Foundation of Computing

Ugo Montanari
Dipartimento di Informatica
Università di Pisa

48 ore - 6 CPU
Second Semester 2021-2022
Exam: a seminar on a related subject
Prerequisites: some elementary knowledge
of logic and algebra

Motivations of the Course

- Need of
  - unifying different approaches
    - where are Maxwell equations?
  - sharing formal developments among different fields
- Category theory is a good candidate
  - abstraction
  - representation independence
  - compositionality
  - canonical constructions
- Often constructions have a precise semantic meaning

Structure of the Course

functions as data	concurrency	interactivity
λ-calculus Alonso Church ‘30 untyped as logic for computable functions Domain equation solved by Dana Scott in 1969-71 Inspired LISP and all functional languages Type theory for polymorphism and abstract data types	**Petri nets** Carl Adam Petri 1962 as a basic model for concurrency Collective/individual semantics Nonsequential processes Graphtransformation, term rewriting Probably most used specification formalism for practical systems	**Calculus of Communicating Systems (CCS), π-calculus** Robin Milner (Turing award ‘91) CCS around 1980, π-calc.1989 Bisimilarity, David Park: most used equivalence of interactive nondeterministic systems. Formal basis of most models for state-based compositional secure distributed systems
cartesian closed categories Pierre-Louis Curien categorical abstract machine 1997	monoidal categories Jose Meseguer-Ugo Montanari Petri nets are monoids 1989	coalgebras-bialgebras Peter Aczel 1988-89 Daniele Turi, Gordon Plotkin 1997

Untyped Lambda Calculus

$(\lambda x.xx)\ (\lambda x.xx) \Rightarrow (\lambda x.xx)\ (\lambda x.xx) \Rightarrow \ldots$

existence of nonterminating computations

Università degli Studi di Pisa
Corso di Laurea Magistrale in Informatica

Anno Accademico 2021-2022

Insegnamento di

Foundation of Computing

Pagina del corso: `http:/pages.di.unipi.it/montanari/FOC.html`

Note di

Lambda Calcolo

Libro di Testo: J. Mitchell, Foundations for Programming Languages, MIT Press, 1996.

,

a cura di
Ugo Montanari
Dipartimento di Informatica
Università degli Studi di Pisa
ugo@di.unipi.it

Symply Typed Lambda Calculus

type constructors available

ES. $\lambda^{\to, \times, unit}$ calculus

Types

$$\sigma ::= b \mid null \mid unit \mid \sigma + \sigma \mid \sigma \times \sigma \mid \sigma \to \sigma$$

null : the empty set
unit : the singleton
b : the type constants (integers, boolean, etc.)
$\sigma + \sigma$: disjoint union
$\sigma \times \sigma$: cartesian product
} associate to the left
$\sigma \to \sigma$: "some" functions – associates to the right

Context Sensitive Syntax

variables are untyped

$\Gamma = \{ x_1 : \sigma_1, \dots, x_K : \sigma_K \}$ type assignment $\quad i \neq j \Rightarrow x_i \neq x_j$

$\Gamma, x : \sigma = \Gamma \cup \{ x : \sigma \}$ implies $x \notin \Gamma$

$\Gamma \triangleright M : \sigma$ typing assertion or typing judgement

Signature

type with constants in B

$\Sigma = \langle B, C \rangle$

set of pairs $c : \sigma$

type constants

term constants

Example

Signature for PCF

type constants: nat, bool

term constants: $0, 1, 2, \ldots$: nat

true, false : bool

plus : nat $\to$ nat $\to$ nat

Eq? : nat $\to$ nat $\to$ bool

$\left.\begin{array}{l} \mathrm{cond}_\sigma : \mathrm{bool} \to \sigma \to \sigma \to \sigma \\ \mathrm{fix}_\sigma : (\sigma \to \sigma) \to \sigma \end{array}\right\}$ for every type σ

Typing rules for $\lambda^{\to}$

Terms

$$M, N ::= c \mid x:\sigma \mid \lambda x:\sigma.\, M \mid M N$$

$$\frac{c : \sigma \in C}{\emptyset \rhd c : \sigma} \quad (\mathrm{cst})$$

$$x : \sigma \rhd x : \sigma \quad (\mathrm{var})$$

$$\frac{\Gamma \rhd M : \sigma}{\Gamma, x:\tau \rhd M : \sigma} \quad (\mathrm{addvar})$$

$$\frac{\Gamma, x:\sigma \rhd M : \tau}{\Gamma \rhd \lambda x:\sigma.\, M : \sigma \to \tau} \quad (\to \mathrm{Intro})$$

$$\frac{\Gamma \rhd M : \sigma \to \tau, \quad \Gamma \rhd N : \sigma}{\Gamma \rhd M N : \tau} \quad (\to \mathrm{Elim})$$

The typing rules define the (correct) typing assertions.

Free variables

$FV(c) = \phi$

$FV(x) = \{x\}$

$FV(\lambda x.M) = FV(M) - \{x\}$

$FV(MN) = FV(M) \cup FV(N)$

Substitutions (in front)

$[N/x]\, x = N$

$[N/x]\, y = y \quad y \neq x$

$[N/x]\, c = c$

$[N/x](PQ) = [N/x]P \;\; [N/x]Q$ subst. precedes application

$[N/x]\lambda y{:}\sigma.M = \lambda z{:}\sigma.\, [N/x][z/y]\, M \quad z \notin FV(\lambda y.M, N)$ and $z \neq x$

Notice that z in the last equation is not unique:
thus $[-/-]-$ defines a relation.
If we let also

$\lambda x{:}\sigma.M = \lambda y{:}\sigma.\, [y/x]\, M \qquad$ (α conversion)

then $[-/-]-$ becomes a function on the corresponding equivalence classes of terms.

Theorem (substitution preserves types)

$\Gamma_1, x:\sigma \triangleright M:\tau \quad \Gamma_2 \triangleright N:\sigma$ implies $\Gamma_1, \Gamma_2 \triangleright [N/x]\, M:\tau$

Typing rules for ×, +, unit, null

$$\frac{\Gamma \rhd M:\sigma \quad \Gamma \rhd N:\tau}{\Gamma \rhd \langle M,N\rangle : \sigma\times\tau} \quad (\times\ \text{Intro})$$

$M, N ::= \langle M, N\rangle \mid \mathrm{Proj}_1^{\sigma,\tau} M \mid \mathrm{Proj}_2^{\sigma,\tau} M$

$$\frac{\Gamma \rhd M:\sigma\times\tau}{\Gamma \rhd \mathrm{Proj}_1^{\sigma,\tau} M:\sigma} \quad (\times\ \text{Elim})_1$$

$$\frac{\Gamma \rhd M:\sigma\times\tau}{\Gamma \rhd \mathrm{Proj}_2^{\sigma,\tau} M:\tau} \quad (\times\ \text{Elim})_2$$

$$\frac{\Gamma \rhd M:\sigma}{\Gamma \rhd \mathrm{Inleft}^{\sigma,\tau} M:\sigma+\tau} \quad (+\ \text{Intro})_1$$

$M, N, P ::= \mathrm{Inright}^{\sigma,\tau} M \mid \mathrm{Inleft}^{\sigma,\tau} M \mid \mathrm{Case}^{\sigma,\tau,\rho} MNP$

$$\frac{\Gamma \rhd M:\tau}{\Gamma \rhd \mathrm{Inright}^{\sigma,\tau} M:\sigma+\tau} \quad (+\ \text{Intro})_2$$

$$\frac{\Gamma \rhd M:\sigma+\tau \quad \Gamma \rhd N:\sigma\to\rho \quad \Gamma \rhd P:\tau\to\rho}{\Gamma \rhd \mathrm{Case}^{\sigma,\tau,\rho} MNP:\rho} \quad (+\text{Elim})$$

Rule (+ Elim) was introduced by Prawitz in 1965.
It is the only correct way of typing variant records

Wrong in Pascal and "classic ML". Correct in CLU (Liskov, 1981).

