# The Synchrony Hypothesis
# or
# The Importance of Being Constructive

inspired by Tom Shiple, Gérard Berry



## What's this talk about ?

- to characterise in a mathematically precise way the class of systems known, informally, as „constructive“ systems
- to present correspondence theorems linking denotational, operational and axiomatic semantics
- to highlight the fact that there are different notions of „causal“ systems depending on the MoCC (model of coordination and communication)

# Synchronous Abstraction

... why constructiveness matters

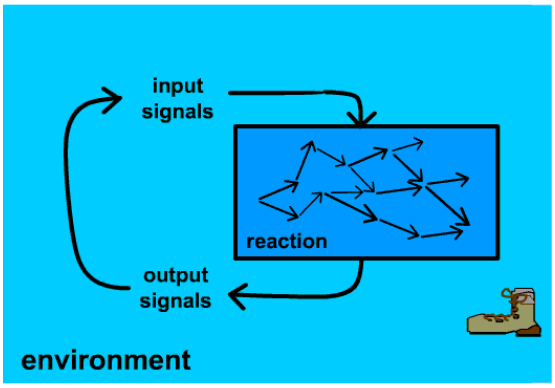


# Synchrony Hypothesis

Environment view:
Reactions are
• atomic
• deterministic
• bounded

↕

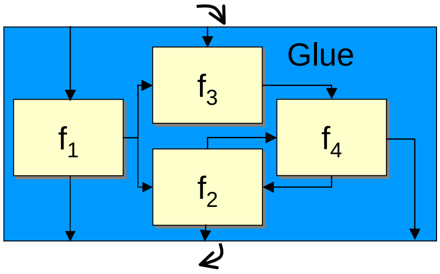
System view:
Reactions may be
• non-atomic
• non-deterministic
• unbounded

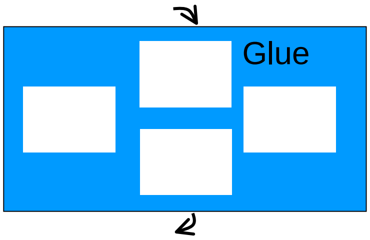



LAGS

## Soups of Soups

Ptolemy
ForSyDe
BIP
SystemC
42
…

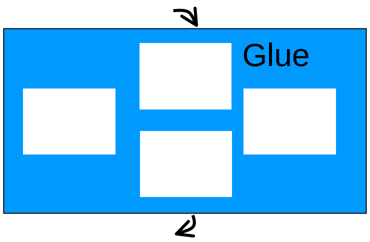


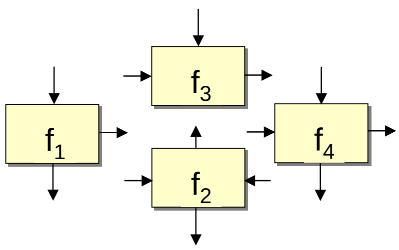




| | |
|---|---|
| Soup | Activation Condition |
| Director | Co-ordination |
| Executive | Orchestration |
| Scheduler | Control Contracts |
| **Delay Model** | Component Protocols |
| … | … |



## Soups of Soups

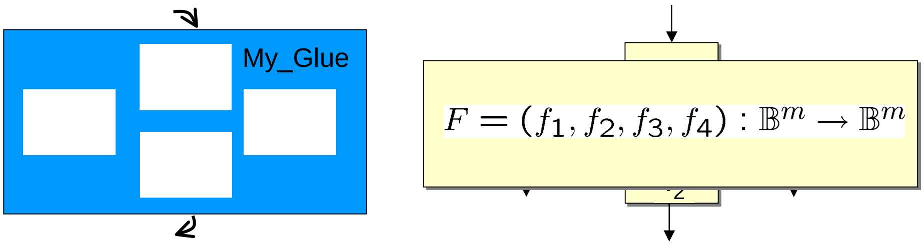


## Operational Semantics for MyGlue

$\text{My\_Glue-exec}(F) : (\text{Time} \to \mathbb{B})^m \to \mathbb{B}$ subset of execution traces

