Apostila de Lógica

Professor Ruy J. Guerra B. de Queiroz
Transcrito por Eduardo Gade Gusmão

Sumário

1 – Introdução

- Você consegue descobrir quais das sentenças abaixo são consistentes?

"Seria errado censurar programas violentos na Televisão, pois o comportamento das pessoas não chega a ser afetado pelo que elas vêem na telinha. Igualmente, seria uma boa idéia se ter mais programas mostrando o lado bom do brasileiro, pois assim calaríamos a boca dos que estão sempre denegrindo o Brasil."

"Nunca desenhei nada na minha vida. Mas se eu sentasse e me concentrasse, bastariam apenas alguns minutos para eu produzir algo tão valioso quanto qualquer coisa que Picasso tenha feito."

"Walter se associou ao clube há dois anos, e tem sido um de seus sócios mais leais desde então. Ano passado ele pagou pelas férias de exatamente aqueles sócios que não pagaram por suas próprias férias."

Lógica: Estudo dos princípios e critérios da argumentação e inferência válidas.

Elementos Básicos de um argumento: Sentenças e, possivelmente, inferências.

Tipos de Sentenças envolvidas em um argumento:

- Proposição: Enunciados que podem ser verdadeiros ou falsos.
- Lógica Aristotélica: Objetos e predicados.

Classificação das Sentenças:

- Simples: "x é P" e "x não é P", onde x é o objeto e P é um predicado.
- Compostas:
  - Universal Afirmativa: "Todo A é B"
  - Universal Negativa: "Nenhum A é B"
  - Existencial Afirmativa: "Algum A é B"
  - Existencial Negativa: "Algum A não é B"

Inferência: Analisar as premissas e chegar a uma conclusão.

Premissa 1	Ex:	Todo A é B	Todo A é B
Premissa 2		Todo B é C	Algum B não é C
...		Todo A é C	Algum C não é A
Premissa n			
Conclusão			

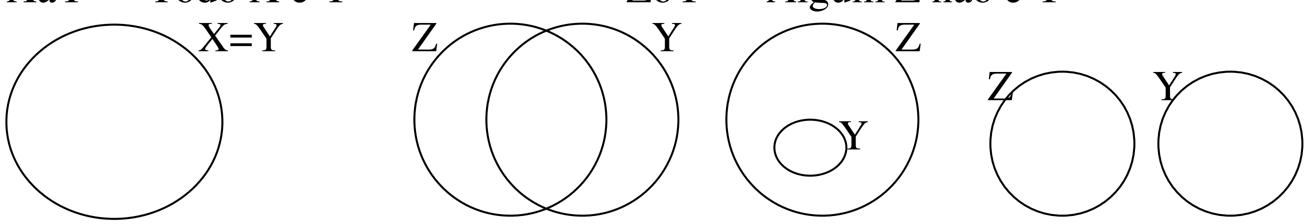
Legenda para os exercícios:
AaB – Todo A é B
AeB – Nenhum A é B
AiB – Algum A é B
AoB – Algum A não é B

Exercícios:

1) XaY, ZoY $\vdash$ XiZ
Resolução:

Premissas:

XaY – "Todo X é Y" ZoY – "Algum Z não é Y"



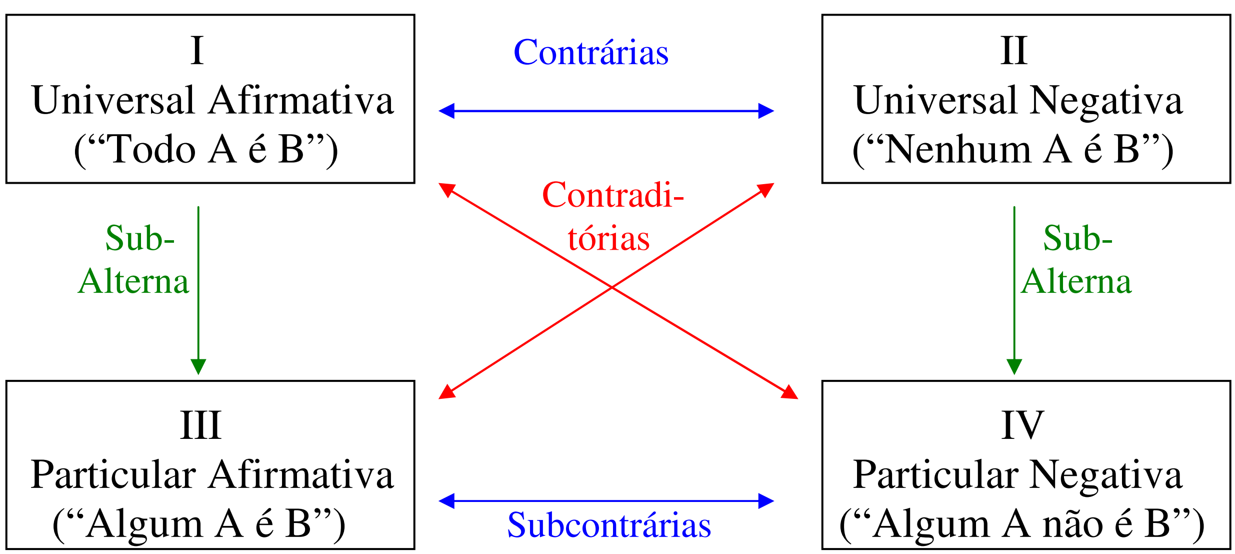
Através do Diagrama de Venn eu formulei todos os casos possíveis para a primeira premissa (apenas um caso, no diagrama mais a esquerda), e para a segunda (todos os outros 3 casos).

Conclusão: XiZ – “Algum X é Z”. Vemos que a conclusão é insegura pois, caso Z e Y sejam disjuntos (diagrama mais a direita), e aplicando à primeira premissa (onde X = Y), nenhum X será Z, gerando uma inconsistência.

Veja que no exemplo anterior eu formulei todos os casos onde a premissa é verdadeira (através do Diagrama de Venn), e consegui achar um caso onde a conclusão não se mostrou correta. Portanto eu afirmo que essa conclusão é insegura, mesmo havendo casos onde ela se mostra consistente.

2) PeR, RiQ $\vdash$ QoR
3) Min, Ron $\vdash$ MeR
4) SoT, UiS $\vdash$ TeU
5) PiQ, ReQ $\vdash$ RiP
6) PoR, QiR $\vdash$ PeQ
7) AeB, BiC $\vdash$ CeA
8) MiN, LeN $\vdash$ MiL

Quadrado da Oposição:



Tabelas-Verdade do Quadrado da Oposição:

I	II	III	IV
V	F	V	F
F	V-F	V-F	V

II	I	III	IV
V	F	F	V
F	V-F	V	F-V

III	I	II	IV
V	V-F	F	V-F
F	F	V	V

IV	I	II	III
V	F	V-F	V-F
F	V	F	V

As tabelas acima mostram, para um dado estado (verdadeiro ou falso), de um dado tipo de sentença (I, II, III, IV, veja o que cada número corresponde no Quadrado da oposição), o(s) estado(s) possível(is) nos outros tipos de sentença.

Definições:

Contradição: Duas Sentenças estão em contradição se, em todas as situações possíveis, elas têm valor-verdade opostos. Diz-se que as sentenças são contraditórias.

Inferências: Uma figura de inferência é uma seqüência de sentenças na qual a última é a conclusão e as restantes são as premissas.

Inferência Logicamente Válida (ou segura): Uma figura de inferência é dita logicamente válida (ou segura) se, em todas as situações nas quais as premissas são verdadeiras, a conclusão também é verdadeira.

Um argumento é uma seqüência ou conjunto de sentenças. Ele é válido se:

I. Não contém contradição, ou seja, não contém duas sentenças S1 e S2 tal que elas sejam contraditórias.
II. Toda figura de inferência utilizada no argumento é logicamente válida.
III. O conjunto de sentenças correspondente ao argumento é consistente.

