Capítulo 2

Técnicas Formais para Representação das Fórmulas Lógicas

Os circuitos lógicos são realizações/implementações de fórmulas lógicas. Neste capítulo apresentamos algumas ferramentas matemáticas .

2.1 Cálculo Proposicional

Estuda as "fórmulas lógicas" obtidas a partir das proposições/variáveis lógicas ou da combinação destas através dos conectivos lógicos ($\vee$, $\wedge$, $\neg$, $\rightarrow$, $\leftrightarrow$).

O conectivo $\leftrightarrow$ estabelece uma relação de **equivalência lógica**.

Definição 2.1 *Duas proposições são ditas* **logicamente equivalentes**, *ou simplesmente* **equivalentes**, *se elas tiverem a mesma tabela-verdade.*

O valor lógico de uma fórmula depende exclusivamente dos valores lógicos de suas variáveis. As **tabelas-verdade** constituem uma forma direta e simples para representar/visualizar esta relação levando em conta todas as possíveis combinações dos valores lógicos (0 e 1) das variáveis. Portanto, para uma função de n variávies deve-se ter 2^n linhas/entradas na tabela-verdade.

Exemplo 2.1 *As tabelas-verdade dos conectivos lógicos:*

Linha	p	q	$p \wedge q$
0	*V*	*V*	*V*
1	*V*	*F*	*F*
2	*F*	*V*	*F*
3	*F*	*F*	*F*

Linha	p	q	$p \vee q$
0	*V*	*V*	*V*
1	*V*	*F*	*V*
2	*F*	*V*	*V*
3	*F*	*F*	*F*

Linha	p	q	$p \rightarrow q$ ($\leftrightarrow \neg p \vee q$)
0	*V*	*V*	*V*
1	*V*	*F*	*F*
2	*F*	*V*	*V*
3	*F*	*F*	*V*

Linha	p	q	$p \leftrightarrow q$
0	*V*	*V*	*V*
1	*V*	*F*	*F*
2	*F*	*V*	*F*
3	*F*	*F*	*V*

Linha	p	$\neg q$
0	*F*	*V*
1	*V*	*F*

As tabelas-verdade para uma fórmula "composta" pode ser obtida seguindo os seguintes passos:

1. Atribuir todas as possívies combinações dos valores lógicos das variáveis.
2. Determinar os valores lógicos das "subfórmulas".
3. Determinar os valores lógicos das combinações das "subfórmulas" e assim progressivamente até obtermos uma única coluna de valores lógicos.

Exemplo 2.2 *Tabelas-verdade de fórmulas mais complexas*

Linha	p	q	$(\neg p) \rightarrow (q \vee p)$		
0	F	F	V	F	F
1	F	V	V	V	V
2	V	F	F	V	V
3	V	V	F	V	V
Passos	1	1	2	3	2

Linha	p	q	r	$(p \vee \neg\, q) \rightarrow (r \leftrightarrow p \wedge q)$				
0	V	V	V	V	F	V	V	V
1	V	V	F	V	F	F	F	V
2	V	F	V	V	V	F	F	F
3	V	F	F	V	V	V	V	F
4	F	V	V	F	F	V	F	F
5	F	V	F	F	F	V	F	F
6	F	F	V	V	V	F	F	F
7	F	F	F	V	V	V	V	F
Passos	1	1	1	3	2	4	3	2

Linha	p	q	r	$p \vee (q \wedge r) \leftrightarrow (p \vee q) \wedge (p \vee r)$					
0	V	V	V	V	V	V	V	V	V
1	V	V	F	V	F	V	V	V	V
2	V	F	V	V	F	V	V	V	V
3	V	F	F	V	F	V	V	V	V
4	F	V	V	V	V	V	V	V	V
5	F	V	F	F	F	V	V	F	F
6	F	F	V	F	F	V	F	F	V
7	F	F	F	F	F	V	F	F	F
Passos	1	1	1	3	4	2	3	2	

2.1.1 Tautologias

Definição 2.2 Tautologia *é uma fórmula que assume o valor lógico Verdadeiro para todas as possíveis combinações das suas variáveis.*

- Identidade
  - $p \vee F \leftrightarrow p$
  - $p \vee V \leftrightarrow V$
  - $p \wedge V \leftrightarrow p$
  - $p \wedge F \leftrightarrow F$

Com uso de tabelas-verdade é fácil mostrar que as fórmulas acimas são tautologias.