$*:\mathrm{unit}$ (unit Intro)

$\mathrm{zero}^{\sigma}:\mathrm{null}\to\sigma$ (null Elim)

Curry-Howard isomorphism

formula ~ type
proof ~ lambda term
proof normalization ~ reduction

Natural deduction system

Propositional intuitionistic version

$\sigma ::= b \mid \sigma \to \sigma \mid \sigma \wedge \sigma \mid \sigma \vee \sigma$

b: propositional symbol

$\Gamma \vdash_M \sigma$ ← the proved formula

hypothesis: Γ; proof structure: M

$x : \sigma \vdash_x \sigma$ different instances, different labels

$$\frac{\Gamma \vdash_M \sigma}{\Gamma, x : \tau \vdash_M \sigma}$$ additional hypotheses can always be assumed

$$\frac{\Gamma \vdash_M \sigma \to \tau \qquad \Gamma \vdash_N \sigma}{\Gamma \vdash_{MN} \tau}$$ modus ponens

$$\frac{\Gamma, x : \sigma \vdash_M \tau}{\Gamma \vdash_{\lambda x:\sigma.M} \sigma \to \tau}$$ discharging the hypothesis σ

$$\frac{c : \sigma \in C}{\phi \vdash_c \sigma}$$ certain formulas are given

A formula σ is <u>provable</u> iff $\exists M.\ \phi \vdash_M \sigma$

A type σ is <u>inhabited</u> iff $\exists M.\ \phi \rhd M : \sigma$

Natural deduction (cont.)

$$\frac{\Gamma \vdash_M \sigma \qquad \Gamma \vdash_N \tau}{\Gamma \vdash_{\langle M,N \rangle} \sigma \wedge \tau}$$

$$\frac{\Gamma \vdash_M \sigma \wedge \tau}{\Gamma \vdash_{\mathrm{Proj}_1^{\sigma,\tau} M} \sigma}$$

$$\frac{\Gamma \vdash_M \sigma \wedge \tau}{\Gamma \vdash_{\mathrm{Proj}_2^{\sigma,\tau} M} \tau}$$

$$\frac{\Gamma \vdash_M \sigma}{\Gamma \vdash_{\mathrm{Inleft}^{\sigma,\tau} M} \sigma \vee \tau}$$

$$\frac{\Gamma \vdash_M \tau}{\Gamma \vdash_{\mathrm{Inright}^{\sigma,\tau} M} \sigma \vee \tau}$$

$$\frac{\Gamma \vdash_M \sigma \vee \tau \quad \Gamma \vdash_N \sigma \to \rho \quad \Gamma \vdash_P \tau \to \rho}{\Gamma \vdash_{\mathrm{Case}^{\sigma,\tau,\rho}_{MNP}} \rho}$$ Prawitz 1965

$$\vdash_* T$$

$$\vdash_{\mathrm{zero}^\sigma} F \to \sigma$$

Negation corresponds to continuations

Even more interesting the predicate version:

- universal quantification corresponds to polimorphism
- existential quantification to abstract data types

Equational Proof System

$\Gamma \triangleright M = N : \tau$ well-typed axioms $\mathcal{E}$ are given

universally quantified variables

$$\frac{\Gamma \triangleright M = N : \tau \in \mathcal{E}}{\Gamma \triangleright M = N : \tau} \quad \text{(axioms)}$$

$$\frac{\Gamma \triangleright M = N : \sigma}{\Gamma, x : \tau \triangleright M = N : \sigma} \quad \text{(add var)}$$

$$\Gamma \triangleright M = M : \sigma \quad \text{(ref)}$$

$$\frac{\Gamma \triangleright M = N : \sigma}{\Gamma \triangleright N = M : \sigma} \quad \text{(sym)}$$

$$\frac{\Gamma \triangleright M = N : \sigma \quad \Gamma \triangleright N = P : \sigma}{\Gamma \triangleright M = P : \sigma} \quad \text{(trans)}$$

equivalence rules

$$\frac{\Gamma, x : \sigma \triangleright M = N : \tau}{\Gamma \triangleright \lambda x : \sigma . M = \lambda x : \sigma . N : \sigma \to \tau} \quad (\xi)$$

$$\frac{\Gamma \triangleright M_1 = M_2 : \sigma \to \tau \quad \Gamma \triangleright N_1 = N_2 : \sigma}{\Gamma \triangleright M_1 N_1 = M_2 N_2 : \tau} \quad (\nu)$$

congruence rules

$$\Gamma \triangleright \lambda x : \sigma . M = \lambda y : \sigma . [y/x] M : \sigma \to \tau \quad y \notin FV(M) \quad (\alpha)$$

$$\Gamma \triangleright (\lambda x : \sigma . M) N = [N/x] M : \tau \quad (\beta)$$

$$\Gamma \triangleright \lambda x : \sigma . M x = M : \sigma \to \tau \quad x \notin FV(M) \quad (\eta)$$

Axioms for $\times$, $+$, unit, null

$\Gamma \rhd \mathrm{Proj}_1^{\sigma,\tau} \langle M, N \rangle = M : \sigma$ (proj$_1$)

$\Gamma \rhd \mathrm{Proj}_2^{\sigma,\tau} \langle M, N \rangle = N : \tau$ (proj$_2$)

$\Gamma \rhd \langle \mathrm{Proj}_1^{\sigma,\tau} M, \mathrm{Proj}_2^{\sigma,\tau} M \rangle = M : \sigma \times \tau$ (surjective pairing)

$\Gamma \rhd \mathrm{Case}^{\sigma,\tau,\rho} (\mathrm{Inleft}^{\sigma,\tau} M) N P = N M : \rho$ (case)$_1$

$\Gamma \rhd \mathrm{Case}^{\sigma,\tau,\rho} (\mathrm{Inright}^{\sigma,\tau} M) N P = P M : \rho$ (case)$_2$

$\Gamma \rhd \mathrm{Case}^{\sigma,\tau,\cancel{\sigma+\tau}\,\rho} M (N \circ \mathrm{Inleft}^{\sigma,\tau}) (N \circ \mathrm{Inright}^{\sigma,\tau}) = N M : \cancel{\sigma+\tau}\,\rho$ (case)$_3$

where $\Gamma \rhd N : (\sigma+\tau) \to (\cancel{\sigma+\tau})\,\rho$

$\Gamma \rhd M : \sigma + \tau$

$$\frac{\Gamma \rhd M : \tau \to \rho \quad \Gamma \rhd N : \sigma \to \tau}{\Gamma \rhd M \circ N : \sigma \to \rho}$$

$\Gamma \rhd M \circ N = \lambda x : \sigma .\, M(Nx) : \sigma \to \rho$

$\Gamma \rhd M = * : \mathrm{unit}$

$\Gamma \rhd M = \mathrm{zero}^\sigma : \mathrm{null} \to \sigma$

Axioms for PCF

$\ldots\ 3+5=8 \ldots \quad \ldots\ T \wedge F = F \ldots$ (basic)

$\mathrm{Eq?}\ n\ n = T \quad \mathrm{Eq?}\ n\ m = F \quad n \neq m$ (Eq?)

if T then M else $N = M$ if F then M else $N = N$ (cond)

$\mathrm{fix}_\sigma = \lambda f : \sigma \to \sigma .\, f(\mathrm{fix}_\sigma f) : (\sigma \to \sigma) \to \sigma$ (fix)

i.e. $\forall M.\ \mathrm{fix}_\sigma M = M(\mathrm{fix}_\sigma M)$

Categories

Partial Algebras

Specification = Signature + Axioms

Signatures give sorts, and operations names and types.

