Faculdades Pitágoras
de Uberlândia

Sistemas de Informação

Apostila de

Lógica Matemática e Computacional

Prof. Walteno Martins Parreira Júnior

www.waltenomartins.com.br

waltenomartins@yahoo.com

2014

Sumário

0 APRESENTAÇÃO

0.1 Metodologia da Disciplina

- Aula expositiva: informação, conhecimento, aprendizagem de conceitos e princípios.
- Atividades de aprendizagem em grupo: desenvolvimento de habilidades e competências, não só da disciplina em questão, mas também habilidade de trabalhar em grupos e equipes. Ênfase em projetos e pesquisas dos alunos, fazendo a relação entre a teoria e o mundo real.
- Atividades Avaliativas (individuais e coletivas)

0.2 Visão geral da Disciplina

- A disciplina de Lógica Matemática e Computacional é fundamental para todos os alunos das áreas de Engenharias e da Computação diante da sua ampla aplicabilidade.
- A Lógica permite formalizar argumentos que podem ser validados ou refutados, usando raciocínio lógico. Estas estruturas de argumentação podem ser utilizadas na elaboração de algoritmos na área de programação.
- Além disso, a disciplina de Lógica Matemática e Computacional prepara os alunos para projetar sistemas digitais, utilizando os diferentes sistemas de numeração e os conceitos de álgebra booleana, temas desta disciplina.

0.3 Objetivos da Disciplina

- Desenvolver no aluno a habilidade de elaborar uma estrutura de argumentação lógica, utilizando uma linguagem formal, tanto para verbalizá-la, como para verificar equivalências e implicações.
- Preparar o aluno para aplicar os conceitos de lógica na modelagem e na construção de algoritmos computacionais.
- Capacitar o aluno a aplicar os conceitos de inferência lógica na construção de dispositivos autônomos e sistemas especialistas.
- Capacitar o aluno a aplicar os conceitos da álgebra booleana na construção de sistemas digitais, assim como a utilização dos diferentes sistemas de numeração.

0.4 Atividades em grupo:

- O aluno deve usar o material indicado pelo professor. Não é possível desenvolver satisfatoriamente uma atividade sem um mínimo de conhecimento do conteúdo ministrado nas aulas expositivas.
- A participação será avaliada a cada encontro. A nota de participação não é nota de presença e sim das atividades efetivamente desenvolvidas.

0.5 Avaliação dos Alunos

- Conhecimentos adquiridos.
- Habilidades e competências específicas da disciplina, principalmente, a competência argumentativa.
- Atitudes: abertura às idéias e aos argumentos dos outros, mostrando disponibilidade para rever suas próprias opiniões; cooperação com os outros, mostrando que a crítica só é eficaz através do diálogo justo e honesto, no seio de uma comunidade.
- Participação efetiva nas aulas e atividades coletivas (não é apenas presença).

1 INTRODUÇÃO À LÓGICA

1.1 Origens

A lógica iniciou seu desenvolvimento na Grécia Aristóteles (384 – 322 AC) e os antigos filósofos gregos passaram a usar em suas discussões sentenças enunciadas nas formas afirmativa e negativa, resultando em grande simplificação e clareza.

Em 1847, Augustus DeMorgam (1806-1871) publicou o tratado Formal Logic. Em 1848, George Boole (1815-1864) escreveu The Mathematical Analysis of Logic e depois publicou um livro sobre o que foi denominado posteriormente de Álgebra de Boole.

Em 1879, Gotlob Frege (1848-1925) contribuiu no desenvolvimento da lógica com a obra Begriffsschrift. As idéias de Frege só foram reconhecidas pelos lógicos mais ou menos a partir de 1905. A escola italiana, que desenvolveu quase toda simbologia da matemática utilizada atualmente, é composta de Giuseppe Peano (1858-1932) e também por Burali-Forti, Vacca, Pieri, Pádoa, Vailati, etc.

Bertrand Russell (1872-1970) e Alfred North Whitehead (1861-1947) iniciam o atual período da lógica com a publicação da obra Principia Mathematica no início do século XX. Também contribuem para o estágio atual, David Hilbert (1862-1943) e sua escola alemã com von Neuman, Bernays, Ackerman e outros.

Em 1938, Claude Shannon mostrou a aplicação da álgebra de Boole na analise de circuitos de relês.

Podemos dizer que a lógica estuda as condições objetivas e ideais para justificar a verdade e não cuida da própria verdade. Ela estuda as condições formais para justificar a verdade, isto é, as condições que o pensamento deve preencher para ser coerente consigo mesmo e demonstrar a verdade já conhecida. A lógica estuda as relações do pensamento consigo mesmo para possibilitar a construção de um contexto correto de justificação, isto é, para a definição argumentativa de premissas corretamente dispostas para uma conclusão justificada. Uma das condições para se ter a verdade demonstrada, e portanto justificada, é que o pensamento seja coerente consigo mesmo, isto é, que siga as leis da razão expressa lingüisticamente.

1.2 Conceitos preliminares.

1.2.1 – Proposição

Definição: todo conjunto de palavras ou símbolos que exprimem um pensamento de sentido completo.

As proposições transmitem pensamentos, isto é, afirmam fatos ou exprimem juízos que formamos a respeito de determinados entes.

Exemplos:
- ✓ Japão está situado no continente africano.
- ✓ A lâmpada da sala está acesa.
- ✓ A cidade de Recife é a capital de Pernambuco.

Na linguagem natural nos acostumamos com vários tipos de proposições ou sentenças:

a) Declarativas
- ✓ Márcio é engenheiro.
- ✓ Todos os homens são mortais.
- ✓ sen $(\pi/_2) = 1$

- ✓ A lua gira em torno da terra.

b) Interrogativas
- ✓ Será que o Roberto vai ao cinema hoje?
- ✓ Quantos alunos faltaram hoje a aula de lógica?
- ✓ O Brasil ganhará a copa do mundo de 2006?

c) Exclamativas
- ✓ Feliz Natal!
- ✓ Vencemos!
- ✓ Passamos no vestibular!

d) Imperativas
- ✓ Não falte as aulas de lógica.
- ✓ Feche a porta.
- ✓ Fique calado.

Estudaremos somente as **proposições declarativas**, pois elas podem ser facilmente classificadas em verdadeiras ou falsas.

1.2.2 – As três leis do pensamento

A lógica adota como regras fundamentais do pensamento os seguintes princípios (ou axiomas):

a) Princípio da Identidade

Se qualquer proposição é verdadeira, então ela é verdadeira.

b) Princípio da Não-Contradição

Uma proposição não pode ser verdadeira e falsa ao mesmo tempo.

c) Princípio do terceiro Excluído

Toda proposição ou é verdadeira ou é falsa, isto é, verifica-se sempre um destes casos e nunca um terceiro caso.

Com base nesses princípios as proposições simples são ou verdadeiras ou falsas - sendo mutuamente exclusivos os dois casos; daí dizer que a lógica clássica é bivalente

Exemplos de proposições falsas:
- ✓ O navegador Vasco da Gama descobriu o Brasil.
- ✓ O escritor francês Dante escreveu Os Lusíadas.
- ✓ número π é um número racional.
- ✓ O México está localizado na América do Sul.

1.2.3 – Valores lógicos das proposições

Definição: chama-se **valor lógico** de uma proposição a **verdade** se a proposição é verdadeira e a **falsidade** se a proposição é falsa.

Os valores lógicos verdade e falsidade de uma proposição designam-se abreviadamente pelas letras **V** e **F** respectivamente. Assim, o que os princípios (axiomas) afirmam é que:

“Toda proposição tem um, e um só, dos valores V e F.”

O valor lógico de uma proposição P é a verdade (V) se P é verdadeira, escrevendo:

$v(P) = V$ e lê-se: “o valor lógico de P é V”

Exemplos:

P: O mercúrio é mais pesado que a água. v(P) = V

Q: O sol gira em torno da Terra. v(Q) = F

R: A lua é um satélite natural da Terra. v(R) = V

1.2.4 – Sentenças abertas

Definição: quando em uma proposição substituímos alguns (ou todos os) componentes por variáveis, obtemos uma sentença (proposição) aberta.

Seja a proposição "Magda é Uberlandense", se substituímos o nome **Magda** pela variável **X**, obteremos a sentença aberta "X é Uberlandense", que não é necessariamente verdadeira e nem falsa.

Exemplos:

P: X é filho de Y.

Q: $x - y = 12$

R: Se x é sobrinho de y, então, x é primo de z.

1.2.5 – Proposições simples (atômicas)

Definição: chama-se **proposição simples**, a proposição que não contém nenhuma outra proposição como parte integrante de si.

Exemplos:

P: Carlos é careca.

Q: Pedro é estudante.

R: O galo põe ovos.

1.2.6 – Conectivos lógicos

Definição: são palavras ou frases que são usadas para formar novas proposição a partir de outras proposições.

Os conectivos usuais em lógica matemática são:

Conectivo	Palavras	Símbolo
Negação	Não	**~ ou ¬**
Conjunção	E	$\wedge$
Disjunção	Ou	$\vee$
Condicional	Se ... então ...	$\rightarrow$
Bicondicional	... Se, e somente se ...	$\leftrightarrow$

1.2.7 – Proposições compostas

Definição: são as proposições formadas por duas ou mais proposições simples e ligadas pelos conectivos lógicos.

Exemplos:

P: Carlos é careca **e** Pedro é estudante.

Q: Carlos é careca **ou** Pedro é estudante.

R: **Se** Carlos é careca, **então** Carlos é infeliz.

S: Carlos é careca **se, e somente se** Pedro é estudante.

1.3 Tabelas-verdade e conectivos.

Segundo o Princípio do terceiro excluído, toda proposição simples P é verdadeira ou é falsa, isto é, tem valor lógico V (verdade) ou o valor lógico F (falsidade). Em uma proposição composta, a determinação do seu valor lógico é feito segundo o princípio: "O valor lógico de qualquer proposição composta depende unicamente dos valores lógicos das proposições simples componentes, ficando por eles univocamente determinado."

Para aplicar este princípio na prática, recorre-se ao uso do dispositivo denominado **Tabela-verdade**, que apresenta todos os possíveis valores lógicos da proposição composta correspondentes a todas as possíveis atribuições de valores lógicos às proposições simples correspondentes.

O número de linhas da tabela-verdade de uma proposição composta está em função do número de proposições simples que a compõem. A tabela-verdade de uma proposição composta com **n** proposições simples contém $\mathbf{2^n}$ linhas.

Portanto, podemos observar:

a) Para uma proposição simples P, o numero de linhas da tabela-verdade será: $2^1 = 2$, representando na tabela-verdade:

P
V
F

b) Para uma proposição composta cujas proposições simples componentes são P e Q, o número de linhas da tabela-verdade será: $2^2 = 4$, que será representada:

P	**Q**
V	V
V	F
F	V
F	F

c) Para uma proposição composta cujas proposições simples componentes são P, Q e R, o número de linhas da tabela-verdade será: $2^3 = 8$, que será representada:

P	**Q**	**R**
V	V	V
V	V	F
V	F	V
V	F	F
F	V	V
F	V	F
F	F	V
F	F	F

Observe que os valores lógicos V e F se alternam de quatro (4) em quatro para a primeira proposição (P), de dois (2) em dois para a Segunda proposição (Q) e de um (1) em um para a terceira proposição (R).

1.3.1 – Operação Lógica Negação

Definição: chama-se negação de uma proposição P a proposição representada por "Não P", cujo valor lógico é a verdade (V) quando P é falsa e a falsidade (F) quando P é verdadeira.

Simbolicamente, a negação de P indica-se com a notação "~P", que se lê: "**NÃO P**"

A tabela-verdade:

P	**~P**
V	F
F	V

v(~P) = ~ v(P)

Para uma proposição P, podemos formar a sua negação de qualquer um dos seguintes modos:

- ✓ "não é verdade que P"
- ✓ "é falso que P"
- ✓ "não" em P

Exemplo:

P: Lídia é estudiosa.

~P: Não é verdade que Lídia é estudiosa.

~P: É falso que Lídia é estudiosa.

~P: Lídia não é estudiosa.

Q: João não foi ao cinema.

~Q: É falso que João não foi ao cinema.

Observações: Pode-se ter algumas variações, por necessidades da língua portuguesa, por exemplo:

P: Todos os homens são elegantes.

~P: Nem todos os homens são elegantes.

Q: Nenhum homem é elegante.

~Q: Algum homem é elegante.

1.3.2 – Operação Lógica Conjunção

Definição: chama-se conjunção de duas proposições P e Q a proposição representada por "P e Q", cujo valor lógico é a verdade (V) quando as proposições P e Q são ambas verdadeiras e a falsidade (F) nos demais casos.

Simbolicamente, a conjunção de duas proposições P e Q indica-se com a notação "$P \wedge Q$", que se lê "**P E Q**". A tabela-verdade:

P	Q	$P \wedge Q$
V	V	V
V	F	F
F	V	F
F	F	F

$v(P \wedge Q) = v(P) \wedge v(Q)$

Exemplo:

P: Pitágoras era Grego Q: Descartes era Francês

$P \wedge Q$: Pitágoras era Grego e Descartes era Francês.

$v(P) = V$ e $v(Q) = V \Rightarrow v(P \wedge Q) = V$

1.3.3 – Operação Lógica Disjunção

Definição: chama-se disjunção de duas proposições P e Q a proposição representada por "P ou Q", cujo valor lógico é a verdade (V) quando ao menos uma das proposições P e Q é verdadeira e a falsidade (F) quando as proposições P e Q são ambas falsas.

Simbolicamente, a conjunção de duas proposições P e Q indica-se com a notação "$P \vee Q$ ", que se lê "**P OU Q**".

A tabela-verdade:

P	Q	$P \vee Q$
V	V	V
V	F	V
F	V	V
F	F	F

$v(P \vee Q) = v(P) \vee v(Q)$

Exemplo:

P: A cidade de Paris é a capital da França

Q: O sol é um satélite artificial da terra

$P \vee Q$: A cidade de Paris é a capital da França ou o sol é um satélite artificial da terra.

$v(P) = V$ e $v(Q) = F \Rightarrow v(P \vee Q) = V$

Na linguagem coloquial a palavra ou tem dois sentidos. Exemplifiquemos:

P: Amilton é bombeiro ou eletricista.

Q: Rosa é mineira ou goiana.

Na proposição Q (Rosa é mineira ou goiana), está afirmando que somente uma das proposições é verdadeira, pois não é possível ocorrer as duas coisas: Rosa ser mineira e goiana ao mesmo tempo.

Na proposição P, diz-se que o ou é inclusivo, já na proposição Q diz-se que o ou é exclusivo. Logo a proposição P é uma disjunção inclusiva ou simplesmente disjunção; e a proposição Q é a disjunção exclusiva.

A disjunção exclusiva de duas proposições P e Q é a proposição composta "$P \underline{\vee} Q$", que se lê: "ou P ou Q" ou se lê "P ou Q, mas não ambos". É a falsidade (F) quando o valor lógico das proposições P e Q forem ambos verdadeiros ou ambos falsos e a verdade (V) quando P e Q tem valores lógicos diferentes.

A tabela-verdade:

P	Q	$P \veebar Q$
V	V	F
V	F	V
F	V	V
F	F	F

$v(P \veebar Q) = v(P) \veebar v(Q)$

1.3.4 – Operação Lógica Condicional

<u>Definição</u>: chama-se proposição condicional de duas proposições P e Q a proposição representada por "se P então Q", cujo valor lógico é a <u>falsidade</u> (F) no caso em que a proposição P é verdadeira e Q é falsa e a <u>verdade</u> (V) nos demais casos.

Simbolicamente, a condicional de duas proposições P e Q indica-se com a notação "$P \rightarrow Q$ ", que se lê:

- "**P SOMENTE SE Q**"
- "p é condição suficiente para q"
- "q é condição necessária para p"
- "p implica (ou acarreta) q"

Na condicional "$P \rightarrow Q$", diz-se que P é o <u>antecedente</u> e Q é o <u>conseqüente</u>. O símbolo "$\rightarrow$" é chamado de <u>símbolo de implicação</u>.

A tabela-verdade:

P	Q	$P \rightarrow Q$
V	V	V
V	F	F
F	V	V
F	F	V

$v(P \rightarrow Q) = v(P) \rightarrow v(Q)$

Observação: Uma condicional $P \rightarrow Q$ <u>não afirma</u> que o conseqüente Q se deduz ou é conseqüência do antecedente P; o que uma condicional afirma é unicamente uma <u>relação</u> entre os valores lógicos do antecedente e do conseqüente de acordo com a tabela-verdade apresentada acima.

Exemplo:

P: Paris é a capital da França

Q: O sol é um satélite natural da terra

$P \rightarrow Q$: Se Paris é a capital da França então o sol é um satélite natural da terra.

$v(P) = V$ e $v(Q) = F \Rightarrow v(P \rightarrow Q) = F$

1.3.5 – Operação Lógica Bi-condicional

Definição: chama-se proposição bi-condicional de duas proposições P e Q a proposição representada por "P se, e somente se Q", cujo valor lógico é a verdade (V) quando as proposições P e Q são ambas verdadeiras ou ambas falsas e a falsidade (F) nos demais casos.

A conjunção da sentença "$P \rightarrow Q$" com a sentença "$Q \rightarrow P$" resulta na sentença "$P \leftrightarrow Q$", assim temos $(P \rightarrow Q) \wedge (Q \rightarrow P)$ equivale a $(Q \leftrightarrow P)$.

Simbolicamente, a bi-condicional de duas proposições P e Q indica-se com a notação "$P \leftrightarrow Q$", que se lê:

- **"P SE, E SOMENTE SE Q"**
- "p é equivalente a q"
- "q se, e somente se p"
- "q é equivalente a p"
- "p é condição necessária e suficiente para q"
- "q é condição necessária e suficiente para p"

A tabela-verdade:

P	Q	$P \leftrightarrow Q$
V	V	V
V	F	F
F	V	F
F	F	V

$v(P \leftrightarrow Q) = v(P) \leftrightarrow v(Q)$

Exemplo:

P: A cidade de Paris é a capital da França

Q: O sol é um satélite natural da terra

$P \leftrightarrow Q$: A cidade de Paris é a capital da França se, e somente se o sol é um satélite natural da terra.

$v(P) = V$ e $v(Q) = F \Rightarrow v(P \leftrightarrow Q) = F$

1.3.6 - Tabela Verdade dos conectores

Resumindo, tem-se a seguinte tabela verdade:

p	q	$p \wedge q$	$p \vee q$	$p \veebar q$	$p \rightarrow q$	$p \leftrightarrow q$
V	V	V	V	F	V	V
V	F	F	V	V	F	F
F	V	F	V	V	V	F
F	F	F	F	F	V	V

p	~p
V	F
F	V

1.3.7 – Ordem De Precedência Dos Conectores

Para reduzir o número de parêntesis necessários em uma proposição composta (fórmula lógica proposicional), estipula-se uma ordem na qual os conectores são aplicados. A ordem de precedência é:

maior

a) conectivos dentro de parêntesis, do mais interno para o mais externo
b) ~
c) $\wedge$
d) $\vee$ ou $\underline{\vee}$
e) $\rightarrow$
f) $\leftrightarrow$

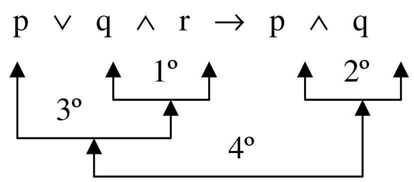
menor

Quando há dois ou mais conectivos de mesma ordem de precedência, o conectivo mais a esquerda na fórmula proposicional tem prioridade sobre o conectivo à direta.

Exemplos:
a) "$p \vee \sim q$" equivale a "$(p \vee (\sim q))$"

b) "$p \vee q \rightarrow r$" equivale a "$((p \vee q) \rightarrow r)$"

c) "$p \vee q \wedge r \rightarrow p \wedge q$" equivale a "$((p \vee (q \wedge r)) \rightarrow (p \wedge q))$"



1.4 Exercícios

1.4.1 - Com o intuito de abordar os conceitos preliminares estudados nesta unidade, responda às seguintes perguntas:

a) Qual é a relação entre sentença e proposição?

b) Toda sentença é uma proposição? E o contrário: toda proposição é uma sentença? Por quê? Dê exemplos.

c) Defina, com suas próprias palavras, o valor lógico.

d) De acordo com a lógica clássica, quais são os valores lógicos possíveis para uma proposição?

e) Compare a teoria dos conjuntos, vista no ensino fundamental, com a lógica clássica discutida nesta unidade.

f) Quais são os três postulados clássicos?

1.4.2 - Represente as seguintes frases em linguagem natural, utilizando lógica formal. Para cada resposta, devem-se especificar as proposições simples extraídas:

a) João mede 1,78m e Maria pesa 60kg.

b) Fortaleza é capital do Maranhão desde que Rio de Janeiro tenha mais de 250 mil habitantes.

c) Ontem o dólar não fechou a R$2,18 ou o índice Bovespa fechou estável.

d) Só irei ao clube se amanhã fizer sol.

1.4.3 - Sejam as seguintes proposições simples:

p: "Tiradentes morreu afogado" e q: "Jaime é gaúcho".

