

LOGICA

Copyright: Tom Vanhoutte

INHOUDSTAFEL

LOGICA	3
1. VOORKENNIS	3
2. INLEIDING	3
3. LOGICA EN KRITISCH DENKEN	3
4. RELATIE MET COMPUTERS	4
5. TURING MACHINE	4
6. ALGORITMES	5
7. WAT IS EEN LOGISCH ARGUMENT?	7
8. WAT ZIJN STATEMENTS?	8
9. STATEMENT OPERATOREN?	8
10. INDUCTIEF OF DEDUCTIEF REDENEREN?	8
10.1. <i>deductief redeneren</i>	8
10.2. <i>Inductief redeneren</i>	9
11. VENN DIAGRAM?	13
11.1. <i>Venn Diagrammen voor I/O</i>	13
11.2. <i>Venn Diagrammen voor universele verklaringen</i>	15

Copyright: Tom Vanhoutte

LOGICA

1. Voorkennis

Deze cursus vereist geen voorkennis.

1. Inleiding

Deze cursus leert u de fundamenteën van logica en laat u zien hoe logisch denken en logische concepten werden toegepast bij het ontwerpen en bouwen van digitale machines. Door vaardigheden in logische analyse te verwerven, verkrijgt u niet alleen belangrijke levensvaardigheden, maar zult u ook beter begrijpen hoe computers werken, waardoor u op de lange termijn een betere technicus wordt. Zoals we later zullen zien, is logica een wetenschap die regels voor goed geordend denken uiteenzet, terwijl kritisch denken voornamelijk de toepassing van die regels is. Veel inleidende studenten hebben de neiging om te denken dat logica wordt opgelegd aan het menselijke denken. Het is beter om te denken aan logica als iets dat filosofen en wiskundigen hebben ontdekt over het menselijke denken. Dus logische regels zijn als natuurwetten: ze werden gevormd door jarenlange observatie van hoe denken werkt wanneer het goed werkt.

2. Logica en Kritisch denken

Hoewel niemand een handige formule heeft ontdekt die alles samenvat, hebben veel mensen door de eeuwen heen diep nagedacht over het denken. Een aantal normen of criteria zijn voorgesteld, getest en goed bevonden als betrouwbare richtlijnen voor gezond verstand en het bereiken van waarheid, of het corresponderen met de realiteit. Bekijk als voorbeeld het statistische denken. Om meer te weten te komen over een grote populatie van dingen, onderzoeken we soms een steekproef en concluderen we iets over de populatie als geheel op basis van wat we in de steekproef waarnemen. Het is logisch dat hoe groter de steekproef is, met betrekking tot de hele populatie, hoe groter de kans dat de conclusie waar is. Het is ook logisch dat hoe meer willekeurig de steekproef wordt geselecteerd, hoe waarschijnlijker de conclusie is dat deze klopt.

Daarom is kritisch denken meer dan alleen kritisch denken. Het is veeleer het denken op basis van criteria die zijn getest en die betrouwbare gidsen zijn voor een gezond oordeel en het bereiken van de waarheid over de kwestie die wordt overwogen. Iemand die elke belangrijke beslissing neemt op basis van "het gooien van de botten" houdt zich bezig met criteriaal denken: het gooien van de botten is het criterium van deze persoon. (Het gooien van de botten is een oude vorm van waarzeggerij waarbij dierlijke botten op een mat worden gegooid en het patroon wordt geïnterpreteerd, meestal door een sjamaan, spirituele lezer of waarzegger van een soort.) Maar zo'n persoon is geen kritische denker -

tenminste zoals de woorden kritische denker tegenwoordig worden gebruikt - voor zijn of haar criterium is niet getest en bleek een over het algemeen betrouwbare gids voor waarheid en gezond verstand te zijn.

