SCE 0117 - Introdução à Lógica Digital

Introdução aos circuitos lógicos

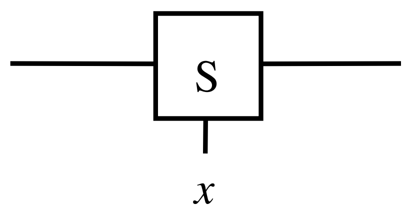
Prof. Vanderlei Bonato

Tópicos da Aula de Hoje

- Variáveis e funções lógicas
- Tabela verdade
- Álgebra Booleana
- Diagrama de Venn
- Processo de síntese

Comportamento de uma chave binária

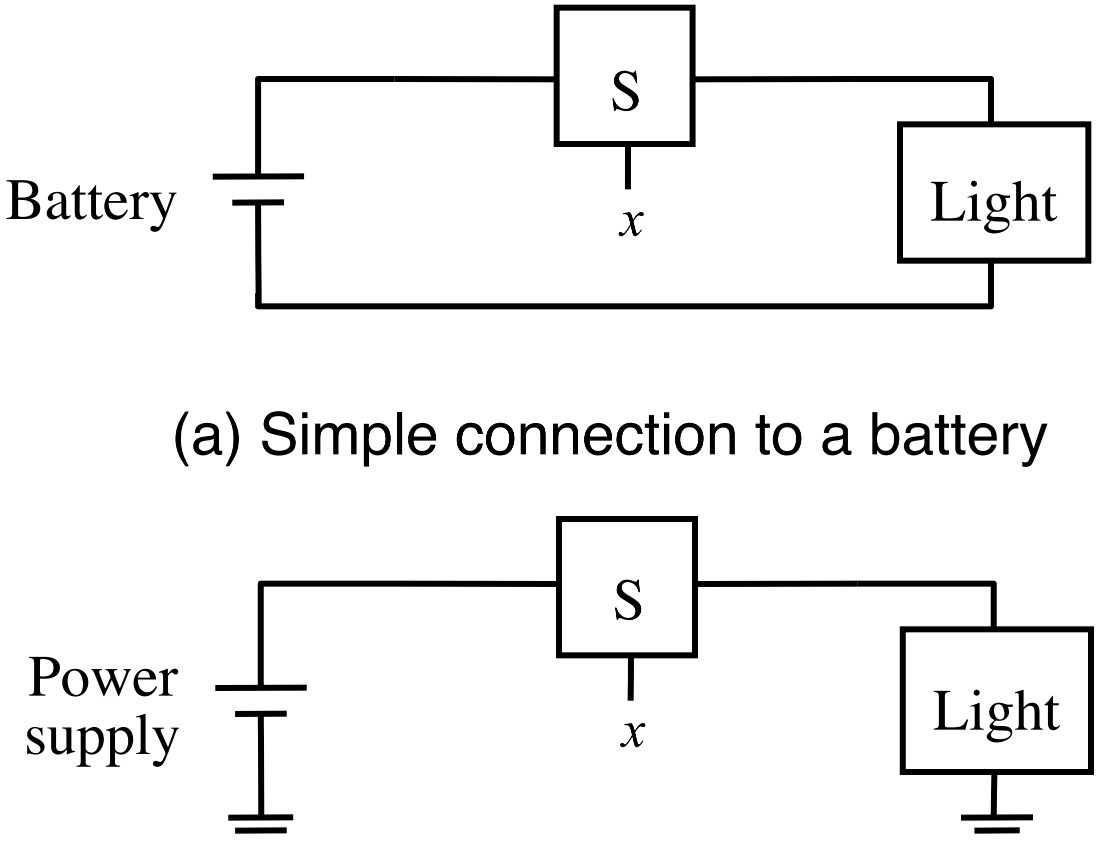
(a) Two states of a switch



(b) Symbol for a switch (x é o controle)

Figure 2.1. A binary switch.

Variáveis e funções



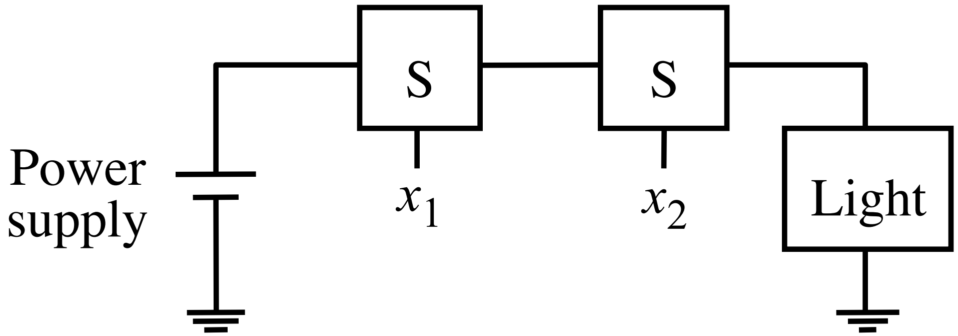
(a) Simple connection to a battery

(b) Using a ground connection as the return path

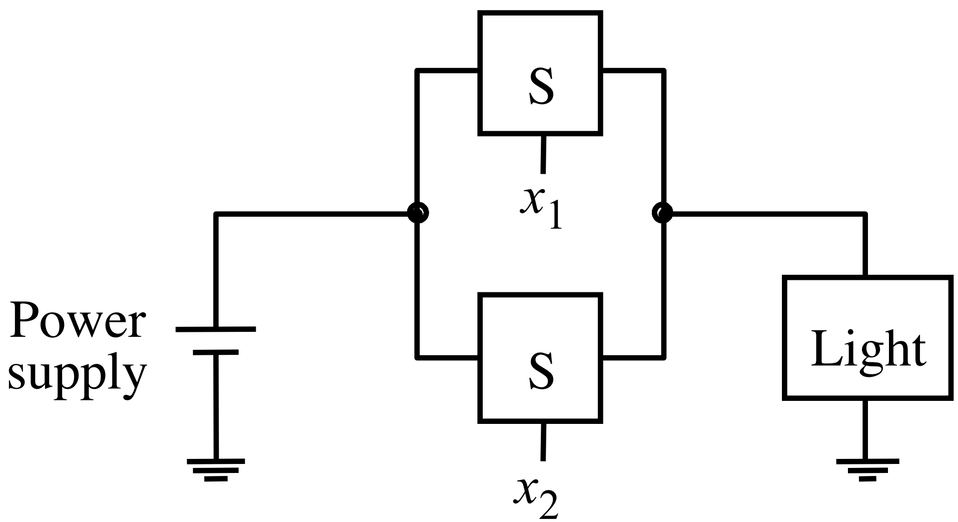
Figure 2.2. A light controlled by a switch.

