Y.C.C.-cursus

BOOLEAANSE ALGEBRA

0001011100
0111101111
0011011101
0000111111
1001101011
1001000011
0110100010
1110000100
1110010101
1100000010
0100100101
1101100011
1110001110
1010111011
1111001001
0001011101
1101111100
1100101100
1001001101
1100010100

Koen Lefever

Y.C.C.-cursus: BOOLEAANSE ALGEBRA.

Iedereen die dieper op zijn computer wil ingaan komt vroeg of laat in contact met computerlogica. Eigenlijk is deze logica identiek aan de symbolische logica of de booleaanse algebra. (Genoemd naar de vader van de symbolische logica: Georges Boole .)

Toen ik in het ledenblad van de Y.C.C., $tring$, vier korte artikels schreef merkte ik dat er interesse is voor dit onderwerp. Aangezien het onmogelijk is om er in $tring$ veel aandacht aan te besteden is dit boekje ontstaan.

Dit boekje bevat 6 delen :

i] De theorie.

ii] Oefeningen op de theorie.

iii] De oplossingen van de oefeningen.

iv] Een woordenlijst.

v] Een paar hulpprogramma's in BASIC.

vi] Een overzichtstabel.

Indien u na het lezen van deze cursus nog problemen heeft dan kunt u steeds met mij contact opnemen:

Koen Lefever,

Lindenstraat 118

1801 Vilvoorde.

i I De theorie

I. SYMBOLISCHE LOGICA.

a) Beweringen.

De zin of uitspraak 'Het regent.' is een bewering die op een bepaalde plaats op een bepaald ogenblik waar of onwaar is. Indien men van een uitspraak niet kan zeggen of ze waar is of niet, dan is ze a priori geen bewering. Ook vragen zijn geen bewering omdat ze noch waar, noch onwaar zijn. (Het antwoord kan wel waar of onwaar zijn.) [ii.1]
In de symbolische logica scrijven we deze beweringen in een waardetafel of waarheidstabel waarin een 1 een ware en een 0 een onware bewering voorstelt:

p	
	De bewering 'Het regent.'
1	is waar: het regent inderdaad.
0	is onwaar: het regent niet.

Als we nu een bewering q nemen met q='De hond blaft niet.' dan is deze bewering onwaar als de hond wel blaft en waar als de hond niet blaft.

b) Samenstellingen.

1) De negatie of ontkenning: symbool $\overline{p}$ of $\neg p$.
Indien ik beweer $\neg p$ (lees: niet-p) dan zeg ik:'Het is niet waar dat het regent.' of m.a.w. 'Het regent niet.' $\neg p$ is waar als p onwaar is en omgekeerd.Als we dit in tabelvorm schrijven dan hebben we volgend resultaat: [ii.2]

p	$\neg p$	De negatie of ontkenning.
1	0	Als p waar is dan is $\overline{p}$ onwaar.
0	1	Als p onwaar is dan is $\overline{p}$ waar.

2) Conjunctie: symbool $p \wedge q$, $p.q$ of $p \times q$.
De bewering 'Het regent <u>en</u> de hond blaft niet.' is pas waar als beide leden van de bewering (= p en q) waar zijn.

p	q	p^q
1	0	0
1	1	1
0	0	0
0	1	0

p^q is enkel waar als p en q gelijktijdig waar zijn.

3) Disjunctie: symbool pvq of p+q (lees: p of q).

De bewering 'De hond blaft of het regent' is reeds waar als een van de leden van de bewering waar is.

p	q	pvq
1	0	1
0	1	1
1	1	1
0	0	0

pvq is enkel onwaar als beide beweringen onwaar zijn.

4) Implicatie: symbool: p -> q (lees: q als p).

De implicatie ('De hond blaft niet als het regent') is een ingewikkelder geval. Om implicatie duidelijk te maken moet u begrijpen dat volgende beweringen waar zijn: 'Brussel ligt in Belgie als 11+4=20' en 'Londen ligt in Duitsland als 10 < 7.' Ze zijn zijn waar omdat de voorwaarden of premissen absurd zijn. De bewering 'Als 5+2=7 dan is de Y.C.C. een club van W.C.-fabrikanten' is onwaar omdat uit een juiste voorwaarde een verkeerd gevolg werd getrokken.

p	q	p->q
1	0	0
1	1	1
0	0	1
0	1	1

zie oef. [ii.3]

5) Equivalentie : symbool : p <-> q (lees: p als en slechts als q)

De equivalentie is waar als beide leden tegelijkertijd waar of onwaar zijn. Dit wil zeggen dat bij de bewering 'De hond blaft niet als en en slechts als het regent.' de equivalentie opgaat als:

1- De hond niet blaft en het regent of

2- De hond blaft en het niet regent.

p	q	p<->q
1	0	0
1	1	1
0	0	1
0	1	0

De equivalentie.

