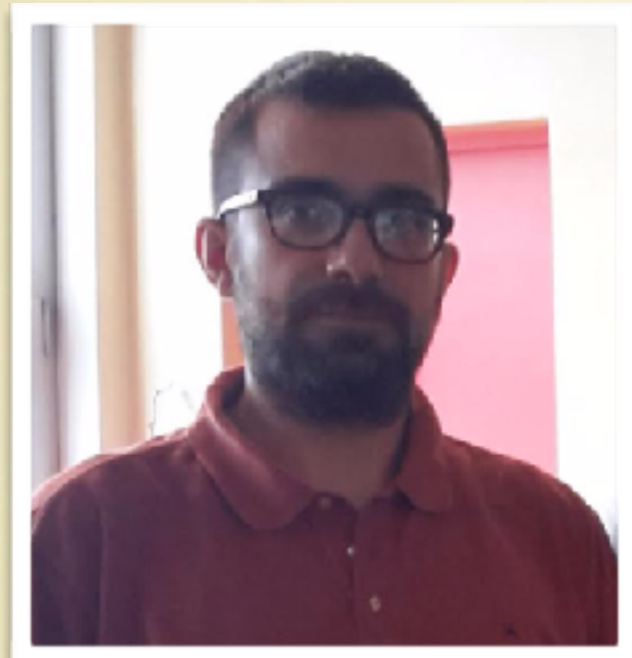
New Foundations for String Diagram Rewriting

Fabio Zanasi
University College London

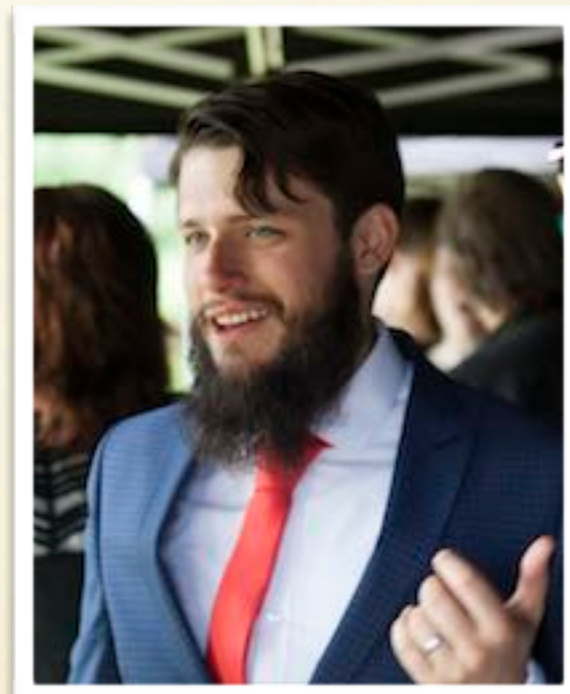
Birmingham, June 2, 2017

Collaborators



Filippo
Bonchi

Fabio
Gadducci

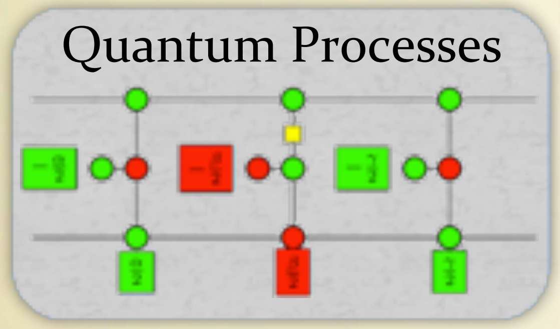


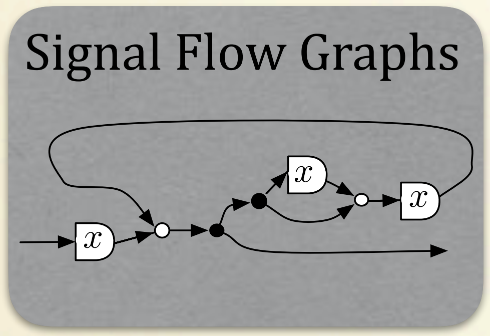
Aleks
Kissinger

Pawel
Sobocinski

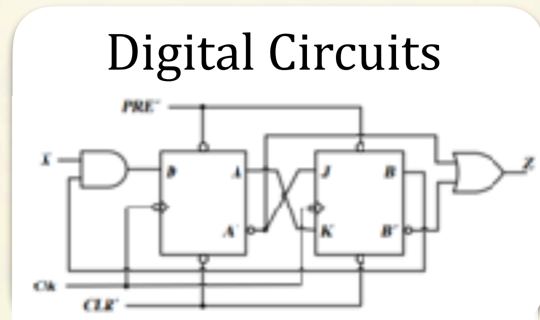
Context and Motivation

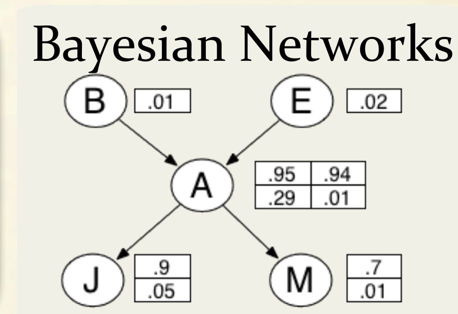
The algebraic approach to network diagrams


Quantum Processes

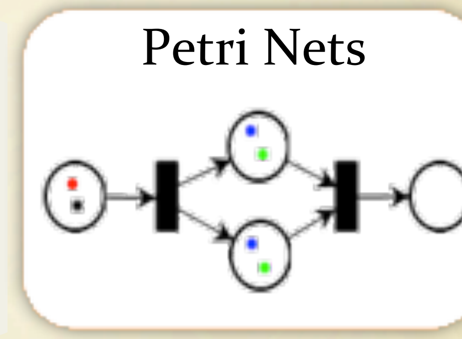


Signal Flow Graphs


Digital Circuits



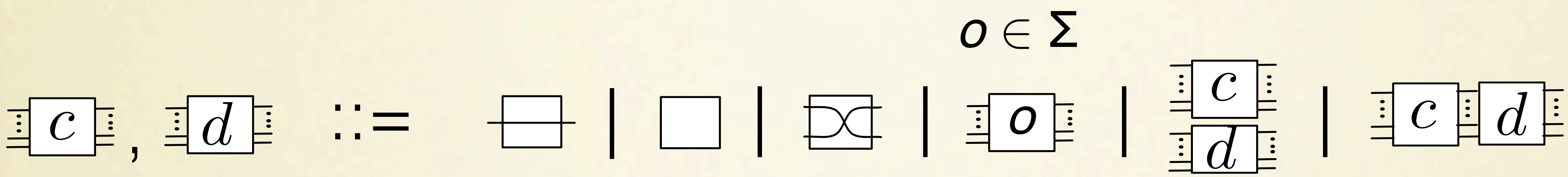
Bayesian Networks


Petri Nets

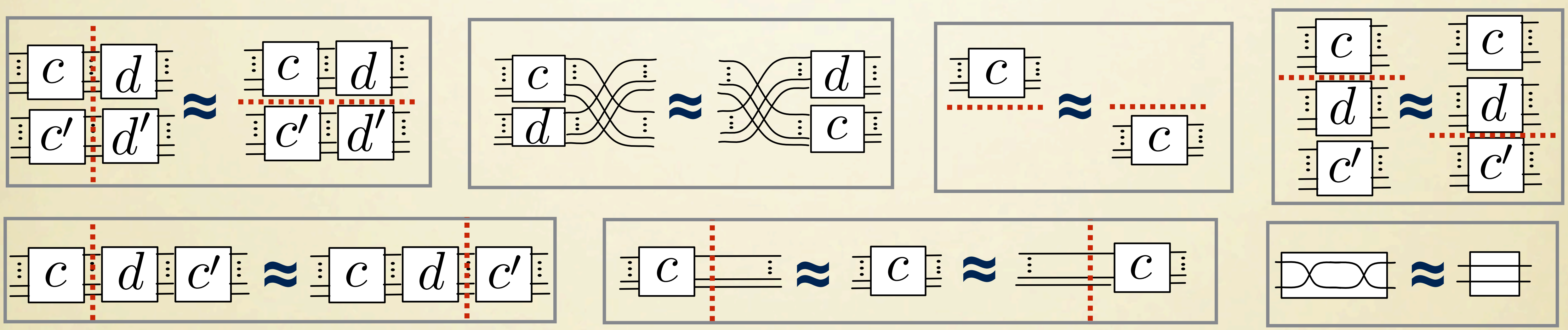
Diagrammatic languages studied using the compositional methods of programming language theory.

