

An opinionated introduction to

Call-by-Push-Value

Guillaume Munch-Maccagnoni

Inria

Team Gallinette, Nantes

January 10th 2020
Coq Andes Summer School



Goal & Motivation

Call-by-Push-Value is:

- A syntax : that extends the λ -calculus with *sums* and control over *evaluation order* (= compatible with side-effects), that decomposes both *call-by-value* and *call-by-name* evaluation strategies.
- A model : an axiomatic notion of *denotational semantics* (= interpret derivations as mathematical objects) that unifies various pre-existing notions of models for effects.

History: British school of denotational semantics. First the models (Scott, Moggi, Fiore), then the syntax (Levy).

(λ -calculus: first the syntax, then the models!)

Period: 1990-2000... and 20 more years to digest!

Goal & Motivation

Opinionated:

- Not a historical presentation, instead focus on basic concepts.
- Show how Call-by-push-value could have arisen (instructively!) from the proof theory of intuitionistic logic, using the same analysis performed for *classical* logic in the same time period by the French school of proof theory (Girard, Danos-Joinet-Schellinx).

Some course material (optional!)

▷ <https://hal.inria.fr/hal-01528857>

Not everything!

- System \mathbf{LJ}_p^n without “!” (Figures 1 & 2, pp. 4-5)
- Expressing the λ -calculus in call-by-value and call-by-name (Figure 4, p. 7)

It might help to have them handy during the course. Then to get more into the technical details:

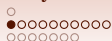
- Confluence (§3.3, p. 18)
- Strong normalisation (§5, p. 37)
- Focusing (§6.4, p. 42)

These sections stand alone and can be read by skipping the rest (which focuses a lot on categorical semantics, which I will not have the time to present).



Curry-Howard in trouble

- Gentle reminders
- From natural deduction to sequent calculus
- Blind spots of the Curry-Howard correspondence



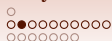
Natural deduction and λ -calculus

Proof theory studies the structure of proofs in the details. So we focus on less expressive logics. Much less: propositional logic!

We fix a set of formulae for the rest of the course:

$$A ::= X \mid A \rightarrow B \mid A \wedge B \mid A \vee B \mid \top \mid \perp$$

- Implication (\rightarrow) “implies”
- Conjunction (\wedge) “and”
- Disjunction (\vee) “or”
- Truth (\top) “true”
- Falsity (\perp) “false”



Natural deduction and λ -calculus

	introduction	elimination
hypothesis	A	
conjunction	$\frac{A \quad B}{A \wedge B}$	$\frac{A \wedge B}{A} \quad \frac{A \wedge B}{B}$
disjunction	$\frac{A}{A \vee B} \quad \frac{B}{A \vee B}$	$\frac{A \vee B \quad A \rightarrow C \quad B \rightarrow C}{C}$
implication	$\frac{[A] \cdots [A] \quad \vdots \quad B}{A \rightarrow B}$	$\frac{A \rightarrow B \quad A}{B}$
truth	$\overline{\top}$	
falsity		$\frac{\perp}{A}$

Natural deduction and λ -calculus

“*Cut-elimination*” (Gentzen):

$$\frac{\frac{[A] \cdots [A]}{\vdots} B}{A \rightarrow B} \quad \frac{\vdots}{A} \left. \vphantom{\frac{\vdots}{A}} \right\} \pi}{B} \quad \triangleright \quad \frac{\left(\frac{\vdots}{A} \right) \pi \cdots \left(\frac{\vdots}{A} \right) \pi}{\vdots} B$$

along with other rules for other pairs of introduction & elimination rules.

\Rightarrow Consistency (no proof of \perp).



Natural deduction and λ -calculus

Howard's "*formulae-as-type notion of construction*":
Cut-elimination = reduction in λ -calculus

$$\frac{\frac{x : [A] \cdots x : [A] \quad t \left\{ \begin{array}{c} \vdots \\ B \end{array} \right\}}{\lambda x. t : A \rightarrow B} \quad \left. \begin{array}{c} \vdots \\ A \end{array} \right\} u}{(\lambda x. t) u : B} \quad \triangleright \quad \left. \begin{array}{c} \vdots \\ A \end{array} \right\} u \cdots \left. \begin{array}{c} \vdots \\ A \end{array} \right\} u \\
 t[u/x] : \left\{ \begin{array}{c} \vdots \\ B \end{array} \right\}$$

I will assume familiarity with binders, substitution...



