



# 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

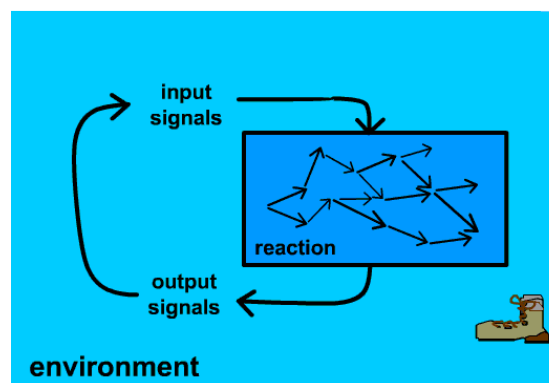
- atomic
- deterministic
- bounded



System view:

Reactions may be

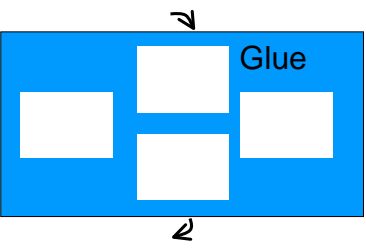
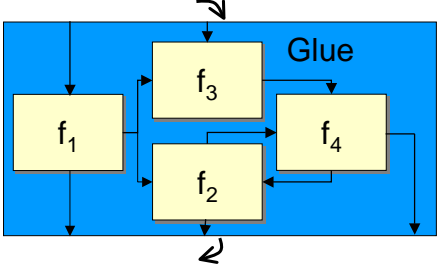
- non-atomic
- non-deterministic
- unbounded



LAGS

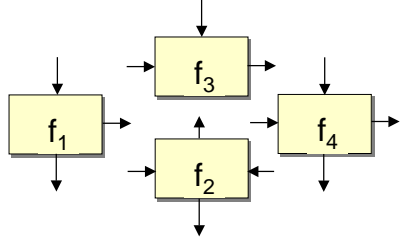
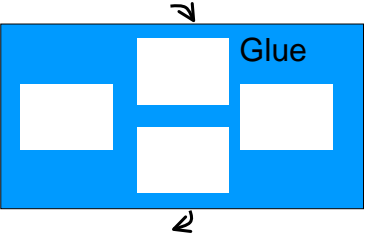
# Soups of Soups

Ptolemy  
ForSyDe  
BIP  
SystemC  
42  
...



Soup	Activation Condition
Director	Co-ordination
Executive	Orchestration
Scheduler	Control Contracts
<b>Delay Model</b>	Component Protocols
...	...

# Soups of Soups





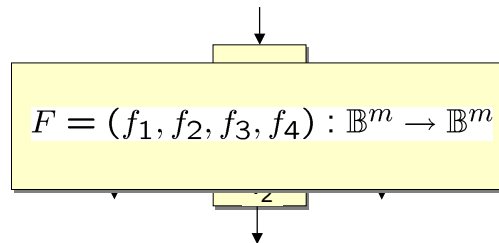
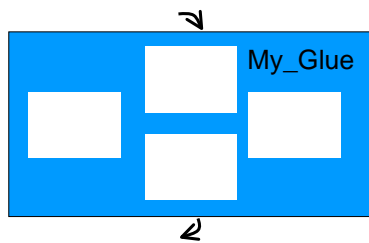
## Operational Semantics for MyGlue



$\text{My\_Glue-exec}(F) : (\text{Time} \rightarrow \mathbb{B})^m \rightarrow \mathbb{B}$  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  $D_i$ .



