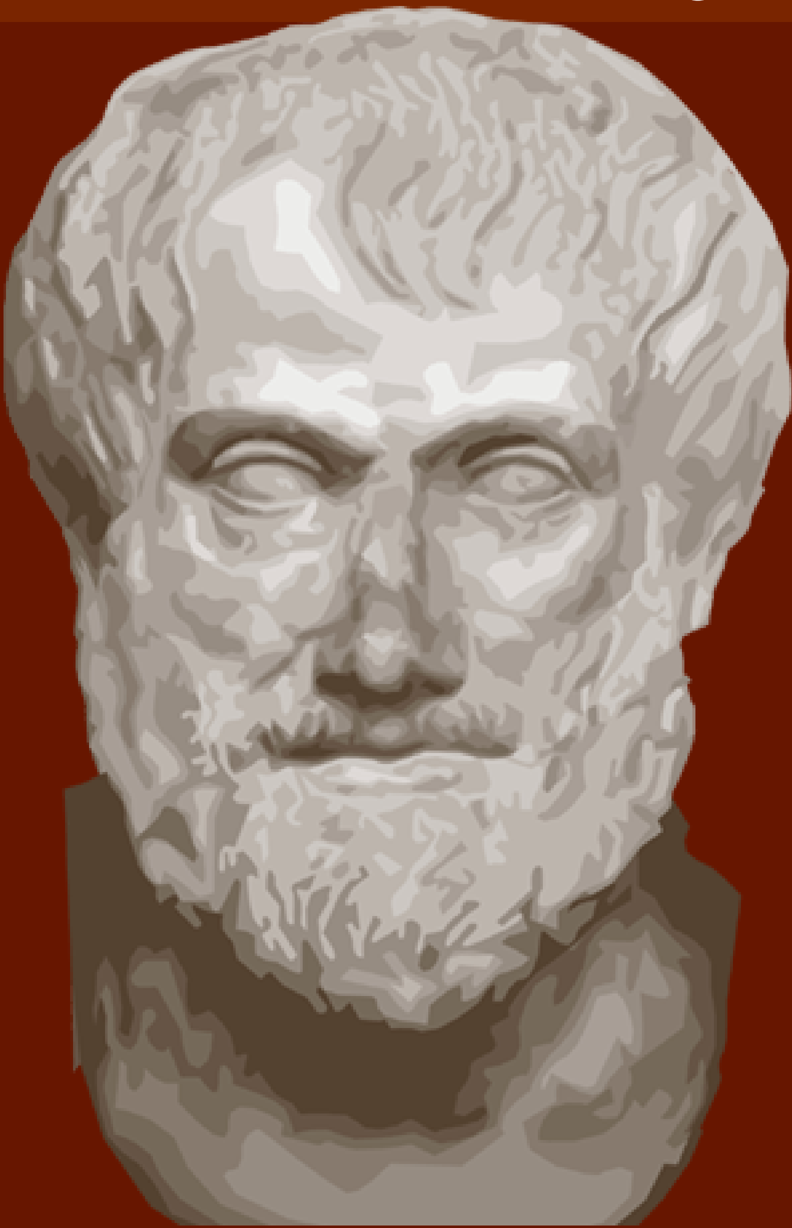
LÓGICA PARA COMPUTAÇÃO



APOSTILA – JOÃO VICTOR DE SÁ FERRAZ COUTINHO

Este material é muito fortemente baseado nas aulas dos professores **Ruy José Guerra Barreto de Queiroz** e **Anjolina Grisi de Oliveira**, que ministram as aulas de Lógica para Computação (IF673), no CIN-UFPE.
Julho, 2018

Sumário

0 Introdução à Lógica **5**

0.1 Argumento 5

Ato de inferência 7

0.2 Consistência 10

Exercícios 12

Parte I Lógica Proposicional

1 Sintaxe **17**

1.1 Alfabeto 18

Expressões legítimas 18

1.2 Conjuntos indutivos 20

Fecho indutivo 22

Conjunto livremente gerado 23

1.3 Funções recursivas sobre PROP 27

Provas de propriedades sobre PROP 29

Exercícios 33

2 Semântica **37**

2.1 Valoração-verdade 37

Teorema da Extensão Homomórfica Única . . 38

Satisfatibilidade 41

2.2 Conjunto funcionalmente completo 44

Exercícios 46

3 Problema da Satisfatibilidade 49

3.1 Método da Tabela-Verdade 49
Complexidade computacional 52
Corretude e completude 52
3.2 Método dos Tableaux Analíticos 52
Complexidade computacional 58
3.3 Método da Resolução 59
Regra da Resolução 61
Complexidade computacional 66
3.4 Método da Dedução Natural 66
Regras de dedução 66
Dedutibilidade 67
Negação 71
As três lógicas 73

Parte II Lógica de Primeira Ordem

4 Sintaxe 77

5 Semântica 79

6 Problema da Satisfatibilidade 81

CAPÍTULO 0 | Introdução à Lógica

A Lógica surgiu há cerca de 350 a.C., com os estudos de Aristóteles (384-322 a.C.). Ele buscou elencar as bases para a chamada **ciência da argumentação**: o estudo da forma dos argumentos, o qual veremos neste capítulo.

0.1 ARGUMENTO

Tomando as clássicas sentenças *"Todo homem é mortal"* e *"Sócrates é homem"*, podemos inferir que *"Sócrates é mortal"*. Note que podemos avaliar as três sentenças como **verdadeiras**, uma vez que elas obedecem o senso comum. Por outro lado, a sentença *"Toda sexta é um dia chuvoso"* é uma sentença claramente **falsa**, pois nem toda sexta é de fato chuvosa. Sentenças desse tipo, que podem ser classificadas como verdadeiras ou falsas (ou seja, que admitem um **valor-verdade**), são chamadas **sentenças declarativas** ou enunciados. Segundo Aristóteles, existem dois tipos de sentenças declarativas:

TIPO I: As que relacionam objetos a categorias.

x é P – *x é um objeto (ou indivíduo) e P é uma categoria (ou coleção).*

x não é P

TIPO II: As que relacionam categorias a categorias.

Todo P é Q – *Universal positiva.*

Nenhum P é Q – *Universal negativa.*

Algum P é Q – *Existencial positiva.*

Algum P não é Q – *Existencial negativa.*

Note que, se tentarmos afirmar "Recife tem praia." e "Recife não tem praia.", duas sentenças do 1º tipo, encontramos um erro lógico, pois é impossível Recife ter praia e não ter ao mesmo tempo. Esse erro – causado quando combinações de sentenças declarativas levam a um absurdo ou impossibilidade lógica – chama-se **contradição** e sua ausência foi identificada por Aristóteles como um requisito fundamental que deve ser atendido por um bom argumento. Para sentenças do tipo (I), identificar a contradição é fácil. Afinal, é impossível x ser P e não ser P ao mesmo tempo. O mesmo não acontece para as sentenças do tipo (II). Assim, para lidar com estas, Aristóteles usou o Quadrado das Oposições:

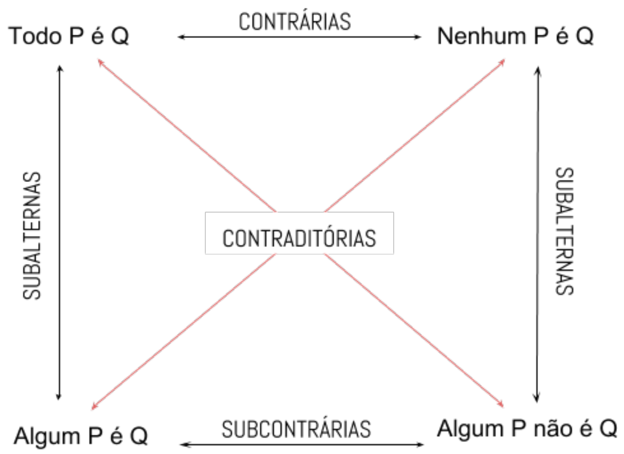


FIGURA 0.1
O Quadrado das Oposições

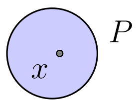
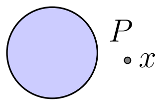
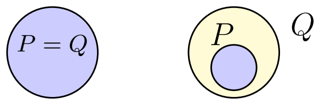
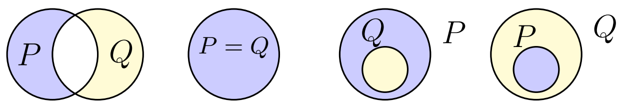
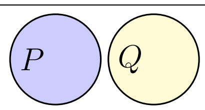
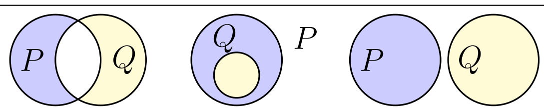
Sentenças **contrárias** não podem ser ambas verdadeiras, mas podem ser ambas falsas. Sentenças **subcontrárias** não podem ser ambas falsas, mas podem ser ambas verdadeiras. Sentenças **contraditórias** não podem assumir o mesmo valor-verdade (ou seja, não podem ser nem ambas verdadeiras nem ambas falsas). Um exemplo de sentenças contraditórias é *"Todas as estrelas são brancas"* e *"Alguma estrela não é branca"*. É fácil identificar que, se todas as estrelas são brancas, é impossível que alguma estrela não seja branca.

ATO DE INFERÊNCIA

Além dos elementos básicos de um argumento definidos por Aristóteles, surge a necessidade de caracterizá-lo como algo não apenas descritivo. Isso é possível ao se verificar que um argumento envolve um ou mais **atos de inferência**: uma sequência de sentenças tomadas como **premissas**, que permitem tirar **conclusões**, sob a forma de uma nova sentença.

$$\begin{array}{rc} & \text{Premissa 1} \\ & \text{Premissa 2} \\ & \ldots \\ & \text{Premissa n} \\ \hline \textbf{logo,} & \text{Conclusão} \end{array}$$

Um ato de inferência é dito **válido** se *se todas as premissas forem verdadeiras, a conclusão também for verdadeira*. Um ato de inferência válido é chamado de **silogismo**. Ao analisar sentenças e atos de inferências, podemos representá-los com os velhos conhecidos Diagramas de Venn:

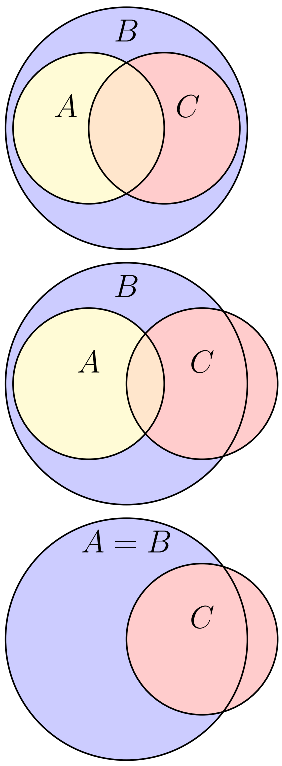
x é P	
x não é P	
Todo P é Q	
Algum P é Q	
Nenhum P é Q	
Algum P não é Q	

Assim, para verificar a validez de um ato de inferência, devemos passar por todas as situações em que as premissas são verdadeiras e verificar que, em todas elas, a conclusão também é verdadeira.

EXEMPLO 0.1 . . . .
Analisemos o ato de inferência abaixo.

$$\begin{array}{rc} & \text{Todo A é B.} \\ & \text{Algum A é C.} \\ \hline \textbf{logo,} & \text{Algum B é C.} \end{array}$$

Se dispormos todas as situações em que as premissas são verdadeiras, temos:

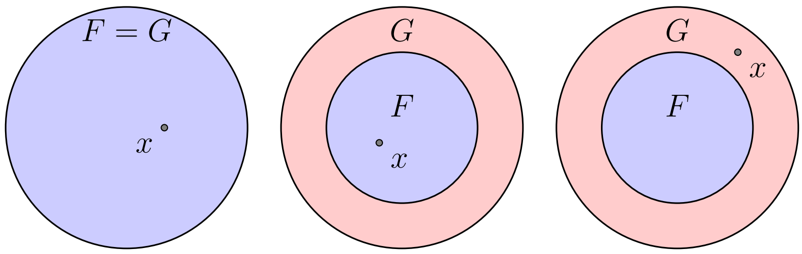


Temos que, em todas elas, a conclusão, *Algum B é C*, é verdadeira. Assim, o ato de inferência é válido.

EXEMPLO 0.2 . . . .
O seguinte ato de inferência é inválido:

$$\begin{array}{rc} & \text{Todo F é G.} \\ & \text{x é G.} \\ \cline{2-2} \textbf{logo,} & \text{x é F.} \end{array}$$

Note a semelhança com a clássica falácia aristotélica *"Todo homem é mortal e Sócrates é mortal, logo Sócrates é homem."*. Analisando-o com os diagramas:



O último diagrama mostra uma situação em que as premissas são verdadeiras, mas a conclusão, não – ou seja, x não é F. Desse modo, o ato de inferência é inválido, pois nem sempre que as premissas são verdadeiras, a conclusão também é.

Você pode se perguntar *mas e quando as premissas não são ou não podem ser todas verdadeiras?* Bem, nesse caso, não nos importa mais se a conclusão é verdadeira ou não, pois ela é consequência das premissas! Atos de inferência – e por consequência, argumentos, como veremos a seguir – que contêm premissas falsas ou contraditórias são ditos **válidos por vacuidade**.

EXEMPLO 0.3 . . . .
Analisemos o seguinte ato de inferência:

$$\begin{array}{rc} & \text{Todas as constelações têm exatamente 5 estrelas.} \\ & \text{O Sol é uma anã amarela.} \\ \cline{2-2} \textbf{logo,} & \text{Faz muito frio em Recife.} \end{array}$$

Obviamente, a conclusão não faz sentido dadas as premissas e é tampouco verdadeira. Porém, nem todas as constelações estelares têm 5 estrelas! Geralmente elas possuem bem mais que isso. Por exemplo, a constelação de Scorpius (Escorpião) possui já de início 13 estrelas "mais brilhantes". Assim sendo, há uma premissa falsa e, portanto, a inferência é válida por vacuidade.

Agora podemos definir argumento apropriadamente.

DEFINIÇÃO 0.1: ARGUMENTO ———————————————

Um argumento é uma coleção de atos de inferência, que por sua vez é uma sequência de sentenças que possuem uma relação entre si. Uma delas é chamada de conclusão, e as demais, as justificativas da conclusão, são chamadas de premissas.

Um argumento é válido se todos os seus atos também são – de modo que, se suas premissas são verdadeiras, a conclusão também é. ---------

0.2 CONSISTÊNCIA . . . .

Mais acima, definimos validez por vacuidade para o caso onde as premissas são falsas ou são contraditórias. Queremos expandir essa condição para qualquer conjunto de sentenças, participantes de um argumento ou não. Temos então, a noção de consistência.

DEFINIÇÃO 0.2: CONSISTÊNCIA DE UM CONJUNTO DE SENTENÇAS –

Dado um conjunto de sentenças, se existe um ato de inferência baseado em algum subconjunto desse conjunto que deduz validamente uma sentença e, com base em outro subconjunto desse conjunto, deduz validamente uma sentença contraditória a anterior, o conjunto é **inconsistente**. Caso contrário, é **consistente**. ---

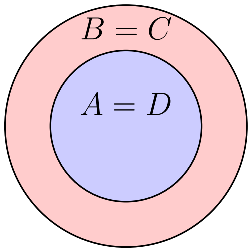
Ou seja, um conjunto de sentenças que não contém contradição entre seus elementos é consistente, de modo que todas as suas sentenças podem ser verdadeiras ao mesmo tempo. Novamente, podemos usar os diagramas de Venn quando possível para identificar consistência – ou a ausência dela.

EXEMPLO 0.4 . . . .

Analisemos o conjunto de sentenças abaixo.

Todo A é B
Todo B é C.
Todo A é D.
Algum B não é D.

Diferentemente da análise de validez de argumentos, não desenhamos diagramas para algumas sentenças e esperamos que outra sentença seja verdadeira também nessa configuração. Aqui, devemos desenhar um diagrama no qual todas as sentenças sejam verdadeiras. Se o conjunto não for consistente, um diagrama não existirá, então deve-se argumentar sua inconsistência. Neste exemplo, o conjunto é consistente, pois o diagrama abaixo ilustra uma situação em que todas as sentenças são verdadeiras ao mesmo tempo.



EXEMPLO 0.5 . . . .
Analisemos o conjunto de sentenças abaixo.

Todo F é G.
Todo G é H.
Algum F não é H.

Das duas primeiras sentenças, podemos inferir que *Todo F é H*. Essa conclusão é uma contradição clara com a terceira sentença, *Algum F não é H*. Assim, por existir contradição entre as sentenças, o conjunto é inconsistente.

EXEMPLO 0.6 . . . .
Esse exemplo trata de um paradoxo:

A sentença abaixo é verdadeira.
A sentença acima é falsa.

Paradoxos são interessantes pois tratam de contradições de sentenças com elas mesmas! Note que, se ambas as sentenças fossem verdadeiras, a primeira seria falsa pela segunda sentença e a segunda seria falsa por consequência disso. Assim, essas sentenças só são verdadeiras se e somente se são falsas, o que obviamente é uma contradição. Logo, o conjunto é inconsistente.

EXERCÍCIOS

1. Defina formalmente os seguintes conceitos:
   (a) Sentença declarativa
   (b) Contradição
   (c) Ato de inferência
   (d) Argumento
   (e) Validez de um argumento
   (f) Validez por vacuidade
   (g) Consistência de um conjunto de sentenças

2. Determine se o argumento a seguir é válido.

   Todo A é B.
   Algum B não é C.
   logo, Todo A é C.

3. Determine se o argumento a seguir é válido.

   Algum A não é B.
   Algum A não é C.
   Todo B é C.
   Algum C não é B.
   logo, Algum A é B.

4. Determine se o argumento a seguir é válido.

   Algum C é A.
   Todo C é B.
   Nenhum A é B.
   logo, Algum A não é B.

5. Determine se o argumento a seguir é válido.

   Todo A é B.
   Algum A é C.
   logo, Algum B não é C.

