Syntactic reasoning for digital circuits

FMCAD 2016 // CSL 2017

Dan R. Ghica
with Achim Jung // Aliaume Lopez // George Kaye

HUAWEI

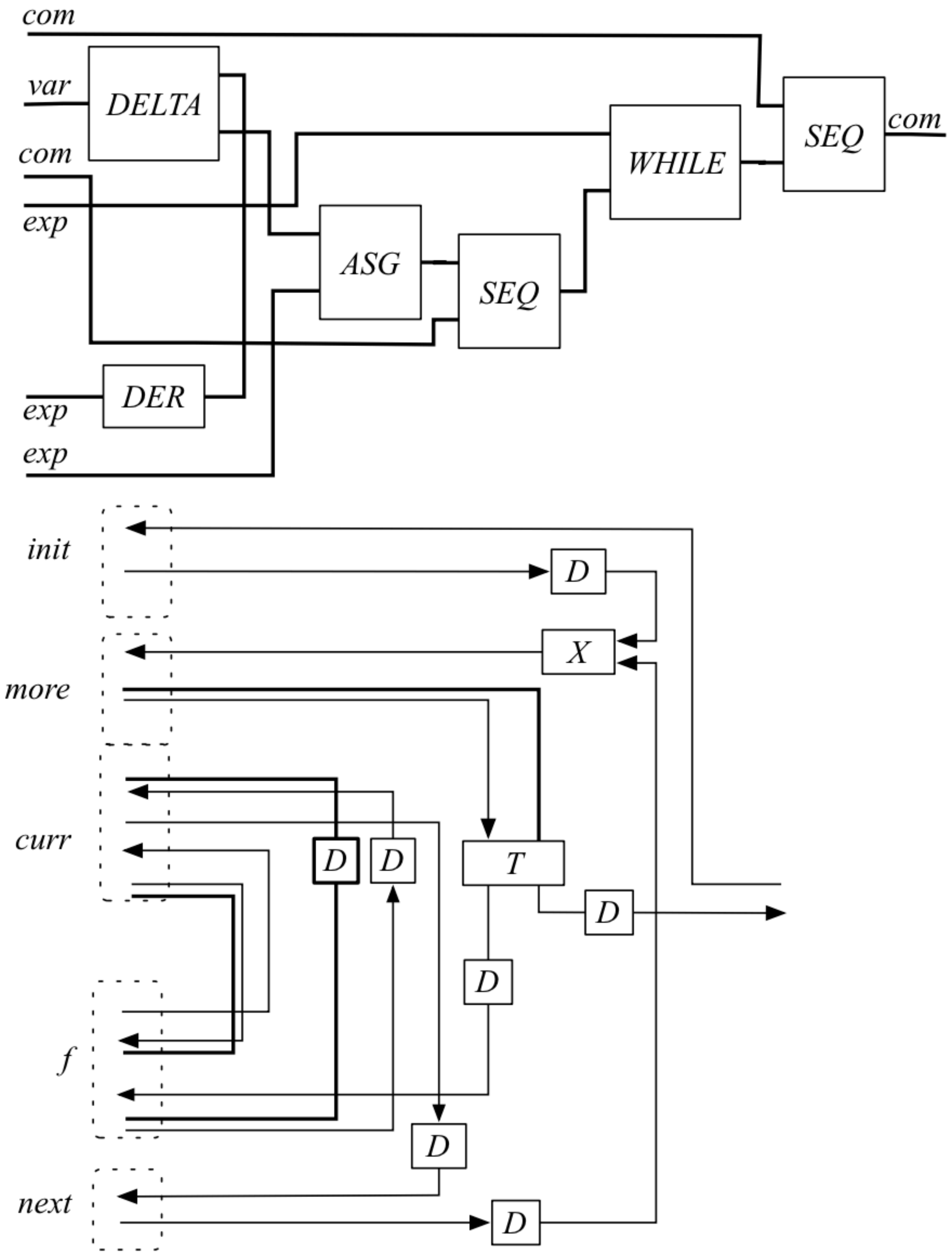
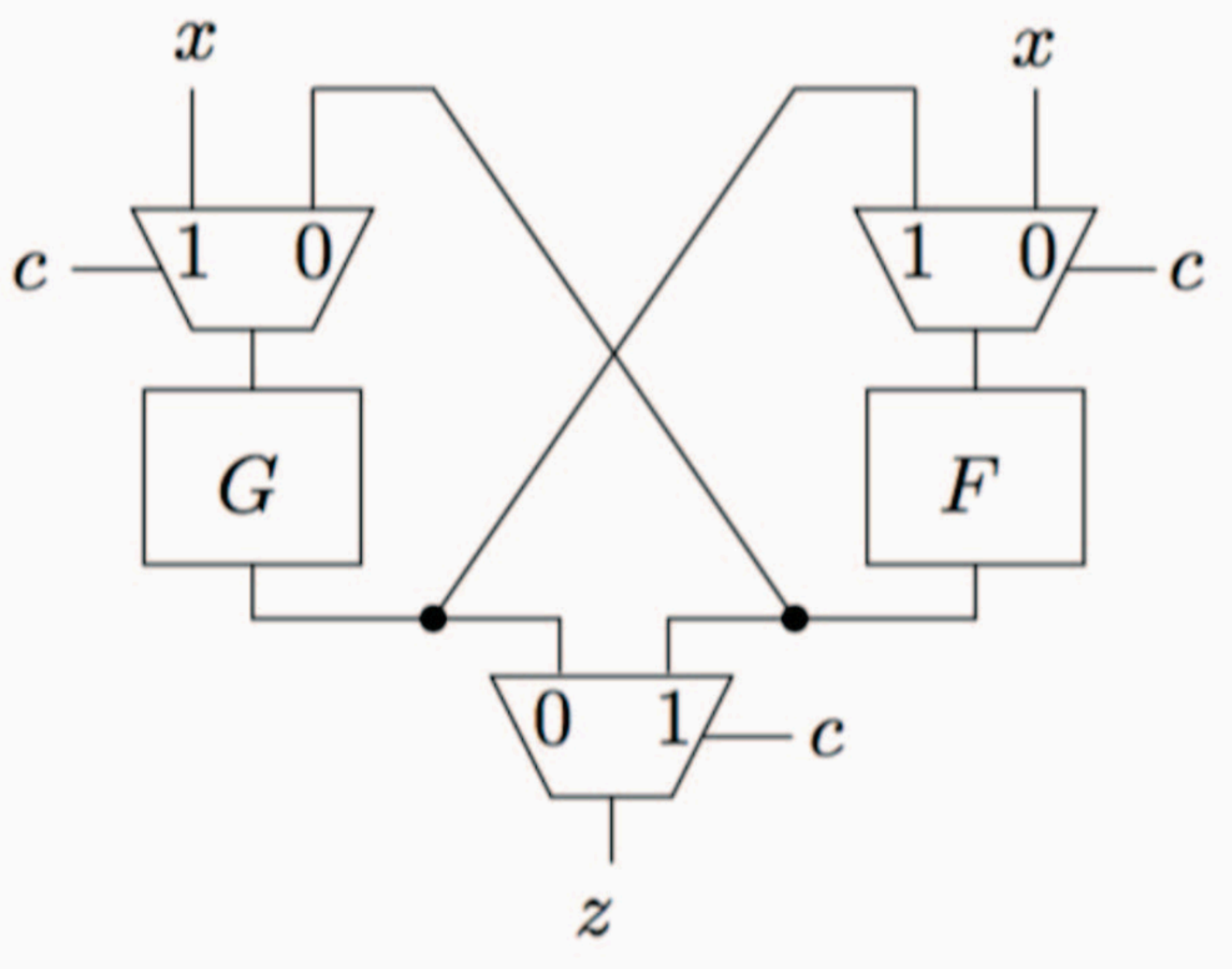


Figure 1. In-place map schematic and implementation

Example. The GoS approach allows the compilation of higher-order, open terms. Consider for example a program that executes in-place map on a data structure equipped with an iterator: $\lambda f : \mathsf{exp} \to \mathsf{exp}.\mathsf{init}; \mathsf{while}(\mathsf{more})(\mathsf{curr} := \mathsf{f}(!\mathsf{curr}); \mathsf{next}) : \mathsf{com}$, where $\mathsf{init} : \mathsf{com}$, $\mathsf{curr} : \mathsf{var}$, $\mathsf{next} : \mathsf{com}$, $\mathsf{more} : \mathsf{exp}$. The interface

Dan R. Ghica, Alex I. Smith, Satnam Singh:
Geometry of synthesis iv: compiling affine recursion into static hardware. ICFP 2011: 221-23

Domain-theoretic simulations are difficult to formalise.
We want *syntactic reasoning* (c.f. operational semantics)



