The Synchrony Hypothesis, Constructive Circuits and Timed Ternary Simulation

Michael Mendler

Faculty of Information Systems and Applied Computer Sciences
University of Bamberg, Germany

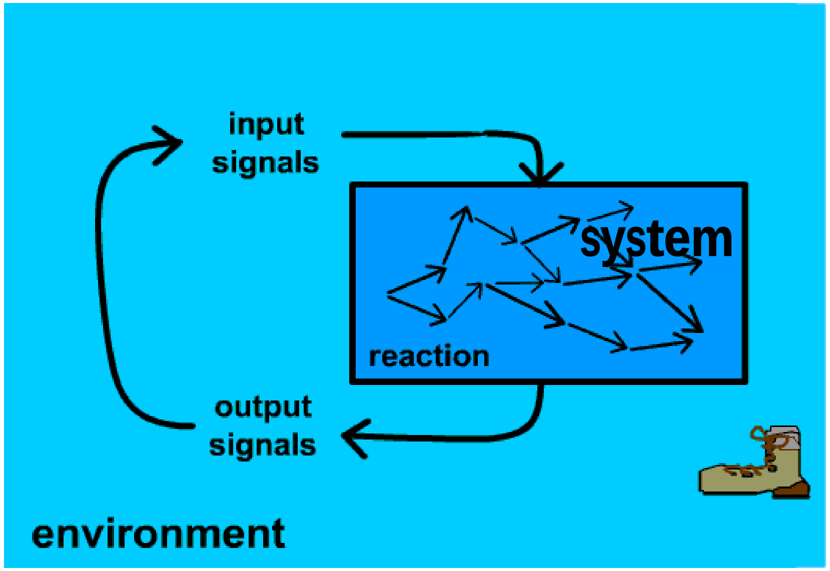
Overview

1. Introduction
2. When is a Circuit Combinational ?
3. When is a Logic Constructive ?
4. An Intuitionistic Modal Logic for Muller Automata (= inertial-delay circuit networks)
5. Constructive Muller Theories & Timed Ternary Simulation
6. Conclusion

1
INTRODUCTION

Synchrony Hypothesis

"A reactive system is faster than its environment, hence reactions can be considered atomic"



High-level Logical View

Reactions are

- discrete, atomic
- deterministic
- functional
- compositional

Low-level System Reality

Reactions may be

- continuous, non-atomic
- non-deterministic
- asynchronous
- causally entangled

The Grand Question

"A reactive system is faster than its environment, hence reactions can be considered atomic"

How to design & implement abstract system reactions so that

- they appear to operate in a functionally atomic way
- robustly and predictable,
- despite unavoidable low-level asynchrony with resource conflicts and scheduling uncertainties

This Class

"A reactive system is faster than its environment, hence reactions can be considered atomic"

We study some lessons learnt from the

- semantics of synchronous programming (e.g. Esterel)
- theory of asynchronous circuits

→ Object of Interest: **Constructive Circuits** [Gérard Berry 1999]

Background Literature

Asynchronous Hardware

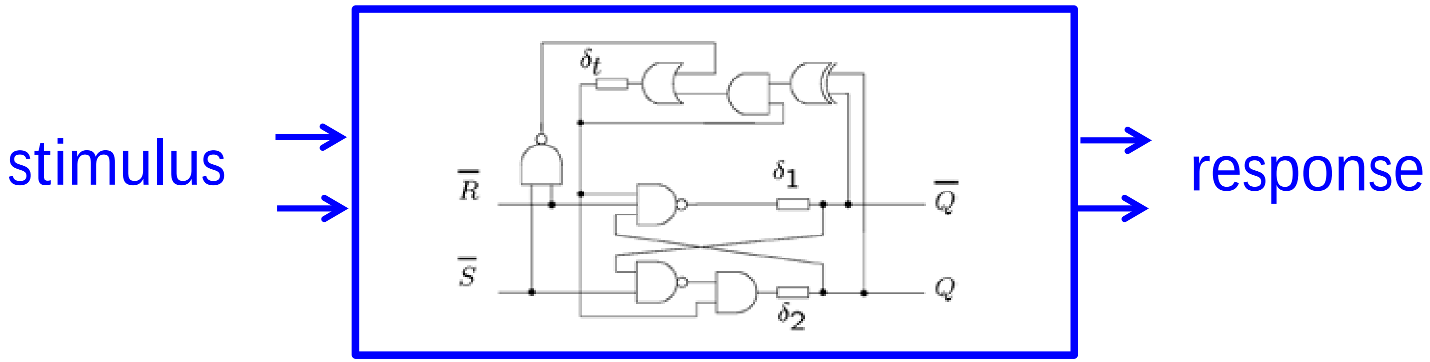
- R.E. Miller: Switching Theory, Vol.2: Sequential Circuits and Machines. John Wiley and Sons, 1965.
- S.H. Unger: Asynchronous Sequential Switching Circuits. Wiley-Interscience 1969
- M. Kishinevsky, A. Kondratyev, A. Taubin, V. Varshavsky: Concurrent Hardware:
- The Theory and Practice of Self-timed Design, Wiley 1994.
- J. A. Brzozowski, C.-J. H. Seger: Asynchronous Circuits. Springer 1995.

Synchronous Programming

- D. Potop-Butucaru, S. A. Edwards, G. Berry: Compiling Esterel. Springer 2002.
- G. Berry: Esterel de A à Z, https://www.college-de-france.fr/fr/agenda/cours/esterel-de-z
- Mendler, Shiple, Berry: Constructive Boolean circuits and the exactness of timed ternary simulation, Formal Methods in System Design, Vol.40, 2012.

2
WHEN IS A CIRCUIT COMBINATIONAL ?

Combinational Circuits

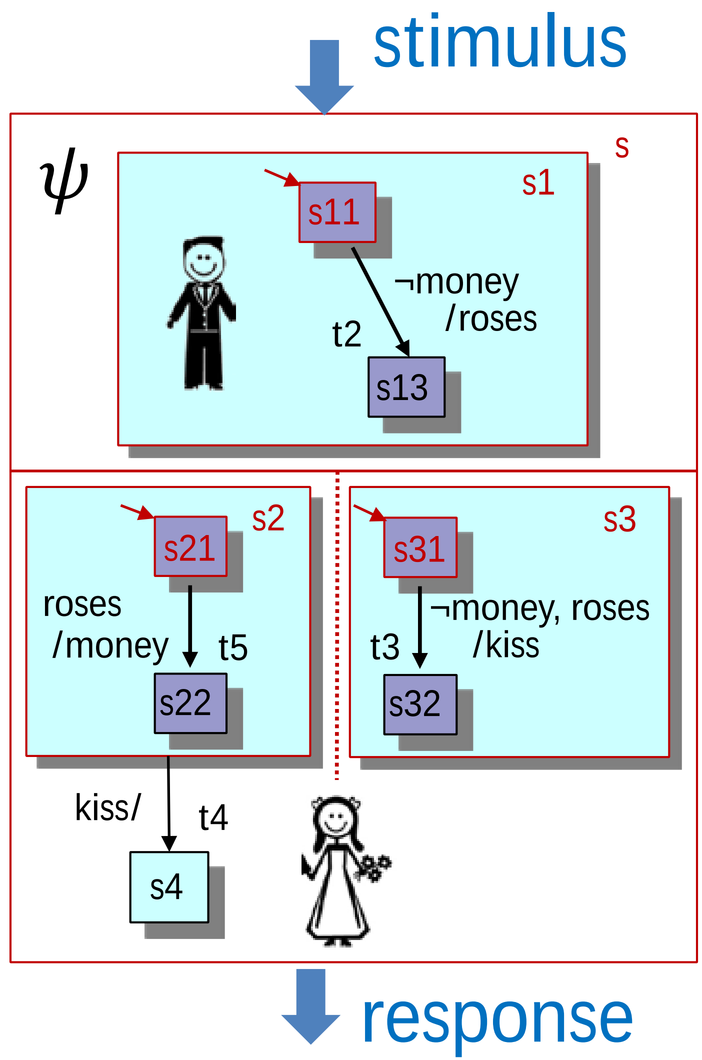


Definition (informal)

A general (possibly cyclic) circuit is combinational if it realises a functional relationship stimulus → response.

