

leren. durven. doen.



# *Logica, Kritisch denken en Informatica*

# Module Logica

## DOELSTELLINGEN

- zelfstandig kunnen studeren (e-learning)
- Engelse taal gewoon worden
- Het leren denken als IT'er 'computer denken'
- Verschillende onderdelen in Logica:
  - **Basis van formele logica:** nodig voor het 'programmeer-denken'
  - **Categoriek denken & VENN diagrammen:** Object geöriënteerd ontwerp en programmatie
  - **Kritisch denken:** Software testen en selectief opzoeken van informatie

# Module Logica

## EVALUATIE

- Examen:

- 30-tal multiple choice vragen (vragen uit elke online-multiple choice test)

- Opdracht Logica: in te dienen ten laatste op **13/10/2020**

### Twee onderdelen:

1. **30-tal vragen:** antwoorden indienen (via Moodle)

2. **2. Eind-test online:**

<https://courses.edx.org/courses/course-v1:Microsoft+DEV262x+1T2018a/course/>  
**print screen maken van resultaat (beeldformaat) en uploaden via Moodle**

# **1: Deductie en Inductie**

**a. Inductieve vs deductieve redeneringen**

**b. Deductie en deductieve syllogismen**

**i. Geldig en ongeldig**

**ii. Vorm van logische redenering**

**iii. Deductieve redeneringen en computer code**

**iv. Waarheidstabel opstellen**

**c. Categorieke logica**

**d. Venn diagrammen**

# **3: Inductief redeneren en Software Testen: Hoe kritisch denken?**

**a. De logica van wetenschap**

**b. Zoeken naar de oorzaak (methoden van Mill)**

**c. Kritisch denken en modern wetenschap**

**d. Toepassen van kritisch denken op software testen**

# Inhoud Online Module Logica

## 1. Inleiding Logica

Wat is Logica?

Proposities en statements

Waarheidstabellen opstellen

## 2. Deductie en Inductie

Inductieve vs deductieve redeneringen

Deductie en deductieve syllogismen

Geldig en ongeldig

Vorm van logische redenering

Deductieve redeneringen en computer code

## 3. Categorieke logica (denken in categorieën)

Venn diagrammen

## 4. Inductief redeneren en Software Testen:

### Hoe kritisch denken?

De logica van wetenschap

Zoeken naar de oorzaak (methoden van Mill)

Kritisch denken en modern wetenschap

Toepassen van kritisch denken op software testen

# 1 Inleiding: Wat is Logica?

- De studie van regels (syllogismen) die beschrijven hoe men correcte verbanden legt tussen (logische) uitspraken.
- Logica gaat over waarheid/onwaarheid van uitspraken
- uitspraak = bewering, kan waar of onwaar zijn in bep. Context

Sinaasappels zijn oranje

Het regent

Morgen gaat het sneeuwen

-> uitspraken zijn ofwel waar, ofwel onwaar, maar nooit allebei

# Propositie (uitspraak die waar/vals is)

propositie of een logische uitspraak=

Een uitspraak *met waar/onwaar eigenschap:*

*Bv Alle katten zijn zoogdieren*

*Sommige honden hebben vlooien*

*Geen enkele leeuw is geschikt als huisdier*

# Logische uitspraken (Statements)

Een **samengestelde logische uitspraak** zijn twee of meer enkelvoudige logische uitspraken die samengebracht worden door een **operator**.

- Jan is thuis EN Piet is thuis
- Er zit een gat in de vloer OF Ik begin dingen te zien
- Je krijgt een boete (true/false)
  - Je rijdt onder Invloed (true/false)
  - EN
  - Je moet in het zakje blazen(true/false)

Oefening:

Geef nog een OF voorbeeld.... Je krijgt een boete (true/false):

-....  
OF  
-....



# Deductie en Inductie

## Deductieve redenering

De logica waarbij men vanuit een algemene uitspraak vertrekt om een conclusie te vormen voor een specifiek geval

## Inductieve redenering

Men vertrekt vanuit een reeks van specifieke gevallen om tot een algemene uitspraak te komen.

