Lógica Matemática - LMA 0001

Rogério Eduardo da Silva - *rogerio.silva@udesc.br*
Claudio Cesar de Sá - *claudio.sa@udesc.br*

Universidade do Estado de Santa Catarina
Departamento de Ciência da Computação

14 de agosto de 2014

Conteúdo Programático:

Método de Ensino

- Aulas expositivas em sala e em laboratório
- Listas de exercícios teóricos e práticos
- Atendimento presencial (sala do professor) e/ou através da lista de emails da disciplina `lma-l@joinville.udesc.br`

Avaliações

- 3 provas teóricas (30% da média semestral cada uma) [de Alencar Filho, 2000, de Souza, 2002]:
  1. Conceitos básicos: tabelas-verdade, formas normais, implicação e equivalências lógica
  2. Argumentação lógica: regras de inferência e de equivalência, demonstração condicional e por absurdo
  3. Lógica de Primeira Ordem: Predicados, quantificadores, particularização/generalização
- Projeto de Implementação Lógica em PROLOG (10% da média semestral)
- Exame Final (caso média semestral < 7.0)
  Data prevista: **30 de Junho de 2014**

Bibliografia Básica Sugerida

- de Alencar Filho, E. (2000).
  Iniciação à Lógica Matemática.
  Editora Nobel.
- de Souza, J. N. (2002).
  Lógica para a Ciência da Computação.
  Editora Campus.

Introdução à Lógica Proposicional


PENGUINS ARE BLACK AND WHITE.
SOME OLD TV SHOWS ARE BLACK AND WHITE.
THEREFORE, SOME PENGUINS ARE OLD TV SHOWS.
GLASBERGEN

O que é Lógica?

O que é Lógica?

"*Conhecimento das formas gerais e regras gerais do pensamento correto e verdadeiro, independentemente dos conteúdos pensados; regras para demonstração científica verdadeira; regras para pensamentos não-científicos; regras sobre o modo de expor o conhecimento; regras para verificação da verdade ou falsidade de um pensamento etc.*"

[Marilena Chaui, "*Convite a Filosofia*", 2002]

Conceitos Introdutórios

- Proposição
  - conjunto de palavras ou símbolos que exprimem um pensamento (fatos ou juízos) de sentido completo
  - Exemplos
    1. A Lua é o satélite da Terra
    2. Recife é a capital de Pernambuco
    3. $\pi > \sqrt{5}$
    4. $1 + 1 = 3$
- Princípios das Proposições

  Não contradição: uma proposição não pode ser verdadeira e falsa ao mesmo tempo

  Terceiro excluído: uma proposição é sempre ou verdadeira ou falsa, não existe terceira opção

Conceitos Introdutórios

- Proposições **Verdadeiras**
  1. $1 + 1 = 2$
  2. A Lua é o satélite natural da Terra
  3. Florianópolis e Recife são capitais de estados
- Proposições **Falsas**
  1. Vasco da Gama descobriu o Brasil
  2. $3 \times 4 < 100 \div 13$
  3. $3 \div 5$ é um número inteiro

Conceitos Introdutórios

Determine "V" ou "F" para as afirmações abaixo

1. O Brasil já ganhou 5 títulos mundiais

Conceitos Introdutórios

Determine “V” ou “F” para as afirmações abaixo

1. O Brasil já ganhou 5 títulos mundiais
2. $\pi = 3.14$

Conceitos Introdutórios

Determine "V" ou "F" para as afirmações abaixo

1. O Brasil já ganhou 5 títulos mundiais
2. $\pi = 3.14$
3. Eu sempre minto

Conceitos Introdutórios

Determine “V” ou “F” para as afirmações abaixo

1. O Brasil já ganhou 5 títulos mundiais
2. $\pi = 3.14$
3. Eu sempre minto

Cuidado com paradoxos!

Conceitos Introdutórios

Determine “V” ou “F” para as afirmações abaixo

1. O Brasil já ganhou 5 títulos mundiais
2. $\pi = 3.14$
3. Eu sempre minto

Cuidado com paradoxos!

I don’t speak English

Função de Avaliação ou Interpretação Lógica

Uma proposição: uma proposição p sempre assume V ou F. Nunca os dois valores simultaneamente, ou nenhum dos dois. Sempre é alguma coisa em V ou F, logo ...

Mapeamento Binário: $f(p) \rightarrow V$ ou
$f(p) \rightarrow F$

Função de Avaliação: $f_{avalia}(p) = \{V, F\}$

Função de Avaliação $f(p)$ ou Interpretação Lógica $\Phi(p)$: $f_{avalia}(p) \equiv \Phi(p)$

Conceitos Introdutórios

Alfabeto

Símbolos Ortográficos: ( )

Constantes Lógicas: *True, False* (**V** e **F** neste curso)

Símbolos Proposicionais: $p, q, r, s, p_1, r_2, \ldots$

Conectores: $\sim (\neg), \wedge, \vee, \rightarrow, \leftrightarrow$

Conceitos Introdutórios

Fórmulas bem formadas (*fbf*)

- Constantes lógicas são fórmulas
- Símbolos proposicionais são fórmulas
- Operação negação: $\sim p$
- Operação conjunção ("e"): $p \wedge q$
- Operação disjunção ("ou"): $p \vee q$
- Operação disjunção exclusiva ("x-ou"): $p \veebar q$
- Operação condicional: $p \rightarrow q$
- Operação bicondicional:$p \leftrightarrow q$

Conceitos Introdutórios

Tabelas-Verdade

- lista todos os possíveis valores lógicos de uma proposição composta, em função das combinações de todos os possíveis valores para cada proposição simples que a compõe
- Exemplo: $p \wedge q$
  - Valores possíveis para "p" = V ou F (1 ou 0)
  - Valores possíveis para "q" = V ou F (1 ou 0)

p	q
V	V
V	F
F	V
F	F

p	q
0	0
0	1
1	0
1	1

Conceitos Introdutórios

Operações Lógicas: **negação** ($\sim$)

- inversão do valor lógico de uma proposição

p	$\sim$ **p**
V	F
F	V

Conceitos Introdutórios

Operações Lógicas: **negação** ($\sim$)

- Exemplos em Linguagem Natural:
  - p : O Sol é uma estrela
  - $\sim p$: O Sol não é uma estrela
  - p : Carlos é um mecânico
  - $\sim p$: Não é verdade que Carlos é um mecânico
  - $\sim p$: É falso que Carlos é um mecânico

Conceitos Introdutórios

Operações Lógicas: **conjunção** ($\wedge$)

- proposição composta que é verdadeira somente quando todas as proposições componentes forem verdadeiras

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

Conceitos Introdutórios

Operações Lógicas: **disjunção** ($\vee$)

- proposição composta que é verdadeira quando pelo menos uma das proposições componentes for verdadeira

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

Conceitos Introdutórios

Operações Lógicas: **disjunção exclusiva** ($\veebar$)

- proposição composta que é verdadeira somente quando exatamente uma das proposições componentes for verdadeira

p	**q**	**p $\veebar$ q**
V	V	F
V	F	V
F	V	V
F	F	F

Conceitos Introdutórios

Operações Lógicas: **condicional** ($\rightarrow$)

- proposição que representa uma relação do tipo: "**se p então q**"
- p é chamado **antecedente**
- q é chamado **consequente**
- $\rightarrow$ é chamado operador **implicação**

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

Conceitos Introdutórios

Operações Lógicas: **condicional** ($\rightarrow$)

- Exemplos em Linguagem Natural:
  - Se Maio tem 31 dias então a Terra é plana
  - Se π é um número real então o Brasil fica na América do Sul

Conceitos Introdutórios

Operações Lógicas: **bi-condicional** ($\leftrightarrow$)

- proposição que representa uma relação do tipo: "**p se e somente se q**"

p	**q**	**p $\leftrightarrow$ q**
V	V	V
V	F	F
F	V	F
F	F	V

Conceitos Introdutórios – Exercícios

Sejam as proposições

p: Está frio
q: Está chovendo

Traduzir para linguagem natural:

1. $\sim p$
2. $p \wedge q$
3. $p \vee q$
4. $p \rightarrow \sim q$
5. $p \leftrightarrow q$
6. $\sim p \wedge \sim q$

Conceitos Introdutórios – Exercícios

Traduzir para linguagem simbólica:

1. Marcos é alto e elegante
2. Marcos é alto mas não é elegante
3. Não é verdade que Marcos é baixo ou elegante
4. Marcos não é nem alto nem elegante
5. Marcos é alto ou é baixo e elegante

Conceitos Introdutórios – Exercícios

Determinar o valor lógico:

1. Roma é a capital da França ou $tg 45^\circ = 1$
2. Não é verdade que 12 é impar
3. $2 + 2 = 4 \wedge$ 11 é primo
4. Se Brasília é a capital do Brasil então $\pi = 0$
5. Se Brasília é a capital do Brasil então argentinos falam espanhol
6. $3 + 2 = 7 \ \wedge \ 5 + 5 = 10 \ \vee \ 10 > 3 \times 3$
7. $\frac{10}{2} = 5 \veebar \ \sim 1 + 1 = 3$

Construção de Tabelas-Verdade

Construção de Tabelas-Verdade

- Número de Linhas $= 2^N$, onde N $=$ número de proposições
- Dois Métodos:
  1. uma coluna por operador
  2. uma coluna por símbolo

Construção de Tabelas-Verdade

Método #1 - Uma coluna por operador.

Exemplo: $\sim (p \wedge \sim q)$

p	**q**
V	V
V	F
F	V
F	F

Construção de Tabelas-Verdade

Método #1 - Uma coluna por operador.

Exemplo: $\sim (p \wedge \sim q)$

p	**q**	$\sim$ **q**
V	V	F
V	F	V
F	V	F
F	F	V

Construção de Tabelas-Verdade

Método #1 - Uma coluna por operador.

Exemplo: $\sim (p \wedge \sim q)$

$\mathbf{p}$	$\mathbf{q}$	$\sim \mathbf{q}$	$\mathbf{p} \wedge \sim \mathbf{q}$
V	V	F	F
V	F	V	V
F	V	F	F
F	F	V	F

Construção de Tabelas-Verdade

Método #1 - Uma coluna por operador.

Exemplo: $\sim (p \wedge \sim q)$

$\mathbf{p}$	$\mathbf{q}$	$\sim \mathbf{q}$	$\mathbf{p} \wedge \sim \mathbf{q}$	$\sim (\mathbf{p} \wedge \sim \mathbf{q})$
V	V	F	F	V
V	F	V	V	F
F	V	F	F	V
F	F	V	F	V

Construção de Tabelas-Verdade

Método #2 - Uma coluna por símbolo.

Exemplo: $\sim (p \wedge \sim q)$

	1			1
$\sim$	**p**	$\wedge$	$\sim$	**q**
	V			V
	V			F
	F			V
	F			F

Construção de Tabelas-Verdade

Método #2 - Uma coluna por símbolo.

Exemplo: $\sim (p \wedge \sim q)$

	1		2	1
$\sim$	**p**	$\wedge$	$\sim$	**q**
	V		F	V
	V		V	F
	F		F	V
	F		V	F

Construção de Tabelas-Verdade

Método #2 - Uma coluna por símbolo.

Exemplo: $\sim (p \wedge \sim q)$

	1	3	2	1
$\sim$	**p**	$\wedge$	$\sim$	**q**
	V	F	F	V
	V	V	V	F
	F	F	F	V
	F	F	V	F

Construção de Tabelas-Verdade

Método #2 - Uma coluna por símbolo.

Exemplo: $\sim (p \wedge \sim q)$

4	1	3	2	1
$\sim$	**p**	$\wedge$	$\sim$	**q**
V	V	F	F	V
F	V	V	V	F
V	F	F	F	V
V	F	F	V	F

Construção de Tabelas-Verdade

Concluindo:

Exemplo: $\sim (p \wedge \sim q)$

$$P_{pq}(VV, VF, FV, FF) = (V, F, V, V)$$

ou

$$P_{pq}(00, 01, 10, 11) = (1, 1, 0, 1)$$

Construção de Tabelas-Verdade - Exercícios

1. $\sim (p \wedge q) \vee \sim (q \leftrightarrow p)$
2. $p \vee \sim r \rightarrow q \wedge \sim r$
3. $(p \rightarrow q) \wedge (q \rightarrow r) \rightarrow (p \rightarrow r)$

Construção de Tabelas-Verdade - Exercícios

1. $\sim (p \wedge q) \vee \sim (q \leftrightarrow p)$
   - $P_{pq}(VV, VF, FV, VV) = (F, V, V, V)$
2. $p \vee \sim r \rightarrow q \wedge \sim r$
3. $(p \rightarrow q) \wedge (q \rightarrow r) \rightarrow (p \rightarrow r)$

Construção de Tabelas-Verdade - Exercícios

1. $\sim (p \wedge q) \vee \sim (q \leftrightarrow p)$
   - $P_{pq}(VV, VF, FV, VV) = (F, V, V, V)$
2. $p \vee \sim r \rightarrow q \wedge \sim r$
   - $P_{pqr}(VVV, VVF, VFV, VFF, FVV, FVF, FFV, FFF) = (F, V, F, F, V, V, V, F)$
3. $(p \rightarrow q) \wedge (q \rightarrow r) \rightarrow (p \rightarrow r)$

Construção de Tabelas-Verdade - Exercícios

1. $\sim (p \wedge q) \vee \sim (q \leftrightarrow p)$
   - $P_{pq}(VV, VF, FV, VV) = (F, V, V, V)$
2. $p \vee \sim r \rightarrow q \wedge \sim r$
   - $P_{pqr}(VVV, VVF, VFV, VFF, FVV, FVF, FFV, FFF) = (F, V, F, F, V, V, V, F)$
3. $(p \rightarrow q) \wedge (q \rightarrow r) \rightarrow (p \rightarrow r)$
   - $P_{pqr}(VVV, VVF, VFV, VFF, FVV, FVF, FFV, FFF) = (V, V, V, V, V, V, V, V)$

Valor Lógico de uma Proposição

- É obtido a partir da substituição das proposições componentes por seus respectivos valores lógicos.
- Exemplo:

p: A Terra é um planeta
q: Maio tem 30 dias

qual o valor lógico da proposição:

- $\sim (p \vee q) \leftrightarrow \sim p \wedge \sim q$

Valor Lógico de uma Proposição

- É obtido a partir da substituição das proposições componentes por seus respectivos valores lógicos.
- Exemplo:

p: A Terra é um planeta
q: Maio tem 30 dias

qual o valor lógico da proposição:

- $\sim (p \vee q) \leftrightarrow \sim p \wedge \sim q$

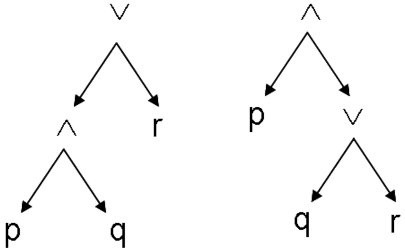
$$\sim (V \vee F) \leftrightarrow \sim V \wedge \sim F$$
$$\sim (V) \leftrightarrow F \wedge V$$
$$F \leftrightarrow F$$
$$V$$

Valor Lógico de uma Proposição - Exercícios

1. Para p = *falso* e q = *falso*, determine $(p \rightarrow q) \rightarrow (p \rightarrow p \wedge q)$
2. Para p = *verdade* e q, r = *falso*, determine $(q \leftrightarrow (r \rightarrow \sim p)) \vee ((\sim q \rightarrow p) \leftrightarrow r)$
3. Para r = *verdade*, determine: $p \rightarrow \sim q \vee r$
4. Para q = *verdade*, determine: $(p \rightarrow q) \rightarrow (\sim q \rightarrow \sim p)$

Uso de Parênteses

- A expressão $p \wedge q \vee r$ pode ser interpretada de duas formas (com valores lógicos distintos):
  1. $(p \wedge q) \vee r$
  2. $p \wedge (q \vee r)$



Uso de Parênteses

- Ordem de Precedência dos conectivos:
  1. $\sim$
  2. $\wedge \ \vee \ \veebar$
  3. $\rightarrow$
  4. $\leftrightarrow$
- Exercício: Monte a representação hierárquica para as expressões abaixo:
  1. $p \wedge \sim q$
  2. $p \wedge q \veebar r$
  3. $\sim (p \rightarrow q \vee \sim r)$
  4. $p \wedge q \rightarrow \sim p$
  5. $p \wedge (q \rightarrow \sim p)$
  6. $q \rightarrow p \leftrightarrow r \wedge s \vee \sim t$

Uso de Parênteses

- Ordem de Precedência dos conectivos:
  1. $\sim$
  2. $\wedge \ \vee \ \veebar$
  3. $\rightarrow$
  4. $\leftrightarrow$
  5. $(\ldots)$
- os símbolos de parênteses "( )" são então utilizados como modificadores da ordem de precedência.