**Definition** F is My_Glue-combinational
if $\forall$ i $\in$ m. $\exists$ time bound $D_i$ and response value $\alpha_i$ such that for all $h \in \text{My\_Glue-exec}(F)$ $h_i$ stabilises to $\alpha_i$ at time $D_i$.




$$F = (f_1, f_2, f_3, f_4) : \mathbb{B}^m \to \mathbb{B}^m$$

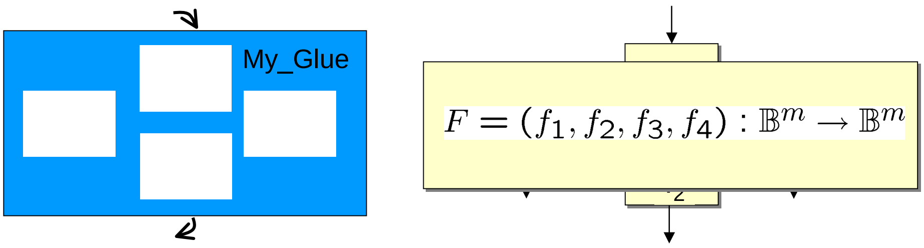


## Denotational Semantics for MyGlue

$\mathbb{B} \subset \mathbb{D}(\mathbb{B})$ $\quad \text{lift}(F) : \mathbb{D}(\mathbb{B})^m \to \mathbb{D}(\mathbb{B})^m$ monadic domain extension

$\text{My\_Glue-fp}(F) = \mu(\text{lift}(F)) : \mathbb{D}(\mathbb{B})^m$ fixed point

**Definition** F is My_Glue-causal if $\text{My\_Glue-fp}(F) \in \mathbb{B}^m$




$$F = (f_1, f_2, f_3, f_4) : \mathbb{B}^m \to \mathbb{B}^m$$

## The Full Abstraction Game

Operational Semantics (F combinational)

My_Glue-exec$(F)$ : (Time $\rightarrow \mathbb{B})^m \rightarrow \mathbb{B}$

soundness and completeness

model theory

? My Synchrony Hypothesis

Axiomatic Semantics

My_Glue-spec$(F)$

proof theory

algebraic decision procedure

My_Glue-fp$(F) = \mu($lift$(F))$ : $\mathbb{D}(\mathbb{B})^m$

Denotational Semantics (F causal)



## Constructive Semantics

Constructive Delay Model

Berry_Glue-exec$(F)$ : $(\mathbb{R}^+ \rightarrow \mathbb{B})^m \rightarrow \mathbb{B}$

soundness and completeness

model theory

? Berry's Synchrony Hypothesis

Constructive Logic

Berry_Glue-spec$(F)$

proof theory

algebraic decision procedure

Berry_Glue-fp$(F) = \mu($lift$(F))$ : $(\mathbb{B} \times \mathbb{R}^+)^m_\perp$

Ternary Simulation

## What's Constructive Logic ?

Classical Logic ☻

Excluded Middle: ☻⊢ $\phi \vee \neg \phi$ for all $\phi$

e.g., ☻⊢ $(P = NP) \vee (P \neq NP)$

Double Negation: ☻⊢ $\neg\neg \phi \equiv \phi$

Constructive Logic ☺:

Disjunction Property: If ☺⊢ $\phi \vee \psi$ then ☺⊢ $\phi$ or ☺⊢ $\psi$

Existential Property: If ☺⊢ $\exists$ x. $\phi(x)$ then there is a (closed) term t such that ☺⊢ $\phi(t)$

Constructive proofs have computational meaning



## What does it buy us ?

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

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

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

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

Constructive reaction is always deterministic and bounded !

## Some Folks‘ lore



## Some Folks‘ lore says ...

- ... denotational semantics: Ternary Algebra is a logic of constructive truth
- ... axiomatic semantics: Intuitionistic Boolean Logic has constructive provability
- ... operational semantics: Inertial Delays are the right real-time interpretation of Ternary Simulation

Really?...

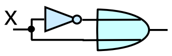
## 1 Denotational Folklore

- Ternary Algebra is $\underbrace{\cancel{\text{like a logic of truth values}}}_{\text{not quite}}$

$\mathbb{S} = \{0, 1, \perp\} = \mathbb{B}_\perp \quad \perp \sqsubseteq 0, 1$ „discrete Scott domain“

$\perp$ unknown, undefined, non-determinism, oscillation, deadlock, metastability, unstable, transient, don‘t care, ...

