Sistemas Digitais I

LESI :: 2º ano

- Álgebra de Boole –

António Joaquim Esteves
João Miguel Fernandes

www.di.uminho.pt/~aje

Bibliografia: secções 3.1 e 4.1, DDPP, Wakerly

DEP. DE INFORMÁTICA
ESCOLA DE ENGENHARIA
UNIVERSIDADE DO MINHO

2. Álgebra de Boole

- Sumário -

- Sinais binários
- Sistemas combinacionais vs. sequenciais
- Portas lógicas
- Álgebra da *comutação (switching)*
- Axiomas
- Teoremas
- Dualidade
- Representação normalizada
- Exemplos

2. Álgebra de Boole

- Introdução -

- O sucesso da tecnologia dos computadores baseia-se em 1º lugar na simplicidade com que se projectam circuitos digitais e na facilidade da sua produção.
- Os circuitos digitais são constituídos por unidades de processamento elementares, designadas por **portas lógicas**, e unidades de memória elementares, designadas por *flip-flops*.
- A simplicidade do projecto de circuitos digitais deve-se ao facto de as entradas e as saídas de cada porta lógica ou *flip-flop* assumir apenas 2 valores: 0 e 1.
- As alterações no valor dos sinais são determinadas pelas leis da **álgebra de Boole**.
- A álgebra de Boole permite **optimizar** funções.
- No projecto de circuitos digitais pode usar-se **técnicas de optimização de outras áreas**.

2. Álgebra de Boole

- Sinais Binários (1) -

- A lógica digital esconde a realidade analógica, ao mapear uma gama infinita de valores reais em apenas 2 valores: 0 e 1.
- A um valor lógico, 0 ou 1, é comum chamar-se um **dígito binário (bit)**.
- Com *n* bits, pode representar-se 2^n entidades distintas.
- Quando um projectista lida com circuitos electrónicos, é comum usar os termos "BAIXO" e "ALTO", em vez de "0" e "1".
- Considerar que 0 é BAIXO e 1 é ALTO, corresponde a usar **lógica positiva**. A correspondência oposta a esta, é designada de **lógica negativa**.

2. Álgebra de Boole

- Sinais Binários (2) -

Technology	State Representing Bit	
	0	1
Pneumatic logic	Fluid at low pressure	Fluid at high pressure
Relay logic	Circuit open	Circuit closed
Complementary metal-oxide semiconductor (CMOS) logic	0–1.5 V	3.5–5.0 V
Transistor-transistor logic (TTL)	0–0.8 V	2.0–5.0 V
Fiber optics	Light off	Light on
Dynamic memory	Capacitor discharged	Capacitor charged
Nonvolatile, erasable memory	Electrons trapped	Electrons released
Bipolar read-only memory	Fuse blown	Fuse intact
Bubble memory	No magnetic bubble	Bubble present
Magnetic tape or disk	Flux direction "north"	Flux direction "south"
Polymer memory	Molecule in state A	Molecule in state B
Read-only compact disc	No pit	Pit
Rewriteable compact disc	Dye in crystalline state	Dye in noncrystalline state

2. Álgebra de Boole

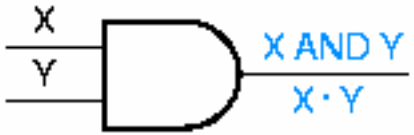
- Sistemas Combinacionais vs. Sequenciais -

- Um sistema lógico **combinacional** é aquele em que as saídas dependem apenas do valor actual das entradas.
- Um sistema combinacional pode ser descrito por uma **tabela de verdade**.
- Além do valor actual das entradas, as saídas dum circuito lógico **sequencial** dependem também da sequência de valores por que passaram as entradas → memória.
- Um sistema sequencial pode ser descrito através duma **tabela de estados**.
- Um sistema combinacional pode conter qualquer número de portas lógicas mas não ciclos de realimentação (*feedback loops*).
- Um ciclo de realimentação é um caminho dum circuito, que permite a um sinal de saída duma porta ser propagado de volta para a entrada dessa porta.
- Regra geral, os ciclos de realimentação introduzem um comportamento sequencial nos circuitos.

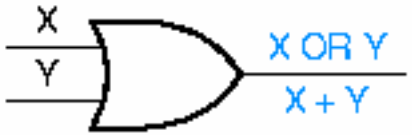
2. Álgebra de Boole

- Portas Lógicas (1) -

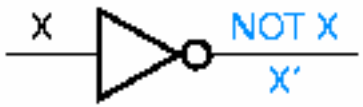
- Com 3 tipos de porta elementares (AND, OR, NOT) consegue construir-se qualquer sistema digital combinacional → formam um **conjunto completo**.



