PAWEL SOBOCINSKI
U. SOUTHAMPTON, U. HAWAI'I AT MĀNOA

PROGRAMMING RECURRENCE RELATIONS

and other compositional stuff…

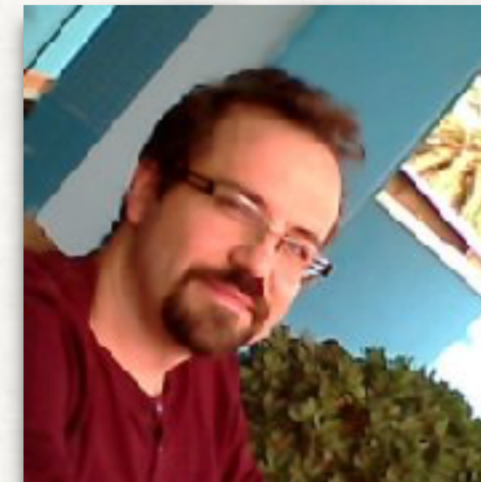
Compositionality Workshop, Simons Institute, 9 December 2016

COLLABORATORS

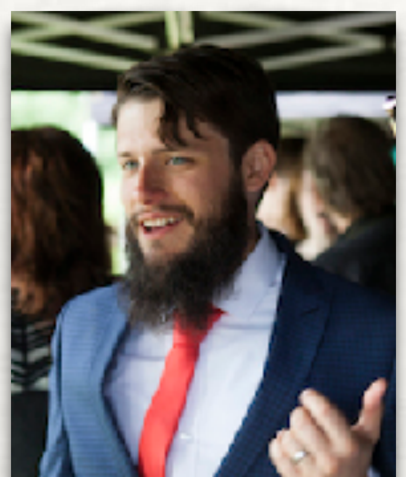

Dusko Pavlovic


Fabio Zanasi


Filippo Bonchi


Fabio Gadducci


Ugo Montanari


Aleks Kissinger


Brendan Fong


Paolo Rapisarda


Bas Westerbaan

Julian Rathke, Owen Stephens, Roberto Bruni, Hernan Melgratti, …

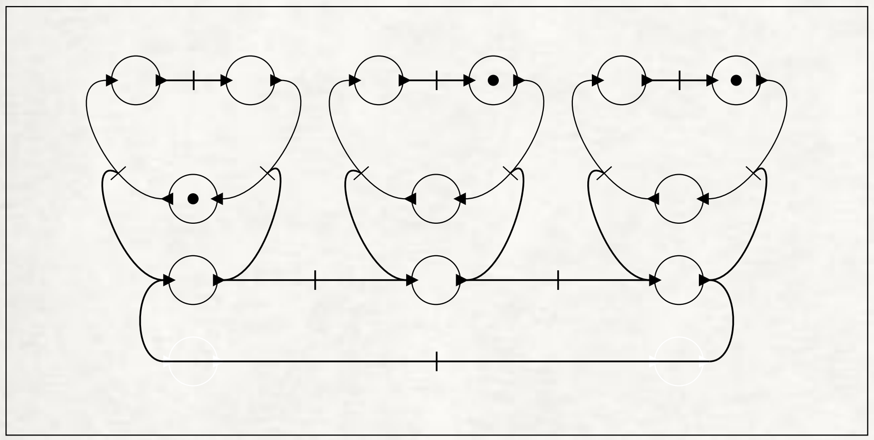
PLAN

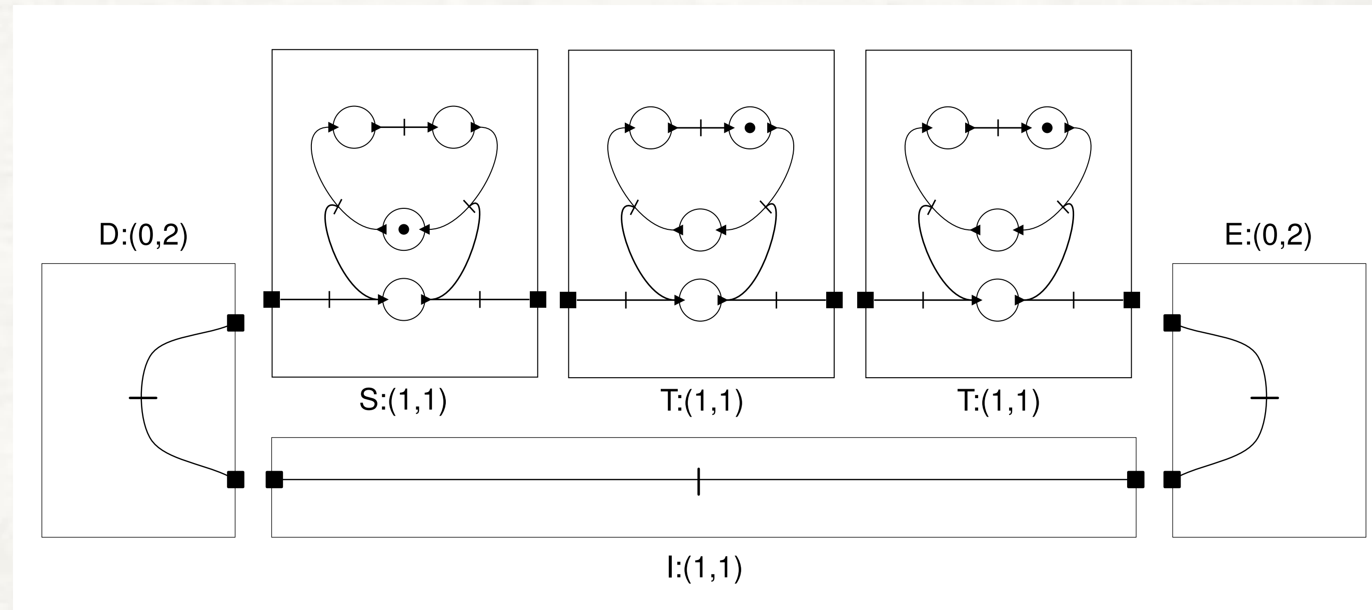
- **Compositional Petri nets**
  - ***verify quicker!***
- Compositional linear algebra
  - *divide by zero!*
- Compositional signal flow graphs
  - *say goodbye to inputs and outputs!*
  - *program recurrence relations!*
- Compositional everything
  - *DPO rewriting*

[[-]] : **Syntax → Semantics**

symmetric monoidal functor,
cf. Lawvere's functorial semantics

COMPOSITIONAL PETRI NETS





$$D\ ;\ ((S\ ;\ T\ ;\ T)\otimes I)\ ;\ E \qquad (\dagger)$$

Fig. 6: A token ring network as a PNB expression

- A compositional approach can be used for
  - faster verification through divide and conquer
  - parametric verification

COMPOSITIONALITY IN PETRI NETS

- **computations** of a net as the arrows of a symmetric monoidal category
  - (Ugo Montanari & Jose Meseguer, Petri nets are monoids, 1980s)
  - inspired a lot of later work on compositional analysis of concurrent computations
- **open petri nets** as the arrows of a symmetric monoidal category
  - Open Petri nets (Baldan, Corradini, Ehrig, Heckel, … 2001–) - glueing nets together along places — as we’ve seen earlier with Blake Pollard’s approach in the stochastic mass action context
  - Petri nets with boundaries — glueing nets together along transitions (Bruni, Melgratti, Montanari, S… 2010—)

