A Fully Compositional Theory of Digital Circuits

George Kaye

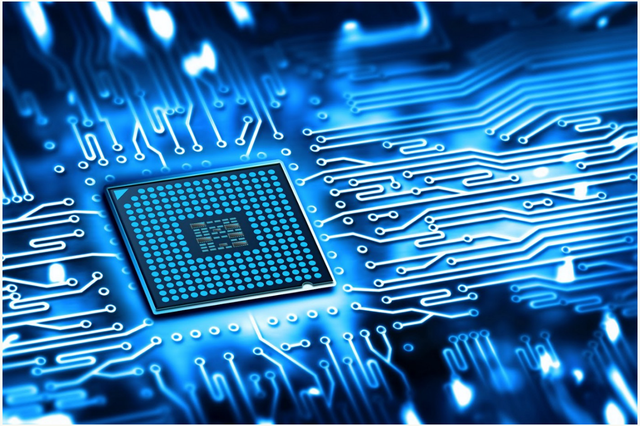
University of Birmingham

05 April 2024

5th Meeting of the Southern and Midlands Logic Seminar

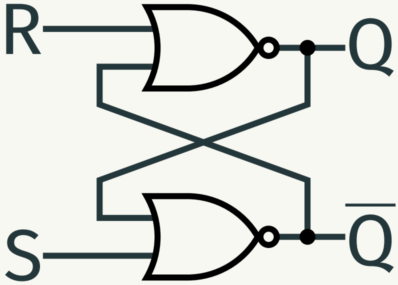
What are we going to be talking about?

Digital circuits!



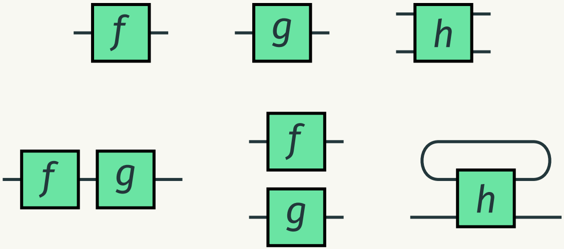
What are we going to be talking about?

Digital circuits!



What are we going to be talking about?

We want a compositional theory of digital circuits.



Using string diagrams removes much of the bureacracy

(also they look pretty)

How did we get here?

The story so far

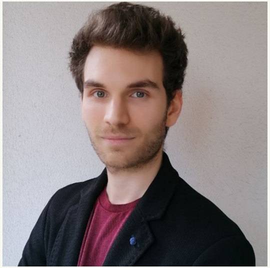
2003

Yves Lafont

'Towards an algebraic theory of Boolean circuits'

2016





Dan Ghica, Achim Jung, Aliaume Lopez

'Diagrammatic semantics for digital circuits'

The story so far

2019

The story so far

‘Do you know category theory’

‘Do you want to do circuits stuff’

‘No’

‘Okay’

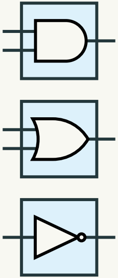
David Sprunger

‘I will help too’

Syntax

Combinational circuit components

gates

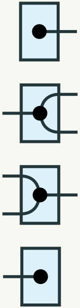


AND gate

OR gate

NOT gate

(co)monoid structure



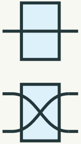
introduce

fork

join

eliminate

categorical structure



identity

symmetry

Light circuits 

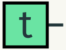
only contain gates and structure.

(actually, we do it more generally than this, but let's keep it simple)

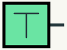
Sequential circuit components

Values

false



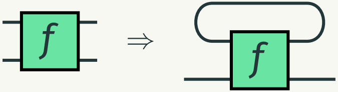
true



short circuit

Delay

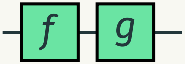
Feedback

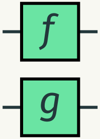


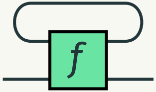
Dark circuits 

may contain delay or feedback.

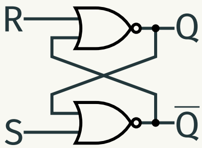
Circuits are morphisms in a freely generated symmetric traced monoidal category (STMC).

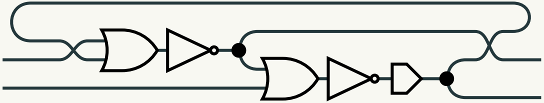






Need an example?