X	Y	X AND Y
0	0	0
0	1	0
1	0	0
1	1	1



X	Y	X OR Y
0	0	0
0	1	1
1	0	1
1	1	1



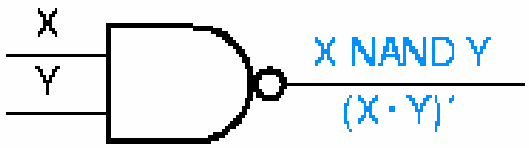
X	NOT X
0	1
1	0

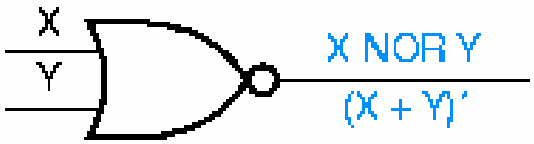
- Os símbolos e as tabelas de verdade do AND e do OR podem ser generalizados para portas com qualquer número de entradas.
- A <u>bolha</u> na saída do inversor representa um comportamento "invertido".

2. Álgebra de Boole

- Portas Lógicas (2) -

- Combinando numa única porta, um NOT com uma função AND ou OR, obtêm-se 2 novas funções lógicas: NAND e NOR.





X	Y	X NAND Y
0	0	1
0	1	1
1	0	1
1	1	0

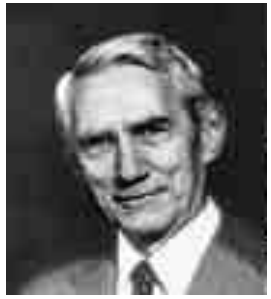
X	Y	X NOR Y
0	0	1
0	1	0
1	0	0
1	1	0

- O símbolo e a tabela de verdade do NAND e do NOR também podem ser generalizados para portas com qualquer número de entradas.

2. Álgebra de Boole

- Álgebra da comutação -

- Em 1854, G. Boole [1815-1865] introduziu o formalismo que ainda usamos para tratar a lógica de forma sistemática → **álgebra de Boole.**

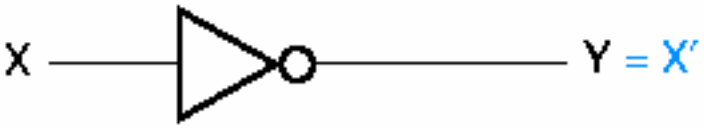


- Em 1938, C. Shannon [1916-2001] utilizou esta álgebra para provar que as propriedades dos circuitos de comutação eléctricos podem ser representados por uma álgebra de Boole com 2-valores → **álgebra da comutação.**
- Usando esta álgebra, pode equacionar-se proposicões *(afirmações)* que serão verdadeiras ou falsas. Combinando-as, geram-se novas proposições e pode concluir-se se elas são verdadeiras ou falsas.
- Shannon usava uma variável simbólica *(por ex. X)* para representar a condição associada a um sinal lógico, em que ele assumia um de 2 valores possíveis *("0" ou "1")*.

2. Álgebra de Boole

- Axiomas (1) -

- Os **axiomas** *(ou postulados)* dum sistema matemático são um conjunto mínimo de definições elementares, que se considera serem verdadeiras.
- O 1º par de axiomas incorpora a **abstracção digital** *(X só pode assumir 2 valores)*:
  (A1) $X=0$ se $X \neq 1$ (A1') $X=1$ se $X \neq 0$
- Este par de axiomas apenas difere na permuta dos símbolos 0 e 1.
- Este princípio aplica-se a todos os axiomas e está na origem da **dualidade**.
- O próximo par de axiomas incorpora a notação de função **inversor**:
  (A2) Se $X=0$, então $X'=1$ (A2') Se $X=1$, então $X'=0$



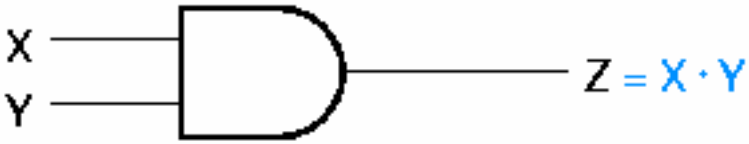
- A plica (') denota a função inversor.