Consistência: Um conjunto S de sentenças é dito inconsistente se existe um subconjunto S1 de S, tal que a figura de inferência que toma S1 como premissa e uma dada sentença P como conclusão é logicamente válida, ao mesmo tempo em que existe um outro subconjunto S2 de S, tal que a figura de inferência que toma S2 como premissa e a contraditória de P como conclusão é logicamente válida.

Um conjunto de sentenças é consistente se não é inconsistente.

Exemplo:

A, B e C são suspeitos de um crime, foram fornecidos os seguintes depoimentos:

A: "B é culpado e C é inocente."
B: "Se A for culpado, então C também é."
C: "Eu sou inocente, mas pelo menos um dos outros dois é culpado."

Como podemos observar, não há contradição aparente, então temos que usar algum método para descobrir alguma relação entre os depoimentos.

Vamos estabelecer "símbolos" para poder escrever as sentenças de uma forma padronizada:

A é culpado: X
B é culpado: Y
C é culpado: Z

Operações:
1- “e”: &, ^
2- “ou”: v
3- “não”: -, ¯, ~, ¬
4- “se então”: →

Escrevendo os depoimentos com esses símbolos:
A: Y ^ ¬Z
B: X → Z
C: ¬Z ^ (X v Y)

- Lógica Simbólica (1830)
  Gottlob Frege (1879)
  “Uma linguagem pura para escrever conceitos”

  - Sintaxe: Variáveis representam sentenças atômicas.

  Operadores:

  “^” = Conjunção “v” = Disjunção
  “¬” = Negação “→” = Implicação
  “∃” = quantificador existencial “∀” = quantificador universal

Símbolos auxiliares: -, (,)

  - Semântica: Leis que regulam o significado de sentenças compostas em relação ao significado das sentenças que a constituem.

Nos próximos capítulos vamos estudar cuidadosamente essas duas características da lógica proposicional (primeira unidade) e da lógica de predicados (segunda unidade).
Porém antes, veremos algumas definições úteis para mostrar o alfabeto simbólico que será visto no próximo capítulo.

Leitura interessante como introdução aos capítulos seguintes:
- http://en.wikipedia.org/wiki/Logic
- http://en.wikipedia.org/wiki/Aristotle
- http://nostalgia.wikipedia.org/wiki/Mathematical_logic
- http://pt.wikipedia.org/wiki/Silogismo

Definições:

Sintaxe da Lógica Simbólica: Estudo da forma das expressões da Lógica Simbólica.

Alfabeto Simbólico:

- Variáveis: x, y, z, ...
- Constantes: 0, 1
- Operadores: ^, v, →, ¬
- Símbolos Auxiliares: (,)

Objetivo Inicial:
Definir precisamente (i.e. matematicamente) o conjunto das expressões “bem formadas”, montadas com o alfabeto simbólico (conforme mostrado acima).

O conjunto das expressões bem formadas (EBF) é definido da seguinte forma:

I. Toda variável é EBF;
II. Toda constante é EBF;
III. Se P é EBF então (¬P) é EBF;
IV. Se P e Q são EBF então (P^Q) é EBF;
V. Se P e Q são EBF então (PvQ) é EBF;
VI. Se P e Q são EBF então (P→Q) é EBF;
VII. Nada mais é EBF;

“Etapas” para a formação do conjunto das EBF:

Etapa 1) EBF = {0, 1, x, y, z}
Etapa 2) EBF = {(¬0),(¬1),(¬x),(¬y),(¬z)}
Etapa 3) EBF = {(0^1), (0^0), (1^0), (1^1), (0^x), (x^0), (0^y), (y^0), (0^z), (z^0), ...}
Etapa 4) etc...

Nas Próximas etapas eu vou aplicando os outros operadores, até formar um conjunto com todos os elementos de todas as etapas.

- ## Conjunto Indutivo:

  Definição: Suponha que A seja um conjunto, e que x seja um subconjunto próprio de A (i.e. x C A). Suponha também que F

seja um conjunto funções sobre A, cada uma com sua aridade.
Um subconjunto Y de A é dito indutivo em relação a x e F se:

I. Y contém x;
II. Y é fechado sob as funções de F;

Observações:

- Aridade: Característica da função relativa aos parâmetros que ela recebe. Por exemplo, a função "eleva ao quadrado" receba um número e retorna o seu quadrado, por isso é uma função unária; Já a função "soma" recebe dois números e retorna a soma deles, portanto é uma função binária.
- O conjunto A já é um conjunto indutivo em relação a F.

Vamos procurar um método para achar o menor dos conjuntos indutivos em relação a x e F.

Que tal aplicar a operação de interseção à "família" de todos os possíveis conjuntos indutivos em relação a x e F.

Essa família seguramente não é vazia, pois pelo menos A está nela. Tome a interseção dessa família. O resultado será o menor dos conjuntos indutivos em relação a x e F (vamos chamá-lo de X^+).

No próximo capítulo vamos definir a lógica proposicional e relacioná-la aos conceitos de conjunto indutivo para definir sua sintaxe.

Lógica Proposicional

- Sintaxe da Lógica Proposicional:
  Estudo da forma das expressões da lógica proposicional.
  Dado o alfabeto simbólico ∑ a partir do qual construiremos as expressões que representarão sentenças da lógica proposicional. Desejamos definir precisamente o conjunto de palavras sobre ∑ que irão fazer parte do que consideramos expressões bem formadas (ou legítimas).
  O conceito de conjunto indutivamente definido é exatamente o que procuramos. (ver definição no capítulo anterior)

Suponha que A seja o conjunto de todas as palavras sobre o alfabeto simbólico ∑ = variáveis U constantes U operadores U símbolos auxiliares. Imagine que X seja o conjunto de "expressões atômicas" formadas por uma variável ou uma constante. Assuma que F seja o conjunto de funções que produzam palavras sobre ∑ ao receber palavras sobre ∑ como entrada. Mais especificamente, suponha que F conterá as funções que terão o papel de produzir expressões contendo os operadores/conectivos.

Ou seja, A = ∑* F = {f^,fv,f¬,f→}
Variáveis = {x,y,z} Constantes = {0,1} Operadores = {^,v,¬,→}

$\begin{cases} \text{f^: A x A} \rightarrow \text{A} \\ \text{w1, w2} \longrightarrow \text{(w1 ^ w2)} \end{cases}$

$\begin{cases} \text{fv: A x A} \rightarrow \text{A} \\ \text{w1, w2} \longrightarrow \text{(w1 v w2)} \end{cases}$

$\begin{cases} \text{f}\rightarrow\text{: A x A} \rightarrow \text{A} \\ \text{w1, w2} \longrightarrow \text{(w1} \rightarrow \text{w2)} \end{cases}$

$\begin{cases} \text{f}\neg\text{: A} \rightarrow \text{A} \\ \text{w} \longrightarrow (\neg\text{w}) \end{cases}$

Exemplo: w1 = "x(→" w2 = ")0^v"
f^(w1,w2) = f^(x(→,)0^v) = (x(→ ^)0^v)
f→ (w1,w2) = f→ (x(→,)0^v) = (x(→ →)0^v)

Desejamos encontrar o menor dos conjuntos indutivos sob X e F. Para isso podemos imaginar uma “família” de subconjuntos de A que são todos indutivos sob X e A. Essa família não é vazia, pois o próprio A pertence a ela. Agora tome a interseção de todos os conjuntos dessa família, o resultado será, naturalmente, o menor dos conjuntos indutivos sob X e F. (como foi visto anteriormente, a notação deste menor conjunto indutivo formado dessa forma é X').

Podemos também imaginar um processo mais “construtivo” de obter o menor conjunto indutivo sob X e F. Trata-se de um processo iterativo que começa pelo X, e vai completando com o que falta para que se chegue a um conjunto que seja fechado sob F, portanto indutivo sob X e F.

