Constructive Boolean Networks and the Exactness of (Timed) Ternary Simulation

Michael Mendler

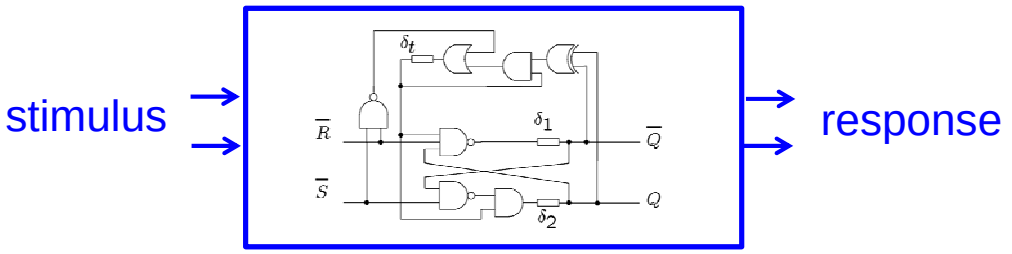
[joint work with Tom Shiple, Gérard Berry;
Formal Methods in System Design, Vol. 40, No. 3, June 2012]

What‘s this talk about ?

- formally characterise the class of Boolean networks known, informally, as „constructive“ systems (Berry, Shiple)
- present correspondence theorems linking denotational, operational and axiomatic semantics
- highlight that there are different notions of „causal“ or „combinational“ systems depending on the MoCC (model of coordination and communication)

COMBINATIONAL NETWORKS

Combinational Networks

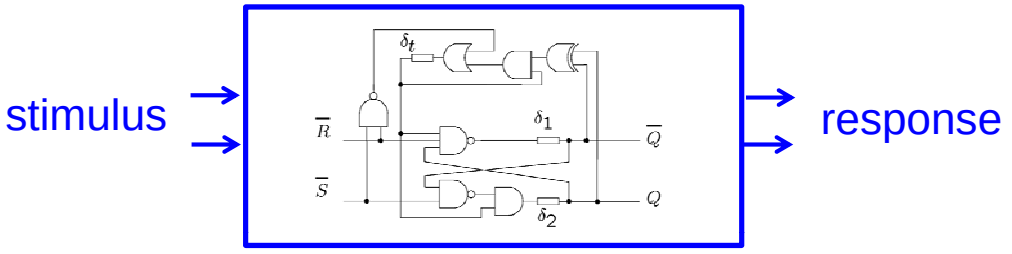


Definition (informal)

A Boolean network (circuit + delay nodes) is combinational if it realises a functional relationship input → output.

- Asynchronous Circuits
- Synchronous Programming (e.g., Esterel, Lustre, …)
- Communicating Mealy machines (e.g., Statecharts)

Combinational Networks



Definition (operational)

Let DEL be a network delay/scheduling model (MoCC).

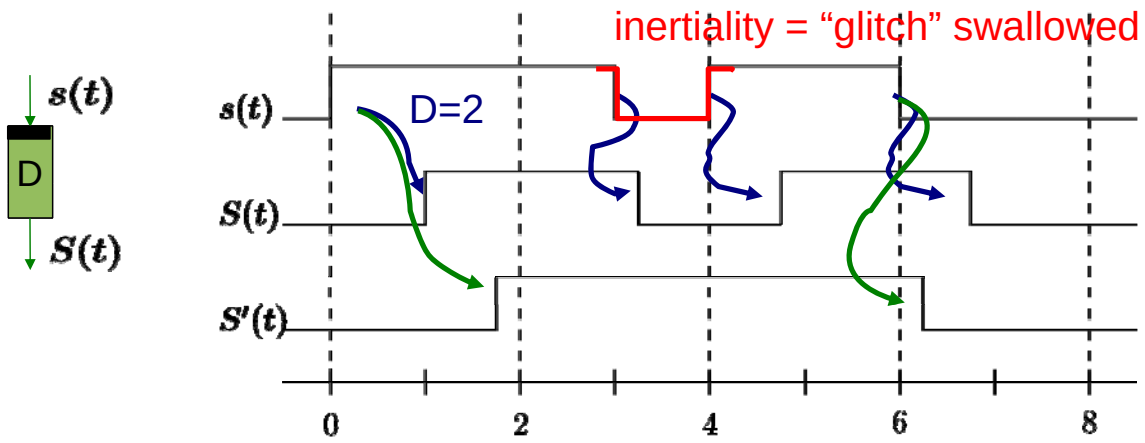
A network is DEL-combinational if for all constant input signals every output node

- stabilizes in bounded time
- to a unique response value

under DEL-execution semantics.

