Cartesian closed categories
CCC

typed functional programming
vs.
category theory

typed lambda-calculi
vs.
category theory

Cartesian categories

Definition: *A category* $\mathbf{K}$ *is Cartesian if it comes equipped with finite products.*

Equivalently:

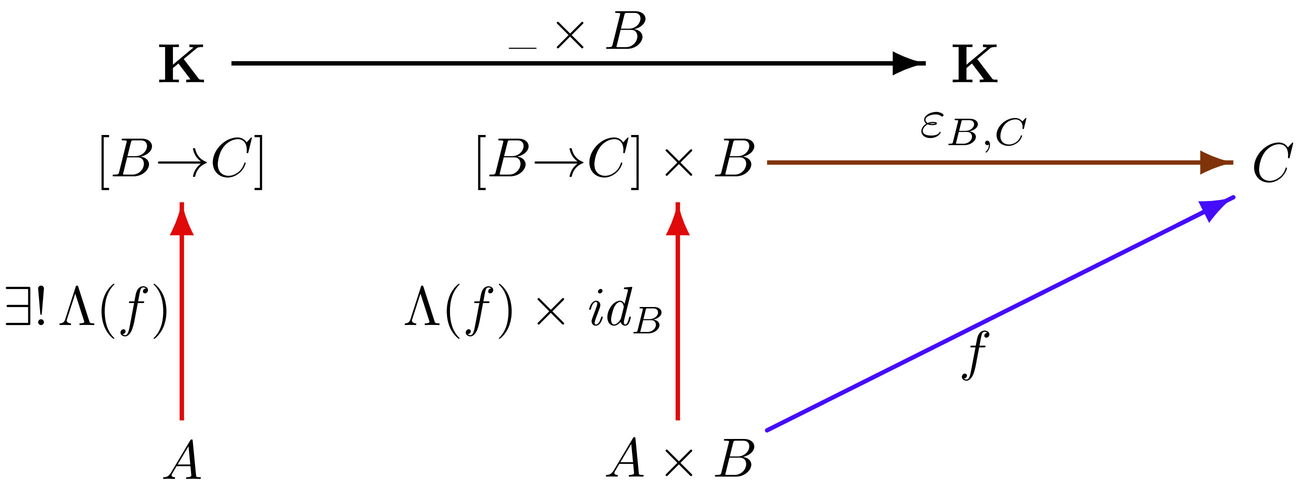
- $1 \in |\mathbf{K}|$ — a terminal object
- $A \times B$ — a product of A and B, for every $A, B \in |\mathbf{K}|$

Examples: $\mathbf{Set}$, $\mathbf{Pfn}$, $\mathbf{Cpo}$, *semilattices*, $\mathbf{Cat}$, $T^{op}_{\Sigma,\Phi}$, ...

Recall the definitions of these categories and the constructions of products in each of them

Cartesian closed categories

Definition: *A Cartesian category* $\mathbf{K}$ *is closed if for all* $B, C \in |K|$ *we indicate* $[B \to C] \in |\mathbf{K}$ *and* $\varepsilon_{B,C} \colon [B \to C] \times B \to C$ *such that for all* $A \in |\mathbf{K}|$ *and* $f \colon A \times B \to C$ *there is a unique* $\Lambda(f) \colon A \to [B \to C]$ *satisfying*
$(\Lambda(f) \times id_B);\varepsilon_{B,C} = f$



Examples: $\mathbf{Set}$, $\mathbf{Cpo}$, $\mathbf{Cat}$, ...
Heyting semilattices ($b \Leftarrow c$ *is such that for all* a, $a \wedge b \leq c$ *iff* $a \leq (b \Leftarrow c)$*)*

Non-examples: $\mathbf{Pfn}$, $T^{op}_{\Sigma,\Phi}$.

Summing up

A category $\mathbf{K}$ is a *Cartesian closed category* (*CCC*) if:

- $\mathbf{C}: \mathbf{K} \to \mathbf{1}$ has a right adjoint $\mathbf{C}_1: \mathbf{1} \to \mathbf{K}$, yielding $1 \in |K|$.
- $\Delta: \mathbf{K} \to \mathbf{K} \times \mathbf{K}$ has a right adjoint $\_ \times \_: \mathbf{K} \times \mathbf{K} \to \mathbf{K}$ with counit given by $\pi_{A,B}: A \times B \to A$ and $\pi'_{A,B}: A \times B \to B$ for $A, B \in |K|$.
- for each $B \in |K|$, $\_ \times B: \mathbf{K} \to \mathbf{K}$ has right adjoint $[B \to \_]: \mathbf{K} \to \mathbf{K}$, with counit given by $\varepsilon_{B,C}: [B \to C] \times B \to C$, for $C \in |\mathbf{K}|$.