Construção de Tabelas-Verdade - Exercícios

1. $\sim (p \vee \sim q)$
2. $\sim (p \rightarrow \sim q)$
3. $p \wedge q \rightarrow p \vee q$
4. $\sim p \rightarrow (q \rightarrow p)$
5. $(p \rightarrow q) \rightarrow p \wedge q$
6. $q \leftrightarrow \sim q \wedge p$
7. $(p \leftrightarrow \sim q) \rightarrow \sim p \wedge q$

Construção de Tabelas-Verdade - Exercícios

Suprimir o maior número de parênteses possível (de forma a não alterar a expressão):

1. $((q \leftrightarrow (r \vee q)) \leftrightarrow (p \wedge (\sim (\sim q))))$
2. $((p \wedge (\sim (\sim q))) \leftrightarrow (q \leftrightarrow (r \vee q)))$
3. $(((p \vee q) \rightarrow (\sim r)) \vee ((((\sim q) \wedge r) \wedge q)))$

Tautologias, Contradições e Contingências

Tautologia ($\blacksquare$):

- Toda proposição $P_{pqr\ldots}$ que, independente dos valores lógicos de suas proposições componentes, resulte sempre em **verdade**.
- Exemplo:

$$\sim (p \wedge \sim p)$$

$\mathbf{p}$	$\sim \mathbf{p}$	$\mathbf{p} \wedge \sim \mathbf{p}$	$\sim (\mathbf{p} \wedge \sim \mathbf{p})$
V	F	F	V
F	V	F	V

$$P_p(V, F) = (V, V) = \blacksquare$$

Tautologias, Contradições e Contingências

Outros exemplos:

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

Tautologias, Contradições e Contingências

Tautologia ($\blacksquare$):

Princípio da Substituição

- Se $P_{pqr\ldots}$ é uma tautologia então $P_{p_0 q_0 r_0 \ldots}$ também será uma tautologia, independente dos valores de $p_0, q_0, r_0, \ldots$

Tautologias, Contradições e Contingências

Contradição ($\square$):

- Toda proposição $P_{pqr...}$ que, independente dos valores lógicos de suas proposições componentes, resulte sempre em **falsidade**.
- Exemplo:

$$p \wedge \sim p$$

p	$\sim$ **p**	**p** $\wedge \sim$ **p**
V	F	F
F	V	F

$$P_p(V, F) = (F, F) = \square$$

Tautologias, Contradições e Contingências

Outros exemplos:

- $p \leftrightarrow \sim p$
- $(p \wedge q) \wedge \sim (p \vee q)$
- $\sim p \wedge (p \wedge \sim q)$

Tautologias, Contradições e Contingências

Contingência:

- Toda proposição $P_{pqr\ldots}$ que não é nem uma tautologia nem uma contradição.
- Exemplo:

$$p \rightarrow \sim p$$

$\mathbf{p}$	$\sim \mathbf{p}$	$\mathbf{p} \rightarrow \sim \mathbf{p}$
V	F	F
F	V	V

$$P_p(V, F) = (F, V)$$

Tautologias, Contradições e Contingências - Exercícios

Classifique as proposições abaixo como tautológicas, contraditórias ou contingentes:

1. $(p \rightarrow p) \vee (p \rightarrow \sim p)$
2. $p \rightarrow (p \rightarrow q \wedge \sim q)$
3. $(p \rightarrow q) \rightarrow (p \vee r \rightarrow q \vee r)$

Implicação Lógica

Implicação Lógica

- Diz-se que uma proposição $P_{pqr\ldots}$ *implica logicamente* uma proposição $Q_{pqr\ldots}$, se estas repectivamente nunca assumirem os valores lógicos "V" e "F" simultaneamente.
- Representação:

$$P_{pqr\ldots} \Rightarrow Q_{pqr\ldots}$$

- Toda proposição implica logicamente uma tautologia: $P_{pqr\ldots} \Rightarrow \blacksquare$
- Só uma contradição implica logicamente outra contradição: $\square \Rightarrow \square$
- Atenção: o símbolo "$\Rightarrow$" define uma **relação** lógica entre duas fórmulas $P_{pqr\ldots}$ e $Q_{pqr\ldots}$
- Por outro lado, o símbolo "$\rightarrow$" define uma **operação** lógica entre duas fórmulas $P_{pqr\ldots}$ e $Q_{pqr\ldots}$. Aqui tem tabela-verdade, no caso acima não.

Implicação Lógica

Sua definição operacional é dada por:

- Para verificar se duas fórmulas se relacionam logicamente entre si, isto é, $P_{pqr\ldots} \Rightarrow Q_{pqr\ldots}$, deve-se contruir as tabelas-verdade de $P_{pqr\ldots}$ e $Q_{pqr\ldots}$
- Em toda linha de avaliação de $P_{pqr\ldots}$ for Verdade, então naquela linha em $Q_{pqr\ldots}$ também deverá ser Verdade
- Em outras palavras, a definição da relação de implicação lógica ($\Rightarrow$), segue a tabela-verdade da operação ou conectivo lógico da implicação ($\rightarrow$)

Implicação Lógica

Propriedades:

- Propriedade Reflexiva: $P_{pqr\ldots} \Rightarrow P_{pqr\ldots}$
- Propriedade Transitiva: se $P_{pqr\ldots} \Rightarrow Q_{pqr\ldots}$ e $Q_{pqr\ldots} \Rightarrow R_{pqr\ldots}$ então $P_{pqr\ldots} \Rightarrow R_{pqr\ldots}$
- Para $p \wedge q, p \vee q$ e $p \leftrightarrow q$:
  - $p \wedge q \Rightarrow p \vee q$
  - $p \wedge q \Rightarrow p \leftrightarrow q$
- Para $p \leftrightarrow q, p \rightarrow q$ e $q \rightarrow p$:
  - $p \leftrightarrow q \Rightarrow p \rightarrow q$
  - $p \leftrightarrow q \Rightarrow q \rightarrow p$

Implicação Lógica

Regras de Inferência:

Adição: $$\begin{aligned} p &\Rightarrow p \vee q \\ q &\Rightarrow p \vee q \end{aligned}$$

Implicação Lógica

Regras de Inferência:

Adição:
$$p \Rightarrow p \lor q$$
$$q \Rightarrow p \lor q$$

Simplificação:
$$p \land q \Rightarrow p$$
$$p \land q \Rightarrow q$$

Implicação Lógica

Regras de Inferência:

Adição:
$$p \Rightarrow p \vee q$$
$$q \Rightarrow p \vee q$$

Simplificação:
$$p \wedge q \Rightarrow p$$
$$p \wedge q \Rightarrow q$$

Silogismo Disjuntivo:
$$(p \vee q) \wedge \sim p \Rightarrow q$$
$$(p \vee q) \wedge \sim q \Rightarrow p$$

Implicação Lógica

Regras de Inferência:

Adição: $p \Rightarrow p \vee q$
$q \Rightarrow p \vee q$

Simplificação: $p \wedge q \Rightarrow p$
$p \wedge q \Rightarrow q$

Silogismo Disjuntivo: $(p \vee q) \wedge \sim p \Rightarrow q$
$(p \vee q) \wedge \sim q \Rightarrow p$

Modus Ponens: $(p \rightarrow q) \wedge p \Rightarrow q$

Implicação Lógica

Regras de Inferência:

Adição:
$$p \Rightarrow p \vee q$$
$$q \Rightarrow p \vee q$$

Simplificação:
$$p \wedge q \Rightarrow p$$
$$p \wedge q \Rightarrow q$$

Silogismo Disjuntivo:
$$(p \vee q) \wedge \sim p \Rightarrow q$$
$$(p \vee q) \wedge \sim q \Rightarrow p$$

Modus Ponens: $(p \rightarrow q) \wedge p \Rightarrow q$

Modus Tollens: $(p \rightarrow q) \wedge \sim q \Rightarrow \sim p$

Implicação Lógica

Regras de Inferência:

Adição:
$$p \Rightarrow p \vee q$$
$$q \Rightarrow p \vee q$$

Simplificação:
$$p \wedge q \Rightarrow p$$
$$p \wedge q \Rightarrow q$$

Silogismo Disjuntivo:
$$(p \vee q) \wedge \sim p \Rightarrow q$$
$$(p \vee q) \wedge \sim q \Rightarrow p$$

Modus Ponens: $(p \rightarrow q) \wedge p \Rightarrow q$

Modus Tollens: $(p \rightarrow q) \wedge \sim q \Rightarrow \sim p$

Silogismo Hipotético: $(p \rightarrow q) \wedge (q \rightarrow r) \Rightarrow (p \rightarrow r)$

Implicação Lógica

Regras de Inferência:

Adição: $p \Rightarrow p \wedge q$
$p \Rightarrow p \vee q$

Simplificação: $p \wedge q \Rightarrow p$
$p \wedge q \Rightarrow q$

Silogismo Disjuntivo: $(p \vee q) \wedge \sim p \Rightarrow q$
$(p \vee q) \wedge \sim q \Rightarrow p$

Modus Ponens: $(p \rightarrow q) \wedge p \Rightarrow q$

Modus Tollens: $(p \rightarrow q) \wedge \sim q \Rightarrow \sim p$

Silogismo Hipotético: $(p \rightarrow q) \wedge (q \rightarrow r) \Rightarrow (p \rightarrow r)$

Princípio da Inconsistência: $\square \Rightarrow p$

- $p \wedge \sim p \Rightarrow q$

Implicação Lógica - Exercicios

1. Mostre que:
   1.1 $q \Rightarrow p \rightarrow q$
   1.2 $q \Rightarrow p \wedge q \leftrightarrow p$
2. Mostre que $p \leftrightarrow \sim q$ não implica $p \rightarrow q$
3. Mostre que $(X \neq 0 \rightarrow X = Y) \wedge X \neq Y \Rightarrow X = 0$

Equivalência Lógica

Equivalência Lógica

- Diz-se que uma proposição $P_{pqr\ldots}$ é logicamente equivalente a uma proposição $Q_{pqr\ldots}$, se as tabelas-verdade dessas proposições forem idênticas.
- Tautologias são sempre equivalentes: $\blacksquare \Leftrightarrow \blacksquare$
- Contradições são sempre equivalentes: $\square \Leftrightarrow \square$

Equivalência Lógica

Propriedades:

Dupla Negação: $p \Leftrightarrow \sim\sim p$

Equivalência Lógica

Propriedades:

Dupla Negação: $p \Leftrightarrow \sim\sim p$

Regra de CLAVIUS: $(\sim p \rightarrow p) \Leftrightarrow p$

Equivalência Lógica

Propriedades:

Dupla Negação: $p \Leftrightarrow \sim\sim p$

Regra de CLAVIUS: $(\sim p \rightarrow p) \Leftrightarrow p$

Regra da Absorção: $(p \rightarrow p \wedge q) \Leftrightarrow (p \rightarrow q)$

Equivalência Lógica

Propriedades:

Dupla Negação: $p \Leftrightarrow \sim\sim p$

Regra de CLAVIUS: $(\sim p \rightarrow p) \Leftrightarrow p$

Regra da Absorção: $(p \rightarrow p \wedge q) \Leftrightarrow (p \rightarrow q)$

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

Equivalência Lógica

Propriedades:

Dupla Negação: $p \Leftrightarrow \sim\sim p$

Regra de CLAVIUS: $(\sim p \rightarrow p) \Leftrightarrow p$

Regra da Absorção: $(p \rightarrow p \wedge q) \Leftrightarrow (p \rightarrow q)$

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

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

Equivalência Lógica

Propriedades:

Dupla Negação: $p \Leftrightarrow \sim\sim p$

Regra de CLAVIUS: $(\sim p \rightarrow p) \Leftrightarrow p$

Regra da Absorção: $(p \rightarrow p \wedge q) \Leftrightarrow (p \rightarrow q)$

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

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

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

Equivalência Lógica - Exercício

Prove que: $(p \wedge \sim q \rightarrow falso) \Leftrightarrow (p \rightarrow q)$

Equivalência Lógica - Exercício

Prove que: $(p \wedge \sim q \rightarrow \texttt{falso}) \Leftrightarrow (p \rightarrow q)$

$\mathbf{p}$	$\mathbf{q}$	$\sim \mathbf{q}$	$\mathbf{p} \wedge \sim \mathbf{q}$	$\mathbf{p} \wedge \sim \mathbf{q} \rightarrow \texttt{falso}$	$\mathbf{p} \rightarrow \mathbf{q}$	$\mathbf{p} \wedge \sim \mathbf{q} \rightarrow \texttt{falso} \Leftrightarrow \mathbf{p} \rightarrow \mathbf{q}$
V	V	F	F	V	V	V
V	F	V	V	F	F	V
F	V	F	F	V	V	V
F	F	V	F	V	V	V

Equivalência Lógica

Dada $p \rightarrow q$ temos:

Proposição Recíproca: $q \rightarrow p$

Proposição Contrária: $\sim p \rightarrow \sim q$

Proposição Contrapositiva: $\sim q \rightarrow \sim p$

Equivalência Lógica

$\mathbf{p}$	$\mathbf{q}$	$\mathbf{p \rightarrow q}$	$\mathbf{q \rightarrow p}$	$\sim \mathbf{p} \rightarrow \sim \mathbf{q}$	$\sim \mathbf{q} \rightarrow \sim \mathbf{p}$
V	V	V	V	V	V
V	F	F	V	V	F
F	V	V	F	F	V
F	F	V	V	V	V

Logo:

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

Equivalência Lógica - Exercícios

1. Determine a proposição contrapositiva
   1.1 Se X é menor que zero então X não é positivo
   1.2 Se X^2 é ímpar então X é ímpar
2. Determine:
   2.1 A contrapositiva da contrapositiva de $p \rightarrow q$
   2.2 A contrapositiva da recíproca de $p \rightarrow q$
   2.3 A contrapositiva da contrária de $p \rightarrow q$
3. Determine:
   3.1 A contrapositiva de $p \rightarrow \sim q$
   3.2 A contrapositiva de $\sim p \rightarrow q$
   3.3 A contrapositiva da recíproca de $p \rightarrow \sim q$
   3.4 A recíproca da contrapositiva de $\sim p \rightarrow \sim q$

Equivalência Lógica - Conectivos de Scheffer

Negação conjunta $(\sim p \wedge \sim q)$ também indicada por $(p \downarrow q)$, portanto

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

Negação disjunta $(\sim p \vee \sim q)$ também indicada por $(p \uparrow q)$, portanto

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

Equivalência Lógica - Exercícios

Demonstrar por tabelas-verdade as seguintes equivalências lógicas:

1. $p \wedge (p \vee q) \Leftrightarrow p$
2. $q \leftrightarrow p \vee q \Leftrightarrow p \rightarrow q$
3. $(p \rightarrow q) \vee (p \rightarrow r) \Leftrightarrow p \rightarrow q \vee r$
4. $p \veebar q \Leftrightarrow (p \vee q) \wedge \sim (p \wedge q)$
5. $(p \downarrow q) \uparrow (q \downarrow p) \Leftrightarrow q \vee p$

Método Dedutivo

Álgebra das Proposições

Propriedades da Conjunção

Idempotente	$p \wedge p \Leftrightarrow p$
Comutativa	$p \wedge q \Leftrightarrow q \wedge p$
Associativa	$(p \wedge q) \wedge r \Leftrightarrow p \wedge (q \wedge r)$
Identidade	$p \wedge \text{true} \Leftrightarrow p$ $p \wedge \text{false} \Leftrightarrow \text{false}$

Álgebra das Proposições

Propriedades da Disjunção

Idempotente	$p \lor p \Leftrightarrow p$
Comutativa	$p \lor q \Leftrightarrow q \lor p$
Associativa	$(p \lor q) \lor r \Leftrightarrow p \lor (q \lor r)$
Identidade	$p \lor \text{true} \Leftrightarrow \text{true}$ $p \lor \text{false} \Leftrightarrow p$

Álgebra das Proposições

Propriedades da Conjunção e da Disjunção

Distributiva	$p \wedge (q \vee r) \Leftrightarrow (p \wedge q) \vee (p \wedge r)$ $p \vee (q \wedge r) \Leftrightarrow (p \vee q) \wedge (p \vee r)$
Absorção	$p \wedge (p \vee q) \Leftrightarrow p$ $p \vee (p \wedge q) \Leftrightarrow p$
Regra de DE MORGAN	$\sim (p \wedge q) \Leftrightarrow \sim p \vee \sim q$ $\sim (p \vee q) \Leftrightarrow \sim p \wedge \sim q$

Álgebra das Proposições - Exercício

Apresente a negação das proposições abaixo:

- É inteligente e estuda
- É médico ou professor

Álgebra das Proposições - Exercício

Apresente a negação das proposições abaixo:

- É inteligente e estuda
- É médico ou professor

Conclusões:

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

Álgebra das Proposições

Negação da Condicional:

- Se $p \rightarrow q \Leftrightarrow \sim p \vee q$ então sua negação é dada por

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

CUIDADO!: Condicional não apresenta as propriedades idempotente, comutativa e associativa.

Álgebra das Proposições

Negação da Bicondicional:

- Se $p \leftrightarrow q \Leftrightarrow (p \rightarrow q) \wedge (q \rightarrow p)$ então sua negação é dada por

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

CUIDADO!: Bicondicional não é idempotente, mas é comutativa e associativa.

Álgebra das Proposições - Exercício

Prove:

1. $p \rightarrow q \vee r \Leftrightarrow (p \rightarrow q) \vee (p \rightarrow r)$
2. $p \rightarrow q \wedge r \Leftrightarrow (p \rightarrow q) \wedge (p \rightarrow r)$

Método Dedutivo

- Seja `falso` $\rightarrow p$ podemos concluir que

Método Dedutivo

- Seja $\texttt{falso} \rightarrow p$ podemos concluir que

$$\sim(\texttt{falso}) \vee p$$
$$\texttt{verdade} \vee p$$
$$\texttt{verdade}$$

Método Dedutivo

- Seja $\texttt{falso} \to p$ podemos concluir que

$$\begin{gathered} \sim(\texttt{falso}) \vee p \\ \texttt{verdade} \vee p \\ \texttt{verdade} \end{gathered}$$

- Seja $p \to \texttt{verdade}$ então:

Método Dedutivo

- Seja $\texttt{falso} \rightarrow p$ podemos concluir que

$$\sim (\texttt{falso}) \vee p$$
$$\texttt{verdade} \vee p$$
$$\texttt{verdade}$$

- Seja $p \rightarrow \texttt{verdade}$ então:

$$\sim p \vee \texttt{verdade}$$
$$\texttt{verdade}$$

Método Dedutivo

Demonstre pelo método dedutivo que a implicação de simplificação

$$p \wedge q \Rightarrow p$$

Método Dedutivo

Demonstre pelo método dedutivo que a implicação de simplificação

$$p \wedge q \Rightarrow p$$

$$p \wedge q \rightarrow p \quad \text{(hipótese inicial)}$$

Método Dedutivo

Demonstre pelo método dedutivo que a implicação de simplificação

$$p \wedge q \Rightarrow p$$

$$\begin{array}{rl} p \wedge q \rightarrow p & \text{(hipótese inicial)} \\ \sim (p \wedge q) \vee p & \text{(equivalência condicional)} \end{array}$$

Método Dedutivo

Demonstre pelo método dedutivo que a implicação de simplificação

$$p \wedge q \Rightarrow p$$

$$\begin{array}{rl} p \wedge q \rightarrow p & \text{(hipótese inicial)} \\ \sim (p \wedge q) \vee p & \text{(equivalência condicional)} \\ (\sim p \vee \sim q) \vee p & \text{(regra de DE MORGAN)} \end{array}$$

Método Dedutivo

Demonstre pelo método dedutivo que a implicação de simplificação

$$p \wedge q \Rightarrow p$$

$$\begin{array}{rl}
p \wedge q \rightarrow p & \text{(hipótese inicial)} \\
\sim (p \wedge q) \vee p & \text{(equivalência condicional)} \\
(\sim p \vee \sim q) \vee p & \text{(regra de DE MORGAN)} \\
\sim p \vee (\sim q \vee p) & \text{(associativa)}
\end{array}$$

Método Dedutivo

Demonstre pelo método dedutivo que a implicação de simplificação

$$p \wedge q \Rightarrow p$$

$$\begin{array}{ll}
p \wedge q \rightarrow p & \text{(hipótese inicial)} \\
\sim (p \wedge q) \vee p & \text{(equivalência condicional)} \\
(\sim p \vee \sim q) \vee p & \text{(regra de DE MORGAN)} \\
\sim p \vee (\sim q \vee p) & \text{(associativa)} \\
\sim p \vee (p \vee \sim q) & \text{(comutativa)}
\end{array}$$

