Wiring Diagrams, a categorical formalism for systems modeling

Christina Vasilakopoulou

University of Patras, Greece

Department of Mathematics and Statistics, Macquarie University

27 February 2020
Goal: categorical framework for modeling and analysis of systems

Analyse the composite system using the analyses of the particular system components and their specific wired interconnection.

▶ System architecture and behavior in single model!
Outline

1. A few categorical concepts
2. The category of wiring diagrams $\text{WD}$
3. Systems as $\text{WD}$-algebras
4. Further directions
A category $\mathcal{C}$ consists of

- a collection of objects $X, Y, Z, \ldots$
- a collection of morphisms $f : X \to Y$
- an identity morphism $1_X : X \to X$ for all $X \in \mathcal{C}$
- a composition rule $X \xrightarrow{f} Y \xrightarrow{g} Z := g \circ f$

such that $h \circ (g \circ f) = (h \circ g) \circ f$ and $1_Y \circ f = f = f \circ 1_X$

Examples

- **Set** the category of sets and functions (usual composition and ids)
- Numerous mathematical structures: $\text{Mon}$, $\text{Grp}$, $\text{Rng}$, $\text{Vect}_k$, $\text{Mod}_R$, $\text{Mat}$, $\text{Top}$, $\text{Man}$, $\text{Hilb}$, $\text{Aff}$, ...
Examples

- **Graph** the category of (directed, multi) graphs and homomorphisms

- **Petri** the category of Petri nets and homomorphisms

- **DDS** the category of discrete dynamical systems (Moore machines)

$S = \{s_1, s_2\}, S \times B \xrightarrow{\text{upd}} S, S \xrightarrow{\text{rdt}} B$

is the NOT machine
Functors

A functor $F : C \to D$ between two categories consists of

- a mapping $X \mapsto FX$ on objects
- a mapping $(f : X \to Y) \mapsto (Ff : FX \to FY)$ on morphisms

such that $F(g \circ f) = Fg \circ Ff$ and $F(1_X) = 1_{FX}$

Examples

- The ‘forgetful’ functor $U : \text{Grp} \to \text{Set}$ discards the group (& group homomorphism) structure, keeping the underlying set (& function)
- $\text{List} : \text{Set} \to \text{Set}$ maps set $S$ to $\text{List}(S)=\{(x_1, \ldots, x_n)|n \in \mathbb{N}, x_i \in S\}$ & function $f : S \to T$ to $\text{List}(f)$ by $(x_1, \ldots, x_n) \mapsto (f(x_1), \ldots, f(x_n))$
- Given a ‘theory’ $\mathcal{T}$, a model or algebra is a functor $A : \mathcal{T} \to \text{Set}$ that materializes abstract $n$-ary operations (groups, rings..)

Categories and functors themselves form a category $\text{Cat}$
A monoidal category $\mathcal{V}$ has a unit $I$ and a tensor product functor

\[ \otimes : \mathcal{V} \times \mathcal{V} \rightarrow \mathcal{V} \]

\[ (X, Y) \mapsto X \otimes Y \]

with $X \otimes (Y \otimes Z) \cong (X \otimes Y) \otimes Z$ and $I \otimes X \cong X \cong X \otimes I$.

Examples

- **Set** with the cartesian product $X \times Y = \{(x, y) | x \in X, y \in Y\}$ and the singleton set $\{\ast\}$ and similarly $(\textbf{Cat}, \times, 1)$
- $(\textbf{Vect}_k, \otimes, k)$ with the tensor product of vector spaces

A monoidal functor $F : \mathcal{V} \rightarrow \mathcal{W}$ comes with ‘comparison’ morphisms

\[ FX \otimes_{\mathcal{W}} FY \rightarrow F(X \otimes_{\mathcal{V}} Y) \text{ and } I_{\mathcal{W}} \rightarrow F(I_{\mathcal{V}}) \]

Monoidal functors $F : \mathcal{V} \rightarrow \textbf{Cat}$ are sometimes called $\mathcal{V}$-algebras.
The monoidal category of wiring diagrams

