Frege's principle

 $\underset{\bigcirc}{\mathsf{Typed}} \ \lambda\text{-calculus}$ 

Towards a NL fragment

## Compositionality & $\lambda$ -calculus

Pascal Amsili

January 2023

Sorbonne !!!

Frege's principle

Typed  $\lambda$ -calculus

Towards a NL fragment

### Overview

### Untyped (pure) $\lambda$ -calculus

Sorbonne III Nouvelle III



Syntax

Let V be a countable set of variables. The set of all well-formed terms,  $\Lambda$ , is defined inductively as follows :



•  $(t_1)t_2 \in \Lambda$   $\forall t_1, t_2 \in \Lambda$ 



 $\underset{\bigotimes}{\mathsf{Typed}} \ \lambda\text{-calculus}$ 

Towards a NL fragment

# Syntax (cont'd)

### The term $(\lambda x.(x)x)y$ has this syntactic structure :

### af(la(x, af(x, x)), y)



Sorbonne III Nouvelle III



Typed  $\lambda$ -calculus

Towards a NL fragment

### Variable substitution

• 
$$x_{[x:=z]} \rightsquigarrow z$$
  
•  $y_{[x:=z]} \rightsquigarrow y \text{ si } y \neq x$   
•  $(M)N_{[x:=z]} \rightsquigarrow (M_{[x:=z]})N_{[x:=z]}$   
•  $\lambda x.M_{[x:=z]} \rightsquigarrow \lambda z.M_{[x:=z]}$   
•  $\lambda y.M_{[x:=z]} \rightsquigarrow \lambda y.M_{[x:=z]}$  if  $x \neq y$ 

Sorbonne ;;; Nouvelle ;;;



Typed  $\lambda$ -calculus

Towards a NL fragment

### Term substitution

• 
$$x_{[x:=t]} \rightsquigarrow t$$
  
•  $y_{[x:=t]} \rightsquigarrow y \text{ si } y \neq x$   
•  $(M)N_{[x:=t]} \rightsquigarrow (M_{[x:=t]})N_{[x:=t]}$   
•  $\lambda y.M_{[x:=t]} \rightsquigarrow \lambda y.M_{[x:=t]}$  if y is not free in t.

Sorbonne !!!



Frege's principle

Typed  $\lambda$ -calculus

Towards a NL fragment

### $\alpha$ equivalence

$$\lambda x.\varphi \equiv \lambda z.\varphi_{[x:=z]}$$

Frege's principle

Typed  $\lambda$ -calculus

Towards a NL fragment

### Convention on variables

Let M be a term, x a variable. By convention, the occurrences of x in M are either all free or all bound.

It can be shown that every term constructed without respecting this convention is  $\alpha$ -equivalent to a term that respects the convention.

Sorbonne ;;;



Frege's principle

Typed  $\lambda$ -calculus

Towards a NL fragment

# $\beta$ equivalence

 $(\lambda x.M)N \equiv M_{[x:=N]}$ 



Sorbonne ;;; Nouvelle ;;;



Frege's principle

Typed  $\lambda$ -calculus

Towards a NL fragment

## Combinators

### A combinator is a closed $\lambda$ -term (ie without free variable)

Sorbonne ;;; Nouvelle ;;;



### Identity

$$I =_{def} \lambda x.x$$

For any term  $t : (I)t \equiv t$ 

Sorbonne ;;; Nouvelle ;;;





$$T =_{\text{def}} \lambda x.\lambda y.x$$
$$F =_{\text{def}} \lambda x.\lambda y.y$$

This encoding allows to encode an if-then-else function :

if P then Q else 
$$R =_{\text{def}} ((P)Q)R$$
.

if P is  $\beta$ -equivalent to T then ((P)Q)R will yield Q, while if P is  $\beta$ -equivalent (or  $\beta$ -reduces) to F, the outcome will be Q.

Untyped (pure)  $\lambda\text{-calculus}$   $_{\odot}$ 

00000

Frege's principle

Typed  $\lambda$ -calculus

Towards a NL fragment

## The IF combinator

$$\mathsf{IF} =_{\mathrm{def}} \lambda b. \lambda t. \lambda f. ((b)t) f$$

$$\begin{array}{l} \mathsf{NOT} =_{\mathrm{def}} \lambda u.((u)\mathsf{F})\mathsf{T} \\ \mathsf{AND} =_{\mathrm{def}} \lambda u.\lambda v.((u)v)\mathsf{F} \\ \mathsf{OR} =_{\mathrm{def}} \lambda u.\lambda v.((u)\mathsf{T})v \end{array}$$

Sorbonne !!!

00000

Frege's principle

Typed  $\lambda$ -calculus

Towards a NL fragment

## Church numerals

$$0 =_{def} \lambda f . \lambda x. x$$
  

$$1 =_{def} \lambda f . \lambda x. (f) x$$
  

$$n =_{def} \lambda f . \lambda x. (f) (f) ... (f) x$$
  
with n times f

Succ =<sub>def</sub> 
$$\lambda n. \lambda f. \lambda x.(f)((n)f)x$$
  
+  $\equiv \lambda m. \lambda n. \lambda f. \lambda x.((m)f)((n)f)x$   
\*  $\equiv \lambda m. \lambda n. \lambda f.(m)(n)f$ 

Sorbonne !!!

