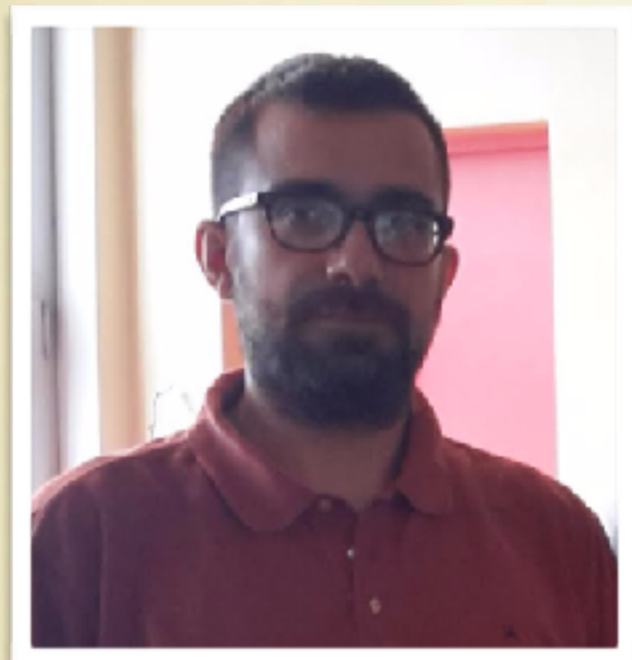
New Foundations for String Diagram Rewriting