Traduzir para linguagem natural, as seguintes proposições compostas:

a) $p \wedge q$

b) $p \rightarrow q$

c) $\sim p \vee q$

d) $q \leftrightarrow \sim p$

1.4.4 - Determinar o valor lógico (V ou F) de cada uma das seguintes proposições.

a) O número 23 é primo.

b) Goiânia é a capital de Tocantins.

c) O número 25 é quadrado perfeito.

d) Todo número divisível por 5 termina com 5.

e) $\pi = 3$

f) $3 > 2$

g) $\pi > 3$

h) $\sqrt{4} = 2$

1.4.5 - Sejam as proposições:

p: O empregado foi demitido. q: O patrão indenizou o empregado.

Escreva em notação simbólica, cada uma das proposições abaixo:

a) O empregado não foi demitido.
b) O patrão não indenizou o empregado.
c) O empregado foi demitido e o patrão indenizou o empregado.
d) É falso que o empregado foi demitido ou o patrão indenizou o empregado.
e) O empregado foi demitido ou o patrão não indenizou o empregado.
f) não é verdade que o empregado não foi demitido.

1.4.6 - Sejam as proposições:

p: Rosas são vermelhas. q: Violetas são azuis. r: Cravos são amarelos.

Escreva em notação simbólica, cada uma das proposições compostas abaixo:

a) Rosas são vermelhas e violetas são azuis.
b) Rosas são vermelhas, ou violetas são azuis ou os cravos são amarelos.
c) Se violetas são azuis, então as rosas são vermelhas e os cravos são amarelos.
d) Rosas são vermelhas se e somente se, as violetas não forem azuis e os cravos não são amarelos.

e) Rosas são vermelhas e, se os cravos não são amarelos então as violetas não são azuis.

1.4.7 - Sejam as proposições:

p: Jô Soares é gordo. q: Jô Soares é artista. r: Paulo é cantor.

Escreva em notação simbólica, cada uma das proposições abaixo:

a) Jô Soares não é gordo.
b) Jô Soares não é artista.
c) Não é verdade que Jô Soares não é gordo
d) Jô Soares é gordo ou artista.
e) Se Jô Soares não é artista então Paulo não é cantor.
f) Jô Soares é artista se e somente se Paulo é cantor

1.4.8 - Sejam as proposições:

p: João joga futebol.

q: Pedro joga tênis.

Traduzir as formulas lógicas para o português.

a) $p \vee q$ b) $p \wedge q$ c) $p \wedge \sim q$ d) $p \rightarrow \sim q$
e) $\sim p \wedge \sim q$ f) $\sim\sim p$ g) $\sim(\sim p \wedge \sim q)$ h) $\sim p \leftrightarrow \sim q$

1.4.9 - Sejam as proposições:

p: A bola é vermelha. q: O bambolê é amarelo.

Traduzir as formulas lógicas para o português.

a) $p \vee q$ b) $p \wedge q$ c) $p \wedge \sim q$
d) $\sim p \wedge \sim q$ e) $\sim\sim p$ f) $\sim(\sim p \wedge \sim q)$
g) $\sim p \leftrightarrow \sim q$ h) $p \rightarrow \sim q$ i) $\sim(\sim q \rightarrow p)$

1.4.10 - Determinar o valor lógico (V ou F) de cada uma das proposições compostas abaixo, sabendo o valor lógico de cada proposição simples v(p) = V e v(q) = F.

a) $p \vee q \veebar p$ b) $p \wedge q \veebar p$ c) $p \wedge \sim q$ d) $p \vee (\sim p \wedge \sim q)$
e) $\sim p \wedge \sim q$ f) $\sim\sim p$ g) $p \leftrightarrow (\sim p \rightarrow \sim q)$ h) $p \leftrightarrow \sim q \rightarrow p$
i) $\sim p \wedge q \veebar p$ j) $p \rightarrow \sim q \veebar p$ k) $p \wedge (\sim q \rightarrow p)$ l) $\sim(r \wedge \sim q) \leftrightarrow p$

1.4.11 - Determinar o valor lógico (V ou F) de cada uma das proposições compostas abaixo, sabendo o valor lógico de cada proposição simples v(p) = V, v(q) = V e v(r) = F.

a) $p \vee q \veebar r$ b) $p \wedge q \veebar r$ c) $p \wedge \sim q \veebar r$ d) $r \vee (\sim p \wedge \sim q)$
e) $\sim p \wedge \sim q \veebar r$ f) $\sim\sim p$ g) $p \leftrightarrow (\sim r \rightarrow \sim q)$ h) $p \veebar r \leftrightarrow \sim q \rightarrow p$
i) $\sim r \wedge q \veebar p$ j) $p \rightarrow \sim q \veebar p$ k) $r \veebar p \wedge (\sim q \rightarrow p)$ l) $\sim(r \wedge \sim q) \leftrightarrow p$

1.4.12 - Construa a tabela-verdade de cada uma das seguintes proposições:

a) $p \vee q \veebar p$ b) $p \wedge q \veebar p$ c) $p \wedge \sim q$
d) $\sim p \wedge \sim q$ e) $\sim\sim p$ f) $p \vee (\sim p \wedge \sim q)$
g) $\sim p \wedge q \leftrightarrow \sim q \rightarrow p$ h) $p \rightarrow \sim q$ i) $p \wedge (\sim q \rightarrow p)$

j) $\sim p \lor r \leftrightarrow \sim q \lor r$ k) $p \land r \to \sim q$ l) $\sim(r \land \sim q) \to p$

1.4.13 - Construir as tabelas-verdade das seguintes proposições:

a) $\sim p \land r \to q \lor \sim r$
b) $p \to (p \to \sim r) \leftrightarrow q \lor r$
c) $p \to r \leftrightarrow q \lor \sim r$
d) $(p \land q \to r) \lor (\sim p \leftrightarrow q \lor \sim r)$

1.4.14 - Determinar P(VF, FV, FF) em cada um dos seguintes casos:

a) $P(p, q) = \sim(\sim p \leftrightarrow q)$
b) $P(p, q) = \sim p \lor q \to p$
c) $P(p, q) = (p \lor q) \land \sim(p \land q)$
d) $P(p, q) = (p \land \sim q) \lor (\sim p \land q)$
e) $P(p, q) = \sim((p \lor q) \land (\sim p \lor \sim q))$
f) $P(p, q) = \sim q \lor p \leftrightarrow q \to \sim p$
g) $P(p, q) = (p \lor q) \land \sim p \to (q \to p)$
h) $P(p, q) = (p \land \sim q) \lor (\sim p \land q) \to p$

1.4.15 - Sabendo que as proposições p, q são **verdadeiras** e que as proposições r e s são **falsa**s, determinar o valor lógico (V ou F) de cada uma das seguintes proposições:

a) $p \land q \to r$
b) $(q \lor r) \land (p \lor s)$
c) $r \lor s \to q$
d) $(r \to s) \land (p \land q)$
e) $q \leftrightarrow p \land s$
f) $(p \land \sim q) \lor r$
g) $p \to \sim(r \land s)$
h) $\sim((r \to p) \lor (s \to q))$
i) $(q \to s) \to r$

2 TAUTOLOGIA, CONTINGÊNCIA E CONTRADIÇÃO

2.1 Tautologia

Definição: denomina-se tautologia a proposição composta que é sempre verdadeira. Na tabela-verdade de uma proposição tautológica, a última coluna (à direita) contém somente V's (verdade).

Em outros termos, tautologia é toda proposição composta P (p,q,r,...) cujo valor lógico é sempre V (verdade), quaisquer que sejam os valores lógicos das proposições simples componentes p, q, r, ...

As tautologias são também denominadas proposições tautológicas ou proposições logicamente verdadeiras.

Exemplo: a proposição " $\sim(p \wedge \sim p)$ " denominado de Princípio da não contradição é tautológica, conforme se vê pela tabela-verdade:

p	~p	$p \wedge \sim p$	$\sim(p \wedge \sim p)$
V	F	F	V
F	V	F	V

Portanto, dizer que uma proposição não pode ser simultaneamente verdadeira e falsa é sempre verdadeiro.

2.2 Contradição

Definição: denomina-se contradição a proposição composta que é sempre falsa. Na tabela-verdade de uma proposição contraditória, a última coluna (à direita) contém somente F's (falsidade).

Como uma tautologia é sempre verdadeira (V), a negação de uma tautologia é sempre falsidade (F), ou seja, é uma contradição e vice-versa.

As contradições são também denominadas proposições contraválidas ou proposições logicamente falsas.

Exemplo: a proposição " $p \leftrightarrow \sim p$ "é uma contradição, conforme se vê pela tabela-verdade:

p	~p	$p \leftrightarrow \sim p$
V	F	F
F	V	F

2.3 Contingência

Definição: denomina-se contingência a proposição composta que pode ser verdadeira e pode ser falsa. Na tabela-verdade de uma proposição contigencial, a última coluna (à direita) contém V's (verdadeiros) e F's (falsidades), cada um pelo menos uma vez.

Em outros termos, contingência é toda proposição composta que não é tautologia e nem contradição.

As contingências são também denominadas proposições contingentes ou proposições indeterminadas.

Exemplo: a proposição " $p \rightarrow \sim p$ "é uma contingência, conforme se vê na tabela-verdade:

p	~p	p → ~p
V	F	F
F	V	V

2.4 Exercícios

2.4.1 - Determinar se as proposições a seguir são tautologia, contradição ou contingência, usando o método da Tabela-verdade.

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

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

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

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

2.4.2 - Determinar se as proposições a seguir são tautologia, contradição ou contingência, usando o método da Tabela-verdade.

a) $q \vee r \leftrightarrow (p \wedge q) \rightarrow (p \wedge r)$

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

c) $\sim p \vee q \leftrightarrow r \vee \sim p \wedge r \leftrightarrow \sim r$

d) $p \wedge q \leftrightarrow p \vee q \rightarrow p$

e) $p \vee q \rightarrow q \wedge r \leftrightarrow \sim p \vee \sim r$

f) $p \wedge r \vee q \rightarrow (p \leftrightarrow q \wedge \sim r$

3 EQUIVALÊNCIA E IMPLICAÇÃO LÓGICA.

3.1 Introdução

Diz-se que uma proposição P implica logicamente, ou simplesmente implica, uma proposição Q, se Q for verdadeira todas as vezes que P também o for. Duas proposições são consideradas logicamente equivalentes se as suas tabelas-verdade forem idênticas, em outras palavras, diz-se que uma proposição P é equivalente a uma proposição Q, se as tabelas-verdade de P e Q são idênticas. Outra alternativa é substituir as relações de implicação (⇒) pela condicional (→) e de equivalência (⇔) pela bicondicional (↔) e verificar se é tautológica, ou seja, sempre verdadeiro. Nesse caso, pode-se afirmar que existe a relação entre as proposições envolvidas.

Neste momento, vale destacar que as implicações e equivalências podem ser demonstradas por meio de tabelas-verdade ou por outros métodos, como dedução e indução. Nesta aula, serão utilizadas tabelas-verdade, enquanto que, na próxima unidade serão utilizadas as regras da álgebra das proposições para demonstrar equivalência e implicação lógica por método dedutivo.

3.2 Conceito de implicação lógica

O estudo da implicação lógica é de grande relevância na lógica. As implicações lógicas que serão tratadas levarão em conta a condicional como implicação material.

O símbolo "→" representa uma **operação** entre duas proposições, resultando uma nova proposição. O símbolo "⇒" indica apenas uma **relação** entre duas proposições dadas.

Definição: Diz-se que uma proposição P(p, q, ...) **implica logicamente** uma proposição Q(p, q, ...), se Q é verdadeira (V) todas as vezes que P é verdadeira (V).

Pode-se dizer Diz-se que uma proposição p implica logicamente uma proposição q quando, em suas tabelas-verdade, não ocorre "VF" nesta ordem.

3.3 Propriedades da implicação lógica: reflexiva e transitiva

São:

a) A condição necessária e suficiente para que uma implicação **p ⇒ q** seja verdadeira é que uma condicional p → q seja uma tautologia;

b) Propriedade reflexiva (R): **p ⇒ p;**

c) Propriedade transitiva (T): Se **p ⇒ q** e **q ⇒ r**, então **p ⇒ r**.

3.4 Demonstração de implicação lógica, comparando-se tabelas-verdade

verificar se p ⇒ q → p

p	q	q → p
V	V	V
V	F	V
F	V	F
F	F	V

Comparando os valores lógicos da coluna p com os valores lógicos da coluna $q \rightarrow p$, verificamos que não ocorre "VF" em nenhuma linha, logo $p \Rightarrow q \rightarrow p$ é uma relação válida.

3.5 Demonstração de implicação lógica, substituindo-se a relação pelo conectivo

verificar se $p \Rightarrow q \rightarrow p$

p	q	$q \rightarrow p$	$p \rightarrow q \rightarrow p$
V	V	V	V
V	F	V	V
F	V	F	V
F	F	V	V

Como a coluna de resultado final da tabela-verdade é uma Tautologia, então a relação é verdadeira.

3.6 Implicação entre Sentenças Abertas

Diz-se que uma sentença aberta implica uma outra sentença aberta quando o conjunto-verdade de uma delas está contido no conjunto-verdade da outra.

3.6.1 - Exemplo: Julgar a sentença $\mathbf{x - 3 = 0 \Rightarrow x^2 = 9}$

Resolução:

Determinando o conjunto-verdade da primeira sentença aberta:
$x - 3 = 0$
temos que $x = 3$
logo, $V_1 = \{ 3 \}$
Determinando o conjunto-verdade da segunda sentença aberta:
$x^2 = 9$
$x = \pm 3$
logo, $V_2 = \{ -3, 3 \}$
Podemos observar que $\{ 3 \} \subset \{ -3, 3 \}$
Portanto, podemos dizer que a implicação é verdadeira,
logo a sentença $\mathbf{x - 3 = 0 \Rightarrow x^2 = 9}$ está correta

3.7 Propriedades das Implicações Lógicas

São:

a) A condição necessária e suficiente para que uma implicação $\mathbf{p \Rightarrow q}$ seja verdadeira é que uma condicional $p \rightarrow q$ seja uma tautologia;

b) Propriedade reflexiva: $\mathbf{p \Rightarrow p}$**;**

c) Propriedade transitiva: Se $\mathbf{p \Rightarrow q}$ e $\mathbf{q \Rightarrow r}$, então $\mathbf{p \Rightarrow r}$.

3.8 Implicações Notáveis

Estas implicações são consideradas notáveis (ou clássicas), pois são argumentos válidos fundamentais, usados para fazer “inferências”, isto é, executar os “passos” de uma demonstração ou de uma dedução. Também chamadas de Regras de Inferência.

3.8.1 - Adição

$p \Rightarrow p \vee q$

$q \Rightarrow p \vee q$

organizando a tabela-verdade tem-se que:

p	q	p ∨ q
V	V	V
V	F	V
F	V	V
F	F	F

Não há “VF”, logo $p \Rightarrow p \vee q$

p	q	p ∨ q
V	V	V
V	F	V
F	V	V
F	F	F

Não há “VF”, logo $q \Rightarrow p \vee q$

3.8.2 - Conjunção

$p \wedge q \Rightarrow p$ e $p \wedge q \Rightarrow q$

$q \wedge p \Rightarrow p$ e $q \wedge p \Rightarrow q$

Provar, usando tabela-verdade as outras propriedades

3.8.3 - Simplificação

$p \wedge q \Rightarrow p$

$p \wedge q \Rightarrow q$

3.8.4 - Simplificação Disjuntiva

$(p \vee q) \wedge (p \vee \sim q) \Rightarrow p$

3.8.5 - Absorção

$p \rightarrow q \Rightarrow p \rightarrow (p \wedge q)$

3.8.6 - Regra Modus Ponens

$(p \rightarrow q) \wedge p \Rightarrow q$

3.8.7 - Regra Modus Tollens

$(p \rightarrow q) \wedge \sim q \Rightarrow \sim p$

3.8.8 - Regra do Silogismo Disjuntivo

$(p \vee q) \wedge \sim p \Rightarrow q$

$(p \vee q) \wedge \sim q \Rightarrow p$

3.8.9 - Silogismo Hipotético

$(p \rightarrow q) \wedge (q \rightarrow r) \Rightarrow p \rightarrow r$

3.8.10 - Dilema Construtivo

$((p \rightarrow q) \wedge (r \rightarrow s) \wedge (p \vee r)) \Rightarrow q \vee s$

3.8.11 - Dilema Destrutivo

$((p \rightarrow q) \wedge (r \rightarrow s) \wedge (\sim p \vee \sim s)) \Rightarrow \sim p \vee \sim r$

3.9 Teorema Contra-Recíproco

A proposição "$p(x) \Rightarrow q(x)$ é verdadeira se, e somente se "$\sim q(x) \Rightarrow \sim p(x)$" é verdadeira. Assim, afirmar "**Se p, então q**" é o mesmo que afirmar "**se ~q, então ~p**". Portanto, "$p(x) \Rightarrow q(x)$" é equivalente a "$\sim q(x) \Rightarrow \sim p(x)$".

Exemplo: A sentença "Se comeu, então matou a fome" é equivalente a "Se não matou a fome, então não comeu".

3.10 Relação Entre Implicações

a) Implicações recíprocas

$p \Rightarrow q$ e $q \Rightarrow p$

Duas proposições recíprocas não são logicamente equivalentes, uma pode ser verdadeira sem que a outra o seja.

b) Implicações Inversas

$p \Rightarrow q$ e $\sim p \Rightarrow \sim q$

Duas proposições inversas não são logicamente equivalentes, uma pode ser verdadeira sem que a outra o seja.

c) Implicações Contrapositivas

$p \Rightarrow q$ e $\sim q \Rightarrow \sim p$

Duas proposições contrapositivas são logicamente equivalentes, sempre que uma é verdadeira, a outra também será.

3.11 Equivalencias entre Implicações

a) $p \Rightarrow q \Leftrightarrow \sim q \Rightarrow \sim p$

b) $p \Rightarrow q \Leftrightarrow \sim (p \wedge \sim q)$

c) $p \Rightarrow q \Leftrightarrow \sim p \vee q$

3.12 Conceito de equivalência lógica

O símbolo "$\leftrightarrow$" representa uma **operação** entre duas proposições, resultando uma nova proposição. O símbolo "$\Leftrightarrow$" indica apenas uma **relação** entre duas proposições dadas.

<u>Definição</u>: Diz-se que uma proposição P(p, q, ...) é **logicamente equivalente** a uma proposição Q(p, q, ...), se as tabelas-verdade destas duas proposições são idênticas.

3.13 Propriedades da equivalência lógica: reflexiva, simétrica e transitiva

a) A condição necessária e suficiente para que uma eqüivalência $\mathbf{p \Leftrightarrow q}$ seja verdadeira é que a bicondicional $\mathbf{p \leftrightarrow q}$ seja uma tautologia;

b) Propriedade Reflexiva (R): **p ⇔ p;**

c) Propriedade Simétrica (S): Se **p ⇔ q,** então **q ⇔ p.**

d) Propriedade Transitiva (T): Se **p ⇔ q** e **q ⇔ r**, então **p ⇔ r**.

3.14 Demonstração de equivalência lógica, comparando-se tabelas-verdade

Dadas as proposições $p \rightarrow q$ e $\sim p \vee q$, a relação de equivalência lógica entre elas é denotada por $p \rightarrow q \Leftrightarrow \sim p \vee q$. Podem-se mostrar as tabelas-verdade:

p	q	$p \rightarrow q$
V	V	V
V	F	F
F	V	V
F	F	V

p	q	$\sim p$	$\sim p \vee q$
V	V	F	V
V	F	F	F
F	V	V	V
F	F	V	V

Como as tabelas-verdade possuem resultados iguais, logo a relação é verdadeira.

3.15 Demonstração de equivalência lógica, substituindo-se a relação pelo conectivo

Dadas as proposições $p \rightarrow q$ e $\sim p \vee q$, a relação de equivalência lógica entre elas é denotada por $p \rightarrow q \Leftrightarrow \sim p \vee q$.

Podemos mostrar na tabela-verdade:

p	q	$p \rightarrow q$	$\sim p$	$\sim p \vee q$	$(p \rightarrow q) \leftrightarrow (\sim p \vee q)$
V	V	V	F	V	V
V	F	F	F	F	V
F	V	V	V	V	V
F	F	V	V	V	V

Como o resultado final da tabela-verdade é uma Tautologia, logo a relação é verdadeira.

3.16 Equivalencia entre Sentenças Abertas

Diz-se que uma sentença aberta é equivalente a uma outra sentença aberta quando o conjunto-verdade da primeira sentença está contido no conjunto-verdade da segunda e o conjunto-verdade do segunda está contido no conjunto-verdade da primeira. Eqüivale dizer que os conjuntos-verdade são iguais.

Exemplo: Julgar a sentença $\mathbf{2x + 3 = x + 5 \Leftrightarrow 7x - 3 = 5x + 1}$

Resolução:

Determinando o conjunto-verdade da primeira sentença aberta:

$2x + 3 = x + 5$
temos que $x = 2$
logo, $V_1 = \{ 2 \}$
Determinando o conjunto-verdade da segunda sentença aberta:
$7x - 3 = 5x + 1$
$x = 2$

logo, $V_2 = \{ 2 \}$
Podemos observar que $v_1 \subset v_2$ e $v_2 \subset v_1$, logo $V_1 = V_2$.
Portanto, podemos dizer que a eqüivalência é verdadeira,
logo a sentença $\mathbf{2x + 3 = x + 5 \Leftrightarrow 7x - 3 = 5x + 1}$ está correta.