Springer Link

Published: 28 March 2012

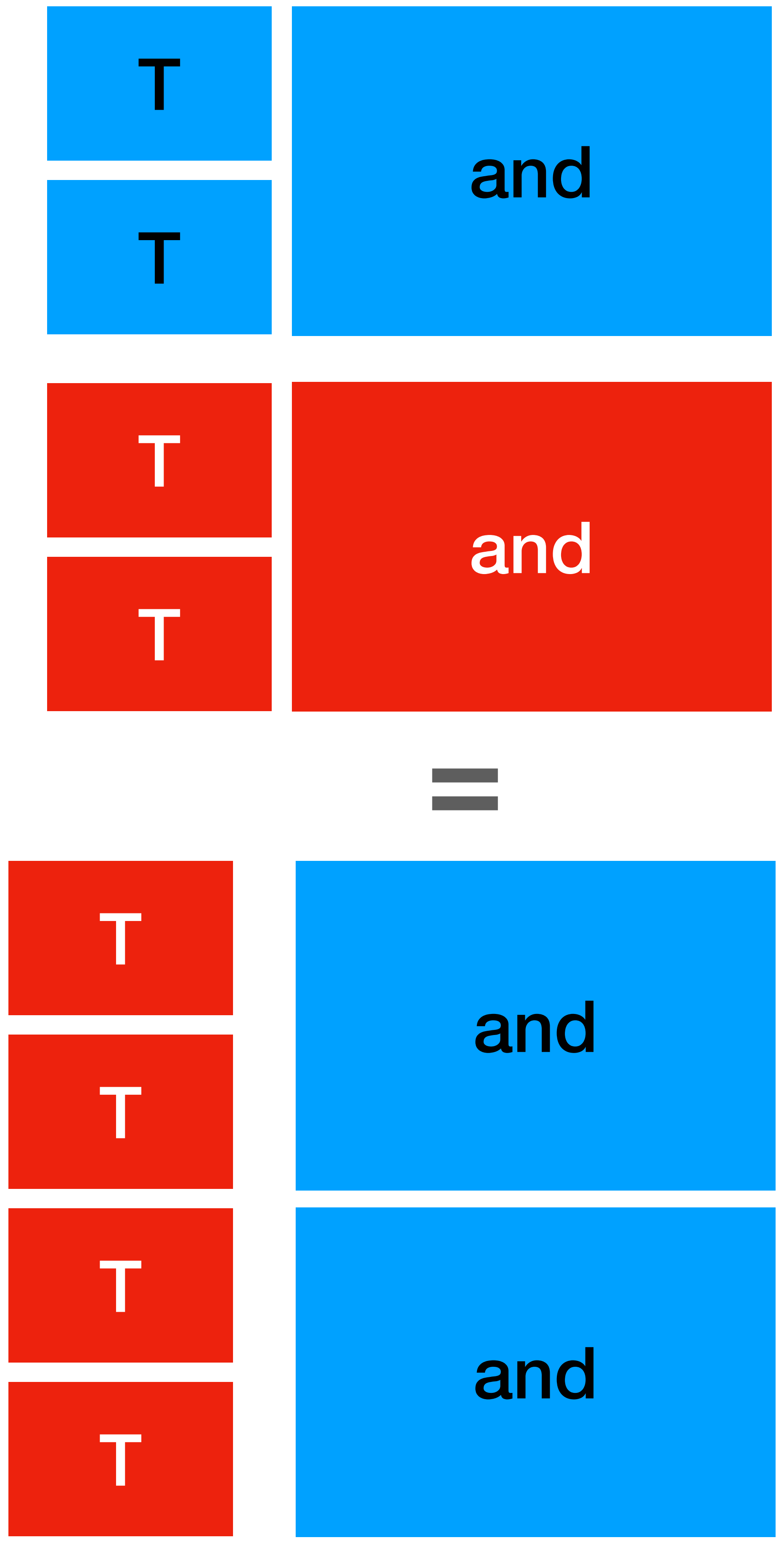
Constructive Boolean circuits and the exactness of timed ternary simulation

Michael Mendler, Thomas R. Shiple & Gérard Berry

Formal Methods in System Design **40**, 283–329(2012) | Cite this article

The problem with compositional syntax

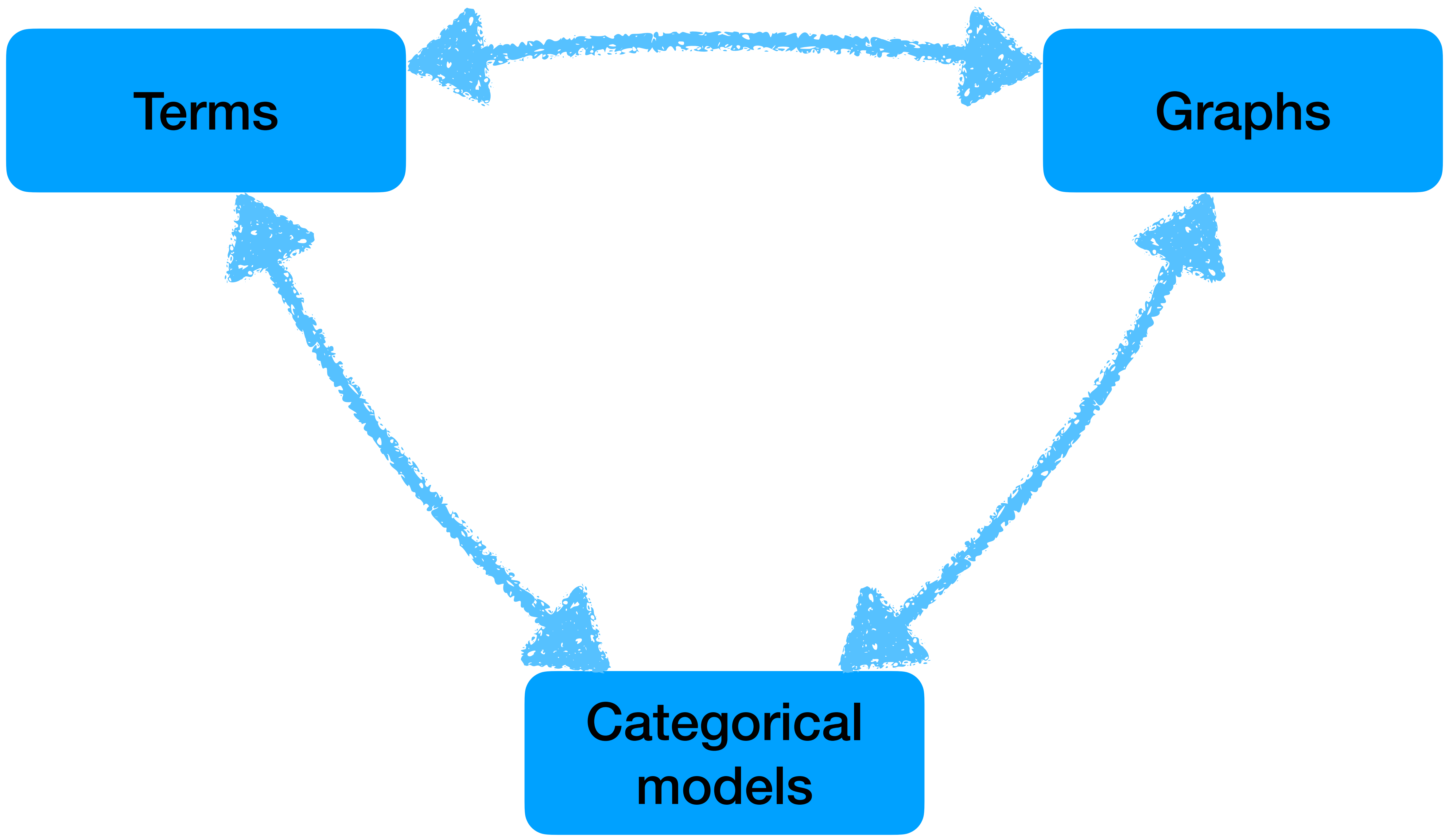
or where are the redexes?