[1.9]

c) Meervoudige samenstellingen.

Vanaf nu gaan we de Nederlandse zinnen weglaten en enkel nog met symbolen werken. Als u hiermee moeilijkheden heeft dan kan u in gedachten nog steeds de zinnen invullen.

We kunnen nu samenstellingen maken met 2 beweringen of BOOLEANS. Wat nu als u volgende opdracht moet oplossen : (p->q)^(q->p)<->(p<->q) ??
Hiervoor hebben we een prioriteitensysteem nodig om te weten waarmee we moeten beginnen. Bij gelijkgestelde logische bewerkingen werken we gewoon van links naar rechts. Hieronder volgt het prioriteitensysteem:

( )
¬
+ x
>
< >

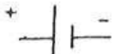
De oplossing is dan :

p	q	p->q	q->p	p<->q	(p->q)^(p->q)	oplossing
1	1	1	1	1	1	1
1	0	0	1	0	0	1
0	1	1	0	0	0	1
0	0	1	[illegible]	1	1	1

Bij de samenstellingen met 2 booleans zijn er steeds 4 rijen omdat alle combinaties van 1 en 0 moeten gevonden worden. Indien er n beweringen zijn, dan zijn er 2^n combinaties. Als we werken met p,q en r dan zijn er 8 (= 2^3) combinaties. Let altijd goed op dat u geen combinaties vergeet of tweemaal schrijft. [ii.5]

II. ELEKTRONICA

In een computer zitten chips die logische functies kunnen uitvoeren: de 'logische poorten'. Hierbij stelt een spanning (meestal 5 Volt) een logische 1 voor. Een logische 0 wordt voorgesteld door de afwezigheid van spanning (= 0 volt.) Omdat de meeste leden leken zijn op het gebied van de elektronica ga ik die chips niet bespreken maar zal ik met eenvoudige schakelingen een en ander trachten duidelijk te maken. U kan eventueel ondestaande schakelingen zelf bouwen. U hoeft hiervoor niet eens te solderen. Hieronder vindt u een onderdelenlijstje :

 : Stroombron (batterij 5 -> 12 Volt)

 : Lampje (6 Volt)

 : Voorschakelweerstand : weerstand die kleiner is dan die van het lampje . De stroom zal bij voorkeur door deze resistor gaan dan door het lampje. Als de stroom maar enigzins kan zal ze hierdoor gaan en niet door het lampje.

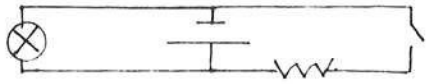
 : Gewone aan/uit schakelaar.

a) Logische 1 & 0

Een gesloten schakelaar en een brandende lamp stellen een losiche 1 voor. Een geopende schakelaar en een gedoofde lamp is een logische 0. Deze schakeling heeft slechts 1 bewering : de schakelaar. De lamp geeft aan of de bewerig waar is of niet.

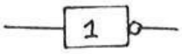
b) Not port : niet-poort , negatie

Bij de NOT brandt de lamp als de schakelaar geopend is en is ze uit als de schakelaar gesloten is. De schakeling is de volgende:

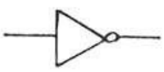


In schema's wordt deze schakeling vervangen door een symbool. Er zijn twee internationaal aanvaarde symbolen: die volgens D.I.N.-norm en die volgens de Amerikaanse normen:

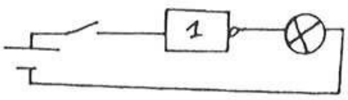
D.I.N.



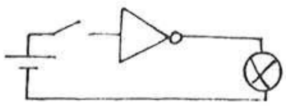
U.S.A.