Up-bounded Inertial Delay (UIN)

[Huffman'54, Miller'65, Brzozowski/Seger'89]

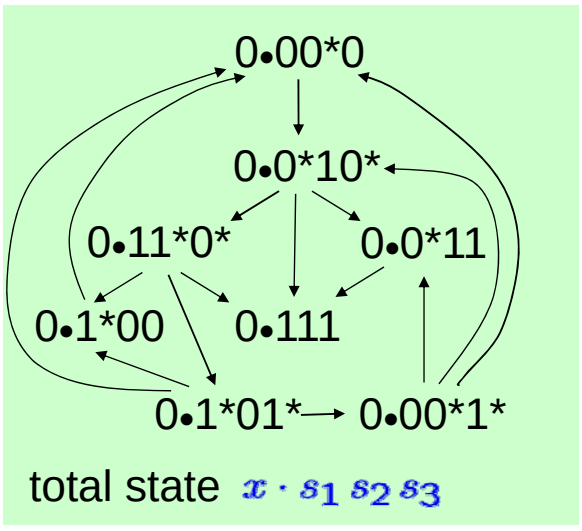


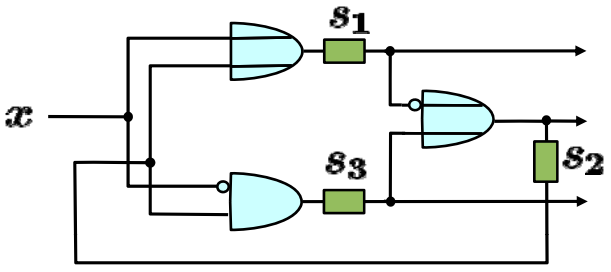
(1) Up-bounded Propagation: The delay cannot remain unstable for longer than D time without changing output

(2) Inertiality: The output only changes if delay is unstable

Example I

- Muller Diagram [Muller'56]
  UIN system trajectories

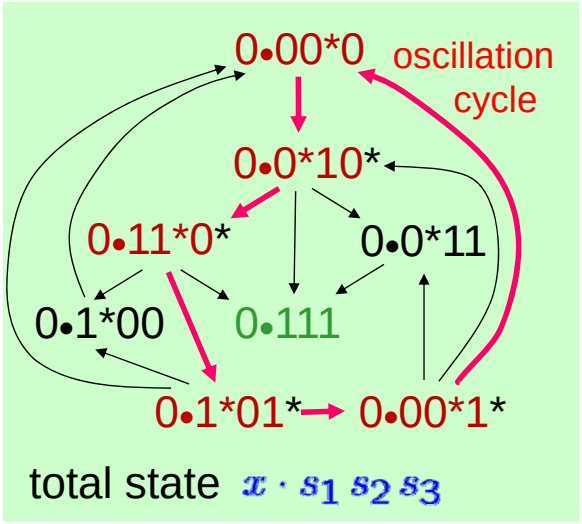


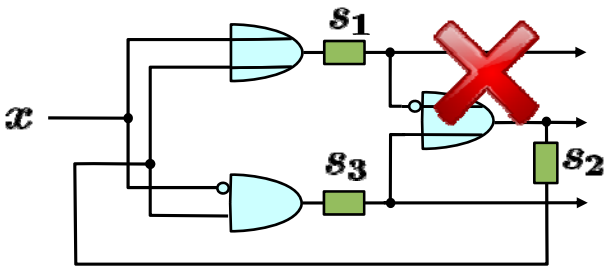


- Up-bounded Inertial Delay (UIN)
- General Multiple Winner Model (GMW) *[Huffman'54, Brzozowski/Yoeli 79]*

Example I

- Muller Diagram [Muller'56]
  UIN system trajectories





not UNI-combinational !

- Up-bounded Inertial Delay (UIN)
- General Multiple Winner Model (GMW)
  [Huffman'54, Brzozowski/Yoeli 79]

Example II

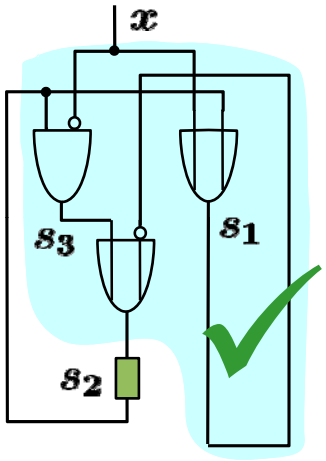
- Muller Diagram
  UIN system trajectories

total state $x \cdot s_2$

0•0* 1•1*

0•1 1•0

stable states



- All possible UIN-trajectories in the General Multiple Winner (GMW) model converge !

UIN-combinational

Boolean Paradise Lost

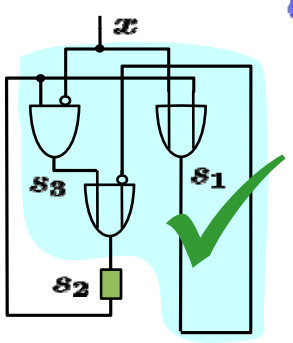
$$s_1 = s_2 + x$$
$$s_2 = \overline{s_1} + s_3$$
$$s_3 = \overline{x} \cdot s_2$$

$\approx$

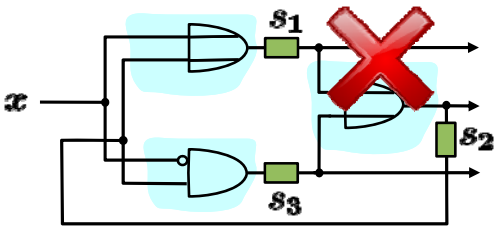
$$s_1 = 1$$
$$s_2 = \overline{x}$$
$$s_3 = \overline{x}$$

