Essential Skills Mathematics

Modulewijzer

Module L1

Propositielogica

Leerdoelen en onderwerpen

Leerdoelen

Na deze module bestudeerd te hebben moet je:

- Kunnen definiëren wat een propositie is
- De definities van de logische operatoren kennen en kunnen toepassen
- Waarheidstabellen kunnen maken en de waarheidswaarde van een logische expressie kunnen evalueren
- De begrippen *tautologie*, *contradictie* en *contingentie* kennen en logische expressies hiermee kunnen indelen en met behulp daarvan proposities kunnen herschrijven.

Onderwerpen

L1.1. Beschrijving van proposities

L1.2. Logische operatoren

L1.3. Waarheidstabellen

L1.4. Herschrijven van expressies

Antwoorden van de opgaven

L1.1. Beschrijving van proposities

De logica is een wetenschap die zich bezig houdt met methoden van redeneren om vast te kunnen stellen of een bepaalde bewering waar is of niet. Logica kan ook goed worden gebruikt om de correctheid van programma's aan te tonen en zelfs correcte programma's af te leiden. Ook bij veel andere wetenschappen wordt logica gebruikt om conclusies te formuleren of problemen op te lossen. In de logica spelen beweringen een grote rol. Als deze beweringen slechts waar of niet waar kunnen zijn, noemen we zo'n bewering een *propositie*. In de informatica is het gebruikelijk om waar en niet waar respectievelijk true en false te noemen. We zullen ons hierbij aansluiten. In algemene situaties, waar de actuele waarde van een propositie niet van belang is, praten we over de waarheidswaarde van een propositie.

Definitie:
Een *propositie* is een bewering die *true* of *false* is, maar niet beide.

Voorbeelden
1. De aarde is een planeet.
2. De ribben van een kubus zijn gelijk.
3. $2 + 9 = 11$.
4. Het oppervlak ingesloten door een cirkel is $2r$.
5. Op 1 augustus 1800 heeft het in Eindhoven de gehele dag hard gewaaid.

Van deze voorbeelden zijn 1, 2 en 3 true, 4 is false en van 5 is de waarheidswaarde op dit moment moeilijk vast te stellen, maar het is wel een bewering die true of false is, zodat het een propositie is.

De volgende beweringen zijn *geen* proposities:
1. $a + b = 7$.
2. $x > 7$.
3. Het socialisme is schuldig aan de huidige crisis.

De waarheidswaarde van de beweringen 1 en 2 kan niet worden vastgesteld zolang de waarden van a, b en x niet bekend zijn. Dat betekent dat deze beweringen zowel true als false kunnen zijn. Op het moment dat de waarden bekend zijn worden de beweringen proposities. De derde bewering is geen propositie omdat de waarheidswaarde voor verschillende personen verschillend zal zijn, het is een mening.

Opgave L1.1.

Geef aan of de volgende uitspraken proposities zijn:

a) Dat is een mooi schilderij.

b) Henk is groter dan Kees.

c) De toets van Module [1] was makkelijk!

Ga nu in Grasple aan de slag met opgaven van dit onderdeel om na te gaan of je de stof goed hebt begrepen.

L1.2. Logische operatoren

In de wiskunde is het gebruikelijk om variabelen voor te stellen met behulp van letters die kunnen worden vervangen door 'echte' waarden. Verder kunnen deze variabelen worden gecombineerd tot grotere gehelen door middel van de bekende operatoren als +, x, -, enz.
In de logica worden dit soort conventies ook gebruikt en noemt men de letters (vaak *p, q, r, …*) propositievariabelen. Deze propositievariabelen kunnen dan met behulp van logische operatoren worden gecombineerd tot grotere gehelen, de propositionele vormen of logische expressies.

Voorbeeld 1:
Gegeven de volgende proposities:
p: De zon schijnt vandaag.
q: Het regent.
r: Ik ben gelukkig.

De bewering 'De zon schijnt vandaag *en* ik ben gelukkig' kan dan ook worden geschreven als: *p* en *r*. Zo is het ook mogelijk een bewering als 'Het regent *of* de zon schijnt vandaag' te schrijven als: *q* of *r*. We noemen 'en' en 'of' de logische operatoren (connectieven).

<u>Negatie</u>
Als *p* een propositie is, dan is de ontkenning van *p* ook een propositie. De negatie (of ontkenning) van *p* neemt immers ook de waarde true of false aan, afhankelijk van de waarde van *p*. De ontkenning van *p* wordt geschreven als $\neg p$ en wordt uitgesproken als *niet p.* Omdat de ontkenningsoperator slechts op één propositie werkt noemen we dit een *unaire* operator. Uit de informele definitie kunnen we afleiden dat indien *p* true is, $\neg p$ false is en omgekeerd. In een tabel kunnen we dat als volgt noteren:

p	$\neg p$
T	F
F	T

Zo'n tabel noemen we een waarheidstabel.