3.17 Propriedades das Equivalências Lógicas

São:

1) A condição necessária e suficiente para que uma eqüivalência $\mathbf{p \Leftrightarrow q}$ seja verdadeira é que a bicondicional $\mathbf{p \leftrightarrow q}$ seja uma tautologia;
2) Propriedade Reflexiva: $\mathbf{p \Leftrightarrow p}$**;**
3) Propriedade Simétrica: Se $\mathbf{p \Leftrightarrow q}$**,** então $\mathbf{q \Leftrightarrow p}$**.**
4) Propriedade Transitiva: Se $\mathbf{p \Leftrightarrow q}$ e $\mathbf{q \Leftrightarrow r}$, então $\mathbf{p \Leftrightarrow r}$.

3.18 Equivalências Notáveis

3.18.1 - Dupla Negação

$\sim\sim p \Leftrightarrow p$

p	~p	~~p
V	F	V
F	V	F

Observe que os valores lógicos das colunas p e ~~p são iguais. Logo a Dupla Negação equivale a afirmação.

3.18.2 - Leis Idempotentes

$p \wedge p \Leftrightarrow p$

$p \vee p \Leftrightarrow p$

Provar, usando tabela-verdade as outras propriedades

3.18.3 - Leis Comutativas

$p \wedge q \Leftrightarrow q \wedge p$

$p \vee q \Leftrightarrow q \vee p$

3.18.4 -Leis Associativas

$p \wedge (q \wedge r) \Leftrightarrow (p \wedge q) \wedge r$

$p \vee (q \vee r) \Leftrightarrow (p \vee q) \vee r$

3.18.5 - Leis de DeMorgan

$\sim(p \wedge q) \Leftrightarrow \sim p \vee \sim q$

$\sim(p \vee q) \Leftrightarrow \sim p \wedge \sim q$

3.18.6 - Leis Distributivas

$p \wedge (q \vee r) \Leftrightarrow (p \wedge q) \vee (p \wedge r)$

$p \vee (q \wedge r) \Leftrightarrow (p \vee q) \wedge (p \vee r)$

3.18.7 - Condicionais

Das proposições:

i) $p \rightarrow q$ (condicional)

ii) $q \rightarrow p$ (recíproca da condicional)

iii) $\sim q \rightarrow \sim p$ (contrapositivo)

iv) $\sim p \rightarrow \sim q$ (recíproca do contrapositivo)

resultam as duas equivalência:

$p \rightarrow q \Leftrightarrow \sim q \rightarrow \sim p$

$q \rightarrow p \Leftrightarrow \sim p \rightarrow \sim q$

3.18.8 - Bicondicionais

$p \leftrightarrow q \Leftrightarrow (p \rightarrow q) \wedge (q \rightarrow p)$

p	**q**	**$p \leftrightarrow q$**	**$p \rightarrow q$**	**$q \rightarrow p$**	**$(p \rightarrow q) \wedge (q \rightarrow p)$**
V	V	V	V	V	V
V	F	F	F	V	F
F	V	F	V	F	F
F	F	V	V	V	V

iguais, logo equivalentes

3.19 Álgebra das proposições.

Para discutir algumas regras que podem ser aplicadas em proposições compostas com o intuito de provar relações de implicação e equivalência lógica, ou mesmo para simplificar uma estrutura de argumentação muito complexa. As propriedades aqui discutidas serão novamente apresentadas em Álgebra Booleana.

Algumas propriedades são válidas tanto para conjunção como para disjunção, enquanto que outras são válidas somente a uma delas, ou em situações em que ambas são envolvidas. O quadro abaixo apresenta as propriedades válidas tanto para disjunção como para conjunção.

<table>
<tr><th colspan="2">Disjunção</th><th colspan="2">Conjunção</th><th>Propriedade</th></tr>
<tr><td colspan="2">$p \wedge p \Leftrightarrow p$</td><td colspan="2">$p \vee p \Leftrightarrow p$</td><td>Idempotente</td></tr>
<tr><td colspan="2">$p \wedge q \Leftrightarrow q \wedge p$</td><td colspan="2">$p \vee q \Leftrightarrow q \vee p$</td><td>Comutativa</td></tr>
<tr><td colspan="2">$(P \wedge p) \wedge r \Leftrightarrow p \wedge (q \wedge r)$</td><td colspan="2">$(p \vee p) \vee r \Leftrightarrow p \vee (q \vee r)$</td><td>Associativa</td></tr>
<tr><td>$p \wedge t \Leftrightarrow p$</td><td>$p \wedge f \Leftrightarrow f$</td><td>$p \vee t \Leftrightarrow t$</td><td>$p \vee f \Leftrightarrow p$</td><td>Identidade</td></tr>
</table>

Destaca-se que, na propriedade da Identidade, tem-se sempre um elemento considerado neutro, pois não influencia na decisão final da proposição, e um elemento absorvente, que por si só, define o resultado final da proposição composta.

Dentre as propriedades que são aplicadas à proposição que contém tanto conjunção como disjunção, encontram-se a distributiva, a de absorção e as famosas regras de De Morgan, apresentadas no quadro abaixo.

Distributiva	Absorção	De Morgan
$p \wedge (q \vee r) \Leftrightarrow (p \wedge q) \vee (p \wedge r)$	$p \wedge (p \vee q) \Leftrightarrow p$	$\sim(p \wedge q) \Leftrightarrow \sim p \vee \sim q$
$p \vee (q \wedge r) \Leftrightarrow (p \vee q) \wedge (p \vee r)$	$p \vee (p \wedge q) \Leftrightarrow p$	$\sim(p \vee q) \Leftrightarrow \sim p \wedge \sim q$

3.20 Exercícios

3.20.1 - Verifique se existe implicação lógica entre as proposições compostas P(p,q,r,...) e Q(p,q,r,...), ou seja, se P(p,q,r,...) $\Rightarrow$ Q(p,q,r,...), usando o método indicado na letra. Para cada letra, deve-se indicar se existe ou não implicação lógica e justificar.

a) Construindo a tabela-verdade, e comparando uma proposição com a outra:

P(p,q,r,...) = q Q(p,q,r,...) = $p \vee q \leftrightarrow p$

b) Construindo a tabela-verdade, e avaliando se a condicional $\leftrightarrow$ é tautológica:

P(p,q,r,...) = $(p \rightarrow q) \wedge q$ e Q(p,q,r,...) = $\sim p$

3.20.2 - Verifique se existe equivalência lógica entre as proposições compostas P(p,q,r,...) e Q(p,q,r,...), ou seja, se P(p,q,r,...) $\Leftrightarrow$ Q(p,q,r,...), usando o método indicado na letra. Para cada letra, deve-se indicar se existe ou não equivalência lógica e justificar.

a) Construindo a tabela-verdade, e comparando uma proposição com a outra:

P(p,q,r,...) = $p \vee q$ e Q(p,q,r,...) = $(p \rightarrow q) \rightarrow p$

b) Construindo a tabela-verdade, e avaliando se a bi-condicional $\leftrightarrow$ é tautológica:

P(p,q,r,...) = $p \wedge q$ e Q(p,q,r,...) = $(p \vee q) \wedge \sim (p \wedge q)$

3.20.3 - Julgar cada uma das seguintes proposições (dizer se são verdadeiras ou falsas as relações):

a) $\sim p \wedge \sim p \Rightarrow \sim p$
b) $\sim p \vee \sim(p \vee q) \Rightarrow p \vee q$
c) $(p \wedge \sim q) \vee (p \wedge \sim q) \Rightarrow p \wedge \sim q$
d) $(p \rightarrow q) \vee r \Rightarrow (p \wedge \sim r) \rightarrow (q \wedge r)$
e) $(p \rightarrow \sim r) \wedge (q \rightarrow \sim p) \Rightarrow r \rightarrow (p \vee q)$
f) $(p \rightarrow q) \rightarrow r \Rightarrow (p \wedge \sim r) \rightarrow \sim q$

3.20.4 - Julgar cada uma das seguintes proposições (dizer se são verdadeiras ou falsas as relações):

a) $\sim p \wedge \sim p \Leftrightarrow \sim p$
b) $\sim p \vee \sim(p \vee q) \Leftrightarrow p \vee q$
c) $(p \wedge \sim q) \vee (p \wedge \sim q) \Leftrightarrow p \wedge \sim q$
d) $(p \rightarrow q) \vee r \Leftrightarrow (p \wedge \sim r) \rightarrow (q \wedge r)$
e) $(p \rightarrow \sim r) \wedge (q \rightarrow \sim p) \Leftrightarrow r \rightarrow (p \vee q)$
f) $(p \rightarrow q) \rightarrow r \Leftrightarrow (p \wedge \sim r) \rightarrow \sim q$

3.20.5 - Considere as proposições: p, q, r, s dadas por:

p: $5 = 8$ q: $4 < 5$ r: $9 > 7$ s: $8 < 10$

e diga se são verdadeiras ou falsas as relações abaixo:

a) $r \Leftrightarrow s$
b) $r \Leftrightarrow q \vee s$
c) $\sim p \Leftrightarrow s \vee r$
d) $p \Leftrightarrow q$
e) $\sim r \Leftrightarrow (q \wedge \sim s)$
f) $\sim r \Leftrightarrow (s \vee \sim q)$
g) $p \Rightarrow q$
h) $\sim r \Rightarrow q \vee s$
i) $\sim r \Rightarrow s \vee r$
j) $(p \rightarrow p) \Rightarrow (q \wedge r)$
k) $\sim r \Rightarrow (q \wedge \sim s)$
l) $\sim r \Rightarrow (s \vee \sim q)$

3.20.6 - Considere as proposições: p, q, r, s dadas por:

p: $7 + 2 = 9$ q: $(7 + 2)^2 = 81$ r: $2^0 = 1$ s: $0^2 = 2$

e dê o valor lógico (verdadeiro ou falso) das relações abaixo:

a) $r \Leftrightarrow s$ b) $r \Leftrightarrow q$ c) $\sim p \Leftrightarrow s$
d) $p \Leftrightarrow q$ e) $\sim r \Leftrightarrow (q \wedge \sim s)$ f) $\sim r \Leftrightarrow (s \vee \sim q)$
g) $p \Rightarrow q$ h) $\sim r \Rightarrow q$ i) $\sim r \Rightarrow s$
j) $(p \rightarrow p) \Rightarrow (q \wedge r)$ k) $\sim r \Rightarrow (q \wedge \sim s)$ l) $\sim r \Rightarrow (s \vee \sim q)$

3.20.7 - Usando as equivalências tautológicas, mostrar que as proposições abaixo podem ser escritas usando os conectivos $\wedge$, $\vee$ e ~. Depois simplifique as expressões o máximo possíveis.

a) $(r \rightarrow (q \rightarrow p)$ b) $q \vee r \rightarrow (q \rightarrow p)$ c) $p \wedge r \rightarrow (q \rightarrow r)$
d) $(r \wedge q) \rightarrow (\sim q \rightarrow p)$ e) $(p \wedge q) \rightarrow (p \wedge \sim q)$ f) $(p \wedge r) \rightarrow ((q \wedge p) \rightarrow r)$
g) $\sim (p \wedge q) \rightarrow p$ h) $(p \wedge q) \rightarrow r \rightarrow (p \vee r)$ i) $\sim (p \wedge q) \rightarrow (p \wedge q)$

3.20.8 - Usando as equivalências tautológicas, mostrar que as proposições abaixo podem ser escritas usando os conectivos: $\vee$ e ~. Depois simplifique as expressões o máximo possíveis.

a) $(r \wedge q) \rightarrow (q \rightarrow p)$ b) $(p \wedge q) \rightarrow p$ c) $(p \wedge r) \rightarrow ((q \wedge p) \rightarrow r)$
d) $(r \rightarrow (q \rightarrow p)$ e) $q \vee r \rightarrow (q \rightarrow p)$ f) $p \wedge r \rightarrow (q \rightarrow r)$

3.20.9 - Usando as equivalências tautológicas, simplificar as proposições abaixo:

a) $(\sim (p \wedge q) \wedge r) \rightarrow (\sim p \vee \sim q)$ b) $(p \wedge q) \rightarrow p \vee p$ c) $(p \wedge q) \rightarrow ((q \wedge p) \rightarrow r)$
d) $\sim (p \wedge q) \vee \sim p \rightarrow (\sim p \vee q)$ e) $(p \wedge q) \rightarrow (q \rightarrow p)$ f) $p \wedge (\sim p \vee q) \Rightarrow p \wedge q$
g) $(p \wedge q) \rightarrow r \rightarrow (p \vee r)$ h) $\sim (p \wedge q) \rightarrow p$ i) $\sim (p \wedge q) \rightarrow (p \wedge q)$
j) $(p \wedge q) \rightarrow ((p \wedge r) \rightarrow (p \vee q))$ k) $\sim (p \wedge r) \rightarrow ((r \rightarrow (q \wedge r))$

4 MÉTODOS PARA DETERMINAÇÃO DA VALIDADE DE FÓRMULAS

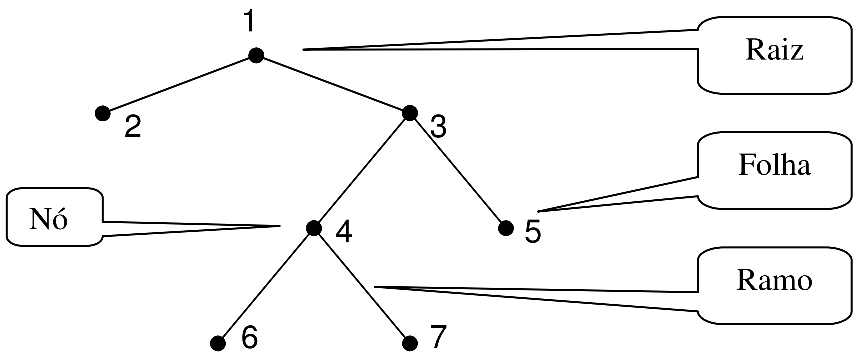
São métodos usados para determinar ou verificar se a fórmula lógica proposicional é válida ou quais os valores lógicos apresentados.

4.1 Método da Tabela-Verdade

O método da Tabela-verdade é o método da força bruta utilizado na determinação da validade de fórmulas da lógica proposicional. No método da tabela-verdade são consideradas todas as possibilidades de valores de verdade associados a esses símbolos proposicionais. Na coluna de resultado, observamos uma das três soluções: Tautologia, Contradição ou Contingência.

4.2 Árvore Semântica

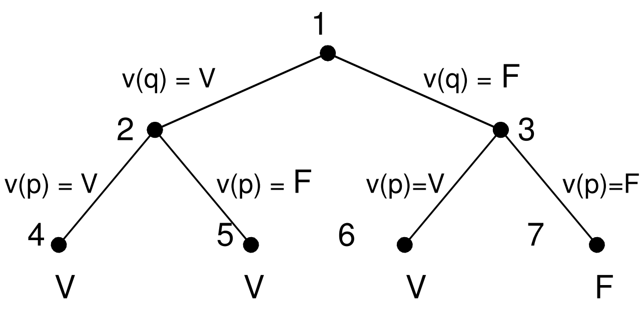
Definição 1: Uma árvore é um conjunto de **nós** ou **vértices** ligados por **arestas** ou **ramos**, conforme é indicado na figura abaixo. Os nós são rotulados por números inteiros, e os nós finais (na figura: 2, 6, 7 e 5) são denominados de folhas.



Definição 2: Uma árvore semântica é uma árvore na qual os vértices internos representam proposições, as arestas representam os valores lógicos de uma proposição e as folhas representam os resultados finais e que serve para determinar a validade de uma fórmula (ou proposição) a partir da estrutura de dados do tipo árvore.

4.2.1 - Exemplos:

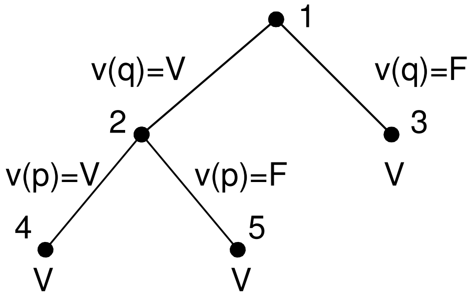
1) Dada a proposição “p ∨ q”, determinar se a proposição é tautologia, contradição ou contingência.



p	q	p ∨ q
V	V	V
V	F	V
F	V	V
F	F	F

portanto, é uma contingência.

2) Dada a proposição "p ∨ ~ (p ∧ q)", determinar se a proposição é tautologia, contradição ou contingência.



nó 2: p ∨ ~ (p ∧ q)
? ? V

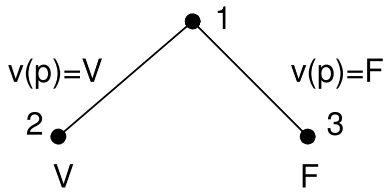
nó 3: p ∨ ~ (p ∧ q)
? ? F
F
V
V

nó 4: p ∨ ~ (p ∧ q)
V V V
V
F
V

nó 5: p ∨ ~ (p ∧ q)
F F V
F
V
V

CONCLUSÃO: como todas as folhas tem valor lógico V (verdade), então a fórmula proposicional é tautologia.

3) Dada a proposição "p ∨ (p ∧ ~ q)", determinar se a proposição é tautologia, contradição ou contingência.



nó 2: p ∨ (p ∧ ~ q)
V V ?
V ?
V

nó 3: p ∨ (p ∧ ~q)
F F ?
F F
F

CONCLUSÃO: Como aparecem V e F, logo temos uma contingência.

4.3 Negação ou Absurdo

Neste método, considera-se inicialmente a negação daquilo que pretende-se demonstrar. Assim, dada uma fórmula proposicional, para demonstrar a sua validade (tautologia), supõe-se que a fórmula não é uma tautologia. A partir desta suposição, deve-se utilizar um conjunto de deduções corretas e concluir um fato contraditório ou absurdo.

Como o resultado é um absurdo, conclui-se que a suposição inicial é falsa. Em outras palavras, se a suposição inicial diz que a fórmula não é uma tautologia, deve-se concluir após obter o absurdo que a não validade da fórmula é um absurdo, logo a fórmula é uma tautologia.

Validade	Supor	Ocorrência
Tautologia	F	Absurdo
Contradição	V	Absurdo
Contingência	V e F	Não ocorrer nenhum Absurdo

4.3.1 - Exemplos:

1) Demonstração de uma tautologia: $p \vee \sim(p \wedge q)$

$p \quad \vee \quad \sim (\quad p \quad \wedge \quad q)$

F

F F

V

V V

a negação da tautologia é "F"

É absurdo, pois a proposição p não pode ser "V" e "F" ao mesmo tempo!

absurdo

Como o valor lógico de p é "V" e "F" ao mesmo tempo, ocorreu um absurdo, então a fórmula é uma tautologia.

Pode-se provar pela tabela verdade que:

p	q	$p \wedge q$	$\sim (p \wedge q)$	$p \vee \sim (p \wedge q)$
V	V	V	F	V
V	F	F	V	V
F	V	F	V	V
F	F	F	V	V

2) Demonstração de uma contradição: $p \leftrightarrow \sim p$

$p \quad \leftrightarrow \quad \sim \quad p$

V

a negação da contradição é "V"

temos duas possibilidades:

a) $p \quad \leftrightarrow \quad \sim \quad p$

V

V V

V F

absurdo

b) $p \quad \leftrightarrow \quad \sim \quad p$

V

F F

F V

absurdo

É absurdo, pois a proposição p não pode ser "V" e "F" ao mesmo tempo!

Como ocorreu absurdo nas duas possibilidades, podemos concluir que a formula é contraditória.

Pode-se provar pela tabela verdade que é contradição.

3) Demonstração de uma tautologia: $(p \rightarrow q) \leftrightarrow (\sim p \rightarrow \sim q)$

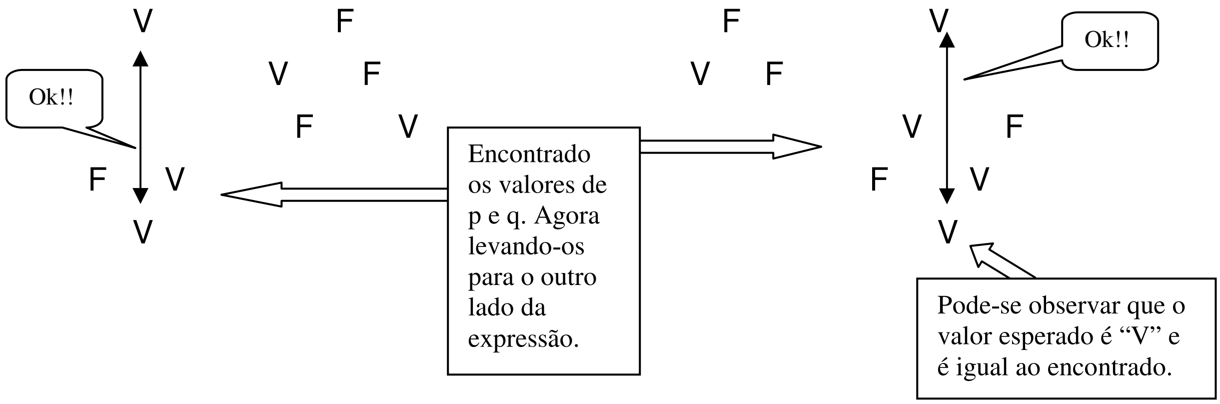
$(p \rightarrow q) \leftrightarrow (\sim p \rightarrow \sim q)$

F ⟵ a negação da tautologia é “F”

Temos duas possibilidades

a) $(p \rightarrow q) \leftrightarrow (\sim p \rightarrow \sim q)$ b) $(p \rightarrow q) \leftrightarrow (\sim p \rightarrow \sim q)$

F F



Como não ocorreu nenhum absurdo, podemos concluir que a fórmula não é uma tautologia, e nada além disto pode ser afirmado.

