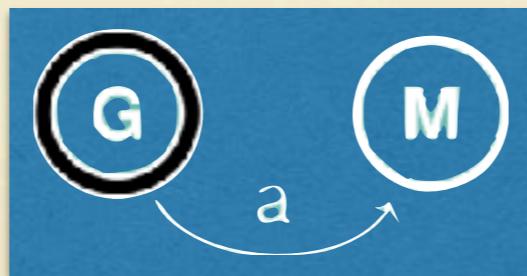


New Foundations for String Diagram Rewriting

Fabio Zanasi
University College London

Graphs as Models
April 23, 2017



Collaborators



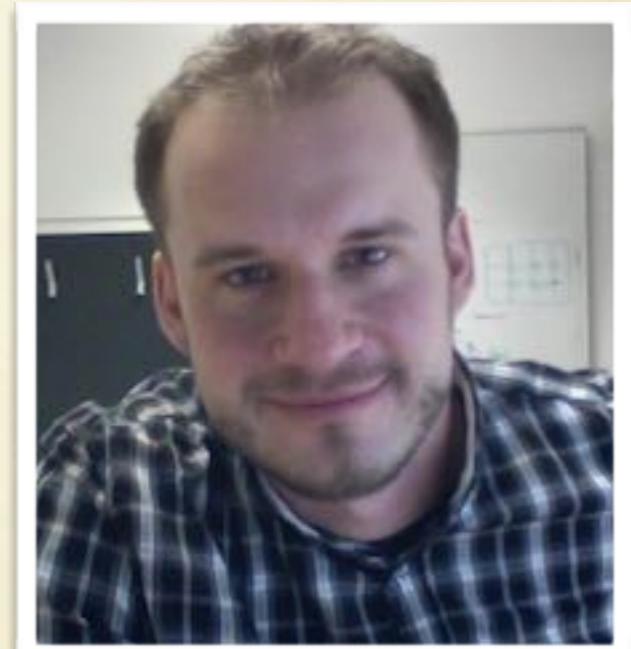
Filippo
Bonchi



Fabio
Gadducci

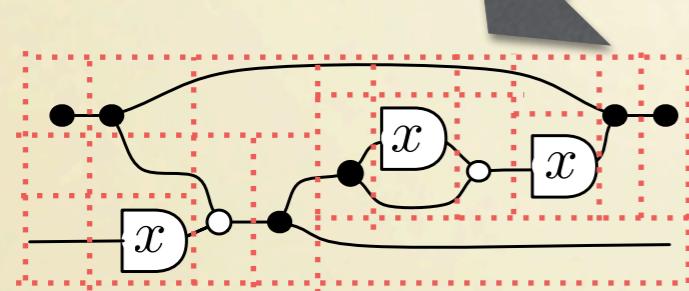
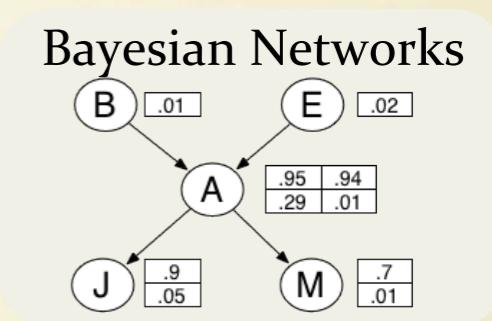
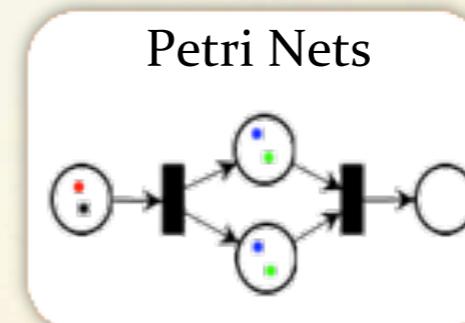
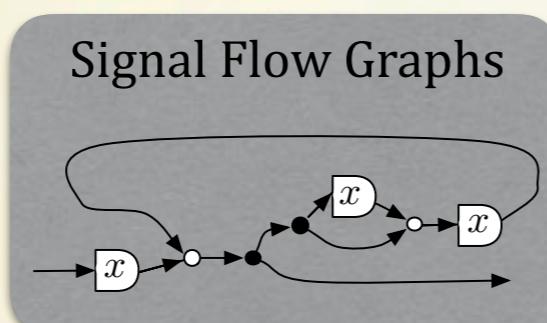
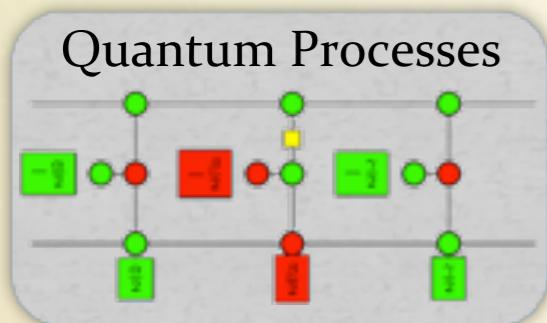


Aleks
Kissinger

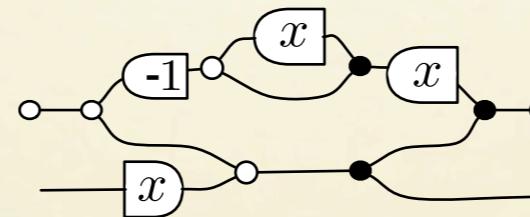


Pawel
Sobocinski

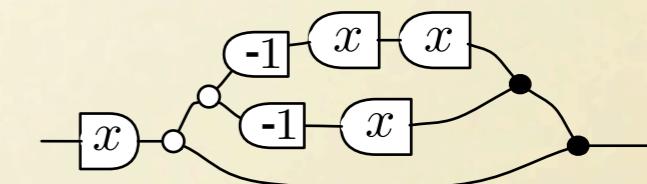
Algebras of network diagrams



\approx



\approx



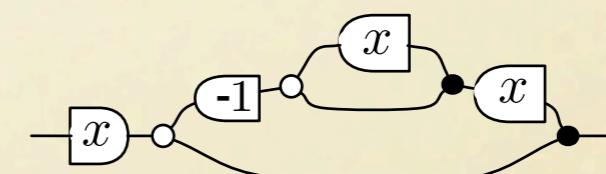
\approx

Fibonacci

$$\frac{x}{1 - x - x^2}$$

A diagram of the generating function for Fibonacci numbers: $x / (1 - x - x^2)$. The fraction is shown with x above the numerator and $1 - x - x^2$ above the denominator.

\approx

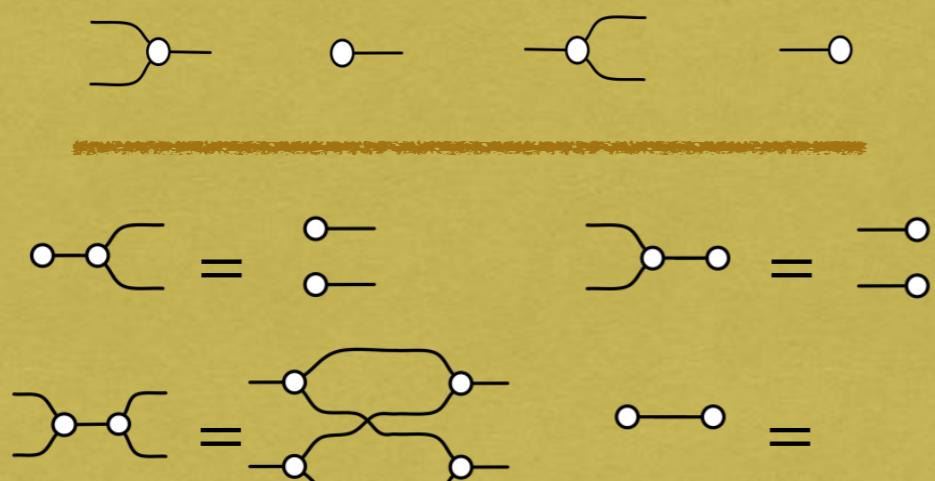


Props

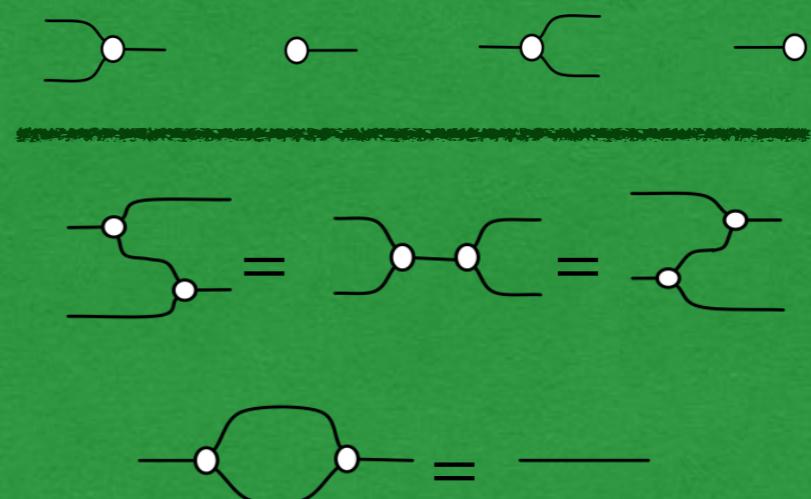
A prop is (just) a symmetric monoidal category with set of objects \mathbb{N}