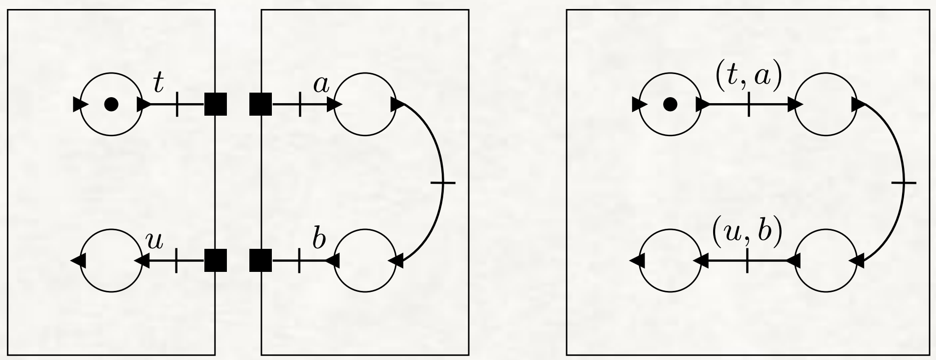
COMPOSITIONAL PETRI NETS

VIA FUNCTORIAL SEMANTICS

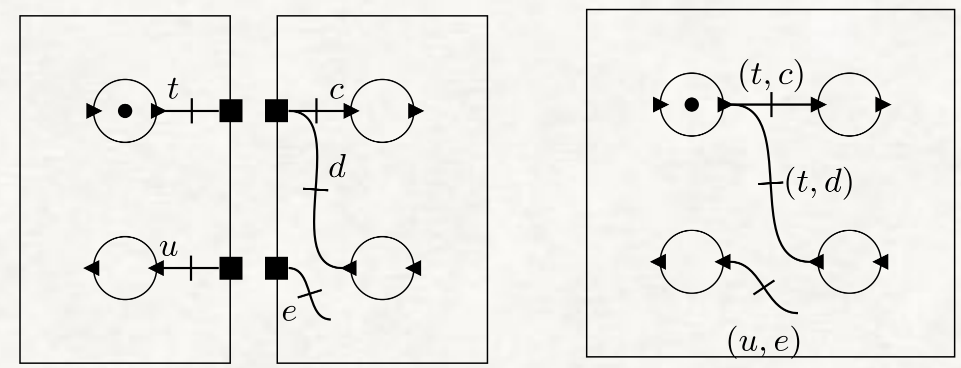
$$[\![-]\!]: \mathbf{Petri} \rightarrow \mathbf{2LTS}$$

Petri and **Aut** are symmetric monoidal categories
and [[-]] is a symmetric monoidal functor

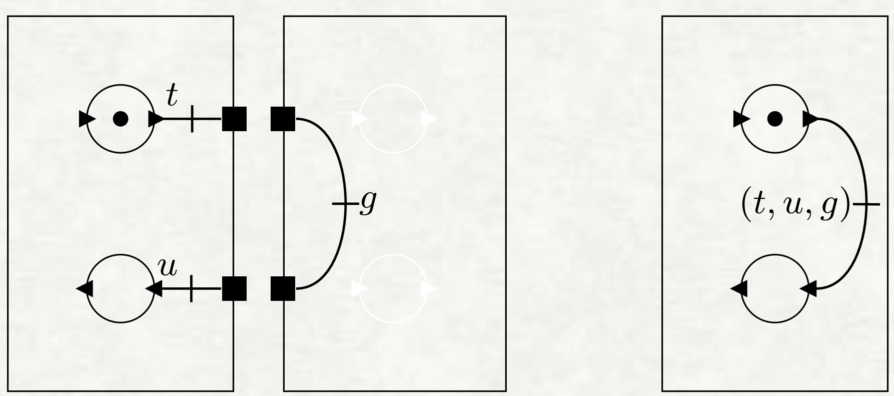
PETRI



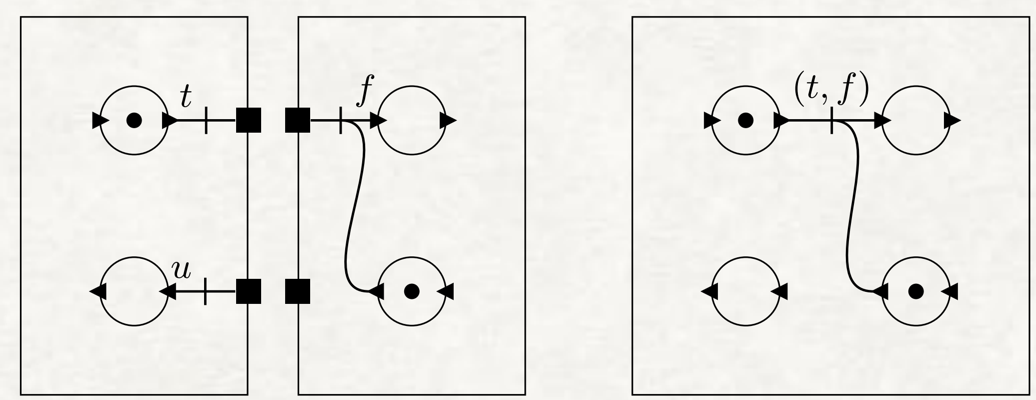
$P : (0,2)$ $Q : (2,0)$ $P\ ;Q : (0,0)$



$P : (0,2)$ $R : (2,0)$ $P\ ;R : (0,0)$



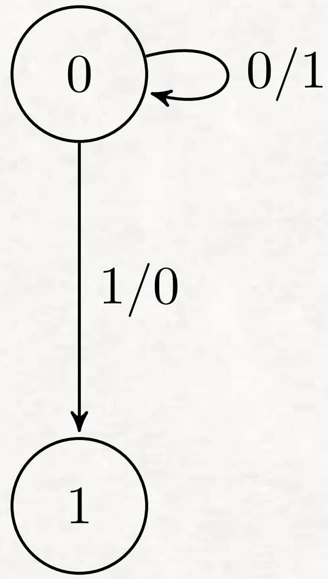
$P : (0,2)$ $S : (2,0)$ $P\ ;S : (0,0)$



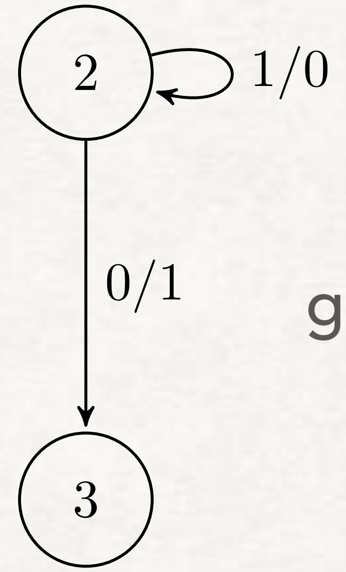
$P : (0,2)$ $T : (2,0)$ $P\ ;T : (0,0)$

2LTS

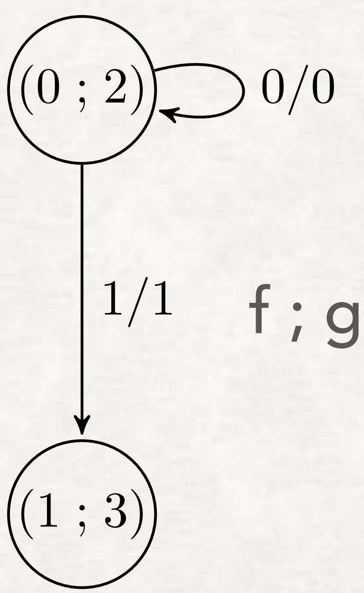
AKA Bob Walters' Span(**Graph**) compositional algebra of transition systems