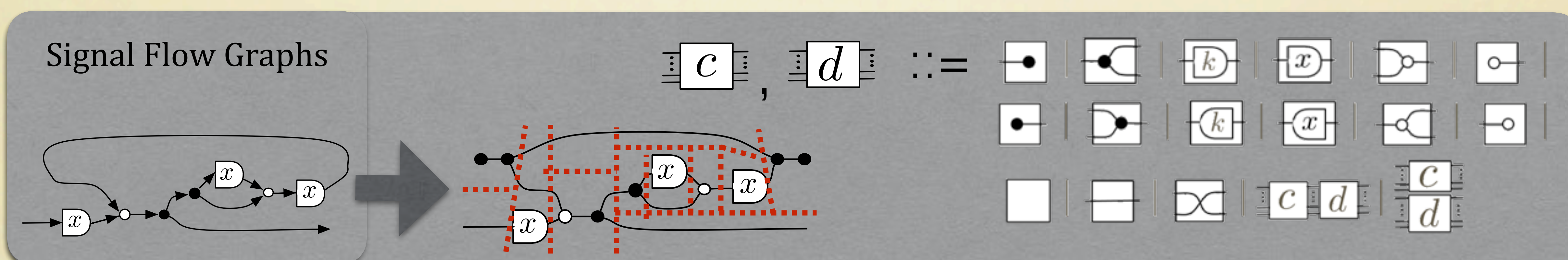
Syntax + **Equations**

Graphical syntax



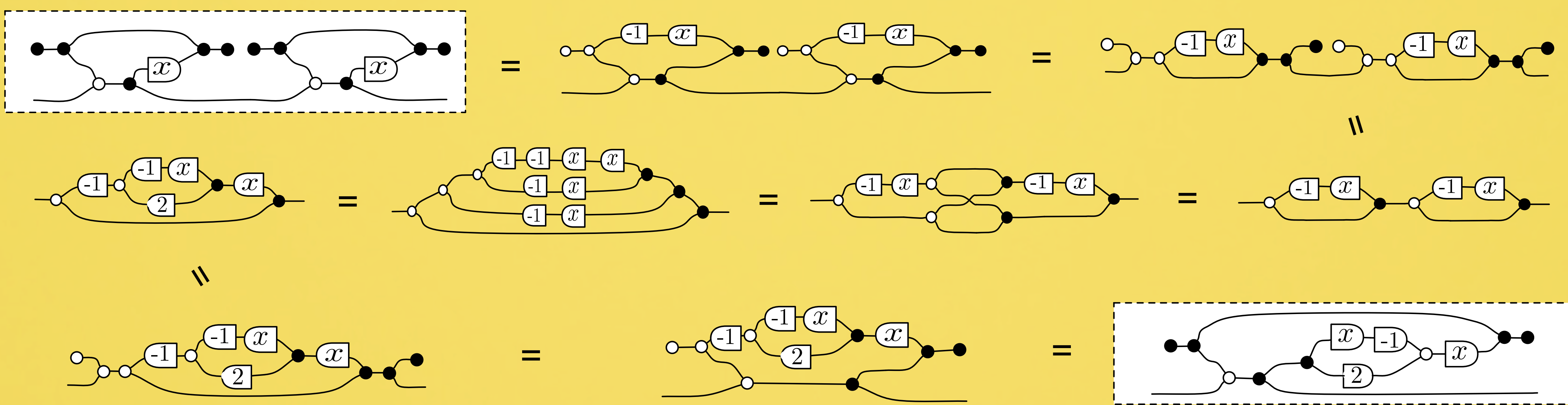
Quotiented by the *laws of symmetric monoidal categories*:



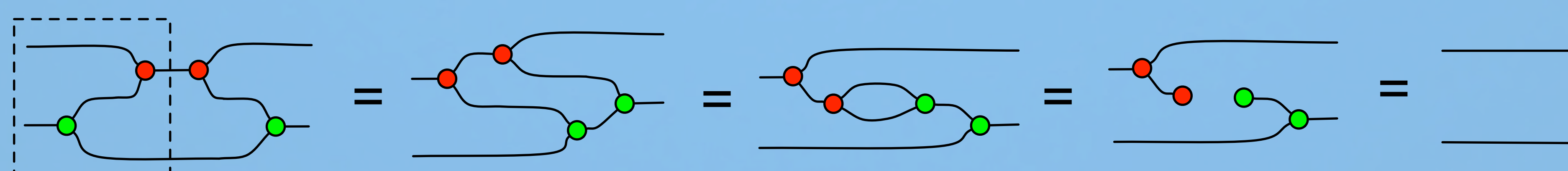


Equational theories of diagrams

Signal flow graphs: circuit equivalence



Quantum processes: the CNOT gate is unitary

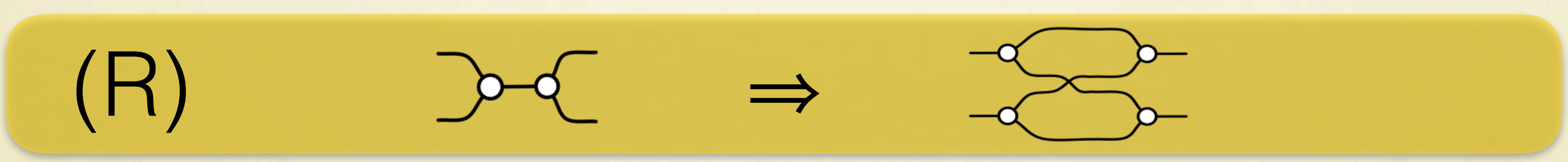


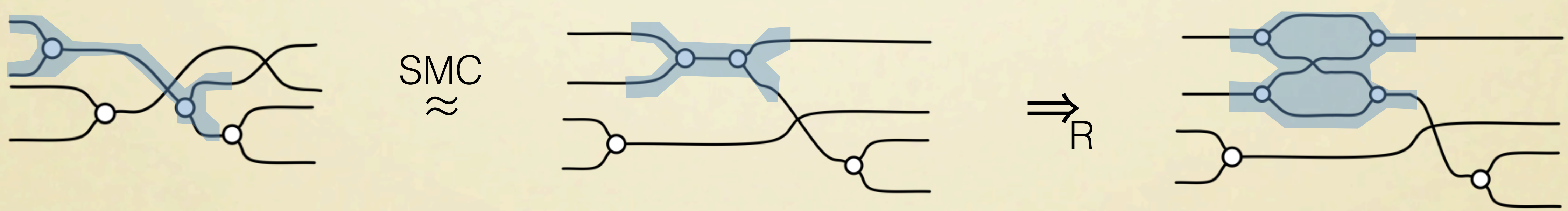
Bayesian reasoning: disintegration

Perspective of this work

Syntax + **Equations**

Modulo the laws of SMCs — **Diagram rewriting system**

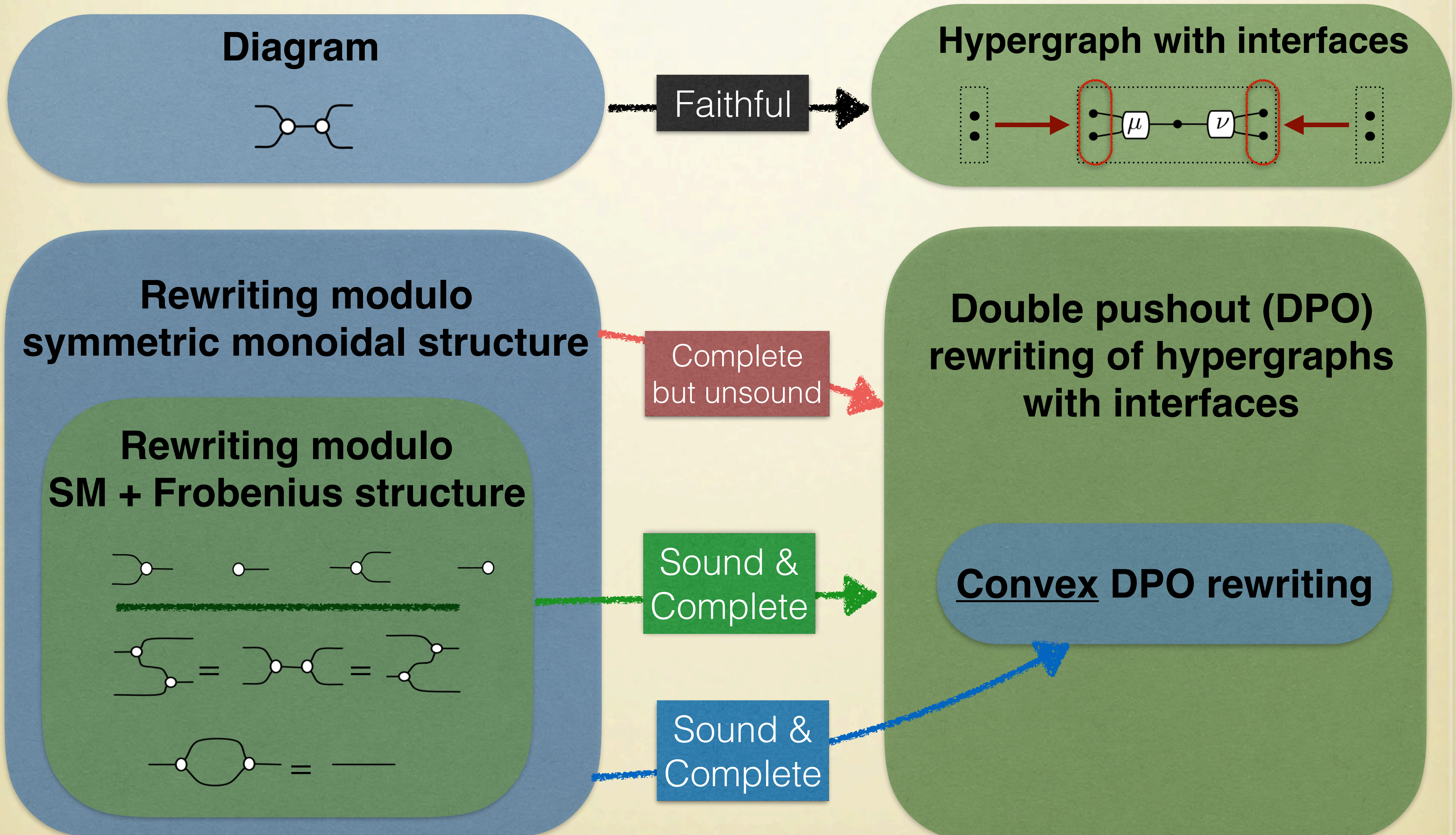




How to implement rewriting modulo symmetric monoidal structure in a simple, yet rigorous way?