$\perp$ ... avoids dangerous classical equalities ☺:




$x + \overline{x} \neq 1$

| NOT | |
|---|---|
| 0 | 1 |
| 1 | 0 |
| $\perp$ | $\perp$ |

$\perp$ ... avoids equalities altogether ☹ !

Ternary logic has no theorems at all,

$x = \perp \wedge y = \perp \not\Rightarrow x = y$

| OR | 0 | 1 | $\perp$ |
|---|---|---|---|
| 0 | 0 | 1 | $\perp$ |
| 1 | 1 | 1 | 1 |
| $\perp$ | $\perp$ | 1 | $\perp$ |

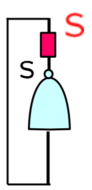


## 2 Axiomatic Folklore

- Constructiveness is not quite about provability in Intuitionistic Boolean Logic !

present s else emit s end

$s = \neg s$ intuitionistically, is equivalent to false

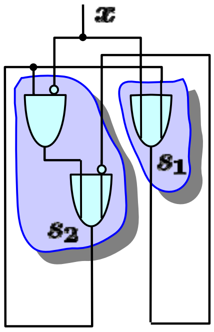



$\Rightarrow \; s = \neg s \vdash \phi$ for any formula $\phi$

$\Rightarrow$ Intuitionism alone doesn‘t help, we must axiomatise delays (scheduling), too !

## 3 Operational Folklore

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

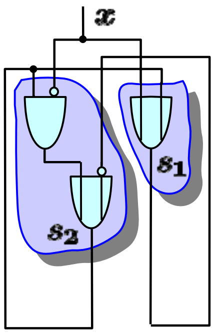



$$s_1 = s_2 + x$$
$$s_2 = \overline{s_1} + \overline{x} \cdot s_2$$



## 3 Operational Folklore

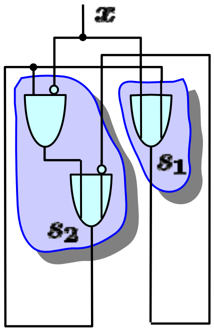
- Inertial Delays are not quite the right operational interpretation of Ternary Simulation !




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

## 3 Operational Folklore

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



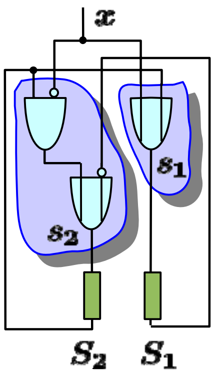

$$s_1 = \perp + x$$
$$s_2 = \perp$$

The ternary fixed point implies that $s_2$ is non-constructive



## 3 Operational Folklore

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




$x \cdot S_1 S_2$ total state

0•1*0 1•0*0*

0•00* 1•0*1

0•0*1 1•11*

0•11 1•10

stable states

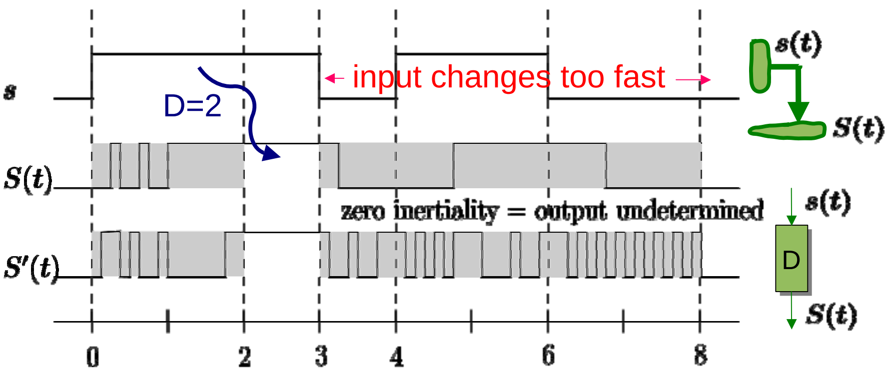
All possible inertial delay system trajectories according to General Multiple Winner model (GMW, (Brzozowski & Seger) converge !