Voorbeeld 2:
Gegeven de volgende propositie:
p: Op dit moment schijnt in Amsterdam de zon.
Dan is de negatie:
$\neg p$: Het is niet zo dat op dit moment de zon schijnt in Amsterdam.
Dit kunnen we natuurlijk ook kunnen lezen als:
$\neg p$: Op dit moment schijnt de zon niet in Amsterdam.

Conjunctie

Twee proposities kunnen tot een nieuwe propositie worden samengesteld met behulp van de binaire operator 'en'. Een binaire operator is een operator die op twee proposities werkt. We noemen dit de conjunctie van de twee oorspronkelijke proposities. De conjunctie van p en q wordt geschreven als: $p \wedge q$ en uitgesproken als: p en q. De conjunctie van p en q is true als p en q beide true zijn en false in alle andere gevallen. Hieronder staat de definitie in de vorm van een tabel:

p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

We beschouwen in het vervolg enkel deze tabel (en de andere in dit hoofdstuk) als de *definitie* van de operator ∧ en niet het informele taalgebruik.

Voorbeelden:
1. ‘Een schimmel is een wit paard’ ∧ ’2+3 = 5'
2. 'Een schimmel is een bruin paard’ ∧ ‘2+3 = 5'
3. 'Een schimmel is een bruin paard’ ∧ '2+3 = 6'
4. 'Een schimmel is een wit paard’ ∧ '2+3 = 6'

In deze voorbeelden is alleen de eerste propositie true en de andere drie zijn false omdat tenminste één van de twee proposities false is.

Programmeervoorbeeld
In controlestructuren van programma's komt het gebruik van logische operatoren vaak voor. De programma opdracht:

```
IF ((x<3) AND (z=2)) THEN x:=(x+z) ELSE x:=(x+2)
```

zorgt ervoor dat x: = (x+z) alleen wordt uitgevoerd als geldt: (x<3) èn (z=2); in alle andere gevallen zal x: = (x+2) worden uitgevoerd.

Merk op dat ((x<3) AND (Z=2)) geen proposities zijn. Of deze uitspraak true of false is, hangt af van de waarden die x en z op een bepaald moment hebben. Indien echter de waarden van x en z bekend zijn (in dit geval zal dat zijn tijdens de uitvoering van het programma), dan is deze uitspraak een propositie geworden (hij is dan immers true of false). Dit soort uitspraken noemen we predikaten en zal worden besproken in Module L2.

Ook combinaties met andere operatoren komen voor:

```
IF (NOT(x>=3) AND NOT (z<>2)) THEN x:=(x+z) ELSE x:=(x+2)
IF (NOT(x>=3) OR (z<>2)) THEN x:=(x+z) ELSE x:=(x+2)
```

Geven hetzelfde resultaat.

<u>Disjunctie</u>
Net zoals de conjunctie als logische voorstelling van het in de taal gebruikte woord 'en' wordt gebruikt is de disjunctie (∨) de voorstelling van 'of'. Zij wordt gedefinieerd met behulp van de volgende tabel:

p	q	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F

Deze definitie toont aan dat $p \vee q$ true is als (tenminste) één van de proposities true is. De disjunctie wordt dus genoteerd als $p \vee q$ en uitgesproken als p of q.

Houd je strikt aan de definitie uit de tabel, want het gebruik van 'of' in natuurlijke taal kan tot misverstanden of dubbelzinnigheden aanleiding geven.
Beschouw hiervoor de zin: ‘Vandaag zal de zon schijnen of het zal regenen’. Iemand die deze uitspraak doet, zal waarschijnlijk niet voor leugenaar worden uitgemaakt, als vandaag de zon schijnt èn het regent.
Maar in de zin: ‘Je moet je kamer opruimen of je moet de afwas doen’, wordt waarschijnlijk niet van je verwacht dat je zowel je kamer opruimt, als ook de afwas doet.

Voorbeelden:
1. 'Een schimmel is een wit paard’ ∨ '2+3 = 5'
2. 'Een schimmel is een bruin paard’ ∨ '2+3 = 5'
3. 'Een schimmel is een wit paard’ ∨ '2+3 = 6'
4. 'Een schimmel is een bruin paard’ ∨ '2+3 = 6'

In deze voorbeelden zijn de eerste drie proposities true en de laatste is false omdat hier deze beide proposities false zijn.

Dus: zoals al vermeld, is de logische 'of' equivalent met de Nederlandse zin: òf het een òf het ander òf beide. Dit wordt ook wel de ‘inclusive or’ genoemd.
Dit in tegenstelling tot de ‘exclusive or’ (XOR) waarvoor geldt: òf het een, òf het ander, maar niet beide.

Implicatie
Een derde binaire operator is de implicatie. Deze operator maakt het mogelijk om 'oorzaak-gevolg' uitspraken in een logische expressie vorm te geven. Denk daarbij bijvoorbeeld aan: ‘Als ik genoeg geld heb, dan koop ik een nieuwe fiets.’