Synthesising combinational circuits is non-trivial…

Example: Romeo & Guilietta

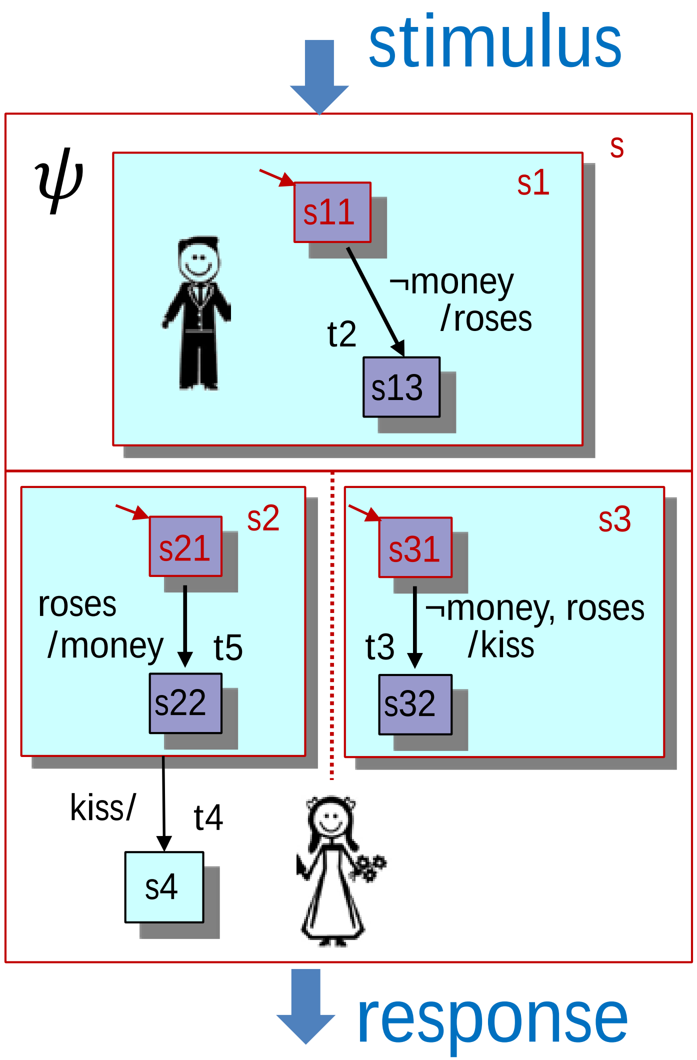


Synchronous Programming Model

Hierarchical, communicating state machines, e.g.:

- Statecharts [D. Harel 1987]
- Esterel [G. Berry 1983, 2000]
- SyncCharts [Ch. André 2003]
- Quartz [K. Schneider, 2009]
- SCCharts [R. von Hanxleden et al. 2014]
- Céu [F. Sant'Anna et al, 2017]
- Blech [F. Gretz & F.-J. Grosch, 2018]

Example: Romeo & Guilietta



Boolean Declarative Semantics

$x = 1$ state x active
signal x present/emitted
transition x fired

$x = 0$ state inactive
signal x absent/not emitted
transition x blocked

Logical Specification

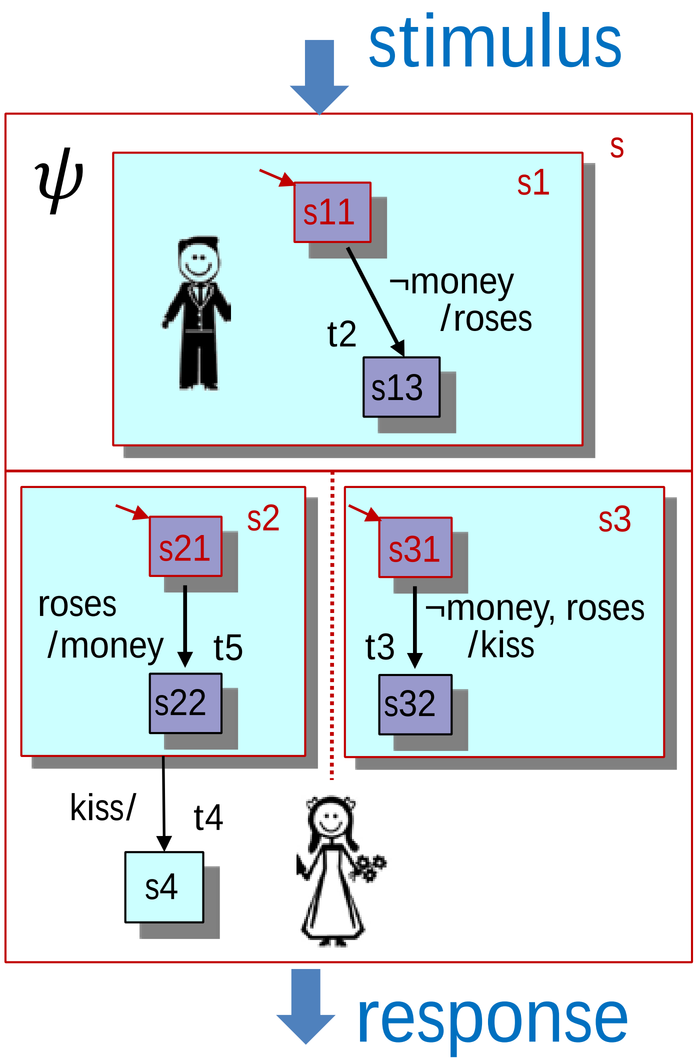
$t2 = \neg \text{money}$

$t3 = \neg \text{money} \wedge \text{roses}$ $\quad$ $\text{roses} = t2$

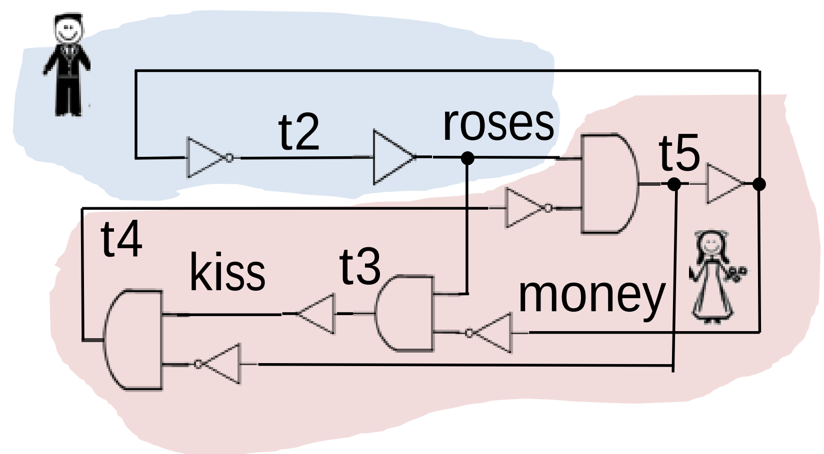
$t4 = \text{kiss} \wedge \neg t5$ $\quad$ $\text{kiss} = t3$

$t5 = \text{roses} \wedge \neg t4$ $\quad$ $\text{money} = t5$

Example: Romeo & Guilietta



Generated Boolean Circuit



Logical Specification

$t2 = \neg \text{money}$

$t3 = \neg \text{money} \wedge \text{roses}$

$t4 = \text{kiss} \wedge \neg t5$

$t5 = \text{roses} \wedge \neg t4$

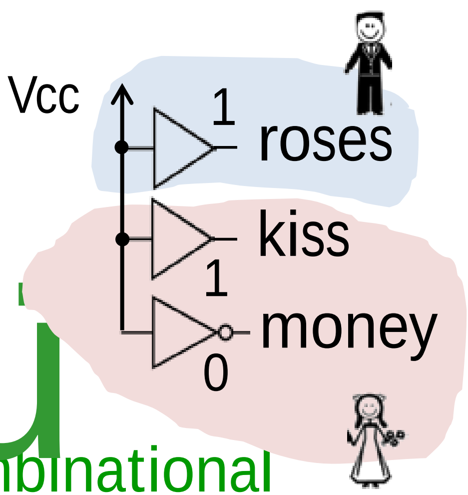
$\text{roses} = t2$

$\text{kiss} = t3$

$\text{money} = t5$

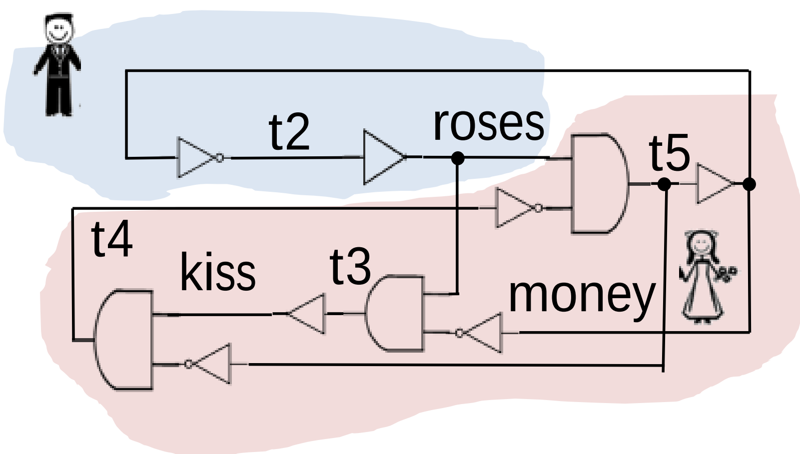
Example: Romeo & Guilietta

Equivalent Circuit



ü combinational

Generated Boolean Circuit



But are these circuits really equivalent ?

