Logica



|  |  |
| --- | --- |
| Auteur | P. Ras |
| Versie | 1 |
| Datum | September 2011 |
| Opleiding | ICTaL, AO en MD |

Inhoud

[1 Inleiding 1](#_Toc320526175)

[1.1 Wat is logica 1](#_Toc320526176)

[1.2 Waarom logica 1](#_Toc320526177)

[1.3 De gekozen onderwerpen 1](#_Toc320526178)

[1.3.1 Talstelsels 1](#_Toc320526179)

[1.3.2 Verzamelingenleer 1](#_Toc320526180)

[1.3.3 Basisalgebra 1](#_Toc320526181)

[1.3.4 Logische puzzels 2](#_Toc320526182)

[1.3.5 Logische testen 2](#_Toc320526183)

[1.3.6 Digitale poorten en poortschakelingen 2](#_Toc320526184)

[1.3.7 Boolean algebra 2](#_Toc320526185)

[1.3.8 Karnaugh diagrammen 2](#_Toc320526186)

[2 Talstelsels 3](#_Toc320526187)

[2.1 Inleiding 3](#_Toc320526188)

[2.2 Decimale talstelsel (deci = 10) of 10-tallig talstelsel 3](#_Toc320526189)

[2.3 Andere talstelsels omrekenen naar decimaal 4](#_Toc320526190)

[2.3.1 Werkwijze 2- t/m 10-tallig talstelsel 4](#_Toc320526191)

[2.3.2 Oefenopgaven 4](#_Toc320526192)

[2.3.3 Werkwijze 11-tallig talstelsel en hoger 5](#_Toc320526193)

[2.3.4 Oefenopgaven 5](#_Toc320526194)

[2.4 Omrekenen van 10-tallig naar een ander talstelsel 6](#_Toc320526195)

[2.4.1 Werkwijze 6](#_Toc320526196)

[2.4.2 Oefenopgaven 7](#_Toc320526197)

[2.5 Omrekenen van 2-tallig naar 16-tallig en andersom 7](#_Toc320526198)

[2.5.1 werkwijze 7](#_Toc320526199)

[2.5.2 Oefenopgaven 8](#_Toc320526200)

[3 Verzamelingenleer 9](#_Toc320526201)

[3.1 Inleiding 9](#_Toc320526202)

[3.2 Voorbeelden van getalverzamelingen 10](#_Toc320526203)

[3.2.1 De Natuurlijke getallen 10](#_Toc320526204)

[3.2.2 De gehele getallen 10](#_Toc320526205)

[3.2.3 De rationale getallen 11](#_Toc320526206)

[3.2.4 De reële getallen 11](#_Toc320526207)

[3.2.5 De complexe getallen 11](#_Toc320526208)

[3.2.6 De lege verzameling 11](#_Toc320526209)

[3.2.7 Het universum 11](#_Toc320526210)

[3.3 Venn-diagrammen 12](#_Toc320526211)

[3.4 Bewerings en bewerkings operatoren 12](#_Toc320526212)

[3.4.1 Bewerings operatoren 12](#_Toc320526213)

[3.4.1.1 “Is element van” en “is geen element van” 12](#_Toc320526214)

[3.4.1.2 “Is een deelverzameling van” en “is geen deelverzameling van” 13](#_Toc320526215)

[3.4.1.3 “… Omvat …” en “… Omvat … niet” 13](#_Toc320526216)

[3.4.2 Bewerkings operatoren 14](#_Toc320526217)

[3.4.2.1 De doorsnede 14](#_Toc320526218)

[3.4.2.2 De vereniging 14](#_Toc320526219)

[3.4.2.3 Uitgezonderd 15](#_Toc320526220)

[3.4.3 Oefenopgaven 15](#_Toc320526221)

[3.5 Complement 16](#_Toc320526222)

[3.6 Oefenopgaven 16](#_Toc320526223)

[3.7 Beweringen en predikaten 17](#_Toc320526224)

[3.7.1 Beweringen 17](#_Toc320526225)

[3.7.2 Voorbeelden 17](#_Toc320526226)

[3.7.3 Predikaten 17](#_Toc320526227)

[3.7.4 Voorbeelden 17](#_Toc320526228)

[3.7.5 Verzamelingen opschrijven met predikaten 18](#_Toc320526229)

[3.8 Oefenopgaven 18](#_Toc320526230)

[3.9 EN (AND) en OF (OR) 19](#_Toc320526231)

[3.10 Oefenopgaven 19](#_Toc320526232)

[3.11 Intervallen en getallenlijnen 20](#_Toc320526233)

[3.11.1 Getallenlijn 20](#_Toc320526234)

[3.11.2 Intervallen 21](#_Toc320526235)

[3.12 Oefenopgaven 22](#_Toc320526236)

[3.13 Samenvatting 23](#_Toc320526237)

[3.13.1 Bijzondere verzamelingen 23](#_Toc320526238)

[3.13.2 Operatoren 23](#_Toc320526239)

[3.13.3 Beweringen en predikaten 23](#_Toc320526240)

[3.13.4 Intervallen 23](#_Toc320526241)

[4 Algebra 24](#_Toc320526242)

[4.1 Inleiding 24](#_Toc320526243)

[4.2 Rekenen met variabelen 24](#_Toc320526244)

[4.3 Haakjes wegwerken 25](#_Toc320526245)

[4.3.1 Inleiding 25](#_Toc320526246)

[4.4 Oefenopgaven 26](#_Toc320526247)

[4.5 Ontbinden in priemfactoren 26](#_Toc320526248)

[4.5.1 Inleiding 26](#_Toc320526249)

[4.5.2 Priemgetallen 26](#_Toc320526250)

[4.5.3 Ontbinden van een getal in priemfactoren 27](#_Toc320526251)

[4.5.4 Wanneer is een getal deelbaar door 2, 3, 5, 7 of 11 28](#_Toc320526252)

[4.6 Oefenopgaven 28](#_Toc320526253)

[4.7 Vereenvoudigen van breuken 29](#_Toc320526254)

[4.8 Oefenopgaven 29](#_Toc320526255)

[4.9 Buiten haakjes halen 30](#_Toc320526256)

[4.10 Oefenopgaven 31](#_Toc320526257)

# Inleiding

## Wat is logica

Een ander woord voor logica is redeneerkunst. In de exacte wetenschappen betekend dit: Uitgaand van axioma’s (aannames) redeneren tot een uitkomst. Deze axioma’s kunnen ontstaan door ervaringen (als je iets boven de grond loslaat, dan valt het naar beneden) of ontstaan uit het brein van zeer slimme mensen. De ervaringsaxioma’s kom je veel tegen bij natuurkunde en scheikunde, de andere kom je vaak tegen bij de wiskunde.

De aannames of axioma’s hoeven niet waar te zijn. Om te kunnen redeneren heb je ze alleen nodig. Je moet namelijk een startpunt hebben.

## Waarom logica

De onderwerpen die je bij logica leert zijn meestal niet belangrijk. Het gaat meer om de manier van denken die je nodig hebt om uitkomsten bij deze onderwerpen te kunnen beredeneren. Als je een beroep krijgt waarbij je moet kunnen programmeren is het belangrijk dat je logisch kunt denken, patronen herkend en analytisch/systematisch je programma’s kunt opbouwen.

De cursus Logica werkt toe naar Boolean algebra. Dit is algebra met variabelen die slechts twee waarden kunnen aannemen. 0 en 1.

Binnen de Boolean algebra werken we met twee operatoren. Een operator is een teken dat een bewerking van getallen en variabelen aangeeft. Al bekende operatoren zijn optellen, aftrekken, vermenigvuldigen, delen, machtsverheffen, worteltrekken, enz.

De operatoren bij Boolean algebra zijn **en (∧)** en **of (∨)**. Ook bij het programmeren kom je deze operatoren veelvuldig tegen (OR, AND).

## De gekozen onderwerpen

### Talstelsels

Dit is een onderwerp waarbij je een paar basisregels leert aan de hand van ons eigen decimale talstelsel. Met deze basisregels kun je vervolgens alle talstelsels (binair, hexadecimaal, 7-tallig, enz.) opbouwen en snappen.

### Verzamelingenleer

Met de kennis van verzamelingenleer krijg je een beter inzicht in de werking van databases. Net als bij databases gaat verzamelingenleer over het groeperen en combineren van data (gegevens).