# Operational Semantics: Non-Inertial Delays

A constructive communication model for
Boolean networks

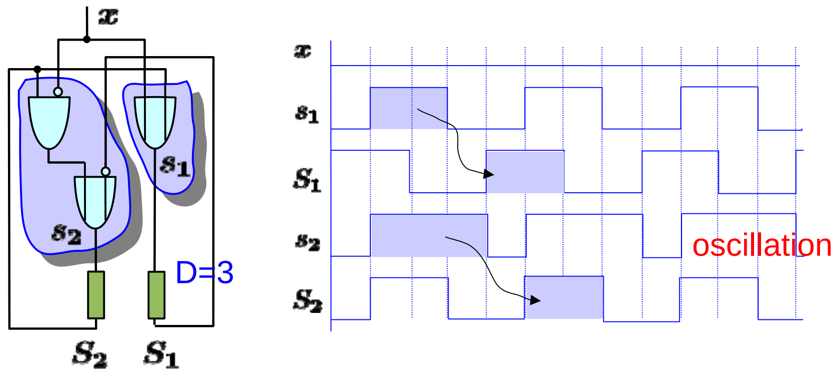


# Up-bounded Non-Inertial (UN-) Delay




(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 (UN-) Delay




Non-inertial (UN-) delays permit oscillation as predicted by ternary simulation.



# Execution of UN-Delay Networks

state nodes $\mathcal{S} \subseteq \mathcal{Z}$ feedback vertex set $\mathcal{S} \cap \mathcal{X} = \emptyset$

input nodes $\mathcal{X} \subseteq \mathcal{Z}$ constant input $a \in \mathbb{B}^{\mathcal{X}}$

Network excitation function $S : (\mathbb{B}^{\mathcal{X}} \times \mathbb{B}^{\mathcal{S}}) \rightarrow \mathbb{B}^{\mathcal{S}}$

## Non-Inertial Network Behaviour

Let UN-exec$(N, a) \subseteq (\mathbb{R}^{+} \rightarrow \mathbb{B})^{\mathcal{Z}}$ be the set of trajectories h such that

- h has all input signals constant at value a
- h is right-continuous and non-Zeno
- h is consistent with network excitation function S and UN-delay scheduling

# Axiomatic Semantics: UN-Logic

A constructive „Boolean“ specification language
for UN-delay networks



# Syntax and Semantics

Syntax $\phi \; ::= \; e \mid \phi \supset \phi \mid \phi \wedge \phi \mid \phi \vee \phi \mid \lozenge_D \phi$

$e \subseteq \mathbb{B}^{\mathcal{Z}}$ Boolean expression in network variables $\mathcal{Z}$

$D \geq 0$ delay parameter

$h[s,t) \models \phi$ states that (right continuous, non-Zeno) trajectory $h : \mathbb{R}^+ \to \mathbb{B}^{\mathcal{Z}}$ *satisfies* $\phi$ in interval $I = [s,t)$:

$$
\begin{array}{rcll}
h\,I & \models & e & \text{iff } \; \forall \tau \in I.\, h(\tau) \models e \text{ classically} \\
h\,I & \models & \phi \wedge \psi & \text{iff } \; h\,I \models \phi \text{ and } h\,I \models \psi \\
h\,I & \models & \phi \vee \psi & \text{iff } \; h\,I \models \phi \text{ or } h\,I \models \psi \\
h\,I & \models & \phi \supset \psi & \text{iff } \; \forall J \subseteq I.\, h\,J \models \phi \Rightarrow h\,J \models \psi \\
h\,[s,t) & \models & \lozenge_D \phi & \text{iff } \; s + D < t \Rightarrow h[s+D,t) \models \phi.
\end{array}
$$

## Basic Properties

$h \models \phi$ iff $\forall I.\ h\, I \models \phi$ $\qquad$ $\models \phi$ iff $\forall h.\ h \models \phi$

Abbreviations $\quad$ *false* $=_{\text{df}} 0$ $\quad$ *true* $=_{\text{df}} 1$

$\phi \equiv \psi =_{\text{df}} (\phi \supset \psi) \wedge (\psi \supset \phi)$ $\qquad$ $\neg\phi =_{\text{df}} \phi \supset$ *false*

