LET'S CODE

pass

Introdução à Lógica Proposicional

Objetivos

Introduzir lógica formal através de variáveis proposicionais (P, Q) e seus símbolos (\(\neg\),\ (\vee\),\(\wedge\),\(\rightarrow\),\(\equiv\)).

1. Introdução

A Lógica é um ramo do conhecimento que geralmente nos é dado como trivial, por ser utilizado tão frequentemente no nosso cotidiano. Computação, matemática, filosofia e até mesmo a nossa própria linguagem estão submissos às amarras do raciocínio lógico, como podemos perceber na seguinte passagem do livro "Alice Através do Espelho":

> "- Quando eu uso uma palavra – ele disse com um tom brincalhão. – Significa precisamente o que eu decido que significa: nem mais nem menos. – O problema é – Alice respondeu. – Se você consegue fazer com que as palavras signifiquem tantas coisas diferentes. – O problema é saber quem é que manda. Isso é tudo."

Ora, quem manda nas nossas palavras somos nós mesmos. Ao analisarmos nosso próprio discurso sob a ótica da lógica formal, veremos que tanto as palavras que escolhemos quanto o formato com o qual as apresentamos é determinante para transmitir a mensagem do modo como queremos. Não só isso, como também perceberemos que os discursos de terceiros podem não ter embasamento lógico algum quando analisados em detalhes.

Para conseguirmos explorar mais rigorosamente a lógica, começaremos com a lógica proposicional, que é a metodologia formal adotada tanto na matemática quanto na filosofia. Nesse contexto, introduziremos as chamadas variáveis proposicionais, as quais servem para traduzir uma ideia para a linguagem não ambígua da lógica formal, o que permite resolver problemas de maneira sistemática e com métodos bem definidos.

1.1 Proposições simples

Proposições simples são chamadas na lógica proposicional de constantes lógicas, podendo ser verdadeiras ou falsas individualmente.

Exemplos:
P: Fui para a praia.
Q: Passei protetor solar.

1.2 Proposições compostas

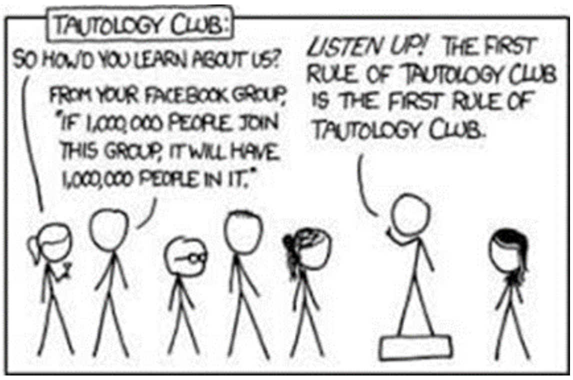
Dadas duas proposições - P e Q - e conectores entre elas, podemos avaliar a veracidade de uma afirmação lógica utilizando a veracidade de P e de Q.

Exemplos:
P e Q: Fui para a praia e passei protetor solar.
P ou Q: Fui para a praia ou passei protetor solar.

1.3 Tautologia

Uma proposição lógica que é sempre verdadeira.

Exemplo:



1.4 Contradição

Uma proposição lógica que é sempre falsa.

Exemplo: Os vírus são seres vivos e não são seres vivos.

2. Tabelas de verdade

2.1 Conector \(\neg\) (negação)

\(\neg P\) falsifica a proposição \(P\):

p	~p
T	F
F	T

or

p	~p
1	0
0	1

Exemplo:
P = Hoje fez sol.
\(\neg\)P = Hoje não fez sol.

2.2 Conector \(\wedge\) (conjunção)

P \(\wedge\) Q é verdadeira apenas se tanto P quanto Q são verdadeiras:

p	q	p∧q
T	T	T
T	F	F
F	T	F
F	F	F

Exemplo:
P = Douglas gosta de programar. Q = Jonas gosta de matemática.
P\(\wedge\)Q = Douglas gosta de programar e Jonas gosta de matemática.

