

Semântica e Gramática Gerativa

Aula 2

Marcelo Ferreira
ferreira10@gmail.com

Universidade de São Paulo

USP, 22 de Agosto de 2012

Semântica e Gramática Gerativa

Aula 2

Marcelo Ferreira
ferreira10@gmail.com

Universidade de São Paulo

USP, 22 de Agosto de 2012

De Conjuntos para Funções

- ▶ Na aula passada: verbos intransitivos denotavam conjuntos.

$$\llbracket \text{trabalha} \rrbracket = \{x: x \text{ trabalha}\}$$

De Conjuntos para Funções

- ▶ Na aula passada: verbos intransitivos denotavam conjuntos.
 $[[\text{trabalha}]] = \{x: x \text{ trabalha}\}$
- ▶ A partir de hoje: verbos intransitivos (e vários outros tipos de constituintes sintáticos) denotarão **funções**.
- ▶ Conforme veremos, lidar com funções nos trará um poder de generalização imenso.

Funções

- ▶ Funções: dispositivos que mapeiam elementos de um conjunto (domínio) em elementos de um outro conjunto (contra-domínio)

Função Sucessor

- ▶ Mapeia números naturais em números naturais.

Para todo $n \in \mathbb{N}$, $S(n) = n + 1$

$S(0) = 1$, $S(1) = 2$, ...

Funções Características

- ▶ O domínio é um conjunto qualquer (p. ex \mathbb{N}) e o contradomínio é o conjunto $\{0,1\}$

$$f(x) = \begin{cases} 1 & \text{se } x \text{ é par} \\ 0 & \text{se } x \text{ é ímpar} \end{cases}$$

- ▶ Essa função *caracteriza* o conjunto dos números pares.

Notação *Lambda* - λ

► Função Sucessor

$$\lambda x : \underbrace{x \in \mathbb{N}}_{\text{domínio}}. \underbrace{x + 1}_{\text{valor}}$$

ou, se o contexto deixar claro o domínio da função

$$\lambda x. x + 1$$

Notação Lambda - λ

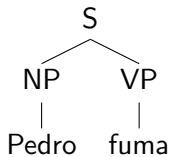
► Função Característica

$$\lambda x : \underbrace{x \in \mathbb{N}}_{\text{domínio}}. \underbrace{x \text{ é par}}_{\text{retorna o valor 1 se}}$$

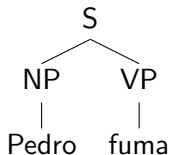
ou, se o contexto deixar claro o domínio da função

$$\lambda x. x \text{ é par}$$

Verbos Intransitivos



Verbos Intransitivos



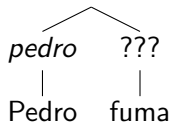
[[S]] = 1 sse Pedro fuma

[[Pedro]] = pedro

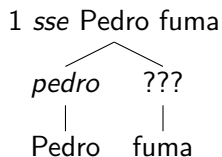
[[fuma]] = ???

Verbos Intransitivos

1 sse Pedro fuma

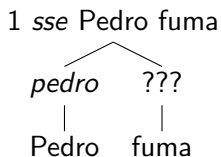


Verbos Intransitivos



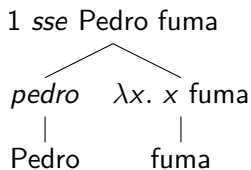
- ▶ Se a extensão de *fuma* for uma função, ela deve levar indivíduos em valores de verdade, de modo que um indivíduo x seja mapeado no valor 1 se x fuma, e no valor 0 se x não fuma.