Boolean-combinational

Boolean Algebra

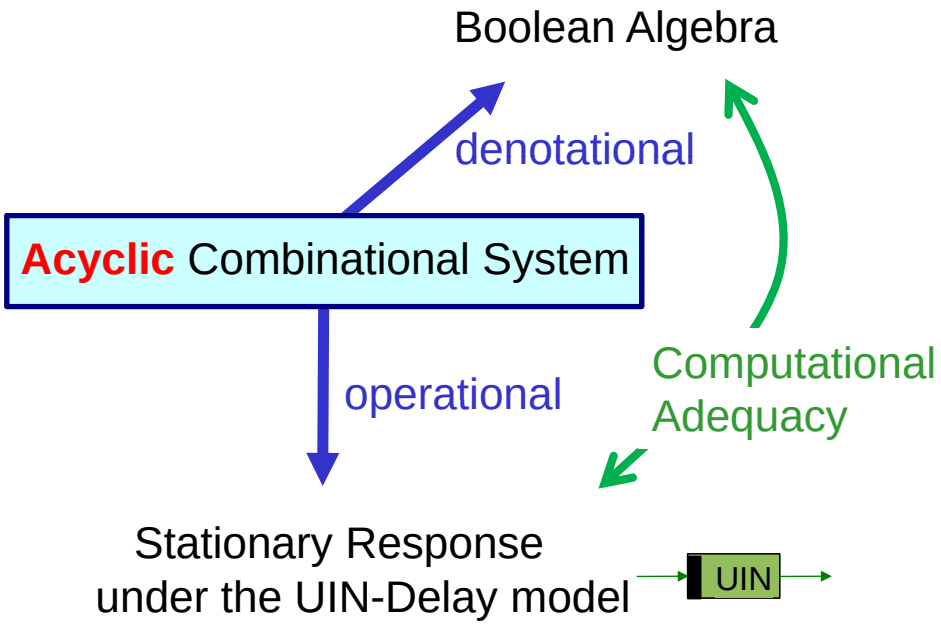


UIN-combinational

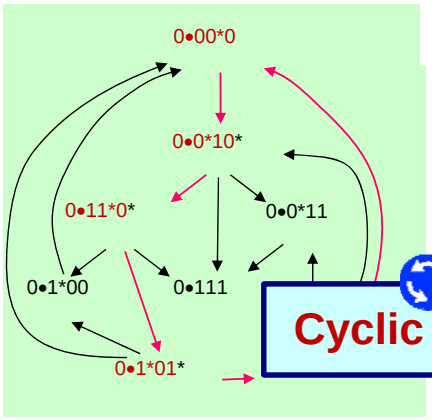


not UIN-combinational

The 2-valued "Steady State Paradise"



Cycles: “Paradise lost ?”



Complexity !

Muller-Moore Automata

denotational

Cyclic Combinational System

operational

Computational Adequacy

Transients irrelevant !

Transient Behaviour under the UIN-Delay model

UIN

Cycles: “Paradise lost ?”

What does the engineer do when 2 Booleans are not enough ? ...

?

denotational

Cyclic Combinational System

operational

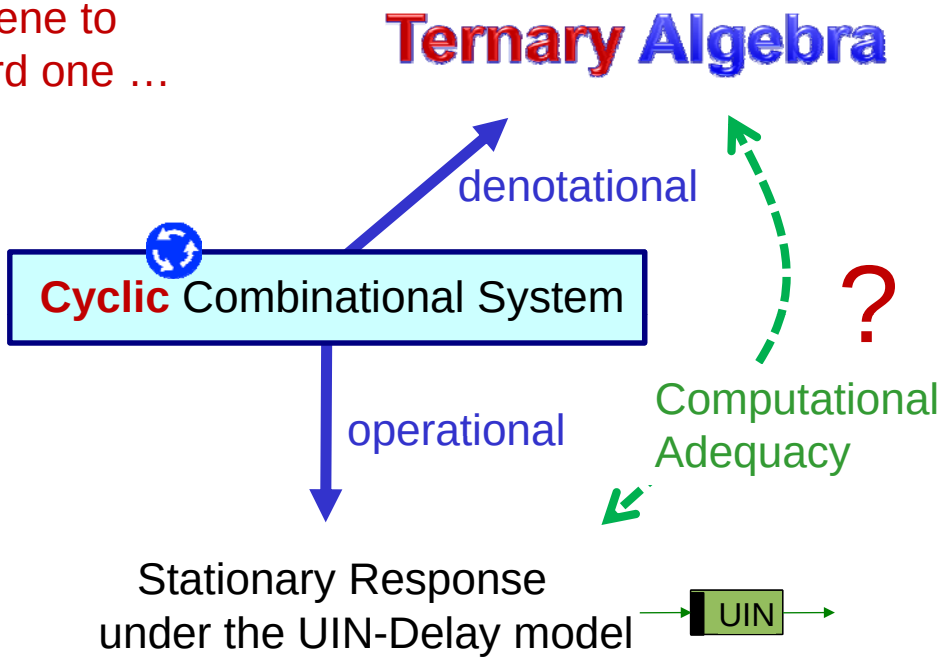
Computational Adequacy

Stationary Response under the UIN-Delay model

UIN

Cycles: “Paradise lost ?”