6. Considere um argumento A formado por um conjunto P de premissas e um conjunto unitário C de conclusões. É possível afirmar algo sobre a validez de A nas situações abaixo? Justifique.
   (a) P é consistente e $P \cup C$ é inconsistente.
   (b) P é inconsistente e $P \cup C$ é inconsistente.
   (c) P é consistente e $P \cup C$ é consistente.
7. Mostre mais 3 diagramas que confirmam que o conjunto de sentenças do exemplo 0.4 é consistente.
8. Determine se o conjunto de sentenças a seguir é consistente.

   Algum A é B.
   Todo C é B.
   Nenhum C é A

9. Determine se o conjunto de sentenças a seguir é consistente.

   Nenhum A é B.
   Algum A não é C.
   Todo B é C.
   Algum A é C.

10. Determine se o conjunto de sentenças a seguir é consistente. *(Dica: lembre da definição de subconjuntos.)*

   Se x não é C então x não é A.
   Nenhum A é B.
   Algum B é C.

11. Determine se o conjunto de sentenças a seguir é consistente.

   y é D.
   Todo B é A.
   Todo C é B.
   Nenhum D é E.
   Todo A é E.
   y é C.

12. A sentença abaixo é o dito **Paradoxo de Russell**. Explique a contradição que se forma na sentença.

"Um conjunto M é o conjunto de todos os conjuntos que não contém a si próprios."

13. O parágrafo abaixo contém o chamado **Paradoxo de Grelling-Nelson**. Identifique a contradição associada a ele.

    *"Uma palavra **autológica** é uma palavra que descreve a si mesma. Palavras que não são autológicas, ou seja, que não descrevem a si mesmas, são ditas **heterológicas**. Por exemplo: 'real' é autológica, já que a palavra 'real' é real. 'Sofisticado' também é autológica, pois é uma palavra sofisticada, bem como 'substantivo', já que a palavra é um substantivo. Em contraste, 'longo', 'monossilábico', 'adjetivo', 'verbo', 'mosca' e 'palavrão' são palavras heterológicas."*

14. Leandro se tornou o presidente da empresa BR-Lógica. Para isso ter acontecido, houve uma votação na qual os seguintes pontos foram levantados:

    1: Todos que ocuparem tal posição são honestos;

    2: Todos que ocuparem tal posição são competentes;

    É tomado como senso comum pelas pessoas que votaram em Leandro os seguintes pontos:

    3: Todos que ocuparem tal posição e são coniventes com infrações à lei são desonestos;

    4: Todos que ocuparem tal posição e são negligentes são também incompetentes;

    São considerados negligentes os presidentes que possuem desconhecimento das atividades da empresa, que não sabem nem o que se passa na sala ao lado. Posteriormente, foram reveladas graves infrações à BR-Lógica, a ponto de elas ocorrerem até na sala ao lado da de Leandro. Ele, então, fez a seguinte afirmação:

    5: "Não estava ciente das infrações."

    Determine se o conjunto de sentenças $\{1, 2, 3, 4, 5\}$ é consistente.

I

LÓGICA PROPOSICIONAL

CAPÍTULO 1 Sintaxe

A lógica de Aristóteles não era plenamente formal, pois não era apática ao conteúdo das sentenças nem ao conhecimento dos sujeitos. Ou seja, era atribuída a uma sentença um valor-verdade baseado na falsidade ou verdade do conhecimento do sujeito (por exemplo, a sentença *"Sócrates é homem"* é verdadeira se considerarmos Sócrates como o filósofo grego, mas pode ser falsa para uma pessoa cujo nome de seu cachorro seja Sócrates). Desse modo, ambiguidades, equívocos e falta de clareza eram problemas frequentemente enfrentados pelos lógicos. Em oposição à essa linha de pensamento, surgiu independentemente ao estudo tradicional da lógica e como um subramo da Matemática, a Lógica Simbólica, por George Boole (1815-1864) e Friedrich Frege (1848-1925). Essencialmente, a Lógica Simbólica trata-se de um método de representação e manipulação símbolica de sentenças. Um exemplo disso está descrito a seguir. Suponha que, em uma cidade, 3 pessoas, A, B e C são suspeitas de um crime. Estes são os depoimentos deles:

A diz: B é culpado e C é inocente.

B diz: Se A é culpado, então C também é.

C diz: Eu sou inocente, mas pelo menos um dos outros dois é culpado.

Avaliar se tal conjunto de depoimentos é consistente usando diagramas de Venn é bem complicado! Felizmente, podemos representá-los por meio de símbolos. Vamos representar as sentenças *"B é culpado"*, *"C é culpado"* e *"A é culpado"* por **variáveis**, x, y e z respectivamente. Usando álgebra booleana, podemos representar tais depoimentos dessa forma:

A diz: $x \wedge \neg y$

B diz: $z \rightarrow y$

C diz: $\neg y \wedge (x \vee z)$

Desse modo, podemos verificar a consistência de um modo que lembre o método de resolver um sistema de equações, bem mais simples. Veremos adiante que as sentenças escritas em símbolos dessa forma são chamadas de proposições e seu estudo compõe a **lógica proposicional**. Neste capítulo, veremos a sintaxe de suas expressões, ou seja, a forma delas.

1.1 ALFABETO

Quando escrevemos expressões matemáticas, geralmente usamos uma quantidade grande e variada de símbolos. Por exemplo, na expressão $(4+x) \times (3-y)$, usamos 9 símbolos diferentes. Os símbolos x e y são as chamadas **variáveis**; os símbolos $+$, $-$ e $\times$ são os **operadores**; 4 e 3 são as **constantes**, enquanto (e) são chamados de **parênteses**. A união de todas as variáveis, operadores, constantes e parênteses – ou seja, o conjunto de todos os símbolos com os quais podemos escrever expressões – forma o **alfabeto**. Na lógica, temos o seguinte alfabeto Σ:

Variáveis: $\{x, y, z, w...\}$

Operadores: $\{\neg, \wedge, \vee, \rightarrow\}$

Parênteses: $\{(,)\}$

Constantes: $\{0, 1\}$

$$\Sigma = \text{Variáveis} \cup \text{Operadores} \cup \text{Parênteses} \cup \text{Constantes}$$

Com esse alfabeto, podemos criar infinitas **palavras** ou **cadeias** diferentes, como $(x \wedge y) \vee \neg w$ e $()(xw\vee$. O conjunto de todas as palavras que podem ser formadas pelo alfabeto da lógica é denotado por $\mathbf{\Sigma^*}$ *(sigma estrela)*.

EXPRESSÕES LEGÍTIMAS

Note que Σ^* contém muitas cadeias que não fornecem nenhuma utilidade prática, pois não são **expressões legítimas** da lógica. É o caso de $()(xw\vee$. Na aritmética, temos algo semelhante com cadeias como $(+)x23*$ ou $-*2\}\{)5$. Buscamos então, definir um conjunto que reúna todas as expressões legítimas da lógica. Por ora, vamos denotá-lo por $<EXPR>$. A figura 1.1 ilustra a relação inicial entre os dois conjuntos.

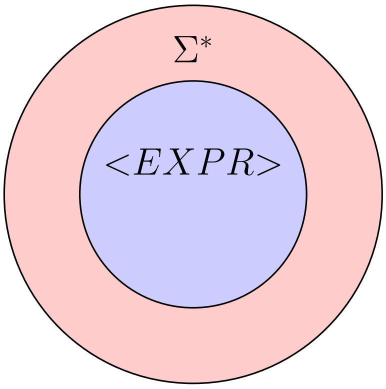


FIGURA 1.1
Diagrama ilustrando Σ^* e $<EXPR> \subseteq \Sigma^*$.

Note que as variáveis e as constantes representam, por si mesmas, expressões legítimas. Agruparemo-as em um conjunto **base**, denotado por X. Note também que podemos produzir qualquer cadeia da lógica por meio de **funções geradoras**, de acordo com os operadores. São elas:

$$
\begin{array}{ll}
f_\neg : \Sigma^* \to \Sigma^* & f_\wedge : \Sigma^* \times \Sigma^* \to \Sigma^* \\
f_\neg(\omega) = (\neg\omega) & f_\wedge(\omega_1, \omega_2) = (\omega_1 \wedge \omega_2) \\
 & \\
f_\vee : \Sigma^* \times \Sigma^* \to \Sigma^* & f_\to : \Sigma^* \times \Sigma^* \to \Sigma^* \\
f_\vee(\omega_1, \omega_2) = (\omega_1 \vee \omega_2) & f_\to(\omega_1, \omega_2) = (\omega_1 \to \omega_2)
\end{array}
$$

Agrupando as funções em um conjunto F, temos o seguinte diagrama:

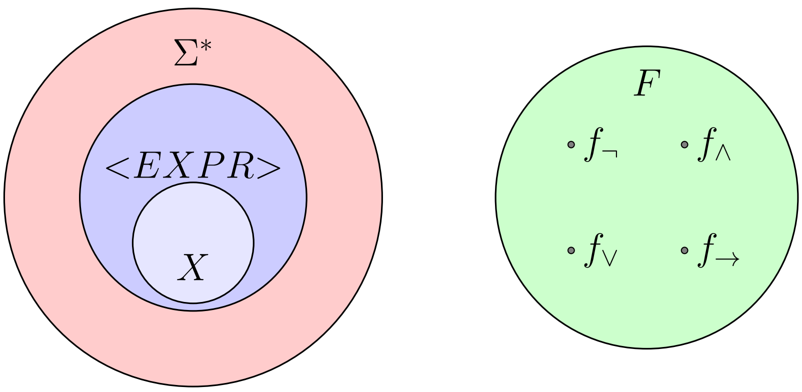


FIGURA 1.2
Linguagem da lógica proposicional.

Assim, podemos definir **indutivamente** o conjunto $<EXPR>$:

DEFINIÇÃO 1.1: CONJUNTO DAS EXPRESSÕES LEGÍTIMAS

O conjunto das expressões legítimas da lógica proposicional é o menor conjunto de cadeias sobre o alfabeto Σ tal que:

- Toda variável é uma expressão legítima;
- Toda constante é uma expressão legítima;
- Se ω for uma expressão legítima, então $(\neg\omega)$ é uma expressão legítima;
- Se ω_1 e ω_2 forem expressões legítimas, então $(\omega_1 \wedge \omega_2)$ é uma expressão legítima;
- Se ω_1 e ω_2 forem expressões legítimas, então $(\omega_1 \vee \omega_2)$ é uma expressão legítima;
- Se ω_1 e ω_2 forem expressões legítimas, então $(\omega_1 \rightarrow \omega_2)$ é uma expressão legítima;

As expressões legítimas da lógica proposicional também são chamadas **proposições**, e denotamos o conjunto das proposições por $\mathbf{PROP}$.

1.2 CONJUNTOS INDUTIVOS

Definições feitas como a do conjunto $<EXPR>$ são chamadas definições indutivas, e o conjunto é chamado **conjunto indutivo**. Sua definição formal é dada a seguir, tente associá-la com a definição do conjunto das expressões legítimas.

DEFINIÇÃO 1.2: CONJUNTO INDUTIVO

Seja A um conjunto qualquer e X um subconjunto próprio de A. Seja F um conjunto de funções sobre A, cada uma com sua aridade. Um subconjunto Y de A é dito **indutivo sobre X e F** se:

- Y contém X;
- Y é fechado sob as funções de F.

EXEMPLO 1.1

Vamos definir indutivamente o conjunto de todas as cadeias binárias que têm o formato $1^{k+1}0^k$, $k > 0$.

Tomando o conjunto desejado por A, iniciemos uma análise breve de A. Note que 110, 11100 e 1111000, por exemplo, estão em A, enquanto ϵ (a cadeia vazia), 0 e 001, por exemplo, não (mas estão no conjunto de todas as cadeias binárias, que denotamos por $\{0,1\}^*$). Por definição, precisamos definir um conjunto base e um conjunto de funções sobre $\{0,1\}^*$. Lembre que a base deve satisfazer a característica do conjunto!

$X = \{\ 110\ \}$

$F = \{\ f(-)\ \}$, onde:

$$f : \{0,1\}^* \rightarrow \{0,1\}^*$$
$$f(\omega) = 1 \cdot \omega \cdot 0$$

O símbolo $\cdot$ denota a operação de **concatenação**. Como A contém X e é fechado sob as funções de F, temos que A é um conjunto indutivo sobre X e F, como planejado. Uma análise comum de ser feita ao definir conjuntos indutivos é determinar o menor e o maior conjuntos indutivos sobre a base e o conjunto de funções. Note que o menor conjunto indutivo sobre X e F é o próprio A, pois qualquer conjunto menor que ele não será fechado sob a função f ou não conterá X; já o maior é facilmente determinado por $\{0,1\}^*$, o conjunto de todas as cadeias binárias, pois é um conjunto que contém X e é fechado sob a função f, e não há outro conjunto maior que ele.

EXEMPLO 1.2 . . . .

Vamos definir indutivamente o conjunto das cadeias palíndromes sobre o alfabeto $\{a,b\}$.

Vamos tomar o conjunto desejado por A. Uma cadeia palíndrome é uma cadeia a qual lê-la da esquerda para a direita é o mesmo que lê-la da direita para a esquerda. Por exemplo, ϵ, aba, $baab$ e $aabbaa$ são palíndromes, enquanto ab, baa e $abbaa$, não. Definindo o conjunto base e o conjunto de funções, temos:

$X = \{\ \epsilon, a, b\ \}$

$F = \{\ f(-), g(-)\ \}$, onde:

$$\begin{array}{ll} f : \{a,b\}^* \rightarrow \{a,b\}^* & g : \{a,b\}^* \rightarrow \{a,b\}^* \\ f(\omega) = a \cdot \omega \cdot a & g(\omega) = b \cdot \omega \cdot b \end{array}$$

Como A contém X e é fechado sob as funções de F, A é um conjunto indutivo sobre X e F, e é o menor. O maior conjunto indutivo sobre X e F é o conjunto de todas as cadeias que podem ser formadas pelo alfabeto $\{a, b\}$, ou seja, $\{a, b\}^*$, uma vez que tal conjunto contém X e é fechado sob as funções de F, e não há outro conjunto maior que ele.

FECHO INDUTIVO

Definições indutivas envolvendo um conjunto base X e um conjunto de funções F geralmente implicam que infinitos conjuntos serão indutivos sobre X e F. Porém, usualmente estamos interessados apenas no menor deles, pois é o que caracteriza matematicamente as propriedades de tal conjunto, chamado de **fecho indutivo de X sob F**. Assim, veremos a seguir dois métodos que nos permite obtê-lo.

Método Top-Down

Tome a interseção da família $\mathbb{F}$ de todos os subconjuntos indutivos Y de Σ^* que contém X e são fechados sob as funções de F. Como o próprio Σ^* é indutivo sobre X e F, a interseção não é vazia. O conjunto resultante é dito **menor conjunto indutivo sobre X e F**.

$$X^+ = \bigcap \mathbb{F}$$

Método Bottom-Up

Desejamos obter o menor dos conjuntos indutivos sobre X e F de uma forma mais computacional. Definimos um processo indutivo, passo a passo, para obtenção desse conjunto, dito **fecho indutivo de X sob F**.

- Seja $X_0 = X$
- $X_{i+1} = X_i \cup \{f(\omega_1, ..., \omega_k) \mid f \in F$, aridade$(f) = k, \omega_1, ..., \omega_k \in X_i\}$

E então, tome:

$$X_+ = \bigcup_{i=0}^{\infty} X_i$$

LEMA 1.3 . . . .
$X^+ = X_+$

Em outras palavras, o menor conjunto indutivo é igual ao fecho indutivo.

PROVA Como procedimento padrão de provas de igualdade de conjuntos, queremos mostrar que *(i)* $X^+ \subseteq X_+$ e que *(ii)* $X_+ \subseteq X^+$.

(i) $X^+ \subseteq X_+$
Como X^+ é a interseção de todos os conjuntos indutivos sobre X e F, basta mostrar que X_+ é indutivo sobre X e F. Assim, temos:

(a) X_+ contém X:
Por definição, X_+ contém $X_0 = X$. Assim, X_+ contém X.

(b) X_+ é fechado sob as funções de F:
Queremos mostrar que para todo $f \in F$ e para todos $\omega_1, ..., \omega_k \in X_+$, $f(\omega_1, ..., \omega_k) \in X_+$. Por definição, existe algum i de forma que $\omega_1, ..., \omega_k \in X_i$ e $f(\omega_1, ..., \omega_k) \in X_{i+1}$. Como X_i e $X_{i+1} \subseteq X_+$, X_+ é fechado sob as funções de F.

(ii) $X_+ \subseteq X^+$
Como X_+ é definido pela união de conjuntos X_i tal que $i \geqslant 0$, queremos mostrar que, para todo i, $X_i \subseteq X^+$. Faremos uma prova por indução matemática sobre i.

- Passo base $(i = 0)$: $X_0 \subseteq X^+$
  Como X^+ é indutivo e $X_0 = X$, $X_0 \subseteq X^+$.
- Passo indutivo $(i = n)$:

  Hipótese indutiva: $X_n \subseteq X^+$
  Tese: $X_{n+1} \subseteq X^+$

  Por definição, $X_{n+1} = X_n \cup \{f(\omega_1, ..., \omega_k) \mid f \in F,$ aridade$(f) = k, \omega_1, ..., \omega_k \in X_n\}$.
  Pela H.I., $X_n \subseteq X^+$. Como X^+ é indutivo, é fechado sob as funções de F e, portanto, a segunda parcela também está contida em X^+. Logo, está provada a tese.

A prova está completa.

CONJUNTO LIVREMENTE GERADO

Como dissemos anteriormente, desejamos caracterizar o conjunto de palavras sobre o alfabeto Σ que são expressões da lógica proposicional. Usando os conceitos de conjunto indutivo, somos capazes de:

- Reconhecer palavras de Σ que são proposições.
- Aplicar funções recursivas sobre o conjunto das proposições para verificar propriedades de sintaxe sobre seus elementos.
- Usar indução matemática para provar uma afirmação do tipo "todo elemento do conjunto das proposições tem a propriedade P".

Para garantir que essas afirmações sejam seguras matematicamente, é necessário que cada proposição tenha apenas uma única linha de formação, ou seja, não tenha **ambiguidade**. Desse modo, o conjunto das proposições precisa ter a propriedade de **leitura única**. Para ilustrar esse conceito, vamos definir indutivamente um conjunto similar ao das expressões aritméticas com apenas uma variável, x. Por exemplo, $(x + x)$ e $(x \times (x + x))$ pertencem ao conjunto, mas xx e $()x+$, não.

