Lambda calculus: Relative clauses

λ-notation

Consider a function:

$$f : \mathbb{N} \mapsto \mathbb{N} \text{ for every } x \in \mathbb{N}.\ f(x) = x + 1$$

The function can be expressed as follows:

$$f = \lambda x : x \in \mathbb{N}. x + 1$$

The (smallest) function which maps every x such that $x \in \mathbb{N}$ to $x + 1$

Semantic types of λ-terms

If x is type of a and E is type of b, then $\lambda x.E$ is type of $a \to b$.

Intransitive verbs

$$[\![\text{dance}]\!]^{M,g} = \begin{bmatrix} \mathsf{a} \dashrightarrow 1 \\ \mathsf{b} \nearrow \\ \mathsf{c} \dashrightarrow 0 \end{bmatrix} \quad \text{i.e.,} \quad \begin{array}{l} f : D_e \mapsto D_t \\ \text{for every } x, f(x) = 1 \\ \text{iff } x \text{ dance} \end{array}$$

$$[\![\text{dance}]\!]^{M,g} = \overbrace{\lambda x}^{e} :\in D_e. \underbrace{\mathsf{dance}(x)}_{t}$$

Transitive verbs

$$\begin{bmatrix} \mathsf{a} \mapsto \begin{bmatrix} \mathsf{a} \mapsto 0 \\ \mathsf{b} \mapsto 0 \\ \mathsf{c} \mapsto 1 \end{bmatrix}, \mathsf{b} \mapsto \begin{bmatrix} \mathsf{a} \mapsto 1 \\ \mathsf{b} \mapsto 0 \\ \mathsf{c} \mapsto 0 \end{bmatrix}, \mathsf{c} \mapsto \begin{bmatrix} \mathsf{a} \mapsto 1 \\ \mathsf{b} \mapsto 0 \\ \mathsf{c} \mapsto 0 \end{bmatrix} \end{bmatrix}$$

$[\![\text{see}]\!]^{M,g} = f : D_e \mapsto D_{e \to t}$
for every x, $f(x) = f' : D_e \mapsto D_t$
for every y, $f'(y) = 1$
iff y see x

$$[\![\text{see}]\!]^{M,g} = \overbrace{\lambda x}^{e} : x \in D_e\, \overbrace{\lambda y}^{e} : y \in D_e. \underbrace{\mathsf{see}(x)(y)}_{t}$$

λ-reduction

η-equivalence

$$f = \lambda x. f(x)$$

Two functions are equivalent iff they return the same values for every argument

$$[\![\text{dance}]\!]^{M,g} = \mathsf{dance} = \lambda x.\mathsf{dance}(x)$$

β-reduction

$$(\lambda x.E_1)(E_2) = E_1[E_2/x]$$

"$E_1[E_2/x]$" is the expression just like E_1, but where every free occurrence of x has been replaced by E_2.

$$\begin{aligned} [\![\text{dance}]\!]^{M,g}([\![\text{Ada}]\!]^{M,g}) &= [\lambda x.\mathsf{dance}(x)](\mathsf{a}) \\ &= \mathsf{dance}(\mathsf{a}) \end{aligned}$$

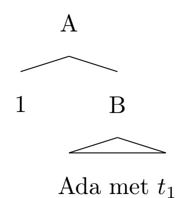
α-equivalence

$$\lambda x.E = \lambda y.E[y/x]$$

A specific choice of a bound variable doesn't matter.

$$[\![\text{dance}]\!]^{M,g} = \lambda x.\mathsf{dance}(x) = \lambda y.\mathsf{dance}(y)$$

λ-abstraction

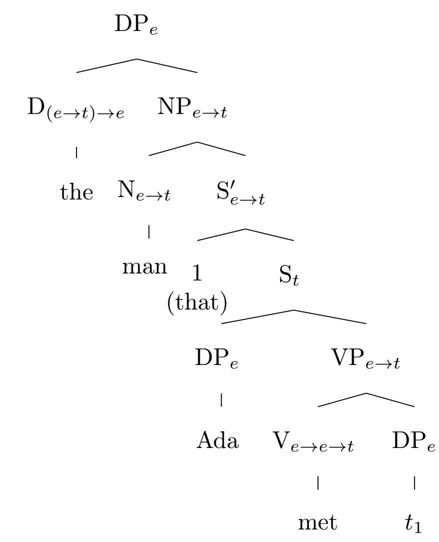


If A is branching node with daughters $n \in \mathbb{N}$ and B of type b, then A has type $a \to b$ and for any assignment g, $[\![\text{A}]\!]^{M,g} = \lambda x.[\![\text{B}]\!]^{g^{n \mapsto x}}$

$$\begin{aligned} &[\![1 \text{ [Ada met } t_1]]\!]^{M,g} \\ &= \lambda x.[\![\text{Ada met } t_1]\!]^{M,g^{1 \mapsto x}} \\ &= \lambda x.[\![\text{met}]\!]^{M,g^{1 \mapsto x}}([\![t_1]\!]^{M,g^{1 \mapsto x}})([\![\text{Ada}]\!]^{M,g^{1 \mapsto x}}) \\ &= \lambda x.\mathsf{met}(g(1))(\mathsf{a}) = \lambda x.\mathsf{met}(x)(\mathsf{a}) \end{aligned}$$

Relative clauses

(1) the man that Ada met



---

1. $[\![t_1]\!]^{M,g} = g(1)$ (the same as a pronoun)
2. $[\![\text{S}']\!]^{M,g} = \lambda x.[\![\text{S}]\!]^{M,g^{1 \mapsto x}} = \lambda x.\mathsf{met}(x)(\mathsf{a})$
3. $[\![\text{the}]\!]^{M,g} = \lambda P.$the entity y s.t. $P(y)$
   defined only if
   $\exists y.[P(y) \wedge \forall x.P(x) \to x = y]$

Predicate Modification

$$\frac{P :: e \to t \qquad Q :: e \to t}{\lambda x.P(x) \wedge Q(x) :: e \to t}$$

---

1. $[\![\text{NP}]\!]^{M,g} = \lambda x.[\![\text{man}]\!]^{M,g}(x) \wedge [\![\text{S}']\!]^{M,g}(x)$
   $= \lambda x.\mathsf{man}(x) \wedge \mathsf{met}(x)(\mathsf{a})$
2. $[\![\text{DP}]\!]^{M,g} = [\![\text{the}]\!]^{M,g}([\![\text{NP}]\!]^{M,g})$
   $=$