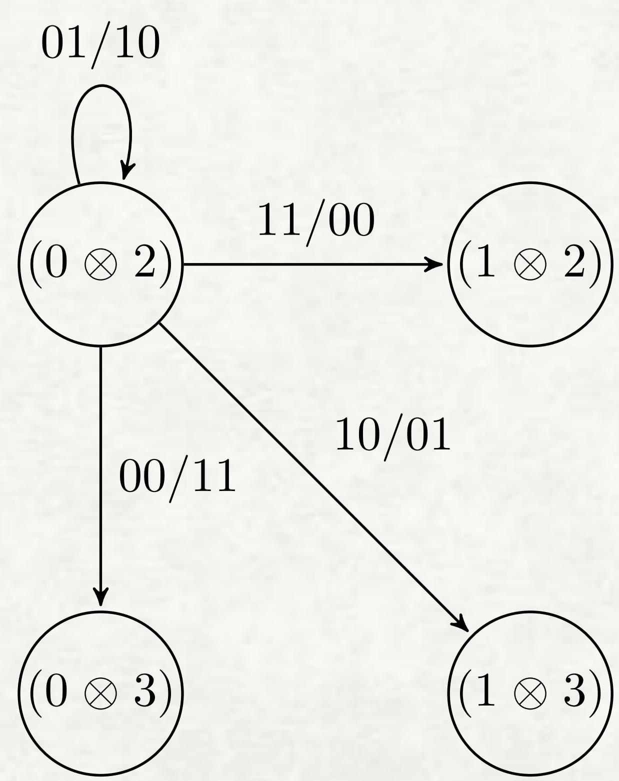
$f: 1 \to 1$



$g: 1 \to 1$

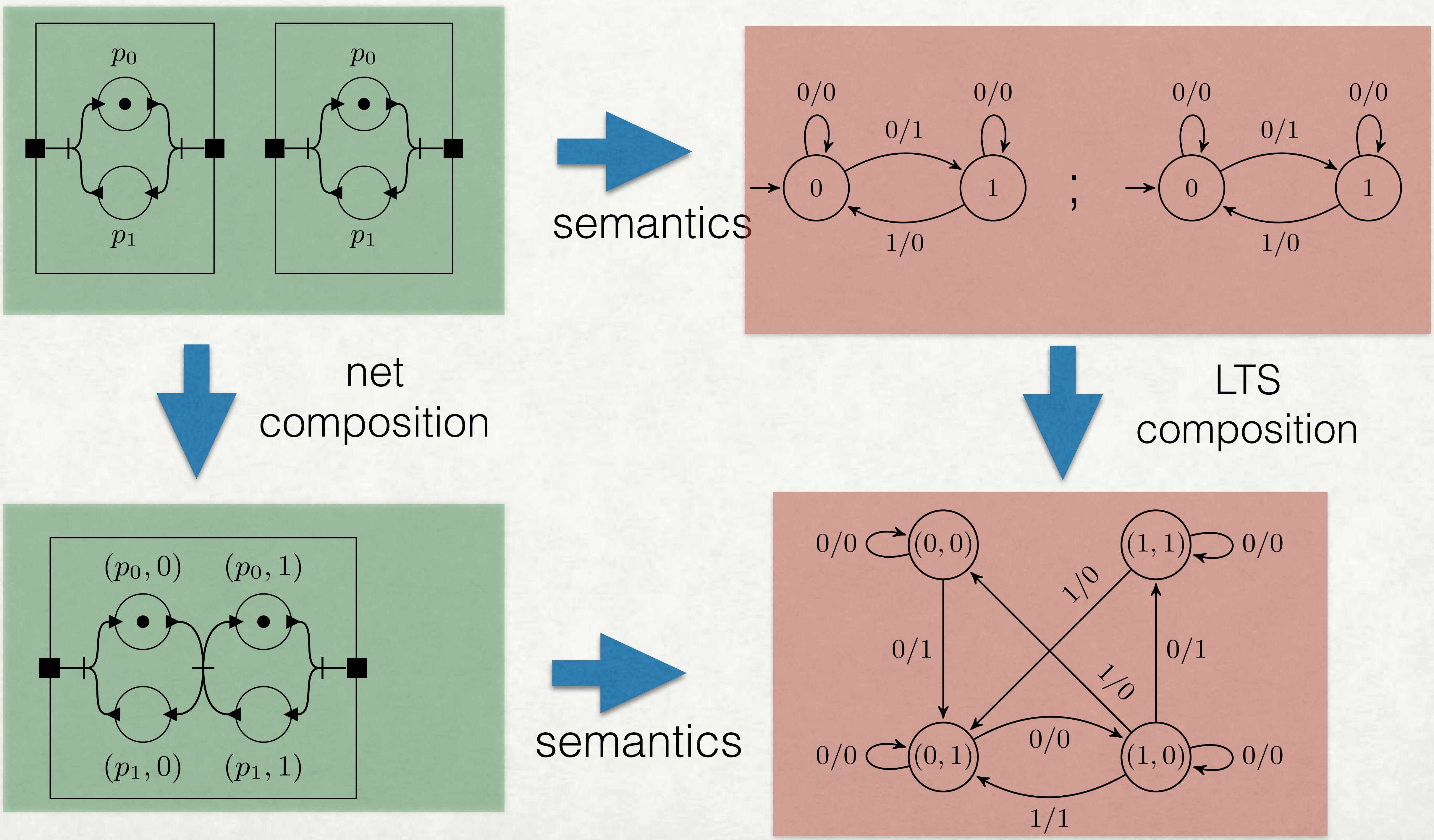


$f ; g: 1 \to 1$



$f \otimes g: 1 \to 1$

[[-]]: Petri → 2LTS



p_0
p_1
p_0
p_1
semantics
0/0
0/1
0/0
0
1
1/0
;
0/0
0/1
0/0
0
1
1/0
net composition
LTS composition
$(p_0, 0)$
$(p_0, 1)$
$(p_1, 0)$
$(p_1, 1)$
semantics
0/0
(0, 0)
(1, 1)
0/0
1/0
0/1
0/1
1/0
0/0
(0, 1)
0/0
(1, 0)
0/0
1/1

THE PROOF IS IN THE PUDDING

- R. Mayr and L. Clemente. *Advanced Automata Minimization*. In PoPL '13.
- F. Bonchi and D. Pous. *Checking NFA Equivalence with Bisimulations up to Congruence*. In PoPL '13.

Penrose tool for reachability checking, compositionally

over	8	31.039	0.008	1.071	**0.003**	3812.00	37.63	141.85	**15.49**
over	32	*M*	*T*	*M*	**0.004**	*M*	*T*	*M*	**15.50**
over	512	*M*	*T*	*M*	**0.003**	*M*	*T*	*M*	**15.52**
over	4096	*M*	*T*	*M*	**0.004**	*M*	*T*	*M*	**16.04**
over	32768	*M*	*T*	*M*	**0.010**	*M*	*T*	*M*	**20.09**
dac	8	**0.001**	0.003	0.017	0.002	**7.51**	33.28	38.85	14.68
dac	32	**0.001**	0.005	0.028	0.002	**7.50**	34.50	49.45	14.68
dac	512	0.005	*T*	255.847	**0.003**	20.62	*T*	6012.00	**14.80**
dac	4096	2.462	*T*	*M*	**0.008**	166.07	*T*	*M*	**15.92**
dac	32768	*T*	*T*	*M*	**0.053**	*T*	*T*	*M*	**24.24**
philo	8	**0.002**	0.003	0.016	0.005	**8.86**	33.22	38.54	17.34
philo	32	*M*	**0.003**	0.017	0.005	*M*	33.53	40.87	**17.35**
philo	512	*M*	0.020	0.086	**0.008**	*M*	41.69	290.77	**17.39**
philo	4096	*M*	7.853	*M*	**0.019**	*M*	172.76	*M*	**17.58**
philo	32768	*M*	*T*	*M*	**1.014**	*M*	*T*	*M*	**21.32**
iter-choice∗	8	0.006	5.025	19.062	**0.002**	36.37	465.17	1570.64	**14.64**
iter-choice∗	32	*M*	*T*	*T*	**0.003**	*M*	*T*	*T*	**14.64**
iter-choice∗	512	*M*	*T*	*T*	**0.006**	*M*	*T*	*T*	**14.71**
iter-choice∗	4096	*M*	*T*	*T*	**0.028**	*M*	*T*	*T*	**15.22**
iter-choice∗	32768	*M*	*T*	*T*	**1.644**	*M*	*T*	*T*	**20.15**
replicator∗	8	**0.001**	/	0.016	0.002	**7.51**	/	38.15	14.72
replicator∗	32	**0.001**	/	0.017	0.002	**7.51**	/	39.41	14.72
replicator∗	512	**0.002**	/	1.023	0.009	**14.72**	/	77.87	14.82
replicator∗	4096	0.062	/	64.046	**0.056**	86.85	/	3256.00	**15.72**
replicator∗	32768	91.646	/	*M*	**3.660**	1524.50	/	*M*	**21.90**
counter∗	8	**0.001**	/	/	0.054	**7.51**	/	/	19.98
counter∗	16	**0.000**	/	/	4.646	**7.51**	/	/	27.98
counter∗	32	**0.001**	/	/	52.072	**7.51**	/	/	50.25
counter∗	64	**0.001**	/	/	*T*	**8.60**	/	/	*T*
hartstone	8	**0.001**	0.002	/	0.062	**7.51**	33.17	/	20.05
hartstone	16	**0.001**	0.003	/	5.073	**7.51**	33.20	/	24.01
hartstone	32	**0.001**	0.002	/	64.062	**7.51**	33.22	/	38.70
hartstone	64	**0.001**	0.002	/	*T*	**8.54**	33.46	/	*T*
token-ring	8	**0.001**	0.007	0.071	1.085	**7.51**	39.96	89.81	20.89
token-ring	16	**1.824**	*T*	*T*	16.038	318.08	*T*	*T*	**29.41**
token-ring	32	*M*	*T*	*T*	**165.461**	*M*	*T*	*T*	**50.19**
token-ring	64	*M*	*T*	*T*	*T*	*M*	*T*	*T*	*T*