Fabio Zanasi
University College London

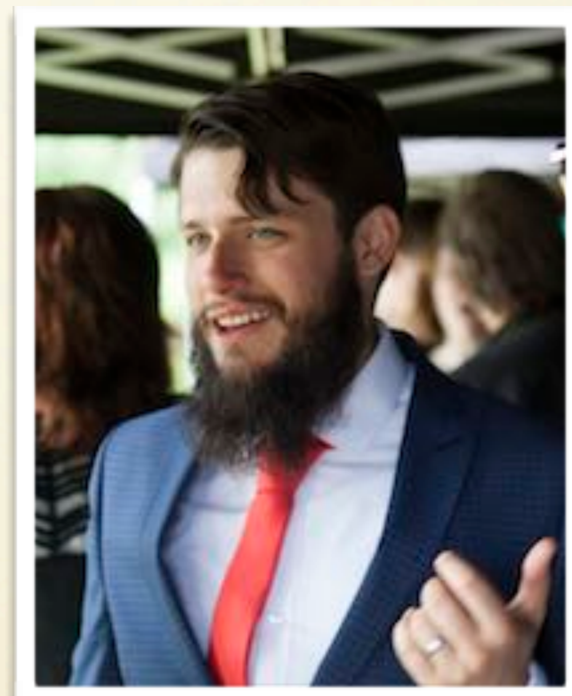
Graphs as Models
April 23, 2017

Collaborators



Filippo Bonchi

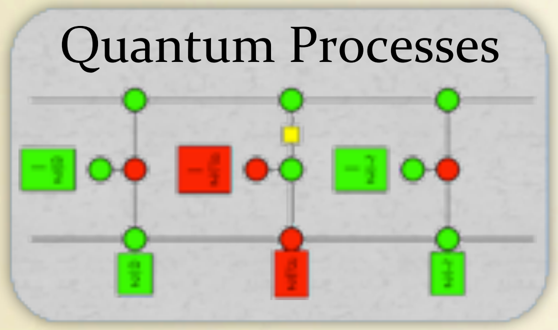
Fabio Gadducci

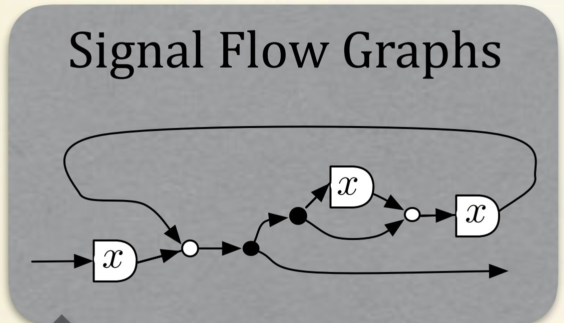


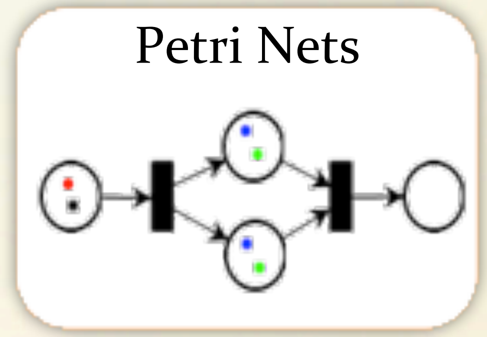
Aleks Kissinger

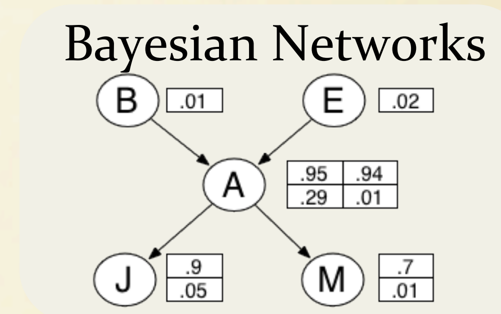
Pawel Sobocinski

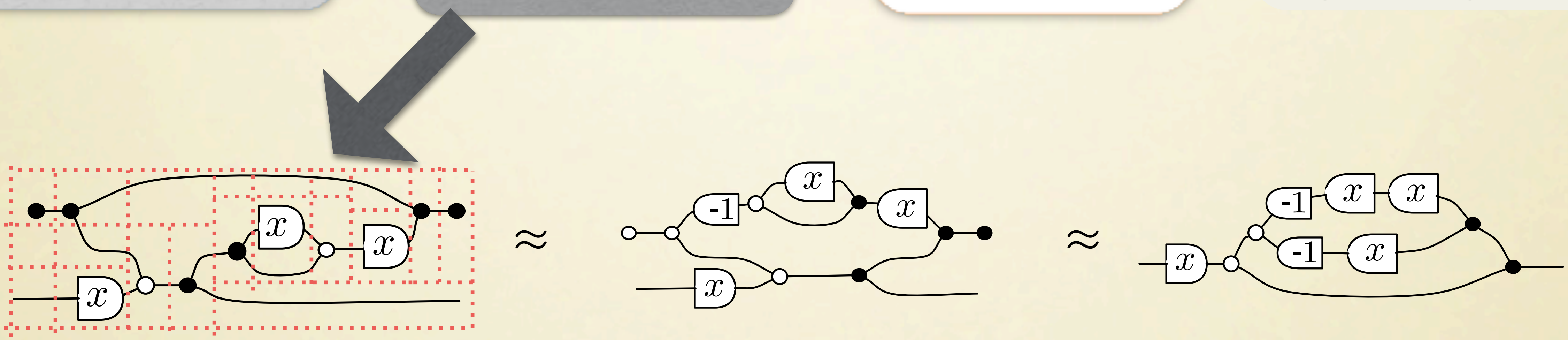
Algebras of network diagrams





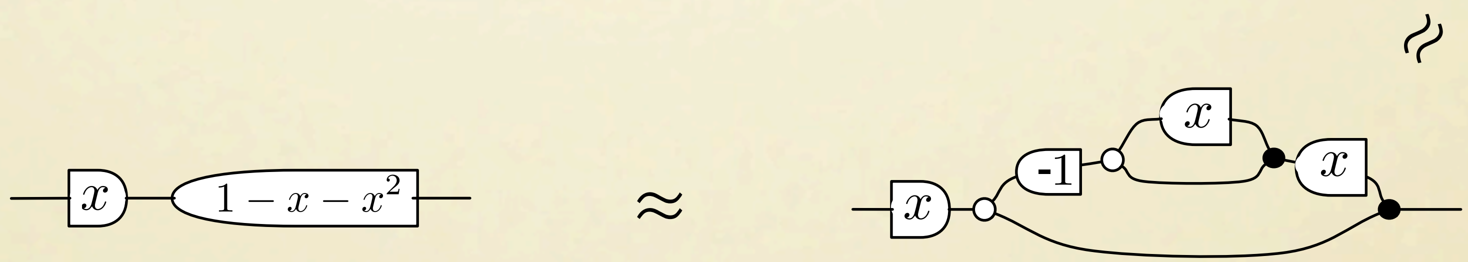






Fibonacci

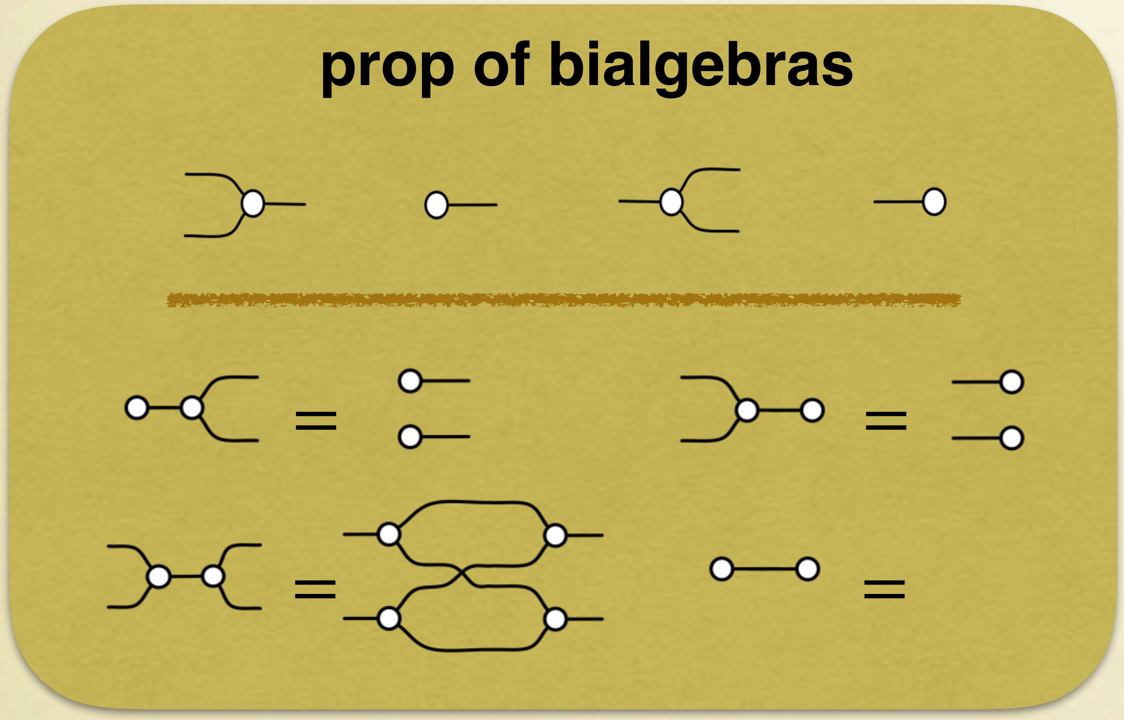
$$\frac{x}{1 - x - x^2}$$

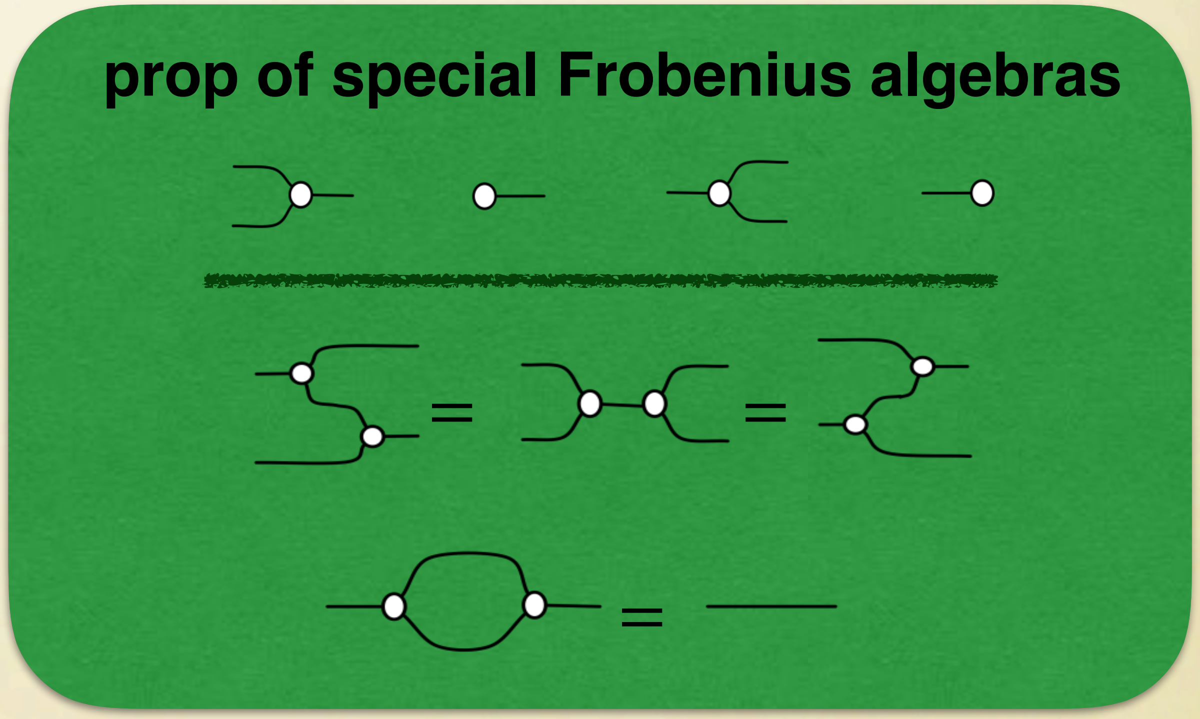


Props

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

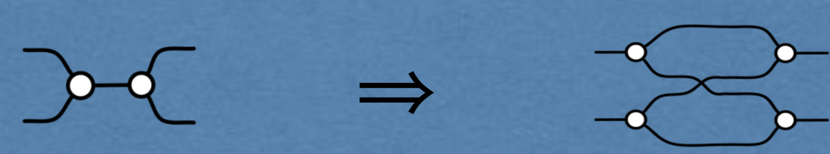


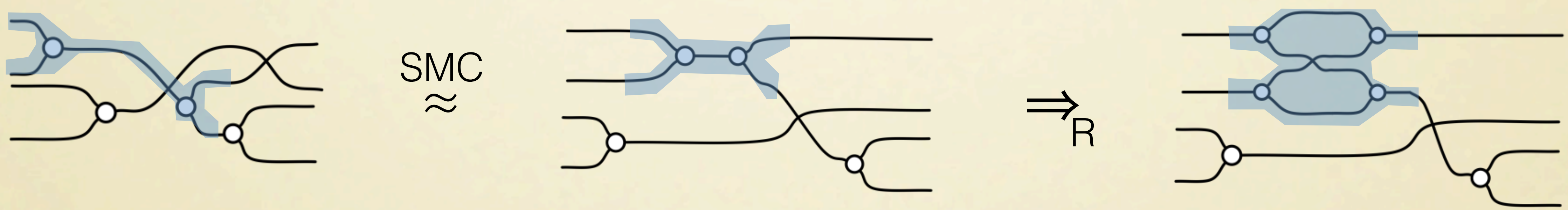


Rewriting in a prop

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

(R)

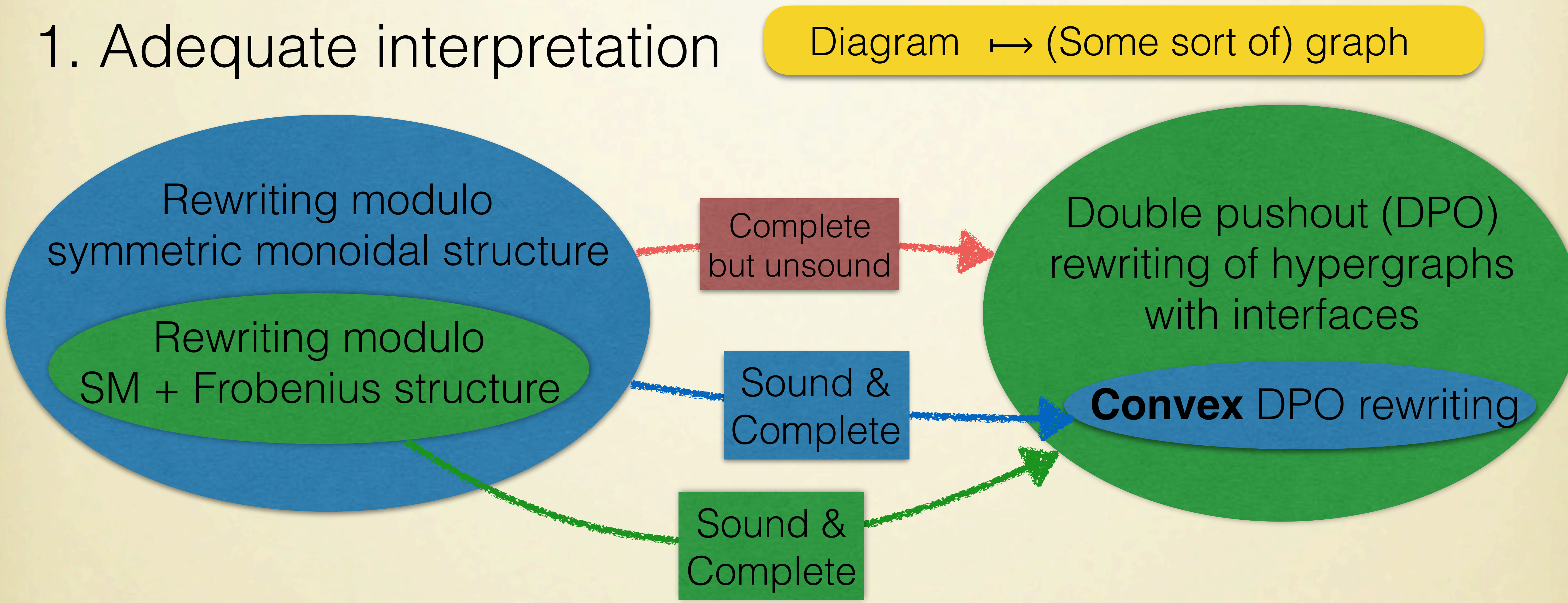




Our question
How to implement rewriting modulo symmetric monoidal structure in a simple, yet rigorous way?