$X = \{ \ x \ \}$

$F = \{ \ f_+(-,-), f_\times(-,-), f_{()}(-) \ \}$, onde:

$$\begin{array}{lll} f_+ : \Sigma^* \times \Sigma^* \to \Sigma^* & f_\times : \Sigma^* \times \Sigma^* \to \Sigma^* & f_{()} : \Sigma^* \to \Sigma^* \\ f_+(\omega_1, \omega_2) = \omega_1 + \omega_2 & f_\times(\omega_1, \omega_2) = \omega_1 \times \omega_2 & f_{()}(\omega) = (\omega) \end{array}$$

O conjunto similar ao das expressões aritméticas com apenas a variável x será, portanto, o fecho indutivo de X sob F. Ele é problemático do ponto de vista computacional, pois se tomarmos a seguinte cadeia:

$$x + x \times x$$

Vemos que, apesar de ser uma palavra legítima do conjunto, não sabemos onde está a relação de precedência entre $+$ e $\times$. Se quiséssemos calcular o valor dessa expressão, poderíamos ter que lidar com erros devido à essa incerteza (por exemplo, se $x = 3$, a expressão poderia assumir 18 e 12 ao mesmo tempo, o que é impossível).

Na lógica proposicional, não é diferente. Veremos mais tarde como calcular o valor-verdade de uma expressão, e precisamos então garantir que todas as proposições não possam assumir dois valores-verdade diferentes. Desse modo, precisamos que $PROP$ atenda certas condições, sistematizadas na definição a seguir.

DEFINIÇÃO 1.3: CONJUNTO LIVREMENTE GERADO

Seja A um conjunto qualquer e X um subconjunto próprio de A. Seja F um conjunto de funções sobre A, cada uma com sua aridade. O fecho indutivo de X sob F é dito **livremente gerado** se:

- As funções de F são injetoras.
- Qualquer duas funções distintas de F possuem conjuntos imagem em relação a X_+ disjuntos.
- Nenhum elemento da base X está no conjunto imagem de alguma função f de F.

Logo, temos que o conjunto que definimos acima não é livremente gerado, pois as funções f_+ e $f_\times$ não possuem conjuntos imagem disjuntos: $f_+(x, f_\times(x,x))$ resulta no mesmo elemento que $f_\times(f_+(x,x),x)$. Veremos adiante mais exemplos de conjuntos que não são livremente gerados.

EXEMPLO 1.4 . . . .
Tomemos uma definição indutiva para o conjunto das cadeias binárias, incluindo a cadeia vazia.

$X = \{\ \epsilon, 0, 1\ \}$

$F = \{\ f(-), g(-)\ \}$, onde:

$$\begin{array}{ll} f : \{0,1\}^* \to \{0,1\}^* & g : \{0,1\}^* \to \{0,1\}^* \\ f(\omega) = 0 \cdot \omega & g(\omega) = 1 \cdot \omega \end{array}$$

Tal conjunto será, portanto, o fecho indutivo X_+ de X sob F. Note que ele é tanto o menor conjunto quanto o maior conjunto indutivo sobre X e F. A análise que queremos fazer é se X_+ é livremente gerado. A resposta é não, pelo fato de que $f(\epsilon) = 0$, que é um elemento da base X. Alternativamente, $g(\epsilon) = 1$ também é um elemento de X. Note que, se definirmos X sem 0 e 1, X_+ passa a ser livremente gerado, pois não haveria mais ambiguidades.

EXEMPLO 1.5 . . . .
Tomemos uma definição indutiva para o conjunto dos naturais múltiplos de 3.

$X = \{\ 0\ \}$

$F = \{ f(-), g(-) \}$, onde:

$$\begin{array}{ll} f : \mathbb{N} \to \mathbb{N} & g : \mathbb{N} \to \mathbb{N} \\ f(x) = 3x & g(x) = x + 3 \end{array}$$

Tal conjunto será, portanto, o fecho indutivo X_+ de X sob F. Diferentemente dos exemplos que vimos até agora, esse trata de números e não de cadeias sobre um alfabeto. Agora, temos a questão: X_+ é livremente gerado? Novamente, a resposta é não, pelo fato de que $f(0) = 0$, que é um elemento da base. Alternativamente, $f(g(0)) = g(g(g(0)))$, implicando que as funções de F não têm conjuntos imagem disjuntos. Se definíssemos F como um conjunto unitário contendo apenas a função g, então X_+ seria livremente gerado.

EXEMPLO 1.6 . . . .

Tomemos uma definição indutiva para o conjunto dos inteiros múltiplos de 5.

$X = \{ 0 \}$

$F = \{ f(-), g(-) \}$, onde:

$$\begin{array}{ll} f : \mathbb{Z} \to \mathbb{Z} & g : \mathbb{Z} \to \mathbb{Z} \\ f(x) = x - 5 & g(x) = |x| \end{array}$$

Tal conjunto será, portanto, o fecho indutivo X_+ de X sob F. Repetindo a análise, X_+ não é livremente gerado pois a função g não é injetora: $g(f(0)) = g(g(f(0)))$.

Nós podemos então provar que as proposições da lógica proposicional não apresentam ambiguidade, ou seja, que PROP é livremente gerado.

TEOREMA 1.7 . . . .

$PROP$ é livremente gerado.

PROVA Queremos provar que $PROP$, como fecho indutivo de X sob F, cumpre as três condições de conjuntos livremente gerados:

- As funções $f_\neg$, $f_\wedge$, $f_\vee$ e $f_\to$ são injetoras, pois quando aplicadas a elementos distintos, resultam em elementos distintos.

- Não há em F duas funções distintas cujos conjuntos imagens sob PROP tenham interseção.
- Nenhum elemento de X pertence ao conjunto imagem sob PROP de alguma função de F.

Desse modo, $PROP$ é livremente gerado.
A prova está completa.

1.3 FUNÇÕES RECURSIVAS SOBRE PROP . . . .

Uma vez que PROP é livremente gerado, podemos definir funções recursivas sobre PROP, uma vez que ele tem a propriedade de leitura única, implicando que o processo de definição recursiva de funções sobre ele é seguro matematicamente. Nessa seção, passaremos por várias funções que podem ser definidas sobre o conjunto das proposições e, então, seremos capazes de provar propriedades de proposições com elas usando indução matemática. Como notação, usaremos as letras gregas φ, ψ, ρ e θ para representar expressões da lógica proposicional.

Para toda função definida, usaremos a expressão $\phi = (x \vee (\neg y))$ para exemplificar a entrada das funções.

Número de parênteses

$f : PROP \to \mathbb{N}$
$f(\varphi) = 0$, se φ é atômica.
$f((\neg\psi)) = f(\psi) + 2$
$f((\rho \wedge \theta)) = f(\rho) + f(\theta) + 2$
$f((\rho \vee \theta)) = f(\rho) + f(\theta) + 2$
$f((\rho \to \theta)) = f(\rho) + f(\theta) + 2$

Proposições atômicas (variáveis ou constantes) não contribuem com parênteses, enquanto cada operador contribui com 2 parênteses para a expressão. Ao receber ϕ, essa função retorna 4.

Número de operadores

$g : PROP \to \mathbb{N}$
$g(\varphi) = 0$, se φ é atômica.
$g((\neg\psi)) = g(\psi) + 1$

$g((\rho \wedge \theta)) = g(\rho) + g(\theta) + 1$
$g((\rho \vee \theta)) = g(\rho) + g(\theta) + 1$
$g((\rho \rightarrow \theta)) = g(\rho) + g(\theta) + 1$

Proposições atômicas não possuem operadores, então não contribuem em nada no número de operadores, enquanto cada operador contribui obviamente com 1. Ao receber ϕ, essa função retorna 2.

Número total de símbolos

$h : PROP \rightarrow \mathbb{N}$
$h(\varphi) = 1$, se φ é atômica.
$h((\neg\psi)) = h(\psi) + 3$
$h((\rho \wedge \theta)) = h(\rho) + h(\theta) + 3$
$h((\rho \vee \theta)) = h(\rho) + h(\theta) + 3$
$h((\rho \rightarrow \theta)) = h(\rho) + h(\theta) + 3$

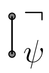
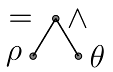
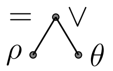
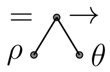
Proposições atômicas contribuem com 1 no número de símbolos, enquanto cada operador contribui com 3: os dois parênteses e o símbolo do próprio operador. Ao receber ϕ, essa função retorna 8.

Subexpressões

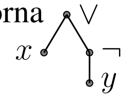
$s : PROP \rightarrow \mathcal{P}(PROP)$
$s(\varphi) = \{\varphi\}$, se φ é atômica.
$s((\neg\psi)) = s(\psi) \cup \{(\neg\psi)\}$
$s((\rho \wedge \theta)) = s(\rho) \cup s(\theta) \cup \{(\rho \wedge \theta)\}$
$s((\rho \vee \theta)) = s(\rho) \cup s(\theta) \cup \{(\rho \vee \theta)\}$
$s((\rho \rightarrow \theta)) = s(\rho) \cup s(\theta) \cup \{(\rho \rightarrow \theta)\}$

Cada subexpressão contribui com si mesma para o conjunto das subexpressões. Ao receber ϕ, essa função retorna $\{x, y, (\neg y), (x \vee (\neg y))\}$.

Árvore Sintática

$i : PROP \rightarrow GRAFO$
$i(\varphi) = \circ\,\varphi$, se φ é atômica.
$i((\neg\psi)) =$ 
$i((\rho \wedge \theta)) =$ 
$i((\rho \vee \theta)) =$ 
$i((\rho \rightarrow \theta)) =$ 

Árvore sintática de uma expressão é uma árvore cujos nós são os símbolos da expressão, exceto pelos parênteses. É possível obter a expressão novamente realizando um encaminhamento em ordem na árvore, adicionando um par de parênteses para cada nó interno. Ao receber ϕ, essa função retorna



Posto

$p : PROP \to \mathbb{N}$
$p(\varphi) = 0$, se φ é atômica.
$p((\neg\psi)) = p(\psi) + 1$
$p((\rho \wedge \theta)) = max(p(\rho), p(\theta)) + 1$
$p((\rho \vee \theta)) = max(p(\rho), p(\theta)) + 1$
$p((\rho \to \theta)) = max(p(\rho), p(\theta)) + 1$

O posto de uma expressão é equivalente à altura da árvore sintática da expressão, onde a altura de uma árvore é computada pela altura do maior ramo da árvore. Proposições atômicas só acrescentam um vértice, então não contribuem para a altura da árvore, enquanto cada operador aumenta em 1 a altura da subárvore em que se encontram (implicando que pode haver outra subárvore com altura maior). Ao receber ϕ, essa função retorna 2.

PROVAS DE PROPRIEDADES SOBRE PROP

Uma vez que definimos funções recursivas sobre PROP, podemos provar propriedades de seus elementos usando indução matemática. Percorreremos algumas provas por indução para ilustrar o processo.

LEMA 1.8 . . . .
Para toda proposição $\varphi \in PROP$, o número de parênteses de φ é par.

PROVA Queremos provar que, para toda proposição φ, $f(\varphi)$ é par. Faremos uma prova por indução sobre a complexidade de φ.

- Passo base (φ é atômica):
  Pela definição de f, $f(\varphi) = 0$ e 0 é par.

- Passo indutivo (φ é da forma $(\neg\psi)$):

  Hipótese indutiva: $f(\psi)$ é par.

  Tese: $f((\neg\psi))$ é par.

Por definição, $f((\neg\psi)) = f(\psi) + 2$. Pela hipótese, $f(\psi)$ é par. Logo, $f(\psi) + 2$ também é par. Está provada a tese.

- Passo indutivo (φ é da forma $(\rho \,\square\, \theta)$, onde $\square \in \{\wedge, \vee, \rightarrow\}$):

  Hipótese indutiva (1): $f(\rho)$ é par.

  Hipótese indutiva (2): $f(\theta)$ é par.

  Tese: $f((\rho \,\square\, \theta))$ é par.

  Por definição, $f((\rho \,\square\, \theta)) = f(\rho) + f(\theta) + 2$. Pela hipótese (1) e (2), $f(\rho)$ é par e $f(\theta)$ é par. Logo, $f(\rho) + f(\theta) + 2$ também é par. Está provada a tese.

A prova está completa.

LEMA 1.9 . . . .
Para toda proposição $\varphi \in PROP$, o posto de φ é no máximo igual ao número de operadores de φ.

PROVA Queremos provar que, para toda proposição φ, $p(\varphi) \leqslant g(\varphi)$. Faremos uma prova por indução sobre a complexidade de φ.

- Passo base (φ é atômica):
  Pela definição de p e g, $p(\varphi) = 0$ e $g(\varphi) = 0$. Temos que $0 \leqslant 0$.

- Passo indutivo (φ é da forma $(\neg\psi)$):

  Hipótese indutiva: $p(\psi) \leqslant g(\psi)$

  Tese: $p((\neg\psi)) \leqslant g((\neg\psi))$

  Por definição, $p((\neg\psi)) = p(\psi) + 1$ e $g((\neg\psi)) = g(\psi) + 1$. Desse modo, podemos desenvolver a tese para:

  $p(\psi) + 1 \leqslant g(\psi) + 1$

  Pela hipótese, $p(\psi) \leqslant g(\psi)$. Logo, $p(\psi) + 1 \leqslant g(\psi) + 1$. Está provada a tese.

- Passo indutivo (φ é da forma $(\rho \,\square\, \theta)$, onde $\square \in \{\wedge, \vee, \rightarrow\}$):

  Hipótese indutiva (1): $p(\rho) \leqslant g(\rho)$

  Hipótese indutiva (2): $p(\theta) \leqslant g(\theta)$

  Tese: $p((\rho \,\square\, \theta)) \leqslant g((\rho \,\square\, \theta))$

Por definição, $p((\rho \Box \theta)) = max(p(\rho), p(\theta)) + 1$ e $g((\rho \Box \theta)) = g(\rho) + g(\theta) + 1$. Desse modo, podemos desenvolver a tese para:

$$max(p(\rho), p(\theta)) + 1 \leqslant g(\rho) + g(\theta) + 1$$

Por propriedade matemática, $max(p(\rho), p(\theta)) \leqslant p(\rho) + p(\theta)$. Pela hipótese (1) e (2), $p(\rho) \leqslant g(\rho)$ e $p(\theta) \leqslant g(\theta)$. Logo, $p(\rho) + p(\theta) \leqslant g(\rho) + g(\theta)$ e, portanto, $p(\rho) + p(\theta) + 1 \leqslant g(\rho) + g(\theta) + 1$.

Como $max(p(\rho), p(\theta)) + 1 \leqslant p(\rho) + p(\theta) + 1$ e $p(\rho) + p(\theta)) + 1 \leqslant g(\rho) + g(\theta) + 1$, por transitividade, $max(p(\rho), p(\theta)) + 1 \leqslant g(\rho) + g(\theta) + 1$. Está provada a tese.

A prova está completa.

LEMA 1.10 . . . .
Para toda proposição $\varphi \in PROP$, o número de subexpressões de φ é no máximo igual a $2n + 1$, onde n é o número de operadores de φ.

PROVA Queremos provar que, para toda proposição φ, $|s(\varphi)| \leqslant 2g(\varphi) + 1$. Faremos uma prova por indução sobre a complexidade de φ.

- Passo base (φ é atômica):
  Pela definição de s e g, $|s(\varphi)| = |\{\varphi\}| = 1$ e $2g(\varphi) + 1 = 2 \times 0 + 1 = 1$. Temos que $1 \leqslant 1$.

- Passo indutivo (φ é da forma $(\neg\psi)$):

  Hipótese indutiva: $|s(\psi)| \leqslant 2g(\psi) + 1$

  Tese: $|s((\neg\psi))| \leqslant 2g((\neg\psi)) + 1$

  Por definição, $|s((\neg\psi))| = |s(\psi) \cup \{(\neg\psi)\}|$ e $2g((\neg\psi)) + 1 = 2(g(\psi) + 1) + 1$. Desse modo, podemos desenvolver a tese para:

  $$|s(\psi) \cup \{(\neg\psi)\}| \leqslant 2g(\psi) + 1 + 2$$

  Pelo Princípio da Inclusão-Exclusão, $|s(\psi) \cup \{(\neg\psi)\}| = |s(\psi)| + |\{(\neg\psi)\}| - |s(\psi) \cap \{(\neg\psi)\}|$. Como $s(\psi)$ e $\{(\neg\psi)\}$ são disjuntos, a igualdade se resume a $|s(\psi)| + 1$. Reescrevendo a tese:

  $$|s(\psi)| + 1 \leqslant 2g(\psi) + 1 + 2$$

  Pela hipótese, $|s(\psi)| \leqslant 2g(\psi)+1$. Logo, $|s(\psi)|+1 \leqslant 2g(\psi)+1+2$. Está provada a tese.

- Passo indutivo (φ é da forma $(\rho \,\square\, \theta)$, onde $\square \in \{\wedge, \vee, \rightarrow\}$):

  Hipótese indutiva (1): $|s(\rho)| \leqslant 2g(\rho) + 1$

  Hipótese indutiva (2): $|s(\theta)| \leqslant 2g(\theta) + 1$

  Tese: $|s((\rho \,\square\, \theta))| \leqslant 2g((\rho \,\square\, \theta)) + 1$

  Por definição, $|s((\rho \,\square\, \theta))| = |s(\rho) \cup s(\theta) \cup \{(\rho \,\square\, \theta)\}|$ e $2g((\rho \,\square\, \theta))+1 = 2(g(\rho)+g(\theta)+1)+1$. Desse modo, podemos desenvolver a tese para:

  $$|s(\rho) \cup s(\theta) \cup \{(\rho \,\square\, \theta)\}| \leqslant 2g(\rho) + 2g(\theta) + 1 + 1 + 1$$

  Pelo Princípio da Inclusão Exclusão, $|s(\rho) \cup s(\theta) \cup \{(\rho \,\square\, \theta)\}| = |s(\rho)| + |s(\theta)| + |\{(\rho \,\square\, \theta)\}| - |s(\rho) \cap s(\theta)| - |s(\rho) \cap \{(\rho \,\square\, \theta)\}| - |s(\theta) \cap \{(\rho \,\square\, \theta)\}| + |s(\rho) \cap s(\theta) \cap \{(\rho \,\square\, \theta)\}|$. As quatro interseções são vazias, pois os conjuntos são disjuntos dois a dois. Assim, a igualdade se resume a $|s(\rho)| + |s(\theta)| + 1$. Reescrevendo a tese:

  $$|s(\rho)| + |s(\theta)| + 1 \leqslant 2g(\rho) + 2g(\theta) + 1 + 1 + 1$$

  Pela hipótese (1) e (2), $|s(\rho)| \leqslant 2g(\rho) + 1$ e $|s(\theta)| \leqslant 2g(\theta) + 1$. Assim, $|s(\rho)|+|s(\theta)|+1 \leqslant 2g(\rho)+2g(\theta)+1+1+1$. Está provada a tese.

