

IntroProof

O aluno João não é um string;
nem é um nome; é uma pessoa mesmo!

João : Person "João" : String João : Name

João tem 3 filhos : ~~Bool~~ Prop

João tem Brasil filhos

← verdadeiro ou falso?
Nenhum dos dois!
É tão errado (type error)
que nem chega perto de ter
chances de ser falso!

Type error, type-check

João tem _____ filhos : $\text{Nat} \rightarrow \text{Prop}$

operação binária no Int

domínio

codomínio

Int : Type

Prop : Type

2 + 4 : Int true OR true : Bool

Int : Type

$$\text{Nat} \rightarrow \text{Prop} : \text{Type}$$

↳ operação binária no Bool

↳ inferimos essa tipagem sem calcular "o valor" de $2 + 4$.

PALAVRINHAS

Objeto vs Prop

Bool vs Prop

Static vs Dynamic

Dados vs Alvo

Usar vs Atacar

Comando vs Comentário

Atômico vs Composto

lec01

2025-08-18

Prop

Como saber se algo é uma Prop?

Dica: veja se faz sentido dizer:

Suponha (que) P ...

Eu acho que P ...

Eu vou demonstrar P ...

Quantos tipos?

ou qualquer outro tipo
atômico que apareceu até
agora: Person, String, ...

Por causa do $(\rightarrow)$ e do Nat já é garantido
que temos uma infinidade de tipos:

Nat , $\text{Nat} \rightarrow \text{Nat}$, $\text{Nat} \rightarrow (\text{Nat} \rightarrow \text{Nat})$, $\text{Nat} \rightarrow (\text{Nat} \rightarrow (\text{Nat} \rightarrow \text{Nat}))$, ...

hw: descreva os habitantes desses tipos

Construindo Props (conectivos lógicos)

-elim
Como usar

-intro
Como atacar

comandos

Split.

Esc-L
Esc-R.

Suponha P.

Para cada comando precisamos saber:

- quando é executável/permitido

- qual o seu efeito
no tabuleiro DADOS-ALVO

&

?

ou

?

$\Rightarrow$

?

$\Leftrightarrow$

?

$\neg$

?

=

?

$\forall$

?

$\exists$

?

⋮

?

?

?

?

?

Split

DEMONS

⋮
Split.

DADOS

⋮
P : Prop
Q : Prop
⋮

ALVO

P & Q



DADOS

⋮
P : Prop
Q : Prop
⋮

ALVO

P

DADOS

⋮
P : Prop
Q : Prop
⋮

ALVO

Q



precisamos "matar" ambos

Esc-L

DEMONS

DADOS

ALVO

⋮
Esc-L.

⋮
P : Prop
Q : Prop
⋮

P ou Q



DADOS

ALVO

⋮
P : Prop
Q : Prop
⋮

P

Esc-R

DEMONS

DADOS

ALVO

⋮
Esc-R.

⋮
P : Prop
Q : Prop
⋮

P ou Q



DADOS

ALVO

⋮
P : Prop
Q : Prop
⋮

Q

Suponha —.

a única coisa que posso supor é a premissa da implicação do alvo

DEMONS

Suponha P

DADOS

P : Prop
Q : Prop
⋮

ALVO

P $\Rightarrow$ Q

Concelho perigoso:

Dá a impressão que tem uma coisa proibida que não pode supor.

« não pode supor teu alvo »

A verdade: Só pode supor algo quando teu alvo é uma implicação $P \Rightarrow Q$ e até nesse caso a única coisa que pode supor é o P.

$P \Rightarrow Q$ é como uma promessa:

« Prometo Q se P. »

« Dado P, consigo Q. »

É uma promessa condicional:
eu não prometi Q se não tiver P

(P é uma Prop) $\rightarrow$

P : Prop
Q : Prop
⋮

Q

(P é verdadeiro) $\rightarrow$

P

por que escrevi em cor laranja?

