Lambda calculs et catégories

Paul-André Melliès

Master Parisien de Recherche en Informatique

Ecole Normale Supérieure

Synopsis of the lecture

1 – The lambda-calculus

2 – The simply-typed lambda-calculus

3 – Elements of rewriting theory

4 – The lambda-calculus with explicit substitutions

First part

Lambda-calculus

The calculus of functions

The pure λ-calculus

Terms $\quad M \quad ::= \quad x \quad | \quad MN \quad | \quad \lambda x.M$

The β-reduction:

$$(\lambda x.M)\,N \quad \longrightarrow \quad M\,[x := N]$$

The η-expansion:

$$M \quad \longrightarrow \quad \lambda x.\,(M\,x)$$

Remark: every term is considered up to renaming $\equiv_\alpha$ of the bound variables, typically:

$$\lambda x.\lambda y.x \quad \equiv_\alpha \quad \lambda z.\lambda y.z$$

Occurrences

The set of occurrences of a λ-term M is defined by induction:

- $\mathbf{occ}(x) = \{\varepsilon\}$
- $\mathbf{occ}(MN) = \{\varepsilon\} \cup \{1 \cdot o \mid o \in \mathbf{occ}(M)\} \cup \{2 \cdot o \mid o \in \mathbf{occ}(N)\}$
- $\mathbf{occ}(\lambda x.M) = \{\varepsilon\} \cup \{1 \cdot o \mid o \in \mathbf{occ}(M)\}$

Note that every occurrence of the λ-term M is labelled by

- an application node App
- a binder λx
- a variable x

Free variables

The set of **free variables** of a λ-term is defined by induction:

- $FV(x) = \{x\}$
- $FV(MN) = FV(M) \cup FV(N)$
- $FV(\lambda x.M) = FV(M) \setminus \{x\}$

Every occurrence of a variable x in a λ-term is

- either free
- or bound by a binder λx above it in the λ-term.

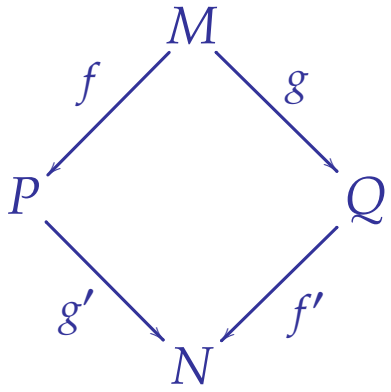
Church-Rosser theorem

Also called confluence theorem.

Given two β-rewriting paths

$$f : M \xrightarrow{*} P \qquad\qquad g : M \xrightarrow{*} Q$$

there exists a λ-term N and two β-rewriting paths f' and g' completing the diagram as



The simply-typed λ-calculus

It is possible to **type** the expressions of the λ-calculus using simple types A, B constructed by the grammar:

$$A, B \ ::= \ \alpha \mid A \Rightarrow B.$$

A **typing context** Γ is a finite sequence

$$\Gamma = (x_1 : A_1, ..., x_n : A_n)$$

where each x_i is a variable and each A_i is a simple type.

A **sequent** is a triple

$$x_1 : A_1, ..., x_n : A_n \vdash P : B$$

where

$$x_1 : A_1, ..., x_n : A_n$$

is a typing context, P is a λ-term and B is a simple type.

The simply-typed λ-calculus

Variable
$$\frac{}{x : A \vdash x : A}$$

Abstraction
$$\frac{\Gamma, x : A \vdash P : B}{\Gamma \vdash \lambda x.P : A \Rightarrow B}$$

Application
$$\frac{\Gamma \vdash P : A \Rightarrow B \qquad \Delta \vdash Q : A}{\Gamma, \Delta \vdash PQ : B}$$

Weakening
$$\frac{\Gamma \vdash P : B}{\Gamma, x : A \vdash P : B}$$

Contraction
$$\frac{\Gamma, x : A, y : A \vdash P : B}{\Gamma, z : A \vdash P[x, y \leftarrow z] : B}$$

Exchange
$$\frac{\Gamma, x : A, y : B, \Delta \vdash P : C}{\Gamma, y : B, x : A, \Delta \vdash P : C}$$

Subject reduction

A λ-term P is simply typed when there exists a sequent

$$\Gamma \quad \vdash \quad P : A$$

which may be obtained by a derivation tree.

