

DCC638 - Introdução à Lógica Computacional
2023.1

Os Fundamentos: Lógica Proposicional

Área de Teoria DCC/UFMG

Introdução

Lógica: Introdução

- A **lógica** é o ramo da filosofia, matemática e ciência da computação que trata das **inferências válidas**.

A lógica é a base do raciocínio matemático e de todo o raciocínio automatizado.

- Ela estuda a **preservação da verdade** durante uma argumentação.

A lógica concerne técnicas que garantem que:

1. partindo de hipóteses verdadeiras,
 2. atinjam sempre conclusões também verdadeiras.
- As regras da lógica dão significado preciso a afirmações matemáticas. Elas são essenciais na construção de **demonstrações matemáticas**.

Lógica: Introdução

- A lógica é fundamental em aplicações em ciência da computação:
 - ① projeto de computadores e desenhos de circuitos,
 - ② especificação de sistemas,
 - ③ escrita de programas de computador,
 - ④ inteligência artificial,
 - ⑤ demonstração automática de teoremas,
 - ⑥ verificação de programas,
 - ⑦ processamento de linguagem natural,
 - ⑧ ...
- A lógica se divide em vários tipos: lógica proposicional, lógica de predicados, lógica de ordem superior, lógicas não-clássicas (como intuicionista e linear), etc.
- Neste curso vamos nos concentrar na **lógica proposicional** e na **lógica de predicados**.

Lógica Proposicional

Proposições

- Uma **proposição** é uma sentença declarativa (ou seja, uma sentença que estabelece um fato) que pode ser verdadeira ou falsa, mas não ambos.
- Exemplo 1

 As seguintes sentenças declarativas são proposições:
 - “*Belo Horizonte é a capital de Minas Gerais.*” (Proposição verdadeira)
 - “*Londres é a capital da França.*” (Proposição falsa)
 - “ $1 + 1 = 2.$ ” (Proposição verdadeira)
 - “ $2 + 2 = 3.$ ” (Proposição falsa)
- Exemplo 2

 As seguintes sentenças não são proposições:
 - “*Que horas são?*” (Não é uma sentença declarativa.)
 - “*Estude com afinco para a prova.*” (Não é uma sentença declarativa.)
 - “ $x + 2 = 3.$ ” (Não é verdadeira nem falsa, pois x é desconhecido.)

Proposições

- Nós usamos letras para denotar **variáveis proposicionais**, ou seja, variáveis que representam proposições:

$$p, q, r, s, t, \dots$$

- O **valor de verdade** de uma proposição pode ser:
 - **verdadeiro**, denotado por V (verdadeiro) ou T (do inglês *true*), ou
 - **falso**, denotado por F (falso ou, em inglês, *false*).
- A área da lógica que lida com proposições é chamada de **lógica proposicional** ou **cálculo proposicional**.

A lógica proposicional foi formalizada pela primeira vez pelo filósofo grego Aristóteles no Século IV AC.

Proposições compostas

- **Proposições atômicas** são aquelas que não podem ser expressas em termos de proposições mais simples.
- **Proposições compostas** podem ser criadas ao se combinarem proposições já existentes.

A combinação de proposições é feita usando **operadores lógicos** ou **conectivos lógicos** como:

- negação (não),
 - conjunção (e),
 - disjunção (ou),
 - implicação (implica),
 - implicação dupla (implica duplamente).
-
- Nós vamos agora estudar estes conectivos.

Conectivos lógicos: Negação

- Seja p uma proposição.

A **negação de p** , denotada por $\neg p$ (ou também \bar{p} , $\sim p$, $!p$), é a afirmação

“Não é o caso de que p .”

Lê-se a proposição $\neg p$ como “não p ”.

O valor de verdade de $\neg p$ é o oposto do valor de verdade de p .

- Tabela da verdade** para a negação $\neg q$ de uma proposição p :

Negação	
p	$\neg p$
T	F
F	T

Conectivos lógicos: Negação

- Exemplo 3 Seja a proposição

p : “O computador do Haniel roda Linux.”

A negação $\neg p$ é: “Não é o caso de que o computador do Haniel rode Linux.”

Forma alternativa da negação $\neg p$: “O computador do Haniel não roda Linux.”

- Exemplo 4 Seja a proposição

q : “Carolina tem pelo menos 25 anos.”

A negação $\neg q$ é: “Não é o caso de que Carolina tenha pelo menos 25 anos.”

Forma alternativa da negação $\neg q$: “Carolina não tem pelo menos 25 anos.”

Mais uma forma de negação $\neg q$: “Carolina tem menos de 25 anos.”

Conectivos lógicos: Conjunção

- Sejam p e q proposições.

A **conjunção de p e q** , denotada por $p \wedge q$, é a afirmação

“ p e q ”.

A conjunção $p \wedge q$ é verdadeira quando ambos p e q são verdadeiros, e é falsa em caso contrário.

- Tabela da verdade** para a conjunção $p \wedge q$ de duas proposições p e q :

Conjunção

p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

Conectivos lógicos: Conjunção

- Exemplo 5 Sejam as proposições:

p : “Hoje é sábado”,

q : “Vou fazer o jantar”.

A conjunção $p \wedge q$ é:

“Hoje é sábado e vou fazer o jantar.”

- Às vezes em linguagem natural usamos “mas” para significar conjunção:

- Exemplo 6 A proposição

“Hoje chove, mas vou sair”

é a conjunção $p \wedge q$ das proposições

p : “Hoje chove”

q : “Hoje vou sair”.

Conectivos lógicos: Disjunção

- Sejam p e q proposições.

A **disjunção de p e q** , denotada por $p \vee q$, é a afirmação

“ p ou q ”.

A disjunção $p \vee q$ é verdadeira quando ao menos um entre p e q é verdadeiro, e é falsa em caso contrário.

- Tabela da verdade para a disjunção $p \vee q$ de duas proposições p e q :

Disjunção

p	q	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F

Conectivos lógicos: Disjunção

- Exemplo 7 Sejam as proposições:

p : “O celular de Alice é azul”,

q : “O celular de Alice é novo”.

