Normalisation by evaluation for digital circuits

George Kaye, Dan R. Ghica, David Sprunger
University of Birmingham

13 December 2021

SYCO 8

Introduction

Digital circuits are ubiquitous in today's society.

Normally evaluated by simulating automata.

What about step-by-step operational semantics?

Previous work: Ghica, Jung and Lopez (2016, 2017)

Not necessarily complete: problems with 'instant' feedback.

This talk

We will first recap the existing categorical circuit framework.

Extend this to properly handle instant feedback.

Verify its correctness with Mealy machines and streams.

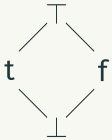
See how this can be used as a ‘normalisation by evaluation’ for digital circuits.

Categorical semantics for digital circuits

Circuit signature

Values $\mathbf{V}$ forming a lattice.

$\mathbf{V} =$



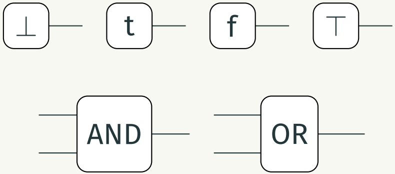
Gate symbols g with associated monotonic functions $\bar{g} : \mathbf{V}^m \to \mathbf{V}$.

$$\{\ \mathrm{AND} : \mathbf{V}^2 \to \mathbf{V},\ \mathrm{OR} : \mathbf{V}^2 \to \mathbf{V}\ \}$$

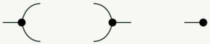
Combinational circuits

Circuits are morphisms in the prop generated freely over a signature Σ.

e.g:

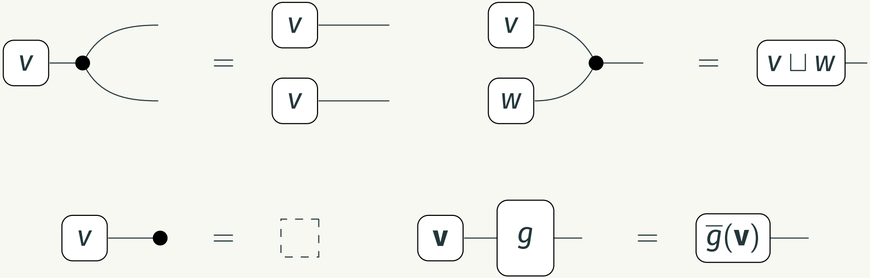


Along with structural generators



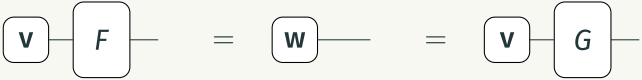
Combinational circuits: axioms

Behaviour defined using axioms.



Combinational circuits: extensional equivalence

We also consider input-output behaviour.



F and *G* are extensionally equivalent.

The prop of combinational circuits

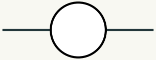
generators $+$ axioms

$+$ quotient by extensional equivalence $=$ $\mathbf{CCirc}_\Sigma$

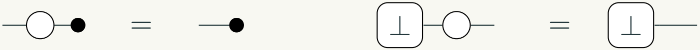
Temporal circuits

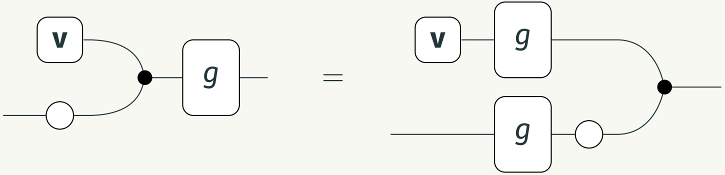
Combinational circuits are boring.

Delay is represented by a new generator

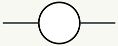


Temporal circuits: axioms





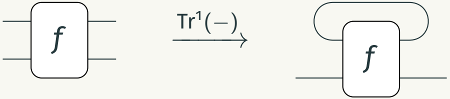
The prop of temporal circuits

$\mathbf{CCirc}_{\Sigma}$ +  + axioms = $\mathbf{TCirc}_{\Sigma}$

Sequential circuits

Sequential circuits have delay and feedback.

We freely add a trace operator.



