

AMBIENTAÇÃO EM ALGORITMOS E LÓGICA DE PROGRAMAÇÃO – I

Aula 1

Prof. Me. Pietro Martins de Oliveira

- Introdução à Lógica Proposicional
 - Proposição
 - Princípio da não contradição
 - Princípio do terceiro excluído
 - Tipos de proposições
 - Conectivos Lógicos
 - Negação, conjunção e disjunção
 - Tabela Verdade
 - Expressões Lógicas

- A lógica, como ciência, estuda o conjunto de regras que regem o processo de pensar e raciocinar.
- **Lógica Matemática** é o uso da lógica formal para estudar o raciocínio matemático.
 - Linguagem simbólica (algébrica)
 - Representação matemática do pensamento
- **Objetivo** fundamental da lógica:
 - Elaborar e analisar critérios para avaliar a validade, ou não, de argumentos

- Lógica Proposicional (Cálculo Proposicional):
 - Estudo de proposições
 - Princípios, técnicas e métodos para distinguir o raciocínio correto do incorreto
- A lógica é de alta importância para a computação
 - Programas são uma sequência de passos lógico-matemáticos (algoritmos) para resolver um problema

AMBIENTAÇÃO EM ALGORITMOS E LÓGICA DE PROGRAMAÇÃO – I

Aula 1

Prof. Me. Pietro Martins de Oliveira
pietro.oliveira@unicesumar.edu.br

Lógica Proposicional

- **Proposição:**

É toda expressão que exprime um pensamento de sentido completo.

- Pode ser classificada como **V** (verdadeira) ou **F** (falsa).

Valor lógico de uma proposição: Se p é uma proposição, então $V(p)=\mathbf{V}$ ou $V(p)=\mathbf{F}$.

Exemplos de proposições:

- O Sol gira em torno da Terra.
- $-1 < 0$.
- O morcego é um mamífero e o pinguim uma ave.
- As nuvens são feitas de algodão.
- O número 13 é ímpar.

Não são proposições:

- Onde fica este endereço?
- Parabéns!
- Faça a pesquisa.

I. Princípio da não contradição:

Uma proposição não pode ser verdadeira e falsa simultaneamente.

II. Princípio do terceiro excluído:

Toda proposição ou é verdadeira ou é falsa; não existe um terceiro valor lógico.

- **Proposições Simples**

- Ao serem decompostas, seu significado torna-se incompleto
- Sem conectivos lógicos.
- Denotadas por letras minúsculas
- Ex.: “O político é honesto.”

- **Proposições Compostas**

- Ao serem decompostas, obtêm-se novas proposições com sentido completo
- Com conectivos lógicos.
- Denotadas por letras maiúsculas
- Ex.: “Todo político é ladrão **ou** eu sou louco.”

- Existem diversos conectivos lógicos para se compor proposições não-atômicas.

Conectivo	Símbolo	
NÃO	\sim \neg	Negação ou modificador
E	\wedge	Conjunção
OU	\vee	Disjunção

- Além destes, existem a implicação condicional, implicação bicondicional, disjunção exclusiva...

Aplicando os conectivos a proposições.
Considere as seguintes proposições:

- ***p***: “*Sou brasileiro*”
- ***q***: “*Tenho um bom salário*”

A negação de p	$\sim p$	Não p
A conjunção de p e q	$p \wedge q$	p e q
A disjunção de p e q	$p \vee q$	p ou q

- Uma proposição deve assumir um valor verdade:
 - Verdadeiro ou Falso.
- Para descobrir o valor (resultado) de uma proposição (expressão lógica) pode-se utilizar as Tabelas Verdade.
- **Tabela verdade:**
 - Reúne todas as combinações de valores possíveis para as proposições atômicas
 - Associa um valor verdade para cada combinação de valores, dependendo dos conectivos usados

AMBIENTAÇÃO EM ALGORITMOS E LÓGICA DE PROGRAMAÇÃO – I

Aula 1

Prof. Me. Pietro Martins de Oliveira
pietro.oliveira@unicesumar.edu.br

Operações Lógicas

Inverte o valor lógico de uma proposição

- Negação de uma proposição p : $\sim p$
- Lê-se: “**não** p ”
- Pode-se prefixar a proposição p por: “*não é verdade que*” (ou equivalente).