2.3 Conector \(\vee\) (disjunção)

P \(\vee\) Q é verdadeira se P é verdadeira ou se Q é verdadeira:

p	q	p∨q
T	T	T
T	F	T
F	T	T
F	F	F

Exemplo:
P = O semáforo está verde. Q = O semáforo está amarelo.
P\(\vee\)Q = O semáforo está verde ou amarelo

2.4 Conector \(\rightarrow\) (implicação)

P \(\rightarrow\) Q é verdadeira se P é verdadeira e Q é verdadeira ou se P é falsa:

p	q	p→q
T	T	T
T	F	F
F	T	T
F	F	T

Exemplo:
P = Faz sol amanhã. Q = Vou ao parque.
P\(\rightarrow\)Q = Se faz sol amanhã, vou ao parque.

2.5 Conector \(\equiv\) (bicondicional)

P \(\equiv\) Q é verdadeira se P e Q são ambas falsas ou ambas verdadeiras:

p	q	$p \leftrightarrow q$
T	T	T
T	F	F
F	T	F
F	F	T

Exemplo:
P = Não faz trânsito. Q = Chego rápido em casa.
P\(\equiv\)Q = Se não faz trânsito, chego rápido em casa. Se chego rápido em casa, não faz trânsito.

2.6 Precedência dos operadores lógicos

Ordem de precedência: $\neg$, $\wedge$, $\vee$, $\Rightarrow$, $\Leftrightarrow$

Exemplos:

$\neg p \wedge q \equiv ((\neg p) \wedge q)$
$p \wedge \neg q \equiv (p \wedge (\neg q))$
$p \wedge q \vee r \equiv ((p \wedge q) \vee r)$
$p \vee q \wedge r \equiv (p \vee (q \wedge r))$
$p \Rightarrow q \Leftrightarrow r \equiv ((p \Rightarrow q) \Leftrightarrow r)$
$p \Leftrightarrow q \Rightarrow r \equiv (p \Leftrightarrow (q \Rightarrow r))$

3. Variações da implicação

3.1 Recíproca

A recíproca é obtida trocando a premissa com a conclusão: Recíproca de $P \rightarrow Q$: $Q \rightarrow P$

Exemplo:
Implicação: Se sei programar, então sei lógica.
Recíproca: Se sei lógica, então sei programar.

3.2 Contrapositiva

A contrapositiva é obtida trocando a premissa com a conclusão e negando ambas: Contrapositiva de \(P \rightarrow Q\): \(\neg Q \rightarrow \neg P\) São equivalentes.

Exemplo:
Implicação: Se sei programar, então sei lógica.
Contrapositiva: Se não sei lógica, então não sei programar.

3.3 Inversa

A inversa é obtida negando a premissa e a conclusão: Inversa de \(P \rightarrow Q\): \(\neg P \rightarrow \neg Q\) Equivalente à recíproca.

Exemplo:
Implicação: Se sei programar, então sei lógica.
Inversa: Se não sei programar, então não sei lógica.

3.4 Negação da implicação

A negação de uma implicação é obtida negando-a: Negação de \(P \rightarrow Q\): \(\neg (P \rightarrow Q)\) Equivalente a \(P \wedge \neg Q\).

Exemplo:
Implicação: Se sei programar, então sei lógica.
Negação da implicação: Sei programar e não sei lógica

Equivalências Lógicas

Objetivo
Apresentar equivalências lógicas úteis para resolução de problemas.

1. Propriedades

1.1 Comutativa

$p \land q \equiv q \land p$
$p \lor q \equiv q \lor p$

Exemplo: Patos são animais e facas são objetos é equivalente a facas são objetos e patos são animais.

1.2 Associativa

$p \land (q \land r) \equiv (p \land q) \land r$
$p \lor (q \lor r) \equiv (p \lor q) \lor r$

Exemplo: Patos são animais e facas são objetos pontudos é equivalente a patos são animais e facas são objetos e facas são pontudas.