Natural deduction and λ -calculus

introduction

$$\frac{}{\Gamma, x:A \vdash x:A}$$

$$\frac{\Gamma \vdash t:A \quad \Gamma \vdash u:B}{\Gamma \vdash (t,u):A \wedge B}$$

$$\Gamma \vdash t:A_i$$

$$\frac{}{\Gamma \vdash \iota_i(t):A_1 \vee A_2}$$

$$\frac{\Gamma, x:A \vdash t:B}{\Gamma \vdash \lambda x.t:A \rightarrow B}$$

$$\frac{}{\Gamma \vdash \lambda ().\top}$$

$$\Gamma \vdash ():\top$$

elimination

$$\frac{\Gamma \vdash t:A \wedge B}{\Gamma \vdash \pi_1(t):A} \quad \frac{\Gamma \vdash t:A \wedge B}{\Gamma \vdash \pi_2(t):B}$$

$$\frac{\Gamma \vdash t:A \vee B \quad \Gamma, x:A \vdash u:C \quad \Gamma, y:B \vdash v:C}{\delta(t,x.u,y.v):C}$$

$$\delta(t,x.u,y.v):C$$

$$\frac{\Gamma \vdash t:A \rightarrow B \quad \Gamma \vdash u:A}{\Gamma \vdash tu:B}$$

$$\frac{\Gamma \vdash t:\perp}{\Gamma \vdash t:A}$$



Natural deduction and λ -calculus

$$(\lambda x.t)u \triangleright t[u/x]$$

$$\pi_1(t, u) \triangleright t$$

$$\pi_2(t, u) \triangleright u$$

$$\delta(\iota_1(t), x.u, y.v) \triangleright u[t/x]$$

$$\delta(\iota_2(t), x.u, y.v) \triangleright v[t/y]$$

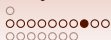


Natural deduction and λ -calculus

Good properties of “constructions”?

About logic:

- *Normalisation?*: The normal form always exists (i.e. the logic is consistent).
- *Analyticity?*: The normal form contains “explicit information” in some sense.

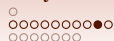


Natural deduction and λ -calculus

Good properties of “constructions”?

About reductions \rightarrow (obtained by extending \triangleright to sub-terms):

- *Confluence?*: If $u \leftarrow^* t \rightarrow^* v$ then there exists w such that $u \rightarrow^* w \leftarrow^* v$ (in particular: if w does not reduce, then it is the unique one).
- *Standardisation?*: Any series of reduction $t \rightarrow^* u$ can be rewritten by applying reductions in a leftmost-outermost order (the leftmost-outermost strategy always finds the normal form).

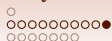


Natural deduction and λ -calculus

Problem 1

All this goes very nicely when you only have to deal with *negative* connectives (e.g. \rightarrow).

But it becomes very difficult technically when one has to deal with *positive* connectives (e.g. \vee).



Natural deduction and λ -calculus

The reason is that in order to reach **informative normal forms**, one has to consider additional reductions called **commuting conversions**:

$$\delta(t, x_1.u_1, x_2.u_2)v \quad \triangleright \quad \delta(t, x_1.(u_1 v), x_2.(u_2 v))$$

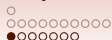
$$\pi_i(\delta(t, x_1.u_1, x_2.u_2)) \triangleright \delta(t, x_1.\pi_i(u_1), x_2.\pi_i(u_2))$$

$$\delta_v(\delta(t, x_1.u_1, x_2.u_2)) \triangleright \delta(t, x_1.\delta_v(u_1), x_2.\delta_v(u_2))$$

where $\delta_v(t) = \delta(t, y_1.v_1, y_2.v_2)$.

Cf. the anomaly in the rule:

$$\frac{\Gamma \vdash t : A \vee B \quad \Gamma, x : A \vdash u : C \quad \Gamma, y : B \vdash v : C}{\delta(t, x.u, y.v) : C}$$



Deductive systems & proof equalities

“*Notion of construction*”, a shift in point of view:

$$\frac{A}{A \vee B} \quad \frac{B}{A \vee B}$$

Deductive systems & proof equalities

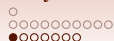
“*Notion of construction*”, a shift in point of view:

$$\frac{A}{A \vee A} \quad \frac{A}{A \vee A}$$

Deductive systems & proof equalities

“*Notion of construction*”, a shift in point of view:

$$\frac{t:A}{\iota_1(t):A \vee A} \quad \frac{t:A}{\iota_2(t):A \vee A}$$



Deductive systems & proof equalities

“*Notion of construction*”, a shift in point of view:

$$\frac{t : A}{\iota_1(t) : A \vee A} \quad \frac{t : A}{\iota_2(t) : A \vee A}$$

$$\iota_1() \neq \iota_2()$$

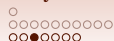
Deductive systems & proof equalities

We now care about *equalities between proofs*. We have a deductive system.

Deductive system:

- Set of formulae $|\mathcal{D}|$
- For all $A, B \in |\mathcal{D}|$, a set of proofs $\mathcal{D}(A, B)$
- For all $A, B, C \in |\mathcal{D}|$, a function

$$\circ : \mathcal{D}(B, C) \times \mathcal{D}(A, B) \rightarrow \mathcal{D}(A, C)$$



Deductive systems & proof equalities

Example: a *category*.

