

Introduction to Formal Semantics

Tutorial Lecture 4: Typed Lambda Calculus

Nicolaie Dominik Dascalu, Dr. Volha Petukhova
Spoken Language System Group
Saarland University



18.05.22



- **Matrixes**
Exercise 1
- **Types & Lambda Abstraction**
Exercises 2
- **Trees & Lambda reduction**
Exercise 3

Reading:

- Coppock, E., and Champollion, L. (2021). Invitation to formal semantics. Manuscript, Boston University and New York University (Ch.5)





Discussion



- Did you have any difficulties understanding **the main concepts**?
- Were the **exercises** difficult?
- Is there something you would like to review from **tutorial 3**?



Exercises



Exercise 1

$$f_{\text{LOVE}} = \{\langle \text{bonnie}, \text{clyde} \rangle, \langle \text{clyde}, \text{bonnie} \rangle, \langle \text{susan}, \text{megan} \rangle\}$$

$$f_{\text{LOVE}} = \begin{bmatrix} \langle \text{bonnie}, \text{bonnie} \rangle \mapsto 0 \\ \langle \text{bonnie}, \text{clyde} \rangle \mapsto 1 \\ \langle \text{bonnie}, \text{megan} \rangle \mapsto 0 \\ \langle \text{bonnie}, \text{susan} \rangle \mapsto 0 \\ \langle \text{clyde}, \text{clyde} \rangle \mapsto 0 \\ \langle \text{clyde}, \text{bonnie} \rangle \mapsto 1 \\ \langle \text{clyde}, \text{megan} \rangle \mapsto 0 \\ \langle \text{clyde}, \text{susan} \rangle \mapsto 0 \\ \langle \text{megan}, \text{megan} \rangle \mapsto 0 \\ \langle \text{megan}, \text{bonnie} \rangle \mapsto 0 \\ \langle \text{megan}, \text{clyde} \rangle \mapsto 0 \\ \langle \text{megan}, \text{susan} \rangle \mapsto 0 \\ \langle \text{susan}, \text{susan} \rangle \mapsto 0 \\ \langle \text{susan}, \text{bonnie} \rangle \mapsto 0 \\ \langle \text{susan}, \text{clyde} \rangle \mapsto 0 \\ \langle \text{susan}, \text{megan} \rangle \mapsto 1 \end{bmatrix}$$

Characteristic Function

$$f_{\text{LOVE} \leftarrow} = \begin{bmatrix} \text{bonnie} \mapsto \begin{bmatrix} \text{bonnie} \mapsto 0 \\ \text{clyde} \mapsto 1 \\ \text{megan} \mapsto 0 \\ \text{susan} \mapsto 0 \end{bmatrix} \\ \text{clyde} \mapsto \begin{bmatrix} \text{bonnie} \mapsto 1 \\ \text{clyde} \mapsto 0 \\ \text{megan} \mapsto 0 \\ \text{susan} \mapsto 0 \end{bmatrix} \\ \text{megan} \mapsto \begin{bmatrix} \text{bonnie} \mapsto 0 \\ \text{clyde} \mapsto 0 \\ \text{megan} \mapsto 0 \\ \text{susan} \mapsto 1 \end{bmatrix} \\ \text{susan} \mapsto \begin{bmatrix} \text{bonnie} \mapsto 0 \\ \text{clyde} \mapsto 0 \\ \text{megan} \mapsto 0 \\ \text{susan} \mapsto 0 \end{bmatrix} \end{bmatrix}$$