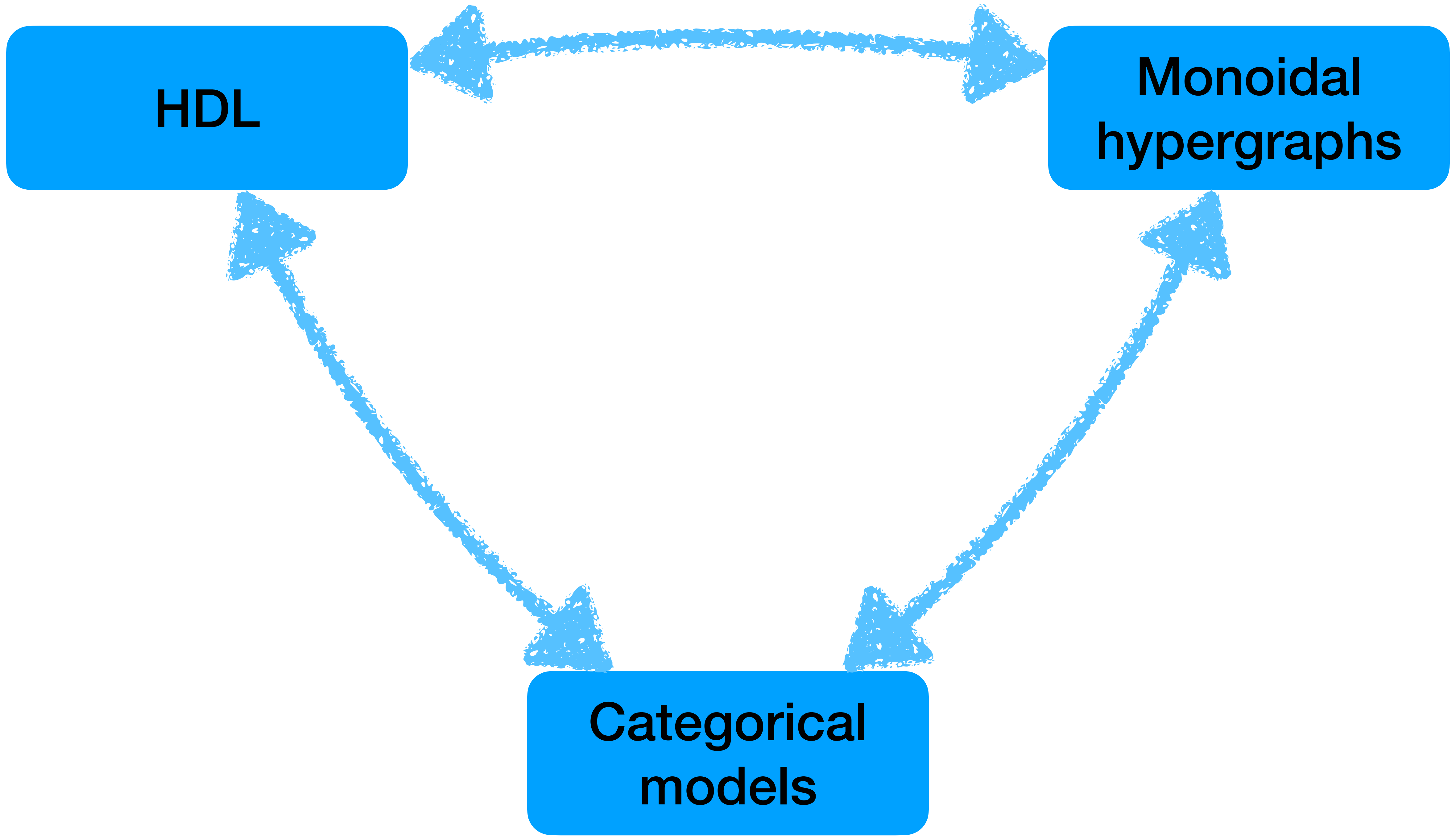
= (T|T).and | (T|T).and = T | T

= (T|T|T|T) . (and|and) = ???

distributive laws are bidirectional and require search strategies for the redex


Terms
Graphs
Categorical models

Categorical / string diagram semantics of digital circuits



Dan R. Ghica, Achim Jung, Aliaume Lopez:
Diagrammatic Semantics for Digital Circuits. CSL 2017: 24:1-24:16

Dan R. Ghica, Achim Jung:
Categorical semantics of digital circuits. FMCAD 2016: 41-48

arXiv.org > math > arXiv:2010.06319

Mathematics > Category Theory

[Submitted on 13 Oct 2020 (v1), last revised 21 Oct 2020 (this version, v2)]

The Graphical Language of Symmetric Traced Monoidal Categories

George Kaye

HUAWEI

Solution: use graph(ical) syntax

Diagrams = graphical representations of graphs

term language:
- definitions
- proof assistants
- inductive proofs

Object	A
Morphism	$f : A \to B$
Identity	$\mathrm{id}_A : A \to A$
Composition	$t \circ s$

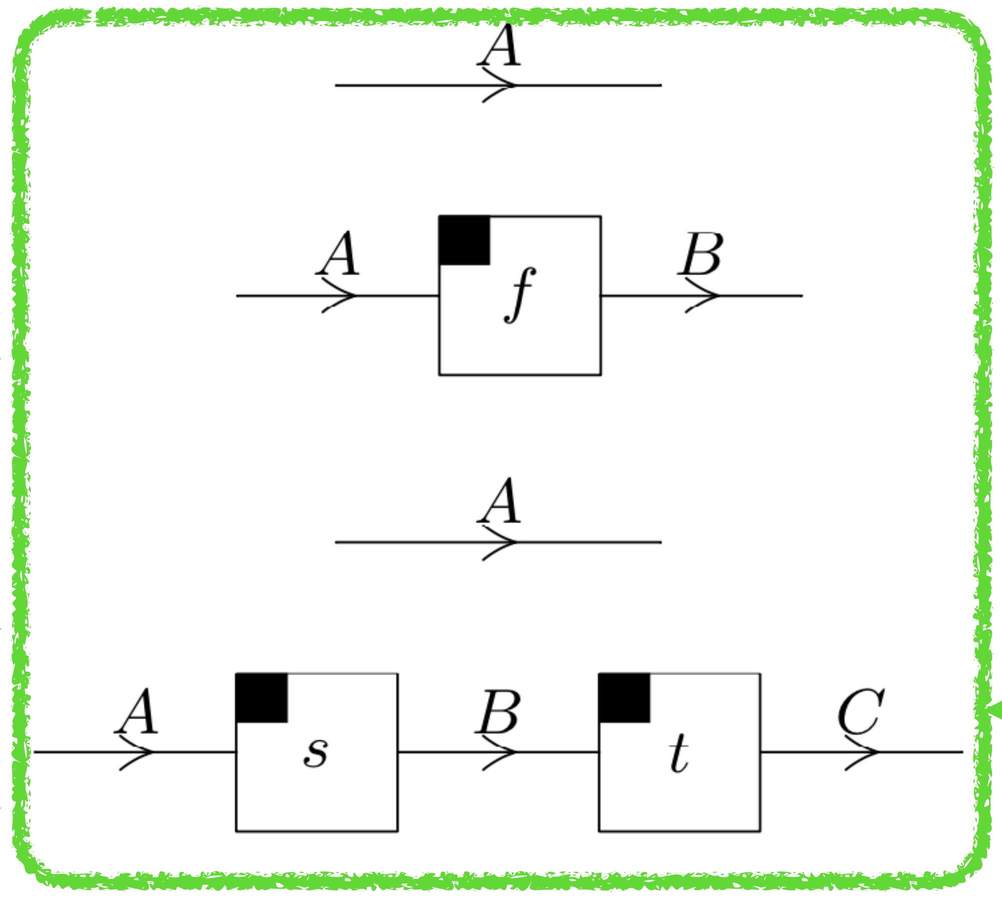


Table 1: The graphical language of categories

graph(ical) language:
- normal forms
- calculations
- tools
- co-inductive proofs

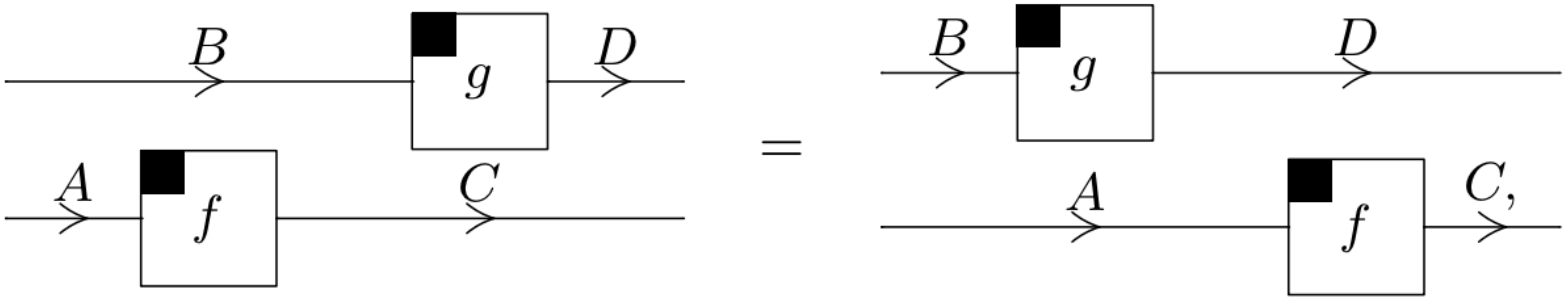
coherence theorems
(soundnes/completeness)

arXiv:0908.3347 [pdf, ps, other] math.CT doi 10.1007/978-3-642-12821-9_4
A survey of graphical languages for monoidal categories
Authors: Peter Selinger

HUAWEI

Bonus!

