A Fully Compositional Theory of Digital Circuits

George Kaye

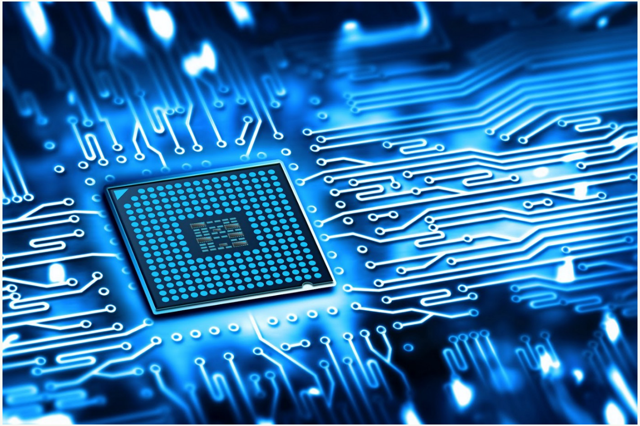
University of Birmingham

21 June 2024

Applied Category Theory 2024 (ACT 2024)

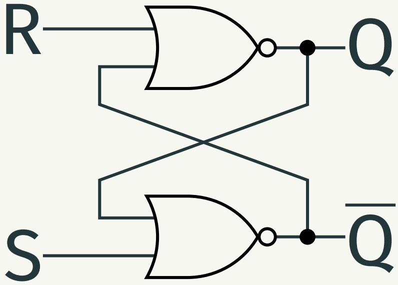
What are we going to be talking about?

Digital circuits!



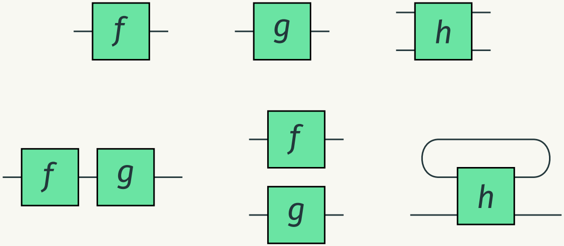
What are we going to be talking about?

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

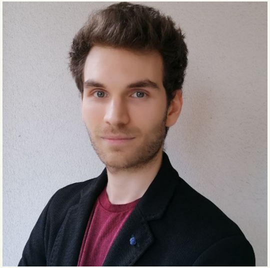
2003

Yves Lafont

'Towards an algebraic theory of Boolean circuits'

2016





Dan Ghica, Achim Jung, Aliaume Lopez

'Diagrammatic semantics for digital circuits'

2019

The story so far

David Sprunger

'Do you know category theory'

'Do you want to do circuits stuff'

'No'

'Okay'

'I will help too'

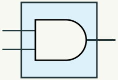
Hold on a second...

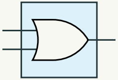
Hold on a second...

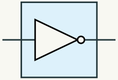
Syntax

Combinational circuit components

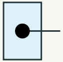
gates

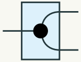
 AND gate

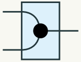
 OR gate

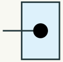
 NOT gate

(co)monoid structure

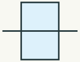
 introduce

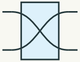
 fork

 join

 eliminate

categorical structure

 identity

 symmetry

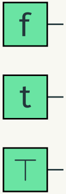
Light circuits 

only contain gates and structure.

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

Sequential circuit components

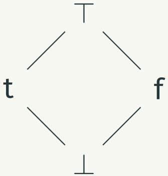
Values



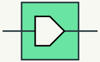
false

true

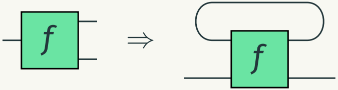
short circuit



Delay



Feedback

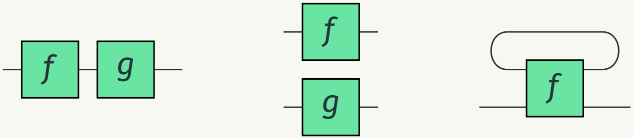


Dark circuits 

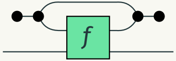
may contain

delay or feedback.

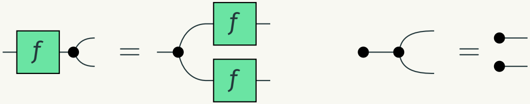
Circuits are morphisms in a freely generated symmetric traced monoidal category (STMC).

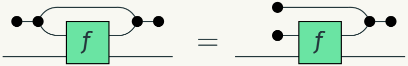


Why not use Frobenius structure?



We want copying...





Where were we?

What is the meaning?

What are the denotational semantics of digital circuits?