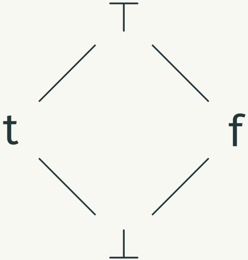


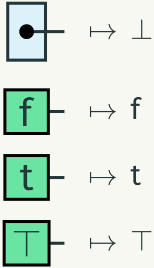
We need some meaning

What is the meaning?

Denotational semantics

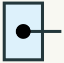
Values are interpreted in a lattice:



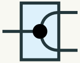


Let's make everything a function

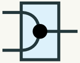
monotone functions $\overline{g} : \mathbf{V}^m \to \mathbf{V}$



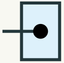
initialise $() \mapsto (\bot)$



copy $x \mapsto (x, x)$



join in the lattice $(x, y) \mapsto x \sqcup y$



discard $x \mapsto ()$

Feedback is interpreted as the least fixed point.

Functions are not enough

How do we model delay?

Streams!

Streams

A stream $\mathbf{V}^\omega$ is an infinite sequence of values.

$$v_0 :: v_1 :: v_2 :: v_3 :: v_4 :: v_5 :: v_6 :: v_7 :: \cdots$$

A stream function $\mathbf{V}^\omega \to \mathbf{V}^\omega$ consumes and produces streams.

$$f(v_0 :: v_1 :: v_2 :: v_3 :: v_4 :: \cdots) = w_0 :: w_1 :: w_2 :: w_3 :: w_4 :: \cdots$$

Interpreting the sequential components

$$\boxed{v}\!-\ () := v :: \perp :: \perp :: \perp :: \cdots$$

 $$(v_0 :: v_1 :: v_2 :: \cdots) := \perp :: v_0 :: v_1 :: v_2 :: \cdots$$

Maybe there are too many streams

Does every stream function $(\mathbf{V}^m)^\omega \to (\mathbf{V}^n)^\omega$ correspond to a circuit?

No.

(but this is to be expected!)

Restricting the stream functions

Circuits are causal.

They can only depend what they've seen so far.

Circuits are monotone.

They are constructed from monotone functions.

Circuits are finitely specified.

Their streams have finitely many stream derivatives.

These are the streams we're looking for

Theorem

A stream function is the interpretation of a sequential circuit if and only if it is ***causal, monotone*** *and has* ***finitely many stream derivatives****.*

Sound and complete denotational semantics!

Suppose we have two circuits
with the same denotation

$$\llbracket -\boxed{f}- \rrbracket = \llbracket -\boxed{g}- \rrbracket$$

What does this tell us about the
structure of these circuits?

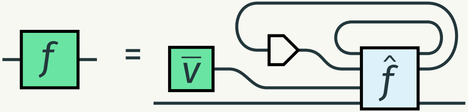
Operational semantics

Reducing it down

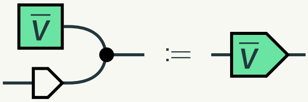
We want to find a set of reductions for digital circuits

We want to reduce circuits to their outputs syntactically in a step-by-step manner

Going global

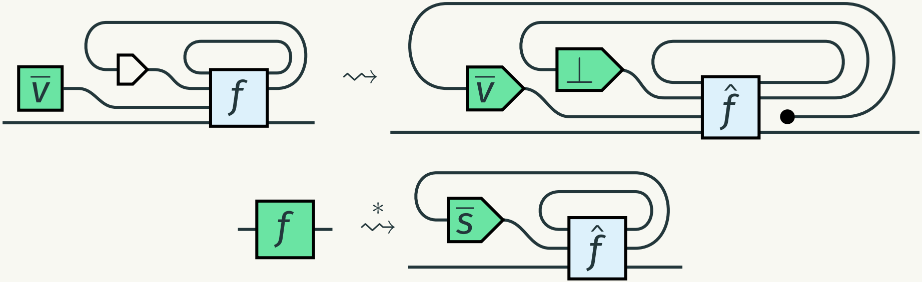


by moving boxes and wires around



$\overline{v}$
:=
$\overline{v}$

‘Register’



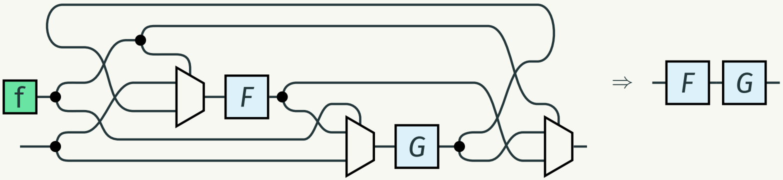
$\overline{v}$
f
⇝
$\overline{v}$
$\perp$
$\hat{f}$
f
$\overset{*}{⇝}$
$\overline{s}$
$\hat{f}$

The sticking point

What are we going to do about the non-delay-guarded trace?