De implicatie wordt geschreven als $p \rightarrow q$ en uitgesproken als p impliceert q.
De definitie in de vorm van een tabel is de volgende:

p	q	$p \rightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T

Hieruit blijkt dat de implicatie altijd de waarde true heeft, tenzij de eerste propositie true is en de tweede false. De eerste twee regels van de tabel zijn gevoelsmatig ook logisch. Voor de derde en vierde regel geldt, dat als de eerste bewering false is, het lastig is een conclusie te trekken. Daarom is afgesproken dat de stelling in dat geval altijd true is.

De propositie p wordt de premisse (ook wel hypothese, oorzaak of antecedent) genoemd en q de conclusie (ook wel gevolg of consequentie). Taalgebruik waarin implicaties een rol spelen komt vaak voor.

Voorbeeld 1:
De propositie $p \rightarrow q$ kan de logische vertaling zijn van zinnen als:
'als p, dan q'
'q volgt uit p'
'p is een voldoende voorwaarde voor q'
'q is het gevolg van p'
'q mits p'

Toch kunnen ook hier problemen ontstaan, zoals blijkt uit het volgende voorbeelden.

Voorbeeld 2:
Beschouw de uitspraak 'Als jij de kamer stoft, dan zal ik afwassen'. Voor de meesten zal dit betekenen dat als de andere de kamer stoft, de eerste persoon de afwas zal doen. Geschreven als implicatie: 'Jij stoft de kamer' → 'Ik was af'. In deze notatie is 'Jij stoft de kamer' de premisse en 'Ik was af' de conclusie. Uit de definitietabel blijkt dat indien de premisse false is 'Jij stoft de kamer niet', dat dan de conclusie toch true kan zijn 'Ik was af'. Waarschijnlijk is dat niet de bedoeling. Een implicatie kan soms raar overkomen.

Voorbeeld 3:
Beschouw de volgende propositie: 'Als er kabouters bestaan dan is de aarde een meloen'. De implicatie op zich is true, want de premisse 'Kabouters bestaan is uiteraard false', maar er kan natuurlijk niet de conclusie aan worden verbonden dat de aarde inderdaad een meloen is. Pas dus op voor situaties waarin de premisse false is!

De implicatie levert vaak meer problemen op dan de andere logische operatoren. Dat komt door de logische regel dat uit een FALSE-hypothese alles kan worden afgeleid. Zo is FALSE → FALSE gelijk aan TRUE. En zelfs FALSE → TRUE is TRUE. Willekeurige proposities invullend levert dat verschrikkelijke onzin op: 'Als 2 oneven is, dan is 7 x 3 = 21' is dus een ware bewering, strikt logisch gezien. Zo ook is 'Als 2 oneven is, dan is 7 x 3 = 22' een ware bewering in strikt logische zin. Deze logica lijkt niet erg zinvol!

Voorbeeld 4:
Toch is er in het spraakgebruik wel iets te herkennen van deze logische afspraak. In de uitspraak 'Als Piet zijn propedeuse haalt, dan vreet ik mijn schoenen op' gaat de spreker er natuurlijk van uit dat de hypothese tòch niet waar zal blijken te zijn. Blijkbaar kun je uit een foute veronderstelling alles afleiden!

Overigens houdt de logica zich niet bezig met de *semantiek* van haar uitspraken. Dat wil zeggen: zij trekt zich niets aan van wat de proposities en de samengestelde proposities voor een betekenis hebben in de dagelijkse wereld; zij houdt zich slechts bezig met strikte afspraken over wat geldig is binnen de door ons mensen opgestelde strakke logische structuur.

Voorbeeld 5:
Een voorbeeld nog om te laten zien wat hier bedoeld wordt: met de regel "TRUE → TRUE wordt TRUE" heeft niemand moeite. Maar wat is de betekenis dan van: 'Als 2 even is, dan is 7 x 3 = 21'? De logica is niet geïnteresseerd in het antwoord op deze vraag. De logica stelt slechts vast dat de waarheidswaarde van deze propositie TRUE is.

Als de logica dus indruist tegen je 'gevoel' dan moet je je maar vasthouden aan de formele afspraken die we in dit hoofdstuk hebben gemaakt: de waarheidstabel heeft het in de logica voor het zeggen, en de 'informele vertaling' van een propositie in een natuurlijke taal als het Nederlands, is vaak dubbelzinnig en voor meerdere uitleg vatbaar. In dit verband onderstrepen we nogmaals de opmerking: We beschouwen in het vervolg enkel de tabellen in dit hoofdstuk als de definitie van de operatoren en niet het informele taalgebruik.

Equivalentie
De laatste operator die we zullen bespreken is de equivalentie of gelijkheid. Deze operator heeft de waarde true indien beide proposities dezelfde waarheidswaarde hebben. We noteren de equivalentie als $p \leftrightarrow q$ en spreken dit uit als p is equivalent met q. Soms wordt de equivalentie uitgesproken als 'p dan en slechts dan als q', ook wel eens afgekort tot 'p desda q'. Ook andere afkortingen komen voor 'p asa q' (als en slechts als') en het Engelse: 'p iff q' (if and only if).