De conclusie van een inductieve bewering is *nooit gegarandeerd*

# Deductief redeneren

## Voorbeeld

Alle mensen zijn sterfelijk (premissie)  
Socrates is een mens (premissie)

Dus, Socrates is sterfelijk (conclusie)

Dit is een deductief syllogism

Syllogisme: een logische redenering die uit 2 beweringen (premissen) bestaat, gevolgd door een daaruit logisch afgeleide conclusie

# Zwakke of Sterke Inductieve redenering

## Zwakke redenering

1. Deze mand bevat 100 appels
2. 3 willekeurig geselecteerde appels uit de mand zijn rijp
3. Dus, waarschijnlijk zijn alle appels in de mand rijp

## Sterke redenering

1. Deze mand bevat 100 appels
2. 80 willekeurig geselecteerde appels uit de mand zijn rijp
3. Dus, waarschijnlijk zijn alle appels in de mand rijp

# Wanneer deductieve redenering

- Een deductieve redenering garandeert dat **wanneer alle premissen waar** zijn, de dat de **conclusie steeds waar is**.
- Indien voor alle premissen de conclusie gegarandeerd is, dan is de redenering **geldig**
- *Indien de conclusie in een redenering niet gegarandeerd is, dan is de redenering **ongeldig***
- **(on)Geldigheid is niet hetzelfde als (on)waarheid**

# Geldigheid

- Bij een **geldige deductie** is het **onmogelijk** dat uit **ware** premissen een **valse** conclusie volgt

*Bij **inductie** is geldigheid niet van toepassing!*

***Geldigheid is enkel van toepassing bij deductieve redeneringen!***

# Voorbeelden deductieve redenering

- **Alle studenten eten pizza**
  - **Sabine is een student**
  - **=> Dus, Sabine eet pizza**
- 
- **Alle atleten gaan naar de fitness**
  - **Benny is een atleet**
  - **=> Dus, Benny gaat naar de fitness**

# Voorbeeld 3: deductieve geldigheid

## Geldige deductieve redenering

1. Alle nachtvlinders zijn nachtdieren
2. Alle nachtvlinders zijn insecten
3. Dus, sommige insecten zijn nachtdieren

In symbolen uitgedrukt:

Let  $V$ =nachtvlinder,  $N$ =nachtdier, and  $I$ =insecten

1. Alle  $V$  zijn  $N$
2. Alle  $V$  zijn  $I$
3. Dus, sommige  $I$  zijn  $N$

# Deductieve Syllogismen

## Modus Ponens

### Vorm van de redenering

If p then q

p

-----

q

$p \rightarrow q$

p

-----

q

Als Jan thuis is, is An thuis  
Jan is thuis

-----

Dus, An is thuis



# Deductive Syllogismen

- **Modus Tollens**
- **Vorm van de redenering:**

If p then q  
Not q

-----  
Therefore Not p

$p \rightarrow q$   
 $\sim q$

-----  
 $\sim p$

Als je slecht geslapen hebt, ben je moe  
Je bent niet moe

-----  
Dus, je hebt niet slecht geslapen

# Opgelet: foutieve redenering

- modus nonsens

If p then q  
q

-----  
~~Therefore p~~

$p \rightarrow q$   
q

-----  
 ~~$\emptyset$~~

Als je slecht geslapen hebt, ben je moe  
Je bent moe

-----  
~~Dus, je hebt slecht geslapen~~

# Deductieve Syllogismen

Reductio ad Absurdum: bewijs uit het ongerijmde

Vorm van de redenering:

Om *p* te bewijzen

Veronderstel  $\sim p$  (niet *p*)

Vanuit de veronderstelling, besluit *q*

Bewijs dat *q* niet mogelijk is, want is absurd

Conclusie: *p* moet waar zijn

# Voorbeeld reductio ad absurdum

- Stel: de aarde plat is
- dan kan men er af vallen.
- Maar niemand kan van de aarde af vallen,
- dus de aarde is niet plat.