$L(x) = x$,
onde $L(x)$ é a função lógica e x a variável de entrada

$L=1$ se $x=1$
$L=0$ se $x=0$



(a) The logical AND function (series connection)



AND: $L(x_1, x_2) = x_1 \cdot x_2$

OR: $L(x_1, x_2) = x_1 + x_2$

(b) The logical OR function (parallel connection)

Figure 2.3. Two basic functions.

Conexão serial-paralela

$$L(x_1, x_2, x_3) = (x_1 + x_2) \cdot x_3$$

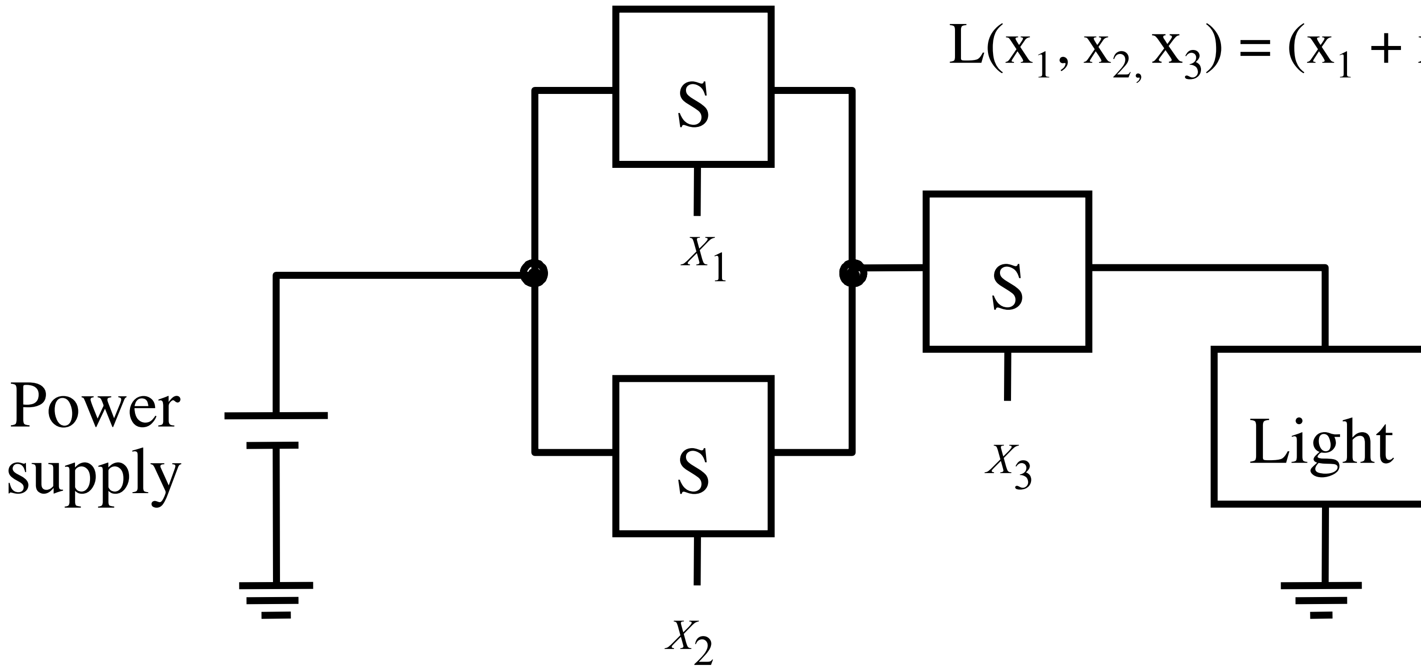


Figure 2.4. A series-parallel connection.

$L(x) = \overline{x}$, também conhecido como inverso, complemento ou operação NOT

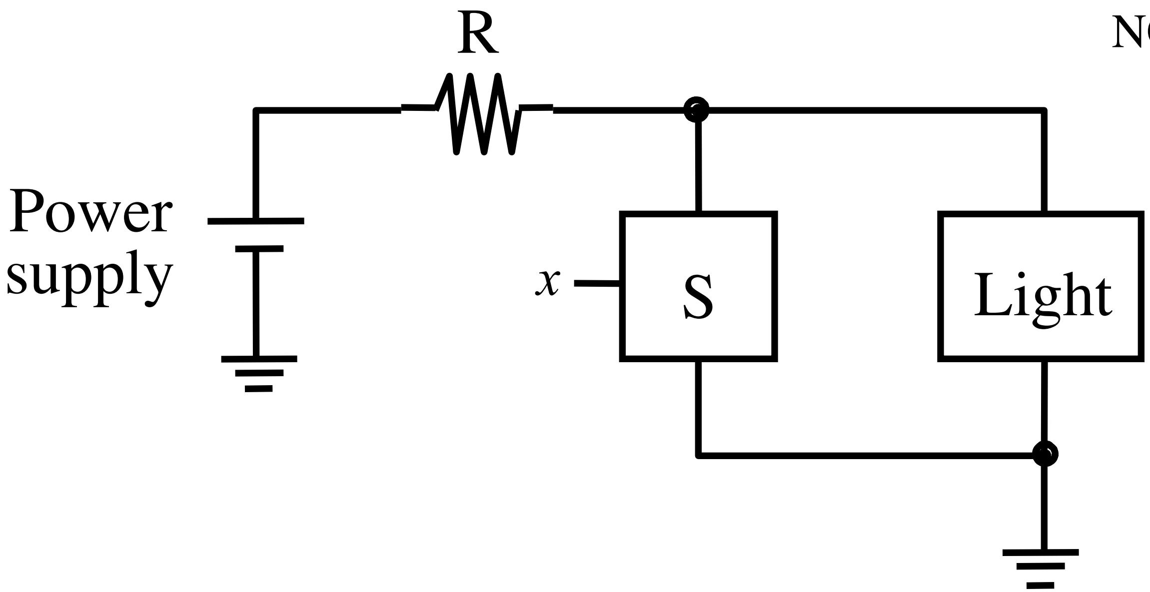


Figure 2.5. An inverting circuit.

Tabela verdade

- Usada para representar operações lógicas
- Na tabela verdade abaixo, as duas primeiras colunas apresentam todas as possíveis combinações de entrada e as colunas restantes a saída das funções AND e OR

x_1	x_2	$x_1 \cdot x_2$	$x_1 + x_2$
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1
		AND	OR

Figure 2.6. A truth table for the AND and OR operations.

Tabela verdade de 3 entradas para AND e OR

x_1	x_2	x_3	$x_1 \cdot x_2 \cdot x_3$	$x_1 + x_2 + x_3$
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	0	1
1	0	0	0	1
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Figure 2.7. Three-input AND and OR operations.