In industry, feedback is usually delay-guarded.

But this rules out some clever circuits!



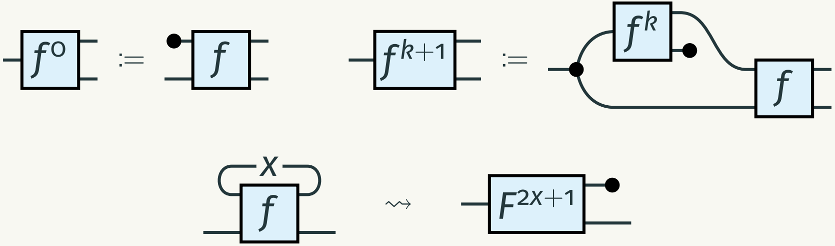
(And also it would be cheating)

V is a finite lattice...

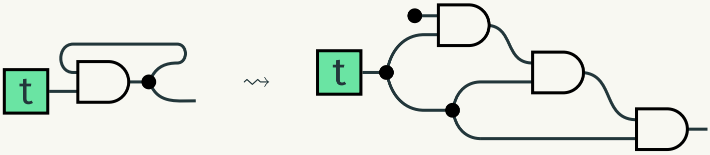
The functions are monotone...

We can compute the least fixed point
in finite iterations!

Getting rid of non-delay-guarded feedback



Getting rid of non-delay-guarded feedback

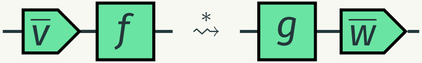


Here’s Mealy

For any circuit

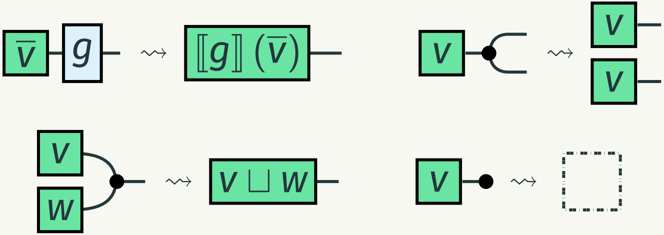
What is the goal

We want to compute the outputs of circuits given some inputs



How does a circuit process a value?

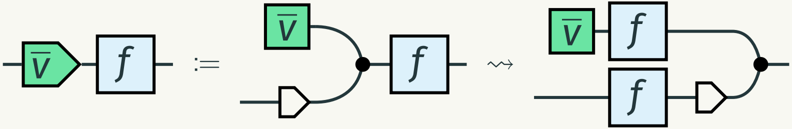
Reducing values



Lemma

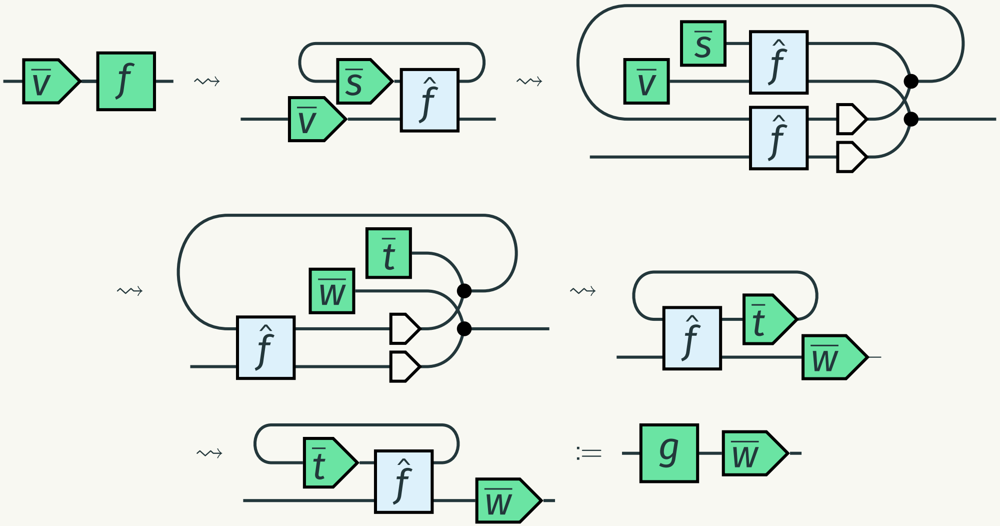
For every $-\boxed{f}-$ *there exists* $\boxed{\overline{w}}-$ *s.t.* $\boxed{\overline{v}}-\boxed{f}- \overset{*}{\rightsquigarrow} \boxed{\overline{w}}-$.