3. Relatie met computers

Een van de belangrijkste ideeën die deze cursus zal onderzoeken, is de relatie tussen logisch en kritisch denken en informatica. Laten we eerst eens kijken naar hoe computers werken om die relatie tot stand te brengen en dan duiken we in de basisprincipes van de formele logica.

4. Turing Machine

Om beter te begrijpen hoe computers "denken", kunnen we beginnen met de eenvoudigste versie van een computer: de Turing-machine. Een Turing-machine is niet echt een machine, maar een model om te beschrijven hoe computers doen wat ze doen.

Het model is bedacht door computerwetenschapper Alan Turing, die door velen wordt beschouwd als de vader van digitaal computergebruik. Turing realiseerde zich dat alle rekenproblemen kunnen worden opgesplitst in een heel eenvoudige taal, een digitale taal: 0 en 1. Dat is vrij eenvoudig, toch? Een manier om na te denken over het inzicht van Turing is in termen van twee toestanden zoals 'aan' en 'uit', 'waar' en 'onwaar', in 'en' uit 'of' ja 'en' nee '. Je kunt een Turing machine maken uit alles dat in twee verschillende toestanden kan bestaan. Eén filosoof suggereerde dat een Turing-machine gemaakt kan worden van vogels die pikken: wanneer het hoofd van de vogel omhoog is, staat het in de "0" -positie en wanneer het omlaag is, bevindt het zich in de "1" positie.

In veel opzichten is de lichtsakelaar op uw muur een digitale machine. Wanneer de lichtsakelaar aan is, is deze in één staat en wanneer deze uit is, bevindt deze zich in een andere - en dat zijn de enige twee toestanden waar de lichtsakelaar zich in kan bevinden. Hoewel niet veel informatie kan worden gecommuniceerd met die schakelaar, alleen door te kijken. Daarmee kunnen we zien in welke toestand het licht is (neem even aan dat je het daadwerkelijke licht niet ziet dat de schakelaar bedient).

Alle moderne computers zijn in essentie erg complex. Turingmachines die schakelaars draaien - heel veel schakelaars - gaan heel snel aan en uit om alle magische dingen te doen die een computer doet.

Hoe kan een switch alle interessante afbeeldingen maken, gegevens in een spreadsheet verwerken, tekstberichten en digitale telefoongesprekken maken, holografische afbeeldingen produceren en alle andere coole dingen die onze computers doen? Het is een

beetje ingewikkelder dan een eenvoudige lichtschakelaar (je wist waarschijnlijk al dat dit zou komen).

De basis van computersystemen en de logica waaruit ze bestaan, zijn opgebouwd uit sets van schakelaars die allemaal samenwerken. Laten we ons voorbeeld van de lichtschakelaar uitbreiden om te zien hoe dit werkt. Stel dat we twee lichtschakelaars hebben die bij gebruik in verschillende combinaties verschillende kleuren lichten produceren. Dit zijn de combinaties die we kunnen maken (met 0 voor uit en 1 voor aan):

Switch 1	Switch 2	Kleur
0	0	Uit
0	1	Rood
1	0	Groen
1	1	Blauw

In de taal van bits, is dit een twee-bits systeem. We hebben twee binaire toestanden die samenwerken om vier verschillende combinaties te vormen die een uitschakeltoestand produceren en drie verschillende gekleurde lichten op toestanden. Moderne computers gebruiken acht bits in combinatie om de fundamentele eenheid van digitale informatie te vormen. Dit apparaat wordt een byte genoemd.

In de allereerste moderne computers werd een byte met zijn bits in verschillende combinaties gebruikt om één enkel teken van tekst weer te geven. De onderstaande tabel toont de bytevoorstelling van de letter "A" in binaire taal:

0100 0000 = 2 sets van elk 4 bits = 8 bits = 1 byte

Wanneer de CPU (centrale verwerkingseenheid - de "hersenen" van de computer ") die digitale combinatie ontvangt, weet deze dat ze een ander signaal naar het scherm moet sturen om de letter "A" te tekenen.