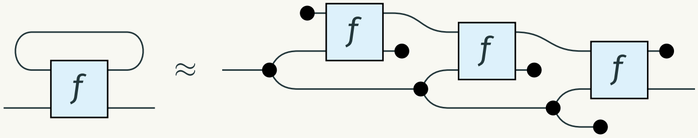
Certain kinds of stream functions!

$$f(v_0 :: v_1 :: v_2 :: \ldots) = w_0 :: w_1 :: w_2 :: \ldots$$

Denotational equivalence

$$[\![\, -\boxed{f}- \,]\!] = [\![\, -\boxed{g}- \,]\!] \Rightarrow -\boxed{f}- \approx -\boxed{g}-$$

We can also eliminate non-delay-guarded feedback



(Kleene fixpoint theorem)

We want something different

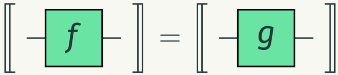
Denotational equivalence obscures the structure of terms

We want to reason more syntactically

Operational semantics	Algebraic semantics
a bit has changed	(pretty much) new

Doing something useful

Suppose we have two circuits
with the same denotation



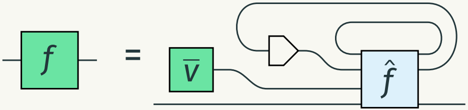
What does this tell us about the
structure of these circuits?

Reducing it down

We want to find a set of reductions for digital circuits

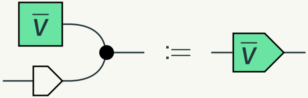
We want to reduce circuits to their outputs syntactically in a step-by-step manner

Going global

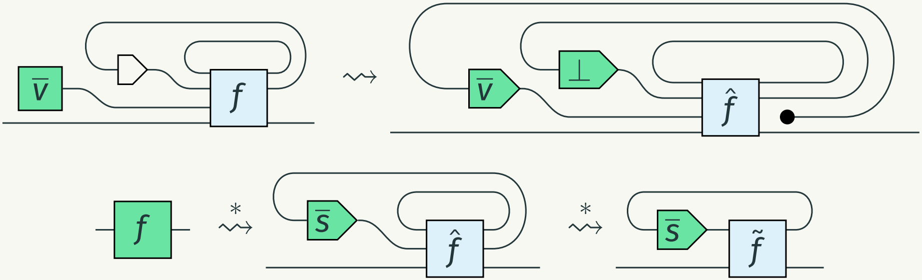


by moving boxes and wires around

Going global

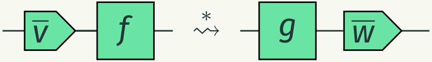


‘Register’



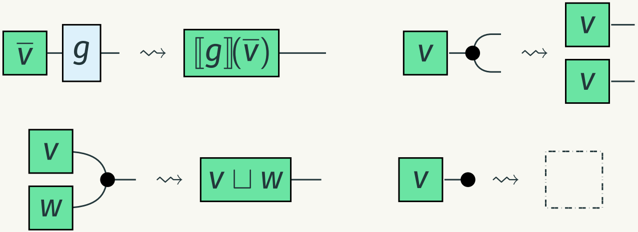
What is the goal

We want to compute the outputs of circuits given some inputs



How does a circuit process a value?

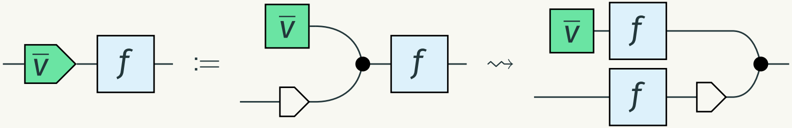
Reducing values



Lemma

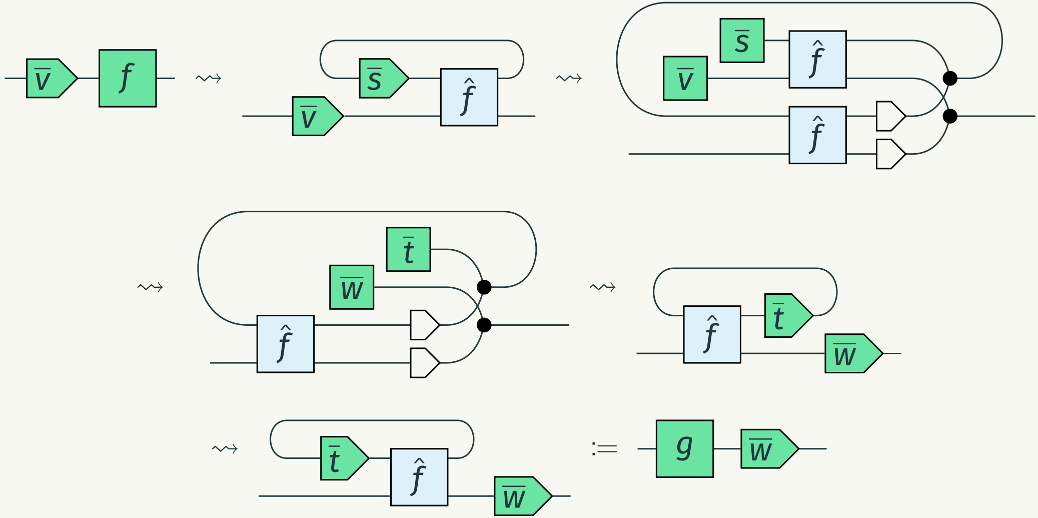
For every $-\boxed{f}-$ *there exists* $\boxed{\overline{w}}-$ *s.t.* $\boxed{\overline{v}}-\boxed{f}- \overset{*}{\rightsquigarrow} \boxed{\overline{w}}-$.