Método Dedutivo

Demonstre pelo método dedutivo que a implicação de simplificação

$$p \wedge q \Rightarrow p$$

$$\begin{array}{cl}
p \wedge q \rightarrow p & \text{(hipótese inicial)} \\
\sim (p \wedge q) \vee p & \text{(equivalência condicional)} \\
(\sim p \vee \sim q) \vee p & \text{(regra de DE MORGAN)} \\
\sim p \vee (\sim q \vee p) & \text{(associativa)} \\
\sim p \vee (p \vee \sim q) & \text{(comutativa)} \\
(\sim p \vee p) \vee \sim q & \text{(associativa)}
\end{array}$$

Método Dedutivo

Demonstre pelo método dedutivo que a implicação de simplificação

$$p \wedge q \Rightarrow p$$

$$\begin{array}{rl}
p \wedge q \rightarrow p & \text{(hipótese inicial)} \\
\sim (p \wedge q) \vee p & \text{(equivalência condicional)} \\
(\sim p \vee \sim q) \vee p & \text{(regra de DE MORGAN)} \\
\sim p \vee (\sim q \vee p) & \text{(associativa)} \\
\sim p \vee (p \vee \sim q) & \text{(comutativa)} \\
(\sim p \vee p) \vee \sim q & \text{(associativa)} \\
(\texttt{verdade}) \vee \sim q & \text{(tautologia)}
\end{array}$$

Método Dedutivo

Demonstre pelo método dedutivo que a implicação de simplificação

$$p \wedge q \Rightarrow p$$

$$\begin{array}{cl}
p \wedge q \rightarrow p & \text{(hipótese inicial)} \\
\sim (p \wedge q) \vee p & \text{(equivalência condicional)} \\
(\sim p \vee \sim q) \vee p & \text{(regra de DE MORGAN)} \\
\sim p \vee (\sim q \vee p) & \text{(associativa)} \\
\sim p \vee (p \vee \sim q) & \text{(comutativa)} \\
(\sim p \vee p) \vee \sim q & \text{(associativa)} \\
(\texttt{verdade}) \vee \sim q & \text{(tautologia)} \\
\texttt{verdade} & \text{(identidade)}
\end{array}$$

Lembrar que: `verdade` $\equiv \blacksquare$

Método Dedutivo

Demonstre pelo método dedutivo que a implicação de adição

$$p \Rightarrow p \vee q$$

$$p \rightarrow p \vee q \quad \text{(hipótese inicial)}$$

Método Dedutivo

Demonstre pelo método dedutivo que a implicação de adição

$$p \Rightarrow p \vee q$$

$$\begin{array}{cl}
p \rightarrow p \vee q & \text{(hipótese inicial)} \\
\sim p \vee (p \vee q) & \text{(equivalência condicional)} \\
(\sim p \vee p) \vee q & \text{(associativa)} \\
(\texttt{verdade}) \vee q & \text{(tautologia)} \\
\texttt{verdade} & \text{(identidade)}
\end{array}$$

Método Dedutivo

Demonstre pelo método dedutivo a regra *Modus Ponens*:

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

Método Dedutivo

Demonstre pelo método dedutivo a regra *Modus Ponens*:

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

$$\begin{array}{rl}
(p \to q) \wedge p \to q & \text{(hipótese inicial)} \\
(\sim p \vee q) \wedge p \to q & \text{(equivalência condicional)} \\
p \wedge (\sim p \vee q) \to q & \text{(comutativa)} \\
(p \wedge \sim p) \vee (p \wedge q) \to q & \text{(distributiva)} \\
(\square) \vee (p \wedge q) \to q & \text{(contradição)} \\
(p \wedge q) \to q & \text{(identidade)} \\
\blacksquare &
\end{array}$$

Método Dedutivo

Demonstre pelo método dedutivo a regra *Modus Tollens*:

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

Método Dedutivo

Demonstre pelo método dedutivo a regra *Modus Tollens*:

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

$$\begin{array}{rl}
(p \rightarrow q) \wedge \sim q \rightarrow \sim p & \text{(hipótese inicial)} \\
(\sim p \vee q) \wedge \sim q \rightarrow \sim p & \text{(equivalência condicional)} \\
(\sim p \wedge \sim q) \vee (q \wedge \sim q) \rightarrow \sim p & \text{(distributiva)} \\
(\sim p \wedge \sim q) \vee \Box \rightarrow \sim p & \text{(contradição)} \\
(\sim p \wedge \sim q) \rightarrow \sim p & \text{(identidade)} \\
\blacksquare &
\end{array}$$

Método Dedutivo

Demonstre pelo método dedutivo a regra *Silogismo Disjuntivo*:

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

Método Dedutivo

Demonstre pelo método dedutivo a regra *Silogismo Disjuntivo*:

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

$$\begin{array}{cl}
(p \vee q) \wedge \sim p \rightarrow q & \text{(hipótese inicial)} \\
(p \wedge \sim p) \vee (q \wedge \sim p) \rightarrow q & \text{(distributiva)} \\
\square \vee (q \wedge \sim p) \rightarrow q & \text{(contradição)} \\
(q \wedge \sim p) \rightarrow q & \text{(identidade)} \\
\blacksquare &
\end{array}$$

Método Dedutivo

Resumo até o momento:

1. $F_{pqrs\ldots} \Leftrightarrow G_{pqrs\ldots}$ significa:
   - 1.1 Que estas duas fórmulas apresentam uma relação de equivalência verdadeira entre si, a TV de F é igual a TV de G
   - 1.2 Ainda, podemos transformar $F_{pqrs\ldots}$ em $G_{pqrs\ldots}$ usando outras regras de equivalência, e vice-versa $G_{pqrs\ldots}$ em $F_{pqrs\ldots}$
2. Quanto $F_{pqrs\ldots} \Rightarrow G_{pqrs\ldots}$ significa:
   - 2.1 Significa que $F_{pqrs\ldots}$ tem uma relação de implicação verdadeira para fórmula $G_{pqrs\ldots}$
   - 2.2 Para mostrar que isto é verdadeiro, demonstra-se que a implicação é uma tautologia, isto é:
     $F_{pqrs\ldots} \rightarrow G_{pqrs\ldots}$ terá que levar a $\blacksquare$
   - 2.3 Ou $F_{pqrs\ldots} \rightarrow G_{pqrs\ldots} \Rightarrow \blacksquare$ usando outras regras de equivalências (verdades entre duas fórmulas)

Método Dedutivo

Onde vamos usar isto?

1. Faz parte o conceito fundamental do que é um teorema lógico
2. Antecipando (mas não muito), um conjunto de fórmulas vai confirmar ou não uma conclusão (C), isto é:

$$\{F_1, F_2, F_3, ...F_n\} \vdash C$$

3. Essencialmente, isto é perguntar se:

$$\{F_1, F_2, F_3, ...F_n\} \Rightarrow C$$

4. Ou se:

$$\{F_1, F_2, F_3, ...F_n\} \rightarrow C \Leftrightarrow \blacksquare$$

Método Dedutivo

- Mas para chegar lá, faltam alguns detalhes de como transformar fórmulas entre si
- Uma fórmula transformada em outra ... mantem-se equivalente!

Método Dedutivo

Redução do números de conectivos:

1. $\wedge, \rightarrow, \leftrightarrow$ em termos de $\sim, \vee$:
   - $p \wedge q \Leftrightarrow \sim (\sim p \vee \sim q)$
   - $p \rightarrow q \Leftrightarrow \sim p \vee q$
   - $p \leftrightarrow q \Leftrightarrow \sim (\sim (\sim p \vee q) \vee \sim (\sim q \vee p))$
2. $\vee, \rightarrow, \leftrightarrow$ em termos de $\sim, \wedge$:
   - $p \vee q \Leftrightarrow \sim (\sim p \wedge \sim q)$
   - $p \rightarrow q \Leftrightarrow \sim (p \wedge \sim q)$
   - $p \leftrightarrow q \Leftrightarrow \sim (p \wedge \sim q) \wedge \sim (\sim p \wedge q)$
3. $\wedge, \vee, \leftrightarrow$ em termos de $\sim, \rightarrow$:
   - $p \vee q \Leftrightarrow \sim p \rightarrow q$
   - $p \wedge q \Leftrightarrow \sim (p \rightarrow \sim q)$
   - $p \leftrightarrow q \Leftrightarrow \sim ((p \rightarrow q) \rightarrow \sim (q \rightarrow p))$

Forma Normal das Proposições

São proposições que (no máximo) contém os conectivos $\sim, \wedge, \vee$.
Tipos:

- **FNC** Formal Normal Conjuntiva
- **FND** Formal Normal Disjuntiva

Forma Normal Conjuntiva - FNC

- contém (no máximo) os conectivos $\sim, \wedge, \vee$
- $\sim$ não aparece repetido ($\sim\sim$)
- $\sim$ não tem alcance sobre $\wedge, \vee$, ou seja, só afeta proposições simples
- $\vee$ não tem alcance sobre $\wedge$ como em $(p \vee (q \wedge r))$

Exemplos de FNC:

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

Forma Normal Conjuntiva - FNC

Como determinar a FNC equivalente?

1. Substituir os conectivos $\leftrightarrow$
2. Substituir os conectivos $\rightarrow$
3. Substituir dupla negações ($\sim\sim$)
4. Substituir negações de parênteses $\sim(X \wedge Y)$ e $\sim(X \vee Y)$
5. Aplicar a regra da distributiva onde $\vee$ tem alcance sobre $\wedge$
6. Simplificar as expressões equivalentes a $\square$ ou $\blacksquare$

Forma Normal Conjuntiva - FNC

Converter a expressão abaixo para sua forma normal conjuntiva

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

Forma Normal Conjuntiva - FNC

Converter a expressão abaixo para sua forma normal conjuntiva

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

1. $\sim ((p \vee q) \wedge \sim q) \wedge \sim (q \wedge r)$
2. $(\sim (p \vee q) \vee \sim\sim q) \wedge (\sim q \vee \sim r)$
3. $((\sim p \wedge \sim q) \vee q) \wedge (\sim q \vee \sim r)$
4. $(\sim p \vee q) \wedge (\sim q \vee q) \wedge (\sim q \vee \sim r)$
5. $(\sim p \vee q) \wedge (\blacksquare) \wedge (\sim q \vee \sim r)$
6. $(\sim p \vee q) \wedge (\sim q \vee \sim r)$

Forma Normal Conjuntiva - FNC

Converter a expressão abaixo para sua forma normal conjuntiva

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

1. $(\sim p \vee q) \leftrightarrow (\sim\sim q \vee \sim p)$
2. $(\sim p \vee q) \leftrightarrow (q \vee \sim p)$
3. $(\sim (\sim p \vee q) \vee (q \vee \sim p)) \wedge (\sim (q \vee \sim p) \vee (\sim p \vee q))$
4. $((\sim\sim p \wedge \sim q) \vee (q \vee \sim p)) \wedge ((\sim q \wedge \sim\sim p) \vee (\sim p \vee q))$
5. $((p \wedge \sim q) \vee (q \vee \sim p)) \wedge ((\sim q \wedge p) \vee (\sim p \vee q))$
6. $(p \vee (q \vee \sim p)) \wedge (\sim q \vee (q \vee \sim p)) \wedge (\sim q \vee (\sim p \vee q)) \wedge (p \vee (\sim p \vee q))$
7. $(p \vee (\sim p \vee q)) \wedge ((\sim q \vee q) \vee \sim p) \wedge (\sim q \vee (q \vee \sim p)) \wedge ((p \vee \sim p) \vee q)$
8. $((p \vee \sim p) \vee q) \wedge ((\sim q \vee q) \vee \sim p) \wedge ((\sim q \vee q) \vee \sim p) \wedge ((p \vee \sim p) \vee q)$
9. $((\blacksquare) \vee q) \wedge ((\blacksquare) \vee \sim p) \wedge ((\blacksquare) \vee \sim p) \wedge ((\blacksquare) \vee q)$
10. $(\blacksquare) \wedge (\blacksquare) \wedge (\blacksquare) \wedge (\blacksquare)$
11. $\blacksquare$

Forma Normal Conjuntiva - FNC

Converter a expressão abaixo para sua forma normal conjuntiva

$$p \leftrightarrow q \vee \sim r$$

1. $(\sim p \vee q \vee \sim r) \wedge (\sim (q \vee \sim r) \vee p)$
2. $(\sim p \vee q \vee \sim r) \wedge ((\sim q \wedge \sim\sim r) \vee p)$
3. $(\sim p \vee q \vee \sim r) \wedge ((\sim q \wedge r) \vee p)$
4. $(\sim p \vee q \vee \sim r) \wedge (\sim q \vee p) \wedge (r \vee p)$

Forma Normal Disjuntiva - FND

- contém (no máximo) os conectivos $\sim, \wedge, \vee$
- $\sim$ não aparece repetido ($\sim\sim$)
- $\sim$ não tem alcance sobre $\wedge, \vee$, ou seja, só afeta proposições simples
- $\wedge$ não tem alcance sobre $\vee$ como em $(p \wedge (q \vee r))$

Exemplos de FND:

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

Forma Normal Disjuntiva - FND

Converter a expressão abaixo para sua forma normal disjuntiva

$$(p \to q) \wedge (q \to p)$$

1. $(\sim p \vee q) \wedge (\sim q \vee p)$
2. $((\sim p \vee q) \wedge \sim q) \vee ((\sim p \vee q) \wedge p)$
3. $(\sim p \wedge \sim q) \vee (q \wedge \sim q) \vee (\sim p \wedge p) \vee (q \wedge p)$
4. $(\sim p \wedge \sim q) \vee (\Box) \vee (\Box) \vee (q \wedge p)$
5. $(\sim p \wedge \sim q) \vee (q \wedge p)$

Forma Normal Disjuntiva - FND

Converter a expressão abaixo para sua forma normal disjuntiva

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

1. $\sim ((p \vee q) \wedge \sim q) \wedge \sim (q \wedge r)$
2. $(\sim (p \vee q) \vee \sim\sim q) \wedge (\sim q \vee \sim r)$
3. $((\sim p \wedge \sim q) \vee q) \wedge (\sim q \vee \sim r)$
4. $(((\sim p \wedge \sim q) \vee q) \wedge \sim q) \vee (((\sim p \wedge \sim q) \vee q) \wedge \sim r)$
5. $(\sim p \wedge \sim q \wedge \sim q) \vee (q \wedge \sim q) \vee (\sim p \wedge \sim q \wedge \sim r) \vee (q \wedge \sim r)$
6. $(\sim p \wedge (\sim q \wedge \sim q)) \vee (\Box) \vee (\sim p \wedge \sim q \wedge \sim r) \vee (q \wedge \sim r)$
7. $(\sim p \wedge \sim q) \vee (\sim p \wedge \sim q \wedge \sim r) \vee (q \wedge \sim r)$

Método Dedutivo

- Se uma proposição só contém os conectivos $\sim, \wedge, \vee$ então se trocarmos cada símbolo $\wedge$ por $\vee$ e vice-versa, dá-se o nome de **dual** à proposição resultante.
- *Exemplo:* $\sim((p \wedge q) \vee \sim r)$ e sua dual $\sim((p \vee q) \wedge \sim r)$
- **Princípio da Dualidade:** se p e q são proposições equivalentes que só contém os conectivos $\sim, \wedge, \vee$, então suas duais p_1 e q_1 são também equivalentes.

Método Dedutivo - Exercícios

1. Simplifique as proposições
   1.1 $\sim(\sim p \rightarrow \sim q)$
   1.2 $\sim(p \vee q) \vee (\sim p \wedge q)$
2. Use o método dedutivo para demonstrar
   2.1 $(p \rightarrow q) \wedge (p \rightarrow r) \Leftrightarrow (p \rightarrow q \wedge r)$
3. Determine a FNC
   3.1 $p \rightarrow q$
   3.2 $p \leftrightarrow \sim p$
   3.3 $\sim p \downarrow (q \veebar p)$
4. Determine a FND
   4.1 $\sim(\sim p \vee \sim q)$
   4.2 $(p \rightarrow q) \vee \sim p$

Demonstração Direta (ou natural)

Demonstração Direta - Argumentos

- Dadas as proposições (*premissas*) $p_1, p_2, \ldots, p_n$ e uma conclusão q, denota-se que:

$$p_1, p_2, \ldots, p_n \vdash q$$

- **Validade de Argumento:** um argumento é considerado válido se

$$p_1 \wedge p_2 \wedge \ldots \wedge p_n \Rightarrow q$$

- Exemplo: $p \vdash p \vee q$ é sempre válido

Demonstração Direta - Argumentos

Argumentos Válidos Fundamentais

Adição (AD) $p \vdash p \vee q$

Simplificação (SIMP) $\begin{array}{c} p \wedge q \vdash p \\ p \wedge q \vdash q \end{array}$

Conjunção (CONJ) $p, q \vdash p \wedge q$

Absorção (ABS) $p \rightarrow q \vdash p \rightarrow (p \wedge q)$

Modus Ponens (MP) $p \rightarrow q, p \vdash q$

Modus Tollens (MT) $p \rightarrow q, \sim q \vdash \sim p$

Demonstração Direta - Argumentos

Argumentos Válidos Fundamentais

Silogismo Disjuntivo (SD) $p \vee q, \sim p \vdash q$
$p \vee q, \sim q \vdash p$

Silogismo Hipotético (SH) $p \rightarrow q, q \rightarrow r \vdash p \rightarrow r$

Dilema Construtivo (DC) $p \rightarrow q, r \rightarrow s, p \vee r \vdash q \vee s$

Dilema Destrutivo (DD) $p \rightarrow q, r \rightarrow s, \sim q \vee \sim s \vdash \sim p \vee \sim r$

Demonstração Direta - Argumentos

Regras de Inferência

- Escreve-se as premissas (em coluna), um traço horizontal e então escreve-se a conclusão
- Exemplo para a regra Modus Ponens (MP)

$$\begin{array}{l} p \to q \\ p \\ \hline q \end{array} \qquad\qquad \begin{array}{l} \sim p \vee r \to s \wedge \sim q \\ \sim p \vee r \\ \hline s \wedge \sim q \end{array}$$

Validade mediante Regras de Inferência

Verifique a validade para $p \rightarrow q, p \wedge r \vdash q$

Validade mediante Regras de Inferência

Verifique a validade para $p \to q, p \wedge r \vdash q$

$$\frac{p \wedge r}{p} \quad \textbf{por (SIMP)}$$

Validade mediante Regras de Inferência

Verifique a validade para $p \rightarrow q, p \wedge r \vdash q$

$$\frac{p \wedge r}{p \qquad \textbf{por (SIMP)}}$$

$$\frac{\begin{array}{l} p \rightarrow q \\ p \end{array}}{q \qquad \textbf{por (MP)}}$$