5. Algoritmes

Computers zijn met name handig omdat ze herhaalbare taken op een voorspelbare manier uitvoeren (zelfs als u gefrustreerd bent over uw computer omdat deze niet doet wat u wil). Wanneer u een tekstverwerker opent of een computerspel speelt, werkt de computer tegen een reeks opdrachten of instructies die een team van programmeurs heeft geschreven en die instructies vertellen de computer wat te doen en wanneer hij deze moet doen. Deze instructies doen altijd hetzelfde en wanneer het programma goed geschreven is, doet het ze consistent en foutloos.

Een algoritme is een methodische reeks stappen die kan worden gebruikt om berekeningen

te maken, problemen op te lossen en beslissingen te nemen. Een algoritme is geen specifieke berekening, maar de methode die is gevolgd bij het maken van de berekening.

Iedereen gebruikt in het dagelijkse leven reeds onbewust algoritmes.

Voorbeeld: BROODJE KAAS

Stap 1: zet de ingrediënten uiteen: kaas, smeerboter, brood

Stap 2: open de pot met smeerboter en jam

Stap 3: Leg twee stukjes brood op een plat oppervlak

Stap 4: Smeer een dunne hoeveelheid smeerboter op een sneetje brood

Stap 5: Plaats een plakje kaas boven op de boterham met smeerboter

Stap 6: Plaats een snee brood op de andere

Omdat dit een zeer onnauwkeurige reeks instructies is, zal elke broodje met kaas een beetje anders er uit zien. Toch bereiken we op een zeer hoog niveau in wezen dezelfde resultaten. Computers volgen vergelijkbare instructies op, maar computersystemen zijn zeer nauwkeurig in de instructies die zij volgen. Deze resulteren over het algemeen in een uitvoer die veel nauwkeuriger is.

Veronderstel het volgende in programmacode:

Copyright: Tom Vanhoutte

```
for (x=0; x < 7; x++)  
{  
    If x is Even Then  
        print "Eet pasta als maaltijd"  
    Else  
        print "Eet een salade als maaltijd"  
}
```

Van programmacode naar logica:

Stelling 1: De dag is oneven of de dag is even

Stelling 2: het is niet zo dat de dag even is

Conclusie: daarom is de dag niet gekend

Dit is een goed begin, maar dit argument vertelt ons niet wat we elke dag zullen eten. Dus we moeten een ander syllogisme toevoegen, een modus ponens genaamd, om ons te helpen met dat probleem. Dit syllogisme voor ons argument ziet er als volgt uit:

Stelling 1: als de dag even is, eten we pasta

Stelling 2: de dag is even

Conclusie: daarom eten we pasta

Natuurlijk hebben we voor de oneven dagen een ander syllogisme nodig:

Stelling 1: als de dag niet gekend is, eten we salade

Stelling 2: De dag is even

Conclusie: Daarom eten we salade

We kunnen deze nu combineren in een enkel argument:

Gegeven: de huidige dag van de week

Stelling 1: De dag is oneven of de dag is even

Stelling 2: als de dag even is, eten we pasta

Stelling 3: als de dag oneven is, eten we salade

Stelling 4: het is niet zo dat de dag [oneven / even] is

Uitgangspunt 5: De dag is [even / oneven]

Conclusie: daarom eten we [pasta / salade]

Copyright: Tom Vanhoutte
Omdat onze disjunctieve syllogism-regel zegt dat als een van de opties niet waar is, de ander waar moet zijn, we de dagen van de week in gebouwen 4 en 5 en de maaltijd die we eten in de conclusie combineren en de dag van de week bepalen welke we kiezen.

6. Wat is een LOGISCH argument?

Voorbeeld:

Een groep personen die in een discussie aan het roepen zijn naar elkaar met elk hun eigen argumenten zijn per definitie geen LOGISCHE argumenten.

De definitie van een argument is het leveren van één of meerdere redenen die iemand van uw statement kan overtuigen.