Verbos Intransitivos



- ▶ Se a extensão de *fuma* for uma função, ela deve levar indivíduos em valores de verdade, de modo que um indivíduo x seja mapeado no valor 1 se x fuma, e no valor 0 se x não fuma.
- ▶ $\llbracket \text{fuma} \rrbracket = \lambda x. x \text{ fuma}$

Verbos Intransitivos



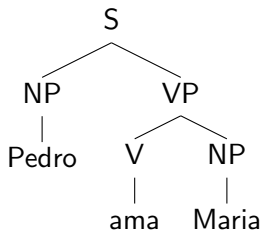
- ▶ $\llbracket S \rrbracket = \llbracket \text{fuma} \rrbracket(\llbracket \text{Pedro} \rrbracket)$

Predicação como Aplicação Funcional

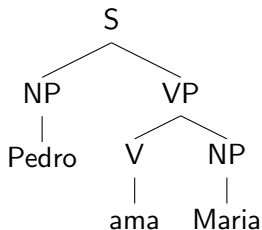
Aplicação Funcional

Seja α um nó ramificado, cujos constituintes imediatos são β e γ .
Se $\llbracket \beta \rrbracket$ é uma função e $\llbracket \gamma \rrbracket$ pertence ao domínio de $\llbracket \beta \rrbracket$, então $\llbracket \alpha \rrbracket = \llbracket \beta \rrbracket(\llbracket \gamma \rrbracket)$.

Verbos Transitivos



Verbos Transitivos



[[S]] = 1 sse Pedro ama Maria

[[Pedro]] = pedro

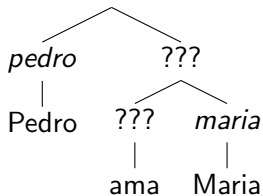
[[Maria]] = maria

[[ama]] = ???

[[VP]] = ???

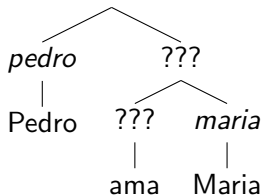
Verbos Transitivos

1 sse Pedro ama Maria



Verbos Transitivos

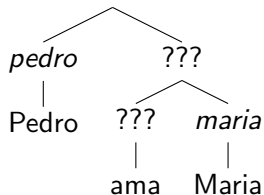
1 se Pedro ama Maria



- ▶ Se a extensão de *ama Maria* for uma função, ela deve levar indivíduos em valores de verdade, de modo que um indivíduo y seja mapeado no valor 1 se y ama Maria, e no valor 0 se y não ama Maria.

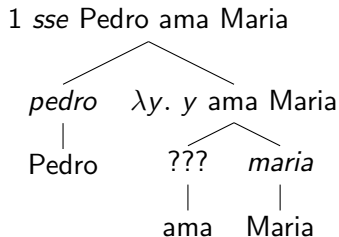
Verbos Transitivos

1 se Pedro ama Maria



- ▶ Se a extensão de *ama Maria* for uma função, ela deve levar indivíduos em valores de verdade, de modo que um indivíduo y seja mapeado no valor 1 se y ama Maria, e no valor 0 se y não ama Maria.
- ▶ $\llbracket \text{ama Maria} \rrbracket = \lambda y. y \text{ ama Maria}$

Verbos Transitivos

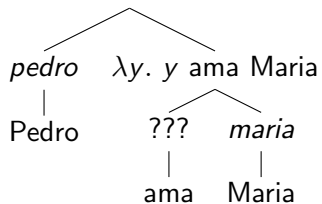


► $\llbracket S \rrbracket = \llbracket \text{ama Maria} \rrbracket(\llbracket \text{Pedro} \rrbracket)$

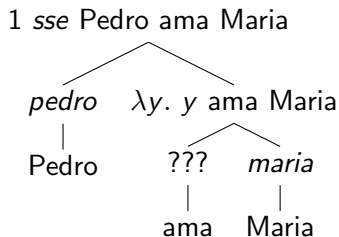
(Aplic. Func.)