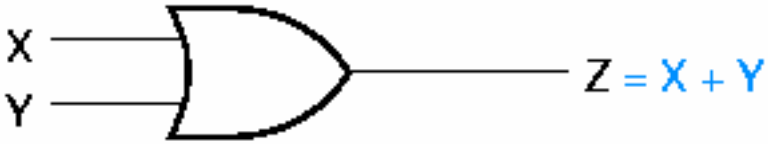
2. Álgebra de Boole

- Axiomas (2) -

- Os últimos 3 pares de axiomas enunciam a definição formal das operações AND *(multiplicação lógica)* e OR *(adição lógica)*:

(A3)	$0 \cdot 0 = 0$	(A3')	$1+1 = 1$
(A4)	$1 \cdot 1 = 1$	(A4')	$0+0 = 0$
(A5)	$0 \cdot 1 = 1 \cdot 0 = 0$	(A5')	$1+0 = 0+1 = 1$





- Por convenção, numa expressão lógica contendo multiplicação e adição, a multiplicação tem **precedência**.
- A expressão $X \cdot Y + Y \cdot Z'$ é equivalente a $(X \cdot Y) + (Y \cdot Z')$.
- Os axiomas A1-A5 e A1'-A5' definem de forma completa a **álgebra Boole**.

2. Álgebra de Boole

- Teoremas (1) -

- Os teoremas são declarações, que se sabe serem verdadeiras, que permitem manipular expressões algébricas de modo a que a análise seja mais simples e/ou a síntese dos circuitos correspondentes seja mais eficiente.

- Teoremas envolvendo apenas uma variável:

(T1)	$X+0 = X$	(T1')	$X \cdot 1 = X$	(Identidades)
(T2)	$X+1 = 1$	(T2')	$X \cdot 0 = 0$	(Elementos nulos)
(T3)	$X+X = X$	(T3')	$X \cdot X = X$	(Idempotência)
(T4)	$(X')' = X$			(Involução)
(T5)	$X+X' = 1$	(T5')	$X \cdot X' = 0$	(Complementos)

- Pode provar-se que estes teoremas são verdadeiros. Prova de T1:
  [X=0] 0+0=0 (verdade, segundo A4')
  [X=1] 1+0=1 (verdade, segundo A5')

2. Álgebra de Boole

- Teoremas (2) -

- Teoremas envolvendo 2 ou 3 variáveis:

(T6)	$X+Y = Y+X$	(T6')	$X \cdot Y = Y \cdot X$	(Comutatividade)
(T7)	$(X+Y)+Z = X+(Y+Z)$	(T7')	$(X \cdot Y) \cdot Z = X \cdot (Y \cdot Z)$	(Associatividade)
(T8)	$X \cdot Y+X \cdot Z = X \cdot (Y+Z)$	(T8')	$(X+Y) \cdot (X+Z) = X+Y \cdot Z$	(Distributividade)
(T9)	$X+X \cdot Y = X$	(T9')	$X \cdot (X+Y) = X$	(Cobertura)
(T10)	$X \cdot Y+X \cdot Y' = X$	(T10')	$(X+Y) \cdot (X+Y') = X$	(Combinação)
(T11)	$X \cdot Y+X' \cdot Z+Y \cdot Z = X \cdot Y+X' \cdot Z$			(Consenso)
(T11')	$(X+Y) \cdot (X'+Z) \cdot (Y+Z) = (X+Y) \cdot (X'+Z)$			

- Atenção: o teorema T8' não é verdadeiro com inteiros ou reais.
- T9 e T10 são usados para minimizar funções lógicas.
- Demonstrar T9: T1' → T8 → T2 → T1'

2. Álgebra de Boole

- Teoremas (3) -

- Vários teoremas importantes são verdadeiros para um nº arbitrário de variáveis

- Teoremas envolvendo *n* variáveis:

(T12) $X+X+ \ldots +X = X$ — Idempotência generalizada

(T12') $X \cdot X \cdot \ldots \cdot X = X$

(T13) $(X_1 \cdot X_2 \cdot \ldots \cdot X_n)' = X_1'+X_2'+ \ldots +X_n'$, — Teoremas de DeMorgan

(T13') $(X_1+X_2+ \ldots +X_n)' = X_1' \cdot X_2' \cdot \ldots \cdot X_n'$

(T14) $[F(X_1,X_2,\ldots,X_n,0,1,+,\cdot)]' = F(X_1',X_2',\ldots,X_n',1,0,\cdot,+)$ — T. de DeMorgan generalizado

(T15) $F(X_1,X_2,\ldots,X_n) = X_1 \cdot F(1,X_2,\ldots,X_n) + X_1' \cdot F(0,X_2,\ldots,X_n)$ — Teoremas da expansão de Shannon

(T15') $F(X_1,X_2,\ldots,X_n) = [X_1+F(0,X_2,\ldots,X_n)] \cdot [X_1'+F(1,X_2,\ldots,X_n)]$

2. Álgebra de Boole

- Teoremas (4) -

- Teorema de DeMorgan (T13 e T13') para n=2:
  $(X \cdot Y)' = X' + Y'$
  $(X + Y)' = X' \cdot Y'$

- O Teorema de DeMorgan estabelece um procedimento para complementar funções lógicas.