Igualdade vs Equivalência

PALAVRINHAS

lec02
2025-08-20

igualdade vs equivalência

intensão vs extensão

proposição vs comando

$\equiv / \Leftrightarrow$ vs $\stackrel{\text{def}}{=} / \stackrel{\text{def}}{\Leftrightarrow}$

$\underbrace{\quad}_{: \alpha \text{ Obj}} =_{\alpha} \underbrace{\quad}_{: \alpha \text{ Obj}} : \text{Prop}$

$\underbrace{\quad}_{\text{Prop}} \Leftrightarrow \underbrace{\quad}_{\text{Prop}} : \text{Prop}$

Exemplos:

"hi" $\stackrel{\text{String}}{=}$ "oi" $\leftarrow$ uma proposição falsa $\rightarrow$ "hi" palíndromo $\Leftrightarrow$ "oi" substring de "boi"

42 $\stackrel{\text{Int}}{=} 7 \cdot 6$ $\leftarrow$ uma proposição verdadeira $\rightarrow$ 7 par $\Leftrightarrow$ "hi" palindromo

"hi" $\stackrel{?}{=} 42$ $\leftarrow$ TYPE ERROR!

3 primo $\Leftrightarrow 7$

Intensão vs Extensão

Ismênia $\underset{\text{Person}}{=}$ a diretora do IMD

Ismênia $\neq$ a diretora do IMD

↳ outra coisa ser a Ismênia,
outra coisa ser a diretora do IMD!

$\{2\} \underset{\text{Set Int}}{=} \{\text{os primos pares}\}$

$\{2\} \neq \{\text{os primos pares}\}$

O que significa $x \in \{2\}$?

Significa: $x = 2$

O que significa $x \in \{\text{os primos pares}\}$?

Significa: x primo & x par.

Definições

Def. $\text{— par} : \text{Int} \rightarrow \text{Prop}$

Seja $n : \text{Int}.$: Cmd

~~Seja $k : \text{Int}.$~~

↖ não!

$n \text{ par} \iff (\forall k : \text{Int}) [n = 2k]$

DADOS

$n : \text{Int}$

~~$k : \text{Int}$~~

Tem como uma definição de predicado ser errada?

Sim! Se discorda do nosso ♥:

- aceita algo que não deveria;
- não aceita algo que deveria.

$3 \text{ par} \iff (\forall k : \text{Int}) [3 = 2k]$
(concordou com o ♥ (tome $k := 1$ para refutar))

$4 \text{ par} \iff (\forall k : \text{Int}) [4 = 2k]$
(discordou (tome $k := 1$ para refutar))

finalmente corrigimos assim:

$$n \text{ par} \stackrel{\text{def}}{\iff} (\exists k : \text{Int}) [n = 2k] : \text{Cmd}$$

↑
agora sim!

Mais uma definição:

$$_{}^2 : \text{Int} \rightarrow \text{Int}$$

$$x^2 \stackrel{\text{def}}{=} x \cdot x$$

Assim temos: $3^2 \equiv 3 \cdot 3 = 9$

Tem como uma def ser errada?

Sim! Novamente: se discorda do ♥

Desafios:

Seja $x : _$. : $Type \rightarrow Cmd$

Seja $_ : Int$. : $Var \rightarrow Cmd$

Teoremas e demonstrações

Θ. Seja $x : \text{Int}$.

Suponha x par.

Demonstre: $(\forall y : \text{Int}) [y \text{ par} \Rightarrow x \cdot y + 2 \text{ par}]$

DEMONS.

Seja $w : \text{Int}$.

Suponha w par.

⋮

?

DADOS

$x : \text{Int}$

$w : \text{Int}$

x par

w par

ALVO

~~$(\forall y : \text{Int}) [\dots]$~~

~~$w \text{ par} \Rightarrow x \cdot w + 2 \text{ par}$~~

$x \cdot w + 2 \text{ par}$



$(\exists k : \text{Int}) [x \cdot w + 2 = 2k]$

Conectivos lógicos

-elim
Como usar

-intro
Como atacar

$P \& Q$?

Split.