A prova está completa.

EXERCÍCIOS

1. Defina formalmente os seguintes conceitos:
   (a) Conjunto indutivo sobre X e F
   (b) Fecho indutivo de X sob F
   (c) Conjunto livremente gerado
2. Qual a importância de um conjunto como PROP ser livremente gerado?
3. Dado um conjunto base X e um conjunto de funções geradoras F, defina os dois métodos para se obter o fecho indutivo de X sob F e prove que os conjuntos que resultam de ambos os métodos são iguais.
4. Defina indutivamente os conjuntos a seguir, identificando: (i) base da indução, (ii) funções geradoras, (iii) fecho indutivo (menor conjunto indutivo) e (iv) maior conjunto indutivo.
   (a) Conjunto das cadeias binárias com formato $0^k1^{2k}, k \geqslant 0$.
   (b) Conjunto das cadeias binárias cujos números correspondentes são pares.
   (c) Conjunto das cadeias binárias de tamanho ímpar.
   (d) Conjunto das cadeias binárias de tamanho ímpar que possuem 0 como o símbolo do meio.
   (e) Conjunto das cadeias binárias de tamanho par que podem ser divididas ao meio resultando em metades idênticas.
   (f) Conjunto das cadeias sobre o alfabeto $\{a, b\}$ que começam e terminam com o mesmo símbolo.
   (g) Conjunto das cadeias sobre o alfabeto $\{[,]\}$ cujas palavras contêm colchetes apropriadamente aninhados.
   Exemplos: ϵ, [], [][[]] e [[[]]] pertencem ao conjunto, enquanto][e [[], não.
   (h) Conjunto das cadeias palíndromes sobre o alfabeto $\{a, b, c\}$.
5. Para cada item da questão anterior, determine se o fecho indutivo é um conjunto livremente gerado. Caso negativo, explicite as condições violadas.

6. O fecho indutivo sob X e F da definição a seguir não é livremente gerado. Determine quem é ele, o porquê de não ser livremente gerado e dê outra definição indutiva que gere o mesmo fecho indutivo, porém dessa vez livremente gerado.

   $X = \{\ 0, 1\ \}$
   $F = \{\ f(-), g(-)\ \}$, onde:

$$
\begin{array}{ll}
f : \mathbb{N} \to \mathbb{N} & g : \mathbb{N} \to \mathbb{N} \\
f(x) = 2x & g(x) = 2x + 2
\end{array}
$$

7. Para cada problema abaixo, defina recursivamente as funções para a formalização do problema e use indução para provar o enunciado.
   (a) Para toda proposição $\varphi \in PROP$, o número de parênteses de φ é igual ao dobro do número de operadores de φ.
   (b) Para toda proposição $\varphi \in PROP$, o número de subexpressões de φ é no máximo igual ao número de nós da árvore sintática de φ.
   (c) Para toda proposição $\varphi \in PROP$, o posto de φ é no máximo igual ao dobro do número de parênteses à esquerda de φ.
   (d) Para toda proposição $\varphi \in PROP$, o número de nós da árvore sintática de φ é no máximo igual ao número total de símbolos de φ.
   (e) Para toda proposição $\varphi \in PROP$, se φ não contém nenhuma ocorrência do símbolo $\neg$, então o número total de símbolos de φ é ímpar.
   (f) Para toda proposição $\varphi \in PROP$, o posto de φ é no máximo igual ao número de nós da árvore sintática de φ.
   (g) Para toda proposição $\varphi \in PROP$, o número total de símbolos de φ é no mínimo igual ao número de operadores de φ.
   (h) Para toda proposição $\varphi \in PROP$, o número de parênteses à esquerda de φ é igual ao número de parênteses à direita de φ.

8. Considere uma população de formigas, divididas entre formigas guardiãs e operárias. As formigas devem migrar para outro reino dispostas em filas indianas. O conjunto L de formigas contém todas as configurações de filas que obedecem as seguintes propriedades:
   - Para cada formiga operária O, é necessário duas formigas guardiãs G acompanhando ao seu lado.

- A configuração tem tamanho ímpar.

Por exemplo, GOGOG, GGG, GOG e GGOGOGG são configurações válidas, enquanto OOO, GG e OGO não são.
Dê uma definição indutiva para L, identificando a base da indução e o conjunto das funções geradoras, e determine se L é livremente gerado.

9. Considere uma sala composta por livros de cálculo e física, os quais estão dispostos nas estantes numa configuração linear e horizontal. O conjunto L de livros contém todas as configurações lineares horizontais que obedecem as seguintes propriedades:

   - Para cada livro de cálculo C, há 3 livros de física F do seu lado direito.
   - A configuração tem tamanho par.

   Assim, configurações como CFFF e FFFF são válidas, porém FFF ou CCFFFF não são.
   Dê uma definição indutiva para L, identificando a base da indução e o conjunto das funções geradoras, e determine se L é livremente gerado.

10. Uma sequência $\varphi_0, ..., \varphi_n$ é chamada uma sequência de formação de φ se $\varphi_n = \varphi$ e para toda $i \leq n$,

    φ_i é atômica, ou

    $\varphi_i = (\neg\varphi_j)$ para algum $j < i$, ou

    $\varphi_i = (\varphi_j \,\square\, \varphi_k)$ para algum $j, k < i$.

    Por exemplo, $\perp, y, z, (\perp \vee y), (\neg(\perp \vee y)), (\neg z)$ e $z, (\neg z)$ são ambas sequências de formação de $(\neg z)$. Note que sequências de formação podem conter "lixo".
    Mostre que, para toda proposição $\varphi \in PROP$, φ possui uma sequência de formação.

CAPÍTULO 2 Semântica

Vimos que, uma vez que PROP é livremente gerado, podemos definir funções recursivamente sobre PROP, pois o resultado será seguro do ponto de vista matemático. Particularmente, estamos interessados em definir uma função que calcula o **valor-verdade** de uma proposição, dado que o valor-verdade das possíveis variáveis que ocorrem nela esteja fixado. Por exemplo, ao tomar as sentenças *"Todo cometa é um corpo celeste"*, *"O Sol é um corpo celeste"* e *"O Sol é um cometa"* como variáveis x, y e z, respectivamente, desejamos calcular o valor-verdade da proposição $((x \wedge y) \rightarrow z)$, uma vez que sabemos que os valores-verdade de x, y e z são verdadeiro, verdadeiro e falso. Neste capítulo, daremos atenção ao significado das expressões da lógica proposicional.

2.1 VALORAÇÃO-VERDADE

Em lógica, uma proposição pode assumir um de dois valores-verdade: **1** ou **0** (verdadeiro ou falso, respectivamente). Tais valores são também chamados de **booleanos**, em homenagem a George Boole. Ele também definiu funções sobre o conjunto dos booleanos, as chamadas **funções booleanas**.

Função NÃO

NÃO de um valor-verdade retorna verdadeiro se, e somente se, o valor-verdade é falso.

$$\begin{array}{c|c} 0 & \mathbf{1} \\ \hline 1 & \mathbf{0} \end{array}$$

Função E

E entre dois valores-verdade retorna verdadeiro se, e somente se, ambos os valores-verdade são verdadeiros.

	0	1
0	**0**	**0**
1	**0**	**1**

Função OU

OU entre dois valores-verdade retorna verdadeiro se, e somente se, pelo menos um dos valores-verdade é verdadeiro.

	0	1
0	**0**	**1**
1	**1**	**1**

Função SE-ENTÃO

SE-ENTÃO entre dois valores-verdade retorna verdadeiro se, e somente se, o primeiro valor-verdade é falso ou o segundo é verdadeiro.

	0	1
0	**1**	**1**
1	**0**	**1**

TEOREMA DA EXTENSÃO HOMOMÓRFICA ÚNICA

Assim, podemos definir **valoração-verdade** como uma função v que atribui a cada variável um booleano ou valor-verdade. Logo, para as variáveis x, y e z definidas na introdução do capítulo, por exemplo, podemos dizer que $v(x) = 1$, $v(y) = 1$ e $v(z) = 0$.

Portanto, para extendermos a função de valoração-verdade para qualquer proposição da lógica proposicional que contenha tais variáveis, um procedimento lógico a se fazer é definir uma função que recursivamente alcança as variáveis e que faz uma correspondência entre os símbolos de operadores e as funções booleanas. Tal procedimento é descrito no teorema a seguir, nomeado **Teorema da Extensão Homomórfica Única**.

TEOREMA 2.1 . . . .
Seja A um conjunto qualquer, e X um subconjunto próprio de A. Seja F um conjunto de funções sobre A, cada uma com sua aridade. Seja B um

outro conjunto, G um conjunto de funções sobre B e seja $d : F \to G$ um mapeamento entre as funções de F e G que preserva a aridade.
Se o fecho indutivo de X sob F for livremente gerado, então para toda função $h : X \mapsto B$, existe uma única função $\hat{h} : X_+ \mapsto B$ tal que:

- $\hat{h}(\omega) = h(\omega)$, se $\omega \in X$.
- $\hat{h}(f(\omega_1, ..., \omega_k)) = g(\hat{h}(\omega_1), ..., \hat{h}(\omega_k))$, para toda função $f \in F \mid aridade(f) = k$, $g = d(f)$ e para toda k-upla $\omega_1, ..., \omega_k$ de elementos de X_+.

A figura abaixo mostra a implicação do teorema na lógica proposicional:

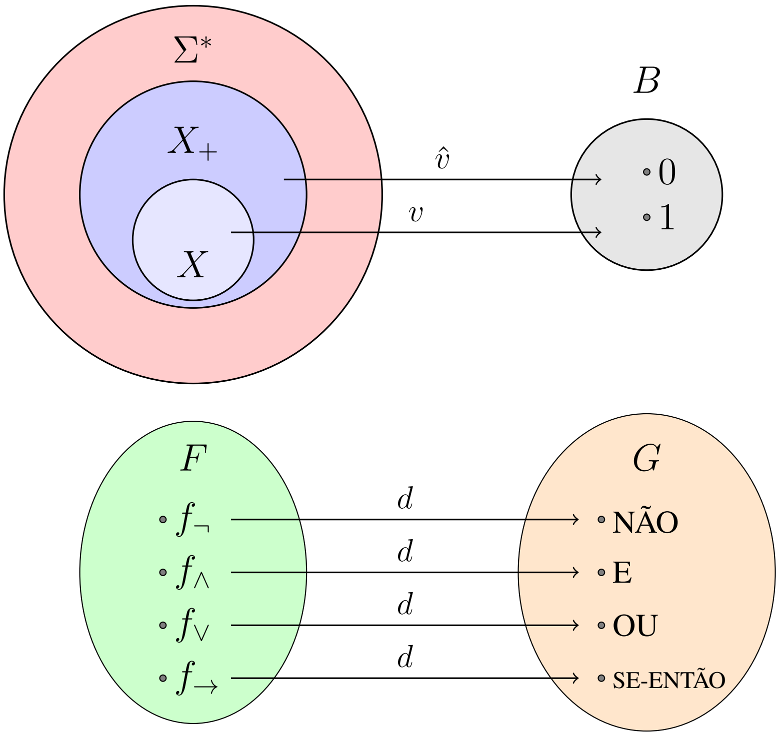


FIGURA 2.1
Implicação do teorema 2.1 na lógica proposicional

Podemos então, tomar a função de valoração-verdade como um caso especial do teorema da extensão homomórfica única:

DEFINIÇÃO 2.1: VALORAÇÃO-VERDADE
A função de **valoração-verdade** é uma função $v : X \mapsto BOOLEANOS$,

tal que sua extensão homomórfica $\hat{v} : X_+ \mapsto BOOLEANOS$ atende as condições:

- $\hat{v}(\omega) = v(\omega)$, se $\omega \in X$
- $\hat{v}(f_\neg(\omega)) = N\tilde{A}O(\hat{v}(\omega))$
- $\hat{v}(f_\wedge(\omega_1, \omega_2)) = E(\hat{v}(\omega_1), \hat{v}(\omega_2))$
- $\hat{v}(f_\vee(\omega_1, \omega_2)) = OU(\hat{v}(\omega_1), \hat{v}(\omega_2))$
- $\hat{v}(f_\rightarrow(\omega_1, \omega_2)) = SE\text{-}ENT\tilde{A}O(\hat{v}(\omega_1), \hat{v}(\omega_2))$

Uma valoração-verdade v **satisfaz** uma proposição φ se $\hat{v}(\varphi) = 1$. Caso contrário, se $\hat{v}(\varphi) = 0$, dizemos que v **refuta** φ.

EXEMPLO 2.2 . . . .

Vamos calcular o valor-verdade da proposição $((x \wedge y) \rightarrow z)$, dado que $v(x) = 1$, $v(y) = 1$ e $v(z) = 0$.

Aplicando a função de valoração-verdade à proposição:

$$\begin{aligned}
&\hat{v}(((x \wedge y) \rightarrow z)) \\
&= SE\text{-}ENT\tilde{A}O(\hat{v}((x \wedge y)), \hat{v}(z)) \\
&= SE\text{-}ENT\tilde{A}O(E(\hat{v}(x), \hat{v}(y)), v(z)) \\
&= SE\text{-}ENT\tilde{A}O(E(v(x), v(y)), 0) \\
&= SE\text{-}ENT\tilde{A}O(E(1, 1), 0) \\
&= SE\text{-}ENT\tilde{A}O(1, 0) = 0
\end{aligned}$$

Temos então, que a valoração $1, 1, 0$ refuta a proposição. Note que esta é a mesma definida no exemplo do início do capítulo. Assim, o resultado foi esperado, pois o Sol e os cometas serem corpos celestes não implica no Sol ser também um cometa.

EXEMPLO 2.3 . . . .

Vamos calcular o valor-verdade da proposição $((\neg z) \wedge (x \vee y))$, dado que $v(x) = 0$, $v(y) = 1$ e $v(z) = 0$.

Aplicando a função de valoração-verdade à proposição:

$\hat{v}(((\neg z) \wedge (x \vee y)))$
$= E(\hat{v}((\neg z)), \hat{v}((x \vee y)))$
$= E(N\tilde{A}O(\hat{v}(z)), OU(\hat{v}(x), \hat{v}(y)))$
$= E(N\tilde{A}O(v(z)), OU(v(x), v(y)))$
$= E(N\tilde{A}O(0), OU(0, 1))$
$= E(1, 1) = 1$

Temos então, que a valoração $0, 1, 0$ satisfaz a proposição.

SATISFATIBILIDADE

Alguns conceitos importantes surgem quando tratamos de valoração-verdade em proposições:

Seja φ uma proposição:

- φ é dita **satisfatível** se existe uma valoração-verdade v que a satisfaz ($\hat{v}(\varphi) = 1$).
- φ é dita **refutável** se existe uma valoração-verdade v que a refuta ($\hat{v}(\varphi) = 0$).
- φ é dita **tautologia** se toda valoração-verdade a satisfaz.
- φ é dita **insatisfatível** se toda valoração-verdade a refuta.

Seja ψ uma proposição:

- φ e ψ são ditas **logicamente equivalentes** ou equivalentes se, para toda valoração-verdade v, $\hat{v}(\varphi) = \hat{v}(\psi)$.

Seja Γ um conjunto de proposições:

- Γ é dito **satisfatível** ou consistente se existe uma valoração-verdade que satisfaz todas as proposições de Γ.
- Γ é dito **insatisfatível** ou inconsistente se não existe uma valoração-verdade que satisfaz todas as proposições de Γ.
- φ é dita **consequência lógica** de Γ ($\Gamma \vDash \varphi$) se toda valoração-verdade que satisfaz Γ também satisfaz φ.

TEOREMA 2.4 . . . .
Seja φ uma proposição e Γ um conjunto de proposições. φ é consequência lógica de Γ se, e somente se, $\Gamma \cup \{(\neg\varphi)\}$ é insatisfatível.

PROVA Como procedimento padrão de provas "se, e somente se", devemos provar a ida e a volta.

IDA: Se φ é consequência lógica de Γ, então $\Gamma\cup\{(\neg\varphi)\}$ é insatisfatível.
Faremos uma prova direta.

1. Suponha que φ é consequência lógica de Γ.
2. Por (1), toda valoração-verdade que satisfaz Γ também satisfaz φ.

   Seja v uma valoração-verdade que satisfaz Γ.
3. Por (2), $\hat{v}(\varphi) = 1$, e, portanto, $\hat{v}((\neg\varphi)) = 0$.
4. Por (3), temos que toda valoração-verdade que satisfaz Γ refuta $(\neg\varphi)$.
5. Por (4), não há valoração-verdade que satisfaz todas as proposições de $\Gamma \cup \{(\neg\varphi)\}$.
6. Por (5), $\Gamma \cup \{(\neg\varphi)\}$ é insatisfatível.

VOLTA: Se $\Gamma \cup \{(\neg\varphi)\}$ é insatisfatível, então φ é consequência lógica de Γ.
Faremos uma prova direta.

1. Suponha que $\Gamma \cup \{(\neg\varphi)\}$ é insatisfatível.
2. Por (1), temos dois casos: Γ é um conjunto insatisfatível ou Γ é um conjunto satisfatível, mas $\Gamma \cup \{(\neg\varphi)\}$ não. Queremos mostrar que, para ambos os casos, $\Gamma \vDash \varphi$.
   - Suponha que Γ é insatisfatível.
     1. Temos que não há valoração-verdade que satisfaz Γ.
     2. Assim, por vacuidade, temos que φ é consequência lógica de Γ.
   - Suponha que Γ é satisfatível, mas $\Gamma \cup \{(\neg\varphi)\}$ não é.
     1. Temos que há pelo menos uma valoração-verdade que satisfaz Γ.
     2. Temos que não há valoração-verdade que satisfaz $\Gamma \cup \{(\neg\varphi)\}$.