Linha	p	F	p∨F ↔ p	
0	V	F	V	V
1	F	F	F	V
Passo	1	1	2	3

Linha	p	V	p∨V ↔ V	
0	V	V	V	V
1	F	V	V	V
Passo	1	1	2	3

Linha	p	V	p∧V ↔ p	
0	V	V	V	V
1	F	V	F	V
Passo	1	1	2	3

Linha	p	F	p∧F ↔ F	
0	V	F	F	V
1	F	F	F	V
Passo	1	1	2	3

- Idempotência
  - $p \wedge p \leftrightarrow p$
  - $p \vee p \leftrightarrow p$

Linha	p	p∧p ↔ p	
0	V	V	V
1	F	F	V
Passo	1	2	3

Linha	p	p∨p ↔ p	
0	V	V	V
1	F	F	V
Passo	1	2	3

- Absorção
  - $p \vee (p \wedge q) \leftrightarrow p$
  - $p \vee (\neg\, p \wedge q) \leftrightarrow p \vee q$
  - $p \wedge (p \vee q) \leftrightarrow p$
  - $p \wedge (\neg\, p \vee q) \leftrightarrow p \wedge q$

Linha	p	q	p ∨ (p ∧ q) ↔ p		
0	V	V	V	V	V
1	V	F	V	F	V
2	F	V	F	F	V
3	F	F	F	F	V
Passo	1	1	3	2	4

Linha	p	q	p ∨ (¬ p ∧ q) ↔ p ∨ q				
0	V	V	V	F	F	V	V
1	V	F	V	F	F	V	V
2	F	V	V	V	V	V	V
3	F	F	F	V	F	V	F
Passo	1	1	4	2	3	5	2

Linha	p	q	p ∧ (p ∨ q) ↔ p		
0	V	V	V	V	V
1	V	F	V	V	V
2	F	V	F	V	V
3	F	F	F	F	V
Passo	1	1	3	2	4

Linha	p	q	p ∧ (¬ p ∨ q) ↔ p ∧ q				
0	V	V	V	V	V	V	V
1	V	F	V	F	F	V	F
2	F	V	V	F	V	V	F
3	F	F	F	F	V	V	F
Passo	1	1	4	2	3	5	2

- Dupla Negação: ¬ ¬ p ↔ p

Linha	p	¬ ¬ p	
0	V	F	V
1	F	V	F
Passo	1	2	3

- Teorema de De Morgan
  - ¬ (p ∨ q) ↔ ¬ p ∧ ¬ q
  - ¬ (p ∧ q) ↔ ¬ p ∨ ¬ q

Linha	p	q	¬ (p ∨ q) ↔ ¬ p ∧ ¬ q					
0	V	V	F	V	V	F	F	F
1	V	F	F	V	V	F	F	V
2	F	V	F	V	V	V	F	F
3	F	F	V	F	V	V	V	V
Passo	1	1	3	2	4	2	3	2

Linha	p	q	¬ (p ∧ q) ↔ ¬ p ∨ ¬ q					
0	V	V	F	V	V	F	F	F
1	V	F	V	F	V	F	V	V
2	F	V	V	F	V	V	V	F
3	F	F	V	F	V	V	V	V
Passo	1	1	3	2	4	2	3	2

- Consenso

  - (p ∧ q) ∨ (¬ p ∧ r) ∨ (q ∧ r) ↔ (p ∧ q) ∨ (¬ p ∧ r)
  - (p ∨ q) ∧ (¬ p ∨ r) ∧ (q ∨ r) ↔ (p ∨ q) ∧ (¬ p ∨ r)

Exemplo 2.3 *Aplicação na síntese de circuitos:*

- *obter ¬ p a partir de p com uso de portas NAND: Pela Idempotência, ¬ p ↔ ¬ (p ∧ p).*
- *obter x ∨ y a partir de x e y usando portas NAND. Pela dupla negação, x ∨ y ↔ ¬ ¬ (x ∨ y). Pelo teorema de Morgan, ¬ ¬ (x ∨ y) ↔ ¬ (¬ x ∧ ¬ y). E finalmente, aplicando a Idempotência, ¬ (¬ x ∧ ¬ y) ↔ ¬ (¬ (x ∧ x) ∧ ¬ (y ∧ y))*