Usando a tabela-verdade, podemos comprovar o resultado:

p	q	$p \rightarrow q$	$\sim p$	$\sim q$	$\sim p \rightarrow \sim q$	$(p \rightarrow q) \leftrightarrow (\sim p \rightarrow \sim q)$
V	V	V	F	F	V	V
V	F	F	F	V	V	V
F	V	V	V	F	F	F
F	F	V	V	V	V	V

Como podemos ver, a fórmula é uma contingência.

4.4 Exercícios:

4.4.1 - Determinar, usando árvore semântica, quais das proposições abaixo são tautologia, contradição ou contingência.

a) $(p \vee (p \wedge q)) \leftrightarrow p$
b) $(\sim p \wedge \sim q) \vee (p \rightarrow q)$
c) $(p \vee \sim q) \leftrightarrow ((p \rightarrow r) \vee q)$
d) $(\sim p \vee \sim q) \leftrightarrow (p \wedge q)$
e) $p \vee (\sim p \leftrightarrow q)$
f) $(\sim p \vee q) \leftrightarrow (p \rightarrow q)$

4.4.2 - Determinar, usando árvore semântica, a validade da relação abaixo:

a) $p \wedge q \rightarrow p$
b) $(p \wedge q) \rightarrow (p \wedge \sim q)$
c) $\sim p \wedge \sim p \leftrightarrow \sim p$
d) $\sim (\sim (p \vee q)) \leftrightarrow p \vee q$
e) $p \vee q \leftrightarrow p$

4.4.3 - Demonstrar, usando o método da negação ou absurdo que a fórmula é uma contradição.

a) $(\sim p \vee \sim q) \leftrightarrow (p \wedge q)$
b) $p \vee (\sim p \leftrightarrow q)$
c) $\sim((p \rightarrow q) \rightarrow ((p \rightarrow q) \vee r))$
d) $(\sim p \vee q) \leftrightarrow (p \rightarrow q)$
e) $(p \vee (p \wedge q)) \leftrightarrow p$
f) $(\sim p \wedge \sim q) \vee (p \rightarrow q)$

4.4.4 - Demonstrar, usando o método da negação ou absurdo que a fórmula é uma tautologia.

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

4.4.5 - Usando o método da negação ou absurdo para mostrar que as proposições são uma tautologia, contradição ou contingência.

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

4.4.6 - Determinar se as proposições a seguir são tautologia, contradição ou contingência, usando os seguintes métodos: Tabela-verdade, Árvore semântica e Negação ou absurdo.

a) $\sim (p \wedge q) \rightarrow \sim p \vee \sim q$
b) $(p \rightarrow q \vee r) \wedge q \rightarrow r \vee p$
c) $(p \leftrightarrow q) \vee p \rightarrow q$
d) $\sim (p \rightarrow q) \rightarrow (p \wedge \sim q)$
e) $\sim (p \wedge q) \leftrightarrow \sim (\sim p \wedge q)$
f) $p \wedge (p \vee q) \rightarrow p$

4.4.7 - Verificar se as proposições são tautologia, contradição ou contingência usando os métodos: tabela-verdade, árvore semântica e Negação ou absurdo.

a) $\sim (p \wedge q) \leftrightarrow \sim p \vee \sim q$
b) $\sim (p \vee q) \rightarrow \sim p \wedge \sim q$
c) $\sim (r \wedge p) \vee q \leftrightarrow \sim (p \wedge q) \vee r$
d) $(p \leftrightarrow q) \vee r \rightarrow (p \rightarrow q) \vee r$
e) $(p \rightarrow \sim r) \wedge (q \rightarrow \sim p) \leftrightarrow r \rightarrow (p \vee q)$
f) $(p \rightarrow q) \rightarrow r \rightarrow (p \wedge \sim r) \rightarrow \sim q$
g) $\sim (p \wedge q) \leftrightarrow \sim (\sim p \wedge q)$
h) $\sim (p \rightarrow q) \rightarrow (p \wedge \sim q)$

5 ÁLGEBRA BOOLEANA

O matemático inglês George Boole (1815-1864) desenvolveu a álgebra booleana em 1854. Boole estava interessado em regras “algébricas” para o raciocínio lógico, semelhante às regras algébricas para o raciocínio numérico. É uma maneira formal para escrever e manipular proposições lógicas como se fosse uma fórmula algébrica. Em 1938, Claude Shannon demonstra a aplicação da álgebra booleana na análise de circuitos à relês.

A álgebra booleana, também conhecida como Álgebra de Boole, determina regras algébricas para o raciocínio lógico assim como existem regras para o raciocínio numérico. Em outras palavras, a álgebra booleana é uma maneira de se utilizar técnicas algébricas para lidar com expressões lógicas. Essas regras capturam o funcionamento das operações lógicas E (AND), OU (OR) e NÃO (NOT), bem como das operações da teoria de conjuntos: soma, produto e complemento. Na Matemática e na Ciência da Computação, a álgebra booleana é o fundamento da matemática computacional, baseada em números binários. Com a álgebra booleana, introduziu-se o conceito de portas lógicas que só processam dois tipos de entradas - verdade ou falsidade, sim ou não, aberto ou fechado, um ou zero. Atualmente, todos os computadores usam a Álgebra de Boole na construção de circuitos digitais que contêm combinações de portas lógicas que produzem os resultados das operações utilizando lógica binária.

5.1 Conceitos preliminares.

A Álgebra booleana é definida como sendo um sistema que opera com funções booleanas aplicadas a variáveis de entradas. Essas variáveis podem receber somente dois valores possíveis: 1 ou 0, aberto ou fechado, sim ou não. Entre as operações básicas estão a adição lógica, ou operação OR, com o símbolo (+), a multiplicação lógica, ou operação AND, com o símbolo (.) e finalmente, a complementação lógica, ou operação NOT, que tem como símbolo uma barra sobreposta ou o ‘ (apóstrofo simples).

5.1.1 – Variáveis e Expressões na Álgebra de Boole

As variáveis booleanas, que são representadas através de letras, podem assumir apenas dois valores: 0 e 1. Expressão booleana é uma expressão matemática cujas variáveis são booleanas e seu resultado assumirá apenas dois valores: 0 e 1.

5.1.2 – Postulados

a) Postulado da Complementação

Chama-se de **A’** o complemento de A. Dizemos “A BARRA”. Poder ser: $\overline{\mathbf{A}}$

i) se A = 0 então A’ = 1

ii) se A = 1 então A’ = 0

através do postulado da complementação, pode-se estabelecer a identidade: A’’ = A

b) Postulado da Adição

i) 0 + 0 = 0

ii) 0 + 1 = 1

iii) 1 + 0 = 1

iv) 1 + 1 = 1

Lembrar que: p ∨ q

através deste postulado, pode-se estabelecer as seguintes identidades:

i) $A + 0 = A$

ii) $A + 1 = 1$

iii) $A + A = A$

iv) $A + A' = 1$

c) Postulado da Multiplicação

i) $0 \bullet 0 = 0$

ii) $0 \bullet 1 = 0$

iii) $1 \bullet 0 = 0$

iv) $1 \bullet 1 = 1$

Lembrar que: $p \wedge q$

através deste postulado, pode-se estabelecer as seguintes identidades:

i) $A \bullet 0 = 0$

ii) $A \bullet 1 = A$

iii) $A \bullet A = A$

iv) $A \bullet A' = 0$

5.2 Teoremas da Álgebra de Boole.

5.2.1 - Introdução

Na álgebra de Boole, estão definidas as operações binárias (+) e (.), e uma operação unitária (‘), e os elementos 0 e 1. Baseado nisso, foram criados diversos teoremas em que, dada uma função booleana, pode-se alterá-la (na maioria das vezes buscando simplificá-la), sem perder o seu valor lógico. Ou seja, uma mesma expressão booleana pode ser escrita de diversas formas sem que seja perdido o sentido desta. Por exemplo, a expressão “A” é igual às expressões “A + A.B” ou “A.B + A.B’” ou ainda “(A + B).(A + B’)”. Com isso, dada uma expressão booleana, pode-se aplicar diversos teoremas a ela, alterando sua estrutura, mas sem alterar o valor lógico dela. Estes teoremas foram apresentados na álgebra das proposições com o mesmo intuito: mostrar como duas proposições compostas, apesar de estruturalmente diferentes, são logicamente equivalentes. Neste tópico tem-se:

- Teoremas da álgebra de Boole.
- Utilização dos teoremas da álgebra de Boole para verificar a equivalência de duas expressões booleanas.
- Utilização dos teoremas da álgebra de Boole para simplificar expressões booleanas complexas.

5.2.2 – Condições

a) Neste primeiro momento, serão usados os operadores:

Lógica Booleana	Lógica Proposicional	conectivo	Exemplo Booleano	Exemplo Proposicional
$\bullet$	$\wedge$	E	$p \bullet q$	$p \wedge q$
+	$\vee$	OU	$p + q$	$p \vee q$
– ou ’	~	NÃO	$(p + q)'$	$\sim (p \vee q)$

b) 1 + 1 = 1, visto que não existe o 2 na lógica booleana.

Define-se, por convenção que 1 + 1 = 1, porque em lógica booleana o 1 corresponde a V (verdade) e como o valor lógico de V ∨ V é V no cálculo preposicional, então temos 1 como resultado.

c) As regras para formar expressões válidas em álgebra booleana são as mesmas para formar fórmulas lógicas no cálculo proposicional:

   i) Uma variável é uma expressão booleana: **p**;

   ii) Se p é uma expressão booleana, então também é o **p’**;

   iii) Se p e q são expressões booleanas, então também são o **p • q** e **p + q**;

   iv) Se p é uma expressão booleana, então também é o **(p)**.

d) As regras esquerda-para-direita e a precedência dos operadores para avaliação de fórmulas lógicas também se aplicam às expressões booleanas, assim:

( ) ↓ maior precedência

’

•

\+ menor precedência

e) Em álgebra booleana, quando duas expressões são equivalentes, usamos o sinal de igualdade para representar. Exemplo:

p • q + p • r + q’ • r = p • q + q’ • r

isto afirma que os valores de ambos os lados da equação são iguais para todos os possíveis valores das variáveis. Provando a igualdade das expressões, temos:

p	q	r	q’	p•q	p•r	q’•r	p•q + p•r	p•q + p•r + q’•r	p•q + q’•r
0	0	0	1	0	0	0	0	0	0
0	0	1	1	0	0	1	0	1	1
0	1	0	0	0	0	0	0	0	0
0	1	1	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0
1	0	1	1	0	1	1	1	1	1
1	1	0	0	1	0	0	1	1	1
1	1	1	0	1	1	0	1	1	1

iguais

5.2.3 - Propriedades

Uma álgebra de Boole é um conjunto S no qual estão definidos duas operações binárias (+ e •) e uma operação unária (’), e que contém dois elementos distintos (0 e 1), tais que as suas propriedades são válidas, quaisquer que sejam A, B, C pertencentes a S.

a) Propriedade Comutativa

   i) A + B = B + A (Adição)

   ii) A • B = B • A (multiplicação)

Usando a tabela-verdade para mostrar a validade:

A	B	A + B	B + A
1	1	1	1
1	0	1	1
0	1	1	1
0	0	0	0

iguais

A	B	A • B	B • A
1	1	1	1
1	0	0	0
0	1	0	0
0	0	0	0

iguais

b) Propriedade Associativa

i) $A + (B + C) = (A + B) + C$ (adição)

ii) $A \bullet (B \bullet C) = (A \bullet B) \bullet C$ (multiplicação)

c) Propriedade Distributiva

$A \bullet (B + C) = A \bullet B + A \bullet C$

5.2.4 – Teoremas de De Morgan

i) 1º Teorema: O complemento do produto é igual à soma dos complementos.

$(A \bullet B)' = A' + B'$

ii) 2º Teorema: O complemento da soma é igual ao produto dos complementos.

$(A + B)' = A' \bullet B'$

5.2.5 – Identidades Auxiliares

a) $A + A \bullet B = A$

b) $A + A' \bullet B = A + B$

c) $(A + B) \bullet (A + C) = A + B \bullet C$

A	B	A • B	A + A • B
1	1	1	1
1	0	0	1
0	1	0	0
0	0	0	0

iguais

A	B	A'	A' • B	A + A' • B	A + B
1	1	0	0	1	1
1	0	0	0	1	1
0	1	1	1	1	1
0	0	1	0	0	0

iguais

5.2.6 - Tabela com as propriedades da Álgebra de Boole

PROPRIEDADES DA ALGEBRA DE BOOLE				
Comutativa:	A + B = B + A	A • B = B • A	P. Adição	P. Multiplicação
Associativa:	A + (B + C) = (A + B) + C	A • (B • C) = (A • B) • C	A + 0 = A	A • 0 = 0
Distributiva:	A • (B + C) = A • B + A • C		A + 1 = 1	A • 1 = A
Teorema de DeMorgan:	(A • B)' = A' + B'	(A + B)' = A' • B´	A + A = A	A • A = A
Identidades Auxiliares	A + A • B = A		A + A'= 1	A • A' = 0
A + A' • B = A + B	(A + B) • (A + C) = A + B • C	Postulado da Complementação	A = 0 → A'= 1	A = 1 → A'= 0

5.3 Métodos para minimização de funções: Método Algébrico e Mapa de Karnaugh.

5.3.1 – Método Algébrico

Uma função booleana qualquer pode ser representada por mais de uma expressão booleana. Para reduzir uma expressão booleana em uma outra equivalente mais simples devemos usar as propriedades da álgebra de Boole, já que elas expressam a equivalência de expressões booleanas. Portanto, usando os conceitos de álgebra de Boole pode-se simplificar expressões lógicas.

5.3.2 – Exemplos de resolução pelo método algébrico:

A seguir exemplos de resolução pelo método algébrico.

1) Simplificar a expressão: $S = A' \bullet B' + A' \bullet B$

   usando a propriedade distributiva:
   $S = A' \bullet (B' + B)$
   usando a identidade da adição: $X' + X = 1$, temos:
   $S = A' \bullet 1$
   pela identidade da multiplicação: $X \bullet 1 = X$, temos:
   $S = A'$

2) Simplificar a expressão: $S = (A + B + C) \bullet (A' + B' + C)$

   usando a propriedade distributiva, temos:
   $S = A \bullet A' + A \bullet B' + A \bullet C + A' \bullet B + B \bullet B' + B \bullet C + A' \bullet C + B' \bullet C + C \bullet C$
   do identidade da multiplicação: $X \bullet X' = 0$, temos:
   $S = A \bullet B' + A \bullet C + A' \bullet B + B \bullet C + A' \bullet C + B' \bullet C + C \bullet C$
   do identidade da multiplicação: $X \bullet X = X$, temos:
   $S = A \bullet B' + A \bullet C + A' \bullet B + B \bullet C + A' \bullet C + B' \bullet C + C$
   colocando C em evidencia (propriedade distributiva), temos:
   $S = A \bullet B' + C \bullet (A + B + A' + B' + 1) + A' \bullet B$
   usando a identidade da adição: $X + X' = 1$ e $X + 1 = 1$, temos:
   $S = A \bullet B' + C \bullet 1 + A' \bullet B$
   usando a identidade da multiplicação: $X \bullet 1 = X$, temos:
   $S = A \bullet B' + C + A' \bullet B$
   portanto:
   $S = (A + B + C) \bullet (A' + B' + C) = A \bullet B' + C + A' \bullet B$

3) Provar que a expressão $(A + B) \bullet (A + C) = A + B \bullet C$ é válida:

   Vamos desenvolver o lado esquerdo, logo: $(A + B) \bullet (A + C)$

   Usando a propriedade distributiva: $A \bullet A + A \bullet B + A \bullet C + B \bullet C$

   Usando a identidade da multiplicação $X \bullet X = X$, temos: $A + A \bullet B + A \bullet C + B \bullet C$

   Usando a Propriedade distributiva: $A \bullet (1 + B + C) + B \bullet C$

   Usando a identidade da soma $X + 1 = 1$, temos: $A \bullet (1) + B \bullet C$

   Usando a identidade da multiplicação $X \bullet 1 = X$, temos: $A + B \bullet C$

   que é a expressão procurada, logo: $(A + B) \bullet (A + C) = A + B \bullet C$ é válida.

4) Dada a expressão $S = A \bullet B \bullet C + A \bullet C' + A \bullet B'$, simplificá-la.

colocando A em evidencia (p. distributiva), temos:
$S = A \bullet (B \bullet C + C' + B')$
aplicando a propriedade associativa, temos:
$S = A \bullet (B \bullet C + (C' + B'))$
aplicando a identidade: $X'' = X$, temos:
$S = A \bullet (B \bullet C + (C' + B')'')$
aplicando o 2º teorema de DeMorgan, temos:
$S = A \bullet (B \bullet C + (C'' \bullet B'')')$
aplicando a identidade: $X'' = X$, temos:
$S = A \bullet (B \bullet C + (C \bullet B)')$
usando a propriedade comutativa, temos:
$S = A \bullet (B \bullet C + (B \bullet C)')$
chamando $B \bullet C$ de Y, logo $(B \bullet C)' = Y'$
$S = A \bullet (Y + Y')$
pela identidade da adição: $X + X' = 1$, temos:
$S = A \bullet 1$
pela identidade da multiplicação: $X \bullet 1 = X$, logo:
$S = A$
portanto, $S = A \bullet B \bullet C + A \bullet C' + A \bullet B' = A$

5) Usando as propriedades da Álgebra de Boole, provar que a expressão

$\mathbf{(A + B) \bullet (A' + C) = A \bullet C + A' \bullet B}$ é válida, indicando a propriedade usada em cada passagem.

$(A + B) \bullet (A' + C) = A \bullet C + A' \bullet B$
Distributiva
$A \bullet A' + A \bullet C + A' \bullet B + B \bullet C$
Identidade da Multiplicação: $X \bullet X' = 0$
$A \bullet C + A' \bullet B + B \bullet C$
Identidade da Soma: $1 = X + X'$
$A \bullet C + A' \bullet B + B \bullet C \bullet (A + A')$
Distributiva
$A \bullet C + A' \bullet B + A \bullet B \bullet C + A' \bullet B \bullet C$
Distributiva
$A \bullet C \bullet (1 + B) + A' \bullet B \bullet (1 + C)$
Identidade da Soma: $1 + X = 1$
$A \bullet C \bullet 1 + A' \bullet B \bullet 1$
Identidade da Multiplicação: $X \bullet 1 = X$
$A \bullet C + A' \bullet B$

5.3.3 – Mapa de Veitch-Karnaugh

Simplifica-se expressões booleanas usando as propriedades, postulados e identidades da álgebra de Boole, no entanto não há um conjunto de procedimentos a serem seguidos, funciona por tentativa e erro, cada solução é diferente da anterior. Uma opção de simplificação é através dos Mapas de Veitch-Karnaugh ou simplesmente Mapas de Karnaugh. Para usar os diagramas de Veitch-Karnaugh é necessário seguir alguns procedimentos.

	$\overline{B}$	B
$\overline{A}$	$\overline{A} \bullet \overline{B}$	$\overline{A} \bullet B$
A	$A \bullet \overline{B}$	$A \bullet B$

Número De Células: 2^N

onde N é o número de variáveis

$2^2 = 4$ células

	$\overline{B}$	$\overline{B}$	B	B
$\overline{A}$	$\overline{A}\,\overline{B}\,\overline{C}$	$\overline{A}\,\overline{B}\,C$	$\overline{A}\,B\,C$	$\overline{A}\,B\,\overline{C}$
A	$A\,\overline{B}\,\overline{C}$	$A\,\overline{B}\,C$	$A\,B\,C$	$A\,B\,\overline{C}$
	$\overline{C}$	C	C	$\overline{C}$

	B’	B’	B	B
A’	A’B’C’	A’B’C	A’B C	A’B C’
A	A B’C’	A B’C	A B C	A B C’
	C’	C	C	C’

5.3.4 – Exemplo de resolução pelo Mapa de Karnaugh:

Dada a função booleana S = F(A,B) = A’ • B’ + A • B, será montado o mapa de Karnaugh.

Solução: Como tem duas proposições (ou variáveis), tem-se 2^2 células, montando o mapa: As parcelas que temos na expressão são transcritas para o mapa com o valor igual a 1:

	$\overline{B}$	B
$\overline{A}$	1	
A		1

completando com zeros as células vazias:

	$\overline{B}$	B
$\overline{A}$	1	0
A	0	1

Portanto, tem-se o mapa montado.

5.3.5 - Obtendo a Simplificação de uma Expressão com 2 Variáveis

1) Dada uma expressão booleana: S = A' • B + A • B' + A • B

2) Desenhar o diagrama. Como tem-se duas variáveis, então tem-se 4 células.

