met SD van 0.0, 0.001en 0.005**

Uit tabel 6.2 is te zien dat als er meer ruis in de data zit er minder fuzzy sets in het optimale systeem zitten. Tevens is te zien dat het model met ruis veel slechter presteert dan zonder ruis. Dit heeft twee oorzaken. De eerste oorzaak is dat de ruis er voor zorgt dat een slechter model gevonden kan worden. De tweede oorzaak is dat er in de valdatie set ook ruis zit.

6.5 Betouwbaarheid van de voorspelling

Als een systeem een voorspelling geeft voor een bepaalde waarde komt altijd de vraag of die voorspelling ook goed is. Beter is nog om te weten hoe goed die voorspelling is, want als bekend is dat de voorspelling niet goed is wil je je geld er niet op zetten. Daarom zou het systeem bij een voorspelling ook een betrouwbaarheids interval, of een standaard deviatie moeten geven, zodat de voorspelling op waarde geschat kan worden. Zo'n standaard deviatie moet dus ook geschat worden.
Een AFS maakt de voorspelling op basis van fuzzy sets. Het ligt dus voor de hand om de standaard deviatie(SD) ook te schatten op basis van de fuzzy sets. De hier toegepaste methode berekend de SD voor elke fuzzy set uit. Als de SD van een voorspelling berekend moet worden wordt het gewogen gemiddelde gebruikt. De wegingen zijn in verhouding van de mate waarin de input variabelen in de fuzzy sets zitten.

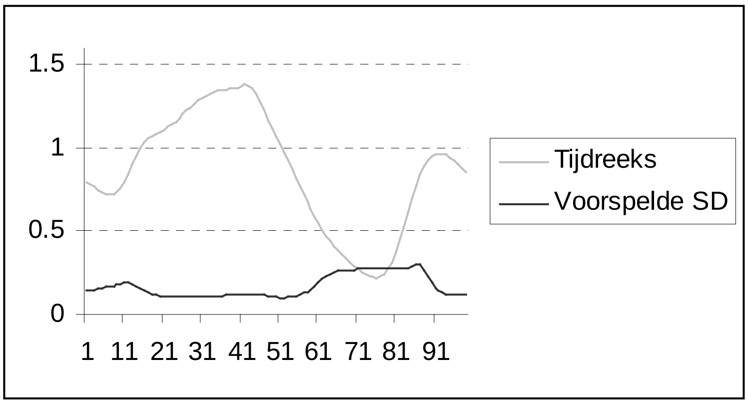
$$SD(x) = \frac{\sum_{i=1}^{n} A_i(x) SD_i}{\sum_{i=1}^{n} A_i(x)}$$

SD(x) is de voorspelling voor de standaard deviatie van de voorspelling voor x
SD_i is de voorspelling van de standaard deviatie voor set I
$A_i(x)$ is de mate waarin het voorbeeld x in fuzzy set i ligt.

De voorspelling voor de standaard deviatie voor een set wordt uit gerekend door voor elk voorbeeld uit de verificatie set het kwadraat van de fout uit te rekenen. En hiervan de wortel van het gewogen gemiddelde te nemen.

$$SD_i = \sqrt{\frac{\sum_{j=1}^{m} A_i(x^j)(y_j - f(x^j))^2}{\sum_{j=1}^{m} A_i(x^j)}}$$

In figuur 6.5 staat een deel van de tijdreeks en de voorspelde standaard deviatie. Het AFS dat hier gebruikt is heeft 3 basis sets per input variable en heeft totaat 40 fuzzy sets. Het is goed te zien dat de SD niet overal gelijk is, maar verschillend is per voorspelling. Voor dit AFS is de voorspelfout groter als de tijdreeks convex loopt en kleiner als de tijdreeks concaaf loop.