Atomic formulas are either equations (with existence interpr.) or existence judgements.

Conditional axioms are Horn clauses.

Homomorphisms $h = \{h_s\}_{s \in S} : A \to B$ are total functions with

$f \in \Sigma_{s_1 \ldots s_k, s}$ $\quad a_i \in A_{s_i}$ $\quad f^A(a_1 \ldots a_k)$ defined

implies

$f^B(h_{s_1}(a_1) \ldots h_{s_k}(a_k))$ defined and

$f^B(h_{s_1}(a_1) \ldots h_{s_k}(a_k)) = h_s(f^A(a_1 \ldots a_k))$

Categories are partial algebras:

$S = \{\text{object}, \text{arrow}\}$

$s, t :$ arrow $\to$ object $\qquad s(f) = a \quad t(f) = b$ is $f : a \to b$

$id :$ object $\to$ arrow $\qquad id(a)$ is id_a, sometimes a

$; \; :$ arrow, arrow $\to$ arrow $\qquad ;(f, g)$ is $f;g$

$$\frac{f : a \to b \quad g : b \to c}{f;g : a \to c}$$

it means: $f;g$ is defined and $s(f;g) = s(a)$ and $t(f;g) = t(b)$

$$\frac{f : a \to b}{id_a ; f = f = f ; id_b}$$

(identity)

$$\frac{f : a \to b \quad g : b \to c \quad h : c \to d}{f;(g;h) = (f;g);h} \quad \text{(associativity)}$$

Axioms can be sometimes written as commuting diagrams

$$\begin{tikzcd} a \arrow[r, "f"'] \arrow[rr, bend left, "f;g"] & b \arrow[r, "g"'] & c \end{tikzcd}$$

The objects of C are $|C|$

The arrows of C from a to b are the homset $C[a,b]$

$$\begin{tikzcd} \bullet \arrow[r, "f"] \arrow[dr, "f"'] & b \arrow[d, "id_b"] \arrow[dr, "g"] & \\ & \bullet \arrow[r, "g"'] & \bullet \end{tikzcd}$$

$$\begin{tikzcd} \bullet \arrow[r, "f"] \arrow[d, "f;g"'] & \bullet \arrow[d, "g;h"] \arrow[dl, "g"'] \\ \bullet \arrow[r, "h"'] & \bullet \end{tikzcd}$$

a isomorphic to b iff there are $f : a \to b$ and $g : b \to a$

$$f;g = id_a \qquad g;f = id_b$$

Most categorical constructions are defined up to isomorphism

Functors are homomorphisms

$F : C \to D \qquad F = \{F_{object}, F_{arrow}\}$

$f : a \to b$ implies $F(f) : F(a) \to F(b)$

$F(id^C_a) = id^D_{F(a)}$

$F(f);^D F(g) = F(f;^C g)$

Examples

a) In-the-small:

Simpler example: Path(G) G is a (binary directed) graph

Objects: the nodes of G

Arrows: $p : n \to n'$ the paths in G.

$p;p'$ the concatenation of paths

$id(n)$ the empty path from n to n

More complex examples: G a complex "graph", e.g. Petri net. the generated category is its model of computation

b) In-the-large:

Simpler example: Set

$|Set|$: the collection of all sets

$Set[S_1, S_2]$: the functions $S_1 \to S_2 = S_2^{S_1}$

More complex examples: sets with structures, morphisms respect structure

PO partial orderings and monotone functions

CPO complete partial orderings and continuous functions

Mon monoids and monoid homomorphisms

Alg_Σ Σ-algebras and Σ-homomorphisms

Vect Vector spaces and linear transformations

Cat Categories and functors

Monoidal Categories

Additional operations and axioms

$- \otimes_o - :$ object object $\to$ object
$- \otimes_a - :$ arrow arrow $\to$ arrow $\quad \Rightarrow \quad \otimes$ generic : tensor

$1_o :$ object
$1_a :$ arrow $\quad \Rightarrow \quad 1$ generic

Monoidal axioms:

$$a \otimes 1 = 1 \otimes a = a$$
$$f \otimes 1 = 1 \otimes f = f$$
$$a \otimes (b \otimes c) = (a \otimes b) \otimes c$$
$$f \otimes (g \otimes h) = (f \otimes g) \otimes h$$

All the categorical operations respect the monoidal structure

$$\frac{f : a \to b \quad g : c \to d}{f \otimes g : a \otimes c \to b \otimes d} \qquad 1 : 1 \to 1$$

$$id(a \otimes b) = id(a) \otimes id(b)$$

$$\frac{f_1 : a_1 \to b_1 \quad g_1 : b_1 \to c_1 \quad f_2 : a_2 \to c_2 \quad g_2 : b_2 \to c_2}{(f_1 ; g_1) \otimes (f_2 ; g_2) = f_1 \otimes f_2 ; g_1 \otimes g_2}$$

; and $\otimes$ represent sequential and parallel composition:

$f_1 : u_1 \to v_1 \quad f_2 : u_2 \to v_2$ basic interleaving equations can be derived

$$f_1 \otimes id(u_2) ; id(v_1) \otimes f_2 = (f_1 ; id(v_1)) \otimes (id\, u_2 ; f_2) = f_1 \otimes f_2$$

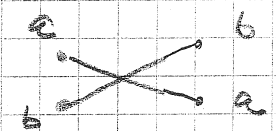
$$f_1 \otimes f_2 = (id(u_1) ; f_1) \otimes (f_2 ; id(v_2)) = id(u_1) \otimes f_2 ; f_1 \otimes id(v_2)$$

Symmetric Monoidal Categories

Additional operations and axioms

$\rho_{-,-}$: object object $\to$ arrow   symmetries

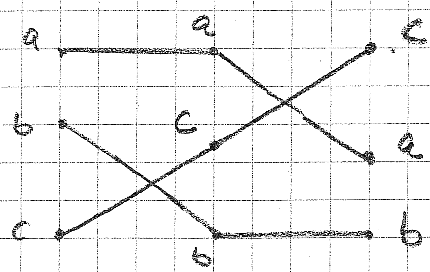
$\rho_{a,b} : a \otimes b \to b \otimes a$ basic intuition: wires



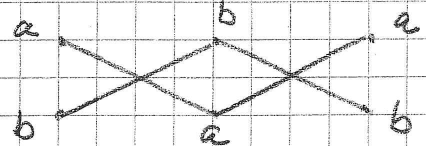
$\rho_{a\otimes b, c} = id_a \otimes \rho_{b,c} ; \rho_{a,c} \otimes id_b$

coherence

$\rho_{1,1} = 1$

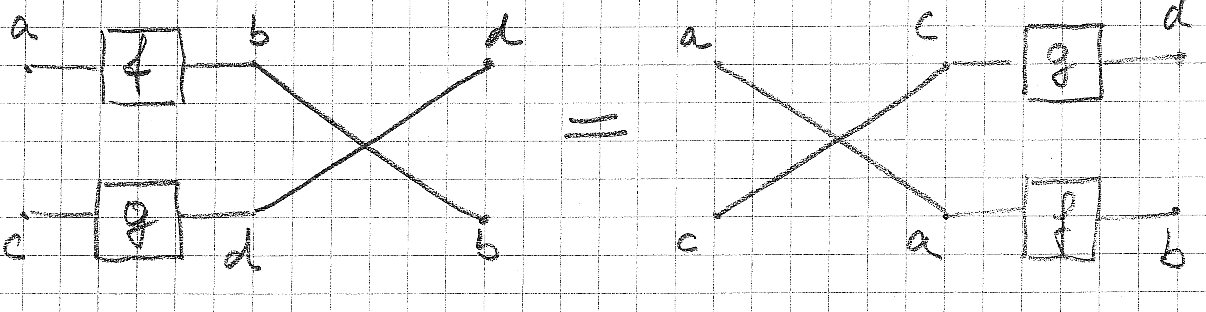


$\rho_{a,b} ; \rho_{b,a} = id_{a \otimes b}$ iso



$$\frac{f : a \to b \quad g : c \to d}{f \otimes g ; \rho_{b,d} = \rho_{a,c} ; g \otimes f}$$

naturality



naturality axiom implies isomorphism of the wire-and-box diagrams

Cartesian Categories (with commutative, chosen products)

