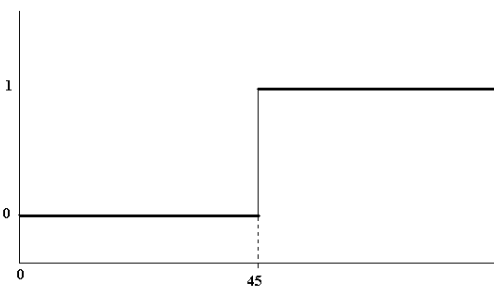


Fuzzy Logic

Wat is waar en wat is niet waar? Volgens wiskundigen is de uitspraak $2 > 1$ waar en $2 < 1$ onwaar. Maar in de praktijk liggen de zaken natuurlijk lang niet zo duidelijk. Bijvoorbeeld van een uitspraak als: "Henk is oud" is moeilijk te zeggen of hij waar of niet waar is, want wanneer is iemand oud? Als hij 40 jaar oud is? Als hij 50 jaar oud is? Omdat een echte wiskundige dit soort vaagheden niet uit kan staan gaan we eens proberen hier wat aan te doen. We komen uit bij 'Fuzzy Logic' — Dave de Jonge

Laten we eens beginnen met wat elementaire logica. In het vakgebied van de logica geven we uitspraken aan als variabelen. Bijvoorbeeld de uitspraak "Henk is ouder dan 45" geven we aan met x . x wordt dan een logische variabele genoemd. Een logische variabele kan twee waarden aannemen: waar of onwaar, oftewel: 0 of 1. Dus als Henk inderdaad ouder is dan 45 geven we dit aan met $x = 1$ en als Henk niet ouder is dan 45 dan geven we dit aan met $x = 0$. We kunnen de waarde van x ook beschouwen als een functie van zijn daadwerkelijke leeftijd l : $x = x(l)$. Dit is dan een stapfunctie.

$$\begin{aligned} x(l) &= 0 \text{ als } l \leq 45 \\ x(l) &= 1 \text{ als } l > 45 \end{aligned}$$



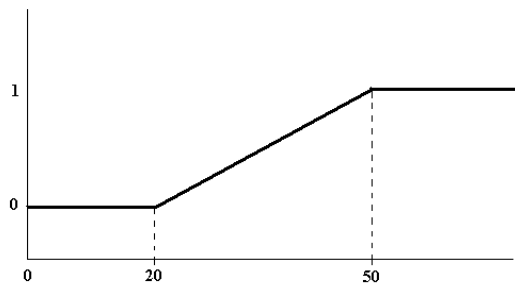
Logische uitspraken kunnen we samenstellen. Stel we hebben bijvoorbeeld een andere uitspraak: "Henk is langer dan 1.80" dan kunnen we deze uitspraak aangeven met y .

Dan betekent $x \wedge y$: "Henk is ouder dan 45 en Henk is langer dan 1.80". Op dezelfde manier geldt:

$x \vee y$ = "Henk is ouder dan 45 of Henk is langer dan 1.80". De uitspraak $x \wedge y$ heeft dus waarde 1 als $x = 1$ en $y = 1$ en heeft waarde 0 in alle andere gevallen. De uitspraak $x \vee y$ heeft waarde 0 als $x = 0$ en $y = 0$ en heeft waarde 1 in alle andere gevallen. Verder hebben we de not-operator: $\neg x$ wil zeggen "Henk is niet ouder dan 45". Dus $\neg x = 0$ als $x = 1$ en $\neg x = 1$ als $x = 0$.

So far so good, maar nu willen we ons dus afvragen: wat doen we als x een vage uitspraak is zoals: "Henk is oud"? Heel simpel: we kennen voor iedere waarde van l een waarde toe aan $x(l)$ die ergens tussen 0 en 1 in ligt. Deze waarde kunnen we dan beschouwen als de mate waarin we de leeftijd l als 'oud' beschouwen. Bijvoorbeeld onder de 20 beschouwen we als niet oud, boven de 50 is oud en daartussenin loopt de waarde van $x(l)$ geleidelijk op:

$$\begin{aligned} X(l) &= 0 && \text{als } l < 20 \\ X(l) &= (l - 20) / 30 && \text{als } 20 \leq l \leq 50 \\ X(l) &= 1 && \text{als } l > 50 \end{aligned}$$



We kunnen nu dus van iemand van 30 jaar niet meer zeggen of hij wel of niet oud is. We kunnen hooguit zeggen dat hij oud is met een graad van $1/3$ op een schaal van 0 tot 1.

We kunnen hetzelfde toepassen op wat abstractere wiskundige uitspraken. Stel we hebben een verzameling A met een deelverzameling U : $U \subset A$. Van ieder element $a \in A$ kunnen we ons nu afvragen of a wel of niet in U zit. Oftewel: we hebben een uitspraak $x(a)$ en die uitspraak $x(a)$ betekent " a zit in U ". Voor iedere a in A heeft $x(a)$ de waarde 0 of 1. (Merk op dat ons eerste voorbeeld hier een speciaal geval van was met A de verzameling van

positieve reële getallen en U de verzameling van alle reële getallen die groter dan 45 zijn.)