What about delays?



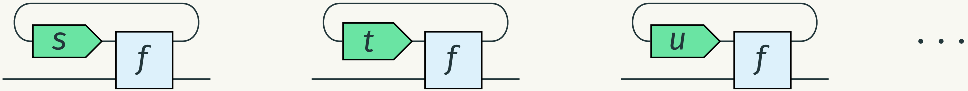
'Streaming'

Catching the jet stream



When are two circuits observationally equivalent?

Circuits have finitely many states...



Maximum number of states: $|\mathbf{V}|^{\text{number of delays}}$

Observe this

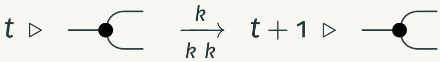
Two circuits are observationally equivalent if the reduction procedure creates the same outputs for all inputs of length $|\mathbf{V}|^{\text{max number of delays}} + \mathbf{1}$.

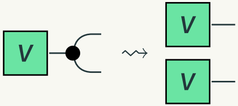
Denotational semantics $\cong$ Operational semantics

A little bit different

This operational semantics is a bit different to some others...

(cf. signal flow graphs)





We want to transform the circuit

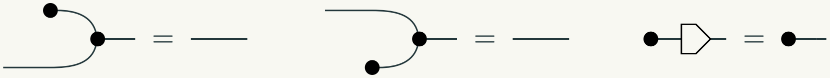
Still takes a while

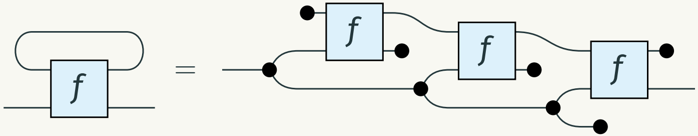
This is a superexponential upper bound for testing circuit equivalence

Can we do better?

Mealy is so back

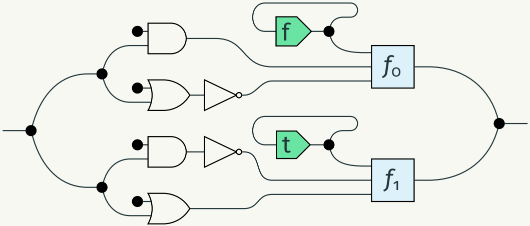
First things first...