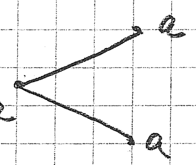
Additional operations and axioms

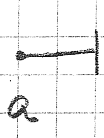
$\nabla_{\_}$: object $\to$ arrow   duplicator

$!_{\_}$: object $\to$ arrow   discharger

$\nabla_a : a \to a \otimes a$ basic intuition

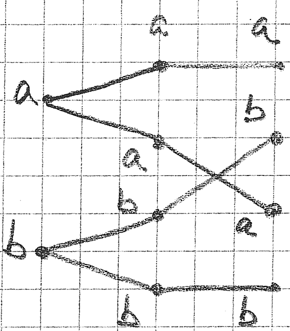
$!_a : a \to 1$





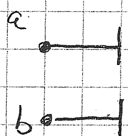
Coherence axioms

$\nabla_{a\otimes b} = \nabla_a \otimes \nabla_b \,;\, id_a \otimes \gamma_{a,b} \otimes id_b$

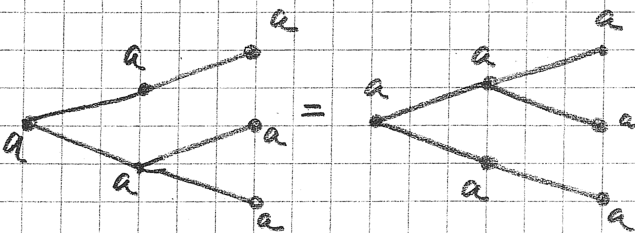


$!_1 = \nabla_1 = 1$

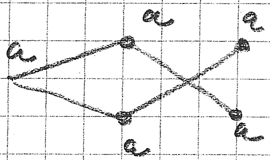
$!_{a\otimes b} = !_a \otimes !_b$



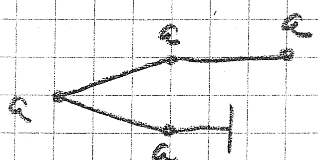
$\nabla_a \,;\, id_a \otimes \nabla_a = \nabla_a \,;\, \nabla_a \otimes id_a$



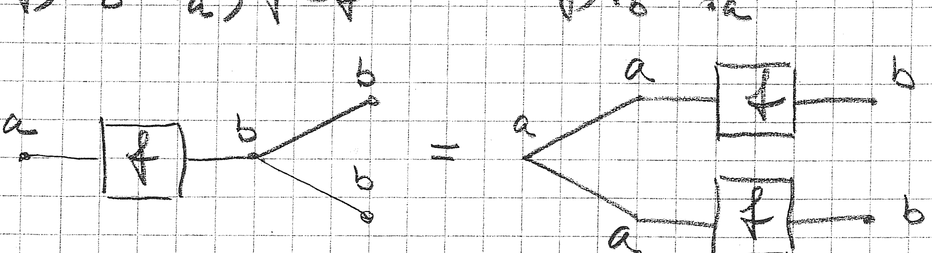
$\nabla_a \,;\, \gamma_{a,a} = \nabla_a$

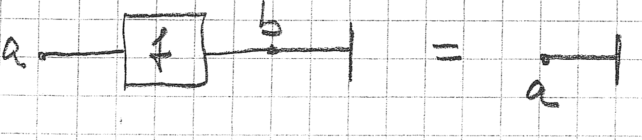


$\nabla_a \,;\, (id_a \otimes !_a) = id_a$



$$\frac{f : a \to b}{f ; \nabla_b = \nabla_a ; f \otimes f} \qquad \frac{f : a \to b}{f ; !_b = !_a} \qquad \text{naturality}$$





The category of variables and substitutions

S, Σ is a signature. Build the category $\underline{Subst}(\Sigma)$ as freely generated by:

$$\frac{s \in S}{s \in |\underline{Subst}(\Sigma)|} \qquad \frac{f \in \Sigma_{s_1 \ldots s_k, s}}{f \in \underline{Subst}(\Sigma)[s_1 \otimes \cdots \otimes s_k, s]}$$

and by the operations and axioms of cartesian categories.
Then $\underline{Subst}(\Sigma)$ has as arrows in the homset

$$\underline{Subs}(\Sigma)[s_1 \otimes \cdots \otimes s_k, s_1' \otimes \cdots \otimes s_h']$$

exactly all substitutions

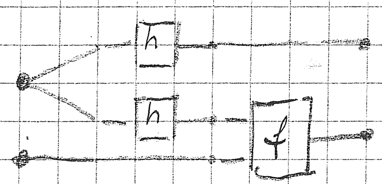
$$\left| \begin{array}{l} y_1 = t_1(x_1, \ldots, x_k) \\ \cdots \\ y_h = t_h(x_1, \ldots, x_k) \end{array} \right. \quad \text{with} \quad \begin{array}{l} x_1 : s_1 \ \ldots \ x_k : s_k \\ y_1 : s_1' \ \ldots \ y_h = s_h' \\ t_i \in T_\Sigma(\{x_1, \ldots, x_k\}) \end{array}$$

Example: Given $S = \{s\}$ $h : s \to s$ $f : s \otimes s \to s$

$y_1 = h(x_1)$
$y_2 = f(h(x_1), x_2)$

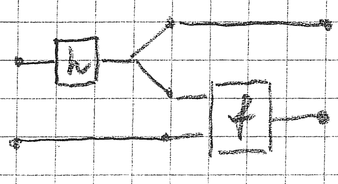
can be represented as

$(\nabla_s \otimes id_s) ; (h \otimes (h \otimes id_s ; f))$



or as

$(h ; \nabla_s \otimes id_s) ; (id_s \otimes f)$



Final Object

object z is final in $\underline{C}$ iff $\forall a \in |\underline{C}|\ \exists!\, l_a : a \to z$

Theorem: the final object is unique up to isomorphism

$$\begin{tikzcd}
z \arrow[r, "id_z = l_z"] \arrow[dr, "l'_z"] \arrow[d, phantom, ""] & z \arrow[d, "l'_z"] \arrow[dl, phantom, "="] \\
z' \arrow[u, "l_{z'}"] \arrow[r, "id_{z'} = l'_{z'}"'] \arrow[ur, phantom, "="] & z'
\end{tikzcd}$$

$l_{z'} ; l'_z = l'_{z'} = id_{z'}$

$l'_z ; l_{z'} = l_z = id_z$

Product Given two objects $a, b \in |\underline{C}|$

$a, b \in |C|$

$a \times b \in |C|$ the product

$\left.\begin{array}{l} \pi_a : a \times b \to a \\ \pi_b : b \times b \to b \end{array}\right\}$ the projections

$$\begin{tikzcd}
 & c \arrow[dl, "f"'] \arrow[d, "h = \langle f, g \rangle"] \arrow[dr, "g"] & \\
a & a \times b \arrow[l, "\pi_a"] \arrow[r, "\pi_b"'] & b
\end{tikzcd}$$

$\forall f, g.\ \exists!\, h$ satisfying the diagram

in **Set**: $h(x) = \langle f(x), g(x) \rangle$

$\pi_a(\langle x, y \rangle) = x$

$\pi_b(\langle x, y \rangle) = y$

or equivalently

$\langle f, g \rangle_c ; \pi_a = f$

$\langle f, g \rangle_c ; \pi_b = g$

$\langle h ; \pi_a , h ; \pi_b \rangle = h$

or equivalently

$\langle -, - \rangle_c : \underline{C}[c, a] \times \underline{C}[c, b] \to C[c, a \times b]$

the pairing operation is a (homset) isomorphism.