1.3 Distributiva

$p \lor (q \land r) \equiv (p \lor q) \land (p \lor r)$
$p \land (q \lor r) \equiv (p \land q) \lor (p \land r)$

Exemplo: Patos são animais ou facas são objetos pontudos é equivalente a patos são animais ou facas são objetos e patos são animais ou facas são pontudas.

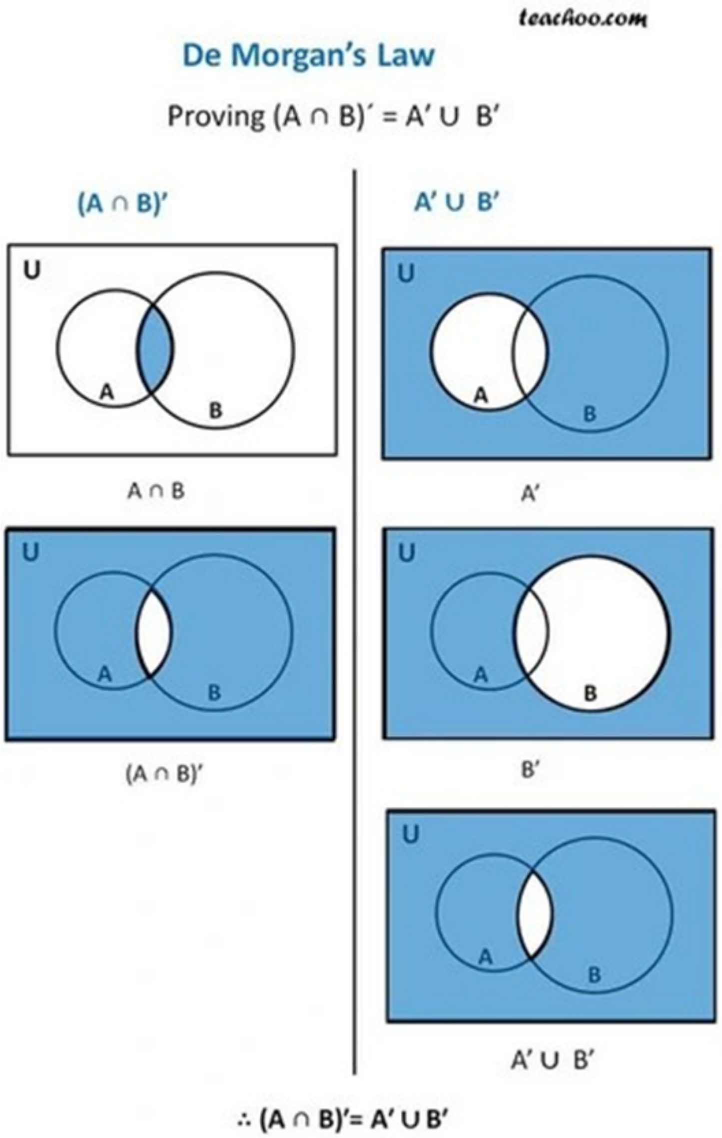
2. Regras de substituição

2.1 Dupla negativa
$\neg\neg p \equiv p$

Exemplo: O inimigo do meu inimigo é meu amigo.

2.2 Primeira lei de De Morgan

A negação da disjunção é a conjunção das negações.

