

Semântica e Gramática Gerativa

Aula 9

Marcelo Ferreira
ferreira10@usp.br

Universidade de São Paulo

Pronomes Ligados

- (1) João odeia o chefe dele.
- (2) Só João odeia o chefe dele.

Pronomes Ligados

- (1) João odeia o chefe dele.
 - (2) Só João odeia o chefe dele.
- ▶ (2) é ambígua de uma forma que (1) não é.

Pronomes Ligados

[Pedro é o chefe de João]

João odeia o chefe dele \equiv João odeia Pedro

Pronomes Ligados

[Pedro é o chefe de João]

João odeia o chefe dele \equiv João odeia Pedro

Só João odeia o chefe dele \equiv

Pronomes Ligados

[Pedro é o chefe de João]

João odeia o chefe dele \equiv João odeia Pedro

Só João odeia o chefe dele \equiv

João odeia Pedro e ninguém mais odeia Pedro.

João odeia Pedro e ninguém mais odeia o próprio chefe.

Co-referência vs. Ligação

(3) João odeia o chefe dele.

$$\llbracket \text{odeia o chefe dele} \rrbracket^g = \begin{cases} \lambda x. x \text{ odeia o chefe do João} & (a) \\ \lambda x. x \text{ odeia o chefe de } x & (b) \end{cases}$$

- ▶ Em (a), o pronome é interpretado anaforicamente, co-referente ao sujeito *João*
- ▶ Em (b), o pronome é interpretado como uma *variável ligada*.

Co-referência vs. Ligação

(4) João odeia o chefe dele.

- ▶ Quando o sujeito da oração é um nome próprio, as interpretações anafórica e de variável ligada não levam a diferenças de significado.
- ▶ $[\lambda x. x \text{ odeia o chefe do João}](\text{joão}) =$
1 sse João odeia o chefe do João
- ▶ $[\lambda x. x \text{ odeia o chefe de } x](\text{joão}) =$
1 sse João odeia o chefe do João

[_{DP} Só João]

- ▶ $\llbracket \text{Só João trabalha} \rrbracket = 1 \text{ sse}$
João trabalha e ninguém mais trabalha

[_{DP} Só João]

- ▶ $\llbracket \text{Só João trabalha} \rrbracket = 1 \text{ sse}$
João trabalha e ninguém mais trabalha
- ▶ Intuição: $\llbracket \text{Só João} \rrbracket$ inspeciona a extensão de VP e verifica se João é o único indivíduo que é levado no valor 1 por esta função.

[_{DP} Só João]

- ▶ $\llbracket \text{Só João trabalha} \rrbracket = 1$ sse
João trabalha e ninguém mais trabalha
- ▶ Intuição: $\llbracket \text{Só João} \rrbracket$ inspeciona a extensão de VP e verifica se João é o único indivíduo que é levado no valor 1 por esta função.
- ▶ $\llbracket \text{Só João} \rrbracket =$
 $\lambda f. f(\text{joão}) = 1 \ \& \ \neg \exists x : x \neq \text{joão} \ \& \ f(x) = 1$

Co-referência vs. Ligação

(5) Só João odeia o chefe dele.

- ▶ Quando o sujeito da oração é não é um nome próprio, as interpretações anafórica e de variável ligada podem levar a diferenças de significado.

- ▶ $\llbracket \text{Só João} \rrbracket = \lambda f. f(\text{joão}) = 1 \ \& \ \neg \exists x : x \neq \text{joão} \ \& \ f(x) = 1$

- ▶ $\llbracket \text{só João} \rrbracket (\lambda x. x \text{ odeia o chefe do João}) = 1$ sse

João odeia o chefe do João e

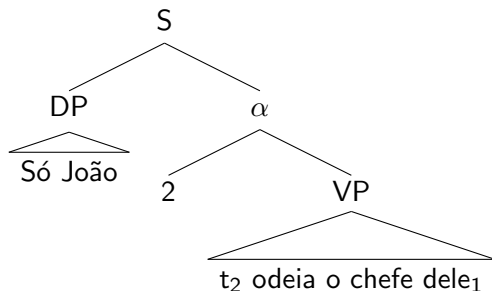
$\neg \exists x : x \neq \text{joão} \ \& \ x \text{ odeia o chefe do João}$