Props can be freely constructed starting from a signature Σ and equations E

prop of bialgebras


$$\begin{array}{c} \text{Top row: } \text{Graph 1} \quad \text{Graph 2} \quad \text{Graph 3} \quad \text{Graph 4} \\ \hline \text{Bottom row: } \text{Graph 5} = \text{Graph 6} \quad \text{Graph 7} = \text{Graph 8} \\ \text{Graph 9} = \text{Graph 10} \quad \text{Graph 11} = \text{Graph 12} \end{array}$$

prop of special Frobenius algebras

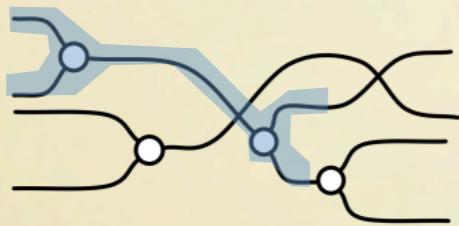

$$\begin{array}{c} \text{Top row: } \text{Graph 1} \quad \text{Graph 2} \quad \text{Graph 3} \quad \text{Graph 4} \\ \hline \text{Bottom row: } \text{Graph 5} = \text{Graph 6} = \text{Graph 7} \\ \text{Graph 8} = \text{Graph 9} \end{array}$$

Rewriting in a prop

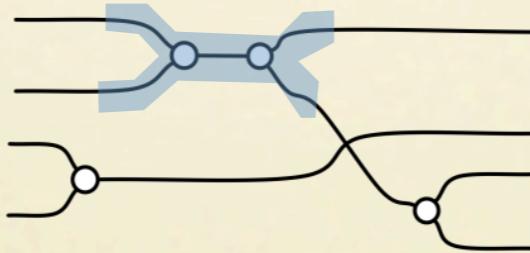
Perspective of this work:
see E as a **rewriting system** on diagrams

(R)

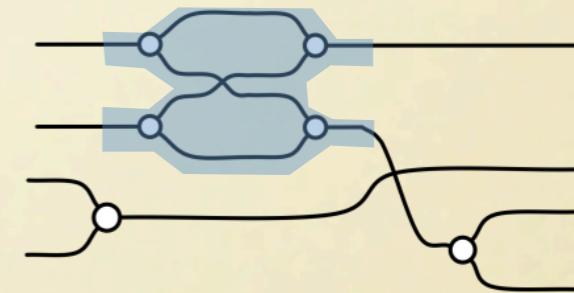
$$\begin{array}{c} \text{Diagram: } \text{Two nodes connected by a horizontal line, with vertical lines extending from each node.} \\ \Rightarrow \\ \text{Diagram: } \text{The same two nodes, but their vertical lines are now crossed.} \end{array}$$