Er geldt:

p	q	$p \leftrightarrow q$
T	T	T
T	F	F
F	T	F
F	F	T

Voorbeeld 1:
Let op dat de equivalentie enkel aangeeft of twee logische operanden dezelfde waarheidswaarde hebben. Aangezien beide operanden in de logische expressie (5 >7) (12 < (5+7)) false zijn heeft de expressie dus in zijn geheel de waarde true.

Voorbeeld 2:
Nog een opmerking over het taalgebruik bij equivalentie. Bekijk als voorbeeld de volgende zin: ´Ik ga alleen afwassen, als jij de kamer stofzuigt`. Op het eerste gezicht lijkt dit een implicatie, maar wij interpreteren deze zin als volgt: ´als jij de kamer stofzuigt, dan ga ik afwassen´, maar ook: ´als ik ga afwassen, dan stofzuig jij de kamer´, want ik was alleen maar dan af, als jij stofzuigt, dus als ik afwas dan moet jij wel aan het stofzuigen zijn. Dit betekent dat het één de ander impliceert én omgekeerd, dat wil zeggen: het is een equivalentie! Kortom: ´p alleen als q vertalen wij met $p \leftrightarrow q$.

Opgave L1.2.1

We definiëren de volgende uitspraken:
- p : Ik heb honger.
- q : Ik eet.
- r : Ik word dik.

Geef van de volgende beweringen het logisch equivalent:
a) Als ik honger heb, eet ik en word ik dik.
b) Ik eet alleen maar als ik honger heb.
c) Ik eet, ongeacht of ik honger heb of niet.
d) Ik word dik als ik eet.

Opgave L1.2.2

Voor de definitie van p , q en r zie vraag 1.2.1. Vertaal de volgende logische expressies in een goed lopende, Nederlandse zin:

a) $q \to r$

b) $\neg q \to p$

c) $(\neg q \to \neg r) \wedge (q \to p)$

Ga nu in Grasple aan de slag met opgaven van dit onderdeel om na te gaan of je de stof goed hebt begrepen.

L 1.3. Waarheidstabellen

Tot nu toe zijn waarheidstabellen gebruikt om de definitie van een operator vast te leggen. Waarheidstabellen kunnen ook worden gebruikt om grotere propositionele vormen te onderzoeken. Het is met behulp van een waarheidstabel mogelijk om bijvoorbeeld te onderzoeken of twee propositionele vormen gelijk zijn (equivalent). Ook kan de waarheidswaarde van een logische expressie worden bepaald (zie voorbeeld 2 hieronder).

Volgens de definitie van equivalentie zijn twee proposities (en dus ook twee propositionele vormen) equivalent als beide operanden (altijd) dezelfde waarheidswaarde hebben. We kunnen dat onderzoeken door 'eenvoudigweg' alle mogelijk gevallen te onderzoeken. Let er wel op dat het aantal te onderzoeken gevallen exponentieel toeneemt naarmate er meer propositie-variabelen in de vorm voorkomen. In het algemeen zal met drie variabelen nog gewerkt kunnen worden (acht gevallen). Vier variabelen vraagt al zestien gevallen te onderscheiden.

Voorbeeld 1:

Stel dat we willen bewijzen dat $p \rightarrow q$ equivalent is met $\neg p \vee q$.

Oftewel: $(p \rightarrow q) \leftrightarrow (\neg p \vee q)$. We gebruiken daarvoor onderstaande tabel.

Het beste is nu, om in de eerste twee kolommen alle mogelijke combinaties van p en q op te sommen. Aangezien $\neg p$ ook voorkomt in één van de proposities, wordt in de derde kolom de waarheidswaarde van $\neg p$ vastgesteld. Vervolgens wordt de waarheidswaarde van $p \rightarrow q$ voor al die combinaties onderzocht en in de vierde kolom vastgesteld. Daarna wordt de waarheidswaarde van $\neg p \vee q$ voor al die combinaties onderzocht en genoteerd in de vijfde kolom. Tenslotte kan de equivalentie van deze twee proposities worden aangegeven in de laatste kolom. Slechts indien in deze laatste kolom overal een T verschijnt hebben we bewezen dat $p \rightarrow q$ equivalent is met $\neg p \vee q$ voor alle mogelijke combinaties van de waarde van de propositievariabelen p en q.

Kolom 1	Kolom 2	Kolom 3	Kolom 4	Kolom 5	Kolom 6
p	q	$\neg p$	$p \rightarrow q$	$\neg p \vee q$	$(p \rightarrow q) \leftrightarrow (\neg p \vee q)$
T	T	F	T	T	T
T	F	F	F	F	T
F	T	T	T	T	T
F	F	T	T	T	T

Opmerking: Het is gebruikelijk om de waarheidswaarde onder de operator te noteren.