BIBLIOGRAPHY

- Owen Stephens, *Compositional Specification and Reachability Checking of Net Systems*, PhD thesis, U. Southampton, 2015
- P. Sobocinski, *Representations of Petri net interactions*, CONCUR 2010
- R. Bruni, H. Melgratti, U. Montanari, *A Connector Algebra for P/T Nets Interactions*, CONCUR 2011
- R. Bruni, H. Melgratti, U. Montanari, P. Sobocinski, *Connector Algebras for C/E and P/T Nets' Interactions*, Log. Meth. Comput. Sci., 2013
- P. Sobocinski, *Nets, relations and linking diagrams*, CALCO 2013
- P. Sobocinski and O. Stephens, *A programming language for spatial distribution of net systems*, Petri Nets `14
- J. Rathke, P. Sobocinski and O. Stephens, *Compositional reachability in Petri nets*, Reachability Problems `14

PLAN

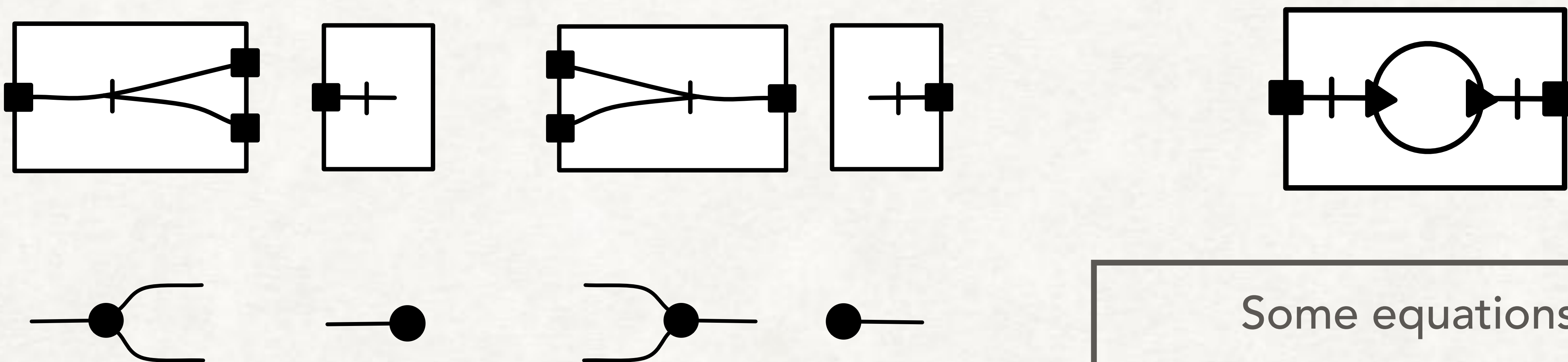
- Compositional Petri nets
  - *verify quicker!*
- **Compositional linear algebra**
  - ***divide by zero!***
- Compositional signal flow graphs
  - *say goodbye to inputs and outputs!*
  - *program recurrence relations!*
- Compositional everything
  - *DPO rewriting*

[[-]] : **Syntax → Semantics**

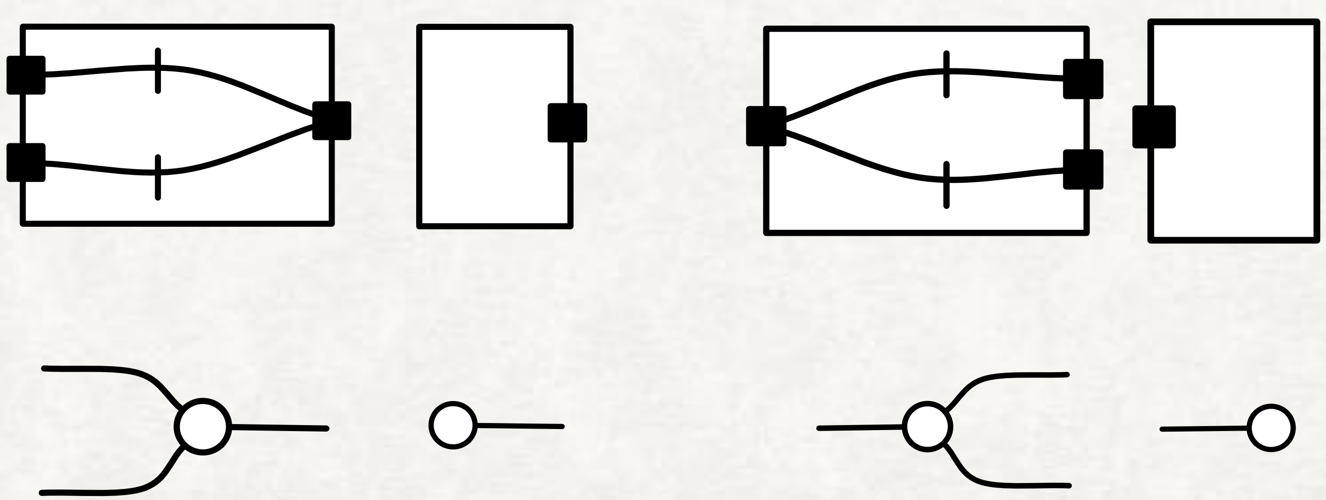
PRESENTING **PETRI** WITH GENERATORS AND RELATIONS

P.S. NETS, RELATIONS AND LINKING DIAGRAMS, CALCO `13

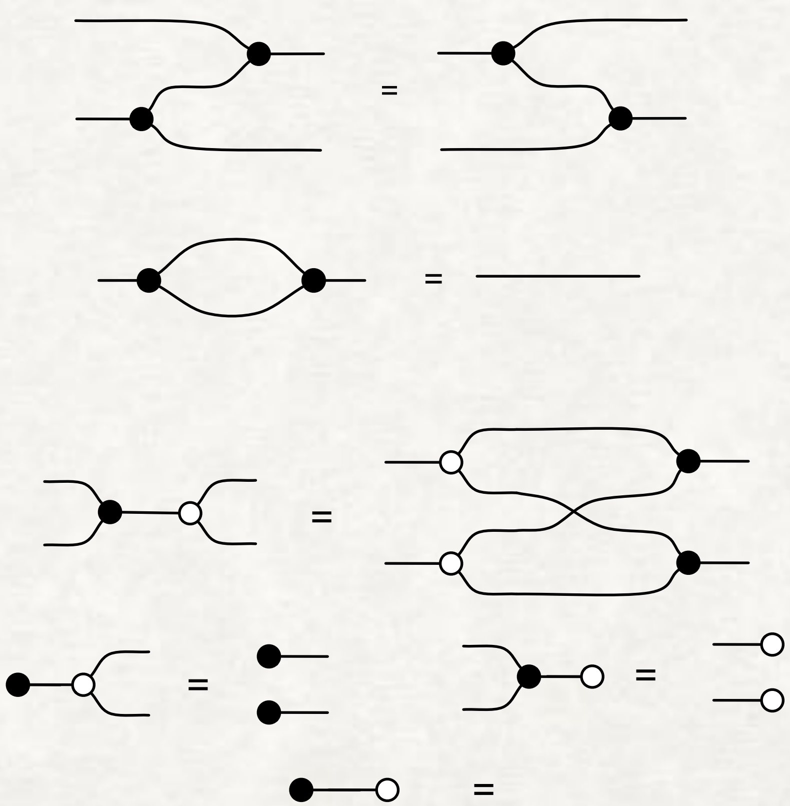
“a transition can connect to more than one place”



“a place can connect to more than one transition”