SMC
 \approx



\Rightarrow_R



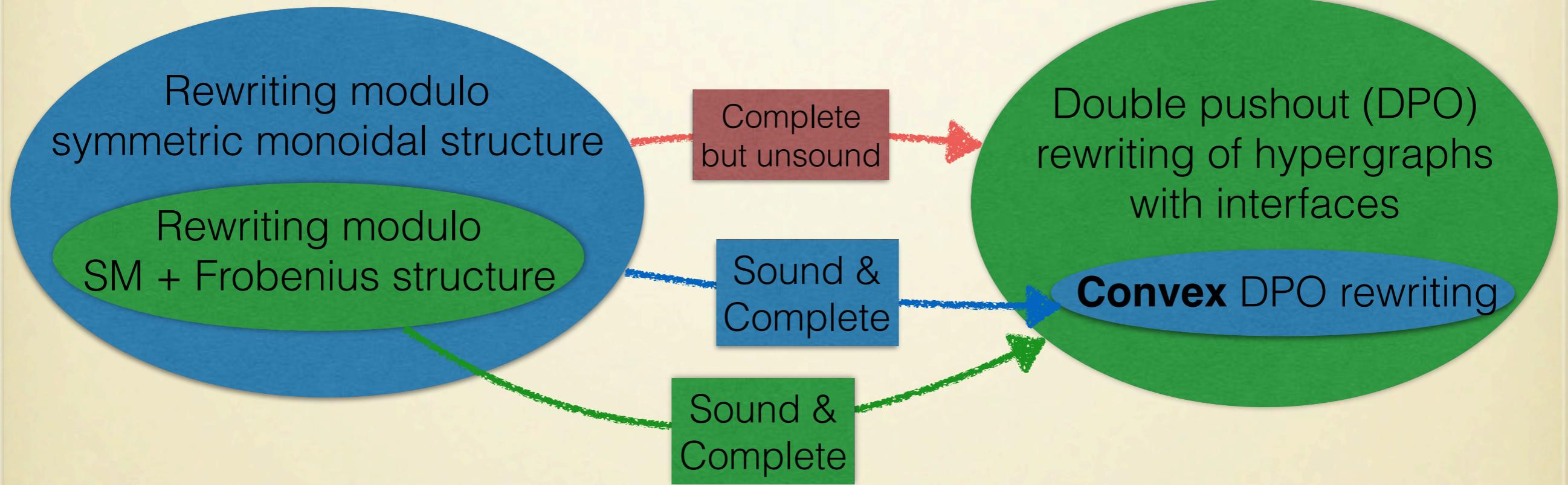
Our question

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

Outline

1. Adequate interpretation

Diagram \mapsto (Some sort of) graph

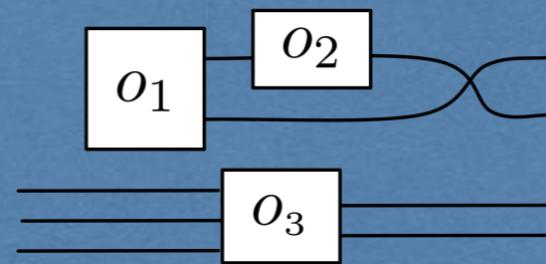


2. Decidability of confluence

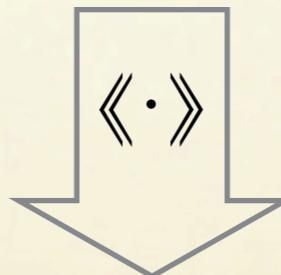
Hypergraph interpretation

prop **$Syn(\Sigma)$** of syntax
freely generated by

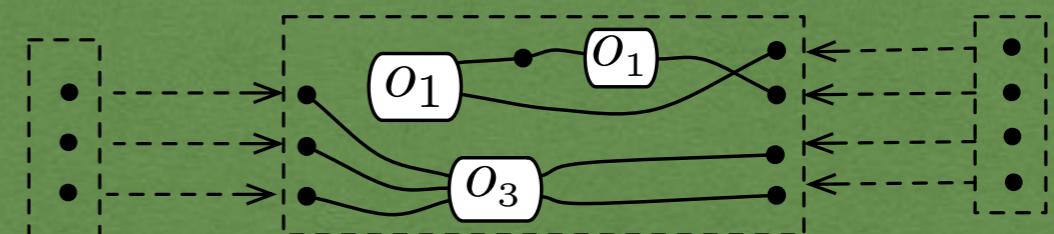
$$\Sigma = \{ [o_1], [o_2], [o_3] \}$$



Operations in Σ ~ Hyperedges
L/R boundary ~ Cospan structure



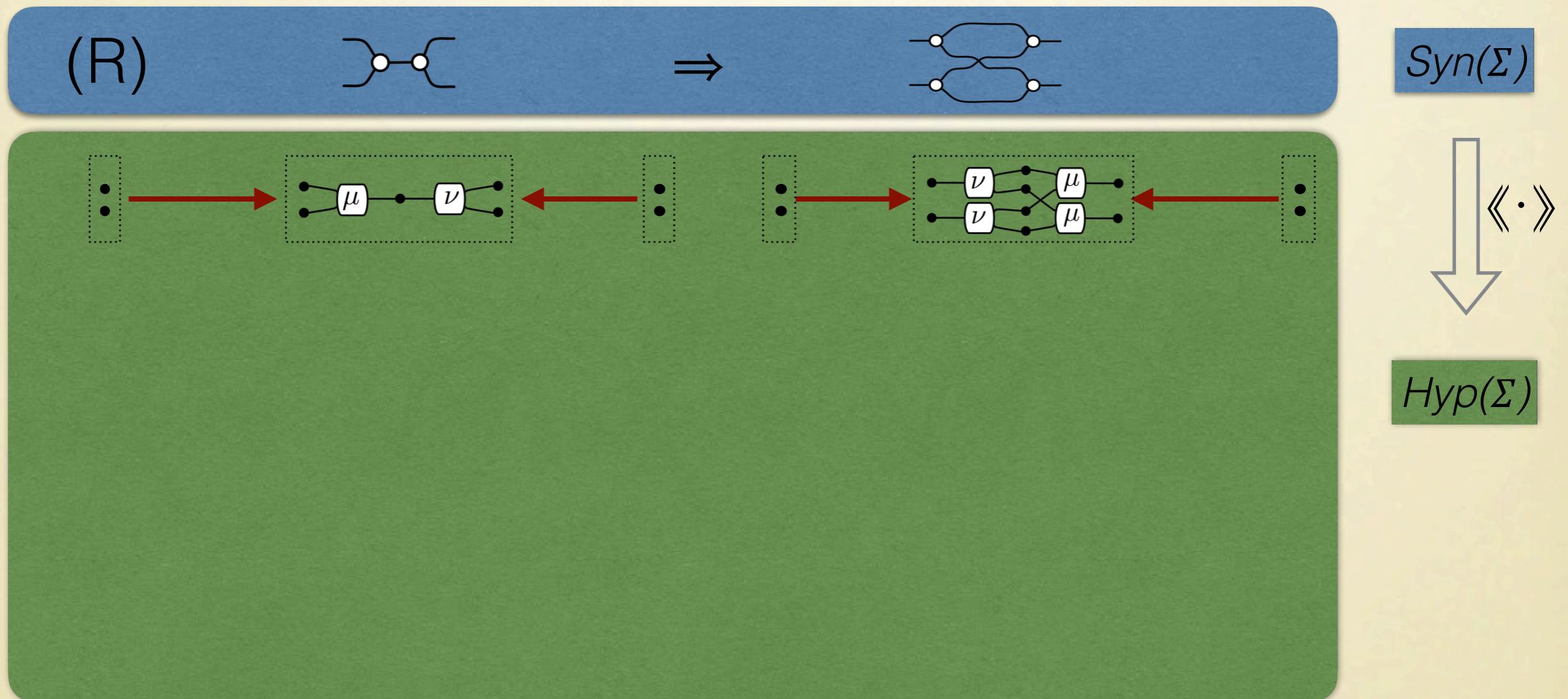
prop **$Csp(Hyp(\Sigma))$** of (discrete) cospans
 Σ -labelled hypergraphs



Proposition $\langle\cdot\rangle : Syn(\Sigma) \rightarrow Csp(Hyp(\Sigma))$ is faithful.

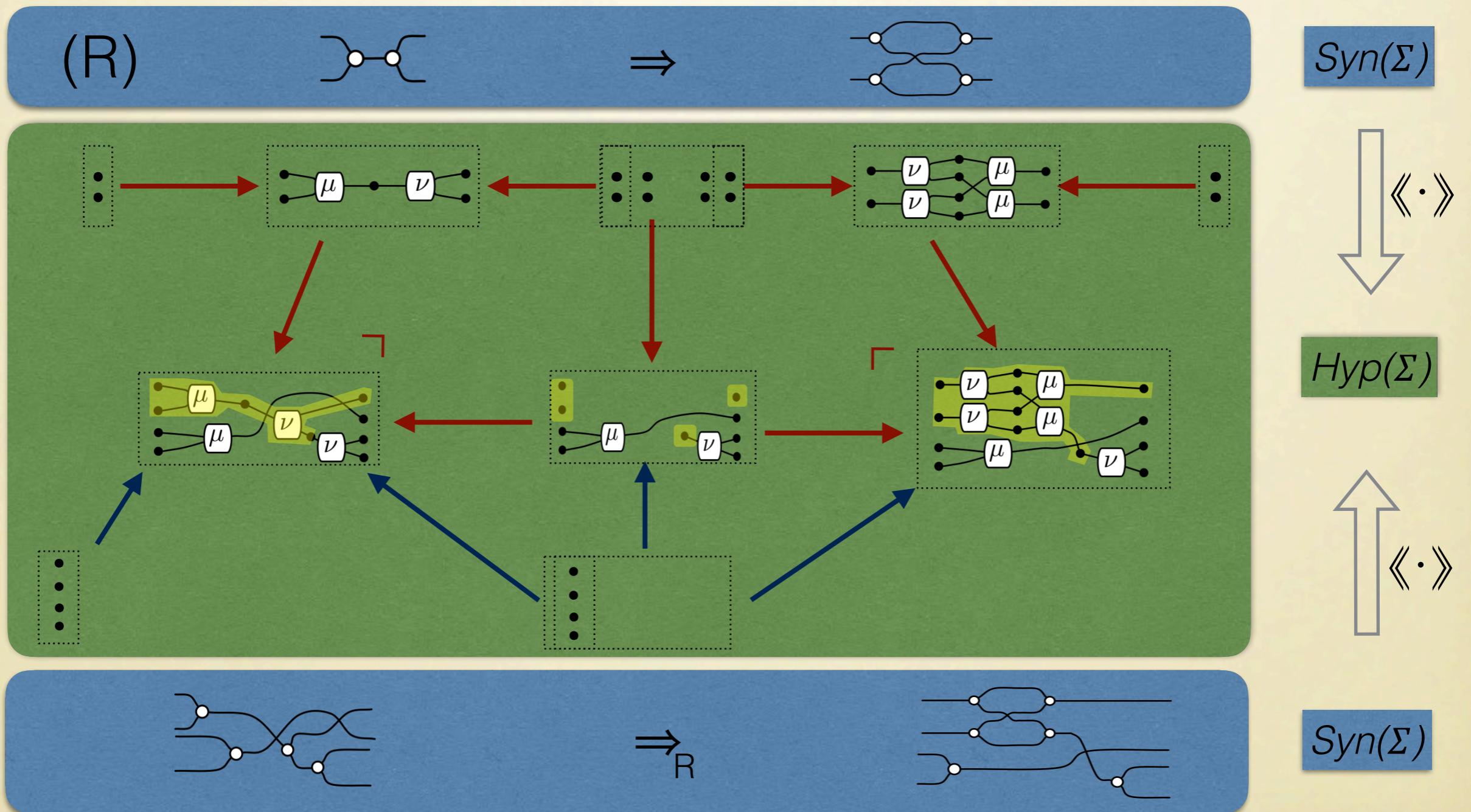
DPO rewriting with interfaces

$Hyp(\Sigma)$ is an *ahdesive category* (Lack & Sobociński)
and thus adapted to DPO rewriting.



DPO rewriting with interfaces

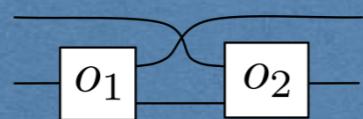
$Hyp(\Sigma)$ is an *ahdesive category* (Lack & Sobociński) and thus adapted to DPO rewriting.