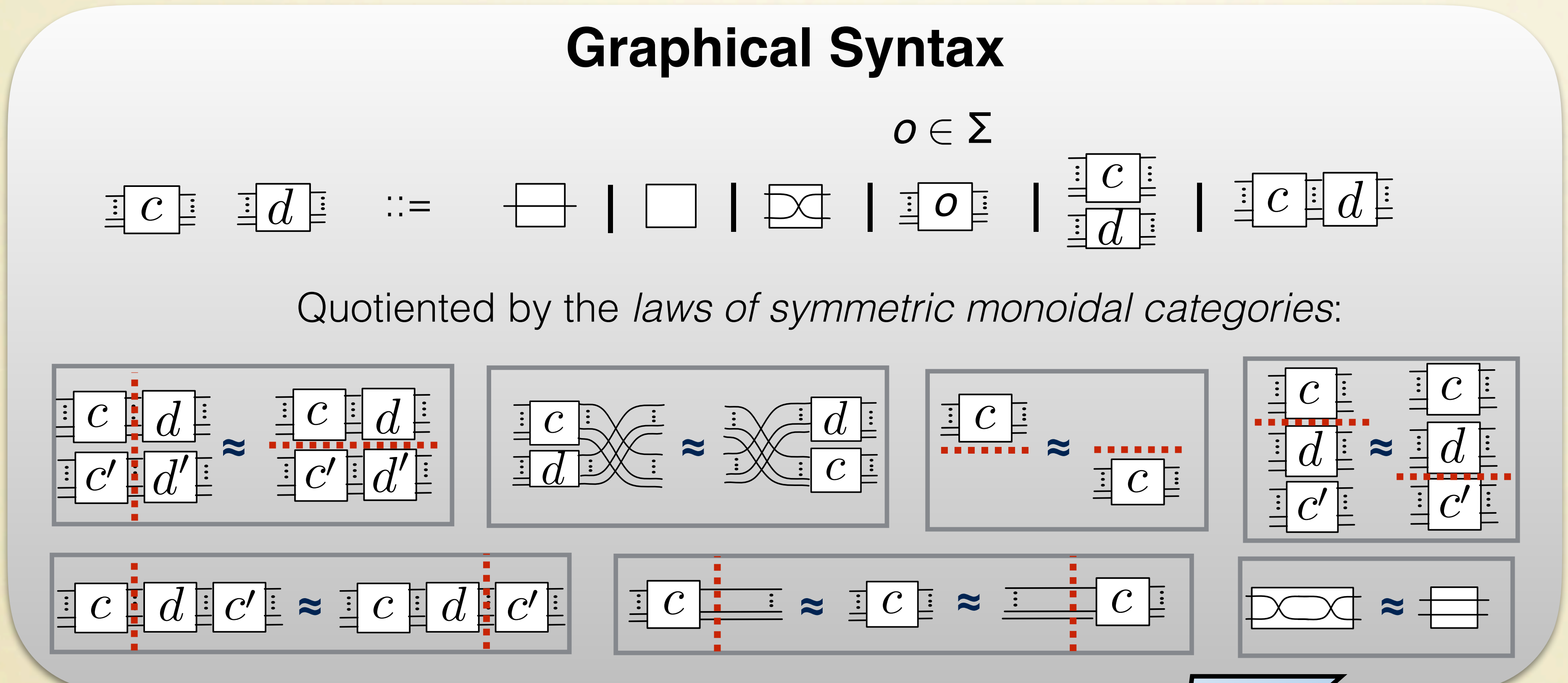
Implementing Diagram Rewriting

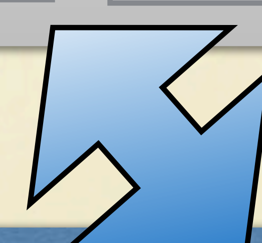
The graph interpretation (LICS'16)



Props

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

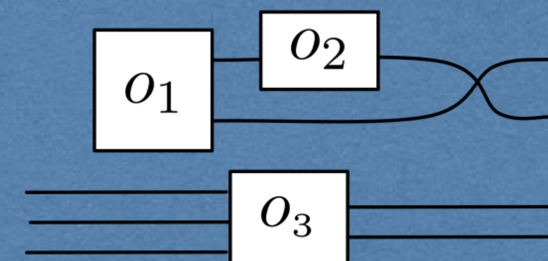




Prop ***Syn(Σ)***
freely generated by { o | $o \in \Sigma$ }

Hypergraph interpretation

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

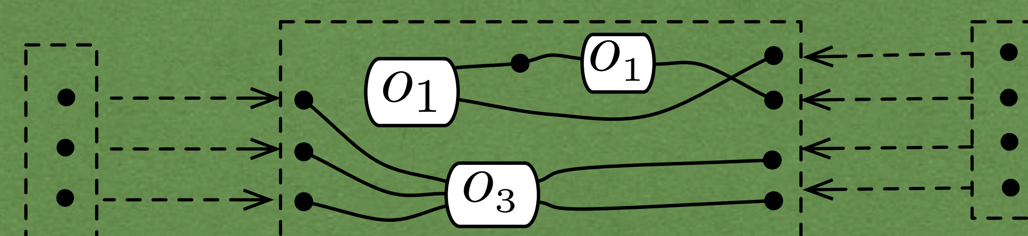


Operations in Σ ~ Hyperedges
L/R boundary ~ Cospan structure

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

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

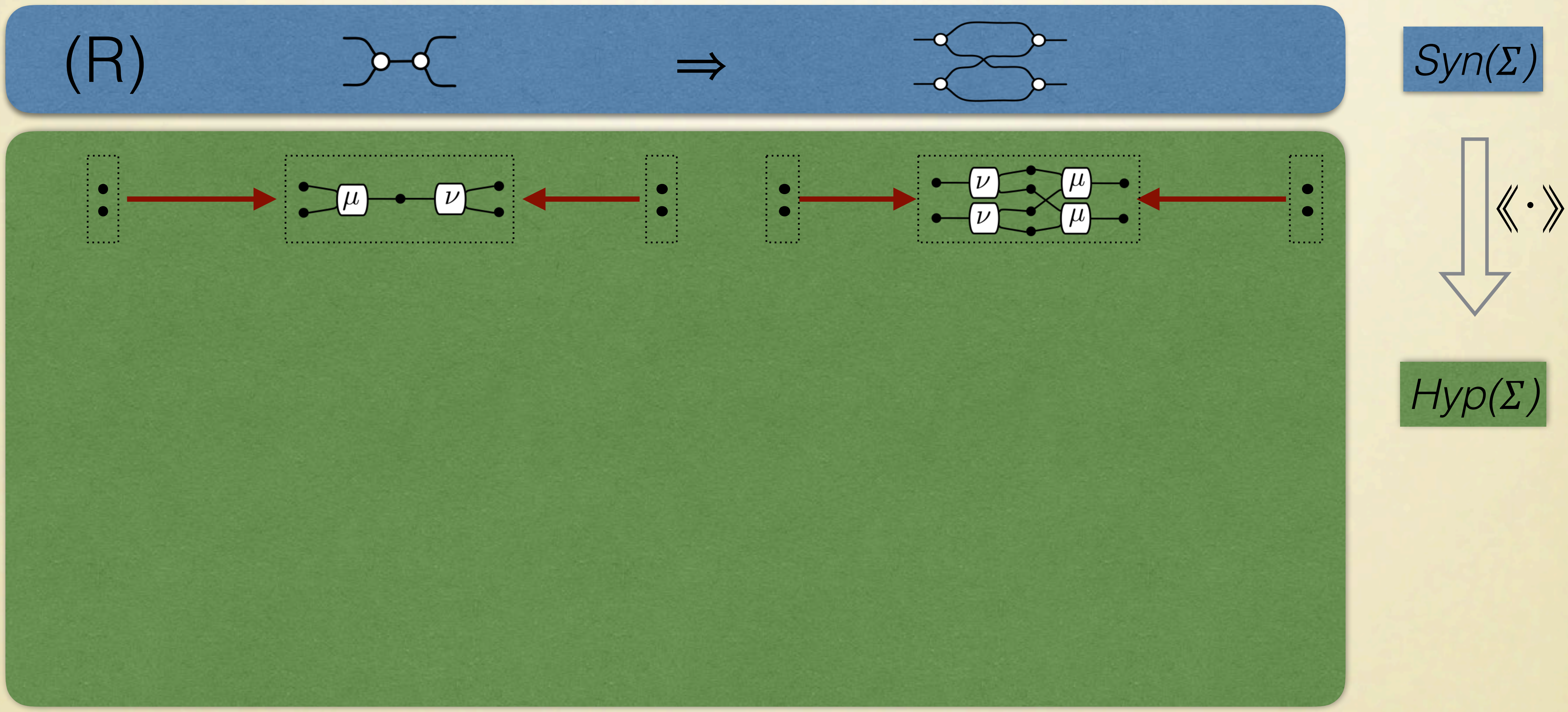
prop ***Csp(Hyp(Σ))*** of (discrete) cospans
of Σ-labelled hypergraphs