A disjunção $p \vee q$ é:

“O celular de Alice é azul ou o celular de Alice é novo.”

Alternativamente, $p \vee q$ é:

“O celular de Alice é azul ou é novo.”



Conectivos lógicos: “Ou inclusivo” versus “ou exclusivo”

- A palavra “ou” tem dois significados diferentes em linguagem natural.
- O conectivo “ou” da disjunção corresponde ao significado de **ou inclusivo**, em que a disjunção é verdadeira se ao menos uma das proposições é verdadeira.
- Exemplo 8

 A disjunção

“Você pode se matricular nesta disciplina se tiver cursado Cálculo ou Programação”

significa que podem se matricular na disciplina:

- alunos que cursaram apenas Cálculo,
- alunos que cursaram apenas Programação,
- alunos que cursaram ambos Cálculo e Programação.

Esta é uma disjunção inclusiva.



Conectivos lógicos: “Ou inclusivo” versus “ou exclusivo”

- O outro significado de “ou” corresponde ao **ou exclusivo**, em que a disjunção é verdadeira se exatamente uma das proposições é verdadeira.

- Exemplo 9 Se você ler na entrada de um conjunto de salas de cinema:

“O ingresso dá direito a assistir à sessão de Star Wars ou à sessão de O Senhor dos Anéis.”

você entende que você pode:

- escolher assistir à sessão de *Star Wars*, mas não à de *O Senhor dos Anéis*,
- escolher assistir à sessão de *O Senhor dos Anéis*, mas não à de *Star Wars*,
- mas você não pode assistir a ambas as sessões de *Star Wars* e de *O Senhor dos Anéis*.

Esta é uma disjunção exclusiva.



Conectivos lógicos: Ou exclusivo

- Sejam p e q proposições.

O **ou exclusivo** de p e q , denotado por $p \oplus q$, é a afirmação

“ou p ou q ”.

O ou exclusivo $p \oplus q$ é verdadeiro quando exatamente um entre p e q é verdadeiro, e é falso em caso contrário.

É comum ler $p \oplus q$ como “ p xor q ” (do inglês *exclusive or*).

- Tabela da verdade para o ou exclusivo $p \oplus q$ de duas proposições p e q :

Ou exclusivo

p	q	$p \oplus q$
T	T	F
T	F	T
F	T	T
F	F	F

Conectivos lógicos: Ou exclusivo

- Exemplo 10 Sejam as proposições

p : “Eu vou à festa hoje”,

q : “Eu vou ficar em casa hoje”.

O ou exclusivo $p \oplus q$ é:

“Hoje ou eu vou à festa, ou eu vou ficar em casa.”



Conectivos lógicos: Proposições condicionais

- Sejam p e q proposições.

A **afirmação condicional** ou **implicação** $p \rightarrow q$ é a afirmação

“se p , então q ”.

A afirmação condicional $p \rightarrow q$ é falsa quando p é verdadeira e q é falsa, e a afirmação é verdadeira em caso contrário.

- Na afirmação condicional $p \rightarrow q$:
 - p é chamada de **hipótese**, **antecedente**, ou **premissa**,
 - q é chamada de **conclusão** ou **consequente**.

Conectivos lógicos: Proposições condicionais

- Tabela da verdade para a proposição condicional $p \rightarrow q$ envolvendo duas proposições p e q :

Implicação

p	q	$p \rightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T

A afirmação condicional $p \rightarrow q$ é falsa quando p é verdadeira e q é falsa, e a afirmação é verdadeira caso contrário.

Conectivos lógicos: Proposições condicionais

- A implicação $p \rightarrow q$ pode ser entendida como uma promessa:

“Se você me garantir p , eu te garanto q .”

A promessa só é quebrada (ou falsa) quando:

- você me garantir p e eu não te garantir q em troca.

A promessa é mantida (ou verdadeira) quando:

- você me garante p e eu te garanto q , ou
- você não me garante p (e neste caso eu sou livre para te garantir q ou não sem quebrar a promessa.)

Dizemos que neste caso a implicação é verdadeira **por vacuidade**, pois a premissa é falsa.

Conectivos lógicos: Proposições condicionais

- Exemplo 11 Considere a implicação abaixo.

“Se eu ganhar o prêmio, eu vou dar uma festa.”

A proposição condicional só é falsa no caso em que eu ganho o prêmio e mesmo assim não dou uma festa.

Se eu não ganhar o prêmio, eu posso dar uma festa ou não, sem assim quebrar minha promessa.

Logo, se eu não ganhar o prêmio, a proposição condicional é verdadeira por vacuidade, independentemente de eu dar ou não uma festa. ●

Conectivos lógicos: Proposições condicionais

- Na lógica proposicional, o valor de verdade da implicação $p \rightarrow q$ só depende do valor de verdade de p e q , e não de seu significado.

Ao contrário do uso da implicação em linguagem natural, não existe necessariamente uma noção de “causa e efeito” entre p e q .

- A implicação lógica $p \rightarrow q$ apenas representa uma noção de “não-deterioração” da verdade:

“A conclusão q é no mínimo tão verdadeira quanto a premissa p .”

Note que a implicação só é falsa quando

“A conclusão q for menos verdadeira que a premissa p ”,

ou seja, quando p for verdadeiro e q for falso.

Conectivos lógicos: Proposições condicionais

- Exemplo 12

 Vamos analisar se as implicações abaixo são verdadeiras ou falsas.

- *"Se o sol emite luz, então queijos são laticínios."*

Proposição verdadeira: premissa e conclusão verdadeiras.

- *"Se $2 + 2 = 3$, então morangos são animais."*

Proposição verdadeira: premissa e conclusão falsas.

- *"Se a semana tem 7 dias, então o Brasil fica na Europa."*

Proposição falsa: premissa verdadeira e conclusão falsa.

- *"Se 9 é primo, então 12 é par."*

Proposição verdadeira: premissa falsa e conclusão verdadeira.