Outline

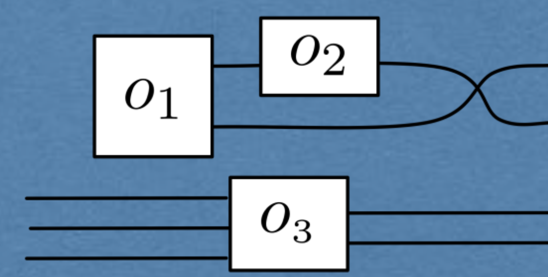
1. Adequate interpretation



2. Decidability of confluence

Hypergraph interpretation

prop ***Syn(Σ)*** of syntax
freely generated by
$\Sigma = \{ \boxed{o_1}, \boxed{o_2}, \boxed{o_3} \}$

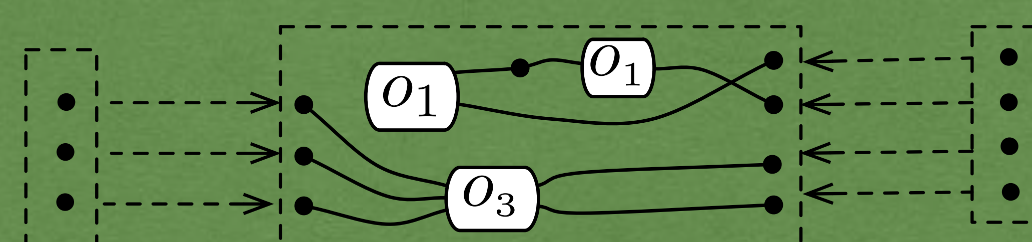


Operations in Σ ~ Hyperedges
L/R boundary ~ Cospan structure

$\langle\!\langle \cdot \rangle\!\rangle$

prop ***Hyp(Σ)*** of
Σ-labelled hypergraphs

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

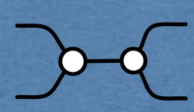


Proposition $\langle\!\langle \cdot \rangle\!\rangle$: *Syn(Σ)* → *Csp(Hyp(Σ))* is faithful.

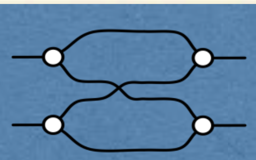
DPO rewriting with interfaces

Hyp(Σ) is an *ahdesive category* (Lack & Sobocinski) and thus adapted to DPO rewriting.

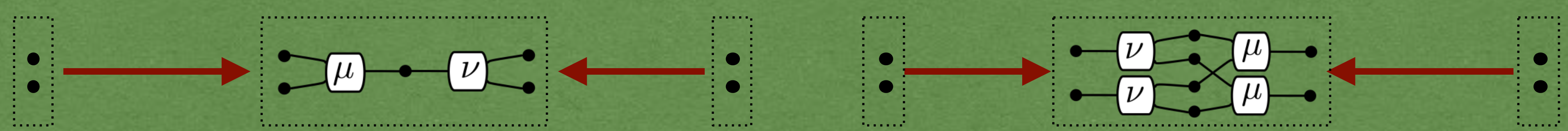
(R)



⇒



Syn(Σ)

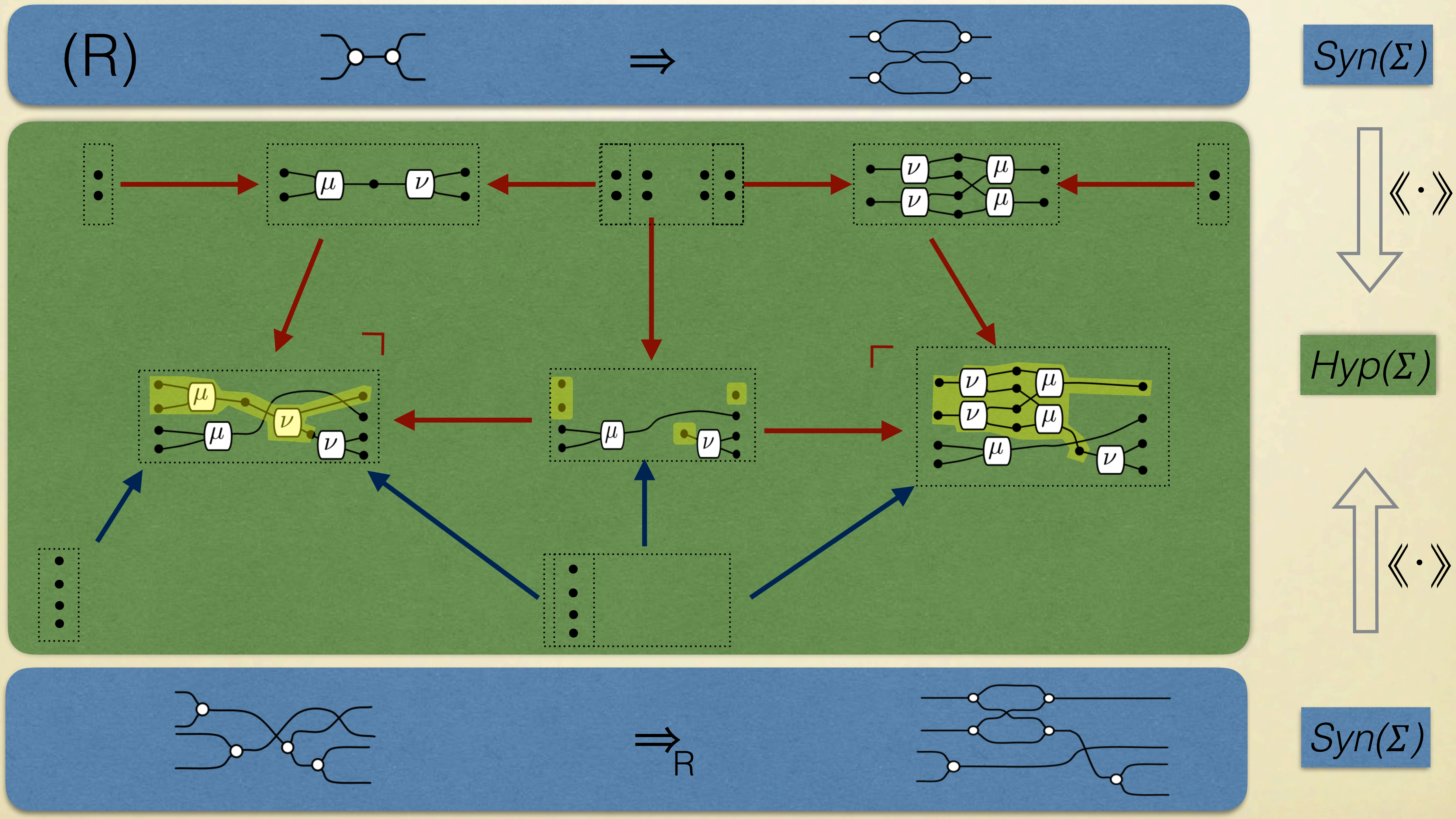


⟪·⟫

Hyp(Σ)

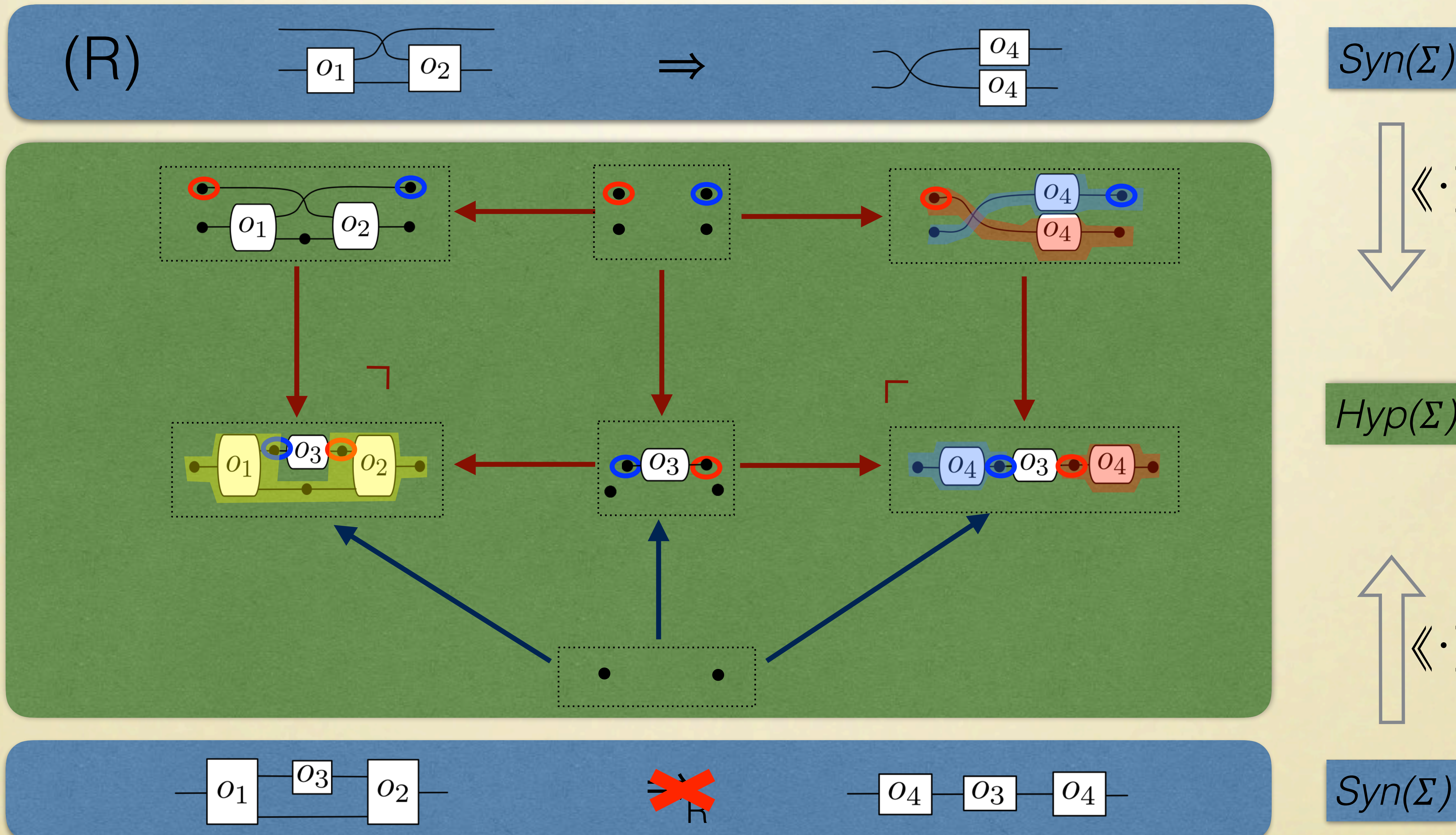
DPO rewriting with interfaces

Hyp(Σ) is an *ahdesive category* (Lack & Sobocinski)
and thus adapted to DPO rewriting.



