Rewriting structured cospans

Daniel Cicala

SYCO 4
22 May 2019

outline

part i. motivation

motivation

Systems abound.

natural sciences	chemical reactions
	ecological systems
	classical and quantum physical systems
social sciences	social networks
	WORLD3 model
engineering	power grid
	hardware and software networks
	logistics

motivation

The grand ambition is...

to create a general mathematical theory for *compositional* systems

motivation

How do we embark on creating a fully general mathematical theory of systems?

look to linguistics

motivation

Syntax vs. Semantics

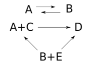
syntax	rules of grammar and sentence composition
semantics	meaning of words and sentences

motivation

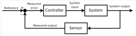
ingredients for syntax

- “alphabet” for systems
- rules for combining “letters” and “words”

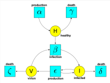
field-specific alphabet examples



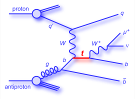
Chemical Reaction Network



Control Network



Petri Net



Feynman Diagram

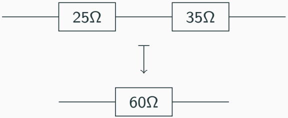
motivation

a toy example illustrating our goals.

We want to

- *connect systems together*

- *'rewrite' systems into equivalent systems*



motivation

- Our goal is to

  ... create syntax for compositional systems *(Baez, Courser)*
  ... onto these terms, introduce rewriting

- Compositional systems requires *composing* together systems to create new systems.

Make systems the arrows of a category!

- To rewrite systems, we borrow from the theory of adhesive categories or, more strictly, topos theory.

Make systems the objects of a topos!

motivation

	make systems arrows in a category
$+$	make systems objects in a topos
	use double categories

part ii. **structured cospans**

part iia. **structured cospans as arrows**

How to read a structured cospan:

$$\texttt{inputs} \rightarrow \texttt{system} \leftarrow \texttt{outputs}$$

This is a diagram in a category. How do we tame this data?

Given an adjunction

$$\mathsf{A} \underset{R}{\overset{L}{\rightleftarrows}} \mathsf{X} \quad (L \dashv R)$$

between topoi a **structured cospan** is a diagram in X of form

$$La \rightarrow x \leftarrow Lb$$

theorem. *(Baez, Courser)*

Given an adjunction

$$\mathsf{A} \underset{R}{\overset{L}{\rightleftarrows}} \mathsf{X} \quad (L \dashv R)$$

between topoi, there is a category ${}_L\mathsf{Csp}$ comprised of

objects	those of A
arrows	structured cospans $La \to x \leftarrow Lb$.

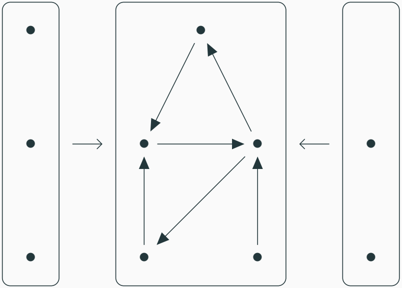
We fit open graphs into this framework using the adjunction

$$\mathsf{Set} \underset{R}{\overset{L}{\rightleftarrows}} \mathsf{Graph} \qquad (L \dashv R)$$

defined by

$$La := \text{ edgeless graph with node set } a$$

$$Rg := \text{ underlying set of nodes of } g$$



is of the form $La \to x \leftarrow Lb$ where

La is a three element set
Lb is a two element set

part iib. structured cospans as objects

The mechanisms of rewriting are designed for *objects* of a category.

definition.

Fix an adjunction

$$A \underset{R}{\overset{L}{\rightleftarrows}} X \quad (L \dashv R)$$

between topoi. The category ${}_L\mathsf{StrCsp}$ has

objects structured cospans $La \to x \leftarrow Lb$

arrows triples (f, g, h) fitting into commuting diagrams

$$\begin{array}{ccccc} La & \longrightarrow & x & \longleftarrow & Lb \\ {\scriptstyle Lf}\downarrow & & {\scriptstyle g}\downarrow & & \downarrow{\scriptstyle Lh} \\ La' & \rightarrow & x' & \leftarrow & Lb' \end{array}$$

The mechanisms for rewriting work for the objects of a *topos*.

theorem. (dc)

The category ${}_L\mathsf{StrCsp}$ is a topos.

part iii. rewriting

part iiia. double pushout rewriting

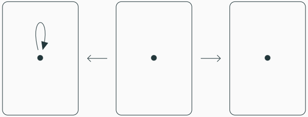
example.

Suppose we model the internet with graphs via

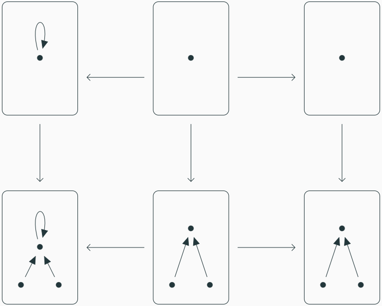
$$\begin{aligned} \text{nodes} :=& \quad \text{websites} \\ \text{edges} :=& \quad \text{links} \end{aligned}$$

but are uninterested in self-linking websites.