Unique Boolean Solution

$\mathsf{roses} = 1$

$\mathsf{kiss} = 1$

$\mathsf{money} = 0$

Boolean Simplification

Logical Specification

$t2 = \neg\mathsf{money}$

$t3 = \neg\mathsf{money} \wedge \mathsf{roses}$

$t4 = \mathsf{kiss} \wedge \neg t5$

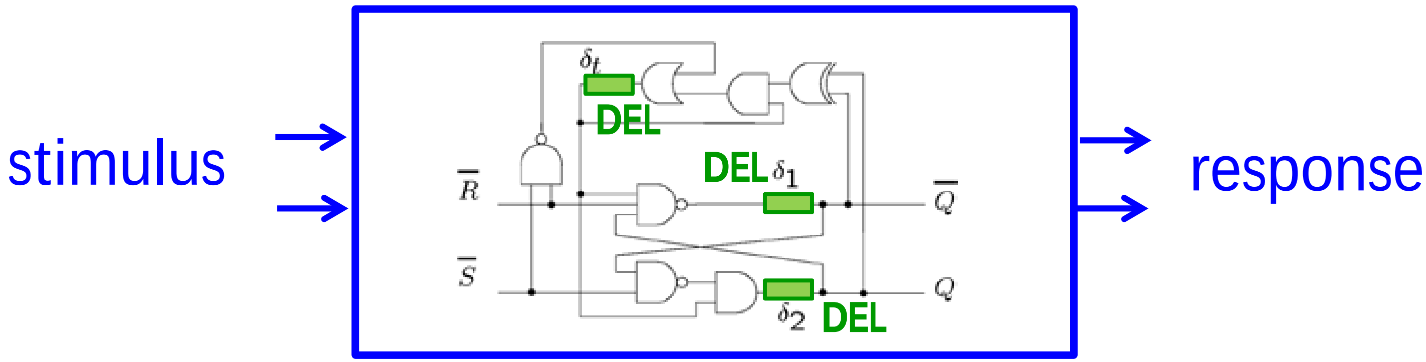
$t5 = \mathsf{roses} \wedge \neg t4$

$\mathsf{roses} = t2$

$\mathsf{kiss} = t3$

$\mathsf{money} = t5$

Combinational Networks



Refined Definition (operational)

Let DEL be a network delay/scheduling model.

A network is DEL-combinational (in fundamental mode) if for all constant input signals every network node

- stabilizes in bounded time
- to a unique response value

under DEL-execution semantics.

Some Delay Models

§ Fixed/up-bounded/bi-bounded Ideal Delay
[e.g., Lam/Brayton'94]

our focus

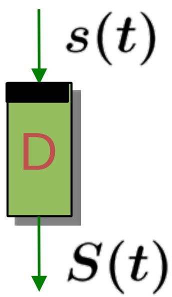
§ **UIN**: Up-bounded Inertial Delay
[Huffman'54, Miller'65, Brzozowski/Seger'89]

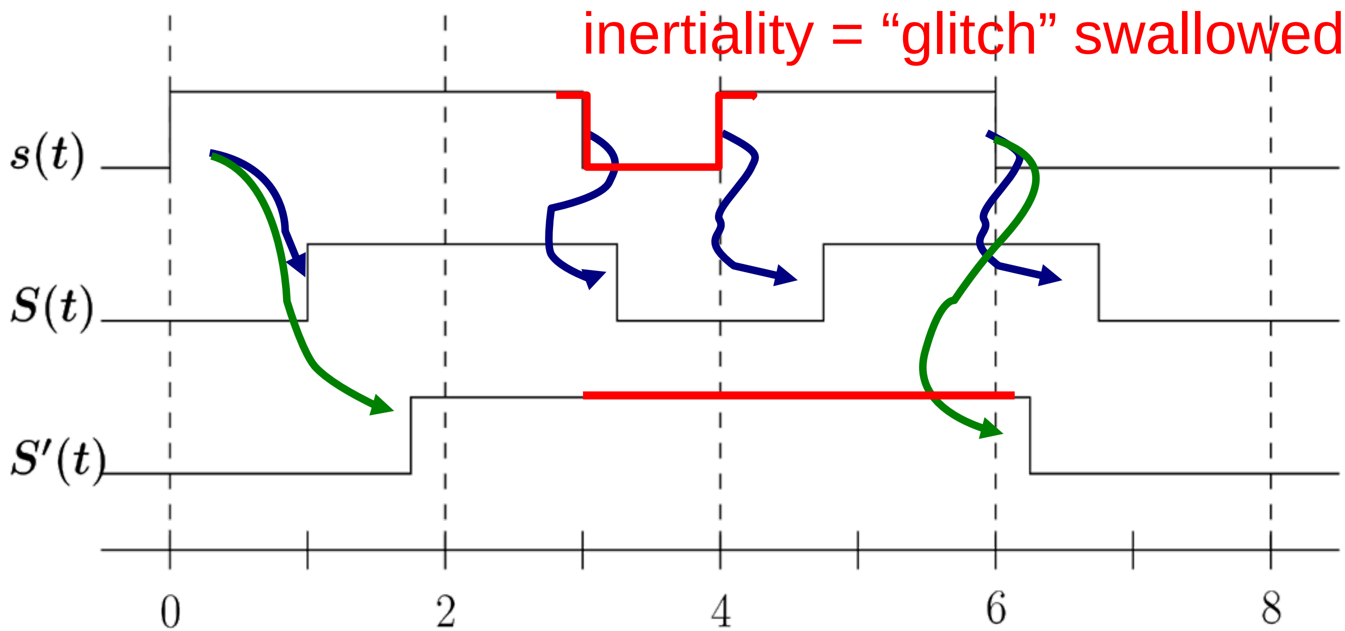
§ **UNI**: Up-bounded Noninertial Delay
[„XBD0", McGeer'92], [„binary chaos", Burch'92]
[„delay-modality", Mendler/Fairtlough'96]

- Bi-bounded Inertial Delay [Brzozowski/Seger'95]

Up-bounded Inertial Delay (UIN)

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





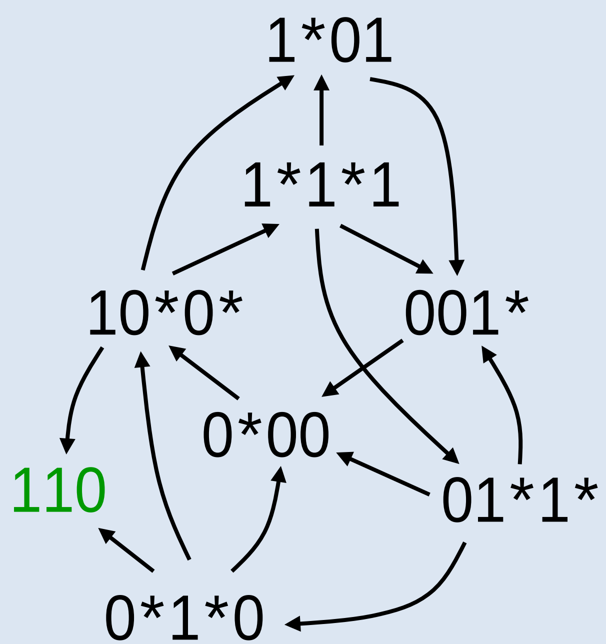
Example
D=2

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

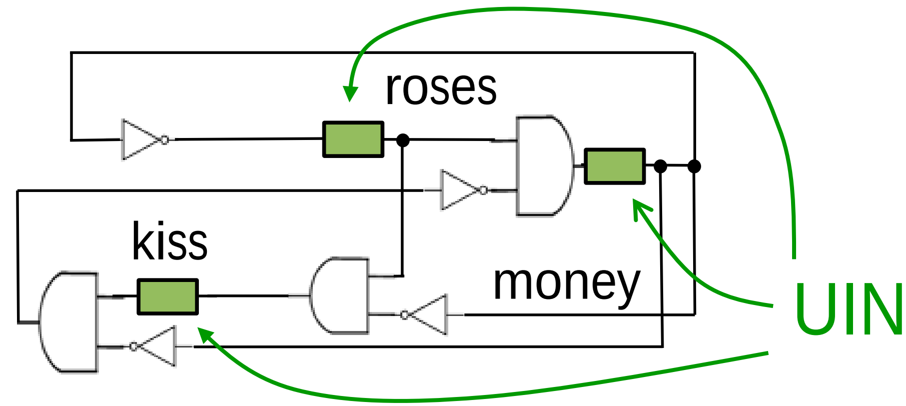
(2) **Inertiality:** The output only changes if delay is unstable