Logische argumentatie bestaat uit 3 zaken:

1. Stelling: reden

2. Conclusie: bewijs dat de stelling juist is

3. Logische relatie: connectie tussen de stelling en de conclusie.

Voorbeeld van een niet logisch argument:

Morgen komt de zon op en Justin Timberlake zijn gazon is groen.

Dit zijn stellingen die de conclusie hebben die juist zijn, maar geen logische relatie hebben.

Dit is dus geen volledig logisch argument.

7. Wat zijn statements?

We weten dus dat een argument bestaat uit een stelling, conclusie en logische relatie.

Kunnen we een zin gebruiken als een stelling? Het antwoord is hier steeds NEEN!

In een zin gebruiken we verklarende statements of gewoon statements.

Deze statements zijn niet altijd WAAR.

Voorbeelden:

- Alle Belgen leven in België
- De meeste bedrijven in Kortrijk zijn afvalverwerkingsbedrijven
- Apple haalde in 2017 een hogere omzet dan Microsoft

Alle voorbeelden proberen een waarheid te definiëren. De eerste 2 bevatten onwaarheden en zijn statements, de onderste is waar en is een verklarende statement.

Alleen verklarende statements kunnen we als statement in logica aanvaarden.

Goed voorbeeld:

- Na is een element in de tabel van Mendeljev

Het is de bedoeling om steeds verklarende elementen als stellingname te vinden, maar dit is niet altijd mogelijk

Welke stellingen zijn verklarend en welke niet?

Stelling: Wat is de tijd?

Stelling: Open het venster?

Stelling: Ik beloof.

Stelling: Mayonaise is wit. => dit is het enige verklarende argument.

8. Statement operatoren?

Meerdere stellingen kunnen tegelijk worden vergeleken met operatoren.

Bijvoorbeeld:

Tom **en** Tim zijn thuis

Tom **of** Tim is thuis

9. Inductief of deductief redeneren?

Inductief redeneren is het zoeken naar trend of patroon die we uiteindelijk veralgemenen.

Bij het veralgemenen weten we NIET ZEKER of de conclusie van inductief redeneren waar zal zijn.

Bij deductief redeneren nemen we een set van data of set van feiten die waar zijn om uiteindelijk naar een resultaat te komen waarvan we weten dat die ook waar zijn. We leiden dus feiten af van bestaande feiten. Dit afleiden noemen we de minor-stelling van de major-stelling afleiden.

9.1. deductief redeneren

Voorbeeld:

Stel dat je een stuk fruit wil eten en je opent je koelkast.

In de koelkast zal je een stuk prei een appel en sla terugvinden. Je weet dat prei en sla geen fruit is. Je weet ook dat alle appels fruit zijn en dat een granny Smith een appel is. Daar leiden we uit af dat Granny Smith fruit is. Dit noemen we **sylogisme**, een vorm van deductief redeneren.

Sylogismen zijn dus deductieve argumenten die in volgende vorm worden geschreven:

A is B, C is A, daarom is C ook B.

Wanneer we bovenstaande nu in Stellingen en Conclusies zouden onderbrengen dan krijgen we het volgende:

Stelling: Alle appels zijn fruit

Stelling: een Granny Smith is een appel

Conclusie: een Granny Smith is een fruit

Voorbeeld:

Stelling: ieder mens moet vrij zijn (major-stelling)

Stelling: jij en ik zijn mensen (minor-stelling)

Conclusie: wij moeten vrij zijn.

Copyright: Tom Vanhoutte

9.2. Inductief redeneren

Voorbeeld:

Stelling: een eend in het park is wit. De tweede, derde en vierde eende zijn ook wit.

Conclusie: alle eenden in het park zijn wit. Ook dit is een sylogisme, maar een deductief sylogisme.