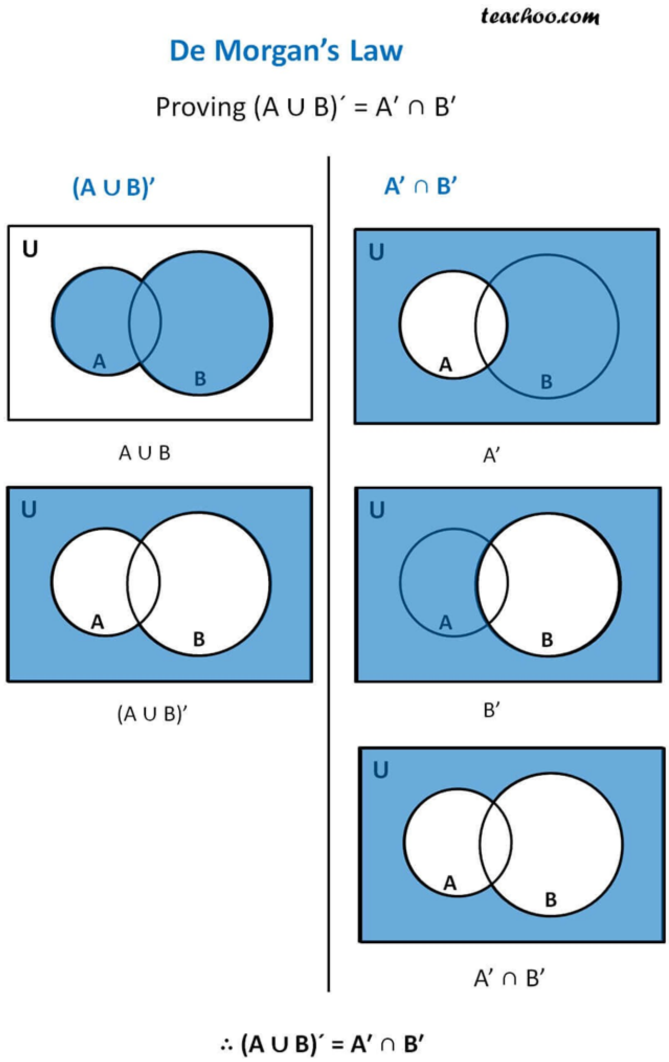


¬(p ∧ q) ≡ (¬p ∨ ¬q)
Forma computacional: “Not (P and Q)” é equivalente a “Not P or not Q.”

Exemplo: “Não está nublado e chovendo” é equivalente a “Não está nublado ou não está chovendo”

2.3 Segunda lei de De Morgan

A negação da conjunção é a disjunção das negações.



$\neg(p \lor q) \equiv (\neg p \land \neg q)$
"Not (P or Q)" é equivalente a "Not P and not Q."

Exemplo: "Não está nublado ou chovendo" é equivalente a "Não está nublado e não está chovendo"

2.4 Implicação-disjunção

$(p \rightarrow q) \equiv (\neg p \lor q)$

Exemplo: "Se sei programar, então sei lógica" $(p \rightarrow q)$ é equivalente a "Não sei programar ou sei lógica" $(\neg p \lor q)$.

3. Regras de inferência

3.1 Eliminação da conjunção

Se p ∧ q é verdade, então p é verdade e q é verdade.
$p \land q \rightarrow p$
$p \land q \rightarrow q$

Exemplo:
Gosto de pinguins e de golfinhos (p ∧ q).
Logo, gosto de pinguins (p).

3.2 Modus ponens (MP)

Sabendo que p → q e sabendo p, podemos afirmar que q deve ser verdade. $(p \rightarrow q) \land p \equiv q$

Exemplo:
Se alguém sabe programar, então a pessoa sabe lógica. (p → q)
Elisa sabe programar (p).
Logo, Elisa sabe lógica (q).

3.3 Modus tollens (MT)

Sabendo que p → q e sabendo ¬q, podemos afirmar que ¬p deve ser verdade. $(p \rightarrow q) \land \neg q \equiv \neg p$

Exemplo:
Se alguém sabe programar, então a pessoa sabe lógica. (p → q)
Rodolfo não sabe lógica (¬q).
Logo, Rodolfo não sabe programar (¬p).

3.4 Modus Tollendo Ponens ou Silogismo disjuntivo

Se p ∨ q e ¬p são verdade, então q é verdade.
$(p \lor q) \land \neg p \rightarrow q$

Exemplo:
Hoje vou ao shopping ou vou estudar lógica. (p ∨ q)
Não fui ao shopping (¬p).
Logo, estudei lógica (q).