- $\text{id}_A \in \mathcal{D}(A, A)$
- $\text{id}_A \circ f = f$
- $f \circ \text{id}_A = f$
- $f \circ (g \circ h) = (f \circ g) \circ h$

Deductive systems & proof equalities

$$\frac{x:A \vdash t:B \quad \frac{y:B \vdash u:C \quad z:C \vdash v:D}{y:B \vdash (\lambda z.v)u:D}}{x:A \vdash (\lambda y.(\lambda z.v)u)t:D}$$

vs.

$$\frac{\frac{x:A \vdash t:B \quad y:B \vdash u:C}{x:A \vdash (\lambda y.u)t:C} \quad z:C \vdash v:D}{x:A \vdash (\lambda z.v)((\lambda y.u)t):D}$$

$$(\lambda y.(\lambda z.v)u)t = (\lambda z.v)((\lambda y.u)t)$$



Deductive systems & proof equalities

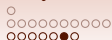
Good properties of “constructions”?

About equality in \mathcal{D} :

- *Non-degenerate?*: Not all sets $\mathcal{D}(A, B)$ have at most one element. (The booleans *true* and *false* are distinct.)
- *A category?*
- *Universal properties?*

$$\mathcal{D}(A \vee B, C) \cong \mathcal{D}(A, C) \times \mathcal{D}(B, C)$$

i.e. extensionality or “eta” rules.



Deductive systems & proof equalities

Problem 2

Having too many equations leads to nonsense for computation. E.g. asking for both a category and all universal properties:

$$u[t^{A \vee B}/y] = \delta(t, x_1^A . u[l_1(x_1)/y], x_2^B . u[l_2(x_2)/y])$$

Impossibility results in certain cases (becomes degenerate).



Deductive systems & proof equalities

In “*A formulae-as-types notion of construction*”, Bill Howard proposed a **connection between reduction and cut-elimination**. However:

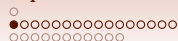
- Howard **did not mention commuting conversions**, only stated without proof the normalisation result for sums *on closed terms*. Gentzen’s cut-elimination is in fact not a theorem of natural deduction, but of **sequent calculus**, which Gentzen invented because natural deduction was too hard to work with directly.
- Gentzen’s cut-elimination in its original formulation is **not designed to define a meaningful notion of equality between proofs**.

The **Curry-Howard correspondence was incomplete**.



Sequent calculus for Curry-Howard

- Gentzen's sequent calculus
- The backbone of Curry-Howard for sequent calculus: the μ - $\tilde{\mu}$ system
- Dealing with connectives : polarisation and focusing
- Proof-theoretic results



Sequent calculus

A *sequent* is a list of formulae, of the following form:

$$A_1, \dots, A_n \vdash B_1, \dots, B_m$$

shortened

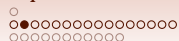
$$\Gamma \vdash \Delta$$

$\Gamma = A_1, \dots, A_n$ is the *antecedent*, and $\Delta = B_1, \dots, B_m$ is the *succedent*.

The meaning of the sequent is as follows: *if all the formulae of the antecedent are true, then at least one formula of the succedent is true*. In other words it is equivalent to:

$$(A_1 \wedge \dots \wedge A_n) \rightarrow (B_1 \vee \dots \vee B_m)$$

Sequent calculus is a formulation of logic where all the rules for connectives are introduction rules (on the left or on the right).



Sequent calculus

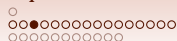
I will consider logics that only use one of the following two forms:

- *Intuitionistic* sequent

$$A_1, \dots, A_n \vdash B$$

- *Classical* sequent

$$\vdash A_1, \dots, A_n$$



Sequent calculus

The rules for the connectives of intuitionistic sequent calculus (**LJ**) are given as follows (**Logical Group**):

right introduction rules

$$\frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \wedge B} (\wedge \vdash)$$

$$\frac{\Gamma \vdash A}{\Gamma \vdash A \vee B} (\vee \vdash_1) \quad \frac{\Gamma \vdash B}{\Gamma \vdash A \vee B} (\vee \vdash_2)$$

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \rightarrow B} (\rightarrow \vdash)$$

$$\frac{}{\Gamma \vdash \top} (\top \vdash)$$

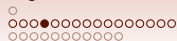
left introduction rules

$$\frac{\Gamma, A \vdash \Delta}{\Gamma, A \wedge B \vdash \Delta} (\wedge_1 \vdash) \quad \frac{\Gamma, B \vdash \Delta}{\Gamma, A \wedge B \vdash \Delta} (\wedge_2 \vdash)$$

$$\frac{\Gamma, A \vdash \Delta \quad \Gamma, B \vdash \Delta}{\Gamma, A \vee B \vdash \Delta} (\vee \vdash)$$

$$\frac{\Gamma \vdash A \quad \Gamma', B \vdash \Delta}{\Gamma, \Gamma', A \rightarrow B \vdash \Delta} (\rightarrow \vdash)$$

$$\frac{}{\Gamma, \perp \vdash \Delta} (\perp \vdash)$$



Sequent calculus

The backbone of sequent calculus is formed by the **Identity Group**:

- The *axiom* rule:

$$\frac{}{A \vdash A} \text{ (ax)}$$

- The *cut* rule:

$$\frac{\Gamma \vdash A, \Delta \quad \Gamma', A \vdash \Delta'}{\Gamma, \Gamma' \vdash \Delta, \Delta'} \text{ (cut)}$$

(where Δ is empty for intuitionistic logic)

All the power of logical consequence in sequent calculus is located in the cut rule.