- Pode usar-se o teorema de DeMorgan para converter expressões AND-OR em expressões OR-AND.

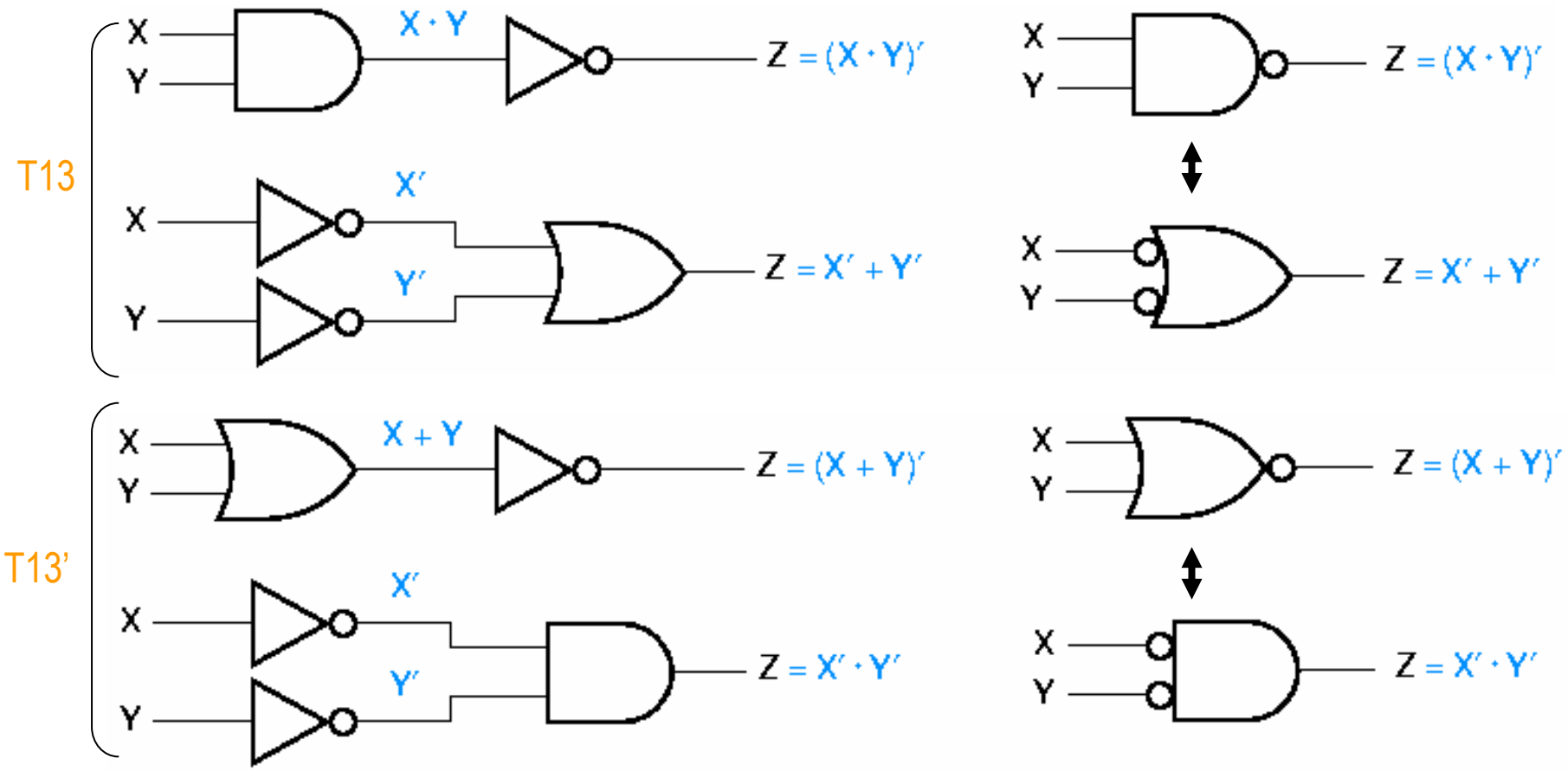
Augustus De Morgan
[1806-1871]

- Exemplo:
  $Z = A' B' C + A' B C + A B' C + A B C'$
  $Z' = (A + B + C') \cdot (A + B' + C') \cdot (A' + B + C') \cdot (A' + B' + C)$

2. Álgebra de Boole

- Teoremas (5) -

Portas lógicas equivalentes segundo os teoremas de DeMorgan:



2. Álgebra de Boole

- Teoremas (6) -

- Como a álgebra de Boole só possui 2 valores, também se pode demonstrar a validade dos teoremas através de **tabelas de verdade**.
- Para isso, constrói-se uma tabela de verdade para cada lado das equações presentes num teorema.