$P \text{ ou } Q$?

Esc-L
Esc-R.

$P \Rightarrow Q$?

Suponha P.

ilusão de opção/escolha

$P \Leftrightarrow Q$?

?

$\neg P$?

?

$u =_{\alpha} v$?

?

$(\forall x : \alpha) [\varphi(x)]$?

Seja : α .

de fato precisamos escolher (um nome válido e bom!) ?

$(\exists x : \alpha) [\varphi(x)]$?

?

⋮

Seja $_ : _$. quais nome são válidos?

DEMONS

Seja w : α .

DADOS

$\vdots$
 ϕ : ?
 $\vdots$

ALVO

$(\forall x : \alpha) [\phi(x)]$



DADOS

$\vdots$
 ϕ : ?
 $\vdots$
 $w : \alpha$

ALVO

$\phi(w)$

Regras de inferência

lec03
2025-08-22

$$\frac{\alpha : \text{Type} \quad \beta : \text{Type}}{\alpha \rightarrow \beta : \text{Type}} \quad (\rightarrow)\text{-Formation}$$

forma geral das regras:

$$\frac{J_1 \quad \dots \quad J_n}{J} \quad \text{label}$$

↑
linha de inferência

← n pode ser 0 premissas

← para referir à regra

← exatamente uma conclusão

$$\frac{}{\text{Person} : \text{Type}} \quad \text{Person-F}$$

$$\frac{}{\text{Nat} : \text{Type}} \quad \text{Nat-F} \quad \dots$$

Tuplas

$$\frac{\alpha : \text{Type} \quad \beta : \text{Type}}{\alpha \times \beta : \text{Type}} \quad (x)\text{-F}$$

Como **construo** habitantes (elementos) do $\alpha \times \beta$?

$$\frac{a : \alpha \quad b : \beta}{\langle a, b \rangle : \alpha \times \beta} \quad (x)\text{-I} \quad \text{Introdução}$$

$\uparrow$
tupla

Como **uso** habitantes do $\alpha \times \beta$?

Notações alternativas:

$w.l$	$\text{outl } w$	$\pi_1 w$	$\text{proj}_1 w$	$\text{fst } w$
$w.r$	$\text{outr } w$	$\pi_2 w$	$\text{proj}_2 w$	$\text{snd } w$

$$\frac{w : \alpha \times \beta}{w.l : \alpha} \quad (x)\text{-E}_L \quad \text{Eliminação}$$

$\uparrow$
Left

$\uparrow$
projectão

$$\frac{w : \alpha \times \beta}{w.r : \beta} \quad (x)\text{-E}_R$$

$\uparrow$
Right

Exemplo

$$\frac{\frac{\text{Person} : \text{Type}}{\text{Person-f}} \quad \frac{\frac{\text{Nat} : \text{Type}}{\text{Nat-f}} \quad \frac{\text{String} : \text{Type}}{\text{Str-f}}}{\text{Nat} \times \text{String} : \text{Type}}_{(\times)\text{-f}}}{\text{Person} \rightarrow (\text{Nat} \times \text{String}) : \text{Type}}_{(\rightarrow)\text{-f}}$$

Abstraindo com buracos

João tem 3 filhos : Prop

João tem _ filhos : $\text{Nat} \rightarrow \text{Prop}$

_ tem 3 filhos : $\text{Person} \rightarrow \text{Prop}$

_ tem _ filhos : $\text{Person} \rightarrow (\text{Nat} \rightarrow \text{Prop})$

ou. ? $(\text{Person} \times \text{Nat}) \rightarrow \text{Prop}$

ou. ? $\text{Nat} \rightarrow (\text{Person} \rightarrow \text{Prop})$

Precedência sintáctica

O que significa $\alpha \times \beta \rightarrow \gamma$? Nada!

Tanto $(\alpha \times \beta) \rightarrow \gamma$ quanto $\alpha \times (\beta \rightarrow \gamma)$ tem significado.

aquela função que recebendo...

retorna...