# Deductieve Syllogismen

## Disjunctie

**p or q**

**Not p**

-----

**Therefore q**

**p v q**

**~p**

-----

**q**

Ofwel treden de Beatles op vanavond **ofwel** treden de Rolling Stones op

Vanavond treden de Beatles **niet** op

-----

**Dus de Rolling Stones treden vanavond op**

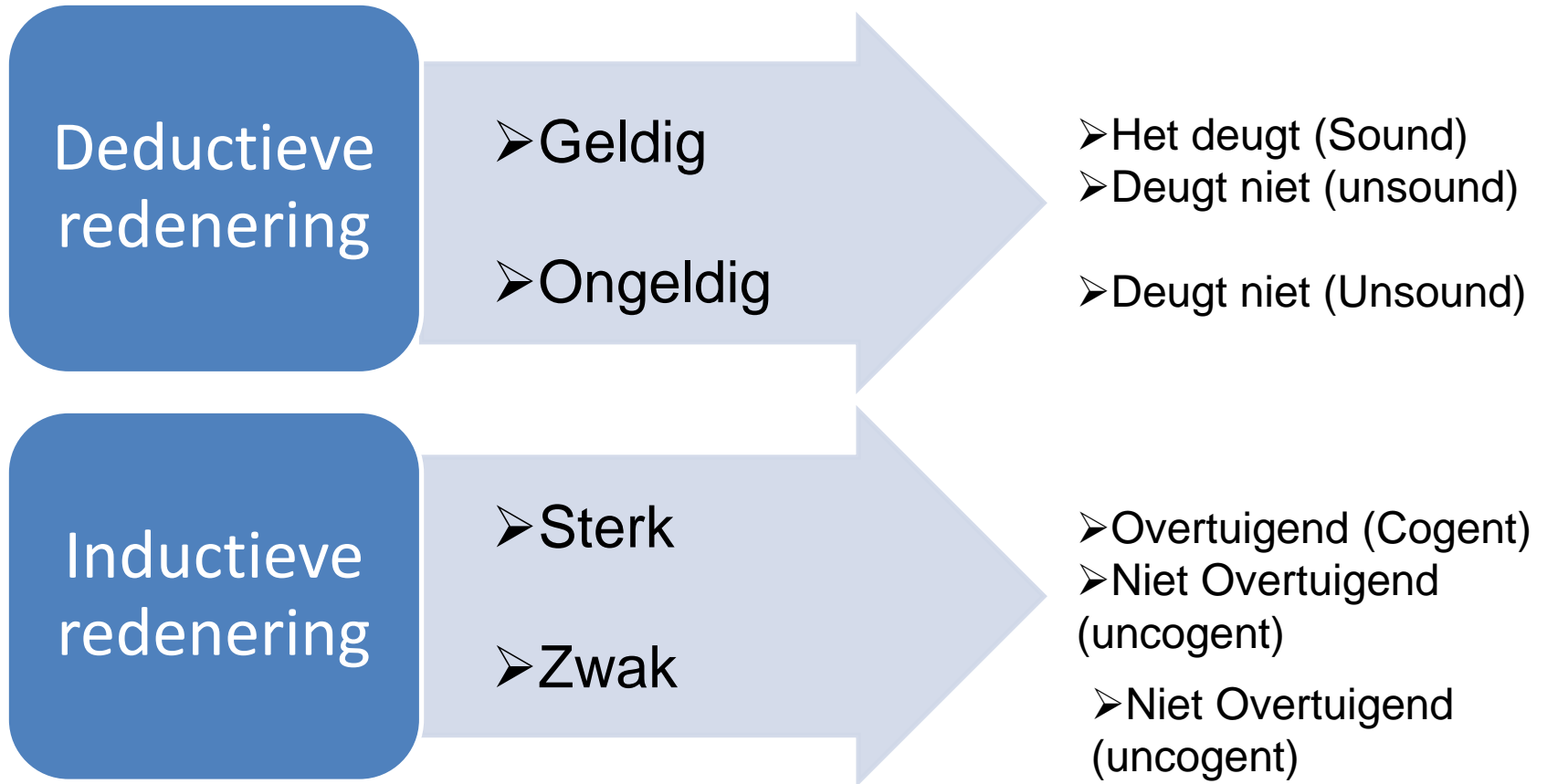
<https://learning.edx.org/course/course-v1:Microsoft+DEV262x+1T2018a/block-v1:Microsoft+DEV262x+1T2018a+type@sequential+block@b8f34124-9792-ad2d-ffd3-84b79faaa59a/block-v1:Microsoft+DEV262x+1T2018a+type@vertical+block@7e4c63d7-a12c-7909-2709-3fcbc2f10aa9>

# Deductieve Syllogisme

## Hypothetische syllogisme

<p>If p then q If q then r</p> <hr/> <p>Therefore if p then r</p>	<p><math>p \rightarrow q</math> <math>q \rightarrow r</math></p> <hr/> <p><math>p \rightarrow r</math></p>	<p>Als Jos gaat zwemmen, dan gaat Els zwemmen Als Els gaat zwemmen, dan gaat Koen zwemmen</p> <hr/> <p>Dus, Als jos gaat zwemmen, dan gaat Koen zwemmen</p>
---	--	---

# Samenvatting Evaluatie van redeneringen



# Herkennen van Deductief/Inductief

- **Deductieve Redenering**: redenering waarbij de conclusie *noodzakelijkerwijze* volgt uit de premissen
- **Inductieve Redenering**: redenering waarbij de conclusie *waarschijnlijk* volgt uit de premissen



# Indicatoren Deductie/Inductie

## **Deductieve Redenering**

- zeker
- absoluut
- noodzakelijkerwijze

## **Inductieve redenering**

- waarschijnlijk
- mogelijkerwijze

# Oefening Deductief of Inductief?

Elke les filosofie die ik tot nu toe heb gevolgd was leuk  
Dus, de volgende les zal ***waarschijnlijk*** leuk zijn.