De schakeling wordt dan:



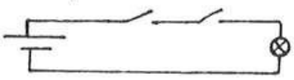
of:



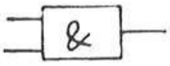
Ook hier is nog steeds 1 bewering.

c) AND-port : logische EN

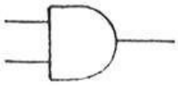
De lamp mag enkel branden als de twee schakelaars tegelijkertijd gesloten zijn.



De symbolen zijn: D.I.N.



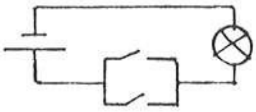
U.S.A.



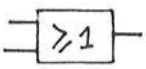
Oefening [ii.6]

d) OR-PORT: logische OF

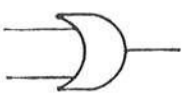
De lamp moet branden als 1 of 2 schakelaars gesloten zijn.



Symbolen: D.I.N.

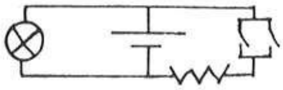


U.S.A.



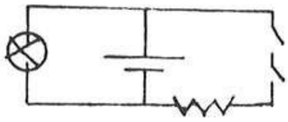
e) NOR-port : NOT OR-port , Niet-of : Het tegengestelde van de OR-port: als een schakelaar dicht is dan wordt de stroom omgeleid.

Symbool: $\overline{p+q}$ of $\neg(p \vee q)$. [ii.7]



f) NAND-port : NOT AND : logische niet-en

Symbool: $\overline{p \times q}$ of $\neg(p \wedge q)$



De symbolen vindt u terug in deel [vi].

g) EX-OR : exclusieve of : symbool: $p \oplus q$

Eigenlijk bestaan er 2 logische OR-ports: de inclusieve en de exclusieve of. Als we niet specifieren of we de IN-OR of de EX-OR bedoelen wordt er steeds de IN-OR bedoeld. Het verschil is dat als zowel de bewering q en p waar zijn de IN-OR opgaat, maar de EX-OR niet. Als ik in het gewoon taalgebruik zeg : 'Ik noem Koen of Lode.' dan bedoel ik meestal de EX-OR: het is niet waarschijnlijk dat ik zowel Koen als Lode zou noemen. In de logica wordt echter de IN-OR bedoeld. De waardetabel van de EX-OR ziet er als volgt uit:

p	q	$p \oplus q$
1	0	1
1	1	0
0	0	0
0	1	1

h) De EX-NOR : Dit is het omgekeerde van de EX-OR. De waardetafel is (niet toevallig) dezelfde als die van de equivalentie.

i) De implicatie : Deze heeft niet veel belang in de elektronica.

j) De zin : Om een voorbeeld te geven van de zin van deze schakelingen geef ik op de volgende pagina een voorbeeld van een binaire rekenmachine die kan optellen. Probeer wat het geeft als je op verschillende ingangen een logische 1 of 0 plaatst. Let wel op: de optellingen gebeuren in het binaire talstelsel:

De '+' stelt in volgende regels geen 'of' voor maar een 'plus' (zoals in 5+3=8 !)

0+0=0

0+1=1

1+0=1

1+1=10

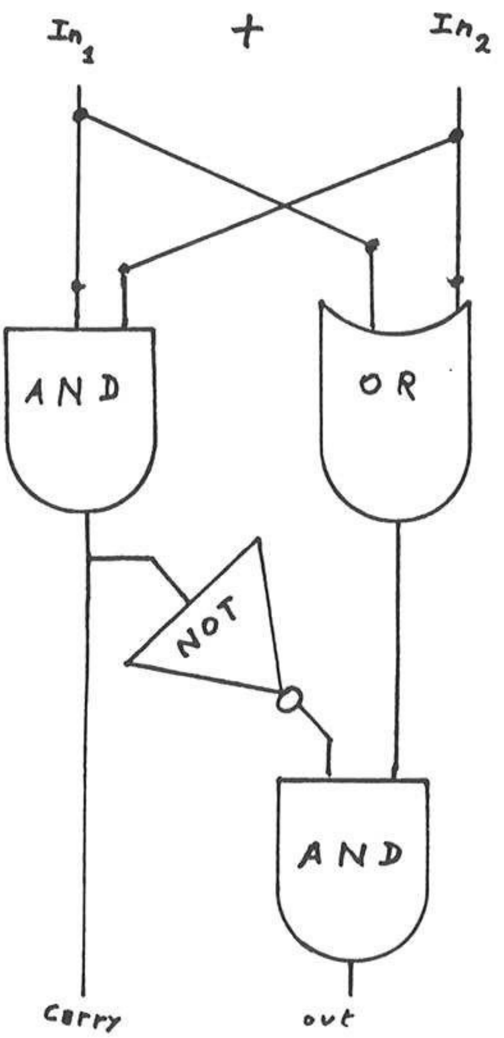
We kunnen 1+1=10 ook schrijven als 1+1=0 met overdracht (carry) 1.