- ▶ $\llbracket \text{só o João} \rrbracket (\lambda x. x \text{ odeia o chefe de } x) = 1$ sse

João odeia o chefe do João e

$\neg \exists x : x \neq \text{joão} \ \& \ x \text{ odeia o chefe de } x$

Co-referência: Sintaxe e Semântica



$$[[\alpha]]^g = \lambda x. [[VP]]^{g[2 \rightarrow x]}$$

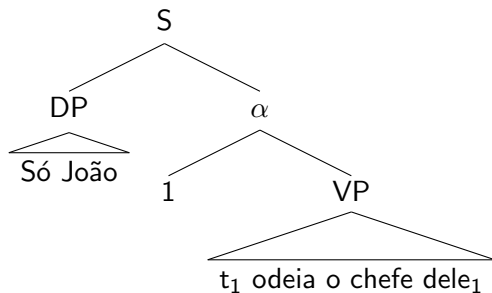
$$[[VP]]^{g[2 \rightarrow x]} = 1 \text{ sse } x \text{ odeia o chefe de } g(1)$$

$$[[\alpha]]^g = \lambda x. x \text{ odeia o chefe de } g(1)$$

$$[[\alpha]]^g = \lambda x. x \text{ odeia o chefe do João}$$

(se $g(1) = \text{João}$)

Ligação: Sintaxe e Semântica



$$[[\alpha]]^g = \lambda x. [[VP]]^g[1 \rightarrow x]$$

$$[[VP]]^g[1 \rightarrow x] = 1 \text{ sse } x \text{ odeia o chefe de } x$$

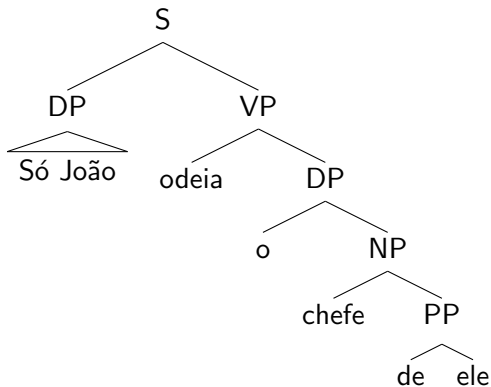
$$[[\alpha]]^g = \lambda x. x \text{ odeia o chefe de } x$$

Ligação Pronominal sem Índices nem atribuições

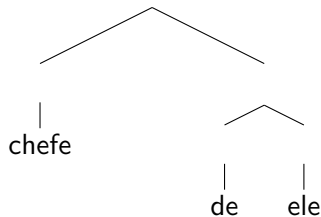
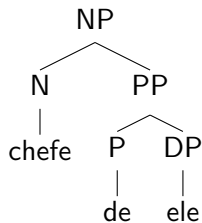
- ▶ O que veremos a seguir é um esboço de uma adaptação para os nossos propósitos de alguns trabalhos de Pauline Jacobson. O principal deles é:

Jacobson, P. (1999) *Towards a Variable-Free Semantics*.
Linguistics and Philosophy 22, pp. 117-184.

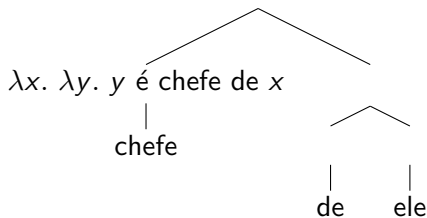
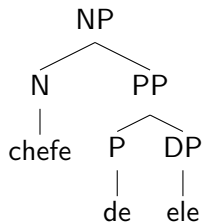
Ligação Pronominal sem Índices nem atribuições



Ligação Pronominal sem Índices nem atribuições

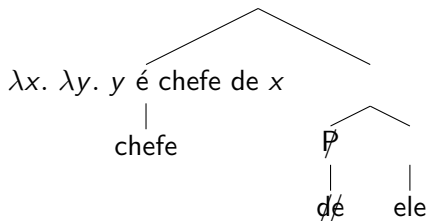
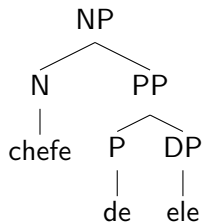