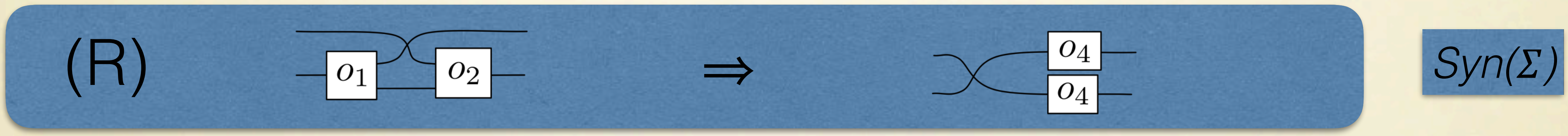
DPO rewriting can be unsound

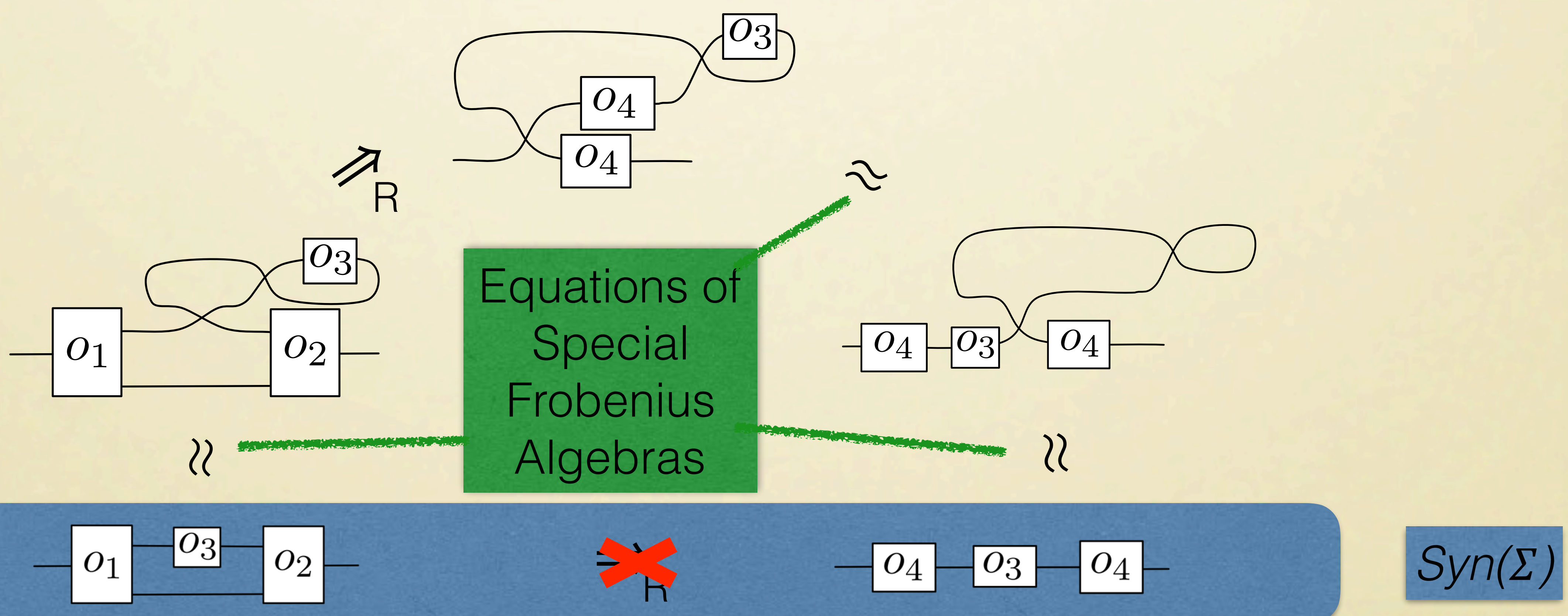
Rewriting in *Hyp(Σ)* is complete but generally not sound



