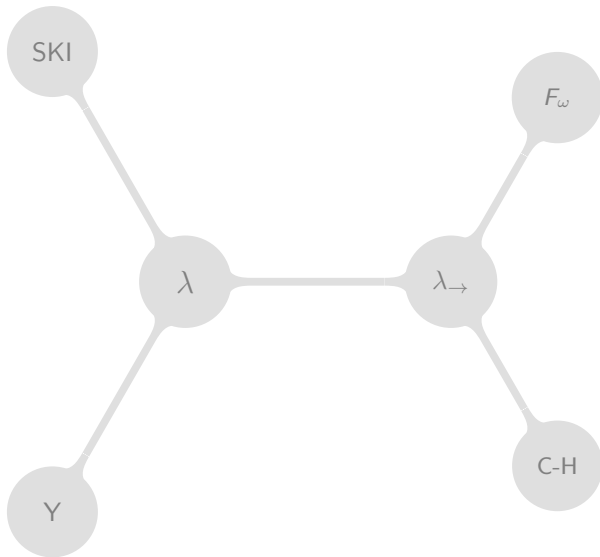


Introduction to Lambda Calculus

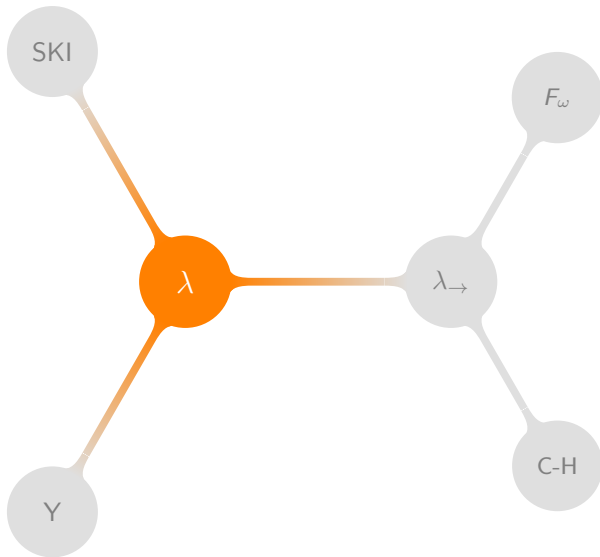
Maciek Makowski (@mmakowski)

30th September 2014

The Plan



Basic Lambda Calculus



Syntax

$\langle term \rangle ::= x$	(variable)
$(\lambda x. \langle term \rangle)$	(abstraction)
$(\langle term \rangle \langle term \rangle)$	(application)

where $x \in \mathbb{X}$ – the set of variables

Syntax

v_1

Syntax

v_1

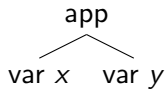
var v_1

Syntax

$x\ y$

Syntax

$x\ y$



Syntax

$\lambda a.b$

Syntax

$\lambda a.b$

abs a
|
var b

Syntax

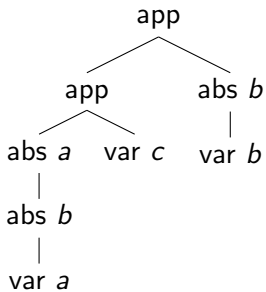
$(\lambda a. \lambda b. a) \ c \ (\lambda b. b)$

Syntax

$\langle term \rangle ::= x$	(variable)
$(\lambda x. \langle term \rangle)$	(abstraction)
$(\langle term \rangle \langle term \rangle)$	(application)