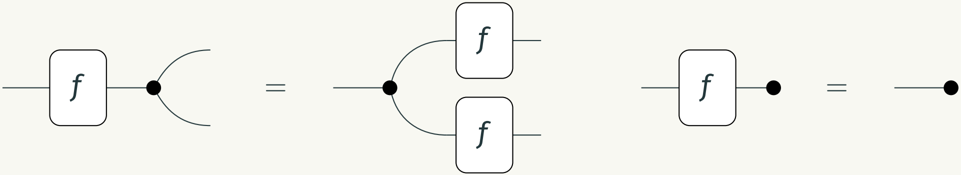
$$\mathbf{TCirc}_\Sigma \quad + \quad \text{trace} \quad = \quad \mathbf{SCirc}_\Sigma$$

Sequential circuits

Theorem (Ghica and Jung, 2016)

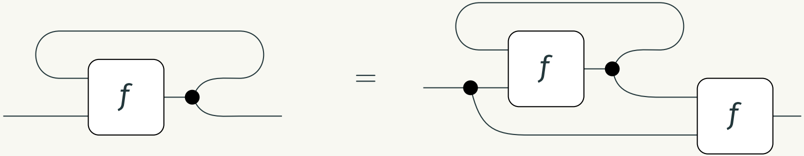
$\mathbf{SCirc}_\Sigma$ is cartesian.

We can copy and discard data.



Sequential circuits: unfolding

Traced cartesian categories admit the unfolding rule.

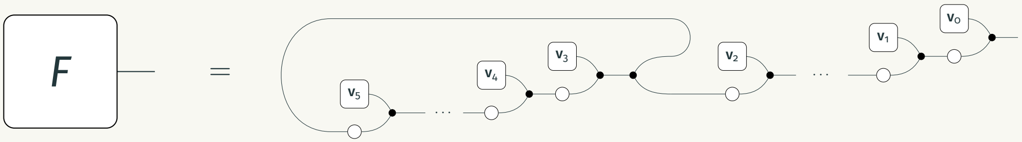


Crucial part of the operational semantics!

Evaluating digital circuits

Productivity

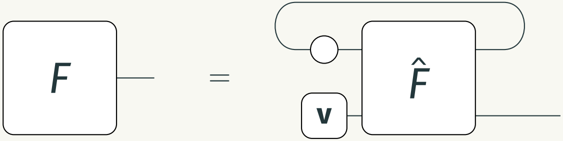
For closed circuits the aim is to reduce to a (possibly infinite) sequence of values.



Circuits that do this are called productive.

Delay-guarded feedback

Some circuits have delay-guarded feedback.



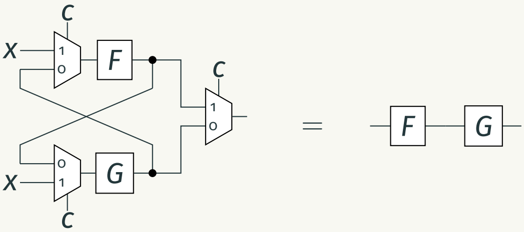
where $\hat{F}$ is combinational.

Theorem (Ghica, Jung and Lopez, 2017)

Circuits with delay-guarded feedback are productive.

Cyclic combinational circuits

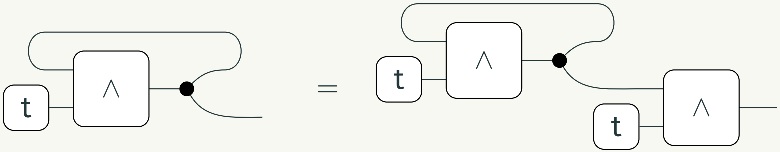
But not all non-delay-guarded circuits are unproductive!



These are called cyclic combinational circuits.

Unproductive circuits

However some non-delay-guarded circuits are unproductive...



Unproductive circuits – what to do?

Ban non-delay-guarded circuits?

This would mean we can't model cyclic combinational circuits.

Also implies we are working in a category with delayed trace.

We would lose the unfolding rule.