2.1.2 Lógica da Argumentação

Conforme já vimos, a lógica se ocupa das relaçõs de consequência entre as premissas e a conclusão de um **argumento correto**. Um argumento de premissas $P_1, P_2, P_3, ..., P_n$ e de conclusão Q é indicado de forma simbólica por:

$$P_1, P_2, P_3, ..., P_n \rightarrow Q$$

Diz-se que um argumento é **válido** se, e somente se, a conclusão for verdadeira quando as premissas $P_1, P_2, P_3, ..., P_n$ forem verdadeiras.

Para demonstrar ou verificar ou testar se um dado argumento é válido ou não com uso de tabelas-verdade, pode-se proceder do seguinte modo:

1. Constrói-se a tabela-verdade envolvendo todas as variáveis, destacando uma coluna para cada premissa e outra para a conclusão.
2. Verifica-se nas colunas de premissas as linhas em que os valores lógicos são todos V. Se em todas essas linhas o valor lógico relativo à coluna da conclusão for também V, então o argumento é válido.

Exemplo 2.4 *Vamos analisar as alternativas de conclusão do seguinte enunciado: "É sabido que se os produtores rurais investem na produção de café e o mercado internacional está em alta, então a maior parte da produção é exportada. Além disso, se a maior parte da produção é exportada, o ministério é prestigiado. Ora, sabemos que hoje, embora o mercado internacional do café esteja em alta, o ministério está desprestigiado. Os produtores rurais estão ou não investindo na produção?"*

Usando as seguintes variáveis lógicas:

- *a: os produtores investem na produção.*
- *b: o mercado está em alta.*
- *c: a maior parte da produção é exportada.*
- *d: o ministério está prestigiado.*

a	b	c	d	$(a \wedge b)$	$\rightarrow c$	$c \rightarrow d$	$\neg\, d$
V	V	V	V	V	V	V	F
V	V	V	F	V	V	F	V
V	V	F	V	V	F	V	F
V	V	F	F	V	F	V	V
V	F	V	V	F	V	V	F
V	F	V	F	F	V	F	V
V	F	F	V	F	V	V	F
V	F	F	F	F	V	V	V
F	V	V	V	F	V	V	F
F	V	V	F	F	V	F	V
F	V	F	V	F	V	V	F
F	V	F	F	F	V	V	V
F	F	V	V	F	V	V	F
F	F	V	F	F	V	F	V
F	F	F	V	F	V	V	F
F	F	F	F	F	V	V	V

Note-se que somente a 12ª linha tem todas as premissas ($(a \wedge b) \rightarrow c$, $c \rightarrow d$, $\neg\, d$) com valor lógico verdadeiro. Portanto, o argumento testado é válido para $\neg\, a$.

Como um argumento $P_1, P_2, P_3, ..., P_n \rightarrow Q$ é válido se, e somente se, a fórmula $P_1 \wedge P_2 \wedge P_3 \wedge ..., P_n \rightarrow Q$, conhecida como **condicional associada**, for tautológica, então uma outra maneira de verificar se um dado argumento $P_1, P_2, P_3, ..., P_n \rightarrow Q$ é válido ou não, através das tabelas-verdade, é construir a sua condicional associada e verificar se ela é ou não uma tautológica.

2.1.3 Outra Tautologias e Regras de Inferência

É evidente que tabelas-verdade são somente práticas para analisar funções lógicas envolvendo um número pequeno de variáveis. Segundo Wakerly, 10

para estudantes e 4 a 5 para os outros. Quando ultrapassar este limite, técnicas mais "sofisticadas" são requeridas.

Apresentaremos aqui algumas mais usuais regras de inferência. Vale ressaltar que todas as regras de inferência são tautologias.