- Para verificar a validade dum teorema, basta que ambos os lados das equações produzam resultados idênticos em todas as combinações de valores das variáveis.

X	Y	$\overline{X}$	$\overline{Y}$	$\overline{X+Y}$	$\overline{X}\bullet\overline{Y}$
0	0	1	1	1	1
0	1	1	0	0	0
1	0	0	1	0	0
1	1	0	0	0	0

- Exemplo: provar os teoremas de DeMorgan [T13 e T13'] para n=2:

  $(X+Y)' = X' \cdot Y'$ $\qquad$ $(X \cdot Y)' = X'+Y'$

X	Y	$\overline{X}$	$\overline{Y}$	$\overline{X\bullet Y}$	$\overline{X}+\overline{Y}$
0	0	1	1	1	1
0	1	1	0	1	1
1	0	0	1	1	1
1	1	0	0	0	0

2. Álgebra de Boole

- Dualidade (1) -

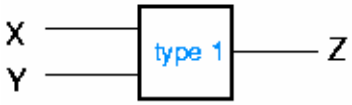
- Os teoremas foram apresentados aos pares.
- A versão primária dum teorema pode ser obtida da versão secundária trocando “0” com “1” e “ ·” com “+”.
- **Princípio da dualidade**: Qualquer teorema ou identidade da Álgebra de Boole continua a ser verdadeiro quando se trocam todos os “0” com “1” e todos os “ ·” com “+”.
- A dualidade é importante porque duplica a utilidade de qualquer axioma/teorema da Álgebra de Boole e da manipulação de funções lógicas.
- O **dual** duma expressão lógica é a mesma expressão em que “+” e “ ·” foram trocados: $F^D(X_1,X_2,...,X_n,0,1,+,\cdot,') = F(X_1,X_2,...,X_n,1,0,\cdot,+,')$.
- Não se deve confundir a dualidade com os teoremas de DeMorgan
  $[F(X_1,X_2,...,X_n,+,\cdot)]' = F(X_1',X_2',...,X_n',\cdot,+)$
  $[F(X_1,X_2,...,X_n,+,\cdot)]' = F^D(X_1',X_2',...,X_n',+,\cdot)$