Als het een les filosofie is, dan is het **zeker** dat de les altijd leuk is.  
De volgende les is een les filosofie  
Dus, de volgende les is leuk

# Evaluatie Deductieve redeneringen

## ➤ Geldigheid (valid)

- Zegt enkel iets over de **structuur** van de redenering
- Zegt enkel dat ***indien de premissen waar*** zijn, dan moet de conclusie ***noodzakelijk*** waar is

## ➤ Deugdelijkheid (sound)

- Gaat over **structuur én waarheid** van redenering
- De redenering is **geldig en alle premissen zijn waar**

# Evaluatie van Inductieve redeneringen

## ➤ Sterkte

- Zegt iets over de structuur van de redenering
- Betekent dat indien de premissen waar zijn, de conclusie ***waarschijnlijk*** waar is

## ➤ Overtuigend

- Zegt iets over de structuur én waarheid van de redenering
- De redenering is sterk en de premissen zijn allemaal waar

# Verkeerde redenering type 1

$p \rightarrow q$

$q$

~~Dus  $p$~~

**Als Jan thuis is, is An thuis**

**An is thuis**

~~**Dus, Jan is thuis**~~

## Verkeerde redenering type 2

$p \rightarrow q$

$\sim p$

~~Dus  $\sim q$~~

Als de hond honger heeft, dan blaft hij

De hond heeft geen honger

~~Dus de hond blaft niet~~

# Vorm van redenering (argument forms) en Computer Programma's

**Scratch**

<https://scratch.mit.edu/>

**Tutorial**

<https://www.youtube.com/watch?v=VlpmkeqJhmQ>

**Karel/Karen de Robot**

<https://www.kareltherobot.ch/karel.html>

# Operatoren

## Operatoren

NIET

EN

OF

ALS...DAN

XOR (exclusive OR)



# (Logische) Operatoren

**Monadische en Dyadische (unaire en binaire) operatoren**

***monadische operator* werken op een enkelvoudige logische uitspraak (statement).**

***dyadische operatoren* werken op 2 enkelvoudige statements.**

# Negatie – Enkelvoudige operator

**Symbool**

**~ NOT**  
**Voorbeeld**

**$\sim A$**

**Jan is niet thuis (het is niet het geval dat Jan thuis is)**

**Symbool in javascript, C, C#:**

**!**

**Voorbeeld:**

**!a**

# Conjunctie (EN)– dyadische operator

**Symbool**

**&& of ^**

**Twee delen in een conjunctie zijn de conjuncten**

**Voorbeeld**

A && B

Jan is thuis **EN** Piet is thuis

**Symbool in javascript, C, C#:**

**&&**

**Voorbeeld:**

a && b

# Conjunctie (OF)– dyadische operator

**Symbool**

**||** of **v**

**Twee delen in een conjunctie zijn de conjuncten**

**Voorbeeld**

**A v B**

**Jan is thuis **OF** Piet is thuis**

**Symbool in javascript, C, C#:**

**||**

**Voorbeeld:**

**a || b**

# ALS...DAN Conditie - implicatie - Operator

Symbool

$\supset$  of  $\rightarrow$

2 delen in de conditie:

De antecedent

De consequent

Voorbeeld:

- **J**  $\rightarrow$  **P** (J=antecedent en P is consequent)
- **Als** Jan thuis is **dan** is Piet thuis

Symbool in javascript, C, C#:

**if (...)** {...}

Voorbeeld:

```
if (J) {  
    P  
}
```

# Waarheidtabel

- Zal alle waar en vals- combinaties van statements en tabelvorm zetten en het resultaat van de operator
- 1 is Waar (true)
- 0 is Valse (false)
- **Voorbeeld: monadische operator NIET(!)**

Vb Waarheidstabel van statement

(A)n is **NIET** thuis

A	!A
1(true)	0(false)
0(false)	1(true)

# Waarheidstabel EN operator

## Voorbeeld: Dyadische operator EN(&&)

- Bv waarheidstabel van
- (A)n is thuis **EN** (B)ert is op café

A	B	A && B
1(true)	1(true)	1(true)
1(true)	0(false)	0(false)
0(false)	1(true)	0(false)
0(false)	0(false)	0(false)

# Waarheidstabel OF operator

## Voorbeeld: Dyadische operator OF(||)

- Bv waarheidstabel van
- (A)n is thuis **OF** (B)ert is op café

A	B	A    B
1(true)	1(true)	1(true)
1(true)	0(false)	1(true)
0(false)	1(true)	1(true)
0(false)	0(false)	0(false)