Símbolos gráficos das operações lógicas

- Cada operação lógica pode ser implementada eletronicamente com transistores, resultando em um elemento de circuito conhecido como porta lógica (*logic gate*)
- Um porta lógica pode possuir uma ou mais entradas, mas uma única saída que é ativada ou não em função da entrada
- Circuitos lógicos podem ser construídos gráficamente usando essas representações (esquemático)

Símbolos gráficos das portas lógicas (AND, OR, NOT)

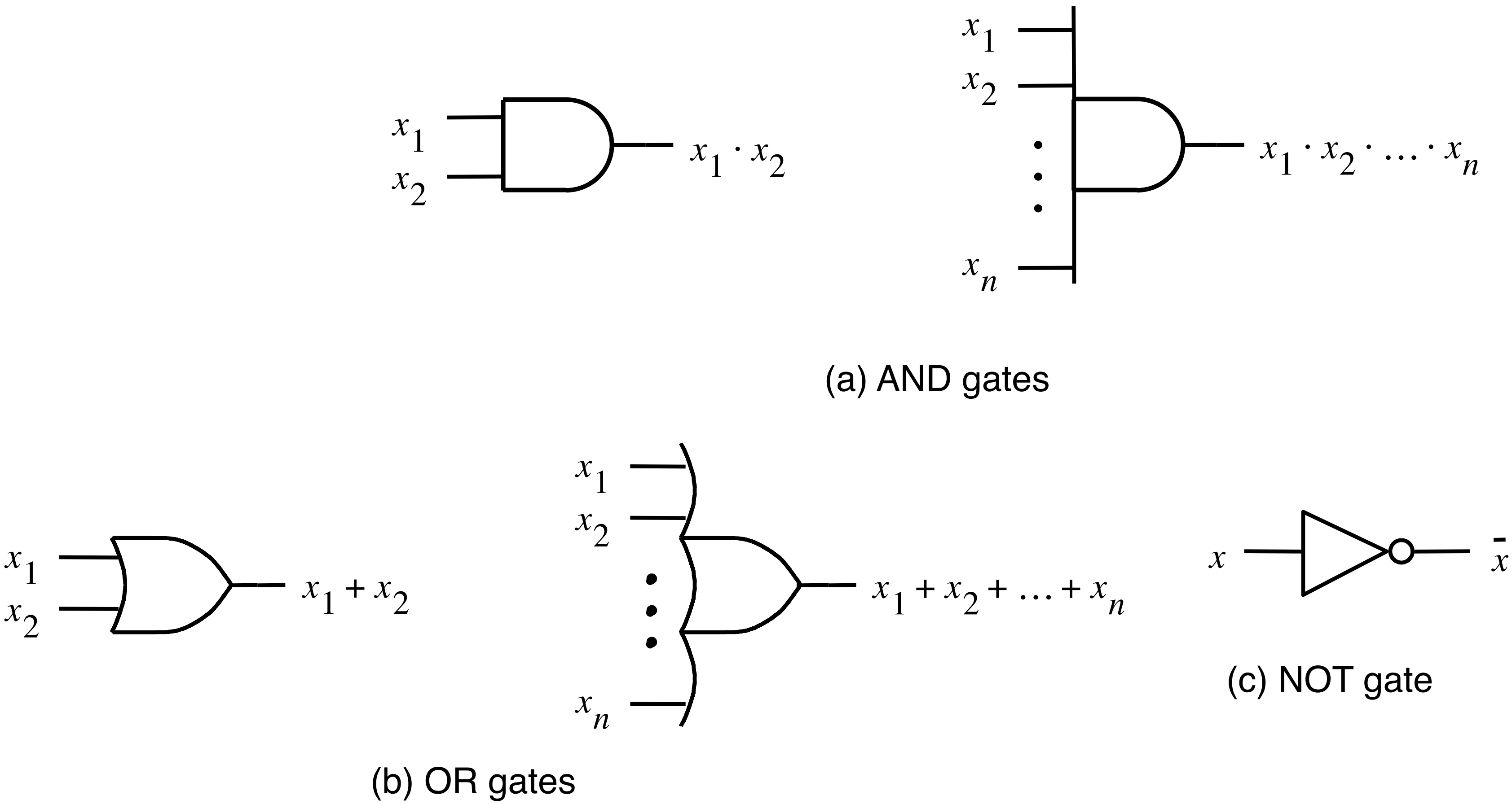


Figure 2.8. The basic gates.

Circuito formado por uma rede de portas lógicas

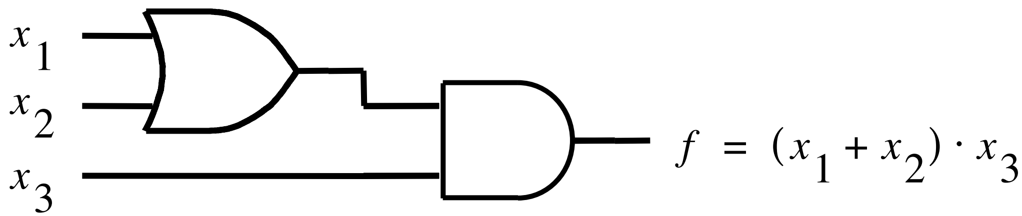


Figure 2.9. The function from Figure 2.4.