Voorbeeld 2:
Nog een voorbeeld, maar dan met drie propositie-variabelen. Onderzocht dient te worden wat de waarheidswaarde van de volgende propositionele vorm is:
$((p \leftrightarrow r) \wedge (r \leftrightarrow q)) \rightarrow (p \leftrightarrow q)$
Daartoe gebruiken we de volgende tabel:

Kolom 1	Kolom 2	Kolom 3	Kolom 4	Kolom 5	Kolom 6	Kolom 7	Kolom 8
p	q	r	$p \leftrightarrow r$	$\wedge$	$r \leftrightarrow q$	$\rightarrow$	$p \leftrightarrow q$
T	T	T	T	T	T	T	T
T	T	F	F	F	F	T	T
T	F	T	T	F	F	T	F
T	F	F	F	F	T	T	F
F	T	T	F	F	T	T	F
F	T	F	T	F	F	T	F
F	F	T	F	F	F	T	T
F	F	F	T	T	T	T	T

Eerst worden de kleinste vormen onderzocht (kolom 4, 6 en 8), waarna uit de kolommen 4 en 6 kolom 5 kan worden herleid, tenslotte gevolgd door kolom 7 (Ga dit na!). Uit de vorm van de zevende kolom blijkt duidelijk dat deze propositionele vorm altijd true is.

Overigens bepaalt het aantal operanden (ook wel atomaire proposities genoemd) uit hoeveel regels een waarheidstabel moet bestaan. Met 3 operanden (p, q en r bijvoorbeeld) zijn er $2 \cdot 2 \cdot 2 = 2^3 = 8$ verschillende combinaties mogelijk van waarheidswaarden. Dus de waarheidstabel van bv. $p \wedge q \vee r$ heeft 8 regels. Om geen enkele combinatie van waarheidswaarden te missen structureren we zo'n waarheidstabel meestal zó:

p	*q*	*r*
T	*T*	*T*
T	*T*	*F*
T	*F*	*T*
T	*F*	*F*
F	*T*	*T*
F	*T*	*F*
F	*F*	*T*
F	*F*	*F*

waarmee we alle 8 verschillende combinaties hebben gehad.

Indien een propositionele vorm altijd de waarde true heeft noemen we deze een *tautologie*, indien de waarde altijd false is een *contradictie* of absurditeit; in alle andere gevallen spreken we van een *contingentie*.

Definitie:
Een tautologie is een propositionele vorm waarvan de waarheidswaarde true is voor alle mogelijke waarden van de propositie-variabelen.

Definitie:
Een contradictie of absurditeit is een propositionele vorm waarvan de waarheidswaarde false is voor alle mogelijke waarden van de propositie-variabelen.

Definitie:
Een propositionele vorm die geen tautologie of contradictie is, is een contingentie.

Voorbeelden:

p is een contingentie;

$p \vee \neg p$ is een tautologie;

$p \wedge \neg p$ is een contradictie.

De laatste vorm wordt ook wel eens omschreven als 'het uitgesloten wonder'.

In sommige gevallen is het niet nodig om alle gevallen te onderzoeken om toch een uitspraak te kunnen doen over de waarheidswaarde van een propositie. In dat geval moet gezocht worden naar een bijzonder geval dat nader moet worden onderzocht.

Voorbeeld:
We willen onderzoeken of $(p \wedge q) \to p$ een tautologie is. Daartoe hoeven we eigenlijk alleen maar te zoeken naar een geval waarin de implicatie false kan worden. De enige manier waarop$(r \to t)$ de waarde false kan hebben is het geval waarin r true is en t false. In ons voorbeeld betekent dit het vinden van een geval waarin $p \wedge q$ true is en p false. Om $p \wedge q$ true te laten zijn moeten zowel p als q de waarde true hebben. Alleen dat geval moeten we dus nader onderzoeken. Het is duidelijk dat dan p niet meer false kan zijn! Deze vorm van onderzoek noemen we wel gepast gevallen onderscheid, of verkorte waarheidstabel indien het onderzoek wordt gedaan met behulp van een tabel:

p	q	$p \wedge q$	$\to$	p
T	T	T	T	T

Contrapositie
De contrapositie is van speciaal belang. Het is een alternatieve manier om een implicatie te beschrijven. Dat wil zeggen: $p \to q$ en $\neg q \to \neg p$ hebben dezelfde waarheidstabel. Let op: hele volksstammen denken dat uit de regel: 'als het regent, dan word ik nat' wel volgt: 'als het niet regent dan word ik niet nat'. Maar dat is niet correct. Als je immers bv. onder de douche staat word je ook nat. Wèl geldt: 'als ik niet nat ben, dan heeft het ook niet geregend.' Deze laatste zin is de contrapositie van de allereerste in dit voorbeeld. In logische bewoordingen geldt dus dat: $(p \to q) \leftrightarrow (\neg q \to \neg p)$ een tautologie is, maar $(p \to q) \leftrightarrow (\neg p \to \neg q)$ niet.