Voorbeeld:

Ik vertrek elke morgen om 8 uur naar het werk. Ik ben altijd op tijd. Ik ga er dan ook van uit dat wanneer ik op 8 uur vertrek, altijd op tijd zal zijn.

Voorbeeld:

De stoelen in de eetkamer zijn rood. De stoelen in de woonkamer zijn rood. Alle stoelen in het huis zijn rood.

Voorbeeld:

Probleem: bevolkingsgroei schatten voor de komende jaren.

Jaar	Bevolking
------	-----------

1950	7500
------	------

1960	7900
------	------

1970 8300
1980 8900
1990 9200
2000 10000
2010 10678
2020

Wanneer we hier een trend in de stijgingen proberen te zoeken zullen we aan een gemiddeld stijgingspercentage komen. We weten niet of de stijging die we berekenen voor 2020 ook waar zal zijn.

Is onderstaand voorbeeld deductief redeneren of inductief redeneren?

6,9,12,15, ...

Hier nemen we de data set van de feiten die waar zijn om uiteindelijk tot de conclusie te komen die ook waar is.

OEFENEN: SYLLOGISMEN

1.

Stelling 1: Sommige huiseigenaren zijn niet Belgisch.

Stelling 2: Geen van de mathematici is een huiseigenaar.

De conclusie is: Sommige mathematici zijn niet Belgisch.

Deze conclusie is **juist**

Deze conclusie is **onjuist**

2.

Stelling 1: Geen van de acteurs is een slaapwandelaar.

Stelling 2: Alle Spanjaarden zijn slaapwandelaars.

De conclusie is: Sommige acteurs zijn Spanjaarden.

Deze conclusie is **juist**

Deze conclusie is **onjuist**

3.

Stelling 1: Alle postzegelverzamelaars zijn chemici.

Stelling 2: Sommige postzegelverzamelaars zijn Braziliaans.

De conclusie is: Geen van de chemici is een Braziliaan.

Deze conclusie is **juist**

Deze conclusie is **onjuist**

4.

Stelling 1: Alle canadezen zijn rechtshandig.

Stelling 2: Alle rechtshandigen zijn opticiens.

De conclusie is: Sommige opticiens zijn Canadees.

Copyright: Tom Vanhoutte

Deze conclusie is **juist**

Deze conclusie is **onjuist**

5.

Stelling 1: Alle bloeddonoren zijn Oostenrijkers.

Stelling 2: Geen van de bloeddonoren is een bewaker.

De conclusie is: Sommige bewakers zijn geen Oostenrijkers.

Deze conclusie is **juist**

Deze conclusie is **onjuist**

6.

Stelling 1: Geen van de Italianen is een wijnrinker.

Stelling 2: Alle dichters zijn wijnrinkers.

De conclusie is: Alle dichters zijn Italiaans.

Deze conclusie is **juist**

Deze conclusie is **onjuist**

7.

Stelling 1: Geen van de bijenhouders is een stratenmaker.

Stelling 2: Alle Argentijnen zijn bijenhouders.

De conclusie is: Sommige stratenmakers zijn niet Argentijns.

Deze conclusie is **juist**

Deze conclusie is **onjuist**

8.

Stelling 1: Geen van de historici is een niet-roker.

Stelling 2: Geen van de niet-rokers is Mexicaans.

De conclusie is: Alle historici zijn Mexicaans.