UN-Logic contains Boolean Algebra

$$e_1 \cdot e_2 \longleftrightarrow e_1 \wedge e_2$$
$$\overline{e} \longleftrightarrow \neg e$$
$$e_1 + e_2 \longleftrightarrow \neg(\neg e_1 \wedge \neg e_2)$$
$$e_1 = e_2 \longleftrightarrow e_1 \equiv e_2$$
$$\overline{e_1} + e_2 \longleftrightarrow e_1 \supset e_2$$

logical equivalences



## Basic Properties

Monotonicity $\quad$ If $J \subseteq I$ and $h\, I \models \phi$, then $h\, J \models \phi$.

UN-Logic is constructive

$h \models \phi \vee \psi$ iff $h \models \phi$ or $h \models \psi$

e.g., $h \models \langle s = 1\rangle + \neg\langle s = 1\rangle$ always true

$h \models \langle s = 1\rangle \vee \neg\langle s = 1\rangle$ only if s constant

Now, what is $\Diamond$ good for?... it saves us from logical fallacies: e.g., $(s = 1) \supset \Diamond_D \neg(s = 1)$ means "region $s = 1$ is transient with life time at most $D$" rather than plain falsity.

## Basic Properties

UN-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")
Types, Functional Programming: strong monads.

$\Diamond \phi$ internalises side effects and computations ...



## Inertial Assignment

**Abbreviation**

Let $s :=_D e$ stand for $(\neg e \supset \Diamond_D \neg s) \wedge (e \supset \Diamond_D s)$

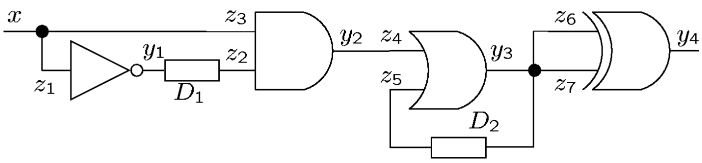
**Proposition**

$h \models s_1 :=_D s_2$ iff $h$ is the execution of an up-bounded non-inertial delay between $s_2$ (input) and $s_1$ (output).

$$e_1 :=_0 e_2 \longleftrightarrow e_1 = e_2$$
$$\Diamond_D e \longleftrightarrow e :=_D 1$$
$$e \longleftrightarrow e :=_0 1$$

logical equivalences

## UN Network Specifications




N

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



## UN Network Specifications

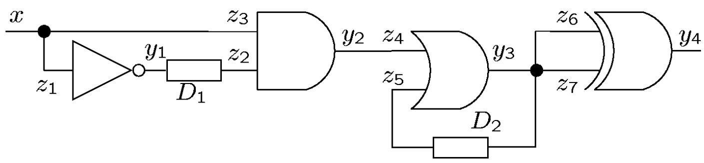
Note: Equational Substitution

$$x :=_{D_1} e_1 \wedge y :=_{D_2} e_2$$
$$\longrightarrow x :=_{D_1 + D_2} e_1\{e_2/y\}$$

in general is not sound unless $D_2 = 0$ !

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

## UN Network Specifications




N

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

$$\Phi_N^{\mathsf{rsnf}} \;\equiv\; z_2 :=_{D_1} \overline{x} \wedge z_5 :=_{D_2} x z_2 + z_5$$

reduced substitution normal form



## Semantical Adequacy

state nodes $\mathcal{S} \subseteq \mathcal{Z}$ feedback vertex set $\mathcal{S} \cap \mathcal{X} = \emptyset$

input nodes $\mathcal{X} \subseteq \mathcal{Z}$ constant input $a \in \mathbb{B}^{\mathcal{X}}$

Network excitation function $S : (\mathbb{B}^{\mathcal{X}} \times \mathbb{B}^{\mathcal{S}}) \to \mathbb{B}^{\mathcal{S}}$

### Semantical Adequacy Theorem

The non-inertial network behaviour UN-exec$(N, a)$ coincides with the models of the formula

$$\Psi_{N,a} \;\equiv_{df}\; \bigwedge_{s_i \in \mathcal{S}} s_i :=_{D_i} S_i \;\wedge\; \bigwedge_{x_i \in \mathcal{X}} x_i = a_i$$