Some equations absorbed by graph isomorphisms

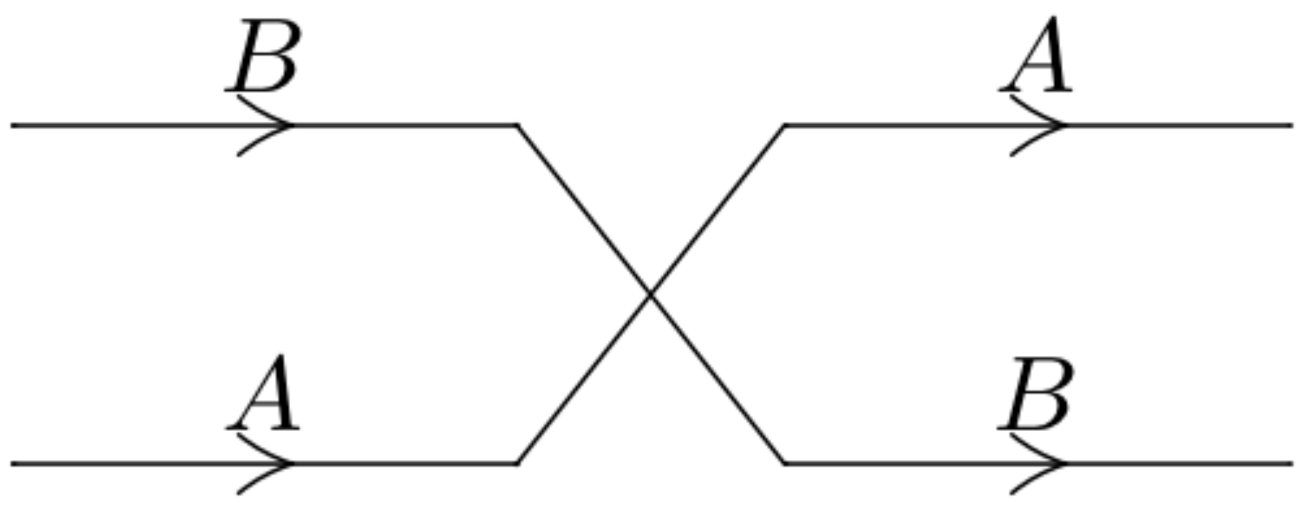


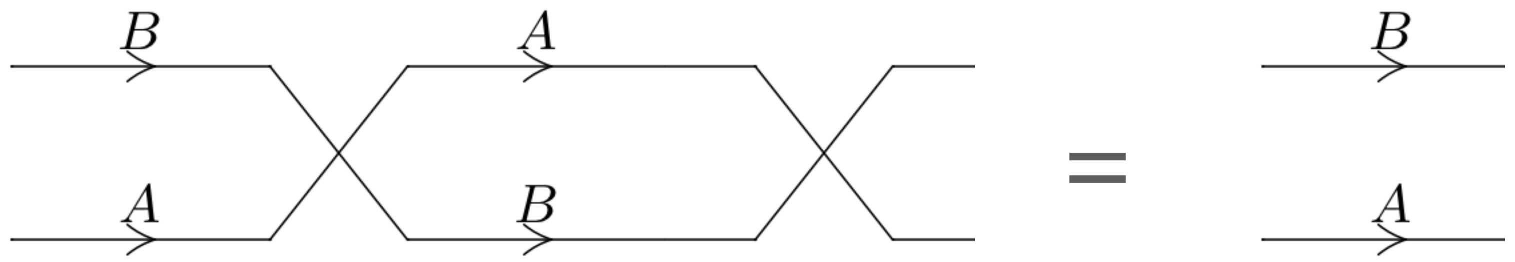
(id|f).(g|id)
= (id.g)|(<u>f.id</u>) distributivity of | ("functoriality of ⊗")
= g|f identity laws
= (<u>g.id</u>)|(id.f) identity laws
= (g|id).(id|f) distributivity of |

Symmetric monoidal category

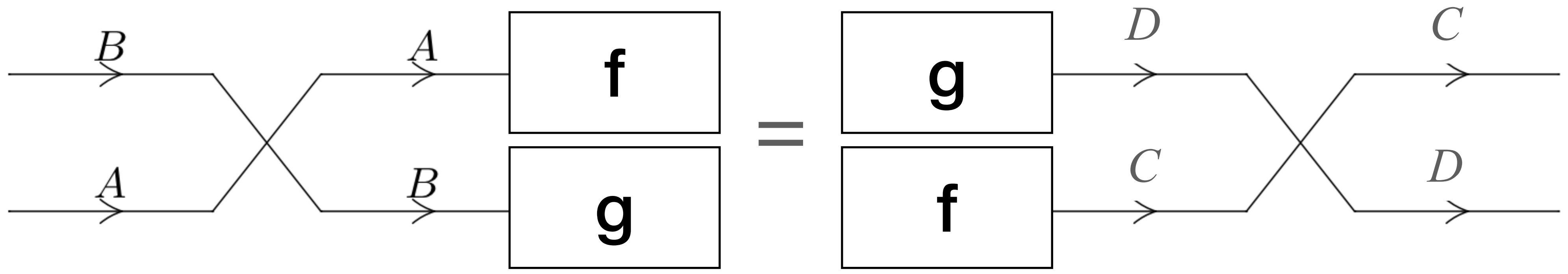
Boxes, wires, "swapping"

Symmetry $c_{A,B}$





c[A,B].c[A,B]=id[A,B], A,B∈ℕ



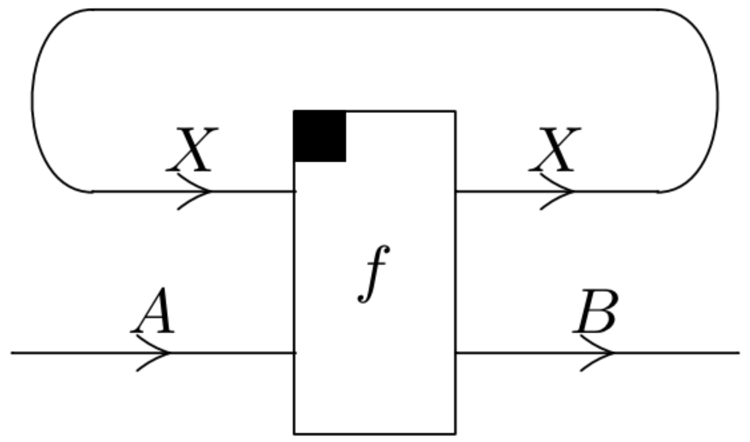
c[A,B].(f|g)=(g|f).c[C,D], A,B,C,D∈ℕ

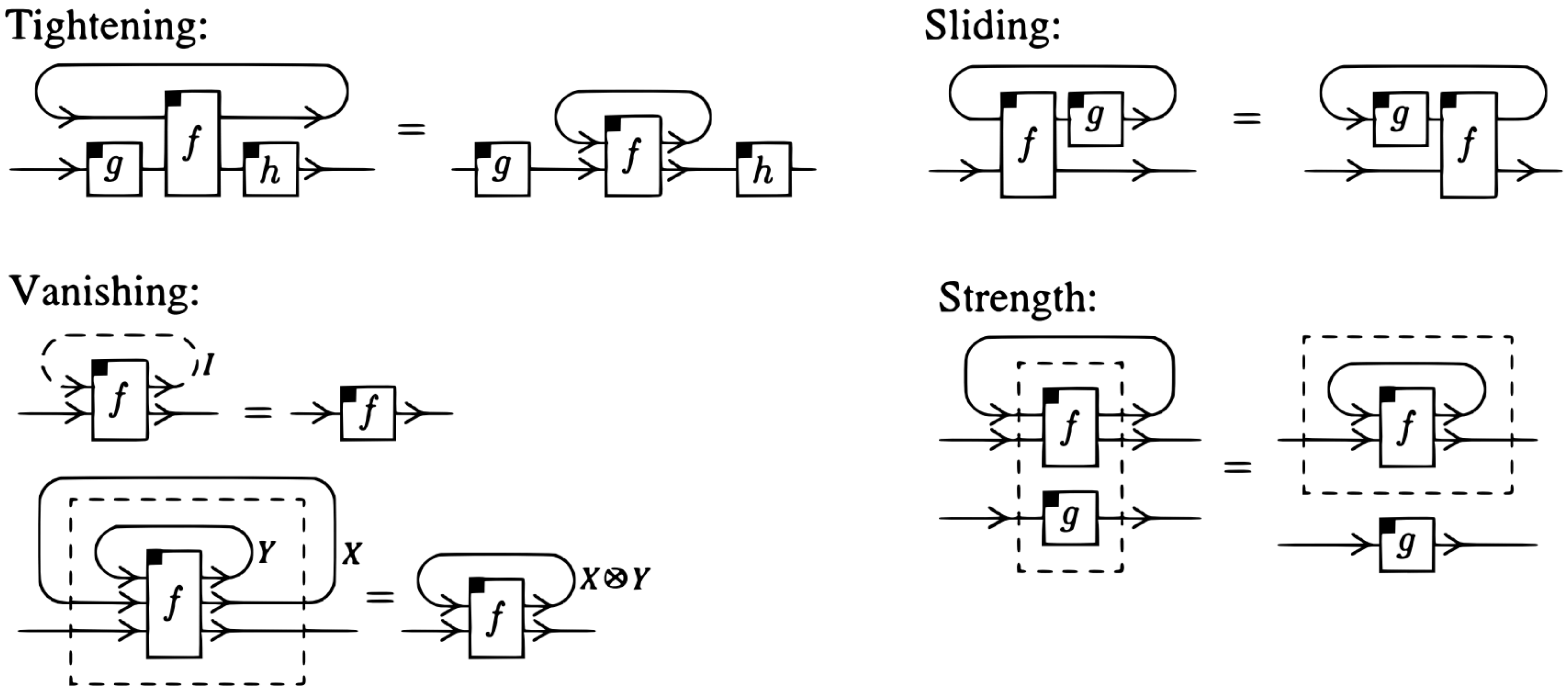
HUAWEI

Feedback = "trace"