Deze conclusie is **juist**

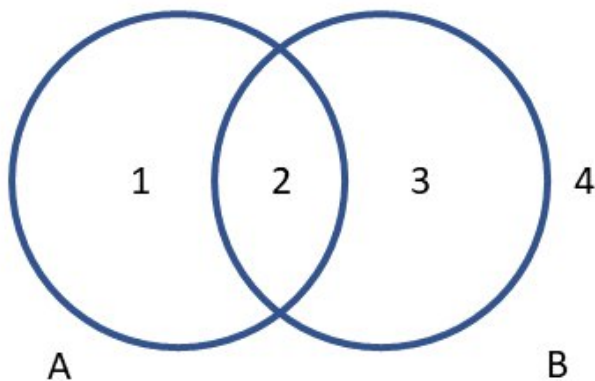
Deze conclusie is **onjuist**

Copyright: Tom Vanhoutte

10. VENN DIAGRAM?

Een Venn-diagram is een reeks overlappende cirkels, waarbij elke cirkel staat voor een categorie of klasse van dingen en waarbij de cirkels zo zijn gerangschikt dat ze de logische relaties tussen de weergegeven categorieën weergeven. Een Venn-diagram voor een enkele categorische instructie heeft twee overlappende cirkels, een voor de categorie met de naam van de betreffende term en een voor de categorie met de predikaatterm, die vier verschillende gebieden of regio's produceert.

Beschouw de volgende categoriale verklaring, All As are Bs en het volgende diagram:



De A-cirkel vertegenwoordigt alle dingen in het universum die A zijn, terwijl de B-cirkel alle dingen vertegenwoordigt die B. zijn. Als de As aardvarkens zijn, vertegenwoordigt de A-cirkel alle aardvarken; als de Bs bruine dingen zijn, vertegenwoordigt de B-cirkel alle dingen die bruin zijn, enzovoort. Dus:

Regio 1 vertegenwoordigt dingen die A zijn, maar niet B.

Regio 2 vertegenwoordigt dingen die zowel A als B. zijn

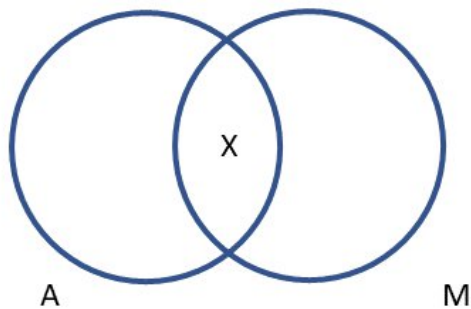
Regio 3 vertegenwoordigt dingen die B zijn, maar geen A.

Regio 4 (het gebied buiten de cirkels) vertegenwoordigt alles dat noch A noch B is.

10.1. Venn Diagrammen voor I/O

De 'I'-verklaring

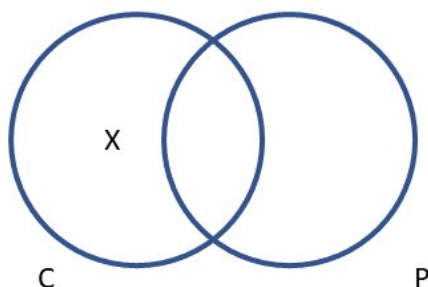
Een specifieke verklaring is alleen waar als de voorwaarden verwijzen naar een of meer werkelijk bestaande dingen, met behulp van duidelijke afkortingen (A voor "aardvarken," etc.), het Venn-diagram voor de I-verklaring "Sommige aardvarkens zijn zoogdieren" ziet er daarom als volgt uit:



Je zou het handig kunnen vinden om je voor te stellen dat alle aardvarkens in het universum afgerond zijn en rondscharrelen voor mieren in de eerste cirkel - de A-cirkel. Stel je ook voor dat alle zoogdieren in het universum verzameld en rusten binnen de tweede cirkel - de M-cirkel. De X geeft aan dat er ten minste één ding bestaat in het deel van het aardvarkgebied dat overlapt met het zoogdiergebied. Dat wil zeggen, sommige aardvarkens zijn zoogdieren. Merk op dat we geen X in regio 3 hebben gezet. Onafhankelijk van het diagram, kunnen we weten dat er wezens zijn in regio 3 - er zijn zoogdieren die geen aardvarkens zijn - maar de I-verklaring vertelt ons dit niet en we willen diagram alleen de informatie die de verklaring geeft, en niets meer. Regio 3 krijgt daarom geen X.