3. Por (1) e (2), toda valoração-verdade que satisfaz Γ refuta $(\neg\varphi)$.
4. Por (3), toda valoração-verdade que satisfaz Γ também satisfaz φ.
5. Por (4), φ é consequência lógica de Γ.

3. Assim, como ambos os casos resultam em $\Gamma \vDash \varphi$, temos que φ é consequência lógica de Γ.

A prova está completa.

TEOREMA 2.5 . . . .
Seja φ uma proposição. φ é satisfatível se, e somente se, $(\neg\varphi)$ é refutável.

PROVA Iremos provar a ida e a volta.

IDA: Se φ é satisfatível, então $(\neg\varphi)$ é refutável.
Faremos uma prova direta.

1. Suponha que φ é satisfatível.
2. Por (1), existe uma valoração-verdade v tal que $\hat{v}(\varphi) = 1$.
3. Por (2), $\hat{v}((\neg\varphi)) = 0$.
4. Por (3), como existe uma valoração-verdade que refuta $(\neg\varphi)$, temos que $(\neg\varphi)$ é refutável.

VOLTA: Se $(\neg\varphi)$ é refutável, então φ é satisfatível.
Faremos uma prova direta.

1. Suponha que $(\neg\varphi)$ é refutável.
2. Por (1), existe uma valoração-verdade v tal que $\hat{v}((\neg\varphi)) = 0$.
3. Por (2), $\hat{v}(\varphi) = 1$.
4. Por (3), como existe uma valoração-verdade que satisfaz φ, temos que φ é satisfatível.

A prova está completa.

TEOREMA 2.6 . . . .
Seja φ uma proposição. φ é tautologia se, e somente se, $\varnothing \vDash \varphi$.

PROVA Iremos provar a ida e a volta.

IDA: Se φ é tautologia, então $\varnothing \vDash \varphi$.
Faremos uma prova direta.

1. Suponha que φ é tautologia.
2. Por (1), todas as valorações-verdade satisfazem φ.
3. Por (2), φ é consequência lógica de qualquer conjunto de proposições, incluindo o conjunto vazio, pois todas as valorações-verdade que satisfazem tal conjunto sempre satisfazem φ.
4. Por (3), $\varnothing \vDash \varphi$.

VOLTA: Se $\varnothing \vDash \varphi$, então φ é tautologia.
Faremos uma prova direta.

1. Suponha que $\varnothing \vDash \varphi$ é tautologia.
2. Por (1), todas as valorações-verdade que satisfazem $\varnothing$ também satisfazem φ.
3. Como $\varnothing$ não possui nenhuma proposição, por vacuidade, temos que todas as valorações-verdade satisfazem $\varnothing$.
4. Por (2) e (3), todas as valorações-verdade satisfazem φ.
5. Por (4), φ é tautologia.

A prova está completa.

2.2 CONJUNTO FUNCIONALMENTE COMPLETO .

Para ganhar desempenho computacional, muitas vezes desejamos escrever proposições em um conjunto menor de símbolos de conectivos que o usual da lógica, $\{\neg, \vee, \wedge, \rightarrow\}$. Para isso ser possível, é necessário que o conjunto possa gerar expressões da lógica proposicional que sejam equivalentes àquelas geradas pelo conjunto usual. Conjuntos de conectivos que atendem a essa condição são chamados **funcionalmente completos**.

Daqui em diante, afrouxaremos um pouco a colocação de parênteses em proposições, para melhorar a clareza. A seguir, mostraremos dois teoremas para ilustrar o conceito. Devemos usar equivalências lógicas, como $(\varphi \rightarrow \psi) \equiv (\neg\varphi \vee \psi)$, $(\varphi \wedge \psi) \equiv \neg(\neg\varphi \vee \neg\psi)$ e $(\varphi \vee \psi) \equiv \neg(\neg\varphi \wedge \neg\psi)$ (as Leis de DeMorgan), para a demonstração de problemas desse tipo.

TEOREMA 2.7 . . . .

O conjunto $\{\vee, \neg\}$ é funcionalmente completo.

PROVA Queremos mostrar que há uma equivalência de qualquer expressão da lógica proposicional com proposições escritas apenas com $\neg$ e $\vee$. Temos:

$$
\begin{aligned}
&\varphi \equiv \varphi \\
&(\neg\psi) \equiv (\neg\psi) \\
&(\rho \wedge \theta) \equiv \neg(\neg\rho \vee \neg\theta) \\
&(\rho \vee \theta) \equiv (\rho \vee \theta) \\
&(\rho \rightarrow \theta) \equiv (\neg\rho \vee \theta)
\end{aligned}
$$

A prova está completa.

TEOREMA 2.8 . . . .
O conjunto $\{\wedge, \neg\}$ é funcionalmente completo.

PROVA Queremos mostrar que há uma equivalência de qualquer expressão da lógica proposicional com proposições escritas apenas com $\neg$ e $\wedge$. Temos:

$$
\begin{aligned}
&\varphi \equiv \varphi \\
&(\neg\psi) \equiv (\neg\psi) \\
&(\rho \wedge \theta) \equiv (\rho \wedge \theta) \\
&(\rho \vee \theta) \equiv \neg(\neg\rho \wedge \neg\theta) \\
&(\rho \rightarrow \theta) \equiv \neg(\rho \wedge \neg\theta)
\end{aligned}
$$

A prova está completa.

EXERCÍCIOS

1. Enuncie o Teorema da Extensão Homomórfica Única.
2. Defina valoração-verdade e explique de que forma o Teorema da Extensão Homomórfica Única se aplica nessa definição.
3. Defina conjunto funcionalmente completo.
4. Determine se as seguintes afirmações são verdadeiras ou falsas.
   (a) Dada uma proposição ψ, é impossível que ψ seja satisfatível e não seja refutável.
   (b) Dada uma proposição ψ, se ψ não for satisfatível então ψ pode ser refutável.
   (c) Dada uma proposição ψ, é possível que ψ seja satisfatível e não-refutável.
   (d) Dada uma proposição ψ e um conjunto de proposições Γ, se $\Gamma \vDash \neg\psi$ então $\Gamma \cup \{\psi\}$ é insatisfatível.
   (e) Dada uma proposição ψ e um conjunto de proposições Γ, se $\Gamma \vDash \psi$ então $\Gamma \cup \{\psi\}$ é satisfatível.
   (f) Dado um conjunto de proposições Γ, se Γ é satisfatível, então todo subconjunto finito de Γ também é satisfatível.
5. Prove que, dado um conjunto de proposições Γ e duas proposições φ e ψ,
$$\Gamma \vDash (\varphi \to \psi) \text{ se, e somente se, } \Gamma \cup \{\varphi\} \vDash \psi$$
6. Prove que, dadas duas proposições φ e ψ,
$$\{\varphi\} \vDash \psi \text{ se, e somente se, } \{\neg\psi\} \vDash \neg\varphi$$
7. Prove que, dadas três proposições φ, ψ e θ,
$$\{\varphi, \neg\psi\} \vDash \theta \text{ e } \{\varphi, \psi\} \vDash \theta \text{ se, e somente se, } \{\varphi\} \vDash \theta$$
8. Prove o Teorema 2.6 usando o Teorema 2.4.
9. Mostre que o conjunto de conectivos $\{\neg, \to\}$ é funcionalmente completo.
10. Argumente o porquê do conjunto de conectivos $\{\wedge, \vee\}$ não ser funcionalmente completo.

11. A função booleana NAND é descrita na tabela a seguir. O símbolo do operador lógico correspondente à função NAND é $\uparrow$. Mostre que $\{\uparrow\}$ é um conjunto funcionalmente completo.

	0	1
0	**1**	**1**
1	**1**	**0**

Dica: encontre a proposição equivalente à $\varphi \uparrow \psi$.

12. A função booleana NOR é descrita na tabela a seguir. O símbolo do operador lógico correspondente à função NOR é $\downarrow$. Mostre que $\{\downarrow\}$ é um conjunto funcionalmente completo.

	0	1
0	**1**	**0**
1	**0**	**0**

Dica: encontre a proposição equivalente à $\varphi \downarrow \psi$.

13. Podemos ver a relação $\vDash (\varphi \to \psi)$ (ou seja, $\varphi \to \psi$ é tautologia) como uma espécie de ordenação. Dizemos que φ é menor que ψ (notação $\varphi < \psi$) se $\vDash (\varphi \to \psi)$ mas $\nvDash (\psi \to \varphi)$.

   (a) Para cada φ e ψ tal que $\varphi < \psi$, encontre σ tal que $\varphi < \sigma < \psi$. *Dica: forme uma proposição σ a partir de φ e ψ tal que $\varphi < \sigma$ e $\sigma < \psi$.*

   (b) Dê um exemplo de uma sequência $\varphi_1, \varphi_2, \varphi_3, ...$ tal que $\varphi_1 < \varphi_2 < \varphi_3 < ...$

CAPÍTULO 3 | Problema da Satisfatibilidade

"Dada uma proposição φ, pergunta-se: φ é satisfatível?". Esse é o enunciado do **problema da satisfatibilidade**, também conhecido como SAT. Ele é um dos problemas mais importantes em computação e foi o primeiro a ser classificado como NP-Completo, por Steven Cook e Leonid Levin, onde ele pertence a classe dos problemas os quais, dada a solução pronta, pode-se verificá-la em tempo polinomial, e, nessa classe, ele é um dos mais difíceis problemas. Ainda não se conhece algoritmos que resolvem SAT rapidamente (e maioria dos pesquisadores crêem que tal algoritmo não existe), mas heurísticas que resolvem o problema atualmente conseguem resolver instâncias de SAT consistindo de milhões de símbolos. Nesse capítulo, estudaremos cinco métodos que resolvem SAT, começando pelo algoritmo de força-bruta que serviu de base para os seguintes: o método da tabela-verdade.

3.1 MÉTODO DA TABELA-VERDADE

O primeiro método algorítmico que resolve SAT foi definido em 1921 pelo filósofo austríaco Ludwig Wittgenstein. Ele consiste em construir uma tabela de valores-verdade, onde suas linhas representam valorações-verdade, e suas colunas, as subexpressões da proposição a qual queremos avaliar sua satisfatibilidade. Se, na última coluna, o valor 1 aparecer, temos que a proposição é satisfatível.

EXEMPLO 3.1

$(y \to \neg z) \to (x \lor z)$ *é satisfatível?*

Vamos construir uma tabela-verdade para $(y \to \neg z) \to (x \lor z)$. Dispomos as suas subexpressões em colunas, e todas as valorações-verdade possíveis para as variáveis nas linhas.

x	y	z	$\neg z$	$y \to \neg z$	$x \lor z$	$(y \to \neg z) \to (x \lor z)$
0	0	0	1	1	0	0
0	0	1	0	1	1	1
0	1	0	1	1	0	0
0	1	1	0	0	1	1
1	0	0	1	1	1	1
1	0	1	0	1	1	1
1	1	0	1	1	1	1
1	1	1	0	0	1	1

Como o valor 1 apareceu na última coluna, temos que $(y \to \neg z) \to (x \lor z)$ é satisfatível.

EXEMPLO 3.2 . . . .
$(x \land z) \to ((x \land z) \lor \neg z)$ é tautologia?

Vamos construir uma tabela-verdade para $(x \land z) \to ((x \land z) \lor \neg z)$. Dessa vez, esperamos que todos os valores da última coluna sejam 1.

x	z	$\neg z$	$x \land z$	$(x \land z) \lor \neg z$	$(x \land z) \to ((x \land z) \lor \neg z)$
0	0	1	0	1	1
0	1	0	0	0	1
1	0	1	0	1	1
1	1	0	1	1	1

Assim, como todos os valores-verdade da última coluna são 1 (implicando que todas as valorações-verdade satisfazem a proposição), temos que $(x \land z) \to ((x \land z) \lor \neg z)$ é tautologia.

EXEMPLO 3.3 . . . .
$\{x \to z, \neg z \land (x \lor y)\} \vDash y \land \neg z$?

Construiremos uma tabela-verdade para as três proposições. Esperamos que, sempre que as proposições do conjunto tenham valor-verdade 1, $y \land \neg z$ também tenha.

x	y	z	$\neg z$	$x \vee y$	$x \to z$	$\neg z \wedge (x \vee y)$	$y \wedge \neg z$
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	0
0	1	0	1	1	**1**	**1**	**1**
0	1	1	0	1	1	0	0
1	0	0	1	1	0	1	0
1	0	1	0	1	1	0	0
1	1	0	1	1	0	1	1
1	1	1	0	1	1	0	0

Vemos que, em todas as vezes que $x \to z$ e $\neg z \wedge (x \vee y)$ são verdadeiras, $y \wedge \neg z$ também é. Assim, $y \wedge \neg z$ é consequência lógica de $\{x \to z, \neg z \wedge (x \vee y)\}$.

EXEMPLO 3.4 . . . .

$\{x, x \vee y, x \vee z\} \vDash x \wedge (y \vee z)$?

Construiremos uma tabela-verdade para as quatro proposições. Esperamos que, sempre que as proposições do conjunto tenham valor-verdade 1, $x \wedge (y \vee z)$ também tenha.

x	y	z	$y \vee z$	$(x \vee y)$	$(x \vee z)$	$x \wedge (y \vee z)$
0	0	0	0	0	0	0
0	0	1	1	0	1	0
0	1	0	1	1	0	0
0	1	1	1	1	1	0
1	0	0	0	**1**	**1**	**0**
1	0	1	1	**1**	**1**	**1**
1	1	0	1	**1**	**1**	**1**
1	1	1	1	**1**	**1**	**1**

Vemos que há um caso onde $x \wedge (y \vee z)$ é falsa, mesmo onde x, $x \vee y$ e $x \vee z$ são verdadeiras. Desse modo, $x \wedge (y \vee z)$ não é consequência lógica de $\{x, x \vee y, x \vee z\}$.

COMPLEXIDADE COMPUTACIONAL

O método da Tabela-Verdade é bem poderoso no sentido que, como ele verifica todas as valorações-verdade possíveis para as variáveis de uma dada proposição, a tabela nos fornece todas as informações possíveis sobre a satisfatibilidade de qualquer expressão. Veremos que outros métodos não agem dessa maneira, dando respostas "limitadas" para entradas também "limitadas". Contudo, tamanha expressividade traz um custo computacional bastante caro: a tabela-verdade de uma proposição com n variáveis e k operadores terá 2^n linhas e, no pior caso, $2k + 1$ colunas (provamos isso no Lema 1.10). Assim, o método tem complexidade $\mathcal{O}(2^n \cdot 2k + 1) = \mathcal{O}(2^n)$, ou seja, exponencial. Custos de tempo exponenciais são altos demais, tornando o método inviável na prática. Outros métodos, apesar de menos expressivos, são mais eficientes para certas entradas. Note que, se nos foi dada uma proposição e uma certa valoração-verdade dessa proposição, se quisermos **verificar** se ela satisfaz a proposição, o custo é reduzido para $2k + 1$, pois só precisamos montar uma linha da tabela.

CORRETUDE E COMPLETUDE

Apesar de inviável, a expressividade do método da Tabela-Verdade permitiu a ele ser a base para todos os métodos seguintes. Os métodos que resolvem SAT devem atender a duas condições:

Corretude: a resposta que o método fornece deve ser a mesma que o método da Tabela-Verdade fornece.

Completude: o método não deve rodar indefinidamente, ou seja, ele deve eventualmente parar e dar uma resposta.

Métodos criados que não atendem a alguma dessas condições não são considerados confiáveis.

3.2 MÉTODO DOS TABLEAUX ANALÍTICOS . . . . .

Na década de 1950, os lógicos Evert Beth e Jaakko Hintikka, holandês e finlandês respectivamente, buscaram independentemente uma formulação intuitiva do conceito de valoração-verdade e encontraram uma noção filosófica de **mundo possível**. Buscando estabelecer uma ponte entre valoração-verdade e mundo possível, formularam um método algorítmico que resolve SAT, que se baseia numa **árvore de possibilidades** a partir de um conjunto de regras simples, conhecido como o método dos tableaux. Com ausência de contradição, o mundo possível seria revelado por um caminho da raiz a uma folha. Mais tarde, em 1970, a fim de melhorar o desempenho do método, o lógico americano Raymond Smullyan definiu uma metodologia para a aplicação de suas regras. A partir de então, o método ficou conhecido como **método dos tableaux analíticos**.

As regras do tableau podem ser dispostas sistematicamente como a seguir. Regras do tipo α não bifurcam a árvore, enquanto regras do tipo β sim.

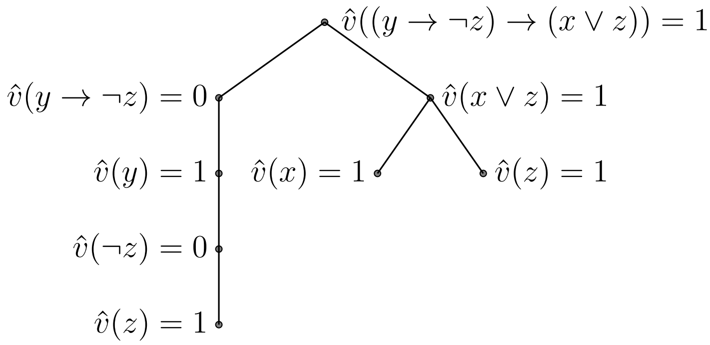
α	β
$\begin{array}{l}\hat{v}(\neg\varphi)=1\\ \hat{v}(\varphi)=0\end{array}$ $\begin{array}{l}\hat{v}(\neg\varphi)=0\\ \hat{v}(\varphi)=1\end{array}$	
$\begin{array}{l}\hat{v}(\varphi\wedge\psi)=1\\ \hat{v}(\varphi)=1\\ \hat{v}(\psi)=1\end{array}$	$\begin{array}{ccc} & \hat{v}(\varphi\wedge\psi)=0 & \\ \hat{v}(\varphi)=0 & & \hat{v}(\psi)=0\end{array}$
$\begin{array}{l}\hat{v}(\varphi\vee\psi)=0\\ \hat{v}(\varphi)=0\\ \hat{v}(\psi)=0\end{array}$	$\begin{array}{ccc} & \hat{v}(\varphi\vee\psi)=1 & \\ \hat{v}(\varphi)=1 & & \hat{v}(\psi)=1\end{array}$
$\begin{array}{l}\hat{v}(\varphi\to\psi)=0\\ \hat{v}(\varphi)=1\\ \hat{v}(\psi)=0\end{array}$	$\begin{array}{ccc} & \hat{v}(\varphi\to\psi)=1 & \\ \hat{v}(\varphi)=0 & & \hat{v}(\psi)=1\end{array}$

Pense em tais regras de forma lógica. Por exemplo, para a proposição $(\varphi \wedge \psi)$ ser verdadeira, é necessário que ambas φ e ψ sejam verdadeiras, logo, elas seguem no mesmo ramo. Já para a proposição $(\varphi \vee \psi)$ ser verdadeira, só é necessário que apenas uma delas, φ ou ψ, seja. Assim, a árvore bifurca, dispondo um ramo para cada possibilidade. Mais detalhes do método serão dados nos exemplos a seguir.