Validade mediante Regras de Inferência

Verifique a validade para $p \wedge q, p \vee r \rightarrow s \vdash p \wedge s$

Validade mediante Regras de Inferência

Verifique a validade para $p \land q, p \lor r \rightarrow s \vdash p \land s$

$$\begin{array}{ll} (1) & p \land q \\ (2) & p \lor r \rightarrow s \\ \hline \end{array}$$

Validade mediante Regras de Inferência

Verifique a validade para $p \wedge q, p \vee r \rightarrow s \vdash p \wedge s$

$$
\begin{array}{lll}
(1) & p \wedge q & \\
(2) & p \vee r \rightarrow s & \\
\hline
(3) & p & \textbf{(1) por (SIMP)}
\end{array}
$$

Validade mediante Regras de Inferência

Verifique a validade para $p \wedge q, p \vee r \rightarrow s \vdash p \wedge s$

$$
\begin{array}{lll}
(1) & p \wedge q & \\
(2) & p \vee r \rightarrow s & \\
\hline
(3) & p & \textbf{(1) por (SIMP)} \\
(4) & p \vee r & \textbf{(3) por (AD)}
\end{array}
$$

Validade mediante Regras de Inferência

Verifique a validade para $p \land q, p \lor r \to s \vdash p \land s$

$$
\begin{array}{lll}
(1) & p \land q & \\
(2) & p \lor r \to s & \\
\hline
(3) & p & \mathbf{(1)\ por\ (SIMP)} \\
(4) & p \lor r & \mathbf{(3)\ por\ (AD)} \\
(5) & s & \mathbf{(2,4)\ por\ (MP)}
\end{array}
$$

Validade mediante Regras de Inferência

Verifique a validade para $p \rightarrow (q \rightarrow r), p \rightarrow q, p \vdash r$

$$\begin{array}{ll} (1) & p \rightarrow (q \rightarrow r) \\ (2) & p \rightarrow q \\ (3) & p \\ \hline \end{array}$$

Validade mediante Regras de Inferência

Verifique a validade para $p \to (q \to r), p \to q, p \vdash r$

$$
\begin{array}{lll}
(1) & p \to (q \to r) & \\
(2) & p \to q & \\
(3) & p & \\
\hline
(4) & q \to r & \mathbf{(1, 3)\ por\ (MP)}
\end{array}
$$

Validade mediante Regras de Inferência

Verifique a validade para $p \to (q \to r), p \to q, p \vdash r$

$$
\begin{array}{lll}
(1) & p \to (q \to r) & \\
(2) & p \to q & \\
(3) & p & \\
\hline
(4) & q \to r & \mathbf{(1,3)\ por\ (MP)} \\
(5) & q & \mathbf{(2,3)\ por\ (MP)}
\end{array}
$$

Validade mediante Regras de Inferência

Verifique a validade para $p \to (q \to r), p \to q, p \vdash r$

$$
\begin{array}{lll}
(1) & p \to (q \to r) & \\
(2) & p \to q & \\
(3) & p & \\
\hline
(4) & q \to r & \mathbf{(1,3)\ por\ (MP)} \\
(5) & q & \mathbf{(2,3)\ por\ (MP)} \\
(6) & r & \mathbf{(4,5)\ por\ (MP)}
\end{array}
$$

Validade mediante Regras de Inferência

Verifique a validade para $p \rightarrow q, p \wedge q \rightarrow r, \sim(p \wedge r) \vdash \sim p$

$$\begin{array}{ll} (1) & p \rightarrow q \\ (2) & p \wedge q \rightarrow r \\ (3) & \sim(p \wedge r) \\ \hline \end{array}$$

Validade mediante Regras de Inferência

Verifique a validade para $p \rightarrow q, p \wedge q \rightarrow r, \sim (p \wedge r) \vdash \sim p$

$$
\begin{array}{ll}
(1) & p \rightarrow q \\
(2) & p \wedge q \rightarrow r \\
(3) & \sim (p \wedge r) \\
\hline
(4) & p \rightarrow p \wedge q \quad \textbf{(1) por (ABS)}
\end{array}
$$

Validade mediante Regras de Inferência

Verifique a validade para $p \to q, p \wedge q \to r, \sim (p \wedge r) \vdash \sim p$

$$
\begin{array}{lll}
(1) & p \to q & \\
(2) & p \wedge q \to r & \\
(3) & \sim (p \wedge r) & \\
\hline
(4) & p \to p \wedge q & \mathbf{(1)\ por\ (ABS)} \\
(5) & p \to r & \mathbf{(2,4)\ por\ (SH)}
\end{array}
$$

Validade mediante Regras de Inferência

Verifique a validade para $p \rightarrow q, p \wedge q \rightarrow r, \sim (p \wedge r) \vdash \sim p$

$$
\begin{array}{lll}
(1) & p \rightarrow q & \\
(2) & p \wedge q \rightarrow r & \\
(3) & \sim (p \wedge r) & \\
\hline
(4) & p \rightarrow p \wedge q & \mathbf{(1)\ por\ (ABS)} \\
(5) & p \rightarrow r & \mathbf{(2,4)\ por\ (SH)} \\
(6) & p \rightarrow p \wedge r & \mathbf{(5)\ por\ (ABS)}
\end{array}
$$

Validade mediante Regras de Inferência

Verifique a validade para $p \to q, p \wedge q \to r, \sim (p \wedge r) \vdash \sim p$

(1)	$p \to q$	
(2)	$p \wedge q \to r$	
(3)	$\sim (p \wedge r)$	
(4)	$p \to p \wedge q$	**(1) por (ABS)**
(5)	$p \to r$	**(2, 4) por (SH)**
(6)	$p \to p \wedge r$	**(5) por (ABS)**
(7)	$\sim p$	**(3, 6) por (MT)**

Validade mediante Regras de Inferência

Verifique a validade para $p \vee q \to r, r \vee q \to (p \to (s \leftrightarrow t)), p \wedge s \vdash s \leftrightarrow t$

$$\begin{array}{ll} (1) & p \vee q \to r \\ (2) & r \vee q \to (p \to (s \leftrightarrow t)) \\ (3) & p \wedge s \\ \hline \end{array}$$

Validade mediante Regras de Inferência

Verifique a validade para $p \vee q \rightarrow r, r \vee q \rightarrow (p \rightarrow (s \leftrightarrow t)), p \wedge s \vdash s \leftrightarrow t$

$$
\begin{array}{lll}
(1) & p \vee q \rightarrow r & \\
(2) & r \vee q \rightarrow (p \rightarrow (s \leftrightarrow t)) & \\
(3) & p \wedge s & \\
\hline
(4) & p & \textbf{(3) por (SIMP)} \\
(5) & p \vee q & \textbf{(4) por (AD)} \\
(6) & r & \textbf{(1, 5) por (MP)} \\
(7) & r \vee q & \textbf{(6) por (AD)} \\
(8) & p \rightarrow (s \leftrightarrow t) & \textbf{(2, 7) por (MP)} \\
(9) & s \leftrightarrow t & \textbf{(4, 8) por (MP)}
\end{array}
$$

Validade mediante Regras de Inferência

Verifique a validade para

$$p \rightarrow \sim q, \sim p \rightarrow (r \rightarrow \sim q), (\sim s \vee \sim r) \rightarrow \sim\sim q, \sim s \vdash \sim r$$

Validade mediante Regras de Inferência

Verifique a validade para

$$p \to \sim q, \sim p \to (r \to \sim q), (\sim s \vee \sim r) \to \sim\sim q, \sim s \vdash \sim r$$

$$\begin{array}{lll}
(1) & p \to \sim q & \\
(2) & \sim p \to (r \to \sim q) & \\
(3) & (\sim s \vee \sim r) \to \sim\sim q & \\
(4) & \sim s & \\
\hline
(5) & \sim s \vee \sim r & \mathbf{(4)\ por\ (AD)} \\
(6) & \sim\sim q & \mathbf{(3,5)\ por\ (MP)} \\
(7) & \sim p & \mathbf{(1,6)\ por\ (MT)} \\
(8) & r \to \sim q & \mathbf{(2,7)\ por\ (MP)} \\
(9) & \sim r & \mathbf{(6,8)\ por\ (MT)}
\end{array}$$

Validade mediante Regras de Inferência

Verifique a validade para

$$p \wedge q \to r, r \to s, t \to \sim u, t, \sim s \vee u \vdash \sim (p \wedge q)$$

Validade mediante Regras de Inferência

Verifique a validade para

$$p \wedge q \rightarrow r, r \rightarrow s, t \rightarrow \sim u, t, \sim s \vee u \vdash \sim (p \wedge q)$$

$$\begin{array}{lll}
(1) & p \wedge q \rightarrow r & \\
(2) & r \rightarrow s & \\
(3) & t \rightarrow \sim u & \\
(4) & t & \\
(5) & \sim s \vee u & \\
\hline
(6) & \sim u & \mathbf{(3,4)\ por\ (MP)} \\
(7) & \sim s & \mathbf{(5,6)\ por\ (SD)} \\
(8) & \sim r & \mathbf{(2,7)\ por\ (MT)} \\
(9) & \sim (p \wedge q) & \mathbf{(1,8)\ por\ (MT)}
\end{array}$$

Validade mediante Regras de Inferência

Verifique a validade para

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

Validade mediante Regras de Inferência

Verifique a validade para

$$p \to q, p \vee (\sim\sim r \wedge \sim\sim q), s \to \sim r, \sim (p \wedge q) \vdash \sim s \vee \sim q$$

(1)	$p \to q$	
(2)	$p \vee (\sim\sim r \wedge \sim\sim q)$	
(3)	$s \to \sim r$	
(4)	$\sim (p \wedge q)$	
(5)	$p \to p \wedge q$	**(1) por (ABS)**
(6)	$\sim p$	**(4, 5) por (MP)**
(7)	$\sim\sim r \wedge \sim\sim q$	**(2, 6) por (SD)**
(8)	$\sim\sim r$	**(7) por (SIMP)**
(9)	$\sim s$	**(3, 8) por (MT)**
(10)	$\sim s \vee \sim q$	**(9) por (AD)**

Validade mediante Regras de Inferência e Equivalência

Regra da Substituição

- Uma proposição p ou apenas parte dela pode ser substituída por uma proposição q equivalente, sendo que a proposição resultante será equivalente a p

Validade mediante Regras de Inferência e Equivalência

Equivalências Notáveis

Idempotência (ID)
$$p \Leftrightarrow p \wedge p$$
$$p \Leftrightarrow p \vee p$$

Comutação (COM)
$$p \wedge q \Leftrightarrow q \wedge p$$
$$p \vee q \Leftrightarrow q \vee p$$

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

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

Dupla Negação (DN) $p \Leftrightarrow \sim\sim p$

Validade mediante Regras de Inferência e Equivalência

Equivalências Notáveis

De Morgan (DM) $\sim (p \wedge q) \Leftrightarrow \sim p \vee \sim q$
$\sim (p \vee q) \Leftrightarrow \sim p \wedge \sim q$

Condicional (COND) $p \rightarrow q \Leftrightarrow \sim p \vee q$

Bicondicional (BICOND) $p \leftrightarrow q \Leftrightarrow (p \rightarrow q) \wedge (q \rightarrow p)$
$p \leftrightarrow q \Leftrightarrow (p \wedge q) \vee (\sim p \wedge \sim q)$

Contraposição (CP) $p \rightarrow q \Leftrightarrow \sim q \rightarrow \sim p$

Exportação–Importação (EI) $p \wedge q \rightarrow r \Leftrightarrow p \rightarrow (q \rightarrow r)$

Validade mediante Regras de Inferência e Equivalência

Demonstrar: $p \rightarrow \sim q, q \vdash \sim p$

Validade mediante Regras de Inferência e Equivalência

Demonstrar: $p \rightarrow \sim q, q \vdash \sim p$

(1)	$p \rightarrow \sim q$	
(2)	q	
(3)	$\sim\sim q \rightarrow \sim p$	**(1) por (CP)**
(4)	$q \rightarrow \sim p$	**(3) por (DN)**
(5)	$\sim p$	**(2, 4) por (MP)**

Validade mediante Regras de Inferência e Equivalência

Demonstrar: $p \rightarrow q, r \rightarrow \sim q \vdash p \rightarrow \sim r$

Validade mediante Regras de Inferência e Equivalência

Demonstrar: $p \rightarrow q, r \rightarrow \sim q \vdash p \rightarrow \sim r$

$$
\begin{array}{lll}
(1) & p \rightarrow q & \\
(2) & r \rightarrow \sim q & \\
\hline
(3) & \sim\sim q \rightarrow \sim r & \mathbf{(2)\ por\ (CP)} \\
(4) & q \rightarrow \sim r & \mathbf{(3)\ por\ (DN)} \\
(5) & p \rightarrow \sim r & \mathbf{(1,4)\ por\ (SH)}
\end{array}
$$

Validade mediante Regras de Inferência e Equivalência

Demonstrar: $p \vee (q \wedge r), p \vee q \rightarrow s \vdash p \vee s$

Validade mediante Regras de Inferência e Equivalência

Demonstrar: $p \vee (q \wedge r), p \vee q \rightarrow s \vdash p \vee s$

(1)	$p \vee (q \wedge r)$	
(2)	$p \vee q \rightarrow s$	
(3)	$(p \vee q) \wedge (p \vee r)$	**(1) por (DIST)**
(4)	$p \vee q$	**(3) por (SIMP)**
(5)	s	**(2, 4) por (MP)**
(6)	$s \vee p$	**(5) por (AD)**
(7)	$p \vee s$	**(6) por (COM)**

Validade mediante Regras de Inferência e Equivalência

Demonstrar: $p \to q, q \leftrightarrow s, t \vee (r \wedge \sim s) \vdash p \to t$

Validade mediante Regras de Inferência e Equivalência

Demonstrar: $p \rightarrow q, q \leftrightarrow s, t \vee (r \wedge \sim s) \vdash p \rightarrow t$

(1)	$p \rightarrow q$	
(2)	$q \leftrightarrow s$	
(3)	$t \vee (r \wedge \sim s)$	
(4)	$(q \rightarrow s) \wedge (s \rightarrow q)$	**(2) por (BICOND)**
(5)	$q \rightarrow s$	**(4) por (SIMP)**
(6)	$p \rightarrow s$	**(1, 5) por (SH)**
(7)	$(t \vee r) \wedge (t \vee \sim s)$	**(3) por (DIST)**
(8)	$t \vee \sim s$	**(7) por (SIMP)**
(9)	$\sim s \vee t$	**(8) por (COM)**
(10)	$s \rightarrow t$	**(9) por (COND)**
(11)	$p \rightarrow t$	**(6, 10) por (SH)**

Validade mediante Regras de Inferência e Equivalência

Princípio da Inconsistência

- Quando duas ou mais proposições não podem ser <u>simultaneamente</u> verdadeiras
- Exemplo:

$$\sim(p \vee \sim q), p \vee \sim r, q \rightarrow r$$

Validade mediante Regras de Inferência e Equivalência

Demonstrar a inconsistência entre: $\sim (p \vee \sim q), p \vee \sim r, q \rightarrow r$

(1)	$\sim (p \vee \sim q)$	
(2)	$p \vee \sim r$	
(3)	$q \rightarrow r$	
(4)	$\sim p \wedge \sim\sim q$	**(1) por (DM)**
(5)	$\sim p \wedge q$	**(4) por (DN)**
(6)	q	**(5) por (SIMP)**
(7)	r	**(3,6) por (MP)**
(8)	$\sim p$	**(5) por (SIMP)**
(9)	$\sim r$	**(2,8) por (SD)**
(10)	$r \wedge \sim r$	**(7,9) por (CONJ)**
(11)	$\square$	**(10) por (CONTR)**

Demonstração Condicional

Demonstração Condicional

- Seja o argumento

$$p_1, p_2, \ldots, p_N \vdash A \rightarrow B$$

- este só é válido se a regra

$$(p_1 \land p_2 \land \ldots \land p_N) \rightarrow (A \rightarrow B) \equiv \blacksquare$$

- Aplicando a regra da importação temos

$$(p_1 \land p_2 \land \ldots \land p_N) \land A \rightarrow B$$

- portanto, conclui-se que:

$$p_1, p_2, \ldots, p_N, A \vdash B$$

Demonstração Condicional

Demonstrar: $p \vee (q \rightarrow r), \sim r \vdash q \rightarrow p$

$$
\begin{array}{ll}
(1) & p \vee (q \rightarrow r) \\
(2) & \sim r \\
\hline
\end{array}
$$

Demonstração Condicional

Demonstrar: $p \vee (q \to r), \sim r \vdash q \to p$

$$
\begin{array}{lll}
(1) & p \vee (q \to r) & \\
(2) & \sim r & \\
(3) & q & \textbf{por (Dem.C)} \vdash p \\
\hline
\end{array}
$$

Demonstração Condicional

Demonstrar: $p \vee (q \rightarrow r), \sim r \vdash q \rightarrow p$

(1)	$p \vee (q \rightarrow r)$	
(2)	$\sim r$	
(3)	q	**por (Dem.C)** $\vdash p$
(4)	$p \vee (\sim q \vee r)$	**(1) por (COND)**
(5)	$(p \vee \sim q) \vee r$	**(4) por (ASSOC)**
(6)	$p \vee \sim q$	**(2, 5) por (SD)**
(7)	p	**(3, 6) por (SD)**

Demonstração Condicional

Demonstrar: $\sim p \rightarrow \sim q \vee r, s \vee (r \rightarrow t), \sim p \vee s, \sim s \vdash q \rightarrow t$

Demonstração Condicional

Demonstrar: $\sim p \rightarrow \sim q \vee r, s \vee (r \rightarrow t), \sim p \vee s, \sim s \vdash q \rightarrow t$

(1)	$\sim p \rightarrow \sim q \vee r$	
(2)	$s \vee (r \rightarrow t)$	
(3)	$\sim p \vee s$	
(4)	$\sim s$	
(5)	q	**por (Dem.C)** $\vdash t$
(6)	$p \rightarrow s$	**(3) por (COND)**
(7)	$\sim p$	**(4, 6) por (MT)**
(8)	$\sim q \vee r$	**(1, 7) por (MP)**
(9)	$q \rightarrow r$	**(8) por (COND)**
(10)	$r \rightarrow t$	**(2, 4) por (SD)**
(11)	$q \rightarrow t$	**(9, 10) por (SH)**
(12)	t	**(5, 11) por (MP)**

Demonstração Condicional

Demonstrar: $\sim p \rightarrow \sim q, r \rightarrow s, (\sim p \wedge t) \vee (r \wedge u) \vdash q \rightarrow s$

Demonstração Condicional

Demonstrar: $\sim p \rightarrow \sim q, r \rightarrow s, (\sim p \wedge t) \vee (r \wedge u) \vdash q \rightarrow s$