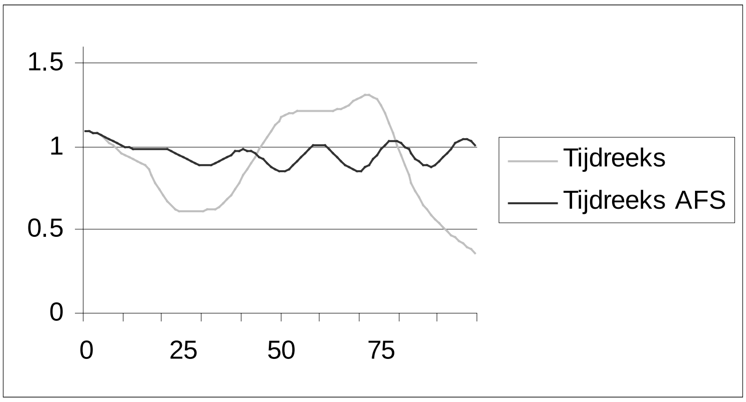
Figuur 6.5 **De tijdreeks en de standaard deviatie van de voorspelling.**

De SD is in de grafiek met 100 vermenigvuldigd.

De SD die hier gebruikt wordt is ook slechts een voorspelling. Er moet nog onderzocht worden hoe goed deze voorspelling is.

6.6 Lange termijn voorspelling

De Additive fuzzy systemen die hier gebruikt worden voorspellen de volgende waarde van de tijdreeks. Het is interesant om te zien of op basis van deze AFS ook een voorspelling op de langere termijn gemaakt kan worden. Om dit te onderzoeken worden de waarden van een tijdreeks voorspeld, waarbij voorspelde waarden weer als invoer genomen worden. Hier is het AFS met 3 basis sets en totaal 40 fuzzy sets gebruikt. Het resultaat is te zien in figuur 6.6



Figuur 6.6 **De lange termijn voorspelling**

Het is goed te zien dat het AFS slecht voorspellingen kan doen op de lange termijn. Dit komt doordat het AFS hierop niet getraind is. Als een voorspelling voor de langere termijn gemaakt moet worden, zal een AFS bemaakt moeten worden die hierop ook getraind is.

6.7 Interpreteren van de resultaten

Een groot voordeel van het gebruik van fuzzy rules voor het modeleren van een systeem is dat de fuzzy rules een semantische betekenis hebben. Dit betekent dat de fuzzy rules geïnterpreteerd kunnen worden. Op deze manier kan de werking van het systeem achterhaald worden en kunnen eventuele fouten in het model eruit gehaald worden.

Hier volgen drie rulebases die achtereenvolgens door het fuzzy beslissingsboom algoritme gegenereerd zijn.

Rulebase 1:
1) Als (x_{t-2} = S of M) en (x_{t-1} = S of M) dan y = M (0.564)
2) Als (x_{t-1} = L) dan y = L (1.348)
3) Als (x_{t-2} = L) en (x_{t-1} = S of M) dan y =L (0.974)

Rulebase 2:
1) Als (x_{t-3} = S of M) en (x_{t-2} = S of M) en (x_{t-1} = S of M) dan y = M (0.553)
2) Als (x_{t-1} = L) dan y = L (1.46)
3) Als (x_{t-2} = L) en (x_{t-1} = S of M) dan y = S (0.325)
4) Als (x_{t-3} = L) en (x_{t-2} = S of M) en (x_{t-1} = S of M) dan y = L (1.364)

Rulebase 3:
1) Als (x_{t-3} = S of M) en (x_{t-2} = S of M) en (x_{t-1} = M) dan y = M (0.731)
2) Als (x_{t-1} = L) dan y = L (1.381)
3) Als (x_{t-2} = L) en (x_{t-1} = S of M) dan y = L (0.912)
4) Als (x_{t-3} = L) en (x_{t-2} = S of M) en (x_{t-1} = S of M) dan y = S (0.265)
5) Als (x_{t-3} = S of M) en (x_{t-2} = S of M) en (x_{t-1} = S) dan y = S (-0.046)