Ligação Pronominal sem Índices nem atribuições



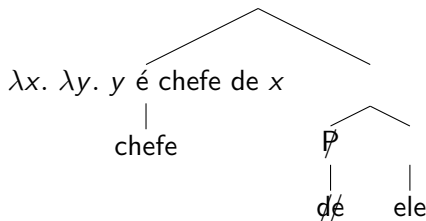
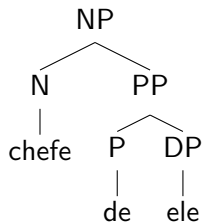
- ▶ $[[\text{chefe}]] = \lambda x. \lambda y. y \text{ é chefe de } x$

Ligação Pronominal sem Índices nem atribuições



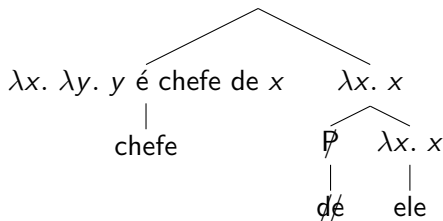
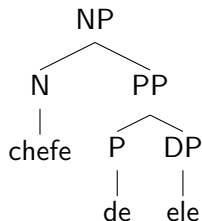
- ▶ $\llbracket \text{chefe} \rrbracket = \lambda x. \lambda y. y \text{ é chefe de } x$
- ▶ A preposição *de* é semanticamente vácuua.

Ligação Pronominal sem Índices nem atribuições



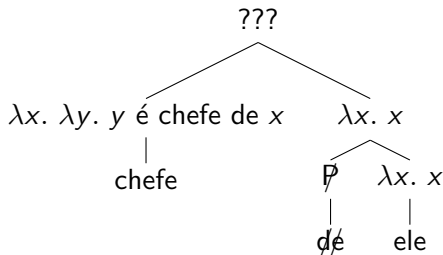
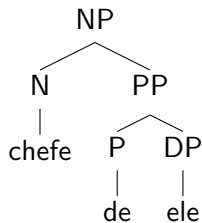
- ▶ $\llbracket \text{chefe} \rrbracket = \lambda x. \lambda y. y \text{ é chefe de } x$
- ▶ A preposição *de* é semanticamente vácuua.
- ▶ $\llbracket \text{ele} \rrbracket =$

Ligação Pronominal sem Índices nem atribuições



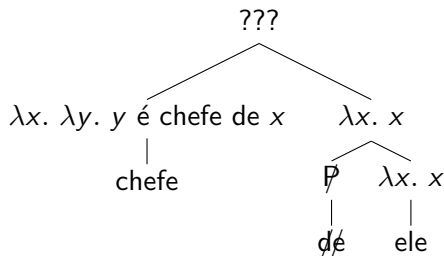
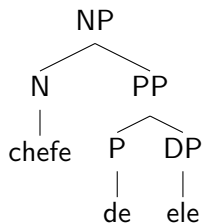
- ▶ $[[\text{chefe}]] = \lambda x. \lambda y. y \text{ é chefe de } x$
- ▶ A preposição *de* é semanticamente vácuua.
- ▶ $[[\text{ele}]] = \lambda x. x$ Pronomes denotam a função identidade!!!

Ligação Pronominal sem Índices nem atribuições



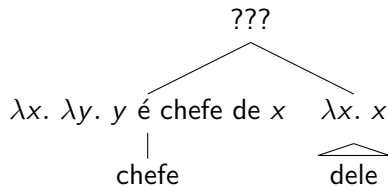
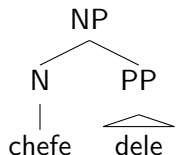
- ▶ $\llbracket \text{chefe} \rrbracket = \lambda x. \lambda y. y \text{ é chefe de } x$
- ▶ A preposição *de* é semanticamente vácuua.
- ▶ $\llbracket \text{ele} \rrbracket = \lambda x. x$ Pronomes denotam a função identidade!!!
- ▶ $\llbracket \text{NP} \rrbracket = ???$

Ligação Pronominal sem Índices nem atribuições



- ▶ Incompatibilidade de tipos: $\llbracket \text{chefe} \rrbracket$ é de tipo $\langle e, et \rangle$ e $\llbracket \text{dele} \rrbracket$ de tipo $\langle e, e \rangle$