(1)	$\sim p \rightarrow \sim q$	
(2)	$r \rightarrow s$	
(3)	$(\sim p \wedge t) \vee (r \wedge u)$	
(4)	q	**por (Dem.C)** $\vdash s$
(5)	$\sim\sim p$	**(1, 4) por (MT)**
(6)	p	**(5) por (DN)**
(7)	$p \vee \sim t$	**(6) por (AD)**
(8)	$\sim\sim p \vee \sim t$	**(7) por (DN)**
(9)	$\sim (\sim p \wedge t)$	**(8) por (DM)**
(10)	$r \wedge u$	**(3, 9) por (SD)**
(11)	r	**(10) por (SIMP)**
(12)	s	**(2, 11) por (MP)**

Demonstração Indireta (ou por Absurdo)

Demonstração por Absurdo

- Consiste em admitir a negação da conclusão ($\sim Q$) como sendo uma nova premissa
- E então, demonstrar logicamente que o novo argumento é inconsistente:

$$p_1, p_2, \ldots, p_N, \sim Q \vdash \square$$

Demonstração por Absurdo

Demonstrar: $p \rightarrow \sim q, r \rightarrow q \vdash \sim (p \wedge r)$

Demonstração por Absurdo

Demonstrar: $p \rightarrow \sim q, r \rightarrow q \vdash \sim (p \wedge r)$

(1)	$p \rightarrow \sim q$	
(2)	$r \rightarrow q$	
(3)	$\sim\sim (p \wedge r)$	**por (Dem.Ab.)** $\vdash \square$
(4)	$p \wedge r$	**(3) por (DN)**
(5)	p	**(4) por (SIMP)**
(6)	r	**(4) por (SIMP)**
(7)	$\sim q$	**(1, 5) por (MP)**
(8)	q	**(2, 6) por (MP)**
(9)	$\sim q \wedge q$	**(7, 8) por (CONJ)**
(10)	$\square$	**(9) por (CONTR)**

Demonstração por Absurdo

Demonstrar: $\sim p \rightarrow q, \sim q \vee r, \sim r \vdash p \vee s$

Demonstração por Absurdo

Demonstrar: $\sim p \to q, \sim q \vee r, \sim r \vdash p \vee s$

(1)	$\sim p \to q$	
(2)	$\sim q \vee r$	
(3)	$\sim r$	
(4)	$\sim (p \vee s)$	**por (Dem.Ab.)** $\vdash \square$
(5)	$\sim p \wedge \sim s$	**(4) por (DM)**
(6)	$\sim p$	**(5) por (SIMP)**
(7)	q	**(1, 6) por (MP)**
(8)	$\sim q$	**(2, 3) por (SD)**
(9)	$q \wedge \sim q$	**(7, 8) por (CONJ)**
(10)	$\square$	**(9) por (CONTR)**

Demonstração por Absurdo

Demonstrar: $\sim p \vee q, \sim q, \sim r \rightarrow s, \sim p \rightarrow (s \rightarrow \sim t) \vdash t \rightarrow r$

Demonstração por Absurdo

Demonstrar: $\sim p \vee q, \sim q, \sim r \rightarrow s, \sim p \rightarrow (s \rightarrow \sim t) \vdash t \rightarrow r$

(1)	$\sim p \vee q$	
(2)	$\sim q$	
(3)	$\sim r \rightarrow s$	
(4)	$\sim p \rightarrow (s \rightarrow \sim t)$	
(5)	t	**por (Dem.C)** $\vdash r$
(6)	$\sim r$	**por (Dem.Ab.)** $\vdash \square$
(7)	$\sim p$	**(1, 2) por (SD)**
(8)	$s \rightarrow \sim t$	**(4, 7) por (MP)**
(9)	s	**(3, 6) por (MP)**
(10)	$\sim t$	**(8, 9) por (MP)**
(11)	$t \wedge \sim t$	**(5, 10) por (CONJ)**
(12)	$\square$	**(11) por (CONTR)**

Demonstração por Absurdo

Demonstre por absurdo:

$$\begin{array}{ll}
(1) & \sim(y \neq 1 \vee z \neq -1) \\
(2) & (x < y \wedge x > z) \wedge z = -1 \rightarrow x = 0 \\
(3) & \sim(y = 1 \vee x = 0) \vee (x < y \wedge x > z) \\
\hline
\vdash & x = 0
\end{array}$$

Demonstração por Absurdo

Demonstre por absurdo:

(1)	$\sim(y \neq 1 \vee z \neq -1)$	
(2)	$(x < y \wedge x > z) \wedge z = -1 \rightarrow x = 0$	
(3)	$\sim(y = 1 \vee x = 0) \vee (x < y \wedge x > z)$	
(4)	$x \neq 0$	**por (Dem.Ab.)** $\vdash \square$
(5)	$y = 1 \wedge z = -1$	**(1) por (DM)**
(6)	$y = 1$	**(5) por (SIMP)**
(7)	$y = 1 \vee x = 0$	**(6) por (AD)**
(8)	$x < y \wedge x > z$	**(3, 7) por (SD)**
(9)	$z = -1$	**(5) por (SIMP)**
(10)	$(x < y \wedge x > z) \wedge z = -1$	**(8, 9) por (CONJ)**
(11)	$x = 0$	**(2, 10) por (MP)**
(12)	$x \neq 0 \wedge x = 0$	**(4, 11) por (CONJ)**
(13)	$\square$	**(12) por (CONTR)**

Introdução à Lógica de Primeira Ordem

Lógica de Primeira Ordem

- Lógica de Primeira Ordem (LPO) é uma extensão à Lógica Proposicional onde cada proposição $p, q, r, \ldots$ é entendida como um conjunto de proposições pertencentes a um dado conjunto (chamado de **domínio**)

$$p(x): \;\; x \in A$$

- $p(x)$ é uma **sentença aberta** em um conjunto A, se e somente se, $p(x)$ se torna uma proposição para todo $x = a, a \in A$
- Se $a \in A$ e $\Phi(p(a)) =$ verdade diz-se que a **satisfaz** $p(x)$
- Exemplo: seja $N = 1, 2, 3, \ldots$
  - $x + 1 > 8$
  - $x^2 - 5x + 6 = 0$

Conjunto-Verdade para Sentenças Abertas

- É definido com o conjunto (V_p) de todos os elementos $a \in A$ tais que $\Phi(p(a)) =$ verdade

- Exemplo: $x + 1 > 8$ em N (números naturais)
  - $V_p = \{x \mid x \in N \wedge x + 1 > 8\} = \{8, 9, 10, \ldots\}$

- Considerações

  Condição Universal $V_p = A$, $\Phi(p(x))$ é verdade para **todo** $x \in A$
  Condição Possível $V_p \subset A$, $\Phi(p(x))$ é verdade para **algum** $x \in A$
  Condição Impossível $V_p = \phi$, $\Phi(p(x))$ é verdade para **nenhum** $x \in A$

Sentenças Abertas – Duas Variáveis

- $p(x, y)$ é uma sentença aberta em dois domínios A e B, se e somente se, $p(x, y)$ se torna uma proposição para todo par $(a, b) \in A \times B$
- Se $a \in A, b \in B$ e $\Phi(p(a, b)) =$ verdade diz-se que (a, b) satisfaz $p(x, y)$
- Exemplo: seja $A = \{1, 2, 3\}$ e $B = \{5, 6\}$
  - $x < y$
  - $y = 2x$

Conjunto-Verdade para Sentenças Abertas

- É o conjunto (V_p) de todos os elementos $(a, b) \in A \times B$ tais que $\Phi(p(a, b)) =$ verdade

$$V_p = \{(x, y) \mid x \in A \wedge y \in B \wedge \Phi(p(x, y))\}$$

- Considerações

  Condição Universal $V_p = A \times B$, $\Phi(p(x, y))$ é verdade para **todo** $(x, y) \in A \times B$

  Condição Possível $V_p \subset A \times B$, $\Phi(p(x, y))$ é verdade para **algum** $(x, y) \in A \times B$

  Condição Impossível $V_p = \phi$, $\Phi(p(x, y))$ é verdade para **nenhum** $(x, y) \in A \times B$

Sentenças Abertas – N-Variáveis

- $p(x_1, x_2, \ldots, x_N)$ é uma sentença aberta em N domínios $A_1, A_2, \ldots, A_N$, se e somente se, $p(x_1, x_2, \ldots, x_N)$ se torna uma proposição para toda tupla $(a_1, a_2, \ldots, a_N) \in A_1 \times A_2 \times \ldots \times A_N$

Exercícios

- Determinar o conjunto-verdade (V_p) para
  1. $2x = 6$
  2. $x - 1 < 4$
  3. $x^2 < 25$
- ...considerando:
  - A = números naturais
  - $A = \{1, 3, 4, 7, 9, 11\}$

Operações Lógicas sobre Sentenças Abertas

- **Conjunção**

$$V_{p\wedge q} = V_p \cap V_q = \{x \in A \mid p(x)\} \cap \{x \in A \mid q(x)\}$$

- **Disjunção**

$$V_{p\vee q} = V_p \cup V_q = \{x \in A \mid p(x)\} \cup \{x \in A \mid q(x)\}$$

- **Negação**

$$V_{\sim p} = \bar{A} = \{x \in A \mid A - p(x)\}$$

Operações Lógicas sobre Sentenças Abertas

- **Condicional**
  - Se $p(x) \rightarrow q(x)$ então $\sim p(x) \vee q(x)$

$$V_{p \rightarrow q} = V_{\sim p} \cup V_q = \{x \in A \mid A - p(x)\} \cup \{x \in A \mid q(x)\}$$

- **Bicondicional**
  - Se $p(x) \leftrightarrow q(x)$ então $(p(x) \rightarrow q(x)) \wedge (q(x) \rightarrow p(x))$

$$V_{p \leftrightarrow q} = (V_{\sim p} \cup V_q) \cap (V_{\sim q} \cup V_p)$$

A álgebra das proposições que era válida para proposições atômicas e compostas, continua válida para sentenças abertas

Exercícios

- Determina o conjunto-verdade para $A = \{1 \dots 10\}$
  1. $x < 7 \wedge$ x é ímpar
  2. x é par $\wedge\, x + 2 \leq 10$
  3. x é primo $\vee\, x + 5 \in A$
  4. $\sim$ (x é primo)
  5. $\sim (x^2 - 3x = 0)$
  6. $x + 5 \in A \rightarrow x < 0$
  7. $x^2 < 12 \leftrightarrow x^2 - 5x + 6 = 0$

Quantificadores

- Expressam relações lógicas de quantidade entre variáveis e seus respectivos domínios

- Tipos:
  - Quantificador Universal ($\forall$)
  - Quantificador Existencial ($\exists$)

Quantificador Universal

- Para uma sentença aberta $p(x)$ em um conjunto não-vazio A, onde para **todo** elemento $x \in A$, temos $\Phi(p(x)) =$ verdade
  - $\Phi(\forall x \in A : p(x)) \equiv p(x_1) \wedge p(x_2) \wedge p(x_3) \wedge \ldots \wedge p(x_N)$
  - ou simplesmente $\Phi(\forall x : p(x))$
  - conclui-se que $V_p = A$

- Exemplo: "*Todo x é mortal no domínio dos seres humanos*"

Quantificador Universal - Exercício

- Seja D = [10 . . . 15]

$$\forall x \in D : \frac{x}{2} \geq 5$$

Interprete e avalia a validade da expressão acima

Quantificador Universal - Exercício

- Seja D = [10 ... 15]

$$\forall x \in D : \frac{x}{2} \geq 5$$

  Interprete e avalia a validade da expressão acima

- $\Phi(\forall x \in D : \frac{x}{2} \geq 5) = \Phi(\frac{10}{2} \geq 5) \wedge \Phi(\frac{11}{2} \geq 5) \wedge \Phi(\frac{12}{2} \geq 5) \wedge \Phi(\frac{13}{2} \geq 5) \wedge \Phi(\frac{14}{2} \geq 5) \wedge \Phi(\frac{15}{2} \geq 5)$

Quantificador Universal - Exercício

- Seja D = [10 . . . 15]

$$\forall x \in D : \frac{x}{2} \geq 5$$

  Interprete e avalia a validade da expressão acima

- $\Phi(\forall x \in D : \frac{x}{2} \geq 5) = \Phi(\frac{10}{2} \geq 5) \wedge \Phi(\frac{11}{2} \geq 5) \wedge \Phi(\frac{12}{2} \geq 5) \wedge \Phi(\frac{13}{2} \geq 5) \wedge \Phi(\frac{14}{2} \geq 5) \wedge \Phi(\frac{15}{2} \geq 5)$

- $\Phi(\forall x \in D : \frac{x}{2} \geq 5) = V \wedge V \wedge V \wedge V \wedge V \wedge V = V$

Quantificador Existencial

- Para uma sentença aberta $p(x)$ em um conjunto não-vazio A, onde para **pelo menos um** elemento $x \in A$, temos $\Phi(p(x)) = \text{verdade}$
  - $\Phi(\exists x \in A : p(x)) \equiv p(x_1) \vee p(x_2) \vee p(x_3) \vee \ldots \vee p(x_N)$
  - ou simplesmente $\Phi(\exists x : p(x))$
  - conclui-se que $V_p \subset A$

- Exemplo: "*Existe pelo menos um x que é homem no domínio dos seres humanos*"

Quantificador Existencial - Exercício

- Seja D = [10 ... 15]

$$\exists x \in D : mod(x, 3) = 0$$

Interprete e avalia a validade da expressão acima

Quantificador Existencial - Exercício

- Seja D = [10 . . . 15]

$$\exists x \in D : mod(x, 3) = 0$$

  Interprete e avalia a validade da expressão acima

- $\Phi(\exists x \in D : mod(x, 3) = 0) = \Phi(mod(10, 3) = 0) \vee \Phi(mod(11, 3) = 0) \vee \Phi(mod(12, 3) = 0) \vee \Phi(mod(13, 3) = 0) \vee \Phi(mod(14, 3) = 0) \vee \Phi(mod(15, 3) = 0)$

- $\Phi(\exists x \in D : mod(x, 3) = 0) = F \vee F \vee V \vee F \vee F \vee V = V$

Variável Aparente e Variável Livre

- Se há um quantificador incidindo sobre uma variável, essa se chama **variável aparente**. Caso contrário, chama-se **variável livre**
- Em outras palavras, variáveis aparentes estarão sempre associadas a domínios, e portanto para tais é possível a sua interpretação (Φ)
- Exemplos:

$$\exists x \in A : 3x + 1 > 10$$

$$x + 3 = -4$$

- Princípio da Substituição de Variáveis Aparentes
  - $\forall x \in A : p(x) \Leftrightarrow \forall y \in A : p(y)$
  - $\exists x \in A : p(x) \Leftrightarrow \exists y \in A : p(y)$

Quantificador de Existência e Unicidade

- Para $x^2 = 16$ sobre o conjunto dos números reais ($\Re$) temos:

$$4^2 = 16 \wedge (-4)^2 = 16 \Leftrightarrow 4 \neq -4$$

portanto conclui-se que:

$$\exists x, y \in \Re : (x^2 = 16 \wedge y^2 = 16 \wedge x \neq y)$$

- Para $x^3 = 27$ temos:

$$\exists x \in \Re : (x^3 = 27)$$

$$(x^3 = 27) \wedge (y^3 = 27) \rightarrow (x = y)$$

portanto:

$$\exists! x \in \Re : (x^3 = 27)$$

Negação de Quantificadores

- Regra de DE MORGAN para quantificadores
- $\sim (\forall x \in A : p(x)) \Leftrightarrow \exists x \in A :\sim p(x)$
- $\sim (\exists x \in A : p(x)) \Leftrightarrow \forall x \in A :\sim p(x)$

Quantificadores – N-Variáveis

Quantificação Parcial

$$\exists x \in A : (2x + y < 7),$$
$$\text{onde } A = \{1 \ldots 5\}$$

Quantificação Múltipla

$$\exists x \in A, \forall y \in B : (2x + y < 7),$$
$$\text{onde } A = \{1 \ldots 5\}, B = \{3, 4, 5\}$$

Negação de Múltiplos Quantificadores

- $\sim (\forall x \exists y : p(x,y)) \Leftrightarrow$
- $\exists x \sim (\exists y : p(x,y)) \Leftrightarrow$
- $\exists x \forall y : \sim p(x,y)$

Lógica de Primeira Ordem

Slides do Prof. José Augusto Baranauskas
Fonte: `http://dfm.ffclrp.usp.br/~augusto`
(Modificados para se adequarem a esta disciplina com autorização do autor)

Lógica Proposicional vs. Lógica de Predicados

- Na lógica proposicional, utilizamos *proposições* para a representação de conceitos. Ex.: p : o céu é azul

- Na lógica de predicados (ou lógica de primeira ordem) utilizaremos:
  1. Objetos pertencentes a um dado domínio (D)
  2. Fatos sobre objetos
  3. Relações ou relacionamento entre os objetos de domínios:

$$D = D_1 \cup D_2 \cup \ldots \cup D_N$$

Lógica Proposicional vs. Lógica de Predicados

- O objetivo (motivação) de se utilizar LPO é aumentar a capacidade de representação de conceitos

- "**Todo homem é mortal**"

- "**Sócrates é um homem**"

- Logo .... "**Sócrates é um mortal**"

Lógica Proposicional vs. Lógica de Predicados

- O objetivo (motivação) de se utilizar LPO é aumentar a capacidade de representação de conceitos
- "**Todo homem é mortal**"

$$\forall x \in HUMANIDADE : (homem(x) \rightarrow mortal(x))$$

- "**Sócrates é um homem**"

$$homem(socrates)$$

- Logo .... "**Sócrates é um mortal**"

$$homem(socrates) \rightarrow mortal(socrates) \text{ por Particularização Universal}$$

$$mortal(socrates) \text{ por Modus Ponens}$$

Lógica de Predicados

- Problema: representar o fato de que todas as mulheres do DCC são bonitas
- Lista de mulheres do DCC: Ana, Laura, Cláudia, Bianca, Fernanda, Paula, Joana, Maria, ...
- Antes, no cálculo proposicional, seria necessário criar uma proposição para cada caso:
  1. p: "*Ana é bonita*"
  2. q: "*Laura é bonita*"
  3. r: "*Cláudia é bonita*"
  4. s: "*Bianca é bonita*"
  5. e assim por diante ...
- Fácil perceber que rapidamente faltarão símbolos proposicionais !

Lógica de Predicados

- Solução: representar o fato através de variáveis relacionadas com o domínio 'DCC '
- $\forall x \in DCC : mulher(x) \rightarrow bonita(x)$
- E ainda, como o domínio é conhecido e fixo, é usual a representação do fato sem explicitar o domínio:

$$\forall x : mulher(x) \rightarrow bonita(x)$$

Lógica de Predicados - Representação

Símbolos Constantes representam objetos específicos do domínio. São representados por letras minúsculas ou números: **a, maria, bola, 10, 7.43**

Símbolos Variáveis representam objetos não-específicos (instanciáveis) que podem assumir apenas os valores de um domínio. São representados por letras maiúsculas: **X, Y, PESSOA**

Símbolos Funcionais representam funções no domínio D: $f : D^N \mapsto D$ onde N é o número de argumentos da função (*Aridade*).