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

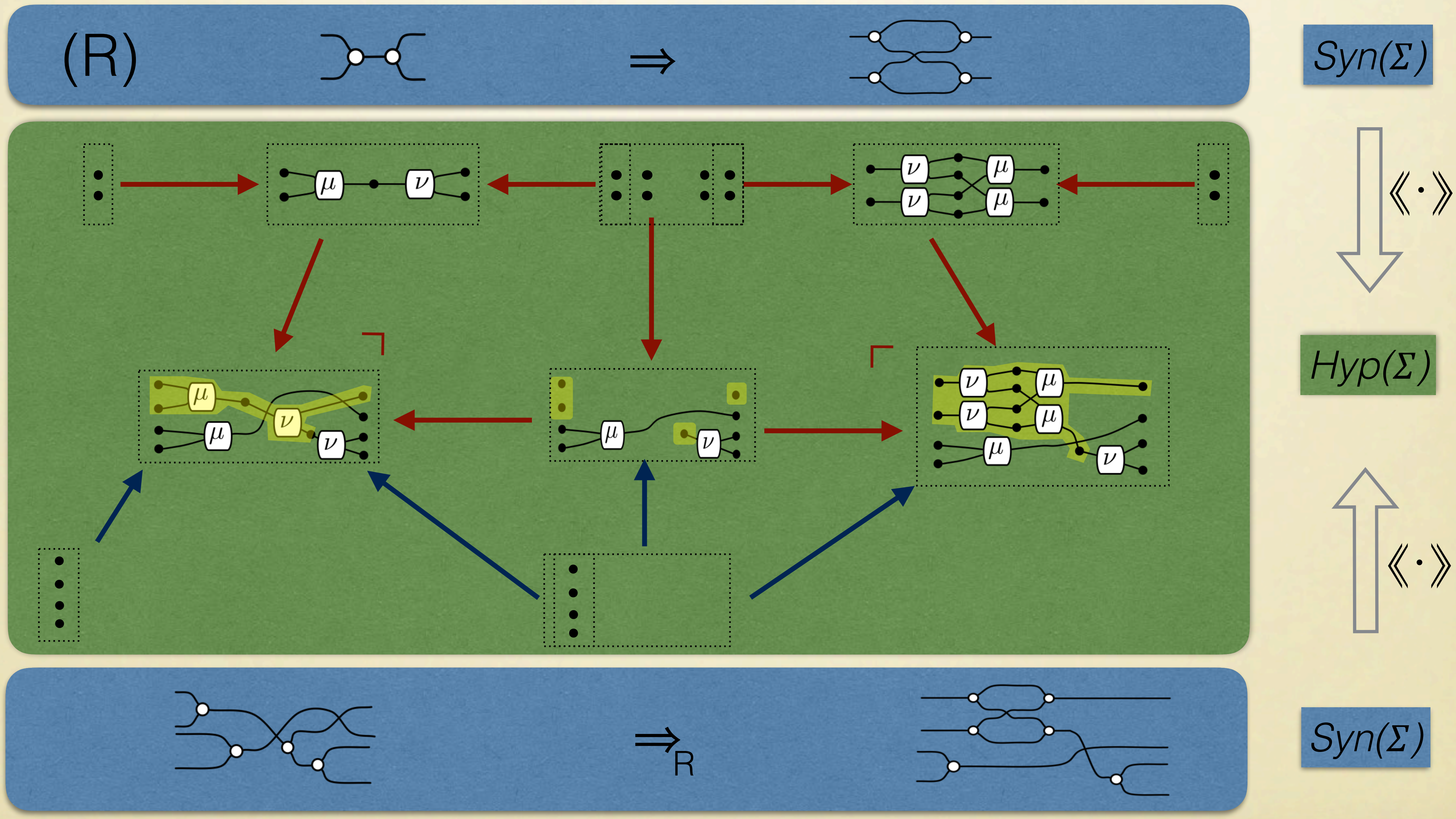
DPO rewriting with interfaces

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



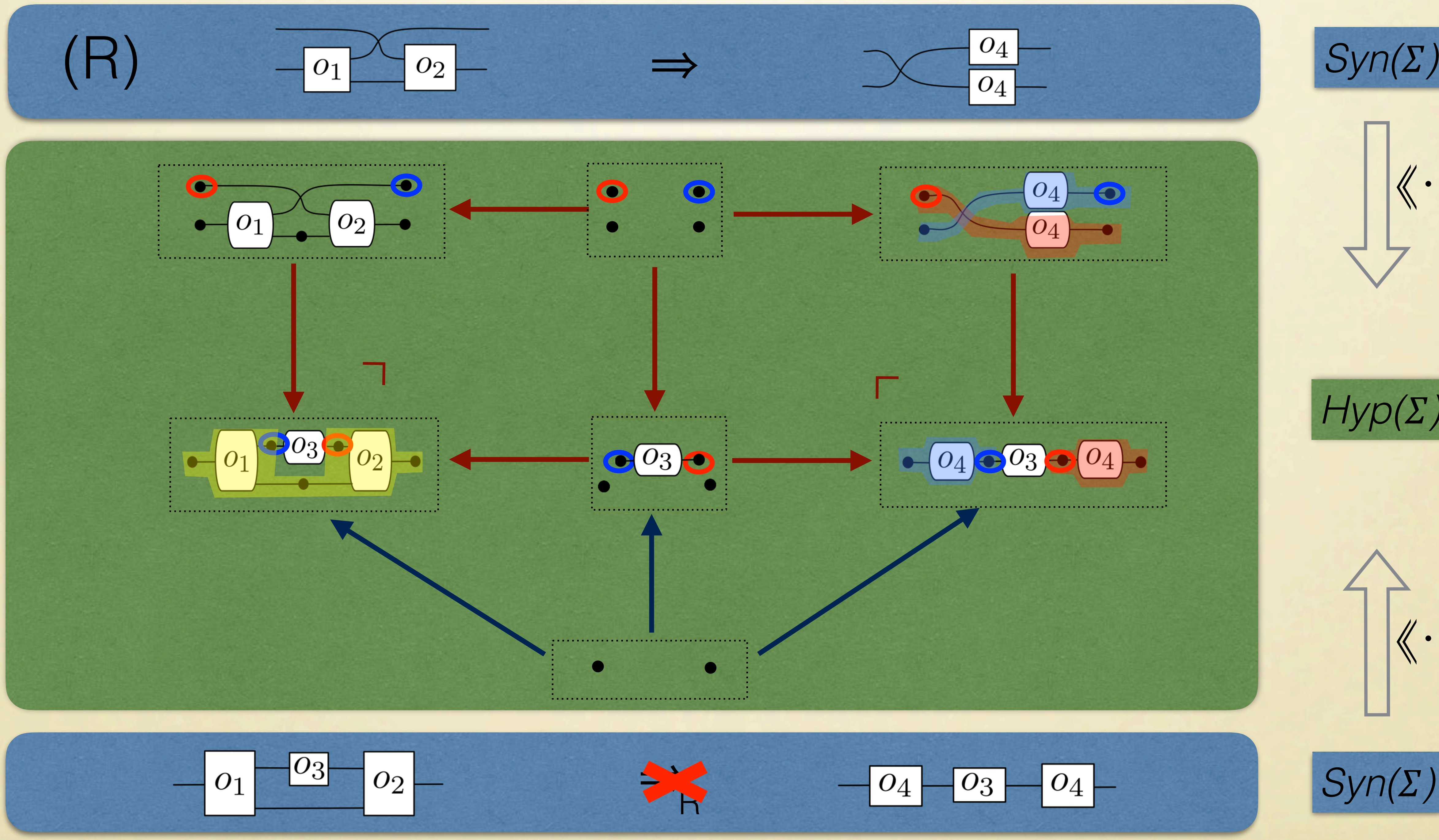
DPO rewriting with interfaces

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



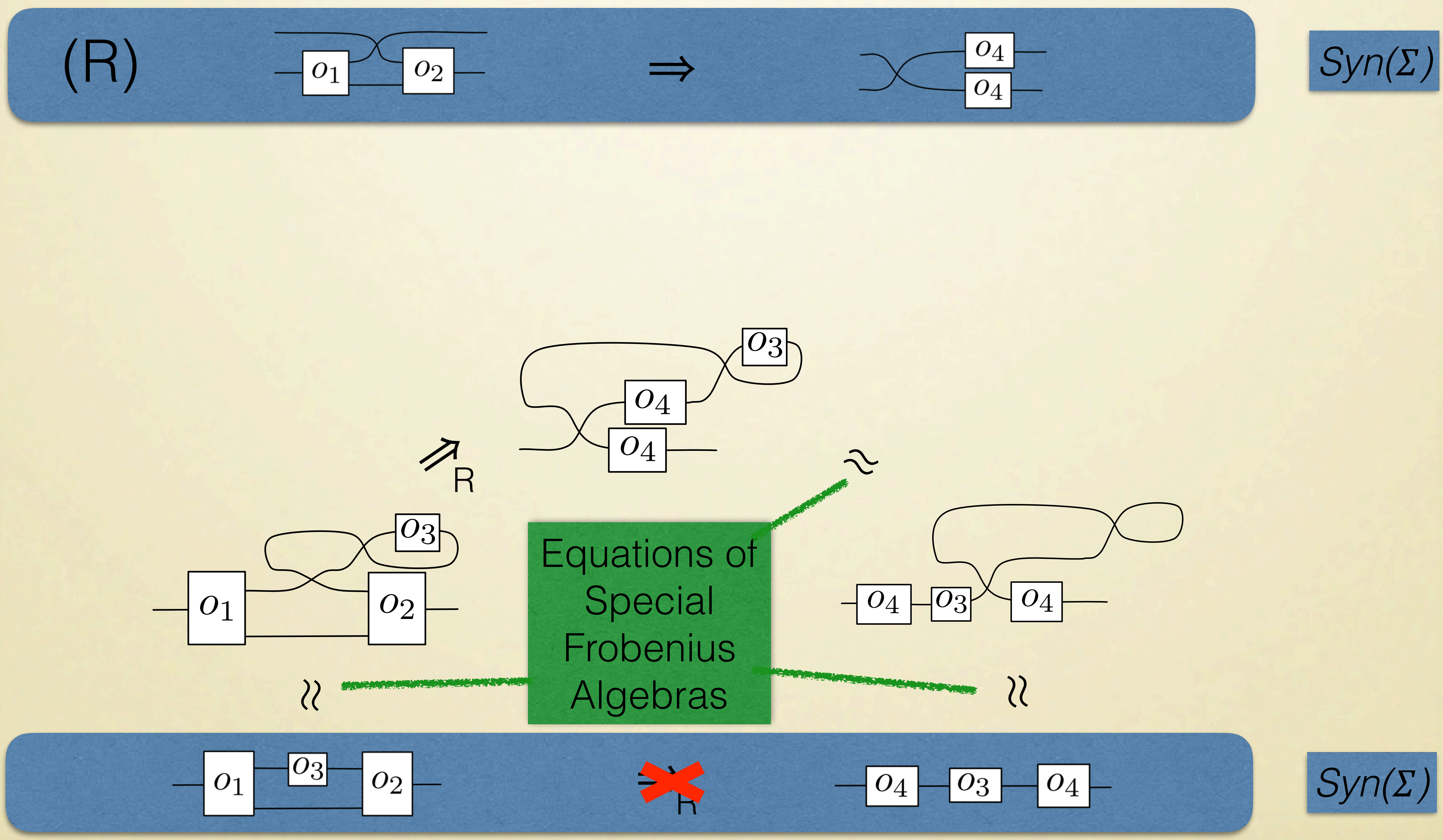
DPO rewriting can be unsound

