UNIVERSITY OF TORONTO | Computer Science

Quantifier Scope Storage

CSC485/2501

December 2, 2021

Outline

- The Language of the Lambda Calculus
- Alpha Conversion
- Beta Reduction
- Logical Representations of Meaning
- Quantifier Storage

The Language of the Lambda Calculus

- Constants, e.g., a, f
- Variables, e.g., x, y (they often look like constants)
- Applications, e.g., $f(a)$
- Abstractions, e.g. $\lambda x.f(x)$

Alpha Conversion

The Basic Idea

$\lambda x.f(x) \Rightarrow \lambda y.f(y)$ [y fresh]

Beta Reduction

The Basic Idea

$(\lambda x.f(x))(a) \Rightarrow f(a)$

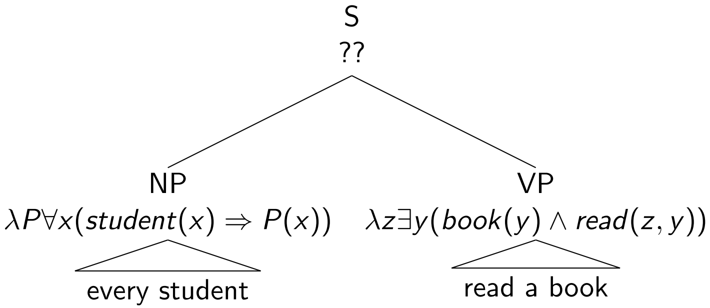
Logical Representations of Meaning

What's one of the logical forms?

Every student read a book

Logical Representations of Meaning

- Every student read a book
  $\forall x(student(x) \Rightarrow \exists y(book(y) \wedge read(x, y)))$

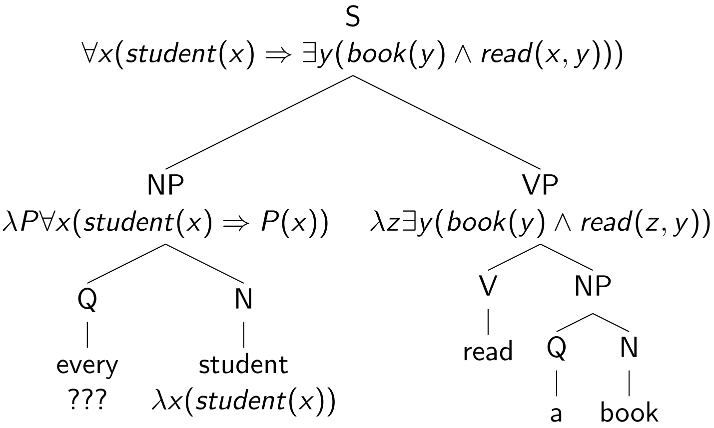


Beta Normalization

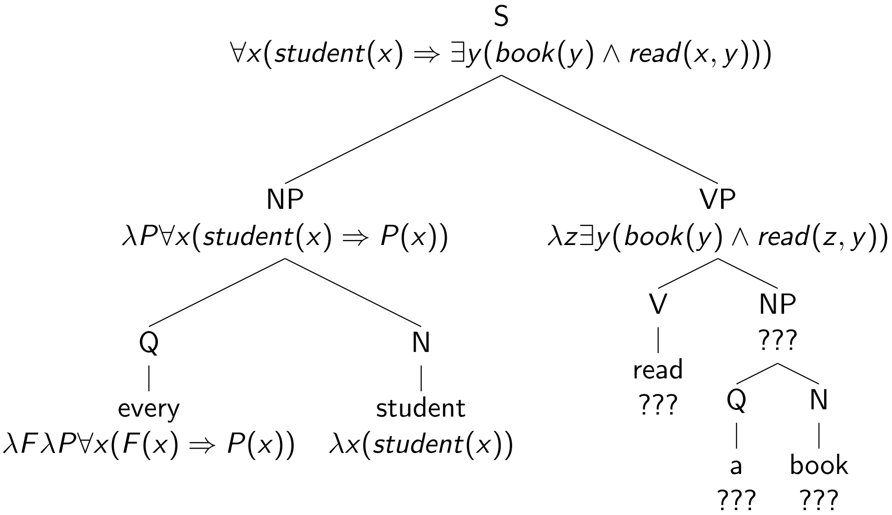
- Every student
  $\lambda P \forall x(student(x) \Rightarrow P(x))$
- read a book
  $\lambda z \exists y(book(y) \wedge read(z,y))$
- Every student read a book

$$\begin{aligned} & \lambda P \forall x(student(x) \Rightarrow P(x))(\lambda z \exists y(book(y) \wedge read(z,y))) \\ \Leftrightarrow_\beta\ & \forall x(student(x) \Rightarrow \lambda z \exists y(book(y) \wedge read(z,y))(x)) \\ \Leftrightarrow_\beta\ & \forall x(student(x) \Rightarrow \exists y(book(y) \wedge read(x,y))) \end{aligned}$$