Dois modos de representar o mesmo circuito lógico

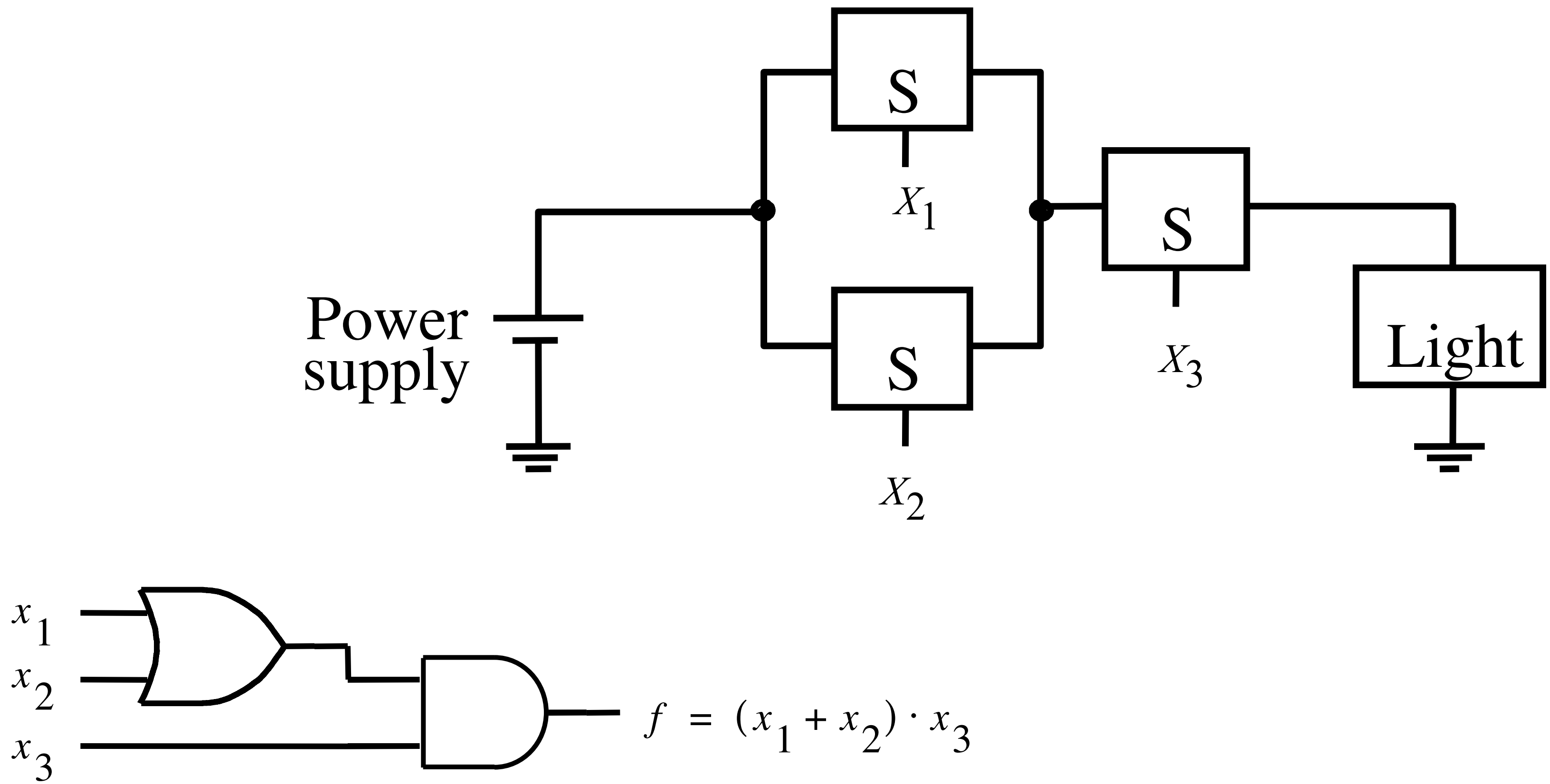
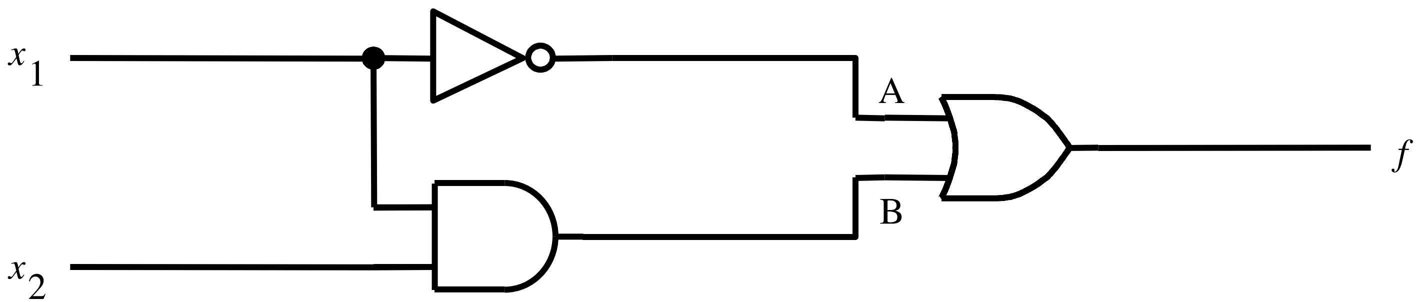


Figure 2.9. The function from Figure 2.4.

Determinar o comportamento de uma função a partir de uma rede de portas lógicas



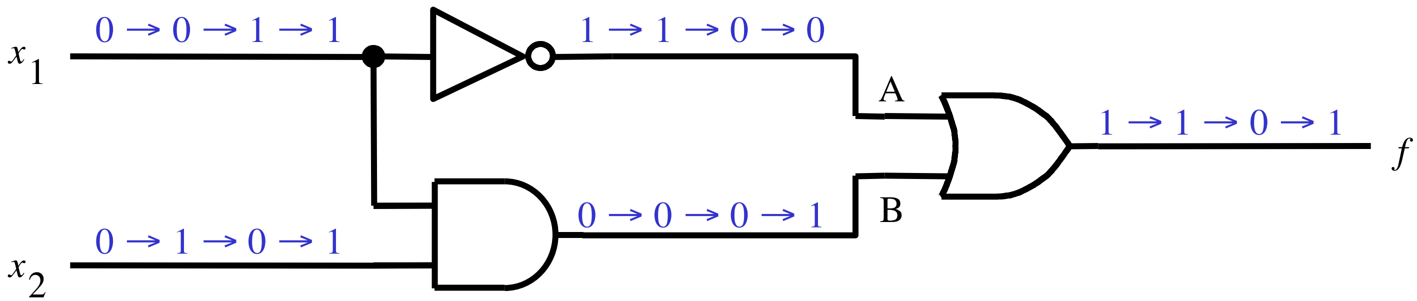
(a) Network that implements $f = \bar{x}_1 + x_1 \cdot x_2$

x_1	x_2	$f(x_1, x_2)$
0	0	1
0	1	1
1	0	0
1	1	1

(b) Truth table for f

Figure 2.10 a Logic network

Processo de Análise:



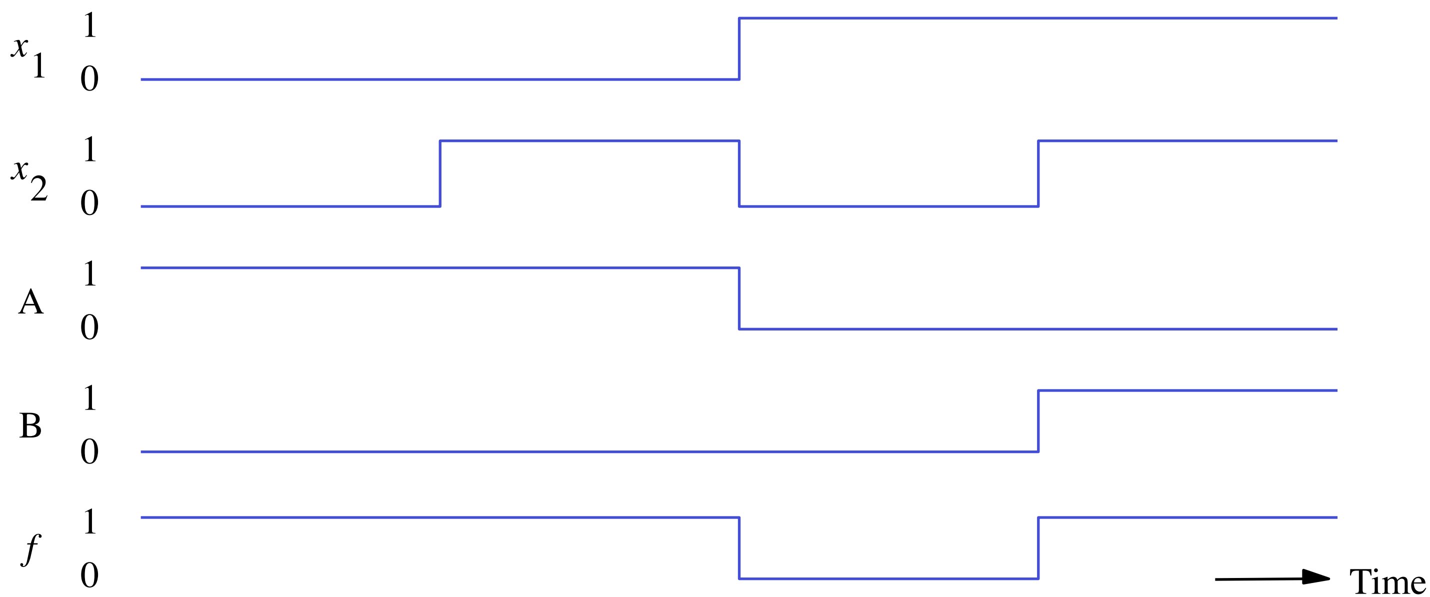
(a) Network that implements $f = \bar{x}_1 + x_1 \cdot x_2$

x_1	x_2	$f(x_1, x_2)$
0	0	1
0	1	1
1	0	0
1	1	1

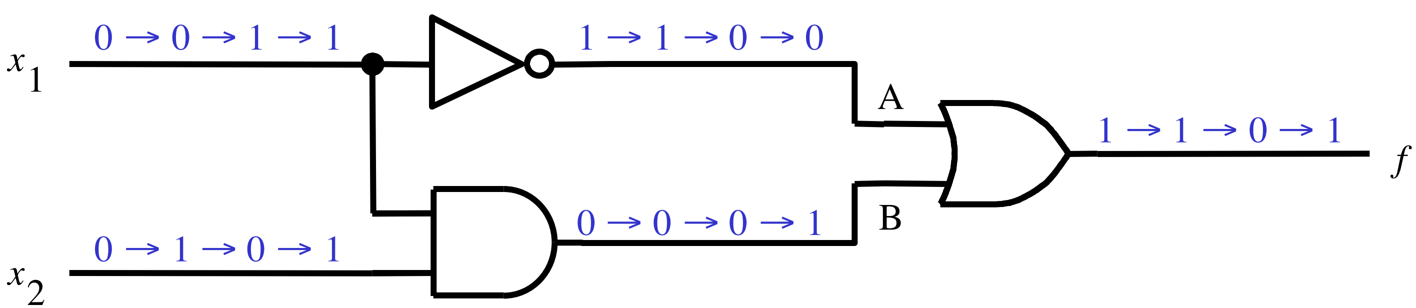
(b) Truth table for f

Figure 2.10 a Logic network

Representação por diagrama de tempo



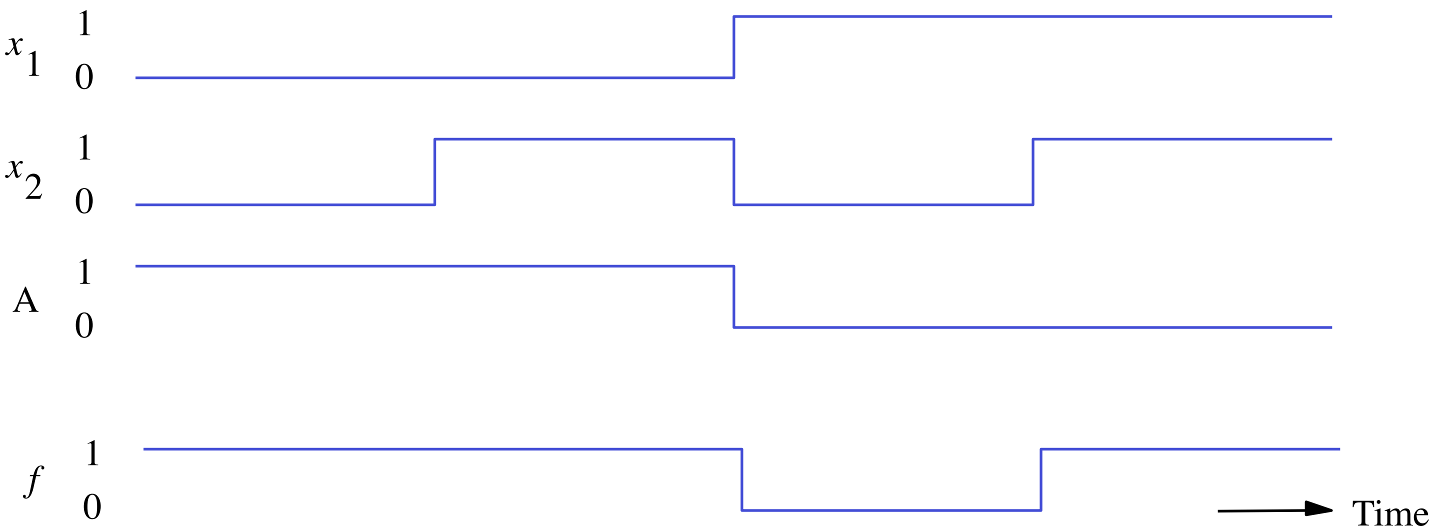
(c) Timing diagram



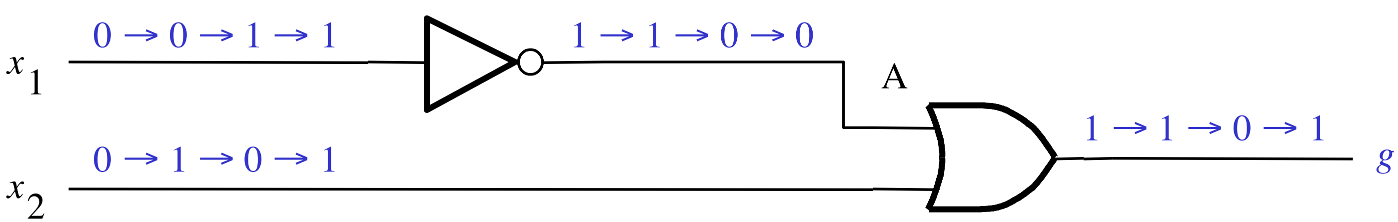
(a) Network that implements $f = \bar{x}_1 + x_1 \cdot x_2$