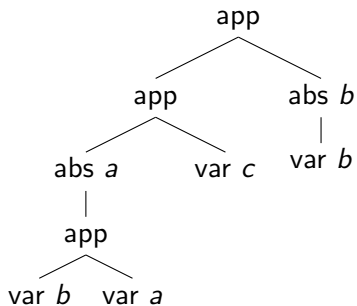
Syntax

$(\lambda a. \lambda b. a) c (\lambda b. b)$



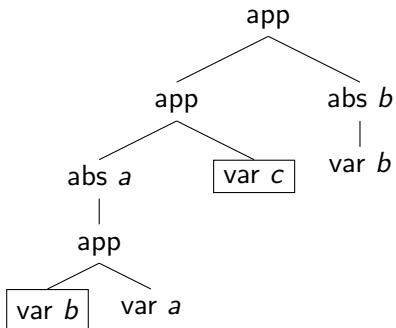
Syntax

$(\lambda a. b \ a) \ c \ (\lambda b. b)$



Syntax

$(\lambda a. \underline{b} \ a) \ \underline{c} \ (\lambda b. b)$



Rewriting

$$(\lambda x.x\ y)\ (\lambda x.x) \longleftarrow_{\alpha} (\lambda a.a\ y)\ (\lambda b.b)$$

Rewriting

$$(\lambda x. x \ y) \ (\lambda z. z) \longrightarrow_{\beta} (\lambda z. z) \ y \longrightarrow_{\beta} y$$