One establishes that the set of simply typed λ-terms is closed under β-réduction:

Subject Reduction:

If $\Gamma \vdash P : A$ and $P \longrightarrow_\beta Q$, then $\Gamma \vdash Q : A$.

Strong normalization

A λ-term P is strongly normalizing when there exists no infinite sequence of β-reductions:

$$P \longrightarrow_\beta P_1 \longrightarrow_\beta P_2 \longrightarrow_\beta \cdots \longrightarrow_\beta P_n \longrightarrow_\beta \cdots$$

Strong normalization:

Every simply typed λ-term P is strongly normalizing.

In particular, the λ-term $\Delta\Delta$ loops:

$$\Delta\Delta \quad \longrightarrow_\beta \quad \Delta\Delta \quad \longrightarrow_\beta \quad \cdots$$

is not simply typed.

minimal logic

Curry-Howard (1)

Variable

$$\frac{}{A \vdash A}$$

Abstraction

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \Rightarrow B}$$

Application

$$\frac{\Gamma \vdash A \Rightarrow B \qquad \Delta \vdash A}{\Gamma, \Delta \vdash B}$$

Weakening

$$\frac{\Gamma \vdash B}{\Gamma, A \vdash B}$$

Contraction

$$\frac{\Gamma, A, A \vdash B}{\Gamma, A \vdash B}$$

Exchange

$$\frac{\Gamma, A, B, \Delta \vdash C}{\Gamma, B, A, \Delta \vdash C}$$

Curry-Howard (1)

simply typed λ-calculus

Variable
$$\frac{}{x : A \vdash x : A}$$

Abstraction
$$\frac{\Gamma, x : A \vdash P : B}{\Gamma \vdash \lambda x.P : A \Rightarrow B}$$

Application
$$\frac{\Gamma \vdash P : A \Rightarrow B \qquad \Delta \vdash Q : A}{\Gamma, \Delta \vdash PQ : B}$$

Weakening
$$\frac{\Gamma \vdash P : B}{\Gamma, x : A \vdash P : B}$$

Contraction
$$\frac{\Gamma, x : A, y : A \vdash P : B}{\Gamma, z : A \vdash P[x, y \leftarrow z] : B}$$

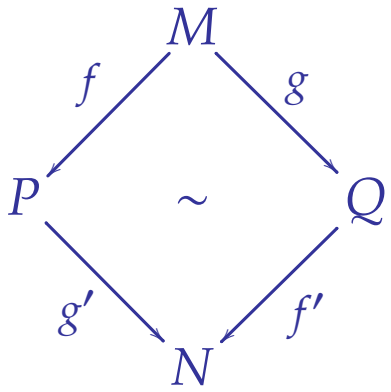
Exchange
$$\frac{\Gamma, x : A, y : B, \Delta \vdash P : C}{\Gamma, y : B, x : A, \Delta \vdash P : C}$$

Algebraic Church-Rosser Theorem

Given two β-rewriting paths

$$f \;:\; M \xrightarrow{*} P \qquad\qquad g \;:\; M \xrightarrow{*} Q$$

there exists a λ-term N and two β-rewriting paths f' and g' completing the diagram as



where $\sim$ denotes the permutation equivalence on rewriting paths.