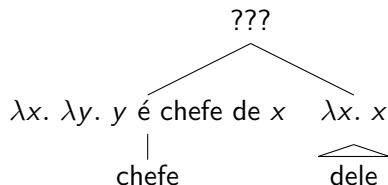
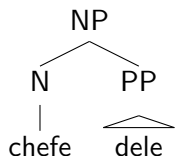
Ligação Pronominal sem Índices nem atribuições



Composição Funcional

$$\left[\begin{array}{c} \diagup \quad \diagdown \\ X_{\langle b,c \rangle} \quad Y_{\langle a,b \rangle} \\ \diagdown \quad \diagup \end{array} \right] = \lambda x_a. \llbracket X \rrbracket (\llbracket Y \rrbracket (x))$$

Ligação Pronominal sem Índices nem atribuições

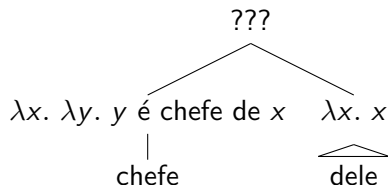
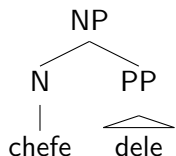


Composição Funcional

$$\left[\begin{array}{c} \diagup \quad \diagdown \\ X_{\langle b,c \rangle} \quad Y_{\langle a,b \rangle} \\ \diagdown \quad \diagup \end{array} \right] = \lambda x_a. \llbracket X \rrbracket (\llbracket Y \rrbracket (x))$$

$$\llbracket \text{NP} \rrbracket = \lambda x_e. \llbracket \text{chefe} \rrbracket (\llbracket \text{dele} \rrbracket (x))$$

Ligação Pronominal sem Índices nem atribuições



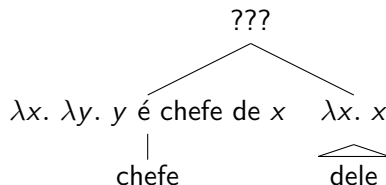
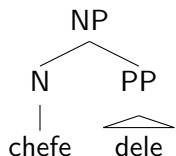
Composição Funcional

$$\left[\begin{array}{c} \diagup \quad \diagdown \\ X_{\langle b,c \rangle} \quad Y_{\langle a,b \rangle} \\ \diagdown \quad \diagup \end{array} \right] = \lambda x_a. \llbracket X \rrbracket (\llbracket Y \rrbracket (x))$$

$$\llbracket \text{NP} \rrbracket = \lambda x_e. \llbracket \text{chefe} \rrbracket (\llbracket \text{dele} \rrbracket (x))$$

$$\llbracket \text{NP} \rrbracket = \lambda x_e. \llbracket \text{chefe} \rrbracket (x)$$

Ligação Pronominal sem Índices nem atribuições



Composição Funcional

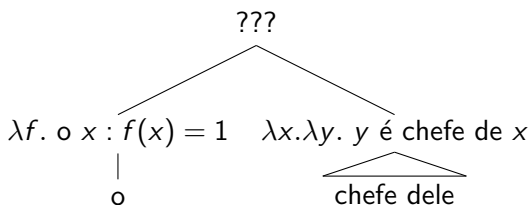
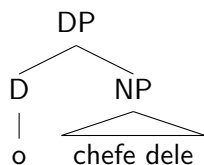
$$\left[\begin{array}{c} \wedge \\ X_{\langle b,c \rangle} \quad Y_{\langle a,b \rangle} \end{array} \right] = \lambda x_a. [X]([Y](x))$$

$$[NP] = \lambda x_e. [\text{chefe}]([\text{dele}](x))$$

$$[NP] = \lambda x_e. [\text{chefe}](x)$$

$$[NP] = [\text{chefe}] \quad !!! \quad \underline{\text{[Compor com a função id. é vácuo!]}}$$