Aan de rulebases is te zien dat in het begin alleen de variablen x_{t-1} tot x_{t-3} gebruikt worden, dit is te verwachten, omdat deze varaiabelen het dichtst bij de huidige waarde liggen en daarom bruikbaarder zijn voor de voorspelling van x_t. Bij de overgang van Rulebase 1 naar Rulebase 2 wordt de regel 1 gesplitst naar de variable x_{t-3}. Zo ontstaan de regels 1 en 4 van rulebase 2. Bij de overgang van Rulebase 2 naar Rulebase 3 wordt regel 1 gesplitst in de regels 1 en 5 van rulebase 3. Hierbij wordt er gesplitst naar variable x_{t-1}. Merk op dat de rulebase nog niet stabiel is, omdat regel 3 als conclusie eerst Large in Rulebase 1 heeft, daarna Small in Rulebase 2 en later weer Large in Rulebase 3. Voor dit AFS wordt de rulebase stabiel, d.w.z dat de outputwaarden van fuzzy sets niet meer veel veranderen, bij ongeveer 10 fuzzy sets.
Voor deze tijdreeks bestaat het uiteindelijk model uit meer dan 50 regels gebruikt. Dit is een aantal dat niet goed te overzien is en daarom moeilijk te interpreteren. Daarom kan voor de interpretatie een model genomen worden dat een kleiner aantal heeft, maar wel al stabiel is. Dit model geeft dan globaal de werking aan van het model dat gebruikt wordt.
Het is opvallend dat naarmate er meer fuzzy sets in het model komen de output waarden van die sets niet meer in het interval [0 .. 1.4] liggen. Dit komt doordat het centrum van de meeste fuzzy sets niet ligt op een plaats waar in werkelijkheid ook punten liggen. De outputwaarden worden dan gemiddeld met andere fuzzy sets, zodat uiteindelijk de punten wel in het interval [0 .. 1.4] komen te liggen. Deze regels kunnen daardoor moeilijk geïnterpreteerd worden. Om dit op te lossen kan het aantal fuzzy sets groter gemaakt worden of moet ervoor gezorgt worden dat de voorbeelden beter over de inputruimte verdeeld zijn, zodat het centrum van de fuzzy sets wel in de buurt van de voorbeelden liggen.

6.7.1 Voorwaarden voor goede interpretatie

Als een AFS goed interpreteerbaar moet zijn, dan zijn er een antal voorwaarden die moeten gelden.
De fuzzy sets hebben een semantische betekenis. Dit betekend dat elke fuzzy set in een taal omgezet kan worden.
Er zijn weinig fuzzy sets tegelijkertijd actief. Dit is nodig zodat het gevolg van één set goed in te schatten is. Dit is de reden dat AFS op basis van gausische basis sets slechter zijn te interpreteren, omdat er dan altijd meerdere fuzzy sets actief zijn.

De maxima van de fuzzy sets liggen in het gebied waar de voorbeelden liggen. Dit is belangrijk, omdat dan een goede schatting is te maken van een waarde van het AFS op basis van één fuzzy set.
Het aantal fuzzy sets is beperkt. Dit om te voorkomen dat door de bomen het bos niet meer gezien kan worden. Om het bos van de bomen te kunnen onderscheiden kan het interpreteren met een kleine boom gebeuren, terwijl voor de benadering een grotere gebruikt wordt.

Het beste model dat gevonden was voor het beschrijven van de tijdreeks is die met 2 basis sets per variabele. Het is duidelijk dat dit model niet aan de laatse twee eisen voldoet. Er moet dus een afweging gemaakt worden tussen een goede benadering en interpreteerbaarheid.

7 Conclusie en toekomstig werk

In dit hoofdstuk zullen de conclusies die getrokken kunnen worden op een rij gezet. Daarnaast zullen de zaken waar verder onderzoek aan gedaan kan worden uiteen worden gezet.

7.1 Conclusie

Om de conclusies van deze scriptie op een rij te zetten zullen de vier vragen van de vraagstelling één voor één afgelopen worden.