Verbos Transitivos

1 sse Pedro ama Maria

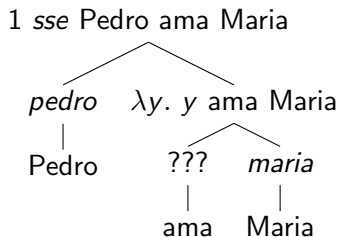


Verbos Transitivos



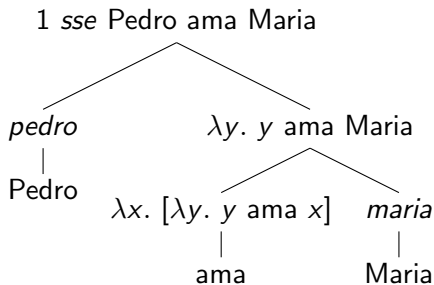
- ▶ Se a extensão de *ama* for uma função, ela deve levar indivíduos em funções, de modo que um indivíduo x seja mapeado em uma função f , a qual mapeia um indivíduo y no valor 1 se y ama x e no valor 0 se y não ama x .

Verbos Transitivos



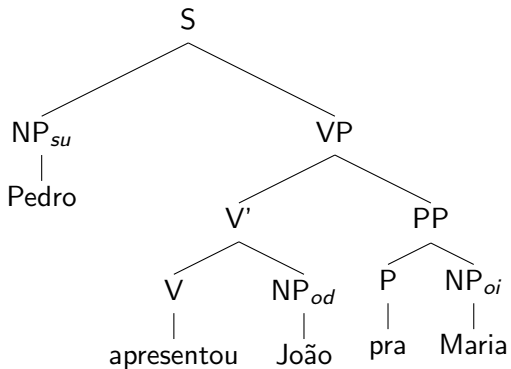
- ▶ Se a extensão de *ama* for uma função, ela deve levar indivíduos em funções, de modo que um indivíduo x seja mapeado em uma função f , a qual mapeia um indivíduo y no valor 1 se y ama x e no valor 0 se y não ama x .
- ▶ $\llbracket \text{ama} \rrbracket = \lambda x. [\lambda y. y \text{ ama } x]$

Verbos Transitivos



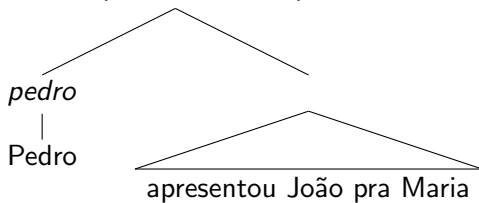
- ▶ $\llbracket S \rrbracket = \llbracket \text{ama Maria} \rrbracket(\llbracket \text{Pedro} \rrbracket)$ (Aplic. Func.)
- ▶ $\llbracket \text{ama maria} \rrbracket = \llbracket \text{ama} \rrbracket(\llbracket \text{Maria} \rrbracket)$ (Aplic. Func.)

Verbos Bitransitivos



Verbos Bitransitivos

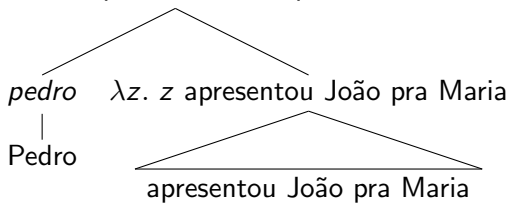
1 sse Pedro apresentou João pra Maria



[[apresentou João pra Maria]] = ???

Verbos Bitransitivos

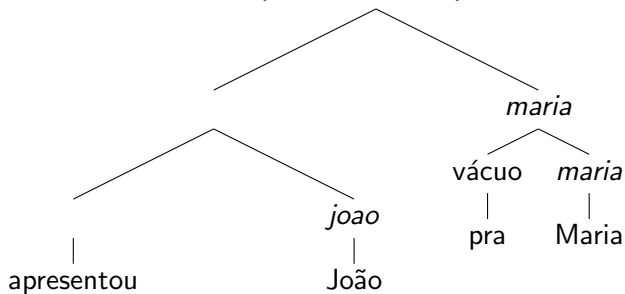
1 sse Pedro apresentou João pra Maria



$\llbracket \text{apresentou João pra Maria} \rrbracket = \lambda z. z$ apresentou João pra Maria