Opgave L1.3.1

Maak voor de volgende propositionele vormen een waarheidstabel:

a) $(p \vee q) \wedge p$

b) $(p \rightarrow q) \wedge (p \vee q)$

c) $p \leftrightarrow (p \wedge (q \vee p))$

Ga nu in Grasple aan de slag met opgaven van dit onderdeel om na te gaan of je de stof goed hebt begrepen.

L1.4. Herschrijven van expressies

Eigenschappen van logische operatoren
Zoals bekend, bestaan er in de rekenkunde bepaalde eigenschappen en afspraken, waaraan operatoren, zoals optelling en vermenigvuldiging, voldoen. Er wordt dan gewerkt met een prioriteitsvolgorde: (Hoe Moeten We Van Die Onvoldoendes Afkomen), oftewel: haakjes uitwerken, machtsverheffen, worteltrekken, vermenigvuldigen, delen, optellen, aftrekken.

Ook bij logische operatoren is er een prioriteitsvolgorde (in dalende prioriteit):
niet ($\neg$) - en ($\wedge$) - of ($\vee$) - implicatie ($\rightarrow$) - gelijkheid ($\leftrightarrow$).

Voorbeeld:
$\neg q \leftrightarrow q \wedge r$ is gelijk aan $(\neg q) \leftrightarrow (q \wedge r), q \vee \neg r \rightarrow s$ is gelijk aan $(q \vee (\neg r)) \rightarrow s.$
We adviseren toch, om van haakjes gebruikt te maken, omdat de logische expressie met haakjes veel makkelijker leest en enig (onbedoeld) misverstand opheft.

Andere eigenschappen die bij rekenkundige operatoren een rol spelen zijn:
- commutativiteit $a \cdot b = b \cdot a$
- associativiteit $(a + b) + c = a + (b + c)$
- distributiviteit $a \cdot (b + c) = a \cdot b + a \cdot c$

Opmerking: Dit geldt natuurlijk niet voor alle rekenkundige bewerkingen; zo zijn de '$-$ ' en de '$/$ ' operatoren niet commutatief ($x - y \neq y - x$)!

Voor de logische operatoren gelden de volgende eigenschappen: '$\vee$ ' en '$\wedge$ 'zijn zowel commutatief als associatief en '$\vee$ ' distribueert over '$\wedge$ ' en '$\wedge$ ' over '$\vee$ '. In logische expressies opgeschreven wordt dit:

*commutativiteit : (a * b) = (b * a)*
$(p \wedge q) \Leftrightarrow (q \wedge p)$
$(p \vee q) \Leftrightarrow (q \vee p)$
$(p \leftrightarrow q) \Leftrightarrow (q \leftrightarrow p)$

associativiteit: (a + b) + c = a + (b + c) = a + b + c
$(p \wedge (q \wedge r)) \Leftrightarrow (p \wedge q) \wedge r) \Leftrightarrow p \wedge q \wedge r$
$(p \vee (q \vee r)) \Leftrightarrow (p \vee q) \vee r) \Leftrightarrow p \vee q \vee r$
$(p \leftrightarrow (q \leftrightarrow r)) \Leftrightarrow ((p \leftrightarrow q) \leftrightarrow r) \Leftrightarrow p \leftrightarrow q \leftrightarrow r$

distributiviteit: $a * (b + c) = a * b + a * c$
$(p \wedge (q \vee r)) \Leftrightarrow (p \wedge q) \vee (p \wedge r))$
$(p \vee (q \wedge r)) \Leftrightarrow (p \vee q) \wedge (p \vee r))$
Verder gelden er nog enkele eigenschappen, zoals:

idempotentie:
$(p \wedge p) \Leftrightarrow p$
$(p \vee p) \Leftrightarrow p$

dubbele ontkenning:
$\neg(\neg p) \Leftrightarrow p$

De Morgan:
$\neg(p \wedge q) \Leftrightarrow (\neg p \vee \neg q)$
$\neg(p \vee q) \Leftrightarrow (\neg p \wedge \neg q)$

Implicatieregel:
$(p \longrightarrow q) \Leftrightarrow (\neg p \vee q)$
Dit schrijven we zo omdat is bewezen dat dit een tautologie betreft.

De correctheid van deze expressies kan eenvoudig via een waarheidstabel worden aangetoond. Vooral de wetten van De Morgan spelen bij het opstellen van programma's een grote rol. Vaak is het mogelijk om ontkenningen in constructies te vereenvoudigen.