Theorem established by Jean-Jacques Lévy in 1978

Redex

Definition. A β-redex is a pair

$$(M, o)$$

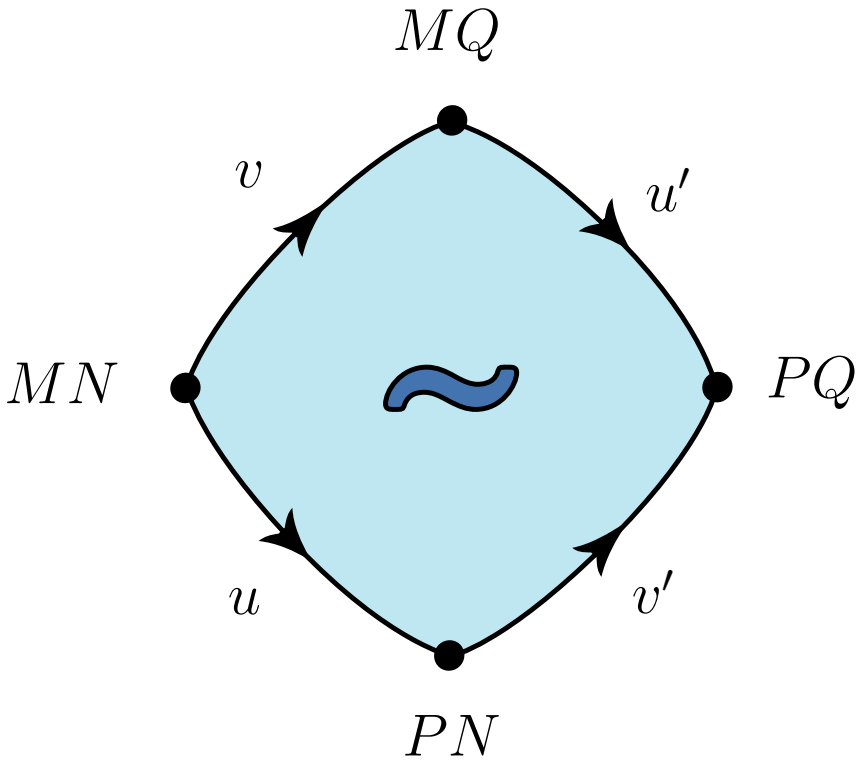
consisting of

- a λ-term M
- an occurrence of the λ-term M such that

$$M_{|o} \quad = \quad (\lambda x.P)\, Q$$

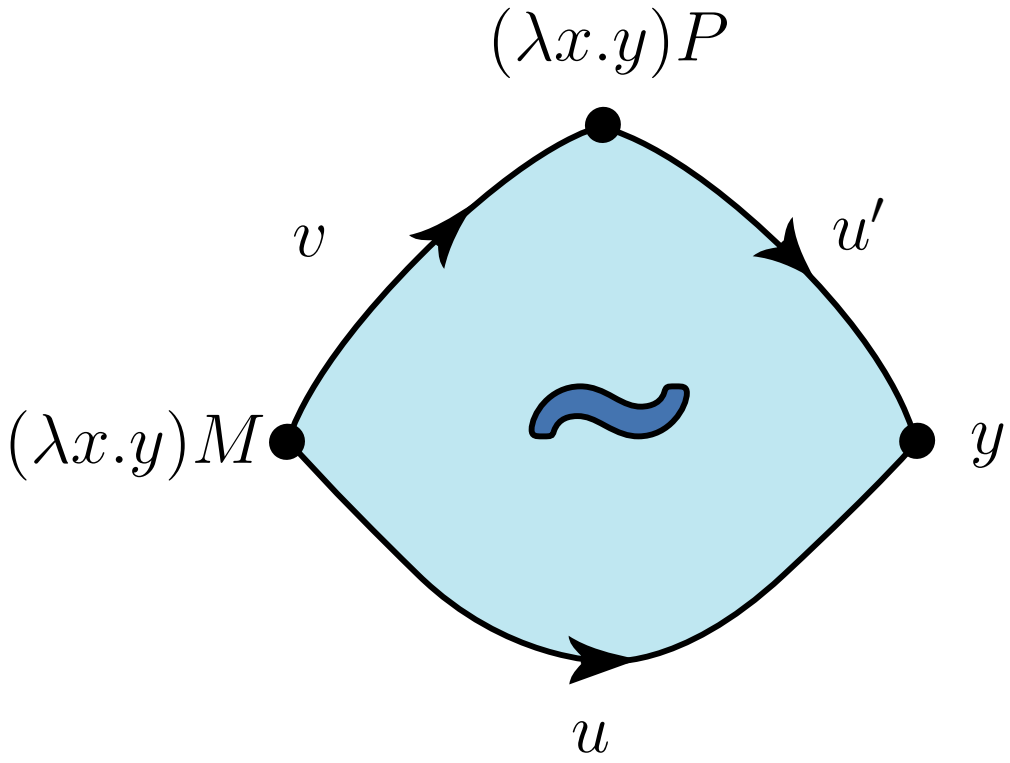
is a β-reduction pattern.