### Basisalgebra

Hier behandelen wij onder andere ontbinden in factoren, haakjes wegwerken en oplossen van eenvoudige vergelijkingen. Deze vaardigheden heb je weer nodig bij Boolean algebra.

### Logische puzzels

Tussendoor gaan we aan de slag met puzzels die een logische denkwijze vereisen. Bijvoorbeeld Sudoku’s en Logigrammen.

### Logische testen

Op gezette tijden doen we ook een aantal logische testen. IQ-testen en Braintraining achtige testen. Zo kunnen we zien of er progressie (vooruitgang) zit in je logisch denkniveau.

### Digitale poorten en poortschakelingen

Digitale poortschakelingen helpen je met het binaire (bi = 2) denken. Een computer denkt ook alleen maar in enen en nullen. Door middel van digitale poorten krijg je inzicht in de werking van de verschillende operatoren. Wat doet een operator met signalen van enen en nullen. Poortschakelingen maken Boolean functies tastbaar.

### Boolean algebra

Het doel van Boolean algebra is om complexe voorwaarden in een programma te vereenvoudigen. Soms vallen voorwaarden helemaal weg. Door dit algebraïsch (systematisch) te benaderen kun je vaak programma’s vereenvoudigen, iets wat de elegantie, het geheugengebruik en de snelheid ten goede komt.

### Karnaugh diagrammen

Met Karnaugh diagrammen kun je hetzelfde als met Boolean algebra, namelijk vereenvoudigen. Het verschil zit in de methode. Met Karnaugh diagrammen doe je dit op een grafische manier. Omdat het grootste deel van de mensheid grafischer dan analytischer is ingesteld wordt deze methode meestal makkelijker gevonden.

# Talstelsels

## Inleiding

Een talstelsel is een methode om met zo min mogelijk symbolen een bepaalde waarde aan te kunnen geven. Je maakt dan getallen. Verschillende toepassingen maken van verschillende talstelsels gebruik. Zo gebruik je voor IP4 netwerkadressen het decimale talstelsel, maar rekent de computer dit om naar een binair (2) adres. 10.10.255.3 wordt 00001010.00001010.11111111.00000011. Bij IP6 en bij sommige oude machinetaal programmeertalen maak je gebruik van het hexadecimale (16) talstelsel. De computer vertaald ook deze getallen naar een binair getal.

In dit hoofdstuk bekijken we ons eigen decimale talstelsel (wij werken decimaal omdat we 10 vingers hebben) en gebruiken we de regels die hierbij horen om andere talstelsels te begrijpen. Het einddoel is dat je van ieder willekeurig talstelsel kunt omrekenen naar ieder ander willekeurig talstelsel.

## Decimale talstelsel (deci = 10) of 10-tallig talstelsel

Als we naar een willekeurig geheel getal kijken dan is dit opgebouwd uit 10 symbolen. namelijk 0, 1, 2, 3, 4, 5, 6, 7, 8 en 9. De plaats waar dit symbool in een getal staat bepaald de waarde van het symbool. Het laatste symbool staat voor het aantal ééntallen, het symbool hiervoor staat voor het aantal tientallen, het symbool daarvoor staat voor het aantal honderdtallen, het symbool daarvoor staat voor het aantal duizendtallen, enzovoorts.

Het getal 3781 is de som (optelling) van 1 ééntal, 8 tientallen, 7 honderdtallen en 3 duizendtallen. Je doet dus 1x1 + 8x10 + 7x100 + 3x1000 = 3781.

Je maakt hier gebruik van de machtreeks van 10 (zie de onderstaande tabel).

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Machtreeks | 100 | 101 | 102 | 103 | 104 | 105 | Enz. |
| Waarde | 1 | 10 | 100 | 1000 | 10000 | 100000 |  |

De symbolen 0 t/m 9 gebruik je als vermenigvuldigingsfactoren voor de machten van 10.

Het grondtal van de machten is 10, en daarom heet dit talstelsel het 10-tallig of decimale talstelsel.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 3 | 7 | 8 | 1 |  |  |  |  |
|  |  |  |  | 1•100 = | 1•1 = | 1 |  |
|  |  |  |  | 8•101 = | 8•10 = | 80 |  |
|  |  |  |  | 7•102 = | 7•100 = | 700 |  |
|  |  |  |  | 3•103 = | 3•1000 = | 3000 |  |
|  |  |  |  | |  | | --- | |  | |  |  | + |
|  |  |  |  |  |  | 3781 |  |

## Andere talstelsels omrekenen naar decimaal

### Werkwijze 2- t/m 10-tallig talstelsel

Alle talstelsels werken op dezelfde manier. Het enige wat anders is, is het grondtal. Zo heeft het binaire talstelsel het grondtal 2, het hexadecimale talstelsel het grondtal 16, enz.

Het aantal symbolen dat een talstelsel gebruikt is ook gelijk aan het grondtal. Het 4-tallig talstelsel gebruikt 0, 1, 2, 3 (4 symbolen) en het 2-tallig talstelsel gebruikt 0 en 1 (2 symbolen).

Bekijk het volgende voorbeeld: 3425 (6) = ? (10)

Er wordt nu gevraagd een omrekening te maken van het 6-tallig naar het 10-tallig talstelsel. Dus het getal 3425 in het 6-tallig talstelsel heeft een waarde van ? in het 10-tallig talstelsel.

Je gaat nu als volgt te werk:

Werk van achteren uit, dit symbool (5) is de vermenigvuldigingsfactor van 60, dan is 2 de vermenigvuldigingsfactor van 61, 4 de vermenigvuldigingsfactor van 62 en 3 de vermenigvuldigingsfactor van 63. Je voert de vermenigvuldigingen uit en telt de gekregen getallen bij elkaar op. In een schema ziet dit er als volgt uit:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 3 | 4 | 2 | 5 |  |  |  |  |
|  |  |  |  | 5x60 = | 5x1 = | 5 |  |
|  |  |  |  | 2x61 = | 2x6 = | 12 |  |
|  |  |  |  | 4x62 = | 4x36 = | 144 |  |
|  |  |  |  | 3x63 = | 3x216 = | 648 |  |
|  |  |  |  | |  | | --- | |  | |  |  | + |
|  |  |  |  |  |  | 809 |  |

### Oefenopgaven

45371

1x9^0 1

7x9^1 63

3x9^2 81

5x9^3 243

4x9^3 2916

Reken de volgende getallen om naar het decimale talstelsel:

2735 (8) = 1245 (10) 45371 (9) = 3304 (10)

12323 (4) = 443 (10) 4235 (6) = (10)

10100111 (2) = 167 (10) 34243 (5) = (10)

10201021 (3) = 2707 (10) 6204 (7) = (10)

67125 (8) = 28245 (10) 51543 (6) = (10)

### Werkwijze 11-tallig talstelsel en hoger

Bij talstelsels hoger dan 10-tallig krijgen wij een probleem. Je hebt dan meer dan 10 symbolen nodig (11-tallig = 11 symbolen, 16-tallig = 16 symbolen, enz.). In ons talstelsel kennen wij de symbolen van 0 t/m 9. Voor bijvoorbeeld het 16-tallig talstelsel hebben wij nog symbolen nodig voor 10, 11, 12, 13, 14 en 15. Hiervoor gebruiken wij de hoofdletters van A t/m F.

Je krijgt dan:

|  |  |  |  |
| --- | --- | --- | --- |
| Symbool | Waarde | Symbool | Waarde |
| 0 | 0 | 8 | 8 |
| 1 | 1 | 9 | 9 |
| 2 | 2 | A | 10 |
| 3 | 3 | B | 11 |
| 4 | 4 | C | 12 |
| 5 | 5 | D | 13 |
| 6 | 6 | E | 14 |
| 7 | 7 | F | 15 |

De werkwijze om een getal om te rekenen blijft onveranderd.

Bekijk het volgende voorbeeld: A6E (16) = ? (10)

Er wordt nu gevraagd een omrekening te maken van het 16-tallig naar het 10-tallig talstelsel. Dus het getal A6E in het 16-tallig talstelsel heeft een waarde van ? in het 10-tallig talstelsel.