Net als wanneer we schrijven: 14+9=3 met overdracht 2.

Digitale rekenmachines en computers zijn hierop gebaseerd. In een C.P.U. (Centrale Verwerkings Eenheid , microprocessor. Vb.: Z80,6502, 8088,MC 6809,NS 32000,1802 ...) zitten een heleboel logische poorten.

In_1 & In_2 zijn schakelaars.

Carry & Out zijn lampjes.



[il.8]

III. Informatica.

Nu dringt de noodzaak zich op van de kennis van het binaire stelsel. De kleine waarden staan in onderstaande kolom, voor de andere getallen verwijs ik naar het programma in deel [v].

0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000

We nemen best de gewoonte aan achter elk binaire getal een B te schrijven om verwarring te voorkomen.

a)

1) De logische functies zijn zeer nuttig als programmeerhlpjes. Een 8-bit getal heeft als maximale waarde 255 (=11111111B). Zo'n 8-bit getal bestaat uit 2 NIBBLES van elk 4 bit:

vb.: $\underbrace{1011}\ \underbrace{0011}$ B

Stel dat u op dit getal (10110011b) de AND uitvoert met het getal 10001010B, dan moet u dit bit per bit oplossen en bekomt u volgend resultaat:

```
     10110011
     10001010
AND ---------
     10000010
```

Om enkel de hoogste nibble van een getal over te houden (en dus de laagste te 'maskeren') moet u AND 11110000B doen! De cijfers in de M.S.N. (Most Significant Nibble) blijven identiek , die van de L.S.N. (Less Significant Nibble) worden geRESET. [fi.9]

2) Sommige computers (o.a. de CBM-64) kunnen dit ook met decimale getallen. De getallen worden eerst naar het binaire stelsel omgerekend en dan wordt de functie op elke bit uitgevoerd. Het nieuw verkregen getal wordt weer naar het decimale stelsel omgerekend. Met hexadecimalen [v] gebeurt hetzelfde.

b

Ook in IF...THEN-strukturen is logica zinvol.

Vbn.: if A=1 or A=2 then 1000

if A=3 and B=7 then 2000

if not(A=3 and B=8) then 3000

Laten we dit laatste voorbeeld eens nader bekijken.

We kunnen p gelijkstellen aan A=3 en q gelijkstellen aan B=8. Dan krijgen we: if not(p and q) then r waarbij r overeenkomt met goto 3000.

Als we dit nu in symbolen schrijven: if ¬ (p ^ q) then r

'r' wordt enkel uitgevoerd indien ¬ (p ^ q) een logische 1 oplevert.

p	q	p^q	oplossing
1	0	0	1
1	1	1	0
0	1	0	1
0	0	0	1

Het is dus voldoende als A≠ 3 of B≠ 8 om naar 3000 te springen.

[ii.10]

einde deel [i].

ii. I Oefeningen op de theorie

[ii.1] Welke van onderstaande uitspraken zijn beweringen ?

a) Onze kat noemt Tomms.

b) Kan een aap vliegen ?

c) Kom naar hier !

d) X is een hoofstad in Europa.

e) De Aarde is de enige planneet waar leven is ontstaan.

[ii.2] Geef de negatie van volgende beweringen?

a) Belgie ligt in Europa.

b) Brussel ligt niet in Frankrijk.

c) Het is niet waar dat Brussel niet in Europa ligt.

[ii.3] Los op:

vb.: $p=1$

$p \wedge q=0$

$q=0$

a) $p=1$

$p \vee q=$

$q=1$

b) $r=0$

$r \Rightarrow s=$

$s=0$

[ii.4] Los op:

$p=1$ en $q=0$: $p \Leftrightarrow q =$

[ii.5] Vul in:

	p	q	r	$p \vee q$	$\neg(p \vee q)$	$\neg(p \vee q) \leftrightarrow r$
1	1	1	1	1	0	0
2	0	1	1			
3	1	0	1			
4	0	0	1			
5	1	1	0			
6	0	1	0			
7	1	0	0			
8	0	0	0			

[ii.6] Teken de schakeling van de logische EN , maar gebruik de symbolen.