Exemplo:

p : Carlos é mecânico.

$\sim p$: Carlos **não** é mecânico.

$\sim p$: **Não é verdade que** Carlos é mecânico.

Tabela Verdade para a Negação

p	$\sim p$
V	F
F	V

Apresenta valor verdadeiro se, e somente se, cada componente for verdadeiro

- Conjunção entre duas proposições p e q : $p \wedge q$
- Lê-se: “ p e q ”
- Reflete noção de simultaneidade.

Valor Lógico da conjunção:

- **Verdadeira**, apenas quando p e q são simultaneamente verdadeiras.
- **Falsa**, em qualquer outro caso.

Tabela Verdade da Conjunção

p	q	$p \wedge q$
V	V	V
V	F	F
F	V	F
F	F	F

Exemplo - considere as seguintes proposições

p e **q** :

- **p** : “Linux é complicado.”
- **q** : “Programar é simples.”

Assim, temos que

- **$p \wedge q$** : “Linux é complicado **e** programar é simples.”

Assume valor verdadeiro se ao menos um dos componentes for verdadeiro

- Disjunção entre duas proposições p e q : $p \vee q$
- Lê-se: “ p ou q ”
- Reflete noção de “ao menos um”.

Valor Lógico da disjunção:

- **Verdadeira**, quando pelo menos uma das proposições é verdadeira.
- **Falsa**, apenas quando p e q são simultaneamente falsas.

Tabela Verdade da Disjunção

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

Exemplo - considere as seguintes proposições

p e **q** :

- **p** : “Linux é de código aberto.”
- **q** : “Programar é para os fortes.”

Assim, temos que

- **$p \vee q$** : “Linux é de código aberto **ou** programar é para os fortes.”

Ordem de precedência para conectivos lógicos:

1	()	Parênteses internos
2	\sim	Negação
3	\wedge	Conjunção
4	\vee	Disjunção

AMBIENTAÇÃO EM ALGORITMOS E LÓGICA DE PROGRAMAÇÃO – I

Aula 1

Prof. Me. Pietro Martins de Oliveira
pietro.oliveira@unicesumar.edu.br

Lógica Proposicional

Exemplo 1

Como aferir o valor verdade de uma expressão lógica?

Exemplo:

1) Expressão: $p \wedge \sim q$

- ***p : “Sou inteligente.”***
- ***q : “Sou bom programador.”***

Como aferir o valor verdade de uma expressão lógica?

Exemplo:

1) Expressão: $p \wedge \sim q$

- ***p : “Sou inteligente.”***
- ***q : “Sou bom programador.”***

- Devemos avaliar todas as combinações de valores verdade possíveis

1) Expressão: $p \wedge \sim q$

- p : “*Sou inteligente.*”
- q : “*Sou bom programador.*”

p	q	$\sim q$	$p \wedge \sim q$
V	V		

1) Expressão: $p \wedge \sim q$

- p : “*Sou inteligente.*”
- q : “*Sou bom programador.*”

p	q	$\sim q$	$p \wedge \sim q$
V	V		
V	F		

1) Expressão: $p \wedge \sim q$

- p : “*Sou inteligente.*”
- q : “*Sou bom programador.*”

p	q	$\sim q$	$p \wedge \sim q$
V	V		
V	F		
F	V		

1) Expressão: $p \wedge \sim q$

- p : “*Sou inteligente.*”
- q : “*Sou bom programador.*”

p	q	$\sim q$	$p \wedge \sim q$
V	V		
V	F		
F	V		
F	F		



1) Expressão: $p \wedge \sim q$

- p : “*Sou inteligente.*”
- q : “*Sou bom programador.*”

p	q	$\sim q$	$p \wedge \sim q$
V	V		
V	F		
F	V		
F	F		

1) Expressão: $p \wedge \sim q$

- p : “*Sou inteligente.*”
- q : “*Sou bom programador.*”