Je gaat nu als volgt te werk:

Werk van achteren uit, dit symbool E (=14) is de vermenigvuldigingsfactor van 160, dan is 6 de vermenigvuldigingsfactor van 161 en A (=10) de vermenigvuldigingsfactor van 162. Je voert de vermenigvuldigingen uit en telt de gekregen getallen bij elkaar op. In een schema ziet dit er als volgt uit:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| A | 6 | E |  |  |  |  |
|  |  |  | 14.160 = | 14.1 = | 14 |  |
|  |  |  | 6.161 = | 6.16 = | 96 |  |
|  |  |  | 10.162 = | 10.256 = | 2560 |  |
|  |  |  | |  | | --- | |  | |  |  | + |
|  |  |  |  |  | 2670 |  |

### Oefenopgaven

Reken de volgende getallen om naar het decimale talstelsel:

AAA (16) = ………(10) 3B7C (16) = ………(10)

666 (16) = ………(10) 22FA (16) = ………(10)

## Omrekenen van 10-tallig naar een ander talstelsel

### Werkwijze

We doen dit aan de hand van een stappenplan met hetzelfde voorbeeld als in de vorige paragraaf: 809 (10) = ? (6)

**Stap 1**

Schrijf de machtreeks op van het talstelsel waar je naartoe gaat.

In dit voorbeeld de machtreeks van 6.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Machtreeks | 60 | 61 | 62 | 63 | 64 | 65 |
| Waarden | 1 | 6 | 36 | 216 | 1296 | 7776 |

**Stap 2**

Bepaal tot waar je de machtreeks nodig hebt.

In dit voorbeeld heb je geen factoren nodig boven de 216. De volgende is namelijk 1296 en dit past niet meer in 809 (de opgave).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Machtreeks | 60 | 61 | 62 | 63 |
| Waarden | 1 | 6 | 36 | 216 |

**Stap 3**

Bepaal uit hoeveel symbolen je getal gaat bestaan.

In ons voorbeeld maak je gebruik van 60, 61, 62 en 63. Je te vormen getal bestaat dan uit 4 symbolen.

**Stap 4**

Bepaal de vermenigvuldigingsfactoren van de machtreeks. Begin bij de grootste.

In ons voorbeeld is de grootste waarde 63 = 216. Deze past **3** keer in 809. 3 keer 216 = 648. Deze waarde trek je van 809 af en dan hou je 161 over.

Vervolgens kijk je hoe vaak 36 in 161 past. Dit is **4** keer. 4 keer 36 = 144. Deze waarde trek je van 161 af en dan hou je 17 over.

Vervolgens kijk je hoe vaak 6 in 17 past. Dit is **2** keer. 2 keer 6 = 12. Deze waarde trek je van 17 af en dan hou je 5 over.

Vervolgens kijk je hoe vaak 1 in 5 past. Dit is **5** keer. 5 keer 1 = 5. Deze waarde trek je van 5 af en dan hou je 0 over.

**Stap 5**

Schrijf het antwoord op.

Ik weet nu dat 809 = 3.63 + 4.62 + 2.61 + 5.60.

Het antwoord is nu: 809 (10) = 3425 (6)

En dit komt overeen met de vorige paragraaf.

### Oefenopgaven

Reken de decimale waarden om in het gewenste talstelsel. Gebruik de 5 stappen uit de vorige paragraaf.

714 (10) = ………(4) 222 (10) = ………(2)

1458 (10) = ………(5) 716 (10) = ………(2)

721 (10) = ………(8) 8417 (10) = ………(12)

664 (10) = ………(3) 23953 (10) = ………(16)

8314 (10) = ………(6) 1438 (10) = ………(16)

## Omrekenen van 2-tallig naar 16-tallig en andersom

### werkwijze

Sommige omrekeningen zijn sneller en eenvoudiger uit te voeren. Wil je van 5-tallig naar 7-tallig kun je het beste met een tussenstap werken: van 5-tallig naar 10-tallig en daarna van 10-tallig naar 7-tallig.

Het gaat eenvoudiger als de twee machtreeksen in elkaar vallen. Bijvoorbeeld van binair (2) naar hexadecimaal (16) en andersom. 16 is namelijk 24. Schrijven we de machtreeksen onder elkaar dan krijg je:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Machtreeks 2 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 210 | 211 | 212 | enz |
| Waarde | 1 | 2 | 4 | 8 | 16 | 32 | 64 | 128 | 256 | 512 | 1024 | 2048 | 4096 |  |
| Machtreeks 16 | 160 |  |  |  | 161 |  |  |  | 162 |  |  |  | 163 | enz |
| Waarde | 1 |  |  |  | 16 |  |  |  | 256 |  |  |  | 4096 |  |

De machtreeks van 16 valt in zijn geheel in de machtreeks van 2. Je kunt ieder symbool uit het hexadecimale talstelsel (0 t/m F) schrijven als een binair getal bestaande uit 4 symbolen. Bekijk dit in de volgende tabel:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Hexadecimaal |  | Binair |  | Hexadecimaal |  | Binair |
| 0 | = | 0000 |  | 8 | = | 1000 |
| 1 | = | 0001 |  | 9 | = | 1001 |
| 2 | = | 0010 |  | A(10) | = | 1010 |
| 3 | = | 0011 |  | B(11) | = | 1011 |
| 4 | = | 0100 |  | C(12) | = | 1100 |
| 5 | = | 0101 |  | D(13) | = | 1101 |
| 6 | = | 0110 |  | E(14) | = | 1110 |
| 7 | = | 0111 |  | F(15) | = | 1111 |

Omdat in onze talstelsels de waarde van een getal een optelling van machten is, kun je bij deze omrekeningen de symbolen gewoon vervangen.

A3C (16) wordt binair 1010 0011 1100.

63F (16) wordt binair 0110 0011 1111.

11001110010101001 (2) moet je van achter uit opdelen in groepjes van 4. Dit wordt dan

0001 1001 1100 1010 1001 (2) en dit wordt hexadecimaal 19CA9.

1010 1110 0100 (2) wordt hexadecimaal AE4.

### Oefenopgaven

Reken de volgende getallen om naar het gewenste talstelsel.

714 (16) = ………(2) 1001001010001111 (2) = ………(16)

1458 (16) = ………(2) 1111111111111 (2) = ………(16)

721 (16) = ………(2) 101010101010101 (2) = ………(16)

664 (16) = ………(2) 100111 (2) = ………(16)

8314 (16) = ………(2) 100100011111010 (2) = ………(16)

# mieren.jpgVerzamelingenleer

## Inleiding

Een verzameling is een aantal objecten die iets met elkaar gemeen hebben. Deze objecten (dingen) noemen we elementen.

Een aantal voorbeelden:

|  |  |  |
| --- | --- | --- |
| **De verzameling** | **De elementen** | **De gemeenschappelijke factor** |
| Een verzameling postzegels | Postzegels. | Gebruikt of bruikbaar als frankering van post. |
| Een club voetballers | Voetballers. | Zijn allemaal lid van deze club. |
| De verzameling voetbalclubs in de eredivisie | Voetbalclubs. | Uit de eredivisie. |
| Een hoop mieren. | Mieren. | Horen bij dezelfde hoop. |
| Een lijn punten. | Punten. | Liggen allemaal op dezelfde lijn. |
| De Nederlandse bevolking. | Mensen. | Wonen in Nederland. |
| De verzameling van gehele getallen. | Getallen. | Hebben allemaal geen cijfers achter de komma. |



Omdat verzamelingen uit van alles kunnen bestaan, kiezen we er in de wiskunde en logica meestal voor om getal verzamelingen te gebruiken. Kies je bijvoorbeeld voor een hoop mieren, dan zul je de elementen moeten gaan tekenen. Bij getal verzamelingen kun je de elementen gewoon opschrijven.

Als je een aantal elementen als een verzameling wilt opschrijven doen we dit altijd tussen acculades, gescheiden door komma’s. Voorbeeld: V = {3, 5, 17}. Dit is de verzameling V met de elementen 3, 5 en 17.

Een vaste afspraak in de verzamelingenleer is dat er in een verzameling **nooit** twee identieke elementen mogen zitten.

## getalverz.pngVoorbeelden van getalverzamelingen

Een aantal getalverzamelingen zijn al internationaal vastgelegd. Hierbij staan de elementen en de naam van de verzameling al vast.