Modus ponens

$$\begin{array}{l} p \\ p \rightarrow q \\ \hline q \end{array} \qquad \begin{array}{l} p \\ q \\ \hline p \wedge q \end{array}$$

Contrapositiva

$$\begin{array}{l} p \rightarrow q \\ \neg\, q \\ \hline \neg\, p \end{array}$$

Redução por absurdo

$$\begin{array}{l} \neg\, p \rightarrow p \\ \hline p \end{array}$$

$$\begin{array}{l} p \wedge q \\ \hline p \end{array}$$

Prova por casos

$$\begin{array}{l} p \rightarrow q \\ \neg\, p \rightarrow q \\ \hline q \end{array} \qquad \begin{array}{l} p \vee q \\ \neg\, p \\ \hline q \end{array} \qquad \begin{array}{l} p \rightarrow q \\ q \rightarrow r \\ \hline p \rightarrow r \end{array}$$

Exemplo 2.5 *Voltamos ao exemplo 2.4. Procuraremos fazer agora uma dedução formal:*

1:	$a \wedge b \rightarrow c$	*Premissa*
2:	$c \rightarrow d$	*Premissa*
3:	b	*Premissa*
4:	$\neg\, d$	*Premissa*
5:	$\neg\neg\, a$	*Suposição*
6:	$\neg\, d \rightarrow \neg\, c$	*Contraposição de (2)*
7:	$\neg\, c$	*Modus Ponens de (4) e (6)*
8:	$\neg\, c \rightarrow \neg\,(a \wedge b)$	*Contraposição de (1)*
9:	$\neg\,(a \wedge b)$	*Modus Ponens de (7) e (8)*
10:	$\neg\, a \vee \neg\, b$	*Teorema de Morgan*
11:	$\neg\, a$	
12:	$\neg\, a$	*Redução por absurdo de (5) e (11)*

Há um metateorema bastante útil na manipulação e simplificação de fórmulas lógicas, conhecido como **Princípio de Dualidade**.

Definição 2.3 Princípio de Dualidade: *Se uma fórmula é válida, então a sua expressão dual é também válida. A expressão dual é obtida trocando-se OR por AND (e vice-versa), 0 por 1 (e vice-versa) e preservando-se a operação NOT.*

Exemplo 2.6 *A expressão dual de*

$$x \vee (y \wedge z) \leftrightarrow (x \vee y) \wedge (x \vee z)$$

é

$$x \wedge (y \vee z) \leftrightarrow (x \wedge y) \vee (x \wedge z)$$

Com uso de tabelas-verdade é fácil verificar a validade das duas expressões.

Outro metateorema que facilita a manipulação e análise de fórmulas lógicas é o **Princípio de Substituição**.

Definição 2.4 Princípio de Substituição: *Seja $P(p, q, \ldots)$ uma tautologia. Então $P(P_1, P_2, \ldots)$ é também uma tautologia para quaisquer proposições $P_1, P_2, \ldots$*

Exemplo 2.7 *A expressão $(p \wedge \neg q) \vee \neg (p \wedge \neg q)$ é uma tautologia, porque sabemos que a fórmula $P \vee \neg P$ é uma tautologia. No caso, $P = (p \wedge \neg q)$.*

2.2 Álgebra de Boole

Em 1854 o matemático George Boole (`http://www-history.mcs.st-and.ac.uk/~history/Mathematicians/Boole.html`) inventou um sistema algébrico que permite expressar e manipular algebricamente as fórmulas lógicas do Cálculo Proposicional.

Seja $\mathcal{B}$ um conjunto com pelo menos dois elementos distintos (0, elemento neutro para adição e 1, elemento neutro para multiplicação) munido de duas operaçõs binárias, adição (+) e multiplicação (·), e uma operação unária, complemento (′). Então, a quadrúpla

$$(\mathcal{B}, +, \cdot, ')$$

é chamada **álgebra de Boole** se valem os seguintes axiomas:

- Existência única de elementos neutros
  - $a + 0 = a$
  - $a \cdot 1 = a$
- Fechamento
  - $a + b \in \mathcal{B}, \quad \forall a, b.$
  - $a \cdot b \in \mathcal{B}, \quad \forall a, b$

- Comutativa
  - $a + b = b + a$
  - $a \cdot b = b \cdot a$
- Distributiva
  - $a + (b \cdot c) = (a + b) \cdot (a + c)$
  - $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$
- Existência (única) de elementos inversos
  - $a + a' = 1$
  - $a \cdot a' = 0$

Os axiomas da álgebra de Boole são duais, isto é, pode-se obter uma expressão em cada axioma a partir da outra trocando-se apenas + por · (e vice-versa), 0 por 1 (e vice-versa) e preservando-se a operação complemento. Portanto, **o dual de qualquer teorema de uma álgebra de Boole é também um teorema**, uma vez que o dual pode ser provado seguindo os mesmos passos do teorema original utilizando-se as expressões duais em cada passo.