Figure 2.10 b Logic network

Otimização de redes lógicas



(c) Timing diagram



(d) Network that implements $g = \bar{x}_1 + x_2$

$f = g \qquad \bar{x}_1 + x_1 \cdot x_2 = \bar{x}_1 + x_2$

Figure 2.10 b Logic network

1849 - George Boole → Álgebra Booleana

1930 – Claude Shannon → Circuitos Chaveados

Muito usada para descrever e analisar circuitos lógicos

Axiomas (postulados):

1a.	$0.0 = 0$
1b.	$1+1 = 1$
2a.	$1.1 = 1$
2b.	$0+0 = 0$
3a.	$0.1 = 1.0 = 0$
3b.	$1+0=0+1 = 1$
4a.	If $x = 0$; $\overline{x} = 1$
4b.	If $x = 1$; $\overline{x} = 0$

Teoremas (regras a partir dos axiomas):

5a.	$x.0 = 0$
5b.	$x+1 = 1$
6a.	$x.1 = x$
6b.	$x+0 = x$
7a.	$x.x = x$
7b.	$x+x = x$
8a.	$x.\overline{x} = 0$
8b.	$x+\overline{x} = 1$
9.	$\overline{\overline{x}} = x$

Princípio de dualidade: trocar 0 por 1 e vice versa e
trocar + por . e vice versa

Álgebra Booleana

Propriedades/identidades para 2 e 3 variáveis:

10a. $x.y = y.x$ Comutativa
10b. $x+y = y+x$
11a. $x.(y.z) = (x.y).z$ Associativa
11b. $x+(y+z) = (x+y)+z$
12a. $x.(y+z) = x.y+x.z$ Distributiva
12b. $x+y.z = (x+y).(x+z)$
13a. $x+x.y = x$ Absorção
13b. $x.(x+y) = x$
14a. $x.y+x.\overline{y} = x$ Combinação
14b. $(x+y).(x+\overline{y}) = x$
15a. $\overline{x.y} = \overline{x}+\overline{y}$ → Teorema de DeMorgan
15b. $\overline{x+y} = \overline{x}.\overline{y}$
16a. $x+\overline{x}.y = x+y$ Consenso
16b. $x.(\overline{x}+y) = x.y$

Prova do Teorema de DeMorgan

$\overline{x.y} = \overline{x} + \overline{y}$ → Teorema de DeMorgan

x	y	$x \cdot y$	$\overline{x \cdot y}$	$\overline{x}$	$\overline{y}$	$\overline{x} + \overline{y}$
0	0	0	1	1	1	1
0	1	0	1	1	0	1
1	0	0	1	0	1	1
1	1	1	0	0	0	0

LHS RHS

Manipulação algébrica usando os axiomas, teoremas e propriedades

- Analisar os exemplos 2.1 e 2.2 do livro texto
- Estes exemplos simples demonstram que é impraticável lidar com expressões complexas deste modo
- Porém, esses teoremas e propriedades provêm a base para a síntese de funções lógicas em ferramentas CAD/EDA

Diagrama de Venn

- Provem uma forma mais intuitiva para entender como duas equações podem ser equivalentes
- Tradicionalmente utilizado na matemática para ilustrar graficamente as várias operações e relações na álgebra de conjuntos
- Na algebra Booleana (B) há somente dois valores (elementos) no universo B = {0,1}
  - Uma expressão envolvendo uma ou mais variáveis tem área marcada onde o valor da expressão é igual a 1

