

Rewriting structured cospans

Daniel Cicala

SYCO 4

22 May 2019

outline

- part i.** motivation
- part ii.** structured cospans
- part iii.** rewriting structured cospans
- part iv.** inductive rewriting

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

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

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



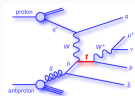
CHEMICAL REACTION
NETWORK



CONTROL NETWORK



PETRI NET



FEYNMAN DIAGRAM

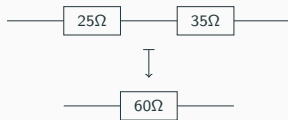
a toy example illustrating our goals.

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:

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

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

Given an adjunction

$$\begin{array}{ccc}
 & L & \\
 A & \begin{array}{c} \curvearrowright \\ \perp \\ \curvearrowleft \end{array} & X \\
 & R &
 \end{array}$$

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

$$La \rightarrow x \leftarrow Lb$$

theorem. (*Baez, Courser*)

Given an adjunction

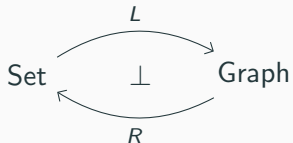
$$\begin{array}{ccc} & L & \\ A & \xrightarrow{\quad} & X \\ & \perp & \\ & R & \end{array}$$

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

objects those of A

arrows structured cospans $La \rightarrow x \leftarrow Lb$.

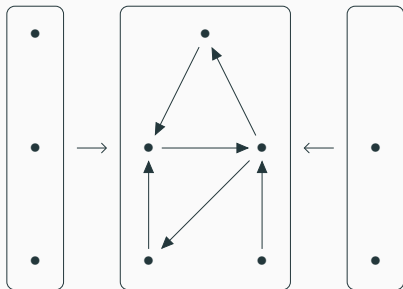
We fit open graphs into this framework using the adjunction



defined by

$La :=$ edgeless graph with node set a

$Rg :=$ underlying set of nodes of g



is of the form $La \rightarrow x \leftarrow Lb$ 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

$$\begin{array}{ccc}
 & L & \\
 A & \begin{array}{c} \curvearrowright \\ \perp \\ \curvearrowleft \end{array} & X \\
 & R &
 \end{array}$$

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

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

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

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

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

theorem. (dc)

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

part iii. rewriting

part iii.a. double pushout rewriting

example.

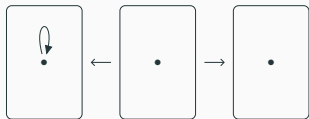
Suppose we model the internet with graphs via

nodes := websites

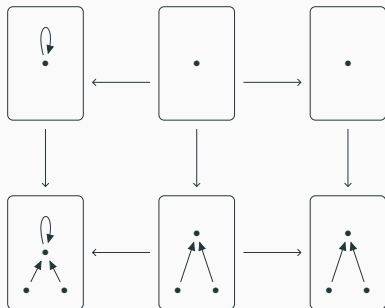
edges := links

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 axiomatized using **adhesive categories**, of which topoi are an example.

definition.

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

$$l \leftarrow k \rightarrow r$$

- 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

$$\begin{array}{ccccc}
 \ell & \leftarrow & k & \longrightarrow & r \\
 \downarrow & \sqcap & \downarrow & \sqcap & \downarrow \\
 g & \leftarrow & d & \longrightarrow & h
 \end{array}$$

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

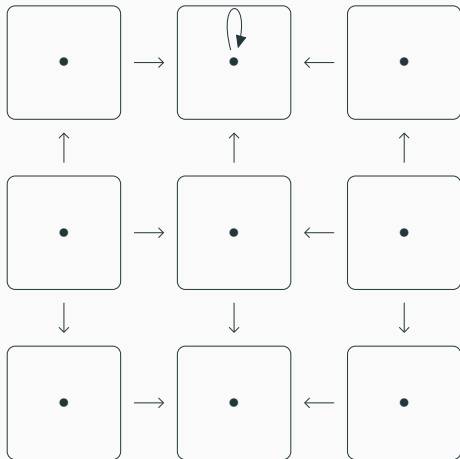
Because ${}_L\text{StrCsp}$ is a topos, we can rewrite structured cospans.