A partir destes axiomas é fácil ver que um conjunto binário $\{0, 1\}$ munido das operações lógicas AND (·), OR (+) e NOT (′) é uma álgebra de Boole, pois estas operações são definidas sobre os elementos deste conjunto na seguinte forma

OR	0	1
0	0	1
1	1	1

AND	0	1
0	0	0
1	0	1

	NOT
0	1
1	0

Ainda mais,

1. temos dois elementos neutros distintos: 0 para OR e 1 para AND,
2. é fácil ver que as operações são comutativas,
3. as operações lógicas AND e OR são distributivas, e
4. 0 OR (NOT(0)) = 1, 1 OR (NOT(1)) = 1, 0 AND (NOT(0)) = 0 e 1 AND (NOT(1)) = 0.

Consequentemente, podemos exprimir uma fórmula lógica como uma função algébrica e manipulá-la como uma equação.

Exemplo 2.8 *Voltamos novamente ao exemplo 2.4. Vamos ver que algebricamente poderemos chegar a mesma conclusão, equacionando as premissas (afirmações que valem logicamente 1)*

- $a \wedge b \rightarrow c \leftrightarrow \neg(a \wedge b) \vee c$: $(a \cdot b)' + c = 1$
- $c \rightarrow d \leftrightarrow \neg c \vee d$: $c' + d = 1$
- b: $b=1$
- $\neg d$: $d' = 1$

Para ficar numa forma mais "tratável", podemos ainda fazer as seguintes substituições:

- $(a \cdot b)' + c = 1$ *por* $(a \cdot b) \cdot ((a \cdot b)' + c) = (a \cdot b) \cdot 1$*, que é equivalente a* $a \cdot b \cdot c = a \cdot b$.
- $c' + d = 1$ *por* $c \cdot (c' + d) = c \cdot 1$*, que é equivalente a* $c \cdot d = c$.
- $d' = 1$ *por* $d = 0$

Temos, portanto, um sistema de equações "binárias":

$$\begin{aligned} a \cdot b \cdot c &= a \cdot b \\ c \cdot d &= c \\ b &= 1 \\ d &= 0 \end{aligned}$$

Por substituição, podemos ver que $c = 0$ *e* $a = 0$.

Através das manipulações algébricas, podemos obter "igualdades" correspondentes às tautologias listadas na seção 2.1.1 (para os interessados nestas manipulações, consulte o livro de Troy Nagle et al.). Tais igualdades poderão ser utilizadas, por exemplo, para simplificar as funções e para derivar expressões equivalentes:

- Idempotência
  - $p + p = p$

  - $p \cdot p = p$
- Absorção
  - $p + p \cdot q = p$
  - $p + p' \cdot q = p + q$
  - $p \cdot (p + q) = p$
  - $p \cdot (p' + q) = p \cdot q$
- Dupla Negação: $p'' = p$
- Teorema de De Morgan
  - $(p_1 + p_2 + \cdots + p_n)' = p_1' \cdot p_2' \cdot \cdots \cdot p_n'$
  - $((p_1 \cdot p_2 \cdot \cdots \cdot p_n)' = p_1' + p_2' + \cdots + p_n'$
- Consenso
  - $(p \cdot q) + (p' \cdot r) + (q \cdot r) = (p \cdot q) + (p' \cdot r)$
  - $(p + q) \cdot (p' + r) \cdot (q + r) = (p + q) \cdot (p' + r)$

Por simplicidade, quando não há ambiguidade na interpretação, omitiremos $\cdot$ para referir à operação de multiplicação.