What about delays?



'Streaming'

Catching the jet stream



When are two circuits observationally equivalent?

Circuits have finitely many states...

Definition

Two circuits with at most c delay components are observationally equivalent if the reduction procedure creates the same outputs for all inputs of length $|\mathbf{V}|^c + 1$.

Observe this

Theorem

Two circuits are observationally equivalent if and only if they are denotationally equivalent.

Sound and complete operational semantics!

Still takes a while

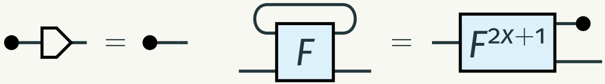
This is a superexponential upper bound for testing circuit equivalence

Can we do better?

Algebraic semantics

Mealy is so back

First things first...



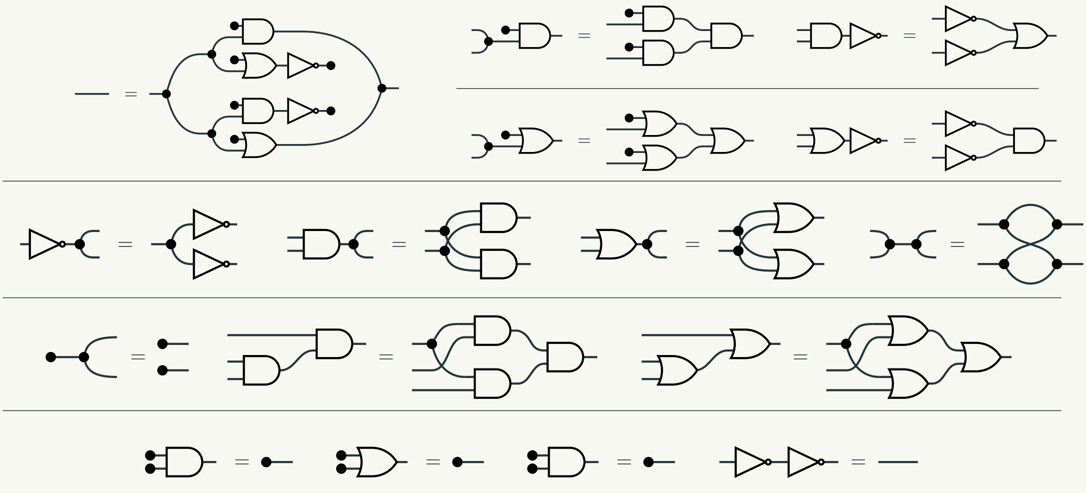
By these equations,

Say we have a procedure $|-|$ for establishing a canonical circuit for a function $f : \mathbf{V}^m \to \mathbf{V}^n$

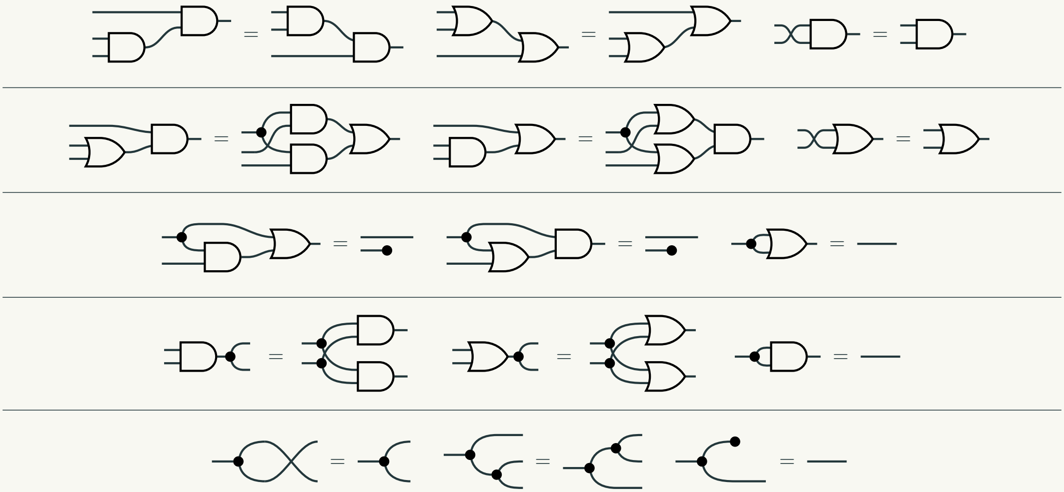
A circuit is normalised if it is in the image of $|-|$

What equations are needed to normalise any circuit?