Example: Romeo & Guilietta

§ Muller Diagram [Muller'56]
UIN system trajectories



total state
$x\ y\ z =$ roses kiss money



$$\text{roses} :=_D \neg\text{money}$$

$$\text{kiss} :=_D \neg\text{money} \wedge \text{roses}$$

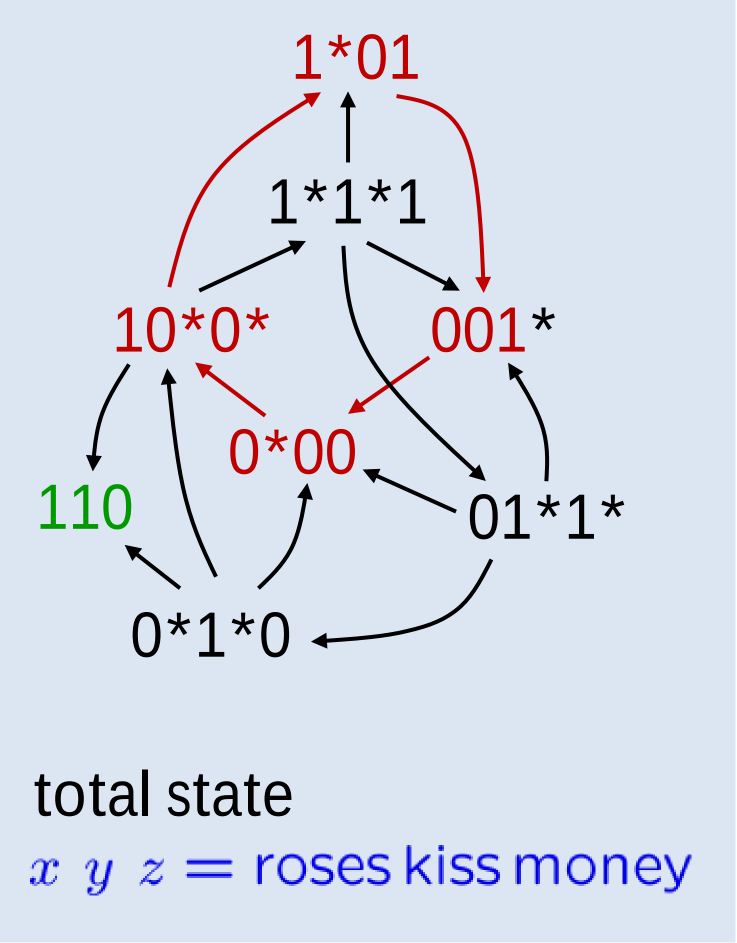
$$\text{money} :=_D \text{roses} \wedge \neg(\text{kiss} \wedge \neg\text{money})$$

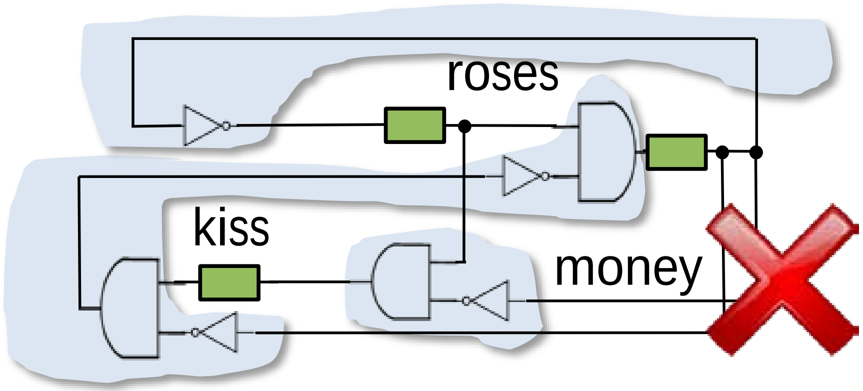
Up-bounded Inertial Delays (UIN)

≊ General Multiple Winner Model (GMW) *[Huffman'54, Brzozowski/Yoeli 79]*

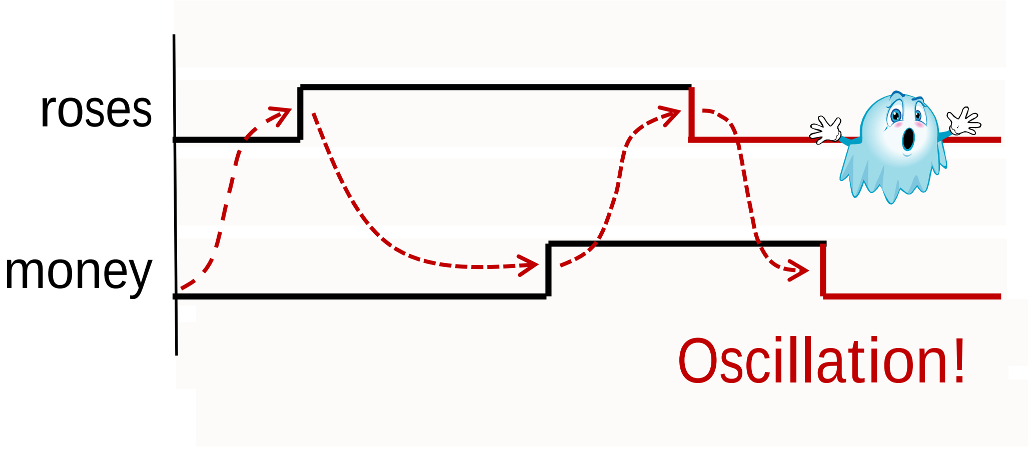
Example: Romeo & Guilietta

§ Muller Diagram [Muller'56]
UIN system trajectories



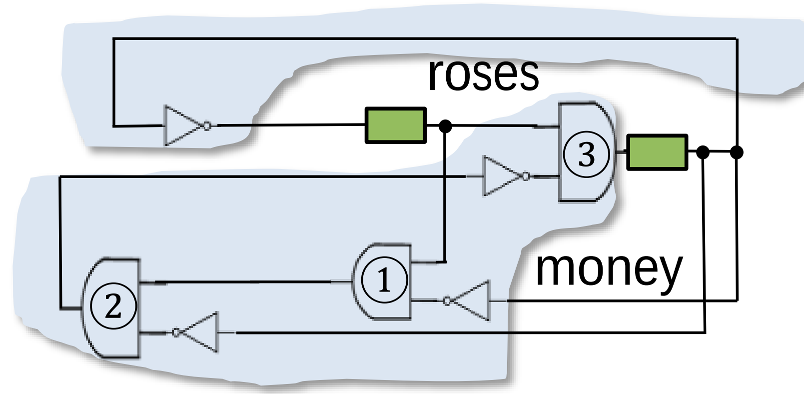


§ functional hazard: the network is not UIN-combinational !



Example Adjusted: No Delay in Kissing

- eliminate variable „kiss“
- only two equations
- ①②③ evaluated atomically



ü

$$\text{roses} :=_D \neg\text{money}$$

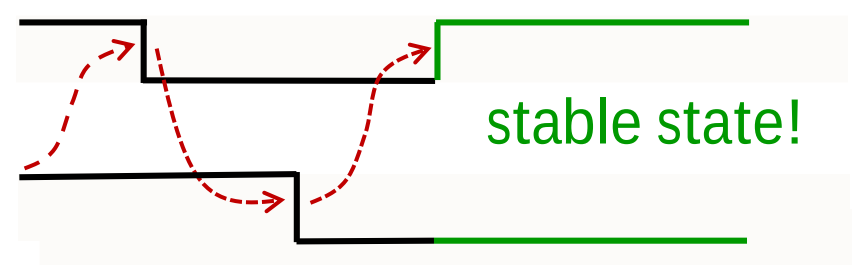
$$\text{money} :=_D \text{roses} \wedge \neg((\neg\text{money} \wedge \text{roses}) \wedge \neg\text{money})$$

1*1
↓
01*
↓
0*0
↓
10

total state

$x\ z = \text{roses money}$

The network is UIN-combinational !



- All UIN-trajectories converge

Delay-Insensitivity

Definition (informal)

A Boolean circuit is DEL-insensitive, if its behaviour is invariant under arbitrary introduction of DEL-delays in the gates' input and output wires.

Question

What would be an abstract specification language that

- extends Boolean algebra (in modest way)
- can identify UIN-combinational/delay-insensitive circuits
- is expressive enough to capture the effect of scheduling delays under causality and sharing („function hazards")?