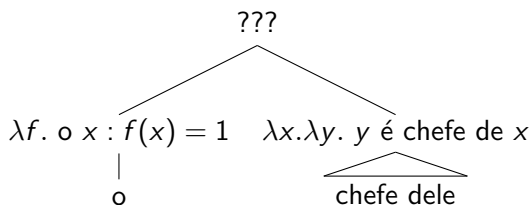
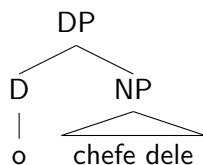
Ligação Pronominal sem Índices nem atribuições



Composição Funcional

$$\left[\left[X_{\langle b,c \rangle} \quad Y_{\langle a,b \rangle} \right] \right] = \lambda x_a. \left[X \right] \left(\left[Y \right] (x) \right)$$

Ligação Pronominal sem Índices nem atribuições

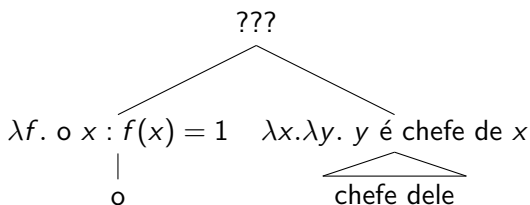
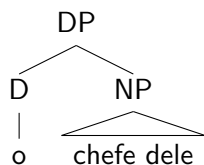


Composição Funcional

$$\left[\left[X_{\langle b,c \rangle} \quad Y_{\langle a,b \rangle} \right] \right] = \lambda x_a. \llbracket X \rrbracket (\llbracket Y \rrbracket (x))$$

$$\llbracket \text{DP} \rrbracket = \lambda x_e. \llbracket o \rrbracket (\llbracket \text{chefe dele} \rrbracket (x))$$

Ligação Pronominal sem Índices nem atribuições



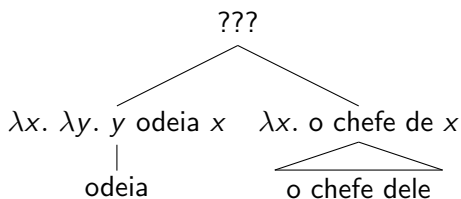
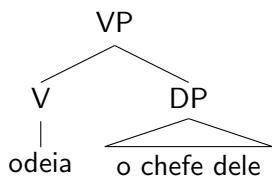
Composição Funcional

$$\left[\left[X_{\langle b,c \rangle} \quad Y_{\langle a,b \rangle} \right] \right] = \lambda x_a. [X]([Y](x))$$

$$[DP] = \lambda x_e. [o]([chefe\ dele](x))$$

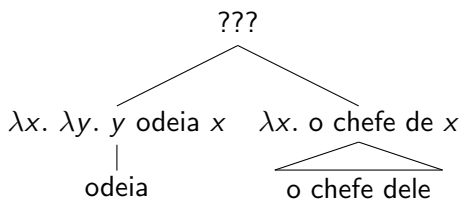
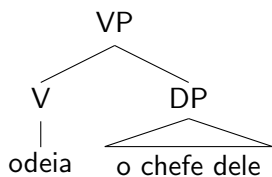
$$[DP] = \lambda x_e. o\ chefe\ de\ x$$

Ligação Pronominal sem Índices nem atribuições



[Chegou a hora de ligar o pronome. Mas sem Abstração Funcional!]

Ligação Pronominal sem Índices nem atribuições

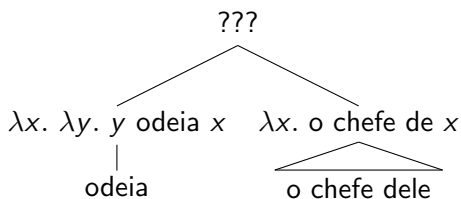
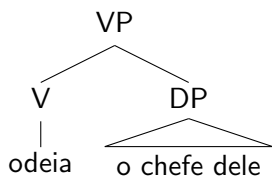


[Chegou a hora de ligar o pronome. Mas sem Abstração Funcional!]