Sequent calculus

The **Structural Group** deals with the bookkeeping of multiplicities of formulae.

- The *weakening* rules:

$$\frac{\Gamma \vdash \Delta}{\Gamma, A \vdash \Delta} \text{ (w}\vdash\text{)} \quad \left(\frac{\Gamma \vdash \Delta}{\Gamma \vdash A, \Delta} \text{ (}\vdash\text{ w)} \right)$$

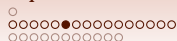
- The *contraction* rules:

$$\frac{\Gamma, A, A \vdash \Delta}{\Gamma, A \vdash \Delta} \text{ (c}\vdash\text{)} \quad \left(\frac{\Gamma \vdash A, A, \Delta}{\Gamma \vdash A, \Delta} \text{ (}\vdash\text{ c)} \right)$$

- The *exchange* rules:

$$\frac{\Gamma, A, B, \Gamma' \vdash \Delta}{\Gamma, B, A, \Gamma' \vdash \Delta} \text{ (ex}\vdash\text{)} \quad \left(\frac{\Gamma \vdash \Delta, A, B, \Delta'}{\Gamma \vdash \Delta, B, A, \Delta'} \text{ (}\vdash\text{ ex)} \right)$$

(*Substructural logics*, like linear logic, try to remove these.)



Sequent calculus

Summary

- **Identity Group:** backbone of logic
- **Structural Group:** bookkeeping of formulae
- **Logical Group:** introduction rules for connectives



Sequent calculus

With the cut rule, all the elimination rules are derivable starting from the left-introduction rules.

- Elimination rule for \rightarrow :

$$\frac{\Gamma \vdash A \rightarrow B \quad \frac{\Gamma \vdash A \quad \overline{B \vdash B}}{\Gamma, A \rightarrow B \vdash B} \text{ (}\rightarrow\text{)} \text{ (cut)}}{\Gamma, \Gamma \vdash B} \text{ (c}\vdash\text{)} \quad \frac{\Gamma, \Gamma \vdash B}{\Gamma \vdash B} \text{ (c}\vdash\text{)}$$

- Elimination rule for \vee :

$$\frac{\Gamma \vdash A \vee B \quad \frac{\Gamma, A \vdash C \quad \Gamma, B \vdash C}{\Gamma, A \vee B \vdash C} \text{ (}\vee\text{)} \text{ (cut)}}{\Gamma, \Gamma \vdash C} \text{ (c}\vdash\text{)} \quad \frac{\Gamma, \Gamma \vdash C}{\Gamma \vdash C} \text{ (c}\vdash\text{)}$$



Sequent calculus

Gentzen's cut-elimination. *The cut rule is admissible in the system without the cut rule.* That is to say, for any derivation that uses the cut rule, one can find a derivation that does not use the cut rule.



The μ - $\tilde{\mu}$ subsystem

The backbone of Curry-Howard for sequent calculus (= computational interpretation of the Identity Group).

Three categories of terms t, e, c associated with three kinds of judgements:

$$\Gamma \vdash t : A \mid \Delta$$

$$\Gamma \mid e : A \vdash \Delta$$

$$c : (\Gamma \vdash \Delta)$$

The formula A is *principal*.

The *antecedent* $\Gamma = (x_1 : A_1, \dots, x_n : A_n)$ and the *succedent* $\Delta = (\alpha_1 : A_1, \dots, \alpha_n : A_n)$ give the types of variables that might appear in t, e and c .



The μ - $\tilde{\mu}$ subsystem

Identity group:

$$\frac{}{x : A \vdash x : A} \text{ (}\vdash\text{-ax)} \qquad \frac{}{\alpha : A \vdash \alpha : A} \text{ (ax}\vdash\text{)}$$

$$\frac{c : (\Gamma \vdash \alpha : A, \Delta)}{\Gamma \vdash \mu\alpha.c : A \mid \Delta} \text{ (}\vdash\text{-}\mu\text{)} \qquad \frac{c : (\Gamma, x : A \vdash \Delta)}{\Gamma \mid \tilde{\mu}x.c : A \vdash \Delta} \text{ (}\tilde{\mu}\vdash\text{)}$$

$$\frac{\Gamma \vdash t : A \mid \Delta \quad \Gamma' \mid e : A \vdash \Delta'}{\langle t \parallel e \rangle : (\Gamma, \Gamma' \vdash \Delta, \Delta')} \text{ (cut)}$$

$$t ::= x \mid \mu\alpha.c$$

$$e ::= \alpha \mid \tilde{\mu}x.c$$

$$c ::= \langle t \parallel e \rangle$$

New binders μ , $\tilde{\mu}$; infinity of variables x ($y, z \dots$) and α ($\beta, \gamma \dots$).