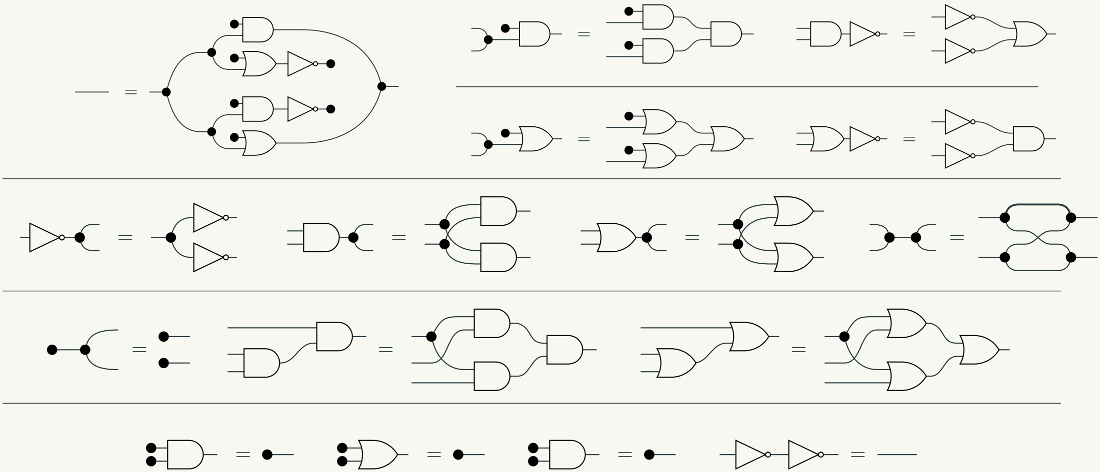
By these equations,

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

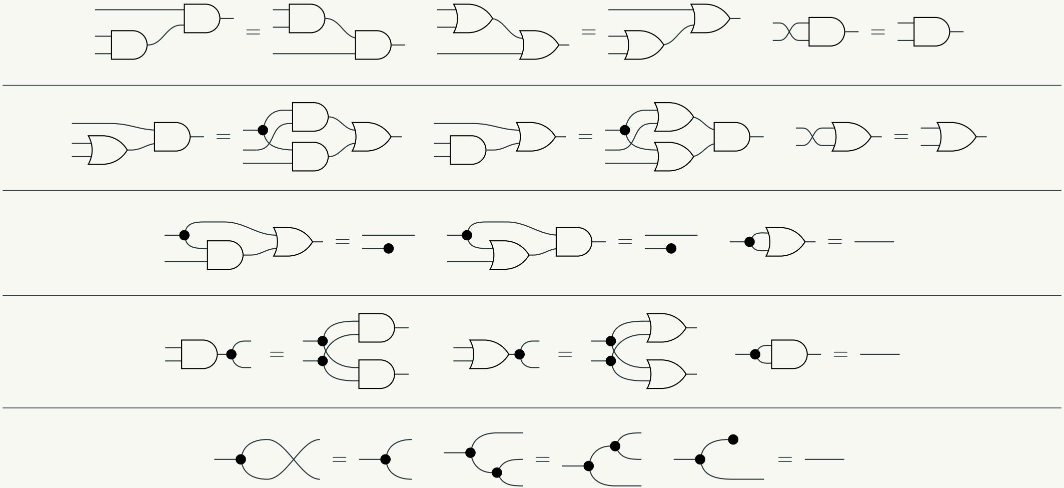


A circuit is normalised if it is in the image of $||-||$

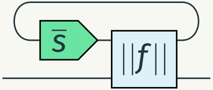
It’s completely normal



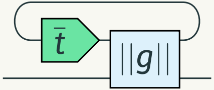
It’s completely normal



How to translate between

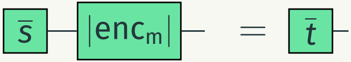


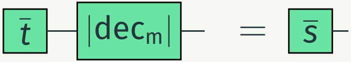
and



?

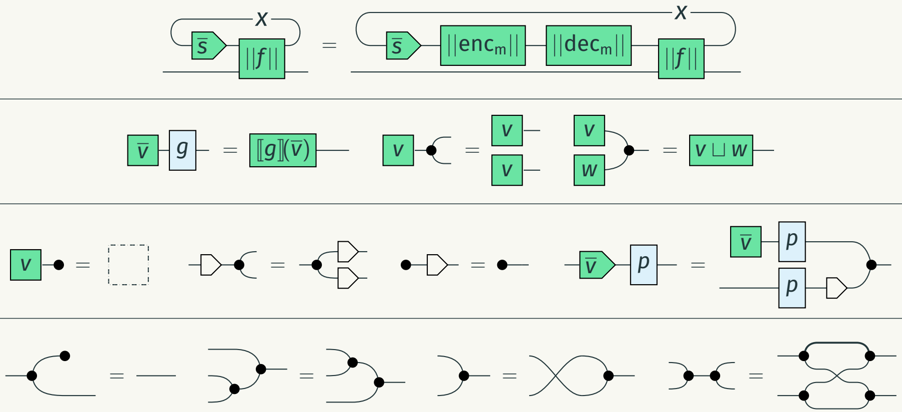
First encode one set of states into the other





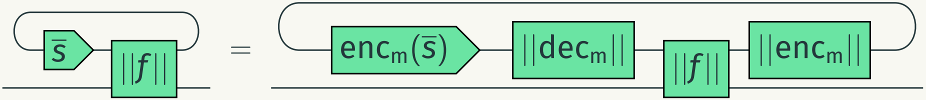
(and for any future states)

Changing the states

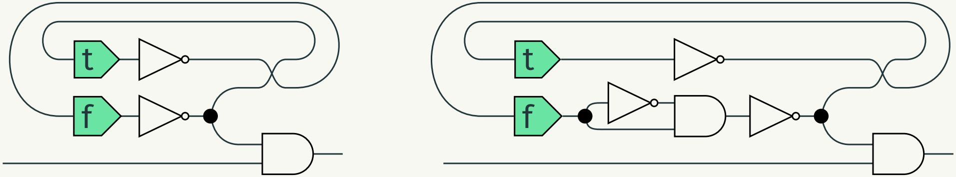


Changing the states

With these equations we can derive

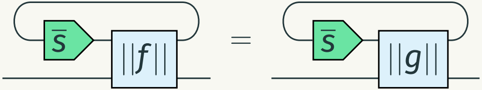


Is this enough?



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

Denotational semantics $\cong$ Algebraic semantics

To the end

Three different semantics for sequential digital circuits