Programmeervoorbeeld:
In een programma moet een opdracht S herhaaldelijk worden uitgevoerd totdat geldt dat x tenminste 0 is of x kleiner is dan 0, maar in dat geval moet y groter dan 0 zijn. In een formele beschrijving ziet dat er als volgt uit: $(x \geq 0) \vee ((x < 0) \wedge (y > 0))$

De constructie die daarvoor zorgt zou kunnen zijn:
WHILE NOT (x >= 0) OR ((x < 0) AND (y>0)) DO S
Deze opdracht zorgt er immers voor dat S steeds wordt uitgevoerd totdat op een bepaald moment wel geldt $(x \geq 0) \vee ((x < 0) \wedge (y > 0))$. De opdracht S moet natuurlijk wel invloed hebben op de waarden van x en y. Als namelijk x en y de waarden zouden behouden zoals die waren voor het begin van deze constructie, dan zal dit stukje van het programma nooit eindigen. Op het moment dat $(x \geq 0) \vee ((x < 0) \wedge (y > 0))$ tot *true* evalueert, wordt de constructie:
WHILE false DO S; en zal S niet meer worden uitgevoerd.
De expressie $\neg ((x \geq 0) \vee ((x < 0) \wedge (y > 0)))$ laat zich gelukkig wel wat vereenvoudigen (rechts staat steeds op welke eigenschap we ons beroepen om de equivalentie met de

volgende expressie aan te tonen):

$\neg ((x \geq 0) \vee ((x < 0) \wedge (y > 0))$ {wet van De Morgan}
$\Leftrightarrow (\neg (x \geq 0) \wedge \neg ((x < 0) \wedge (y > 0))$ {wet van De Morgan}
$\Leftrightarrow (\neg (x \geq 0) \wedge (\neg (x < 0) \vee \neg (y > 0))$ {eigenschap van >,< en ≥}
$\Leftrightarrow ((x < 0) \wedge ((x \geq 0) \vee (y \leq 0))$ {distributiviteit}
$\Leftrightarrow ((x < 0) \wedge (x \geq 0)) \vee ((x < 0) \wedge (y \leq 0))$ {eigenschap getallen}
$\Leftrightarrow F \vee ((x < 0) \wedge (y \leq 0))$ {false-true regels}
$\Leftrightarrow ((x<0) \wedge (y \leq 0))$
Deze afleiding vereenvoudigt dus de constructie tot:
WHILE (x < 0) AND (y<=0) DO S

Zoals uit het bovenstaand voorbeeld blijkt, kan op grond van de beschreven eigenschappen het gebruik van waarheidstabellen worden vermeden (zeker gewenst als er meer propositie-variabelen in een expressie voorkomen). De techniek hierbij is om (delen van) de expressie te herschrijven totdat de gewenste vorm is verkregen. Het is hierbij gebruikelijk om de argumenten voor de herschrijving als een soort commentaar in de afleiding op te nemen, zoals in het voorgaande voorbeeld al is gedaan.

De herschrijfregels zijn gebaseerd op het toepassen van *tautologieën*. Over de schrijfwijze van deze logische wetten is nog wel het één en ander op te merken.

In de equivalentie $p \leftrightarrow q$ is het teken een logische operator, met een bepaald rekenvoorschrift. Zoals gebleken kan het berekenen van de uitkomst van logische expressies soms leiden tot de conclusie dat er sprake is van een tautologie.
Zo is de equivalentie $(p \rightarrow q) \leftrightarrow (\neg p \vee q)$ een tautologie, maar de equivalentie $(p \wedge q) \leftrightarrow (\neg p \rightarrow q)$ is dat niet. Daardoor is de eerste een herschrijfregel en de tweede niet.
We noemen een equivalentie die een tautologie is ook wel een l*ogische equivalentie*.

Er is een schrijfwijze waarmee we meteen het verschil duidelijk kunnen maken tussen een logische equivalentie en een ´gewone´ equivalentie. We gebruiken de $\Leftrightarrow$ voor logische equivalenties.

Conclusie:
We hebben bewezen hebben dat $(p \rightarrow q) \leftrightarrow (\neg p \vee q)$ een tautologie is. Daarom zullen we voortaan schrijven: $(p \rightarrow q) \Leftrightarrow (\neg p \vee q)$.
In herschrijvingen waar je gebruik maakt van de logische equivalenties van het fomuleblad ‘Logische Equivalenties’ zul je ook zelf van het teken gebruik moeten maken.

In het algemeen vindt men het gebruik van herschrijfregels moeilijker dan het maken van waarheidstabellen. Beide technieken zijn geschikt om de logische equivalentie van twee proposities aan te tonen. Herschrijven is echter een noodzakelijke vaardigheid, die bij het programmeren en het vereenvoudigen van digitale schakelingen nog vaak van pas zal komen. De reden voor de moeilijkheidsgraad ligt in het feit dat er niet één methode is aan te geven wat je moet doen. Vaak zijn er meerdere regels van toepassing op een propositie. Welke regel je dan moet gebruiken ligt aan wat je wilt bereiken, en de 'verkeerde' weg inslaan is een onvermijdelijke stap naar dieper inzicht en grotere ervaring. Gewoon blijven proberen!

We geven nu twee voorbeelden van het gebruik van herschrijfregels, de gebruikte namen verwijzen naar de regels op het fomuleblad 'Logische Equivalenties'. Waar het symbool * wordt gebruikt wordt aangegeven dat deze regel meerdere keren is toegepast.