Op deze manier kunnen we het concept van een *fuzzy set* definiëren. Zoals de naam al zegt is het een vage versie van het begrip verzameling. Een fuzzy set X is in feite een functie $x(a)$, met waarden tussen 0 en 1, op een verzameling A . Als voor een gegeven element a geldt: $x(a) = 0,25$ dan kunnen we dus niet meer zeggen of a wel of niet in X zit maar hooguit een vage uitspraak doen als: “ a zit in X met een graad van 0,25 op een schaal van 0 tot 1”. De functie $x(a)$ wordt de *waarheidsfunctie* van X genoemd. Als $x(a)$ alleen de waarden 0 of 1 aan zou nemen, dan zouden we X weer als een gewone verzameling kunnen beschouwen, namelijk de verzameling van alle punten a waarvoor geldt $x(a) = 1$.

Merk op dat dit een beetje aan quantum mechanica doet denken, maar toch zit er een belangrijk verschil in: in quantummechanica wordt de golf-functie geïnterpreteerd als een kans dat we een deeltje ergens aantreffen. De golf-functie moet daarom genormaliseerd zijn. Fuzzy logic kan in sommige gevallen ook wel als een kansverdeling geïnterpreteerd worden, maar dat hoeft niet. De waarheidsfunctie hoeft dan ook niet genormaliseerd te zijn. Bijvoorbeeld in het geval van de uitspraak “Henk is oud” kan dit niet als een kans geïnterpreteerd worden. Het is immers niet zo dat als Henk 30 jaar is, er een kans van 33% is dat hij oud is. Met andere woorden: zijn leeftijd zit ergens tussen jong en oud in, maar het is niet zo dat hij de ene keer oud is en de andere keer niet. Wel kunnen we quantummechanica als een speciaal geval van fuzzy logic beschouwen.

Fuzzy Logic wordt echter pas interessant wanneer we uitspraken gaan combineren met de operatoren $\wedge$, $\vee$ en $\neg$. Hoe moeten we deze operatoren interpreteren wanneer we ze gaan toepassen op fuzzy uitspraken? Hierover zijn vele discussies gaande en er is dan ook niet één algemeen aanvaardde standaard. Echter, de volgende definities worden wel het meest gebruikt:

$$x(a) \wedge x(b) = \min(x(a), x(b))$$

$$x(a) \vee x(b) = \max(x(a), x(b))$$

$$\neg x(a) = 1 - x(a)$$

We hebben nu iets heel raars: in de klassieke logica is de uitspraak $x \wedge \neg x$ altijd onwaar ongeacht of x wel of niet waar is. Immers een stelling kan

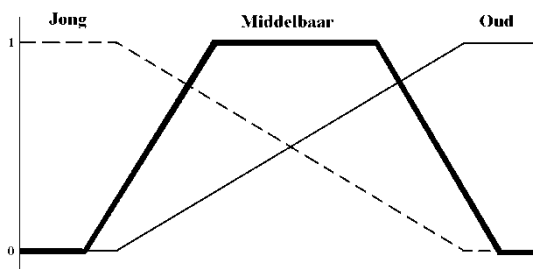
nooit waar en onwaar tegelijk zijn. Maar in fuzzy logic kan het bijvoorbeeld zo zijn dat een stelling ‘half waar’ is ($x = 0,5$) waardoor we krijgen:

$$x \wedge \neg x = \min(x, 1 - x) = 0,5$$

De stelling “ x is waar en niet waar tegelijk” is zelf ook half waar!

Linguistic variables

Voor de variabele ‘leeftijd’ kunnen we dus een waarheidsfunctie ‘oud’ definiëren, maar we kunnen net zo goed een waarheidsfunctie ‘jong’ definiëren, of een waarheidsfunctie ‘middelbaar’. We hebben dan drie functies op het domein van de positieve reële getallen en dus hebben we voor iedere leeftijd drie waarden die aangeven in hoeverre we die leeftijd respectievelijk beschouwen als ‘jong’, ‘middelbaar’ en ‘oud’.



We kunnen dan de leeftijd van Henk op een kwalitatieve manier beschrijven: niet met één getal dat zijn leeftijd aangeeft, maar met drie getallen tussen 0 en 1 die respectievelijk aangeven in welke mate Henk ‘jong’, ‘middelbaar’ en ‘oud’ is. We noemen ‘leeftijd’ dan een *linguistic variable*. Omdat we de leeftijd min of meer met woorden omschrijven. Het omzetten van een gewone variabele naar een linguistic variable wordt ook wel *fuzzification* genoemd. We vervangen de leeftijd 45 bijvoorbeeld door het drietal: (0.2, 0.8, 0.7) wat wil zeggen dat Henk jong genoemd kan worden met zekerheid 0.2, middelbaar met 0.8 en oud met 0.7.