p	q	$\sim q$	$p \wedge \sim q$
V	V	F	
V	F		
F	V		
F	F		

1) Expressão: $p \wedge \sim q$

- p : “*Sou inteligente.*”
- q : “*Sou bom programador.*”

p	q	$\sim q$	$p \wedge \sim q$
V	V	F	
V	F	V	
F	V		
F	F		

1) Expressão: $p \wedge \sim q$

- p : “*Sou inteligente.*”
- q : “*Sou bom programador.*”

p	q	$\sim q$	$p \wedge \sim q$
V	V	F	
V	F	V	
F	V	F	
F	F		

1) Expressão: $p \wedge \sim q$

- p : “*Sou inteligente.*”
- q : “*Sou bom programador.*”

p	q	$\sim q$	$p \wedge \sim q$
V	V	F	
V	F	V	
F	V	F	
F	F	V	



1) Expressão: $p \wedge \sim q$

- p : “*Sou inteligente.*”
- q : “*Sou bom programador.*”

p	q	$\sim q$	$p \wedge \sim q$
V	V	F	
V	F	V	
F	V	F	
F	F	V	

1) Expressão: $p \wedge \sim q$

- p : “*Sou inteligente.*”
- q : “*Sou bom programador.*”

p	q	$\sim q$	$p \wedge \sim q$
V	V	F	F
V	F	V	
F	V	F	
F	F	V	

1) Expressão: $p \wedge \sim q$

- p : “*Sou inteligente.*”
- q : “*Sou bom programador.*”

p	q	$\sim q$	$p \wedge \sim q$
V	V	F	F
V	F	V	V
F	V	F	
F	F	V	

1) Expressão: $p \wedge \sim q$

- p : “*Sou inteligente.*”
- q : “*Sou bom programador.*”

p	q	$\sim q$	$p \wedge \sim q$
V	V	F	F
V	F	V	V
F	V	F	F
F	F	V	

1) Expressão: $p \wedge \sim q$

- p : “*Sou inteligente.*”
- q : “*Sou bom programador.*”

p	q	$\sim q$	$p \wedge \sim q$
V	V	F	F
V	F	V	V
F	V	F	F
F	F	V	F

1) Expressão: $p \wedge \sim q$

- p : “*Sou inteligente.*”
- q : “*Sou bom programador.*”

p	q	$\sim q$	$p \wedge \sim q$
V	V	F	V
V	F	V	V
F	V	F	F
F	F	V	V



AMBIENTAÇÃO EM ALGORITMOS E LÓGICA DE PROGRAMAÇÃO – I

Aula 1

Prof. Me. Pietro Martins de Oliveira
pietro.oliveira@unicesumar.edu.br

Lógica Proposicional

Exemplo 2

2) Expressão: $E = (p \wedge \sim q) \vee (r \wedge \sim(p \vee q))$

- Em expressões complexas, é importante saber o resultado das “subexpressões”:

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

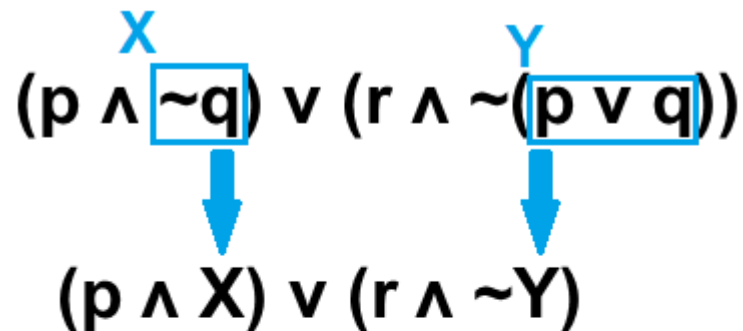
2) Expressão: $E = (p \wedge \sim q) \vee (r \wedge \sim(p \vee q))$

- Em expressões complexas, é importante saber o resultado das “subexpressões”:

$$(p \wedge \overset{X}{\boxed{\sim q}}) \vee (r \wedge \overset{Y}{\sim(\boxed{p \vee q})})$$