Matematicamente:

$$\begin{cases} X_o = X \\ X_{n+1} = X_n \cup \{f(x1,...,xk) \,/\, (x1,...,xk) \in X_n,\ f \in F,\ \text{aridade}(F) = k\} \end{cases}$$

Agora tome a união de todos os Xi , $0 \leq i <$ infinito
Obtemos assim, no limite, um conjunto que contém X e é fechado sob F, portanto é o fecho indutivo de X sob F. Denotamos esse conjunto por: X_{+} .

Resta-nos mostrar que esses dois métodos que descrevemos para obter o nosso menor conjunto indutivo sob X e F, resultam num mesmo subconjunto de A.

Lema: $X_{+} = X^{+}$

Prova: (esboço)

i) X^{+} está contido ou é igual a X_{+}
1- X_{+} contém X? 2- X_{+} é fechado sob F?

ii) X^+ está contido ou é igual a X^+

Base: X_o está contido ou é igual a X^+
Caso Indutivo:
Hipótese de Indução: X_k está contido ou é igual a X^+
Tese: X_{k+1} está contido ou é igual a X^+

$X_{k+1} = X_k \cup \{\underline{f\quad\quad}\}$
$X_o \cup X_1 \cup X_2 \cup \ldots$ está contido ou é igual a X^+.
X_i está contido ou é igual a X^+ $\quad 0 \leq i <$ infinito

- Conjuntos Livremente Gerados

Conforme dissemos anteriormente, desejamos caracterizar precisamente (i.e. matematicamente) o conjunto das palavras sobre o alfabeto simbólico $\sum$ = variáveis U constantes U operadores U símbolos auxiliares, que sejam expressões da lógica proposicional. Usando os conceitos de conjunto indutivo chegamos a um ponto em que temos condições de:

I. Reconhecer as palavras sobre $\sum$ que são "proposições" da lógica proposicional.
II. Usar o método da "prova por indução" para provar uma afirmação do tipo "Todo elemento do conjunto das proposições tem a propriedade de ...".

- Definição (Conjunto Livremente Gerado):

Seja A e X conjuntos, X está contido em A, e F um conjunto de funções sobre A, cada uma com sua aridade. O fecho indutivo X^+ de X sob F será livremente gerado se:

I. Todas as funções de F forem injetoras.
II. Para quaisquer duas funções f e g de F, o conjunto imagem de f em relação a X^+ e o conjunto imagem de g em relação a X^+ são disjuntos.

III. Nenhum elemento de base X é resultado de algum f de F aplicado a argumentos vindos de X_+.

Lema: O conjunto PROP das proposições da lógica proposicional é livremente gerado.

Observação: Todo conjunto livremente gerado tem a chamada "propriedade da leitura única", i.e., qualquer elemento tem uma única formação. Isso significa que o processo de definição recursiva de funções sobre conjunto livremente gerado é seguro matematicamente.

- Definindo Recursivamente funções sobre PROP:

  Legenda: E = expressão atômica;
  E1,E2 = expressões atômicas ou não;

1) Contando o número de parênteses (de uma dada proposição):

   - $f: \text{PROP} \longrightarrow \text{IN}$ (naturais)

$$\begin{cases} - f(E) = 0 & - f((\neg E1)) = f(E1)+2 \\ - f((E1 \wedge E2)) = f(E1)+f(E2)+2 & - f((E1 \vee E2)) = f(E1)+f(E2)+2 \\ - f((E1 \rightarrow E2)) = f(E1)+f(E2)+2 & \end{cases}$$

2) Contando o número de operadores:

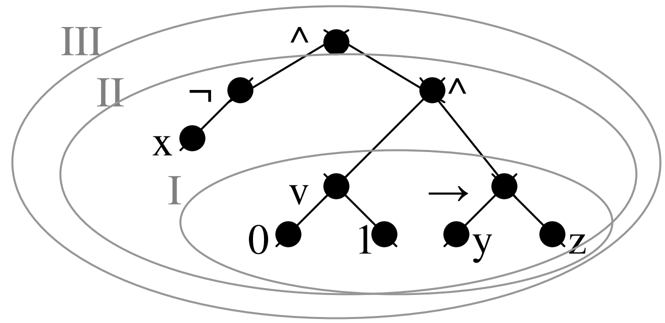
   - $g: \text{PROP} \longrightarrow \text{IN}$

$$\begin{cases} - g(E) = 0 & - g((\neg E1)) = g(E1)+1 \\ - g((E1 \wedge E2)) = g(E1)+g(E2)+1 & - g((E1 \vee E2)) = g(E1)+g(E2)+1 \\ - g((E1 \rightarrow E2)) = g(E1)+g(E2)+1 & \end{cases}$$

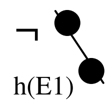
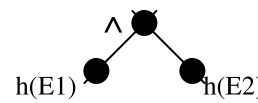
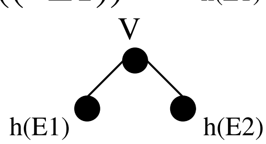
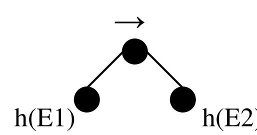
3) Montando a árvore sintática:

Exemplo de árvore sintática:

I. (0v1) , (y→z)
II. ((0v1) ^ (y→z)) , (¬x)
III. ((¬x) ^ ((0v1) ^ (y→z)))



○ h: PROP ⟶ (representação gráfica de árvores)

- h(E) =  - h((¬E1)) =  - h((E1^E2)) = 
- h((E1vE2)) =  - h (E1→E2) = 

4) Calculando o Posto de uma árvore:

○ p: PROP ⟶ IN

- p(E) = 0
- p((¬E1)) = 1 + p(E1)
- p((E1→E2)) = 1 + max(p(E1),p(E2))
- p((E1^E2)) = 1 + max(p(E1),p(E2))
- p((E1vE2)) = 1 + max(p(E1),p(E2))

5) Obtendo o conjunto das subexpressões:

○ s: PROP ⟶ PROP

- s(E) = {E}
- s((¬E1)) = {(¬E1)} U s(E1)
- s((E1→E2)) = {(E1→E2)} U s(E1) U s(E2)
- s((E1^E2)) = {(E1^E2)} U s(E1) U s(E2)
- s((E1vE2)) = {(E1vE2)} U s(E1) U s(E2)

- Lema: Para toda proposição φ o número de parênteses de φ é par.

Prova: Por indução sobre a complexidade de φ

I) Caso Base: φ é atômica

A função f conta o número de parênteses de uma dada expressão. (veja sua definição no tópico anterior)

$f(\varphi) = 0$ e 0 é par c.q.d. (como queria demonstrar)

II) Casos Indutivos:

i) φ é da forma $(\neg\psi)$:

Hipótese de indução: $f(\psi)$ é par.

Tese: $f((\neg\psi))$ é par.

Como f(PSI) é par, então $f(\psi) + 2$ também é.

Como $f((\neg\psi)) = f(\psi) + 2$.

Logo, $f((\neg\psi))$ também é par. c.q.d.

ii) φ é da forma $(\rho \square \omega)$:

Onde $\square = \wedge, \vee, \rightarrow$

Hipótese de indução: $f(\rho)$ é par e $f(\omega)$ é par.

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

Da definição de f:

$f((\rho \square \omega)) = f(\rho) + f(\omega) + 2$

Como $f(\rho)$ é par, $f(\omega)$ também é par.

Então $f(\rho) + f(\omega) + 2$ também é par.

Logo $f((\rho \square \omega))$ é par. c.q.d.

- Lema: Para todo $\varphi \in$ PROP, o número de subexpressões de φ é no máximo igual ao dobro do número de operadores de φ + 1. (i.e. $|s(\varphi)| \leq 2g(\varphi) + 1$).

Prova: Por indução sobre a complexidade de φ.

I) Caso Base: φ é atômica.

Então $s(\varphi) = \{\varphi\}$, e $g(\varphi) = 0$

Daí $|s(\varphi)| = 1$. Logo $|s(\varphi)| \leq 2.0 + 1$ c.q.d.

II) Casos Indutivos:

i) φ é da forma $(\neg \psi)$:

Hipótese de indução: $|s(\psi)| \leq 2.g(\psi) + 1$

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

Temos que provar que a tese é verdadeira, tendo a hipótese de indução como verdadeira.

