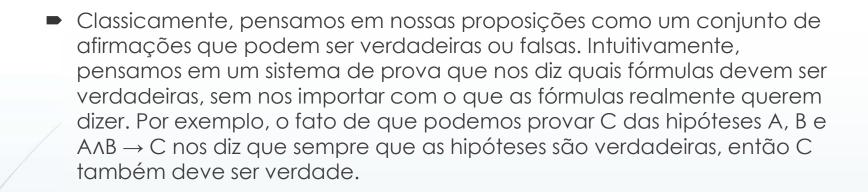
Capitúlo 6 – Semântica da Lógica Proposicional



Para que isto faça mais sentido precisamos saber sobre quais condições uma fórmula proposicional é verdadeira. Fórmulas e provas formais são noções sintáticas, isto é, são representadas por símbolos e estruturas simbólicas. A verdade é uma noção semântica, na medida em que atribui um tipo de significado a certas fórmulas.



- Dado um conjunto de hipóteses, Γ, e uma fórmula, A, podemos derivar A de Γ?
- Que fórmulas podem ser derivadas de Γ?
- Que hipóteses são necessárias para derivar A?
- Semanticamente, consideramos o seguinte:
- Dada a atribuição de "valores de verdade" às variáveis que compõem a fórmula A, A será verdadeiro ou falso?
- Existe alguma atribuição de verdade que faz A verdade?
- Quais são as atribuições de verdade que fazem A verdade?

6.1. Valores de verdade e atribuições

- Antes de mais nada, precisamos de uma noção para valor verdade. Valor verdade pode ser considerado como a relação de uma proposição com "a verdade". Os dois valores considerados até o momento foram "verdade" e "falso". Usaremos os símbolos T e F para representá-los informalmente; Eles representam os valores que T e ⊥ representam em dedução natural. No LEAN, usaremos true ou false.
- A próxima noção que precisamos é a de uma atribuição verdade, que é uma função que atribui um valor de verdade a cada variável pertencente a um conjunto de variáveis. Nomeamos as variáveis de P, Q, R, ... e as fórmulas arbitrárias de A, B, C, Por exemplo, a função v, que é uma atribuição verdade ao conjunto {P, Q, R, S} é definida da seguinte maneira:

• $v(P) := \mathbf{T}$

• $v(Q) := \mathbf{F}$

• $v(R) := \mathbf{F}$

v(S) := T

- Intuitivamente, uma atribuição de verdade descreve um possível "estado do mundo", ou seja, se uma proposição é válida naquele mundo atual.
- Uma vez que temos uma atribuição verdade v para um conjunto de variáveis proposicionais, podemos estendê-lo a uma **função de valoração** \bar{v} que atribui um valor de verdadeiro ou falso a cada fórmula proposicional, dependendo do valor verdade que adotamos para nossas variáveis.
- Esta função é definida de forma recursiva, ou seja, as fórmulas são avaliadas de baixo para cima, de modo que o valor atribuído a uma fórmula composta é determinado pelos valores atribuídos aos seus componentes. \bar{v} está definida da seguinte maneira:
 - $\bar{v}(\top) = \mathbf{T}$
 - $\bar{v}(\perp) = \mathbf{F}$
 - \(\bar{v}(\ell) = v(\ell)\), where \(\ell\) is any propositional variable.
 - $\bar{v}(\neg A) = \mathbf{T}$ if $\bar{v}(A)$ is \mathbf{F} , and vice versa.
 - v̄(A ∧ B) = T if v̄(A) and v̄(B) are both T, and F otherwise.
 - $\bar{v}(A \vee B) = \mathbf{T}$ if at least one of $\bar{v}(A)$ and $\bar{v}(B)$ is \mathbf{T} ; otherwise \mathbf{F} .
 - \$\bar{v}(A \to B) = \mathbf{T}\$ if either \$\bar{v}(B)\$ is \$\mathbf{T}\$ or \$\bar{v}(A)\$ is \$\mathbf{F}\$, and \$\mathbf{F}\$ otherwise. (Equivalently, \$\bar{v}(A \to B) = \mathbf{F}\$ if \$\bar{v}(A)\$ is \$\mathbf{T}\$ and \$\bar{v}(B)\$ is \$\mathbf{F}\$, and \$\mathbf{T}\$ otherwise.)