Voorbeeld 1:

$(p \rightarrow (q \rightarrow r)) \rightarrow ((p \wedge q) \rightarrow r)$	pas de implicatieregel toe
$\Leftrightarrow (\neg p \vee (\neg q \vee r)) \rightarrow (\neg(p \wedge q) \vee r)$	pas associativiteit toe
$\Leftrightarrow (\neg p \vee \neg q \vee r) \rightarrow (\neg(p \wedge q) \vee r)$	pas De Morgan toe
$\Leftrightarrow (\neg p \vee \neg q \vee r) \rightarrow (\neg p \vee \neg q) \vee r)$	pas associativiteit toe
$\Leftrightarrow (\neg p \vee \neg q \vee r) \rightarrow (\neg p \vee \neg q \vee r)$	pas False/True regels toe
$\Leftrightarrow T$	

Voorbeeld 2:

$(p \wedge q) \rightarrow (p \vee q)$	pas de implicatieregel toe
$\Leftrightarrow \neg(p \wedge q) \vee (p \vee q)$	pas De Morgan toe
$\Leftrightarrow (\neg p \vee \neg q) \vee (p \vee q)$	pas associativiteit toe
$\Leftrightarrow (\neg p \vee \neg q \vee p \vee q)$	pas commutativiteit toe
$\Leftrightarrow (\neg p \vee p \vee \neg q \vee q)$	pas associativiteit toe
$\Leftrightarrow (\neg p \vee p) \vee (\neg q \vee q)$	pas False/True regels toe
$\Leftrightarrow T \vee T$	pas idempotentie toe
$\Leftrightarrow T$	

Opgave L1.4.

Vereenvoudig de volgende expressies doormiddel van herschrijven:

a) $(q \rightarrow p) \wedge (\neg p \rightarrow q) \wedge (q \rightarrow q)$

b) $((\neg p \vee q) \vee (\neg p \vee r)) \rightarrow (q \vee r)$

Ga nu in Grasple aan de slag met opgaven van dit onderdeel om na te gaan of je de stof goed hebt begrepen.

Antwoorden van de opgaven

Opgave L1.1.

a) Geen propositie

b) Propositie

c) Geen propositie

Opgave L1.2.1

a) $p \rightarrow (q \wedge r)$
b) $q \leftrightarrow p$
c) q
d) $q \rightarrow r$

Opgave L1.2.2

a) Als ik eet dan word ik dik.

b) Als ik niet eet dan heb ik honger.

c) Als ik niet eet dan word ik niet dik en als ik eet dan heb ik honger.

Opgave L1.3.

a) $(p \vee q) \wedge p$

p	q	$(p \vee q) \wedge p$
T	T	T
T	F	T
F	T	F
F	F	F

b) $(p \rightarrow q) \wedge (p \vee q)$

p	q	$(p \rightarrow q) \wedge (p \vee q)$
T	T	T
T	F	F
F	T	T
F	F	F

c) $p \leftrightarrow (p \wedge (q \vee p))$

p	q	$p \leftrightarrow (p \wedge (q \vee p))$
T	T	T
T	F	T
F	T	T
F	F	T

Opgave L1.4.

a) $(q \to p) \wedge (\neg p \to q) \wedge (q \to q)$

$(q \to p) \wedge (\neg p \to q) \wedge (q \to q)$ {Implicatieregel 3x}
$\Leftrightarrow (\neg q \vee p) \wedge (p \vee q) \wedge (\neg q \vee q)$ {False/True regel}
$\Leftrightarrow (\neg q \vee p) \wedge (p \vee q) \wedge$ T {Distributieregel}
$\Leftrightarrow p \vee (\neg q \wedge q) \wedge$ T {False/True regel}
$\Leftrightarrow p \vee$ F $\wedge$ T {False/True regel}
$\Leftrightarrow p$

b) $((\neg p \vee q) \vee (\neg p \vee r)) \to (q \vee r)$

$((\neg p \vee q) \vee (\neg p \vee r)) \to (q \vee r)$ {Associativiteit}
$\Leftrightarrow (\neg p \vee q \vee \neg p \vee r) \to (q \vee r)$ {Implicatieregel}
$\Leftrightarrow \neg(\neg p \vee q \vee \neg p \vee r) \vee (q \vee r)$ {Simplificatie}
$\Leftrightarrow \neg(\neg p \vee q \vee r) \vee (q \vee r)$ {De Morgan}
$\Leftrightarrow (p \wedge \neg q \wedge \neg r) \vee (q \vee r)$ {Distributieregel}
$\Leftrightarrow (p \vee (q \vee r)) \wedge (\neg(q \vee r) \vee (q \vee r))$ {False/True regel}
$\Leftrightarrow (p \vee (q \vee r)) \wedge$ T {Associativiteit}
$\Leftrightarrow (p \vee q \vee r) \wedge$ T {False/True regel}
$\Leftrightarrow (p \vee q \vee r)$