## UN-Calculus

Let $\Phi, \Theta$ be sets of network formulas.
We derive sequents $\Phi \vdash \Theta$ according to the following rules.

$$\frac{D \leq E \qquad R \subseteq S}{\Phi, \Diamond_D R \;\vdash\; \Diamond_E S, \Theta}\ \Diamond id \qquad\qquad \frac{}{\Phi \vdash \Diamond_E 1, \Theta}\ \Diamond true$$

$$\frac{\Phi \vdash \Diamond_D S, \Theta \quad \Phi \vdash \Diamond_E T, \Theta \quad S \cap T \subseteq R \quad F \geq \mathit{max}(D, E)}{\Phi \vdash \Diamond_F R, \Theta}\ \Diamond \wedge R$$

$$\frac{\Phi, R \supset \Diamond_D S \vdash \Diamond_E R, \Theta \qquad S \subseteq T \qquad F \geq D + E}{\Phi, R \supset \Diamond_D S \vdash \Diamond_F T, \Theta}\ \supset\!\Diamond L$$



## Theorems

**Soundness** $\quad \Phi \vdash \Theta \;\Rightarrow\; \Phi \models \Theta.$

**Completeness*** $\quad \Phi \models \Theta \Rightarrow \Phi \vdash \Theta.$

**Constructiveness**

If $\Phi \vdash \Theta$, then there exists a single $\theta \in \Theta$ such that $\Phi \vdash \theta$.

What do these theorems buy us ? ...

*under some weak assumptions

## Reactivity of UN-Networks

$\forall h \in$ UN-exec$(N, a).\, h_i$ stabilises

$\forall h \in$ UN-exec$(N, a).\, \exists D.\, \exists \alpha.\, h_i[D, \infty) = \alpha$

$$\Downarrow$$

bounded reaction

no non-determinism

no metastability

$\exists D.\, \exists \alpha.\, \forall h \in$ UN-exec$(N, a).\, h_i[D, \infty) = \alpha$

$\exists D.\, \exists \alpha.\, \forall h \in$ UN-exec$(N, a).\, h_i$ stabilises to $\alpha$ at time $D$



## Reactivity of UN-Networks

$\forall h \in$ UN-exec$(N, a).\, h_i$ stabilises

$\forall h \in$ UN-exec$(N, a).\, \exists D.\, \exists \alpha.\, h_i[D, \infty) = \alpha$

Quantifier swap !

$\exists D.\, \exists \alpha.\, \forall h \in$ UN-exec$(N, a).\, h_i[D, \infty) = \alpha$

$\exists D.\, \exists \alpha.\, \forall h \in$ UN-exec$(N, a).\, h_i$ stabilises to $\alpha$ at time $D$

## Reactivity of UN-Networks

$\forall h \in \text{UN-exec}(N,a).\, h_i$ stabilises

$\forall h \in \text{UN-exec}(N,a).\, \exists D.\, \exists \alpha.\, h_i[D,\infty) = \alpha$

$\forall h.\, h \models \Psi_{N,a} \Rightarrow \exists D.\, \exists \alpha.\, h \models \Diamond_D(s_i = \alpha)$ adequacy

$\Psi_{N,a} \models \{\Diamond_D(s_i = \alpha) \mid D \in \mathbb{R}^+, \alpha \in \mathbb{B}\}$

$\Psi_{N,a} \vdash \{\Diamond_D(s_i = \alpha) \mid D \in \mathbb{R}^+, \alpha \in \mathbb{B}\}$ completeness

$\exists D.\, \exists \alpha.\, \Psi_{N,a} \vdash \Diamond_D(s_i = \alpha)$ constructiveness

$\exists D.\, \exists \alpha.\, \Psi_{N,a} \models \Diamond_D(s_i = \alpha)$ soundness

$\exists D.\, \exists \alpha.\, \forall h \in \text{UN-exec}(N,a).\, h_i[D,\infty) = \alpha$

$\exists D.\, \exists \alpha.\, \forall h \in \text{UN-exec}(N,a).\, h_i$ stabilises to $\alpha$ at time $D$



## Reactivity of UN-Networks

### Corollary

For every UN-network $N$ there exists a steady state response function

$$[\![N]\!] : \mathbb{B}^{\mathcal{X}} \to (\mathbb{B} \times \mathbb{R}^+)^{\mathcal{S}}_{\perp}$$

such that for all $\alpha, D$ such that:

- $[\![N]\!]_i(a) \neq \perp$ is the minimal stabilisation time and stable value of $s_i$ under $a$
- If $[\![N]\!]_i(a) = \perp$, then $s_i$ oscillates in at least one history.