# Waarheidstabel XOR operator

Voorbeeld: Dyadische operator XOR

**A XOR B IS ENKEL TRUE WANNEER A verschillend is van B**

- Bv waarheidstabel van
- (A) **XOR** (B)

A	B	A XOR B
1(true)	1(true)	0(false)
1(true)	0(false)	1(true)
0(false)	1(true)	1(true)
0(false)	0(false)	0(false)

# Waarheidstabel voorbeelden

- | $D$ | $W$ | $D \text{ EN } W$ | $A$ | $B$ | ofwel $A$ ofwel $B$ | $A$ | $B$ | $\neg(A \wedge B)$ | $\neg A \vee \neg B$ |
|-----|-----|-------------------|-----|-----|---------------------|-----|-----|--------------------|----------------------|
| 0   | 0   | 0                 | 0   | 0   | 0                   | 0   | 0   | 1                  | 1                    |
| 0   | 1   | 0                 | 0   | 1   | 1                   | 0   | 1   | 1                  | 1                    |
| 1   | 0   | 0                 | 1   | 0   | 1                   | 1   | 0   | 1                  | 1                    |
| 1   | 1   | 1                 | 1   | 1   | 0                   | 1   | 1   | 0                  | 0                    |

$A$	$B$	$C$	$(A \wedge B) \vee C$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

$A$	$B$	$C$	$(A \wedge B) \vee (A \wedge C)$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

# Opdracht: een waarheidstabel opstellen

p	q	r	s	t				(q	&&	[(r		s)	&&	t])
T	F	F	T	T				F		F		T		T

# Categoriek redeneren- Inhoud

- **Wat is een categorieke redenering**
- **Sommige, Alle, geen**
- **Voorbeelden**
- **Venn diagrammen**

# Categorieke redenering

- Een **categorieke redenering (argument)** is een deductieve redenering die beweringen maakt over categorieën

# Categoriek redeneren

- Alle S zijn P
  - Alle katten zijn zoogdieren.
- Geen S zijn P
  - Geen katten zijn honden
- Sommige S zijn P
  - Sommige zoogdieren zijn katten
- Sommige S zijn geen P
  - Sommige zoogdieren zijn geen katten

# Categorisch denken: Sommige, Alle, geen

## Voorbeelden:

- Alle katten zijn zoogdieren
- Sommige katten zijn huisdieren
- Geen enkele leeuw is een huisdier

# Categorische syllogismen

## Categorische syllogismen

Bestaan enkel uit categorische uitspraken

### Voorbeeld:

Alle walvissen zwemmen

Alle walvissen zijn zoogdieren

Dus, sommige zoogdieren zwemmen

Alle huisdieren zijn dieren

Sommige slangen zijn huisdieren

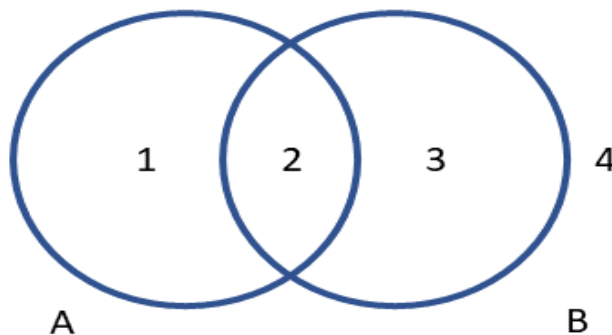
Dus, sommige slangen zijn dieren



# Categoriek denken - Venndiagram

Venndiagram is hulpmiddel(tool) bij het categoriek denken

Bv A=langharige dieren B=dieren met staart



Gebied 1 zijn A(langharige dieren), maar **niet** B(zonder staart)

Gebied 2 zijn A **EN** B (langharige dieren en dieren met staart)

Gebied 3 zijn dieren met staart (B), maar **niet** langharig (A).

Gebied 4 gebied buiten de 2 cirkels: alles dat **niet** langharig dier(A) **en niet** dier met staart is (B)