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

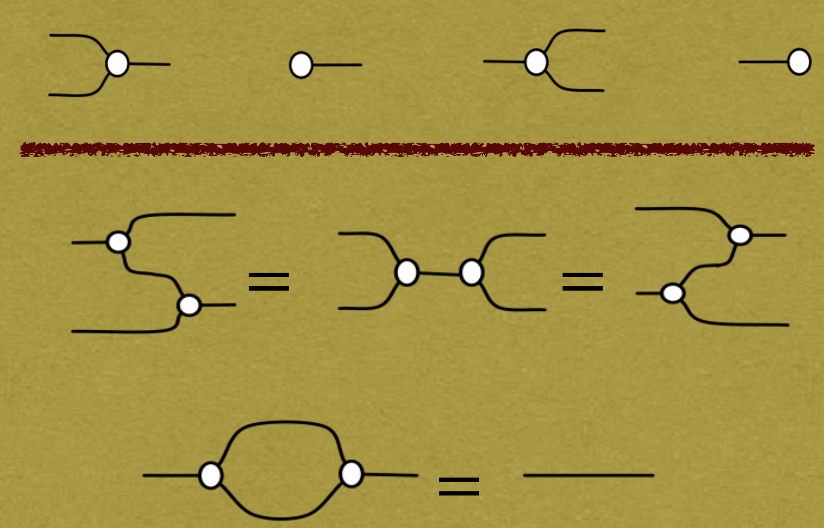


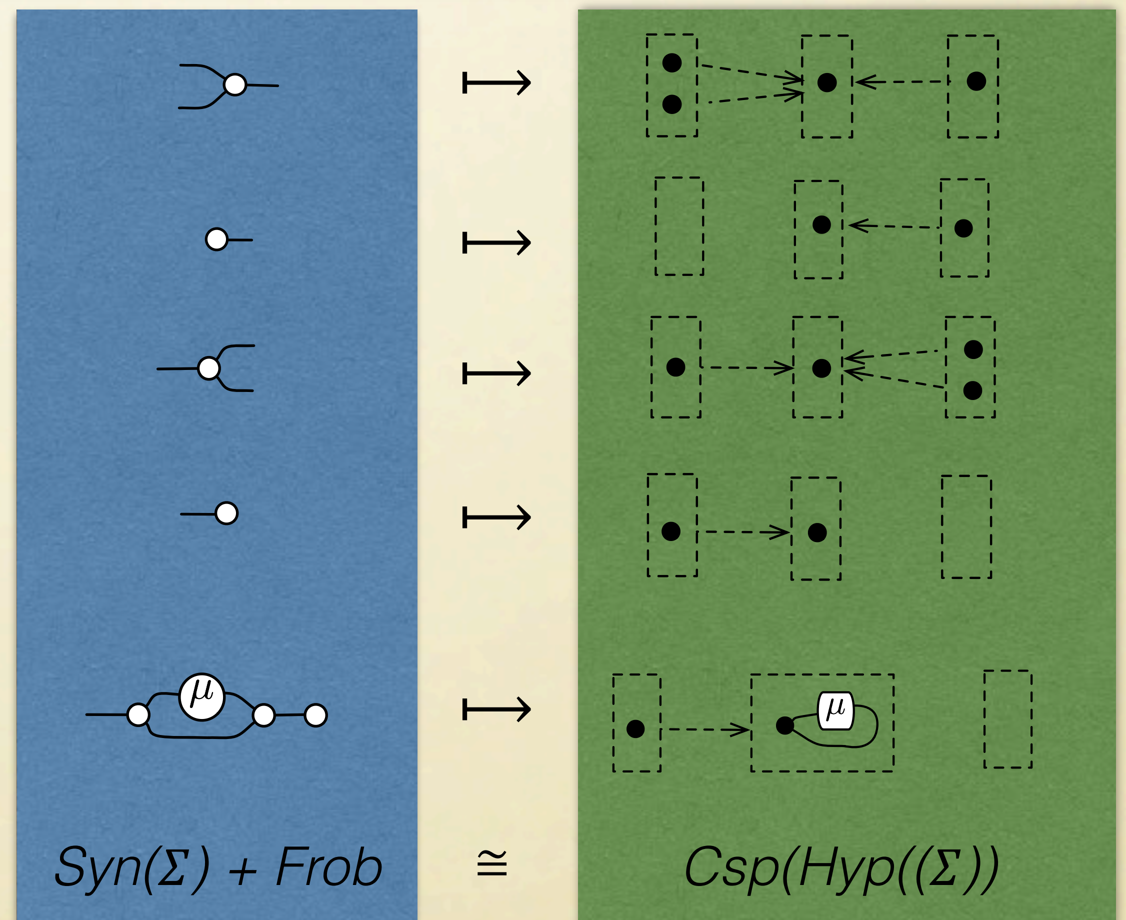
Frobenius makes DPO rewriting sound

Theorem I

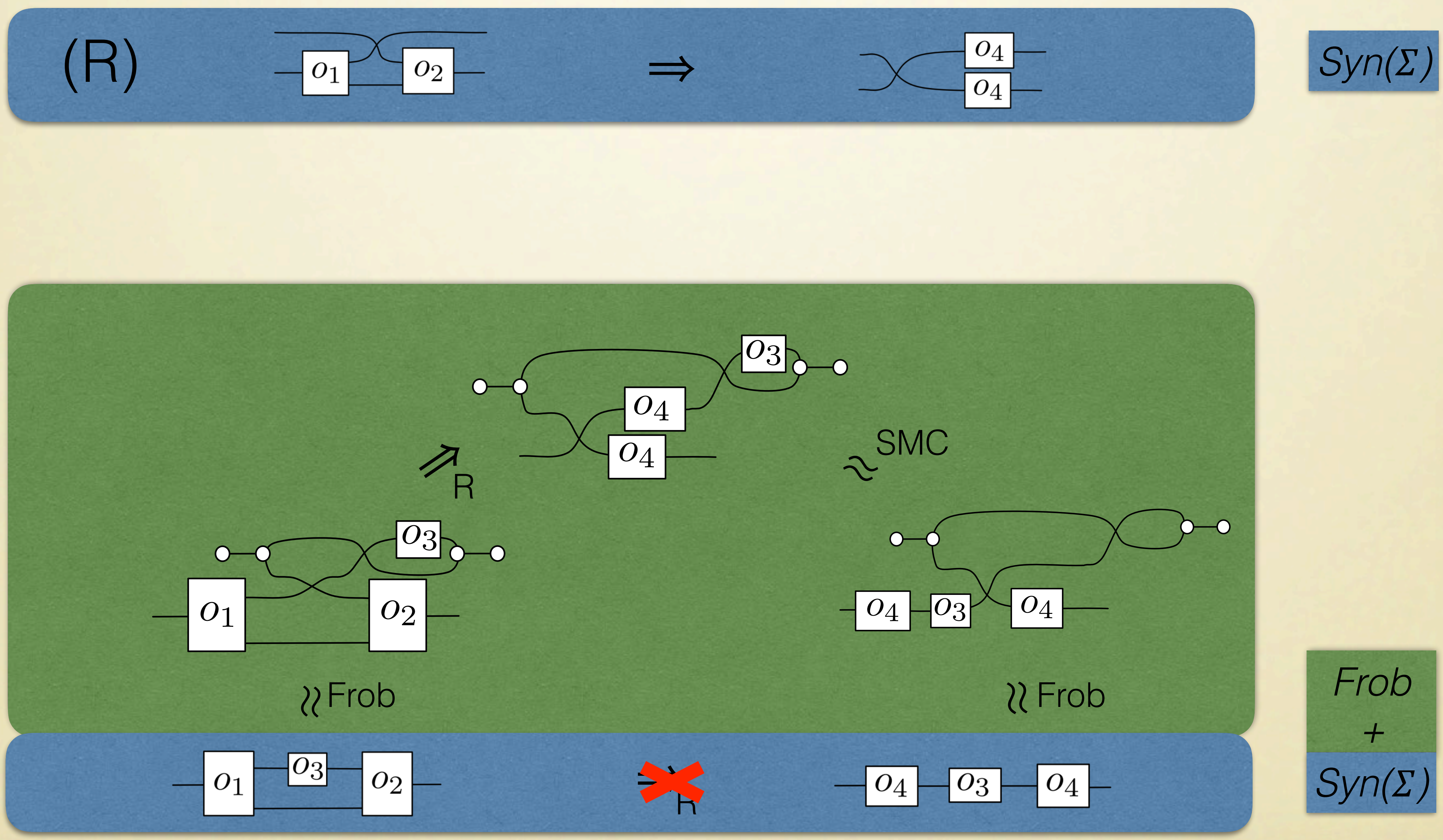
DPO rewriting with interfaces is *sound and complete*
for symmetric monoidal categories
with a chosen *special Frobenius structure*.