	$\overline{B}$	B
$\overline{A}$		
A		

3) Para cada parcela que aparece na expressão, deve-se colocar o numeral um na célula correspondente.

	$\overline{B}$	B
$\overline{A}$		1
A	1	1

4) Completar com o numeral zero (0) as células restantes que estão vazias.

	$\overline{B}$	B
$\overline{A}$	0	1
A	1	1

5) Para obter a expressão simplificada, devemos observar:

   i) Agrupar as células adjacentes (na horizontal ou vertical) onde aparecem o numeral 1, um mesmo numeral 1 pode ser componente de mais de um par. Todos os numerais 1 do diagrama devem pertencer a pelo menos um agrupamento;

	$\overline{B}$	B
$\overline{A}$	0	1
A	1	1

par 1

	$\overline{B}$	B
$\overline{A}$	0	1
A	1	1

par 2

   ii) O par 1 ocupa a região do diagrama onde A é igual a 1, logo o seu valor será: par 1 = A;

   iii) O par 2 ocupa a região do diagrama onde B é igual a 1, logo o seu valor será par 2 = B

   iv) Somando os pares para obter a expressão simplificada: S = A + B

5.3.6 - Simplificação de uma Expressão com 3 ou mais Variáveis

1) Dada a expressão booleana: S = A' • B • C + A • B' + A • B + A' • B' • C'

2) Desenhar o diagrama. Como temos três variáveis, teremos 8 células.

	$\overline{A}$	$\overline{A}$	A	A
$\overline{C}$				
C				
	$\overline{B}$	B	B	$\overline{B}$

3) Para cada parcela que aparece na expressão, colocar o numeral um na célula correspondente.

	$\overline{A}$	$\overline{A}$	A	A
$\overline{C}$	1		1	1
C		1	1	1
	$\overline{B}$	B	B	$\overline{B}$

4) Completar com o numeral zero (0) as células restantes que estão vazias.

	$\overline{A}$	$\overline{A}$	A	A
$\overline{C}$	1	0	1	1
C	0	1	1	1
	$\overline{B}$	B	B	$\overline{B}$

5) Para obter a expressão simplificada, devemos observar:

   i) Agrupar as células adjacentes (na horizontal ou vertical) onde aparecem o numeral 1, um mesmo numeral 1 pode ser componente de mais de um agrupamento. Todos os numerais 1 do diagrama devem pertencer a pelo menos um agrupamento;

	$\overline{A}$	$\overline{A}$	A	A
$\overline{C}$	1	0	1	1
C	0	1	1	1
	$\overline{B}$	B	B	$\overline{B}$

grupo 1

	$\overline{A}$	$\overline{A}$	A	A
$\overline{C}$	1	0	1	1
C	0	1	1	1
	$\overline{B}$	B	B	$\overline{B}$

grupo 2

	$\overline{A}$	$\overline{A}$	A	A
$\overline{C}$	1	0	1	1
C	0	1	1	1
	$\overline{B}$	B	B	$\overline{B}$

grupo 3

Observando todos os grupos ao mesmo tempo:

	$\overline{A}$	$\overline{A}$	A	A
$\overline{C}$	1	0	1	1
C	0	1	1	1
	$\overline{B}$	B	B	$\overline{B}$

ii) Devemos tentar montar os maiores agrupamentos possíveis, mas eles devem ser potência de 2, assim: 1, 2, 4, 8, 16, ...

iii) O agrupamento 1 ocupa a região do diagrama onde A é igual a 1, logo o seu valor será: grupo 1 = A;

iv) O agrupamento 2 ocupa a região do diagrama onde C é igual a 1 e B é igual a 1, logo o seu valor será par 2 = B • C;

v) O agrupamento 3 ocupa a região do diagrama onde $\overline{C}$ é igual 0 e $\overline{B}$ é igual 0, logo o seu valor será: grupo 3 = B’ • C’;

vi) Somando os pares, obtém a expressão simplificada: S = A + B • C + B’ • C’

5.3.7 – Observações Sobre as Simplificações

O mapa de Karnaugh usa as algumas regras de simplificação de expressões booleanas pelo agrupamento de células adjacentes.

a) Agrupamentos não podem conter qualquer célula com o numeral 0 (zero);

b) Agrupamentos podem ser horizontais ou verticais, mas não na diagonal;

c) Agrupamentos podem conter 1, 2, 4, 8, ou em geral 2^n células;

d) Todo agrupamento deve ser o maior possível;

e) Toda célula contendo o numeral 1 deve pertencer a pelo menos um agrupamento;

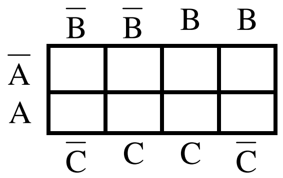
f) Os agrupamentos podem estar sobrepostos parcialmente, mas devem ter pelo menos um elemento novo que não pertença a nenhum outro agrupamento; Se ao desenvolver os agrupamentos, algum agrupamento tiver todos os seus elementos pertencentes a outros agrupamentos, este deve ser eliminado;

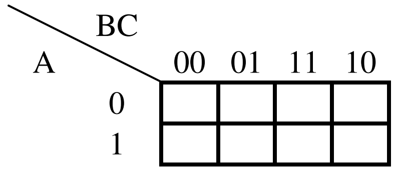
g) Os agrupamentos podem “dar a volta” no diagrama; A célula mais a esquerda em uma linha pode ser agrupada com a célula mais a direita da mesma linha, assim como a célula superior de uma coluna pode ser agrupada com a célula inferior da mesma coluna;

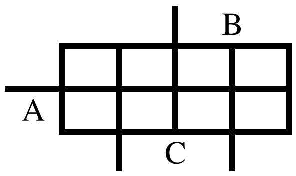
h) Deve-se formar o menor número de agrupamentos possíveis, desde que não contrarie nenhuma das regras anteriores.

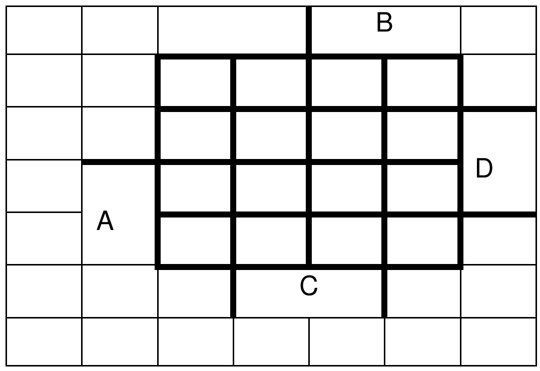
5.3.8 – Outras Formas de Representação do Mapa de Karnaugh

O diagrama pode ser representado de várias formas, mas o resultado será sempre o mesmo.

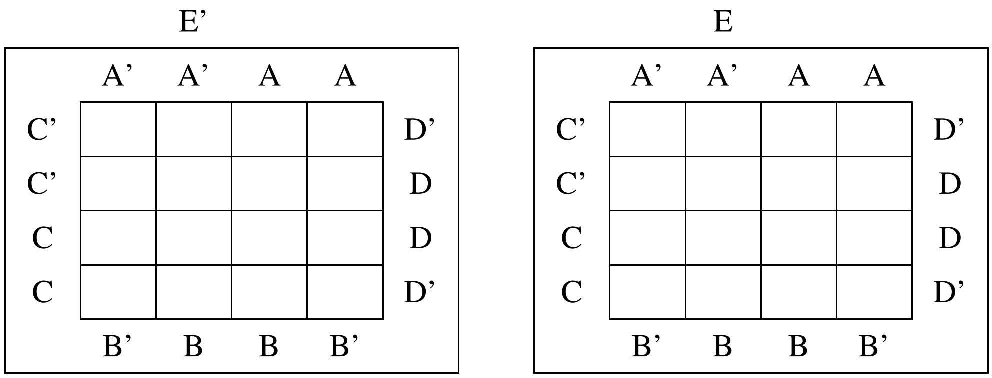








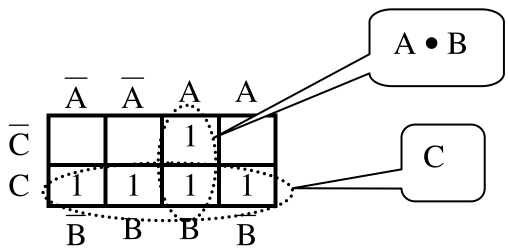
Mapa com 5 variáveis:



5.3.9 – Soma de Produtos

Todos os exemplos e exercícios apresentados até o momento são denominados de soma de produtos. Como pode ser observado, cada parcela da expressão é um produto de variáveis e que são somadas para formar a expressão. É denominada de Forma Normal Disjuntiva.

Exemplo 1: S = A • B + A’ • B • C + B’ • C



A expressão simplificada é: $S = A \bullet B + C$

Exemplo 2:

Encontrar a expressão booleana definida pela tabela-verdade, representá-la no mapa de karnaugh e encontrar a expressão mais simples.

A	B	C	S
1	1	1	1
1	1	0	1
1	0	1	1
1	0	0	0
0	1	1	1
0	1	0	0
0	0	1	0
0	0	0	0

$A \bullet B \bullet C$

$A \bullet B \bullet C'$

$A \bullet B' \bullet C$

$A' \bullet B \bullet C$

$S_1 = A \bullet B \bullet C + A \bullet B \bullet C' + A \bullet B' \bullet C + A' \bullet B \bullet C$

Montando o Mapa de Karnaugh:

	A'	A'	A	A
C'	0	0	1	0
C	0	1	1	1
	B'	B	B	B'

A solução simplificada é:

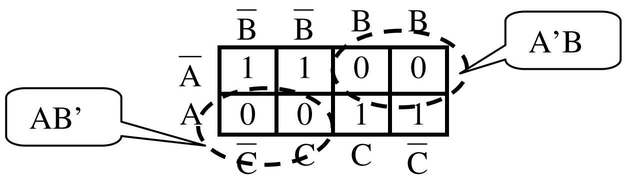
$S_1 = A \bullet B + A \bullet C + B \bullet C$

5.3.10 – Produtos da Soma

Como o próprio nome indica, cada parcela da expressão é uma soma e que serão reunidas através de produtos para formar a expressão. É denominada de Forma Normal Conjuntiva.

Para desenvolver o Mapa de Karnaugh e achar a expressão na forma de Produto da Soma, desenvolve-se:

a) Os agrupamentos com os numerais zero (0) no lugar do numeral 1 (um) da mesma forma que estudada anteriormente;

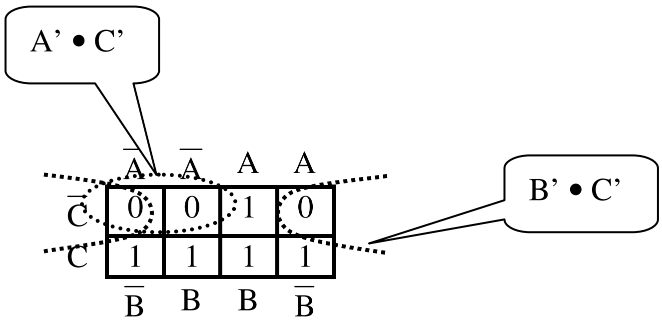


b) Montar a expressão, onde cada agrupamento será somado para formar a expressão, mas como foi desenvolvido a partir das células com numeral 0 (zero), o resultado será negado;
Z' = A' • B + A • B'

c) Para encontrar a expressão, tem-se que inicialmente negar os dois lados e assim eliminar a negação no resultado;
Z'' = (A' • B + A • B') '

d) Aplicar o teorema de DeMorgan duas vezes, encontrando assim, a expressão na forma de Produto da Soma.
Z = (A' • B)' • (A • B')'
Z = (A'' + B') • (A' + B'')
Z = (A + B') • (A' + B)

Exemplo: S = A • B + A' • B • C + B' • C



Montando a expressão simplificada:

S' = A' • C' + B' • C'

Para encontrar uma solução, é necessário usar a propriedade negação e assim:

S'' = (A' • C' + B' • C')'

S = (A' • C')' • (B' • C')'

S = (A'' + C'') • (B'' + C'')

S = (A + C) • (B + C)

5.4 Exercícios

5.4.1 - Usando as propriedades da Algebra de Boole, simplificar as expressões abaixo, indicando a propriedade usada em cada passagem..

a) S = A’ • B’ + A’ • B
b) S = (A + B) • (A + C)
c) S = A’ • B + B’ • C + B • C + A’ • B’ • C
d) S = A’ • B • C + A’ • B’ • C + A • B’

5.4.2 - Usando as expressões abaixo, simplificá-las usando os conceitos (Propriedades) da álgebra de Boole.

a) S = f(A,B,C) = A • B • C’ + A’ • B’ • C’ + B’ • C + A’ • C + A’ • B’
b) S = f(A,B,C) = A • B • C’ + A • B’ • C’ + A’ • B’
c) S = f(A,B,C) = (A + B) • (A’ + C’)
d) S = f(A,B,C) = (A’ • B + A’ • B) • (B’ + C)
e) S = f(A,B,C,D) = A • B • C + A’ • B • C + A • C + B • C + A • D’ + B • D
f) S = f(A,B,C,D) = A • B • D + A’ • B • C + B’ • C • D + B • C • D

5.4.3 - Identificar a expressão Booleana que gera o mapa de Karnaugh abaixo:

a)

	$\overline{A}$	$\overline{A}$	A	A	
$\overline{C}$	0	0	1	0	D
$\overline{C}$	1	0	0	0	$\overline{D}$
C	1	0	0	1	$\overline{D}$
C	0	0	0	0	D
	$\overline{B}$	B	B	$\overline{B}$	

b)

	$\overline{A}$	$\overline{A}$	A	A
$\overline{C}$	0	0	1	0
C	1	1	0	1
	$\overline{B}$	B	B	$\overline{B}$

5.4.4 - Identificar a expressão Booleana mais simples (soma de produto) a partir do mapa de Karnaugh abaixo:

a)

	$\overline{A}$	$\overline{A}$	A	A	
$\overline{C}$	0	1	1	1	D
$\overline{C}$	0	1	1	0	$\overline{D}$
C	0	0	0	0	$\overline{D}$
C	0	1	1	0	D
	$\overline{B}$	B	B	$\overline{B}$	

b)

	$\overline{A}$	$\overline{A}$	A	A	
$\overline{C}$	1	1	0	1	D
$\overline{C}$	1	1	0	1	$\overline{D}$
C	1	1	0	1	$\overline{D}$
C	1	1	0	1	D
	$\overline{B}$	B	B	$\overline{B}$	

c)

	$\overline{A}$	$\overline{A}$	A	A	
$\overline{C}$	1	1	1	0	D
$\overline{C}$	1	1	0	0	$\overline{D}$
C	1	0	0	1	$\overline{D}$
C	1	0	1	1	D
	$\overline{B}$	B	B	$\overline{B}$	

d)

	$\overline{A}$	$\overline{A}$	A	A	
$\overline{C}$	1	0	1	0	D
$\overline{C}$	1	0	0	0	$\overline{D}$
C	1	0	0	1	$\overline{D}$
C	1	0	1	1	D
	$\overline{B}$	B	B	$\overline{B}$	

e)

	$\overline{A}$	$\overline{A}$	A	A
$\overline{C}$	0	1	1	0
C	1	0	0	1
	$\overline{B}$	B	B	$\overline{B}$

f)

	$\overline{A}$	$\overline{A}$	A	A
$\overline{C}$	1	0	0	0
C	1	1	0	1
	$\overline{B}$	B	B	$\overline{B}$

g)

	$\overline{A}$	$\overline{A}$	A	A
$\overline{C}$	1	1	0	0
C	1	1	0	1
	$\overline{B}$	B	B	$\overline{B}$

h)

	$\overline{A}$	$\overline{A}$	A	A
$\overline{C}$	1	1	0	1
C	0	1	0	1
	$\overline{B}$	B	B	$\overline{B}$

5.4.5 - Identificar a expressão Booleana mais simples (Produto da Soma) a partir do mapa de Karnaugh abaixo.

a)

	$\overline{A}$	$\overline{A}$	A	A
$\overline{C}$	1	0	0	1
C	0	1	1	0
	$\overline{B}$	B	B	$\overline{B}$

b)

	$\overline{A}$	$\overline{A}$	A	A
$\overline{C}$	0	0	1	1
C	0	1	1	0
	$\overline{B}$	B	B	$\overline{B}$

c)

	$\overline{A}$	$\overline{A}$	A	A	
$\overline{C}$	1	0	0	1	D
$\overline{C}$	0	1	1	0	$\overline{D}$
C	0	1	1	1	$\overline{D}$
C	1	0	0	1	D
	$\overline{B}$	B	B	$\overline{B}$	

d)

	$\overline{A}$	$\overline{A}$	A	A	
$\overline{C}$	1	1	1	0	D
$\overline{C}$	0	0	1	0	$\overline{D}$
C	1	0	1	1	$\overline{D}$
C	1	1	1	1	D
	$\overline{B}$	B	B	$\overline{B}$	

e)

	$\overline{A}$	$\overline{A}$	A	A	
$\overline{C}$	1	0	0	1	D
$\overline{C}$	0	1	1	0	$\overline{D}$
C	0	1	1	0	$\overline{D}$
C	0	0	0	1	D
	$\overline{B}$	B	B	$\overline{B}$	

f)

	$\overline{A}$	$\overline{A}$	A	A	
$\overline{C}$	1	1	1	0	D
$\overline{C}$	1	0	0	0	$\overline{D}$
C	1	1	1	0	$\overline{D}$
C	1	1	1	0	D
	$\overline{B}$	B	B	$\overline{B}$	

5.4.6 - Identificar a expressão Booleana mais simples (produto de soma) a partir do mapa de Karnaugh abaixo:

a)

	$\overline{A}$	$\overline{A}$	A	A	
$\overline{C}$	1	0	0	1	D
$\overline{C}$	0	1	1	0	$\overline{D}$
C	0	1	1	1	$\overline{D}$
C	1	0	0	1	D
	$\overline{B}$	B	B	$\overline{B}$	

b)

	$\overline{A}$	$\overline{A}$	A	A	
$\overline{C}$	1	1	1	0	D
$\overline{C}$	1	1	1	0	$\overline{D}$
C	1	0	1	1	$\overline{D}$
C	1	1	1	0	D
	$\overline{B}$	B	B	$\overline{B}$	

c)

	$\overline{A}$	$\overline{A}$	A	A
$\overline{C}$	0	0	0	1
C	0	0	1	0
	$\overline{B}$	B	B	$\overline{B}$

d)

	$\overline{A}$	$\overline{A}$	A	A
$\overline{C}$	0	0	1	1
C	0	1	1	0
	$\overline{B}$	B	B	$\overline{B}$

e)

	$\overline{A}$	$\overline{A}$	A	A	
$\overline{C}$	1	0	0	1	D
$\overline{C}$	0	1	1	0	$\overline{D}$
C	0	1	1	0	$\overline{D}$
C	0	0	0	1	D
	$\overline{B}$	B	B	$\overline{B}$	

f)

	$\overline{A}$	$\overline{A}$	A	A	
$\overline{C}$	0	1	1	0	D
$\overline{C}$	1	0	0	0	$\overline{D}$
C	1	1	1	0	$\overline{D}$
C	0	1	1	0	D
	$\overline{B}$	B	B	$\overline{B}$	

5.4.7 - Seja a expressão booleana dada a seguir. Representá-la no mapa de Karnaugh e depois simplificá-la usando soma de produto.

a) $S = A' \bullet B' + A' \bullet B$
b) $S = A' \bullet B + A' \bullet B \bullet C + A \bullet C$
c) $S = A' \bullet B + B' \bullet C + B \bullet C + A' \bullet B' \bullet C$
d) $S = A' \bullet B \bullet C \bullet D + A' \bullet D + A \bullet B \bullet C' + A \bullet B \bullet D'$
e) $S = A \bullet B \bullet C + A' \bullet B \bullet C + A \bullet C + B \bullet C + A \bullet D' + B \bullet D$
f) $S = A \bullet B \bullet D + A' \bullet B \bullet C + B' \bullet C \bullet D + B \bullet C \bullet D$

5.4.8 - Seja a expressão booleana dada a seguir. Representá-la no mapa de Karnaugh e depois simplificá-la usando produto da soma.

a) $S = f(A,B,C) = A \bullet B' \bullet C' + B \bullet C' + A \bullet C' + A' \bullet B$
b) $S = f(A,B,C) = A \bullet B' \bullet C + A' \bullet B' \bullet C' + A' \bullet B$
c) $S = f(A,B,C) = (B + D) \bullet (A' + C') + A' \bullet B \bullet C$
d) $S = f(A,B,C) = (A' \bullet B + A' \bullet B') \bullet (D' + C)$
e) $S = f(A,B,C,D) = A \bullet B \bullet C + A' \bullet B \bullet C + A \bullet C + B \bullet C + A \bullet D' + B \bullet D$
f) $S = f(A,B,C,D) = A \bullet B \bullet D + A' \bullet B \bullet C + B' \bullet C \bullet D + B \bullet C \bullet D$

5.4.9 - Seja a expressão booleana dada a seguir. Faça a simplificação utilizando os 3 métodos (pelas propriedades, Mapa de Karnaugh usando Soma de produtos e mapa de Karnaugh usando Produto da soma)

a) $S = f(A,B,C) = A \bullet B' \bullet C' + B \bullet C' \bullet D' + A \bullet C' \bullet D' + A' \bullet B$
b) $S = f(A,B,C,D) = A \bullet B \bullet C + A \bullet B' \bullet C + A \bullet C + B \bullet C + A \bullet D + B' \bullet D$
c) $S = f(A,B,C,D) = A \bullet B \bullet D + A' \bullet B \bullet C + B \bullet C \bullet D' + B \bullet C \bullet D$
d) $S = f(A,B,C,D) = A \bullet B \bullet C \bullet D + A' \bullet D + A \bullet B \bullet C' + A \bullet B \bullet C + A' \bullet B \bullet D$
e) $S = f(A,B,C,D) = A \bullet B \bullet C + A \bullet B \bullet D + A \bullet B \bullet C' + A \bullet D' + A \bullet C \bullet D'$

6 PORTAS LÓGICAS.

6.1 Introdução

George Boole desenvolveu a álgebra booleana em 1854. É uma maneira formal para escrever e manipular proposições lógicas como se fosse uma fórmula algébrica. Em 1938, Claude Shannon demonstra a aplicação da álgebra booleana na análise de circuitos à relês.