The μ - $\tilde{\mu}$ subsystem

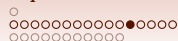
Activation:

$$\frac{c : (\Gamma, x : A \vdash \Delta)}{\Gamma \mid \tilde{\mu}x.c : A \vdash \Delta} \quad (\vdash \mu) \qquad \frac{c : (\Gamma \vdash \alpha : A, \Delta)}{\Gamma \vdash \mu\alpha.c : A \mid \Delta} \quad (\tilde{\mu} \vdash)$$

Deactivation:

$$\frac{\Gamma \vdash t : A \mid \Delta \quad \overline{\mid \alpha : A \vdash \alpha : A} \quad (\text{ax}\vdash)}{\langle t \parallel \alpha \rangle : (\Gamma \vdash \alpha : A, \Delta)} \quad (\text{cut})$$

$$\frac{\overline{x : A \vdash x : A} \quad (\vdash\text{ax}) \quad \Gamma \mid e : A \vdash \Delta}{\langle x \parallel e \rangle : (\Gamma, x : A \vdash \Delta)} \quad (\text{cut})$$



The μ - $\tilde{\mu}$ subsystem

Activating then deactivating amounts to doing nothing:

$$\langle \mu\alpha.c \parallel \beta \rangle \triangleright c[\beta/\alpha]$$

$$\langle y \parallel \tilde{\mu}x.c \rangle \triangleright c[y/x]$$

Deactivating then activating amounts to doing nothing:

$$\mu\alpha.\langle t \parallel \alpha \rangle \triangleright t \quad (\alpha \notin t)$$

$$\tilde{\mu}x.\langle x \parallel e \rangle \triangleright e \quad (x \notin e)$$

μ - $\tilde{\mu}$ is a system to let you freely choose and switch between principal formulae. Its reduction rules do the bookkeeping.

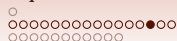
The μ - $\tilde{\mu}$ subsystem

Do we have a category?

$$\frac{\Gamma \vdash t : A \mid \frac{c : (x : A \vdash \alpha : B) \quad | e : B \vdash \Delta}{\langle \mu \alpha . c \parallel e \rangle : (x : A \vdash \Delta)}}{\langle t \parallel \tilde{\mu} x . \langle \mu \alpha . c \parallel e \rangle \rangle : (\Gamma \vdash \Delta)}$$

vs.

$$\frac{\Gamma \vdash t : A \mid \frac{c : (x : A \vdash \alpha : B)}{\langle t \parallel \tilde{\mu} x . c \rangle : (\Gamma \vdash \alpha : B)}}{\langle \mu \alpha . \langle t \parallel \tilde{\mu} x . c \rangle \parallel e \rangle : (\Gamma \vdash \Delta)} \quad | e : B \vdash \Delta$$



The μ - $\tilde{\mu}$ subsystem

$$\langle t \parallel \tilde{\mu}x. \langle \mu\alpha.c \parallel e \rangle \rangle =? \langle \mu\alpha. \langle t \parallel \tilde{\mu}x.c \rangle \parallel e \rangle$$

Yes whenever either:

$$\langle t \parallel \tilde{\mu}x.c \rangle = c[t/x]$$

for t, c arbitrary, or:

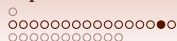
$$\langle \mu\alpha.c \parallel e \rangle = c[e/\alpha]$$

for c, e arbitrary.

1. Choose either (but the choice is arbitrary),
2. Choose both, but then we have the weird equality:

$$c[\tilde{\mu}x.c'/\alpha] = \langle \mu\alpha.c \parallel \tilde{\mu}x.c' \rangle = c'[\mu\alpha.c/x],$$

3. Do not choose (make no assumption about associativity for the moment).



The μ - $\tilde{\mu}$ subsystem

Do not choose: distinguish between terms that one can substitute with from ones that one cannot.

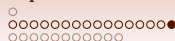
Introduce new categories V, S along with the rules:

$$\langle V \parallel \tilde{\mu}x.c \rangle \triangleright c[V/x] \qquad \langle \mu\alpha.c \parallel S \rangle \triangleright c[S/\alpha]$$

We have so far:

$$\begin{aligned} V &::= x \\ t &::= V \mid \mu\alpha.c \\ S &::= \alpha \\ e &::= S \mid \tilde{\mu}x.c \\ c &::= \langle t \parallel e \rangle \end{aligned}$$

“Values” & “Stacks”: first step towards Call-by-push-value, simply by refusing to make an assumption!



The μ - $\tilde{\mu}$ subsystem

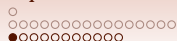
What about the Structural Group? e.g.

$$\frac{\Gamma, x : A, y : A \vdash t : B \mid}{\Gamma, x : A \vdash t[y/x] : B \mid} \text{ (c}\vdash\text{)} \qquad \frac{\Gamma \vdash t : B \mid}{\Gamma, x : A \vdash t : B \mid} \text{ (w}\vdash\text{)}$$

Merge *renaming*, *collapsing* and *reordering* of variables into a single rule indexed by a **structure map** $\sigma : \Gamma, \Delta \rightarrow \Gamma', \Delta'$ substituting variables for variables:

$$\frac{\Gamma \vdash t : A \mid \Delta}{\Gamma' \vdash t[\sigma] : A \mid \Delta'} \qquad \frac{\Gamma \mid e : A \vdash \Delta}{\Gamma' \mid e[\sigma] : A \vdash \Delta'} \qquad \frac{c : (\Gamma \vdash \Delta)}{c[\sigma] : (\Gamma' \vdash \Delta')}$$

Simplifies a lot of things technically & easier to extend to linear logics.