$(w \mapsto \text{age } (w.l) + \text{length } (w.r)) : (\text{Person} \times \text{String}) \rightarrow \text{Nat}$

$\langle \text{João}, \text{length} \rangle$

$: \text{Person} \times (\text{String} \rightarrow \text{Nat})$

Atribuímos ao $_ \times _$ precedência sintáctica maior que a do $_ \rightarrow _$.

A partir desta linha, o $\alpha \times \beta \rightarrow \gamma$ ganhou significado:

$$\alpha \times \beta \rightarrow \gamma \equiv (\alpha \times \beta) \rightarrow \gamma$$

Conectivos lógicos

lec04
2025-08-29

(elim)
Como usar

(intro)
Como atacar

$P \ \& \ Q$

Ext-L $\frac{P \& Q}{P}$
Ext-R $\frac{P \& Q}{Q}$

$P \text{ ou } Q$

Separe em casos a partir de $\underline{P \text{ ou } Q}$

$P \Rightarrow Q$

App $\underline{P \Rightarrow Q}$ no $\underline{P}$ para obter $\underline{Q}$

Split.

Esc-L
Esc-R.

Suponha $\underline{P}$

$\neg P$

?

?

$u =_{\alpha} v$

...

...

$(\forall x : \alpha) [\varphi(x)]$

App $\underline{(\forall x : \alpha) [\varphi(x)]}$ no $\underline{a} : \alpha$ para obter $\underline{\varphi(a)}$

Seja $\underline{\quad} : \alpha$

$(\exists x : \alpha) [\varphi(x)]$

Seja $\underline{a} : \alpha$ tal que $\underline{\varphi(a)}$

Escolho $\underline{\quad} : \alpha$

--ALVO: $\varphi(a)$

inferível
 $\varphi : \alpha \rightarrow \text{Prop}$

Conectivos definidos (ou açucar sintáctico)

$$P \Leftrightarrow Q \stackrel{\text{def}}{\iff} (P \Rightarrow Q) \ \& \ (Q \Rightarrow P)$$

$$P \text{ ou } Q \stackrel{\text{def}}{\iff} (P \ \& \ (\neg Q)) \text{ ou } (Q \ \& \ (\neg P))$$

? $\Uparrow$ $\Downarrow$?

$$(P \text{ ou } Q) \ \& \ \neg (P \ \& \ Q)$$

$$P \Leftarrow Q \stackrel{\text{def}}{\iff} Q \Rightarrow P \quad \exists_{\geq 1}$$

$$(\forall x \in A) [\phi(x)]$$

$$(\forall x : \alpha) [x \in A \Rightarrow \phi(x)]$$

$$(\exists x \in A) [\phi(x)]$$

$$(\exists x) [x \in A \ \& \ \phi(x)]$$

$\exists_{\leq 1}$

$$(\exists! x : \alpha) [\phi(x)] \stackrel{\text{def}}{\iff} (\exists x : \alpha) [\phi(x)] \ \& \ (\forall u, v : \alpha) [\phi(u) \ \& \ \phi(v) \Rightarrow u = v]$$

$\uparrow$ existe único $x : \alpha$ tal que $\phi(x)$

$$(\forall x : \alpha \ \psi) [\phi(x)] \stackrel{\text{def}}{\iff} (\forall x : \alpha) [\psi(x) \Rightarrow \phi(x)]$$

$\uparrow$ todo x legal ϕ ia

$\phi : \alpha \rightarrow \text{Prop}$

$\psi : \alpha \rightarrow \text{Prop}$

$$(\exists x : \alpha \ \psi) [\phi(x)] \stackrel{\text{def}}{\iff} (\exists x : \alpha) [\psi(x) \ \& \ \phi(x)]$$

$\uparrow$ existe x legal que ϕ ia

Regras de inferência : F, I, E

$$\frac{P \Rightarrow Q \text{ true} \quad P \text{ true}}{Q \text{ true}} (\Rightarrow)\text{-E} \quad \text{— como usar}$$

$$\frac{P \text{ true} \quad Q \text{ true}}{P \& Q \text{ true}} (\&)\text{-I} \quad \text{— como atacar}$$

$$\frac{f : \alpha \rightarrow \beta \quad a : \alpha}{f a : \beta} \quad ?$$

$$\frac{P \text{ prop} \quad Q \text{ prop}}{P \& Q \text{ prop}} (\&)\text{-F} \quad \text{— como formar}$$

$$\frac{P \text{ prop} \quad Q \text{ prop}}{P \Rightarrow Q \text{ prop}} (\Rightarrow)\text{-F}$$

... e os outros?