Proposições condicionais em linguagem natural

- Implicações aparecem na matemática e na linguagem natural em diversas formas.

A afirmação condicional $p \rightarrow q$ pode ser expressa como:

- | | |
|---------------------------------|--------------------------------|
| ● “se p , então q ” | ● “ q a menos que $\neg p$ ” |
| ● “se p , q ” | ● “ p implica q ” |
| ● “ q se p ” | ● “ p somente se q ” |
| ● “ q quando p ” | ● “ q sempre que p ” |
| ● “ p é suficiente para q ” | ● “ q segue de p ” |
| ● “ q é necessário para p ” | |

(Após fazer exercícios o suficiente, você vai se acostumar com as várias formas da implicação e tudo vai parecer mais natural!)

Proposições condicionais em linguagem natural

- Exemplo 13 Sejam as proposições:

p : “Está fazendo sol.”

q : “Eu vou ao clube.”

A implicação $p \rightarrow q$ pode ser escrita em linguagem natural como:

- “Se estiver fazendo sol, eu vou ao clube.”
- “Estar fazendo sol é condição suficiente para eu ir ao clube.”
- “Eu vou ao clube a menos que não esteja fazendo sol.”
- “O fato de eu ir ao clube segue do fato de estar fazendo sol.”
- “Eu vou ao clube sempre que faz sol.”
- “Faz sol somente se eu vou ao clube.”



Proposições condicionais: Conversa, contrapositiva e inversa

- Dada uma implicação $p \rightarrow q$:
 - sua forma **contrapositiva** é a implicação $\neg q \rightarrow \neg p$,
 - sua forma **conversa** é a implicação $q \rightarrow p$,
 - sua forma **inversa** é a implicação $\neg p \rightarrow \neg q$.

Proposições condicionais: Conversa, contrapositiva e inversa

- Exemplo 14 Seja a proposição

“Bruno vai bem na prova sempre que estuda com afinco.”

Esta implicação pode ser escrita como $p \rightarrow q$, em que

p é a proposição *“Bruno estuda com afinco”*, e

q é a proposição *“Bruno vai bem na prova”*.

- A contrapositiva $\neg q \rightarrow \neg p$ é a proposição

“Se Bruno não foi bem na prova, então ele não estudou com afinco.”

- A conversa $q \rightarrow p$ é a proposição

“Se Bruno foi bem na prova, ele estudou com afinco.”

- A inversa $\neg p \rightarrow \neg q$ é a proposição

“Se Bruno não estudou com afinco, ele não foi bem na prova.”

Conectivos lógicos: Proposições bicondicionais

- Sejam p e q proposições.

A **afirmação bicondicional** ou **implicação dupla** $p \leftrightarrow q$ é a afirmação

“ p se, e somente se, q ”.

A afirmação bicondicional $p \leftrightarrow q$ é verdadeira quando p e q têm o mesmo valor de verdade, e é falsa em caso contrário.

- Em linguagem natural é comum expressar $p \leftrightarrow q$ como:
 - “ p é necessário e suficiente para q .”
 - “ p sse q .” Note que usamos “sse” com dois “s”.(Em inglês, usa-se o “iff” com dois “f”.)

Conectivos lógicos: Proposições bicondicionais

- Tabela da verdade para a implicação dupla $p \leftrightarrow q$ entre duas proposições p e q :

Implicação dupla

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

- A proposição bicondicional $p \leftrightarrow q$ é verdadeira sempre que ambos $p \rightarrow q$ e $q \rightarrow p$ são verdadeiros, e ela é falsa em caso contrário.

Tabela da verdade de proposições compostas

- Nós introduzimos a negação e os conectivos lógicos de disjunção, conjunção, ou exclusivo, implicação e implicação dupla.
- Nós podemos usar estes operadores para expressar proposições cada vez mais complexas.
- Para determinar o valor de verdade de proposições compostas, podemos usar **tabelas da verdade**.

Exemplo 15 Tabela da verdade para a expressão $(p \vee \neg q) \rightarrow (p \wedge q)$:

p	q	$\neg q$	$p \vee \neg q$	$p \wedge q$	$(p \vee \neg q) \rightarrow (p \wedge q)$
T	T	F	T	T	T
T	F	T	T	F	F
F	T	F	F	F	T
F	F	T	T	F	F



Ordem de precedência dos operadores lógicos

- Note que a aritmética básica adota a seguinte precedência de operadores: a exponenciação tem precedência sobre a multiplicação, que tem precedência sobre a soma. Por exemplo:

$$\textcircled{1} \quad 5 + 4 \times 3^2 = 5 + (4 \times (3^2)) = 5 + (4 \times 9) = 5 + (36) = 41.$$

- Da mesma forma, em uma proposição composta a ordem de avaliação dos operadores é:

1) negação: \neg

3) disjunção: \vee

5) implicação dupla \leftrightarrow

2) conjunção: \wedge

4) implicação: \rightarrow

- Exemplos:

$$\textcircled{1} \quad p \vee \neg q \wedge r \quad \text{equivale a} \quad p \vee ((\neg q) \wedge r).$$

$$\textcircled{2} \quad p \rightarrow q \vee r \quad \text{equivale a} \quad p \rightarrow (q \vee r).$$

$$\textcircled{3} \quad \neg p \vee q \leftrightarrow p \rightarrow q \quad \text{equivale a} \quad ((\neg p) \vee q) \leftrightarrow (p \rightarrow q).$$

Operadores lógicos bit-a-bit

- Computadores representam informação usando **bits**, que são dígitos binários 0 ou 1.

O bit 1 representa T (verdadeiro), e o bit 0 representa F (falso).

- Conectivos lógicos podem ser utilizados para operar sobre **variáveis Booleanas**, cujos valores são bits.

A conjunção é chamada de **AND**, a disjunção é chamada de **OR**, e o ou exclusivo é chamado de **XOR** (*exclusive or*).