prop of special Frobenius algebras

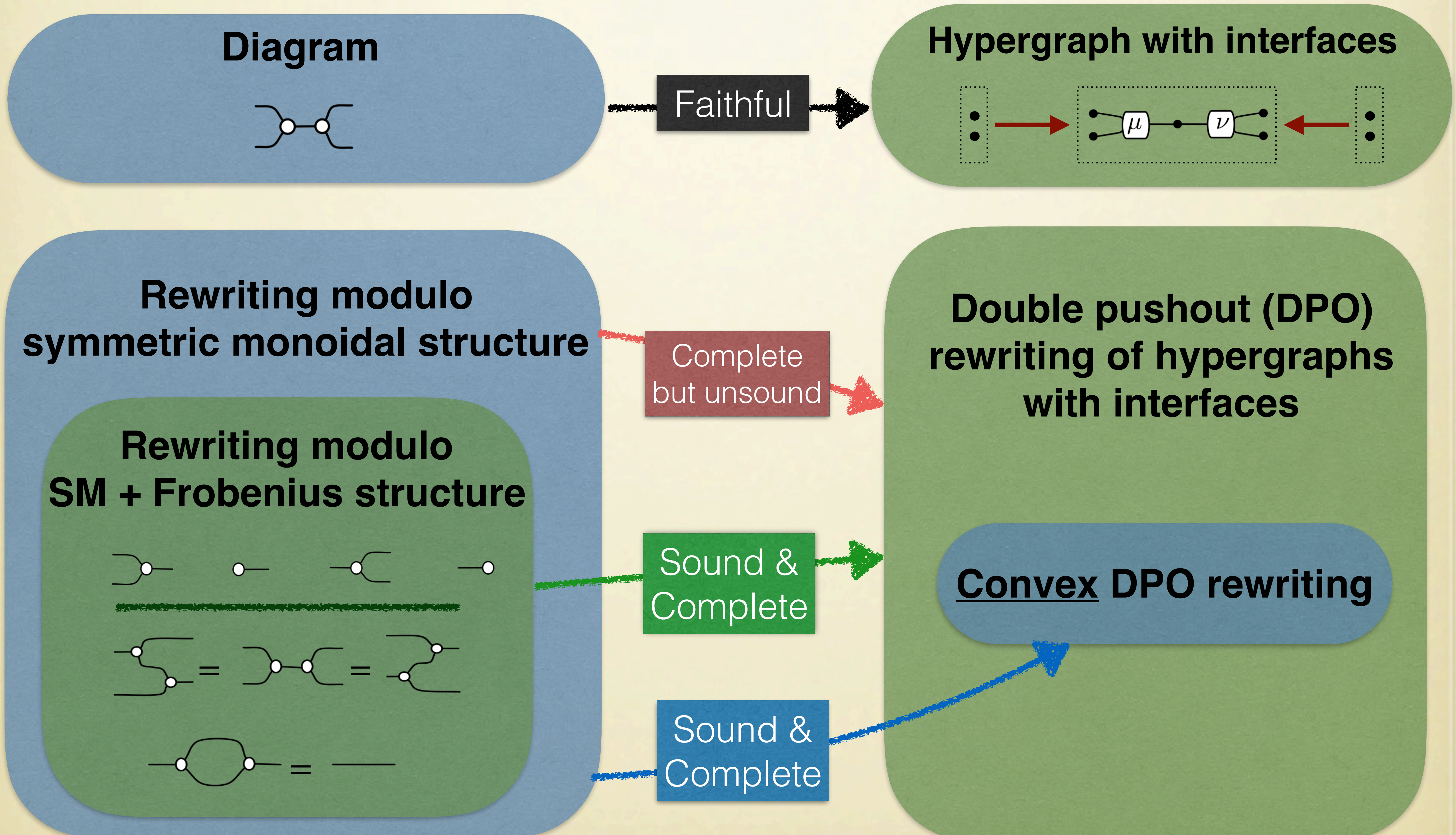




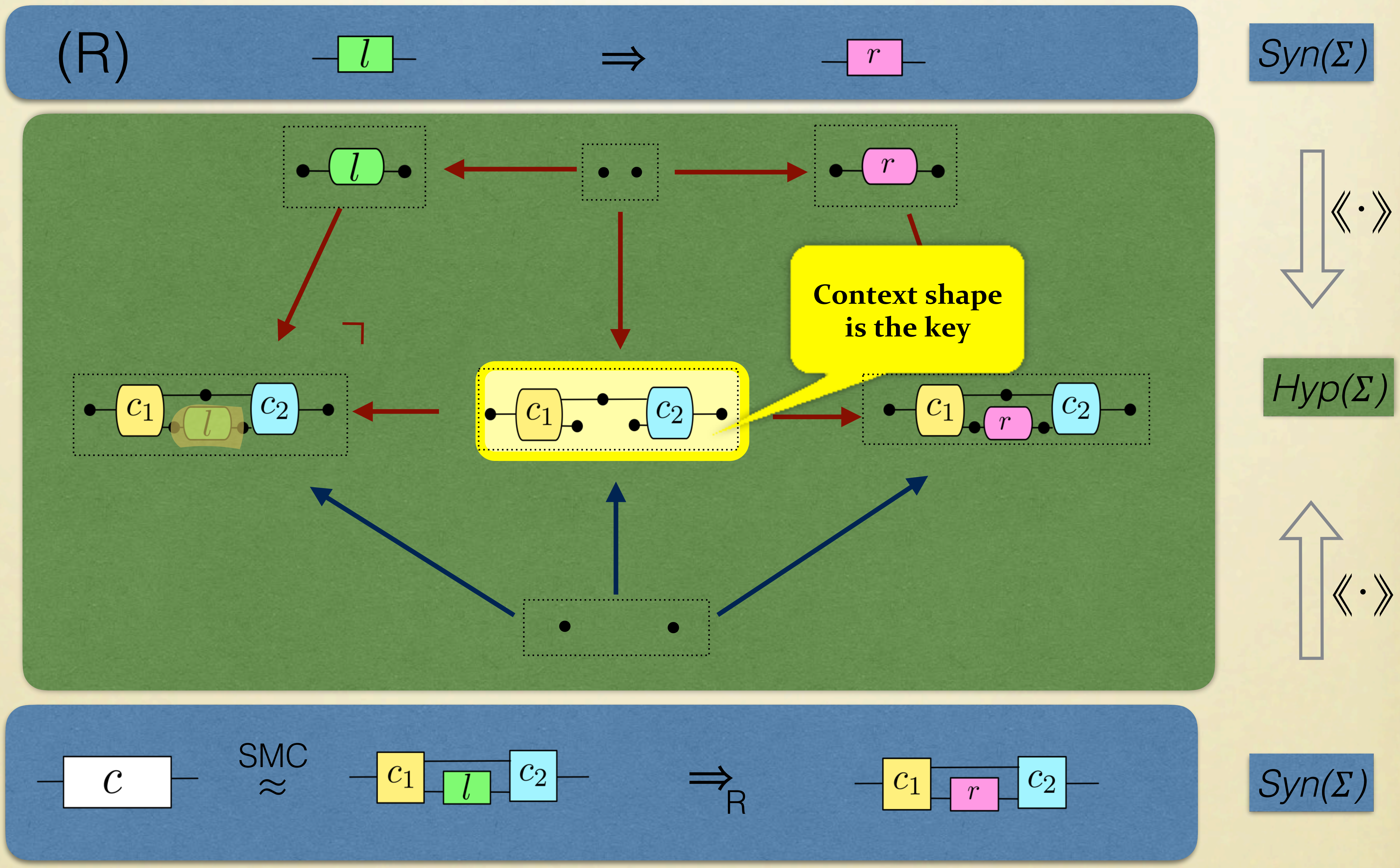
Frobenius makes DPO rewriting sound



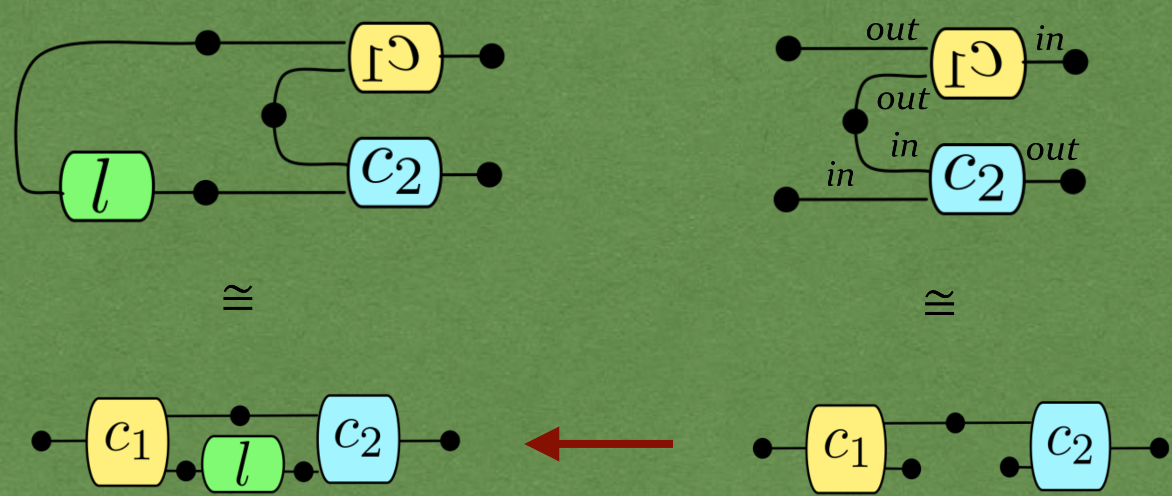
Where we are, so far



How does sound DPO rewriting look like?



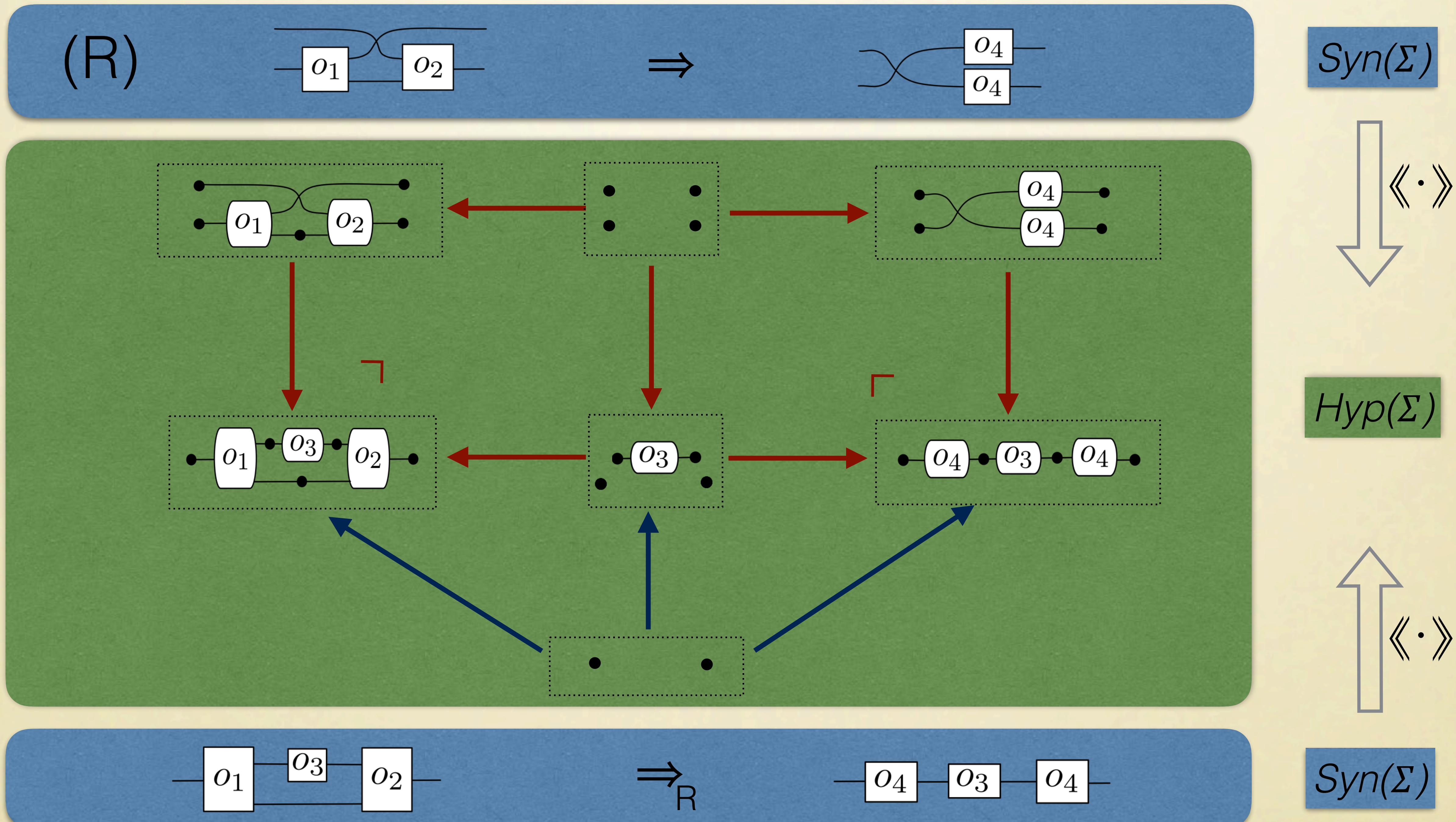
How does sound DPO rewriting look like?