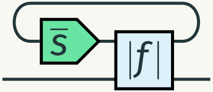
It’s completely normal

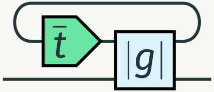


It’s completely normal



Changing the states

How to translate between 

and 

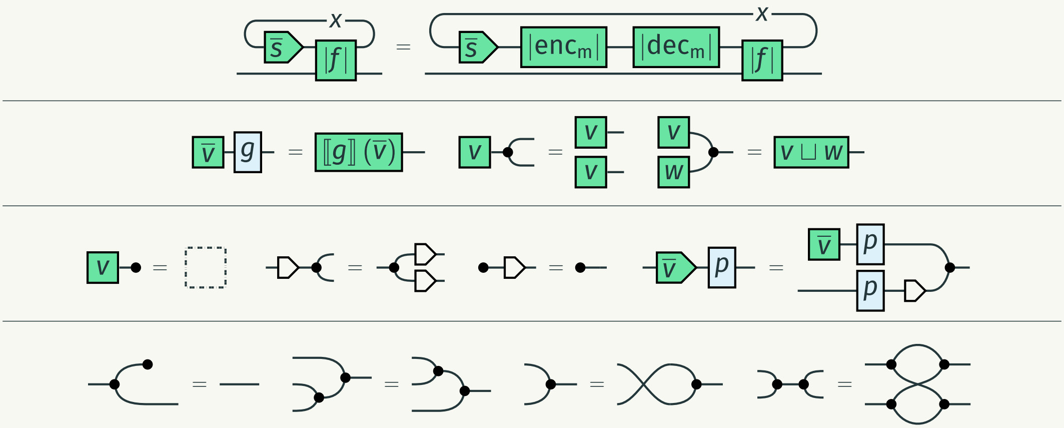
?

First encode one set of states into the other

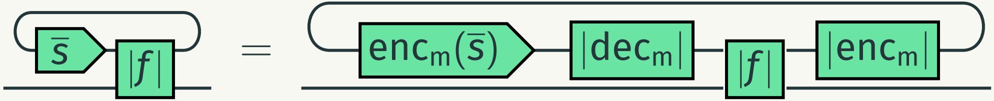
$$\boxed{\bar{s}}-\boxed{|\mathrm{enc}_{\mathrm{m}}|}- \;=\; \boxed{\bar{t}}- \qquad\qquad \boxed{\bar{t}}-\boxed{|\mathrm{dec}_{\mathrm{m}}|}- \;=\; \boxed{\bar{s}}-$$

(and for any future states)

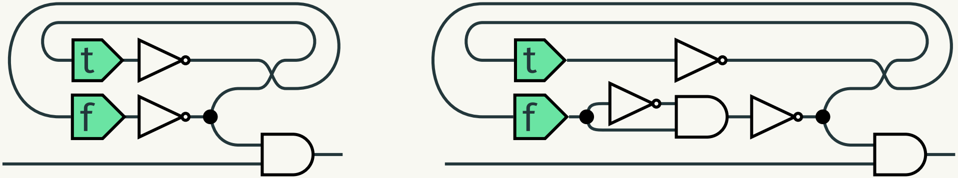
Changing the states



With these equations we can derive

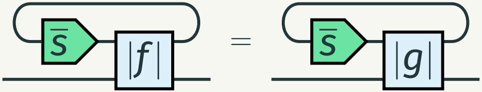


Is this enough?



The cores may not have the same semantics!

Think about what matters



where f and g ‘agree on the states that matter’

It is complete

By the completeness of the denotational semantics, each stream function has a corresponding encoded circuit...

Theorem

Two circuits are equal by the equations if and only if they are denotationally equal.

Sound and complete algebraic semantics!

Three different semantics for sequential digital circuits

$$\begin{tikzcd}
 & \text{Denotational} \arrow[dl, leftrightarrow, "\cong"'] \arrow[dr, leftrightarrow, "\cong"] & \\
\text{Operational} \arrow[rr, leftrightarrow, "\cong"'] & & \text{Algebraic}
\end{tikzcd}$$