Theorem: the product is unique up to isomorphism

$$\begin{tikzcd}
 & a\times b \arrow[dl, "\pi_a"'] \arrow[d, "\langle \pi_a, \pi_b \rangle"] \arrow[dr, "\pi_b"] & \\
a & a\otimes b \arrow[l, "p_a"'] \arrow[r, "p_b"] \arrow[d, "\langle p_a, p_b \rangle"] & b \\
 & a\times b \arrow[ul, "\pi_a"] \arrow[ur, "\pi_b"'] &
\end{tikzcd}$$

All the triangles commute: thus also

$$\langle \pi_a, \pi_b \rangle ; \langle p_a, p_b \rangle = id_{a\times b}$$

since there is one arrow from $a\times b$ to $a\times b$ satisfying the diagram. Similarly for $id_{a\otimes b}$.

The <u>product</u> of arrows: $f: a \to c$ $g: b \to d$ defined if there are products $a\times b$ and $c\times d$

$$f\times g = \langle \pi_a ; f, \pi_b ; g \rangle : a\times b \to c\times d$$

Theorem: $\langle h, k \rangle ; f\times g = \langle h;f, k;g \rangle$

$$\begin{tikzcd}
 & e \arrow[dl, "h"'] \arrow[d, "\langle h,k \rangle"] \arrow[dr, "k"] & \\
a \arrow[d, "f"'] & a\times b \arrow[l, "\pi_a"] \arrow[r, "\pi_b"'] \arrow[d, "f\times g"] & b \arrow[d, "g"] \\
c & c\times d \arrow[l, "\pi_c"] \arrow[r, "\pi_d"'] & d
\end{tikzcd}$$

The two triangles and the two squares commute. Thus also the outer triangles commute: but there is one arrow from e to $c\times d$ which makes them commute

$$\langle h, k \rangle ; f\times g = \langle h;f, k;g \rangle$$

Cartesian Categories

i) There is a terminal object 1.

ii) Every pair a,b has the categorical product

$$(a\times b,\ \pi_1^{a,b} : a\times b \to a,\ \pi_2^{a,b} : a\times b \to b)$$

Two definitions of cartesian categories:
A: in terms of $\otimes, \rho, \nabla, !$ and their axioms
B: in terms of final object and categorical product
Are they the same?

Theorem: Every A-cartesian category is B-cartesian

i) The terminal object is 1 (the identity of the monoid of objects)

ii) The product is defined as $\langle f,g\rangle = \nabla_c ; f\otimes g$, $a\times b = a\otimes b$

$\pi_a = id_a \otimes !_b$
$\pi_b = !_a \otimes id_b$

The unique la from a to 1 is $!_a : a\to 1$

$$\langle f,g\rangle ; \pi_a \stackrel{?}{=} f \qquad \nabla_c ; f\otimes g ; id_a \otimes !_b = \nabla_c ; (f;id_a)\otimes(g;!_b) =$$
$$= \nabla_c ; f\otimes !_c = \nabla_c ; (id_c ; f)\otimes(!_c ; id_1)$$
$$= \nabla_c ; (id\otimes !_c) ; f = id ; f = f$$

$$\langle h; id_a\otimes !_b,\ h; !_a\otimes id_b\rangle \stackrel{?}{=} h$$

$$\nabla_c ; (h; id_a\otimes !_b)\otimes(h; !_a\otimes id_b) = \nabla_c ; h\times h ; id_a\otimes !_b \otimes !_a \otimes id_b =$$
$$= h ; \nabla_{a\otimes b} ; id_a \otimes !_b \otimes !_a \otimes id_b = h ; \nabla_a\otimes\nabla_b ; id_a \otimes \rho_{a,b}\otimes id_b ; id_a\otimes !_b \otimes !_a \otimes id_b =$$
$$= h ; \nabla_a\otimes\nabla_b ; (id_a ; id_a)\otimes(\rho_{a,b} ; !_{b\otimes a})\otimes(id_b ; id_b) = h ; \nabla_a\otimes\nabla_b ; id_a \otimes !_a \times !_b \otimes id_b =$$
$$= h ; (\nabla_a ; id_a\otimes !_a)\otimes(\nabla_b ; !_b \otimes id_b) = h ; id_a \otimes id_b = h ; id_{a\times b} = h$$

Theorem: In a B-cartesian category, the monoidal axioms on objects hold up to isomorphism

Just one case: a and $1\times a$ are isomorphic

$$\begin{tikzcd}
 & 1\times a \arrow[ddl, "\pi_1 = l_{1\times a}"'] \arrow[d, "\pi_a"] \arrow[ddr, "\pi_a"] & \\
 & a \arrow[dl, "l_a"'] \arrow[d, "\langle l_a, id_a\rangle"] \arrow[dr, "id_a"] & \\
1 & 1\times a \arrow[l, "\pi_1"] \arrow[r, "\pi_a"'] & a
\end{tikzcd}$$

the two arrows are:

$\langle l_a, id_a\rangle : a \to 1\times a$

$\pi_a : 1\times a \to a$

All triangle commute: thus

$\langle l_a, id_a\rangle ; \pi_a = id_a$ by constr.

$\pi_a ; \langle l_a, id_a\rangle = id_{1\times a}$ unique arrow

In a B-cartesian category, there may be several objects/arrows $(a\times b, \pi_1, \pi_2), (a\times' b, \pi_1', \pi_2'), \ldots$ however all of them are isomorphic

Similarly for final objects.

However if one chooses products and final object, then

$a\otimes b = a\times b \qquad f\otimes g = f\times g = \langle \pi_a ; f, \pi_b ; g\rangle$

$\rho_{a,b} = \langle \pi_2^{a,b}, \pi_1^{a,b}\rangle$

$\nabla_a = \langle id_a, id_a\rangle$

$!_a = l_a$

Given a B-cartesian category, an A-cartesian category can be obtained by identifying isomorphic objects (skeletal construction).

Cartesian Closed Categories (CCC) caso di Set

Exponent in a cartesian category $\underline{C}$.