Conectivos definidos

lec05

2025-09-01

$$P \iff Q \stackrel{\text{def}}{\iff} (P \Rightarrow Q) \ \& \ (Q \Rightarrow P)$$

$$P \text{ ou } Q \stackrel{\text{def}}{\iff} (P \ \& \ (\neg Q)) \text{ ou } (Q \ \& \ (\neg P))$$

$$P \Leftarrow Q \stackrel{\text{def}}{\iff} Q \Rightarrow P$$

$$(\exists! x : \alpha) [\varphi(x)] \stackrel{\text{def}}{\iff} \underbrace{(\exists x : \alpha) [\varphi(x)]}_{\exists \geq 1 \text{ (existência)}} \ \& \ \underbrace{(\forall u, v : \alpha) [\varphi(u) \ \& \ \varphi(v) \Rightarrow u = v]}_{\exists \leq 1 \text{ (unicidade)}}$$

↑ existe único $x : \alpha$ tal que $\varphi(x)$

$$(\forall x : \alpha \ \psi) [\varphi(x)] \stackrel{\text{def}}{\iff} (\forall x : \alpha) [\psi(x) \Rightarrow \varphi(x)]$$

↑ todo x legal φ ia

$\varphi : \alpha \rightarrow \text{Prop}$

$\psi : \alpha \rightarrow \text{Prop}$

$$(\exists x : \alpha \ \psi) [\varphi(x)] \stackrel{\text{def}}{\iff} (\exists x : \alpha) [\psi(x) \ \& \ \varphi(x)]$$

↑ existe x legal que φ ia

Casos especiais onde $\psi(x) :\Leftrightarrow x \in A$

$$(\forall x \in A)[\phi(x)] \iff (\forall x : \alpha)[x \in A \Rightarrow \phi(x)]$$

$$(\exists x \in A)[\phi(x)] \iff (\exists x : \alpha)[x \in A \ \& \ \phi(x)]$$

Conectivos lógicos

	(elim) Como usar	(intro) Como atacar
$P \ \& \ Q$	Ext-L $\frac{P \ \& \ Q}{P}$ Ext-R $\frac{P \ \& \ Q}{Q}$	Split.
$P \text{ ou } Q$	Separe em casos a partir de $\underline{P \text{ ou } Q}$.	Esc-L Esc-R.
$P \Rightarrow Q$	App $\underline{P \Rightarrow Q}$ no $\underline{P}$ para obter $\underline{Q}$.	Suponha $\underline{P}$.

$\perp$ ← bottom

Contradição. ←
EX FALSO QUODLIBET

(nada)

$u =_{\alpha} v$

...

...

$(\forall x : \alpha) [\varphi(x)]$ App $\underline{(\forall x : \alpha) [\varphi(x)]}$ no $\underline{a} : \alpha$ para obter $\underline{\varphi(a)}$.

Seja $\underline{\quad} : \alpha$.

$(\exists x : \alpha) [\varphi(x)]$ Seja $\underline{a} : \alpha$ tal que $\underline{\varphi(a)}$.

Escolho $\underline{\quad} : \alpha$.

--ALVO: $\varphi(a)$

inferível
 $\varphi : \alpha \rightarrow \text{Prop}$

Negação

$$\neg P \overset{\text{as}}{\iff} P \Rightarrow \perp$$

«refutar P » significa «demonstrar $\neg P$ »

"Leis de lógica": comutatividade de (&)

Sejam $P, Q : \text{Prop}$.