### De Natuurlijke getallen

De verzameling van de natuurlijke getallen heet N. De natuurlijke getallen zijn alle gehele positieve getallen. In wiskundige notatie geeft dit:

N = {1, 2, 3, 4, ….}

De puntjes geven aan dat de regelmaat van de eerste getallen tot oneindig wordt voortgezet.

Of elementen wel of niet tot een verzameling horen kunnen we in een bewering weergeven:

Het getal 1 **is een element** van N, of in wiskundige notatie: 1 ∈ N.

Het getal -3 **is geen element van** N, of in wiskundige notatie: -3 ∉ N.

Als we twee elementen van N bij elkaar optellen, zit de uitkomst altijd in N.

Als we twee elementen van N van elkaar aftrekken, zit de uitkomst niet altijd in N.

We gaan nu de verzameling van de Natuurlijke getallen uitbreiden.

### De gehele getallen

De verzameling van de gehele getallen heet Z. De gehele getallen zijn alle gehele positieve en negatieve getallen. In wiskundige notatie geeft dit:

Z = {…, -2, -1, 0, 1, 2, …}

In Z kunnen we steeds twee getallen bij elkaar optellen of van elkaar aftrekken. De uitkomst zal altijd weer in Z zitten.

Voorbeelden:

2 + 10 = 12 (2, 10 en 12 ∈ Z)

10 - 13 = -3 (10, 13 en -3 ∈ Z)

-6 - (-3) = -3 (-6, -3 en -3 ∈ Z)

-3 - -7 = -10 (-3, -7 en -10 ∈ Z)

De uitkomsten van delingen binnen Z geven niet altijd uitkomsten binnen Z. Daarom gaan we de verzameling van de gehele getallen nog verder uitbreiden.

### De rationale getallen

De verzameling van de rationale getallen heet Q. De rationale getallen zijn alle getallen die je kunt maken door twee getallen uit Z op elkaar te delen. In wiskundige notatie geeft dit:

Q =

We kunnen nu zeggen dat x ∈ Q als geldt x met p ∈ Z en q ∈ Z en q ≠ 0, dit laatste omdat je niet mag delen door 0.

Toch hebben we nog niet alle getallen. Bijvoorbeeld valt nog niet onder de definitie van Q. Daarom gaan we de verzameling van de rationale getallen nog verder uitbreiden.

### De reële getallen

De verzameling van de reële getallen heet R. De reële getallen zijn alle getallen die je kunt verzinnen en gebruiken in het dagelijkse leven. In wiskundige notatie geeft dit:

R =

Er is nog één uitbreiding op de getallenverzamelingen.

### De complexe getallen

Op de middelbare school heb je geleerd dat je geen wortel kunt trekken uit een negatief getal. Dit kan echter wel. We zeggen nu dat . Je kunt dan ieder getal schrijven als . Deze getallen noemen we complexe getallen.

De extra mogelijkheden die het rekenen met complexe getallen biedt, hebben geleid tot allerlei nuttige toepassingen in vooral alles wat met trillingen en golven te maken heeft, zoals het grootste deel van de natuurkunde, de elektrotechniek, de meet- en regeltechniek en vele andere technische disciplines.

De verzameling van de complexe getallen noemen we C.

We zullen niet verder ingaan op de complexe getallen omdat dit een heel nieuw stukje wiskunde is. We volstaan met de wetenschap dat deze getallen en hierdoor ook de verzameling C bestaan.

### De lege verzameling

Een getalverzameling zonder elementen noemen we **de lege verzameling**. Ook hier is een symbool voor:

Ø = de lege verzameling = { }.

### Het universum

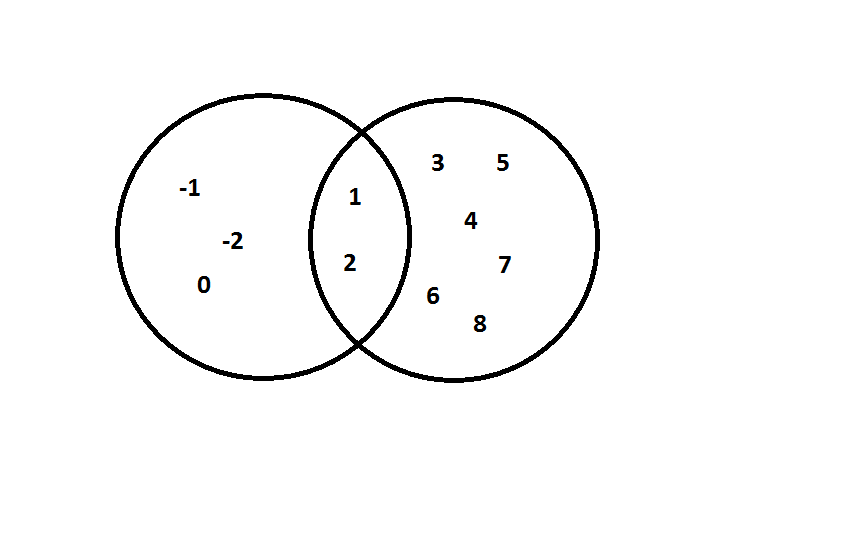
Het universum U is de verzameling met alle elementen die bij een bepaald onderwerp horen.

Als we zeggen U = {1, 2, 3, 4}, dan bestaan 5, 6, enz. niet.

## Venn-diagrammen

Een Venn-diagram is een grafische manier om verzamelingen op te schrijven. Neem bijvoorbeeld de volgende twee verzamelingen:

V = {-2, -1, 0, 1, 2} en W = {1, 2, 3, 4, 5, 6, 7, 8}



**V W**

De linker cirkel is V, de rechter cirkel is W. Het overlappende deel bevat de elementen die zowel in V als in W zitten.

## Bewerings en bewerkings operatoren

Een wiskundige of logische bewerking noemen we een operatie. Het symbool dat de bewerking aangeeft noemen we een operator.

We kennen al behoorlijk veel operatoren, bijvoorbeeld:

+ ,­, ×, ÷, 2, √, enz.

Deze operatoren hebben één getal of twee getallen als invoer, en één getal als uitvoer. Deze operatoren noemen we **bewerkingsoperatoren**.

Er bestaan ook logische of **beweringsoperatoren**. Deze hebben als uitvoer **TRUE** of **FALSE** (of **waar** en **niet waar**, of **hoog** en **laag**, of **0** en **1**)

In de verzamelingenleer heb je ook operatoren. Deze operatoren hebben verzamelingen als invoer.

### Bewerings operatoren

Deze operatoren hebben als uitkomst **waar** of **niet waar**.

#### “Is element van” en “is geen element van”

Deze hebben we al eerder gezien:

3 ∈ N (3 is een element van N) uitvoer: Waar

-2 ∈ N (-2 is een element van N) uitvoer: Niet Waar

3 ∉ N (3 is geen element van N) uitvoer: Niet Waar

-2 ∉ N (-2 is geen element van N) uitvoer: Waar

#### “Is een deelverzameling van” en “is geen deelverzameling van”

We zeggen: Een verzameling A is een deelverzameling van verzameling B (A ⊂ B) als alle elementen van verzameling A in verzameling B zitten.

Voorbeelden:

Neem drie verzamelingen:

Dan:

V ⊂ W **TRUE** Want alle elementen van V zitten in W.

V ⊂ X **FALSE** Want {1} zit niet in X.

W ⊂ V **FALSE** Want {0}, {5}, {6}, {7} en {8} zitten niet in V.

W ⊂ X **FALSE** Want {0}, {1} en {8} zitten niet in X.

X ⊂ V **FALSE** Want {5}, {6} en {7} zitten niet in V.

X ⊂ W **TRUE** Want alle elementen van X zitten in W.

Andersom geldt voor ⊄ (is geen deelverzameling van)

V ⊄ W **FALSE**

V ⊄ X **TRUE**

W ⊄ V **TRUE**

W ⊄ X **TRUE**

X ⊄ V **TRUE**

X ⊄ W **FALSE**

#### “… Omvat …” en “… Omvat … niet”

We zeggen: Een verzameling A omvat verzameling B (A ⊃ B) als alle elementen van verzameling B in verzameling A zitten.

Dit is eigenlijk het omgekeerde van een deelverzameling.

Voorbeelden:

Neem weer de drie verzamelingen:

Dan:

V ⊃ W **FALSE**

V ⊃ X **FALSE**

W ⊃ V **TRUE** Want alle elementen van V zitten in W.