Given two object a, b their exponent is an object

$$a \to b \equiv b^a \quad \text{better using the second notation}$$

together with an evalution map

$$eval_{a,b} : b^a \times a \to b$$

such that $\forall f : c \times a \to b \; \exists! h : c \to b^a$ satisfying the diagram

$$\begin{tikzcd} c \arrow[d, "h = \Lambda^{ab}_c f"] \\ b^a \end{tikzcd} \qquad \begin{tikzcd} c \times a \arrow[r, "f"] \arrow[d, "h \times id_a"'] & b \\ b^a \times a \arrow[ur, "eval_{a,b}"'] & \end{tikzcd}$$

or equivalently the axioms

$$(\Lambda^{ab}_c f) \times id_a \, ; eval_{ab} = f \qquad \sim \beta \text{ conversion}$$

$eval_{ab} : (a \to b) \times a \to b$

$$\Lambda^{ab}_c \, h \times id_a \, ; eval_{a,b} = h \qquad \sim \eta \text{ conversion}$$

$eval_{ab} = \lambda(f, x_a) \, f x_a$

or equivalently the homset isomorphism curry $f \, x_c x_a = f(x_c, x_a)$

uncurry $g \, (x_a x_b) = g \, x_a x_b$

$$\underline{C}[c \times a, b] \simeq \underline{C}[c, b^a] \quad \text{with} \quad F(h) = h \times id_a \, ; eval_{a,b}$$

curry $f \times id(a)$; eval $ab = (\lambda x_c . \lambda x_a . f(x_c, x_a))$

A cartesian category $\underline{C}$ is closed if every pair of objects a, b has an exponent $(b^a, eval_{a,b})$.

Equivalently, $\underline{C}$ is closed if functor $\_ \times a : \underline{C} \to \underline{C}$ has a right adjoint.

20bis

Set has exponentials

$b^a = \{h : a \to b\}$ $\qquad \Lambda^{ab}_c = \text{curry} = \lambda \underbrace{f}_{c\times a \to b} . \underbrace{\lambda x_c . \lambda x_a . \underbrace{f\langle x_c, x_a\rangle}_{b}}_{c \to a \to b}$

$\underbrace{\qquad\qquad\qquad}_{(c\times a \to b)\to(c\to a\to b)}$

$\text{eval}_{ab} = \lambda\langle g, x_a\rangle . \underbrace{\underbrace{g}_{b^a}\underbrace{x_a}_{a}}_{b}$

$\underbrace{\qquad\qquad\qquad}_{(b^a\times a)\to b}$

$h_1 ; h_2 = \lambda x . h_2(h_1 x)$

$\left(\Lambda^{ab}_c f\right) \times id_a ; \text{eval}_{ab} \stackrel{?}{=} f$

$\lambda\langle x_c, x_a\rangle . \left(\lambda\langle g, x'_a\rangle . g x'_a\right)\left(\left(\lambda\langle x'_c, x''_a\rangle\langle(\lambda x'''_a . f\langle x'_c, x'''_a\rangle), x''_a\rangle\right)\langle x_c, x_a\rangle\right) =$

$\lambda\langle x_c, x_a\rangle . \left(\lambda\langle g, x'_a\rangle . g x'_a\right)\langle(\lambda x'''_a . f\langle x_c, x'''_a\rangle), x_a\rangle =$

$\lambda\langle x_c, x_a\rangle . \left(\lambda x'''_a . f\langle x_c, x'''_a\rangle\right) x_a =$

$\lambda\langle x_c, x_a\rangle . f\langle x_c, x_a\rangle = f$ $\qquad$ per η convention

$\underbrace{\text{uncurry}}_{(c\to a\to b)\to(c\times a)\to b} = \lambda g . \underbrace{\lambda\langle x_c, x_a\rangle . \underbrace{\underbrace{g}_{c\to a\to b} x_c x_a}_{b}}_{(c\times a)\to b}$

$\underbrace{\qquad\qquad\qquad}_{(c\to a\to b)\to(c\times a)\to b}$

Interpreting $\lambda^{unit, \times, \to}$ on a CCC

Given $\Sigma = \langle B, C \rangle$ we consider a Σ-CCC, i.e. a CCC $\underline{C}$ with

i) chosen final object $(1, !_a)$, products $(a \times b, \pi_1^{ab}, \pi_2^{ab})$ and exponents $(b^a, eval_{ab})$

ii) an interpretation function $\hat{\_}$ with

$b \in B \Rightarrow \hat{b} \in |\underline{C}|$ basic types as objects

extended by structural recursion to all types

$[\![b]\!] = \hat{b}$

$[\![unit]\!] = 1$

$[\![\sigma \times \tau]\!] = [\![\sigma]\!] \times [\![\tau]\!]$

$[\![\sigma \to \tau]\!] = [\![\tau]\!]^{[\![\sigma]\!]}$

also without $[\![-]\!]$: types and objects are just identified

iii) $c : \sigma \in C \Rightarrow \hat{c} \in \underline{C}[1, [\![\sigma]\!]]$

extended by structural recursion to all typing assertions

Here we need to consider Γ as a list of declarations rather than a set. Thus we need a different addvar perm.

$$\frac{x_{f(1)} : \sigma_{f(1)}, \dots, x_{f(m)} : \sigma_{f(m)} \rhd M : \tau}{x_1 : \sigma_1, \dots, x_n : \sigma_n \rhd M : \tau} \qquad f : [1, \dots, m] \to [1, \dots, n] \quad m \leq n \text{ injective}$$

$[\![\Gamma \rhd M : \sigma]\!] \quad [\![\Gamma]\!] \to [\![\sigma]\!]$

Type assignments are mapped to objects

$[\![\emptyset]\!] = 1$ $\qquad [\![\Gamma, x : \sigma]\!] = [\![\Gamma]\!] \times [\![\sigma]\!]$

$$[\![x_1 : \sigma_1, \dots, x_n : \sigma_n \rhd M : \sigma]\!] = X_f^{\sigma_1 \dots \sigma_n} ; [\![x_{f(1)} : \sigma_{f(1)}, \dots, x_{f(m)} : \sigma_{f(m)} \rhd M : \sigma]\!]$$

$$X_f^{\sigma_1 \dots \sigma_n} : (\dots((1 \times \sigma_1) \times \sigma_2) \times \dots) \times \sigma_n \to (\dots((1 \times \sigma_{f_1}) \times \sigma_{f_2}) \times \dots) \times \sigma_{f_m}$$

typing assertions are mapped to arrows by recursion on the typing rules of $\lambda^{\to, \times, unit}$

$$[\![x:\sigma \triangleright x:\sigma]\!] = \pi_2^{1,\sigma}$$

$$[\![\Gamma \triangleright M:\tau]\!] : [\![\Gamma]\!] \to [\![\tau]\!]$$

$$[\![\phi \triangleright * : unit]\!] = id_1$$

$$[\![\phi \triangleright c:\sigma]\!] = \hat{c} : 1 \to \sigma$$

$$[\![\Gamma \triangleright Proj_1^{\sigma,\tau} M:\sigma]\!] = [\![\Gamma \triangleright M:\sigma\times\tau]\!] \,;\, \pi_1^{\sigma,\tau}$$

$$[\![\Gamma \triangleright Proj_2^{\sigma,\tau} M:\tau]\!] = [\![\Gamma \triangleright M:\sigma\times\tau]\!] \,;\, \pi_2^{\sigma,\tau}$$

$$[\![\Gamma \triangleright \langle M,N\rangle : \sigma\times\tau]\!] = \langle [\![\Gamma \triangleright M:\sigma]\!], [\![\Gamma \triangleright N:\tau]\!] \rangle$$

$$[\![\Gamma \triangleright MN:\tau]\!] = \langle [\![\Gamma \triangleright M:\sigma\to\tau]\!], [\![\Gamma \triangleright N:\sigma]\!] \rangle \,;\, eval_{\sigma,\tau}$$

where $[\![\Gamma \triangleright M:\sigma\to\tau]\!] : [\![\Gamma]\!] \to \tau^\sigma$