Redex permutations



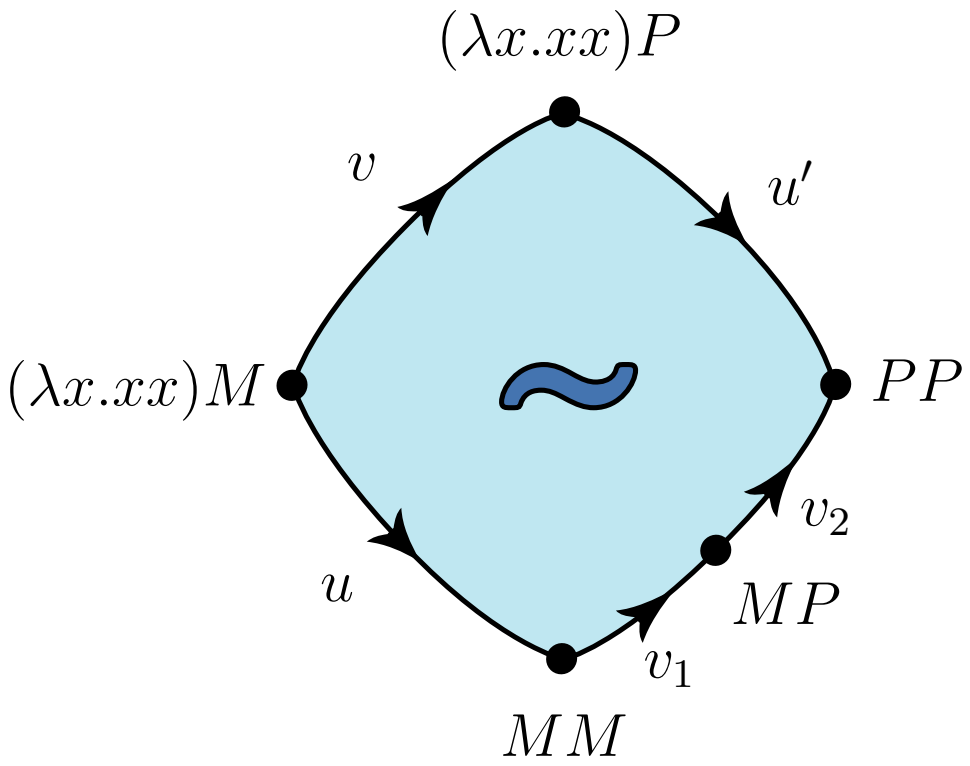
The two redexes $u : M \to P$ and $v : N \to Q$ are disjoint.

Redex permutations



The redex u erases the redex $v : M \to P$.

Redex permutations



The redex u duplicates the redex $v : M \to P$.

Rewriting paths modulo permutations

An important problem of rewriting theory: compare the several paths which rewrite **a λ-term P into its normal form Q.**