- De eerste vraag was het onderzoeken hoe er geredeneerd wordt in de fuzzy logica. Dit was meteen de moeilijkste doelstelling. In de theorie is nog geen eenduidige manier gevonden is om te redeneren, er zijn namelijk verschillende implicaties die gebruikt kunnen worden. De manier waarop met deze implicatie geredeneerd wordt is wel gelijk. In de praktijk wordt er echter niet gebruik gemaakt van een implicatie, maar van de carthesische doorsnede of het carthesische produkt. De reden hiervoor is dat het gebruik van een implicatie geen goede uitkomsten geeft als deze op de zelfde manier gebruikt worden als in de praktijk.
- De tweede vraag bestond uit het onderzoeken van de representatie capaciteiten van AFS. In hoofdstuk drie kwam naar voren, dat veel verschillende AFS gebouwd kunnen worden. De eigenschappen van de AFS hangt af van de basis sets die gekozen zijn en de parameters van het AFS. Bij gebruik van trigular basis sets is het eenduidig te bepalen wat de verschillende parameters voor een invloed uitoefenen op het resultaat van het AFS. De c_i zet de functie op de plaats van het centrum van de fuzzy sets op de waarde c_i. De w_i's zorgen voor afgeleide van de functie in het centrum van de fuzzy set. Een directe relatie tussen de verhouding van het kwadraat van de wegingen van twee sets en de afgeleide in het centrum van deze sets is gevonden. Bij gebruik van gausische basis sets zijn er meer parameters die gezet kunnen worden en is de invloed van de parameters minder duidelijk. De globale invloed is wel te bepalen, maar het uiteindelijke resultaat is altijd afhankelijk van de parameters van alle sets.
- De derde vraag bestond uit het zoeken naar een algoritme dat geen last heeft van de 'curse of dimentionality'. Hiervoor is het fuzzy beslissingsboom algoritme ontworpen. Dit algoritme zorgt ervoor dat niet noodzakelijk alle variabelen in alle fuzzy sets gebruikt hoeven te worden en de model complexiteit gekozen kan worden waarbij de generalisatie capaciteiten maximaal zijn.
- De vierde vraag was het bepalen van de voorwaarden waaronder een AFS geïnterpreteerd kan worden. Het bleek dat er vier belangrijke voorwaarden zijn. De eerste is dat de fuzzy sets linguïstische betekenis hebben. De tweede dat er weinig set tegelijkertijd actief zijn. De derde dat de centra van de fuzzy sets liggen in een gebied waar de voorbeelden ook liggen. De vierde was dat het aantal fuzzy sets beperkt blijft. Omdat het beste AFS meestal niet aan deze voorwaarden zal voldoen moet een afweging gemaakt worden tussen een goed interpreteerbaar AFS enerzijds en goede generalisatie anderzijds.

7.2 Toekomstig werk

De volgende zaken zijn interesant om in de toekomst verder onderzoek aan te verichten:

- Het zoeken naar en theoretische onderbouwing van de redeneer methode die in de praktijk met fuzzy sets gebruikt wordt.

- De representatie capaciteiten van AFS met trigular basis sets zijn in deze scriptie goed onderzocht. De representatie capaciteiten van AFS met gausische basis sets zijn nog niet goed onderzocht. Hieraan zou meer onderzoek vericht kunnen worden.
- Het fuzzy beslissingsboom algoritme zou getest kunnen worden op meerdere problemen om zo te onderzoeken voor welke problemen dit algoritme goede oplossingen geeft en voor welke problemen dit algoritme slechte oplossingen geeft.
- Een andere heuristiek dan de fuzzy variantie zou gebruikt kunnen worden voor het bepalen van de fuzzy set die als eerste gedeeld wordt.
- Een aangepast algoritme kan ontwikkeld worden voor het oplossen van het minimaliserings probleem (5.2), waarbij die structuur van fuzzy sets gekozen wordt die het beste generaliseren.
- Het bepalen van het optimaal aantal benodigde basis sets voor elke variabele.
- Het testen van de voorspelling van de standaard deviatie van de voorspelling.

8 A Referenties

[Fraleigh/Beauregard]	John B. Fraleigh en Raymond A. Beauregard “Linear algebra”, 1990
[Klir]	George J.Klir en Bo Yuan, “Fuzzy sets and fuzzy logic, theory and applications”, 1995
[Berg/Ettes]	Jan van den Berg en Dennis Ettes, “Representation and Learning capabilities of Additive Fuzzy Systems”, 1998
[Wang/Mendel]	Li-Xin Wang en Jerry M. Mendel, “Generation fuzzy rules by learning from examples”, 1991
[Bishop]	Christopher M. Bishop, “Neural Networks for pattern Recognition”, 1995
[Zadeh]	L. A. Zadeh, “Fuzzy sets”, Information and control, 1965
[Hoogland]	H. Hoogland, “FlexMath: de logica van de fuzzy control”, 1997
[Jang]	Jyh-Shing Roger Jang, “ANFIS: Adaptive-Network-Based Fuzzy Inference System”, IEEE Trans. on Systems, Man and Cybernetics, 1993
[Kosko]	B. Kosko, 'Fuzzy Engineering', Prentice Hall, 1997