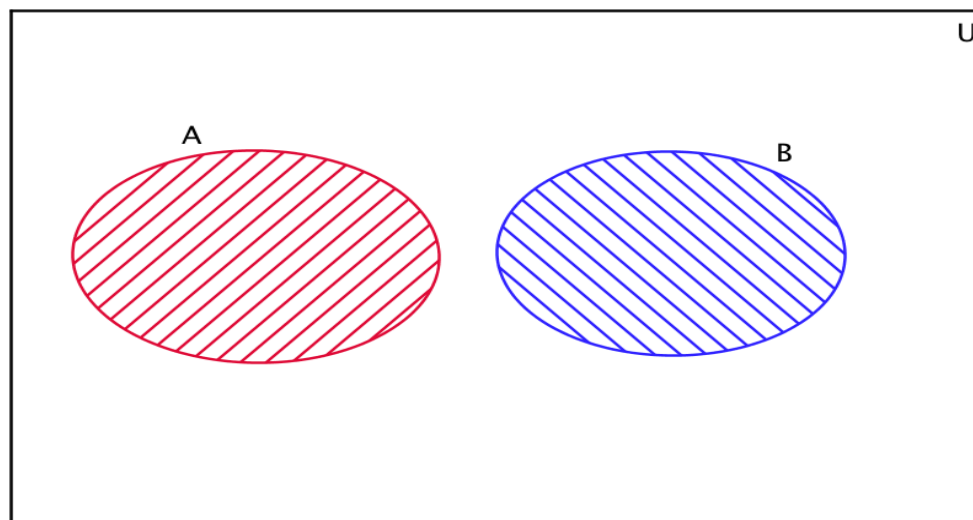
- **Vraag: In welk gebied zit Samson (Sint Bernard hond) ?**