Rewriting

TODO: multiple redexes, Church-Rosser

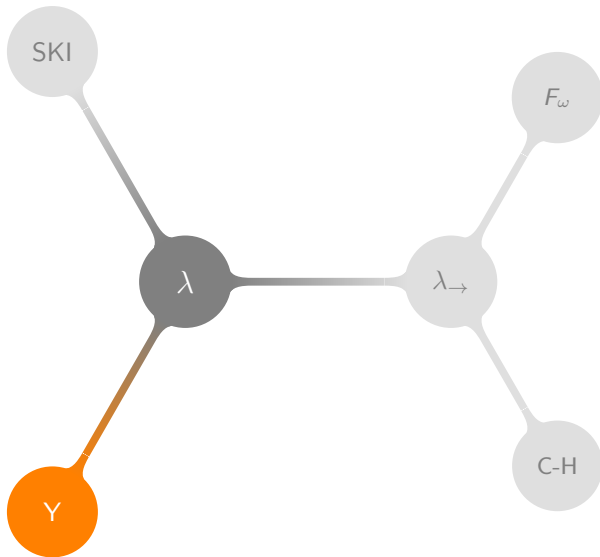
Rewriting

TODO: Ω

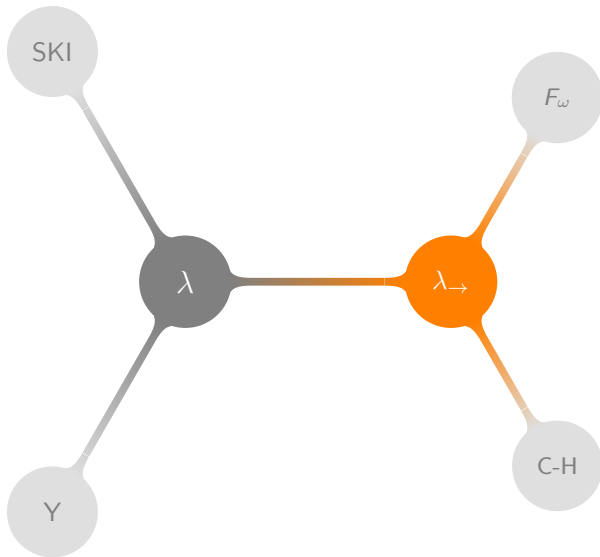
Semantics

TODO: functions

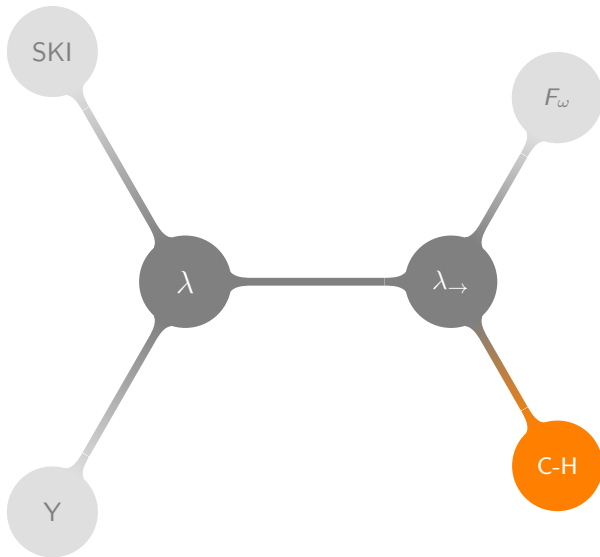
Programming in Lambda Calculus



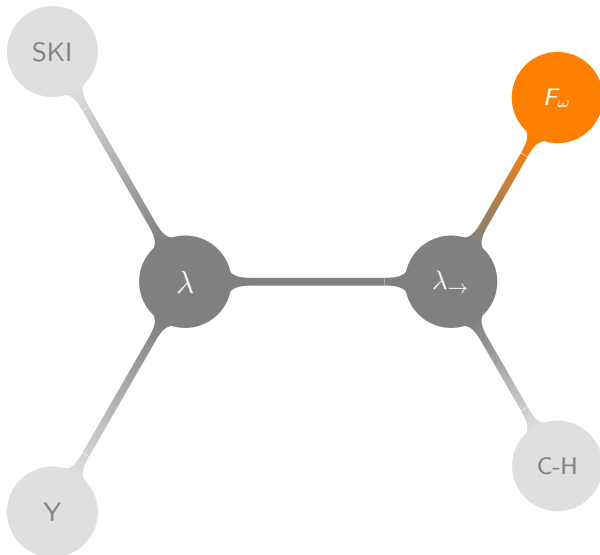
Simple Types



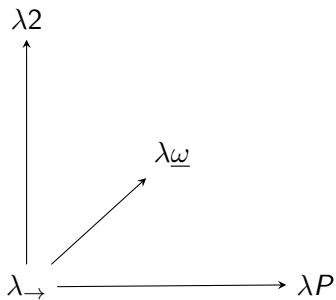
Curry-Howard Correspondence



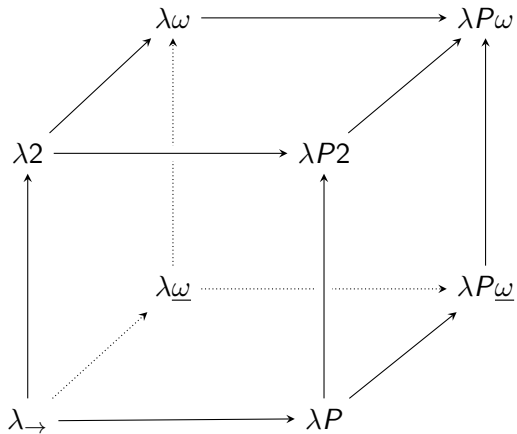
More Types



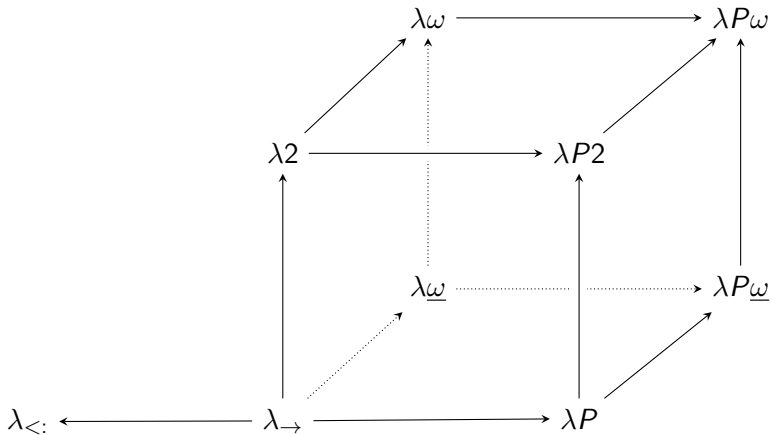
The Lambda Cube



The Lambda Cube



Subtyping



Subtyping