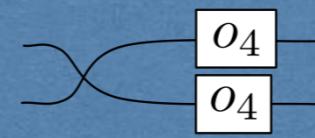
DPO rewriting can be unsound

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

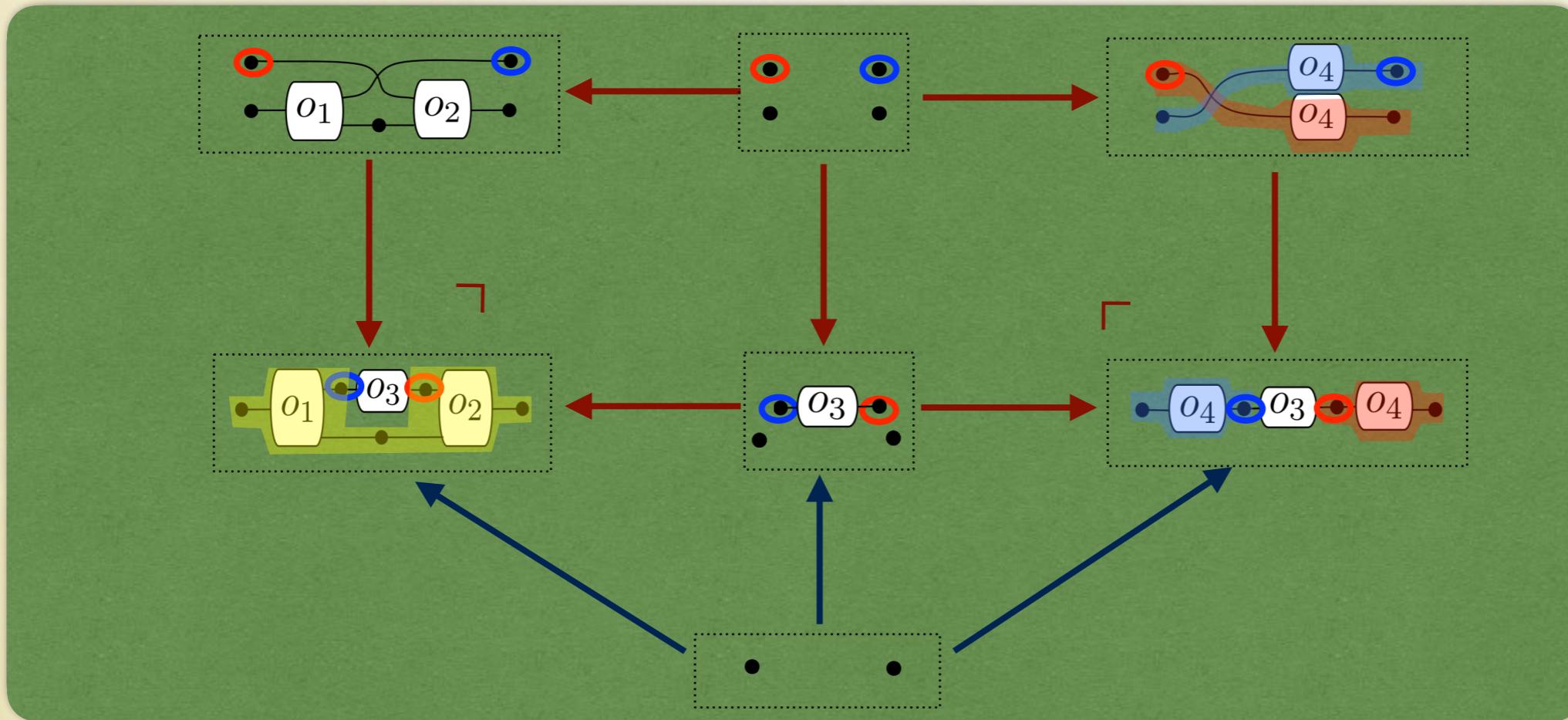
(R)



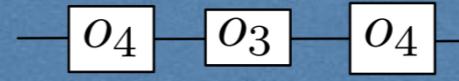
\Rightarrow



$Syn(\Sigma)$



\times

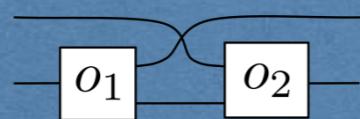


$Syn(\Sigma)$

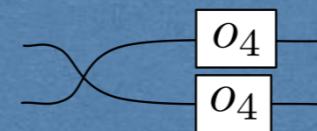
DPO rewriting can be unsound

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

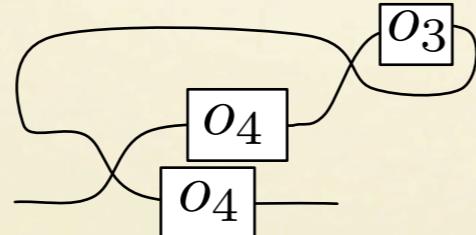
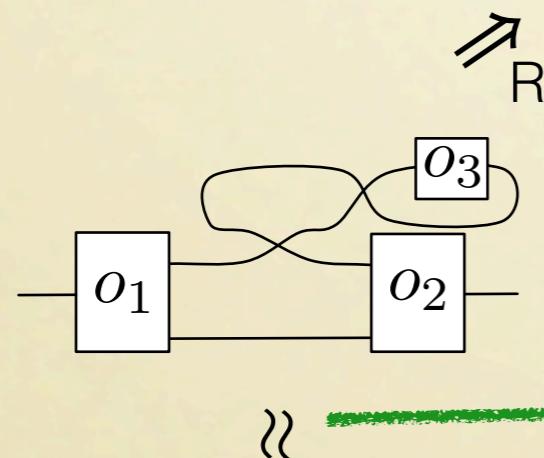
(R)



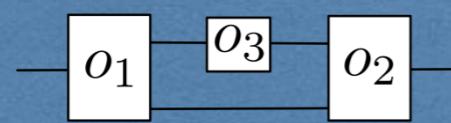
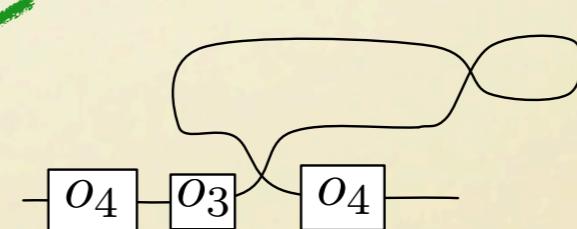
\Rightarrow



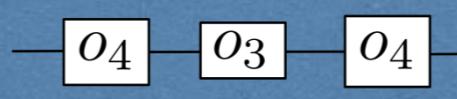
$Syn(\Sigma)$



Equations of
Special
Frobenius
Algebras



$\not\models_h$



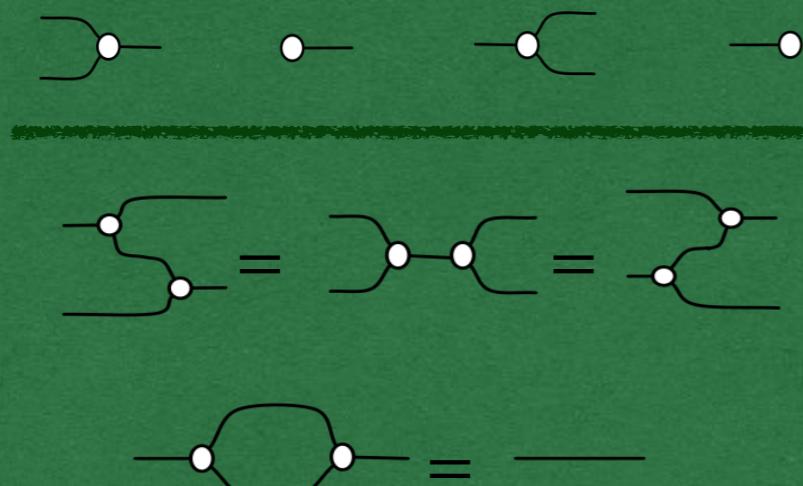
$Syn(\Sigma)$

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*.

prop of special Frobenius algebras

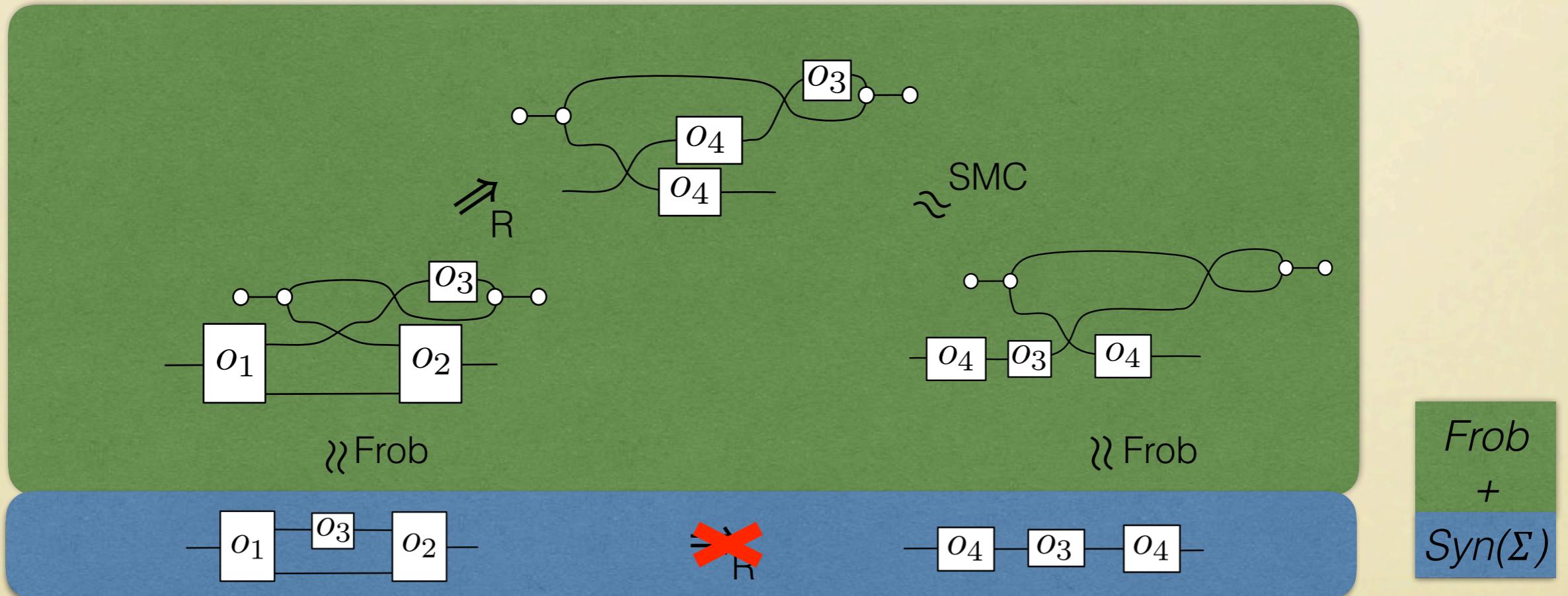


$$\begin{array}{c} \text{Syn}(\Sigma) + \text{Frob} \\ \xrightarrow{\quad} \\ \begin{array}{c} \text{Csp}(\text{Hyp}((\Sigma))) \\ \cong \end{array} \end{array}$$