A rewrite rule that removes a loop is given by



A rewrite rule derived from this is



Double pushout rewriting was axiomatised using **adhesive categories**, of which topoi are an example.

definition.

- A **rewrite rule** is a span with monic legs in a topos:

$$\ell \leftarrowtail k \rightarrowtail r$$

- A **grammar** is a pair (X, P) with X a topos and P a set of rewrite rules in X.

definition.

- Given a grammar, a **derived rewrite rule** is one that appears at the bottom of a DPO diagram

$$\begin{array}{ccccc} \ell & \leftarrowtail & k & \rightarrowtail & r \\ \downarrow & \urcorner & \downarrow & \ulcorner & \downarrow \\ g & \leftarrowtail & d & \rightarrowtail & h \end{array}$$

  with the top row belonging to P.

- The **rewrite relation** on a grammar $g \rightsquigarrow^* h$ is the transitive and reflexive closure of the relation induced by the derived rewrite rules.

part iii. rewriting

part iiib. rewriting structured cospans

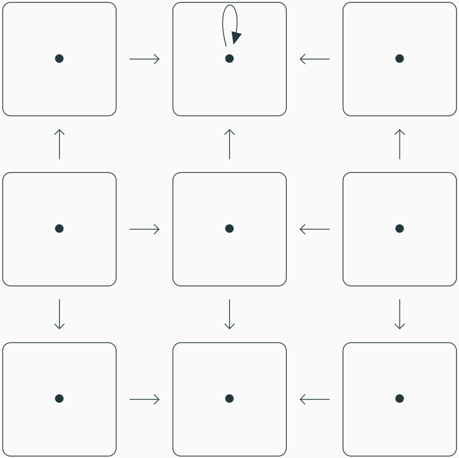
Because ${}_L\mathsf{StrCsp}$ is a topos, we can rewrite structured cospans.

A **rewrite rule of structured cospans** is a commuting diagram of form

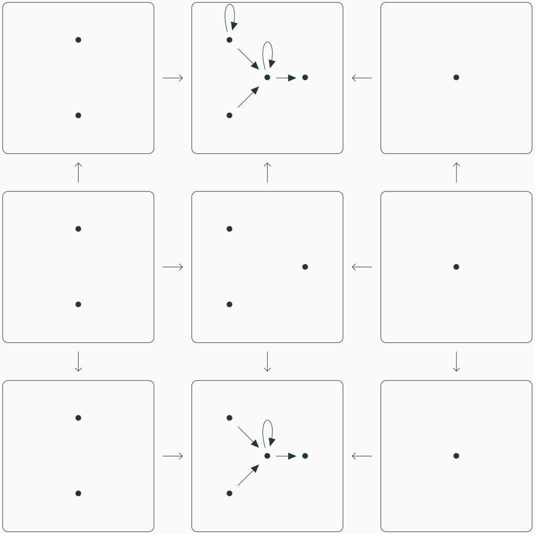
$$
\begin{array}{ccccc}
La & \longrightarrow & x & \longleftarrow & Lb \\
\cong\uparrow & & \uparrow & & \uparrow\cong \\
Lc & \longrightarrow & y & \longleftarrow & Ld \\
\cong\downarrow & & \downarrow & & \downarrow\cong \\
Le & \longrightarrow & z & \longleftarrow & Lf
\end{array}
$$

taken up to isomorphism.

Here is a rewrite rule of open graphs



Here is a derived rewrite rule of open graphs



theorem. (dc)

For any adjunction

$$\mathsf{A} \underset{R}{\overset{L}{\rightleftarrows}} \mathsf{X} \qquad (L \dashv R)$$

between topoi with L preserving pullbacks, there is a symmetric monoidal double category ${}_L\mathbb{R}\mathrm{ewrite}$ comprised of

objects	the objects of A
ver. arrows	isomorphisms in A
hor. arrows	structured cospans $La \to x \leftarrow Lb$
squares	rewrites of structured cospans $\begin{array}{ccccc} La & \to & x & \leftarrow & Lb \\ \cong\uparrow & & \uparrow & & \uparrow\cong \\ Lc & \to & y & \leftarrow & Ld \\ \cong\downarrow & & \downarrow & & \downarrow\cong \\ Le & \to & z & \leftarrow & Lf \end{array}$

part iv. inductive rewriting

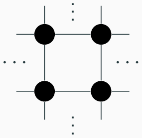
part iva. background

Given a closed system, we want to capture all of its rewritings.

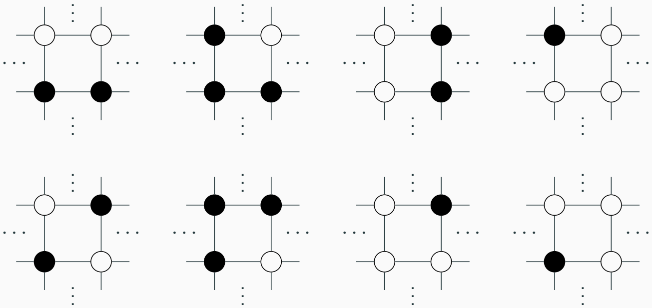
The previous section discussed *operational rewriting*, where the class of rewritings is obtained by *applying* rewrite rules.