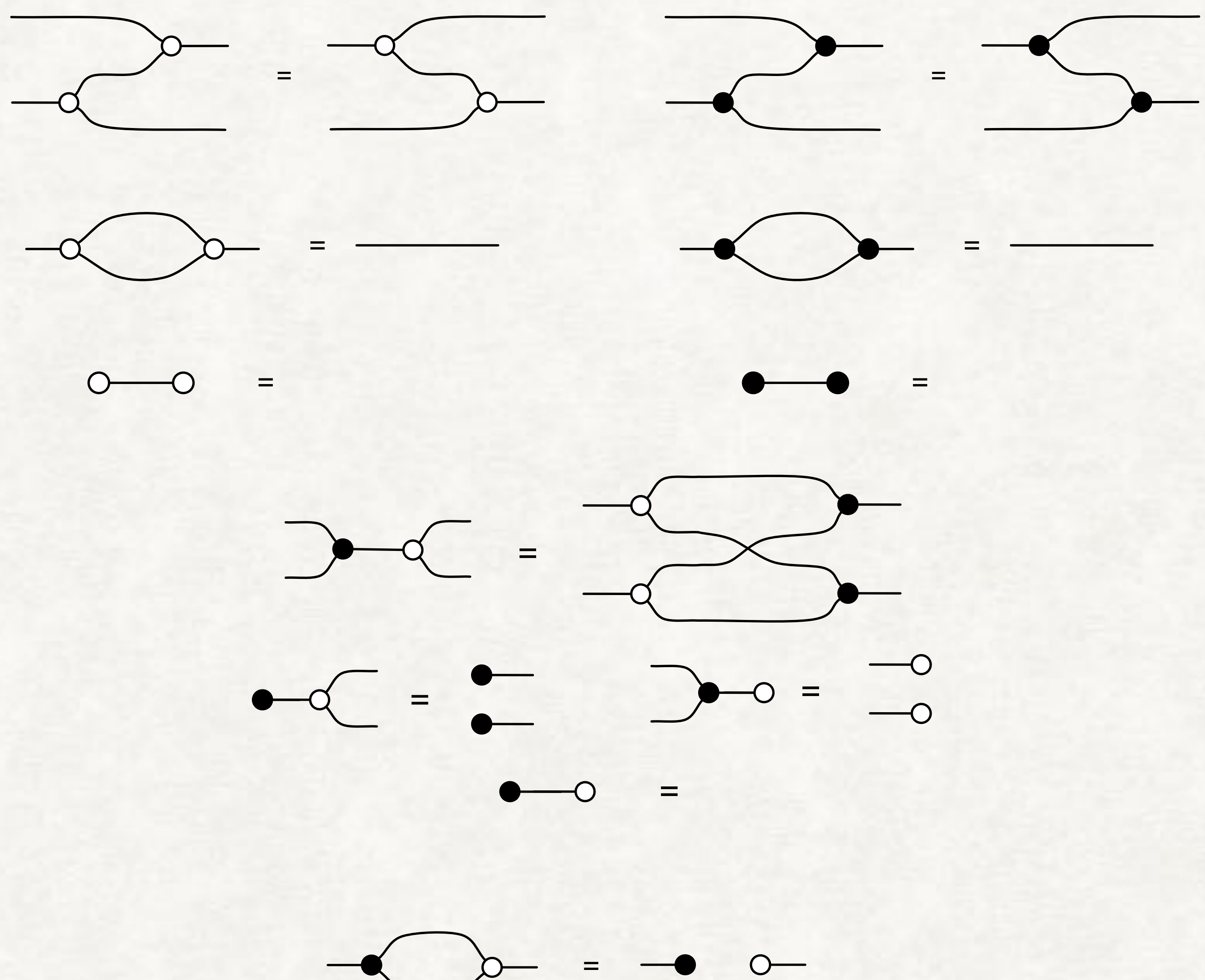


Some equations

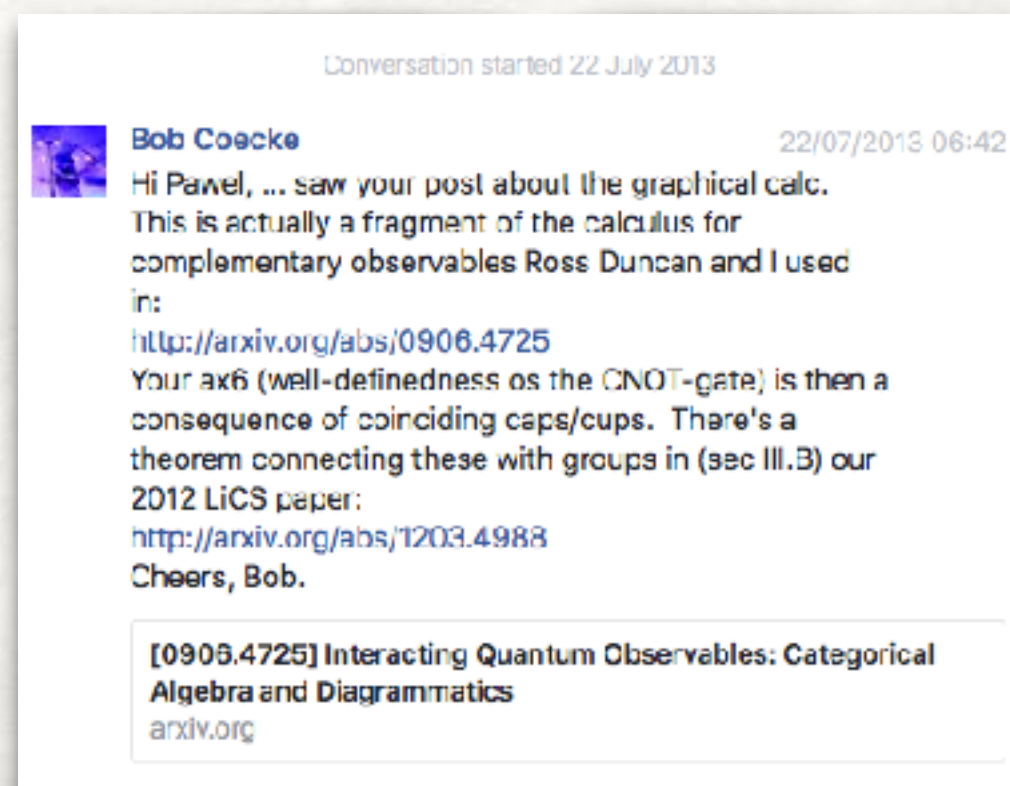


IN A MORE PERFECT, SYMMETRIC WORLD

F. Bonchi, P.S., F. Zanasi, Interacting Hopf Algebras are Frobenius, FoSSaCS `14



What is this thing?