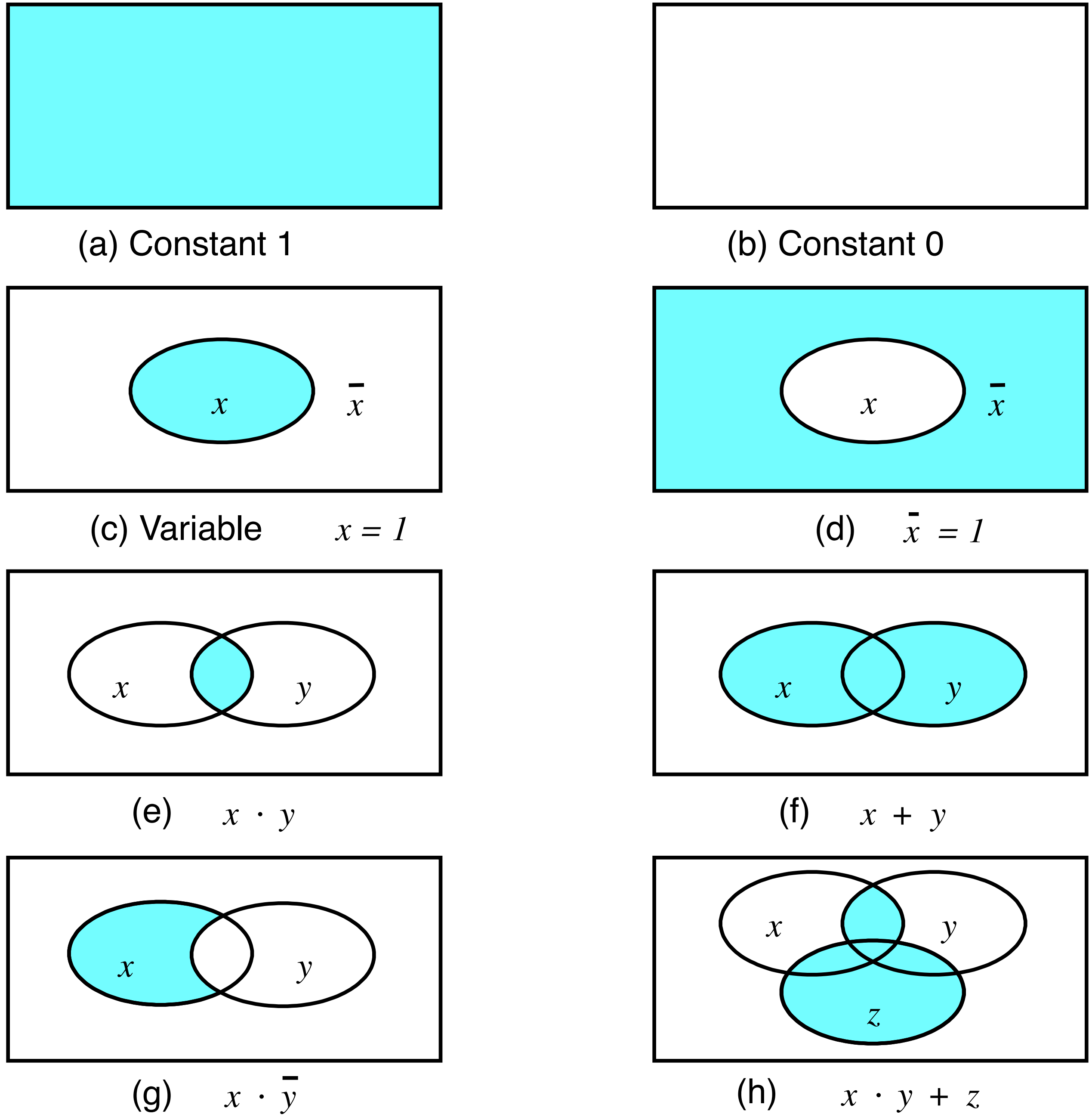


Figure 2.12. The Venn diagram representation.