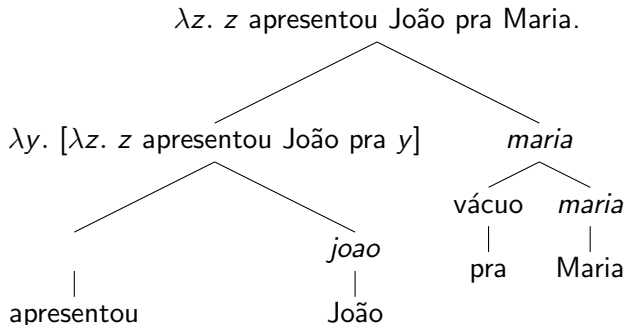
Verbos Bitransitivos

$\lambda z. z$ apresentou João pra Maria.



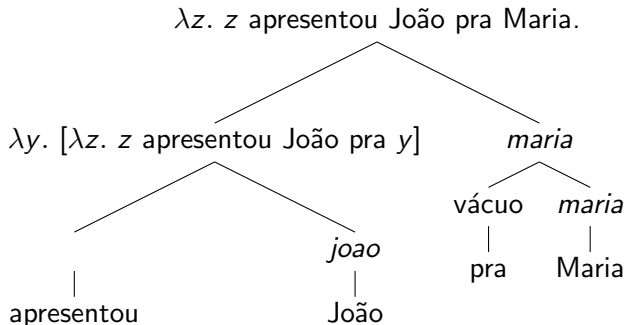
[[apresentou João]] = ???

Verbos Bitransitivos



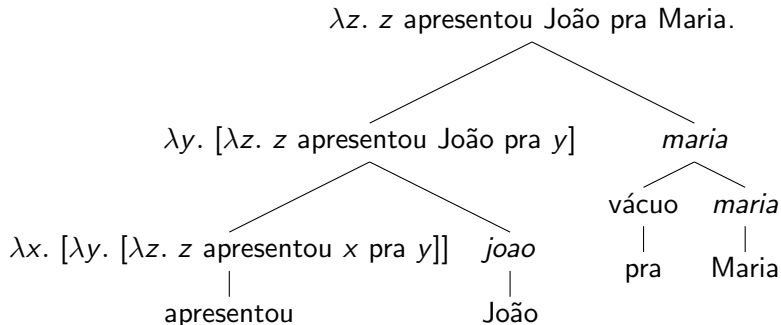
$\llbracket \text{apresentou João} \rrbracket = \lambda y. [\lambda z. z \text{ apresentou João pra } y]$

Verbos Bitransitivos



[[apresentou]] = ???

Verbos Bitransitivos



$\llbracket \text{apresentou} \rrbracket = \lambda x. [\lambda y. [\lambda z. z \text{ apresentou } x \text{ pra } y]]$

Domínios Semânticos

Domínios Semânticos

D_e : conjunto dos indivíduos

Domínios Semânticos

D_e : conjunto dos indivíduos

D_t : conjunto dos valores de verdade ($\{0,1\}$)

Domínios Semânticos

D_e : conjunto dos indivíduos

D_t : conjunto dos valores de verdade ($\{0,1\}$)

$D_{\langle\sigma,\tau\rangle}$: conjunto das funções de D_σ em D_τ .

Domínios Semânticos

D_e : conjunto dos indivíduos

D_t : conjunto dos valores de verdade ($\{0,1\}$)

$D_{\langle\sigma,\tau\rangle}$: conjunto das funções de D_σ em D_τ .

A denotação de um nome próprio pertence a D_e

Domínios Semânticos

D_e : conjunto dos indivíduos

D_t : conjunto dos valores de verdade ($\{0,1\}$)

$D_{\langle\sigma,\tau\rangle}$: conjunto das funções de D_σ em D_τ .

A denotação de um nome próprio pertence a D_e

A denotação de uma sentença pertence a D_t

Domínios Semânticos

D_e : conjunto dos indivíduos

D_t : conjunto dos valores de verdade ($\{0,1\}$)

$D_{\langle\sigma,\tau\rangle}$: conjunto das funções de D_σ em D_τ .