Não possuem valor lógico. Exemplo:

$$idade(joao) \mapsto \Phi(idade(joao)) = 23$$

Símbolos Predicados representam relação ou propriedade p de um ou mais objetos no domínio D $p : D^N \mapsto \{V, F\}$. São representados por nome que iniciam com letra minúscula.

Exemplo:

$$gosta(X, Y)$$

$$empresta(Fulano, Objeto, Alguem)$$

Lógica de Predicados - Representação

Termos é o menor elemento para construção de fatos e regras de primeira ordem. Constantes, variáveis e funções são exemplos de termos.
É denominado *tupla* a um conjunto de termos: $(t_1, t_2, \ldots, t_N)$

Átomo é um símbolo predicado aplicado a uma tupla de termos: $p(t_1, t_2, \ldots, t_N)$. Exemplos:

$$gosta(joao, maria)$$

$$irmao(pedro, X)$$

$$empresta(maria, livro, mae(joao))$$

Símbolo de Igualdade usado para indicar que dois símbolos se referem ao mesmo objeto: $pai(joao) = henrique$

Símbolos Conectivos continuam válidos os símbolos $\wedge, \vee, \veebar, \sim, \rightarrow, \leftrightarrow$ com o acréscimo dos símbolos $\forall, \exists$

Lógica de Predicados - Propriedades

Particularização dada uma fórmula com variáveis aparentes (associadas a um domínio), **particularizar** significa associar um dos elementos pertencentes ao domínio à variável. Denota-se por **variável/valor**. Tipos: Particularização Universal (PU) ou Particularização Existencial (PE)
Por exemplo: dado $\forall X \in D : gato(X)$ e $D = [frajola, felix, garfield]$; uma possível PU é dada por $gato(frajola) \;\; PU \; (X/frajola)$

Generalização dado um domínio (ou vários), **generalizar** uma fórmula (para $\forall$ ou $\exists$) desde que as variáveis envolvidas não estejam relacionadas a outras já existentes. Tipos: Generalização Universal (GU) ou Generalização Existencial (GE)
Exemplo: dado o conjunto $D = [frajola, garfield, snoopy]$; uma possível GE seria $\exists X \in D : gato(X) \;\; GU \; em \; D$

Lógica de Predicados - Exercícios $LN \Rightarrow LPO$

Traduzir da linguagem natural (LN) para linguagem de primeira ordem (LPO):

1. Todos os homens são mortais
2. Alguns gatos são amarelos
3. Nenhuma baleia é peixe
4. Nem tudo que reluz é ouro
5. Meninas e meninos gostam de brincar
6. Leite e banana são nutritivos

Lógica de Predicados - Exercícios $LN \Rightarrow LPO$

Traduzir da linguagem natural (LN) para linguagem de primeira ordem (LPO):

1. Todos os homens são mortais

$$\forall X : (homem(X) \rightarrow mortal(X))$$

2. Alguns gatos são amarelos

$$\exists X : (gato(X) \wedge amarelo(X))$$

3. Nenhuma baleia é peixe

$$\forall X : (baleia(X) \rightarrow \sim peixe(X))$$

4. Nem tudo que reluz é ouro

$$\exists X : (reluz(X) \wedge \sim ouro(X))$$

$$\sim (\forall X : (reluz(X) \rightarrow ouro(X)))$$

5. Meninas e meninos gostam de brincar

$$\forall X : (menino(X) \vee menina(X) \rightarrow gosta(X, brincar))$$

6. Leite e banana são nutritivos

$$\forall X : (leite(X) \vee banana(X) \rightarrow nutritivo(X))$$

Lógica de Predicados - Exercícios $LPO \Rightarrow LN$

Traduzir da linguagem de primeira ordem (LPO) para linguagem natural (LN):

1. $\forall X \exists Y : (pessoa(X) \wedge pessoa(Y) \wedge engana(X, Y) \rightarrow engana(X, X))$
2. $\forall X \exists Y : (humano(X) \rightarrow erro(Y) \wedge faz(X, Y))$
3. $\forall X \exists Y : (erro(Y) \wedge \sim faz(X, Y) \rightarrow \sim humano(X))$

Lógica de Predicados - Exercícios $LPO \Rightarrow LN$

Traduzir da linguagem de primeira ordem (LPO) para linguagem natural (LN):

1. $\forall X \exists Y : (pessoa(X) \land pessoa(Y) \land engana(X, Y) \rightarrow engana(X, X))$

   "Pessoas que enganam outras pessoas, enganam a si mesmas"

2. $\forall X \exists Y : (humano(X) \rightarrow erro(Y) \land faz(X, Y))$

   "Todas as pessoas cometem erros"

3. $\forall X \exists Y : (erro(Y) \land \sim faz(X, Y) \rightarrow \sim humano(X))$

   "Não é humano quem não erra"

Inferência em LPO

- O processo para especificação de conhecimento inferível em LPO é dado através de **fatos** e **regras**
- *Fatos* representam enumerações dos elementos de um domínio. São informações que se conhece como verdadeiras. É através de um conjunto de fatos que é possível se realizar a operação de particularização. Exemplo: $gato(felix)$
- *Regras* são generalizações de domínios para situações específicas. Exemplo: $\forall X \; gato(X) \rightarrow agil(X)$
- **Exercício:** Prove que "Felix é ágil"

Inferência em LPO

- O processo para especificação de conhecimento inferível em LPO é dado através de **fatos** e **regras**
- *Fatos* representam enumerações dos elementos de um domínio. São informações que se conhece como verdadeiras. É através de um conjunto de fatos que é possível se realizar a operação de particularização. Exemplo: $gato(felix)$
- *Regras* são generalizações de domínios para situações específicas. Exemplo: $\forall X: \ gato(X) \rightarrow agil(X)$
- **Exercício:** Prove que "Felix é ágil"

$$agil(felix)$$

Inferência em LPO

$$
\begin{array}{ll}
(1) & gato(felix) \\
(2) & \forall X:\ gato(X) \rightarrow agil(X) \quad \vdash agil(felix) \\
\hline
\end{array}
$$

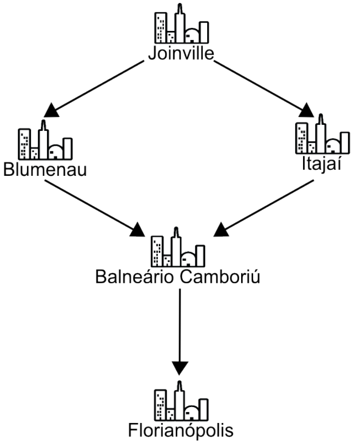
Inferência em LPO

$$
\begin{array}{lll}
(1) & gato(felix) & \\
(2) & \forall X:\ gato(X) \rightarrow agil(X) & \vdash agil(felix) \\
\hline
(3) & gato(felix) \rightarrow agil(felix) & \mathbf{(2)\ por\ (PU)\ \ X/felix} \\
(4) & agil(felix) & \mathbf{(1,3)\ por\ (MP)}
\end{array}
$$

Inferência em LPO

- Considere uma lista de cidades: **Joinville, Itajaí, Blumenau, Balneário Camboriú, Florianópolis**
- Considere agora que essas cidades são conectadas por estradas, conforme a figura a seguir
- Determine se existe algum **caminho** entre duas cidades.
- Há um caminho entre duas cidades se:
  1. as duas cidades são conectadas por uma estrada
  2. existe uma cidade intermediária (escala) que é conectada à cidade de origem a partir da qual há um caminho para a cidade destino
- Demonstre que há um caminho que ligue a cidade de **Joinville** à cidade de **Florianópolis**

Inferência em LPO



Inferência em LPO - Passo #1 = Definir o domínio

$$
\begin{array}{ll}
(1) & estrada(joinville, itajai) \\
(2) & estrada(joinville, blumenau) \\
(3) & estrada(itajai, balneariocamboriu) \\
(4) & estrada(blumenau, balneariocamboriu) \\
(5) & estrada(balneariocamboriu, florianopolis) \\
\hline
\end{array}
$$

Inferência em LPO - Passo #2 = Definir as regras

(1)	$estrada(joinville, itajai)$
(2)	$estrada(joinville, blumenau)$
(3)	$estrada(itajai, balneariocamboriu)$
(4)	$estrada(blumenau, balneariocamboriu)$
(5)	$estrada(balneariocamboriu, florianopolis)$
(6)	$\forall Origem\ \exists Destino : estrada(Origem, Destino) \rightarrow caminho(Origem, Destino)$
(7)	$\forall Origem\ \exists Destino : estrada(Origem, Escala) \wedge caminho(Escala, Destino) \rightarrow caminho(Origem, Destino)$

Inferência em LPO - Passo #3 = Definição da Hipótese

$$
\begin{array}{ll}
(1) & estrada(joinville, itajai) \\
(2) & estrada(joinville, blumenau) \\
(3) & estrada(itajai, balneariocamboriu) \\
(4) & estrada(blumenau, balneariocamboriu) \\
(5) & estrada(balneariocamboriu, florianopolis) \\
(6) & \forall Origem\ \exists Destino\ :\ estrada(Origem, Destino) \rightarrow caminho(Origem, Destino) \\
(7) & \forall Origem\ \exists Destino\ :\ estrada(Origem, Escala) \wedge caminho(Escala, Destino) \rightarrow caminho(Origem, Destino) \\
\hline
\end{array}
$$

$$\vdash caminho(joinville, florianopolis)$$

Inferência em LPO - Passo #4 = Inferência

(1)	$estrada(joinville, itajai)$		
(2)	$estrada(joinville, blumenau)$		
(3)	$estrada(itajai, balneariocamboriu)$		
(4)	$estrada(blumenau, balneariocamboriu)$		
(5)	$estrada(balneariocamboriu, florianopolis)$		
(6)	$\forall Origem\ \exists Destino : estrada(Origem, Destino) \rightarrow caminho(Origem, Destino)$		
(7)	$\forall Origem\ \exists Destino : estrada(Origem, Escala) \wedge caminho(Escala, Destino) \rightarrow caminho(Origem, Destino)$		
(8)	$estrada(joinville, itajai) \wedge caminho(itajai, florianopolis) \rightarrow caminho(joinville, florianopolis)$	**(7) por (PU)**	$Origem / joinville$ $Escala / itajai$ $Destino / florianopolis$

$$\vdash caminho(joinville, florianopolis)$$

Inferência em LPO - Passo #4 = Inferência

$$
\begin{array}{ll}
(1) & estrada(joinville, itajai) \\
(2) & estrada(joinville, blumenau) \\
(3) & estrada(itajai, balneariocamboriu) \\
(4) & estrada(blumenau, balneariocamboriu) \\
(5) & estrada(balneariocamboriu, florianopolis) \\
(6) & \forall Origem\ \exists Destino : estrada(Origem, Destino) \rightarrow caminho(Origem, Destino) \\
(7) & \forall Origem\ \exists Destino : estrada(Origem, Escala) \wedge caminho(Escala, Destino) \\
 & \rightarrow caminho(Origem, Destino) \\
\hline
(8) & estrada(joinville, itajai) \wedge caminho(itajai, florianopolis) \rightarrow caminho(joinville, florianopolis) \quad \mathbf{(7)\ por\ (PU)} \quad \begin{array}{l} Origem / joinville \\ Escala / itajai \\ Destino / florianopolis \end{array} \\
(9) & estrada(itajai, balneariocamboriu) \wedge caminho(balneariocamboriu, florianopolis) \\
 & \rightarrow caminho(itajai, florianopolis) \quad \mathbf{(7)\ por\ (PU)} \quad \begin{array}{l} Origem / itajai \\ Escala / balneariocamboriu \\ Destino / florianopolis \end{array}
\end{array}
$$

$$\vdash caminho(joinville, florianopolis)$$

Inferência em LPO - Passo #4 = Inferência

(1)	$estrada(joinville, itajai)$		
(2)	$estrada(joinville, blumenau)$		
(3)	$estrada(itajai, balneariocamboriu)$		
(4)	$estrada(blumenau, balneariocamboriu)$		
(5)	$estrada(balneariocamboriu, florianopolis)$		
(6)	$\forall Origem\ \exists Destino : estrada(Origem, Destino) \rightarrow caminho(Origem, Destino)$		
(7)	$\forall Origem\ \exists Destino : estrada(Origem, Escala) \wedge caminho(Escala, Destino) \rightarrow caminho(Origem, Destino)$		
(8)	$estrada(joinville, itajai) \wedge caminho(itajai, florianopolis) \rightarrow caminho(joinville, florianopolis)$	**(7) por (PU)**	$Origem / joinville$ $Escala / itajai$ $Destino / florianopolis$
(9)	$estrada(itajai, balneariocamboriu) \wedge caminho(balneariocamboriu, florianopolis) \rightarrow caminho(itajai, florianopolis)$	**(7) por (PU)**	$Origem / itajai$ $Escala / balneariocamboriu$ $Destino / florianopolis$
(10)	$estrada(balneariocamboriu, florianopolis) \rightarrow caminho(balneariocamboriu, florianopolis)$	**(6) por (PU)**	$Origem / balneariocamboriu$ $Destino / florianopolis$

$$\vdash caminho(joinville, florianopolis)$$

Inferência em LPO - Passo #4 = Inferência

(1)	$estrada(joinville, itajai)$		
(2)	$estrada(joinville, blumenau)$		
(3)	$estrada(itajai, balneariocamboriu)$		
(4)	$estrada(blumenau, balneariocamboriu)$		
(5)	$estrada(balneariocamboriu, florianopolis)$		
(6)	$\forall Origem\ \exists Destino : estrada(Origem, Destino) \rightarrow caminho(Origem, Destino)$		
(7)	$\forall Origem\ \exists Destino : estrada(Origem, Escala) \wedge caminho(Escala, Destino)$ $\rightarrow caminho(Origem, Destino)$		
(8)	$estrada(joinville, itajai) \wedge caminho(itajai, florianopolis) \rightarrow caminho(joinville, florianopolis)$	$(\mathbf{7})$ **por** $(\mathbf{PU})$	$Origem / joinville$ $Escala / itajai$ $Destino / florianopolis$
(9)	$estrada(itajai, balneariocamboriu) \wedge caminho(balneariocamboriu, florianopolis)$ $\rightarrow caminho(itajai, florianopolis)$	$(\mathbf{7})$ **por** $(\mathbf{PU})$	$Origem / itajai$ $Escala / balneariocamboriu$ $Destino / florianopolis$
(10)	$estrada(balneariocamboriu, florianopolis) \rightarrow caminho(balneariocamboriu, florianopolis)$	$(\mathbf{6})$ **por** $(\mathbf{PU})$	$Origem / balneariocamboriu$ $Destino / florianopolis$
(11)	$caminho(balneariocamboriu, florianopolis)$	$(\mathbf{5}, \mathbf{10})$ **por** $(\mathbf{MP})$	

$$\vdash caminho(joinville, florianopolis)$$

Inferência em LPO - Passo #4 = Inferência

(1)	$estrada(joinville, itajai)$		
(2)	$estrada(joinville, blumenau)$		
(3)	$estrada(itajai, balneariocamboriu)$		
(4)	$estrada(blumenau, balneariocamboriu)$		
(5)	$estrada(balneariocamboriu, florianopolis)$		
(6)	$\forall Origem\ \exists Destino : estrada(Origem, Destino) \rightarrow caminho(Origem, Destino)$		
(7)	$\forall Origem\ \exists Destino : estrada(Origem, Escala) \wedge caminho(Escala, Destino)$ $\rightarrow caminho(Origem, Destino)$		
(8)	$estrada(joinville, itajai) \wedge caminho(itajai, florianopolis) \rightarrow caminho(joinville, florianopolis)$	**(7) por (PU)**	$Origem / joinville$ $Escala / itajai$ $Destino / florianopolis$
(9)	$estrada(itajai, balneariocamboriu) \wedge caminho(balneariocamboriu, florianopolis)$ $\rightarrow caminho(itajai, florianopolis)$	**(7) por (PU)**	$Origem / itajai$ $Escala / balneariocamboriu$ $Destino / florianopolis$
(10)	$estrada(balneariocamboriu, florianopolis) \rightarrow caminho(balneariocamboriu, florianopolis)$	**(6) por (PU)**	$Origem / balneariocamboriu$ $Destino / florianopolis$
(11)	$caminho(balneariocamboriu, florianopolis)$	**(5, 10) por (MP)**	
(12)	$estrada(itajai, balneariocamboriu) \wedge caminho(balneariocamboriu, florianopolis)$	**(3, 11) por (CONJ)**	

$$\vdash caminho(joinville, florianopolis)$$

Inferência em LPO - Passo #4 = Inferência

(1)	$estrada(joinville, itajai)$		
(2)	$estrada(joinville, blumenau)$		
(3)	$estrada(itajai, balneariocamboriu)$		
(4)	$estrada(blumenau, balneariocamboriu)$		
(5)	$estrada(balneariocamboriu, florianopolis)$		
(6)	$\forall Origem\ \exists Destino : estrada(Origem, Destino) \rightarrow caminho(Origem, Destino)$		
(7)	$\forall Origem\ \exists Destino : estrada(Origem, Escala) \wedge caminho(Escala, Destino) \rightarrow caminho(Origem, Destino)$		
(8)	$estrada(joinville, itajai) \wedge caminho(itajai, florianopolis) \rightarrow caminho(joinville, florianopolis)$	**(7) por (PU)**	$Origem / joinville$ $Escala / itajai$ $Destino / florianopolis$
(9)	$estrada(itajai, balneariocamboriu) \wedge caminho(balneariocamboriu, florianopolis) \rightarrow caminho(itajai, florianopolis)$	**(7) por (PU)**	$Origem / itajai$ $Escala / balneariocamboriu$ $Destino / florianopolis$
(10)	$estrada(balneariocamboriu, florianopolis) \rightarrow caminho(balneariocamboriu, florianopolis)$	**(6) por (PU)**	$Origem / balneariocamboriu$ $Destino / florianopolis$
(11)	$caminho(balneariocamboriu, florianopolis)$	**(5, 10) por (MP)**	
(12)	$estrada(itajai, balneariocamboriu) \wedge caminho(balneariocamboriu, florianopolis)$	**(3, 11) por (CONJ)**	
(13)	$caminho(itajai, florianopolis)$	**(9, 12) por (MP)**	