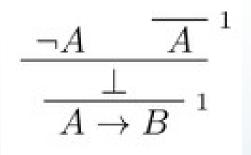
- As regras de disjunção e conjunção são fáceis de entender. A regra da implicação pode ser um pouco mais complicada, mas como visto antes, ela só será falsa se a primeira proposição for verdadeira e a segunda for falsa. Olhe para estes exemplos:
- "Se eu tiver duas cabeças, então os círculos são quadrados".
- "Se eu tivesse duas cabeças, então círculos seriam quadrados".
- A segunda frase é um exemplo de uma implicação contrafatual. Ele afirma algo sobre como o mundo poderia ser, se as coisas fossem diferentes do que realmente são. A implicação material afirma algo sobre o modo como o mundo é agora, e não o modo como poderia ter sido. Como é falso que eu tenha duas cabeças, a afirmação "se eu tiver duas cabeças, então círculos são quadrados" é verdade.

- Observação: Se B é verdadeiro, podemos provar que A → B sem saber nada sobre A.
- → A regra de introdução da implicação nos diz: dado B, podemos concluir A
 → B e depois cancelar a suposição de A, se esta tiver sido feita. Se não utilizarmos A na confecção da prova, esta ainda é válida, só mais fraca.

```
variables A B : Prop
variable hB : B

example : A → B :=
assume hA : A,
show B, from hB
```

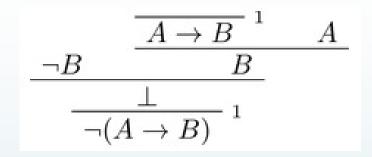
Da mesma forma, se A é falso, podemos provar A → B sem saber nada sobre B:



```
variables A B : Prop
variable hnA : ¬ A

example : A → B :=
assume hA : A,
show B, from false.elim (hnA hA)
```

■ Sabendo que A é verdadeiro e B é falso, podemos provar \neg (A \rightarrow B):



```
variables A B : Prop
variable hA : A
variable hnB : ¬B

example : ¬ (A → B) :=
assume h : A → B,
have hB : B, from h hA,
show false, from hnB hB
```

Dada uma fórmula qualquer, podemos tentar encontrar uma "atribuição de verdade" que nos dirá se ela é verdadeira ou falso. Podemos utilizar o LEAN para avaliar estas fórmulas. No exemplo a seguir, iremos verificar que podemos atribuir qualquer conjunto de valores aos símbolos proposicionais A, B, C, D e E. Ao utilizarmos o comando eval em alguma entrada, o LEAN declarará o valor da expressão.

```
-- Definimos os valores verdade aqui

def A := tt

def B := ff

def C := tt

def D := tt

def E := ff

def test (p : Prop) [decidable p] : string := if p then "verdadeiro" else "falso"

#eval test ((A ∧ B) V ¬ C)

#eval test (A → D)

#eval test (C → (D V ¬E))

#eval test (¬(A ∧ B ∧ C ∧ D))
```

```
    [Lean] "falso" (11, 1)
    [Lean] "verdadeiro" (12, 1)
    [Lean] "verdadeiro" (13, 1)
    [Lean] "verdadeiro" (14, 1)
```

6.2 - Tabela Verdade

- Usaremos algo chamado de tabela de verdade para descobrir quando uma fórmula é ou não é é verdadeira. No lado esquerdo da tabela de verdade, colocamos todas as possíveis atribuições de verdade para as presentes "letras" proposicionais. No lado direito, colocaremos o valor de verdade de toda a fórmula sob a atribuição correspondente.
- Aqui estão as tabelas verdade de nossos conectivos lógicos mais básicos:

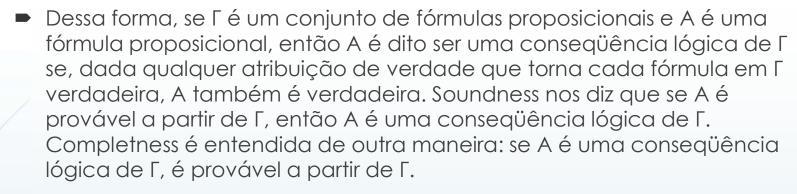
$B \mid A \wedge B$	$A \mid B \mid$	Į
TT	$\mathbf{T} \mid \mathbf{T}$	
F F	TF	
T F	FT	
F F	F F	
F F T	T F F T	

$\vee B$	A	B	$A \rightarrow B$
\mathbf{T}	\mathbf{T}	\mathbf{T}	T
T	\mathbf{T}	F	F
\mathbf{T}	\mathbf{F}	T	T
F	\mathbf{F}	F	T

 As vezes, escrever os valores verdade de uma sub-fórmula de uma fórmula maior pode nos ajudar.

6.3. Soundness and Completeness

- Uma fórmula proposicional é provável se existe uma prova dela em algum sistema dedutivo escolhido.
- Uma fórmula proposicional é dita ser uma tautologia, ou válida, se for verdadeira independente da atribuição verdade.
- "Provabilidade" é uma noção sintática que afirma a existência de uma prova para uma fórmula.
- A validade é uma noção semântica, na medida em que tem a ver com atribuições de verdade e avaliações.
- A afirmação de que cada fórmula provável é válida é conhecida como soundness, e a afirmação de que podemos provar todas as fórmulas válidas é conhecida como completeness.



Com as regras de dedução natural, uma fórmula A é provável a partir de um conjunto de hipóteses {B1, B2, ..., Bn} se e somente se a fórmula B1∧B2∧ ··· ∧Bn → A for provável sem precisar de hipóteses. Então, pelo menos, para conjuntos finitos de fórmulas Γ, as duas declarações de soundness e completeness são equivalentes.

- Provar soundness é mais fácil que provar completeness. Queremos mostrar que sempre que A é provável a partir de um conjunto de hipóteses Γ, então A é uma consequência lógica de Γ.
- No caso da dedução natural, basta mostrar que soundness possui as provas mais básicas e esta é preservada sob cada regra de inferência.
- O caso base é fácil: a regra de suposição diz que A é provável a partir da hipótese A, e claramente toda atribuição verdade que faz A true faz A true. Os passos indutivos envolvem verificar se as regras de inferência que escolhemos condizem com os valores semânticos que temos.
- Por exemplo, suponha que a última regra seja a regra and introduction. Nesse caso, temos uma prova de A a partir de Γ e uma provade B a partir de Δ . Quando as combinamos, formamos uma prova de AAB a partir das hipóteses em Γ U Δ . Indutivamente, podemos assumir que A é uma consequência lógica de Γ e que B é uma consequência lógica de Δ . Seja v atribuição verdade que torne cada fórmula em Γ U Δ verdadeira. Com isso, A será True e B também será True. Pela definição da função de avaliação, \bar{v} [AAB] = T.

- Provar completeness é mais difícil. Basta mostrar que se A é qualquer tautologia, então A é provável. Uma estratégia é mostrar que a dedução natural pode simular o método das tabelas de verdade.
- Por exemplo, suponha que A seja construído a partir de variáveis proposicionais B e C. Então, em dedução natural, devemos ser capazes de provar a fórmula abaixo, com uma parte da disjunção para cada linha da tabela verdade. Feito isso, podemos utilizar cada pedaço da disjunção para avaliar cada expressão que compõe A de acordo com os valores verdade atribuídos, até termos uma prova de A concreta.

$$(B \land C) \lor (B \land \neg C) \lor (\neg B \land C) \lor (\neg B \land \neg C)$$