A denotação de um nome próprio pertence a D_e

A denotação de uma sentença pertence a D_t

A denotação de um verbo intransitivo pertence a $D_{\langle e,t\rangle}$

Domínios Semânticos

D_e : conjunto dos indivíduos

D_t : conjunto dos valores de verdade ($\{0,1\}$)

$D_{\langle\sigma,\tau\rangle}$: conjunto das funções de D_σ em D_τ .

A denotação de um nome próprio pertence a D_e

A denotação de uma sentença pertence a D_t

A denotação de um verbo intransitivo pertence a $D_{\langle e,t\rangle}$

A denotação de um verbo transitivo pertence a $D_{\langle e,\langle e,t\rangle\rangle}$

Domínios Semânticos

D_e : conjunto dos indivíduos

D_t : conjunto dos valores de verdade ($\{0,1\}$)

$D_{\langle\sigma,\tau\rangle}$: conjunto das funções de D_σ em D_τ .

A denotação de um nome próprio pertence a D_e

A denotação de uma sentença pertence a D_t

A denotação de um verbo intransitivo pertence a $D_{\langle e,t\rangle}$

A denotação de um verbo transitivo pertence a $D_{\langle e,\langle e,t\rangle\rangle}$

A denotação de um verbo transitivo pertence a

Domínios Semânticos

D_e : conjunto dos indivíduos

D_t : conjunto dos valores de verdade ($\{0,1\}$)

$D_{\langle\sigma,\tau\rangle}$: conjunto das funções de D_σ em D_τ .

A denotação de um nome próprio pertence a D_e

A denotação de uma sentença pertence a D_t

A denotação de um verbo intransitivo pertence a $D_{\langle e,t\rangle}$

A denotação de um verbo transitivo pertence a $D_{\langle e,\langle e,t\rangle\rangle}$

A denotação de um verbo transitivo pertence a $D_{\langle e,\langle e,\langle e,t\rangle\rangle\rangle}$

Tipos Semânticos

Tipos Semânticos

e e t são tipos semânticos;

Tipos Semânticos

e e t são tipos semânticos;

Se σ e τ são tipos semânticos, então $\langle \sigma, \tau \rangle$ é um tipo semântico;

Tipos Semânticos

e e t são tipos semânticos;

Se σ e τ são tipos semânticos, então $\langle \sigma, \tau \rangle$ é um tipo semântico;

Nada mais é um tipo semântico.

Tipos Semânticos

e e t são tipos semânticos;

Se σ e τ são tipos semânticos, então $\langle \sigma, \tau \rangle$ é um tipo semântico;

Nada mais é um tipo semântico.

Se uma denotação pertence a D_α , essa denotação é de tipo α

Tipos Semânticos

e e t são tipos semânticos;

Se σ e τ são tipos semânticos, então $\langle \sigma, \tau \rangle$ é um tipo semântico;

Nada mais é um tipo semântico.

Se uma denotação pertence a D_α , essa denotação é de tipo α

A denotação de um nome próprio é de tipo e

Tipos Semânticos

e e t são tipos semânticos;

Se σ e τ são tipos semânticos, então $\langle \sigma, \tau \rangle$ é um tipo semântico;

Nada mais é um tipo semântico.

Se uma denotação pertence a D_α , essa denotação é de tipo α

A denotação de um nome próprio é de tipo e

A denotação de uma sentença é de tipo t

Tipos Semânticos

e e t são tipos semânticos;

Se σ e τ são tipos semânticos, então $\langle \sigma, \tau \rangle$ é um tipo semântico;

Nada mais é um tipo semântico.

Se uma denotação pertence a D_α , essa denotação é de tipo α

A denotação de um nome próprio é de tipo e

A denotação de uma sentença é de tipo t

A denotação de um verbo intransitivo é de tipo $\langle e, t \rangle$

Tipos Semânticos

e e t são tipos semânticos;

Se σ e τ são tipos semânticos, então $\langle \sigma, \tau \rangle$ é um tipo semântico;

Nada mais é um tipo semântico.