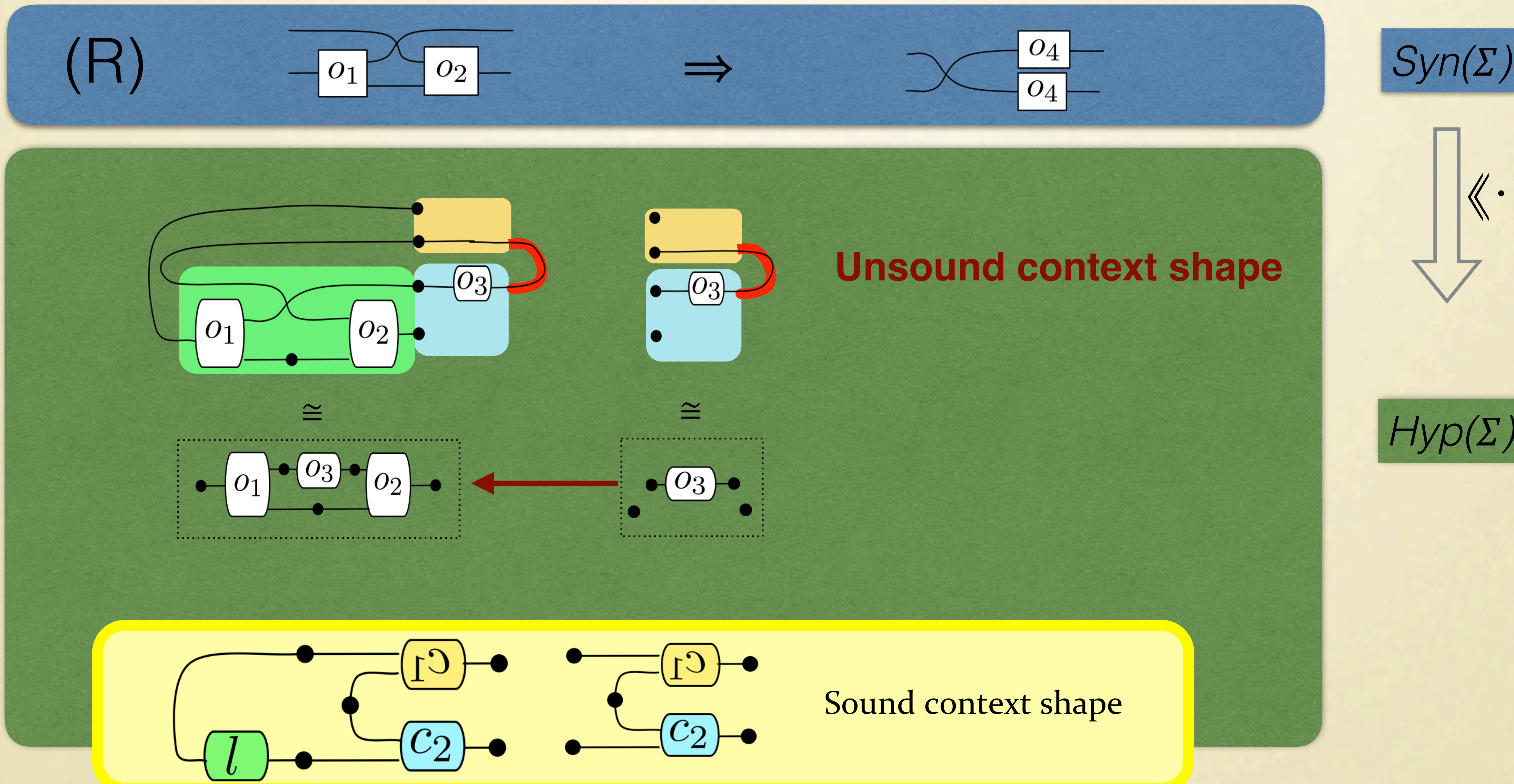
Leading Intuition
a rewriting steps
is sound
iff
the rewriting context
has this shape

Hyp(Σ)

Back to the soundness counterexample



Back to the soundness counterexample



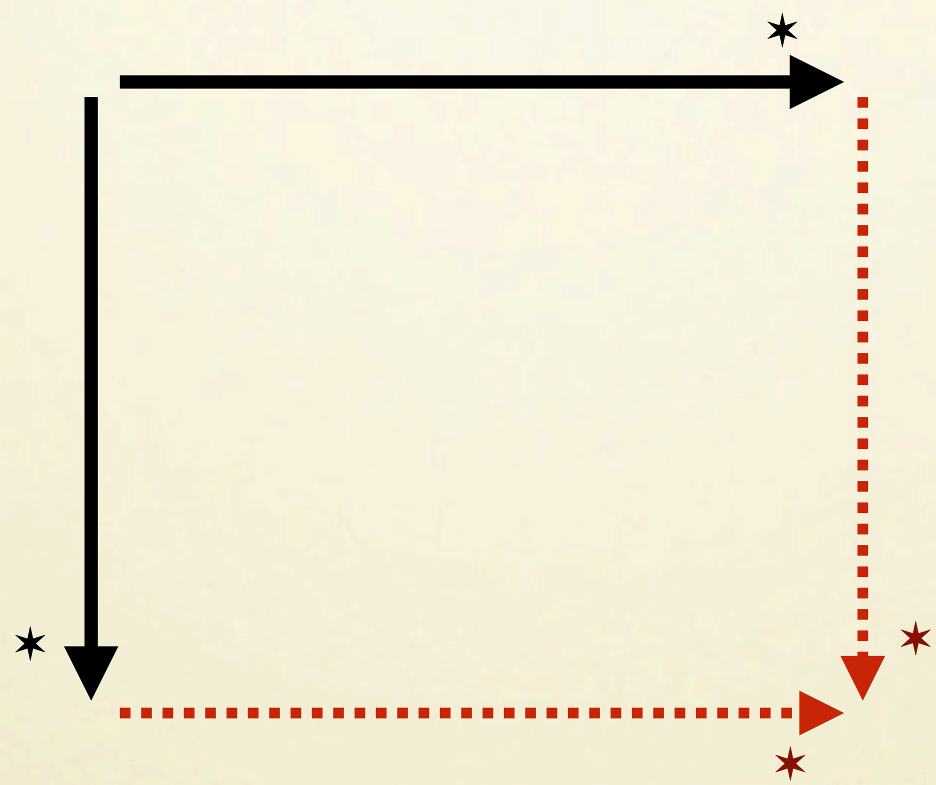
Convex DPO rewriting is sound

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.

Decidability of Confluence

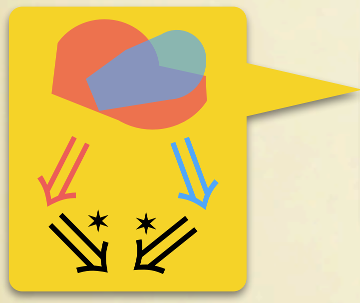
Confluence, abstractly



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

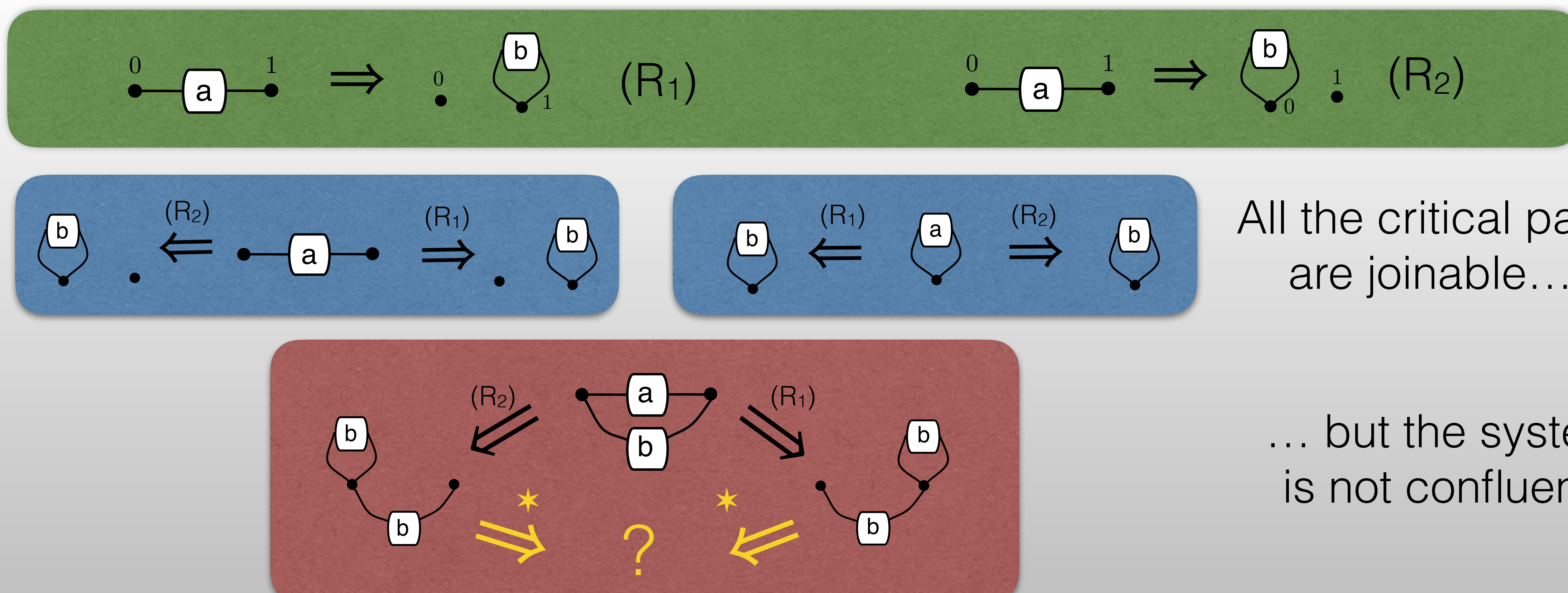
Decidability of Confluence

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



All the critical pairs are joinable ⟺ The system is confluent (Knuth-Bendix)