Spelling this out

- $1 \in |\mathbf{K}|$
  - for $A \in |K|$: $\langle\rangle_A \colon A \to 1$ such that $\langle\rangle_A = f$ for all $f \colon A \to 1$.
- for $A, B \in |\mathbf{K}|$, $A \times B \in |\mathbf{K}|$, $\pi_{A,B} \colon A \times B \to A$, $\pi'_{A,B} \colon A \times B \to B$:
  - for $C \in |\mathbf{K}|$, for $f \colon C \to A$, $g \colon C \to B$: $\langle f, g\rangle \colon C \to A \times B$ such that
    - $\langle f, g\rangle;\pi_{A,B} = f$ and $\langle f, g\rangle;\pi'_{A,B} = g$
    - for $h \colon C \to A \times B$, $h = \langle h;\pi_{A,B}, h;\pi'_{A,B}\rangle$
- for $B, C \in |\mathbf{K}|$, $[B \to C] \in |\mathbf{K}|$, $\varepsilon_{B,C} \colon [B \to C] \times B \to C$:
  - for $A \in |\mathbf{K}|$, for $f \colon A \times B \to C$: $\Lambda(f) \colon A \to [B \to C]$ such that
    - $(\Lambda(f) \times id_B);\varepsilon_{B,C} = f$
    - for $h \colon A \to [B \to C]$, $\Lambda((h \times id_B);\varepsilon_{B,C}) = h$.

Typed λ-calculus with products

Types

The set $\mathcal{T}$ of *types* $\tau \in \mathcal{T}$ is such that

- $1 \in \mathcal{T}$
- $\tau \times \tau' \in \mathcal{T}$, for all $\tau, \tau' \in \mathcal{T}$
- $\tau \to \tau' \in \mathcal{T}$, for all $\tau, \tau' \in \mathcal{T}$

Note: $\mathcal{T}$ need not be the least such that. . .

Contexts

Contexts Γ are of the form:

- $x_1{:}\tau_1, \ldots, x_n{:}\tau_n$,
  where $n \geq 0$, $x_1, \ldots, x_n$ are distinct variables, and $\tau_1, \ldots, \tau_n \in \mathcal{T}$

Typed terms in contexts

$$\Gamma \vdash t \colon \tau$$

Typing/formation rules coming next

Omitting the usual definitions, like:

- *free variables* $FV(M)$,
- *substitution* $M[N/x]$, etc.

Same for the usual simple properties, like:

- weakening — context extension;
- subject reduction;
- uniqueness of types;
- removing unused variables from contexts;

  etc

Typing rules

$$\frac{}{x_1{:}\tau_1, \ldots, x_n{:}\tau_n \vdash x_i : \tau_i}$$

$$\frac{x_1{:}\tau_1, \ldots, x_n{:}\tau_n \vdash M : \tau}{x_1{:}\tau_1, \ldots, x_{i-1}{:}\tau_{i-1}, x_{i+1}{:}\tau_{i+1}, \ldots, x_n{:}\tau_n \vdash \lambda x_i{:}\tau_i.M : \tau_i{\to}\tau}$$

$$\frac{\Gamma \vdash M : \tau{\to}\tau' \qquad \Gamma \vdash N : \tau}{\Gamma \vdash MN : \tau'}$$

$$\frac{}{\Gamma \vdash \langle\rangle : 1} \qquad \frac{\Gamma \vdash M : \tau \qquad \Gamma \vdash N : \tau'}{\Gamma \vdash \langle M, N\rangle : \tau{\times}\tau'}$$

$$\frac{}{\Gamma \vdash \pi_{\tau,\tau'} : \tau{\times}\tau'{\to}\tau} \qquad \frac{}{\Gamma \vdash \pi'_{\tau,\tau'} : \tau{\times}\tau'{\to}\tau'}$$

Semantics

Let $\mathbf{K}$ be an arbitrary but fixed CCC.