Se uma denotação pertence a D_α , essa denotação é de tipo α

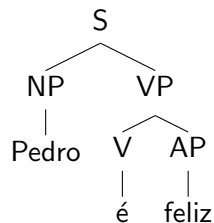
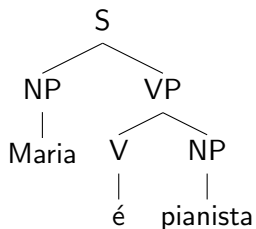
A denotação de um nome próprio é de tipo e

A denotação de uma sentença é de tipo t

A denotação de um verbo intransitivo é de tipo $\langle e, t \rangle$

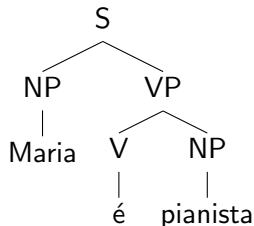
...

Predicados Não-Verbais



- ▶ Vamos tratar o verbo *ser* como sendo semanticamente vácuo.

Nomes Comuns

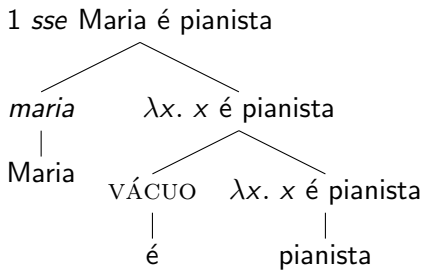


$\llbracket S \rrbracket = 1$ sse Maria é pianista

$\llbracket VP \rrbracket = \llbracket NP \rrbracket = \llbracket \text{pianista} \rrbracket$

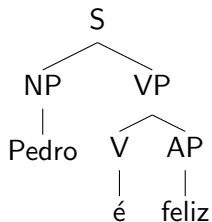
$\llbracket \text{pianista} \rrbracket = \lambda x. x \text{ é pianista}$

Nomes Comuns



$$[[S]] = [[é pianista]]([[Maria]]) \quad (AF)$$

Adjetivos

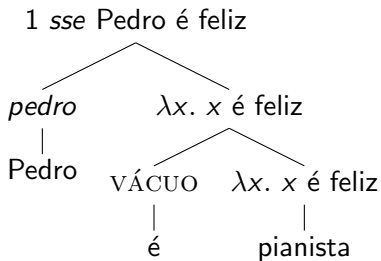


$\llbracket S \rrbracket = 1$ sse Pedro é feliz

$\llbracket VP \rrbracket = \llbracket AP \rrbracket = \llbracket feliz \rrbracket$

$\llbracket feliz \rrbracket = \lambda x. x \text{ é feliz}$

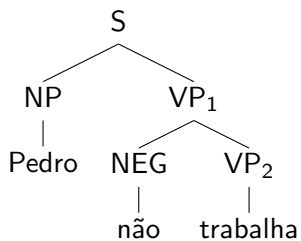
Adjetivos



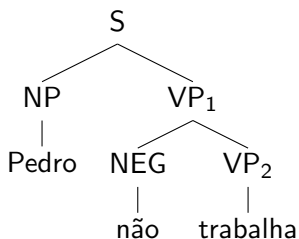
$$\llbracket S \rrbracket = \llbracket \text{é feliz} \rrbracket(\llbracket \text{Pedro} \rrbracket)$$

$$(AF)$$

Negação



Negação



$\llbracket S \rrbracket = 1$ sse Pedro não trabalha

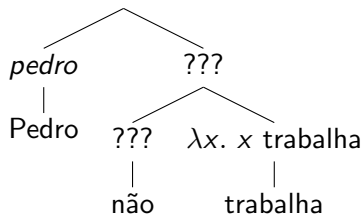
$\llbracket \text{Pedro} \rrbracket = \text{pedro}$

