PROGRAMMING
RECURRENTNESS
RELATIONS
and other compositional stuff…

Compositionality Workshop, Simons Institute, 9 December 2016
• 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 \rightarrow Semantics

symmetric monoidal functor,
cf. Lawvere’s functorial semantics
COMPOSITIONAL PETRI NETS

A compositional approach can be used for

- faster verification through divide and conquer
- parametric verification

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

D ; ((S ; T ; T) ⊗ I) ; E

\[ D ; ((S ; T ; T) \otimes I) ; E \]
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 FUNCTIONAL SEMANTICS

\[ \text{[[-]]}: \text{Petri} \rightarrow 2\text{LTS} \]

Petri and Aut are symmetric monoidal categories
and \text{[[-]]} is a symmetric monoidal functor
The transition presentation is quite intuitive. See the appendix for a formal treatment.

We describe this operation informally with examples because the graphical synchronisations of the second net—this is a general requirement for composition to be defined: the size of the right boundary of the first net agrees with the size of the left boundary; we illustrate this operation in Fig. 3. In each of the examples, the size connected to any boundary port in the composed net. The situation for transition nets that do not agree on the size of their common boundary cannot be synchronised. Given nets

\[ X \]

\[ Y \]

their composition is denoted \( P ; Q \). In general, transitions of the composed net—called the minimal subset of transitions of the individual component—will be subsets of transitions of the individual component. The most interesting operation on PNBs is synchronisation along a common boundary; we illustrate this operation in Fig. 3. Consider the top left quadrant of Fig. 3. The composed net

\[ P : (0, 2) \]

\[ Q : (2, 0) \]

\[ P ; Q : (0, 0) \]

is now fully specified and will not be further altered because it is not connected to any boundary port in the composed net. The situation for transition nets

\[ P : (0, 2) \]

\[ R : (2, 0) \]

\[ P ; R : (0, 0) \]

has a transition \( t \)

\[ P : (0, 2) \]

\[ S : (2, 0) \]

\[ P ; S : (0, 0) \]

\[ P : (0, 2) \]

\[ T : (2, 0) \]

\[ P ; T : (0, 0) \]

\[ (t, a) \]

\[ (u, b) \]

\[ (t, u, g) \]

\[ (t, c) \]

\[ (t, d) \]

\[ (u, e) \]
2LTS

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

Given a finite LTS, we can consider only its runs that start and finish in specified states. When considering LTSs we often want to ignore the particular label and state names and instead concentrate on the transition connections, allowing us to identify those singly-labelled transitions. These labelling shorthand are illustrated in Fig.

Example 2.3.

Chapter 2 Preliminaries

To do so, we identify a certain state as being accepting. Given a finite LTS, we can consider only its runs that start and finish in specified states.

As we have seen, synchronous composition is only defined when the internal bound-ary sizes match up. Tensor composition, however, is defined for any pair of 2-LTSs, where we write states of a synchronous composition as

Given a finite LTS, we can consider only its runs that start and finish in specified states.

Given a 2-LTS, where we write states of a synchronous composition as

Chapter 2 Preliminaries

An example of synchronous and tensor composition is given in Fig.

Figure 2.2: Example 2-LTSs

Figure 2.3: Compact vs Expanded label notation

Chapter 2 Preliminaries

An example of synchronous and tensor composition is given in Fig.

Figure 2.2: Example 2-LTSs

Figure 2.3: Compact vs Expanded label notation

Chapter 2 Preliminaries

An example of synchronous and tensor composition is given in Fig.

Figure 2.2: Example 2-LTSs

Figure 2.3: Compact vs Expanded label notation
empty set of transitions is vacuously mutually independent.

Compositionality means resulting net is illustrated in Fig. 20a. Its semantics can be obtained directly by

For example, consider the Büchi LTS of Fig. 19b with itself, as illustrated in Fig. 20b.

Let \((p_0, 0)\) and \((p_0, 1)\),( \((p_1, 0)\) and \((p_1, 1)\))

\[\text{net composition} \rightarrow \text{semantics} \rightarrow \text{LTS composition} \rightarrow \text{semantics}\]
THE PROOF IS IN THE PUDDING


Penrose tool for reachability checking, compositionally

• P. Sobocinski, *Representations of Petri net interactions*, CONCUR 2010


• 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?
- Interacting Hopf Algebras (IH)
- presentation of $\text{LinRel}_Q$
- 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

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 $\rightarrow$ Semantics
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

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

- Diagrams of \( \text{IH}_{k[x]} \) are an algebra of signal flow graphs (Bonchi, S, Zanasi, A categorical semantics of signal flow graphs, CONCUR 2014)

- Symmetric monoidal theory \( \text{IH}_{k[x]} \) was independently found by Baex and Erbele in Categories in Control, see also Jason Erbele’s thesis Categories in Control: Applied PROPs.

- \( \text{IH}_{k[x]} \) 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, …
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

\[
\frac{1 + x}{1 - x - x^2}
\]

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

(this is the string diagram notation for the polynomial fraction)
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

  - 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
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 $\Sigma$ is a symmetric monoidal signature, and $S_{\Sigma}$ is the free PROP on $\Sigma$

- $\text{Frob}$ is the theory of special Frobenius monoids

\[
S_{\Sigma} + \text{Frob} \cong \text{Csp} (\text{Hyp}_{\Sigma})
\]

- $\text{Hyp}_{\Sigma}$ is adhesive

DPO on $\text{Hyp}_{\Sigma}$ rewriting modulo symmetric monoidal + special Frobenius
convex DPO on $\text{Hyp}_{\Sigma}$ 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