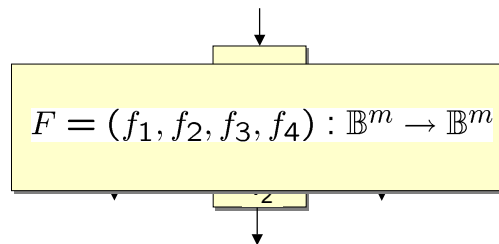
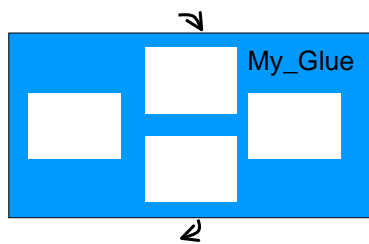
## Denotational Semantics for MyGlue



$\mathbb{B} \subset \mathbb{D}(\mathbb{B})$   $\text{lift}(F) : \mathbb{D}(\mathbb{B})^m \rightarrow \mathbb{D}(\mathbb{B})^m$  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  $\text{My\_Glue-fp}(F) \in \mathbb{B}^m$



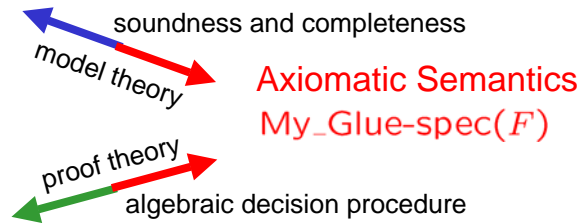
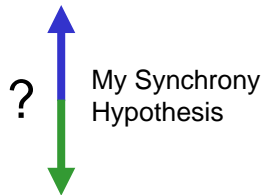


## The Full Abstraction Game



Operational Semantics (F combinational)

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



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

Denotational Semantics (F causal)

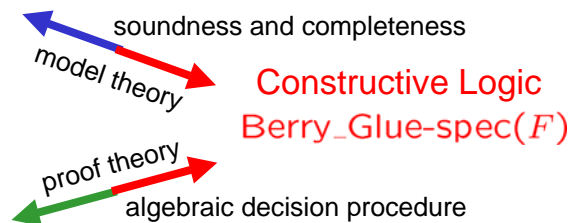
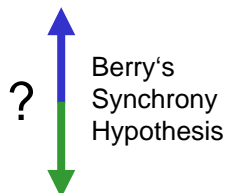


## Constructive Semantics



Constructive Delay Model

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



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

Ternary Simulation



## What's Constructive Logic ?



### Classical Logic ☹

Excluded Middle:  $\odot \vdash \phi \vee \neg \phi$  for all  $\phi$   
e.g.,  $\odot \vdash (P = NP) \vee (P \neq NP)$

Double Negation:  $\odot \vdash \neg\neg \phi \equiv \phi$

### Constructive Logic ☺:

Disjunction Property: If  $\odot \vdash \phi \vee \psi$  then  $\odot \vdash \phi$  or  $\odot \vdash \psi$

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

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

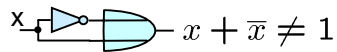
not quite

- Ternary Algebra is like a logic of truth values

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

$\perp$  unknown, undefined, non-determinism, oscillation, deadlock, metastability, unstable, transient, don't care, ...

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



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

Ternary logic has no theorems at all,

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

NOT		
0	1	1
1	0	0
$\perp$	$\perp$	$\perp$

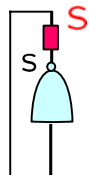
OR			
0	1	$\perp$	$\perp$
0	1	1	1
1	1	1	1
$\perp$	$\perp$	1	$\perp$

# 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  $\phi$

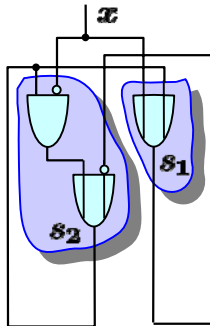
$\Rightarrow$  Intuitionism alone doesn't help, we must axiomatise delays (scheduling), too !