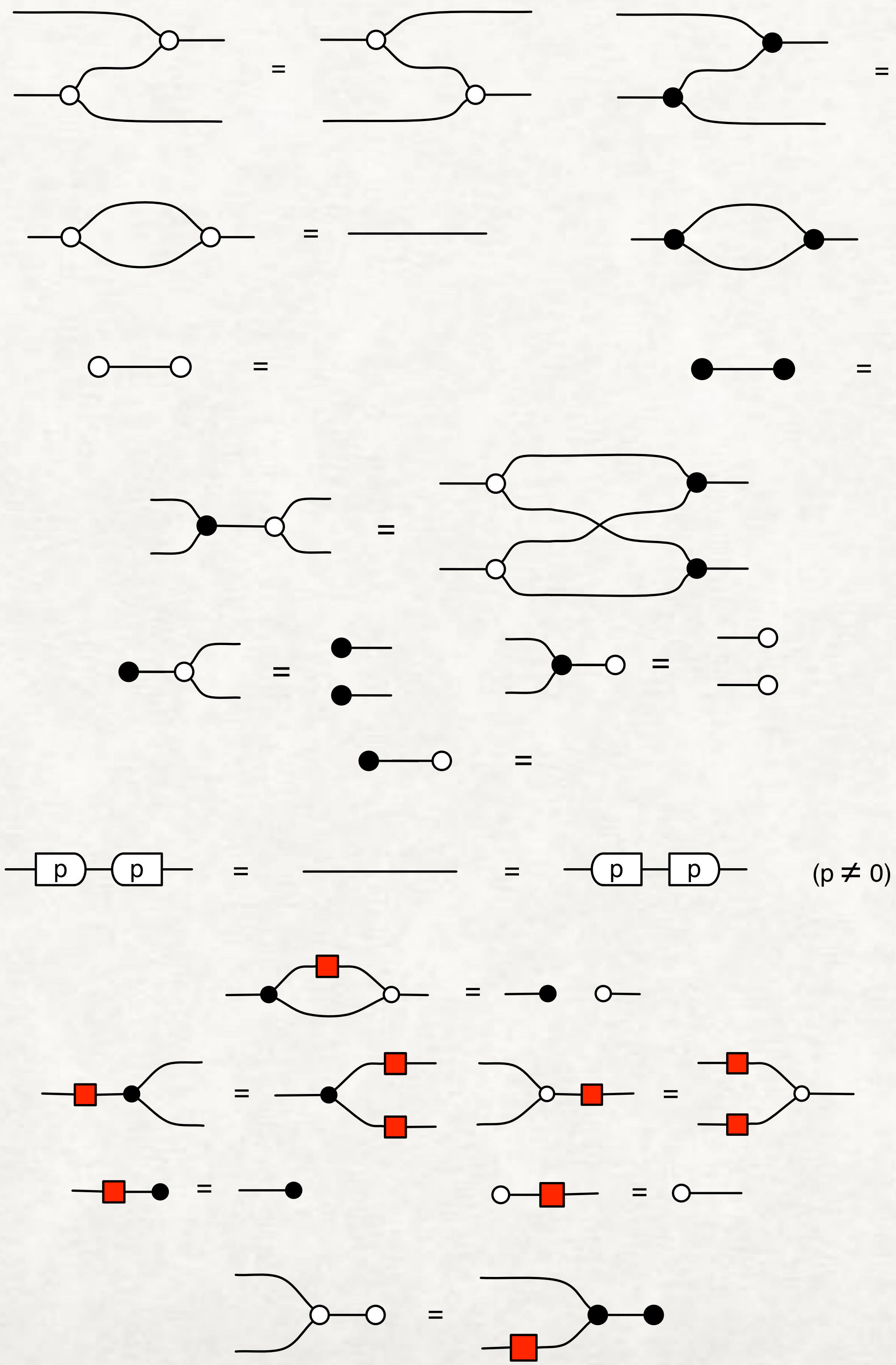
Conversation started 22 July 2013

Bob Coecke 22/07/2013 06:42

Hi Pawel, ... saw your post about the graphical calc.
This is actually a fragment of the calculus for complementary observables Ross Duncan and I used in:
http://arxiv.org/abs/0906.4725
Your ax6 (well-definedness os the CNOT-gate) is then a consequence of coinciding caps/cups. There's a theorem connecting these with groups in (sec III.3) our 2012 LiCS paper:
http://arxiv.org/abs/1203.4988
Cheers, Bob.

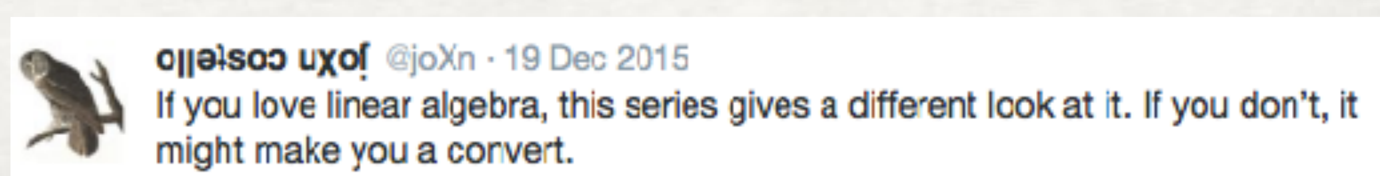
[0906.4725] Interacting Quantum Observables: Categorical Algebra and Diagrammatics
arxiv.org

GRAPHICAL LINEAR ALGEBRA



[[-]] : IH → $\mathbf{LinRel}_{\mathbb{Q}}$

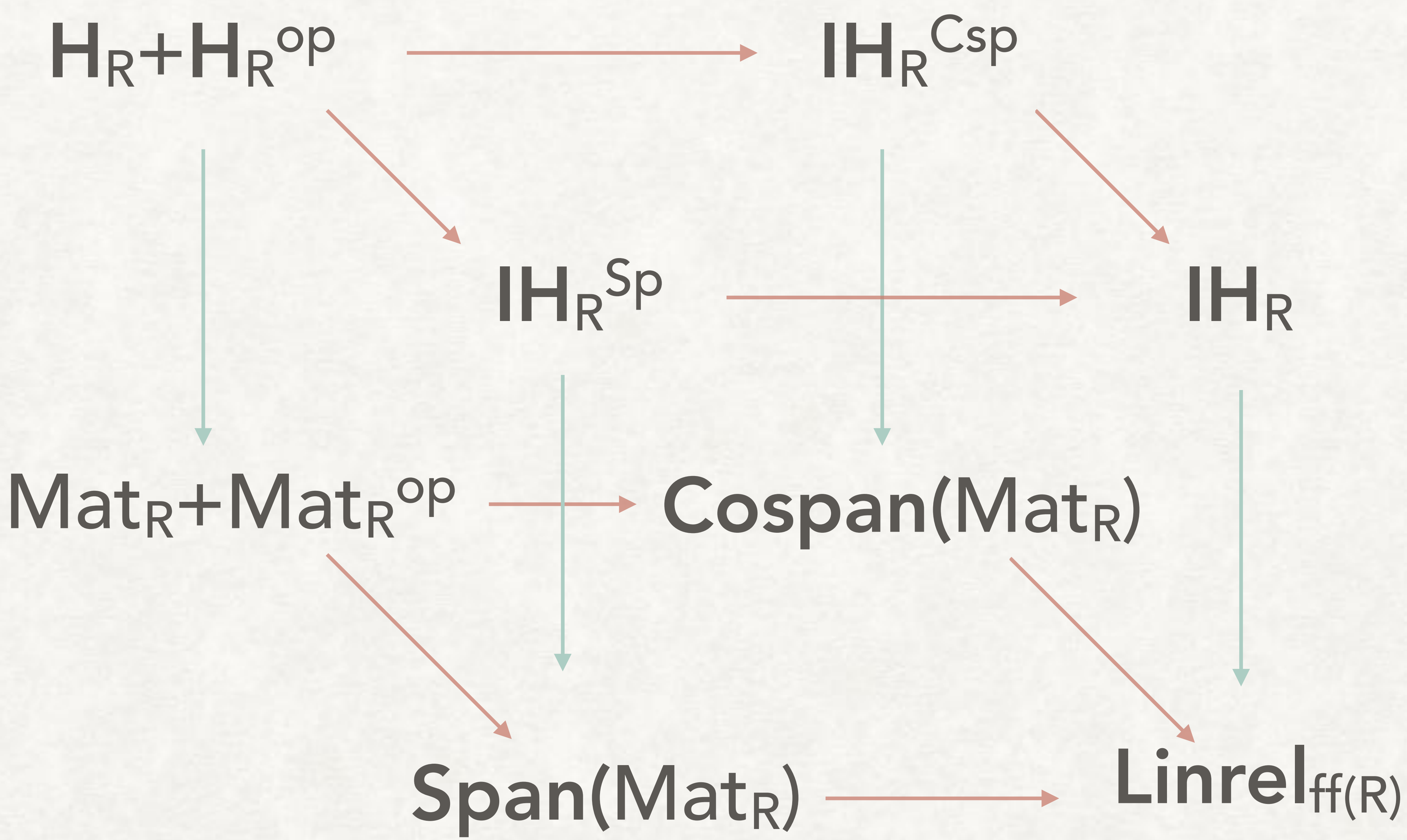
- Interacting Hopf Algebras (**IH**)
- presentation of $\mathbf{LinRel}_{\mathbb{Q}}$
- algebra of fractions follows from algebra of diagrams
- but nothing stops you from dividing by zero…
- diagrammatic playground for elementary linear algebra
- https://GraphicalLinearAlgebra.net



THE GENERAL MATHEMATICAL STORY

OBTAIN COMPOSITIONAL THEORIES, COMPOSITIONALLY

building on S. Lack, Composing PROPs, TAC 2004



Bonchi, S, Zanasi, *Interacting Hopf Algebras*, J Pure Applied Algebra, 2016

Fabio Zanasi, Interacting Hopf Algebras, the theory of linear systems, PhD thesis, 2015

PLAN

- Compositional Petri nets
  - *verify quicker!*
- Compositional linear algebra
  - *divide by zero!*
- **Compositional signal flow graphs**
  - ***say goodbye to inputs and outputs!***
  - ***program recurrence relations!***
- Compositional everything
  - *DPO rewriting*

[[-]] : **Syntax → Semantics**

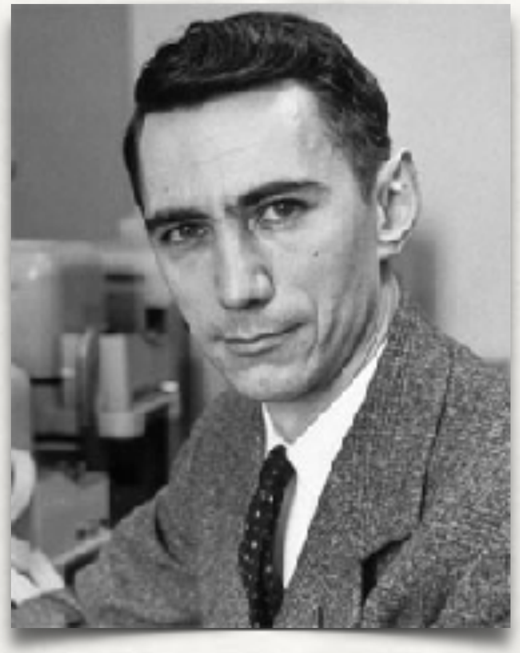
CLAUDE SHANNON

(1916 - 2001)

The Theory and Design of Linear Differential Equation Machines*

* Report to National Defense Research Council, January, 1942.