Ligação

$$\left[\begin{array}{c} \diagup \quad \diagdown \\ X_{\langle a, et \rangle} \quad Y_{\langle e, a \rangle} \\ \diagdown \quad \diagup \end{array} \right] = \lambda x_e. \left[X \right] \left(\left[Y \right] (x) \right) (x)$$

Ligação Pronominal sem Índices nem atribuições



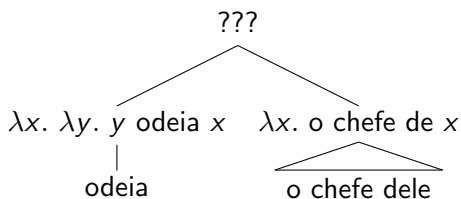
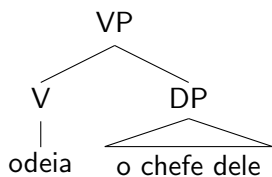
[Chegou a hora de ligar o pronome. Mas sem Abstração Funcional!]

Ligação

$$\left[\begin{array}{c} \diagup \quad \diagdown \\ X_{\langle a, et \rangle} \quad Y_{\langle e, a \rangle} \\ \diagdown \quad \diagup \end{array} \right] = \lambda x_e. \llbracket X \rrbracket (\llbracket Y \rrbracket (x)) (x)$$

$$\llbracket VP \rrbracket = \lambda x_e. \llbracket odeia \rrbracket (\llbracket o chefe dele \rrbracket (x)) (x)$$

Ligação Pronominal sem Índices nem atribuições



[Chegou a hora de ligar o pronome. Mas sem Abstração Funcional!]

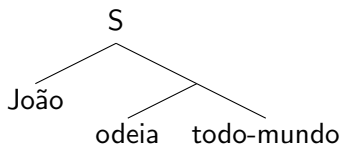
Ligação

$$\left[\begin{array}{c} \diagup \quad \diagdown \\ X_{\langle a, et \rangle} \quad Y_{\langle e, a \rangle} \\ \diagdown \quad \diagup \end{array} \right] = \lambda x_e. [X]([Y](x))(x)$$

$$[VP] = \lambda x_e. [odeia]([o \text{ chefe dele}](x))(x)$$

$$[VP] = \lambda x_e. x \text{ odeia o chefe de } x$$

Composição Funcional é poderosa demais!



Composição Funcional

$$\left[\begin{array}{c} \diagup \quad \diagdown \\ X_{\langle b,c \rangle} \quad Y_{\langle a,b \rangle} \\ \diagdown \quad \diagup \end{array} \right] = \lambda x_a. [X]([Y](x))$$

$$[\text{odeia}] = \lambda x. \lambda y. y \text{ odeia } x \quad \text{tipo } \langle e, et \rangle$$

$$[\text{todo mundo}] = \lambda f. \forall z : f(z) = 1 \quad \text{tipo } \langle et, t \rangle$$

$$[\text{odeia todo mundo}] = \lambda x_e. [\text{todo mundo}]([\text{odeia}](x)) \quad (\text{CF})$$

$$[\text{odeia todo mundo}] = \lambda x. \forall z : z \text{ odeia } x$$

$$[S] = 1 \text{ sse } \forall z : z \text{ odeia João} \quad (\text{AF})$$

Limitando Composição Funcional

- ▶ Nosso objetivo com CF foi o de passar “pra cima” a posição argumental de um pronome até que a posição do ligador aparecesse e Ligação se aplicasse. Vamos então limitar seu uso a casos envolvendo constituintes contendo pronomes.
- ▶ Em um sistema *type-driven* isso não é trivial e nossa solução é apenas um paliativo. Apesar de pronomes e constituintes que os contem denotarem funções que levam indivíduos (tipo e) em algum tipo de entidade (tipo α), não associaremos tais denotações ao tipo $\langle e, \alpha \rangle$, mas sim a um novo tipo complexo $\langle e; \alpha \rangle$. Podemos assim redefinir CF:

Limitando Composição Funcional

Composição Funcional

Sejam a e b dois tipos quaisquer. Se Z é um nó ramificado cujos constituintes imediatos são X (de tipo $\langle a, b \rangle$) e Y (de tipo $\langle e; a \rangle$), então a denotação de Z é de tipo $\langle e; b \rangle$ e definida da seguinte forma:

$$\left[\begin{array}{c} \diagup \quad \diagdown \\ X_{\langle a, b \rangle} \quad Y_{\langle e; a \rangle} \end{array} \right] = \lambda x_e. \llbracket X \rrbracket (\llbracket Y \rrbracket (x))$$