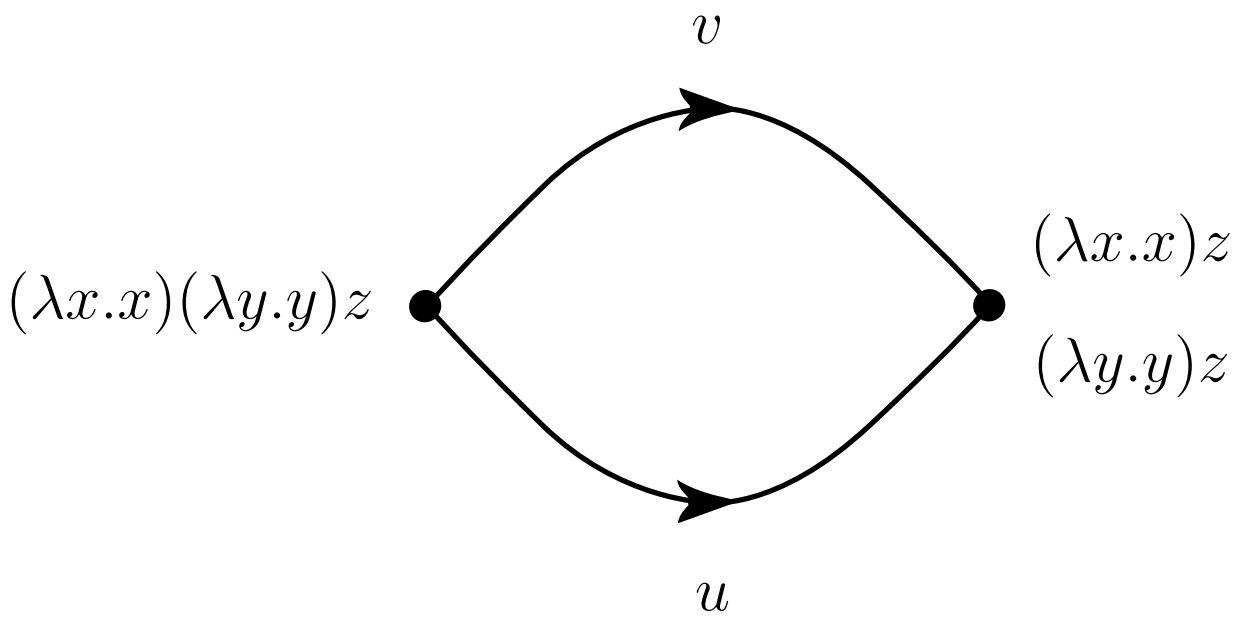
Corollary

Every two rewriting paths to the normal form

$$f, g \quad : \quad P \quad \longrightarrow \quad Q$$

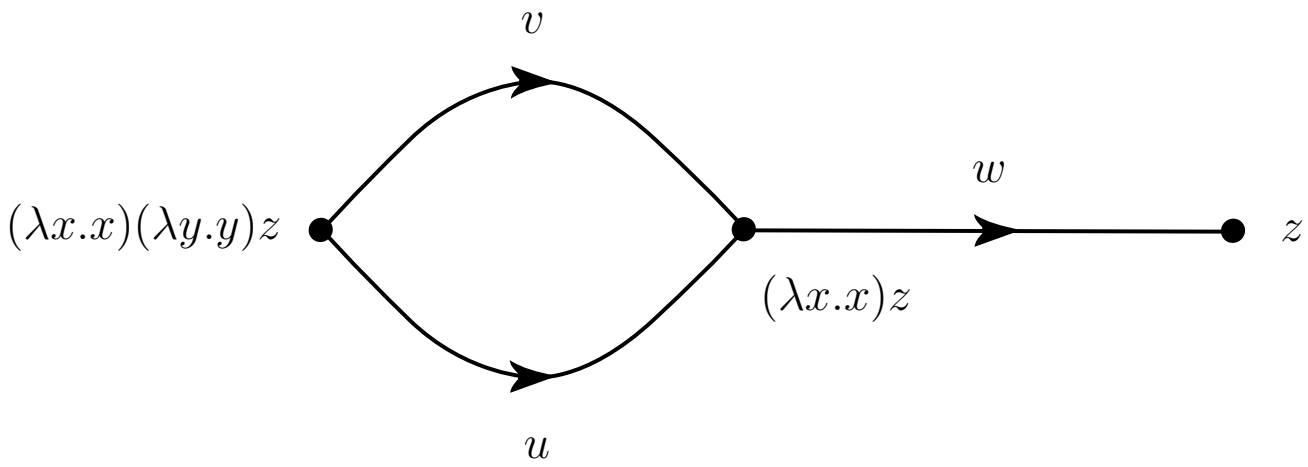
are equal modulo a series of redex permutations.