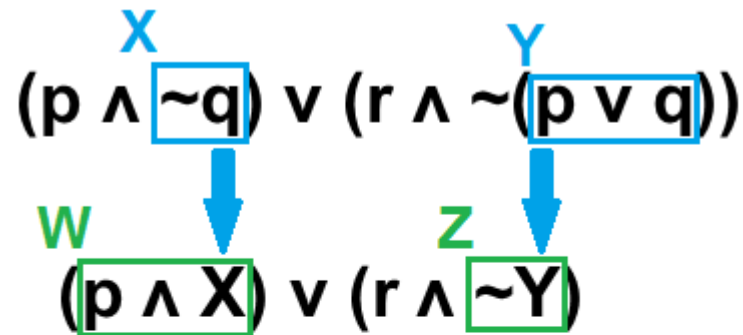
2) Expressão: $E = (p \wedge \sim q) \vee (r \wedge \sim(p \vee q))$

- Em expressões complexas, é importante saber o resultado das “subexpressões”:



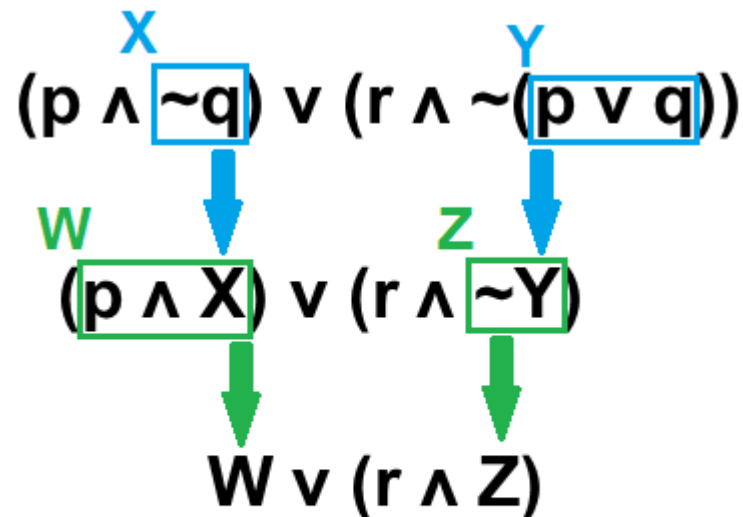
2) Expressão: $E = (p \wedge \sim q) \vee (r \wedge \sim(p \vee q))$

- Em expressões complexas, é importante saber o resultado das “subexpressões”:



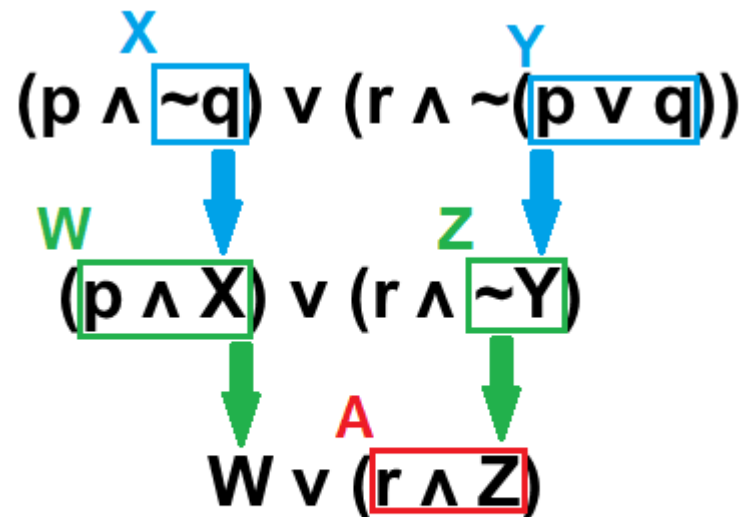
2) Expressão: $E = (p \wedge \sim q) \vee (r \wedge \sim(p \vee q))$

- Em expressões complexas, é importante saber o resultado das “subexpressões”:



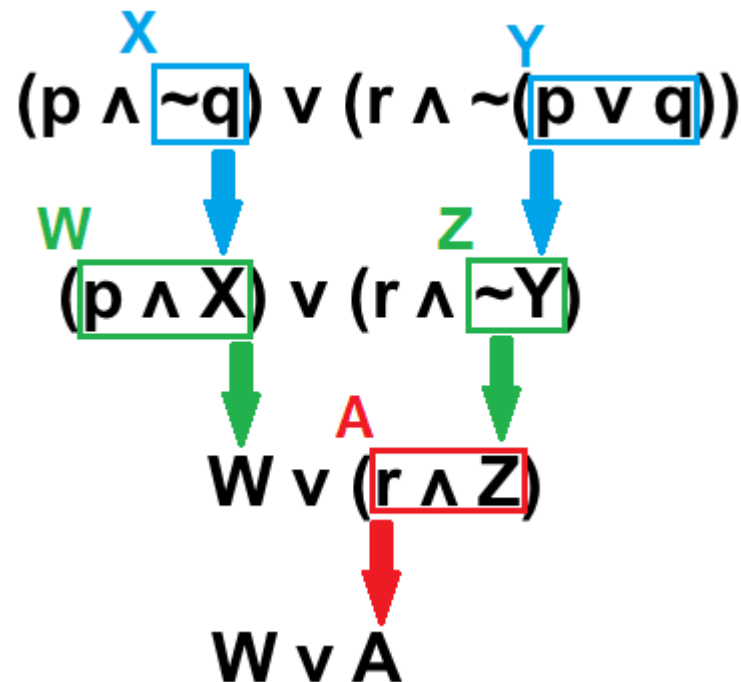
2) Expressão: $E = (p \wedge \sim q) \vee (r \wedge \sim(p \vee q))$

- Em expressões complexas, é importante saber o resultado das “subexpressões”:



2) Expressão: $E = (p \wedge \sim q) \vee (r \wedge \sim(p \vee q))$

- Em expressões complexas, é importante saber o resultado das “subexpressões”:



2) Expressão: $E = (p \wedge \sim q) \vee (r \wedge \sim(p \vee q))$

- Em expressões complexas, é importante saber o resultado das “subexpressões”:
- $X = \sim q$
- $Y = p \vee q$
- $W = p \wedge X$
- $Z = \sim Y$
- $A = r \wedge Z$
- $E = W \vee A$

$$2) E = (p \wedge \sim q) \vee (r \wedge \sim(p \vee q))$$

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

$$2) E = (p \wedge \sim q) \vee (r \wedge \sim(p \vee q))$$

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

$$2) E = (p \wedge \sim q) \vee (r \wedge \sim(p \vee q))$$

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

$$2) E = (p \wedge \sim q) \vee (r \wedge \sim(p \vee q))$$

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

2) $E = (p \wedge \sim q) \vee (r \wedge \sim(p \vee q))$

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

X

$$2) E = (p \wedge \sim q) \vee (r \wedge \sim(p \vee q))$$

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

W

$$2) E = (p \wedge \sim q) \vee (r \wedge \sim(p \vee q))$$

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

Y

2) $E = (p \wedge \sim q) \vee (r \wedge \sim(p \vee q))$

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

Z

2) $E = (p \wedge \sim q) \vee (r \wedge \sim(p \vee q))$

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

Exemplo

2) $E = (p \wedge \sim q) \vee (r \wedge \sim(p \vee q))$

Expressão
Completa!

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

- **Algoritmos e Lógica de Programação:**
 - É preciso desenvolver seu raciocínio lógico
 - É preciso desenvolver seu raciocínio matemático
 - Para desenvolver um programa, você precisa saber se suas instruções (proposições) fazem sentido
 - Estruturas condicionais e de repetição são baseadas em condições
 - Expressões lógico-relacionais

- Introdução à Lógica Proposicional
 - Proposição
 - Princípio da não contradição
 - Princípio do terceiro excluído
 - Tipos de proposições
 - Conectivos Lógicos
 - Negação, conjunção e disjunção
 - Tabela Verdade
 - Expressões Lógicas

AMBIENTAÇÃO EM ALGORITMOS E LÓGICA DE PROGRAMAÇÃO

Prof. Me. Pietro Martins de Oliveira
pietro.oliveira@unicesumar.edu.br