DPO rewriting can be unsound

Rewriting in *Hyp(Σ)* is complete but generally not sound



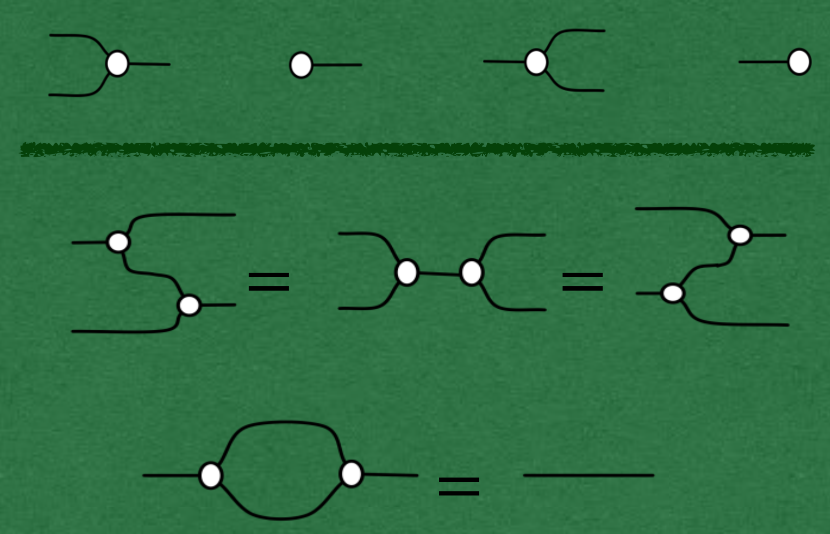


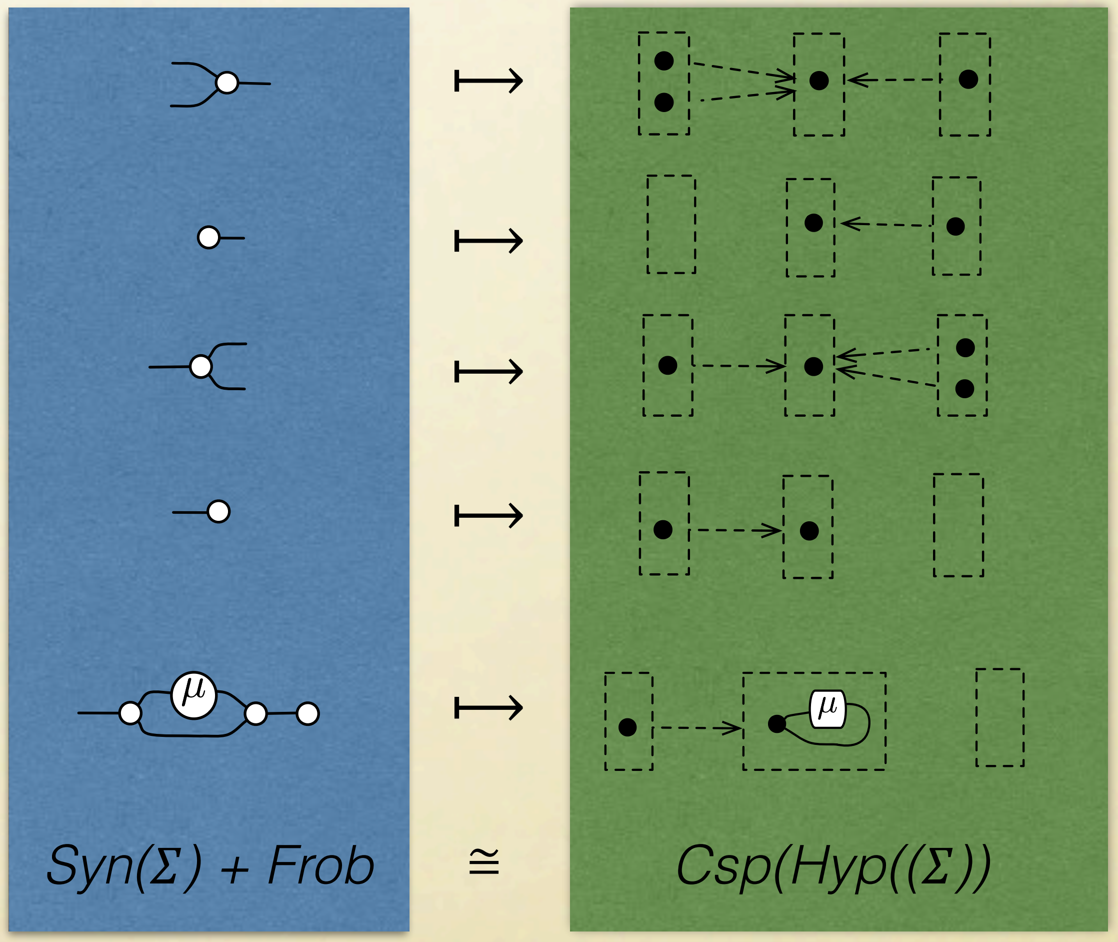
Frobenius makes DPO rewriting sound

Theorem I

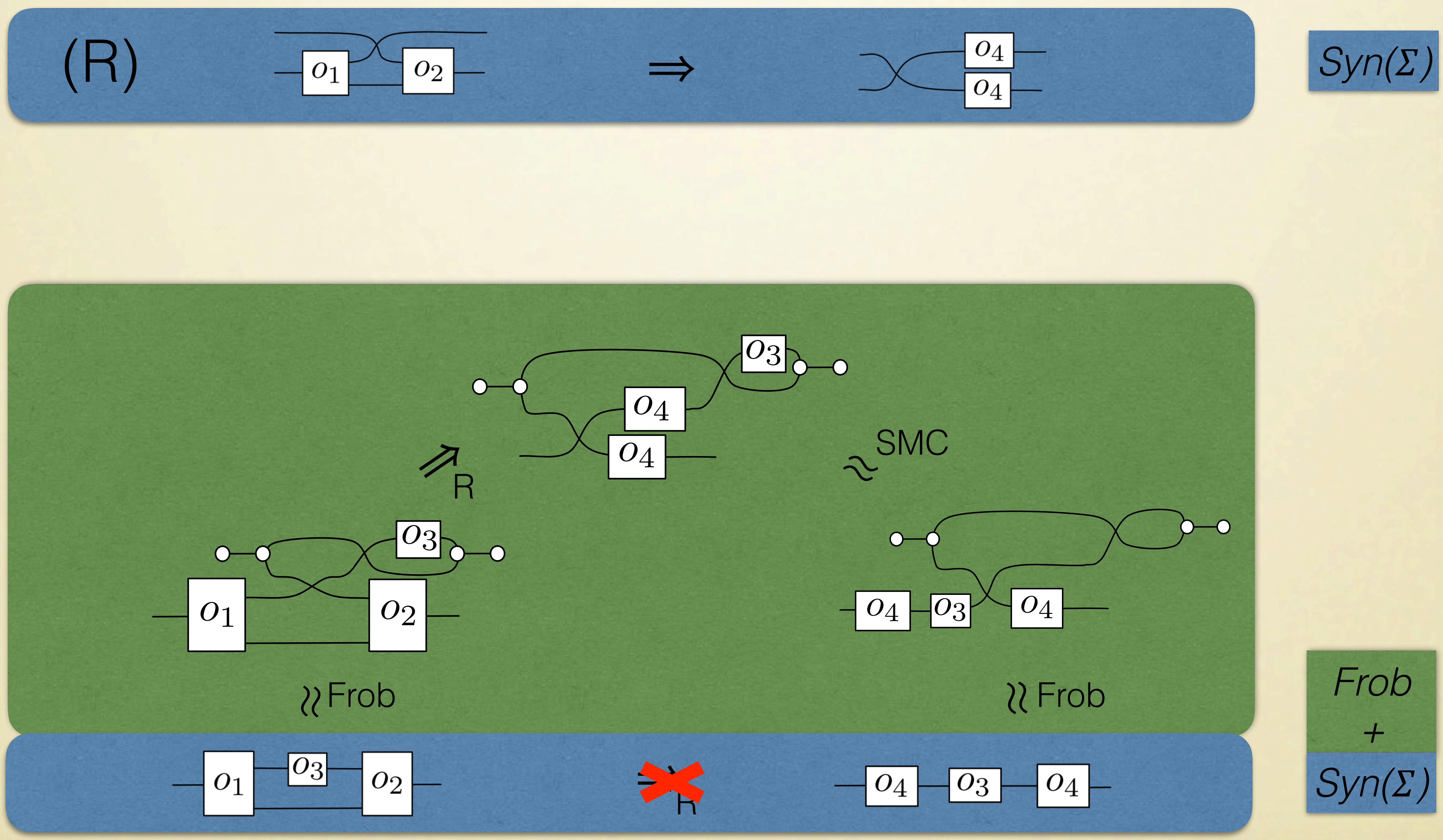
DPO rewriting with interfaces is *sound and complete*
for symmetric monoidal categories
with a chosen *special Frobenius structure*.