Interludio Logico

3
WHEN IS A LOGIC CONSTRUCTIVE ?

Principles of Classical Logic

1. Aristotelian Truth: Every sentence is either true or false
2. Truth Functionality: Truth of a composite sentence is a function of the thruth value of its constituents

A	B	$A \supset B$
F	F	T
F	T	T
T	F	F
T	T	T

A	B	$\neg B$	$A \wedge \neg B$	$(A \wedge \neg B) \supset F$
F	F	T	F	T
F	T	F	F	T
T	F	T	T	F
T	T	F	F	T

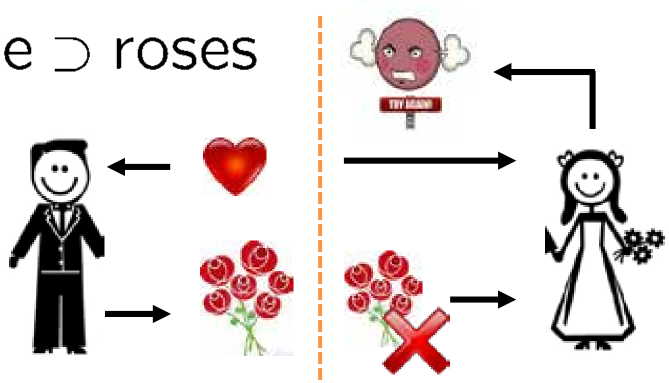
Sherlock Holmes Principle:

„If all contradictory scenarios have been excluded, what remains must be the truth“

Classical Logic is Reactively Inadequate

constructively distinct

$\text{love} \supset \text{roses}$



$(\text{love} \wedge \neg\text{roses}) \supset \text{false}$

classically equivalent

Omniscience of Classical Logic

From the classical “Principle of Omniscience“ the following is provable ... *[Bishop, Bridges: Constructive Analysis, Springer 1985]*

$$\vdash_{cl} \forall \text{Marriage} \in \text{Universe}.\ \exists \text{magic\_day} \in \text{Marriage}.\ (\text{love}(\text{magic\_day}) \supset \forall \text{day} \in \text{Marriage}.\ \text{love}(\text{day}))$$

QUANDO,
QUANDO,
QUANDO ...?

... yet, by all we know, constructively, this is nonsense!

Constructivity & Reactivity in System Design ...

$\Psi_{AI,\vec{a}}$ = set of all reactions of system AI under environment $\vec{a}$

s = boolean output signal of AI system

Assume $\Psi_{AI,\vec{a}}$ is constructive. Then ...

Thesis

Constructive Reactions (in constructive logic) are combinational & delay-insensitive !

- functionally determinate
- time-bounded
- stable, convergent, predictable, ...

es to 1

0 1

4

MULLER LOGIC:

AN (INTUITIONISTIC) MODAL LOGIC FOR MULLER AUTOMATA

Muller Logic in a Nutshell

Syntax

The formulas Φ of Muller Logic are given by

ϕ	$::=$	e	boolean expression over wire variables $\mathcal{Z}$.
	$\mid$	$\phi \wedge \phi$	conjunction
	$\mid$	$\phi \vee \phi$	disjunction
	$\mid$	$\phi \supset \phi$	(intuitionistic) implication
	$\mid$	$\Diamond_D \phi$	bounded delay with $D \in \mathbb{R}$
	$\mid$	$\Box \phi$	inertiality

The semantics is intuitionistic on time intervals ...

Muller Logic in a Nutshell

Definition

- A Muller theory Φ is a conjunction (or set) of formulas

$$\phi \;::=\; e \mid e \supset \Diamond_D e \mid e \supset \Box e$$

where e is a boolean expression over wires $\mathcal{Z}$ and D ∈ $\mathbb{R}$.

Semantics

$h \in \mathbb{R}^+ \to \mathbb{B}^{\mathcal{Z}}$ non-zeno, right-continuous waveform

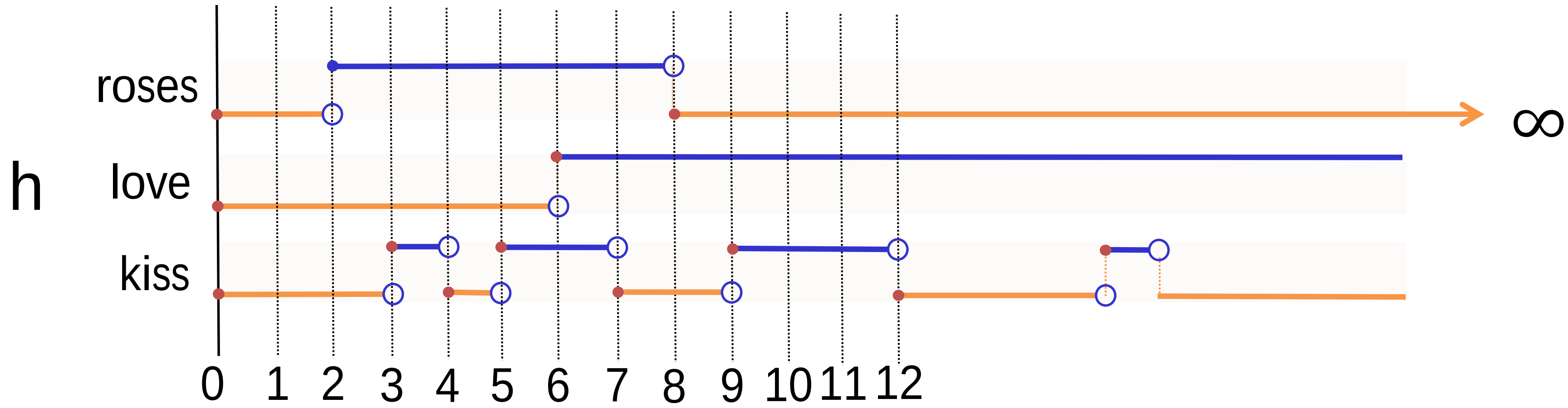
$I = [s, t)$ time interval

$h; I \models e$ "h;I remains in region e"

$h; I \models e_1 \supset \Diamond_D e_2$ "h;I must enter e_2 within D time, inside e_1"

$h; I \models e_1 \supset \Box e_2$ "h;I cannot ever enter $\overline{e_2}$ from inside e_1"

Example



$$h; [5, 8) \models \mathsf{roses} \cdot (\mathsf{love} + \mathsf{kiss})$$

“roses and always love or kiss“

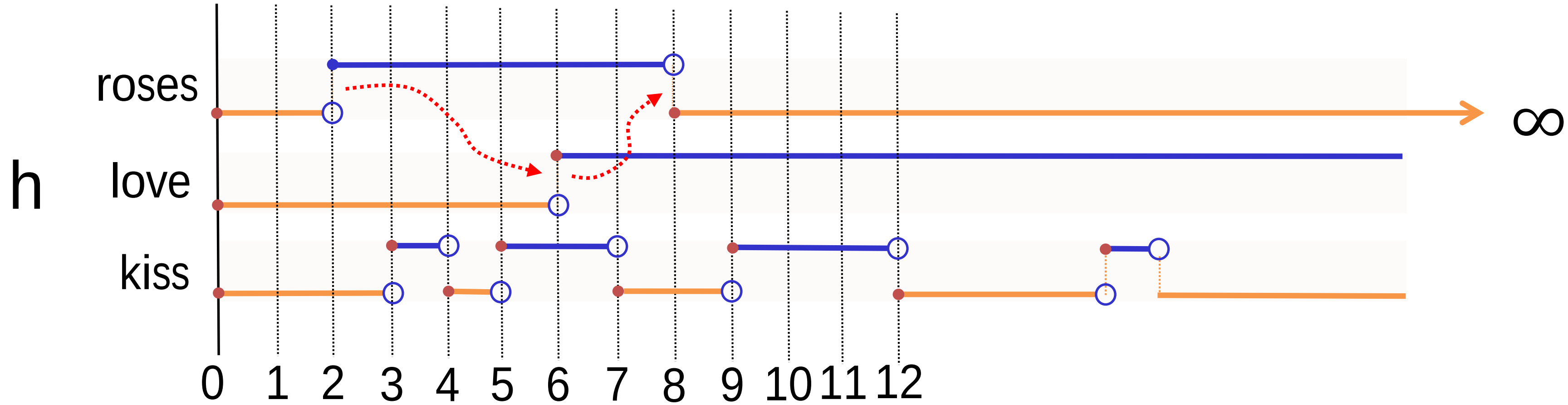
$$h; [0, \infty) \models \overline{\mathsf{roses}} \supset (\mathsf{love} \vee \overline{\mathsf{love}})$$

“while without roses, no change in love“

Boolean expressions hold pointwise throughout the interval

$\Diamond$ Modality for Propagation Delay („Set-up“)

$$h;[s,t) \models \Diamond_D \phi \quad \text{iff} \quad s + D < t \Rightarrow h;[s+D,t) \models \phi.$$