# Venn diagrammen

A: Alle blonde mensen

B: Alle roodharige mensen

Betekenis: Geen A zijn B

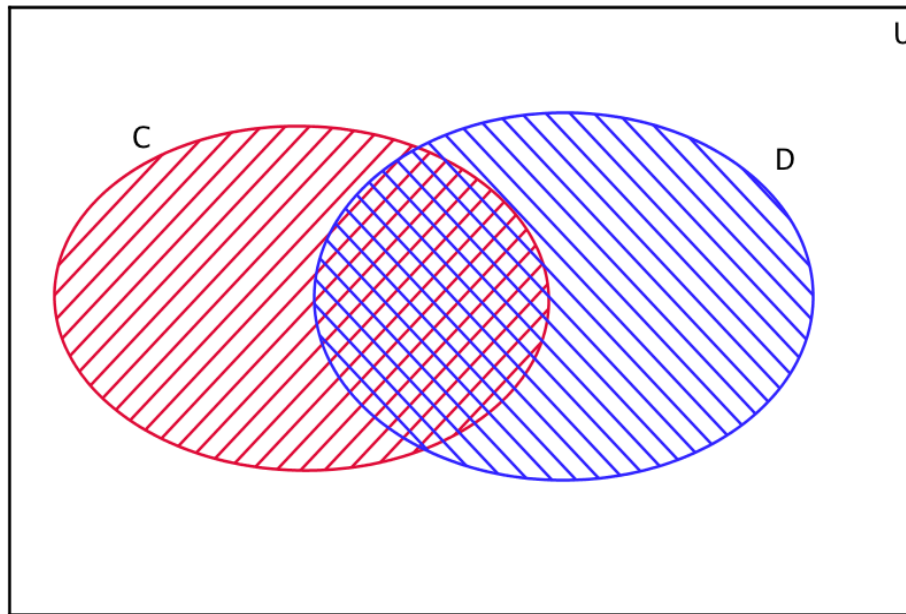


# Venn diagrammen

**C: langharige dieren**

**D: dieren met staart**

**Sommige D zijn B (of ten minste 1 C is ook C)**

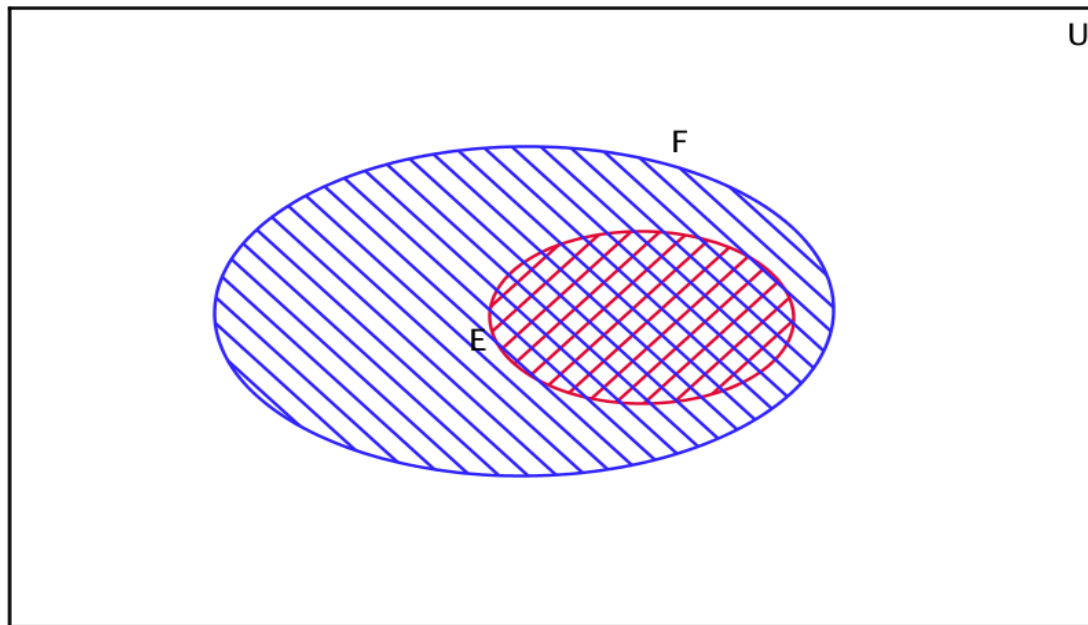


# Venn diagrammen

**F: de dieren**

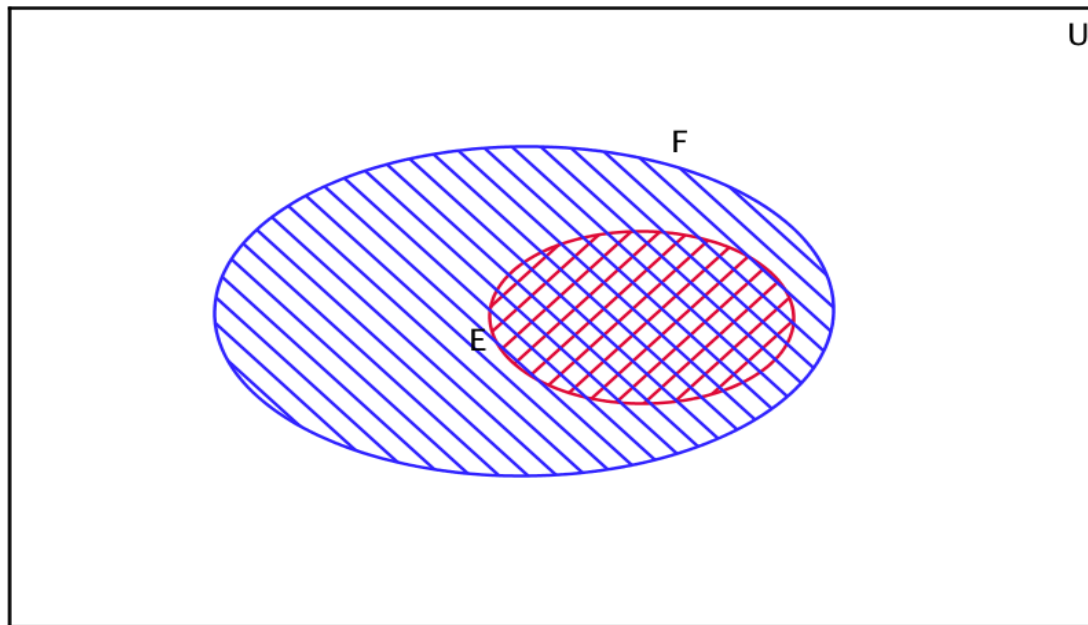
**E: de honden**

**Alle E zijn F**



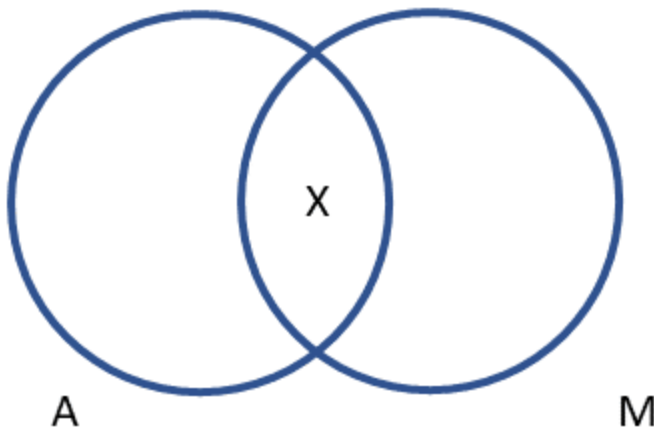
# Oefening Venndiagram

Alle katten(E) zijn zoogdieren(F)