W ⊃ X **TRUE** Want alle elementen van X zitten in W.

X ⊃ V **FALSE**

X ⊃ W **FALSE**

En omgekeerd geldt:

V omvat W niet **TRUE**

V omvat X niet **TRUE**

W omvat V niet **FALSE**

W omvat X niet **FALSE**

X omvat V niet **TRUE**

X omvat W niet **TRUE**

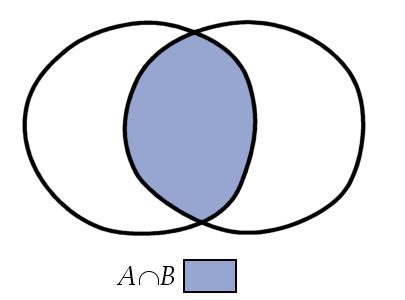
### Bewerkings operatoren

We behandelen een drietal bewerkingsoperatoren.

#### De doorsnede

We zeggen: De doorsnede van twee verzamelingen A en B (A ∩ B, spreek uit als “A doorsnede B”) is de (nieuwe) verzameling met alle elementen die in A zitten en in B zitten.

Voorbeelden:

Neem de verzamelingen V = {1, 2, 3, 4, 5}, W = {4, 5, 6, 7, 8} en X = {2, 4, 6, 8, 10}.

De verzameling P = V ∩ W = {4, 5}.

De verzameling Q = V ∩ X = {2, 4}.

De verzameling R = X ∩ W = {4, 6, 8}.

#### De vereniging

We zeggen: De vereniging van twee verzamelingen A en B (A ∪ B, spreek uit als “A verenigd B”) is de (nieuwe) verzameling met alle elementen van A en B samen.

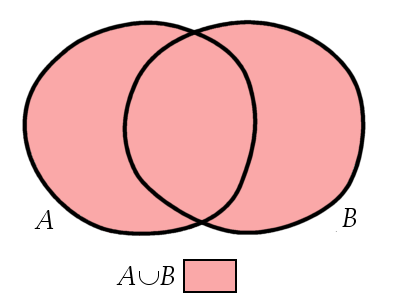
Voorbeelden:

Neem weer de verzamelingen V = {1, 2, 3, 4, 5}, W = {4, 5, 6, 7, 8} en X = {2, 4, 6, 8, 10}.

De verzameling P = V ∪ W = {1, 2, 3, 4, 5, 6, 7, 8}.

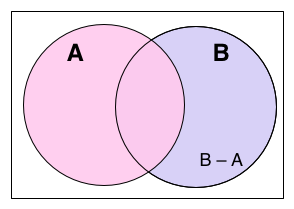
De verzameling Q = V ∪ X = {1, 2, 3, 4, 5, 6, 8, 10}.

De verzameling R = X ∪ W = {2, 4, 5, 6, 7, 8, 10}.

Let er op dat er nooit twee dezelfde elementen in een verzameling staan.

#### Uitgezonderd

We zeggen: De verzameling A uitgezonderd de verzameling B, V = A \ B (spreek uit als “A uitgezonderd B”, soms staat er A - B) is de (nieuwe) verzameling met alle elementen van A verminderd met de elementen van B. V noemen we de verschilverzameling.

Voorbeelden:

Neem de verzameling V = {1, 2, 3, 4, 5}.

De verzameling P = V \ {1, 2} = {3, 4, 5,}.

De verzameling Q = V \ {2, 4, 6, 8, 10} = {1, 3, 5}.

De verzameling R = V \ V = ∅.

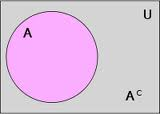
### Oefenopgaven

1. Zijn de volgende beweringen True of False (Waar of Niet waar) en toon dit aan met een Venn-diagram:

|  |  |
| --- | --- |
| * 1. N ⊂ Q | * 1. Z ⊂ Q |
| * 1. N ⊂ R | * 1. N ⊂ Z ⊂ Q ⊂ R ⊂ C |
| * 1. Z ⊂ N | * 1. N ⊂ N |

1. V = {1, 2, 3, 4} en W = {-1, 0, 1, 2, 3, 4, 5, 6}
   1. Is V ⊂ W?
   2. Is W ⊂ V?
2. V = {1, 2, 3} en W = {1, 3, 2}
   1. Is V ⊂ W?
   2. Is W ⊂ V?
3. Bepaal de verzameling:
   1. N ∩ Z
   2. Z ∩ R
   3. R ∩ Z
4. V = {0, 2, 4, 6, 8, 10, 12} en W = {0, 3, 5, 8, 11}
   1. Bepaal V ∩ W.
5. V = {0, 1, 2, 3, 4, …} en W = {-1, -2, -3}
   1. Bepaal V ∩ W.

## Complement

We zeggen: VC is het complement van V als geldt dat V ∩ VC = ∅ **en** V ∪ VC = U (Universum).

Er bestaat alleen een complement als je weet wat U is (als U gedefinieerd is).

## Oefenopgaven

1. Schrijf alle verzamelingen V op die voldoen aan:
   1. V ⊂ {0, 1, 2}
2. V = {a, b, c, d, e}, W = {a, c, g} en U = {a, b, c, d, e, f, g, h, i}

Bepaal de volgende verzamelingen:

|  |  |
| --- | --- |
| 1. V ∩ W | 1. V ∪ W |
| 1. VC | 1. WC |
| 1. VC ∩ WC | 1. (V ∩ W) ∪ VC |
| 1. V ∩ (W ∪ VC) | 1. WC ∩ W |
| 1. WC ∩ V | 1. WC ∪ V |

1. Laat met behulp van een Venn-diagram zien, dat:
   1. (A ∩ B) ∩ C = A ∩ (B ∩ C)
   2. A ∩ (B ∪ C) = (A ∩ B) ∪ (A ∩ C)
   3. Als A ⊂ B, dan A ∪ B = B en A ∩ B = A

## Beweringen en predikaten

### Beweringen

Een bewering is een opmerking die WAAR of NIET WAAR kan zijn. We gebruiken internationaal de termen TRUE (WAAR) en FALSE (NIET WAAR).

We hebben al een aantal beweringsoperatoren behandeld. Zo kennen we ∈ en ∉, ⊂ en ⊄ en ⊃ en . We kennen nog een groep beweringsoperatoren, de vergelijkende beweringsoperatoren. Hieronder zie je een lijstje met deze operatoren en de al bekende:

|  |  |
| --- | --- |
| **Vergelijkende beweringsoperatoren** | |
| < | Kleiner dan |
| > | Groter dan |
| ≤ | Kleiner of gelijk aan |
| ≥ | Groter of gelijk aan |
| = | Is gelijk aan |
| ≠ | Is niet gelijk aan |
| **Overige beweringsoperatoren** | |
| ∈ | Is element van |
| ∉ | Is geen element van |
| ⊂ | Is deelverzameling van |
| ⊄ | Is geen deelverzameling van |
| ⊃ | Omvat |
|  | Omvat niet |

### Voorbeelden

2 < 4 is een bewering, een bewering die TRUE is.

-6 ≥ 8 is een bewering, een bewering die FALSE is.

3,27 ∈ Q is een bewering, een bewering die TRUE is.

### Predikaten

Een predikaat P(x), is een bewering waarin een onbekende voorkomt. Als we in het predikaat een element voor x invullen, wordt het predikaat een bewering.

Predikaten worden vaak gebruikt om gebieden van getallen aan te geven.

### Voorbeelden

Neem een verzameling V = {1, 2, 3, 4}.

x ∈ V is een predikaat.

Vullen we voor x het getal 1 in, dan ontstaat de bewering 1 ∈ V, die TRUE is.

Vullen we voor x het getal 8 in, dan ontstaat de bewering 8 ∈ V, die FALSE is.

### Verzamelingen opschrijven met predikaten

Stel ik wil een verzameling V opschrijven van alle getallen in de verzameling R, die tussen 3 en 5 liggen.

Dit kunnen we niet doen met de manieren die we tot nog toe hebben behandeld. Het zijn ten slotte een oneindig aantal elementen (3,00000001 en 3,0000000000002 enz.).

We schrijven dit nu als volgt op:

**V = {x ∈ R | 3 < x < 5} = Predikatennotatie**

We ontleden even wat hier staat:

V De naam van de verzameling.

x ∈ R Het bereik van de verzameling (dit is het universum U).

| Dit teken betekend: waarvoor geldt dat.