A eletrônica digital, que usa dígitos binários, utiliza alguns circuitos lógicos básicos conhecidos como portas OU, E, NÃO, ... através da utilização conveniente desses circuitos, podemos "implementar" todas as expressões geradas pela álgebra de Boole. Há uma relação entre a estrutura de álgebra de Boole e os diagramas para os circuitos elétricos em computadores, calculadoras, dispositivos industriais de controle, sistemas de telecomunicações, etc.

Uma das principais áreas de utilização da álgebra booleana é a de circuitos lógicos, em que representamos as funções booleanas através de gráficos, símbolos padronizados conhecidos como portas lógicas. As portas lógicas buscam combinar diversos termos de uma função booleana que, juntas, irão realizar uma determinada função. Com esta forma de representação, podemos simbolizar as expressões básicas da álgebra booleana (AND, OR ou NOT), e partindo delas gerar funções mais complexas. Tanto as expressões básicas, como o conjunto delas podem conter várias variáveis de entrada, mas somente uma saída.

6.2 Circuitos Lógicos

Chamamos interruptor ao dispositivo ligado a um ponto do circuito elétrico, que pode assumir um dos dois estados: fechado (1) ou aberto (0). Quando fechado, o interruptor permite que a corrente elétrica passe através do ponto, e quando aberto nenhuma corrente elétrica pode passar pelo ponto. Representação:

Por conveniência, representamos os interruptores da forma:

neste caso, somente conhecemos o estado do interruptor se tivermos a indicação de que A=1 ou A=0.

6.3 Variáveis e Expressões na Álgebra de Boole

As variáveis booleanas, que são representadas através de letras, podem assumir apenas dois valores: 0 e 1.

Expressão booleana é uma expressão matemática cujas variáveis são booleanas e seu resultado assumirá apenas dois valores: 0 e 1.

6.4 Postulados

a) Postulado da Complementação

Chamamos de **A’** o complemento de A. Dizemos “A BARRA”. Poder ser: $\overline{\mathbf{A}}$

i) se A = 0 então A’ = 1

ii) se A = 1 então A’ = 0

através do postulado da complementação, podemos estabelecer a seguinte identidade: A’’ = A

b) Postulado da Adição

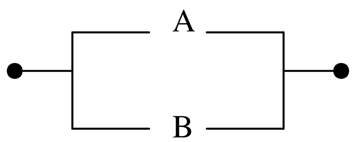
i) 0 + 0 = 0

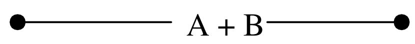
ii) 0 + 1 = 1

iii) 1 + 0 = 1

iv) 1 + 1 = 1

O Circuito lógico que executa o postulado da adição é o circuito OU, representado por “dois interruptores em paralelo”, com a representação:





c) Postulado da Multiplicação

i) 0 • 0 = 0

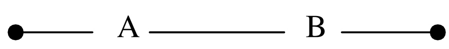
ii) 0 • 1 = 0

iii) 1 • 0 = 0

iv) 1 • 1 = 1

através deste postulado, podemos estabelecer as seguintes identidades:

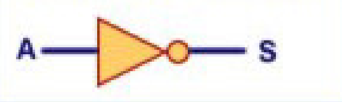
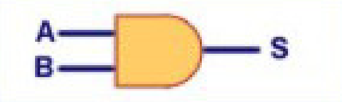
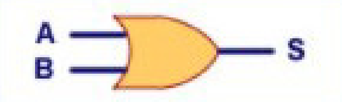
O circuito lógico que executa o postulado da multiplicação booleana é o circuito E, representado por dois interruptores em série, cuja representação é:



6.5 Representação

A representação gráfica dos operadores da álgebra booleana é feita mediante símbolos padronizados por normas internacionais chamados blocos ou portas lógicas.

As portas lógicas são as bases dos circuitos lógicos e tem por finalidade combinar as diferentes variáveis booleanas de modo a realizar determinada função booleana.

NOME	Símbolo Gráfico	Símbolo Algébrico
NOT		$S = \overline{A}$ ou $S = A'$
AND		$S = A . B$ ou $S = AB$
OR		$S = A + B$

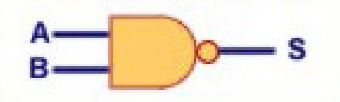
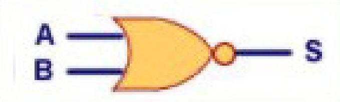
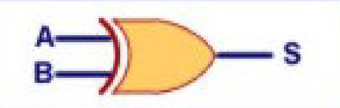
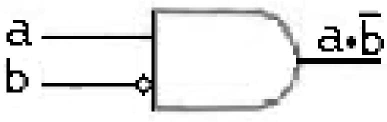
NOME	Símbolo Gráfico	Símbolo Algébrico
NAND		$S = (\overline{A\,B})$
NOR		$S = (\overline{A + B})$
XOR		$S = A \oplus B$

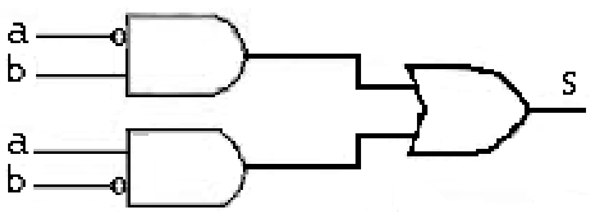
Fig. Portas lógicas. Fonte: TONIN (2008, p. 20)

6.6 Exemplos de exercícios de portas lógicas:

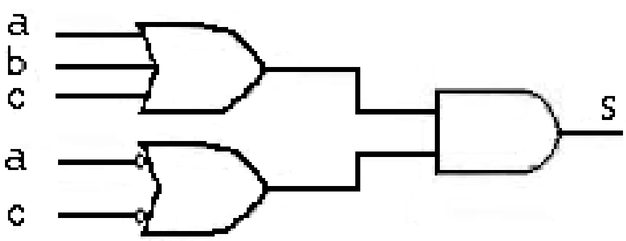
a) Representar mediante portas lógicas a função booleana S = f(A,B) = **A • B’**



b) Representar mediante portas lógicas a função booleana S = f(A,B) = **A’ • B + A • B’**



c) Determinar a função booleana correspondente ao circuito lógico abaixo:



S = (A + B + C) • (A’ + C’)

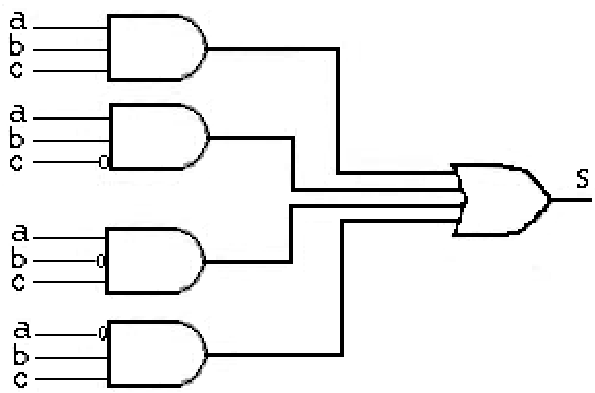
d) Representar em circuito lógico a função booleana definida pela tabela-verdade abaixo:

A	B	C	S
1	1	1	1
1	1	0	1
1	0	1	1
1	0	0	0
0	1	1	1
0	1	0	0
0	0	1	0
0	0	0	0

→ A • B • C
→ A • B • C'
→ A • B' • C
→ A' • B • C

S = A • B • C + A • B • C' + A • B' • C + A' • B • C

O circuito lógico que representa a tabela é:

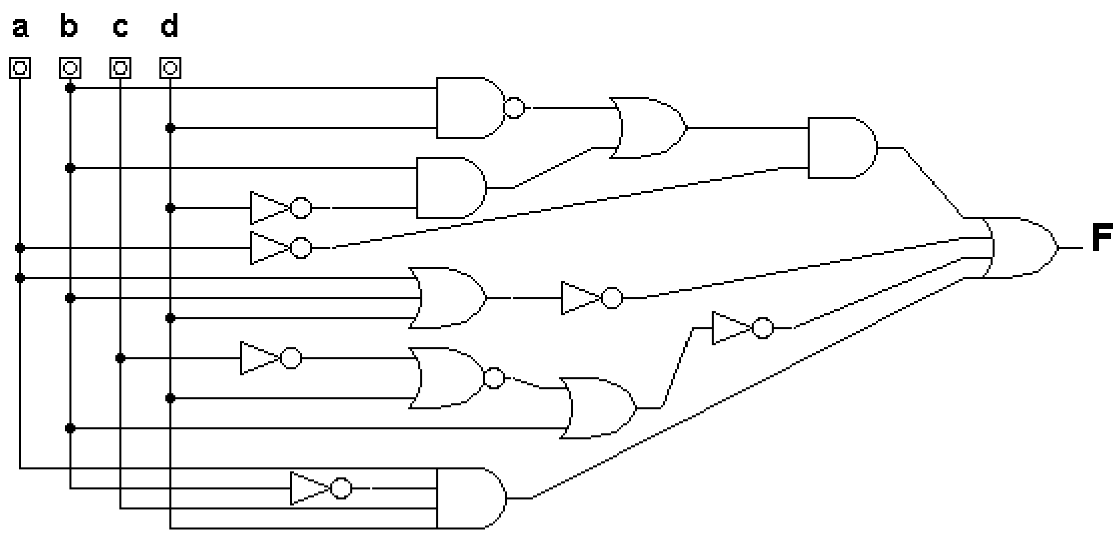


6.7 Exercícios

6.7.1 - Representar a expressão booleana utilizando portas lógicas.

a) S = f(A,B,C) = A • B' • C' + A • B • C' + A • C' + A' • B
b) S = f(A,B,C) = A • B' • C + (A' • B' • C')' + (A • B' • C)'
c) S = f(A,B,C) = (A + B) • (A' + C') • (A • B' • C)
d) S = f(A,B,C) = (A' • B) + (A • B' • C) • (A • B' + C)
e) S = f(A,B,C,D) = A • B • C + (A' • B • C)' + (A • C + B • C)'
f) S = f(A,B,C,D) = A • B • D + A' • B • C + B' • C • D + B • C • D

6.7.2 – Encontrar a expressão booleana que representa o circuito abaixo.



6.7.3 - Dada a função x = a’bc + abcd’ + a’b’cd + bcd

a) criar o mapa de Karnaugh desta função;

b) simplificar a função, se possível, através do mapa de Karnaugh gerado na alternativa a;

c) gerar o circuito lógico correspondente à função simplificada na alternativa b.

6.7.4 - Encontrar uma função existente na tabela abaixo, e depois através de portas lógicas representar a função encontrada.

a)

a	b	c	X
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

b)

a	b	c	X
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

6.7.5 - Seja a expressão booleana dada a seguir. Representá-la no mapa de Karnaugh e depois simplificá-la usando soma de produto. Gerar o circuito lógico correspondente à função simplificada.

a) S = f(A,B,C) = A • B’ • C’ + B • C’ + A • C’ + A’ • B

b) S = f(A,B,C) = A • B’ • C + A’ • B’ • C’ + A’ • B

c) S = f(A,B,C) = (A + B) • (A’ + C’)

d) S = f(A,B,C) = (A’ • B + A’ • C) • (B’ + C)

e) S = f(A,B,C,D) = A • B • C + A’ • B • C + A • C + B • C + A • D’ + B • D

f) S = f(A,B,C,D) = A • B • D + A’ • B • C + B’ • C • D + B • C • D

6.7.6 - Usando as expressões abaixo, simplificá-las usando o Mapa de Karnaugh (Produto da soma) e depois representá-las em Circuitos Lógicos, usando as portas lógicas: Not, And e Or.

a) S = f(A,B,C) = A • B • C’ + B’ • C + A’ • C + A’ • B’

b) S = f(A,B,C) = A • B • C’ + A • B’ • C’ + A’ • B’

c) S = f(A,B,C) = (A + B) • (A’ + C’)

d) S = f(A,B,C) = (A’ • B + A’ • B) • (B’ + C)

e) S = f(A,B,C,D) = A • B • C + A’ • B • C + A • C + B • C + A • D’ + B • D

f) S = f(A,B,C,D) = A • B • D + A’ • B • C + B’ • C • D + B • C • D

6.7.7 - Usando as expressões abaixo, simplificá-las usando o Mapa de Karnaugh (Produto da soma) e depois representá-las em Circuitos Lógicos, usando as portas lógicas: Not, And, Or, Nand e Nor.

a) S = f(A,B,C) = A • B • C’ + B’ • C + A’ • C + A’ • B’

b) S = f(A,B,C) = A • B • C’ + A • B’ • C’ + A’ • B’

c) S = f(A,B,C) = (A + B) • (C’ + D’)

d) S = f(A,B,C) = (A’ • B + A’ • B) • (C’ + D) + A’ • B • C + B’ • C • D

e) S = f(A,B,C,D) = A • B • C + A’ • B • C + A • C + B • C + A • D’ + B • D

f) S = f(A,B,C,D) = A • B • D + A’ • B • C + B’ • C • D + B • C • D

6.7.8 - Identificar a expressão Booleana mais simples (soma de produto) a partir do mapa de Karnaugh abaixo e depois representá-las em Circuitos Lógicos, usando as portas lógicas: Not, And, Or:

a)

	$\overline{A}$	$\overline{A}$	A	A	
$\overline{C}$	0	1	1	1	D
$\overline{C}$	0	1	1	0	$\overline{D}$
C	0	0	0	0	$\overline{D}$
C	0	1	1	0	D
	$\overline{B}$	B	B	$\overline{B}$	

b)

	$\overline{A}$	$\overline{A}$	A	A	
$\overline{C}$	1	1	0	1	D
$\overline{C}$	1	1	0	1	$\overline{D}$
C	1	1	0	1	$\overline{D}$
C	1	1	0	1	D
	$\overline{B}$	B	B	$\overline{B}$	

c)

	$\overline{A}$	$\overline{A}$	A	A	
$\overline{C}$	1	1	1	0	D
$\overline{C}$	1	1	0	0	$\overline{D}$
C	1	0	0	1	$\overline{D}$
C	1	0	1	1	D
	$\overline{B}$	B	B	$\overline{B}$	

d)

	$\overline{A}$	$\overline{A}$	A	A	
$\overline{C}$	1	1	1	0	D
$\overline{C}$	1	1	0	0	$\overline{D}$
C	1	0	0	1	$\overline{D}$
C	1	0	1	1	D
	$\overline{B}$	B	B	$\overline{B}$	

e)

	$\overline{A}$	$\overline{A}$	A	A
$\overline{C}$	0	1	1	0
C	1	0	0	1
	$\overline{B}$	B	B	$\overline{B}$

f)

	$\overline{A}$	$\overline{A}$	A	A
$\overline{C}$	1	0	0	0
C	1	1	0	1
	$\overline{B}$	B	B	$\overline{B}$

g)

	$\overline{A}$	$\overline{A}$	A	A
$\overline{C}$	1	1	0	0
C	1	1	0	1
	$\overline{B}$	B	B	$\overline{B}$

h)

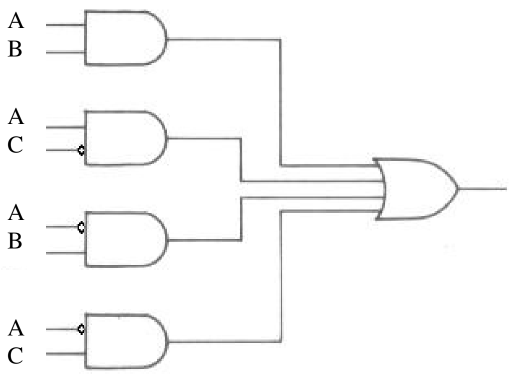
	$\overline{A}$	$\overline{A}$	A	A
$\overline{C}$	1	1	0	1
C	0	1	0	1
	$\overline{B}$	B	B	$\overline{B}$

6.7.9 - Seja a expressão booleana dada a seguir. Representá-la no mapa de Karnaugh e depois simplificá-la. Representar a expressão simplificada usando portas lógicas.

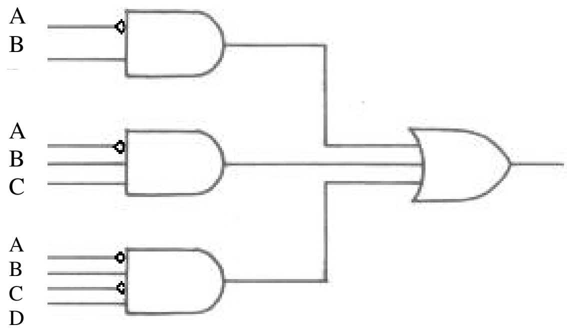
a) S = f(A,B,C) = A • B + B’ • C + A’ • B’ + A’ • B • C
b) S = f(A,B,C) = A • B • C + B’ • A + A • C + A’ • B
c) S = f(A,B,C) = A • B • C’ + B • C’ + A’ • C’ + A • B’
d) S = f(A,B,C,D) = A • B • C + A • B’ • C + A • C + B • C + A’ • D + B’ • D
e) S = f(A,B,C,D) = A • B • D + A’ • B • C + B • C’ • D’ + B • C • D + A • B • D’

6.7.10 - Escrever a expressão booleana representada pelo circuito a seguir. Representá-la no mapa de Karnaugh e depois simplificá-la. Representar a expressão simplificada usando portas lógicas.

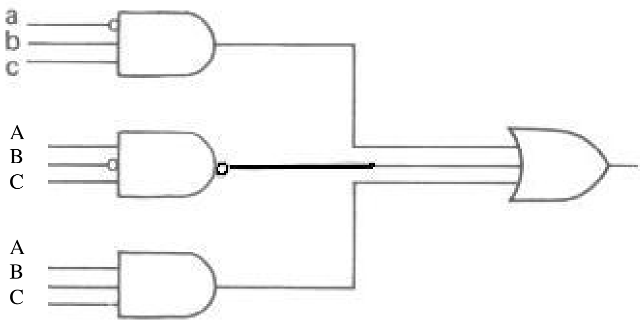
a)


A
B
A
C
A
B
A
C

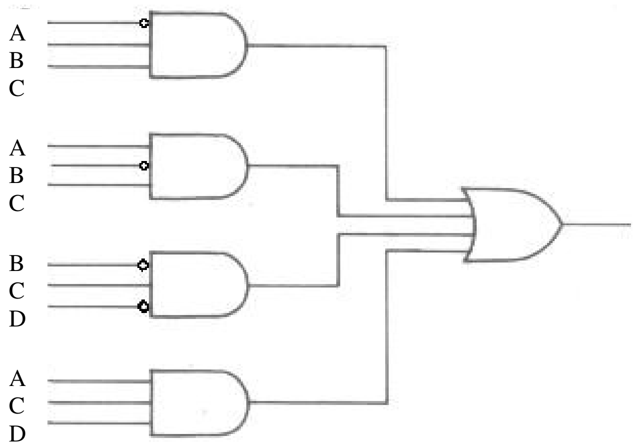
b)


A
B
A
B
C
A
B
C
D

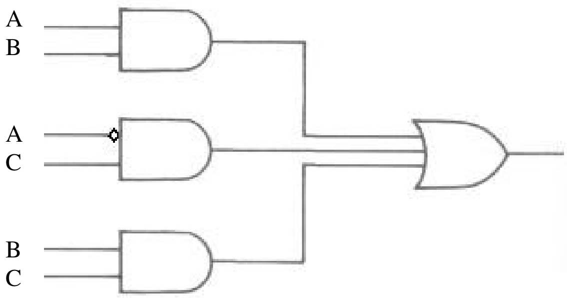
c)


a
b
c
A
B
C
A
B
C

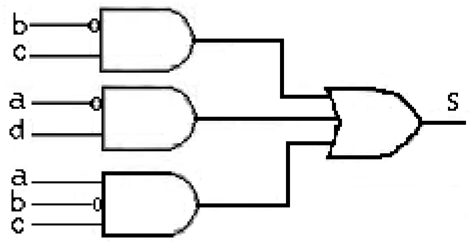
d)


A
B
C
A
B
C
B
C
D
A
C
D

e)


A
B
A
C
B
C

f)


b
c
a
d
a
b
c
s

7 REGRAS DE INFERÊNCIA

7.1 Argumentos

Chama-se de argumento toda a afirmação de que várias proposições (p1, p2, ..., pn) tem por consequencia uma outra proposição q. As proposições p1, p2, ..., pn são as premissas, e a proposição q é a conclusão do argumento. Um argumento é descrito da seguinte forma: p, $p \rightarrow q$, $q \rightarrow r \vdash r$ onde:

$p, p \rightarrow q, q \rightarrow r$	$\vdash$	r
Premissas		conclusão

7.1.1 Validade de um argumento através da Tabela-verdade

Um argumento é valido quando para todas as linhas da tabela-verdade onde as premissas forem verdadeiras, a conclusão também é verdadeira.