2. Álgebra de Boole

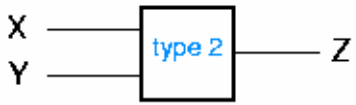
- Dualidade (2) -

Aplicação prática da expressão do dual

Função eléctrica

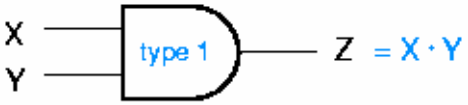


X	Y	Z
LOW	LOW	LOW
LOW	HIGH	LOW
HIGH	LOW	LOW
HIGH	HIGH	HIGH

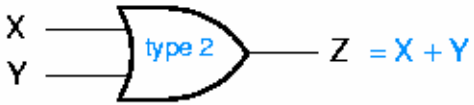


X	Y	Z
LOW	LOW	LOW
LOW	HIGH	HIGH
HIGH	LOW	HIGH
HIGH	HIGH	HIGH

Lógica positiva



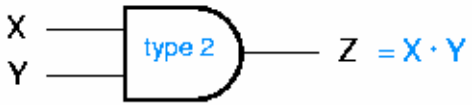
X	Y	Z
0	0	0
0	1	0
1	0	0
1	1	1



X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	1

Lógica negativa

X	Y	Z
1	1	1
1	0	1
0	1	1
0	0	0



X	Y	Z
1	1	1
1	0	0
0	1	0
0	0	0

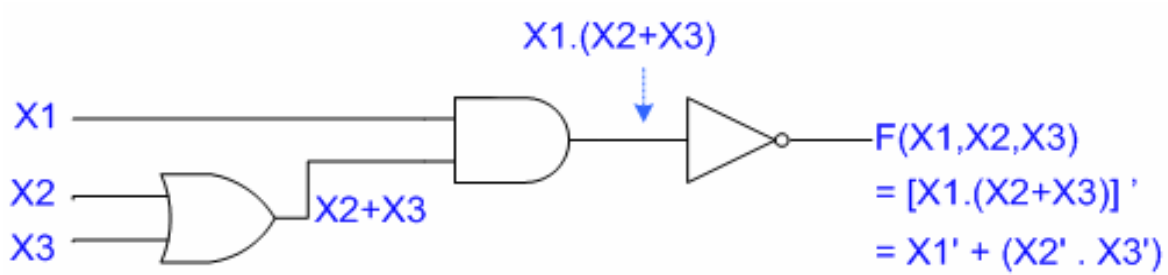
dual

dual

2. Álgebra de Boole

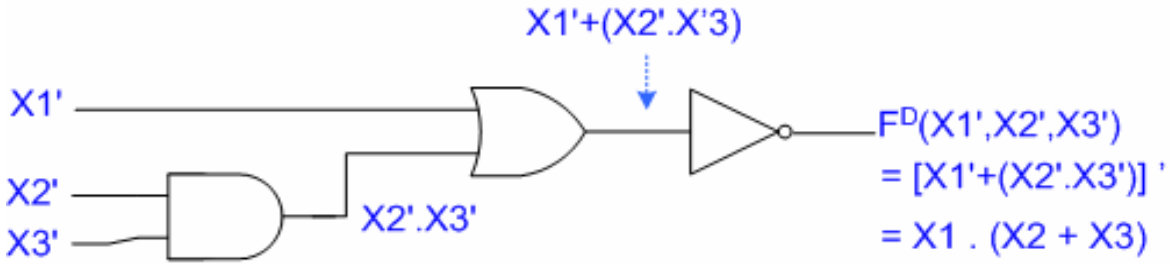
- Dualidade (3) -

- Lógica positiva



X1	X2	X3	F
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

- Lógica negativa



X1'	X2'	X3'	F^D
1	1	1	0
1	1	0	0
1	0	1	0
1	0	0	0
0	1	1	0
0	1	0	1
0	0	1	1
0	0	0	1

- Ilustração prática de [F(X1,X2,...,Xn)]' = F^D(X1',X2',...,Xn')

2. Álgebra de Boole

- Representação normalizada (1) -

- A representação mais elementar duma função lógica é a tabela de verdade.
- A **tabela de verdade** indica qual é a saída do circuito para cada combinação de entradas possível.
- A tabela de verdade duma função de n-variáveis possui 2^n linhas

Row	X	Y	Z	F
0	0	0	0	F(0,0,0)
1	0	0	1	F(0,0,1)
2	0	1	0	F(0,1,0)
3	0	1	1	F(0,1,1)
4	1	0	0	F(1,0,0)
5	1	0	1	F(1,0,1)
6	1	1	0	F(1,1,0)
7	1	1	1	F(1,1,1)

Row	X	Y	Z	F
0	0	0	0	1
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	1
5	1	0	1	0
6	1	1	0	1
7	1	1	1	1

- Existem 2^8 funções lógicas de 3 variáveis ≠ ($2^3 = 8$ linhas e $\{0,1\} = 2$ valores).

2. Álgebra de Boole

- Representação normalizada (2) -

- Como as tabelas de verdade apenas são viáveis com poucas variáveis, é conveniente saber convertê-las para expressões algébricas.
- Um **literal** é uma variável ou o complemento duma variável. EX: X, Y, X'
- Um **termo de produto** é um literal ou um produto lógico de 2 ou mais literais.

  EX: Z' , $W \cdot X \cdot Y$, $W \cdot X' \cdot Y'$
- A **soma-de-produtos** (SOP) é uma soma lógica de termos de produto.
  EX: $Z' + W \cdot X \cdot Y$
- Um **termo de soma** é um literal ou uma soma lógica de 2 ou mais literais.

  EX: Z' , W+X+Y , W+X'+Y'
- O **produto-de-somas** (POS) é um produto lógico de termos de soma.
  EX: $Z' \cdot (W+X+Y)$

2. Álgebra de Boole

- Representação normalizada (3) -

- Um **termo normal** é um termo de produto, ou de soma, em que cada variável só aparece uma vez.
  Exemplos de termos não-normais: $W \cdot X \cdot X' \cdot Z'$, $W'+Y'+Z+W'$
- Um **mintermo** de n-variáveis é um termo de produto normal com **n** literais.

  Exemplos com 4 variáveis: $W \cdot X \cdot Y \cdot Z'$, $W' \cdot X' \cdot Y \cdot Z$
- Um **maxtermo** de n-variáveis é um termo de soma normal com **n** literais.