She asks Kleene to
give her a third one …



TIMED TERNARY SIMULATION

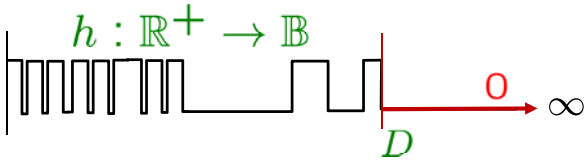
Timed Ternary Algebra

- Recursion theory [Kleene‘52]
- Asynchronous Circuits (hazards, races, oscillation)
  [Yoeli/Rinon‘64, Eichelberger‘65, Roth‘66]
  [Bryant‘87] CMOS transistor-level
- Ternary simulation [Yoeli/Brzozowski‘77, Brzozowski/Seger‘95]
- Cyclic combinational circuits
  [Burch/et.al.‘93, Malik‘93, Shiple‘96]
  [Huang/Parng/Shyu‘91] Timed D-calculus
  [Fairtlough/Mendler‘96] Real-time interpretation
  [Namjoshi/Kurshan‘99, Backes/Fett /Riedel‘08] Refined algorithm
- Synchronous programming [Berry‘99, Schneider/Brandt/Schuele‘04]

$\mathbf{x} = [0, D]$

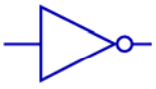
$\mathbb{TS} = \mathbb{B}_\perp \otimes \mathbb{R}^+_\infty$

Abstraction

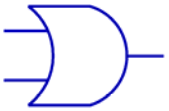


Timed Ternary Algebra

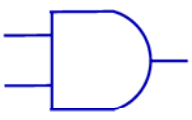
DEL(d)	
$[\alpha, s]$	$[\alpha, s+d]$
$\perp$	$\perp$



NOT	
$[\alpha, s]$	$[\overline{\alpha}, s]$
$\perp$	$\perp$

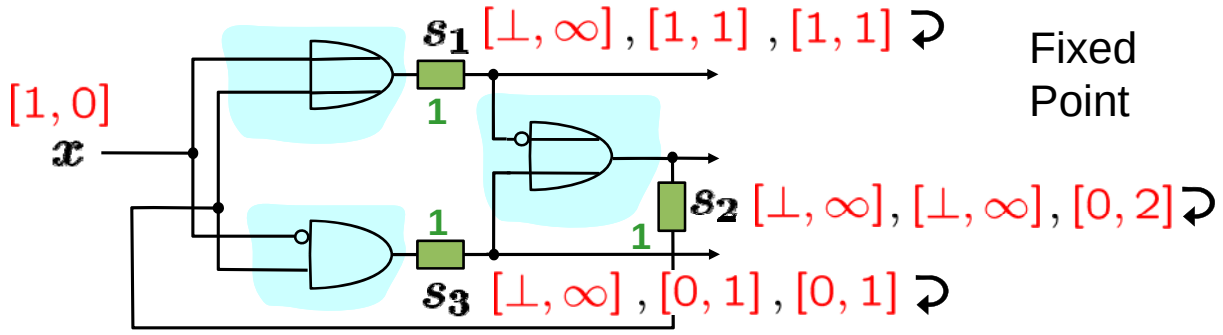


OR	$[0,t]$	$[1,t]$	$\perp$
$[0,s]$	$[0, max(s,t)]$	$[1,t]$	$\perp$
$[1,s]$	$[1,s]$	$[1, min(s,t)]$	$[1,s]$
$\perp$	$\perp$	$[1,t]$	$\perp$



AND	$[0,t]$	$[1,t]$	$\perp$
$[0,s]$	$[0, min(s,t)]$	$[0,s]$	$[0,s]$
$[1,s]$	$[0,t]$	$[1, max(s,t)]$	$\perp$
$\perp$	$[0,t]$	$\perp$	$\perp$

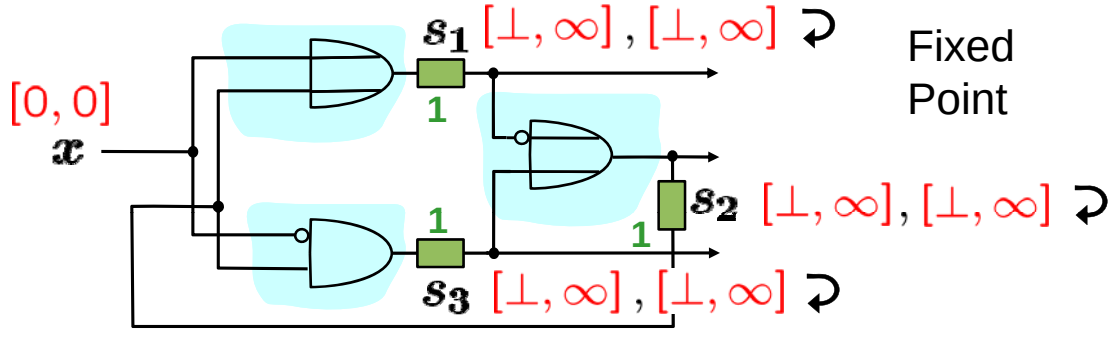
Example I



Fixed Point

$$\mathbf{s}_1^{i+1} = [let\ s_2 = \mathbf{s}_2^i\ in\ s_2 + x, 1]$$
$$\mathbf{s}_2^{i+1} = [let\ (s_1, s_3) = (\mathbf{s}_1^i, \mathbf{s}_3^i)\ in\ \overline{s_1} + s_3, 1]$$
$$\mathbf{s}_3^{i+1} = [let\ s_2 = \mathbf{s}_2^i\ in\ \overline{x} \cdot s_2, 1]$$