Curried Function



Exercise 2

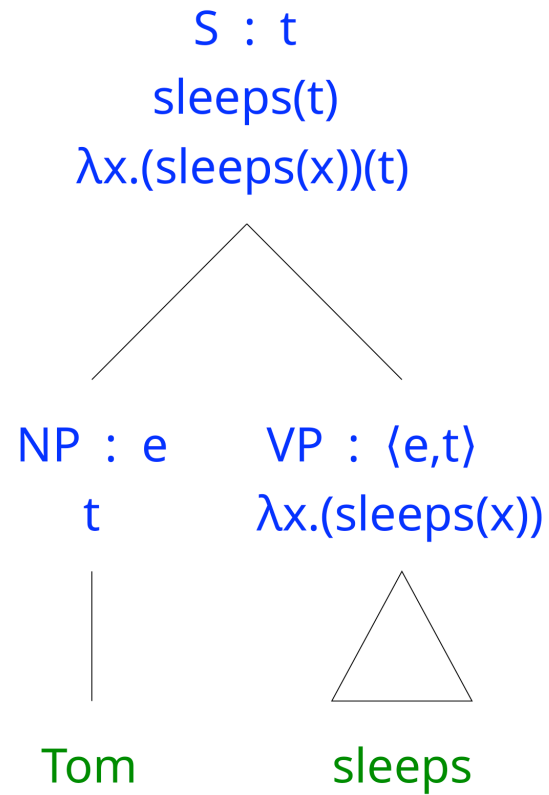
- (1) Mary loves herself : $\langle e, \langle e, t \rangle \rangle \mapsto \lambda x. [\text{loves}(x, x)]$
- (2) All philosophers are evil : $\langle \langle e, t \rangle, \langle \langle e, t \rangle, t \rangle \rangle \mapsto \lambda P. \lambda E. \forall x. [P(x) \rightarrow E(x)]$
- (3) John likes cats : $\langle e, \langle e, t \rangle \rangle \mapsto \lambda x. \lambda y. [\text{likes}(y, x)]$
- (4) Cloe gave Mark the keys : $\langle e, \langle e, \langle e, t \rangle \rangle \rangle \mapsto \lambda x. \lambda y. \lambda z. [\text{gave}(z, y, x)]$
- (5) Yoda floats : $\langle e, t \rangle \mapsto \lambda x. [\text{floats}(x)]$
- (6) Anakin is Luke's father : $\langle e, \langle e, t \rangle \rangle \mapsto \lambda x. \lambda y. [\text{fatherOf}(y, x)]$
- (7) Some bagels are gluten free : $\langle \langle e, t \rangle, \langle \langle e, t \rangle, t \rangle \rangle \mapsto \lambda B. \lambda G. \exists x. [B(x) \wedge G(x)]$
- (8) Eddie lives in Seattle : $\langle e, \langle e, t \rangle \rangle \mapsto \lambda x. \lambda y. [\text{livesIn}(y, x)]$
- (9) Old man : $\langle \langle e, t \rangle, \langle e, t \rangle \rangle \mapsto \lambda O. \lambda M. \lambda x. [O(x) \wedge M(x)]$
- (10) Someone's thirsty : $\langle \langle e, t \rangle, t \rangle \mapsto \lambda P. \exists x. [P(x)]$

P, Q, O, M, are arbitrary.
these are not:
old(x), man(x)



Exercise 3

1



Exercise 3

2

remember that we can
adopt different notation
styles:

Loves(x)(y)

Loves(x,y)