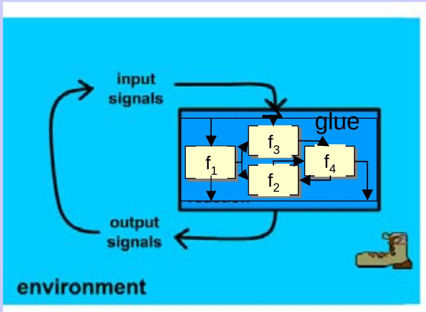
How do we compute $[\![N]\!]$?

## Denotational Semantics: Timed Ternary Simulation

A sound and complete algebraic decision procedure for UN-Logic



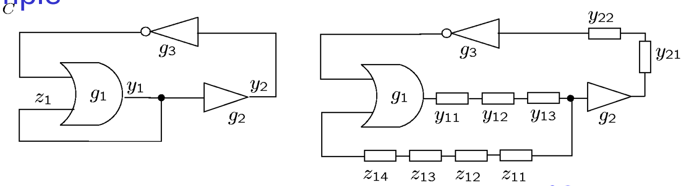
## Classes of Synchronous Causality



## Soup of Delays

Every multi-set of vertices $V \subseteq \mathcal{Z} = \{y_1, y_2, z_1\}$ determines a network $C(V)$ as follows: For every $v \in V$ with multiplicity $n$ we break vertex $v$ by exactly $n$ delay elements connected up in series.

### Example




$C(\{3y_1, 2y_2, 4z_1\})$

$C(\{l{\cdot}y_1, m{\cdot}y_2, k{\cdot}z_1\})$ is

- $k = m = 0$: both UN-combinational and UI-combinational
- $k = 0$, $l = 1$, $m \geq 1$: UI-combinational but not UN-combinational
- neither UN-combinational nor UI-combinational in all other cases.



## Soup of Delays

### Definition

Let $C$ be a circuit and DEL a delay-model. $C$ is

- DEL-speed-independent (*DEL-SI*) if all gate-delay networks obtained from $C$ are DEL-combinational;
- DEL-delay-insensitive (*DEL-DI*) if all gate, input and wire-delay networks of $C$ are DEL-combinational,

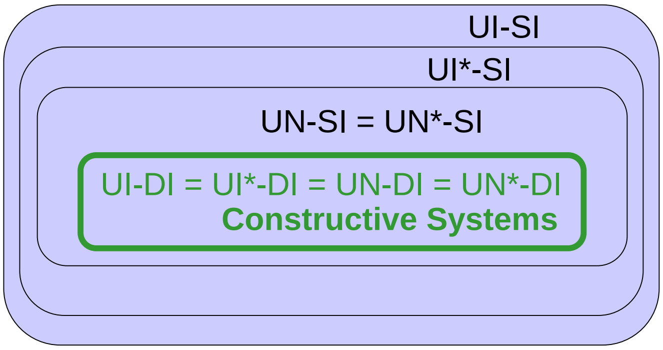
where in each case the multiplicity of delays is 1.

We can strengthen the notions to say that $C$ is

- *DEL*-SI* if $C$ is DEL-SI, and
- *DEL*-DI* if $C$ is *DEL-DI*

for arbitrary multiplicity of delays.

# Some Synchronous Causality Classes






# Summary

**Theorem**

The following statements are equivalent:

- A system (decomposition) S is constructive, i.e., provably stable in **UN-Logic**
- The **ternary simulation** of S in the chosen state variables generates Boolean solutions
- S stabilises in bounded time to a unique steady state under **non-inertial delay** assumptions