Example I



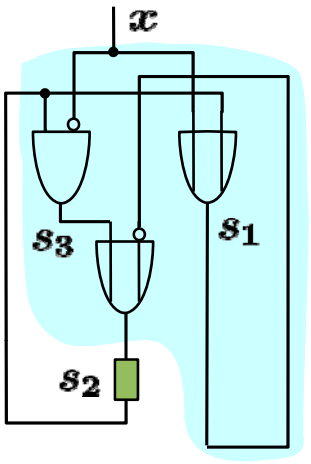
Fixed Point

$$\mathbf{s}_1^{i+1} = [let\ s_2 = \mathbf{s}_2^i\ in\ s_2 + x, 1]$$
$$\mathbf{s}_2^{i+1} = [let\ (s_1, s_3) = (\mathbf{s}_1^i, \mathbf{s}_3^i)\ in\ \overline{s_1} + s_3, 1]$$
$$\mathbf{s}_3^{i+1} = [let\ s_2 = \mathbf{s}_2^i\ in\ \overline{x} \cdot s_2, 1]$$

Example II

$$
\begin{aligned}
x &= [0,0] \\
s_2 &= [\bot,\infty], [1,1] \circlearrowright
\end{aligned}
$$

$$
\begin{aligned}
x &= [1,0] \\
s_2 &= [\bot,\infty], [0,1] \circlearrowright
\end{aligned}
$$

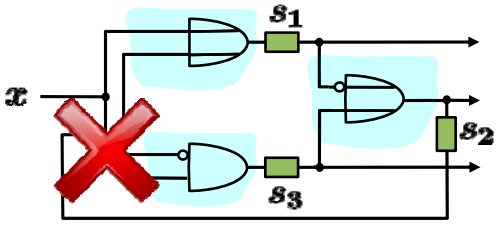


$$
\begin{aligned}
\mathbf{s}_2^{i+1} &= [\mathit{let}\, s_2 = \mathbf{s}_2^i \;\mathit{in}\; \overline{(s_2 + x)} + (\overline{x} \cdot s_2), 1] \\
&= [\mathit{let}\, s_2 = \mathbf{s}_2^i \;\mathit{in}\; \overline{x}, 1] \\
&= [\overline{x}, 1]
\end{aligned}
$$

Ternary Combinational Network

Definition

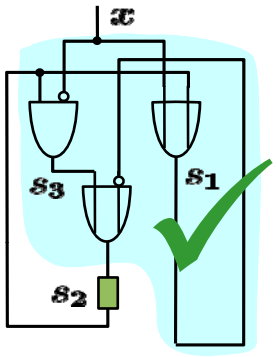
A network (fixed set of state nodes) is ternary combinational if the least fixed point of its timed ternary extension produces bounded-time Boolean values for all (static) input vectors.



not ternary combinational
not UIN-combinational

ternary combinational

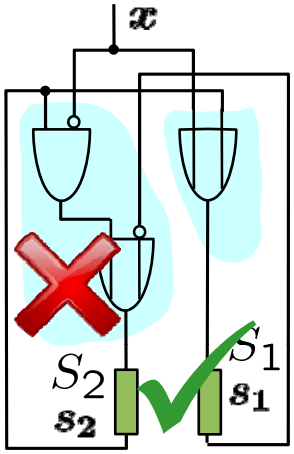
UIN-combinational



Problem Solved ?

THE ABSTRACTION GAP

UIN-Delay Full Abstraction Problem



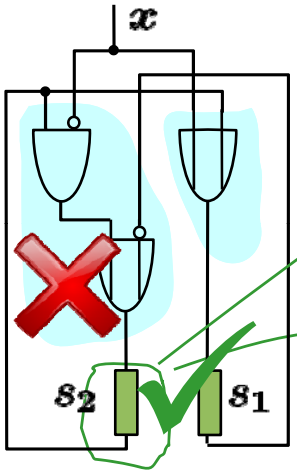
$$S_1 = \bot \wedge s_1 = \bot \wedge S_2 = \bot \wedge s_2 = \bot$$

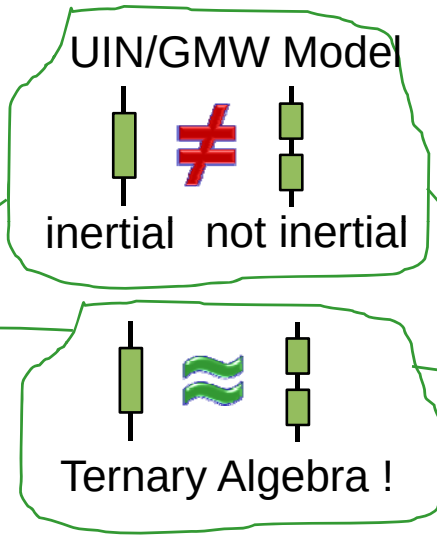
$$S_1 = s_1 \vee S_2 = s_2$$

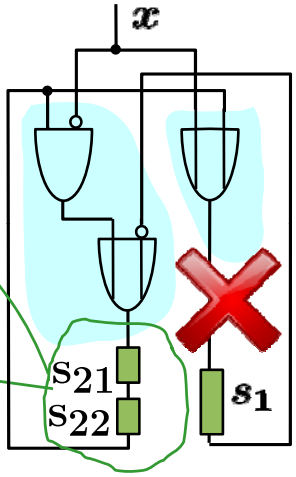
not ternary comb. ? Ternary simulation too abstract ?

UIN-comb. UIN delays too strongly synchronised ?