Exemplo 2.9 *Simplifique as seguintes expressões:*

1.

$$\begin{aligned} w &= (x + x'y)(x + y) + (x + x')(x + y) \\ &= (x + y)(x + y) + 1(x + y) \\ &= (x + y) + (x + y) \\ &= (x + y) \end{aligned}$$

2.

$$\begin{aligned} w &= x'y(z + y'z) + y'z \\ &= x'yz + x'yy'z + y'z \\ &= x'yz + 0 + y'z \end{aligned}$$

$$\begin{aligned} &= x'yz + y'z \\ &= x'z + 0 + y'z \\ &= (x' + y')z \\ &= (xy)'z \end{aligned}$$

3.

$$\begin{aligned} w &= ab + ab'c \\ &= a(b + b'c) \\ &= a(b + c) \end{aligned}$$

4.

$$\begin{aligned} w &= ab + a'cd'e + b'cd' \\ &= ab + bcd'e + a'cd'e + b'cd' \\ &= ab + a'cd'e + cd'e + b'cd' \\ &= ab + cd'e + b'cd' \end{aligned}$$

Exemplo 2.10 *Obtenha uma expressão equivalente que possa ser implementada com uso de portas NAND de duas entradas:*

$$\begin{aligned} w &= x + y = (x + y)'' = ((x + y)')' = \\ &= (x'y')' = ((xx)'(yy)')' \end{aligned}$$

Exemplo 2.11 *Obtenha uma expressão equivalente que possa ser implementada com uso de portas NOR de duas entradas:*

$$\begin{aligned} w &= xy = (xy)'' = ((xy)')' = \\ &= (x' + y')' = ((x + x)' + (y + y)')' \end{aligned}$$

Finalmente, vale ressaltar que se pode ainda mostrar que um conjunto $\mathcal{B}_\backslash$ de sequências de n bits é também uma álgebra de Boole sob adição, multiplicação e complemento, bit a bit.

Exemplo 2.12 *Seja* $\mathcal{C} = \{00, 01, 10, 11\}$. *Se definirmos as seguintes operações sobre eles*

+	00	01	10	11
00	00	01	10	11
01	01	01	11	11
10	10	11	10	11
11	11	11	11	11

·	00	01	10	11
00	00	00	00	00
01	00	01	00	01
10	00	00	10	10
11	00	01	10	11

	NOT
00	11
01	10
10	01
11	00

é fácil verificar agora que o conjunto $\{00, 01, 10, 11\}$ *sob as operações acima definidas é uma álgebra de Boole.*

2.3 Álgebra de Conjuntos

O conceito de classe ou conjunto de objetos é um dos mais fundamentais em toda a Matemática. Podemos pensar num conjunto como sendo uma coleção de elementos que possuem uma determinada propriedade comum. Um **conjunto vazio**, $\emptyset$, é aquele que não possue nenhum elemento. Por outro lado, um **conjunto universo** engloba todos os elementos que satisfazem uma determinada propriedade de interesse.

São definidas as seguintes operações sobre os conjuntos:

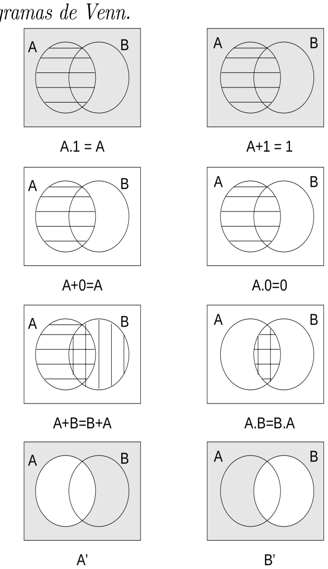
- **união** (+) de dois conjuntos A e B consiste num outro conjunto C de todos os elementos que pertencem a A ou a B ou a ambos.
- **interseção** ($\cdot$) de dois conjuntos A e B consiste num outro conjunto C de elementos que pertencem tanto a A como a B.
- **complemento** ($'$) de um conjunto A em relação ao conjunto universo U consiste num outro conjunto A' de elementos que pertencem a U, mas não pertencem a A.
- **diferença** (-) entre dois conjuntos A e B $(A - B)$ consiste num outro conjunto C de elementos que pertencem a A e não pertencem a B.

Dado um conjunto universo U. Pode-se mostrar que um sistema constituído de conjuntos contidos em U, incluindo o conjunto vazio ($\emptyset$) e o conjunto-universo (U), e munido das operações união (+), interseção ($\cdot$) e complemento ($'$) é uma álgebra de Boole.