- Types denote objects, $[\![\tau]\!] \in |\mathbf{K}|$, satisfying:
  - $[\![1]\!] = 1$
  - $[\![\tau \times \tau']\!] = [\![\tau]\!] \times [\![\tau']\!]$
  - $[\![\tau \rightarrow \tau']\!] = [[\![\tau]\!] \rightarrow [\![\tau']\!]]$
- So do contexts:
  - $[\![x_1{:}\tau_1, \ldots, x_n{:}\tau_n]\!] = [\![\tau_1]\!] \times \ldots \times [\![\tau_n]\!]$
- Terms denote morphisms:

$$[\![M]\!] \colon [\![\Gamma]\!] \rightarrow [\![\tau]\!]$$

defined by induction on the derivation of $\Gamma \vdash M : \tau$ (coming next).

Semantics of λ-terms

- $[\![x_i]\!] = \pi_i \colon [\![\Gamma]\!] \to [\![\tau_i]\!]$ for $\Gamma = x_1{:}\tau_1, \ldots, x_n{:}\tau_n$, where π_i is the obvious projection.
- $[\![\lambda x_i{:}\tau_i.M]\!] = \Lambda(\rho;[\![M]\!]) \colon [\![\Gamma']\!] \to [[\![\tau_i]\!]{\to}[\![\tau]\!]]$ for $\Gamma = x_1{:}\tau_1, \ldots, x_n{:}\tau_n$, $\Gamma' = x_1{:}\tau_1, \ldots, x_{i-1}{:}\tau_{i-1}, x_{i+1}{:}\tau_{i+1}, \ldots, x_n{:}\tau_n$, and $\Gamma \vdash M \colon \tau$, where $\rho \colon [\![\Gamma']\!] \times [\![\tau_i]\!] \to [\![\Gamma]\!]$ is the obvious isomorphism.
- $[\![M\,N]\!] = \langle [\![M]\!], [\![N]\!] \rangle;\varepsilon_{[\![\tau]\!],[\![\tau']\!]} \colon [\![\Gamma]\!] \to [\![\tau']\!]$ for $\Gamma \vdash M \colon \tau{\to}\tau'$ and $\Gamma \vdash N \colon \tau$.
- $[\![\langle\rangle]\!] = \langle\rangle_{[\![\Gamma]\!]} \colon [\![\Gamma]\!] \to 1$
- $[\![\langle M, N\rangle]\!] = \langle [\![M]\!], [\![N]\!] \rangle \colon [\![\Gamma]\!] \to [\![\tau]\!] \times [\![\tau']\!]$ for $\Gamma \vdash M \colon \tau$ and $\Gamma \vdash N \colon \tau'$.
- $[\![\pi_{\tau,\tau'}]\!] = \Lambda(\pi'_{[\![\Gamma]\!],[\![\tau]\!]\times[\![\tau']\!]};\pi_{[\![\tau]\!],[\![\tau']\!]}) \colon [\![\Gamma]\!] \to [[\![\tau]\!] \times [\![\tau']\!]{\to}[\![\tau]\!]]$ and
  $[\![\pi'_{\tau,\tau'}]\!] = \Lambda(\pi'_{[\![\Gamma]\!],[\![\tau]\!]\times[\![\tau']\!]};\pi'_{[\![\tau]\!],[\![\tau']\!]}) \colon [\![\Gamma]\!] \to [[\![\tau]\!] \times [\![\tau']\!]{\to}[\![\tau']\!]]$

Equational β, η-calculus

Judgements:

$$\Gamma \vdash M = N : \tau$$

for $\tau \in \mathcal{T}$, $\Gamma \vdash M : \tau$, $\Gamma \vdash N : \tau$

Axioms:

(β) $\Gamma \vdash (\lambda x{:}\tau.M)N = M[N/x] : \tau'$, for $\Gamma \vdash \lambda x{:}\tau.M : \tau \to \tau'$, $\Gamma \vdash N : \tau$

(η) $\Gamma \vdash \lambda x{:}\tau.Mx = M : \tau \to \tau'$, for $\Gamma \vdash M : \tau \to \tau'$, $x \notin dom(\Gamma)$

- $\Gamma \vdash M = \langle\rangle : 1$, for $\gamma \vdash M : 1$
- $\Gamma \vdash \pi_{\tau,\tau'}\langle M, N\rangle = M : \tau$ and $\Gamma \vdash \pi'_{\tau,\tau'}\langle M, N\rangle = N : \tau'$, for $\Gamma \vdash M : \tau$, $\Gamma \vdash N : \tau'$
- $\Gamma \vdash M = \langle \pi_{\tau,\tau'}M, \pi'_{\tau,\tau'}M\rangle : \tau \times \tau'$, for $\Gamma \vdash M : \tau \times \tau'$

Rules: reflexivity, symmetry, transitivity, congruence.