UIN-Delay Full Abstraction Problem





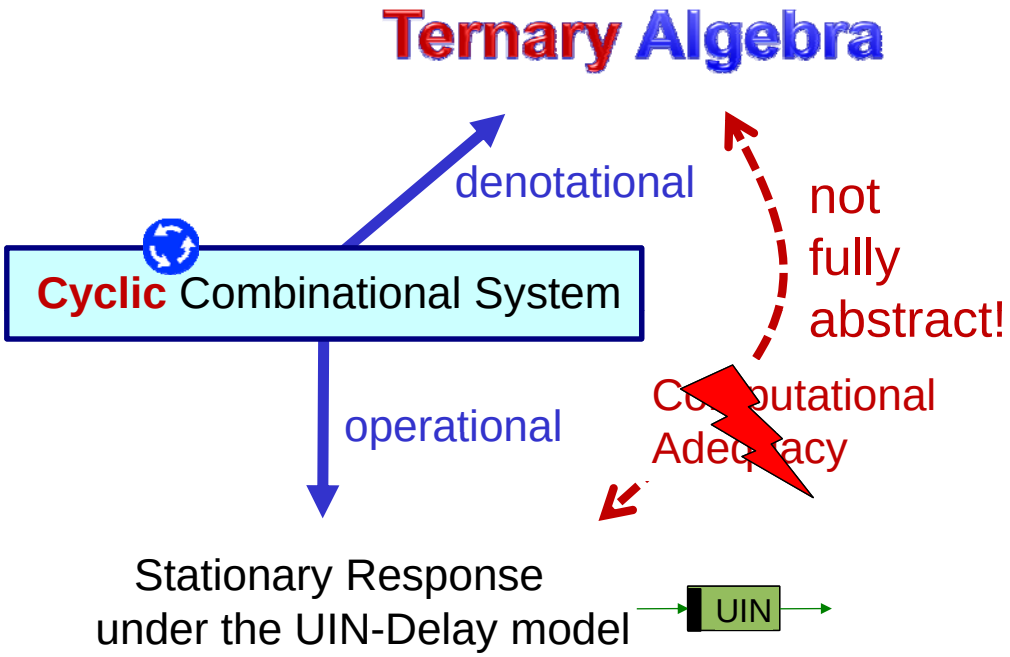


UIN-comb.
not ternary comb.

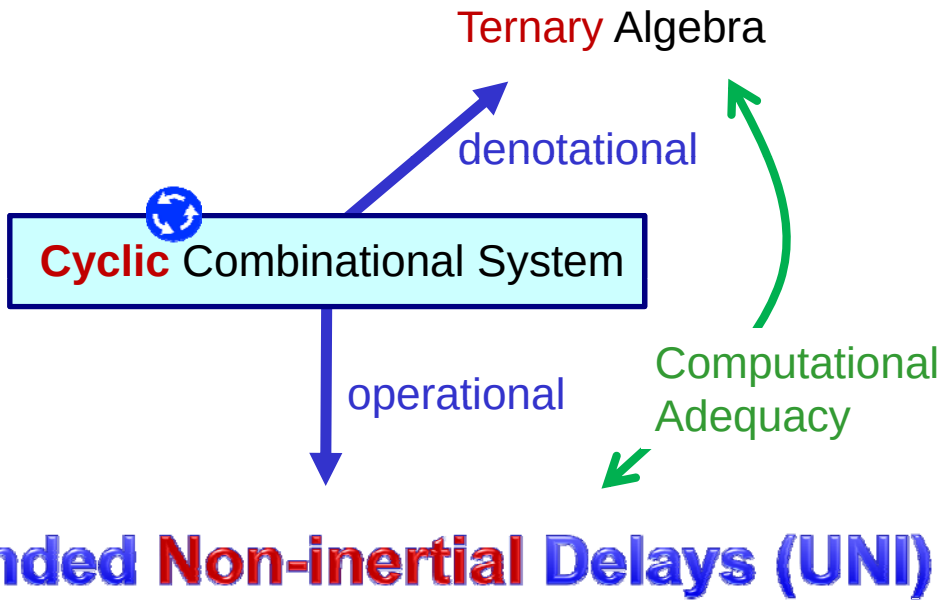
not UIN-comb.
not ternary comb.

- Inertial Delays / GMW are not quite the right operational interpretation of Ternary Simulation !

“Paradise lost ?”



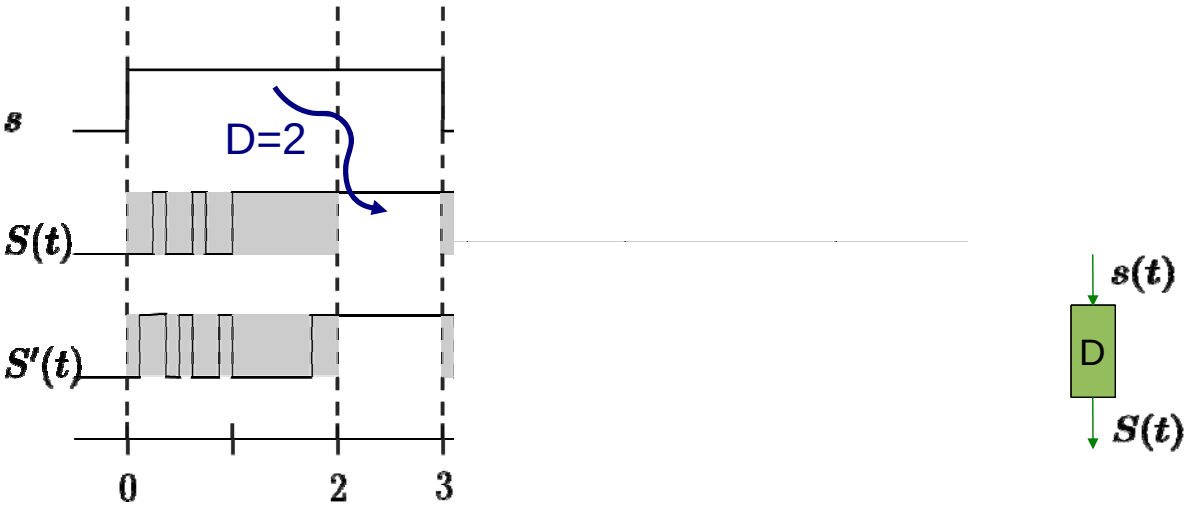
“Paradise lost ?”