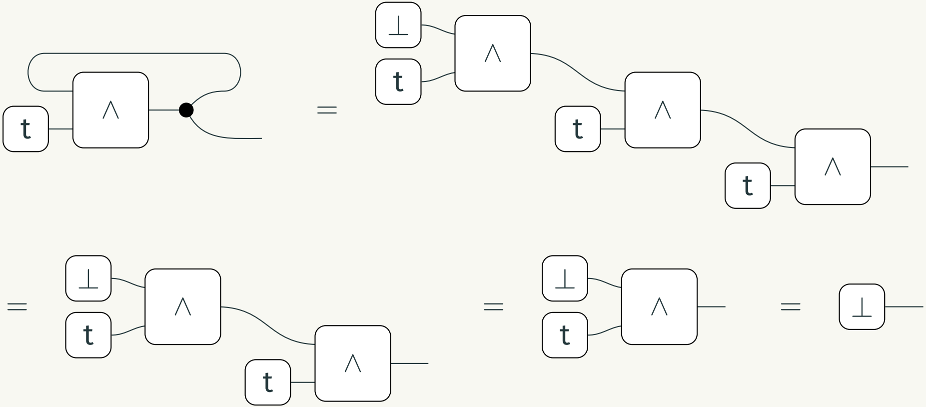
Eliminating 'instant' feedback

Our gates are monotonic, so they must have a least fixed point...

$$f^i(\bot) = f^{i+1}(\bot)$$

Because the value set **V** is finite, we can always find this fixpoint!

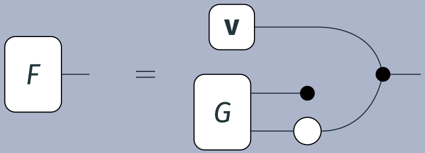
Eliminating ‘instant’ feedback



Productivity, redux

Theorem

For any circuit F *, there exists values* **v** *and circuit* G *such that*



All circuits are productive!

Is this all correct?

We have extended the existing axiomatisation of digital circuits to work with instant feedback.

But is it correct?

We must compare with the denotational semantics.

The denotational semantics of circuits

Circuits $m \to n \Rightarrow$ functions $(\mathbf{V}^m)^\omega \to (\mathbf{V}^n)^\omega$.

How do we translate a circuit into a stream?

We will take a route via Mealy machines.

Mealy machines

Mealy machine

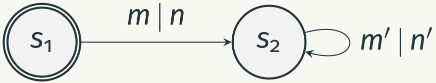
Mealy machines are a kind of finite state machine.

Sets of states S, inputs M, outputs N

Given input $m \in M$ and state $s_1 \in S$ we have:

- next state $T(s_1)(m) = s_2 \in S$
- output $O(s_1)(m) = n \in N$.

A Mealy machine also has a start state.



Mealy machines: bisimilarity

Mealy machines have a notion of bisimilarity.

If two machines are observationally equivalent, then they are bisimilar.

Props of mealy machines

By setting M and N to powers of $\mathbf{V}$, we can define a prop of Mealy machines.

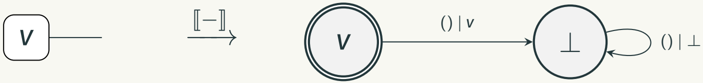
A morphism $m \to n$ is a Mealy machine with inputs $\mathbf{V}^m$ and outputs $\mathbf{V}^n$.

We can define composition, tensor, trace...

We interpret circuits as Mealy machines using a functor $[\![-]\!]$.

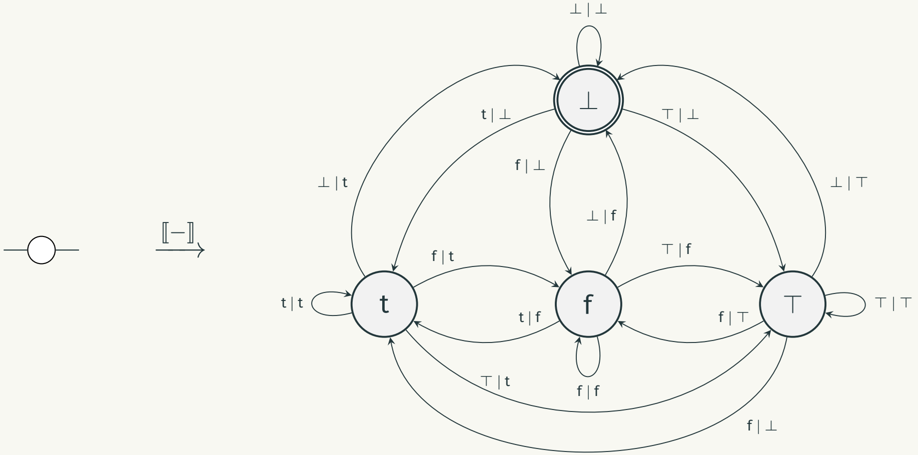
Interpreting SCirc_Σ

Source of state in our circuits: values...