Tabela da verdade para os operadores AND, OR e XOR.

x	y	x AND y	x OR y	x XOR y
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

Operadores lógicos bit-a-bit

- Uma **string de bits** é uma sequência de zero ou mais bits.

O comprimento de uma string é o número de bits nesta string.

- Exemplos:

❶ 101 é uma string de bits de comprimento 3.

❷ 0100111 é uma string de bits de comprimento 7,

- Operadores binários podem ser aplicados **bit a bit** em duas strings de bits de mesmo comprimento. O resultado é uma string de mesmo comprimento.

- Exemplo 16

	0110
	1101
<hr/>	
resultado de <i>AND</i> bit a bit:	0100
resultado de <i>OR</i> bit a bit:	1111
resultado de <i>XOR</i> bit a bit:	1011

Aplicações da Lógica Proposicional

Aplicações da lógica proposicional: Introdução

- A lógica tem importantes aplicações na matemática, ciência da computação, e diversas outras disciplinas:
 - ➊ tradução de sentenças em linguagem natural, frequentemente ambíguas, para uma linguagem precisa,
 - ➋ especificação de circuitos lógicos,
 - ➌ solução de quebra-cabeças (o que é essencial para inteligência artificial),
 - ➍ automatização do processo de construção de demonstrações matemáticas,
 - ➎ ...
- Nesta seção vamos exemplificar algumas destas aplicações práticas da lógica proposicional.

Traduzindo sentenças em linguagem natural

- Sentenças em linguagem natural são frequentemente ambíguas, o que pode causar problemas de comunicação.
- Traduzir sentenças em linguagem natural para proposições compostas remove a ambiguidade.
- Uma vez traduzidas para proposições lógicas, estas sentenças podem ser analisadas quanto ao seu valor de verdade.

Traduzindo sentenças em linguagem natural

- Exemplo 17 Seja a sentença em linguagem natural:

“Você não pode andar na montanha russa se você tiver menos que 1.50m de altura, a menos que você tenha mais de 16 anos.”

Podemos traduzí-la para uma proposição composta, usamos as seguintes proposições:

- q : “você pode andar na montanha russa”,
- r : “você tem menos que 1.50m de altura”,
- s : “você tem mais de 16 anos”.

A sentença em linguagem natural é, então, traduzida para:

$$(r \wedge \neg s) \rightarrow \neg q.$$

(Note que esta não é a única maneira de representar a expressão lógica, mas é uma maneira válida.)

Especificação de sistemas

- Traduzir sentenças de linguagem natural para linguagem lógica é parte essencial da especificação de sistemas de hardware e software.

Exemplo 18 Exprese a especificação abaixo como uma proposição composta.

“A resposta automática não pode ser enviada quando o sistema de arquivos está cheio.”

Solução. Podemos traduzir a especificação para uma proposição composta, usando as seguintes proposições:

- r : *“a resposta automática pode ser enviada”*,
- c : *“o sistema de arquivos está cheio”*.

A especificação fica, então, traduzida para:

$$c \rightarrow \neg r.$$

Especificação de sistemas

- Especificações de sistemas devem ser **consistentes**, isto é, não devem conter requisitos conflitantes que poderiam ser usadas para derivar uma contradição.

Quando as especificações não são consistentes, não é possível desenvolver um sistema que satisfaça todos os requisitos.

- **Exemplo 19** Determine se a seguinte especificação de sistema é consistente:
 - *“A mensagem de diagnóstico é armazenada no buffer ou é retransmitida.”*
 - *“A mensagem de diagnóstico não está armazenada no buffer.”*
 - *“Se a mensagem de diagnóstico estiver armazenada no buffer, ela será retransmitida.”*

Solução. Para determinar se essas especificações são consistentes, primeiro vamos representá-las como expressões lógicas:

- p : *“A mensagem de diagnóstico é armazenada no buffer.”*
- q : *“A mensagem de diagnóstico é retransmitida.”*

Especificação de sistemas

- Exemplo 19 (Continuação)

Assim, a especificação do sistema pode ser reescrita como:

- $p \vee q$

- $\neg p$

- $p \rightarrow q$

Podemos verificar se especificação é consistente com uma tabela da verdade que mostra se é possível satisfazer todos os três requisitos ao mesmo tempo.

p	q	$p \vee q$	$\neg p$	$p \rightarrow q$
T	T	T	F	T
T	F	T	F	F
F	T	T	T	T
F	F	F	T	T

Note que a especificação do sistema é consistente, e ela é satisfeita quando $p = F$ e $q = T$.



Especificação de sistemas

- Exemplo 20 Suponha que adicionemos à especificação do exemplo anterior o seguinte requisito:

- “A mensagem de diagnóstico não é retransmitida.”*

Neste caso a especificação do sistema continua consistente?

Solução. O novo requisito pode ser representado logicamente como $\neg q$, e a tabela da verdade atualizada é a seguinte.

p	q	$p \vee q$	$\neg p$	$p \rightarrow q$	$\neg q$
T	T	T	F	T	F
T	F	T	F	F	T
F	T	T	T	T	F
F	F	F	T	T	T

Pode-se notar que agora é impossível satisfazer todos os requisitos ao mesmo tempo e, portanto, a especificação do sistema é inconsistente.

Resolução de problemas e quebra-cabeças

- **Quebra-cabeças** (ou **puzzles**) que podem ser resolvidos usando raciocínio lógico são conhecidos como **quebra-cabeças** (ou **puzzles**) **lógicos**.
- Exemplo 21

 Considere uma ilha em que há apenas dois tipos de habitantes: cavaleiros, que só falam a verdade, e cavilosos, que só falam mentiras.

Você encontra duas pessoas, A e B .

A pessoa A diz:

“ B só diz a verdade”,

enquanto a pessoa B diz:

“ A e eu somos pessoas de tipos diferentes”.

Qual o tipo de A e o tipo de B ?

Solução.

Desafio para o(a) estudante!