De 'O'-verklaring

Het volgende is het Venn-diagram voor de O-zin "Sommige katten zijn geen huisdieren":



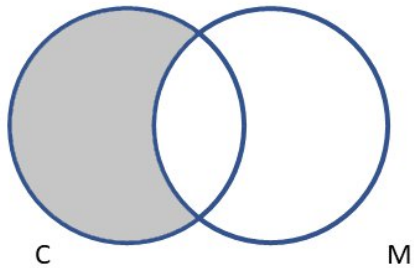
Regio 1 staat voor dingen die katten zijn maar geen huisdieren, regio 2 staat voor katten die huisdieren zijn en regio 3 staat voor huisdieren die geen katten zijn. De X vertelt ons dat er ten minste één "binnen" gebied 1 is.

Opmerking: als er geen X of schaduw in een gebied verschijnt, betekent dit niet dat er niets in het gebied bestaat; het geeft eerder aan dat er niets bekend is over het gebied. Het betekent alleen dat we geen informatie over het gebied hebben. Dus, voor zover we weten, is het gebied mogelijk leeg of bevat het een of meer dingen.

10.2. Venn Diagrammen voor universele verklaringen

De 'A'-verklaring

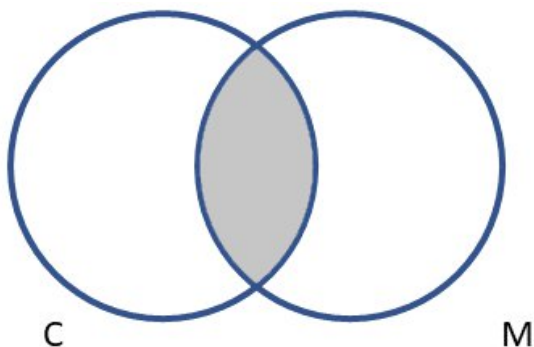
Venn-diagram voor de A-verklaring "Alle katten zijn zoogdieren", ziet er als volgt uit:



Hier plaatsen we regio 1 voor een lege set. Er zijn namelijk geen katten die geen zoogdieren zijn - die categorie is "leeg" omdat onze A-verklaring ons vertelt dat alles wat een kat is ook een zoogdier is.

De 'E'-verklaring

De corresponderende universele negatieve of E-verklaring in dit geval wordt begrepen als te beweren: "Niets is een kat en een zoogdier" of "Geen katten zijn zoogdieren." Het Venn-diagram ziet er als volgt uit:



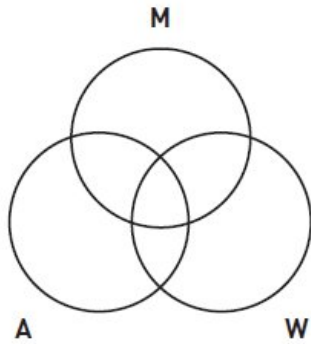
III VANHOULLE

In dit diagram verduisteren we alleen regio 2 of het gebied dat zowel katten als zoogdieren overlapt. We verdoezelen regio 3 niet, omdat we geen beweringen doen over de hele klasse van zoogdieren. Onze E-verklaring vertelt ons alleen dat als iets een kat is, het geen zoogdier is. Dus die categorie is leeg of leeg.

Venn-diagram: geldige argumenten

In dit onderwerp lopen we door het proces van het maken van een Venn-diagram voor een geldig argument om te laten zien hoe het werkt.

Laten we beginnen met het syllogisme van Barbara



Alle zoogdieren zijn warmbloedige wezens.

Alle aardvarken zijn zoogdieren.

Daarom zijn alle aardvarken warmbloedige wezens.

Onze eerste stap is om het argument af te korten, de conclusie onderaan te plaatsen:

Alle M zijn W.

Alle A zijn M.

Alle A zijn W.

Ten tweede tekenen we de drie overlappende cirkels. Voor stap 3 labelen we de cirkels:

Copyright: Tom Vanhoutte