3 < x < 5 3 moet kleiner zijn dan x **en** x moet kleiner zijn dan 5. Oftewel x ligt tussen 3 en 5.

We spreken de regel als volgt uit:

De verzameling V met elementen x uit R, waarvoor geldt dat x tussen 3 en 5 ligt.

## Oefenopgaven

1. We beschouwen het predikaat P(x) = x ∈ N. Bepaal voor de volgende waarden van x of dit TRUE of FALSE is.
   1. x = 3
   2. x = 17,5
   3. x = -16
   4. x = 23
   5. x = √12
   6. x = 0
   7. x = √16
2. Schrijf de volgende verzamelingen in de predikatennotatie:
   1. De verzameling V met elementen x uit R, waarvoor geldt dat x tussen -3 en 3 ligt.
   2. De verzameling W met elementen x uit R, waarvoor geldt dat x groter of gelijk is aan 1.
   3. De verzameling X met elementen x uit R, waarvoor geldt dat x kleiner dan 4 is.
   4. De verzameling Y met elementen x uit N, waarvoor geldt dat x tussen -3 en 5 ligt.
   5. Schrijf de verzameling uit opgave d. op in de accolade notatie (losse elementen).
   6. De verzameling Z met elementen x uit R, waarvoor geldt dat x groter of gelijk is aan -1 en kleiner of gelijk is aan 3.

## EN (AND) en OF (OR)

De logische operatoren **EN** en **OF** zijn bij het programmeren belangrijke logische operatoren. Zo belangrijk dat er een complete algebra voor is ontworpen. We noemen dit de Boolean algebra of binaire algebra.

Als er tussen twee beweringen een **EN-operator** staat, dan krijg je alleen TRUE als beide beweringen TRUE zijn.

Als er tussen twee beweringen een **OF-operator** staat, dan krijg je TRUE als één van beide of beide beweringen TRUE zijn.

Het teken voor EN is:

Het teken voor OF is:

In een zogenaamde waarheidstabel geeft dit:

|  |  |
| --- | --- |
| De **EN**-functie (**AND**) | |
|  | 0 |
|  | 0 |
|  | 0 |
|  | 1 |

|  |  |
| --- | --- |
| De **OF**-functie (**OR**) | |
|  | 0 |
|  | 1 |
|  | 1 |
|  | 1 |

Er bestaat een verband tussen de bewerkingen en en tussen de bewerkingen en .

worden gebruikt tussen predikaten.

worden gebruikt tussen verzamelingen.

Als je zoekt naar dan is dit

Als je zoekt naar dan is dit

## Oefenopgaven

Schrijf de opgaven hieronder op als hierboven en teken een Venn-diagram.

1. A={1, 2, 3, 4, 5} en B={0, 4, 8, 16, 32}.
   1. Bepaal
   2. Bepaal

## Intervallen en getallenlijnen

### Getallenlijn

Getalverzamelingen kun je behalve met Venn-diagrammen ook grafisch weergeven via een getallenlijn.

Voorbeeld 1:

V = {x ∈ R | 3 < x < 5}

Getallenlijn1.wpg

Voorbeeld 2:

W = {1, 2, 3, 4}

Getallenlijn3.wpg

Voorbeeld 3:

X = {x∈ R | x ≤ 0}

Afbeelding2.wpg

Zoals je hierboven kunt zien geef je een gebied aan met een streep. De grenzen of losse waarden geef je aan met een stip.

Een **gesloten stip** betekend dat de betreffende waarde **wel** meedoet in de verzameling.

Een **open stip** betekend dat de betreffende waarde **niet** meedoet in de verzameling.

### Intervallen

Als het Universum van een getallenverzameling gelijk is aan R, kun je verzamelingen ook als een interval opschrijven. Bij een interval schrijf je de randen van een getalgebied op. Alle waarden tussen deze randen horen bij de verzameling. Rondom de verzameling schrijf je haken die aangeven of de randen wel of niet bij de verzameling horen.

Een **rechte haak [ of ]** betekend dat de betreffende waarde **wel** meedoet in de verzameling.

Een **puntige haak 〈 of 〉** betekend dat de betreffende waarde **niet** meedoet in de verzameling.

Voorbeeld 1:

Getallenlijn1.wpgV = {x ∈ R | 3 < x < 5}

De getallenlijn:

De bijhorende intervalnotatie: **〈3,5〉**

Voorbeeld 2:

Getallenlijn3.wpgW = {1, 2, 3, 4}

De getallenlijn:

De bijhorende intervalnotatie: Geen interval wat U ≠ R

Voorbeeld 3:

Afbeelding2.wpgX = {x∈ R | x ≤ 0}

De getallenlijn:

De bijhorende intervalnotatie: **〈←,0]**

**〈 2,5 〉** noemen we een open interval.

**[ 2,5 ]** noemen we een gesloten interval.

**[ 2,5 〉** noemen we een links gesloten interval.

## Oefenopgaven

1. Geef de volgende intervallen weer met behulp van een getallenlijn en met de predikatennotatie:
2. [ 3, 4 ]
3. [ 3, 4 〉
4. 〈 1, 6 〉
5. [ -2, 2 〉
6. 〈 ←, 2 ]
7. 〈 6, → 〉
8. [ -2, → 〉
9. 〈 ←, → 〉
10. Geef de volgende verzamelingen weer met behulp van een getallenlijn en met de intervalnotatie:
11. { x∈R | 1 < x < 7}
12. { x∈R | 4 ≤ x < 8}
13. { x∈R | x < 2}
14. { x∈R | x ≥ -3}
15. { x∈R | x > 2 ∨ 0 ≤ x < 1}
16. { x∈R | -5 ≤ x ≤ -3 ∨ x > 6}
17. Gegeven de volgende getallenlijn:



Welke antwoorden zijn correct en welke niet? Motiveer je antwoorden!

1. { x∈R | -1 < x ≤ 0}
2. { x∈Z | 0 ≤ x < -1}
3. { x∈R | 0 ≤ x < -1}
4. { x∈R | -1 ≤ x < 0}
5. { x∈R | x ≤ 0 ∧ x > -1}
6. { x∈R | x ≤ 0 ∨ x > -1}
7. { x∈R | x ≤ 0 ∩ x > -1}
8. { x∈R | x ≤ 0 ∪ x > -1}
9. 〈 -1, 0]
10. [ -1, 0〉

## Samenvatting

### Bijzondere verzamelingen

De verzameling van de natuurlijke getallen N.

De verzameling van de gehele getallen Z.

De verzameling van de rationale getallen Q.

De verzameling van de reële getallen R.

De verzameling van de complexe getallen C.

### Operatoren

Verzamelingen zijn altijd een deel van U (Universum).

Deelverzameling: V is een deelverzameling van W (**V⊂W**) als voor alle v∈V geldt dat v∈W.

Omvat: V omvat W (**V⊃W**) als voor alle w∈W geldt dat w∈V.

Doorsnede: V∩W = {x∈U | x∈V ∧ x∈W}.

Vereniging: V∪W = {x∈U | x∈V ∨ x∈W}.

Complement: VC (ten opzichte van U) = {x∈U | x∉V}

### Beweringen en predikaten

Bewering: is een uitspraak over een bekend element (getal); men kan zeggen of de uitspraak waar (TRUE) of niet waar (FALSE) is.

Predikaat: is een uitspraak over een onbekend element (letter, variabele); men kan **niet** zeggen of de uitspraak waar (TRUE) of niet waar (FALSE) is.

De symbolen ∩ en ∪ worden gebruikt bij verzamelingen.

De symbolen ∧ (en) en ∨ (of) worden gebruikt bij predikaten.

### Intervallen

Een interval is een verzameling getallen in R, waarbij de grenzen aangegeven worden met rechte haken ( [ ], de grenzen horen wel bij de verzameling) of puntige haken ( 〈 〉, de grenzen horen niet bij de verzameling).

# Algebra

## Inleiding

Bij programmeren krijg je heel veel te maken met Booleaanse beweringen. Dat wil zeggen beweringen die TRUE of FALSE kunnen zijn. Bij iedere Booleaanse bewering wordt er gekeken of er wel (TRUE) of niet (FALSE) aan voldaan wordt.