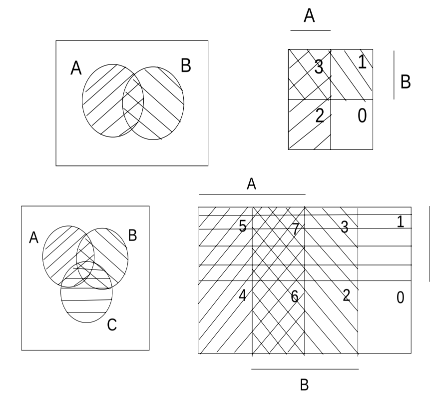
Uma maneira muito ilustrativa de se mostrar as diversas relações e operações entre um pequeno número de conjuntos é utilizar os **diagramas de**

Venn, desenvolvidos pelo matemático John Venn (`http://www-history.mcs.st-and.ac.uk/~history/Mathematicians/Venn.html`). No diagrama de Venn o conjunto-universo U é representado usualmente por um retângulo e os outros subconjuntos por figuras conexas (usualmente, círculos) dentro dele. A área dentro da figura representa o conteúdo de um conjunto e a externa, o conteúdo do seu complemento. Com uso destes diagramas, Venn mostrou as 256 diferentes relações booleanas entre três subconjuntos!

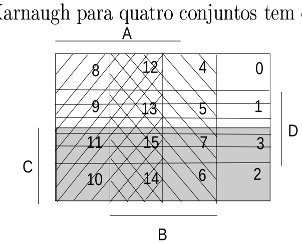
Exemplo 2.13 *Seja um conjunto-universo U e dois conjuntos A e B contidos nele. Vamos verificar a validade de alguns axiomas da álgebra de Boole com uso dos diagramas de Venn.*



Uma forma gráfica alternativa para os diagramas de Venn são os **mapas de Karnaugh** – uma das mais conhecidas ferramentas para minimização das funções lógicas. A seguir é exemplificada a equivalência entre os diagrams de Venn e os mapas de Karnaugh



Um mapa de Karnaugh para quatro conjuntos tem o seguinte aspecto



Para fazer uso dos mapas de Karnaugh/diagramas de Venn na análise ou síntese de uma função lógica, podemos associar uma variável lógica a um conjunto, como ilustra o seguinte exemplo.

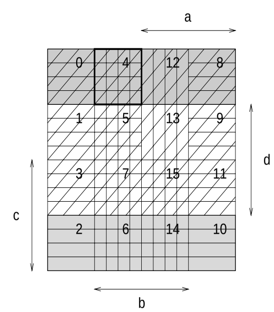
Exemplo 2.14 *Vamos mostrar agora que com uso da noção de conjuntos podemos também resolver o problema proposto no exemplo 2.4. Para tal, associamos a cada variável lógica um conjunto. Temos, então, quatro conjuntos: $\{a\}$, $\{b\}$, $\{c\}$ e $\{d\}$.*

Ainda mais, os quatro "fatos conhecidos" podem ser traduzidas em termos de conjuntos:

- $a \wedge b \rightarrow c \Rightarrow \{\{a\} \cdot \{b\}\}' + \{c\}$
- $c \rightarrow d \Rightarrow \{c\}' + \{d\}$
- $b \Rightarrow \{b\}$

- $\neg\, d \Rightarrow \{d\}\prime$

Através do seguinte mapa de Karnaugh, pode-se inferir que a única região que satisfaz os quatro fatos é a região rotulada de 4, quando a e c não pertencem à região.

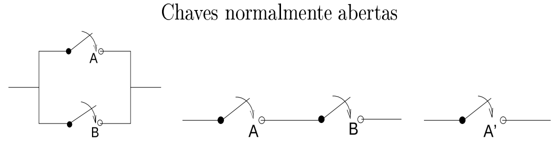


2.4 Álgebra de Chaveamento

Em 1938 Shannon (`http://www-history.mcs.st-and.ac.uk/~history/Mathematicians/Shannon.html`) mostrou como se usa a álgebra booleana para analisar e descrever o comportamento de circuitos lógicos constituídos por relés eletromecânicos (chaves de contato) – componente digital mais popular naquela época. Nesta álgebra, conhecida por **álgebra de chaveamento**, a condição de contato dos relés, aberto e fechado, é representada por uma variável X que pode assumir um dos dois valores, 0 e 1. Hoje em dia, estes dois valores podem corresponder a uma grande variedade de condições físicas – tensão no nível ALTO e BAIXO, luz ligada e apagada, capacitor carregado e descarregado, fusível intato ou aberto, e assim por diante.