prop of special Frobenius algebras



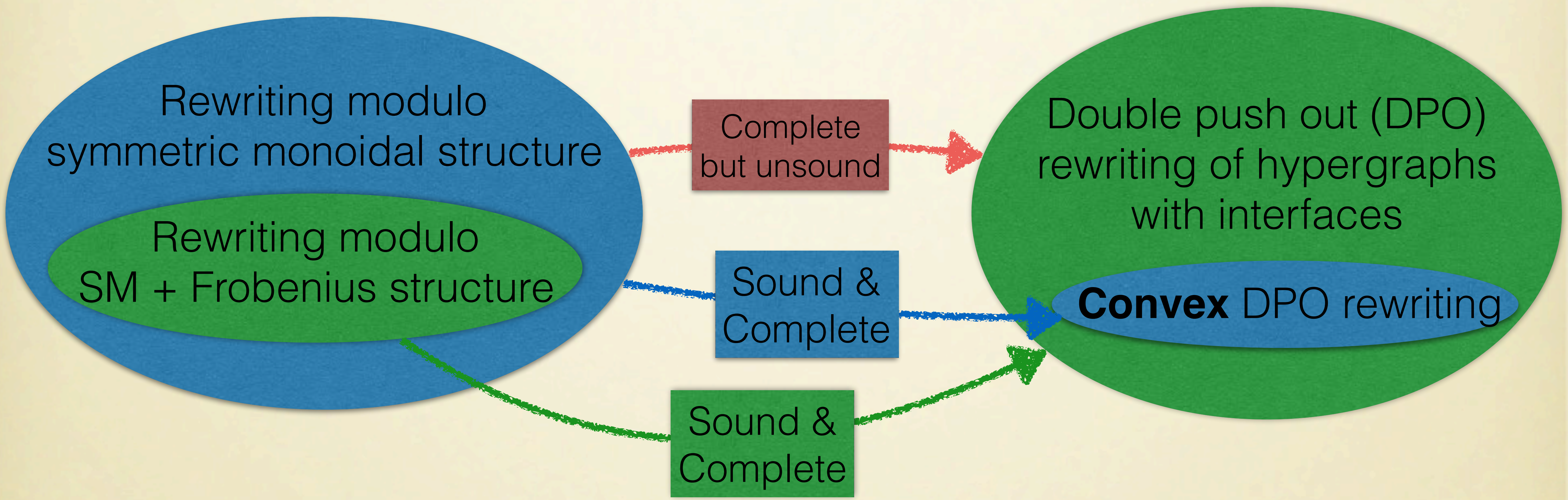


Frobenius makes DPO rewriting sound



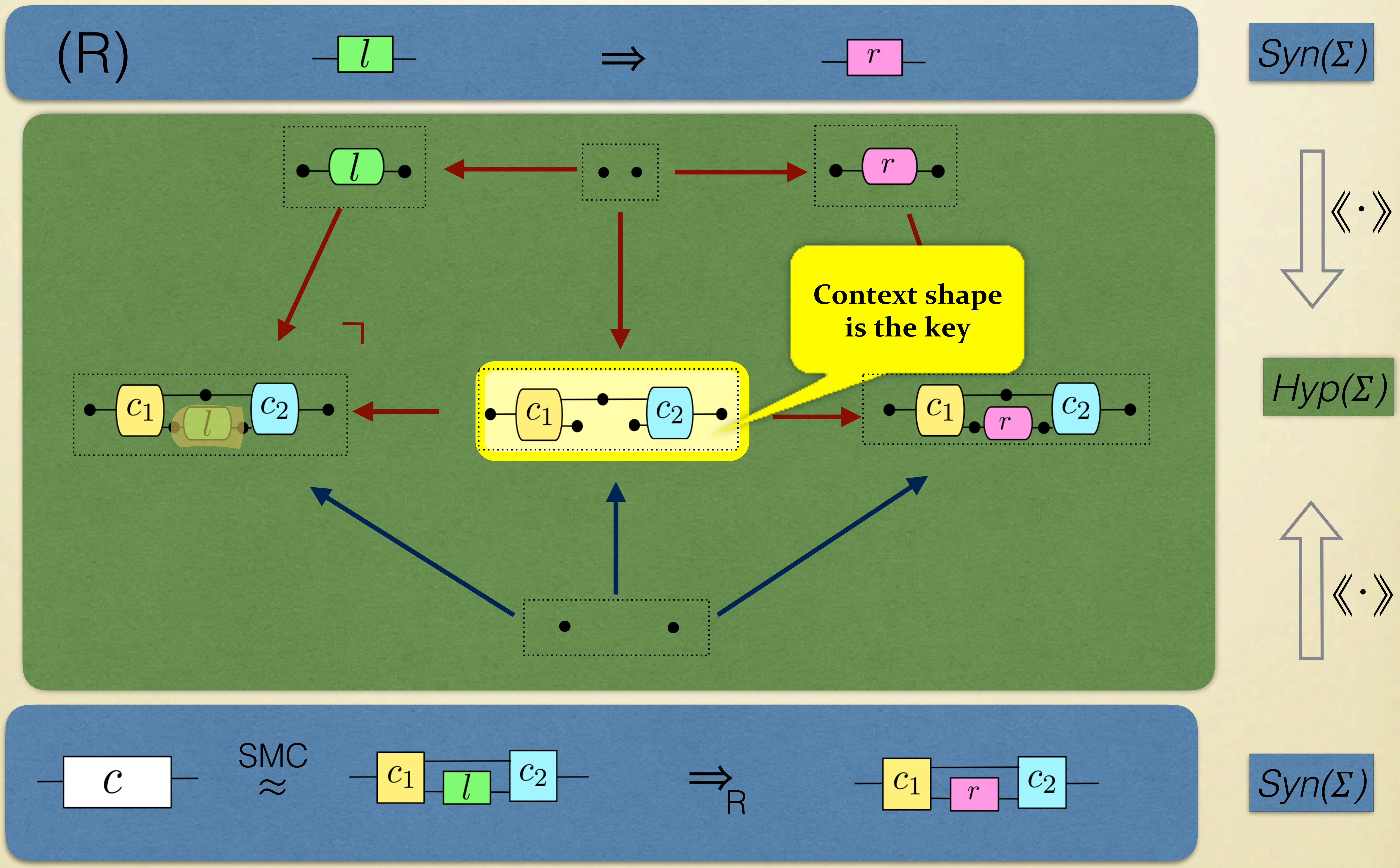
Where we are, so far

1. Adequate interpretation

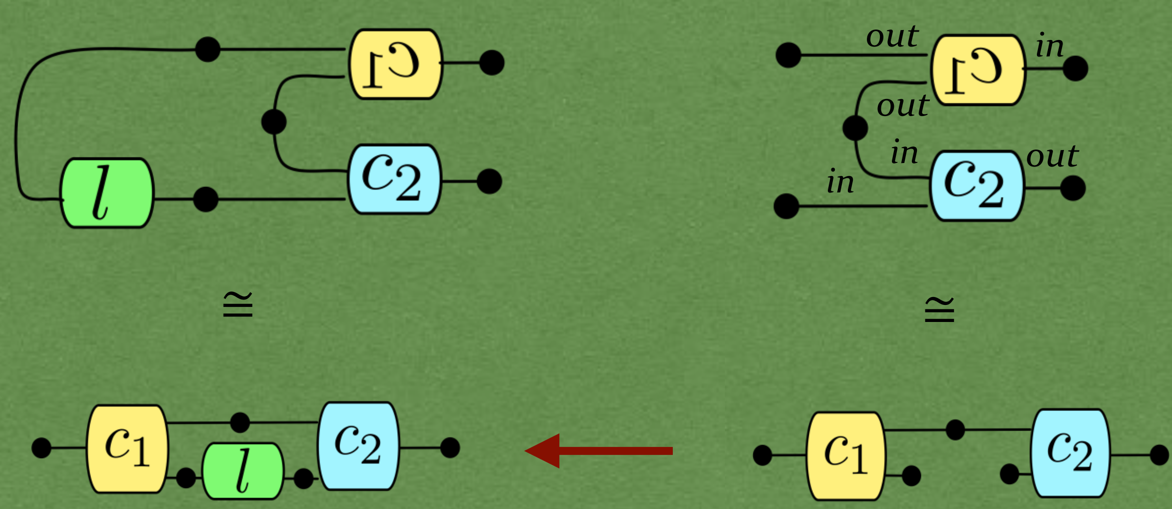


2. Decidability of confluence

How does sound DPO rewriting look like?



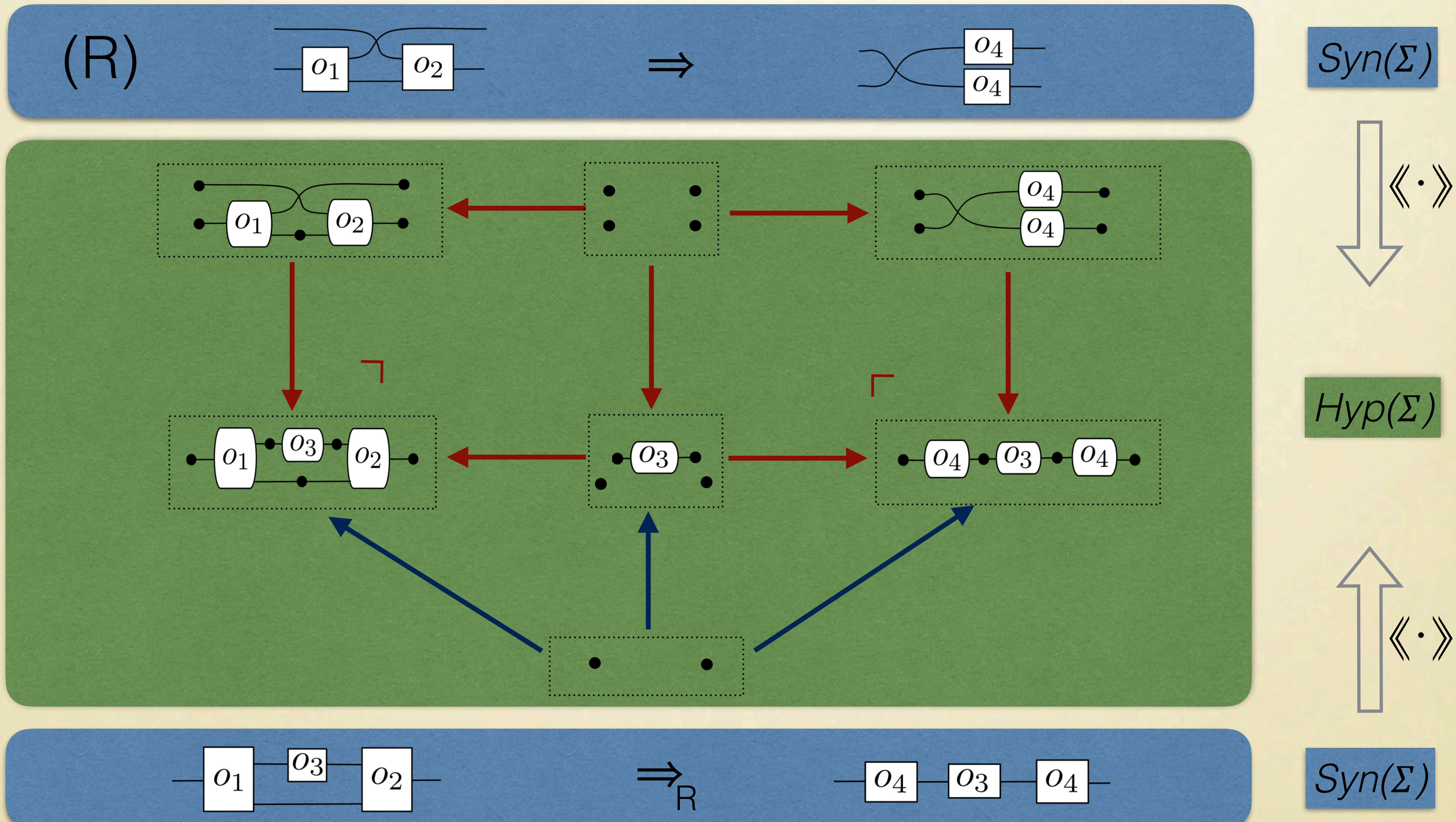
How does sound DPO rewriting look like?



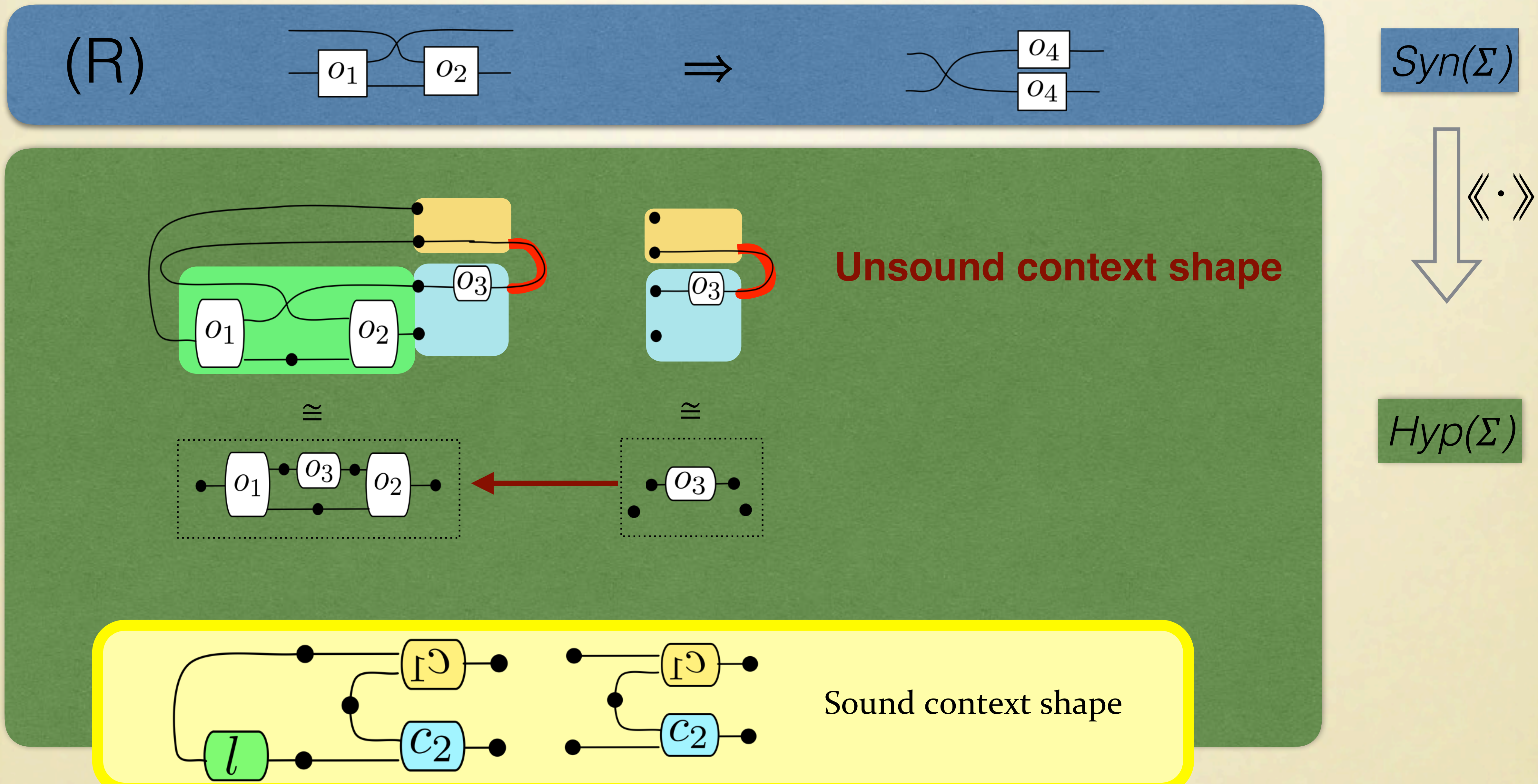
Leading Intuition
a rewriting steps
is sound
iff
the rewriting context
has this shape