$$\vdash caminho(joinville, florianopolis)$$

Inferência em LPO - Passo #4 = Inferência

(1)	$estrada(joinville, itajai)$		
(2)	$estrada(joinville, blumenau)$		
(3)	$estrada(itajai, balneariocamboriu)$		
(4)	$estrada(blumenau, balneariocamboriu)$		
(5)	$estrada(balneariocamboriu, florianopolis)$		
(6)	$\forall Origem\ \exists Destino : estrada(Origem, Destino) \rightarrow caminho(Origem, Destino)$		
(7)	$\forall Origem\ \exists Destino : estrada(Origem, Escala) \wedge caminho(Escala, Destino)$ $\rightarrow caminho(Origem, Destino)$		
(8)	$estrada(joinville, itajai) \wedge caminho(itajai, florianopolis) \rightarrow caminho(joinville, florianopolis)$	$\mathbf{(7)}$ **por** $\mathbf{(PU)}$	$Origem\,/\,joinville$ $Escala\,/\,itajai$ $Destino\,/\,florianopolis$
(9)	$estrada(itajai, balneariocamboriu) \wedge caminho(balneariocamboriu, florianopolis)$ $\rightarrow caminho(itajai, florianopolis)$	$\mathbf{(7)}$ **por** $\mathbf{(PU)}$	$Origem\,/\,itajai$ $Escala\,/\,balneariocamboriu$ $Destino\,/\,florianopolis$
(10)	$estrada(balneariocamboriu, florianopolis) \rightarrow caminho(balneariocamboriu, florianopolis)$	$\mathbf{(6)}$ **por** $\mathbf{(PU)}$	$Origem\,/\,balneariocamboriu$ $Destino\,/\,florianopolis$
(11)	$caminho(balneariocamboriu, florianopolis)$	$\mathbf{(5, 10)}$ **por** $\mathbf{(MP)}$	
(12)	$estrada(itajai, balneariocamboriu) \wedge caminho(balneariocamboriu, florianopolis)$	$\mathbf{(3, 11)}$ **por** $\mathbf{(CONJ)}$	
(13)	$caminho(itajai, florianopolis)$	$\mathbf{(9, 12)}$ **por** $\mathbf{(MP)}$	
(14)	$estrada(joinville, itajai) \wedge caminho(itajai, florianopolis)$	$\mathbf{(1, 13)}$ **por** $\mathbf{(CONJ)}$	

$$\vdash caminho(joinville, florianopolis)$$

Inferência em LPO - Passo #4 = Inferência

(1)	$estrada(joinville, itajai)$		
(2)	$estrada(joinville, blumenau)$		
(3)	$estrada(itajai, balneariocamboriu)$		
(4)	$estrada(blumenau, balneariocamboriu)$		
(5)	$estrada(balneariocamboriu, florianopolis)$		
(6)	$\forall Origem\ \exists Destino : estrada(Origem, Destino) \rightarrow caminho(Origem, Destino)$		
(7)	$\forall Origem\ \exists Destino : estrada(Origem, Escala) \wedge caminho(Escala, Destino)$ $\rightarrow caminho(Origem, Destino)$		
(8)	$estrada(joinville, itajai) \wedge caminho(itajai, florianopolis) \rightarrow caminho(joinville, florianopolis)$	$\mathbf{(7)}$ **por** $\mathbf{(PU)}$	$Origem / joinville$ $Escala / itajai$ $Destino / florianopolis$
(9)	$estrada(itajai, balneariocamboriu) \wedge caminho(balneariocamboriu, florianopolis)$ $\rightarrow caminho(itajai, florianopolis)$	$\mathbf{(7)}$ **por** $\mathbf{(PU)}$	$Origem / itajai$ $Escala / balneariocamboriu$ $Destino / florianopolis$
(10)	$estrada(balneariocamboriu, florianopolis) \rightarrow caminho(balneariocamboriu, florianopolis)$	$\mathbf{(6)}$ **por** $\mathbf{(PU)}$	$Origem / balneariocamboriu$ $Destino / florianopolis$
(11)	$caminho(balneariocamboriu, florianopolis)$	$\mathbf{(5, 10)}$ **por** $\mathbf{(MP)}$	
(12)	$estrada(itajai, balneariocamboriu) \wedge caminho(balneariocamboriu, florianopolis)$	$\mathbf{(3, 11)}$ **por** $\mathbf{(CONJ)}$	
(13)	$caminho(itajai, florianopolis)$	$\mathbf{(9, 12)}$ **por** $\mathbf{(MP)}$	
(14)	$estrada(joinville, itajai) \wedge caminho(itajai, florianopolis)$	$\mathbf{(1, 13)}$ **por** $\mathbf{(CONJ)}$	
(15)	$caminho(joinville, florianopolis)$	$\mathbf{(8, 14)}$ **por** $\mathbf{(MP)}$	

$$\vdash caminho(joinville, florianopolis)$$

Inferência em LPO - Exercício

Traduzir os conhecimentos abaixo para LPO (fatos ou regras):

1. Marcos era um homem
2. Todos os homens e mulheres são pessoas
3. Marcos nasceu em Pompeia
4. Todos os que nasceram em Pompeia eram romanos
5. Cesar era um soberano
6. Todos os romanos eram leais a Cesar ou o odiavam
7. As pessoas só tentam assassinar soberanos aos quais não são leais
8. Marcos tentou assassinar Cesar

Prove que: Marcos odeia Cesar

Inferência em LPO - Exercício

(1)	$homem(marcos)$
(2)	$\forall X : homem(X) \vee mulher(X) \rightarrow pessoa(X)$
(3)	$pompeano(marcos)$
(4)	$\forall X : pompeano(X) \rightarrow romano(X)$
(5)	$soberano(cesar)$
(6)	$\forall X : romano(X) \rightarrow leal\_a(X, cesar) \vee odeia(X, cesar)$
(7)	$\forall X\ \exists Y : pessoa(X) \wedge soberano(Y) \wedge tenta\_assassinar(X, Y) \rightarrow \sim leal\_a(X, Y)$
(8)	$tenta\_assassinar(marcos, cesar)$

$$\vdash odeia(marcos, cesar)$$

Inferência em LPO - Exercício

(1)	$homem(marcos)$	
(2)	$\forall X: homem(X) \vee mulher(X) \rightarrow pessoa(X)$	
(3)	$pompeano(marcos)$	
(4)	$\forall X: pompeano(X) \rightarrow romano(X)$	
(5)	$soberano(cesar)$	
(6)	$\forall X: romano(X) \rightarrow leal\_a(X, cesar) \vee odeia(X, cesar)$	
(7)	$\forall X\, \exists Y: pessoa(X) \wedge soberano(Y) \wedge tenta\_assassinar(X, Y) \rightarrow \sim leal\_a(X, Y)$	
(8)	$tenta\_assassinar(marcos, cesar)$	
(9)	$pessoa(marcos) \wedge soberano(cesar) \wedge tenta\_assassinar(marcos, cesar) \rightarrow \sim leal\_a(marcos, cesar)$	**(7) por (PU)** $X/marcos$, $Y/cesar$
(10)	$homem(marcos) \vee mulher(marcos) \rightarrow pessoa(marcos)$	**(2) por (PU) X/marcos**
(11)	$homem(marcos) \vee mulher(marcos)$	**(1) por (AD)**
(12)	$pessoa(marcos)$	**(10, 11) por (MP)**
(13)	$pessoa(marcos) \wedge soberano(cesar)$	**(5, 12) por (CONJ)**
(14)	$pessoa(marcos) \wedge soberano(cesar) \wedge tenta\_assassinar(marcos, cesar)$	**(8, 13) por (CONJ)**
(15)	$\sim leal\_a(marcos, cesar)$	**(9, 14) por (MP)**
(16)	$pompeano(marcos) \rightarrow romano(marcos)$	**(4) por (PU) X/marcos**
(17)	$romano(marcos)$	**(3, 16) por (MP)**
(18)	$romano(marcos) \rightarrow leal\_a(marcos, cesar) \vee odeia(marcos, cesar)$	**(6) por (PU) X/marcos**
(19)	$leal\_a(marcos, cesar) \vee odeia(marcos, cesar)$	**(17, 18) por (MP)**
(20)	$odeia(marcos, cesar)$	**(15, 19) por (SD)**

$$\vdash odeia(marcos, cesar)$$

Introdução à Programação Lógica

Programação Lógica

- Programação Lógica é um paradigma de programação baseado em linguagens **declarativas**
- Um programa declarativo rompe com a noção de sequencialidade de instruções lógicas; ao invés disso, trabalha com os conceitos de conhecimento declarativo e procedimental
- Conhecimento **declarativo** é aquele conhecimento que é especificado cujo uso não foi definido
- Conhecimento **procedimental** são informações de controle sobre o uso do conhecimento declarativo

Programação Lógica

- O processo de se programar através desse paradigma é o que se segue:

1. Modelagem do(s) domínio(s)
2. Modelagem dos fatos conhecidos acerca do problema
3. Modelagem das regras conhecidas acerca do problema
4. Consulta à base de conhecimento
5. Especificação da inferência desejada (predicado)

Em caso de sucesso, o motor de inferência retorna o predicado com suas variáveis **unificadas**

Programação Lógica - Unificação

- É o processo do PROLOG reconhecer predicados como sendo <u>similares</u>
- A fim de garantir a similiridade entre predicados os seguintes critérios precisam ser satisfeitos:
  1. mesmo nome de predicado
  2. mesma quantidade de parâmetros (aridade)
  3. mesmos valores literais na mesma ordem especificada (caso existam)
- Exemplos:
  - $homem(joao) \Leftrightarrow homem(X) \equiv MATCH$
  - $predicado(X, Marcos) \Leftrightarrow predicado(Y) \equiv$ NOT MATCH
- No caso da unificação ocorrer com sucesso, o motor de inferência substitui as variáveis presentes pelos seus respectivos valores unificados. Exemplo: no primeiro exemplo acima **X/joao**, ou seja, a variável X é substituída pela constante literal *joao*

Programação Lógica - PROLOG

- A linguagem de programação lógica **PROLOG** é um exemplo de linguagem declarativa
- Algumas restrições de representação de predicados são impostas pela linguagem:
  1. Todas as expressões são sucedidas pelo símbolo ponto **(.)**
  2. Os quantificadores $\forall$ e $\exists$ não são implementados explicitamente. O que significa que precisam ser representados de forma implícita (como conjunções ou disjunções)
  3. Os operadores lógicos possuem simbologia específica. A saber:

$$\wedge \equiv , \quad \vee \equiv ; \quad \rightarrow \equiv :-$$

  4. As regras de implicação lógica podem conter apenas um predicado no seu consequente:

$$homem(X) \wedge humano(X) \rightarrow mortal(X)$$

e ainda, são especificados em sua forma recíproca:
consequente $\rightarrow$ **antecedente**:

$$mortal(X) : -homem(X), humano(X)$$

Exemplo de Programação Lógica (I)

- *"Hoje fui a uma festa e apresentado a três casais. Os maridos tinham esposas e profissões distintas. Após alguns goles, me confundi quem era casado com quem e suas profissões. Apenas lembro de alguns fatos. Então me ajude a descobrir quem são os casais. Eis os fatos que eu me lembro:"*[1]

  1. O médico é casado com a Maria
  2. Paulo é advogado
  3. Patrícia não é casada com Paulo
  4. Carlos não é médico
  5. Ah! Luis e Lúcia também estavam na festa
  6. Lembro também que alguém era engenheiro

[1] Retirado da Revista Coquetel - Problemas de Lógica

Exemplo de Programação Lógica (II)

Modelagem do Problema:

- Podemos identificar três domínios:

  Homens (H) Pedro, Carlos, Luis
  Mulheres (M) Patrícia, Lúcia, Maria
  Profissões (P) Médico, Engenheiro, Advogado

- Queremos identificar tuplas-3 da forma (H, M, P)
- A solução do problema é então uma lista com 3 dessas tuplas:

$$(H1, M1, P1), (H2, M2, P2), (H3, M3, P3)$$

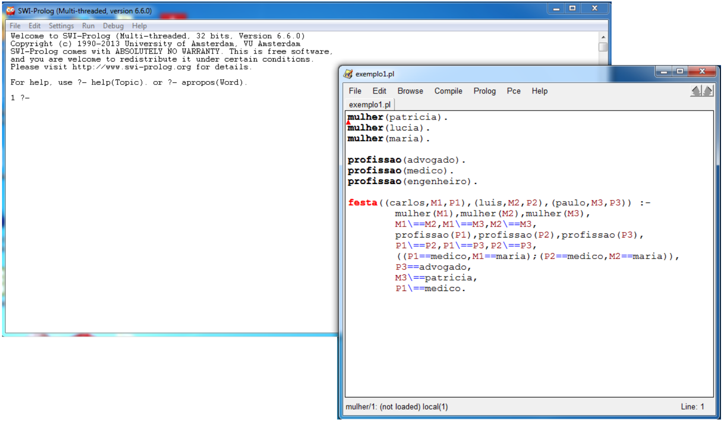
Exemplo de Programação Lógica (III)

- O processo para desenvolvimento e execução de um programa em PROLOG é o seguinte:

1. Editoração do código do programa em um ou mais arquivos texto (usualmente *.pl)
2. É necessário ter um motor de inferência PROLOG como **SWI-Prolog**[2] ou **tkEclipse**
3. Consulta ao arquivo fonte através do menu **File** | **Consult**
4. Em caso de sucesso na consulta, basta digitar no prompt do ambiente (**?-**) a inferência desejada (não esqueça do "." !)

[2] `http://www.swi-prolog.org/`

Exemplo de Programação Lógica (IV)



Exemplo de Programação Lógica (V)

exemplo1.pl

```
profissao (medico).
profissao (engenheiro ).
profissao (advogado).

mulher(maria).
mulher( lucia ).
mulher( patricia ).

festa (( carlos , M1, P1), ( luis , M2, P2), (paulo, M3, P3)) :-
        mulher(M1), mulher(M2), mulher(M3),
        M1\==M2,M1\==M3,M2\==M3,
        profissao (P1), profissao (P2), profissao (P3),
        P1\==P2,P1\==P3,P2\==P3,
        ((P1 == medico,M1 == maria);(P2 == medico, M2 == maria)),
        P3 == advogado,
        M3 \== patricia,
        P1 \== medico.
```

Inferência PROLOG com Sentenças Abertas

- O que nós acabamos de fazer foi uma inferência (em uma base de conhecimento) a partir de uma sentença aberta
- Ou seja, na prática o que o PROLOG foi construir uma enumeração de todas as possíveis combinações de particularização das variáveis envolvidas (produto cartesiano) e retornou aquela particularização que satisfez ao predicado inferido
- Notem que, dependendo do problema podem haver mais de uma particularização que satisfaz o problema. O Prolog irá apenas mostrar inicialmente a primeira destas. Caso se deseje ver outras possíveis respostas, use a tecla "**;**" (no tkEclipse há um botão na interface para isso)
- "**.**" encerra o processo

Inferência PROLOG com Sentenças Abertas

- Mas como funciona o processo de determinação da particularização no Prolog ?
- Através de uma abordagem recursiva denominada **backtracking** (retrocesso)
- Inicialmente, o sistema particulariza cada variável do predicado sendo inferido com a primeira opção disponível na base de conhecimento
- Em seguida, para a <u>última</u> variável inferida, uma próxima particularização é determinada
- Quando não houverem mais particularizações possíveis para a última variável, então o processo é realizado para a penúltima variável de forma análoga (e assim por diante, para as demais variáveis)

Inferência PROLOG com Sentenças Abertas

prodcartesiano.pl

```
a(1).
a(2).
a(3).
b( alfa ).
b(beta).
b(gama).
produto(X,Y) :- a(X),b(Y).
```

Inferência PROLOG com Sentenças Abertas

- Ao inferirmos a sentença "**produto(A,B).**" obtemos como resultado todas as particularizações para as variáveis A e B que assumem os valores produzidos para as variáveis X e Y respectivamente: **X/A Y/B**
- O resultado obtido é:

$$
\begin{array}{ll}
A = 1 & B = alfa \\
A = 1 & B = beta \\
A = 1 & B = gama \\
A = 2 & B = alfa \\
A = 2 & B = beta \\
A = 2 & B = gama \\
A = 3 & B = alfa \\
A = 3 & B = beta \\
A = 3 & B = gama
\end{array}
$$

Inferência PROLOG com Sentenças Abertas

- Sempre que uma variável é particularizada por algum predicado, ela permanecerá particularizada pelo restante da interpretação (Φ) daquele predicado (variáveis não mudam de valor durante uma interpretação)
- Neste caso, predicados que já tenham suas variáveis particularizadas se tornam 'fatos' e são interpretadas pelo Prolog como proposições (V ou F)

Inferência PROLOG com Sentenças Abertas

prodcartesiano2.pl

```
a(1).
a(2).
a(3).
b(1).
b(3).
b(5).
c(X,Y) :- a(Y), b(X).
produto(X,Y) :- a(X),b(Y),c(X,Y).
```

Qual a sequência de particularizações para o predicado “produto(A,B).” ?

