New Foundations for String Diagram Rewriting

Fabio Zanasi
University College London
Collaborators

Filippo Bonchi

Fabio Gadducci

Aleks Kissinger

Pawel Sobocinski
Props: algebras of network diagrams

A prop is (just) a symmetric monoidal category with set of objects \( \mathbb{N} \).

Props can be freely constructed starting from a signature \( \Sigma \) and equations \( E \).

**Prop of bialgebras**

**Prop of special Frobenius algebras**
Rewriting in a prop

Perspective of this work:
see $E$ as a **rewriting system** on diagrams

Our question
How to implement rewriting modulo symmetric monoidal structure in a simple, yet rigorous way?
Outline

1. Adequate interpretation
   - Rewriting modulo symmetric monoidal structure
   - Rewriting modulo SM + Frobenius structure
   - Complete but unsound
   - Sound & Complete

2. Decidability of confluence
   - Double pushout (DPO) rewriting of hypergraphs with interfaces
   - Convex DPO rewriting

Diagram $\rightarrow$ (Some sort of) graph
Hypergraph interpretation

prop \( \text{Syn}(\Sigma) \) of syntax freely generated by 
\[ \Sigma = \{ \ o_1, \ o_2, \ o_3 \} \]

Operations in \( \Sigma \) \sim Hyperedges
L/R boundary \sim Cospan structure

Proposition \( \langle \cdot \rangle : \text{Syn}(\Sigma) \rightarrow \text{Csp}(\text{Hyp}(\Sigma)) \) is faithful.
DPO rewriting with interfaces

$\text{Hyp}(\Sigma)$ is an adhesive category (Lack & Sobocinski) and thus adapted to DPO rewriting.
Hyp(\Sigma) is an ahdesive category (Lack & Sobocinski) and thus adapted to double-pushout rewriting.
DPO rewriting can be unsound

Rewriting in $Hyp(\Sigma)$ is complete but generally not sound

\[
\text{(R)} \quad o_1 \quad o_2 \quad \Rightarrow \quad \begin{array}{c}
\text{Syn}(\Sigma) \\
\langle \cdot \rangle
\end{array}
\]

\[
\begin{array}{c}
\text{Hyp}(\Sigma) \\
\langle \cdot \rangle
\end{array}
\quad \quad
\begin{array}{c}
\text{Syn}(\Sigma) \\
\langle \cdot \rangle
\end{array}
\]
DPO rewriting can be unsound

Rewriting in $Hyp(\Sigma)$ is complete but generally not sound

Equations of Special Frobenius Algebras
Frobenius makes DPO rewriting sound

Theorem I
DPO rewriting with interfaces is sound and complete for symmetric monoidal categories with a chosen special Frobenius structure.
Frobenius makes DPO rewriting sound

(R) \[ o_1 \xrightarrow{\text{Frob}} o_2 \]

\[ \Rightarrow \]

\[ \frac{o_4}{o_4} \]

\[ \text{Syn}(\Sigma) \]

\[ \overset{\text{Frob}}{\sim} \]

\[ \text{SMC} \]

\[ \overset{\text{Frob}}{\sim} \]

\[ \text{Syn}(\Sigma) \]
Where we are, so far

1. Adequate interpretation

   Rewriting modulo symmetric monoidal structure
   Complete but unsound

   Rewriting modulo SM + Frobenius structure
   Sound & Complete

   Double push out (DPO) rewriting of hypergraphs with interfaces
   Convex DPO rewriting

2. Decidability of confluence
How does sound DPO rewriting look like?

Context shape is the key

Context	shape	is	the	key
How does sound DPO rewriting look like?

Leading Intuition

A rewriting step is sound iff the rewriting context has this shape

$\text{Hyp}(\Sigma)$
Back to the soundness counterexample

\( \text{Hyp}(\Sigma) \)
Back to the soundness counterexample

\[(R) \quad \约 \quad \Rightarrow \quad \text{Syn}(\Sigma)\]

Unsound context shape

\[\cong \quad \text{Sound context shape}\]
Convex DPO rewriting is sound

**Theorem I**
DPO rewriting with interfaces is *sound and complete* for symmetric monoidal categories with a chosen *special Frobenius structure*.

**Theorem II**
Convex DPO rewriting with interfaces is *sound and complete* for symmetric monoidal categories.
Where we are, so far

1. Adequate interpretation

- Rewriting modulo symmetric monoidal structure
- Rewriting modulo SM + Frobenius structure
- Complete but unsound

Double push out (DPO) rewriting of hypergraphs with interfaces

Sound & Complete

Convex DPO rewriting

2. Decidability of confluence
Confluence, abstractly

If $E$ is confluent & terminating
then $x \xrightarrow{E} y$ becomes decidable.
Decidability of Confluence

In term rewriting, confluence is **decidable** for terminating systems

All the critical pairs are joinable

The system is confluent

(Knuth-Bendix)

In DPO (hyper)graph rewriting, confluence is **undecidable** (Plump)

All the critical pairs are joinable...

... but the system is not confluent.
Interfaces to the Rescue

Theorem In DPO rewriting with interfaces, confluence is decidable.
Confluence is decidable

**Theorem I**
DPO rewriting with interfaces is *sound and complete* for symmetric monoidal categories with a chosen *special Frobenius structure*.

**Theorem II**
Convex DPO rewriting with interfaces is *sound and complete* for symmetric monoidal categories.
Confluence is decidable

**Theorem I bis**
DPO rewriting with interfaces is *sound and complete* for symmetric monoidal categories with a chosen *special Frobenius structure*.

Confluence is decidable for terminating rewriting systems on such categories.

**Theorem II**
Convex DPO rewriting with interfaces is *sound and complete* for symmetric monoidal categories.
Theorem I bis
DPO rewriting with interfaces is *sound and complete* for symmetric monoidal categories with a chosen *special Frobenius structure*.

Confluence is decidable for terminating rewriting systems on such categories.

Theorem II bis
Convex DPO rewriting with interfaces is *sound and complete* for symmetric monoidal categories.

Confluence is decidable for *connected* terminating rewriting systems on such categories.
Conclusions

**Adequacy**

Rewriting modulo symmetric monoidal structure

Rewriting modulo SM + Frobenius structure

Double push out (DPO) rewriting of hypergraphs with interfaces

Convex DPO rewriting

**Confluence**

Decidable for connected systems

Decidable

**Termination**

Commutativity does not terminate

Proposal: interpret commutative operators as nodes of a new sort

\[
\begin{align*}
\text{SMC} & \approx \ \begin{array}{c}
\overset{\text{in}}{1} \overset{\text{μ}}{0} \overset{\text{out}}{1}
\end{array} \Rightarrow_C \begin{array}{c}
\overset{\text{in}}{1} \overset{\text{μ}}{0} \overset{\text{out}}{1}
\end{array} \Rightarrow_C \begin{array}{c}
\overset{\text{in}}{1} \overset{\text{μ}}{0} \overset{\text{out}}{1}
\end{array} \approx \ \begin{array}{c}
\overset{\text{in}}{1} \overset{\text{μ}}{0} \overset{\text{out}}{1}
\end{array} \Rightarrow_C \cdots
\end{align*}
\]

Terminating term rewriting systems

Terminating DPO-with-interface systems

Confluence for ground objects: undecidable (Kapur et al.)

Confluence: decidable (Knuth-Bendix)

Decidable (Plump)

Decidable