Just "hygiene" rules

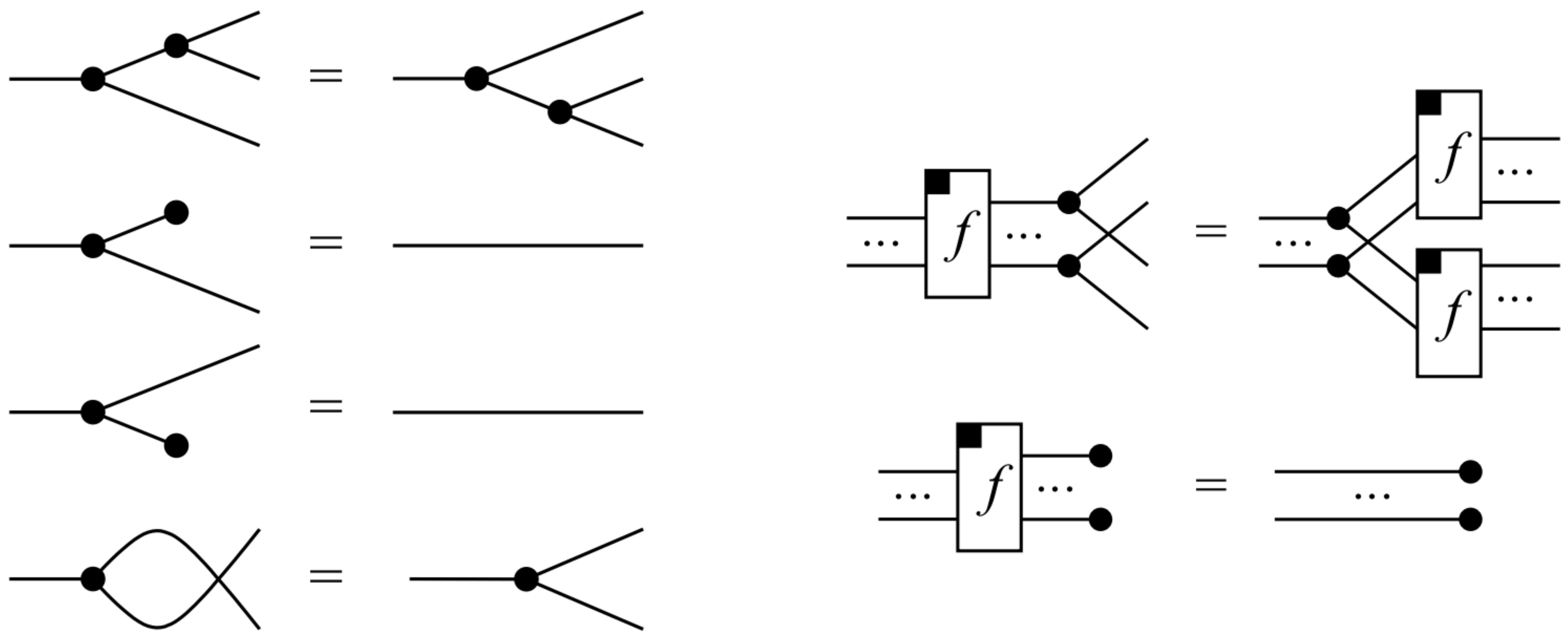
$\mathrm{Tr}^X_{\mathrm{R}}\ f =$





The tensor is a product

A key ingredient: copying and deletion



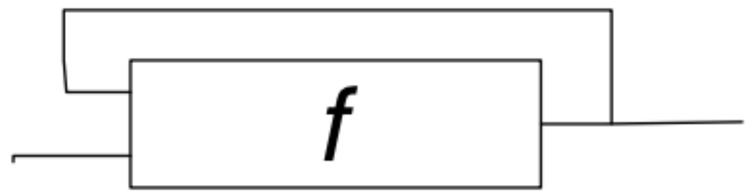
Commutative comonoid axioms

Naturality

Table 8: Graphical representation of some product axioms

Trace + Product = Iteration

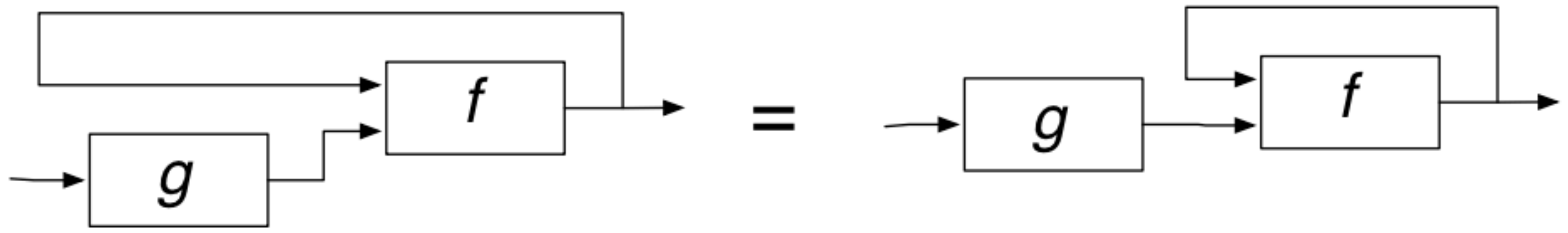
Proper equations for feedback ("dataflow-style")



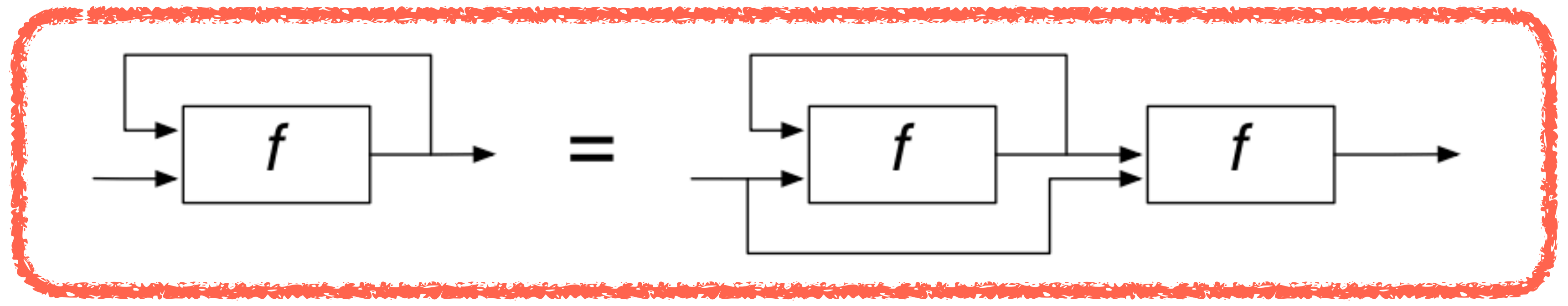
$$\mathrm{iter}^n(f) = \mathrm{Tr}^n(f \cdot (\Delta_n \otimes n)) : m \to n$$

which satisfies the following equations:

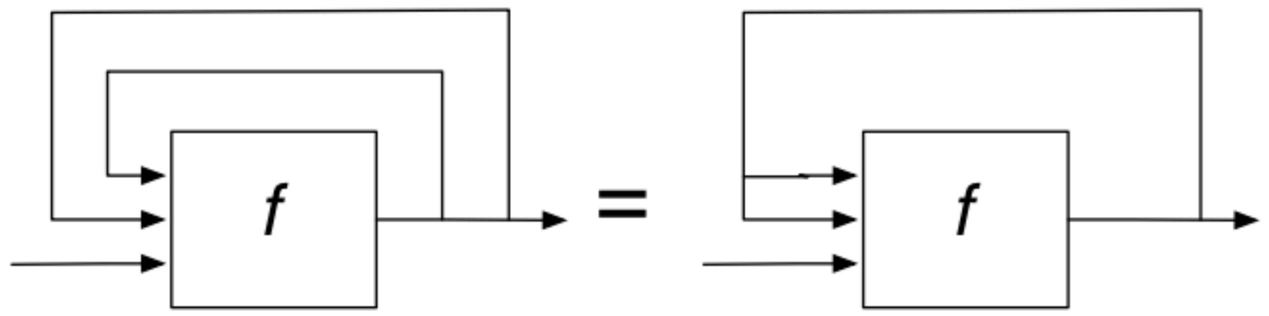
Naturality: $\mathrm{iter}((g \otimes n) \cdot f) = g \cdot \mathrm{iter}(f)$ for any $g : k \to m$.



Iteration: $\mathrm{iter}(f) = \langle m, \mathrm{iter}(f) \rangle \cdot f$



Diagonal: $\mathrm{iter}^n(\mathrm{iter}^n(f)) = \mathrm{iter}^n(((\langle n, n \rangle \otimes m) \cdot f)$.