Focalisation & polarisation

It remains for us to deal with the *Logical Group* (the connectives!).

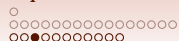
Main insight (from Danos, Joinet and Shellinx): **start with the η rules, and define the remaining reduction in order to be compatible with them.**

Focalisation & polarisation

η rules in sequent calculus:

$$\frac{}{A \rightarrow B \vdash A \rightarrow B} \text{(ax)} = \frac{\frac{}{A \vdash A} \text{(ax)} \quad \frac{}{B \vdash B} \text{(ax)}}{A \rightarrow B, A \vdash B} (\rightarrow \vdash)}{A \rightarrow B \vdash A \rightarrow B} (\vdash \rightarrow)$$

$$\frac{}{A \vee B \vdash A \vee B} \text{(ax)} = \frac{\frac{}{A \vdash A} \text{(ax)}}{A \vdash A \vee B} (\vdash \vee_1) \quad \frac{\frac{}{B \vdash B} \text{(ax)}}{B \vdash A \vee B} (\vdash \vee_2)}{A \vee B \vdash A \vee B} (\vee \vdash)$$



Focalisation & polarisation

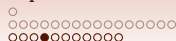
Idea: pattern-matching!

$$\frac{\Gamma \vdash t : A_i \mid}{\Gamma \vdash \iota_i(t) : A_1 \vee A_2 \mid} \quad \frac{c_1 : (\Gamma, x : A \vdash \Delta) \quad c_2 : (\Gamma, y : B \vdash \Delta)}{\Gamma \mid \tilde{\mu}[x.c_1 \mid y.c_2] : A \vee B \vdash \Delta}$$

$$\frac{}{\mid \alpha : A \vee B \vdash A \vee B}$$

=

$$\frac{\frac{\frac{x : A \vdash x : A \mid}{x : A \vdash \iota_1(x) : A \vee B \mid}}{\langle \iota_1(x) \parallel \alpha \rangle : (x : A \vdash \alpha : A \vee B)} \quad \frac{\frac{y : B \vdash y : B \mid}{y : B \vdash \iota_2(y) : A \vee B \mid}}{\langle \iota_2(y) \parallel \alpha \rangle : (y : B \vdash \alpha : A \vee B)}}{\mid \tilde{\mu}[x.\langle \iota_1(x) \parallel \alpha \rangle \mid y.\langle \iota_2(y) \parallel \alpha \rangle] : A \vee B \vdash \alpha : A \vee B}$$



Focalisation & polarisation

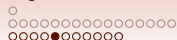
Idea: pattern-matching!

$$\frac{\Gamma \vdash t : A \quad \Gamma' \mid e : B \vdash \Delta}{\Gamma, \Gamma' \mid t \cdot e : A \rightarrow B \vdash \Delta} \qquad \frac{c : (\Gamma, x : A \vdash \alpha : B)}{\Gamma \vdash \mu(x.\alpha).c : A \rightarrow B \mid}$$

$$\frac{}{x : A \rightarrow B \vdash x : A \rightarrow B \mid}$$

=

$$\frac{\frac{\frac{y : A \vdash y : A \quad \mid \alpha : B \vdash \alpha : B}{y : A \mid y \cdot \alpha : A \rightarrow B \vdash \alpha : B}}{\langle x \parallel y \cdot \alpha \rangle : (x : A \rightarrow B, y : A \vdash \alpha : B)}}{x : A \rightarrow B \vdash \mu(y \cdot \alpha). \langle x \parallel y \cdot \alpha \rangle : A \rightarrow B \mid}$$



Focalisation & polarisation

$$V ::= x$$

$$t ::= V \mid \mu\alpha.c \mid \mu(x.\alpha).c \mid \iota_i(t)$$

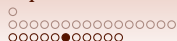
$$S ::= \alpha$$

$$e ::= S \mid \tilde{\mu}x.c \mid \tilde{\mu}[x.c \mid y.c'] \mid t.e$$

$$c ::= \langle t \parallel e \rangle$$

$$S =_{\eta} \tilde{\mu}[x.\langle \iota_1(x) \parallel S \rangle \mid y.\langle \iota_2(y) \parallel S \rangle] \quad (x, y \notin S)$$

$$V =_{\eta} \mu(y.\alpha).\langle V \parallel y.\alpha \rangle \quad (y, \alpha \notin V)$$



Focalisation & polarisation

Which reduction rules?

$$\langle \iota_i(t) \parallel \tilde{\mu}[x_1.c_1 \mid x_2.c_2] \rangle \triangleright^? c_i[t/x_i]$$

$$\langle \mu(x.\alpha).c \parallel t.e \rangle \triangleright^? c[t/x, e/\alpha]$$

Problematic as before! Implies substitution with arbitrary t or e .