- This report deals with the general theory of machines constructed from the following five types of mechanical elements.
  - Integrators
  - Adders or differential gears
  - Gear boxes
  - Shaft junctions
  - Shafts



INTERACTING HOPF ALGEBRAS

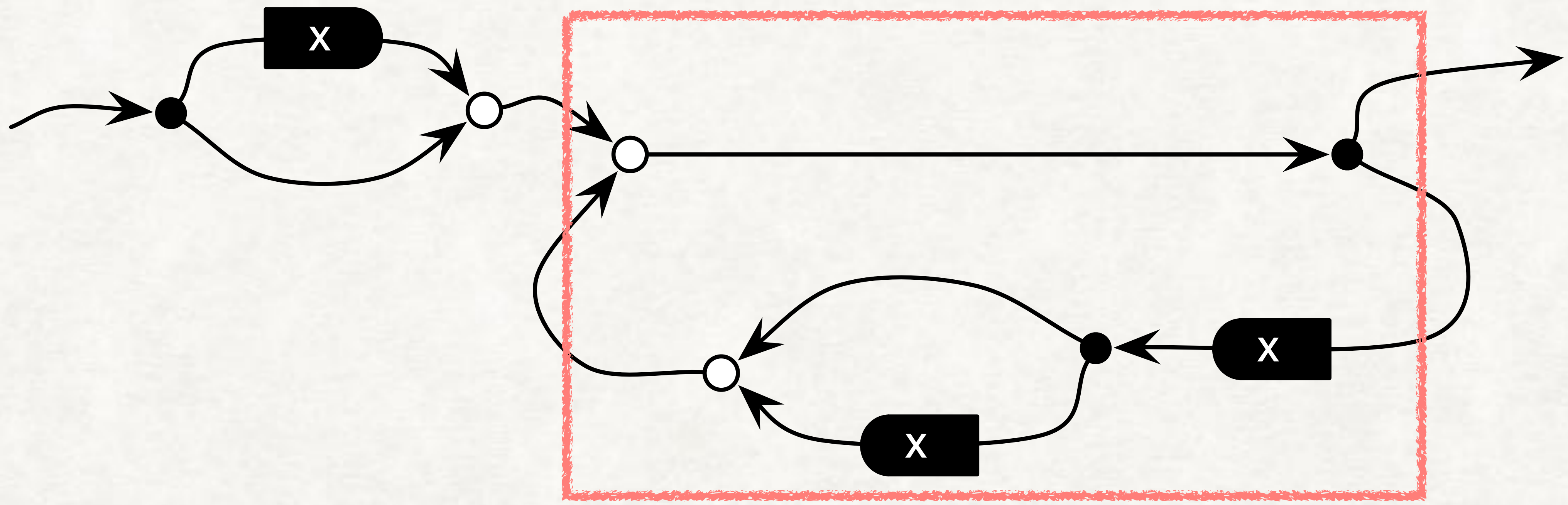
$$[[-]] : \mathbf{IH}_{k[x]} \to \mathbf{LinRel}_{k(x)} \to \mathbf{LinRel}_{k((x))}$$

isomorphism

faithful, not full

- Diagrams of IHk[x] are an algebra of signal flow graphs (Bonchi, S, Zanasi, A categorical semantics of signal flow graphs, CONCUR 2014)
- symmetric monoidal theory IHk[x] was independently found by Baex and Erbele in *Categories in Control*, see also Jason Erbele's thesis *Categories in Control: Applied PROPs*.
- $\mathbf{IH}_{k[x]}$ can be thought of as a process calculus, with an operational semantics (Bonchi, S, Zanasi, *Full Abstraction for Signal Flow Graphs*, PoPL 2015)

HOW TO UNDERSTAND FEEDBACK WITH TEARING?



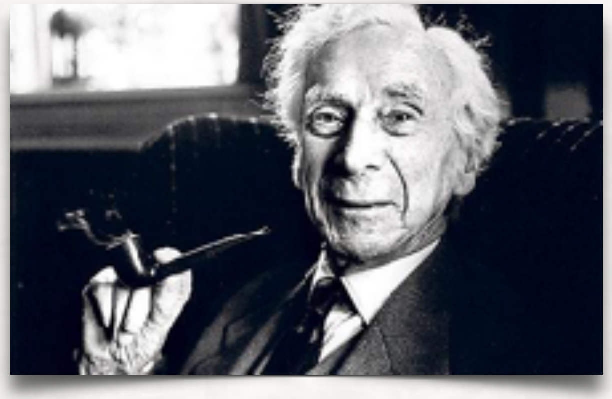
What are these things exactly?

the inputs and outputs are confused…

SOLUTION 1: INTRODUCE FEEDBACK AS A (SCARY) **ADDITIONAL, PRIMITIVE OPERATION**

cf. Kleene star, traced monoidal categories, …

DIRECTION OF FLOW CONSIDERED HARMFUL



The law of causality, I believe, like much that passes muster among philosophers, is a relic of a bygone age, surviving like the monarchy, only because it is erroneously supposed to do no harm.

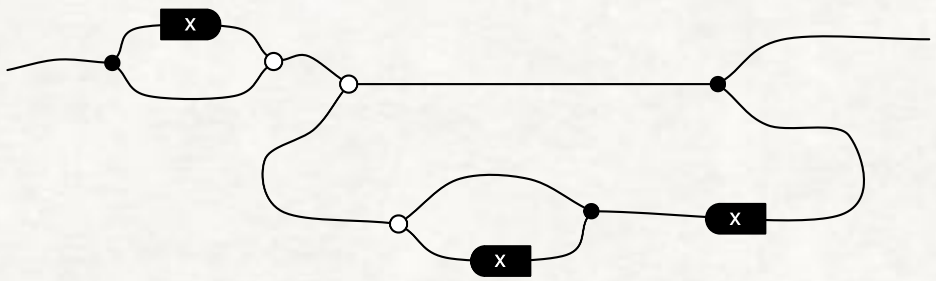
Bertrand Russell, *On the notion of cause*, 1912

It is remarkable that the idea of viewing a system in terms of inputs and outputs, in terms of cause and effect, kept its central place in systems and control theory throughout the 20th century. Input/output thinking is not an appropriate starting point in a field that has modeling of physical systems as one of its main concerns.

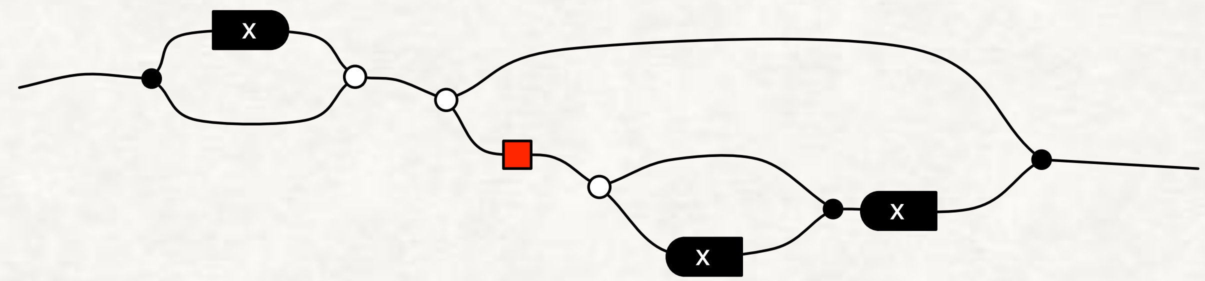
Jan Willems, *The Behavioral Approach to Open and Interconnected Systems*, 2007