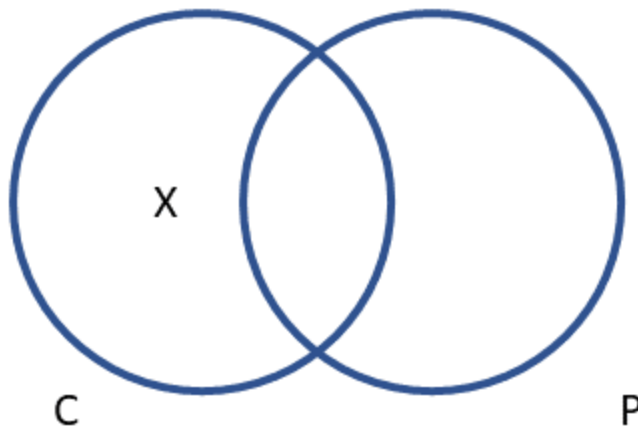
# Oefening Venndiagram

Sommige langharige(A) dieren hebben een staart(M)



# Oefening: Maak venndiagram

Sommige katten(C) zijn **geen** huisdieren(P)



# Opdracht Logisch redeneren /20 punten

1. Geef een voorbeeld van Deductieve redenering
2. Geef een voorbeeld van Inductieve redenering
3. Geef een voorbeeld van geldige deductieve redenering
4. Geef een voorbeeld van een zwakke inductieve redenering
5. Geef een voorbeeld van een geldige deductieve redenering die niet deugt
6. Geef een voorbeeld van een foutieve redenering van het type 1 & 2
7. Geef een voorbeeld van de modus tollens, modus ponens en de disjunctie
8. Geef 2 dyadische operatoren en hun symbolen in Javascript
9. Stel de waarheidstabel op van  $\neg(A \ \&\& \ B)$
10. Stel de waarheidstabel op van  $\neg A \ || \ \neg B$
11. Stel de waarheidstabel op van  $(q \ \&\& \ (r \ || \ s) \ \&\& \ t)$
12. Stel het venndiagram op voor «Geen katten zijn honden »
13. Stel het venndiagram op voor «Sommige katten zijn straatkatten»
14. Stel het venndiagram op voor « Sommige katten zijn geen straatkatten »
15. Stel het venndiagram op voor « Alle auto's zijn voertuigen»



leren. durven. doen.



# Kritisch denken

# Wat is kritisch denken?

## Verband tussen Logica en kritisch denken

### Logica

- zorgt ervoor dat we dieper gaan nadenken over een probleem
- Verleent de vaardigheid van het redeneren met goede argumenten
- Onderzoekt ideeën en test deze op waarheid

# Kritisch denken: Methode van Socrates

2 belangrijke vragen:

- Wat bedoel je **precies** ermee?
- Welke **bewijzen** heb er hiervoor?

# Kritisch denken in Software Design

## Vraag

Voor wie is de software?

Voor wie is het niet

Waarom maken we deze software?

Wat moet de software doen?

Wat moet ze niet doen?

Bestaat er al software dat dit doet?

Wat is het eenvoudigste ontwerp van de software die de doelstellingen bereikt?

# Kritisch denken in Software Testing

Een tester moet **kritisch denken** en veronderstellen dat er fouten in de software zijn ("bugs") en hij moet ze proberen te vinden

# Logica in de wetenschap

## Hypothetico-Deductieve Methode:

1. **Vorm hypothese**
2. **Maak voorspellingen op basis van deze hypothese**
3. **Test de hypothese**

**-> Confirmatie en Disconfirmatie**

# Confirmatie of disconfirmatie

## Confirmatie

- Als hypethese juist is, dan zou P zich voor moeten doen
- P doet zich voor
- Dus, de hypothese is waarschijnlijk juist

## Disconfirmatie

- Als hypothese juist is, dan zou P zich voor moeten doen
- P doet zich **niet** voor
- Dus, de hypothese is waarschijnlijk niet juist

# Methode van Mill – logisch systeem

Filosoof John Stuart Mill (1806–1873): logisch systeem gepubliceerd

**Oorzaak en effect: bv ziek worden van studenten na lunch**

- **Methode van overeenkomst**

	Fries	Salad	Fish Sticks	Burger	Soup
Student 1	X	X	--	--	X
Student 2	--	X	X	--	--
Student 3	X	X	--	--	X
Student 4	--	X	X	X	--



# Methode van Mill

## Methode van de verschillen

Student	Salad	Burrito	Fried Rice	Burger	Sick after Lunch
Jan	X	X	X	X	X
Pat	X	X	X	--	--

# Methode van Mill

## Methode van residu

= methode van gezond verstand

- A, B, en C geven effecten X, Y and Z.
- A is de oorzaak van X.
- B is de oorzaak van Y.
- Dus, C is waarschijnlijk de oorzaak van Z

# *Online cursus logica*

**[HTTPS://COURSES.EDX.ORG/COURSES/COURSE-  
V1:MICROSOFT+DEV262X+1T2018A/COURSE/](https://courses.edx.org/courses/course-v1:MICROSOFT+DEV262X+1T2018A/course/)**