Frege's principle

Typed  $\lambda$ -calculus

Towards a NL fragment

## Overview

### Untyped (pure) $\lambda$ -calculus

Syntax Substitution Equivalences Combinators

### Frege's principle

Typed  $\lambda$ -calculus

Type theory Montague's language

#### Towards a NL fragment

Simple sentence Roadmap for the fragment Quantified sentences Excursus : Generalized Quantifiers Transitive verbs Negation Other phenomena

Untyped (pure) λ-calculus

Frege's principle

Typed  $\lambda$ -calculus

Towards a NL fragment

# Frege's principle

The meaning of an expression is uniquely determined by the meanings of its parts and their mode of combination.

Motivation : Humbolt's view on finite means for infinite sentences

Sorbonne ;;;



## Consequences

- Locality principle : lexical items have a meaning that is independant of the expression they occur in.
- Substitution principle : synonymous expressions may be substituted for each other without changing the meaning of the complex expression in which they occur.
- Parts of well formed sentences have « meaning »
- Meanings can be « composed » : Frege's saturation idea

 $\lambda$ -terms can represent individual meanings and functional application can represent semantic composition.

Frege's principle



Towards a NL fragment

### Overview

### Untyped (pure) $\lambda$ -calculus

Syntax Substitution Equivalences Combinators

Frege's principle

#### Typed $\lambda$ -calculus

Type theory Montague's language

#### Towards a NL fragment

Simple sentence Roadmap for the fragment Quantified sentences Excursus : Generalized Quantifiers Transitive verbs Negation Other phenomena



Typed  $\lambda$ -calculus

Towards a NL fragment



- 1. *e* is a type
- 2. t is a type
- 3. if a and b are types, then  $\langle a, b \rangle$  is a type

Sorbonne ;;; Nouvelle ;;;





Towards a NL fragment



- 1. e is a type
- 2. t is a type
- 3. if a and b are types, then  $\langle a, b \rangle$  is a type
  - $D_e = A$
- $D_t = \{0, 1\}$
- $D_{\langle a,b
  angle} =$  the set of mappings from  $D_a$  to  $D_b$ .

Frege's principle





# Meaningful expressions

For a, b types :

- variables and individual constants of type *a* belong to *ME<sub>a</sub>*.
- if  $\alpha \in ME_{\langle a,b \rangle}$  and  $\beta \in ME_a$  then  $(\alpha)\beta \in ME_b$ .
- if u is a variable of type a and  $\alpha \in ME_b$ , then  $\lambda u.\alpha \in ME_{(a,b)}$ .
- if  $\varphi$  and  $\psi$  are in  $ME_t$ , then the following expressions are also in  $ME_t$ :  $\neg \varphi$ ,  $(\varphi \land \psi)$ ,  $(\varphi \lor \psi)$ ,  $(\varphi \to \psi)$ .
- if  $\varphi$  is in  $ME_t$  and u is a type a variable, then  $\forall u\varphi$  and  $\exists u\varphi$  are in  $ME_t$ .

Sorbonne III Nouvelle III

Frege's principle

Typed  $\lambda$ -calculus

Towards a NL fragment

### Overview

#### Untyped (pure) $\lambda$ -calculus

Syntax Substitution Equivalences Combinators rege's principle

#### Typed $\lambda$ -calculus

Type theory Montague's language

#### Towards a NL fragment

Simple sentence Roadmap for the fragment Quantified sentences Excursus : Generalized Quantifiers Transitive verbs Negation Other phenomena

Untyped (pure) $\lambda$ -calculus	Frege's principle	Typed $\lambda$ -calculus	Towards a NL fragment
000			0 000000 0000000 0000000



S	$\rightarrow$	NP	VP
[[ <i>S</i> ]]	$\leftarrow$	([[ <i>VP</i> ]])	[ <i>NP</i> ]
0	$\leftarrow$	(2)	1
NP	$\rightarrow$	PN	
0	$\leftarrow$	1	
VP	$\rightarrow$	V	
0	$\leftarrow$	1	
0 <i>PN</i>	$\leftarrow \rightarrow$	1 Sam	
0 <i>PN</i> 0	$\leftarrow \to \leftarrow \leftarrow$	1 Sam s	
0 <i>PN</i> 0 <i>V</i>	$\begin{array}{c} \leftarrow \\ \rightarrow \\ \leftarrow \\ \rightarrow \end{array}$	1 Sam <i>s</i> sleeps	

Sorbonne ;;; Nouvelle ;;;

Untyped (pure) λ-calculus

Frege's principle

Typed  $\lambda$ -calculus

Towards a NL fragment

# Roadmap for the fragment

- A cat enters
- Sam likes Pam
- Everyone likes Pam
- Everyone likes an actress
- Sam is mortal
- Sam met a tall person
- Sam doesn't sleep

 $\exists x (Cx \land Ex) \\ Lsp (or ((L)s)p \\ \forall x (Px \rightarrow Lxp) \\ \forall x (Px \rightarrow \exists y (Ay \land Lxy)) \\ Ms \\ \exists x ((Px \land Tx) \land Msx) \\ \neg Ss$