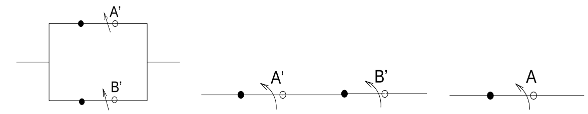
Pode-se associar a cada chave normalmente aberta/fechada uma variável lógica que aparece na forma na qual a chave deve permitir/bloquear a passagem de informação ao ser fechada/aberta.

Chaves normalmente abertas



A + B　　A · B　　A ′

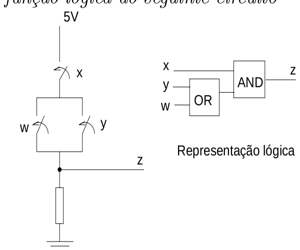
Chaves normalmente fechadas



A + B　　A · B　　A ′

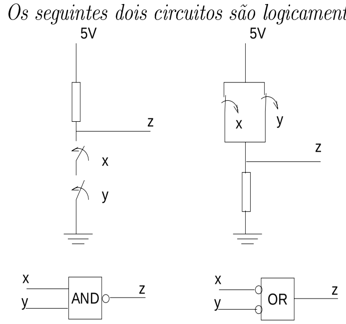
Todas as funções lógicas podem ser implementadas através das combinações entre estas configurações básicas, como ilustra os seguintes exemplos.

Exemplo 2.15 *A função lógica do seguinte circuito*



é equivalente à expressão $z = x \cdot (y + w)$.

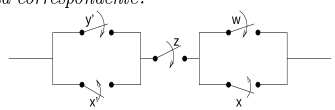
Exemplo 2.16 *Os seguintes dois circuitos são logicamente equivalentes.*

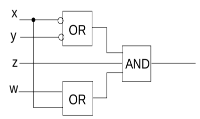


A correspondente função booleana do primeiro circuito é $z = (x \cdot y)\prime$ *e a do segundo é* $z = (x\prime + y\prime)$.

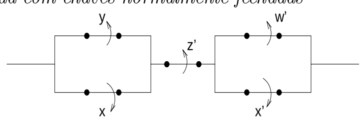
Uma variável pode estar associada a mais de uma chave, se estas chaves operam sempre condicionados aos estados desta variável

Exemplo 2.17 *Seja um circuito de chaves normalmente abertas, qual é a expressão booleana correspondente?*





O circuito só ficará fechado se $(y'+x')z(w+x)$. Esta função pode também ser implementada com chaves normalmente fechadas



2.5 Uma Operação Lógica Útil: XOR

A implementação de uma função lógica com uso das portas AND, OR and NOT é direta, pois elas realizam exatamenta as três operações da álgebra de Boole. Porém, no mercado são oferecidas portas "eletrônicas" que implementem outras funções lógicas mais complexas e comuns nos circuitos lógicos, como a função XOR (ou-exclusivo), cuja tabela-verdade é:

Linha	p	q	p ⊕ q
0	V	V	F
1	V	F	V
2	F	V	V
3	F	F	F

É fácil verificar que

$$\mathrm{p} \oplus \mathrm{q} \leftrightarrow \mathrm{p}' \cdot \mathrm{q} + \mathrm{p} \cdot \mathrm{q}'$$

Algumas relações úteis envolvendo $\oplus$ incluem:

- $\mathrm{a} \oplus \mathrm{a} = 0$
- $\mathrm{a} \oplus \mathrm{a}' = 1$

- $\mathrm{a} \oplus 0 = \mathrm{a}$
- $\mathrm{a} \oplus 1 = \mathrm{a}'$
- $\mathrm{a} \oplus \mathrm{b} = \mathrm{b} \oplus \mathrm{a}$
- $\mathrm{a} \oplus (\mathrm{b} \oplus \mathrm{c}) = (\mathrm{a} \oplus \mathrm{b}) \oplus \mathrm{c}$