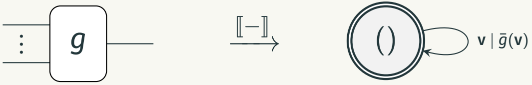
Interpreting SCirc_Σ

…and delay.



Interpreting $\mathbf{SCirc}_\Sigma$

Gates don’t have any ‘internal state’.



To interpret circuit morphisms, combine with composition, tensor, trace...

Interpreting $\mathbf{SCirc}_\Sigma$

Do all the axioms of $\mathbf{SCirc}_\Sigma$ hold in this prop? Yes.

Theorem

For any $F, G \in \mathbf{SCirc}_\Sigma$, if $F = G$ then $[\![F]\!]$ and $[\![G]\!]$ are bisimilar.

Can we return to a circuit from an arbitrary Mealy machine?

Should be possible, rudimentary task in circuit design.

But first, how do we get to streams?

Streams

The final Mealy coalgebra

A Mealy machine is a coalgebra of the functor $R(S) = (S \times N)^M$ in **Set**.

This means there is the notion of a final coalgebra...

The final Mealy coalgebra

$$
\begin{array}{ccc}
S & \xrightarrow{\quad h \quad} & M^\omega \to N^\omega \\
\big\downarrow{\scriptstyle \langle O,T \rangle} & & \big\downarrow{\scriptstyle \langle \mathrm{hd},\mathrm{tl} \rangle} \\
(N \times S)^M & \xrightarrow{\quad Rh \quad} & (N \times (M^\omega \to N^\omega))^M
\end{array}
$$

We compute the unique map $h : S \to (M^\omega \to N^\omega)$ as $O :: O \circ T :: O \circ T^2 :: \cdots$

The resulting stream is the outputs over time given an input stream.

Periodic streams

For now, we focus on closed circuits.

Our circuits are finite in nature.

They may produce an infinite sequence of outputs, but it will be periodic.

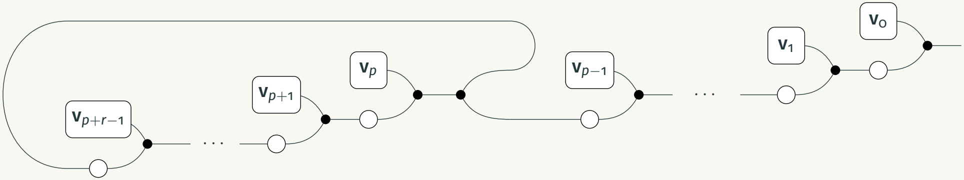
A stream σ is periodic if it has a finite prefix and an infinitely reoccurring period.

$$\mathbf{v}_0 :: \mathbf{v}_1 :: \cdots :: \mathbf{v}_{p-1} :: \mathbf{v}_p :: \mathbf{v}_{p+1} :: \cdots \mathbf{v}_{p+r-1} :: \mathbf{v}_p :: \mathbf{v}_{p+1} :: \cdots$$

From periodic streams to closed circuits

In the closed case, the carrier of the final coalgebra is N^ω.

$$\mathbf{v}_0 :: \mathbf{v}_1 :: \cdots :: \mathbf{v}_{p-1} :: \mathbf{v}_p :: \mathbf{v}_{p+1} :: \cdots \mathbf{v}_{p+r-1} :: \mathbf{v}_p :: \mathbf{v}_{p+1} :: \cdots$$



From periodic streams to closed circuits

$$\mathbf{SCirc}_\Sigma + \text{only closed circuits} = \mathbf{CSCirc}_\Sigma$$

$$\text{Periodic streams over } \mathbf{V}^n \text{ for some n} = \mathbf{PCStream}_\mathbf{V}$$

Theorem

$\mathbf{CSCirc}_\Sigma \cong \mathbf{PCStream}_\mathbf{V}$.

Normalisation by evaluation

We can translate a closed circuit morphism into a Mealy machine, then into a periodic stream and back to a circuit.

The resulting circuit will be a waveform of values.

This is a form of normalisation by evaluation.

Conclusion

Conclusion

Refined the categorical framework for circuits to eliminate instant feedback.

Reduce any circuit to a waveform of values.

Circuits implement Mealy machines, so we can show…

Isomorphism between closed circuits and periodic streams.

Can be used as a form of normalisation by evaluation.

Next step: open circuits – translating arbitrary Mealy machines back to circuits