The „right“ operational interpretation of Timed Ternary Simulation:

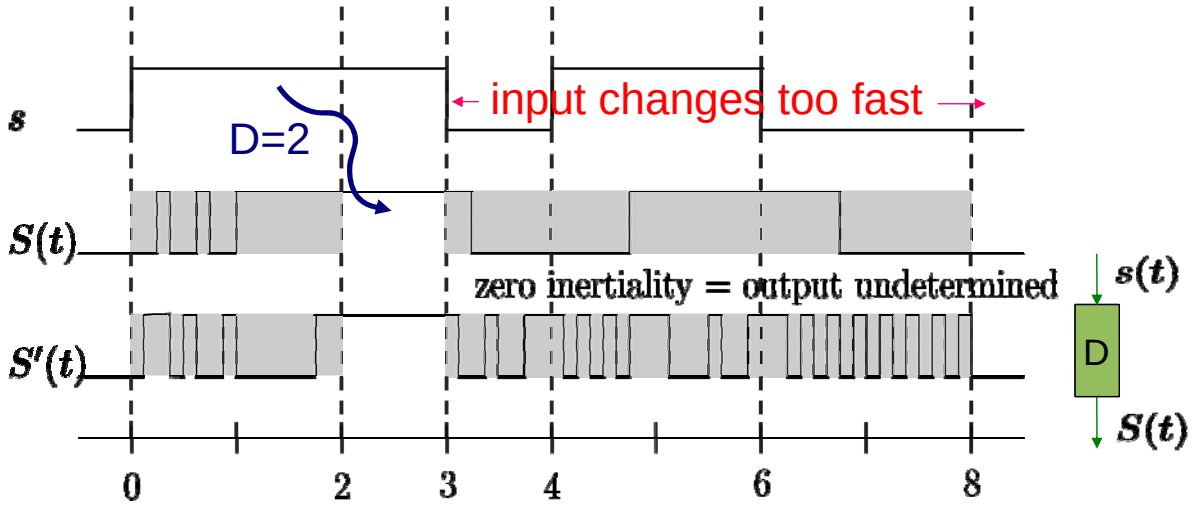
NON-INERTIAL DELAYS

Up-bounded Non-Inertial (UNI-)Delay



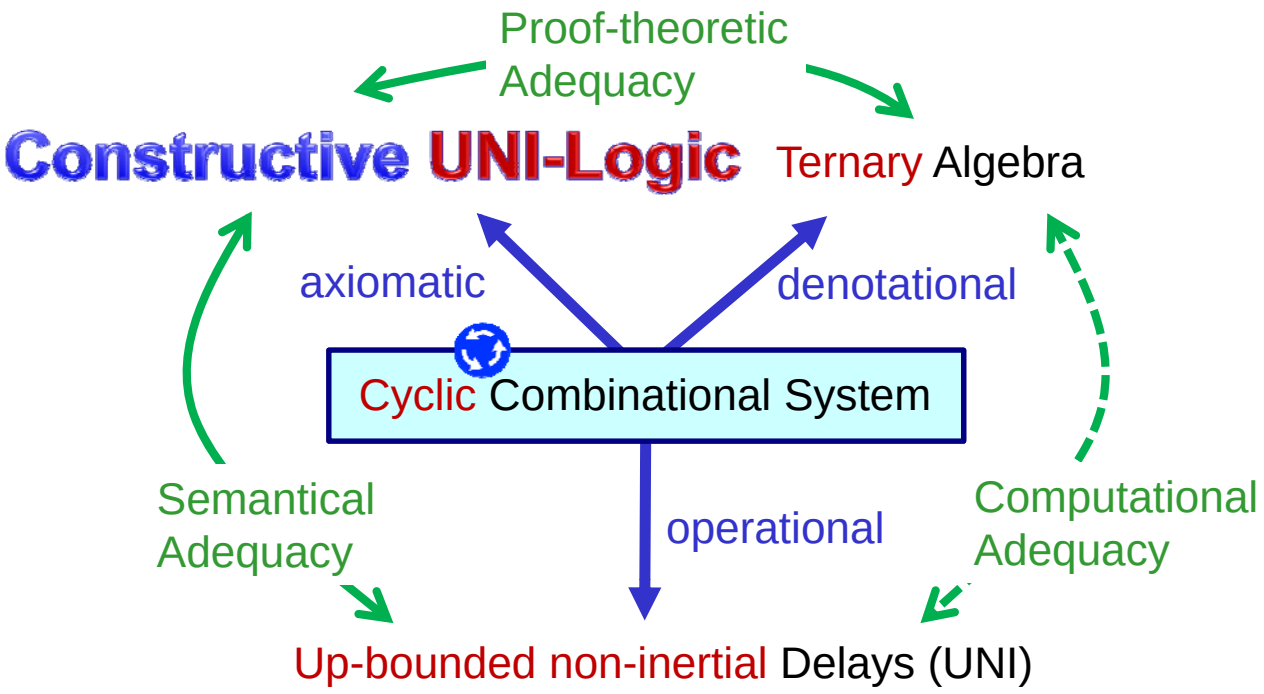
(1) Up-bounded Propagation: If the input remains stable for longer than D time, then the output stabilises to new value.
(2) Non-inertial: If input changes, output totally uncontrolled until new value has propagated through.