Voorbeeld:

<?php

if ($naam == “Piet”) {

echo “hallo Piet”;

} else {

echo “hallo vreemdeling”;

}

?>

De bewering ($naam == “Piet”) is een Booleaanse bewering. Er wordt hier gekeken of de variabele $naam **wel** “Piet” is (uitvoer “hallo Piet”) of dat hij **niet** “Piet” is (uitvoer “hallo vreemdeling”).

Om hier efficiënt mee om te kunnen gaan krijg je bij Logica in het tweede deel Booleaanse algebra. Als we hier mee aan de slag gaan heb je een aantal algebra basisvaardigheden nodig.

Algebra is het rekenen met variabelen. Basisvaardigheden hierbij zijn:

* Haakjes wegwerken
* Ontbinden in priemfactoren
* Ontbinden in factoren
* Rekenen

Deze onderwerpen komen in dit hoofdstuk aan de orde.

## Rekenen met variabelen

Waarom rekenen met variabelen? Meestal reken je iets uit om een situatie te kunnen voorspellen. Hoeveel massa kan een brug dragen? Hoeveel moet ik minder eten om af te vallen? Hoeveel bier kan ik kopen voor het geld dat ik heb? Hoelang moet ik sparen voordat ik een i-pad 2 kan kopen? Hoeveel personen heb ik nodig om die auto aan te duwen? Enz.

Vaak weet je al wel hoe een berekening moet, maar weet je nog niet met welke getallen de berekening moet worden uitgevoerd. De getallen die je nog niet kent vervang je dan door verschillende letters. Deze letters noemen we variabelen (**variabel** betekend **niet constant**).

Omdat je bij het opstellen van de berekeningen variabelen in vergelijkingen krijgt, kun je nog geen antwoorden berekenen. Je weet immers nog niet welke waarden de variabelen vertegenwoordigen. Toch kun je met behulp van vaste rekenregels een aantal berekeningen van zo’n vergelijking alvast uitvoeren. We noemen dit “de vergelijking vereenvoudigen”. Het rekenen met variabelen noemen we **algebra**.

## Haakjes wegwerken

### Inleiding

Een vergelijking bestaat uit **factoren** (een onderdeel van een vermenigvuldiging) en **termen** (een onderdeel van een optelling). Heb je bijvoorbeeld 6X (6 keer X), dan zijn 6 en X factoren. Heb je bijvoorbeeld X-6 (X min 6), dan zijn X en (-6) termen. Bij 3X2+6X-8 heb je 3 termen, (3X2), (6X) en (-8).

Soms, als je een vergelijking opstelt vanuit een praktijksituatie, krijg je uitdrukkingen met haakjes. Bijvoorbeeld:

Iedere medewerker krijgt bij de Albert Heijn een zaterdagtoeslag van 3 euro per uur. Wat verdient een willekeurige medewerker op een zaterdag?

We hebben hier te maken met 2 variabelen. Het aantal gewerkte uren op zaterdag A en het normale uurloon van een medewerker U. De uitkomst is het verdiende bedrag V. We krijgen dan de volgende vergelijking: V = A.(U+3)

Een uitdrukking met haakjes. Deze bestaat uit twee factoren, A en (U+3). Om de haakjes nu weg te werken moet je alle termen van de eerste factor (dit is alleen A) vermenigvuldigen met alle termen van de tweede factor (dit zijn U en +3). Om dit visueel te maken laat ik het ook zien met pijltjes:

V = A.( U + 3 ) = A.U + 3.A

Tweede voorbeeld:

Werk in de volgende vergelijking de haakjes weg: Y = (X+3).(3X-6)

Y = (X+3).(3X-6) = 3X2 – 6X + 9X – 18 = 3X2 + 3X – 18

We noemen dit ook wel de vogelbekmethode omdat de pijlen hierboven lijken op een vogelbek.

In de laatste stap van het voorbeeld hiervoor worden twee termen (-6X en +3X) samengevoegd. Je mag termen samenvoegen als ze dezelfde macht van X hebben. Dus 3X2 en -5X3 mag je niet samenvoegen, maar 3X2 en -5X2 wel. Dit wordt dan -2X2.

## Oefenopgaven

Werk de haakjes weg en vereenvoudig waar mogelijk:

1

* 1. 3(x+4)
  2. 6(x-5)
  3. 7(x2+x)
  4. 2(x2-3)
  5. 4(x3-2x2)
  6. 8x(x-7)
  7. 2x2(x2+4)
  8. 3x(x2-2x+4)
  9. 6x2(2x2-3x-1)
  10. 7x3(x2-x+5)

2

1. (x+2)(x+3)
2. (x+4)(x+1)
3. (x+2)(x-3)
4. (x-2)(x-3)
5. (x-7)(x+5)
   1. (x2+2)(x+3)
   2. (x2+2)(x-3)
   3. (x3-7)(x+3)
   4. (x+2)(x3+3x2)
   5. (x2+2x)(x3+3x2)

3

1. (x+2)(x2+3x+2)
2. (x-3)(x2-4x+1)
3. (x3+2)(x2+3x-3)
4. (x2-3)(x2-5x-2)
5. (7x3+2x2)(x2+3x+2)
6. (x2+x+3)(x2+3x+2)
7. (x2+x-3)(x2-3x+2)
8. (x2-x-3)(x2-3x-2)
9. (2x2+7x+3)(3x2+5x+2)
10. (x5+x4-3x3)(x2-3x+2)

## Ontbinden in priemfactoren

### Inleiding

Voordat we gaan ontbinden in factoren met variabelen, gaan we eerst getallen ontbinden in priemfactoren. Ontbinden in priemfactoren betekend dat we een getal gaan schrijven als een vermenigvuldiging (factoren) van priemgetallen.

### Priemgetallen

Priemgetallen zijn gehele getallen met een bijzondere eigenschap. Deze getallen zijn alleen deelbaar, zodanig dat de uitkomst weer een geheel getal is, door 1 en door zichzelf en >1. In verzamelingen notatie betekend dit:

a is een priemgetal **en** a ∈ Z **en** (a/b) ∈ Z, dan geldt b=1 of b=a en a>1.

De eerste 30 priemgetallen zijn dan:

2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97, 101, 103, 107, 109, 113.

### Ontbinden van een getal in priemfactoren

Ieder getal kun je nu ontbinden in priemfactoren. Om de factoren te vinden deel je het getal net zo lang door priemgetallen (vanaf de kleinste) tot je 1 als uitkomst krijgt. De delers zijn nu je factoren.

**Voorbeeld:**

Ontbindt het getal 1260 in priemfactoren.

**Oplossing:**

We kijken of 1260 deelbaar is door 2 (het eerste priemgetal).

Ja, en de uitkomst is 630. **2**

Nu kijken we of 630 ook deelbaar is door 2.

Ja, en de uitkomst is 315. **2**

Nu kijken we of 315 ook deelbaar is door 2.

Nee, dan kijken we of 315 deelbaar is door 3 (het volgende priemgetal).

Ja, en de uitkomst is 105. **3**

Nu kijken we of 105 ook deelbaar is door 3.

Ja, en de uitkomst is 35. **3**

Nu kijken we of 35 ook deelbaar is door 3.

Nee, dan kijken we of 35 deelbaar is door 5 (het volgende priemgetal).

Ja, en de uitkomst is 7. **5**

Nu kijken we of 7 ook deelbaar is door 5.

Nee, dan kijken we of 35 deelbaar is door 7 (het volgende priemgetal).

Ja, en de uitkomst is 1. **7**

We kunnen nu opschrijven:

**1260 = 2.2.3.3.5.7** (controleer dit met je rekenmachine).

Het steeds opnieuw doen van dezelfde stap heeft bij programmeren een eigen naam. We noemen dit **itereren** en één zo’n stap noemen we een **iteratie**.

**Voorbeeld 2:**

Ontbindt het getal 3822 in priemfactoren.

**Oplossing:**

Is 3822 deelbaar door 2? Ja 2

Uitkomst: 1911 deelbaar door 2? Nee, door 3? Ja 3

Uitkomst: 637 deelbaar door 3? Nee, door 5? Nee, door 7?Ja 7

Uitkomst: 91 deelbaar door 7? Ja 7

Uitkomst: 13 deelbaar door 7? Nee, door 11? Nee, door 13? Ja 13

Uitkomst: 1.