Equivalências Proposicionais

Equivalência de proposições: Introdução

- Um passo importante na resolução de muitos problemas é a substituição de uma afirmação por outra com mesmo valor de verdade.
- Nesta seção vamos estudar como determinar se duas proposições compostas têm sempre o mesmo valor de verdade.
- O conceito de equivalência de proposições lógicas é análogo ao conceito de equivalência algébrica: duas expressões algébricas são equivalentes quando elas têm o mesmo valor para qualquer atribuição de suas variáveis.

1 As expressões

$$(x + 1)^2 \quad \text{e} \quad (x^2 + 2x + 1)$$

são equivalentes porque elas têm o mesmo valor para toda atribuição real a x .

2 As expressões

$$x^2 \quad \text{e} \quad (x + 2)$$

não são equivalentes porque, apesar de terem o mesmo valor para algumas atribuições de x ($x = -1$ ou $x = 2$), isso não ocorre sempre (quando $x = 0$).

Equivalência de proposições: Introdução

- Primeiro, vamos categorizar os tipos de expressões compostas:
 - uma **tautologia** é uma expressão sempre verdadeira independentemente o valor de verdade das variáveis que nela aparecem;
 - uma **contradição** é uma expressão sempre falsa independentemente o valor de verdade das variáveis que nela aparecem;
 - uma **contingência** é uma expressão que não é nem uma tautologia, nem uma contradição.
- **Exemplo 22** A tabela da verdade abaixo mostra que $(p \wedge \neg p)$ é uma contradição, enquanto a expressão $(p \vee \neg p)$ é uma tautologia.

p	$\neg p$	$p \wedge \neg p$	$p \vee \neg p$
T	F	F	T
F	T	F	T



Equivalências lógicas

- Duas proposições compostas p e q são **logicamente equivalentes** se $p \leftrightarrow q$ é uma tautologia.

A notação $p \equiv q$ denota que p e q são logicamente equivalentes.

- Uma maneira de determinar se $p \equiv q$ é usando tabelas da verdade.
- Exemplo 23 Mostre que $p \rightarrow q$ e $\neg p \vee q$ são logicamente equivalentes.

Solução.

Tabela da verdade para $p \rightarrow q$ e $\neg p \vee q$:

p	q	$\neg p$	$\neg p \vee q$	$p \rightarrow q$
T	T	F	T	T
T	F	F	F	F
F	T	T	T	T
F	F	T	T	T

Como a coluna correspondente a $p \rightarrow q$ e a coluna correspondente a $\neg p \vee q$ possuem sempre o mesmo valor de verdade, $(p \rightarrow q) \leftrightarrow (\neg p \vee q)$ é uma tautologia.

Logo $p \rightarrow q \equiv (\neg p \vee q)$.



Equivalências lógicas

- Exemplo 24 Mostre que $\neg(p \vee q)$ e $\neg p \wedge \neg q$ são logicamente equivalentes.

Solução. Tabela da verdade para $\neg(p \vee q)$ e $\neg p \wedge \neg q$:

p	q	$p \vee q$	$\neg(p \vee q)$	$\neg p$	$\neg q$	$\neg p \wedge \neg q$
T	T	T	F	F	F	F
T	F	T	F	F	T	F
F	T	T	F	T	F	F
F	F	F	T	T	T	T

Uma vez que a coluna correspondente a $\neg(p \vee q)$ e a coluna correspondente a $\neg p \wedge \neg q$ possuem sempre o mesmo valor de verdade, $\neg(p \vee q) \leftrightarrow (\neg p \wedge \neg q)$ é uma tautologia.

Logo, $\neg(p \vee q) \equiv \neg p \wedge \neg q$.

A equivalência que demonstramos é uma das **Leis de De Morgan**:

Leis de De Morgan

$\neg(p \vee q) \equiv \neg p \wedge \neg q$
$\neg(p \wedge q) \equiv \neg p \vee \neg q$

Equivalências lógicas

- Exemplo 25 Mostre que $p \vee (q \wedge r)$ e $(p \vee q) \wedge (p \vee r)$ são logicamente equivalentes.

Solução. Tabela da verdade para $p \vee (q \wedge r)$ e $(p \vee q) \wedge (p \vee r)$:

p	q	r	$q \wedge r$	$p \vee (q \wedge r)$	$p \vee q$	$p \vee r$	$(p \vee q) \wedge (p \vee r)$
T	T	T	T	T	T	T	T
T	T	F	F	T	T	T	T
T	F	T	F	T	T	T	T
T	F	F	F	T	T	T	T
F	T	T	T	T	T	T	T
F	T	F	F	F	T	F	F
F	F	T	F	F	F	T	F
F	F	F	F	F	F	F	F

Como as colunas correspondentes a $p \vee (q \wedge r)$ e $(p \vee q) \wedge (p \vee r)$ possuem sempre o mesmo valor de verdade, $p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$. •

A equivalência que demonstramos é a **lei da distributividade da disjunção sobre a conjunção**.

Equivalências lógicas

- Algumas equivalências lógicas importantes:

Nome	Equivalência
Leis de identidade	$p \wedge T \equiv p$ $p \vee F \equiv p$
Leis de dominância	$p \wedge F \equiv F$ $p \vee T \equiv T$
Leis de idempotência	$p \wedge p \equiv p$ $p \vee p \equiv p$
Lei da dupla negação	$\neg(\neg p) \equiv p$
Leis de comutatividade	$p \wedge q \equiv q \wedge p$ $p \vee q \equiv q \vee p$
Leis de associatividade	$(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$ $(p \vee q) \vee r \equiv p \vee (q \vee r)$
Leis de distributividade	$p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$ $p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$
Leis de De Morgan	$\neg(p \wedge q) \equiv \neg p \vee \neg q$ $\neg(p \vee q) \equiv \neg p \wedge \neg q$
Leis de absorção	$p \vee (p \wedge q) \equiv p$ $p \wedge (p \vee q) \equiv p$
Leis da negação	$p \wedge \neg p \equiv F$ $p \vee \neg p \equiv T$

Equivalências lógicas

- Equivalências lógicas envolvendo proposições condicionais:

Equivalências
$p \rightarrow q \equiv \neg p \vee q$
$p \rightarrow q \equiv \neg q \rightarrow \neg p$
$p \vee q \equiv \neg p \rightarrow q$
$p \wedge q \equiv \neg(p \rightarrow \neg q)$
$\neg(p \rightarrow q) \equiv p \wedge \neg q$
$(p \rightarrow q) \wedge (p \rightarrow r) \equiv p \rightarrow (q \wedge r)$
$(p \rightarrow r) \wedge (q \rightarrow r) \equiv (p \vee q) \rightarrow r$
$(p \rightarrow q) \vee (p \rightarrow r) \equiv p \rightarrow (q \vee r)$
$(p \rightarrow r) \vee (q \rightarrow r) \equiv (p \wedge q) \rightarrow r$

- Equivalências lógicas envolvendo proposições bicondicionais:

Equivalências
$p \leftrightarrow q \equiv (p \rightarrow q) \wedge (q \rightarrow p)$
$p \leftrightarrow q \equiv \neg p \leftrightarrow \neg q$
$p \leftrightarrow q \equiv (p \wedge q) \vee (\neg p \wedge \neg q)$
$\neg(p \leftrightarrow q) \equiv p \leftrightarrow \neg q$

Usos das Leis de De Morgan

- As duas equivalências lógicas conhecidas como Leis de De Morgan são particularmente importantes.
- A lei

$$\neg(p \vee q) \equiv \neg p \wedge \neg q$$

diz que a negação da disjunção é a conjunção das negações.

- Exemplo 26 Use as Leis de De Morgan para negar a proposição:

“Danilo vai assistir a Star Wars ou vai assistir a O Senhor dos Anéis.”

Solução. Esta proposição pode ser escrita como $p \vee q$, onde p é *“Danilo vai assistir a Star Wars”*, e q é *“Danilo vai assistir a O Senhor dos Anéis”*.

Pela lei de De Morgan, a negação é $\neg(p \vee q) \equiv \neg p \wedge \neg q$, que se traduz em

“Danilo não vai assistir a Star Wars e nem vai assistir a O Senhor dos Anéis.”



Usos das Leis de De Morgan

- A lei de De Morgan

$$\neg(p \wedge q) \equiv \neg p \vee \neg q$$

diz que a negação da conjunção é a disjunção das negações.

- Exemplo 27 Use as Leis de De Morgan para negar a proposição:

“Estela tem um celular e um computador.”

Solução. Esta proposição pode ser escrita como $p \wedge q$, onde p é *“Estela tem um celular”*, e q é *“Estela tem um computador”*.

Pelas Leis de De Morgan, a negação é $\neg(p \wedge q) \equiv \neg p \vee \neg q$, que se traduz em

“Estela não tem um celular ou ela não tem um computador.”



Construindo novas equivalências lógicas

- Como vimos, tabelas da verdade podem ser utilizadas para verificar equivalências lógicas.
- No geral, para expressões envolvendo n variáveis lógicas são necessárias 2^n linhas na tabela.

Para n grande, pode ser inconveniente construir a tabela da verdade. Por exemplo:

① Quanto $n = 3$, temos $2^n = 8$

② Quando $n = 10$, temos $2^n = 1024$

- Uma alternativa é utilizar equivalências lógicas já conhecidas para derivar novas equivalências lógicas diretamente.

Construindo novas equivalências lógicas

- Exemplo 28 Mostre que $\neg(p \rightarrow q)$ e $p \wedge \neg q$ são logicamente equivalentes.

Solução.

$$\begin{aligned}\neg(p \rightarrow q) &\equiv \neg(\neg p \vee q) && \text{(pela tabela de equiv. de condicionais)} \\ &\equiv \neg(\neg p) \wedge \neg q && \text{(pelas Leis de De Morgan)} \\ &\equiv p \wedge \neg q && \text{(pela lei da dupla negação)}\end{aligned}$$

Note que esta equivalência faz sentido intuitivamente: dizer que uma implicação é falsa ($\neg(p \rightarrow q)$) é o mesmo que dizer que sua hipótese é verdadeira mas sua conclusão é falsa ($p \wedge \neg q$).



Construindo novas equivalências lógicas

- Exemplo 29 Mostre que $\neg(p \vee (\neg p \wedge q))$ e $\neg p \wedge \neg q$ são logicamente equivalentes.

Solução.

$$\begin{aligned}\neg(p \vee (\neg p \wedge q)) &\equiv \neg p \wedge \neg(\neg p \wedge q) && \text{(pelas Leis de De Morgan)} \\ &\equiv \neg p \wedge (\neg(\neg p) \vee \neg q) && \text{(pelas Leis de De Morgan)} \\ &\equiv \neg p \wedge (p \vee \neg q) && \text{(pela lei da dupla negação)} \\ &\equiv (\neg p \wedge p) \vee (\neg p \wedge \neg q) && \text{(pela lei da distributividade)} \\ &\equiv F \vee (\neg p \wedge \neg q) && \text{(porque } \neg p \wedge p \equiv F) \\ &\equiv (\neg p \wedge \neg q) \vee F && \text{(pela lei da comutatividade)} \\ &\equiv \neg p \wedge \neg q && \text{(pela lei de identidade)}\end{aligned}$$



Construindo novas equivalências lógicas

- Exemplo 30 Mostre que $(p \wedge q) \rightarrow (p \vee q)$ é uma tautologia.

Solução.

$$\begin{aligned}(p \wedge q) \rightarrow (p \vee q) &\equiv \neg(p \wedge q) \vee (p \vee q) && \text{(equivalência de condicionais)} \\ &\equiv (\neg p \vee \neg q) \vee (p \vee q) && \text{(pela Lei de De Morgan)} \\ &\equiv (\neg p \vee p) \vee (\neg q \vee q) && \text{(comutatividade e associatividade)} \\ &\equiv T \vee T && \text{(pela lei de negação)} \\ &\equiv T && \text{(pela lei de dominância)}\end{aligned}$$



Satisfatibilidade

- Uma expressão composta é **satisfazível** quando existe uma atribuição de valores de verdade a suas variáveis que torna a expressão verdadeira.