Ora, pela definição de função que obtém o conjunto das subexpressões de uma dada proposição, temos que:

$s(\neg \psi) = \{\neg \psi\} \cup s(\psi)$

$|s(\neg \psi)| = |\{\neg \psi\} \cup s(\psi)|$

$|s(\neg \psi)| = |\{\neg \psi\}| + |s(\psi)| - |\{\neg \psi\} \cap s(\psi)|$

Como $\{\neg \psi\}$ e $s(\psi)$ são disjuntos temos:

$|s(\neg \psi)| = 1 + |s(\psi)|$

Também, pela definição da função que conta o número de operadores de uma dada proposição temos que:

$g(\neg \psi) = 1 + g(\psi)$

Daí, temos que mostrar que:

$|s(\neg \psi)| \leq 2.g(\neg \psi) + 1$

Isto é: $1 + |s(\psi)| \leq 2.g(\psi) + 1 + 1$

Da Hipótese de indução:

$1 + |s(\psi)| \leq 2.g(\psi) + 1 + 1$

Logo: $1 + |s(\psi)| \leq 2.g(\psi) + 1 + 1 + 1$ c.q.d.

ii) φ é da forma $(\rho \square \omega)$:

Hipótese de indução: $\begin{cases} |s(\rho)| \leq 2.g(\rho) + 1 \\ |s(\omega)| \leq 2.g(\omega) + 1 \end{cases}$

Tese: $|s(\rho \square \omega)| \leq 2.g(\rho \square \omega) + 1$

Temos que provar, mais uma vez, que a tese é verdadeira, tendo a hipótese de indução como verdadeira.

Ora, pela definição da função que obtém o conjunto das subproposições de uma dada proposição, temos que:

$s(\rho \square \omega) = \{(\rho \square \omega)\} \cup s(\rho) \cup s(\omega)$

$|s(\rho \square \omega)| = |\{(\rho \square \omega)\} \cup (s(\rho) \cup s(\omega))|$

$= |\{(\rho \square \omega)\}| + |s(\rho) \cup s(\omega)| - |\{(\rho \square \omega)\} \cap (s(\rho) \cup s(\omega))|$

Como $\{(\rho \square \omega)\}$ e $(s(\rho) \cup s(\omega))$ são disjuntos, a terceira parcela é nula. Daí:

$|s(\rho \square \omega)| = 1 + |s(\rho) \cup s(\omega)|$

$|s(\rho \square \omega)| = 1 + |s(\rho)| + |s(\omega)| - |s(\rho) \cap s(\omega)|$

Pela definição da função que conta o número de operadores, temos:

$g(\rho \square \omega) = 1 + g(\rho) + g(\omega)$

Então a tese que temos que provar é:

$1 + |s(\rho)| + |s(\omega)| - |s(\rho) \cap s(\omega)| \leq 2.(1 + g(\rho) + g(\omega)) + 1$

Da hipótese de indução:

$|s(\rho)| + |s(\omega)| \leq (2.g(\rho) + 1) + (2.g(\omega) + 1)$

Logo:

$1 + |s(\rho)| + |s(\omega)| \leq (2.g(\rho) + 1) + (2.g(\omega) + 1) + 1$

Daí: $1 + |s(\rho)| + |s(\omega)| \leq 2(g(\rho) + g(\omega) + 1) + 1$

Logo:

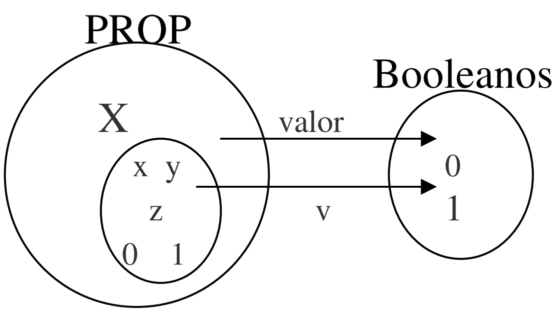
$1 + |s(\rho)| + |s(\omega)| - |s(\rho) \cap s(\omega)| \leq 2(g(\rho) + g(\omega)) + 1$ c.q.d.

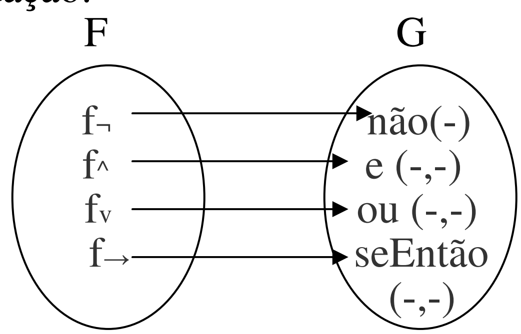
Semântica da Lógica Proposicional

Agora vamos estudar o significado das expressões da lógica proposicional.

Dado que o conjunto das proposições PROP é livremente gerado, sabemos que se podem definir funções recursivamente sobre PROP que o resultado será garantidamente seguro do ponto de vista matemático.
Particularmente, estamos interessados em definir uma função que calcula o valor de uma proposição recursivamente, baseada no fato de que o valor das possíveis variáveis que nela ocorrem está fixado.

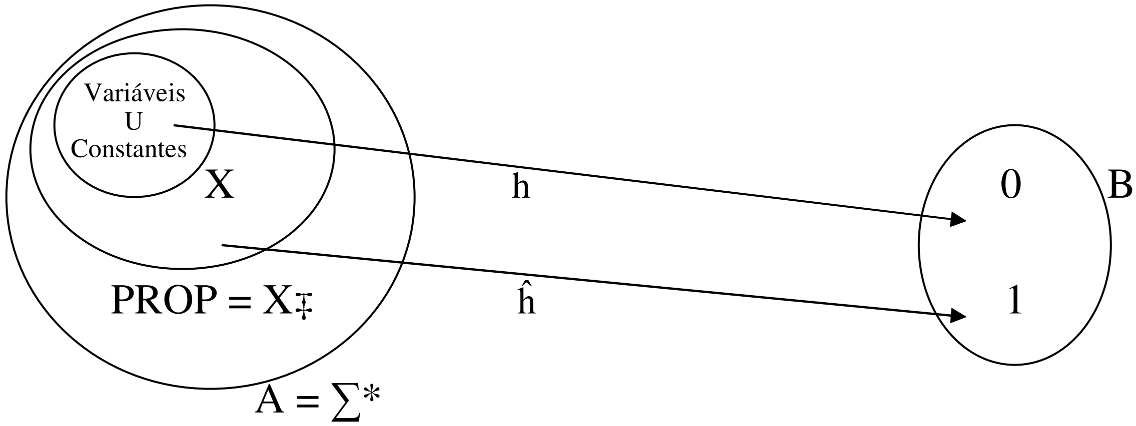
Assim deparamos com a seguinte situação:





- Teorema da extensão homomórfica única: Seja A um conjunto, X C A, e F um conjunto de funções sobre A. Seja B um outro conjunto, e G um conjunto de funções sobre B tal que existe uma associação $d: F \rightarrow G$ entre cada função de F com uma função de G com mesma aridade. Se o fecho indutivo de X sob F, isto é, $X_{\ddagger}$, for livremente gerado então, para toda função $h: X \rightarrow B$ existe uma única função $\hat{h}: X_{\ddagger} \rightarrow B$ tal que:

I. $\hat{h}(x) = h(x)$ se $x \in X$;
II. $\hat{h}(\, f_i(x_1, \ldots, x_k)\,) = g_i(\, \hat{h}(x_1), \ldots, \hat{h}(x_k)\,)$
$g_i = d(f_i)$



Obs: Quando X‡ = PROP e B = {0,1},

o $h: X \to \{0,1\}$ e o $\hat{h}: PROP \to \{0,1\}$ serão tais que:

I. $\hat{h}(x) = h(x)$ se $x \in X$;

II. $\hat{h}(\neg\psi) = \text{NÃO}(\hat{h}(\psi))$;

III. $\hat{h}((\omega \wedge \rho)) = \text{E}(\hat{h}(\omega), \hat{h}(\rho))$;

IV. $\hat{h}((\omega \vee \rho)) = \text{OU}(\hat{h}(\omega), \hat{h}(\rho))$;

V. $\hat{h}((\omega \to \rho)) = \text{SE-ENTÃO}(\hat{h}(\omega), \hat{h}(\rho))$.

- Definição de função de valoração-verdade: Seja PROP o conjunto de proposições da lógica proposicional. Então uma função de valoração verdade h é uma função que atribui a cada variável um valor booleano, ou valor verdade.