Soundness

Given $\Gamma \vdash M : \tau$ and $\Gamma \vdash N : \tau$

if $\Gamma \vdash M = N : \tau$ then $[\![M]\!] = [\![N]\!]$

Proof: :-)

Just check that the axioms and rules of the equational β, η-calculus are sound w.r.t. the semantics in any CCC. For example:

(η) for $\Gamma \vdash M : \tau \to \tau'$, $x \notin dom(\Gamma)$, given the isomorphism $\rho \colon [\![\Gamma]\!] \times [\![\tau]\!] \to [\![\Gamma, x{:}\tau]\!]$:
$[\![\lambda x{:}\tau.Mx]\!] = \Lambda(\rho;[\![Mx]\!]) = \Lambda(\rho;(\langle [\![M]\!], [\![x]\!]\rangle;\varepsilon_{[\![\tau]\!],[\![\tau']\!]})) =$
$\Lambda(\langle \rho;[\![M]\!], \rho;[\![x]\!]\rangle;\varepsilon_{[\![\tau]\!],[\![\tau']\!]}) = \Lambda(([\![M]\!] \times id_{[\![\tau]\!]});\varepsilon_{[\![\tau]\!],[\![\tau']\!]}) = [\![M]\!]$.

Warning: The "real work" is in the proof of soundess for (β), where induction on the structure of terms is needed.

Completeness

Given $\Gamma \vdash M : \tau$ and $\Gamma \vdash N : \tau$,

if in every CCC, $[\![M]\!] = [\![N]\!]$ then $\Gamma \vdash M = N : \tau$

Proof: It is enough to prove this for terms in the empty context.

Define a CCC $\boldsymbol{\lambda}$:

- Category $\boldsymbol{\lambda}$:
  - objects are all types: $|\boldsymbol{\lambda}| = \mathcal{T}$
  - morphisms are λ-terms modulo equality: $\boldsymbol{\lambda}(\tau, \tau') = \{M \mid \vdash M : \tau{\rightarrow}\tau'\}/{\approx}$, where $M \approx N$ iff $\vdash M = N : \tau'{\rightarrow}\tau$
  - composition: $[M]_{\approx};[N]_{\approx} = [\lambda x{:}\tau.N(Mx)]_{\approx}$, for $\vdash M : \tau{\rightarrow}\tau'$, $\vdash N : \tau'{\rightarrow}\tau''$
  - identities: $id_\tau = [\lambda x{:}\tau.x]_{\approx}$.

- Products in $\boldsymbol{\lambda}$:
  - terminal object $1 \in |\boldsymbol{\lambda}|$, with $\langle\rangle_\tau = [\lambda x{:}\tau.\langle\rangle]_\approx$
  - binary product $\tau\times\tau'$, with $\pi_{\tau,\tau'} = [\pi_{\tau,\tau'}]_\approx$, $\pi'_{\tau,\tau'} = [\pi'_{\tau,\tau'}]_\approx$ and pairing $\langle [M]_\approx, [N]_\approx\rangle = [\langle M, N\rangle]_\approx$ for $\vdash M : \tau$, $\vdash N : \tau'$.
- Exponent in $\boldsymbol{\lambda}$: $\tau\to\tau'$, with $\varepsilon_{\tau,\tau'} = [\lambda x{:}(\tau\to\tau')\times\tau.(\pi_{\tau\to\tau',\tau}x)(\pi'_{\tau\to\tau',\tau}x)]_\approx$ and $\Lambda([M]_\approx) = [\lambda x{:}\tau.\lambda y{:}\tau'.M\langle x, y\rangle]_\approx$, for $\vdash M : \tau\times\tau'\to\tau''$.

Now: if in every CCC, $[\![M]\!] = [\![N]\!]$, then this holds in particular in $\boldsymbol{\lambda}$, and so $\Gamma \vdash M = N : \tau$.

To wrap this up: add constants of arbitrary types

SUMMING UP:

CCCs coincide with λ-calculi