Ésik. Comp. *Ling. and Comp Lang*'80 // Cazanescu. *TCS*'85 // Hasegawa. *Edinburgh*'97

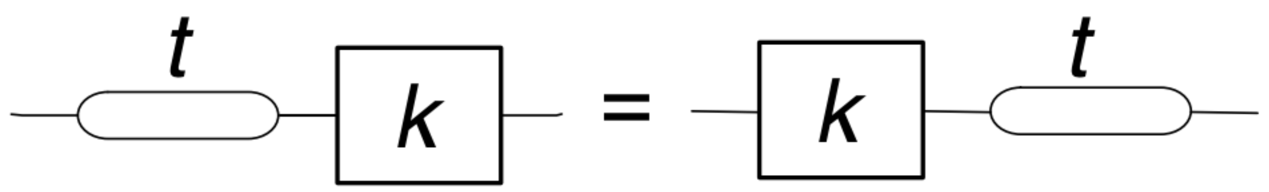
From iteration to digital circuit models

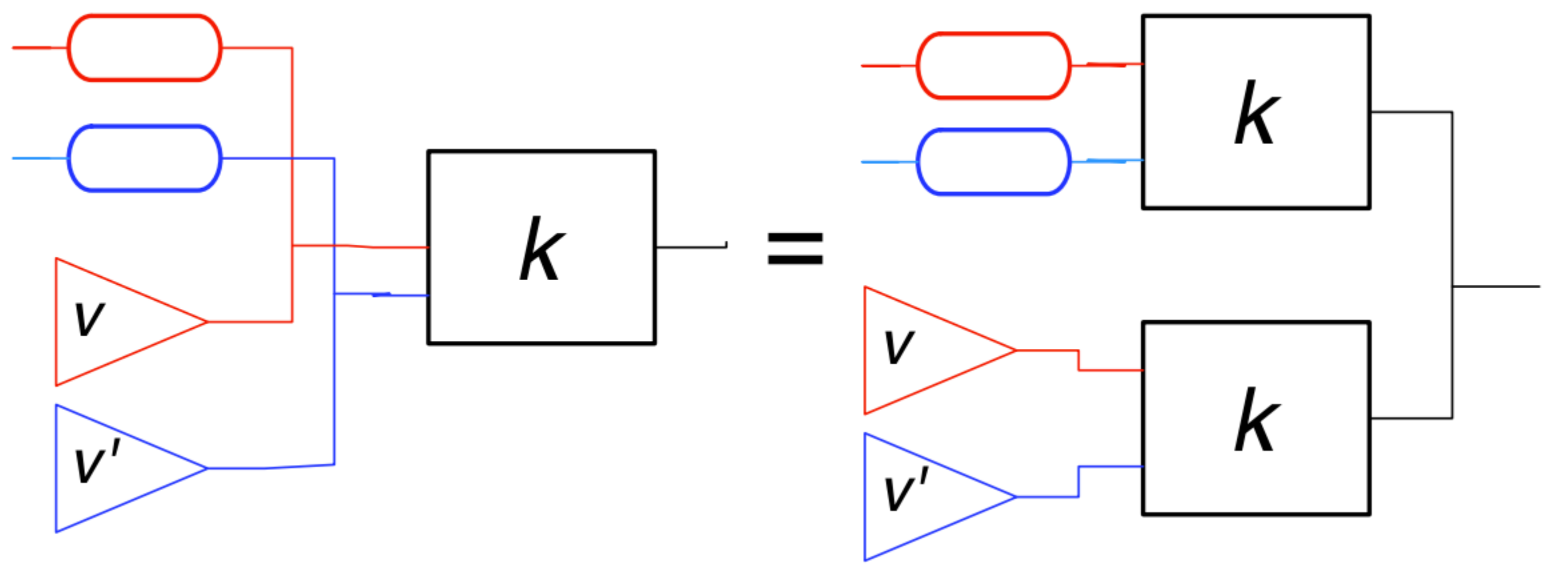
Special morphisms and axioms

- **values**: morphisms **v**:0→1 with a lattice structure
- **gates**: morphisms **k**:*n*→1 extensional and monotonic
    - **join**: special gate implementing join on the lattice of values
- **delays**: family of morphisms **δ**[*t*]:1→1indexed by elements of a monoid *t*∈***T***

Special axioms

Timelessness ("retiming") and Streaming



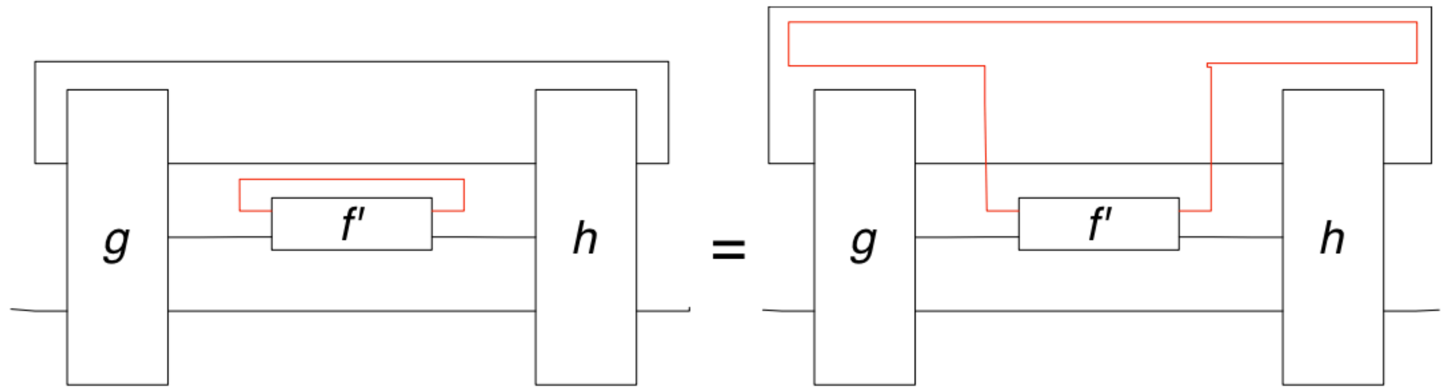


Important: in general *join* does not induce a co-product

The power of graphical reasoning

Normal forms

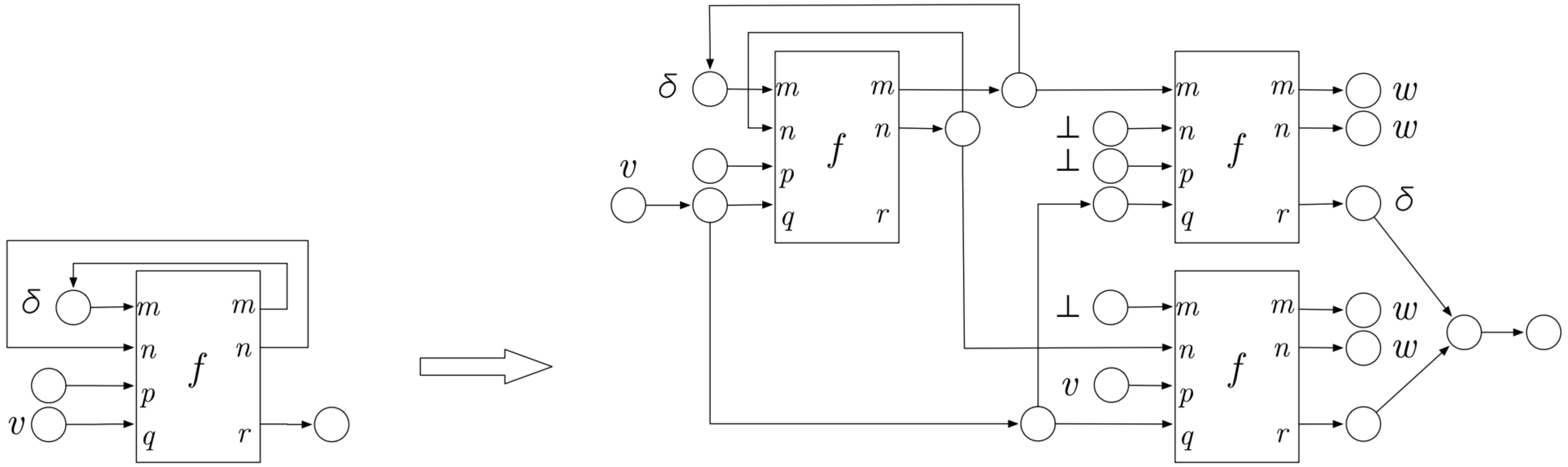
Lemma 5 (Global trace). *For any morphism f in a SMTC PROP there exists a trace-free morphisms $\hat{f}$ such that $f = \mathrm{Tr}^A(\hat{f})$ for some object A.*



Don't get trapped unfolding unproductive local feedbacks!

▶ **Proposition 21.** *Given a graph representing a passified, global trace, global delay circuit, $f : m + n + p + q \to m + n + r$, the following rewrite rule is sound:*

Passification and global delay-form are similar to global trace. (generalisation of Mealy machines)



The proof of this key proposition is purely diagrammatic.