Hyp(Σ)

Back to the soundness counterexample



Back to the soundness counterexample



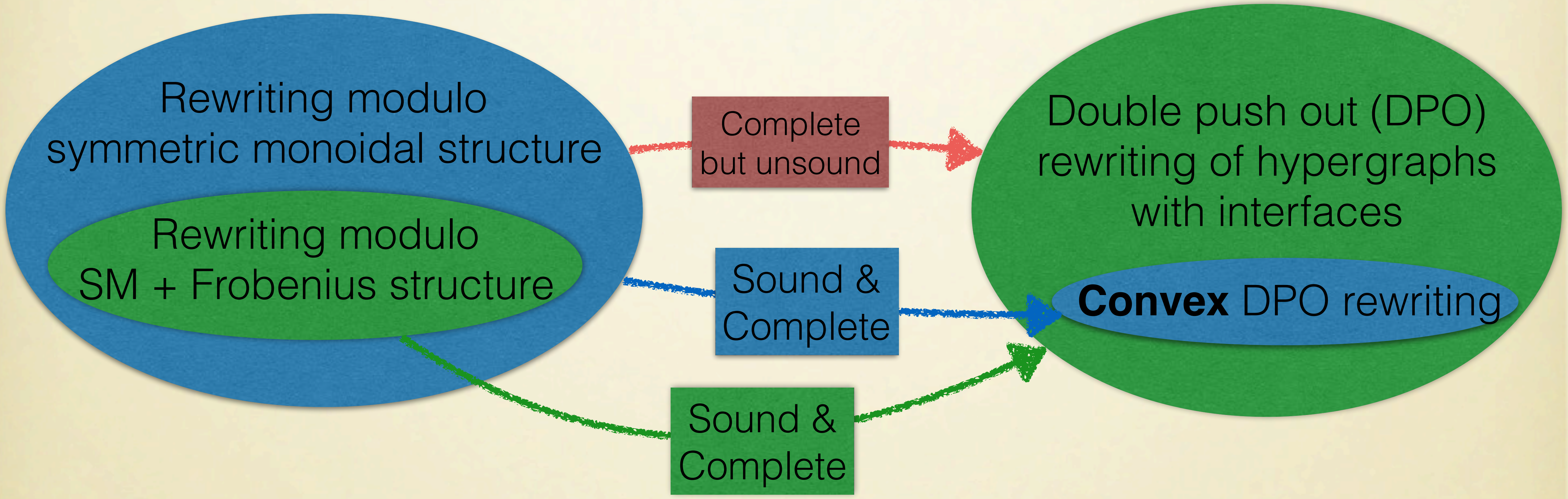
Convex DPO rewriting is sound

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.

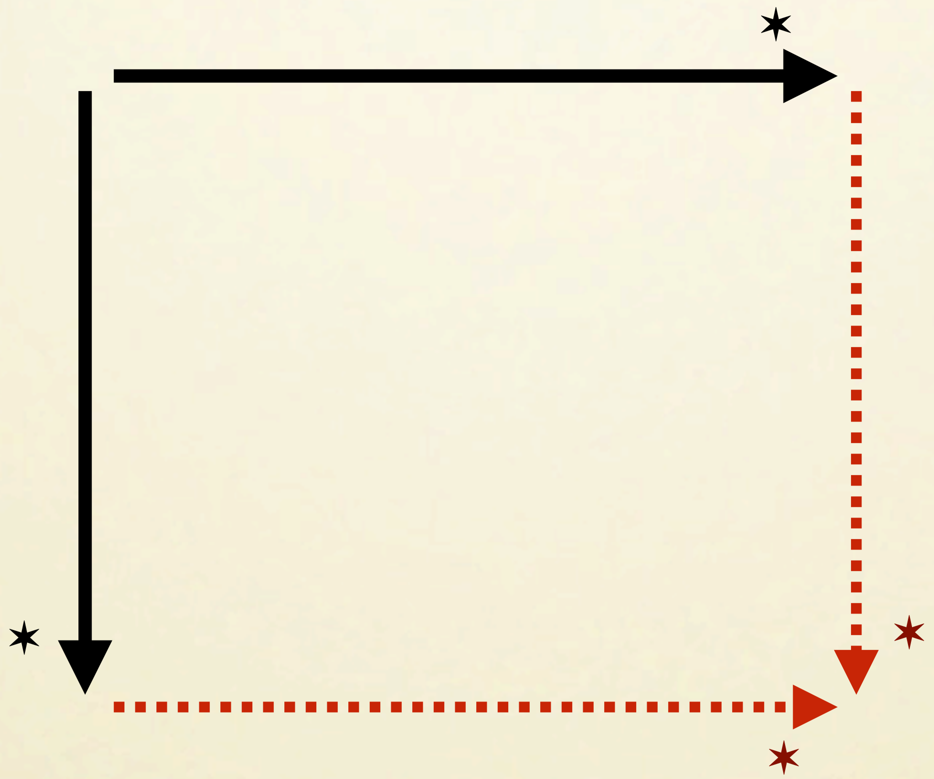
Where we are, so far

1. Adequate interpretation



2. Decidability of confluence

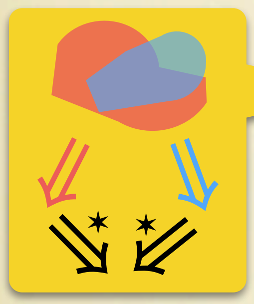
Confluence, abstractly



If E is confluent & terminating
then $x \overset{E}{=} y$ becomes decidable.

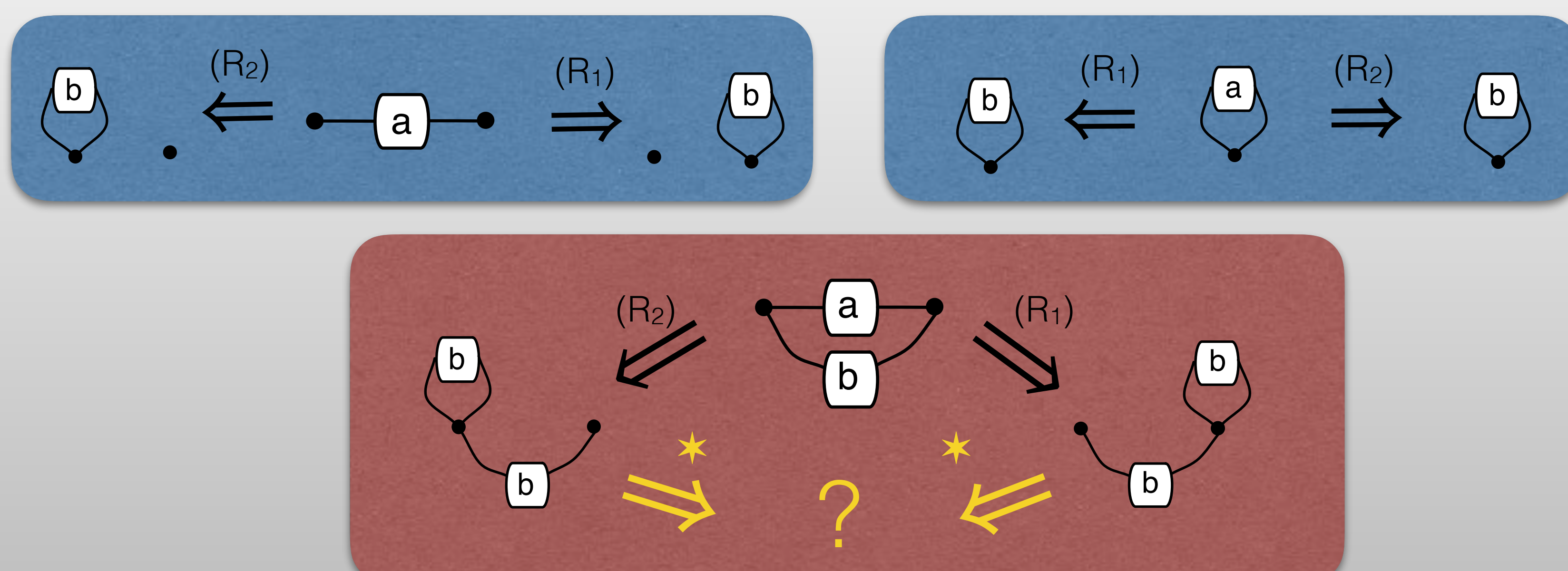
Decidability of Confluence

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



All the critical pairs are joinable $\Longleftrightarrow$ The system is confluent (Knuth-Bendix)