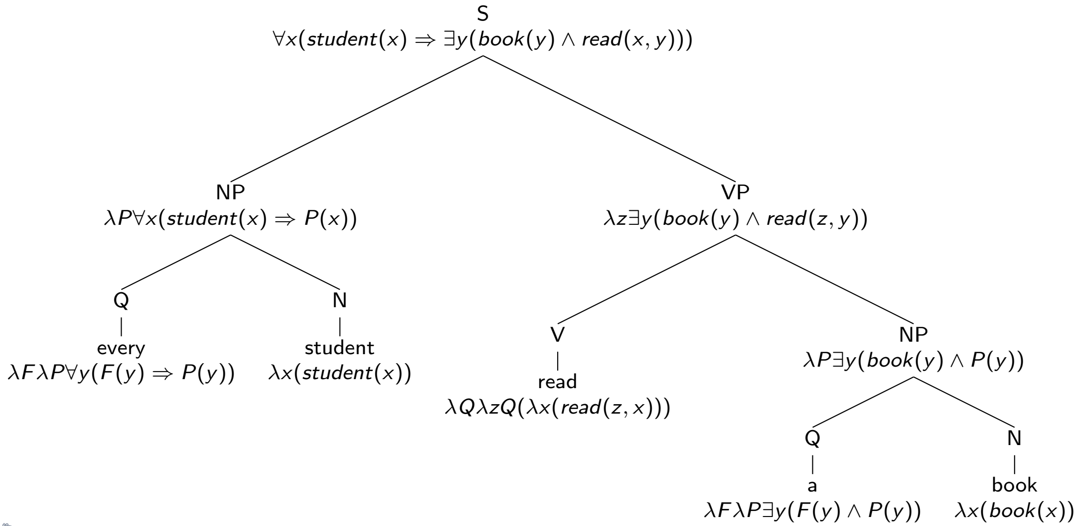
Beta Reduction: Practice



Beta Reduction: Practice



Beta Reduction: Solution to Practice



Quantifier Storage

Every student read a book

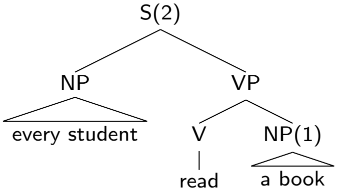
- $\forall x(student(x) \Rightarrow \exists y(book(y) \land read(x, y)))$
- $\exists y(book(y) \land \forall x(student(x) \Rightarrow read(x, y)))$

How to get the second reading?

Quantifier Storage

- Storage at (1)

Logic Statement	Quantifier Storage
$\lambda G \exists y(book(y) \land G(y))$	$\emptyset$
$\Rightarrow \lambda F.F({\color{red}x})$	$[\langle {\color{red}x}; \lambda G \exists y(book(y) \land G(y))\rangle]$



- Retrieve at (2)

$$\forall y, student(y) \Rightarrow read(y, {\color{red}x}) \quad [\langle {\color{red}x}; \lambda G \exists y(book(y) \land G(y))\rangle]$$

$$\lambda G \exists z(book(z) \land G(z))(\lambda {\color{red}x}(\forall y, student(y) \Rightarrow read(y, {\color{red}x}))) \quad []$$

$$\exists z(book(z) \land (\lambda {\color{red}x}(\forall y, student(y) \Rightarrow read(y, {\color{red}x}))(z)) \quad []$$

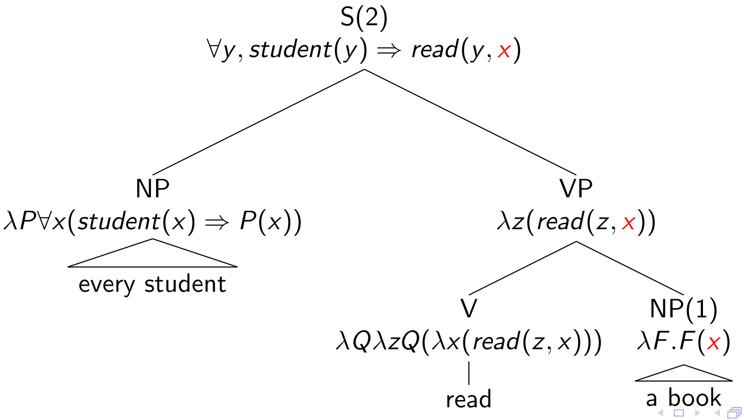
$$\exists z(book(z) \land (\forall y, student(y) \Rightarrow read(y, z)) \quad []$$

Quantifier Storage

- Storage at (1)

$$\lambda G \exists y(book(y) \wedge G(y)) \qquad \emptyset$$

$$\Rightarrow \lambda F.F(x) \qquad [\langle x; \lambda G \exists y(book(y) \wedge G(y))\rangle]$$



UNIVERSITY OF TORONTO

Quantifier Storage

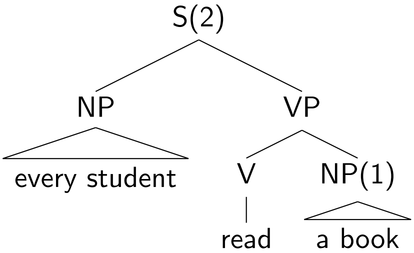
- Retrieve at (2)

$$\forall y.student(y) \Rightarrow read(y, \textcolor{red}{x}) \qquad [\langle \textcolor{red}{x}; \lambda G \exists y(book(y) \wedge G(y))\rangle]$$

$$\lambda G \exists z(book(z) \wedge G(z))(\lambda \textcolor{red}{x}(\forall y.student(y) \Rightarrow read(y, \textcolor{red}{x}))) \qquad []$$

$$\exists z(book(z) \wedge (\lambda \textcolor{red}{x}(\forall y.student(y) \Rightarrow read(y, \textcolor{red}{x}))(z)) \qquad []$$

$$\exists z(book(z) \wedge (\forall y.student(y) \Rightarrow read(y, z)) \qquad []$$



UNIVERSITY OF TORONTO