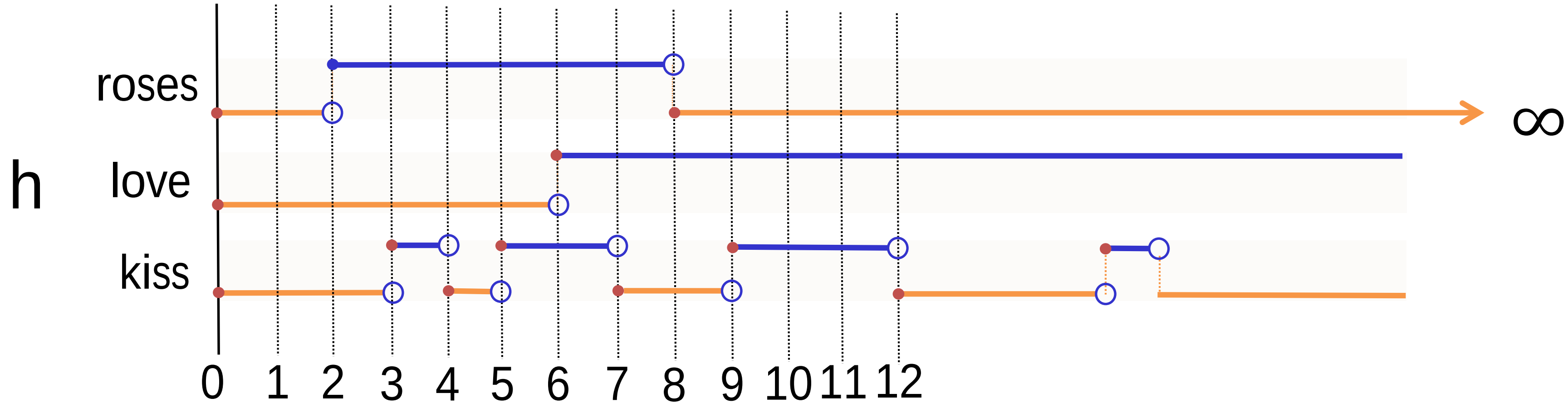


$$h;[0,\infty) \models (\text{roses} \supset \Diamond_4 \text{love}) \wedge (\text{love} \supset \Diamond_2 \overline{\text{roses}})$$

$$h;[0,\infty) \models \text{kiss} \supset \Diamond_3 \text{false}$$

□ Modality for Inertiality („Hold“)

$$h; [s, t) \models \Box\phi \quad \text{iff} \quad s < t \Rightarrow \exists \delta > 0.\ h; [s, t + \delta) \models \phi$$



$$h; [0, \infty) \models (\text{love} \supset \Box\text{love}) \wedge (\overline{\text{roses}} \supset \Box\overline{\text{roses}})$$

$$h; [0, \infty) \models (\overline{\text{love}} + \text{kiss}) \supset \Box\text{roses}$$

(Up-bounded) Muller Theories

Definition

- A Muller theory Φ is a conjunction of formulas

$$\phi \;::=\; e \mid e \supset \Diamond_D e \mid e \supset \Box e$$

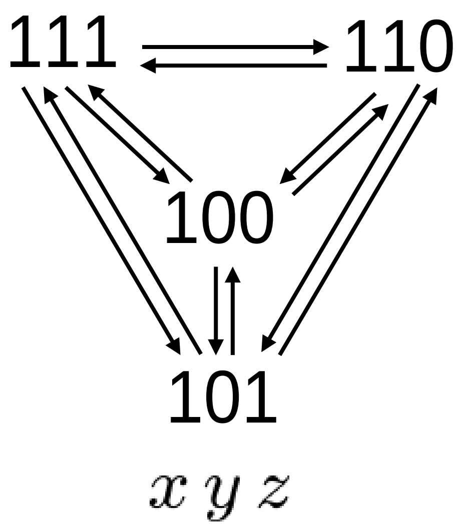
where e is a boolean expression over wires $\mathcal{Z}$ and D ∈ $\mathbb{R}$.

Theorem

- Muller theories Φ specify the (timed) General Multiple Winner behaviour $\mathrm{GMW}(\Phi)$ upbounded inertial delay (UIN) Boolean networks.

Specifying GMW Automata in Muller Logic

$$\Phi = \{ x \}$$



(self-loops omitted)

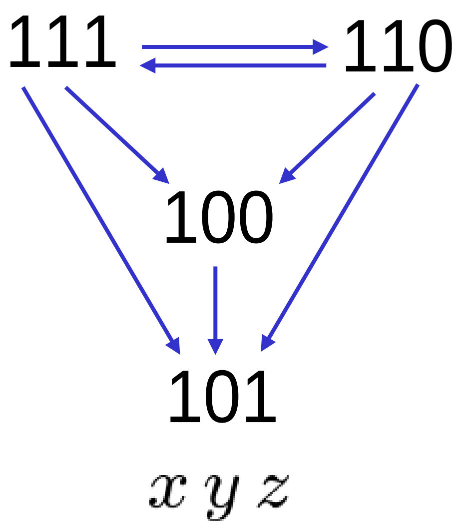
Specifying GMW Automata in Muller Logic

$$\Phi = \{ x,$$

inertiality

$$\overline{y} \supset \Box \overline{y}$$

$$\overline{y} \cdot z \supset \Box z \}$$



Specifying GMW Automata in Muller Logic

$$\Phi = \{ x,$$

inertiality

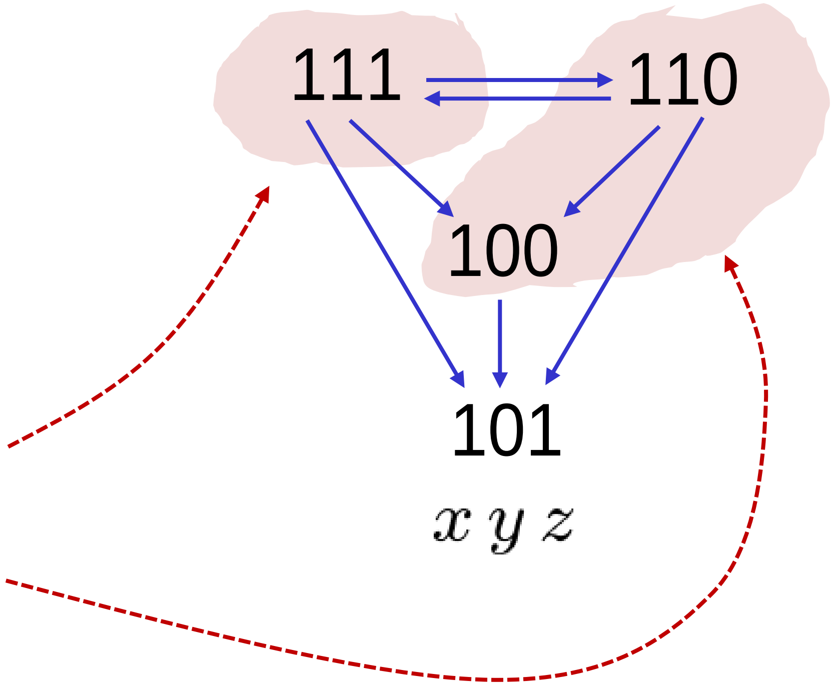
$$\overline{y} \supset \Box \overline{y}$$

$$\overline{y} \cdot z \supset \Box z$$

contraction

$$y \cdot z \supset \Diamond_D \mathit{false}$$

$$\overline{z} \supset \Diamond_E \mathit{false} \; \}$$



bounded regions

Specifying GMW Automata in Muller Logic

$$\Phi = \{\, x,$$

inertiality

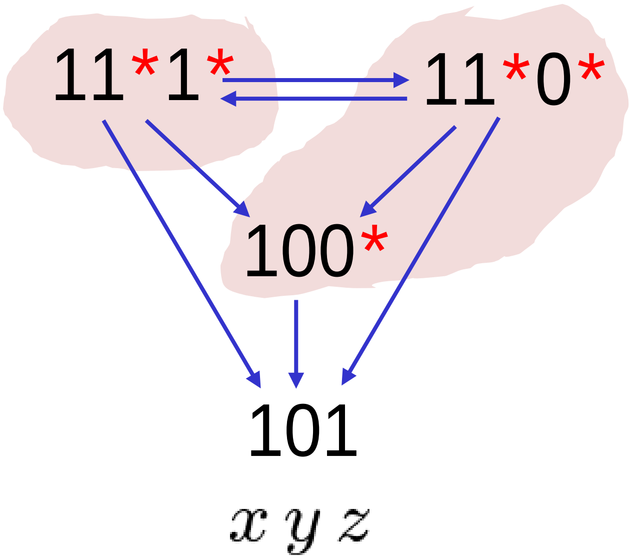
$$\overline{y} \supset \Box \overline{y}$$