Demonstre: $P \& Q \Rightarrow Q \& P$.

DEMONS

Suponha $pq : P \& Q$. ^{(1) Pq}

Ext-R ^{(1) Pq} para obter Q . ^{(2) q}

Split

Parte-L:

Imediato ^{(2) q} .

Parte-R:

Ext-L ^{(1) Pq} para obter P . ^{(3) P}

Imediato ^{(3) P} .

DADOS

$P : \text{Prop}$

$Q : \text{Prop}$

$pq : P \& Q$

$q : Q$

ALVO

~~$(P \& Q) \Rightarrow (Q \& P)$~~

~~$Q \& P$~~

DADOS

$P : \text{Prop}$

$Q : \text{Prop}$

$pq : P \& Q$

$q : Q$

ALVO

Q

DADOS

$P : \text{Prop}$

$Q : \text{Prop}$

$pq : P \& Q$

$q : Q$

$p : P$

ALVO

P

Dupla negação

lec06
2025-09-03

$$P : \text{Prop} \vdash P \Rightarrow \neg \neg P$$

DEMONS

Suponha P .

Suponha $\neg P$.

App $\neg P$ no P para obter $\perp$.
 $P \Rightarrow \perp$

Contradição. $\blacksquare$

DADOS

P

$$\neg P \Leftrightarrow P \Rightarrow \perp$$

$\perp$

ALVO

~~$$P \Rightarrow \neg \neg P$$~~

$$\neg \neg P$$



$$\neg P \Rightarrow \perp$$



$$(P \Rightarrow \perp) \Rightarrow \perp$$

$\perp$

$$\vdash P \Leftarrow \neg\neg P$$

DEMONS

Suponha $\neg\neg P$.

Vou demonstrar: $\neg P$

Sup P .

(Devemos desistir!)

$$\neg\neg P \Rightarrow \perp$$

App $\neg\neg P$ no $\neg P$.

Contradição.

DADOS

$\neg\neg P$

P

ALVO

~~$\neg P$~~

$\perp$

DADOS

$\neg\neg P$



$$\neg P \Rightarrow \perp$$

$\neg P$

ALVO

~~$P \Leftarrow \neg\neg P$~~

P

LEM $\leftarrow$ magia ☆

LEM : $(\forall P : \text{Prop}) [P \vee \neg P]$

LEM[P] : $P \vee \neg P$

Com palavras:

« Para qualquer proposição P , temos $P \vee \neg P$. »

Como esquema de regra de inferência:

$$\frac{}{P \vee \neg P} \text{LEM}[P]$$

Como comando:

☆ LEM P.

-- DADOS: ..., $P \vee \neg P$

☆ - contaminação contagiosa!

$\vdash \neg\neg(P \vee \neg P)$ A irrefutabilidade do LEM[P]

DEMONS

Sup $\neg(P \vee \neg P)$. -- ALVO: $\perp$

Vou demons: $P \vee \neg P$

Esc-R.

Sup P .

Vou demons: $P \vee \neg P$

Esc-L.

Imediato.

-- DADOS: ..., $P \vee \neg P$

App $\neg(P \vee \neg P)$ no $P \vee \neg P$.

Contradição.

-- DADOS: ..., $P \vee \neg P$

App $\neg(P \vee \neg P)$ no $P \vee \neg P$.

Contradição. $\blacksquare$

DADOS

$\neg(P \vee \neg P)$

P

$P \vee \neg P$

$\perp$

ALVO

~~$P \vee \neg P$~~

~~$\neg P$~~

$\perp$

DADOS

$\neg(P \vee \neg P)$

P

ALVO

~~$P \vee \neg P$~~

P

ALVO

$\neg\neg(P \vee \neg P)$

$\neg(P \vee \neg P) \Rightarrow \perp$

$\perp$

DADOS

$\neg(P \vee \neg P) \Leftrightarrow (P \vee \neg P) \Rightarrow \perp$

$P \vee \neg P$

$\perp$