Observação: Como PROP é livremente gerado, o Teorema da Extensão Homomórfica Única garante que para cada valoração-verdade $h: X \to \{0,1\}$, existe uma única extensão $\hat{h}: PROP \to \{0,1\}$, que é exatamente a função que calcula o valor-verdade de uma dada expressão.

- Definição de valoração que satisfaz proposição: Dizemos que uma valoração-verdade $w: X \to \{0,1\}$ satisfaz uma proposição φ se $\hat{w}(\varphi) = 1$.

Exemplo:

- Seja h: x $\longrightarrow$ {0,1} a seguinte valoração verdade:

$$\begin{cases} h(x) = 0; \\ h(y) = 1; \\ h(z) = 0; \end{cases}$$

Vamos calcular o valor-verdade da proposição $\varphi = ((\neg z) \wedge (x \vee y))$
Sabemos que o valor será dado pela expressão única de h, dada por:

$\hat{h}$: PROP $\longrightarrow$ {0,1} tal que:

$\hat{h}((\neg z)\wedge(x\vee y))$
$= E(\hat{h}(\neg z), \hat{h}(x\vee y))$
$= E(\text{NÃO}(\hat{h}(z)), \text{OU}(\hat{h}(x), \hat{h}(y)))$
$= E(\text{NÃO}(h(z)), \text{OU}(h(x), h(y)))$
$= E(\text{NÃO}(0), \text{OU}(0,1))$
$= E(1, 1)$
$= 1$

Portanto o valor-verdade de φ é 1.

O Problema da Satisfatibilidade e Correlatos

Lembrando:

- Definição (valoração-verdade): Uma valoração-verdade, ou simplesmente valoração, é uma função w que associa um valor-verdade, isto é, um valor booleano, a cada variável.
- Definição (valoração satisfaz a uma proposição): Uma valoração w satisfaz uma proposição φ se a sua extensão $\hat{w}$ aplicada a φ dá como resultado o valor-verdade "1" (ou seja, o valor booleano correspondente ao "verdadeiro").

Definição de Satisfatibilidade:

Seja φ uma proposição:

➢ φ é dita satisfatível se existe pelo menos uma valoração w que a satisfaz. (i.e., $\hat{w}(\varphi) = 1$).

➢ φ é chamada de tautologia se toda a valoração a satisfaz.

➢ φ é dita refutável se existe pelo menos uma valoração que não a satisfaz. (i.e., $\hat{w}(\varphi) = 0$).

➢ φ é chamada de insatisfatível se nenhuma valoração a satisfaz.

Dada uma proposição φ pergunta-se:

➢ φ é satisfatível? (SAT)

➢ φ é refutável? (REFUT)

➢ φ é tautological (TAUT)

➢ φ é insatisfatível (INSAT)

Exemplo: Seja $\varphi = ((x \rightarrow (y \rightarrow z)) \rightarrow ((\neg z) \vee (x \wedge (\neg y))))$

Pergunta-se: φ é satisfatível?

Para resolver o problema proposto existem diversos métodos, que estudaremos neste curso. Para ter uma visão inicial, o primeiro método mostrado será o da tabela-verdade, onde nós definimos as operações booleanas que iremos usar (^,v, →,¬), e a partir disso calculamos a valoração da expressão por partes.

Operações Booleanas:

NÃO	
X	Não(x)
0	1
1	0

OU		
X	Y	Ou(x,y)
0	0	0
0	1	1
1	0	1
1	1	1

E		
X	Y	E(x,y)
0	0	0
0	1	0
1	0	0
1	1	1

SE-ENTÃO		
X	Y	Se-ent(x,y)
0	0	1
0	1	1
1	0	0
1	1	1

Construindo a tabela-verdade para φ:

X	Y	Z	¬Z	¬Y	(X^(¬Y))	(¬Z)v(X^(¬Y))	Y→Z	X→(Y→Z)	φ
0	0	0	1	1	0	1	1	1	1
0	0	1	0	1	0	0	1	1	0
0	1	0	1	0	0	1	0	1	1
0	1	1	0	0	0	0	1	1	0
1	0	0	1	1	1	1	1	1	1
1	0	1	0	1	1	1	1	1	1
1	1	0	1	0	0	1	0	0	1
1	1	1	0	0	0	0	1	1	0

Conclusões:

- φ é satisfatível.
- φ é refutável.
- φ não é tautologia.
- φ não é insatisfatível.

- Conjunto de Proposições:

Suponha que S seja um conjunto de proposições, dizemos que S é satisfatível se existe pelo menos uma valoração que satisfaz a todas as proposições de S.

S será insatisfatível quando não for satisfatível.

Exemplo 2: Seja S = {(Y^(¬Z)), (X→Z), ((¬Z)^(XvY))}

Pergunta-se: S é satisfatível?

X	Y	Z	¬Z	X→Z	(Y^(¬Z))	(XvY)	(¬Z)^(XvY)
0	0	0	1	1	0	0	0
0	0	1	0	1	0	0	0
0	1	0	1	1	1	1	1
0	1	1	0	1	0	1	0
1	0	0	1	0	0	1	1
1	0	1	0	1	0	1	0
1	1	0	1	0	1	1	1
1	1	1	0	1	0	1	0

Portanto o conjunto S é satisfatível pois existe um "mundo" onde todas as valorações das proposições que compõem o conjunto são verdadeiras(círculos).

- Conseqüência lógica: Dizemos que uma proposição φ é uma conseqüência lógica do conjunto S se toda valoração que satisfaz S também satisfaz φ.

No exemplo anterior, a proposição (Y^(¬Z)) é conseqüência lógica do conjunto T formado pelas outras duas.

A proposição (¬Z)^(XvY) também é conseqüência lógica do conjunto T´ formado pelas duas restantes.

Já a proposição (X→Z) não é conseqüência lógica do conjunto T´´ formado pelas outras duas. (Veja tudo isso na tabela)

- Relacionamento entre os conceitos em torno da propriedade de satisfatibilidade

    - Teorema (1): φ é uma conseqüência lógica de S se, e somente se, S U { $\neg\varphi$ } é insatisfatível.

    - Teorema (2): Seja φ uma proposição. φ é satisfatível se, e somente se, $\neg\varphi$ é refutável.

Prova(do teorema(2)):

I. Se φ é satisfatível então $\neg\varphi$ é refutável.
Suponha que φ é satisfatível.
Ora, então existe w tal que $\hat{w}(\varphi) = 1$.
Claro que $\hat{w}(\neg\varphi) = 0$. Então existe valoração que não satisfaz $\neg\varphi$. Logo $\neg\varphi$ é refutável.

II. Se $\neg\varphi$ é refutável então φ é satisfatível.
Suponha que $\neg\varphi$ seja refutável.
Então existe w tal que $\hat{w}(\neg\varphi) = 0$.
Claro que $\hat{w}(\varphi) = 1$. Logo, existe w que satisfaz φ. Portanto φ é satisfatível.

Considerações sobre o custo computacional do problema SAT

Problema SAT: Dada uma expressão φ da lógica proposicional pergunta-se: φ é satisfatível?

Observação: Note que para resolver uma instância do problema SAT, temos que realizar um certo número de operações booleanas, que depende do "tamanho" da expressão de entrada φ.

Que tal tentar encontrar uma forma razoavelmente precisa para estimar o número máximo de operações que devemos realizar para chegar a uma solução para o SAT?