EXEMPLO 3.5 . . . .

$(y \to \neg z) \to (x \vee z)$ *é satisfatível?*

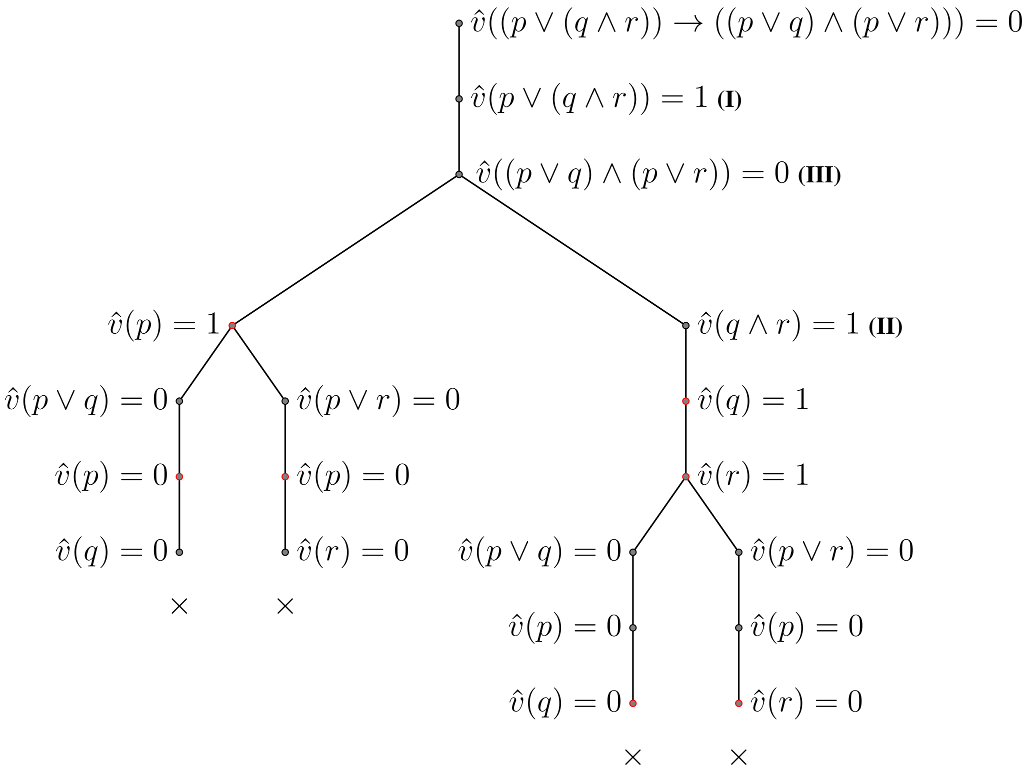
Em outras palavras, *existe um mundo possível no qual* $(y \to \neg z) \to (x \vee z)$ *é verdadeira?* Construímos a árvore de possibilidades para a proposição perguntando ao método se é possível a proposição ser verdadeira. Ou seja, iniciamos com o nó $\hat{v}((y \to \neg z) \to (x \vee z)) = 1$ e, a partir dele, aplicamos as regras dos tableaux até não haver mais nós que podemos aplicar regras.



A árvore resultante indica os mundos possíveis. Estes, caso existam, são determinados por caminhos da raiz às folhas. Desse modo, a árvore nos mostra que os mundos possíveis existem quando y e z são verdadeiras ou x é verdadeira ou z é verdadeira, independentemente dos valores-verdade das outras variáveis. Dessa forma, como existe um mundo possível onde $(y \to \neg z) \to (x \vee z)$ é verdadeira, temos que ela é satisfatível.

EXEMPLO 3.6 . . . .
$(p \vee (q \wedge r)) \to ((p \vee q) \wedge (p \vee r))$ *é tautologia?*

Pense em como vamos formular a árvore inicial. Se você acha que começamos a árvore com $\hat{v}((p \vee (q \wedge r)) \to ((p \vee q) \wedge (p \vee r))) = 1$, bem, **está errado**. Caso fizermos isso, estamos perguntando ao tableau se há um mundo possível no qual a proposição é verdadeira. Caso ela seja de fato tautologia, é óbvio que haverá! O que realmente queremos saber é se ela é verdadeira em todos os mundos possíveis. Porém, Tableaux Analíticos não é um método exaustivo, ou seja, ele não verifica todas as possibilidades. Assim, podemos contornar esse problema com uma manobra lógica, perguntando ao tableau se é possível a proposição ser falsa, e esperamos que o método negue. Assim, nossa pergunta será: *existe um mundo possível no qual* $(p \vee (q \wedge r)) \to ((p \vee q) \wedge (p \vee r))$ *é falsa?*

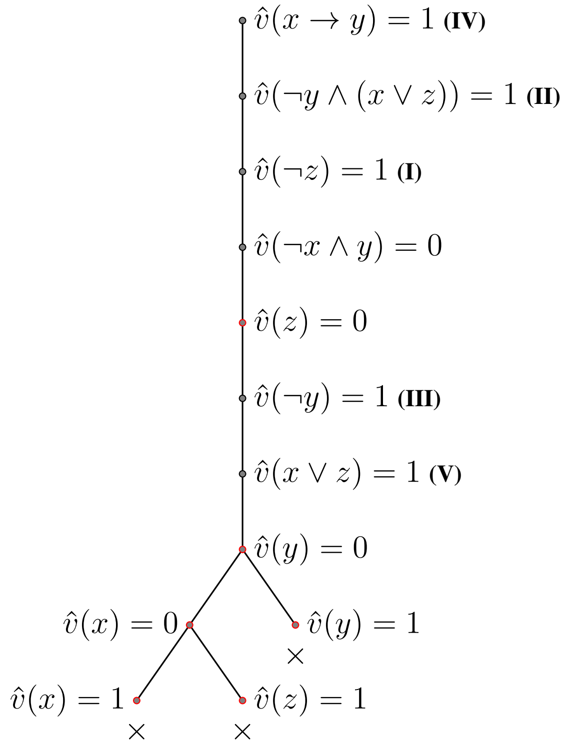


A escolha de aplicação das regras não é determinística, ou seja, não há uma sequência certa de regras para serem aplicadas. Por exemplo, poderíamos ter expandido o nó (III) antes do nó (I) ou ter expandido o nó (III) antes do nó (II). Contudo, uma boa estratégia para a aplicação das regras é **preferir as do tipo** α, pois a árvore formada é menor.

Dizemos que um ramo está **fechado** (simbolizado por $\times$) quando há uma contradição no ramo, ou seja, nós $\hat{v}(\varphi) = 1$ e $\hat{v}(\varphi) = 0$ ocorrem, onde φ é uma proposição atômica. Assim, note que ocorre uma contradição em todos os ramos do tableau. Quando isso acontece, dizemos que o tableau está **fechado** (e é dito **aberto** quando não está fechado). Tableaux fechados simbolizam inexistência de mundos possíveis. Assim, como não há um mundo possível no qual $(p \vee (q \wedge r)) \to ((p \vee q) \wedge (p \vee r))$ é falsa, temos que ela é sempre verdadeira. Assim, ela é tautologia.

EXEMPLO 3.7 . . . .
$\{x \to y, \neg y \wedge (x \vee z), \neg z\} \vDash \neg x \wedge y$?

Queremos saber se, sempre que as proposições do conjunto forem verdadeiras, $\neg x \wedge y$ também seja. Porém, de forma similar ao exemplo anterior, não conseguimos avaliar todas as situações com o método dos Tableaux. Desse modo, iremos perguntar ao tableau se é possível as proposições do conjunto serem verdadeiras, mas $\neg x \wedge y$ ser falsa, e novamente esperamos que o método negue. Assim, nossa pergunta será: *existe um mundo possível no qual $x \rightarrow y$, $\neg y \wedge (x \vee z)$ e $\neg z$ são verdadeiras e $\neg x \wedge y$ é falsa?*

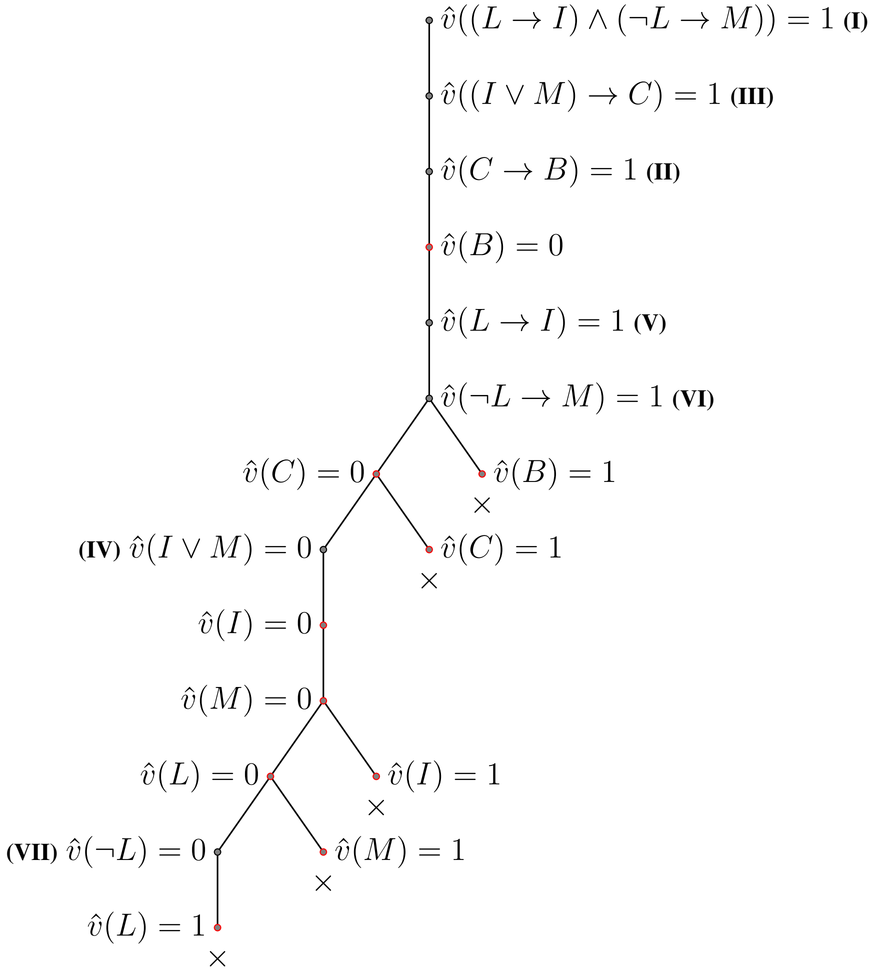


A sequência de aplicação das regras está explícita no lado direito dos nós. Como todos os ramos estão fechados, o tableau está fechado, implicando que não há mundo possível onde as proposições do conjunto são verdadeiras e a de fora é falsa. Logo, $(\neg x \wedge y)$ é consequência lógica de $\{(x \rightarrow y), (\neg y \wedge (x \vee z)), (\neg z)\}$.

Note que, uma vez encontrada uma contradição em um ramo, podemos encerrar a computação nele, visto que ele será inevitavelmente fechado. Assim, algo curioso a se notar nessa árvore é que não precisamos expandir o nó $\hat{v}(\neg x \wedge y) = 0$. Isso significa que as proposições do conjunto não podem ser todas verdadeiras! Por vacuidade, qualquer proposição seria consequência lógica desse conjunto.

EXEMPLO 3.8 . . . .
$\{(L \to I) \wedge (\neg L \to M), (I \vee M) \to C, C \to B\} \vDash B$?

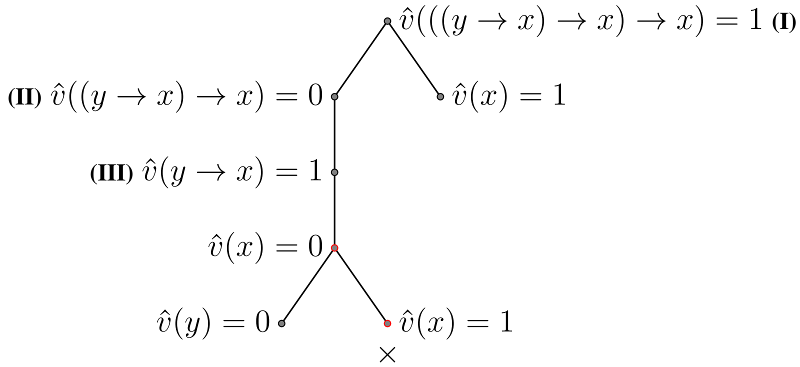
Vamos perguntar se *existe um mundo possível no qual* $(L \to I) \wedge (\neg L \to M)$, $(I \vee M) \to C$ *e* $C \to B$ *são verdadeiras e* B *é falsa.* Construiremos uma árvore de possibilidades para o problema.



Temos que o tableau está fechado, logo, não há mundo possível no qual as proposições do conjunto são verdadeiras, mas a de fora é falsa. Portanto, B é consequência lógica de $\{(L \to I) \wedge (\neg L \to M), (I \vee M) \to C, C \to B\}$.

EXEMPLO 3.9 . . . .
$(((y \to x) \to x) \to x)$ *é insatisfatível?*

Vamos perguntar se *existe um mundo possível no qual* $(((y \to x) \to x) \to x)$ *é verdadeira.* Construiremos uma árvore de possibilidades para o problema.



Temos que há pelo menos um mundo possível no qual a fórmula é verdadeira (mais especificamente, quando x e y são falsas ou quando x é verdadeira). Logo, $(((y \to x) \to x) \to x)$ não é insatisfatível. Observe que podemos confirmar que o método não explora todas as possibilidades ao notar que a valoração $1, 1$ também satisfaz a proposição, mas ela não consta na árvore como mundo possível.

COMPLEXIDADE COMPUTACIONAL

As grandes vantagens do método dos Tableaux Analíticos são ser intuitivo e mais eficiente que a tabela-verdade que alguns casos. Porém, não há realmente uma maneira de determinar esses casos. Para muitas outras entradas, o número de operações do Tableaux é grande demais, equiparável ao da Tabela-Verdade, tornando ineficiente. Por exemplo, analisando a instância:

$$(((x \wedge y) \vee (x \wedge \neg y)) \vee ((\neg x \wedge y) \vee (\neg x \wedge \neg y))) \text{ \textit{é tautologia?}}$$

Notamos que a proposição é semanticamente simples pois qualquer valoração fará alguma das disjunções ser verdadeira, logo, ela é tautologia. Porém, a solução do método dos Tableaux é grande e demorada demais. Para uma entrada genérica, supondo – no pior caso – que toda operação faz a árvore bifurcar, o método terá que computar 2^n operações, pois o número de nós em cada nível é multiplicado por 2. Assim, seu custo computacional é o mesmo da Tabela-Verdade: $\mathcal{O}(2^n)$.

3.3 MÉTODO DA RESOLUÇÃO ....

Em 1965, na década seguinte à formulação do método dos tableaux analíticos, o matemático americano John Alan Robinson publicou um artigo descrevendo um método algorítmico que resolve SAT com 2 características: **é eficiente para certas entradas** e **é eficiente em reconhecê-las**. A ideia básica do método consiste na noção de que uma fórmula conjuntiva (composta por conjunções) é insatisfatível se alguma de suas subfórmulas for insatisfatível. Assim, o método só receberia entradas nesse formato.

Uma proposição é dita **literal** se for atômica ou a negação de uma atômica. São exemplos de literais x, $\neg y$ e z. Disjunções de literais, como $(x \vee \neg y)$, z e $(w \vee z)$, são ditas **cláusulas**. Finalmente, proposições que são formadas por conjunções de cláusulas, como $(x \vee \neg y) \wedge (w \vee z)$, são ditas estarem na **forma normal conjuntiva** (FNC).

TEOREMA 3.10 ....
Para toda $\varphi \in PROP$, existe $\varphi' \in PROP$ tal que:

- φ é logicamente equivalente a φ';
- φ' está na FNC.

PROVA Em outras palavras, queremos provar que é possível transformar qualquer proposição para outra equivalente e que esteja na FNC. Faremos uma prova por indução.

- Passo base (φ é atômica):
  φ é equivalente a ela mesma e φ está na FNC. Assim, podemos tomar $\varphi' = \varphi$.

- Passo indutivo (φ é da forma $(\neg\psi)$):

  Hipótese indutiva: existe ψ' tal que $\psi' \equiv \psi$ e ψ' está na FNC.

  Tese: existe $(\neg\psi)'$ tal que $(\neg\psi)' \equiv (\neg\psi)$ e $(\neg\psi)'$ está na FNC.

  Pela hipótese, podemos tomar $\psi' \equiv C_1 \wedge C_2 \wedge ... \wedge C_k$, onde $C_i = (L_{i1} \vee L_{i2} \vee ... \vee L_{im_i})$.
  Assim, $(\neg\psi) \equiv \neg(C_1 \wedge C_2 \wedge ... \wedge C_k)$.
  Por aplicações sucessivas da Lei de De Morgan,
  $(\neg\psi) \equiv \neg C_1 \vee \neg C_2 \vee ... \vee \neg C_k$
  Logo, $(\neg\psi) \equiv (\neg L_{11} \wedge ... \wedge \neg L_{1m_1}) \vee ... \vee (\neg L_{m1} \wedge ... \wedge \neg L_{km_k})$.