SOLUTION 2: THROW AWAY ARROW HEADS.
THINK OF BEHAVIOUR AS A **RELATION**, NOT AS A FUNCTION

SUSTAINABLE RABBIT FARMING



=

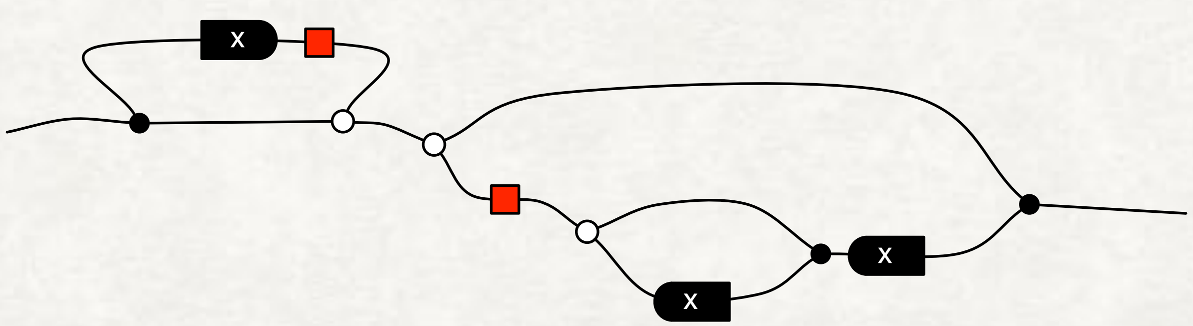


(this is the string diagram notation for the polynomial fraction

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

which is the **generating function** for 1,2,3,5,8…)

=

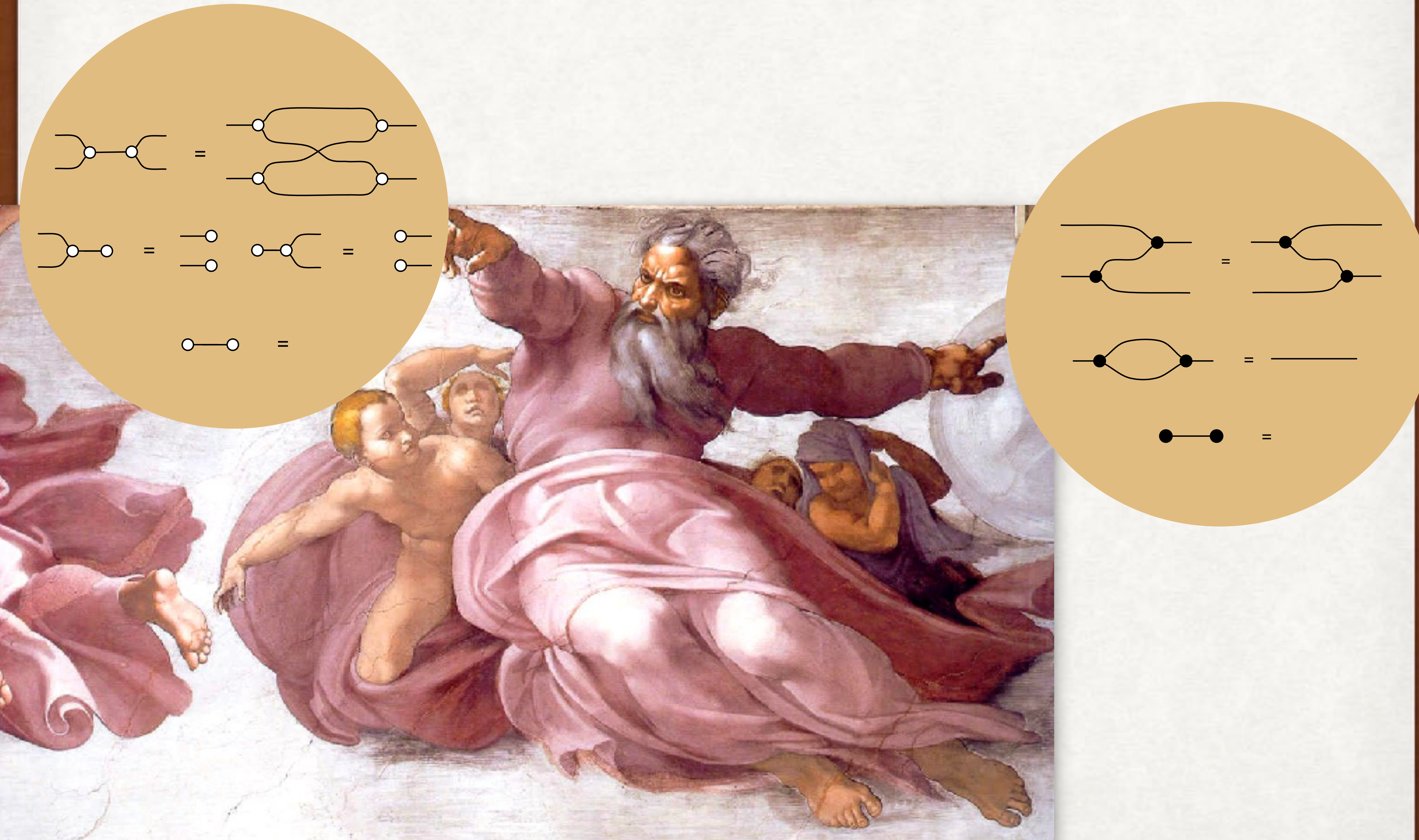


PLAN

- Compositional Petri nets
  - *verify quicker!*
- Compositional linear algebra
  - *divide by zero!*
- Compositional signal flow graphs
  - *say goodbye to inputs and outputs!*
  - *program recurrence relations!*
- **Compositional everything**
  - *DPO rewriting!*

[[-]] : **Syntax** → **Semantics**

IN THE BEGINNING ...



THERE ARE MANY EXAMPLES

- Frobenius: cartesian bicategories -> bicategories of relations, hypergraph categories (Carboni and Walters, *Cartesian Bicategories I*, J Pure Applied Algebra 1987)
  - if you want a monoid structure to be a map, it must be a homomorphic wrt to comonoid structure -> bialgebra
- quantum circuits (Abramsky, Coecke, Duncan, Pavlovic, …)
  - Frobenius = choice of basis, bialgebra = complementarity
- calculus of stateless connectors (Bruni, Montanari), Petri nets, algebras of synchronisation
  - Frobenius = copying, bialgebra = non-determinism, mutual exclusion
- signal flow graphs, electrical circuits, hydraulic systems, time invariant linear dynamical systems
  - Frobenius = feedback, bialgebra = additive structure

SLOGAN

FROBENIUS + BIALGEBRA IS A HIGH LEVEL PROGRAMMING LANGUAGE FOR COMPOSITIONAL MATHEMATICS

Fong, Rapisarda, S., A categorical approach to open and interconnected dynamical systems, LiCS 2016

- Equationally, we weaken **IH** ever so slightly by replacing one equation with a weaker version
- But under the hood, we have to deal with mathematical machine code
  - $k[x]$ is replaced with $k[x,x^{-1}]$
  - linear relations with corelations
  - invoking some industrial strength mathematical control theory…

IMPLEMENTATION: REWRITING WITH DPO

Bonchi, Gadducci, Kissinger, S., Zanasi, *Rewriting modulo symmetric monoidal structure*, LiCS 2016

Bonchi, Gadducci, Kissinger, S., Zanasi, *Confluence of graph rewriting with interfaces*, submitted

- Idea - by orienting the equations of a symmetric monoidal theory we obtain a rewriting system where matches are found modulo symmetric monoidal structure
- Suppose that **Σ** is a symmetric monoidal signature, and $\mathbf{S}_\Sigma$ is the free PROP on **Σ**
- **Frob** is the theory of special Frobenius monoids

$$\boxed{\mathbf{S}_\Sigma + \mathbf{Frob} \cong \mathrm{Csp}(\mathbf{Hyp}_\Sigma)}$$

- Hyp_Σ is adhesive

DPO on Hyp_Σ $\Leftrightarrow$ rewriting modulo symmetric monoidal + special Frobenius

convex DPO on Hyp_Σ $\Leftrightarrow$ rewriting modulo symmetric monoidal

PLAN

- **Compositional Petri nets**
  - *verify quicker!*
- **Compositional linear algebra**
  - *divide by zero!*
- **Compositional signal flow graphs**
  - *say goodbye to inputs and outputs!*
  - *program recurrence relations!*
- **Compositional everything**
  - *DPO rewriting*