$[\![\Gamma \triangleright N:\sigma]\!] : [\![\Gamma]\!] \to \sigma$

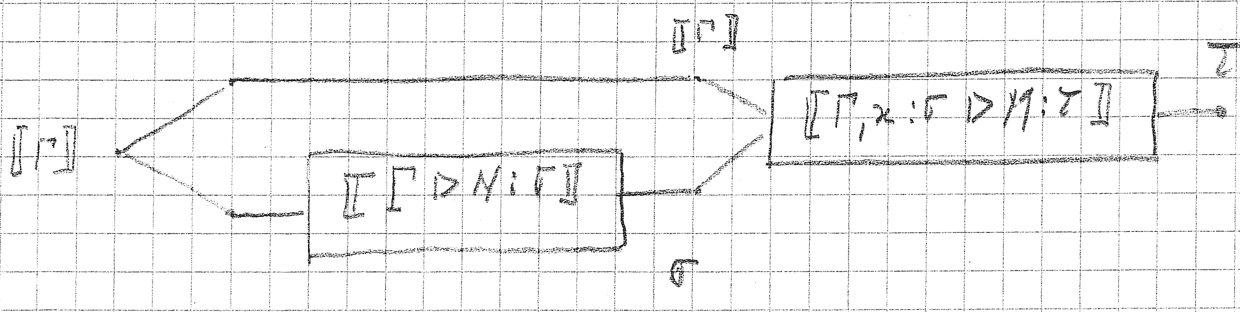
$$[\![\Gamma \triangleright \lambda x:\sigma.M : \sigma\to\tau]\!] = \Lambda^{\sigma,\tau}_{[\![\Gamma]\!]}\, [\![\Gamma, x:\sigma \triangleright M:\tau]\!]$$

where $[\![\Gamma, x:\sigma \triangleright M:\tau]\!] : [\![\Gamma]\!]\times\sigma \to \tau$

$$[\![\Gamma \triangleright M:unit]\!] = l_{[\![\Gamma]\!]} = !_{[\![\Gamma]\!]}$$

Substitution Lemma

$$[\![\Gamma \triangleright [N/x]\, M:\tau]\!] = \langle id_{[\![\Gamma]\!]}, [\![\Gamma \triangleright N:\sigma]\!] \rangle \,;\, [\![\Gamma, x:\sigma \triangleright M:\tau]\!]$$



$$[\![t' [{}^{t}/_{x}]]\!] \rho = [\![t']\!] \rho [{}^{[\![t]\!] \rho}/_{x}]$$

Soundness and Completeness

Provable properties

$\mathcal{E} \vdash \Gamma \triangleright M = N : \sigma$ holds if $\Gamma \triangleright M = N : \sigma$ can be derived from $\mathcal{E}$ using the rules of the λ-calculus

Satisfaction

$\underline{C} \models \Gamma \triangleright M = N : \sigma \iff [\![\Gamma \triangleright M : \sigma]\!] = [\![\Gamma \triangleright N : \sigma]\!]$ in $\underline{C}$

Equational soundness

$\mathcal{E} \vdash \Gamma \triangleright M = N : \sigma$ implies $\underline{C} \models \mathcal{E} \Rightarrow \underline{C} \models \Gamma \triangleright M = N : \sigma$

$\mathcal{E} \models \Gamma \triangleright M = N : \sigma$

Validity

$\forall \underline{C}.\ \underline{C} \models \Gamma \triangleright M = N : \sigma \Rightarrow \phi \vdash \Gamma \triangleright M = N : \sigma$ (not useful here)

every valid formula is provable

Deductive completeness

$\forall C$

$\mathcal{E} \models \Gamma \triangleright M = N : \sigma \Rightarrow \mathcal{E} \vdash \Gamma \triangleright M = N : \sigma$

Minimal model completeness (implies deductive)

Let $\mathcal{E}$ be a syntactic theory ($\mathcal{E} = \{E \mid \mathcal{E} \vdash E\}$ closed)
Then there exist a model $\underline{C}^{\mathcal{E}}$ with

$\mathcal{E} \vdash \Gamma \triangleright M = N : \sigma \iff \underline{C}^{\mathcal{E}} \models \Gamma \triangleright M = N : \sigma$

Soundness of CCC as a model of $\lambda^{unit, \times, \to}$

We have to prove that

$\mathcal{E} \vdash \Gamma \rhd M = N : \tau$ implies $\underline{C} \models \mathcal{E} \Rightarrow \underline{C} \models \Gamma \rhd M = N : \tau$

by rule induction. We assume $\underline{C} \models \mathcal{E}$ and we prove

$\underline{P}(\Gamma \rhd M = N : \tau) \overset{def}{\iff} [\![\Gamma \rhd M : \tau]\!] = [\![\Gamma \rhd N : \tau]\!]$ in $\underline{C}$

(axioms) $\dfrac{\Gamma \rhd M = N : \tau \in \mathcal{E}}{\Gamma \rhd M = N : \tau}$ obvious since we assumed $\underline{C} \models \mathcal{E}$

(~~addvar~~ perm) $\dfrac{x_{f(1)} : \tau_{f(1)}, \ldots, x_{f(m)} : \tau_{f(m)} \rhd \underbrace{M = N : \tau}}{x_1 : \tau_1, \ldots, x_n : \tau_n \rhd M = N : \tau}$ obvious since

$$[\![x_1 : \tau_1, \ldots, x_n : \tau_n \rhd M : \tau]\!] = X_f^{\tau_1 \ldots \tau_n} ; [\![x_{f(1)} : \tau_{f(1)}, \ldots, x_{f(m)} : \tau_{f(m)} \rhd M : \tau]\!]$$

(ref, sym, trans) obvious for $[\![-]\!] = [\![-]\!]$

(ξ) $\dfrac{\Gamma, x : \sigma \rhd M = N : \tau}{\Gamma \rhd \lambda x : \sigma . M = \lambda x : \sigma . N : \sigma \to \tau}$ $[\![\Gamma, x : \sigma \rhd M : \tau]\!] = [\![\Gamma, x : \sigma \rhd N : \tau]\!]$ implies $[\![\Gamma \rhd \lambda x : \sigma . M]\!] = [\![\lambda x : \sigma . N : \sigma \to \tau]\!]$?

Obvious, since

$$[\![\Gamma \rhd \lambda x : \sigma . M : \sigma \to \tau]\!] = \Lambda_{[\![\Gamma]\!]}^{\sigma, \tau}\left([\![\Gamma, x : \sigma \rhd M : \tau]\!]\right)$$

$$[\![\Gamma \rhd \lambda x : \sigma . N : \sigma \to \tau]\!] = \Lambda_{[\![\Gamma]\!]}^{\sigma, \tau}\left([\![\Gamma, x : \sigma \rhd N : \tau]\!]\right)$$

(ν) very similar

(α) $\Gamma \triangleright \lambda x{:}\sigma.M = \lambda y{:}\sigma.[y/x]M : \sigma \to \tau \qquad y \notin FV(M)$

We prove by structural induction on M a stronger property

$[\![\rho(\Gamma \triangleright M : \tau)]\!] = [\![\Gamma \triangleright M : \tau]\!]$ for every injective substitution ρ operating on <u>all</u> the variables of Γ and M

names do not count: only their position do.

→ obvious proof.