A 2-dimensional hole



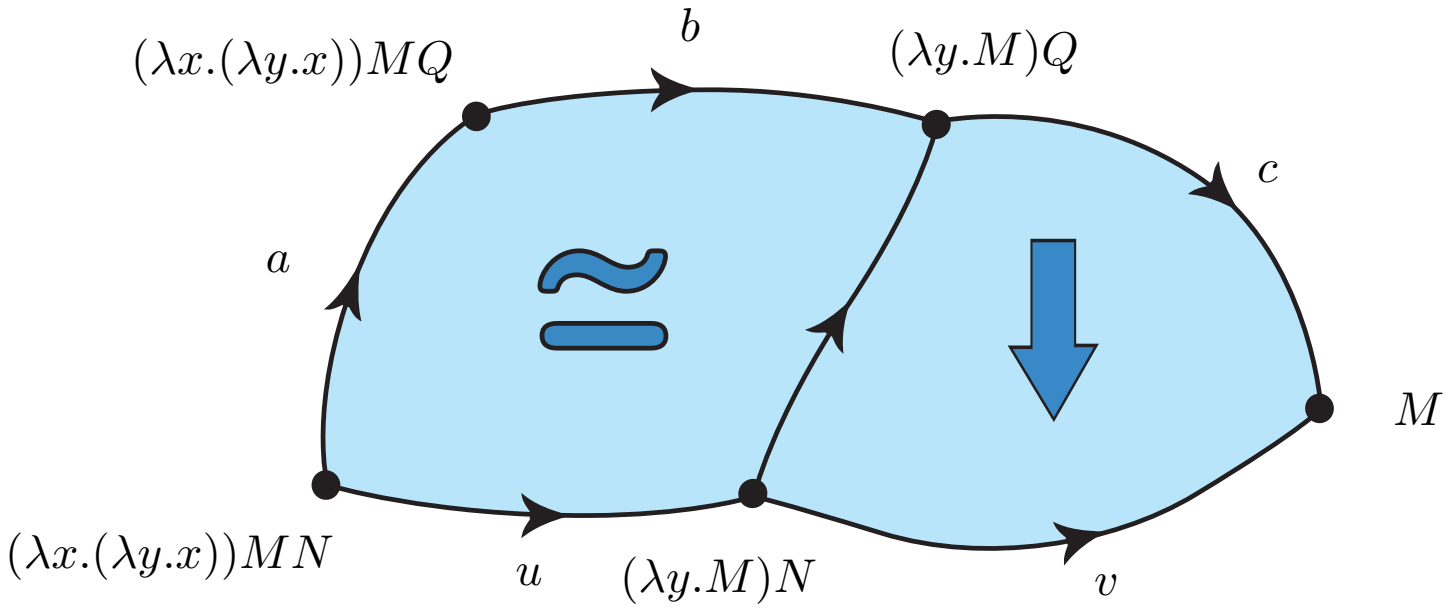
The two redexes u and v are not equivalent modulo permutation.

The 2-dimensional hole continued



The two paths $u \cdot w$ and $v \cdot w$ are equivalent modulo permutation.

Geometry of rewriting



A standardization theorem will be established in the course

The λ-calculus with de Bruijn indices

Variable
$$\frac{}{\Gamma, A \vdash 1 : A}$$

Abstraction
$$\frac{\Gamma, A \vdash P : B}{\Gamma \vdash \lambda\, P : A \Rightarrow B}$$

Application
$$\frac{\Gamma \vdash P : A \Rightarrow B \qquad \Gamma \vdash Q : A}{\Gamma \vdash P\, Q : B}$$

Weakening
$$\frac{\Gamma \vdash P : B}{\Gamma, A \vdash P\,[\uparrow]\ : B}$$

where $P\,[\uparrow]$ denotes the λ-term P where each free variable has been incremented.