There is a category $\textbf{WD}$ where

- **objects** are pairs of sets $X = (X_{\text{in}}, X_{\text{out}})$

  ![Diagram of object](image)

  think of $X$ as a placeholder for systems, with input/output info values in $X_{\text{in}}, X_{\text{out}}$

- **morphisms** are functions $(X_{\text{out}} \times Y_{\text{in}} \xrightarrow{\phi_{\text{in}}} X_{\text{in}}, X_{\text{out}} \xrightarrow{\phi_{\text{out}}} Y_{\text{out}})$

  ![Diagram of morphism](image)

  think of $\phi_{\text{in/out}}$ expressing the flow of info through the ports

- **tensor product** $X \otimes Y = (X_{\text{in}} \times Y_{\text{in}}, X_{\text{out}} \times Y_{\text{out}})$

  ![Diagram of tensor product](image)

  think of parallel execution of processes
Worked-out Example

Start with three boxes $\mathbb{R} \xrightarrow{X} \mathbb{R}$, $\mathbb{R} \xrightarrow{Y} \mathbb{R}$ and $\mathbb{R} \xrightarrow{Z} \mathbb{R}$ where all input and output data of possible processes are real numbers, interconnected as in

This is a morphism $X \otimes Y \otimes Z \to A$ in the category $\textbf{WD}$ expressed as

$$\begin{align*}
\phi_{\text{in}} & : (X \otimes Y \otimes Z)_{\text{out}} \times (X \otimes Y \otimes Z)_{\text{out}} \\ \phi_{\text{out}} & : (X \otimes Y \otimes Z)_{\text{out}} \\ A_{\text{in}} & : \mathbb{R} \times \mathbb{R} \times \mathbb{R} \\ \pi_{12456} & : \mathbb{R} \times \mathbb{R} \times \mathbb{R} \times \mathbb{R} \times \mathbb{R} \\
\end{align*}$$
Systems as algebras for the wiring diagram category

A WD-algebra, namely a monoidal functor

\[ F : \text{WD} \rightarrow \text{Cat} \]

\[ X = (X_{\text{in}}, X_{\text{out}}) \rightarrow FX \]

subsystems category

\[ Y = (Y_{\text{in}}, Y_{\text{out}}) \rightarrow FY \]

composite system functor

\[ \phi \]

\[ F\phi \]

gives \textit{semantics} to boxes, \textit{composite operation} to wiring diagrams and \textit{parallelizing operation} to subsystems via \[ FX \times FY \rightarrow F(X \otimes Y) \]
Case study: Algebra of Contracts

To each box \( X_{\text{in}} \xrightarrow{\bullet} X_{\text{out}} \) assign category of contracts, i.e. relations

\[
R \subseteq X_{\text{in}} \times X_{\text{out}}
\]

To each wiring diagram \( \bullet \) assign formula

that, given contracts on subsystems, produces contract on composite.

Algebra machinery is concrete, flexible and scalable

- composite contract is completely decided by \( R_X, R_Y, R_Z \) and wiring
- can easily replace some subcontract by any other
- computes regardless of number of boxes or ‘difficulty’ of contracts

\[
R_X = [4, 5] \times [4, 5], \quad R_Y = [8, 9] \times [8, 9], \\
R_Z = [3, 5] \times [7, 9] \times \mathbb{R} \times [0, 1] \text{ produce } \\
R_A = [4, 5] \times [8, 9] \times \mathbb{R} \times [0, 1]
\]
...and the story just began

The theory has been developing, using various categorical tools

- main algebra examples: discrete & continuous dynamical systems, subalgebras e.g. linear time invariant systems, safety contracts etc.
- algebra maps are also important: translation between semantics
- *abstract systems*: a very general, all-inclusive framework
- time is incorporated: discrete, continuous and hybrid versions

Move towards interdisciplinary settings

- analysis of Cyber-Physical Systems using this formalism
- connections to other areas e.g. biological and chemical systems
- transfer of perspective and results between WD, databases (*lenses*), logic (*Dialectica*) and machine learning (*learners*)
Thank you for your attention!