# Introduction

Mathematical Semantics of Computer Systems

Syntax ⟷ Semantics

- We strive for mathematically clean semantical models.

- Category theory
  圏論

  A mathematical language

  * from algebraic topology, ~'50
  * used in many disciplines in (pure) math.
  * used in CS / semantics, in the following principal ways

    + the Curry-Howard-Lambek correspondence

      λ-calculus

      intuitionistic logic

      Cartesian closed categories

$\simeq$ + Coalgebra

automata

+ graph rewriting

- categories give us abstraction & genericity
- uniform theory of various kinds of automata (det., non-det., probabilistic, weighted, quantum, ...)

## Verification

input: a system model $\mathcal{M}$

a specification $\varphi$

(a property to be desired)

output: $\mathcal{M} \models \varphi$

"yes", and a mathematical proof for $\mathcal{M} \models \varphi$

"no", (and hopefully a counter example)

"satisfies"

Textbooks

first CT book

[Awodey]  Easy to follow esp. for logicians & CS people

[Leinster]

[MacLane]. Categories for the Working Mathematician.

[Lambek & Scott].

[Jacobs'12]. Intro. to Coalg.

[Jacobs'99] fibration

$$
\begin{tikzcd}
FX & FR \arrow[l, hook'] \arrow[r] & FY \\
X \arrow[u] & R \arrow[l, hook'] \arrow[r] \arrow[u] & Y \arrow[u]
\end{tikzcd}
$$

Commutative diagrams in the language of CT

Definition (category)

A category is given by

$$\mathbb{C} = \Big( \overbrace{\mathcal{O},\ A}^{\text{"sets"}},\ \overbrace{(id_X)_{X\in\mathcal{O}},\ (\circ_{X,Y,Z})_{X,Y,Z\in\mathcal{O}}}^{\text{"operations"}} \Big)$$

where $A = \big(A(X,Y)\big)_{X,Y\in\mathcal{O}}$

- $\mathcal{O}$ is a "collection" of objects
- $A(X,Y)$ is a "collection" of arrows from an object $X$ to another obj. $Y$
- $id_X \in A(X,X)$ is called the identity arrow over $X$, and
- $\circ_{X,Y,Z} : A(Y,Z) \times A(X,Y) \to A(X,Z)$

$f \in A(X,Y)$

$f : X \to Y$ in $\mathbb{C}$

$X \circlearrowleft id_X$

composition

$X \xrightarrow{f} Y \xrightarrow{g} Z$

$g \circ f$

subject to

- the identity law

$$\left[\begin{tikzcd} X \arrow[r, "f"] \arrow[dr, "f"'] & Y \arrow[d, "id_Y"] \arrow[dr, "g"] & \\ & Y \arrow[r, "g"'] & Z \end{tikzcd}\right]$$

equations

For each $X, Y, Z \in \mathcal{O}$

$f \in A(X,Y)$

$g \in A(Y,Z)$

$\circ_{X,Y,Y}$ $\circ_{Y,Y,Z}$

$$id_Y \circ f = f, \quad g \circ id_Y = g$$

the associativity law

$$\left[\begin{tikzcd} X \arrow[r, "f"] \arrow[dr, "g\circ f"'] & Y \arrow[d, "g"] \arrow[dr, "h\circ g"] & \\ & Z \arrow[r, "h"] & V \end{tikzcd}\right]$$

"$h\circ g\circ f$" makes sense

$$h\circ(g\circ f) = (h\circ g)\circ f.$$

---

an algebra = a set
+ operations
+ equations.

---

## Some terminologies

- $X \xrightarrow{f} Y$ in $\mathbb{C}$

f's domain

f's codomain

$$\begin{tikzcd} X \arrow[r, "f"] \arrow[dr, "g\circ f"'] & Y \arrow[d, "g"] \\ & Z \end{tikzcd} \quad \lfloor \mathbb{C}$$

$g\circ f$ "g after f"

$=$

$g \circ_{X,Y,Z} f$

follows function composition:
$(g\circ f)(x) = g(f(x))$

$=$

$f ; g$

## Example

- The category **Sets** of sets and functions

  obj. sets

  arr. functions between them.

  id. $\mathrm{id}_X : X \longrightarrow X$ in **Sets**

  is given by the function

$$\begin{array}{ccc} X & \longrightarrow & X \\ \rotatebox{90}{$\in$} & & \rotatebox{90}{$\in$} \\ x & \longmapsto & x \end{array}$$

  comp. Given $X \xrightarrow{f} Y \xrightarrow{g} Z$ in **Sets**

  we define $g \circ f : X \longrightarrow Z$ in **Sets**

  by the function

$$\begin{array}{ccc} X & \longrightarrow & Z \\ \rotatebox{90}{$\in$} & & \rotatebox{90}{$\in$} \\ x & \longmapsto & g(f(x)) \end{array}$$

NB - $\{\text{sets}\}$ is not a set.

it is a proper class. a small set