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

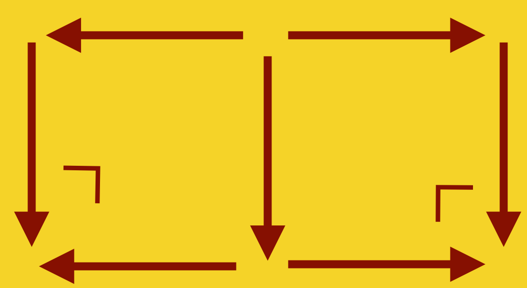


All the critical pairs are joinable…

… but the system is not confluent.

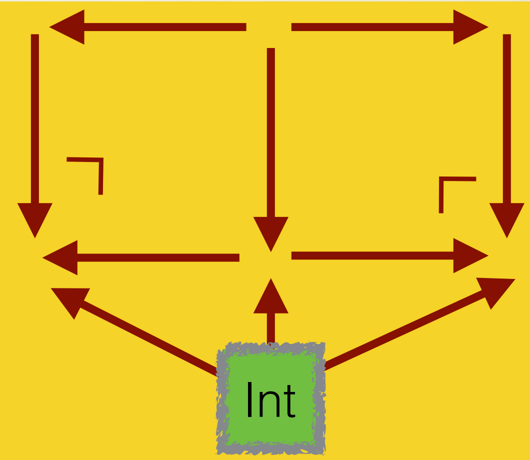
Interfaces to the Rescue

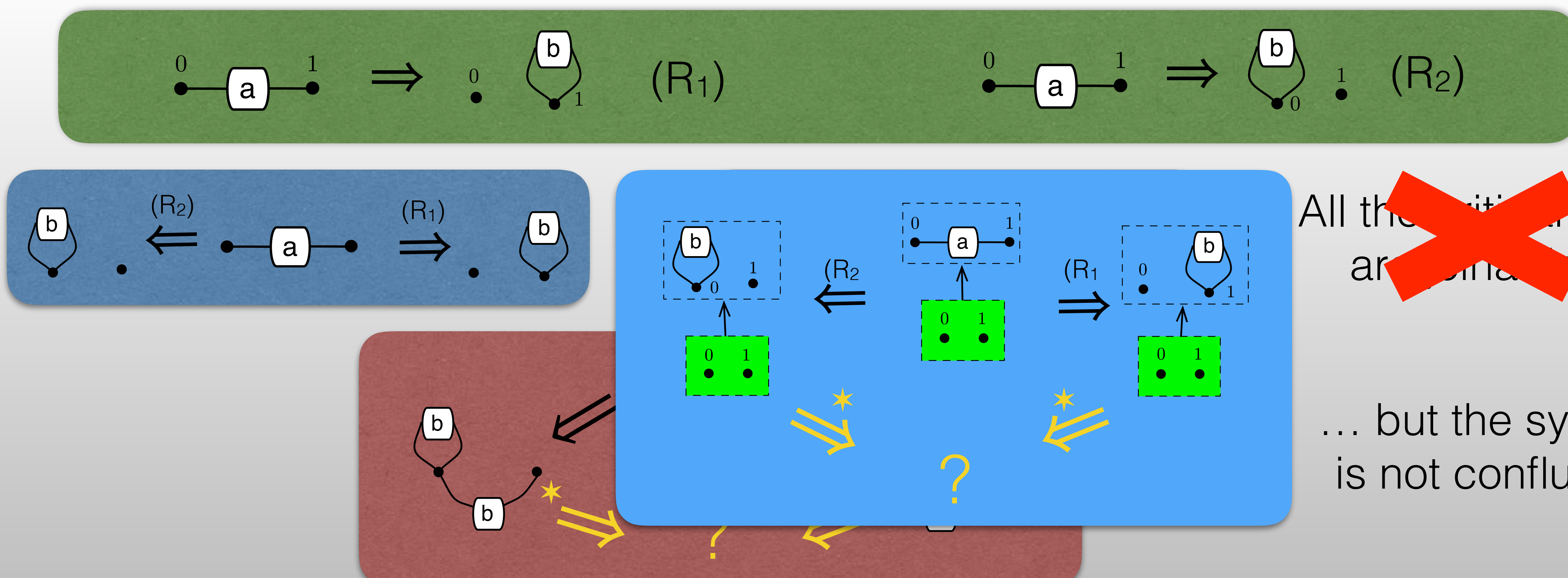
DPO rewriting



VS

DPO rewriting with interfaces





~~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.

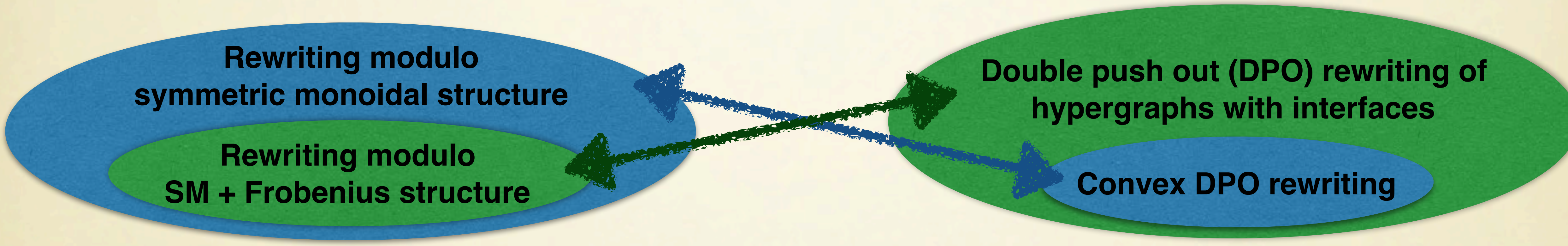
Theorem II bis

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

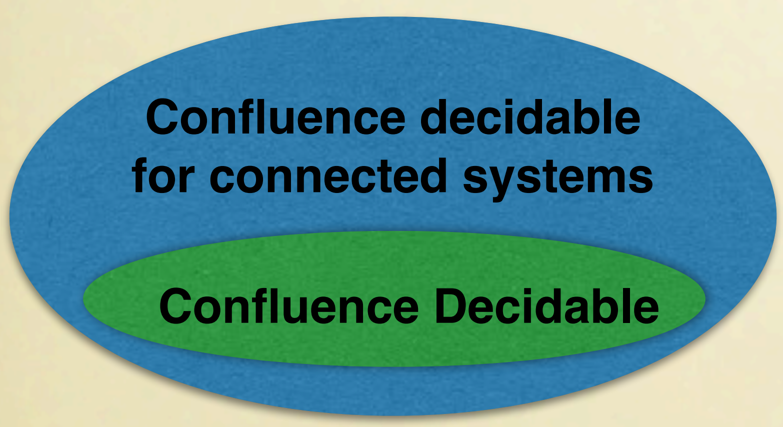
Confluence is decidable for *connected* terminating rewriting systems on such categories.

Conclusions

Adequacy



Confluence



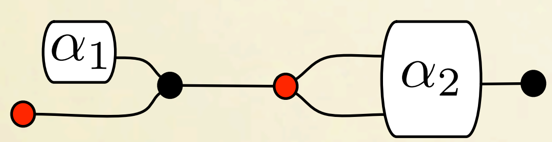
	Terminating term rewriting systems	Terminating DPO-with-interface systems
Confluence for ground objects	***undecidable (Kapur et al.)***	***undecidable (Plump)***
Confluence	***decidable (Knuth-Bendix)***	***decidable***

What’s next

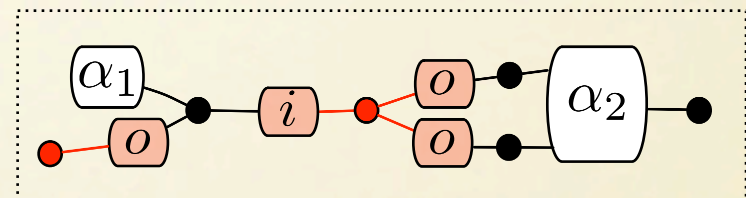
Rewriting modulo *multiple* Frobenius structures

Each Frobenius structure gets absorbed by a different node sort

Syntax



Bipartite hypergraph (kind of)



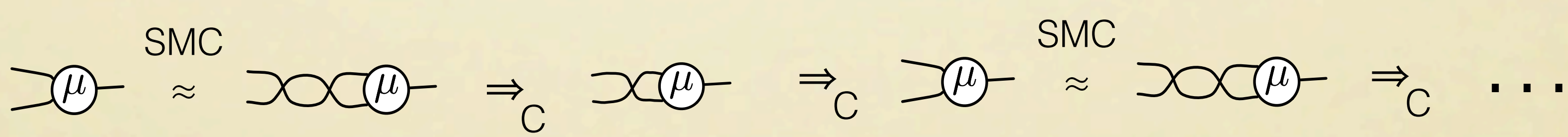
$Syn(\Sigma,A) + \sum_{a\in A} Frob^a \cong Csp(Hyp((\Sigma,A))$

Applications:

Diagrammatic algebras with multiple Frobenius structures (ZX-calculus, calculus of signal flow diagrams, calculus of stateless connectors, graphical linear algebra…)

Nominal calculi (each name is a Frobenius structure)

Termination for commutative operators



Algebraic analysis of bipartite graphs