▶ **Lemma 22.** *A passified, global-trace, global-delay circuit can be unfolded in a time linear in the size of its graph representation.*

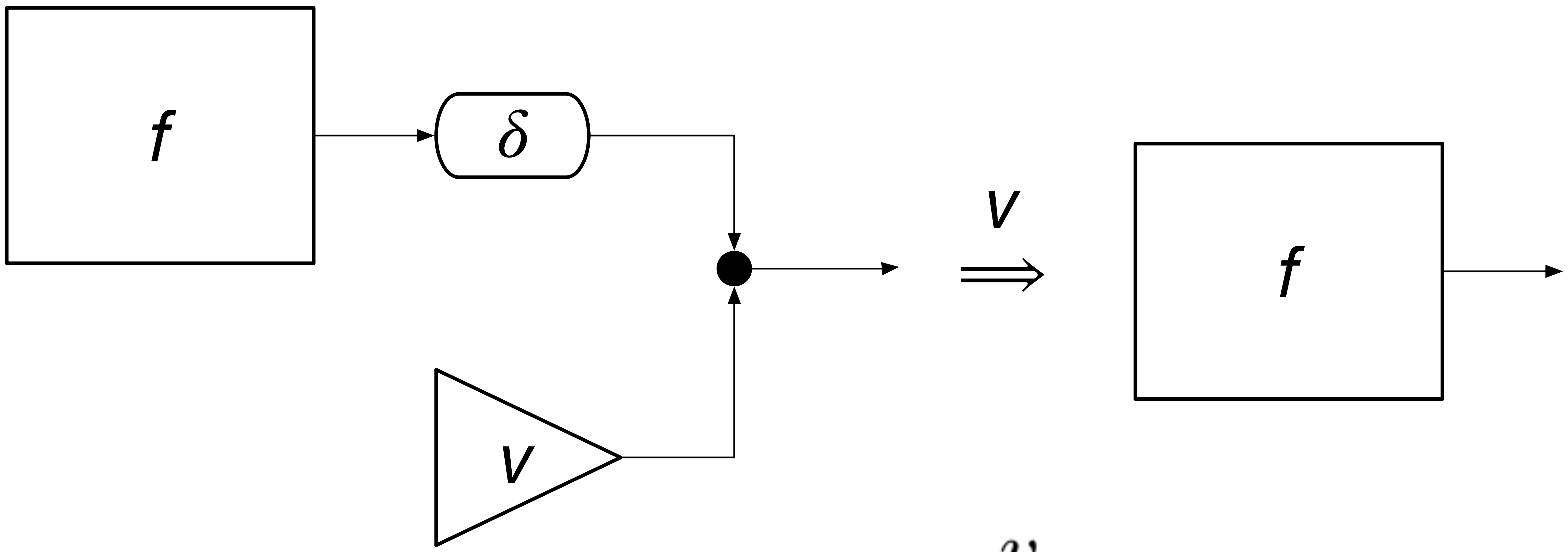
Efficient symbolic execution.

HUAWEI

Key theorem (I)

Characterisation of productivity

$$v :: f = (v \otimes (f \cdot \delta)) \cdot \curlyvee$$

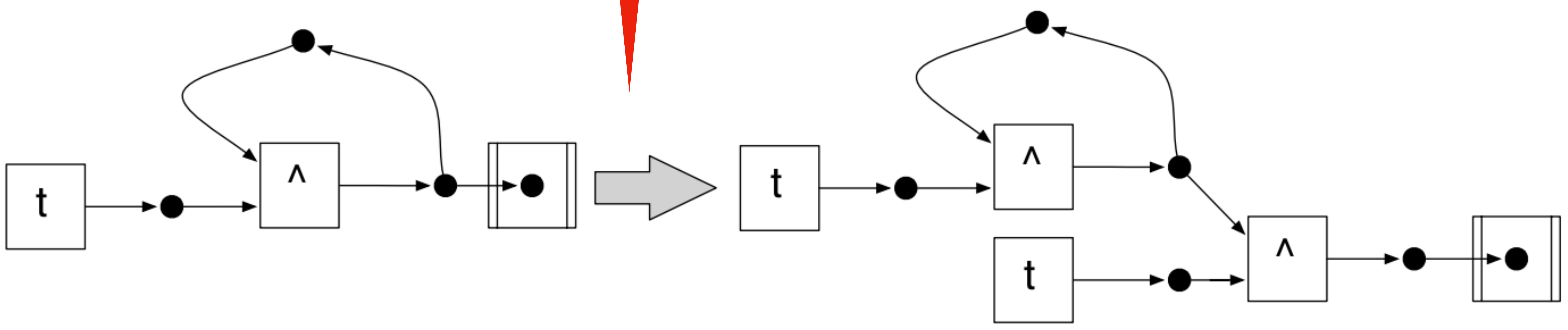


$$v :: f \overset{v}{\Longrightarrow} f.$$

Key theorem (II)

Characterisation of productivity

instantaneous feedback can lead to unproductive execution

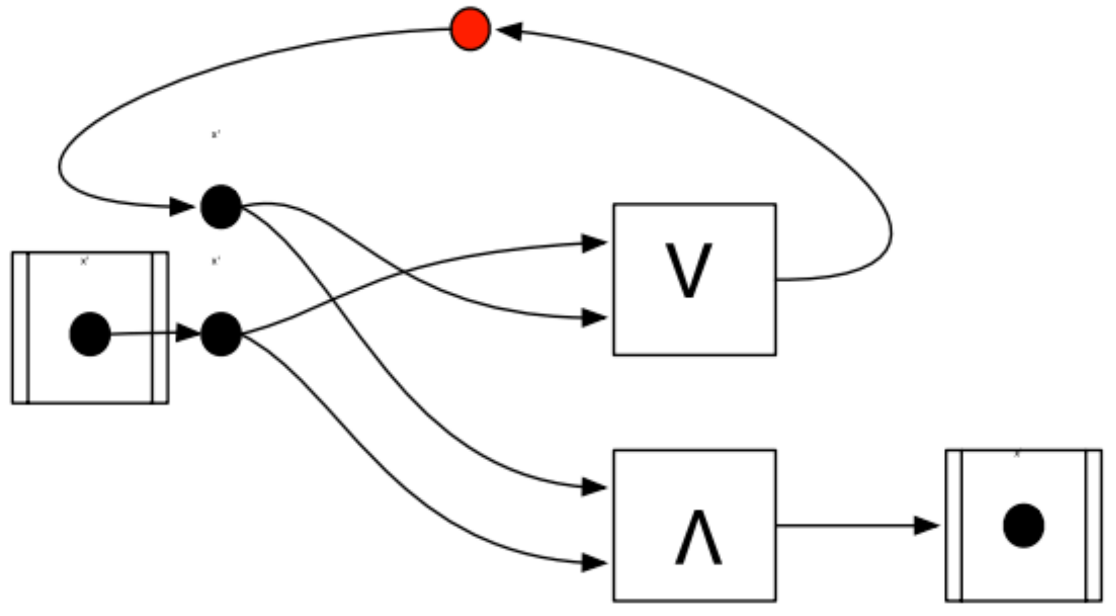


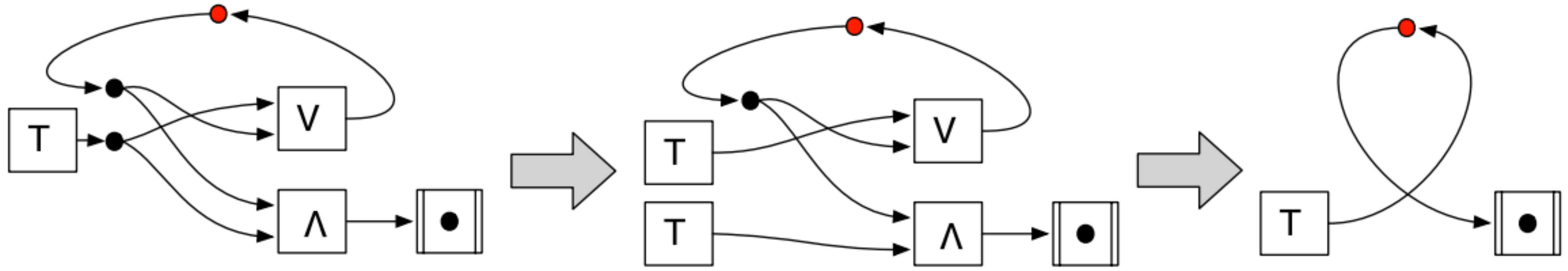
▶ **Theorem 25.** *If a closed, global-trace, global-delay circuit is unproductive after one unfolding then it will always be unproductive.*