Better:

$$\langle \iota_i(V) \parallel \tilde{\mu}[x_1.c_1 \mid x_2.c_2] \rangle \triangleright c_i[V/x_i]$$

$$\langle \mu(x.\alpha).c \parallel V.S \rangle \triangleright c[V/x, S/\alpha]$$

Focalisation & polarisation

Now what about:

$$\langle \iota_i(t) \parallel \tilde{\mu}[x_1.c_1 \mid x_2.c_2] \rangle \triangleright^? \langle t \parallel \tilde{\mu}x_i.c_i \rangle$$

$$\langle \mu(x.\alpha).c \parallel t.e \rangle \triangleright^? \langle t \parallel \tilde{\mu}x.\langle \mu\alpha.c \parallel e \rangle \rangle$$

This is actually *definable* from the rules we have just set up assuming $\tilde{\mu}[x_1.c_1 \mid x_2.c_2] \in S$ and $\mu(x.\alpha).c \in V$:

$$\iota_i(t) \stackrel{\text{def}}{=} \mu\alpha.\langle t \parallel \tilde{\mu}x.\langle \iota_i(x) \parallel \alpha \rangle \rangle$$

$$t.e \stackrel{\text{def}}{=} \tilde{\mu}y.\langle t \parallel \tilde{\mu}x.\langle \mu\alpha.\langle y \parallel x.\alpha \rangle \parallel e \rangle \rangle$$

Focalisation is the phenomenon by which introduction rules hide cuts.

Focalisation & polarisation

$$V ::= x \mid \iota_i(V) \mid \mu(x.\alpha).c$$

$$t ::= V \mid \mu\alpha.c$$

$$S ::= \alpha \mid V.S \mid \tilde{\mu}[x.c \mid y.c']$$

$$e ::= S \mid \tilde{\mu}x.c$$

$$c ::= \langle t \parallel e \rangle$$

$$\langle \iota_i(V) \parallel \tilde{\mu}[x_1.c_1 \mid x_2.c_2] \rangle \triangleright c_i[V/x_i]$$

$$\langle \mu(x.\alpha).c \parallel V.S \rangle \triangleright c[V/x, S/\alpha]$$



Focalisation & polarisation

Final question: how to reduce a binder against a binder?

$$\langle \mu\alpha.c \parallel \tilde{\mu}x.c' \rangle \triangleright ?$$

Hint: be compatible with η rules.

If the common type of x and α is $A \rightarrow B$ then:

$$\langle \mu\alpha.c \parallel \tilde{\mu}x.c' \rangle = \langle \mu(y.\beta). \langle \mu\alpha.c \parallel y.\beta \rangle \parallel \tilde{\mu}x.c' \rangle$$

We are forced to reduce as follows:

$$\langle \mu\alpha.c \parallel \tilde{\mu}x.c' \rangle \triangleright c'[\mu\alpha.c/x]$$

If the common type is $A \vee B$ then:

$$\langle \mu\alpha.c \parallel \tilde{\mu}x.c' \rangle = \langle \mu\alpha.c \parallel \tilde{\mu}[y_1.\langle \iota_1(y_1) \parallel \tilde{\mu}x.c' \rangle \mid y_2.\langle \iota_2(y_2) \parallel \tilde{\mu}x.c' \rangle] \rangle$$

We are forced to reduce as follows:

$$\langle \mu\alpha.c \parallel \tilde{\mu}x.c' \rangle \triangleright c[\tilde{\mu}x.c'/\alpha]$$



Focalisation & polarisation

Polarisation: let the order of reduction be determined by the polarity of the formula.*

Distinguish positive from negative cuts and binders:

$$V ::= x \mid \iota_i(V) \mid \mu(x \cdot \alpha).c \mid \mu^\ominus \alpha.c$$

$$t ::= V \mid \mu^+ \alpha.c$$

$$S ::= \alpha \mid V \cdot S \mid \tilde{\mu}[x.c \mid y.c'] \mid \tilde{\mu}^+ x.c$$

$$e ::= S \mid \tilde{\mu}^\ominus x.c$$

$$c ::= \langle t \parallel e \rangle^+ \mid \langle t \parallel e \rangle^\ominus$$

(*: beware, the polarity is not always determined by η expansions!)

Focalisation & polarisation

What about pairs?