### 3 Operational Folklore



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



$$s_1 = s_2 + x$$

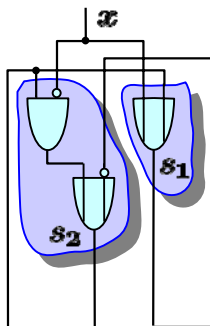
$$s_2 = \overline{s_1} + \overline{x} \cdot s_2$$



### 3 Operational Folklore



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

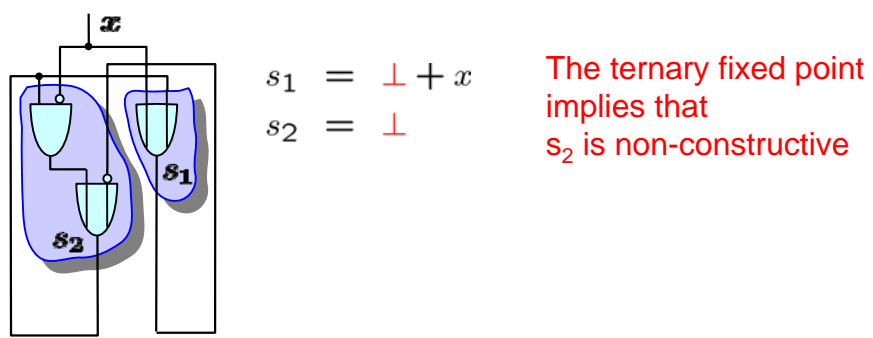


$$s_1 = \perp + x$$

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

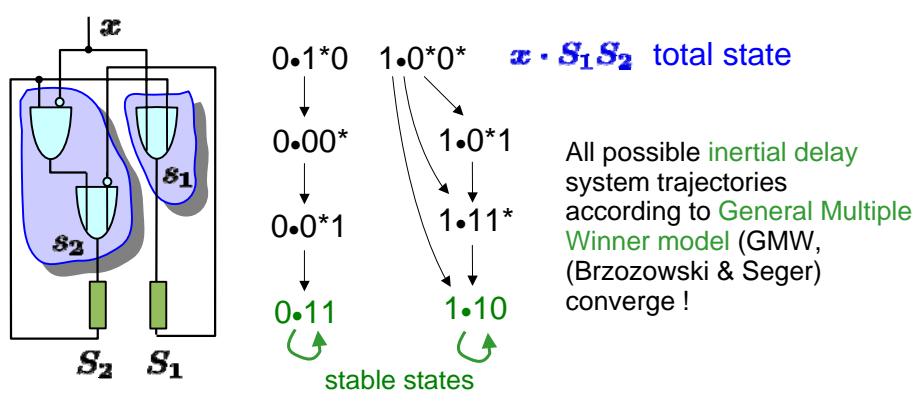
3 Operational Folklore

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



3 Operational Folklore

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



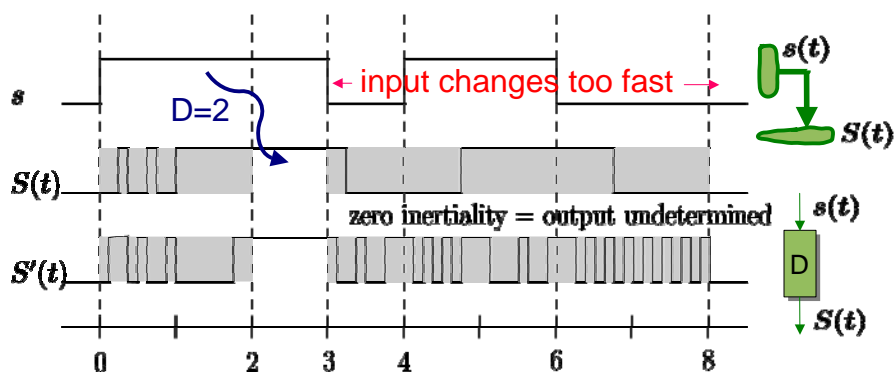


## Operational Semantics: Non-Inertial Delays

A constructive communication model for  
Boolean networks



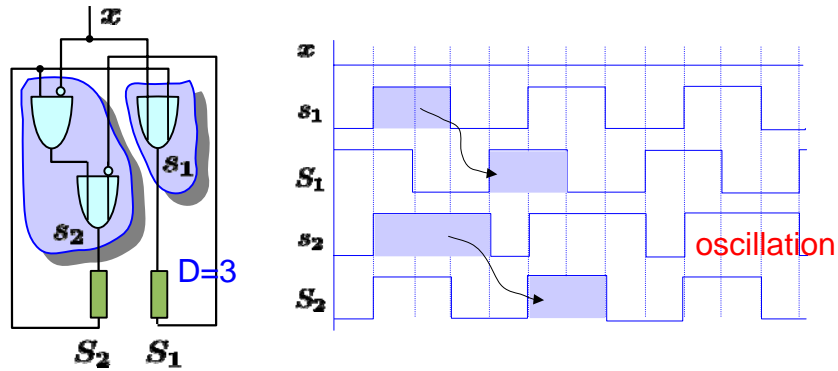
## Up-bounded Non-Inertial (UN-) Delay



- (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  $S \subseteq Z$       feedback vertex set  $S \cap X = \emptyset$   
input nodes  $X \subseteq Z$       constant input  $a \in \mathbb{B}^X$   
Network excitation function  $S : (\mathbb{B}^X \times \mathbb{B}^S) \rightarrow \mathbb{B}^S$

### Non-Inertial Network Behaviour

Let  $\text{UN-exec}(N, a) \subseteq (\mathbb{R}^+ \rightarrow \mathbb{B})^Z$  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**  $\phi ::= e \mid \phi \supset \phi \mid \phi \wedge \phi \mid \phi \vee \phi \mid \diamond_D \phi$   
 $e \subseteq \mathbb{B}^Z$  Boolean expression in network variables  $Z$   