Estimativa:

Seja φ uma expressão. Assuma que:

"n": número de operadores em φ.

"k": número de variáveis listadas em φ.

Sabemos do conceito de valoração-verdade que vamos ter que testar (2 elevado a k) valorações.

Também vimos anteriormente que o número de subexpressões de φ é no máximo igual ao número de operadores + 1. Daí, para cada uma dessas (2 elevado a k) valorações, vamos ter que fazer no máximo 2n+1 operações.

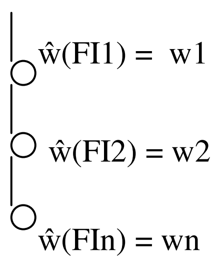
Método dos “Tableaux” Analíticos

- Evert Beth (1954)
- Jaakko Hintikka (1955)

Seja S um conjunto de proposições, Suponha que estejamos buscando uma valoração w: X $\rightarrow$ BOOL, tal que $\hat{w}(\varphi)$, para cada $\varphi \in S$, seja dado por $\hat{w}(\varphi_1) = w_1$, $\hat{w}(\varphi_2) = w_2$, ..., $\hat{w}(\varphi_n) = \varphi_n$, onde $v_i(1 \leq i \leq n)$ é um valor booleano.

Vamos chamar esse conjunto de condições de “condições iniciais do tableaux”.

Agora, vamos montar uma “árvore de possibilidades” que toma cada equação contida nas condições iniciais como nós de uma árvore, um imediatamente abaixo do seguinte:

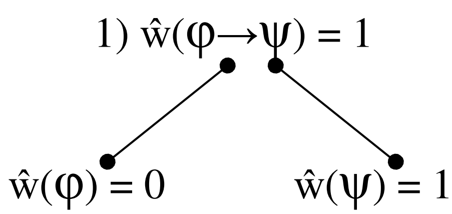


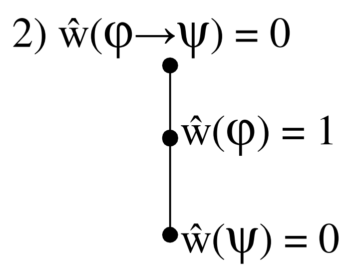
Para cada nó da árvore inicial que contém uma fórmula composta, aplique uma das “regras do tableaux”.(ver as regras abaixo).

Repita esse procedimento, sempre expandindo a árvore em direção às folhas, até que não haja mais nenhum nó com fórmula composta que não tenha sido devidamente simplificado por uma das regras do tabelaux.

Agora, percorra todos os caminhos da raiz até uma folha, e utilize como solução para o problema original todo caminho que não apresente contradição. Essas soluções, se existirem, serão os “mundos possíveis”.

Regras do tableaux:





3) $\hat{w}(\varphi \wedge \psi) = 0$ $\hat{w}(\varphi) = 0$ $\quad$ $\hat{w}(\varphi) = 0$	4) $\hat{w}(\varphi \wedge \psi) = 1$ $\hat{w}(\varphi) = 1$ $\hat{w}(\varphi) = 1$
5) $\hat{w}(\varphi v \psi) = 1$ $\hat{w}(\varphi) = 1$ $\quad$ $\hat{w}(\varphi) = 1$	6) $\hat{w}(\varphi v \psi) = 0$ $\hat{w}(\varphi) = 0$ $\hat{w}(\varphi) = 0$

Exemplos:

1) $\varphi = ((X \rightarrow (\neg Y)) \rightarrow ((\neg Z) \wedge (Y \rightarrow (\neg X))))$

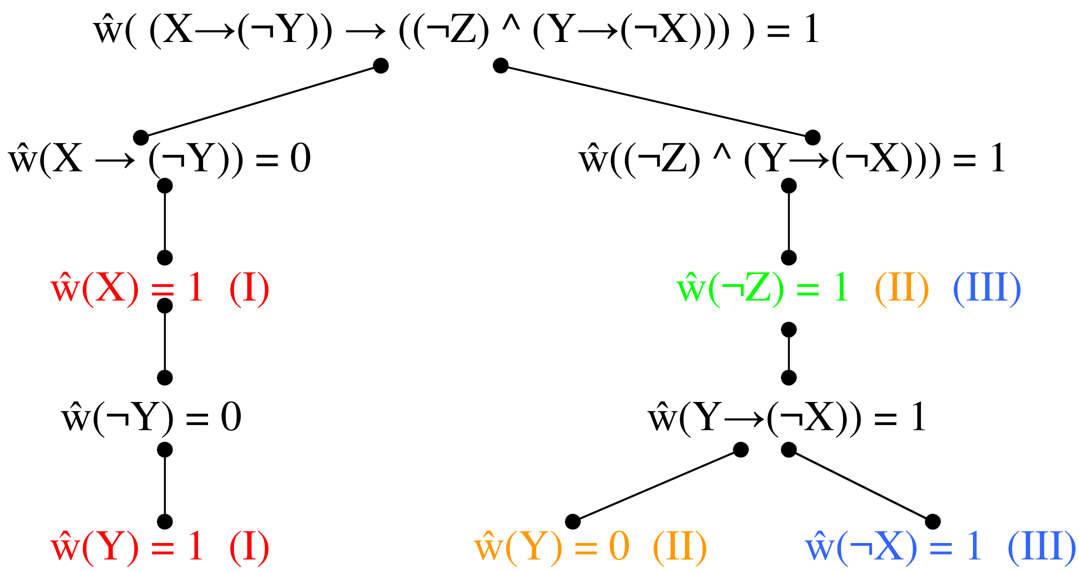
φ é satisfatível?

Quero saber se existe pelo menos uma valoração que satisfaz a expressão φ. Em outras palavras, quero saber se existe um mundo possível no qual φ seja verdadeira.

Como dito anteriormente, vamos construir a árvore de possibilidades para a expressão φ. Vamos começar com φ e então ir “desmontando” esta expressão em subexpressões através das regras do tableaux, até não haver mais subexpressões no qual possamos usar alguma regra. Fazemos isso de cima para baixo, i.e., da raiz em direção as folhas.

Observe que nós não paramos no momento que acharmos todas as versões atômicas de cada variável presente na expressão, e sim quando não há mais subexpressões para aplicar as regras. Veja que se parássemos ao achar as subexpressões atômicas das variáveis difícilmente encontraríamos uma contradição em algum mundo, pois as contradições geralmente vêm na forma: X e ¬X, no mesmo mundo.

Árvore de possibilidades:



(Observe que na parte direita da árvore eu não calculei os NOT, por ser trivial e ocupar espaço desnecessário. Mas veja que o $\neg Z$, por exemplo, iria "descender" dos nós: "$\hat{w}(Y) = 0$" e "$\hat{w}(\neg X) = 1$").

Mundo possível: caminho da raiz até uma folha.

Mundo (I): $\hat{w}(X) = 1$, $\hat{w}(Y) = 1$, $\hat{w}(Z) = ?$
Mundo (II): $\hat{w}(X) = ?$, $\hat{w}(Y) = 0$, $\hat{w}(Z) = 0$
Mundo (III): $\hat{w}(X) = 0$, $\hat{w}(Y) = ?$, $\hat{w}(Z) = 0$

Veja que juntando todos os mundos nós temos 5 possibilidades (pois a possibilidade E = possibilidade F) diferentes de valorações que satisfazem à expressão:

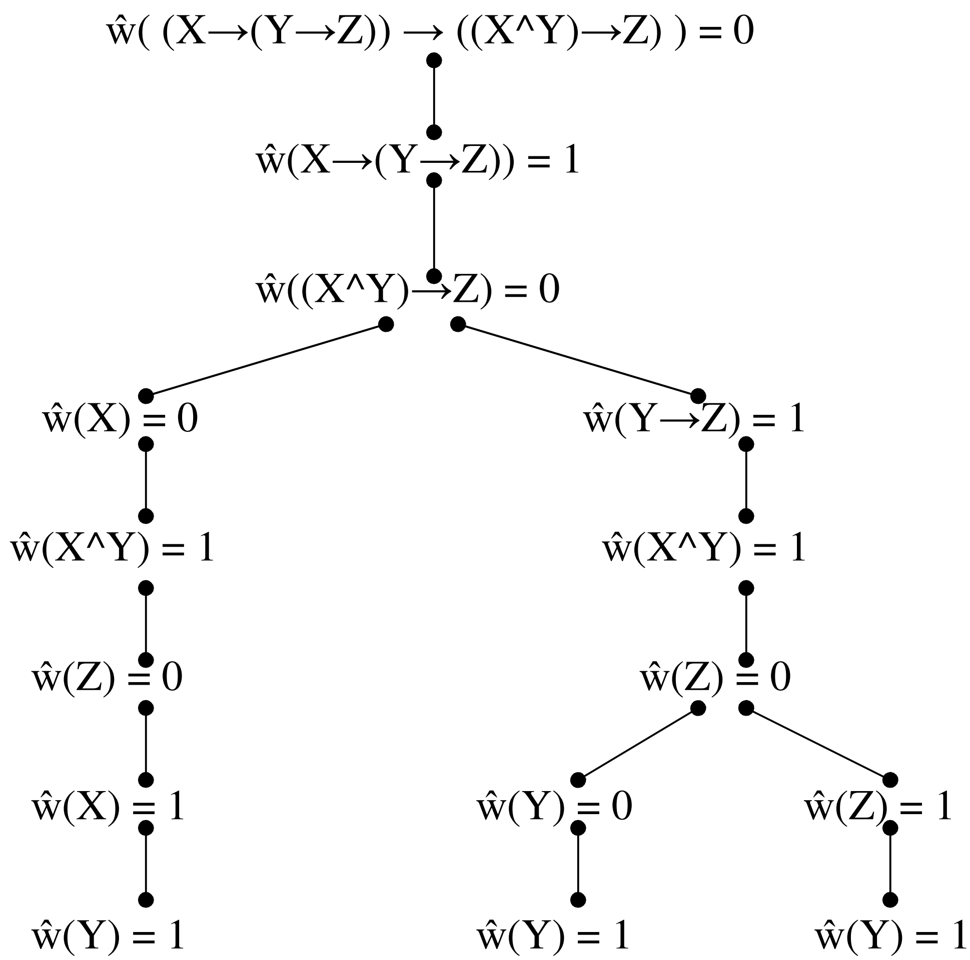
Possibilidade A: $\hat{w}(X) = 1$, $\hat{w}(Y) = 1$, $\hat{w}(Z) = 0$
Possibilidade B: $\hat{w}(X) = 1$, $\hat{w}(Y) = 1$, $\hat{w}(Z) = 1$
Possibilidade C: $\hat{w}(X) = 0$, $\hat{w}(Y) = 0$, $\hat{w}(Z) = 0$
Possibilidade D: $\hat{w}(X) = 1$, $\hat{w}(Y) = 0$, $\hat{w}(Z) = 0$
Possibilidade E: $\hat{w}(X) = 0$, $\hat{w}(Y) = 0$, $\hat{w}(Z) = 0$
Possibilidade F: $\hat{w}(X) = 0$, $\hat{w}(Y) = 1$, $\hat{w}(Z) = 0$

E já que existem esses 5 mundos possíveis, a expressão φ é satisfatível!

2) $\varphi = ((X \rightarrow (Y \rightarrow Z)) \rightarrow ((X \wedge Y) \rightarrow Z))$
φ é tautologia?

Dessa vez eu vou verificar se existe algum mundo possível onde $\hat{w}(\varphi) = 0$. Se existir tal mundo, φ não é tautologia, pois para ser tautologia, todas as valorações devem ser "1".

Árvore de possibilidades:



A partir da árvore de possibilidades, nós percebemos que existem contradições em todos os mundos possíveis, portanto, como não existe um mundo onde $\hat{w}(\varphi) = 0$, φ é tautologia!

3) Vamos ver como fazemos para descobrir se uma proposição é conseqüência lógica de um conjunto de proposições através do método dos tableaux.

Veja as seguintes sentenças:

I. O unicórnio, se é lenda é imortal, e se não é lenda é mamífero.
II. O unicórnio, se é imortal ou mamífero, é chifrudo.
III. O unicórnio, se é chifrudo, então é bruxaria.
IV. O unicórnio é lenda.
V. O unicórnio é bruxaria.

Queremos verificar se as sentenças IV e V são conseqüências do conjunto S formado pelas sentenças I, II e III.

Para fazer isso nós temos que criar uma árvore de possibilidades onde as valorações das sentenças do conjunto S são "1" e a valoração da sentença no qual se quer provar é "0". Poderíamos verificar se mais de uma sentença é conseqüência lógica de S na mesma árvore, mas é mais prudente que se verifique uma sentença de cada vez.

Primeiro vamos definir os símbolos que utilizaremos para representar as sentenças acima, e então mostrar o começo de como seria a árvore das possibilidades apenas para a sentença IV (ver se ela é conseqüência lógica das três anteriores).

Linguagem Simbólica:

1) "O unicórnio é lenda" = l
2) "O unicórnio é imortal" = i
3) "O unicórnio é mamífero" = m
4) "O unicórnio é chifrudo" = c
5) "O unicórnio é bruxaria" = b

Dessa forma as sentenças podem ser escritas:

I. $((l \rightarrow i) \wedge ((\neg l) \rightarrow m))$
II. $((i \vee m) \rightarrow c)$
III. $(c \rightarrow b)$
IV. l
V. b

Árvore de possibilidades:

$\hat{w}((l \rightarrow i) \wedge ((\neg l) \rightarrow m)) = 1$

$\hat{w}((i \vee m) \rightarrow c) = 1$

$\hat{w}(c \rightarrow b) = 1$

$\hat{w}(l) = 0$

...

A partir dessa configuração inicial da árvore, basta desenvolver o método dos tableaux. No próximo exemplo mostrarei porque não desenvolvi aqui este exemplo.

4) φ = (((X^Y)v((¬X)^Y)) v ((X^(¬Y))v((¬X)^(¬Y)))) Verifique se φ é tautologia?

Tente desenvolver o exemplo 4. Você verá que, como no exemplo 3, a árvore gerada será muito grande, mostrando que nem sempre o método dos tableaux é a melhor solução para problemas SAT.

Existem valores onde o número de operações (no tableaux) é igual à do método da tabela-verdade, tornando esse método ineficiente.

No próximo capítulo nós vamos estudar um método chamado de “método da resolução”, que irá resolver mais rapidamente certos casos, onde a aplicação do tableaux se mostra demorado.

O Método da resolução

- Forma Normal Conjuntiva (FNC)

  - Literal: Um literal é uma fórmula atômica ou a negação de uma fórmula atômica.

  - Cláusula: Uma cláusula é uma disjunção(v) de literais.

  - Forma Normal Conjuntiva: Uma fórmula está na forma normal conjuntiva se for uma conjunção(^) de cláusulas.

Equivalências Lógicas:

1) $\neg\neg\varphi \equiv \varphi$ 2) $\varphi \rightarrow \psi \equiv \neg\varphi \vee \psi$ 3) $\neg(\varphi \wedge \psi) \equiv (\neg\varphi)\vee(\neg\psi)$
4) $\neg(\varphi \vee \psi) \equiv (\neg\varphi)\wedge(\neg\psi)$ 5) $\varphi \wedge \varphi \equiv \varphi$ 6) $\varphi \vee \varphi \equiv \varphi$
7) $\varphi \vee (\neg\varphi) \equiv T$ 8) $\varphi \wedge (\neg\varphi) \equiv F$ 9) $\varphi \wedge F \equiv F$
10) $\varphi \vee F \equiv \varphi$ 11) $\varphi \wedge T \equiv \varphi$ 12) $\varphi \vee T \equiv T$

Exemplos:

I) $(L\rightarrow I) \wedge (\neg L\rightarrow M) \equiv (\neg L \vee I) \wedge (\neg\neg L \vee M) \equiv (\neg L \vee I) \wedge (L \vee M)$
II) $(I \vee M) \rightarrow C \equiv \neg(I \vee M) \vee C \equiv (\neg I \wedge \neg M) \vee C \equiv (C \vee \neg I) \wedge (C \vee \neg M)$

- Teorema: Para todo $\varphi \in$ PROP, existe $\varphi \in$ PROP tal que:
  I. φ é logicamente equivalente a φ';
  II. φ' está na FNC (Forma Normal Conjuntiva).