Limitando Ligação

- ▶ Seguindo a mesma linha, vamos também limitar a aplicação de Ligação a casos envolvendo pronomes

Ligação Pronominal

Seja a um tipo qualquer. Se Z é um nó ramificado cujos constituintes imediatos são X (de tipo $\langle a, et \rangle$) e Y (de tipo $\langle e; a \rangle$), então a denotação de Z é de tipo $\langle e; t \rangle$ e definida da seguinte forma:

$$\left[\begin{array}{c} \diagup \quad \diagdown \\ X_{\langle a, et \rangle} \quad Y_{\langle e; a \rangle} \end{array} \right] = \lambda x_e. \llbracket X \rrbracket (\llbracket Y \rrbracket (x))(x)$$

Pronomes Livres

- ▶ Em nosso sistema anterior, a denotação de sentenças com pronomes livres, como *João gosta dela*, era relativizada a uma atribuição. Agora, que não dispomos mais de atribuições, tais sentenças não denotarão valores de verdade, mas sim funções de indivíduos em valores de verdade. A ideia é que quando proferidas em um contexto em que um indivíduo esteja saliente, obtemos um valor de verdade aplicando a função a esse indivíduo.

Pronomes Livres

Só João odeia o chefe dele [ele = Pedro, Lucas, João, ...]

[[o chefe dele]] = $\lambda x. \text{o chefe de } x$ (CF 2x)

[[odeia]] = $\lambda z. \lambda y. y \text{ odeia } z$

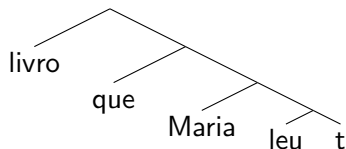
[[odeia o chefe dele]] = $\lambda x. \lambda y. y \text{ odeia o chefe de } x$ (CF)

[[Só João]] = $\lambda f. f(\text{joão}) = 1 \ \& \ \neg \exists z : z \neq \text{joão} \ \& \ f(z) = 1$

[[Só João odeia o chefe dele]] =
 $\lambda x. \text{João odeia o chefe de } x \ \& \ \neg \exists z : z \neq \text{joão} \ \& \ z \text{ odeia o chefe de } x$ (CF)

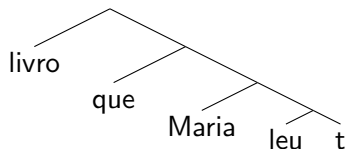
Orações Relativas

- ▶ Em nosso sistema anterior, vestígios eram tratados como pronomes. Podemos manter essa característica em nosso novo sistema, só que agora dispensando índices e atribuições.



Orações Relativas

- Em nosso sistema anterior, vestígios eram tratados como pronomes. Podemos manter essa característica em nosso novo sistema, só que agora dispensando índices e atribuições.



$$\llbracket t \rrbracket = \lambda x. x$$

tipo $\langle e; e \rangle$

$$\llbracket \text{Maria leu } t \rrbracket = \lambda x. \text{Maria leu } x$$

tipo $\langle e; t \rangle$

$$\llbracket \text{que} \rrbracket = \lambda f_{\langle e; t \rangle}. \lambda g_{\langle e, t \rangle}. \lambda x. g(x) = 1 \ \& \ f(x) = 1$$

$$\llbracket \text{que Maria leu } t \rrbracket = \lambda g_{\langle e, t \rangle}. \lambda x. g(x) = 1 \ \& \ \text{Maria leu } x$$

$$\llbracket \text{livro que Maria leu } t \rrbracket = \lambda x. x \ \text{é livro} \ \& \ \text{Maria leu } x$$

Considerações Finais

- ▶ Por um lado, este novo sistema dispensa o uso de atribuições e índices, além da regra de Abstração Funcional.
- ▶ Por outro lado, necessita de duas novas regras: Composição e Ligação
- ▶ Podemos declarar um vencedor conceitual? Difícil dizer.
- ▶ Para argumentos a favor de um sistema sem atribuições e índices, ver os trabalhos de Pauline Jacobson, em especial *Towards a Variable-Free Semantics*, em *Linguistics and Philosophy* 22, 1999. pp. 117-184.