7.2 Processo de inferência

Inferência é o processo pelo qual se chega a uma proposição, firmada na base de uma ou outras mais proposições aceitas como ponto de partida do processo. O Argumento é chamado de premissa e o valor de conclusão. As conclusões são deduzidas a partir das premissas. Caso o estado das premissas esteja vazio, então a conclusão é dita ser o axioma da lógica.

Regras de inferência são regras sintáticas que produzem enunciados válidos em um sistema formal. A partir de um conjunto de proposições podemos derivar outras seguindo estas regras.

Regras de inferência têm as seguintes características:

- Se a Hipótese for verdadeira, então a Conclusão é verdadeira
- Verificação de tipos é baseada em inferência. Se E1 e E2 tem certos tipos, então E3 tem um certo tipo.
- Regras de inferência são uma notação compacta para comandos de implementação.
- Inicia-se com um sistema simplificado de regras ao qual adiciona-se novas características gradualmente
- As premissas são regras sem hipóteses

Na lógica formal, as regras de inferência são normalmente determinadas nas seguinte forma:

premissa 1
premissa 2
...
premissa n
----
conclusão

A validade das regras de inferência pode ser comprovada facilmente com o uso das tabelas-verdade. Com o auxílio destas formas elementares de argumentos podemos construir estruturas argumentativas muito complexas.

7.2.1 Modus Ponens (MP)

$P \rightarrow Q$
P

Q

Se o antecedente de um condicional for verdadeiro, o seu consequente necessariamente é verdadeiro. Por esta razão, esta regra também é chamada de afirmação do antecedente. A validade desta regra é comprovada simplesmente observando a tabela verdade das proposições condicionais. Violações desta regra resultam nas falácias *afirmação do consequente* e *negação do antecedente*, onde a conclusão não segue necessariamente das premissas.

7.2.2 Modus Tollens (MT)

$P \rightarrow Q$
$\sim Q$

$\sim P$

Ao contradizer o consequente, somos obrigados a contradizermos também o antecedente. Novamente isto pode ser verificado simplesmente olhando a tabela verdade das proposições condicionais. Esta regra é também chamada de contradição do consequente.

7.2.3 Silogismo Hipotético (SH)

$P \rightarrow Q$
$Q \rightarrow R$

$P \rightarrow R$

A implicação possui a propriedade transitiva, isto é, se A implica B e B implica C, então A implica C através de B.

7.2.4 Silogismo Disjuntivo (SD)

$P \vee Q$
$\sim Q$

P

Se uma disjunção é verdadeira e uma das proposições componentes se revela falsa, então a outra proposição é necessariamente verdadeira.

7.2.5 Dilema Construtivo (DC)

$(P \rightarrow Q) \wedge (R \rightarrow S)$
$P \vee R$

$Q \vee S$

Dilemas são situações em que somos obrigados a aceitar uma de duas consequências que não são muito agradáveis. Deixando a agradabilidade de lado, esta regra de inferência se baseia na regra Modus Ponens. Tomando apenas a primeira parte da conjunção da primeira premisa, *(p –> q)*, afirmamos *p*, e pela regra Modus ponens, somos obrigados a concluir *q*. Fazendo o mesmo procedimento com o outro lado da conjunção da primeira premissa, concluímos que s. Então o dilema consiste em que, ao afirmar *p* ou *r*, somos obrigados a concluir *q* ou *s*.

7.2.6 Dilema Destrutivo (DD)

(P → Q) ^ (R → S)
~Q v ~S

~P v ~R

É exatamente o oposto do Dilema Construtivo. Baseia-se na regra Modus Tollens.

7.2.7 Absorção (ABS)

P → Q

P → (P ^ Q)

Dada uma condicional, pode-se deduzir dela uma condicional que tem como antecedente o mesmo antecedente da primeira e como consequente uma conjunção as duas proposições que figuravam na primeira condicional. Uma reflexão sobre a tabela verdade das condicionais é capaz de mostrar como esta inferência é válida.

7.2.8 Simplificação (SIMP)

P ^ Q

P

Em uma conjução verdadeira, pode-se concluir que cada um dos seus componentes é verdadeiro de forma independente. Muito simples.

7.2.9 Conjunção (CONJ)

P
Q

P ^ Q

Praticamente o inverso da regra anterior, se dois enunciados são verdadeiros independentemente, isso é condição suficiente para que juntos formem uma conjunção verdadeira.

7.2.10 Adição (AD)

P

P v Q

Dada uma proposição verdadeira, a partir dele pode-se deduzir uma disjunção verdadeira com qualquer outro enunciado que escolhermos. Isto tem ligação com a regra Silogismo

Disjuntivo, pois se o enunciado que escolhermos, *q*, porventura for falso, em nada afetará a verdade da proposição p e da disjunção por elas formada.

7.2.11 Exemplo

Usando as regras de inferência devidamente explicadas, pode-se analisar o argumento apresentado abaixo:

- Se os 8 casos suspeitos de Nova York forem gripe suína, então o governo deverá tomar medidas efetivas para conter a disseminação da doença.
- E se o governo deve tomar medidas para conter a disseminação da doença, então as escolas de NY serão fechadas.
- (Quer dizer,) se os 8 casos suspeitos forem de gripe suína, então as escolas de NY serão fechadas.
- Os suspeitos podem ter gripe Influenza do tipo A ou ter gripe suína.
- (Foi descoberto que) a gripe dos suspeitos não é Influenza do tipo A.
- Logo, as escolas de NY serão fechadas.

Variáveis proposicionais:

- *P: os 8 casos suspeitos de Nova York são de gripe suína*
- *Q: o governo deve tomar medidas efetivas para conter a disseminação da doença*
- *R: as escolas de NY serão fechadas*
- *S: os suspeitos podem ter gripe Influenza do tipo A*

Tradução formal:

$P \rightarrow Q$
$Q \rightarrow R$
$P \rightarrow R$
$S \vee P$
$\sim S$
\------
R

Primeira regra usada:

- Silogismo Hipotético (SH)

$P \rightarrow Q$
$Q \rightarrow R$
\---------
$P \rightarrow R$

Segunda regra usada:

- Silogismo Disjuntivo (SD)

S v P
~S
\-------
P

Explicação:

No argumento, a conclusão *p* do Silogismo Disjuntivo ficou subentendida. Daí, de repente, concluiu-se que *r*. A razão desta grande omissão, frequente na vida real, é que se a negação de *s* resulta em *p*, e, como visto no Silogismo Hipotético, *p* resulta em *r*, conclui-se definitivamente que a negação de *s* resulta em *r*.

7.2.12 Revisão

REGRAS DE INFERÊNCIA.: A fórmula α implica tautologicamente a fórmula β e indicamos $\alpha \Rightarrow \beta$ se e somente se a fórmula $\alpha \rightarrow \beta$ é uma tautologia .

Regras	Sigla	Fórmulas Atômicas	Fórmulas Compostas
Modus Ponens	MP	$p \wedge (p \rightarrow q) \Rightarrow q$	A, A→ B / B
Modus Tollens	MT	$\sim q \wedge (p \rightarrow q) \Rightarrow \sim p$	~ B, A→ B / ~ A
Silogismo Hipotético	SH	$(p\rightarrow q) \wedge (q \rightarrow r) \Rightarrow (p \rightarrow r)$	A → B, B → C / A → C
Silogismo Disjuntivo	SD	$(p \vee q) \wedge \sim p \Rightarrow q$	~ A, A ∨ B / B
Simplificação	SM	$p \wedge q \Rightarrow p$	A ∧ B / A
Adição	AD	$p \Rightarrow p \vee q$	A / A ∨ B
Eliminação	EL	$(p \rightarrow (q \vee r)) \wedge\sim q \Rightarrow p \rightarrow r$	~ B , (A → (B∨ C) / A → C
Prova por Casos	CS	$(p \rightarrow r) \wedge (q \rightarrow r) \Rightarrow (p \vee q) \rightarrow r$	A → C, B → C / (A ∨ B) → C

7.2.13 Exercícios resolvidos

Na coluna direita estão indicadas as linhas e as regras de inferência que produziram a fórmula da coluna esquerda.

a) Derive U das seguintes premissas:

1.	$P \rightarrow (Q \wedge R)$	**premissa**
2.	$(Q \wedge R) \rightarrow S$	**premissa**
3.	$S \rightarrow (T \vee (\neg T \rightarrow U))$	**premissa**
4.	P	**premissa**
5.	$\neg T$	**premissa**
6.	$(Q \wedge R)$	1,4 Modus Ponens
7.	S	2,6 Modus Ponens
8.	$(T \vee (\neg T \rightarrow U))$	3,7 Modus Ponens
9.	$(\neg T \rightarrow U)$	5,8 Silogismo Disjuntivo
10.	U	5,9 Modus Ponens

b) Derive (R ∧ S)

1.	$\neg\neg P$	**premissa**
2.	$Q \rightarrow (R \wedge S)$	**premissa**
3.	$T \leftrightarrow \neg\neg Q$	**premissa**
4.	$T \vee \neg P$	**premissa**
5.	$T \rightarrow \neg\neg Q$	3 Bicondicional
6.	T	1,4 Silogismo Disjuntivo
7.	$\neg\neg Q$	5,6 Modus Ponens
8.	Q	7 Dupla Negação
9.	$(R \wedge S)$	2,8 Modus Ponens

c) Derive P → ¬R

1.	$P \rightarrow \neg Q$	**premissa**
2.	$\neg Q \rightarrow \neg R$	**premissa**
3.	$\mid P$	*Hipótese*
4.	$\mid \neg Q$	1,3 Modus Ponens
5.	$\mid \neg R$	2,4 Modus Ponens
6.	$P \rightarrow \neg R$	3-5 Regra de prova condicional

d) Derive ¬(P ∧ Q)

1.	$P \rightarrow \neg Q$	**premissa**
2.	$\mid P \wedge Q$	*Hipótese* (por contradição)
3.	$\mid P$	2 Simplificação
4.	$\mid \neg Q$	1,3 Modus Ponens
5.	$\mid Q$	2 Simplificação
6.	$\mid Q \wedge \neg Q$	4,5 Conjunção
7.	$\neg(P \wedge Q)$	2-6 Redução ao Absurdo

e) Derive (P ∨ Q) ∧ (P ∧ R)

1.	$(P \vee Q) \rightarrow R$	**premissa**
2.	$R \wedge P$	**premissa**
3.	P	2 Simplificação
4.	$P \vee Q$	3 Adição
5	. R	1,4 Modus Ponens
6.	$P \wedge R$	3,5 Conjunção
7.	$(P \vee Q) \wedge (P \wedge R)$	4,6 Conjunção

`

7.3 Exercícios

7.3.1 Verificar a validade dos seguintes argumentos

a) $p \rightarrow q,\ p \wedge r\ q$
b) $p \wedge q,\ p \vee r \rightarrow s\ p \wedge s$.
c) $p \rightarrow (q \rightarrow r),\ p \rightarrow q,\ p\ r$
d) $p \vee q \rightarrow r,\ (r \vee q) \rightarrow (p \rightarrow s \leftrightarrow t)),\ p \wedge s\ s \leftrightarrow t$

7.3.2 Demonstrar a validade do argumento

P1: Se Pedro tem a mesma altura que João então João tem a mesma altura que Luis.
P2: Se Pedro tem a mesma altura que Luis então Pedro tem a mesma altura que Antônio.
P3: Ou Pedro tem a mesma altura que João ou a altura de Pedro é 1,80 m.
P4: Se a altura de Pedro é 1,80 m então a altura de Pedro mais 0,20 m = 2,00 m.
P5: Mas a altura de Pedro mais 0,20 não é igual a 2,00m.
Portanto: Pedro tem a mesma altura que Antônio.

8 SISTEMAS DE NUMERAÇÃO

Os sistemas de numeração tem por objetivo prover símbolos e convenções para representar quantidades, de forma a registrar a informação quantitativa e poder processá-la. A representação de quantidades se faz com os números. Na antiguidade, duas formas de representar quantidades foram inventadas. Inicialmente, os egípcios, criaram um sistema em que cada dezena (uma mão-cheia de nosso exemplo anterior) era representada por um símbolo diferente. Usando por exemplo os símbolos # para representar uma centena, & para representar uma dezena e @ representando uma unidade (símbolos escolhidos ao acaso), teríamos que ###&&@ representaria 321.

Relembremos ainda um outro um sistema, o sistema de numeração romano. Eram usados símbolos (letras) que representavam as quantidades, como por exemplo: I (valendo 1), V (valendo 5), X (valendo 10), C (valendo 100), etc. A regra de posicionamento determinava que as letras que representavam quantidades menores e precediam as que representavam quantidades maiores, seriam somadas; se o inverso ocorresse, o menor valor era subtraído do maior (e não somado). Assim, a quantidade 128 era representada por CXXVIII = 100 + 10 + 10 + 5 + 1 + 1 + 1 = 128. Por outro lado, a quantidade 94 era representada por XCIV = (-10 + 100) + (-1 + 5) = 94.

Nesses sistemas, os símbolos tinham um valor intrínseco, independente da posição que ocupavam na representação (sistema numérico não-posicional). Um grande problema desse sistema é a dificuldade de realizar operações com essa representação. Experimente multiplicar CXXVIII por XCIV! Assim, posteriormente foram criados sistemas em que a posição dos algarismos no número passou a alterar seu valor (sistemas de numeração posicionais).

Nos sistemas de numeração posicionais (ver adiante Sistemas de Numeração Posicionais), o valor representado pelo algarismo no número depende da posição em que ele aparece na representação. O primeiro sistema desse tipo foi inventado pelos chineses. Eram usados palitos, sendo 1 a 5 palitos dispostos na vertical para representar os números 1 a 5; de 6 a 9 eram representados por 1 a 4 palitos na vertical, mais um palito na horizontal (valendo 5) sobre os demais. Cada número era então representado por uma pilha de palitos, sendo uma pilha de palitos para as unidades, outra para as dezenas, outra para as centenas, etc. Esse sistema, com as pilhas de palitos dispostas em um tabuleiro, permitia a realização das quatro operações aritméticas. Não existia representação para o zero (o espaço relativo ficava vazio).O tabuleiro aritmético (chamado swan-pan), além das quatro operações, era usado na álgebra e na solução de equações. Essa técnica era chamada de Método do Elemento Celestial.

O Alfabeto e o Ábaco No Oriente Médio, por esses tempos, criou-se uma das mais importantes invenções da humanidade: o alfabeto. Na antigüidade, usava-se um símbolo para representar cada conceito ou palavra. Assim, eram necessários milhares de símbolos para representar todos os objetos, ações, sentimentos, etc - como são ainda hoje algumas linguagens. Como decorar todos? O grande achado foi decompor a linguagem em alguns poucos símbolos e regras básicas. Uma conseqüência de fundamental importância para nossos estudos de informática foi possibilitar a ordenação alfabética (essa é uma tarefa típica dos computadores).

Nessa época, foi também criado o ábaco - uma calculadora decimal manual.

Os Algarismos e o Zero Por volta do ano de 650, os hindus inventaram um método de produzir papel (que antes já fora inventado pelos chineses) e seus mateméticos criaram uma representação para os números em que existiam diferentes símbolos para os as unidades, incluindo um símbolo para representar o zero. Essa simples criação permitiu que se processasse a aritmética decimal e se fizesse contas - no papel! Bom, depois de milhares de anos em que todos os cálculos eram feitos com calculadoras (ábacos, swan-pan, etc) finalmente era possível calcular sem auxílio mecânico, usando um instrumento de escrita e papel. A matemática criada pelos hindus foi aprendida pelos árabes (que depois foram copiados pelos europeus). Por volta de 830, um matemático persa (chamado Al-khwarismi, que inspirou o nome algarismo) escreveu um livro (Al-gebr we'l Mukabala, ou álgebra) em que apresentava os algarismos hindus. E esse livro, levado para a Europa e traduzido, foi a base da matemática do Renascimento.

8.1 Sistemas de Numeração Posicionais

Desde quando se começou a registrar informações sobre quantidades, foram criados diversos métodos de representar as quantidades.

O método ao qual estamos acostumados usa um sistema de numeração posicional. Isso significa que a posição ocupada por cada algarismo em um número altera seu valor de uma potência de 10 (na base 10) para cada casa à esquerda.

Por exemplo, no sistema decimal (base 10), no número 125 o algarismo 1 representa 100 (uma centena ou 10^2) , o algarismo 2 representa 20 (duas dezenas ou $1x10^1$) e o algarismo 5 representa 5 mesmo (5 unidades ou $5x10^0$). Assim, em nossa notação,

$125 = 1x10^2 + 2x10^1 + 5x10^0$

Base de um Sistema de Numeração A base de um sistema é a quantidade de algarismos disponível na representação. A base 10 é hoje a mais usualmente empregada, embora não seja a única utilizada. No comércio pedimos uma dúzia de rosas ou uma grosa de parafusos (base 12) e também marcamos o tempo em minutos e segundos (base 60).

Os computadores utilizam a base 2 (sistema binário) e os programadores, por facilidade, usam em geral uma base que seja uma potência de 2, tal como 24 (base 16 ou sistema hexadecimal) ou eventualmente ainda 23 (base 8 ou sistema octal).

Na base 10, dispomos de 10 algarismos para a representação do número: 0, 1, 2, 3, 4, 5, 6, 7, 8 e 9. Na base 2, seriam apenas 2 algarismos: 0 e 1. Na base 16, seriam 16: os 10 algarismos aos quais estamos acostumados, mais os símbolos A, B, C, D, E e F, representando respectivamente 10, 11, 12, 13, 14 e 15 unidades. Generalizando, temos que uma base b qualquer disporá de b algarismos, variando entre 0 e (b-1).

A representação 125,3810 (base 10) significa $1x10^2 + 2x10^1 + 5x10^0 + 3x10^{-1} + 8x10^{-2}$

Generalizando, representamos uma quantidade N qualquer, numa dada base b, com um número tal como segue:

$N^b = a_{n-1}.b^{n-1} + ... + a_2.b^2 + a_1.b^1 + a_0.b^0 + a_{-1}.b^{-1} + a_{-2}.b^{-2} + ... + a_{-n}.b^{-n}$ sendo que

$a_{n-1}.b^{n-1} + ... + a_2.b^2 + a_1.b^1 + a_0.b^0$ é a parte inteira e

$a_{-1}.b^{-1} + a_{-2}.b^{-2} + ... + a_{-n}.b^{-n}$ é a parte fracionária.

Intuitivamente, sabemos que o maior número que podemos representar, com n algarismos, na base b, será o número composto n vezes pelo maior algarismo disponível naquela base (ou seja, b-1). Por exemplo, o maior número que pode ser representado na base 10 usando 3 algarismos será 999 (ou seja, $10^3 - 1 = 999$).

Generalizando, podemos ver que o maior número inteiro N que pode ser representado, em uma dada base b, com n algarismos (n "casas"), será $N = b^n - 1$. Assim, o maior número de 2 algarismos na base 16 será FF_{16} que, na base 10, equivale a $255_{10} = 16^2 - 1$.

Representação Binária Os computadores modernos utilizam apenas o sistema binário, isto é, todas as informações armazenadas ou processadas no computador usam apenas DUAS grandezas, representadas pelos algarismos 0 e 1. Essa decisão de projeto deve-se à maior facilidade de representação interna no computador, que é obtida através de dois diferentes níveis de tensão. Havendo apenas dois algarismos, portanto dígitos binários, o elemento mínimo de informação nos computadores foi apelidado de bit (uma contração do inglês binary digit).

Na base 2, o número "10" vale dois. Mas se $10_2 = 2_{10}$, então dez é igual a dois?

Não, dez não é e nunca será igual a dois!

Na realidade, "10" não significa necessariamente "dez". Nós estamos acostumados a associar "10" a "dez" porque estamos acostumados a usar o sistema de numeração decimal. O número 10_2 seria lido "um-zero" na base 2 e vale 2_{10} (convertido para "dois" na base dez), 10_5 seria lido "um-zero" na base 5 e vale 5_{10} (convertido para "cinco" na base dez), 10_{10} pode ser lido como "um-zero" na base 10 ou então como "dez" na base dez, 10_{16} seria lido "um-zero" na base 16 e vale 16_{10} (convertido para "dezesseis" na base dez), etc.

Portanto, 10 só será igual a dez se - e somente se - o número estiver representado na base dez!

Uma curiosidade: o número "10_b" vale sempre igual à base, porque em uma dada base b os algarismos possíveis vão sempre de 0 a (b - 1)! Como o maior algarismo possível em uma dada base b é igual a (b-1), o próximo número será (b - 1 + 1 = b) e portanto será sempre 10 e assim, numa dada base qualquer, o valor da base será sempre representado por "10"!