In fact pairs can be treated either as positives or as negatives. We consider them separate connectives (\otimes pos. and $\&$ neg.)

$$\begin{array}{c}
 \frac{\Gamma \vdash t : A \quad \Gamma' \vdash u : B}{\Gamma, \Gamma' \vdash t \otimes u : A \otimes B} \quad (\vdash \otimes) \qquad \frac{c : (\Gamma, x : A, y : B \vdash \Delta)}{\Gamma \mid \tilde{\mu}(x \otimes y).c : A \otimes B \vdash \Delta} \quad (\otimes \vdash) \\
 \\
 \frac{c : (\Gamma \vdash \alpha : A) \quad c' : (\Gamma \vdash \beta : B)}{\Gamma \vdash \mu \langle \alpha.c ; \beta.c' \rangle : A \& B} \quad (\vdash \&) \qquad \frac{\Gamma \mid e : A_i \vdash \Delta}{\Gamma \mid \pi_i.e : A_1 \& A_2 \vdash \Delta} \quad (\&_i \vdash)
 \end{array}$$

Then \wedge (= comma on the left of \vdash) is defined by cases

$A \wedge B$	A_+	A_-
B_+	$A \otimes B$	$A \otimes B$
B_-	$A \otimes B$	$A \& B$



Stop worrying and love evaluation order

- Mission accomplished
- Computational interpretation as abstract machines
- Impossibility results: learn to live with evaluation order!
- Direct vs. indirect models



Results

Rewriting

The rewriting system is very simple (it is an *orthogonal higher-order rewriting system*). We get for free:

- **Confluence** (cf. course material, §3.3, p. 18)
- **Standardisation**

by using theorems from the literature (or application by hand of the traditional proofs for the λ -calculus without sums).

Results

Logic: normalisation

The type system fits very well proofs by logical relations/predicates based on *orthogonality*, so for instance we get a proof of:

- **Strong normalisation**: all reduction paths are finite (cf. course material, §5, p. 37)

again using a proof that is a generalisation of that for System F. (I forgot to mention: our sequent calculus extends to second order *for free*, with quantifiers \forall negative and \exists positive.)

We can state cut-elimination: **for any derivation, there is an *equivalent* derivation whose only uses of the cut rule are deactivations.**



Results

Logic: focusing

The standard **focusing** proof-search algorithm is obtained by looking at the shape of η -expanded normal terms.

(Completeness proof included!)

See course material (§6.4, p. 42) for details and perspectives for this term-based technique.



Results

Logic: constructive classical logic

This method was originally applied to Gentzen's classical sequent calculus **LK** in the seminal paper by Danos, Joinet and Shellinx (“*A new deconstructive logic: Linear Logic*”, *Journal of Symbolic Logic*, 1997).

They reconstructed a constructive interpretation of classical logic invented by Girard closely related to *call/cc*.

They did not have a term interpretation, only pure sequents, so the paper is hard to read and the technical details very tedious—Curry-Howard for sequent calculus saves us here! Applying focusing, we get the conservativity of classical logic over intuitionistic logic for **purely positive formulae**. (An example of **analyticity** of cut-free proofs.)



Computational relevance

Gentzen-Landin correspondence

- **Gerhard Gentzen** (1909-1945): natural deduction and sequent calculus, cut-elimination theorem.
- **Peter J. Landin** (1930-2009): SECD machine, control operators to model jumps. (Among others!)



Computational relevance

Gentzen-Landin correspondence

Remember the derivation of natural deduction rules from sequent calculus rules.

$$\lambda x.t \stackrel{\text{def}}{=} \mu(x.\alpha). \langle t \parallel \alpha \rangle$$

$$t u \stackrel{\text{def}}{=} \mu\alpha. \langle t \parallel u.\alpha \rangle$$

$$\langle t u \parallel S \rangle \triangleright \langle t \parallel u.S \rangle$$

$$\langle \lambda x.t \parallel V.S \rangle \triangleright \langle t[V/x] \parallel S \rangle$$

Push-enter abstract machines.

Exercise! Convince yourself that the positive and negative interpretations of the λ -calculus with sums (cf. course material, Figure 4, p. 7) compute respectively in call-by-value and call-by-name.



Computational relevance

Categorical semantics?

In the end, can we get a non-degenerate categorical interpretation? Here are two examples of impossibility results if we assume associativity of composition:

- **Classical logic.** Consider the equality:

$$c \triangleleft \langle \mu_{-}.c \parallel \tilde{\mu}_{-}.c' \rangle \triangleright c'$$

for c, c' arbitrary!

- **Recursion.** Consider the function $\text{not} : \top \vee \top \rightarrow \top \vee \top$ that sends *true* on *false* and *false* on *true*. Its fixed point is a boolean equal to its own negation.

They have non-degenerate models of Call-by-push-value (respectively CPS and domains).



Computational relevance

Direct vs. indirect semantics

There are two ways around.

- Either model the deductive system **directly**: axiomatize polarisation as a category but where some associativities fail.
- Or ask for two categories plus some structure to mediate between the two. Namely, an adjunction between a category of “values” and a category of “stacks” (so-called *adjunction model*). One has to specify a non-trivial interpretation of the deductive system into this notion of model, so the model is **indirect**.



Computational relevance

Direct vs. indirect semantics

There is a correspondence between the two approaches! It is based on identifying a semantic notion of values and stacks:

- **Thunkables** = algebraic values: $\forall c, \langle t \parallel \tilde{\mu}x.c \rangle = c[t/x]$
- **Linears** = algebraic stacks: $\forall c, \langle \mu\alpha.c \parallel e \rangle = c[e/\alpha]$

References

For bibliographic references and more historical context,
refer to the course material:

▷ <https://hal.inria.fr/hal-01528857>