$$\overline{y} \cdot z \supset \Box z$$

contraction

$$y \cdot z \supset \Diamond_D \mathit{false}$$

$$\overline{z} \supset \Diamond_E \mathit{false} \,\}$$

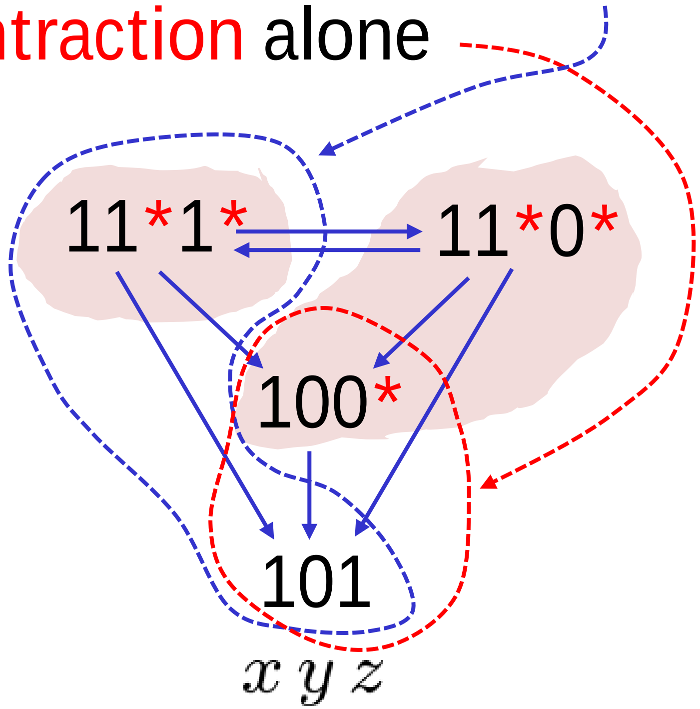


Fairness: „The system trajectory cannot infinitely remain inside a transient region“

Specifying GMW Automata in Muller Logic

$\Phi \models z \supset \Diamond_D \overline{y}$ forced by (contraction and) inertiality

$\Phi \models \overline{y} \supset \Diamond_E z$ forced by contraction alone



$$\Phi = \{ x,$$

inertiality

$$\overline{y} \supset \Box \overline{y}$$

$$\overline{y} \cdot z \supset \Box z$$

contraction

$$y \cdot z \supset \Diamond_D \mathit{false}$$

$$\overline{z} \supset \Diamond_E \mathit{false} \}$$

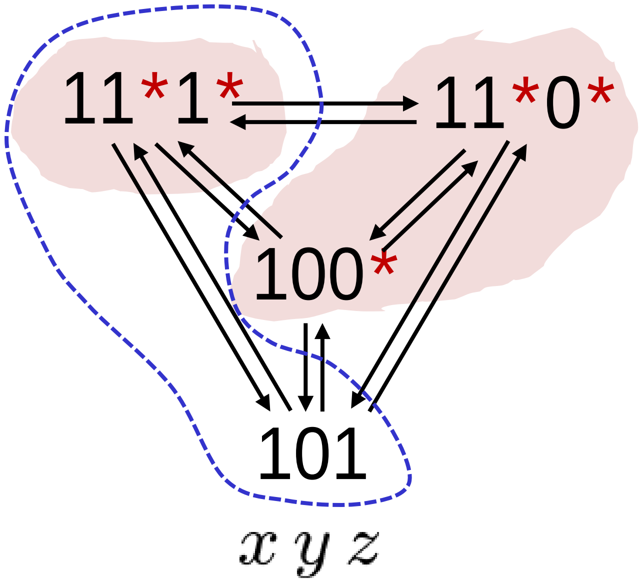
Fairness: „The system trajectory cannot infinitely remain inside a transient region“

Specifying GMW Automata in Muller Logic

$\Phi \not\models z \supset \Diamond_D \overline{y}$ ~~forced by inertiality~~

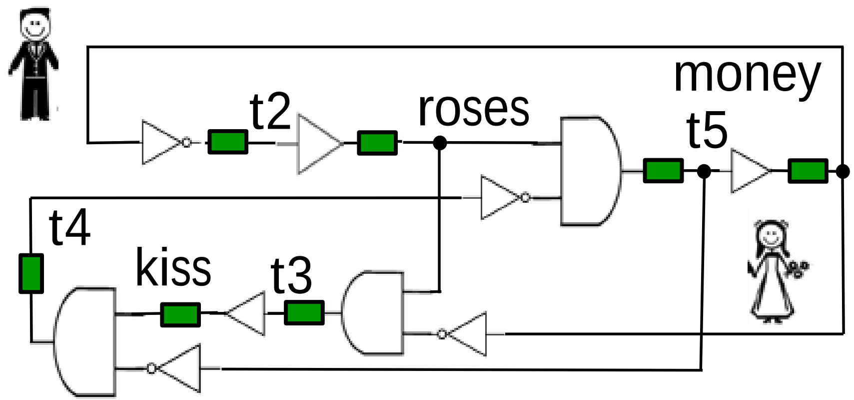
$\Phi \models \overline{y} \supset \Diamond_E z$ forced by contraction

$$\Phi = \{ x, \quad y \cdot z \supset \Diamond_D \mathit{false} \quad \overline{z} \supset \Diamond_E \mathit{false} \}$$



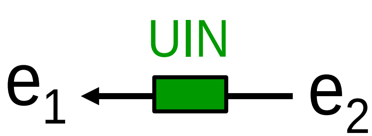
Purely non-inertial theory may lose stabilistion !

Upbounded Inertial Delay (Romeo & Guilietta)



$$t2 :=_D \overline{\text{money}}$$
$$t3 :=_D \overline{\text{money}} \cdot \text{roses}$$
$$t4 :=_D \text{kiss} \cdot \overline{t5}$$
$$t5 :=_D \text{roses} \cdot \overline{t4}$$
$$\text{money} :=_D t5$$
$$\text{roses} :=_D t2$$
$$\text{kiss} :=_D t3$$

$$\Phi_{UIN}$$

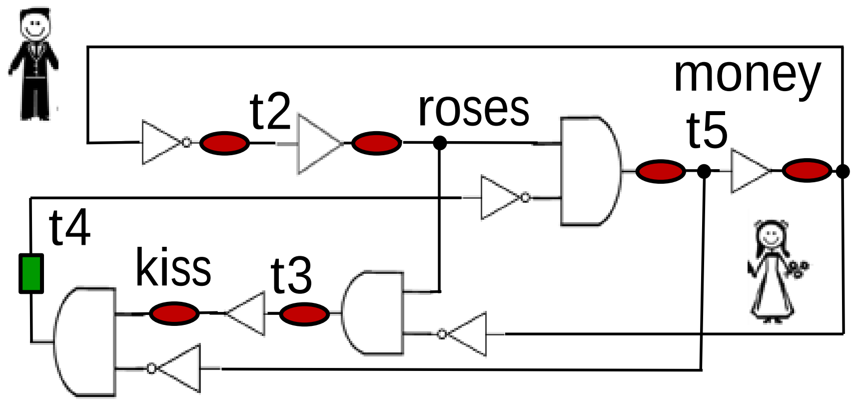


$e_1 :=_D e_2$ stands for

$$(\overline{e_2} \supset \Diamond_D \overline{e_1}) \wedge (e_2 \supset \Diamond_D e_1)$$
$$(e_1 \cdot e_2 \supset \Box e_1) \wedge (\overline{e_1} \cdot \overline{e_2} \supset \Box \overline{e_1})$$

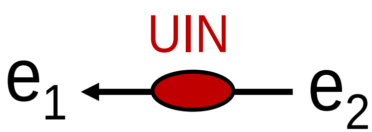
$$\Phi_{UIN} \models \Diamond(\text{roses} \cdot \text{kiss} \cdot \overline{\text{money}})$$

Upbounded Non-Inertial Delay (Romeo & Gulietta)



$$t2 :\approx_D \overline{\mathrm{money}}$$
$$t3 :\approx_D \overline{\mathrm{money}} \cdot \mathrm{roses}$$
$$t4 :\approx_D \mathrm{kiss} \cdot \overline{t5}$$
$$t5 :\approx_D \mathrm{roses} \cdot \overline{t4}$$
$$\mathrm{money} :\approx_D t5$$
$$\mathrm{roses} :\approx_D t2$$
$$\mathrm{kiss} :\approx_D t3$$

$$\Phi_{UNI}$$



$e_1 :\approx_D e_2$ stands for

$$(\overline{e_2} \supset \Diamond_D \overline{e_1}) \wedge (e_2 \supset \Diamond_D e_1)$$

$$\Phi_{UNI} \not\models \Diamond(\mathrm{roses} \cdot \mathrm{kiss} \cdot \overline{\mathrm{money}})$$

5

CONSTRUCTIVE MULLER THEORIES & TIMED TERNARY SIMULATION

Constructive Muller Theories

Definition

- A Muller theory Φ is constructive if
  $\Phi \vDash \Diamond_D(e_1 \vee e_2)$ implies $\Phi \vDash \Diamond_D e_1$ or $\Phi \vDash \Diamond_D e_2$.
- Φ is stabilising if $\forall z \in \mathcal{Z}$ there is D with $\Phi \vDash \Diamond_D(z \vee \neg z)$.
- A Muller theory Φ is non-inertial if it does not contain the $\Box$ operator.