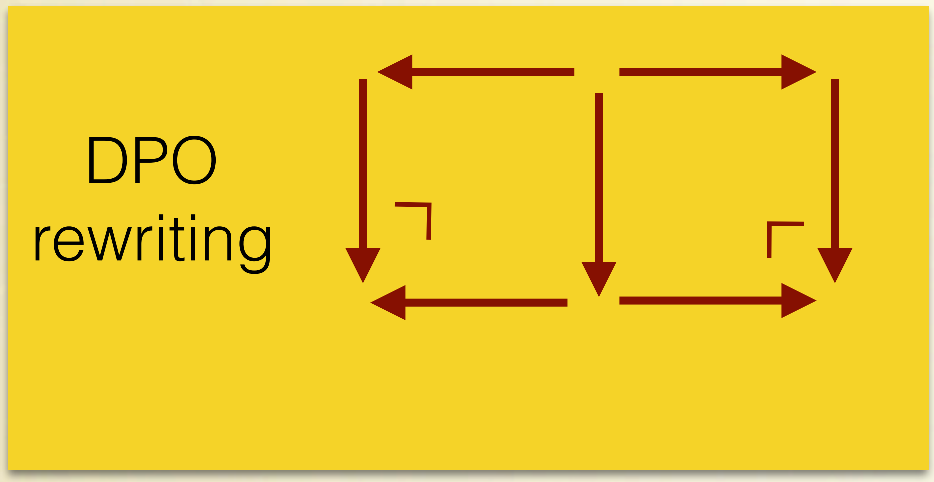
In DPO (hyper)graph rewriting, confluence is **undecidable** (Plump)



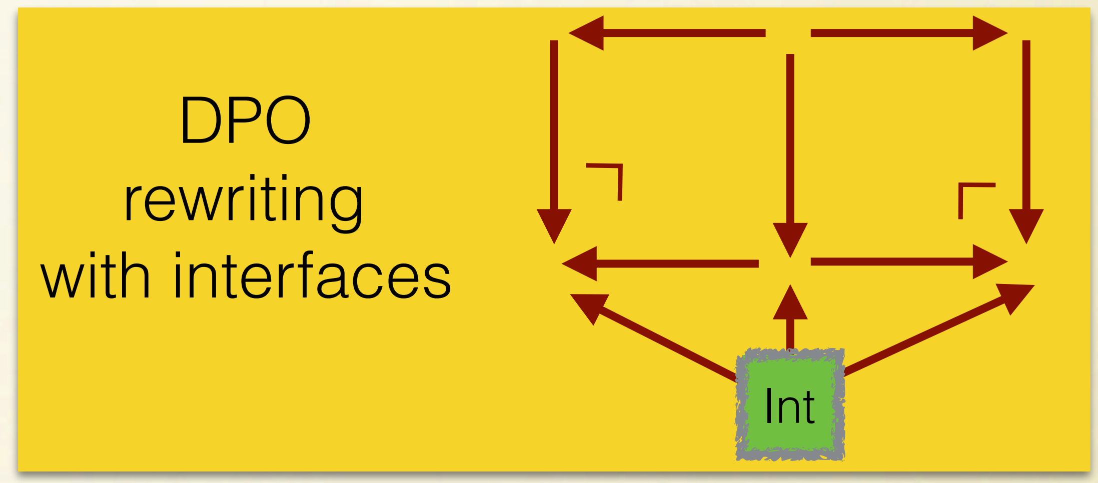
All the critical pairs are joinable…

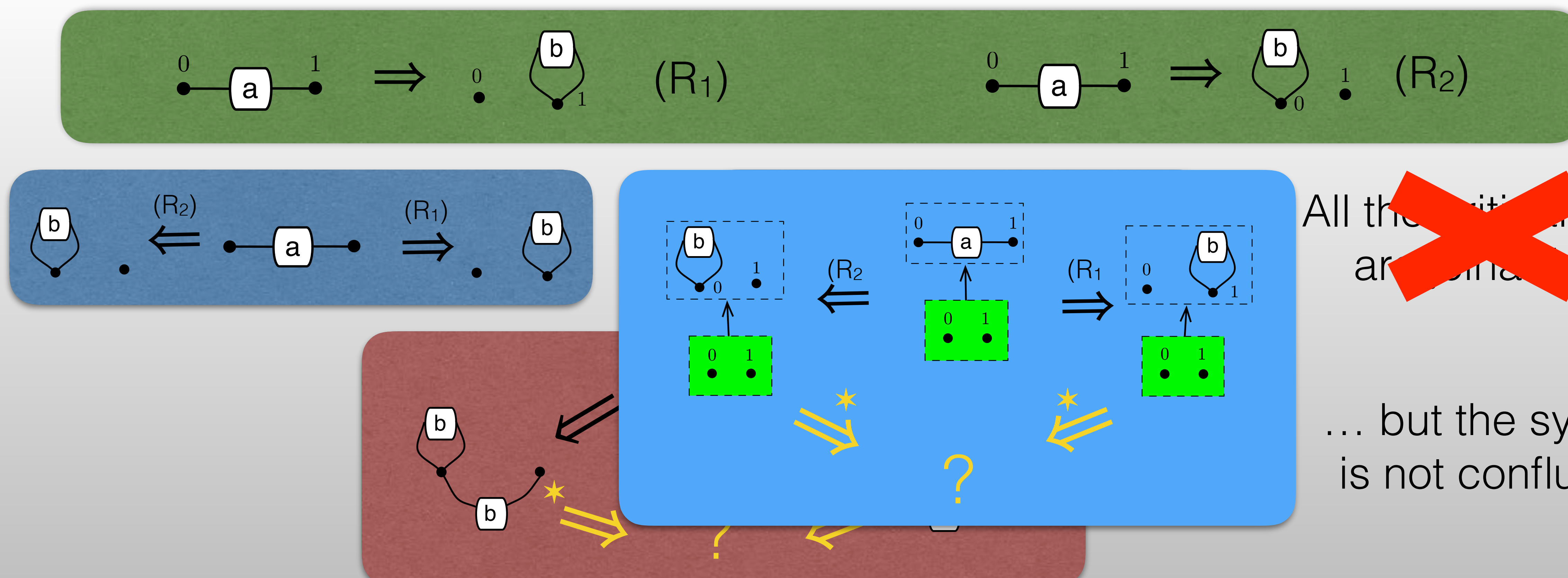
… but the system is not confluent.

Interfaces to the Rescue



VS





All th … pairs
ar … e…

… but the system
is not confluent.

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.

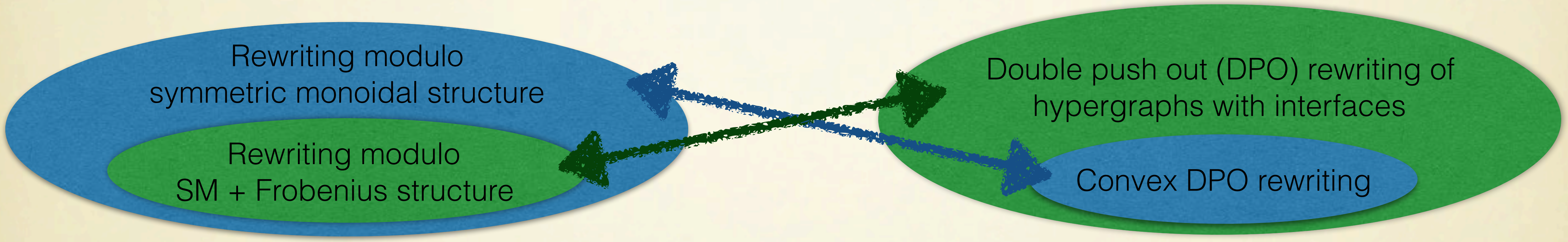
Theorem II bis

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

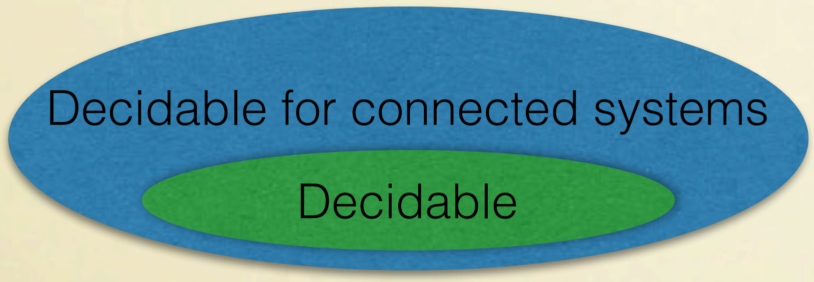
Confluence is decidable for *connected* terminating rewriting systems on such categories.

Conclusions

Adequacy



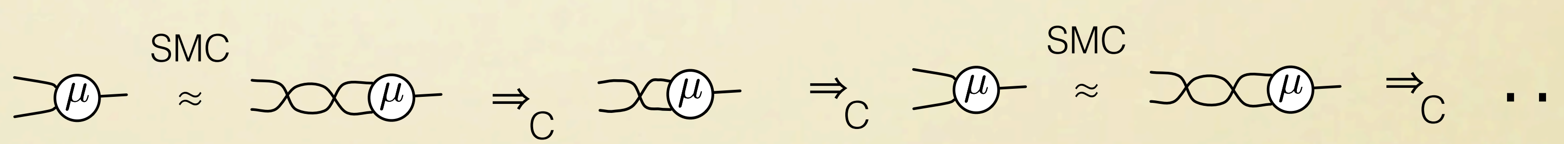
Confluence



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

Termination

Commutativity does not terminate



Proposal: interpret commutative operators as nodes of a new sort