Dus:

**3822 = 2.3.7.7.13** (controleer dit met je rekenmachine).

### Wanneer is een getal deelbaar door 2, 3, 5, 7 of 11

Natuurlijk heb je de beschikking over een rekenmachine. Toch kan het handig zijn om uit je hoofd te kijken of een getal door één van de priemgetallen deelbaar is.

Deelbaar door 2: Een getal is deelbaar door 2 als het getal even is. Dat betekend dat het laatste cijfer van dit getal een 0, 2, 4, 6 of 8 is.

Deelbaar door 3 of 9: Een getal is deelbaar door 3 of 9 als de som (optelling) van de losse cijfers deelbaar is door 3 of 9.

**Voorbeeld:** 3524812752 is deelbaar door 3 omdat

3+5+2+4+8+1+2+7+5+2= 39 en 39 is deelbaar door 3.

Deelbaar door 5: Een getal is deel baar door 5 als het laatste cijfer een 0 of een 5 is.

Deelbaar door 7: Deel het getal van achteruit op in groepjes van 3 cijfers. (34276134 wordt dan 34, 276 en 134). Tel op en trek af om en om (dus 34-276+134). Indien nodig herhaal je dit tot er een getal van maximaal 3 cijfers overblijft (in het voorbeeld hiervoor komt er -108 uit). Als de uitkomst deelbaar door 7 is, is het oorspronkelijke getal ook deelbaar door 7 (ons voorbeeld dus niet).

**Voorbeeld:** 3524812753 is wel deelbaar door 7, want

Opdelen geeft 3, 524, 812 en 753. Optellen en aftrekken geeft 3-524+812-753 = -462 en dit is deelbaar door 7 (-66).

Deelbaar door 11: Neem de losse cijfers van het getal. Tel op en trek af om en om. Als de uitkomst deelbaar is door 11, dan is het oorspronkelijke getal ook deelbaar door 11.

**Voorbeeld:** 5058540795902 is deelbaar door 11 want

5-0+5-8+5-4+0-7+9-5+9-0+2=11 en dit is deelbaar door 11.

Deelbaar door 13: Hier geldt dezelfde regel als voor 7.

**Voorbeeld:** 102113856 is deelbaar door 13 want 102-113+856=845 en 845 is deelbaar door 13 dus het oorspronkelijke getal ook.

Deelbaar door niet priemgetallen:

Een getal is deelbaar door een ander getal als de eerste deelbaar is door alle priemfactoren van de tweede.

**Voorbeeld:** Een getal is deelbaar door bijvoorbeeld 6 als hij deelbaar is door 2 **EN** door 3. Omdat ieder getal is te ontbinden in priemgetallen kun je deze regel voor alle getallen toepassen.

## Oefenopgaven

1. Ontbindt de volgende getallen in priemfactoren:
   1. 12
   2. 210
   3. 6300
   4. 147
   5. 2133
   6. 30030
   7. 5265
   8. 137350
   9. 3016
   10. 7125321
2. Laat met de deelbaarheidsregels zien dat de volgende beweringen TRUE of FALSE zijn:
   1. 12 is deelbaar door 3
   2. 123429864239 is deelbaar door 2
   3. 12345678 is deelbaar door 3
   4. 534274524 is deelbaar door 5
   5. 687743946 is deelbaar door 9
   6. 38427767 is deelbaar door 7
   7. 852233915 is deelbaar door 13
   8. 247451072 is deelbaar door 11
   9. 375887676 is deelbaar door 6
   10. 136835045655 is deelbaar door 21

## Vereenvoudigen van breuken

Het ontbinden in priemfactoren kun je nu gebruiken om breuken te vereenvoudigen. Als je het getal in de teller ontbindt in priemfactoren, en je doet hetzelfde met de noemer, kun je overeenkomstige factoren in de teller en de noemer wegstrepen.

**Voorbeeld:**

Vereenvoudig de breuk:

Eerst ga je de teller en de noemer ontbinden in priemfactoren:

546 = 2.3.7.13

2340 = 2.2.3.3.5.13

De breuk wordt dan:

Boven en onder de breukstreep is een 2, een 3 en een 13 weggestreept.

## Oefenopgaven

Vereenvoudig waar mogelijk de volgende breuken:

1

## Buiten haakjes halen

Eerder zagen wij dat:

a( b + c ) = ab + ac

Natuurlijk kun je dit ook andersom uitwerken.

Je zoekt dan iets gemeenschappelijks in alle termen. Als alle termen een gemeenschappelijke factor hebben, dan kun je deze **buiten haakjes halen** (of **ontbinden in factoren**, immers, je maakt van een optelling van termen een vermenigvuldiging).

**Voorbeeld 1:**

Ontbind de volgende uitdrukking in factoren: p2 + pq

**Oplossing:**

Alle termen van deze uitdrukking hebben een **p**. Deze **p** mag je dan buiten haakjes halen. Binnen de haakjes moet je **p** dan wegdelen:

**Voorbeeld 2:**

Ontbind de volgende uitdrukking in factoren: 4ab – 4b2

**Oplossing:**

Alle termen van deze uitdrukking hebben een **4** en een **b**. Deze **4b** mag je dan buiten haakjes halen. Binnen de haakjes moet je **4b** dan wegdelen:

**Voorbeeld 3:**

Ontbind de volgende uitdrukking in factoren: 6x - 9y

**Oplossing:**

Alle termen van deze uitdrukking hebben een **3** (eigenlijk staat er 2.3.x – 3.3.y). Deze **3** mag je dan buiten haakjes halen. Binnen de haakjes moet je **3** dan wegdelen:

**Voorbeeld 4:**

Ontbind de volgende uitdrukking in factoren:

**Oplossing:**

Alle termen van deze uitdrukking hebben een **3a2**.Deze **3a2** mag je dan buiten haakjes halen. Binnen de haakjes moet je **3a2** dan wegdelen:

## Oefenopgaven

Ontbind in factoren:

1

* 1. ab + ac
  2. pq – pr
  3. cd + ce
  4. xy – xz
  5. ab + bc
  6. pq – qr
  7. ac – bc
  8. xz + yz
  9. 2a – 2b
  10. 5x + 5y
  11. 12p – 12q
  12. 7c + 7d

2

1. 5a – 10b
2. 8a + 16
3. a2 + 7a
4. 6a – 3b
5. 6c + 4d
6. a2 + a
7. ab – 5b
8. x2 – 2xy
9. ap + p
10. 10a + 15b
11. 4x2 – 8x
12. 2a2 + 2ab

3

1. a2b + ab2
2. 3ab + 6a
3. p2q2 – pq
4. 8a3 + 18a
5. 6p2 + 9q2
6. 2p2q – 6pq2
7. 9x2y2 + 6x2y
8. 14x3 – 21xy
9. 9p2 – 12pq
10. 18x2 – 24x
11. 22a2b + 33ab
12. 27p3 – 45p2

4

1. –ab + ac
2. –pq – pr
3. –ap – 2p
4. –xy + x
5. –3a + 3b
6. –6p – 12q
7. –7x + 21y
8. –8z – 4
9. –3a2b – 6ab
10. –8pq – 4p
11. –12x2 + 6x
12. –8a2 + 12a

5

1. ab – ac + ad
2. px + qr – rx
3. a2 – ab + 2a
4. x3 + x2 – x
5. 2a2 + 4ab + 6ac
6. 9pq – 12pr + 3p
7. 4x2y – 12xy2 + 20xy
8. 10a3 – 5a2 +2a
9. –pq + pr – 2p
10. –2ab – ac + 4a
11. –6x3 – 4x2 – 8x
12. –10a2b – 15ab + 5a

## De som-product regel

De som–product regel is een manier om tweedegraads vergelijkingen te ontbinden in factoren. Een tweedegraads vergelijking heeft de algemene vorm:

ax2 + bx + c

Deze vergelijking heet tweedegraads omdat onze variabele (x) als hoogste macht een 2 heeft. Zo is de vergelijking 2x5 + 3x2 + 6 een vijfdegraads vergelijking. En 3x + 5 een eerstegraads vergelijking.

De a, b en c noemen we constanten. Dit zijn vaste getallen.

Om de som-product regel te kunnen gebruiken moet je er voor zorgen dat de a in de algemene vorm van de tweedegraads vergelijking gelijk is aan 1. Dus a = 1. Dit kun je voor elkaar krijgen door het getal a buiten haakjes te halen.