The diagram illustrates the relationship between the category of synthesis ($\text{Syn}(\Sigma)$) and Frobenius structures (Frob) and the category of csp over hypographs ($\text{Csp}(\text{Hyp}((\Sigma)))$). The equivalence is shown through a series of morphisms (indicated by arrows) mapping from the left side to the right side.

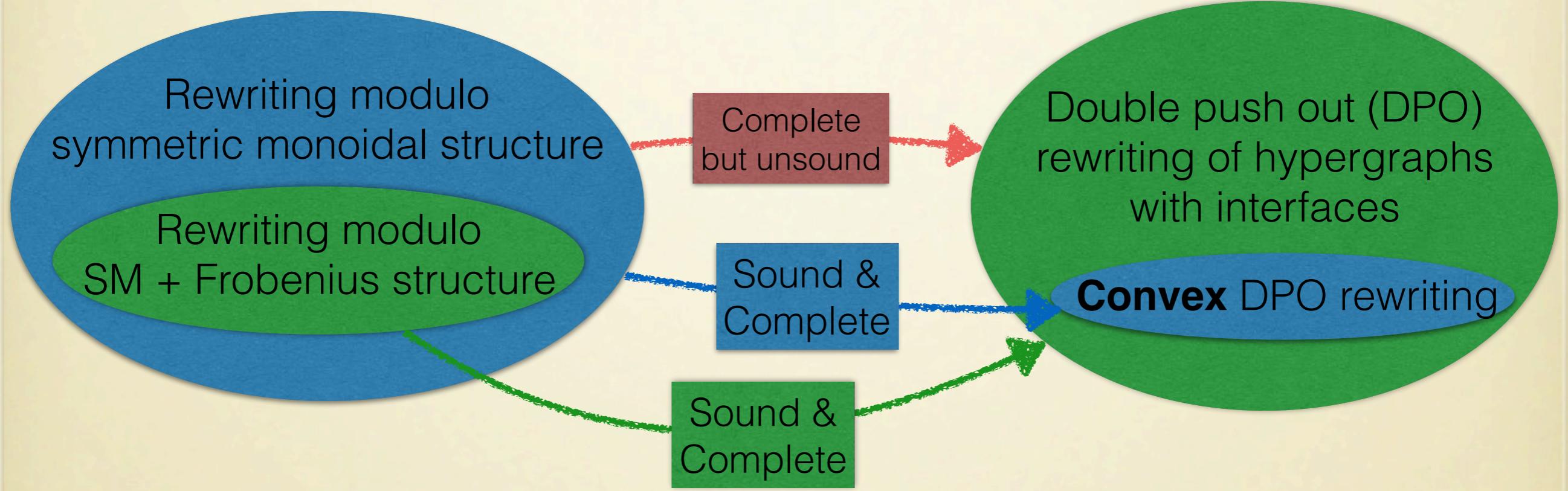
- The first arrow maps a single dot with two outgoing lines to a hypograph where a dot is connected to two dashed boxes, each containing a dot.
- The second arrow maps a dot with one outgoing line to a hypograph where a dot is connected to a dashed box containing a dot, which is further connected to another dashed box containing a dot.
- The third arrow maps a dot with two incoming lines to a hypograph where a dot is connected to two dashed boxes, each containing a dot, with arrows pointing towards the central dot.
- The fourth arrow maps a dot with one incoming line to a hypograph where a dot is connected to a dashed box containing a dot.
- The fifth arrow maps a dot with two outgoing lines followed by a dot with two incoming lines to a hypograph where a dot is connected to a dashed box containing a dot, which is connected to another dashed box containing a dot.
- The sixth arrow maps a dot with one outgoing line followed by a dot with one incoming line to a hypograph where a dot is connected to a dashed box containing a dot.
- The seventh arrow maps a dot with one outgoing line followed by a dot with one incoming line to a hypograph where a dot is connected to a dashed box containing a dot, which is connected to another dashed box containing a dot.
- The eighth arrow maps a dot with one outgoing line followed by a dot with one incoming line to a hypograph where a dot is connected to a dashed box containing a dot.
- The ninth arrow maps a dot with one outgoing line followed by a dot with one incoming line to a hypograph where a dot is connected to a dashed box containing a dot, which is connected to another dashed box containing a dot.
- The final arrow maps a dot with one outgoing line followed by a dot with one incoming line to a hypograph where a dot is connected to a dashed box containing a dot, which is connected to another dashed box containing a dot.

Frobenius makes DPO rewriting sound



Where we are, so far

1. Adequate interpretation



2. Decidability of confluence

How does sound DPO rewriting look like?

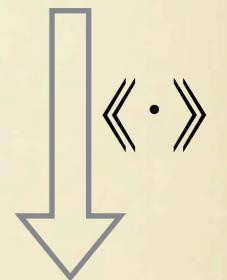
(R)

\boxed{l}

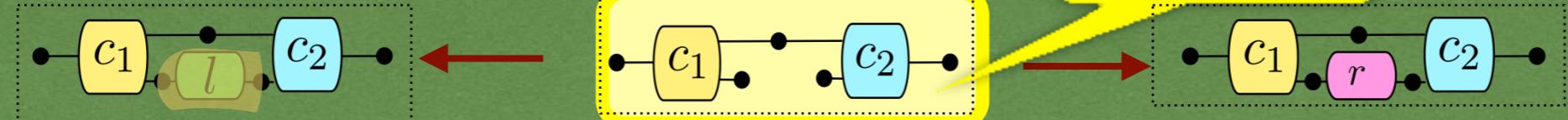
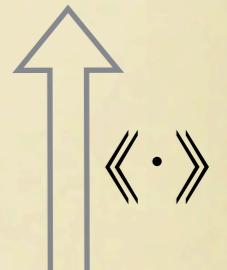
\Rightarrow

\boxed{r}

$Syn(\Sigma)$



$Hyp(\Sigma)$



\boxed{c}

\approx_{SMC}

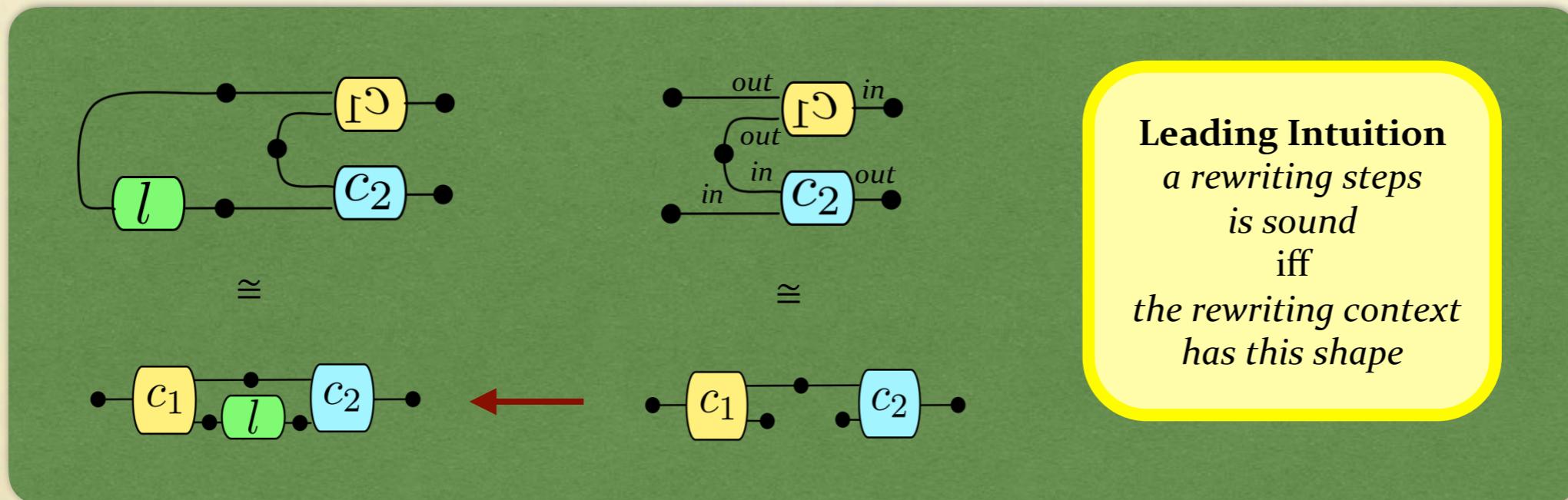
$c_1 \boxed{l} c_2$

\Rightarrow_R

$c_1 \boxed{r} c_2$

$Syn(\Sigma)$

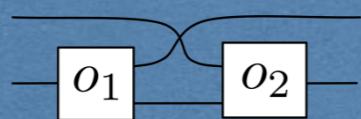
How does sound DPO rewriting look like?