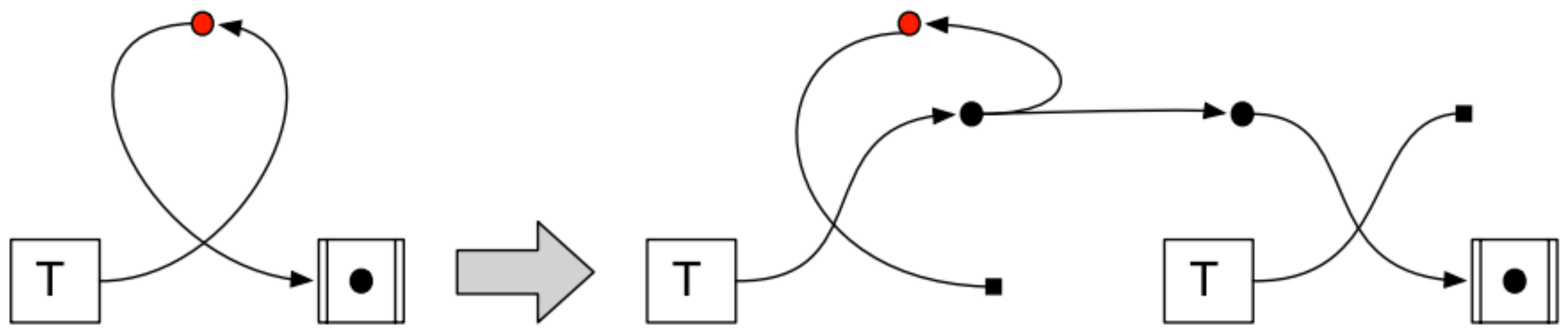
HUAWEI

Example: Cyclic combinational circuits

combinational feedback // evaluation



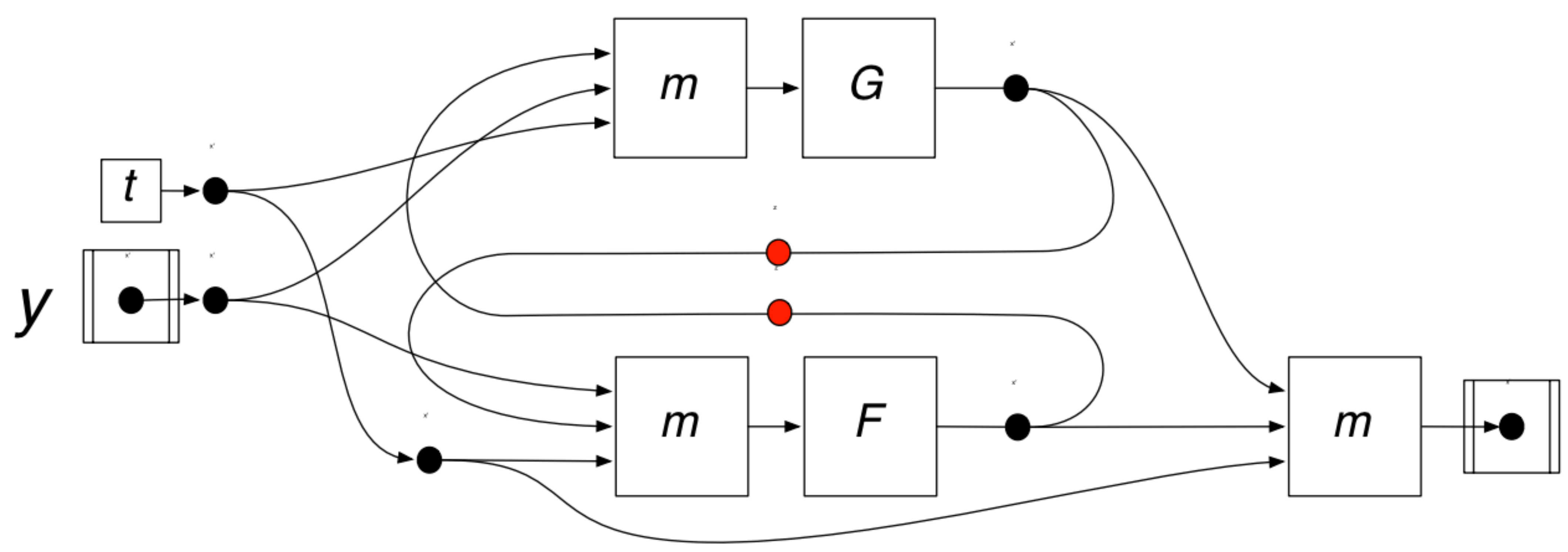


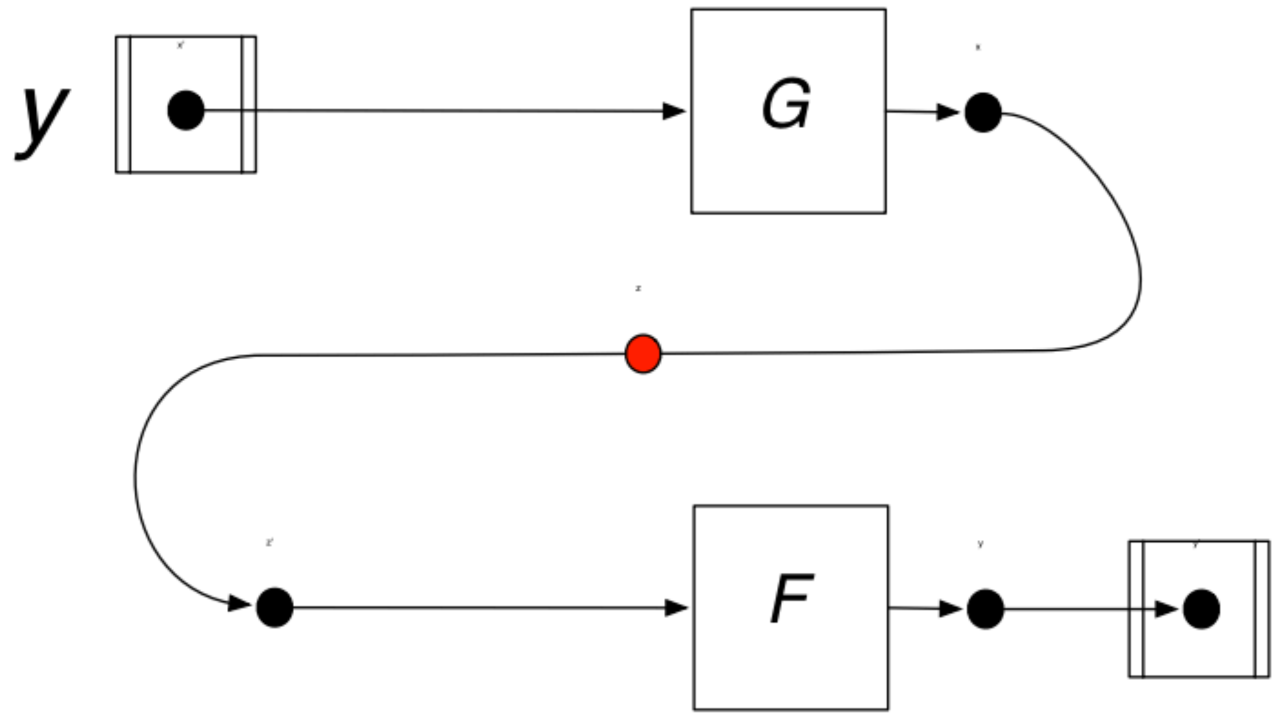


HUAWEI

Example: Cyclic combinational circuits

spurious feedback // partial evaluation

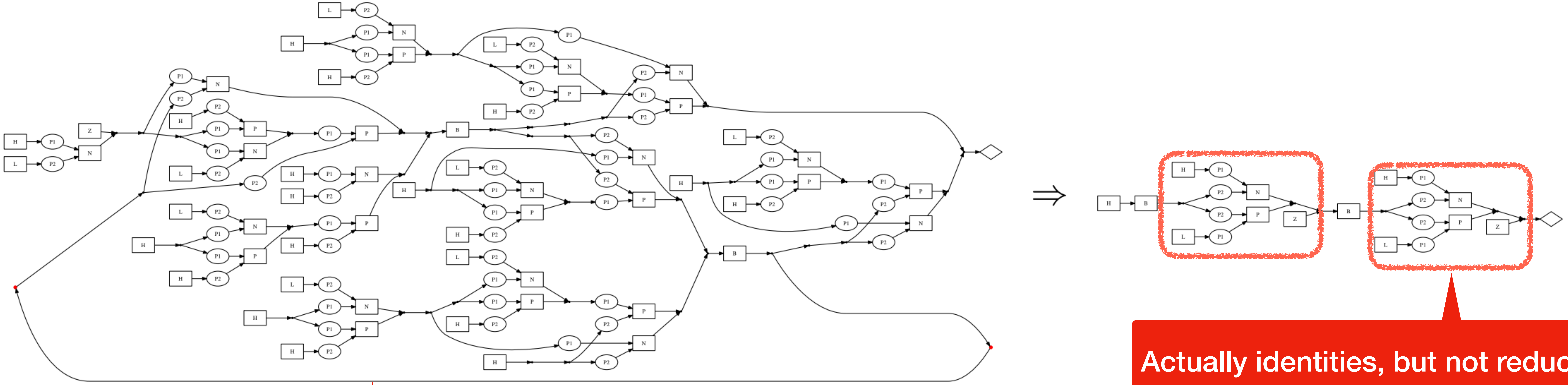




HUAWEI

Example: Pre-logical circuits

Transistor-level modelling // partial evaluation



Same circuit, implemented with pass-through CMOS logic.

Actually identities, but not reducible equationally with what we have.

A foundational issue

Diagrammatic reasoning is sound and complete

arXiv:2010.06319 [pdf, other] math.CT

The Graphical Language of Symmetric Traced Monoidal Categories

Authors: George Kaye

Abstract: We examine a variant of hypergraphs that we call linear hypergraphs, with the aim of creating a sound and complete graphical language for symmetric traced monoidal categories (STMCs). We first define the category of linear hypergraphs as a full subcategory of conventional (simple) hypergraphs, in which each vertex is either the source or the target of exactly one edge. The morphisms of a freely ge... ▽ More

Submitted 21 October, 2020; **v1** submitted 13 October, 2020; **originally announced** October 2020.

Comments: updated notation and appendices, technical report, 56 pages

Bonus: equations can be expressed as a graph rewriting system

Corollary 65. *$LHyp_\Sigma$ is a partial adhesive category.*

Graphs / Graph rewriting ⇔ Categorical TSMCC terms / Equations

Thank you!

HUAWEI