$\llbracket \text{trabalha} \rrbracket = \lambda x. x \text{ trabalha}$

$\llbracket \text{não} \rrbracket = ???$

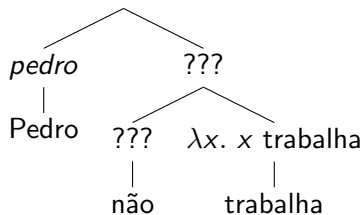
Negação

1 sse Pedro não trabalha



Negação

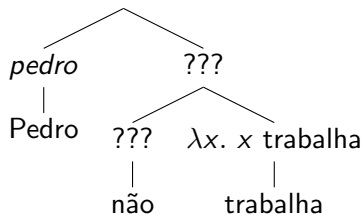
1 sse Pedro não trabalha



$\llbracket \text{não trabalha} \rrbracket = \lambda x. x \text{ não trabalha}$

Negação

1 sse Pedro não trabalha

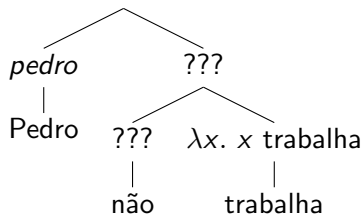


$\llbracket \text{não trabalha} \rrbracket = \lambda x. x \text{ não trabalha}$

$\llbracket \text{não trabalha} \rrbracket = \lambda x. \llbracket \text{trabalha} \rrbracket(x) = 0$

Negação

1 sse Pedro não trabalha



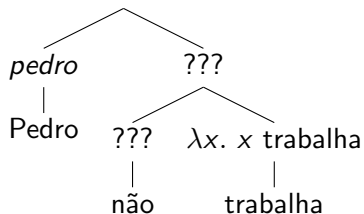
$\llbracket \text{não trabalha} \rrbracket = \lambda x. x \text{ não trabalha}$

$\llbracket \text{não trabalha} \rrbracket = \lambda x. \llbracket \text{trabalha} \rrbracket(x) = 0$

$\llbracket \text{não} \rrbracket =$

Negação

1 sse Pedro não trabalha

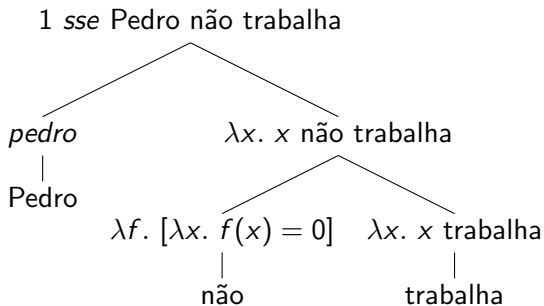


$\llbracket \text{não trabalha} \rrbracket = \lambda x. x \text{ não trabalha}$

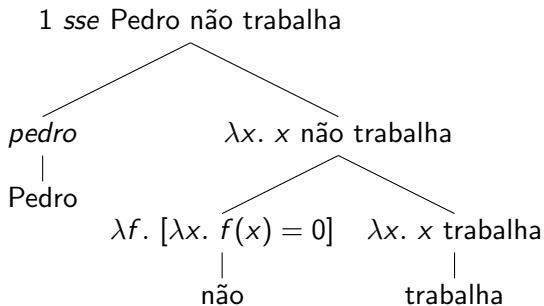
$\llbracket \text{não trabalha} \rrbracket = \lambda x. \llbracket \text{trabalha} \rrbracket(x) = 0$

$\llbracket \text{não} \rrbracket = \lambda f. [\lambda x. f(x) = 0]$

Negação



Negação



$\llbracket S \rrbracket = \llbracket \text{não trabalha} \rrbracket(\llbracket \text{Pedro} \rrbracket)$ (AF)

$\llbracket \text{não trabalha} \rrbracket = \llbracket \text{não} \rrbracket(\llbracket \text{trabalha} \rrbracket)$ (AF)

$\llbracket \text{Pedro não trabalha} \rrbracket = (\llbracket \text{não} \rrbracket(\llbracket \text{trabalha} \rrbracket))(\llbracket \text{Pedro} \rrbracket)$ (AF 2x)