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

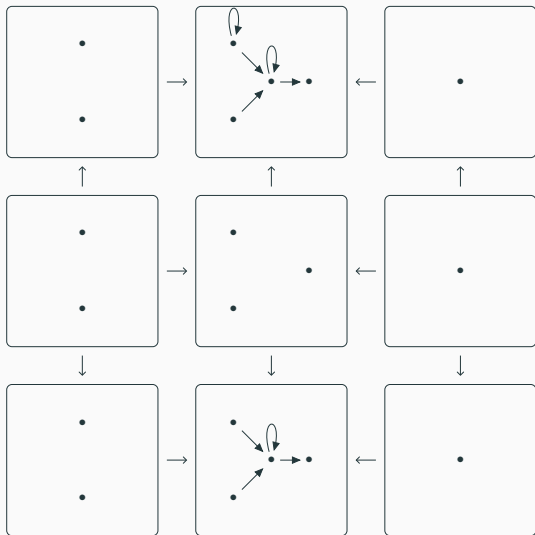
$$\begin{array}{ccccc} La & \longrightarrow & x & \longleftarrow & Lb \\ \uparrow \mathbb{R} & & \uparrow & & \uparrow \mathbb{R} \\ Lc & \longrightarrow & y & \longleftarrow & Ld \\ \downarrow \mathbb{R} & & \downarrow & & \downarrow \mathbb{R} \\ Le & \longrightarrow & z & \longleftarrow & Lf \end{array}$$

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

$$\begin{array}{ccc}
 & L & \\
 A & \begin{array}{c} \curvearrowright \\ \perp \\ \curvearrowleft \end{array} & X \\
 & R &
 \end{array}$$

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

- objects** the objects of A
- ver. arrows** isomorphisms in A
- hor. arrows** structured cospans $La \rightarrow x \leftarrow Lb$
- squares** rewrites of structured cospans

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

part iv. inductive rewriting

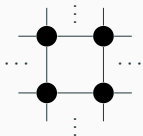
part iva. background

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

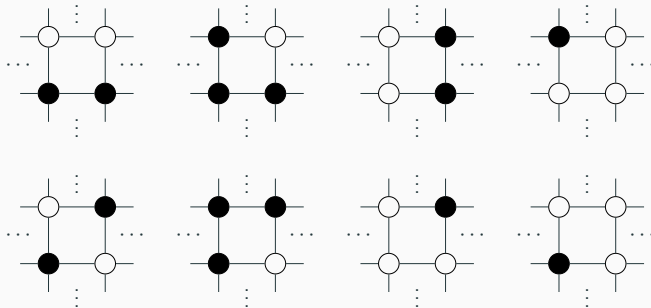
The previous section discussed *operational rewriting*, where the class of rewritings is obtained by applying rewrite rules.

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



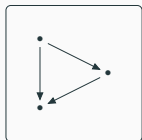
The basic open subsystems come from a grammar.

starting data.

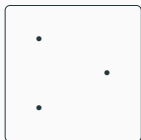
- _ a grammar (X, P) for X a topos
- _ $L \dashv R: A \rightleftarrows X$ with monic counit & L pullback preserving

example.

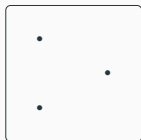
$L \dashv R$: Set \rightleftarrows Graph has a monic counit.



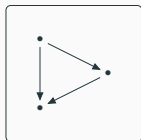
\xrightarrow{LR}



action



$\xrightarrow{\varepsilon}$



counit

definition.

Given

- _ a grammar (X, P)
- _ $L \dashv R: A \rightleftarrows X$ with monic counit ε

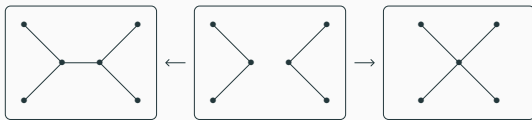
a **discrete grammar** (X, P_{LR}) has rewrite rules

$$l \leftarrow k \xleftarrow{\varepsilon} LRk \xrightarrow{\varepsilon} k \rightarrow r$$

for each rewrite rule

$$l \leftarrow k \rightarrow r$$

If P has a rewriting rule



the associated rule in P_{LR} is



part iv inductive rewriting

part ivb. characterisation results

theorem. (dc)

- (X, P) is a grammar
- $L \dashv R: X \rightleftharpoons A: R$ has a monic counit ε
- $\ell \leftarrow k \rightarrow r$ in P implies $\text{Sub}(k)$ 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 $({}_L\text{StrCsp}, P)$ to its *language*,

$$\text{Lang}({}_L\text{StrCsp}, P),$$

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
- $(L \dashv R): X \rightleftarrows A$ has a monic counit

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

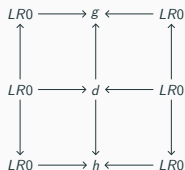
$$\begin{array}{ccccc}
 LR0 & \longrightarrow & \ell & \longleftarrow & LRk \\
 \uparrow \parallel R & & \uparrow & & \uparrow \parallel R \\
 LR0 & \longrightarrow & LRk & \longleftarrow & LRk \\
 \uparrow \parallel R & & \downarrow & & \downarrow \parallel R \\
 LR0 & \longrightarrow & r & \longleftarrow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

for each $\ell \leftarrow k \rightarrow r$ in P .

theorem. (dc)

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

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



**this generalizes work by Gadducci and Heckel*

the end