Se tal atribuição de valores não existe, a expressão é **não-satisfazível** ou **insatisfazível**.

O problema de determinar se uma expressão é satisfazível é chamado de **problema de satisfatibilidade (SAT)**.

- Note que uma expressão composta é satisfazível quando ela é uma tautologia ou uma contingência (mas não uma contradição).

Além disso, uma expressão é insatisfazível quando ela é uma contradição.

Satisfatibilidade

- Uma atribuição de valores de verdade às variáveis de uma expressão composta que torna a mesma verdadeira é chamada de **solução** do problema de satisfatibilidade em questão.
- Para mostrar que uma expressão é satisfazível basta encontrar uma solução.
Já para mostrar que uma expressão é insatisfazível, temos que mostrar que nenhuma atribuição de valores verdade às suas variáveis é uma solução.
Podemos fazer isso usando tabelas da verdade, mas isso pode ser trabalhoso...

Satisfatibilidade

- Exemplo 31 Determine se cada uma das expressões abaixo é satisfazível.

① $(p \vee \neg q) \wedge (q \vee \neg r) \wedge (r \vee \neg p)$

③ $(p \leftrightarrow q) \wedge (q \leftrightarrow r) \wedge (p \oplus r)$

② $(p \vee q \vee r) \wedge (\neg p \vee \neg q \vee \neg r)$

Solução.

- ① $(p \vee \neg q) \wedge (q \vee \neg r) \wedge (r \vee \neg p)$ é satisfazível: a atribuição $p = T, q = T, r = T$ é uma solução que torna a expressão verdadeira.
- ② $(p \vee q \vee r) \wedge (\neg p \vee \neg q \vee \neg r)$ é satisfazível: a atribuição $p = T, q = F, r = F$ é uma solução que torna a expressão verdadeira.
- ③ $(p \leftrightarrow q) \wedge (q \leftrightarrow r) \wedge (p \oplus r)$ é instatisfazível: uma tabela da verdade mostra que nenhuma atribuição de valores a p, q, r torna a expressão verdadeira.

Uma alternativa à construção da tabela da verdade é notar que a expressão não tem solução porque ela exige que p e q tenham o mesmo valor de verdade ($p \leftrightarrow q$), assim como q e r ($q \leftrightarrow r$), mas também exige que p e r tenham valores diferentes ($p \oplus r$), o que é impossível de acontecer ao mesmo tempo.

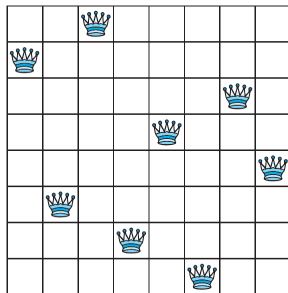
Apêndice - O problema das n rainhas e Solucionadores SAT

Satisfatibilidade: Aplicações

- Muitos problemas em ciência da computação (robótica, inteligência artificial, visão computacional, ...) e em outras áreas (genética, planejamento de cronogramas, ...) podem ser modelados como problemas de satisfatibilidade.
- Exemplo 32** **O problema da n rainhas.** O problema consiste em posicionar n rainhas em um tabuleiro de xadrez de tamanho $n \times n$ sem que nenhuma rainha possa “atacar” outra rainha.

Isso quer dizer que não pode haver duas rainhas posicionadas em uma mesma linha, em uma mesma coluna, ou em uma mesma diagonal.

Por exemplo, uma solução para o problema quando há $n = 8$ rainhas é dada ao lado.



Satisfatibilidade: Aplicações

- Exemplo 32 (Continuação)

Podemos modelar o problema das n rainhas como um problema de satisfatibilidade como a seguir.

Sejam n^2 variáveis $p(i, j)$, com

- $i = 1, 2, \dots, n$

- $j = 1, 2, \dots, n.$

Dada uma configuração do tabuleiro:

$$p(i, j) = \begin{cases} T, & \text{se há uma rainha na linha } i \text{ e coluna } j, \text{ e} \\ F, & \text{se não há uma rainha na linha } i \text{ e coluna } j. \end{cases}$$

No exemplo anterior, temos que

- ① $p(6, 2) = p(2, 1) = T$

- ② $p(3, 4) = p(5, 4) = F.$

- Exemplo 32 (Continuação)

Uma questão central do problema é identificar quando duas casas do tabuleiro estão na mesma diagonal.

Note que dizer que duas casas (i, j) e (i', j') estão na mesma diagonal significa dizer que para ir da casa (i, j) à casa (i', j') precisamos mover o mesmo número de casas na vertical (linhas) e na horizontal (colunas).

Em outras palavras, (i, j) e (i', j') estão na mesma diagonal se a distância entre as linhas e as colunas de (i, j) e (i', j') for a mesma:

$$\begin{aligned} i - i' &= j - j' && \text{ou} \\ i - i' &= -(j - j') . \end{aligned}$$

Satisfatibilidade: Aplicações

- Exemplo 32 (Continuação)

Agora estamos prontos para verificar as condições necessárias para o problema ter solução:

- Há pelo menos uma rainha em cada linha:

$$Q_1 : \bigwedge_{i=1}^n \bigvee_{j=1}^n p(i, j)$$

Por exemplo, no caso de um tabuleiro 3×3 , a condição acima equivale a:

$$\begin{aligned} Q_1 : & (p(1, 1) \vee p(1, 2) \vee p(1, 3)) \wedge \\ & (p(2, 1) \vee p(2, 2) \vee p(2, 3)) \wedge \\ & (p(3, 1) \vee p(3, 2) \vee p(3, 3)) \end{aligned}$$

Satisfatibilidade: Aplicações

- Exemplo 32 (Continuação)

Note que, para o problema ter solução, as seguintes condições têm que ser satisfeitas:

- Há no máximo uma rainha em cada linha:

$$Q_2 : \bigwedge_{i=1}^n \bigwedge_{j=1}^{n-1} \bigwedge_{k=j+1}^n (\neg p(i, j) \vee \neg p(i, k))$$