[ii.7] Probeer de waarheidstabel van de NOR-poort op te stellen.

[ii.8] Maak volgende binaire optelling: 1010 + 1001. Schrijf de uitkomst zonder afzonderlijke carry's.

[ii.9] Los op:

a) 10100101B and 10100010B

b) 11010101B and 0B

c) 10101010B exor 11110000B

d) not 10010110 B

[ii.10]

A. Los op:

a) 45 or 15.

b) 10 nor 23.

c) 11 nand 10.

B. Ontleed: if not(A=5 or B=0) then 5000

[illegible] deel 1 [illegible]

[iii.1]

a) Dit is een bewering.

b) Dit is geen bewering.

c) Dit is geen bewering.

d. Dit is een OPEN bewering. Ze gaat op als X de steden Brussel,Londen,Bonn... voorstelt. Als X Peking,Vilvoorde,de beenhouwer... voorstelt, dan gaat ze niet op. De goede X'en behoren tot de OPLOSSINGSVERZAMELING van ...

e) Omdat men hierover niets met zekerheid kan zeggen is dit geen bewering.

[iii.2]

a) Belgie ligt niet in Europa.

b) Brussel ligt in Frankrijk.

c) Brussel ligt niet in Europa.

[iii.3]

a) $p \vee q = 1$

b) $r \to s = 1$

[iii.4] $p \leftrightarrow q = 0$

[iii.5]

	p	q	r	$p \vee q$	$\neg(p \vee q)$	$\neg(p \vee q) \leftrightarrow r$
1	1	1	1	1	0	0
2	0	1	1	1	0	0
3	1	0	1	1	0	0
4	0	0	1	0	1	1
5	1	1	0	1	0	0
6	0	1	0	1	0	0
7	1	0	0	1	0	0
8	0	0	0	0	1	0

[iii.6]

[iii.7]

p	q	pvq	nor
1	0	1	0
1	1	1	0
0	0	0	1
0	1	1	0

[iii.8]

1010

1001

+ ----

10011

[iii.9]

a) 10100101

10100010

and --------

10100000

b) 11010101 and 0 = 0

d) not 10010110 = 01101001

c) 10101010

11110000

exor--------

01011010

[iii.10]

A.a) 65 0100 0001

15 0000 1111

or ---------

79 0100 1111

b) 10 0000 1010

18 0001 1100

nor ————————

225 1110 0001

c) 11 0000 1011

107 0110 1011

nand ————————

244 1111 0100

B. if ¬(p v q) then r

p	q	pvq	oplossing
0	1	1	0
0	0	0	1
1	1	1	0
1	0	1	0

Einde deel Ciii)

iv] Woordenlijst.

Bit: Een logische of binaire '1' of '0'

Byte: 8 bits (cfr. bit)

Carry: overdracht.

C.P.U.: Central Processing Unit. [i.II.j]

Nibble : 4 bit [i.III a]

Inverter: NOT-port [i.II b]

D.I.N.: Duitse Industrie Norm.

Boolean: cfr. bit

Y.C.C.: Youth Computer Club

Digitaal: uitgedrukt in binaire getallen. (Omgekeerde: analoog)

L.S.B.: Less Significant Bit. Vb.: 1001010<u>1</u>

M.S.B.: Most Significant Bit. Vb.: <u>1</u>0010101

Hexadecimaal: 16-talstelsel : kent de cijfers 1,2,3,4,5,6,7,8,9,A,B,C,
D,E & F.

Open bewering: [iii.1]

Einde deel [iv]

BIN <--> DEC

Met dit eenvoudig programma kan u omrekenen van binair naar decimaal & vice versa.