(β) $[\![\Gamma \triangleright (\lambda x{:}\sigma.M)N : \tau]\!] \stackrel{?}{=} [\![\Gamma \triangleright [N/x]M : \tau]\!]$

$[\![\Gamma \triangleright (\lambda x{:}\sigma.M)N : \tau]\!] = \langle \Lambda^{\sigma,\tau}_{[\![\Gamma]\!]}[\![\Gamma, x{:}\sigma \triangleright M{:}\tau]\!], [\![\Gamma \triangleright N{:}\sigma]\!]\rangle ; eval_{\sigma,\tau} =$

$= \langle id_{[\![\Gamma]\!]}, [\![\Gamma \triangleright N{:}\sigma]\!]\rangle ; \Lambda^{\sigma,\tau}_{[\![\Gamma]\!]}[\![\Gamma, x{:}\sigma \triangleright M{:}\tau]\!] \times id_\sigma ; eval_{\sigma,\tau}$

remember that $\langle h;f, k;g\rangle = \langle h,k\rangle ; f \times g$

$= \langle id_{[\![\Gamma]\!]}, [\![\Gamma \triangleright N{:}\sigma]\!]\rangle ; [\![\Gamma, x{:}\sigma \triangleright M{:}\tau]\!]$ for an axiom of CCC

$= [\![\Gamma \triangleright [N/x]M : \tau]\!]$ for the substitution lemma QED.

$$\begin{tikzcd}
{[\![\Gamma]\!]} \arrow[d, "\Lambda^{\sigma,\tau}_{[\![\Gamma]\!]}[\![M]\!]"'] \\
\tau^\sigma
\end{tikzcd}
\qquad
\begin{tikzcd}
{[\![\Gamma]\!]} \arrow[d, "{\langle id_{[\![\Gamma]\!]}, [\![N]\!]\rangle}"] & \\
{[\![\Gamma]\!] \times \sigma} \arrow[r, "{[\![M]\!]}"] \arrow[d, "\Lambda^{\sigma,\tau}_{[\![\Gamma]\!]}[\![M]\!] \times id_\sigma"'] & \tau \\
\tau^\sigma \times \sigma \arrow[ur, "eval_{\sigma,\tau}"'] &
\end{tikzcd}$$

eddvau!

(η) $[\![\Gamma \rhd \lambda x:\sigma.\, Mx : \sigma \to \tau]\!] \stackrel{?}{=} [\![\Gamma \rhd M : \sigma \to \tau]\!] \qquad x \neq FV(M)$

$[\![\Gamma \rhd \lambda x:\sigma.\, Mx]\!] = \Lambda^{\sigma,\tau}_{[\![\Gamma]\!]} \langle [\![\Gamma, x:\sigma \rhd M]\!], [\![\Gamma, x:\sigma \rhd x]\!] \rangle ; eval_{\sigma,\tau}$

Lemma: $\langle [\![\Gamma, x:\sigma \rhd M]\!], [\![\Gamma, x:\sigma \rhd x]\!] \rangle = [\![\Gamma \rhd M]\!] \times id_\sigma$ if $x \notin M$

$[\![\Gamma \rhd M]\!] \times id_\sigma = \langle [\![\Gamma \rhd M]\!] \times id_\sigma ; \pi_1^{\sigma\to\tau,\sigma}, [\![\Gamma \rhd M]\!] \times id_\sigma ; \pi_2^{\sigma\to\tau,\sigma} \rangle$

$= \langle [\![\Gamma, x^\sigma \rhd M]\!] \cancel{;} [\![\Gamma, x^\sigma \rhd x^\sigma]\!] \rangle$

$[\![\Gamma \rhd \lambda x:\sigma.\, Mx]\!] = \Lambda^{\sigma,\tau}_{[\![\Gamma]\!]} [\![\Gamma \rhd M]\!] \times id ; eval_{\sigma,\tau} = [\![\Gamma \rhd M]\!]$ Q.E.D.

$$\begin{tikzcd}
{[\![\Gamma]\!]} \arrow[d, "\Lambda^{\sigma,\tau}_{[\![\Gamma]\!]}([\![M]\!] \times id_\sigma ; eval_{\sigma,\tau})"] \\
\tau^\sigma
\end{tikzcd}
\qquad
\begin{tikzcd}
{[\![\Gamma \rhd M]\!] \times id_\sigma} \arrow[r, "{[\![M]\!] \times id_\sigma ; eval_{\sigma,\tau}}"] \arrow[d, "{[\![\Gamma \rhd M]\!] \times id_\sigma}"] & \tau \\
\tau^\sigma \times \sigma \arrow[ur, "eval_{\sigma,\tau}"'] &
\end{tikzcd}$$

(proj$_1$) $[\![\Gamma \rhd Proj_1^{\sigma,\tau} \langle M, N \rangle]\!] \stackrel{?}{=} [\![\Gamma \rhd M]\!]$ idem for (proj$_2$)

$[\![\Gamma \rhd Proj_1^{\sigma,\tau} \langle M, N \rangle]\!] = \langle [\![\Gamma \rhd M]\!], [\![\Gamma \rhd N]\!] \rangle ; \pi_1^{\sigma,\tau} = [\![\Gamma \rhd M]\!]$

(SP) $[\![\Gamma \rhd \langle Proj_1^{\sigma,\tau} M, Proj_2^{\sigma,\tau} M \rangle]\!] = [\![\Gamma \rhd M]\!]$

$[\![\Gamma \rhd \langle Proj_1^{\sigma,\tau} M, Proj_2^{\sigma,\tau} M \rangle]\!] = \langle [\![\Gamma \rhd M]\!] ; \pi_1^{\sigma,\tau}, [\![\Gamma \rhd M]\!] ; \pi_2^{\sigma,\tau} \rangle = [\![\Gamma \rhd M]\!]$

(unit) $[\![\Gamma \rhd M : unit]\!] = [\![\Gamma \rhd * : unit]\!] = l_{[\![\Gamma]\!]}$

Completeness of CCC as a model of $\lambda^{unit,\times,\to}$

Minimal model completeness

Given the syntactic theory $\mathcal{E}$, we build the algebraic specification Γ which includes $\mathcal{E}$ and all the axioms of cartesian closed categories.
We then construct the category $Alg(\Gamma)$ of the models of Γ.

Thus:

- The objects of $Alg(\Gamma)$ are CCC which satisfy $\mathcal{E}$.
- The arrows of $Alg(\Gamma)$ are cartesian closed functors: e.g. $F(\Lambda_c^{ab} f) = \Lambda_{F(c)}^{F(a)\,F(b)} F(f)$
- The initial object of $Alg(\Gamma)$ is the minimal model of $\mathcal{E}$

$$\mathcal{E} \vdash \Gamma \rhd M = N : \sigma \quad \Longleftrightarrow \quad \underline{C}^{\mathcal{E}} \models \Gamma \rhd M = N : \sigma$$

Models of $\lambda^{\to, \times, unit, +, null}$

Cocartesian categories: dual of cartesian categories
invert directions of arrows

Initial object: 0

$\forall a \in |\underline{C}|. \exists! \ t_a : 0 \to a$

Coproduct: $a, b \in |\underline{C}|$
$a+b \in |\underline{C}|$
$\left.\begin{array}{l} q_a : a \to a+b \\ q_b : b \to a+b \end{array}\right\}$ injections

$$\begin{tikzcd} & c & \\ a \arrow[ur, "f"] \arrow[r, "q_a"'] & a+b \arrow[u, "h = {[f,g]}"] & b \arrow[ul, "g"'] \arrow[l, "q_b"] \end{tikzcd}$$

$\forall f : a \to c. \forall g : b \to c. \exists! \ a+b \to c. \ f = q_a ; h \ \wedge \ g = q_b ; h$

$C[a+b, c] \simeq C[a, c] \times C[b, c]$

$[q_a ; h, q_b ; h] = h$ $\qquad$ $q_a ; [f, g] = f$
$\qquad\qquad\qquad\qquad\quad\ q_b ; [f, g] = f$

Cocartesian categories: - initial object
- coproduct for every pair a, b.

Models with union and null types: cocartesian CCC

Inleft, Inright and Case are interpreted
using q_a, q_b and $[f, g]$.