  Exemplos com 4 variáveis: $W+X+Y+Z'$, $W'+X'+Y+Z$
- Há uma correspondência entre a tabela de verdade e os mintermos e maxtermos.
- Um mintermo é um termo de produto que é 1 numa linha da tabela de verdade.
- Um maxtermo é um termo de soma que é 0 numa linha da tabela de verdade.

2. Álgebra de Boole

- Representação normalizada (4) -

Mintermos e maxtermos para uma função de 3-variáveis F(X,Y,Z)

Row	X	Y	Z	F	***Minterm***	***Maxterm***
0	0	0	0	F(0,0,0)	$X' \cdot Y' \cdot Z'$	$X + Y + Z$
1	0	0	1	F(0,0,1)	$X' \cdot Y' \cdot Z$	$X + Y + Z'$
2	0	1	0	F(0,1,0)	$X' \cdot Y \cdot Z'$	$X + Y' + Z$
3	0	1	1	F(0,1,1)	$X' \cdot Y \cdot Z$	$X + Y' + Z'$
4	1	0	0	F(1,0,0)	$X \cdot Y' \cdot Z'$	$X' + Y + Z$
5	1	0	1	F(1,0,1)	$X \cdot Y' \cdot Z$	$X' + Y + Z'$
6	1	1	0	F(1,1,0)	$X \cdot Y \cdot Z'$	$X' + Y' + Z$
7	1	1	1	F(1,1,1)	$X \cdot Y \cdot Z$	$X' + Y' + Z'$

2. Álgebra de Boole

- Representação normalizada (5) -

- Um mintermo de n-variáveis pode ser representado por um inteiro com n-bits (o <u>número</u> do <u>mintermo</u>).

- No mintermo **i**, uma variável surge complementada se o bit correspondente na representação binária de **i** for 0; senão, a variável é não-complementada

  Por exemplo, à linha 5 (101) corresponde o mintermo $X \cdot Y' \cdot Z$

- No maxtermo **i**, uma variável surge complementada se o bit correspondente na representação binária de **i** for 1; senão, a variável é não-complementada

  Por exemplo, à linha 5 (101) corresponde o maxtermo $X'+Y+Z'$

- Para que a especificação dos mintermos e maxtermos faça sentido, é preciso conhecer o número de variáveis da função e a sua ordem. (X,Y,Z nos exemplos)

2. Álgebra de Boole

- Representação normalizada (6) -

Row	X	Y	Z	F
0	0	0	0	1
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	1
5	1	0	1	0
6	1	1	0	1
7	1	1	1	1

- A partir da correspondência entre a tabela de verdade e os <u>mintermos</u>, pode derivar-se uma representação algébrica dessa função lógica.
- A **soma canónica** duma função lógica é uma soma dos mintermos que correspondem a linhas da tabela de verdade para as quais a função é 1.
- Exemplo - a partir da tabela ao lado obtém-se:
  $F = \sum_{X,Y,Z} m(0,3,4,6,7) = X' \cdot Y' \cdot Z' + X' \cdot Y \cdot Z + X \cdot Y' \cdot Z' + X \cdot Y \cdot Z' + X \cdot Y \cdot Z$

  A notação $\sum_{X,Y,Z} m(0,3,4,6,7)$ identifica 1 <u>lista de mintermos</u> e representa a soma dos mintermos 0, 3, 4, 6 e 7 envolvendo as variáveis X, Y e Z.
- À lista de mintermos também se pode dar o nome de *on-set* da função lógica.

2. Álgebra de Boole

- Representação normalizada (7) -

- A partir da correspondência entre a tabela de verdade e os maxtermos, pode derivar-se uma representação algébrica dessa função lógica.
- O **produto canónico** duma função lógica é um produto dos maxtermos que correspondem a linhas da tabela de verdade para as quais a função é 0.

Row	X	Y	Z	F
0	0	0	0	1
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	1
5	1	0	1	0
6	1	1	0	1
7	1	1	1	1

- Exemplo - a partir da tabela ao lado obtém-se:
  $F = \prod_{X,Y,Z} M(1,2,5) = (X+Y+Z') \cdot (X+Y'+Z) \cdot (X'+Y+Z')$
  A notação $\prod_{X,Y,Z} M(1,2,5)$ identifica uma lista de maxtermos e representa o produto dos maxtermos 1, 2 e 5 envolvendo as variáveis X, Y e Z.
- À lista de maxtermos também se pode dar o nome de *off-set* da função lógica.