 $D \geq 0$  delay parameter

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

$hI \models e$  iff  $\forall \tau \in I. h(\tau) \models e$  classically  
 $hI \models \phi \wedge \psi$  iff  $hI \models \phi$  and  $hI \models \psi$   
 $hI \models \phi \vee \psi$  iff  $hI \models \phi$  or  $hI \models \psi$   
 $hI \models \phi \supset \psi$  iff  $\forall J \subseteq I. hJ \models \phi \Rightarrow hJ \models \psi$   
 $h[s, t] \models \diamond_D \phi$  iff  $s + D < t \Rightarrow h[s + D, t] \models \phi$ .



## Basic Properties



$$h \models \phi \text{ iff } \forall I. h I \models \phi \qquad \models \phi \text{ iff } \forall h. h \models \phi$$

Abbreviations	$false =_{df} 0$	$true =_{df} 1$
$\phi \equiv \psi =_{df} (\phi \supset \psi) \wedge (\psi \supset \phi)$		$\neg \phi =_{df} \phi \supset false$

### UN-Logic contains Boolean Algebra

$e_1 \cdot e_2$	$\longleftrightarrow$	$e_1 \wedge e_2$	logical equivalences
$\bar{e}$	$\longleftrightarrow$	$\neg e$	
$e_1 + e_2$	$\longleftrightarrow$	$\neg(\neg e_1 \wedge \neg e_2)$	
$e_1 = e_2$	$\longleftrightarrow$	$e_1 \equiv e_2$	
$\bar{e}_1 + e_2$	$\longleftrightarrow$	$e_1 \supset e_2$	



## Basic Properties



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

### UN-Logic is constructive

$$h \models \phi \vee \psi \text{ iff } h \models \phi \text{ or } h \models \psi$$

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).