Por exemplo, no caso de um tabuleiro 3×3 , a condição acima equivale a:

$$\begin{aligned} Q_2 : & ((\neg p(1, 1) \vee \neg p(1, 2)) \wedge (\neg p(1, 1) \vee \neg p(1, 3)) \wedge (\neg p(1, 2) \vee \neg p(1, 3))) \wedge \\ & ((\neg p(2, 1) \vee \neg p(2, 2)) \wedge (\neg p(2, 1) \vee \neg p(2, 3)) \wedge (\neg p(2, 2) \vee \neg p(2, 3))) \wedge \\ & ((\neg p(3, 1) \vee \neg p(3, 2)) \wedge (\neg p(3, 1) \vee \neg p(3, 3)) \wedge (\neg p(3, 2) \vee \neg p(3, 3))) \end{aligned}$$

Satisfatibilidade: Aplicações

- Exemplo 32 (Continuação)

Note que, para o problema ter solução, as seguintes condições têm que ser satisfeitas:

- Há no máximo uma rainha em cada coluna:

$$Q_3 : \bigwedge_{j=1}^n \bigwedge_{i=1}^{n-1} \bigwedge_{k=i+1}^n (\neg p(i,j) \vee \neg p(k,j))$$

Por exemplo, no caso de um tabuleiro 3×3 , a condição acima equivale a:

$$\begin{aligned} Q_3 : & ((\neg p(1,1) \vee \neg p(2,1)) \wedge (\neg p(1,1) \vee \neg p(3,1)) \wedge (\neg p(2,1) \vee \neg p(3,1))) \wedge \\ & ((\neg p(1,2) \vee \neg p(2,2)) \wedge (\neg p(1,2) \vee \neg p(3,2)) \wedge (\neg p(2,2) \vee \neg p(3,2))) \wedge \\ & ((\neg p(1,3) \vee \neg p(2,3)) \wedge (\neg p(1,3) \vee \neg p(3,3)) \wedge (\neg p(2,3) \vee \neg p(3,3))) \end{aligned}$$

(Note que Q_3 e Q_1 (toda linha contém pelo menos uma rainha) juntas implicam que cada coluna contém pelo menos uma rainha.)

Satisfatibilidade: Aplicações

- Exemplo 32 (Continuação)

Note que, para o problema ter solução, as seguintes condições têm que ser satisfeitas:

- Nenhuma diagonal contém duas rainhas:

$$Q_4 : \bigwedge_{i=2}^n \bigwedge_{j=1}^{n-1} \bigwedge_{k=1}^{\min(i-1, n-j)} (\neg p(i, j) \vee \neg p(i-k, k+j))$$

$$Q_5 : \bigwedge_{i=1}^{n-1} \bigwedge_{j=1}^{n-1} \bigwedge_{k=1}^{\min(n-i, n-j)} (\neg p(i, j) \vee \neg p(i+k, j+k))$$

(Encontrar a expansão das condições acima para o caso de um tabuleiro 3×3 fica como exercício para o(a) estudante!)

Satisfatibilidade: Aplicações

- Exemplo 32 (Continuação)

Juntando todas as condições, concluímos que as soluções para o problema das n rainhas são dadas pelas atribuições de valores de verdade para as variáveis $p(i, j)$, com $i = 1, 2, \dots, n$ e $j = 1, 2, \dots, n$, que satisfazem

$$Q = Q_1 \wedge Q_2 \wedge Q_3 \wedge Q_4 \wedge Q_5 .$$

Como você pode notar, esta fórmula Q possui muitas variáveis proposicionais.

Resolver o problema à mão (com, e.g., tabelas da verdade) é trabalhoso!

Técnicas de satisfatibilidade permitem verificar quantas soluções existem:

- Para $n = 8$ rainhas, há 92 maneiras de se posicionarem as rainhas no tabuleiro.
- Para $n = 16$ rainhas, o número sobe para 14 772 512.

(Em outros cursos vocês poderão estudar estas técnicas a fundo!)



Satisfatibilidade: Solucionadores SAT

- Como vimos no exemplo anterior, alguns problemas matemáticos podem ser modelados como um problema de satisfatibilidade em que precisamos determinar se (e, em caso positivo, de quantas formas) uma expressão proposicional pode ser satisfeita.
- Uma tabela de verdade pode ser usada para determinar se uma proposição composta é satisfazível ou, equivalentemente, se a sua negação é uma tautologia.

(Você pode, como exercício, provar que a negação de uma proposição composta insatisfazível é uma tautologia e que a negação de uma proposição composta que é uma tautologia é insatisfazível.)

- Isto funciona bem quando a proposição composta tem um pequeno número de variáveis, mas se torna impraticável quando o número de variáveis cresce.

Por exemplo, se uma proposição composta tem 20 variáveis, a tabela da verdade tem $2^{20} = 1\,048\,576$ linhas!

Ninguém faria isso à mão, é preciso um computador!

Satisfatibilidade: Solucionadores SAT

- Em várias aplicações práticas, o número de variáveis é muito maior que 20, podendo chegar a centenas, milhares ou milhões de variáveis!
 - Pergunta: Quando existem 1 000 variáveis, verificar cada uma das 2^{1000} linhas da tabela da verdade demoraria quanto tempo em nossos computadores mais rápidos?
 - Resposta: Trilhões de anos! (Muito mais tempo que o universo tem de vida!)

Não se conhece nenhum procedimento (ou algoritmo, ou programa de computador) para se determinar em tempo razoável se uma proposição composta com tantas variáveis é satisfazível.

(Vocês vão revisar esta questão no curso, o famoso **problema P vs. NP** .)

- No entanto, houve progresso em desenvolver métodos resolver problemas de satisfatibilidade que surgem em aplicações práticas, como Sudoku (que também pode ser resolvido usando satisfatibilidade!).

(Em outros cursos vocês poderão estudar estas técnicas a fundo!)