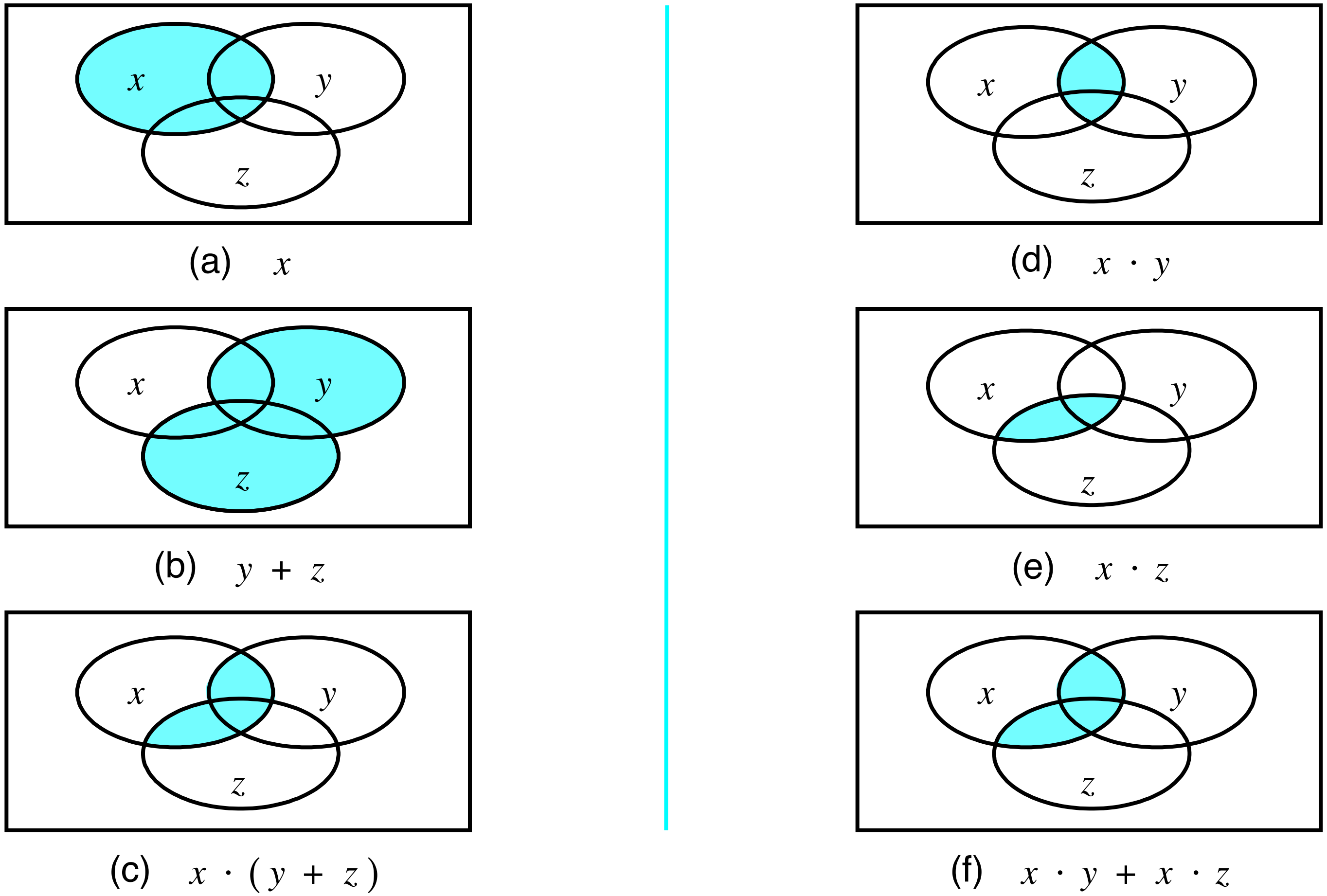


Figure 2.13. Verification of the distributive property

$$x \,.\, (y + z) = x \,.\, y + x \,.\, z$$

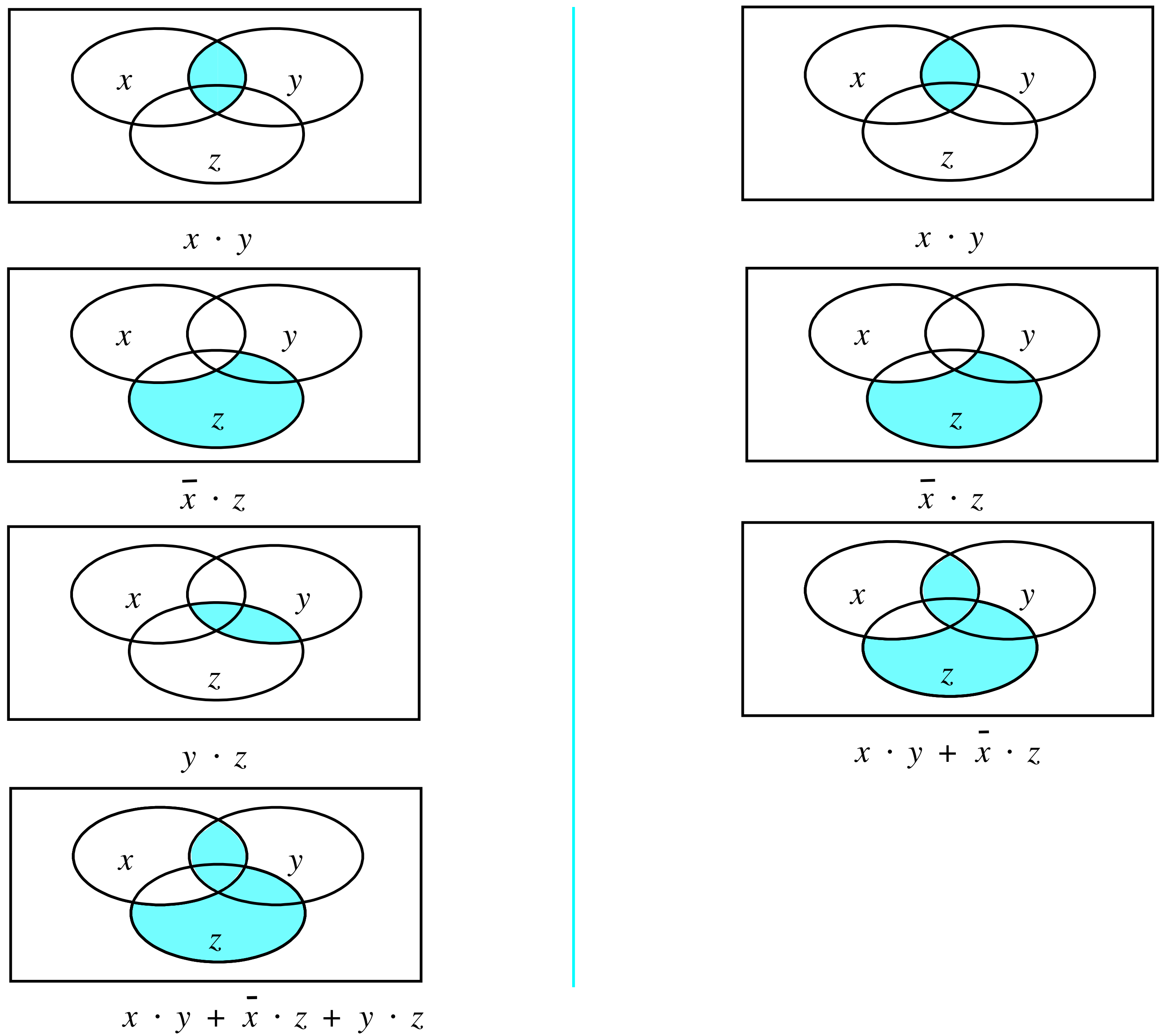


Figure 2.14. Verification of $x \,.\, y + \bar{x} \,.\, z + y \,.\, z = x \,.\, y + \bar{x} \,.\, z$

Notações e terminologias

- AND = . = ^
- OR = + = $\vee$
- NOT = $\bar{x}$ = x' = !x = ~x = NOT x
- Operação AND conhecida como produto e OR como soma
- Precedência de operações
  - NOT, AND e OR
- Para simplificar a aparência das expressões lógicas é comum omitir o operador "."
  - xy+z

Síntese usando portas lógicas AND, OR e NOT

x_1	x_2	$f(x_1, x_2)$
0	0	1
0	1	1
1	0	0
1	1	1

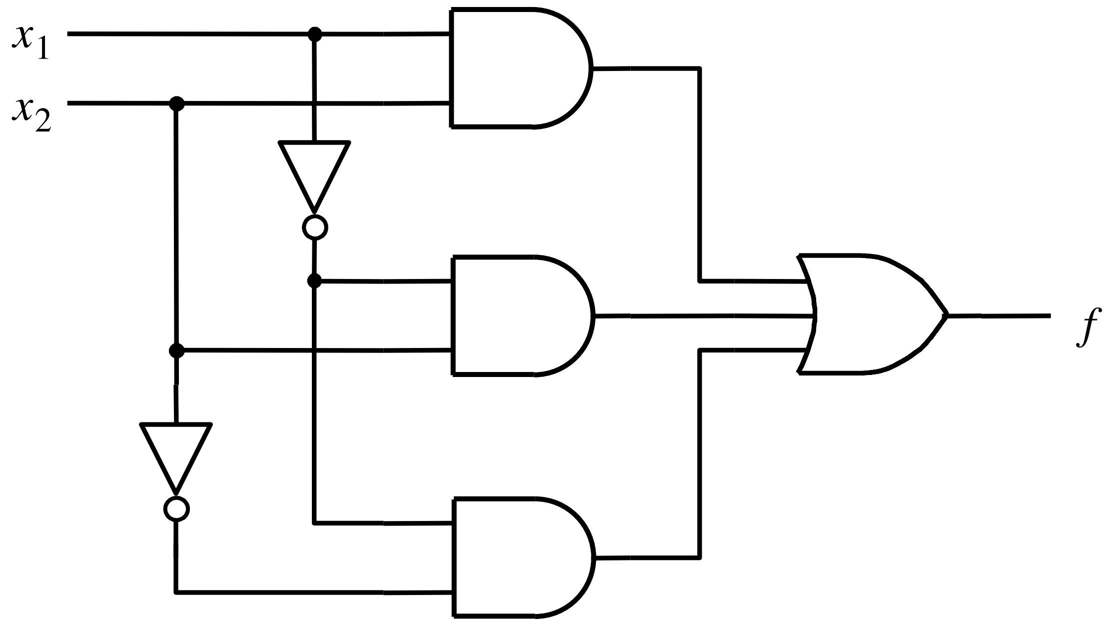
Figure 2.15. A function to be synthesized.

- Um procedimento possível para implementar o circuito lógico da tabela verdade seria criar um termo de produto para cada caso em que a função produz 1 na saída
- $f(x_1,x_2)=x_1x_2 + \overline{x}_1\overline{x}_2 + \overline{x}_1x_2$

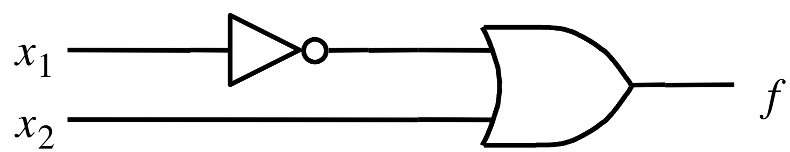
Síntese usando portas lógicas AND, OR e NOT

- Exercício: Mostre com álgebra Booleana como podemos obter o circuito da figura 2.16 (b) a partir de 2.16 (a)

- O processo que gera um circuito a partir de uma descrição do comportamento funcional desejado é chamado de síntese



(a) Canonical sum-of-products



(b) Minimal-cost realization

Figure 2.16. Two implementations of a function in Figure 2.15.

Fim!