The λ-calculus with explicit substitutions

Terms $\quad M \quad ::= \quad \mathbf{1} \quad | \quad MN \quad | \quad \lambda M \quad | \quad M[s]$

Substitutions $\quad s \quad ::= \quad id \quad | \quad \uparrow \quad | \quad M \cdot s \quad | \quad s \circ t$

Key idea: replace the β-rule of the λ-calculus

$$(\lambda x.M)\, N \quad \longrightarrow \quad M\,[x := N]$$

by the Beta-rule of the $\lambda\sigma$-calculus

$$(\lambda\, M)\, N \quad \longrightarrow \quad M\,[N \cdot id]$$

where the substitution is explicit – and thus similar to a closure.

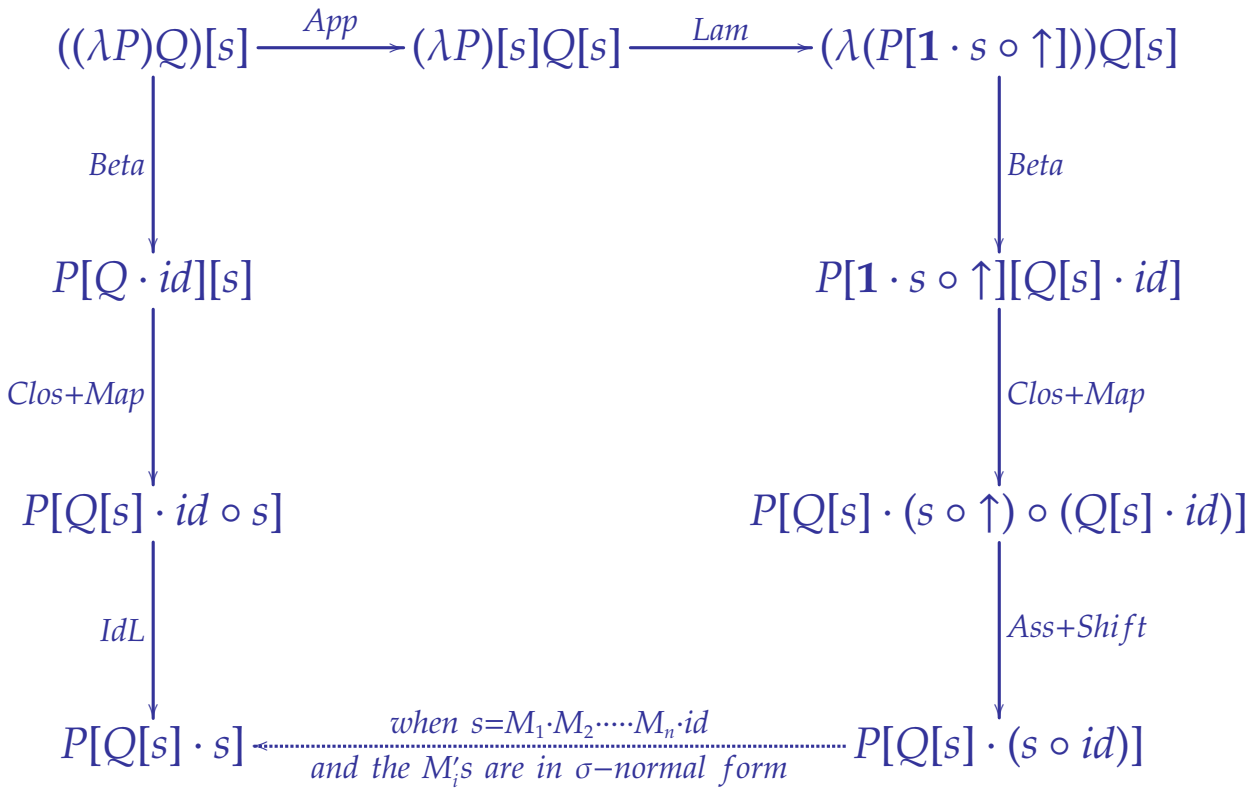
The eleven rewriting rules of the $\lambda\sigma$-calculus