Koen Lefever

```
10 REM * BIN <--> DEC * K.LEFEVER *
20 PRINT:PRINT"Menu:" : PRINT"-----"
30 CLR : REM * GEHEUGENS WISSEN *
40 PRINT"1)      Decimaal --> Binair"
50 PRINT"2)      Binair   --> Decimaal"
60 INPUT "(1/2) ";IN
70 ON IN GOTO 100,200
80 GOTO 60
100 REM * DEC --> BIN *
110 INPUT"DEC=";D
120 IF D=0 THEN PRINT"BIN= 0":END
130 IF D/2= INT(D/2) THEN B$="0"+B$
140 IF D/2<>INT(D/2) THEN B$="1"+B$
150 IF D=1 THEN PRINT"BIN=";B$:RUN
160 D=INT(D/2)
170 GOTO 130
200 REM * BIN --> DEC *
210 INPUT"BIN= ";B$
220 IF B$="0" THEN PRINT"DEC= 0":END
230 L=LEN(B$)
240 FOR Q=0 TO L
250 IF MID$(B$,L-Q,1)="1" THEN D=D+INT(2^Q)
260 NEXT Q
270 PRINT "DEC=";D
280 RUN
```

Hexadecimaal omrekening.

De hexadecimale cijfers zijn: 0,1,2,3,4,5,6,7,8,9,A, B,C,D,E,F

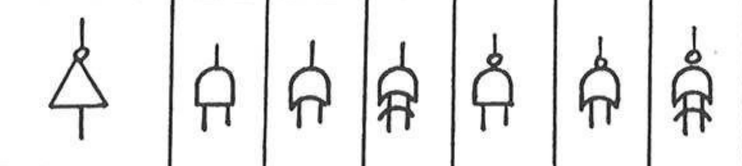
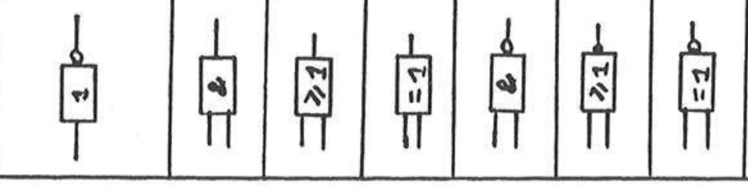
```
1 REM * HEX   -->   DEC *
5 REM * K.Lefever * YCC *
10 INPUT"HEX= ";A$
20 B=LEN(A$)
30 FOR C=0 TO B
40 D$=MID$(A$,B-C,1)
50 IF D$="A" THEN D$="10"
60 IF D$="B" THEN D$="11"
70 IF D$="C" THEN D$="12"
80 IF D$="D" THEN D$="13"
90 IF D$="E" THEN D$="14"
100 IF D$="F" THEN D$="15"
110 D=VAL(D$)
120 E=E+INT(D*(16^C))
130 NEXT C
140 PRINT"DEC=";E
150 RUN
```

```
1 REM * DEC   -->   HEX *
5 REM * K.Lefever * YCC *
10 PRINT
20 INPUT"DEC= ";A
30 B=A/16
40 C=INT(B)
50 D=16*C
60 E=A-D
70 E$=STR$(E)
80 IF E$="10" THEN E$="A"
90 IF E$="11" THEN E$="B"
100 IF E$="12" THEN E$="C"
110 IF E$="13" THEN E$="D"
120 IF E$="14" THEN E$="E"
130 IF E$="15" THEN E$="F"
140 F$=E$+F$
150 A=C
160 IF C=0 THEN PRINT"HEX= ";F$ : RUN
170 GOTO 30
```

Booleaanse algebra (deel 4)

Dit is het slot van de eerste Y.C.C.-cursus. Deze overzichtstabel is een samenvatting van de hele cursus.

K.Lefever

INPUT		NOT A	NOT B	AND	OR	EX-OR	NAND	NOR	EX-NOR	
A	B	$\overline{A}$	$\overline{B}$	A.B	A+B	A⊕B	$\overline{A.B}$	$\overline{A+B}$	$\overline{A⊕B}$	
p	q	¬p	¬q	p^q	p∨q		¬(p^q)	¬(p∨q)	p↔q	⇒ ⇔
0	0	1	1	0	0	0	1	1	1	1
0	1	1	0	0	1	1	1	0	0	1
1	0	0	1	0	1	1	1	0	0	1
1	1	0	0	1	1	0	0	0	1	1
Ndl.		NIET		EN	OF	EX-OF	NEN	NOF	EX-NOF	
Naam:		negatie		conj.	disj.				equiv.	tautologie
U.S.A.										
D.I.N.										

Afkortingen:

conj. : conjunctie

disj. : disjunctie

equiv. : equivalentie

EX : exclusief (symbool: ⊕)

Ndl.: Nederlands

D.I.N.: Duitse Industrie Norm