$$e_1 :=_0 e_2 \longleftrightarrow e_1 = e_2$$

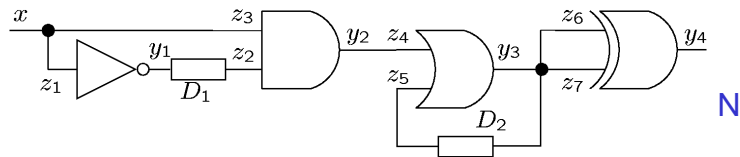
$$\diamond_D e \longleftrightarrow e :=_D 1$$

$$e \longleftrightarrow e :=_0 1$$

logical equivalences



## UN Network Specifications



$$\begin{aligned} \Phi_N \equiv & y_1 = \bar{z}_1 \wedge y_2 = z_3 z_2 \wedge y_3 = z_4 + z_5 \wedge \\ & y_4 = z_6 \bar{z}_7 + \bar{z}_6 z_7 \wedge \\ & z_1 = x \wedge z_2 :=_{D_1} y_1 \wedge z_3 = x \wedge z_4 = y_2 \wedge \\ & z_5 :=_{D_2} y_3 \wedge z_6 = y_3 \wedge z_7 = y_3. \end{aligned}$$



## UN Network Specifications



Note: Equational Substitution

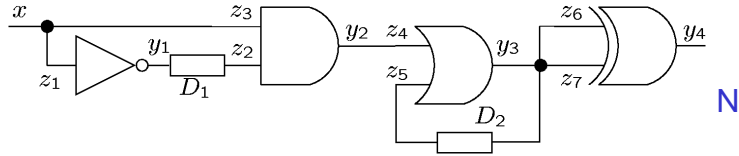
$$\begin{aligned} \mathbf{x} :=_{D_1} \mathbf{e}_1 \wedge \mathbf{y} :=_{D_2} \mathbf{e}_2 \\ \longrightarrow \mathbf{x} :=_{D_1 + D_2} \mathbf{e}_1 \{ \mathbf{e}_2 / \mathbf{y} \} \end{aligned}$$

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

$$\begin{aligned} \Phi_N \equiv & y_1 = \bar{z}_1 \wedge y_2 = z_3 z_2 \wedge y_3 = z_4 + z_5 \wedge \\ & y_4 = z_6 \bar{z}_7 + \bar{z}_6 z_7 \wedge \\ & z_1 = x \wedge z_2 :=_{D_1} y_1 \wedge z_3 = x \wedge z_4 = y_2 \wedge \\ & z_5 :=_{D_2} y_3 \wedge z_6 = y_3 \wedge z_7 = y_3. \end{aligned}$$



## UN Network Specifications



$$\begin{aligned} \Phi_N \equiv & y_1 = \bar{z}_1 \wedge y_2 = z_3 z_2 \wedge y_3 = z_4 + z_5 \wedge \\ & y_4 = z_6 \bar{z}_7 + \bar{z}_6 z_7 \wedge \\ & z_1 = x \wedge z_2 :=_{D_1} y_1 \wedge z_3 = x \wedge z_4 = y_2 \wedge \\ & z_5 :=_{D_2} y_3 \wedge z_6 = y_3 \wedge z_7 = y_3. \end{aligned}$$

$$\Phi_N^{\text{rsnf}} \equiv z_2 :=_{D_1} \bar{x} \wedge z_5 :=_{D_2} x z_2 + z_5 \quad \text{reduced substitution normal form}$$



## 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  $S : (\mathbb{B}^{\mathcal{X}} \times \mathbb{B}^{\mathcal{S}}) \rightarrow \mathbb{B}^{\mathcal{S}}$

### Semantical Adequacy Theorem

The non-inertial network behaviour  $\text{UN-exec}(N, a)$  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.

$$\frac{D \leq E \quad R \subseteq S}{\Phi, \diamond_D R \vdash \diamond_E S, \Theta} \diamond id \qquad \frac{}{\Phi \vdash \diamond_E 1, \Theta} \diamond true$$

$$\frac{\Phi \vdash \diamond_D S, \Theta \quad \Phi \vdash \diamond_E T, \Theta \quad S \cap T \subseteq R \quad F \geq \max(D, E)}{\Phi \vdash \diamond_F R, \Theta} \diamond \wedge R$$

$$\frac{\Phi, R \supset \diamond_D S \vdash \diamond_E R, \Theta \quad S \subseteq T \quad F \geq D + E}{\Phi, R \supset \diamond_D S \vdash \diamond_F T, \Theta} \supset \diamond L$$