Beta	$(\lambda M)N$	$\rightarrow$	$M[N \cdot id]$
App	$(MN)[s]$	$\rightarrow$	$M[s]N[s]$
Abs	$(\lambda M)[s]$	$\rightarrow$	$\lambda(M[\mathbf{1} \cdot (s \circ \uparrow)])$
Clos	$M[s][t]$	$\rightarrow$	$M[s \circ t]$
VarCons	$\mathbf{1}[M \cdot s]$	$\rightarrow$	M
VarId	$\mathbf{1}[id]$	$\rightarrow$	$\mathbf{1}$
Map	$(M \cdot s) \circ t$	$\rightarrow$	$M[t] \cdot (s \circ t)$
IdL	$id \circ s$	$\rightarrow$	s
Ass	$(s_1 \circ s_2) \circ s_3$	$\rightarrow$	$s_1 \circ (s_2 \circ s_3)$
ShiftCons	$\uparrow \circ (M \cdot s)$	$\rightarrow$	s
ShiftId	$\uparrow \circ id$	$\rightarrow$	$\uparrow$

The eleven critical pairs of the $\lambda\sigma$-calculus

$App + Beta$	$(\lambda M)[s](N[s])$	$\overset{App}{\leftarrow}$	$((\lambda M)N)[s]$	$\overset{Beta}{\rightarrow}$	$M[N \cdot id][s]$
$Clos + App$	$(MN)[s \circ t]$	$\overset{Clos}{\leftarrow}$	$(MN)[s][t]$	$\overset{App}{\rightarrow}$	$(M[s](N[s]))[t]$
$Clos + Abs$	$(\lambda M)[s \circ t]$	$\overset{Clos}{\leftarrow}$	$(\lambda M)[s][t]$	$\overset{Abs}{\rightarrow}$	$(\lambda(M[\mathbf{1} \cdot s \circ \uparrow]))[t]$
$Clos + VarId$	$\mathbf{1}[id \circ s]$	$\overset{Clos}{\leftarrow}$	$\mathbf{1}[id][s]$	$\overset{VarId}{\rightarrow}$	$\mathbf{1}[s]$
$Clos + VarCons$	$\mathbf{1}[(M \cdot s) \circ t]$	$\overset{Clos}{\leftarrow}$	$\mathbf{1}[M \cdot s][t]$	$\overset{VarCons}{\rightarrow}$	$M[t]$
$Clos + Clos$	$M[s][t \circ t']$	$\overset{Clos}{\leftarrow}$	$M[s][t][t']$	$\overset{Clos}{\rightarrow}$	$M[s \circ t][t']$
$Ass + Map$	$(M \cdot s) \circ (t \circ t')$	$\overset{Ass}{\leftarrow}$	$((M \cdot s) \circ t) \circ t'$	$\overset{Map}{\rightarrow}$	$(M[t] \cdot s \circ t) \circ t'$
$Ass + IdL$	$id \circ (s \circ t)$	$\overset{Ass}{\leftarrow}$	$(id \circ s) \circ t$	$\overset{IdL}{\rightarrow}$	$s \circ t$
$Ass + ShiftId$	$\uparrow \circ (id \circ s)$	$\overset{Ass}{\leftarrow}$	$(\uparrow \circ id) \circ s$	$\overset{ShiftId}{\rightarrow}$	$\uparrow \circ s$
$Ass + ShiftCons$	$\uparrow \circ ((M \cdot s) \circ t)$	$\overset{Ass}{\leftarrow}$	$(\uparrow \circ (M \cdot s)) \circ t$	$\overset{ShiftCons}{\rightarrow}$	$s \circ t$
$Ass + Ass$	$(s \circ s') \circ (t \circ t')$	$\overset{Ass}{\leftarrow}$	$((s \circ s') \circ t) \circ t'$	$\overset{Ass}{\rightarrow}$	$(s \circ (s' \circ t)) \circ t'$

A dangerous critical pair



This critical pair leads to a counter-example to **strong normalization** of the simply-typed $\lambda\sigma$-calculus.