Por aplicações sucessivas da Distributividade:
$(\neg\psi)' \equiv (\neg L_{11} \vee \neg L_{m1}) \wedge ... \wedge (\neg L_{11} \vee \neg L_{km_k}) \wedge ... \wedge (\neg L_{1m_1} \vee \neg L_{m1}) \wedge ... \wedge (\neg L_{1m_1} \vee \neg L_{km_k})$
Temos que $(\neg\psi)'$ é logicamente equivalente a $(\neg\psi)$ e $(\neg\psi)'$ está na FNC. Assim, está provada a tese.

- Passo indutivo (φ é da forma $(\rho \wedge \theta)$):

  Hipótese indutiva (1): existe ρ' tal que $\rho' \equiv \rho$ e ρ' está na FNC.

  Hipótese indutiva (2): existe θ' tal que $\theta' \equiv \theta$ e θ' está na FNC.

  Tese: existe $(\rho \wedge \theta)'$ tal que $(\rho \wedge \theta)' \equiv (\rho \wedge \theta)$ e $(\rho \wedge \theta)'$ está na FNC.

Pelas hipóteses (1) e (2), temos que $(\rho' \wedge \theta')$ é logicamente equivalente a $(\rho \wedge \theta)$ e está na FNC. Logo, podemos tomar $(\rho \wedge \theta)' \equiv (\rho' \wedge \theta')$. Assim, está provada a tese.

- Passo indutivo (φ é da forma $(\rho \vee \theta)$):

  Hipótese indutiva (1): existe ρ' tal que $\rho' \equiv \rho$ e ρ' está na FNC.

  Hipótese indutiva (2): existe θ' tal que $\theta' \equiv \theta$ e θ' está na FNC.

  Tese: existe $(\rho \vee \theta)'$ tal que $(\rho \vee \theta)' \equiv (\rho \vee \theta)$ e $(\rho \vee \theta)'$ está na FNC.

Pelas hipóteses (1) e (2), temos que $\rho' \equiv C_1 \wedge C_2 \wedge ... \wedge C_k$ e $\theta' \equiv C'_1 \wedge C'_2 \wedge ... \wedge C'_{k'}$.
Assim, $(\rho \vee \theta) \equiv (C_1 \wedge C_2 \wedge ... \wedge C_k) \vee (C'_1 \wedge C'_2 \wedge ... \wedge C'_{k'})$

Por aplicações sucessivas da Distributividade:
$(\rho \vee \theta)' \equiv (C_1 \vee C'_1) \wedge (C_1 \vee C'_2) \wedge ... \wedge (C_1 \vee C'_{k'}) \wedge (C_2 \vee C'_1) \wedge (C_2 \vee C'_2) \wedge ... \wedge (C_k \vee C'_{k'})$
Temos que $(\rho \vee \theta)'$ é logicamente equivalente a $(\rho \vee \theta)$ e $(\rho \vee \theta)'$ está na FNC. Assim, está provada a tese.

- Passo indutivo (φ é da forma $(\rho \rightarrow \theta)$):

  Hipótese indutiva (1): existe ρ' tal que $\rho' \equiv \rho$ e ρ' está na FNC.

  Hipótese indutiva (2): existe θ' tal que $\theta' \equiv \theta$ e θ' está na FNC.

  Tese: existe $(\rho \rightarrow \theta)'$ tal que $(\rho \rightarrow \theta)' \equiv (\rho \rightarrow \theta)$ e $(\rho \rightarrow \theta)'$ está na FNC.

Pelas hipóteses (1) e (2), temos que $\rho' \equiv C_1 \wedge C_2 \wedge ... \wedge C_k$ e $\theta' \equiv C'_1 \wedge C'_2 \wedge ... \wedge C'_{k'}$.
Assim, $(\rho \to \theta) \equiv (\neg\rho \vee \theta) \equiv (\neg C_1 \vee \neg C_2 \vee ... \vee \neg C_k) \vee (C'_1 \wedge C'_2 \wedge ... \wedge C'_{k'})$

Por aplicações sucessivas da Distributividade:
$(\rho \to \theta)' \equiv (\neg C_1 \vee \neg C_2 \vee ... \vee \neg C_k \vee C'_1) \wedge ... \wedge (\neg C_1 \vee \neg C_2 \vee ... \vee \neg C_k \vee C'_{k'})$

Cada $\neg C_i \equiv (\neg L_{i1} \wedge ... \wedge \neg L_{im_i})$. Assim:
$(\rho \to \theta)' \equiv ((\neg L_{i1} \wedge ... \wedge \neg L_{km_k}) \vee ... \vee C'_i) \wedge ... \wedge ((\neg L_{i1} \wedge ... \wedge \neg L_{im_k}) \vee ... \vee C'_{k'})$
Por aplicações sucessivas da Distributividade:
$(\rho \to \theta)' \equiv (\neg L_{i1} \vee C'_i) \wedge ... \wedge (\neg L_{km_k} \vee C'_i) \wedge ... \wedge (\neg L_{i_1} \vee C'_{k'}) \wedge ... \wedge (\neg L_{km_k} \vee C'_{k'})$
Temos que $(\rho \to \theta)'$ é logicamente equivalente a $(\rho \to \theta)$ e $(\rho \to \theta)'$ está na FNC. Assim, está provada a tese.

A prova está completa.

REGRA DA RESOLUÇÃO

O método consiste no seguinte princípio:

$$(L_1 \vee L_2) \wedge (\neg L_1 \vee L_3) \equiv (L_1 \vee L_2) \wedge (\neg L_1 \vee L_3) \wedge (L_2 \vee L_3)$$

Para quaisquer cláusulas C_1 e C_2, se existir um literal L_1 em C_1 que é **complementar** a um literal L_2 em C_2, construa uma disjunção com os literais das cláusulas, exceto L_1 e L_2. A cláusula construída é chamada de **resolvente** de C_1 e C_2. Se temos duas cláusulas unitárias, então o resolvente delas, se existir, é a **cláusula vazia** $()$. Dessa forma, se um conjunto de cláusulas é insatisfatível, podemos usar a regra da resolução para gerar a cláusula vazia a partir do conjunto.

EXEMPLO 3.11 . . . .
$(A \vee B \vee \neg C) \wedge (C \vee \neg D) \wedge (\neg B) \wedge (B \vee D) \wedge (\neg A)$ é insatisfatível?

Queremos saber se é possível encontrar a cláusula vazia a partir das cláusulas da proposição, que já se encontra na FNC. Vamos rotular as cláusulas e então usar a regra da resolução para propagá-las.

$$(A \vee B \vee \neg C)^{C_1} \wedge (C \vee \neg D)^{C_2} \wedge (\neg B)^{C_3} \wedge (B \vee D)^{C_4} \wedge (\neg A)^{C_5}$$

C_1 e C_3: $(A \lor \neg C)^{C_6}$

C_2 e C_6: $(A \lor \neg D)^{C_7}$

C_4 e C_7: $(A \lor B)^{C_8}$

C_5 e C_8: $(B)^{C_9}$

C_3 e C_9: $()$

A cláusula vazia foi encontrada. Assim, em face de uma contradição, temos que $(A \lor B \lor \neg C) \land (C \lor \neg D) \land (\neg B) \land (B \lor D) \land (\neg A)$ é insatisfatível.

Desse exemplo, é válido fazer duas observações: assim como os Tableaux, a escolha para a aplicação da regra não é determinística. Poderíamos ter propagado C_1 com C_2 ou C_2 com C_4, por exemplo. Além disso, nada impede a propagação de uma cláusula com outra que participou de sua criação. Por exemplo, C_7 e C_5. De modo geral, desde que a regra da resolução seja atendida, **é possível propagar qualquer cláusula com qualquer cláusula**.

EXEMPLO 3.12 . . . .
$\neg(A \lor (B \land C)) \lor E \lor A \lor \neg(\neg C \lor \neg D) \lor \neg(A \lor D \lor E)$ *é tautologia?*

Como vimos, o método da Resolução age tentando provar a existência da cláusula vazia. Assim, **suas entradas são sempre do tipo "φ *é insatisfatível?*"**. Desse modo, usaremos uma equivalência:

$$\varphi \text{ é tautologia se, e somente se, } \neg\varphi \text{ é insatisfatível}$$

Portanto, nossa pergunta será:
$\neg(\neg(A \lor (B \land C)) \lor E \lor A \lor \neg(\neg C \lor \neg D) \lor \neg(A \lor D \lor E))$ *é insatisfatível?*

Finalmente, transformaremos nossa proposição para uma equivalente na FNC, rotularemos as suas cláusulas e então iremos propagá-las.

$$\neg(\neg(A \lor (B \land C)) \lor E \lor A \lor \neg(\neg C \lor \neg D) \lor \neg(A \lor D \lor E))$$
$$\equiv (A \lor (B \land C)) \land \neg E \land \neg A \land (\neg C \lor \neg D) \land (A \lor D \lor E)$$
$$\equiv (A \lor B) \land (A \lor C) \land \neg E \land \neg A \land (\neg C \lor \neg D) \land (A \lor D \lor E) \textbf{ (FNC)}$$

Assim, temos:

$$(A \lor B)^{C_1} \land (A \lor C)^{C_2} \land \neg E^{C_3} \land \neg A^{C_4} \land (\neg C \lor \neg D)^{C_5} \land (A \lor D \lor E)^{C_6}$$

C_2 e C_4: $(C)^{C_7}$

C_5 e C_7: $(\neg D)^{C_8}$

C_4 e C_6: $(D \vee E)^{C_9}$

C_8 e C_9: $(E)^{C_{10}}$

C_3 e C_{10}: $()$

A cláusula vazia foi encontrada. Logo, diante de uma contradição, a proposição é insatisfatível, logo, sua negação, $\neg(A \vee (B \wedge C)) \vee E \vee A \vee \neg(\neg C \vee \neg D) \vee \neg(A \vee D \vee E)$, é tautologia.

Podemos usar certas estratégias ao propagar cláusulas. Por exemplo, para cada cláusula, podemos varrer o conjunto de cláusulas em busca de literais complementares e propagar sempre que possível. Obviamente essa não é uma estratégia eficiente do ponto de vista computacional. Porém outra – a que usamos – é tomar pares de cláusulas de modo que sempre uma delas é unitária. Cláusulas unitárias são os únicos tipos de cláusula que permitem outra “diminuir de tamanho”, permitindo assim um meio eficiente de encontrar a cláusula vazia. Tal estratégia chama-se **propagação da cláusula unitária**.

Um planejamento prévio também é importante. Nesse exemplo, estávamos em busca da cláusula (E) para encontrar uma contradição com $(\neg E)$. O literal E encontra-se apenas na cláusula C_6, portanto, as propagações foram feitas de forma a obter E em C_6 (implicando que precisamos “remover” os literais A e D da cláusula).

EXEMPLO 3.13 . . . .
$\{(C \to D), \neg(B \vee D), A \to (B \vee C)\} \vDash \neg A$*?*

De modo a tornar a instância aceitável pelo método da Resolução, usaremos a equivalência (Teorema 2.4):

$$\Gamma \vDash \varphi \text{ se, e somente se, } \Gamma \cup \{(\neg\varphi)\} \text{ é insatisfatível}$$

Portanto, nossa pergunta será:
$\{(C \to D), \neg(B \vee D), A \to (B \vee C), A\}$ *é insatisfatível?*

Tomaremos conjunções de todas as proposições do conjunto, transformaremos a proposição resultante para a FNC, rotularemos as suas cláusulas e iremos propagá-las.

$(C \to D) \land \neg(B \lor D) \land (A \to (B \lor C)) \land A$

$\equiv (\neg C \lor D) \land \neg B \land \neg D \land (\neg A \lor B \lor C) \land A$ **(FNC)**

Assim, temos:

$$(\neg C \lor D)^{C_1} \land \neg B^{C_2} \land \neg D^{C_3} \land (\neg A \lor B \lor C)^{C_4} \land A^{C_5}$$

C_1 e C_3: $(\neg C)^{C_6}$

C_2 e C_4: $(\neg A \lor C)^{C_7}$

C_6 e C_7: $(\neg A)^{C_8}$

C_5 e C_8: $()$

A cláusula vazia foi encontrada. Logo, diante de uma contradição, temos que $\{(C \to D), \neg(B \lor D), A \to (B \lor C), A\}$ é insatisfatível, e, portanto, $\neg A$ é consequência lógica de $\{(C \to D), \neg(B \lor D), A \to (B \lor C)\}$.

EXEMPLO 3.14 . . . .
$(y \to \neg z) \to (x \lor z)$ *é satisfatível?*

De modo a tornar a instância aceitável pelo método da Resolução, usaremos a equivalência:

φ é satisfatível se, e somente se, φ não é insatisfatível

Portanto, nossa pergunta será:
$(y \to \neg z) \to (x \lor z)$ *é insatisfatível?* Dessa vez, esperamos que o método negue.

Transformaremos a proposição para a FNC, rotularemos as suas cláusulas e iremos propagá-las.

$(y \to \neg z) \to (x \lor z)$

$\equiv \neg(y \to \neg z) \lor (x \lor z)$

$\equiv \neg(\neg y \lor \neg z) \lor (x \lor z)$

$\equiv (y \land z) \lor (x \lor z)$

$\equiv (y \lor x \lor z) \land (z \lor x \lor z)$

$\equiv (y \lor x \lor z) \land (z \lor x)$ **(FNC)**

Assim, temos:

$$(y \vee x \vee z)^{C_1} \wedge (z \vee x)^{C_2}$$

Note que não podemos aplicar a regra da resolução em C_1 e C_2, as únicas cláusulas do conjunto, pois não há literais complementares. Desse modo, não conseguimos encontrar a cláusula vazia a partir desse conjunto de cláusulas e, portanto, a proposição não é insatisfatível. Logo, $(y \rightarrow \neg z) \rightarrow (x \vee z)$ é satisfatível.

EXEMPLO 3.15 . . . .
$\{(L \wedge M) \rightarrow P, I \rightarrow \neg P, M, I\} \vDash \neg L$?

Nossa pergunta será:
$\{(L \wedge M) \rightarrow P, I \rightarrow \neg P, M, I, L\}$ é insatisfatível?

Tomaremos conjunções de todas as proposições do conjunto, transformaremos a proposição resultante para a FNC, rotularemos as suas cláusulas e iremos propagá-las.

$$((L \wedge M) \rightarrow P) \wedge (I \rightarrow \neg P) \wedge M \wedge I \wedge L$$

$$\equiv (\neg(L \wedge M) \vee P) \wedge (\neg I \vee \neg P) \wedge M \wedge I \wedge L$$

$$\equiv (\neg L \vee \neg M \vee P) \wedge (\neg I \vee \neg P) \wedge M \wedge I \wedge L \textbf{ (FNC)}$$

Assim, temos:

$$(\neg L \vee \neg M \vee P)^{C_1} \wedge (\neg I \vee \neg P)^{C_2} \wedge M^{C_3} \wedge I^{C_4} \wedge L^{C_5}$$

C_1 e C_5: $(\neg M \vee P)^{C_6}$

C_3 e C_6: $(P)^{C_7}$

C_2 e C_7: $(\neg I)^{C_8}$

C_4 e C_8: $()$

A cláusula vazia foi encontrada. Logo, diante de uma contradição, temos que $\{(L \wedge M) \rightarrow P, I \rightarrow \neg P, M, I, L\}$ é insatisfatível e, portanto, $\neg L$ é consequência lógica de $\{(L \wedge M) \rightarrow P, I \rightarrow \neg P, M, I\}$.

COMPLEXIDADE COMPUTACIONAL

Foi provado que o custo de tempo do método da Resolução é da ordem de $\mathcal{O}(2^n)$, o mesmo do da Tabela-Verdade. Porém, o método tem um diferencial bastante importante.

Nos anos 1950, o matemático também americano Alfred Horn estudou intensamente as propriedades de um determinado tipo de expressões da lógica proposicional que tinham um certo apelo à representação de "condições implicam em uma consequência":

$$(x_1 \wedge x_2 \wedge ... \wedge x_n) \rightarrow y$$

$$\equiv \neg(x_1 \wedge x_2 \wedge ... \wedge x_n) \vee y$$

$$\equiv (\neg x_1 \vee \neg x_2 \vee ... \vee \neg x_n \vee y)$$

Cláusulas nesse formato, que **possuem no máximo um literal positivo**, são chamadas **cláusulas de Horn**. Quando o método da Resolução recebe como entrada uma proposição na FNC cujas cláusulas são todas de Horn, é provado que seu custo computacional é $\mathcal{O}(n)$, ou seja, linear. Por esse motivo, cláusulas de Horn são a base para a programação lógica.

3.4 MÉTODO DA DEDUÇÃO NATURAL . . . .

Em meados de 1934, o matemático alemão Gerhard Gentzen buscou definir um método de formalização dos procedimentos dedutivos utilizados "naturalmente" pelos matemáticos em suas demonstrações. Daí, se propôs a encontrar um conjunto de regras simples de deduções que correspondesse ao conjunto de passos dedutivos em provas matemáticas. Ao invés de partir do conceito de valoração-verdade, seu método partiu da noção de **regra de dedução**, que independe de uma interpretação em valores booleanos. Seu método ficou conhecido como **método da dedução natural**.

Provas em dedução natural são provas a partir de **hipóteses** ou **premissas**. Em outras palavras, em qualquer prova, existe um conjunto finito de hipóteses e uma conclusão, e a **árvore de dedução** mostra que a conclusão segue das hipóteses.

REGRAS DE DEDUÇÃO

Para cada operador lógico, Gentzen definiu duas regras de dedução, de dois tipos.

Regras de introdução

São as regras de dedução que determinam as condições mínimas para que se possa deduzir uma proposição cujo operador principal fosse o em questão. Proposições em colchetes, como $[A]$, simbolizam premissas.

$$\frac{A \quad B}{A \wedge B} \qquad \Bigg| \qquad \frac{A}{A \vee B} \qquad \frac{B}{A \vee B} \qquad \Bigg| \qquad \cfrac{\begin{matrix}[A] \\ \ldots \\ B\end{matrix}}{A \to B}$$

Novamente, pense nas regras de forma lógica. Para provas do tipo $P \to Q$, por exemplo, como a ida e a volta da prova dos teoremas 2.4 a 2.6, é necessário supor P e tentar deduzir Q a partir disso. Caso esse procedimento seja bem sucedido, é possível deduzir $P \to Q$.