Obs.: Toda vez que um número for apresentado sem que seja indicado em qual sistema de numeração ele está representado, estenderemos que a base é dez. Sempre que outra base for utilizada, a base será obrigatoriamente indicada.

Um dia pode ser que os computadores se tornem obrigatórios e sejamos todos forçados por lei a estudar a aritmética em binário! Mas, mesmo antes disso, quem programa computadores precisa conhecer a representação em binário! Vamos começar entendendo as potências de dois:

Repr.Binária	**Potência**	**Repr.Decimal**
1	2^0	1
10	2^1	2
100	2^2	4
1000	2^3	8
10000	2^4	16
100000	2^5	32
1000000	2^6	64
10000000	2^7	128
100000000	2^8	256
1000000000	2^9	512
10000000000	2^{10}	1.024

Depois (e só depois) de compreender bem a tabela acima, fazendo a devida correlação com a representação decimal, é conveniente decorar os valores da tabela. As conversões entre base dois e base dez e as potências de dois são utilizadas a todo momento e seria perda de tempo estar toda hora convertendo. Da mesma forma que, uma vez entendido o mecanismo da multiplicação, decoramos a taboada, é muito mais efetivo saber de cor a tabela acima que fazer as contas de conversão toda vez que for necessário.

A representação binária é perfeitamente adequada para utilização pelos computadores. No entanto, um número representado em binário apresenta muitos bits, ficando longo e passível de erros quando manipulado por seres humanos normais como por exemplo os programadores, analistas e engenheiros de sistemas (bem, não tão normais assim ...). Para facilitar a visualização e manipulação por programadores de grandezas processadas em computadores, são usualmente adotadas as representações octal (base 8) e principalmente hexadecimal (base 16). Ressaltamos mais uma vez que o computador opera apenas na base 2 e as representações octal e hexadecimal não são usadas no computador, elas se destinam apenas à manipulação de grandezas pelos programadores.

Representação em Octal e em Hexadecimal Em projetos de informática (isto é, nos trabalhos realizados pelos programadores, analistas e engenheiros de sistemas), é usual representar quantidades usando sistemas em potências do binário (octal e principalmente hexadecimal), para reduzir o número de algarismos da representação e conseqüentemente facilitar a compreensão da grandeza e evitar erros. No sistema octal (base 8), cada tres bits são representados por apenas um algarismo octal (de 0 a 7). No sistema hexadecimal (base 16), cada quatro bits são representados por apenas um algarismo hexadecimal (de 0 a F).

A seguir, apresentamos uma tabela com os números em decimal e sua representação correspondente em binário, octal e hexadecimal:

Base 10	Base 2	Base 8	Base 16
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

Nota: a base 16 ou sistema hexadecimal pode ser indicada também por um "H" ou "h" após o número; por exemplo: FFH significa que o número FF (ou 255 em decimal) está em hexadecimal. Não confundir o "H" ou "h" com mais um dígito, mesmo porque em hexadecimal só temos algarismos até "F" e portanto não existe um algarismo "H".

Exemplo: Como seria a representação do número 16_{10} em binário, octal e hexadecimal? Solução: Seria respectivamente 10000_2, 20_8 e 10_{16}.

8.2 Conversões entre Bases

Vamos analisar agora as regras gerais para converter números entre duas bases quaisquer.

8.2.1 Conversão entre bases 2, 8 e 16

As conversões mais simples são as que envolvem bases que são potências entre si. Vamos exemplificar com a conversão entre a base 2 e a base 8. Como $2^3 = 8$, separando os bits de um número binário em grupos de tres bits (começando sempre da direita para a esquerda) e convertendo cada grupo de tres bits para seu equivalente em octal, teremos a representação do número em octal. Por exemplo:

$10101001_2 = 10.101.001_2$ (separando em grupos de 3, sempre começando da direita para a esquerda) Sabemos que $010_2 = 2_8$; $101_2 = 5_8$; $001_2 = 1_8$ portanto $10101001_2 = 251_8$

Se você ainda não sabe de cor, faça a conversão utilizando a regra geral. Vamos agora exemplificar com uma conversão entre as bases 2 e 16. Como $2^4 = 16$, basta separarmos em grupos de 4 bits (começando sempre da direita para a esquerda) e converter. Por exemplo:

$11010101101_2 = 110.1010.1101_2$ (separando em grupos de 4 bits, sempre começando da direita para a esquerda) Sabemos que $110_2 = 6_{16}$; $1010_2 = A_{16}$; $1101_2 = D_{16}$; portanto $11010101101_2 = 6AD_{16}$

Vamos agora exercitar a conversão inversa. Quanto seria $3F5_h$ (lembrar que o H está designando "hexadecimal") em octal? O método mais prático seria converter para binário e em seguida para octal. Por exemplo:

$3F5_h = 11.1111.0101_2$ (convertendo cada dígito hexadecimal em 4 dígitos binários) = = 1.111.110.1012 (agrupando de tres em tres bits) = = 1765_8 (convertendo cada grupo de três bits para seu valor equivalente em octal).

8.2.2 Conversão de Números em uma base b qualquer para a base 10

Vamos lembrar a expressão geral já apresentada:

$N_b = a_{n-1}.b^{n-1} + ... + a_2.b^2 + a_1.b^1 + a_0.b^0 + a_{-1}.b^{-1} + a_{-2}.b^{-2} + ... + a_{-n}.b^{-n}$

A melhor forma de fazer a conversão é usando essa expressão. Tomando como exemplo o número 101101_2, vamos calcular seu valor representado na base dez. Usando a expressão acima, fazemos:

$101101_2 = 1x2^5 + 0x2^4 + 1x2^3 + 1x2^2 + 0x2^1 + 1x2^0 = 32 + 0 + 8 + 4 + 0 + 1 = 45_{10}$

Podemos fazer a conversão de números em qualquer base para a base 10 usando o algoritmo acima. Exemplos:

a) Converter 4F5H para a base 10. Solução:

Lembramos que o H significa que a representação é hexadecimal (base 16). Sabemos ainda que $F_{16}=15_{10}$. Então:

$4x16^2 + 15x16^1 + 5x16^0 = 4x256 + 15x16 + 5 = 1024 + 240 + 5 = 1269_{10}$

b) Converter 3485_9 para a base 10. Solução:

$3x9^3 + 4x9^2 + 8x9^1 + 5x9^0 = 3x729 + 4x81 + 8x9 + 5 = 2187 + 324 + 72 + 5 = 2588_{10}$.

c) Converter $7G_{16}$ para a base 10. Solução:

Uma base b dispõe dos algarismos entre 0 e (b-1). Assim, a base 16 dispõe dos algarismos 0 a F e portanto o símbolo G não pertence à representação hexadecimal.

d) Converter $1001{,}01_2$ para a base 10. Solução:

$1x2^3 + 0x2^2 + 0x2^1 + 1x2^0 + 0x2^{-1} + 1x2^{-2} = 8 + 0 + 0 + 1 + 0 + 0{,}25 = 9{,}25_{10}$

e) Converter $34{,}3_5$ para a base 10. Solução:

$3x5^1 + 4x5^0 + 3x5^{-1} = 15 + 4 + 0{,}6 = 19{,}6_{10}$

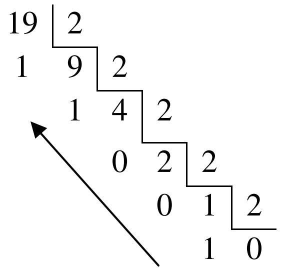
f) Converter $38{,}3_8$ para a base 10. Solução:

Uma base b dispõe dos algarismos entre 0 e (b-1). Assim, a base 8 dispõe dos algarismos 0 a 7 e portanto o algarismo 8 não existe nessa base. A representação 38,3 não existe na base 8.

8.2.3 Conversão de Números da Base 10 para uma Base b qualquer

A conversão de números da base dez para uma base qualquer emprega algoritmos que serão o inverso dos acima apresentados. O algoritmo será entendido melhor pelo exemplo que por uma descrição formal. Vamos a seguir apresentar os algoritmos para a parte inteira e para a parte fracionária:

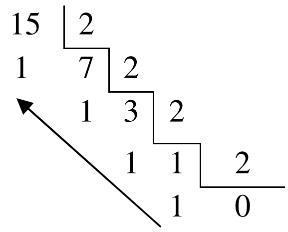
Parte Inteira: O número decimal será dividido sucessivas vezes pela base; o resto de cada divisão ocupará sucessivamente as posições de ordem 0, 1, 2 e assim por diante até que o resto da última divisão (que resulta em quociente zero) ocupe a posição de mais alta ordem. Veja o exemplo da conversão do número 19_{10} para a base 2:



$19_{10} = 010011_2$

Experimente fazer a conversão contrária (retornar para a base 10) e ver se o resultado está correto.

Parte Fracionária Se o número for fracionário, a conversão se fará em duas etapas distintas: primeiro a parte inteira e depois a parte fracionária. Os algoritmos de conversão são diferentes. O algoritmo para a parte fracionária consiste de uma série de multiplicações sucessivas do número fracionário a ser convertido pela base; a parte inteira do resultado da primeira multiplicação será o valor da primeira casa fracionária e a parte fracionária será de novo multiplicada pela base; e assim por diante, até o resultado dar zero ou até encontrarmos o número de casas decimais desejado. Por exemplo, vamos converter $15{,}65_{10}$ para a base 2, com 5 e com 10 algarismos fracionários:



$15_{10} = 01111_2$

0,65	x	2	=	1,30
0,30	x	2	=	0,60
0,60	x	2	=	1,20
0,20	x	2	=	0,40
0,40	x	2	=	0,80
0,80	x	2	=	1,60
0,60	x	2	=	1,20
0,20	x	2	=	0,40
0,40	x	2	=	0,80
0,80	x	2	=	1,60

$0{,}65_{10} = 10100_2$

$0{,}65_{10} = 1010011001_2$

$15{,}65_{10} = 1111{,}1010011001_2$

Obs.: Em ambos os casos, a conversão foi interrompida quando encontramos o número de algarismos fracionários solicitados no enunciado. No entanto, como não encontramos resultado 0 em nenhuma das multiplicações, poderíamos continuar efetuando multiplicações indefinidamente até encontrar (se encontrarmos) resultado zero. No caso de interrupção por chegarmos ao número de dígitos especificado sem encontramos resultado zero, o resultado encontrado é aproximado e essa aproximação será função do número de algarismos que calcularmos. Fazendo a conversão inversa, encontraremos:

a) Com 5 algarismos fracionários:

Parte inteira: $1111_2 = 15_{10}$

Parte fracionária: $0{,}10100_2 = 1x2^{-1} + 0x2^{-2} + 1x2^{-3} + 0x2^{-4} + 0x2^{-5} = 0{,}5 + 0{,}125 =$ 0,62510

b) Com 10 algarismos fracionários:

Parte inteira: $1111_2 = 15_{10}$

Parte fracionária: $0{,}1010011001_2 = 1x2^{-1} + 0x2^{-2} + 1x2^{-3} + 0x2^{-4} + 0x2^{-5} + 1x2^{-6} + 1x2^{-7} + 0x2^{-8} + 0x2^{-9} + 1x2^{-10} = 1/2 + 1/8 + 1/64 + 1/128 + 1/1024 = 0{,}5 + 0{,}125 + 0{,}015625 + 0{,}0078125 + 0{,}0009765625 = 0{,}6494140625_{10}$

Ou seja, podemos verificar (sem nenhuma surpresa) que, quanto maior for a quantidade de algarismos considerados, melhor será a aproximação.

Resumindo

Quanto a conversão de valores com parte inteira e parte fracionária, deve-se realizar as operações de forma separada. A conversão da parte inteira é obtida dividindo-se o número decimal pelo valor da base desejada; o resto encontrado é o algarismo menos significativo do valor na base B (mais á direita). Em seguida, divide-se o quociente encontrado pela base B; o resto é o algarismo significativo seguinte (à esquerda); e assim sucessivamente, vão-se dividindo os quocientes pelo valor da base até se obter quociente de valor zero. Em cada divisão, o resto encontrado é um algarismo significativo do número na nova base; o primeiro resto encontrado é o valor do algarismo menos significativo (mais à direita), e o último resto será o algarismo mais significativo (mais à esquerda).

Para a parte fracionária do número a ser convertido, seleciona a parte fracionária e multiplica pela base a ser convertida (B), e o valor encontrado é constituído de duas partes (a parte inteira, mesmo que seja zero e a parte fracionária). A parte inteira é o primeiro algarismo procurado (o primeiro algarismo a direita da virgula) e a parte fracionária é novamente multiplicada pela base e obtém-se um novo resultado (em duas partes distintas, a inteira e a fracionária), onde a inteira será o segundo algarismo a direita da virgula e a parte fracionária é novamente multiplicada pela base e assim sucessivamente.

8.2.4 Conversão de Números entre duas Bases quaisquer

Para converter números de uma base b para uma outra base b' quaisquer (isso é, que não sejam os casos particulares anteriormente estudados), o processo prático utilizado é converter da base b dada para a base 10 e depois da base 10 para a base b' pedida.

Exemplo: Converter 435 para $(?)_9$.

$43_5 = (4 \text{ x } 5^1 + 3 \text{ x } 5^0)_{10} = 23_{10}$ ==> 23/9 = 2 (resto 5) logo $43_5 = 23_{10} = 25_9$

Números com parte fracionária

Quando há uma parte fracionária no número a ser convertido, a melhor opção é fazer a conversão do número (parte inteira e fracionária) para a base 10 e depois fazer a conversão da base 10 para a outra base, seguindo as instruções apresentadas anteriormente.

8.3 Aritmética em Binário

8.3.1 Operação direta

A tabuada da soma aritmética em binário é muito simples.

São poucas regras:

0 + 0 = 0
0 + 1 = 1
1 + 0 = 1
1 + 1 = 0 (e "vai 1" para o dígito de ordem superior)
1 + 1 + 1 = 1 (e "vai 1" para o dígito de ordem superior)

Exemplo: Efetuar $011100_2 + 011010_2$

Observações:

a) Lembre-se: soma-se as colunas da direita para a esquerda, tal como uma soma em decimal.

b) No exemplo, são usadas, em seqüência, da direita para a esquerda, todas as regrinhas acima.

c) Na primeira linha, é indicado o resultado do "vai um".

d) Por simplicidade, no exemplo estamos considerando os dois números positivos. Solução:

```
11-----> "vai um"
011100
011010 +
 ----------
110110
```

Vamos ver agora a tabuada da subtração:

0 - 0 = 0
0 - 1 = 1 ("vem um do próximo")
1 - 0 = 1
1 - 1 = 0

Obs.: Como é impossível tirar 1 de zero, o artifício é "pedir emprestado" 1 da casa de ordem superior. Ou seja, na realidade o que se faz é subtrair 1 de 10 e encontramos 1 como resultado, devendo então subtrair 1 do dígito de ordem superior (aquele 1 que se "pediu emprestado"). Vamos lembrar que esse algoritmo é exatamente o mesmo da subtração em decimal a que já estamos acostumados desde o curso primário.

Exemplo: Efetuar $111100_2 - 011010_2$

Observações:

a) Lembre-se: subtraem-se as colunas da direita para a esquerda, tal como uma subtração em decimal.

b) No exemplo, são usadas, em seqüência, da direita para a esquerda, todas as regrinhas acima.

c) Na primeira linha, é indicado o resultado do "vem um".

d) Por simplicidade, no exemplo estamos considerando os dois números positivos. Solução:

----02-> "vem um"
11100
01010 -

10010

8.3.2 Complemento a Base

A implementação do algoritmo da subtração em computadores é complexa, requerendo vários testes. assim, em computadores a subtração em binário é feita por um artifício. O método utilizado é o "Método do Complemento a Base" que consiste em encontrar o complemento do número em relação à base e depois somar os números. Os computadores funcionam sempre na base 2, portanto o complemento à base será complemento a dois. Computadores encontram o complemento a dois de um número através de um algoritmo que pode ser assim descrito:

a) Se o número é positivo, mantenha o número (o complemento de um número positivo é o próprio número)

b) Se o número é negativo:

- inverta o número negativo ou o subtraendo na subtração (todo 1 vira zero, todo zero vira um);
- some 1 ao número em complemento ;
- some as parcelas (na subtração, some o minuendo ao subtraendo) ;
- se a soma em complemento acarretar "vai-um" ao resultado, ignore o transporte final).

Como exemplo, vamos usar o algoritmo acima na subtração 1101 - 1100 = 0001

mantém o minuendo	-->	101
inverte o subtraendo	-->	011
soma minuendo e subtraendo	-->	0000
soma 1	-->	0001
ignora o "vai-um"	-->	001

8.3.3 Multiplicação em binário

Vamos ver agora a tabuada da multiplicação:

0 x 0 = 0
0 x 1 = 0

1 x 0 = 0
1 x 1 = 1

No entanto, também a multiplicação em computadores é feita por um artifício: para multiplicar um número A por n, basta somar A com A, n vezes.

Por exemplo, 4 x 3 = 4 + 4 + 4.

E a divisão também pode ser feita por subtrações sucessivas! O que concluímos? Que qualquer operação aritmética pode ser realizada em computadores apenas através de somas (diretas ou em complemento).

8.4 Aritmética em outras bases

8.4.1 Operação direta

A tabuada da soma aritmética em uma base qualquer é simples.

Por exemplo, na base 4, as regras são:

0 + 0 = 0 0 + 1 = 1 1 + 0 = 1 1 + 1 = 2 2 + 0 = 2	... 3 + 2 = 1 [5 – 4 (da base) = 1 e "vai 1" para o dígito de ordem superior] 3 + 3 = 2 [6 – 4 (da base) = 2 e "vai 1" para o dígito de ordem superior]

Exemplo: Efetuar 02323 + 02110

Observações:

- Lembre-se: soma-se as colunas da direita para a esquerda, tal como uma soma em decimal.
- No exemplo, são usadas, em seqüência, da direita para a esquerda, todas as regrinhas acima.
- Na primeira linha, é indicado o resultado do "vai um".
- Por simplicidade, no exemplo estamos considerando os dois números positivos. Solução:

```
11-----> "vai um"
02323
02110 +
----------
11033
```

Vamos ver agora a tabuada da subtração:

0 - 0 = 0
0 - 1 = 3 ("vem um do próximo", que é igual a base: 4, logo 4 + 0 = 4, 4 – 1 = 3)
1 - 0 = 1
1 - 1 = 0
...
2 – 3 = 3 ("vem um do próximo", que é igual a base: 4, logo 4 + 2 = 6, 6 – 3 = 3)
...
3 – 2 = 1
3 – 3 = 0

Obs.: Como é impossível tirar 1 de zero, o artifício é "pedir emprestado" 1 da casa de ordem superior. Ou seja, na realidade o que se faz é subtrair 1 de 10 e encontramos 3 como resultado (1 de ordem superior é do tamanho da base, logo igual a 4), devendo então subtrair 1 do dígito de ordem superior (aquele 1 que se "pediu emprestado" e que vale 4). Vamos lembrar que esse algoritmo é exatamente o mesmo da subtração em decimal a que já estamos acostumados desde o curso primário.

Exemplo: Efetuar 21321 - 02310

Observações:

- Lembre-se: subtrai-se as colunas da direita para a esquerda, tal como uma subtração em decimal.
- No exemplo, são usadas, em seqüência, da direita para a esquerda, as regrinhas acima.
- Na primeira linha, é indicado o resultado do "vem um".
- Por simplicidade, no exemplo estamos considerando os dois números positivos. Solução:

15-----> "vem um"
21321
02310 -

13011

As mesmas regras usada para as operações em base 4 valem para as outras bases como 8, 16, 5 e 7. Observando que o maior algarismo é a (base – 1).

Outros Exemplos:

11
3AC78$_{16}$
26 441$_{16}$ **+**

610B9$_{16}$

11
34723$_{8}$
23334$_{8}$ **+**

60257$_{8}$

2 10 12 11 10
3 1 3 2 0$_{4}$
0 2 3 3 3$_{4}$ **-**
----
2 2 3 2 1$_{4}$

BIBLIOGRAFIA:

ABE, J. M. & SCALZITTI, A. & SILVA FILHO, J.I. Introdução à Lógica para a Ciência da Computação. Arte & Ciência, 2001.

ALENCAR, E.A. Iniciação à Lógica Matemática. Editora Nobel, 1995.

COPI, I.M. Introdução à Lógica. Editora Mestre Jou, 1981.

DAGHLIAN, J. Lógica e Álgebra de Boole. Editora Atlas, 1995.

IDOETA, I. I. & CAPUANO, F.G. Elementos de Eletronica Digital. Livros Erica Editora Ltda, 1999.

JONOFON, S. Raciocínio Lógico, Vol. I, 9ª Edição. Editora JONOFON, 2000.

TONIN, N. Apostila de lógica para a computação. Erechim: Universidade Regional Integrada, 2008.

Nota do Professor:

Este trabalho é um resumo do conteúdo da disciplina, para facilitar o desenvolvimento das aulas, devendo sempre ser complementado com estudos nos livros recomendados e o desenvolvimento dos exercícios indicados em sala de aula e também da resolução das listas de exercícios propostos.

Material também disponível no site do Professor:
www.waltenomartins.com.br/lmc_apo.pdf