Inductive rewriting builds this class from a set of basic rewritings.

- Decompose a closed system into "basic" open subsystems



- Rewrite basic open subsystems to generate all rewritings



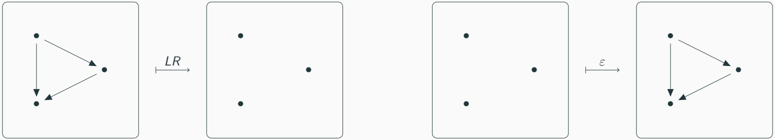
The basic open subsystems come from a grammar.

starting data.

- a grammar (X, P) for X a topos
- $L \dashv R \colon \mathsf{A} \rightleftarrows \mathsf{X}$ with monic counit & L pullback preserving

example.

$L \dashv R$: Set $\rightleftarrows$ Graph has a monic counit.



action counit

definition.

Given

- a grammar (X, P)
- $L \dashv R \colon \mathsf{A} \rightleftarrows \mathsf{X}$ with monic counit ε

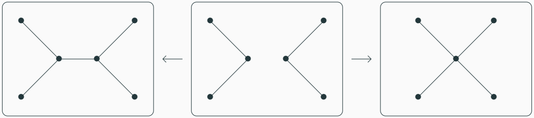
a **discrete grammar** (X, P_{LR}) has rewrite rules

$$\ell \leftarrowtail k \xleftarrow{\varepsilon} LRk \xrightarrow{\varepsilon} k \rightarrowtail r$$

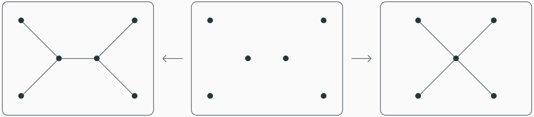
for each rewrite rule

$$\ell \leftarrowtail k \rightarrowtail r$$

If P has a rewriting rule



the associated rule in P_{LR} is



part iv inductive rewriting

part ivb. characterization results

theorem. (dc)

- (X, P) is a grammar
- $L \dashv R \colon \mathsf{X} \rightleftarrows \mathsf{A}$: R has a monic counit ε
- $\ell \leftarrow k \rightarrow r$ in P implies $\mathrm{Sub}(k)$ has all meets.

The rewriting relation for (X, P) and (X, P_{LR}) are equal.

**this generalizes a result in DPO graph rewriting by Ehrig, et. al.*

definition.

We can functorially assign a grammar $({}_L\mathsf{StrCsp}, P)$ to its *language*,

$$\mathrm{Lang}({}_L\mathsf{StrCsp}, P),$$

the double category comprised of

objects	objects from A
vert. arrows	invertible legged spans in A
hor. arrows	structured cospans
squares	generated by the rewrites derived from P

definition.

- (X, P) is a grammar.
- $(L \dashv R) \colon \mathsf{X} \rightleftarrows \mathsf{A}$ has a monic counit

Define $({}_L\mathsf{StrCsp}, \widehat{P}_{LR})$ to have rewrites

$$\begin{array}{ccccc} LR0 & \longrightarrow & \ell & \longleftarrow & LRk \\ {\scriptstyle\cong}\uparrow & & \uparrow & & \uparrow{\scriptstyle\cong} \\ LR0 & \longrightarrow & LRk & \longleftarrow & LRk \\ {\scriptstyle\cong}\downarrow & & \downarrow & & \downarrow{\scriptstyle\cong} \\ LR0 & \longrightarrow & r & \longleftarrow & LRk \end{array} \quad \text{and} \quad \begin{array}{ccccc} LRk & \longrightarrow & \ell & \longleftarrow & LR0 \\ {\scriptstyle\cong}\uparrow & & \uparrow & & \uparrow{\scriptstyle\cong} \\ LRk & \longrightarrow & LRk & \longleftarrow & LR0 \\ {\scriptstyle\cong}\downarrow & & \downarrow & & \downarrow{\scriptstyle\cong} \\ LRk & \longrightarrow & r & \longleftarrow & LR0 \end{array}$$

for each $\ell \leftarrowtail k \rightarrowtail r$ in P.

theorem. (dc)

- (X, P) is a grammar
- $(L \dashv R) \colon \mathsf{X} \rightleftarrows \mathsf{A}$ has monic counit
- $\ell \leftarrowtail k \rightarrowtail r$ in P implies $\mathrm{Sub}(k)$ has all meets
- $g,\ h \in \mathsf{X}$

$g \rightsquigarrow^{*} h$ if and only if $\mathrm{Lang}({}_L\mathsf{StrCsp}, \widehat{P}_{LR})$ has a square

$$\begin{array}{ccccc} LR0 & \longrightarrow & g & \longleftarrow & LR0 \\ \uparrow & & \uparrow & & \uparrow \\ LR0 & \longrightarrow & d & \longleftarrow & LR0 \\ \downarrow & & \downarrow & & \downarrow \\ LR0 & \longrightarrow & h & \longleftarrow & LR0 \end{array}$$

**this generalizes work by Gadducci and Heckel* the end