2. Álgebra de Boole

- Representação normalizada (8) -

- É fácil converter uma lista de mintermos numa lista de maxtermos.
- Para uma função de n-variáveis, os mintermos e maxtermos pertencem ao conjunto $\{0, 1, \ldots, 2^n-1\}$.
- Uma lista de mintermos ou de maxtermos é um sub-conjunto destes números.
- Para mudar dum tipo de lista para o outro, utiliza-se o sub-conjunto de números complementar.
- Exemplos:

$$\sum_{A,B,C} m(0,1,2,3) = \prod_{A,B,C} M(4,5,6,7)$$
$$\sum_{X,Y} m(1) = \prod_{X,Y} M(0,2,3)$$
$$\sum_{W,X,Y,Z} m(1,2,3,5,8,12,13) = \prod_{W,X,Y,Z} M(0,4,6,7,9,10,11,14,15)$$

2. Álgebra de Boole

- Representação normalizada (9) -

- Foram apresentadas 5 formas distintas de representar funções lógicas combinacionais:
  - A tabela de verdade
  - A soma algébrica de mintermos (a soma canónica)
  - A lista de mintermos, com notação $\sum$
  - O produto algébrico de maxtermos (o produto canónico)
  - A lista de maxtermos, com notação $\prod$
- Qualquer destas representações contém exactamente a mesma informação.
- A partir duma delas, pode derivar-se cada uma das outras 4 aplicando uma regra de conversão simples.

2. Álgebra de Boole

- Exemplos (1) -

- 1. Considere $F = X \cdot Y + X \cdot Y' \cdot Z + X' \cdot Y \cdot Z$. Obtenha a expressão de F' na forma produto de somas

$$
\begin{aligned}
F' &= (X.Y + X.Y'.Z + X'.Y.Z)' \\
&= (X.Y)' \cdot (X.Y'.Z)' \cdot (X'.Y.Z)' \\
&= (X'+Y').(X'+Y+Z').(X+Y'+Z')
\end{aligned}
$$

- 2. Escreva a função $G(X,Y,Z) = X + Y \cdot Z$ como uma lista de mintermos

$$
\begin{aligned}
G &= X + Y \cdot Z \\
&= X \cdot (Y+Y') \cdot (Z+Z') + Y \cdot Z \cdot (X+X') && [T5] \\
&= XYZ + XYZ' + XY'Z + XY'Z' + XYZ + X'YZ \\
&= X'YZ + XY'Z' + XY'Z + XYZ' + XYZ && [T3] \\
&= \sum_{X,Y,Z} m(3,4,5,6,7)
\end{aligned}
$$

2. Álgebra de Boole

- Exemplos (2) -

- 3. Obtenha o produto de maxtermos para a função H = X' ·Y' + X ·Z

```
H = X'Y' + XZ
  = (X'Y'+X)(X'Y'+Z)                [T8'] e+A.B=(e+A).(e+B)
  = (X'+X).(Y'+X).(X'+Z).(Y'+Z)     [T8'] e+A.B=(e+A).(e+B)
  =      1 .(X+Y').(X'+Z).(Y'+Z)    [T5 ] X+X'=1
```

- Em cada soma da expressão anterior falta uma variável:
  - X+Y' = X+Y'+ZZ' = (X+Y'+Z)(X+Y'+Z') [T5'] [T8']
  - X'+Z = X'+Z+YY' = (X'+Y+Z)(X'+Y'+Z) [T5'] [T8']
  - Y'+Z = Y'+Z+XX' = (X+Y'+Z)(X'+Y'+Z) [T5'] [T8']

- Combinando estes termos:

H = (X+Y'+Z)(X+Y'+Z')(X'+Y+Z)(X'+Y'+Z)(X+Y'+Z)(X'+Y'+Z)

= $\prod_{X,Y,Z} M(2,3,4,6)$

2. Álgebra de Boole

- Exemplos (3) -

- 4. Obtenha a lista de maxtermos para H = X' ·Y' + X ·Z, usando a tabela de verdade

X	Y	Z	H
0	0	0	**1**
0	0	1	**1**
0	1	0	**0**
0	1	1	**0**
1	0	0	**0**
1	0	1	**1**
1	1	0	**0**
1	1	1	**1**

- A partir da tabela obtém-se:
  $H = \prod_{X,Y,Z} M(2,3,4,6)$
  $H = \sum_{X,Y,Z} m(0,1,5,7)$
- Compare esta solução com a que se obteve no exemplo 3.

- 5. Obtenha uma expressão para J = XYZ + XYZ' + XY'Z + X'YZ com um número reduzido de operadores

```
J = XYZ + XYZ' + XYZ + XY'Z + XYZ + X'YZ      [T3] X+X=X
  = XY(Z+Z') + X(Y+Y')Z + (X+X')YZ
  = XY+XZ+YZ                                 [T5] X+X'=1
```