Prolog - Exercício (I)

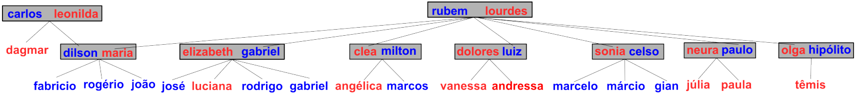
1. Traduzir o exercício dos caminhos e estradas visto em aula, para Prolog
2. Repetir a inferência "**caminho(joinville, florianopolis).**" para confirmar se a programação está correta
3. Inferir outros caminhos (válidos e inválidos)
4. Acrescentar novas cidades e estradas ao mapa original

Prolog - Exercício (II)

1. Traduzir o exercício do soberano visto em aula, para Prolog
2. Repetir a inferência "**odeia(marcos, cesar).**" para confirmar se a programação está correta

Prolog - Exercício (III)

1. Construir um programa em PL para descrever relações familiares em uma árvore genealógica
2. Considere como fatos: homem, mulher, pai, casal
3. Sugestão: use sua própria família como exemplo
4. Construa regras para descrever os seguintes predicados: irmão, irmã, avô, avó, tio, tia, primo, prima



Prolog - Exercício (III)

familia-pessoas.pl

```
homem(carlos).
homem(rubem).
homem(dilson).
homem(fabricio).
homem(rogerio).
homem(joao).
homem(gabriel).
homem(jose).
homem(rodrigo).
homem(gabrielzinho).
homem(milton).
homem(marcos).
homem(luiz).
homem(celso).
homem(marcelo).
homem(marcio).
homem(gian).
homem(paulo).
homem(hipolito).
```

```
mulher( leonilda ).
mulher( lourdes ).
mulher(maria ).
mulher(dagmar).
mulher( elizabeth ).
mulher( luciana ).
mulher( clea ).
mulher( angelica ).
mulher( dolores ).
mulher(vanessa ).
mulher(andressa ).
mulher(sonia ).
mulher(neura ).
mulher( julia ).
mulher(paula ).
mulher(olga ).
mulher(temis ).
```

Prolog - Exercício (III)

familia-pais.pl

```
pai( carlos ,  dilson ).
pai( carlos ,  dagmar).
pai( dilson ,  fabricio ).
pai( dilson ,  rogerio ).
pai( dilson ,  joao ).
pai(rubem, maria).
pai(rubem, gabriel ).
pai(rubem, clea ).
pai(rubem, luiz ).
pai(rubem, sonia ).
pai(rubem, paulo).
pai(rubem, olga ).
pai( gabriel ,  jose ).
pai( gabriel ,  luciana ).
pai( gabriel ,  rodrigo ).
pai( gabriel ,  gabrielzinho ).
pai(milton,  angelica ).
pai(milton,  marcos).
pai( luiz ,  vanessa ).
pai( luiz ,  andressa ).
pai( celso ,  marcelo ).
pai( celso ,  marcio ).
pai( celso ,  gian ).
pai(paulo,  julia ).
pai(paulo,  paula ).
pai( hipolito ,  temis ).
```

```
casal( carlos ,  leonilda ).
casal( dilson ,  maria ).
casal(rubem, lourdes ).
casal( gabriel ,  elizabeth ).
casal(milton,  clea ).
casal( luiz ,  dolores ).
casal( celso ,  sonia ).
casal(paulo,  neura ).
casal( hipolito ,  olga ).
```

Prolog - Exercício (III)

familia-relacoes.pl

```
mae(X,Y) :- mulher(X), pai(W,Y), casal(W,X).
avo(X,Y) :- homem(X), pai(W,Y), pai(X,W).
avo(X,Y) :- homem(X), mae(W,Y), pai(X,W).
avoh(X,Y) :- mulher(X), pai(W,Y), mae(X,W).
avoh(X,Y) :- mulher(X), mae(W,Y), mae(X,W).
irmao(X,Y) :- homem(X), pai(W,X), pai(W,Y), X\==Y.
irma(X,Y) :- mulher(X), pai(W,X), pai(W,Y), X\==Y.
irmaos(X,Y) :- irmao(X,Y) ; irma(X,Y).
tio (X,Y) :- homem(X), pai(W,Y), irmao(X,W) .
tio (X,Y) :- homem(X), mae(W,Y), irmao(X,W).
tio (X,Y) :- homem(X), pai(W,Y), irma(Z,W), casal(X,Z).
tio (X,Y) :- homem(X), mae(W,Y), irma(Z,W), casal(X,Z).
tia (X,Y) :- mulher(X), pai(W,Y), irma(X,W) .
tia (X,Y) :- mulher(X), mae(W,Y), irma(X,W).
tia (X,Y) :- mulher(X), pai(W,Y), irmao(Z,W), casal(Z,X).
tia (X,Y) :- mulher(X), mae(W,Y), irmao(Z,W), casal(Z,X).
primo(X,Y) :- homem(X), pai(W,X), pai(T,Y), irmao(W,T).
primo(X,Y) :- homem(X), pai(W,X), mae(T,Y), irmao(W,T).
primo(X,Y) :- homem(X), mae(W,X), pai(T,Y), irma(W,T).
primo(X,Y) :- homem(X), mae(W,X), mae(T,Y), irma(W,T).
prima(X,Y) :- mulher(X), pai(W,X), pai(T,Y), irmao(W,T).
prima(X,Y) :- mulher(X), pai(W,X), mae(T,Y), irmao(W,T).
prima(X,Y) :- mulher(X), mae(W,X), pai(T,Y), irma(W,T).
prima(X,Y) :- mulher(X), mae(W,X), mae(T,Y), irma(W,T).
```

Introdução à Lógica Nebulosa

Thumé, G. S. (2012).
Estudo e implementação de um modelo EBDI para agentes cognitivos aplicado a atores virtuais.
Trabalho de conclusão de curso (tcc), Universidade do Estado de Santa Catarina (UDESC), Joinville, SC, Brasil.

Introdução à Lógica Nebulosa

- Joinville é uma cidade grande, média ou pequena ?
- 100 km/h é uma velocidade muito lenta, lenta, média, rápida, muito rápida ?
- $30^\circ C$ é uma temperatura quente, morna ou fria ?
- Quem é mais inteligente: Albert Einstein, Isaac Newton ou Stephen Hawking ?
- Galinha frita é um prato intragável, nojento, saboroso, delicioso ?

Introdução à Lógica Nebulosa

- Existem domínios e predicados que não podem ser tratados de forma precisa e bem definida.
- Exemplos:
  - Vai chover amanhã ?
  - João é mais alto que Pedro
  - Este produto é muito caro
- Até agora, nós estudamos a representação *monotônica* da lógica, ou seja: dada uma representação qualquer, sua interpretação é um valor booleano

$$\Phi(P) = \textbf{V} \text{ ou } \textbf{F} = [0, 1]$$

e precisamos de uma forma de se representar aspectos imprecisos ou sem limites bem definidos (**Princípio da Incerteza**)

Introdução à Lógica Nebulosa

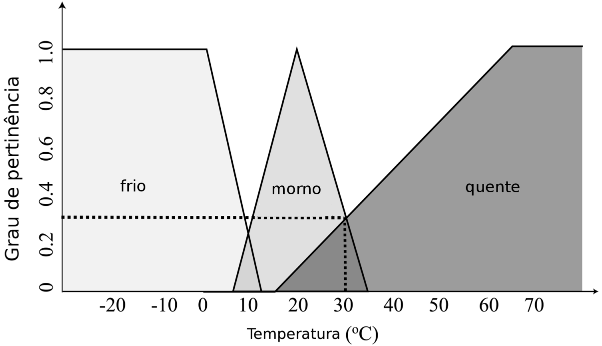
- Lógica Nebulosa (ou *Fuzzy*) é uma extensão à *lógica de primeira ordem* que considera a modelagem da incerteza
- Permite a categorização de atributos em conjuntos nebulosos (ou *Fuzzy Sets*). Neste tipo de representação, o valor de um atributo é representado em um intervalo fechado $[0 \ldots 1]$
- Essa teoria define que um dado atributo α apresenta grau de pertinência (δ) a certo conjunto nebuloso através de uma função dada por:

$$\delta = \mu(\alpha)$$

Por exemplo: como classificar a temperatura $30^\circ C$ (quente, morno ou frio) ?

Introdução à Lógica Nebulosa - Exemplo

- a temperatura $30^\circ C$ pode ser categorizada como $\delta = \{0/\mathit{frio}; 0.3/\mathit{morno}; 0.3/\mathit{quente}\}$ conforme demonstrado pela figura abaixo:



Operações em Lógica Nebulosa

Operações nebulosas nada mais são do que operações entre conjuntos:

Negação representa o conjunto de valores **não pertinentes** a uma dada classe fuzzy. Por exemplo: o conjunto de temperaturas que não são quentes.

$$\overline{\delta} = 1 - \mu(\alpha)$$

Conjunção dado um atributo α e seus graus de pertinência a duas classes $\mu_A(\alpha)$ e $\mu_B(\alpha)$, a conjunção representa o conjunto de valores que são **pertinentes** tanto à classe A quanto à classe B, ou seja:

$$\delta_{A \wedge B} = min(\mu_A(\alpha), \mu_B(\alpha))$$

Disjunção a disjunção representa o conjunto de valores que são pertinentes a pelo menos uma das classes A ou B:

$$\delta_{A \vee B} = max(\mu_A(\alpha), \mu_B(\alpha))$$

Inferência em Lógica Nebulosa

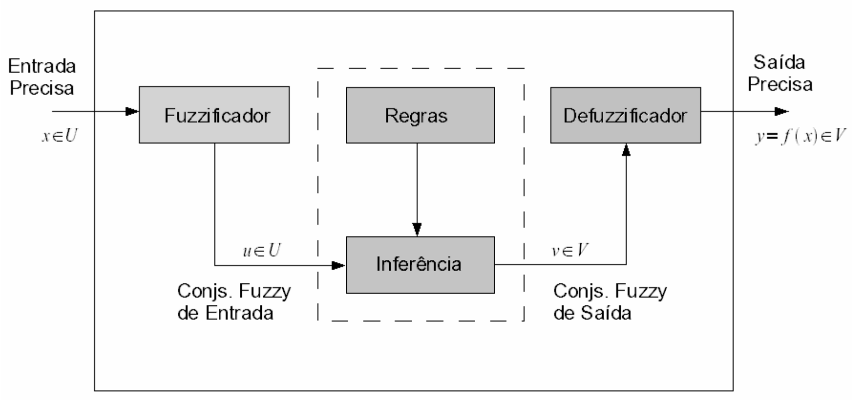
- Enquanto que o raciocínio (ou inferência) monotônico significava provar a validade de um conjunto de premissas $\alpha_1, \alpha_2, \ldots, \alpha_n \vdash \beta$, na lógica nebulosa, o que se quer validar é: dado o grau de pertinência das premissas, qual seria o grau de pertinência da conclusão:

$$\mu(\alpha_1), \mu(\alpha_2), \ldots, \mu(\alpha_n) \vdash \mu(\beta)$$

- O processo para se raciocinar a partir de lógica nebulosa segue quatro passos:

1. **Fuzzificação:** determina-se o grau de pertinência das variáveis de interesse $\delta_i = \mu_i(\alpha)$ para cada conjunto nebuloso
2. **Avaliação:** utilizando os graus de pertinência obtidos, avalia quais regras na base de conhecimento são ativadas (apresentam pertinência não nula).
3. **Composição:** com base nos resultados da avaliação das regras ativadas, deve-se compor estes resultados em um polígono resultante de saída.
4. **Defuzzificação:** traduzir o polígono resultante de saída no valor fuzzy final.

Introdução à Lógica Nebulosa



Introdução à Lógica Nebulosa - Fuzzificação

- Etapa responsável por converter as 'entradas precisas' (valores discretos) em sua categorização em conjuntos Fuzzy

- Este processo consiste em:
  1. Definição dos conjuntos Fuzzy e respectivos intervalos (polígonos)
  2. Determinação dos graus de pertinência do atributo de entrada

$$\delta_i = \mu_i(\alpha)$$

Introdução à Lógica Nebulosa - Fuzzificação

- Consideremos o seguinte cenário: temos um NPC o qual mapeia seus níveis de **saúde** e qualidade da **armadura** e queremos determinar o **risco** deste sofrer um ataque durante o jogo.
- Imagine uma situação onde temos os seguintes atributos de entrada a serem fuzzificados:

$$\text{saude } = 45\% \text{ e armadura } = 65\%$$

Introdução à Lógica Nebulosa - Fuzzificação

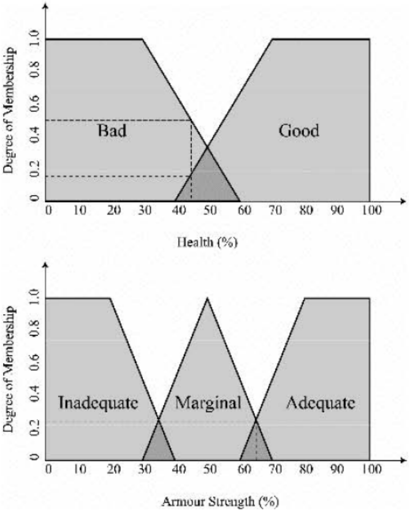
- Imagine os seguintes conjuntos Fuzzy de entrada:
- Para a saúde:

Ruim	{ (0, 1); (30, 1); (60, 0) }
Boa	{ (40, 0); (70, 1); (100, 1) }

- Para a armadura:

Inadequada	{ (0, 1); (20, 1); (40, 0) }
Marginal	{ (30, 0); (50, 1); (70, 0) }
Adequada	{ (60, 0); (80, 1); (100, 1) }

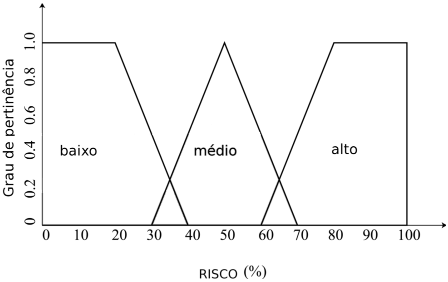
Introdução à Lógica Nebulosa - Fuzzificação



Introdução à Lógica Nebulosa - Fuzzificação

- Imagine ainda os seguintes conjuntos Fuzzy de saída:
- Para o risco:

Baixo { (0, 1); (20, 1); (40, 0) }
Médio { (30, 0); (50, 1); (70, 0) }
Alto { (60, 0); (80, 1); (100, 1) }



Introdução à Lógica Nebulosa - Fuzzificação

- O resultado do processo de fuzzificação neste caso seria:
  1. $\delta_{saude} = \{0.5/\text{Bad}; 0.17/\text{Good}\}$
  2. $\delta_{armadura} = \{0/\text{Inadequate}; 0.25/\text{Marginal}; 0.25/\text{Adequate}\}$

Introdução à Lógica Matemática - Avaliação

- A fase de avaliação utiliza uma base de conhecimento contendo as regras Fuzzy a serem executadas pelo *agente*
- Essas regras são escritas na forma

If **x** *is* **A** *then* **y** *is* **B**

onde x, y representam as variáveis fuzzy (entrada ou saída) e A, B representam os conjuntos Fuzzy; a operação denotada por *is* representa os graus de pertinência da variável no conjunto Fuzzy específico

Introdução à Lógica Matemática - Avaliação

```
if saude is Good AND armadura is Adequate then risco is baixo
if saude is Bad OR armadura is Marginal then risco is medio
if saude is Bad AND armadura is Inadequate then risco is alto
```

Introdução à Lógica Matemática - Avaliação

- O processo de avaliação consiste nos seguintes passos:

1. Substituição dos operadores ‘is’ pelos seus respectivos graus de pertinência
2. Determinação das operações lógicas existentes nas regras: AND, OR, NOT
3. As regras cujo grau de pertinência resultante sejam superiores a 0 (zero), diz-se que foram **ativadas**
4. O grau de pertinência resultante é então atribuído ao operador ‘is’ do atributo de saída

Introdução à Lógica Matemática - Avaliação

Passo #1: substituição dos graus de pertinência

```
1  if saude is Good (0.17) AND armadura is Adequate (0.25) then risco is baixo
2  if saude is Bad (0.5) OR armadura is Marginal (0.25) then risco is medio
3  if saude is Bad (0.5) AND armadura is Inadequate (0) then risco is alto
```

Passo #2: determinação das operações lógicas

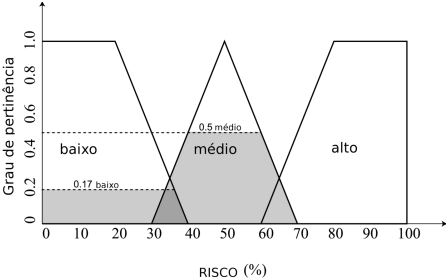
```
1  if saude is Good AND (0.17) armadura is Adequate then risco is baixo
2  if saude is Bad OR (0.5) armadura is Marginal then risco is medio
3  if saude is Bad AND (0) armadura is Inadequate then risco is alto
```

Passo #3: ativação das regras Fuzzy

```
1  if saude is Good AND armadura is Adequate then risco is baixo (0.17)
2  if saude is Bad OR armadura is Marginal then risco is medio (0.5)
```

Introdução à Lógica Matemática - Composição

- O resultado da fase de avaliação é um conjunto de graus de pertinência para o atributo de saída
- A etapa de composição tem por objetivo produzir a união desses graus de pertinência a fim de preparar o processo para a defuzzificação
- Esta etapa simplesmente "corta" os polígonos dos conjuntos Fuzzy nos pontos determinados pelos graus de pertinência obtidos, conforme ilustrado pela figura abaixo:



Introdução à Lógica Matemática - Defuzzificação

- Consiste em se transformar os conjuntos Fuzzy resultantes da composição em um valor de saída
- Apesar de existirem várias abordagens possíveis de serem aplicadas nessa etapa, uma das mais comuns é o cálculo do *centróide*
- Centróide significa calcular o **centro de gravidade** do polígono resultante (área hachurada) através da fórmula

$$G = \frac{\sum_{x=a}^{b} \mu(x)x}{\sum_{x=a}^{b} \mu(x)}$$

Introdução à Lógica Matemática - Defuzzificação

- Consiste em se transformar os conjuntos Fuzzy resultantes da composição em um valor de saída
- Apesar de existirem várias abordagens possíveis de serem aplicadas nessa etapa, uma das mais comuns é o cálculo do *centróide*
- Centróide significa calcular o **centro de gravidade** do polígono resultante (área hachurada) através da fórmula

$$G = \frac{\sum_{x=a}^{b} \mu(x) x}{\sum_{x=a}^{b} \mu(x)}$$

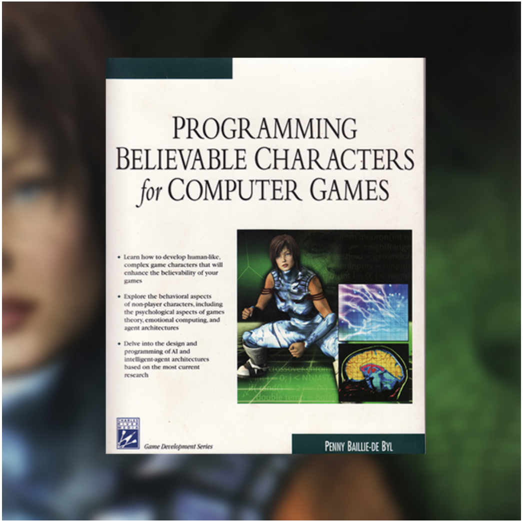
- Para o nosso exemplo, teríamos:

$$G = \frac{0(0.17) + 10(0.17) + 20(0.17) + 30(0.17) + 40(0.5) + 50(0.5) + 60(0.5) + 70(0.0) + 80(0.0) + 90(0.0) + 100(0.0)}{0.17 + 0.17 + 0.17 + 0.17 + 0.5 + 0.5 + 0.5 + 0 + 0 + 0 + 0} = 39.1$$

Introdução à Lógica Matemática - Defuzzificação

- **Conclusão:** para o caso onde o nível de saúde do NPC está em 45% e sua capacidade de armadura está em 65%, o NPC corre 39.1% de risco em caso de ataque
- Restaria agora utilizar ao agente utilizar regras de tomada de decisão autônoma (p.ex. via regras em lógica de primeira ordem) para decidir pela ação mais apropriada a executar neste caso.

Programando Agentes Autônomos



Gostei !!!!

- Computação Cognitiva Aplicada - CoCA

  `http://www2.joinville.udesc.br/~coca`

- D.R.A.M.A. - Developing Rational Agents to Mimic Actors

  `http://drama.musa.cc`