Esboço da Prova: (por indução sobre a complexidade de φ)

Base: φ é atômica

Casos Indutivos:

I) φ é da forma $\neg\psi$
Hipótese de indução: existe $\psi' \equiv \psi$ e ψ' na FNC
Tese: existe $\varphi' \equiv \varphi$ e φ' na FNC

Como ψ' está na FNC, $\neg\psi'$ pode ser colocada na FNC aplicando-se as equivalências lógicas, começando pelas leis de DeMorgan. O resultado será a φ que procuramos.

II) φ é da forma $(\rho \wedge \omega)$

Hipótese de indução: existe $\rho' \equiv \rho$ tal que ρ' está na FNC

Existe $\omega' \equiv \omega$ tal que ω' está na FNC

Tese: existe $\omega' \equiv \omega$ tal que ω' está na FNC

φ' vai ser $(\rho \wedge \omega')$ e já está na FNC.

- O Método da Resolução:

Consolidado num artigo de 1965 por John Alan Robinson, o método da resolução foi concebido com o propósito claro de tornar computacional – mente viável a solução algorítmica do problema SAT (e correlatos), mesmo que para uma classe restrita (e computacionalmente reconhecível) de sentenças.

Observações Chave:

1) Para uma fórmula na FNC ser insatisfatível, é suficiente que uma de suas cláusulas seja insatisfatível.
2) Uma fórmula na FNC do tipo: $(X \vee Y) \wedge (\neg X \vee Z) \wedge (\neg Y \vee \neg Z)$ é logicamente equivalente a: $(X \vee Y) \wedge (\neg X \vee Z) \wedge (Y \vee Z)$

(Prove a observação 2 através da tabela-verdade)

3) Para toda φ E PROP:

- φ é insatisfatível sse φ não é satisfatível.
- φ é insatisfatível sse $\neg\varphi$ é uma tautologia.
- φ é insatisfatível sse $\neg\varphi$ não é refutável.
- Um conjunto S de sentenças é satisfatível sse S não é insatisfatível (o conjunto de suas sentenças não é insatisfatível).
- φ é uma conseqüência lógica de $\{\psi_1, \psi_2, ..., \psi_n\}$ sse $\psi_1 \wedge \psi_2 \wedge ... \wedge \psi_n \wedge \neg\varphi$ é insatisfatível.

No exemplo do unicórnio:

- B é conseqüência lógica do conjunto S = {((L → ¬I) ^ (¬L → M)), ((I v M) → C), (C → ¬B)}?

Essa pergunta equivale a:
((L → ¬I) ^ (¬L → M)) ^ ((I v M) → C) ^ (C → ¬B) ^ (¬B) é insatisfativel?

O primeiro passo é mexer com a sentença ate chegar na FNC, e então nomeamos todas as cláusulas.

((L → ¬I) ^ (¬L → M)) ^ ((I v M) → C) ^ (C → ¬B) ^ (¬B)
≡ ((¬L v I) ^ (¬¬L v M)) ^ (¬(I v M) v C) ^ (¬C v B) ^ (¬B)
≡ (¬L v I) ^ (L v M) ^ (¬I ^ ¬M) v C) ^ (¬C v B) ^ (¬B)
≡ (¬L v I) ^ (L v M) ^ (¬I v C) ^ (¬M v C) ^ (¬C v B) ^ (¬B)
C_1 C_2 C_3 C_4 C_5 C_6

O segundo passo é criar novas cláusulas para ver se gera alguma contradição.

- C_1 ^ C_2 ≡ (I v M) (C_7)
- C_1 ^ C_3 ≡ (¬L v C) (C_8)
- C_2 ^ C_4 ≡ (L v C) (C_9)
- C_9 ^ C_5 ≡ (¬I v B) (C_{10})
- C_4 ^ C_5 ≡ (¬M v B) (C_{11})
- C_7 ^ C_{11} ≡ (I v B) (C_{12})
- C_{10} ^ C_{12} ≡ (B v B) ≡ B (C_{13})

Veja que as cláusulas de número C_6 e C_{13} geram uma contradição ((¬B) ^ (B)), pois eles geram uma cláusula C_{14} que é vazia. Daí nós vemos que a expressão é insatisfatível. E com isso nos mostramos que a expressão B é uma conseqüência lógica do conjunto S.

Sentenças Condicionais

(“Sentenças de Horn”, em homenagem a Alfred Horn, que as estudou cuidadosamente nos anos 1950).

Uma sentença condicional é uma sentença do tipo:
$(X_1 \wedge X_2 \wedge \ldots \wedge X_n) \rightarrow Y$, onde $X_1, X_2, \ldots, X_n$ e Y são sentenças atômicas.
Chamamos $X_1 \wedge X_2 \wedge \ldots \wedge X_n$ de corpo da Regra e Y de objetivo da regra. Observe que a sentença $(X_1 \wedge X_2 \wedge \ldots \wedge X_n) \rightarrow Y$ é equivalente a $\neg(X_1 \wedge X_2 \wedge \ldots \wedge X_n) \vee Y$.

Observação: O custo computacional (ou seja, o número máximo de testes e operações) do método da resolução é, em geral, da ordem de uma exponencial, i.e., O(2 elevado a n), onde n é o número de cláusulas. Porém, se a sentença de entrada for composta de cláusulas de Horn, o número de possibilidades é reduzido drasticamente, e fica da ordem de um valor constante k, multiplicado pelo número de cláusulas, i.e., O(n), onde n é o número de cláusulas. E para reconhecer uma sentença na FNC como constituída de apenas cláusulas de Horn, temos que visitar cláusula a cláusula e verificar se existe mais de um literal positivo. Daí se n for o número de variáveis, esse número de visitas/testes fica na ordem de n.k, ou seja O(n.k), que é baixo.

- Métodos Algorítmicos para resolver o “SAT”

1) Tabela Verdade:
   - Vantagens: Simplicidade, confiabilidade.
   - Desvantagens: Custo computacional.

2) Tableaux Analíticos:
   - Vantagens: Baseado num conceito intuitivo de “mundos possíveis”, regras são mais “sensatas”, às vezes mais eficientes.
   - Desvantagens: Nem sempre mais eficiente, e desprovido de método eficiente de descobrir os casos nos quais é de fato mais eficiente.

3) Resolução:
   - Vantagens: é demonstravelmente mais eficiente para casos que são eficientemente reconhecíveis (fórmulas de Horn).
   - Desvantagens: Nem sempre é mais eficiente e não é intuitivo.

- Lema: Seja φ uma fórmula da lógica proposicional. φ é uma tautologia sse φ é uma conseqüência lógica do conjunto vazio de premissas.

Prova:

(Ida) "Se φ é uma tautologia, então φ é conseqüência lógica de {}"

Hipótese de indução: φ é uma tautologia.

Tese: φ é uma conseqüência lógica de {}

Como φ é uma tautologia, então qualquer v satisfaz φ. Também qualquer v satisfaz {}. Logo, para toda valoração v que satisfaz {}, v também satisfaz φ. Daí, φ é conseqüência lógica de {}.

(Volta) "Se φ for conseqüência lógica de {}, então φ é tautologia"

Hipótese de indução: φ é conseqüência lógica de {}.

Tese: φ é uma tautologia.

Como φ é conseqüência lógica de {}, para toda valoração v que satisfaz {}, v também satisfaz φ. Mas qualquer valoração satisfaz φ. Logo, qualquer valoração também satisfaz φ.

Daí φ é uma tautologia.

Método da Dedução Natural