Theorem *[derived from Mendler, Shiple, Berry 2012]*

- Every non-inertial Muller theory is constructive.
- Stabilisation can be decided by timed ternary simulation...

(Timed) Ternary Algebra

§ Recursion theory [Kleene‘52]

§ Asynchronous Circuits & Fault-modelling (hazards, races, oscillation)

[Yoeli/Rinon‘64, Eichelberger‘65, Roth‘66]

[Bryant‘87] CMOS transistor-level simulation

§ Analysis of Muller Automata

[Yoeli/Brzozowski‘77, Brzozowski/Seger‘95] A/B-algorithms

§ Cyclic Combinational Circuits

[Burch/et.al.‘93, Malik‘93, Shiple‘96]

[Huang/Parng/Shyu‘91] Timed D-Calculus

[Fairtlough/Mendler’96,Mendler/Shiple/Berry‘2012]

modal logic/real-time interpretation

[Namjoshi/Kurshan‘99, Backes/Fett/Riedel‘2008]

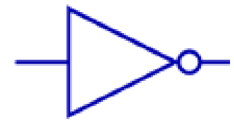
improved (untimed) Algorithm

§ Synchronous programming

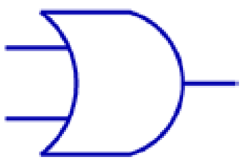
[Berry‘99, Schneider/Brandt/Schüle’2004, ...]

Timed Ternary Algebra

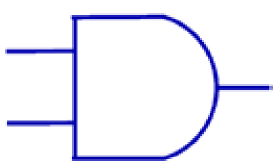
DEL(d)	
$[\alpha, s]$	$[\alpha, s+d]$
$\bot$	$\bot$



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



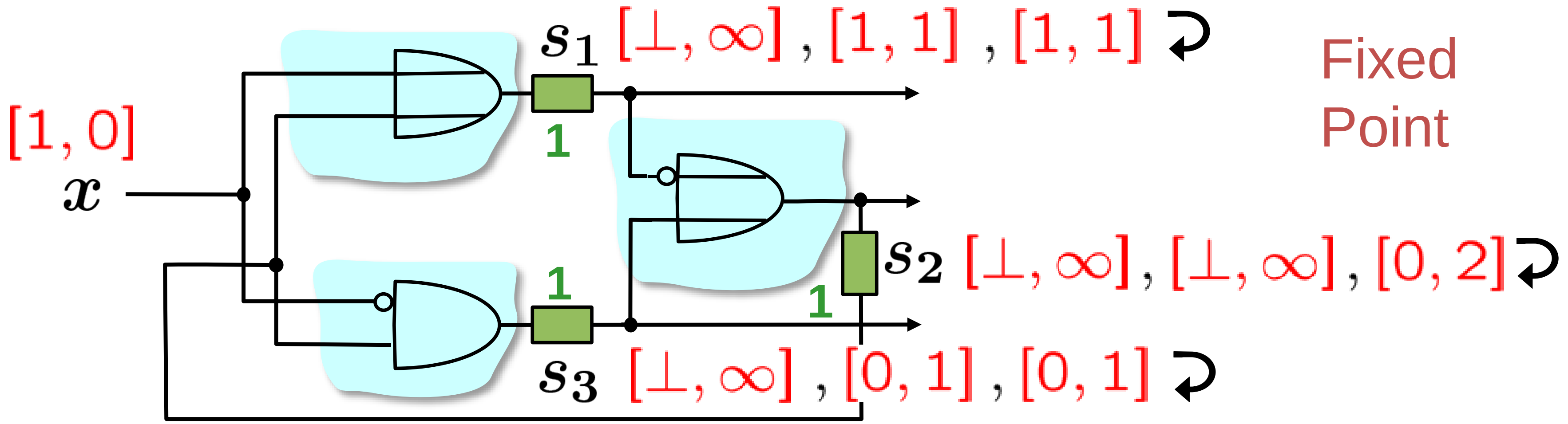
OR	$[0, t]$	$[1, t]$	$\bot$
$[0, s]$	$[0, max(s, t)]$	$[1, t]$	$\bot$
$[1, s]$	$[1, s]$	$[1, min(s, t)]$	$[1, s]$
$\bot$	$\bot$	$[1, t]$	$\bot$



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

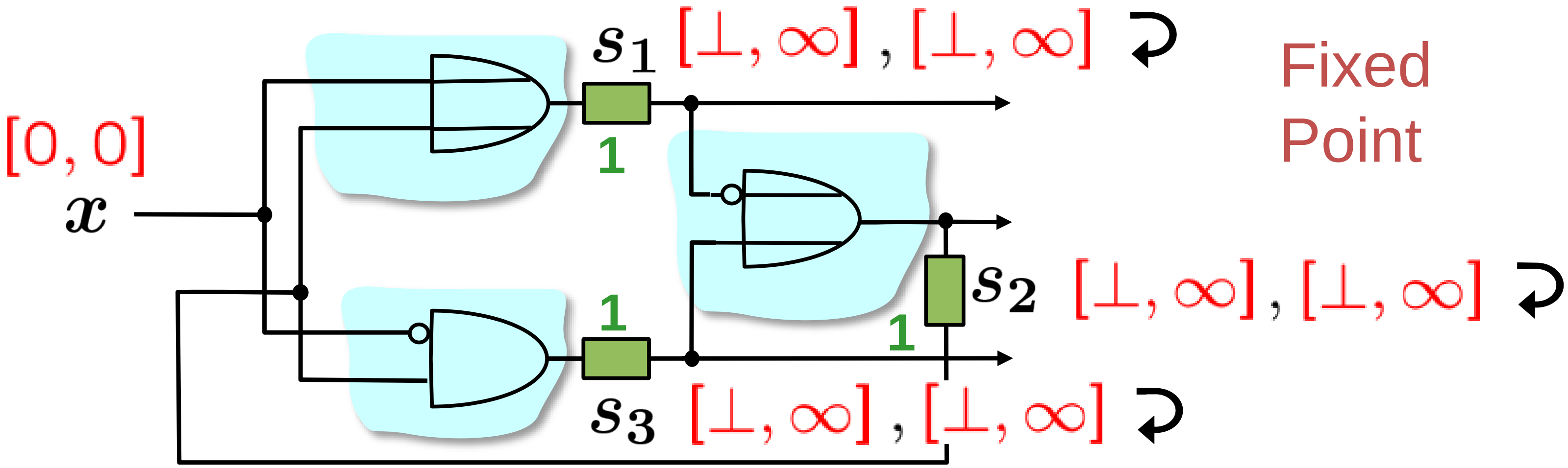
$\bot = [0, \infty) = [1, \infty) = [\bot, t)$

Example I



s_1 $[\bot, \infty]$, $[1, 1]$, $[1, 1]$
$[1, 0]$
x
1
1
1
s_2 $[\bot, \infty]$, $[\bot, \infty]$, $[0, 2]$
s_3 $[\bot, \infty]$, $[0, 1]$, $[0, 1]$
Fixed
Point

Example I



Fixed Point

Constructive Networks

Theorem

The following statements are equivalent:

- A network N is semantically stabilising in **non-inertial Muller-Logic** *axiomatic*
- The **ternary simulation** of N generates Boolean solutions for the state variables *denotational*
- N reaches in bounded time a unique steady state under **non-inertial delay** assumptions *operational*

6
CONCLUSION

Summary

- Intuitionistic Muller Logic (NEW!)
  - expressively adequate specification langage for Booean Networks
  - for inertial and non-inertial delay models
- Timed Ternary Simulation as an algorithmic decision procedure for non-inertial delay networks
- Definition of „Constructive Circuits“ (G. Berry) as networks that are stabilising
  - in constructive Muller Logic (axiomatic)
  - in timed ternary simulation (denotational)
  - under non-inertial delay scheduling (operational)

Open Research Problems

- Complete axiomatisation of Muller Logic
- Computational complexity of decision procedures
- Proof that complete (input and output) inertial delay networks are equivalent to non-inertial delay networks and thus constructive
- Exact separation between delay-insensitive and speed-independent networks in Muller Logic.

Nota Bene: These results are relevant (mutatis mutandis) to distributed systems at higher levels of abstraction, too (RTL, distributed shared memory, middleware, …)