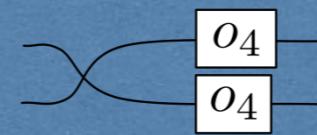
$Hyp(\Sigma)$

Back to the soundness counterexample

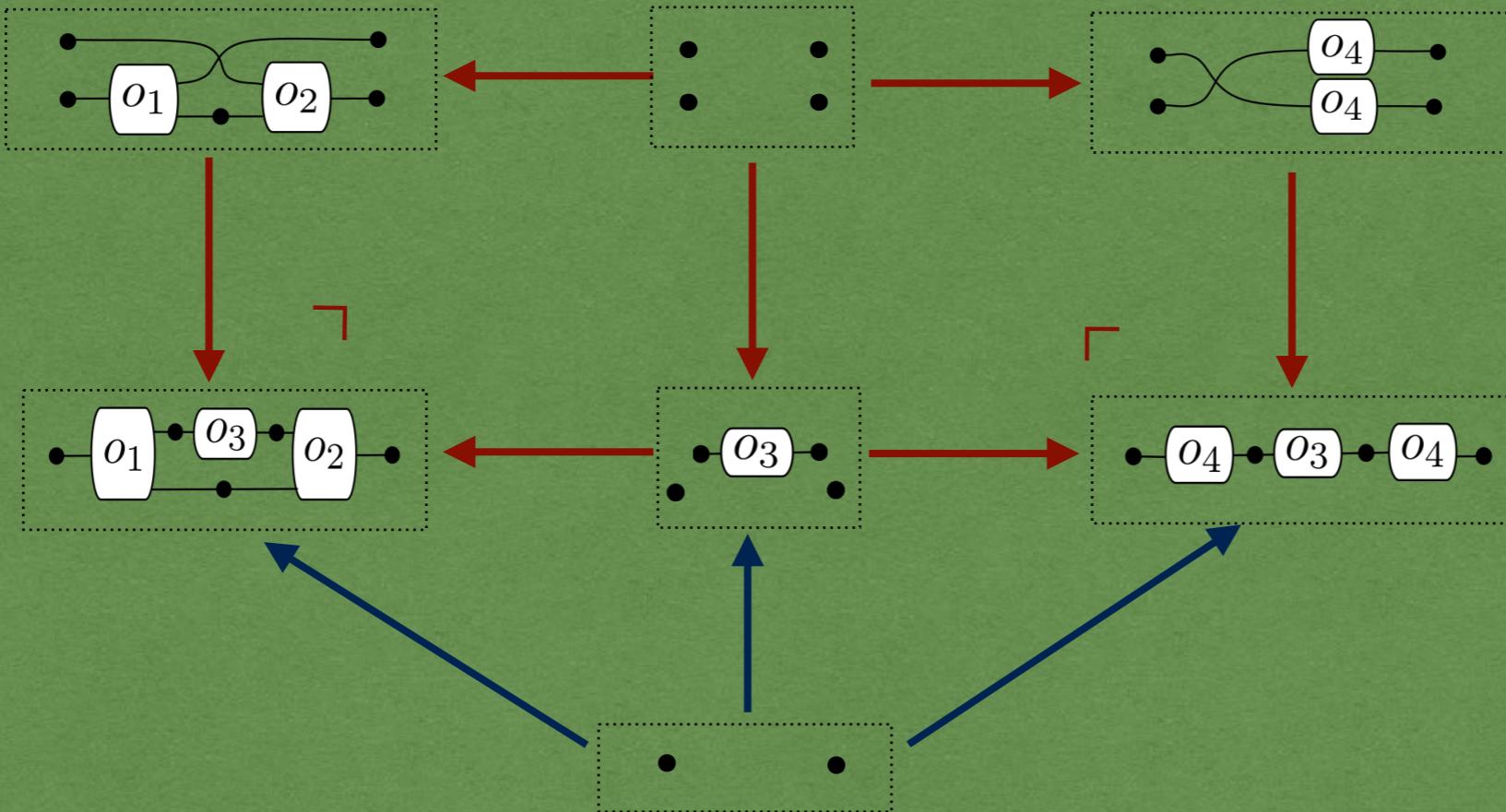
(R)



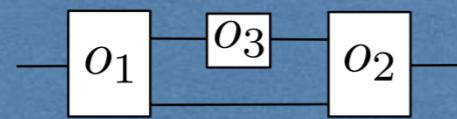
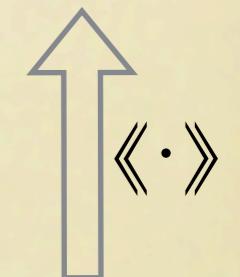
\Rightarrow



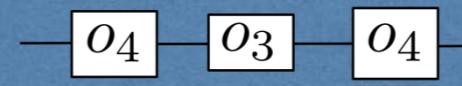
$Syn(\Sigma)$



$Hyp(\Sigma)$



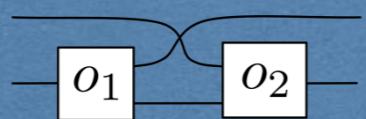
\Rightarrow_R



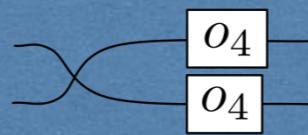
$Syn(\Sigma)$

Back to the soundness counterexample

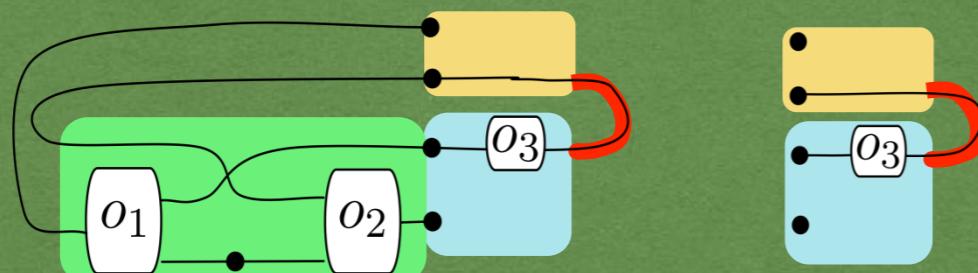
(R)



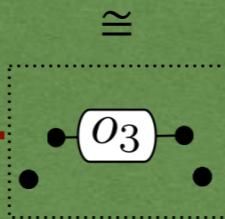
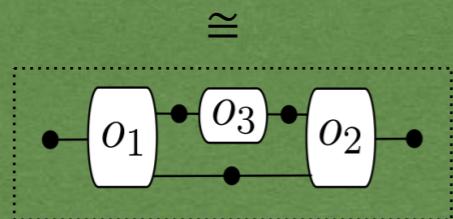
\Rightarrow