Regras de eliminação

São as regras de dedução que determinam as consequências imediatas que poderiam ser obtidas a partir de uma proposição cujo operador principal fosse o em questão.

$$\frac{A \wedge B}{A} \qquad \frac{A \wedge B}{B} \qquad \Bigg| \qquad \cfrac{A \vee B \qquad \begin{matrix}[A] & & [B] \\ \ldots & & \ldots \\ \phi & & \phi\end{matrix}}{\phi} \qquad \Bigg| \qquad \frac{A \quad A \to B}{B}$$

A regra de eliminação da disjunção, apesar de parecer confusa, é idêntica à construção de uma prova por casos, como a vista na volta da prova do teorema 2.4, onde nós supomos ambos os casos separadamente e tentamos chegar à uma mesma conclusão de ambos.

DEDUTIBILIDADE

Em contrapartida aos métodos anteriores, a pergunta que o método da dedução natural procura responder é *“φ é dedutível?”*.

Uma sentença φ é **dedutível** a partir de um conjunto de sentenças Γ (notação: $\Gamma \vdash \varphi$) se existe uma árvore de dedução cuja raiz é φ, cujos vértices são aplicações das regras de dedução e cujas folhas são sentenças de Γ ou suposições adicionais devidamente descartadas.

Pode-se estabelecer uma equivalência entre consequência lógica e dedutibilidade, apesar de serem conceitos diferentes: $\Gamma \vdash \varphi \leftrightarrow \Gamma \vDash \varphi$.

EXEMPLO 3.16 . . . .
$\{A, A \to B\} \vdash A \land B$?

Em outras palavras, queremos saber se há uma árvore de dedução cuja raiz é $A \land B$, cujos vértices são aplicações das regras de dedução e cujas folhas são A, $A \to B$ e qualquer premissa adicional que possa vir a surgir. Temos:

$$\dfrac{\dfrac{A \qquad A \to B}{B}\,E\!\to}{A \land B}\,I\land$$

Uma resolução por dedução natural sempre deve suceder um plano de prova. Queremos provar que, a partir das sentenças do conjunto, é possível provar $A \land B$. Para isso, é necessário que tanto A quanto B sejam verdadeiros, ou seja, sejam vértices na árvore. Uma vez que A é verdadeiro (pois supomos que todas as sentenças do conjunto são verdadeiras durante a prova), precisamos apenas provar que B também é, e esse é o nosso *objetivo*.

Assim, segue que podemos usar a regra de eliminação da implicação ($E\!\to$) com $A \to B$ para deduzir B. Como A e B são verdadeiras, podemos usar a regra de introdução da conjunção ($I\land$) para deduzir $A \land B$. Como existe uma árvore de prova em que $A \land B$ é sua raiz, seus vértices são aplicações das regras de dedução e A e $A \to B$ são suas folhas, temos que $A \land B$ é dedutível a partir de $\{A, A \to B\}$.

EXEMPLO 3.17 . . . .
$\{P \to Q, Q \to R\} \vdash P \to (Q \land R)$?

Em outras palavras, queremos saber se há uma árvore de dedução cuja raiz é $P \to (Q \land R)$, cujos vértices são aplicações das regras de dedução e cujas folhas são $P \to Q$, $Q \to R$ e qualquer premissa adicional que possa vir a surgir. Temos:

$$\dfrac{\dfrac{\dfrac{\dfrac{[P]^1 \qquad P \to Q}{Q}\,E\!\to \qquad Q \to R}{R}\,E\!\to}{Q \land R}\,I\land}{P \to (Q \land R)}\,I\!\to (1)$$

Para provarmos $P \to (Q \wedge R)$, seguiremos um procedimento idêntico ao de uma prova direta: suporemos P e, a partir dessa suposição, tentaremos deduzir $(Q \wedge R)$. Esse é o nosso primeiro passo e o que gerou o vértice $[P]^1$, que é uma premissa adicional. Segue então que, para deduzir $(Q \wedge R)$, precisamos de ambos Q e R como verdadeiros.

Assim, uma vez que supomos que P é verdadeira, podemos usar a eliminação da implicação em $P \to Q$ para deduzir Q e, com esta, podemos usar novamente a eliminação da implicação em $Q \to R$ para deduzir R. Como ambas Q e R são verdadeiras, podemos deduzir $(Q \wedge R)$. Segue então que, como a partir da suposição de P alcançamos $(Q \wedge R)$, podemos usar a introdução da implicação para deduzir $P \to (Q \wedge R)$. Nesse momento, há um **descarte** da premissa P, simbolizado por (1). Em uma árvore de prova, **todas as premissas adicionais devem ser devidamente descartadas**. É possível fazer tal descarte com 3 regras de dedução: a introdução da implicação, a eliminação da disjunção e outra que veremos mais tarde.

Logo, como existe uma árvore de dedução em que $P \to (Q \wedge R)$ é sua raiz, seus vértices são aplicações das regras de dedução e $P \to Q$, $Q \to R$ e $[P]$ são suas folhas, temos que $P \to (Q \wedge R)$ é dedutível a partir de $\{P \to Q, Q \to R\}$.

EXEMPLO 3.18 . . . .

$\{P \to (Q \to R)\} \vDash Q \to (P \to R)$?

Como o método da dedução natural age tentando provar a existência de árvores de dedução, devemos associar as instâncias do problema da satisfatibilidade à noção de dedutibilidade. Nesse caso, temos uma equivalência direta:

$$\Gamma \vDash \varphi \text{ se, e somente se, } \Gamma \vdash \varphi$$

O que nos permite agir como se nossa instância fosse "$\{P \to (Q \to R)\} \vdash Q \to (P \to R)?$". Assim, queremos saber se existe uma árvore de dedução cuja raiz é $Q \to (P \to R)$, cujos vértices são aplicações das regras de dedução e cujas folhas são $P \to (Q \to R)$ e qualquer premissa adicional que possa vir a surgir. Temos:

$$\dfrac{\dfrac{\dfrac{[Q]^1 \qquad \dfrac{[P]^2 \qquad P \to (Q \to R)}{Q \to R}\,E\!\to}{R}\,E\!\to}{P \to R}\,I\!\to (2)}{Q \to (P \to R)}\,I\!\to (1)$$

Para provarmos $Q \to (P \to R)$, precisamos supor Q e então tentar deduzir $P \to R$. Nesse momento, podemos usar o mesmo raciocínio para o consequente: para sermos capazes de deduzir $P \to R$, precisamos supor P e a partir disso tentar deduzir R. Assim, é aceitável pensar que deduziremos R antes de $P \to R$, pois este precisa daquele. Essa sequência de suposições geram as premissas $[Q]$ e $[P]$, respectivamente.

Para deduzirmos R, podemos usar a premissa P e a eliminação da implicação em $P \to (Q \to R)$ para obtermos $Q \to R$ e então novamente aplicamos a regra em $Q \to R$ para obtermos R, uma vez que Q também é uma premissa. Como usamos P na obtenção de R, podemos usar a introdução da implicação para deduzir $P \to R$, descartando P. Finalmente, como utilizamos Q na obtenção de $P \to R$, podemos usar novamente a introdução da implicação para deduzir $Q \to (P \to R)$, descartando Q.

Assim, como existe uma árvore de dedução em que $Q \to (P \to R)$ é sua raiz, seus vértices são aplicações das regras de dedução e $P \to (Q \to R)$, $[P]$ e $[Q]$ são suas folhas, temos que $Q \to (P \to R)$ é dedutível a partir de $\{P \to (Q \to R)\}$.

EXEMPLO 3.19 . . . .
$(A \to B) \to ((C \vee A) \to (C \vee B))$ *é tautologia?*

Usaremos as equivalências $\Gamma \vDash \varphi \leftrightarrow \Gamma \vdash \varphi$ e (teorema 2.6)

$$\varphi \text{ é tautologia se, e somente se, } \vDash \varphi$$

para então transformar a instância do problema em outra aceitável pelo método da dedução natural. Nossa pergunta será:
$\vdash (A \to B) \to ((C \vee A) \to (C \vee B))$*?* Dessa vez, esperamos obter uma árvore de dedução cujas folhas serão, no máximo, premissas adicionais. Temos:

$$\dfrac{\dfrac{[C \vee A]^2 \qquad \dfrac{[C]^3}{C \vee B}\,I\vee \qquad \dfrac{\dfrac{[A]^4 \qquad [A \to B]^1}{B}\,E\!\to}{C \vee B}\,I\vee}{\dfrac{C \vee B}{(C \vee A) \to (C \vee B)}\,I\!\to (2)}\,E\vee\,(3)\,(4)}{(A \to B) \to ((C \vee A) \to (C \vee B))}\,I\!\to (1)$$

Para provarmos $(A \to B) \to ((C \vee A) \to (C \vee B))$, precisamos da introdução da implicação, significando que precisamos supor $A \to B$ e tentar provar $(C \vee A) \to (C \vee B)$ a partir disso. E, para provarmos este, seguimos o mesmo raciocínio: supomos $C \vee A$ e tentamos provar $C \vee B$ a partir disso. Para provarmos $C \vee B$, basta que apenas pelo menos um deles, C ou B, sejam verdadeiros.

Assim, a única premissa em que podemos aplicar alguma regra inicialmente é $C \vee A$. Faremos um procedimento idêntico ao de uma prova por casos: suporemos cada caso, C e A, separadamente, e esperamos que ambos os casos levem à uma mesma conclusão. Como nosso objetivo é provar $C \vee B$, segue que da premissa C podemos usar a introdução da disjunção para deduzir $C \vee B$, enquanto que, para A, podemos usar a introdução da implicação em $A \to B$ para deduzir B, e, portanto, podemos usar a introdução da disjunção para deduzir $C \vee B$. Como ambos os casos levam à uma mesma sentença, podemos deduzí-la usando a eliminação da disjunção e descartando as suposições adicionais dos casos, A e C. Como, a partir de $C \vee A$, conseguimos deduzir $C \vee B$, podemos usar a introdução da implicação para deduzir $(C \vee A) \to (C \vee B)$ e descartar $C \vee A$. Finalmente, como usamos $A \vee B$ para deduzir a sentença anterior, podemos usar a introdução da implicação para deduzir $(A \to B) \to ((C \vee A) \to (C \vee B))$ e descartar $A \to B$.

Logo, como existe uma árvore de dedução em que sua raiz é $(A \to B) \to ((C \vee A) \to (C \vee B))$, seus vértices são aplicações das regras de dedução e suas folhas são $[C \vee A]$, $[C]$, $[A]$ e $[A \to B]$, temos que $(A \to B) \to ((C \vee A) \to (C \vee B))$ é dedutível a partir do conjunto vazio e, portanto, é tautologia.

NEGAÇÃO

O operador de negação não possui uma regra de dedução própria. Porém, foi observado que:

$$(\neg\varphi) \equiv (\neg\varphi \vee \bot) \equiv (\varphi \to \bot)$$

Assim, usamos as regras da implicação para lidar com a negação.

EXEMPLO 3.20 . . . .
$(\neg A \vee B) \to (\neg B \to \neg A)$ *é tautologia?*

Pela equivalência entre tautologia e dedutibilidade, nossa pergunta será: $\vdash (\neg A \vee B) \to (\neg B \to \neg A)$? Em outras palavras, esperamos haver uma árvore de dedução cuja raiz é $(\neg A \vee B) \to (\neg B \to \neg A)$, cujos vértices são aplicações das regras de dedução e cujas folhas serão premissas adicionais devidamente descartadas. Substituindo as negações pelas implicações equivalentes, temos:

$$\dfrac{\dfrac{\dfrac{\dfrac{[(A \to \bot) \vee B]^1 \qquad \dfrac{[A]^3 \qquad [A \to \bot]^4}{\bot}\,E{\to} \qquad \dfrac{[B]^5 \qquad [B \to \bot]^2}{\bot}\,E{\to}\ (4)\ (5)}{\bot}\,E\vee}{A \to \bot}\,I{\to}\ (3)}{(B \to \bot) \to (A \to \bot)}\,I{\to}\ (2)}{((A \to \bot) \vee B) \to ((B \to \bot) \to (A \to \bot))}\,I{\to}\ (1)$$

Para provarmos $(\neg A \vee B) \to (\neg B \to \neg A)$, precisamos supor $\neg A \vee B$ e tentar deduzir $\neg B \to \neg A$. Para sermos capazes de deduzir este, precisamos supor $\neg B$ e tentar deduzir $\neg A$. Finalmente, para podermos deduzir esse último, precisamos supor A e tentar deduzir $\bot$ (lembrando que usamos as regras da implicação para tratar a negação). Assim, nosso objetivo é $\bot$.

Segue que podemos usar a eliminação da disjunção em $(A \to \bot) \vee B$. Supomos cada caso, $A \to \bot$ e B, e tentaremos chegar à uma mesma conclusão a partir das suposições. Podemos usar a premissa A e a eliminação da implicação em $A \to \bot$ para deduzir $\bot$ (note que, como A e $\neg A$ são verdadeiras, tal situação representa uma contradição). Podemos fazer o mesmo com o outro caso, usando a eliminação da implicação em $B \to \bot$ para deduzir $\bot$. Como chegamos a uma mesma sentença, finalizamos a eliminação da disjunção deduzindo $\bot$ e descartando $A \to \bot$ e B. Nesse momento, como usamos A para deduzir $\bot$, podemos usar a introdução da implicação para deduzir $A \to \bot$ e descartar A. Faremos o mesmo em dois passos seguintes, deduzindo $(B \to \bot) \to (A \to \bot)$ e descartando $B \to \bot$ e deduzindo $((A \to \bot) \vee B) \to ((B \to \bot) \to (A \to \bot))$ e descartando $(A \to \bot) \vee B$.

Assim, como existe uma árvore de dedução em que sua raiz é $(\neg A \vee B) \to (\neg B \to \neg A)$, seus vértices são aplicações das regras de dedução e suas folhas são $[A]$, $[\neg A]$, $[B]$, $[\neg B]$ e $[\neg A \vee B]$, temos que $(\neg A \vee B) \to (\neg B \to \neg A)$ é dedutível a partir do conjunto vazio e, portanto, é tautologia.

EXEMPLO 3.21 . . . .
$(y \to \neg z) \to (x \vee z)$ *é satisfatível?*

Usando a equivalência entre tautologia e dedutibilidade e:

φ é satisfatível se, e somente se, $\neg\varphi$ não é tautologia

Nossa pergunta será:
$\vdash \neg((y \to \neg z) \to (x \vee z))$? Dessa vez, esperamos que o método negue. Substituindo as negações pelas implicações equivalentes, temos:

$$[(y \to (z \to \bot)) \to (x \vee z)]^1$$

Para deduzirmos $\neg((y \to \neg z) \to (x \vee z))$, precisamos supor $(y \to \neg z) \to (x \vee z)$ e tentar deduzir $\bot$. Nesse momento, não temos mais nenhuma regra que podemos aplicar! Para prosseguirmos, precisaríamos de $(y \to \neg z)$ para usar a eliminação da implicação, mas é impossível obter essa sentença. Assim, não existe uma árvore cuja raiz é $\neg((y \to \neg z) \to (x \vee z))$, seus vértices são aplicações das regras de dedução e suas folhas são suposições adicionais devidamente descartadas. Logo, $\neg((y \to \neg z) \to (x \vee z))$ não é dedutível a partir do conjunto vazio e, portanto, não é tautologia. Assim, $(y \to \neg z) \to (x \vee z)$ é satisfatível.

AS TRÊS LÓGICAS

Até agora, vimos árvores de dedução cujas demonstrações seguem apenas as 6 regras básicas que definimos inicialmente. Mostraremos com os exemplos a seguir que o método da Dedução Natural não é correto com apenas tais regras.

EXEMPLO 3.22 . . . .
$\neg C \to (C \to D)$ é tautologia?

Nossa pergunta será:
$\vdash \neg C \to (C \to D)$? Substituindo a negação pela implicação equivalente, temos:

$$\frac{[C]^2 \qquad [C \to \bot]^1}{\bot}$$

Para provarmos $\neg C \to (C \to D)$, precisamos supor $\neg C$ e tentar deduzir $C \to D$. Assim, precisamos supor C e tentar deduzir D. Nosso objetivo é D.

Segue que podemos usar a eliminação da implicação em $C \to \bot$ para deduzir $\bot$. Porém, não conseguimos mais aplicar nenhuma regra (podemos ainda aplicar a introdução da implicação, mas levará no mesmo destino). Temos então, que $\neg C \to (C \to D)$ não é dedutível a partir do conjunto vazio e, portanto, não é tautologia.

Contudo, temos que a Tabela-Verdade mostra o contrário:

C	D	$\neg C$	$C \to D$	$\neg C \to (C \to D)$
0	0	1	1	1
0	1	1	1	1
1	0	0	0	1
1	1	0	1	1

Para que o método atinja sua corretude, é necessário definirmos mais uma regra de introdução:

$$\frac{\bot}{\varphi}$$

Se φ é atômica, podemos aplicar essa regra, cujo nome é **Princípio da Explosão** ou **Ex Falsum Quolibet**: "a partir do falso, deduzimos qualquer coisa".

EXEMPLO 3.23 . . . .
$\{D, C \to R, B \to (C \vee \neg D)\} \vdash (\neg A \wedge B) \to (D \vee R)$?

Queremos saber se existe uma árvore de dedução cuja raiz é $(\neg A \wedge B) \to (D \vee R)$, cujos vértices são aplicações das regras de dedução e cujas folhas são sentenças do conjunto e quaisquer premissas adicionais que possam vir a surgir. Usaremos a equivalência da negação com a implicação.

$$\dfrac{\dfrac{\dfrac{\dfrac{[(A \to \bot) \wedge B]^1}{B}\,E\wedge \qquad B \to (C \vee (D \to \bot))}{C \vee (D \to \bot)}\,E\!\to \qquad \dfrac{\dfrac{[C]^2 \qquad C \to R}{R}\,E\!\to}{D \vee R}\,I\vee \qquad \dfrac{\dfrac{\dfrac{D \qquad [D \to}{\bot}}{R}}{D \vee}}{D \vee R}}{(\neg A \wedge B) \to (D \vee R)}$$

II

LÓGICA DE PRIMEIRA ORDEM

CAPÍTULO 4

Sintaxe

CAPÍTULO 5

Semântica

CAPÍTULO

6

Problema da Satisfatibilidade