Fuzzy control

Voor wiskundigen en logici heel interessant allemaal misschien, maar het blijkt dat dit vakgebied juist vooral in de praktijk erg goed toegepast kan worden. Er zijn talloze elektrische apparaten op de markt (van wasmachines tot treinen en zelfs hele fabrieksprocessen) die daadwerkelijk gebruik ma-

ken van fuzzy logic. Dit wordt *Fuzzy Control* genoemd. Men moet hierbij denken aan programma's die opdrachtregels bevatten van de vorm:

```
IF temperature IS high AND speed IS low
THEN increase pressure slightly
```

Hier is 'temperature' een linguistic variable waarvan bekeken wordt in hoeverre die 'high' is en is 'speed' een linguistic variable waarvan bekeken wordt in hoeverre die 'low' is. De vraag is nu waarom we deze vaagheid gebruiken in plaats van een klassieke programmeerregel zoals:

```
IF temperature > 80 AND speed < 25
THEN pressure++
```

Welnu, Fuzzy Control heeft een aantal voordelen. Zo zou dit bovenstaande klassieke systeem bijvoorbeeld een abrupte sprong maken als de temperatuur boven de gegeven grenswaarde van 80 zou komen. Fuzzy control laat echter toe dat het systeem nu een geleidelijke verandering ondergaat wanneer de temperatuur stijgt: des te meer de temperatuur 'high' is, des te meer de druk verhoogd wordt. Dit soort geleidelijke veranderingen zijn in principe ook wel zonder fuzzy logic te realiseren, maar voor complexe systemen zouden daar zeer ingewikkelde differentiaalvergelijkingen aan te pas komen die ten eerste heel veel rekentijd zouden kosten en die tweede misschien zelfs wel helemaal niet oplosbaar zijn.

Bovendien is het misschien van tevoren nog helemaal niet duidelijk bij welke waarden van 'temperatuur' het systeem zich moet aanpassen. Fuzzy control laat toe dat we eerst het programma schrijven zonder expliciete getalswaarden in te voeren en het systeem pas later door middel van 'trial and error' laten leren wat we precies onder 'hoge temperatuur' moeten verstaan. Met andere woorden: de precieze waarheidsfuncties hoeven achteraf pas gedefinieerd te worden als alle code al geschreven is en kunnen eenvoudig aangepast worden als dat nodig blijkt te zijn. Het systeem is hiermee veel flexibeler geworden.

Maar hoe werkt dat nou precies? Het proces van fuzzy control vindt in drie stappen plaats: de eerste stap is fuzzification: de analoge meetwaarden die we met een thermometer of een willekeurige andere sensor hebben gevonden worden met behulp van

bepaalde waarheidsfuncties vertaald naar linguistic variables. Bijvoorbeeld: We meten een temperatuur van 45 graden. Deze krijgt voor de termen 'low', 'moderate' en 'high' de waarden 0.2, 0.8 en 0.3 toegekend. Vervolgens worden de opdrachtregels uitgevoerd. Hoe hoger de temperatuur, hoe meer de druk wordt opgevoerd dus na het nalopen van alle programmeerregels blijkt of de output-variabele (in dit voorbeeld de druk) verhoogd of verlaagd moet worden en in welke mate. Neem het volgende voorbeeld:

```
IF x
THEN pressure IS low
IF y
THEN pressure IS high.
```

x en y zijn hier twee 'fuzzy' logische uitspraken die een waarheidswaarde tussen 0 en 1 hebben, bijvoorbeeld $x = 0.2$ en $y = 0.9$. Voor pressure hebben we twee waarheidsfuncties: 'low' en 'high'. De waarheidsfunctie voor 'low' wordt dan met 0.2 vermenigvuldigd en de waarheidsfunctie voor 'high' wordt met 0.9 vermenigvuldigd (althans, in dit voorbeeld werkt het zo, in het algemeen zijn er vele manieren om fuzzy control te implementeren).

Deze twee functies worden bij elkaar opgeteld zodat we nu één functie hebben die voor elke mogelijk output-waarde van 'pressure' een waarheidswaarde geeft. Uiteindelijk moet er natuurlijk wel één echte exacte waarde uitkomen. Het proces waarbij we de linguistic variable weer naar een gewone waarde omzetten wordt *defuzzification* genoemd. Ook dit kan op vele manieren, we kunnen bijvoorbeeld kijken voor welke druk deze functie zijn maximum aanneemt en dat wordt dan uiteindelijk de druk van het systeem.

Het is interessant om te zien hoe we machines met behulp van dit soort logica op een steeds 'menselijker' manier kunnen laten werken. Het is nog de vraag hoe ver dit nog door zal gaan. Misschien zeggen we in de toekomst wel tegen onze auto dat hij 'flink hard' moet rijden of zeggen we tegen onze mp3-speler dat hij een 'rustig' muziekje af moet spelen. Wie weet, maar voorlopig blijft onze toekomst fuzzy.