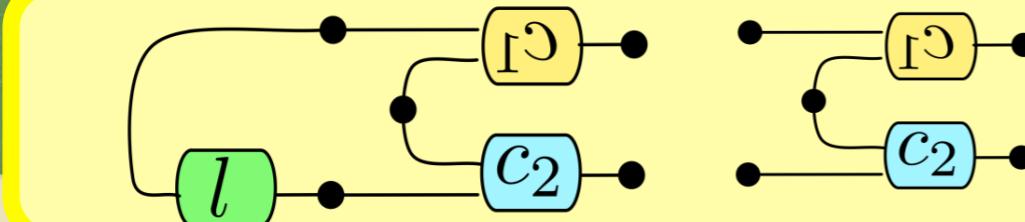
$Syn(\Sigma)$



Unsound context shape



$Hyp(\Sigma)$



Sound context shape

Convex DPO rewriting is sound

Theorem I

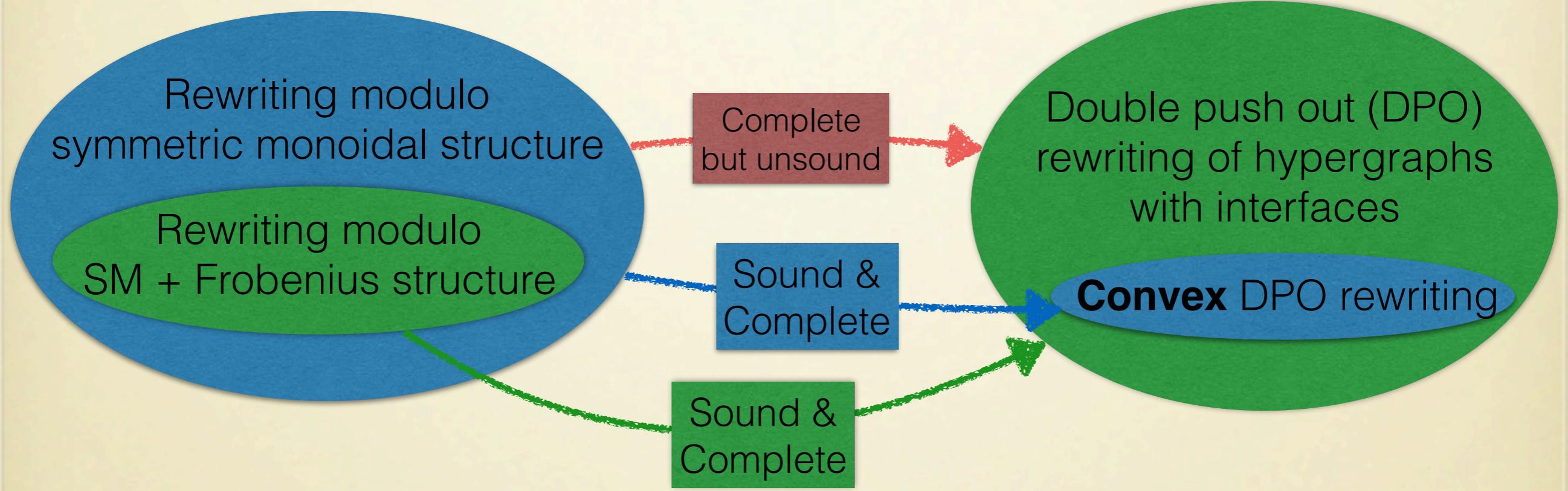
DPO rewriting with interfaces is *sound and complete* for symmetric monoidal categories with a chosen *special Frobenius structure*.

Theorem II

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

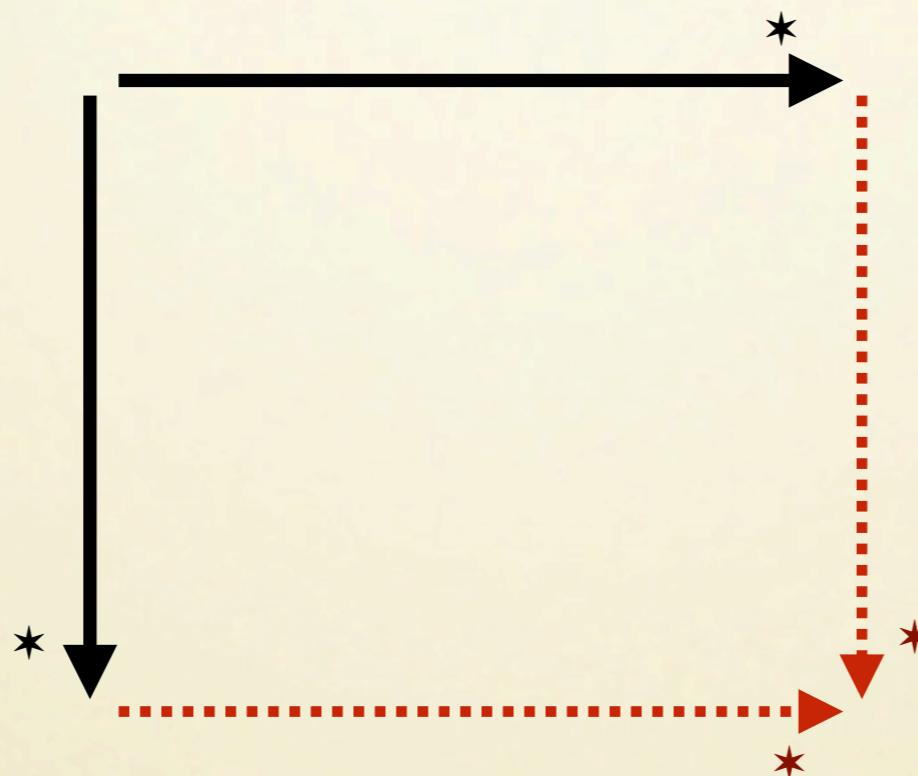
Where we are, so far

1. Adequate interpretation



2. Decidability of confluence

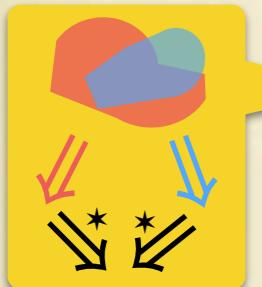
Confluence, abstractly



If E is confluent & terminating
then $x \stackrel{E}{=} y$ 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)

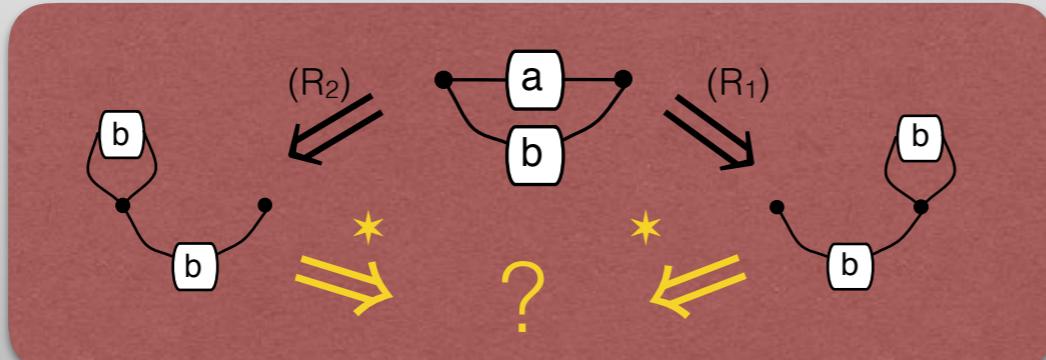
$$\begin{array}{c} \bullet \\[-1ex] \text{---} \\[-1ex] \boxed{a} \\[-1ex] \text{---} \\[-1ex] \bullet \end{array} \stackrel{0}{\quad} \Rightarrow \quad \bullet \quad \stackrel{1}{\quad} \quad \begin{array}{c} \bullet \\[-1ex] \text{---} \\[-1ex] \text{b} \\[-1ex] \text{---} \\[-1ex] \bullet \end{array} \quad (R_1)$$

$$\begin{array}{c} \bullet \\[-1ex] \text{---} \\[-1ex] \boxed{a} \\[-1ex] \text{---} \\[-1ex] \bullet \end{array} \stackrel{0}{\quad} \Rightarrow \quad \begin{array}{c} \bullet \\[-1ex] \text{---} \\[-1ex] \text{b} \\[-1ex] \text{---} \\[-1ex] \bullet \end{array} \quad \stackrel{1}{\quad} \quad \bullet \quad (R_2)$$

$$\begin{array}{c} \bullet \\[-1ex] \text{---} \\[-1ex] \text{b} \\[-1ex] \text{---} \\[-1ex] \bullet \end{array} \quad \stackrel{(R_2)}{\leftarrow} \quad \bullet \quad \stackrel{(R_1)}{\Rightarrow} \quad \begin{array}{c} \bullet \\[-1ex] \text{---} \\[-1ex] \text{b} \\[-1ex] \text{---} \\[-1ex] \bullet \end{array}$$