Up-bounded Non-Inertial (UNI-)Delay



(1) Up-bounded Propagation: If the input remains stable for longer than D time, then the output stabilises to new value.
(2) Non-inertial: If input changes, output totally uncontrolled until new value has propagated through.

How Do We Prove UNI-Delay Adequacy ?



The „right“ logical interpretation of Timed Ternary Simulation:

CONSTRUCTIVE UNI-LOGIC

Basic Properties

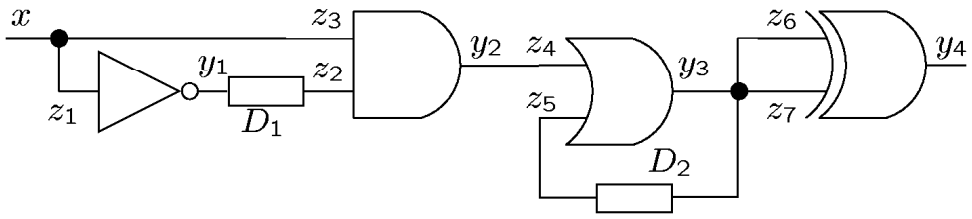
UNI-logic satisfies the axioms

$$\phi \supset \Diamond_D \phi$$
$$\Diamond_D \Diamond_E \phi \supset \Diamond_{D+E} \phi$$
$$\Diamond_D \phi \wedge \Diamond_E \psi \supset \Diamond_{max(D,E)} (\phi \wedge \psi)$$

and the rule $\models \phi \supset \psi \Rightarrow \models \Diamond_D \phi \supset \Diamond_D \psi$.

Logic: lax modality (pronounced "LAGS")
[*Fairtlough & Mendler 97*]
Types, Functional Progr.: strong monads [*Moggi 91*]

UN Network Specifications



N

$$\Phi_N \equiv y_1 = \overline{z_1} \wedge y_2 = z_3 z_2 \wedge y_3 = z_4 + z_5 \wedge$$
$$y_4 = z_6\overline{z_7} + \overline{z_6}z_7 \wedge$$
$$z_1 = x \wedge z_2 :=_{D_1} y_1 \wedge z_3 = x \wedge z_4 = y_2 \wedge$$
$$z_5 :=_{D_2} y_3 \wedge z_6 = y_3 \wedge z_7 = y_3.$$

$\phi_1 :=_D \phi_2$ stands for $(\neg\phi_2 \supset \Diamond_D \neg\phi_1) \wedge (\phi_2 \supset \Diamond_D \phi_1)$

Lindenbaum/Henkin Construction

For every input $\vec{a}$ and state vertex s_i:

UNI-Logic Proof Theory

$\Psi_{N,\vec{a}} \vdash \Diamond_E \langle s_i = \alpha \rangle$

Completeness Soundness

UNI-Logic Model Theory

$\Psi_{N,\vec{a}} \models \Diamond_E \langle s_i = \alpha \rangle$

Proof-theoretic Adequacy

Semantical Adequacy

Exactness Correctness

Ternary Simulation

$[\alpha, E] \sqsubseteq_\varepsilon [\![N]\!]_i(\vec{a}).$

UNI-Delay Execution

$\forall h \in \text{UNI-exec}(N, \vec{a}).\ s_i[E, \infty) = \alpha$

$x = \bot$ „known undefined“ $\leftrightarrow$ „unknown defined“

WHY CONSTRUCTIVENESS MATTERS…

What does constructiveness buy us ?

$\Psi_{N,\vec{a}}$ set of UNI-trajectories of network N and input $\vec{a}$

Disjunction Property

$\Psi_{N,\vec{a}} \vdash$ s stabilises to 0 $\vee$ s stabilises to 1

$\Rightarrow \Psi_{N,\vec{a}} \vdash$ s stabilises to 0 or $\Psi_{N,\vec{a}} \vdash$ s stabilises to 1

Existential Property

$\Psi_{N,\vec{a}} \vdash \exists$ t. s stabilises at time t

$\Rightarrow$ for some delay bound D,
$\Psi_{N,\vec{a}} \vdash$ s stabilises at time D

Constructive reaction is always deterministic and bounded !

CONCLUSION

Summary

Theorem — **„Constructive Networks“**

The following statements are equivalent:

- A network N is provably stable in **UNI-Logic** — axiomatic
- The **ternary simulation** of N in the chosen state variables generates Boolean solutions — denotational
- N stabilises in bounded time to a unique steady state under **non-inertial delay** assumptions — operational

Research Directions

- Expressiveness and Complexity of UNI-Logic ?
- Efficient implementation of timed ternary simulation, symbolically for arbitrary input
- What would be a sound and complete Logic for
  - Up-bounded inertial delays ?
  - bi-bounded delays, transport delays, …
- Comprehensive classification of combinational circuit hierarchies

Thank you !