$\lambda x.\lambda y.[likes(y,x)]$

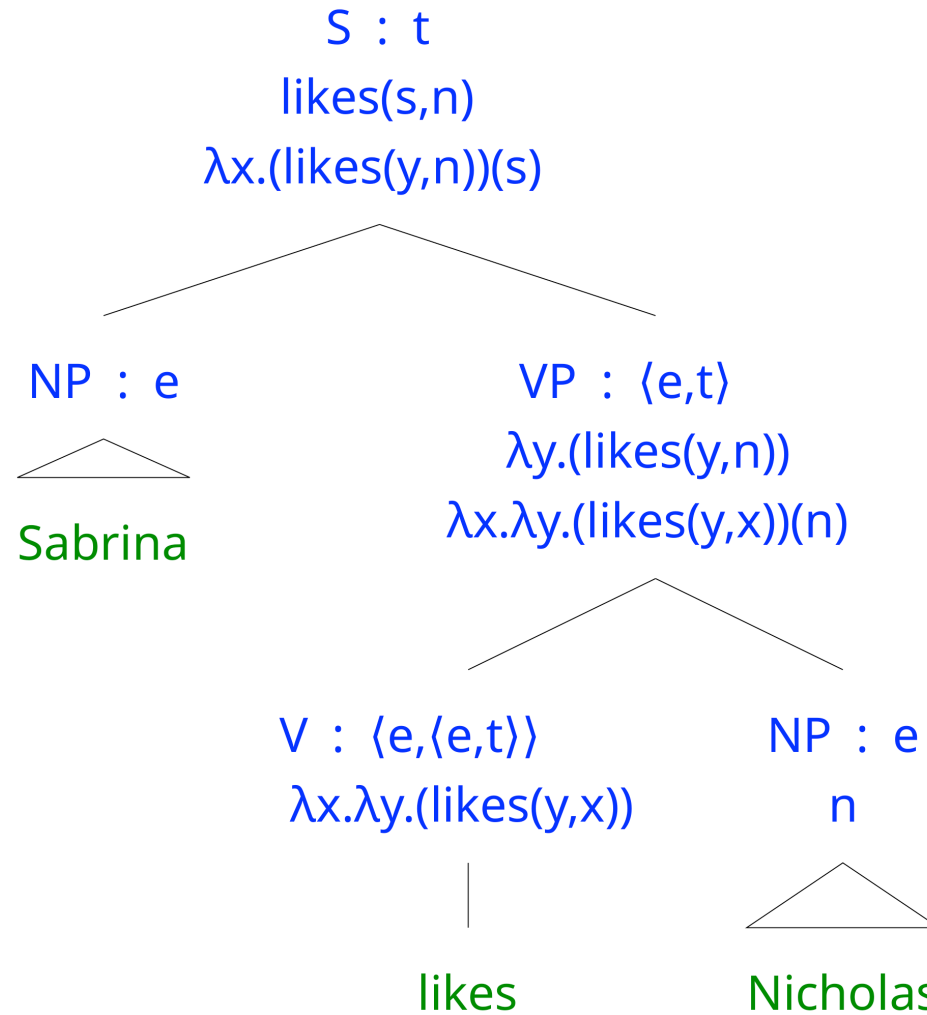
$\lambda y.\lambda x.[likes(x,y)]$

$\lambda y\lambda x[likes(x,y)]$

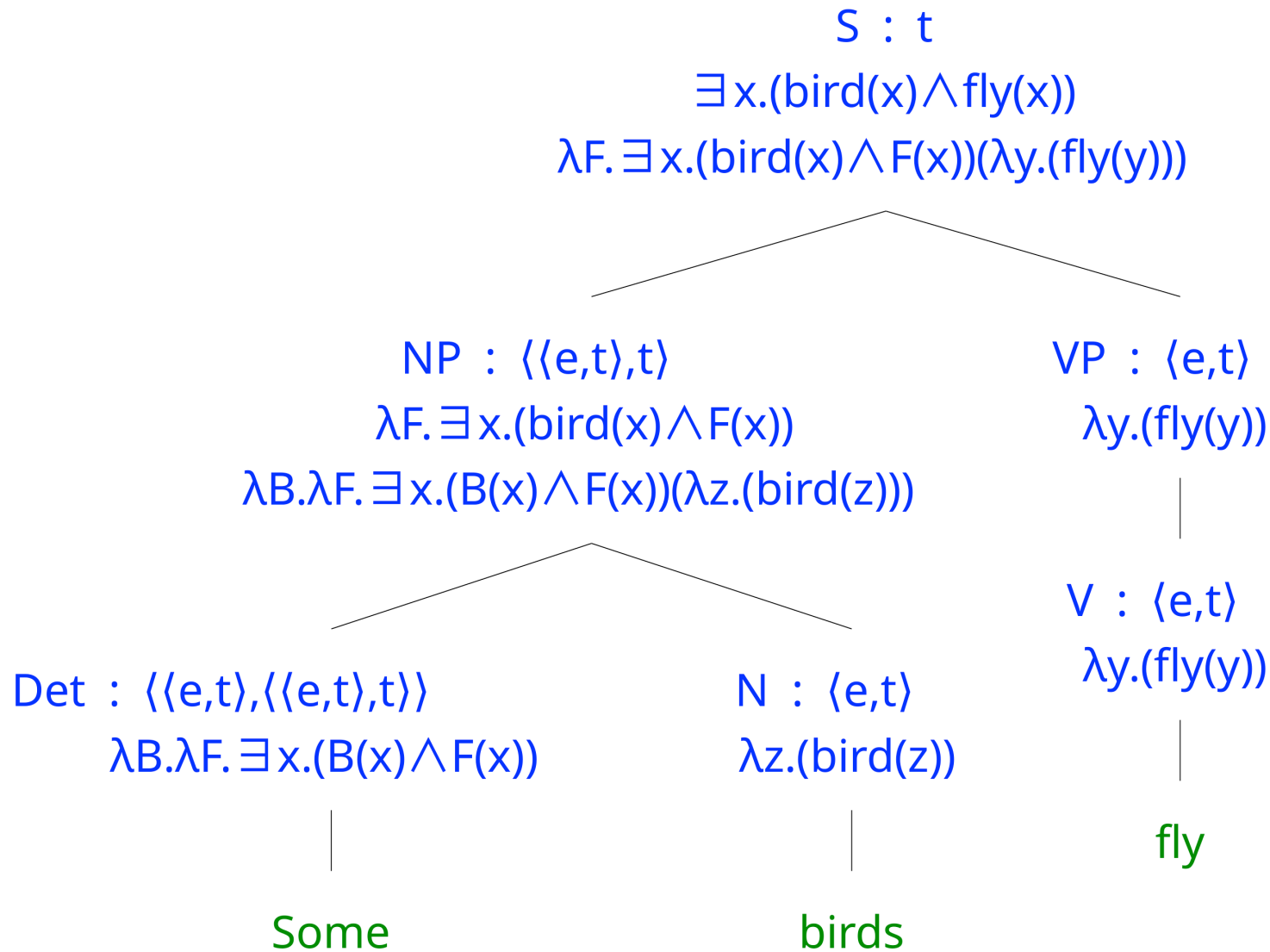
$\lambda y\lambda x[Likes(x,y)]$

$\lambda y\lambda x(Likes(x,y))$

$[\lambda y\lambda x(Likes(x,y))]$




3



**Thank you all
for the kind
attention!**





**If you need further help
or have additional
questions,
please contact us.**