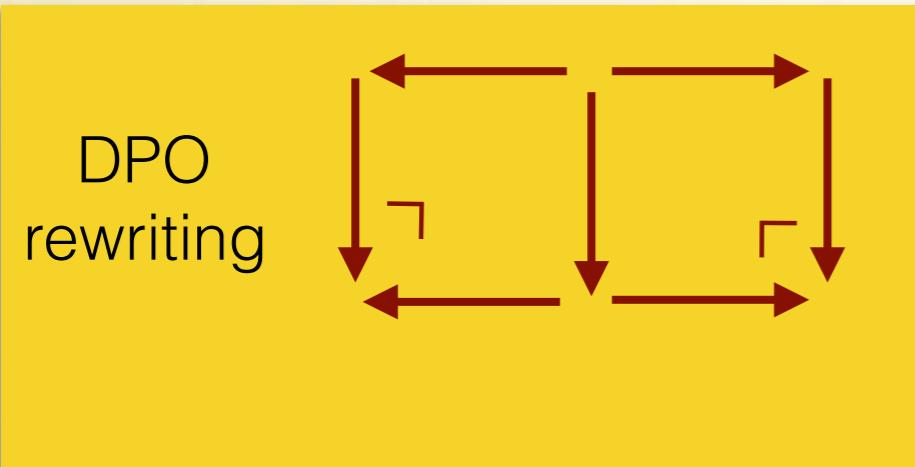
$$\begin{array}{c} \bullet \\[-1ex] \text{---} \\[-1ex] \text{b} \\[-1ex] \text{---} \\[-1ex] \bullet \end{array} \quad \stackrel{(R_1)}{\leftarrow} \quad \begin{array}{c} \bullet \\[-1ex] \text{---} \\[-1ex] \text{a} \\[-1ex] \text{---} \\[-1ex] \bullet \end{array} \quad \stackrel{(R_2)}{\Rightarrow} \quad \begin{array}{c} \bullet \\[-1ex] \text{---} \\[-1ex] \text{b} \\[-1ex] \text{---} \\[-1ex] \bullet \end{array}$$

All the critical pairs
are joinable...



... but the system
is not confluent.

Interfaces to the Rescue

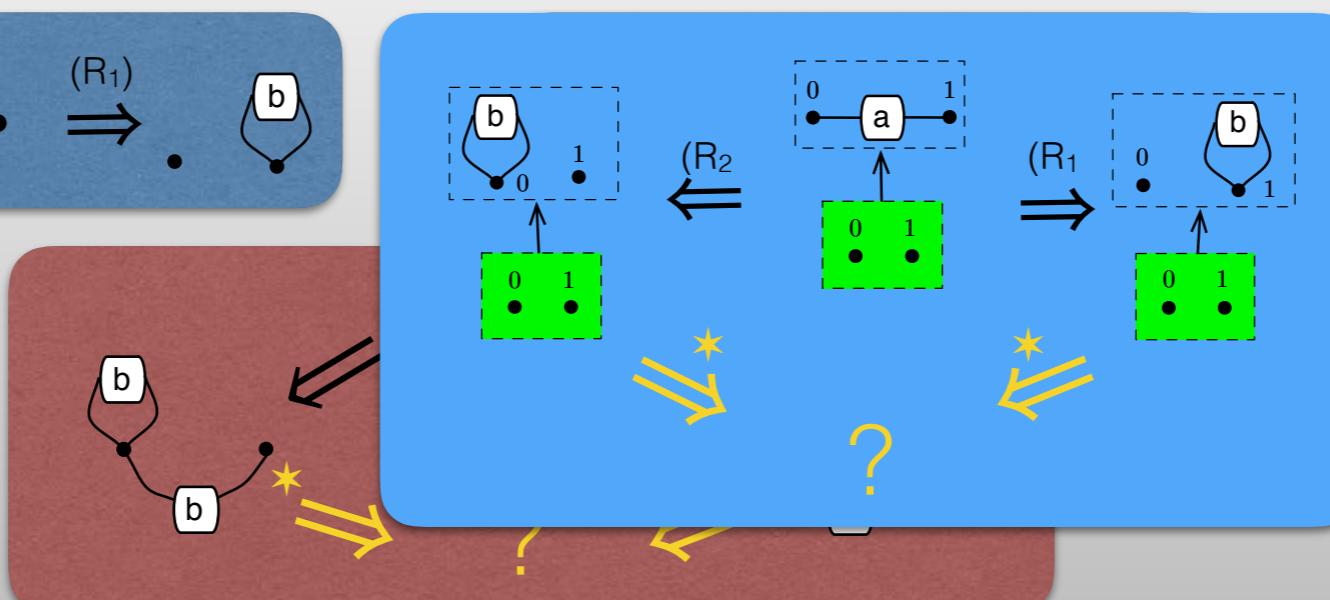
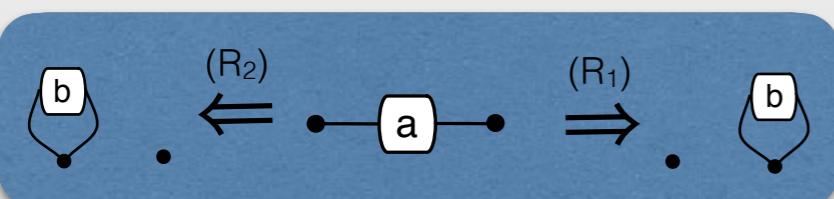


VS



$$\begin{array}{ccc} \text{0} & \text{---} & \text{1} \\ \bullet & \text{---} & \text{a} & \text{---} & \bullet \\ & \Rightarrow & & \bullet & \text{---} & \text{b} \\ & & & & \bullet & \text{---} & \text{1} \\ & & & & & \bullet & \\ & & & & & & (R_1) \end{array}$$

$$\begin{array}{ccc} \text{0} & \text{---} & \text{1} \\ \bullet & \text{---} & \text{a} & \text{---} & \bullet \\ & \Rightarrow & & \bullet & \text{---} & \text{b} \\ & & & & \bullet & \text{---} & \text{0} \\ & & & & & \bullet & \\ & & & & & & (R_2) \end{array}$$



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.

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.

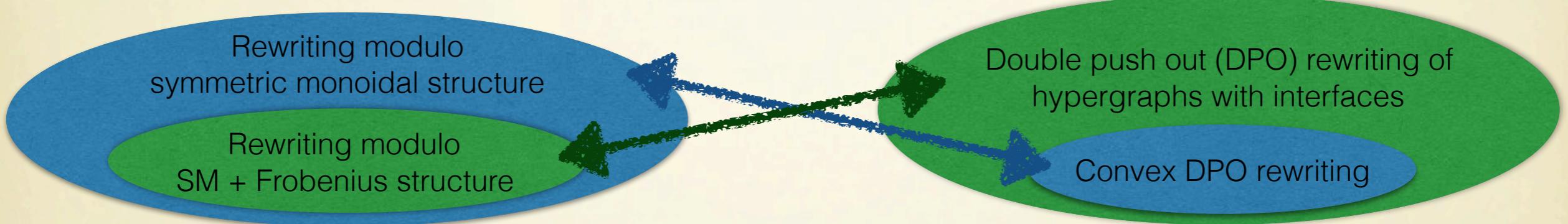
Theorem II bis

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

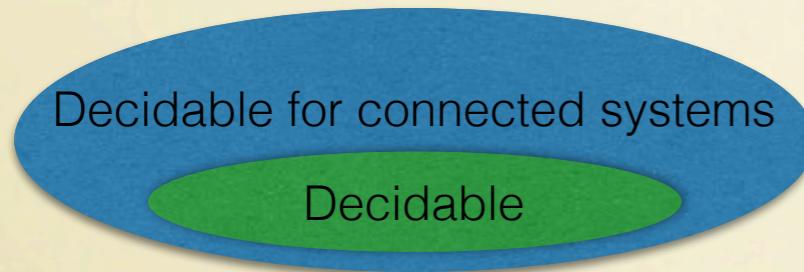
Confluence is decidable for *connected* terminating rewriting systems on such categories.

Conclusions

Adequacy



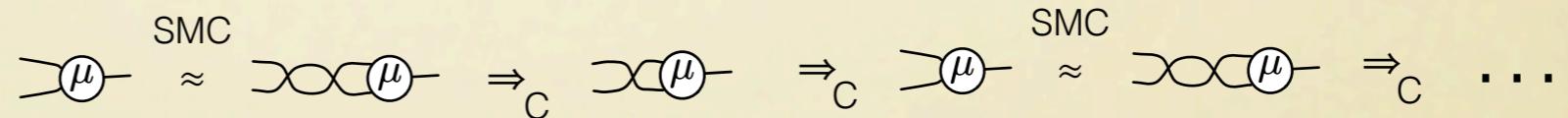
Confluence



	Terminating term rewriting systems	Terminating DPO-with-interface systems
Confluence for ground objects	<i>undecidable (Kapur et al.)</i>	<i>undecidable (Plump)</i>
Confluence	<i>decidable (Knuth-Bendix)</i>	decidable

Termination

Commutativity does not terminate



Proposal: interpret commutative operators as nodes of a new sort