## Theorems



Soundness

$$\Phi \vdash \Theta \Rightarrow \Phi \models \Theta.$$

Completeness\*

$$\Phi \models \Theta \Rightarrow \Phi \vdash \Theta.$$

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



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

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



bounded reaction  
no non-determinism  
no metastability

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

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



## Reactivity of UN-Networks



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

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

Quantifier swap !

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

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

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

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

$\exists D. \exists \alpha. h_i[D, \infty) = \alpha$

}

adequacy

completeness

constructiveness

soundness

$\forall h. h \models \Psi_{N,a} \Rightarrow \exists D. \exists \alpha. h \models \Diamond_D(s_i = \alpha)$   
 $\Psi_{N,a} \models \{\Diamond_D(s_i = \alpha) \mid D \in \mathbb{R}^+, \alpha \in \mathbb{B}\}$   
 $\Psi_{N,a} \vdash \{\Diamond_D(s_i = \alpha) \mid D \in \mathbb{R}^+, \alpha \in \mathbb{B}\}$   
 $\exists D. \exists \alpha. \Psi_{N,a} \vdash \Diamond_D(s_i = \alpha)$   
 $\exists D. \exists \alpha. \Psi_{N,a} \models \Diamond_D(s_i = \alpha)$

$\exists D. \exists \alpha.$

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

}

adequacy

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

**Corollary**

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

$$\llbracket N \rrbracket : \mathbb{B}^X \rightarrow (\mathbb{B} \times \mathbb{R}^+)_{\perp}^S$$

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

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

How do we compute  $\llbracket N \rrbracket$ ?

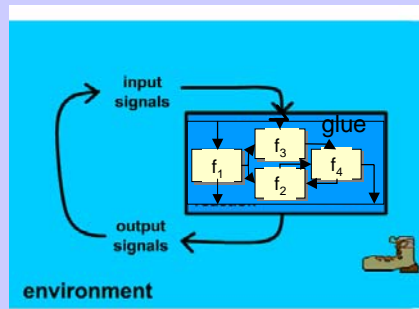


## Denotational Semantics: Timed Ternary Simulation

A sound and complete algebraic decision  
procedure for UN-Logic



## Classes of Synchronous Causality



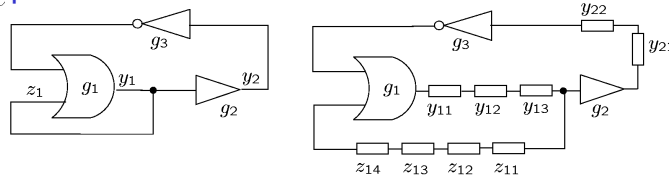


## Soup of Delays



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

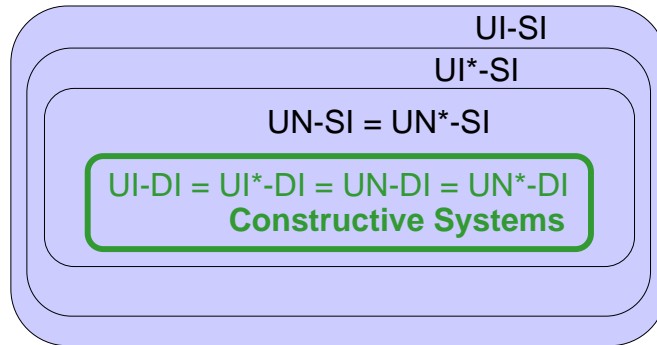
Let  $C$  be a circuit and DEL a delay-model.  $C$  is

- **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.



## 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