Correct-by-construction asynchronous implementation of modular synchronous specifications

Benoît Caillaud - IRISA, Rennes, France
Dumitru Potop - INRIA, Rocquencourt, France

Outline

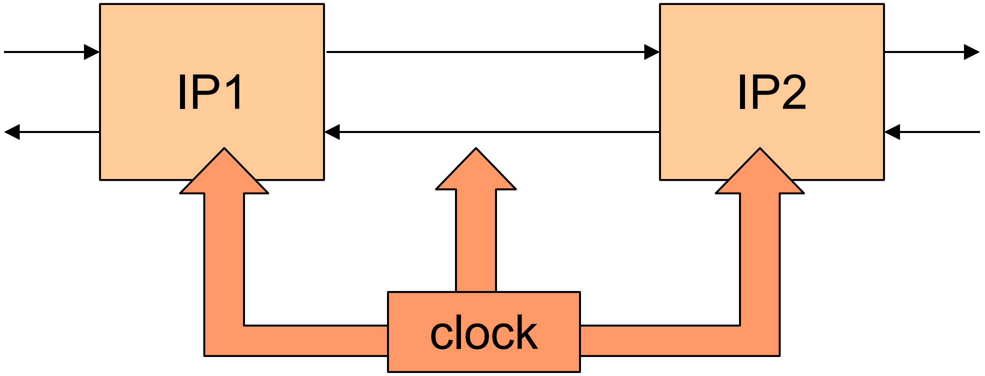
- Motivation: Asynchronous implementation of synchronous specifications
  - GALS architectures
  - Desired efficient implementation
- Formal model
  - Correctness
- Correctness criteria
  - Microstep weak endochrony
  - Microstep weak isochrony
- Conclusion

Synchrony, asynchrony, GALS

- Synchronous specification
  - Global clock ⇒ ease of specification & verification
  - Popular, efficient tools for system design (digital circuits, safety-critical systems)
- Distributed implementation
  - Distributed software, complex digital circuits (SoC/NoC), heterogeneous systems
  - Loosely-connected components (asynchronous FIFOs...)
- GALS architectures = good implementation model
  - Synchronous components, asynchronous communication
  - Problem: preserve semantic consistency between synchronous specification and GALS implementation

What we want

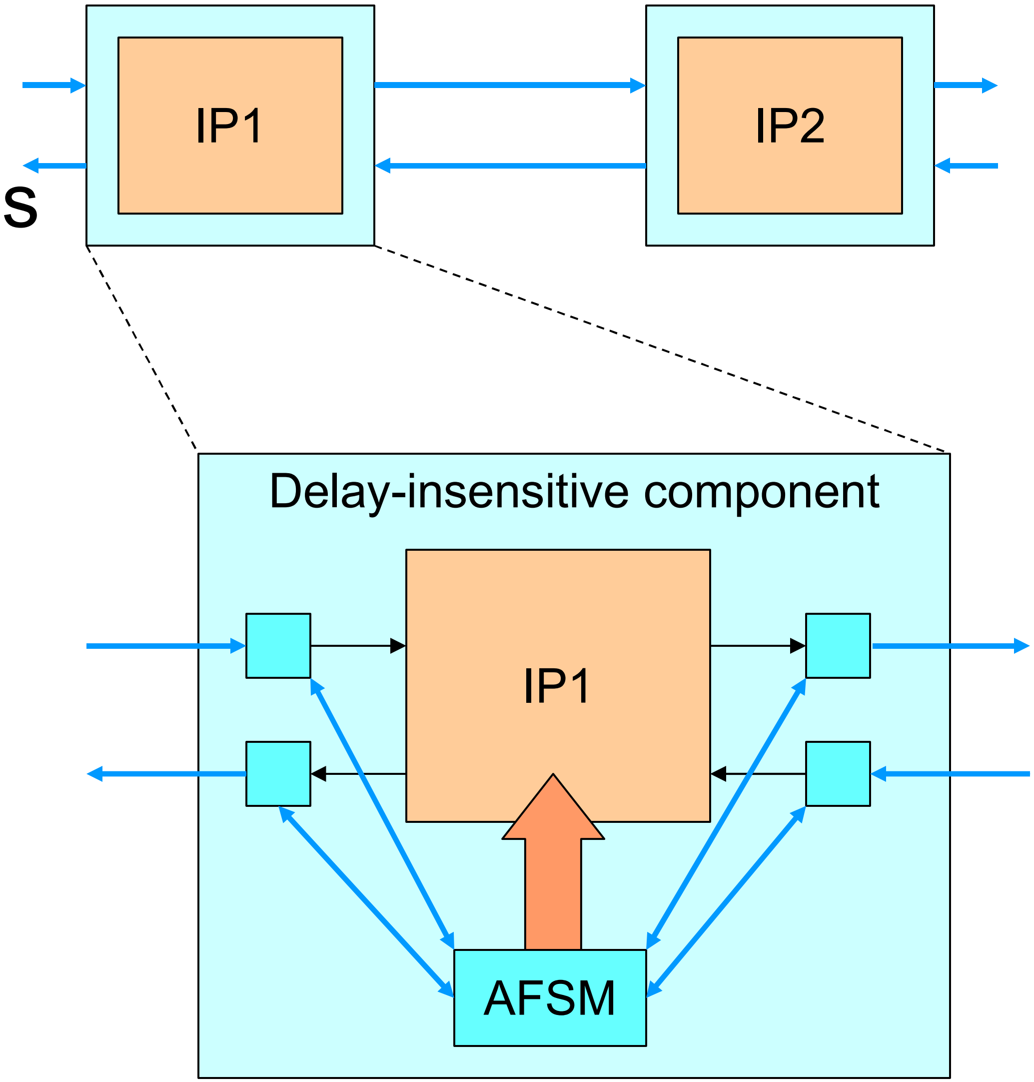
1. Take a modular synchronous specification



What we want

1. Take a modular synchronous specification

2. Replace comm. with asynchronous FIFOs, wrappers



3. Preserve:
   - Functionality
   - Correctness
     - No “extra” traces
     - No deadlocks

   (Kahn processes)
   - Parallelism

Previous work

- Latency-insensitive systems
  - Carloni & Sangiovanni-Vincentelli (1999)
  - Goal: independence from communication delays
  - Global synchrony: system speed = slowest component speed
- Endo/isochronous systems
  - Benveniste, Caillaud, Le Guernic (1999)
    - Version: Generalized latency-insensitive circuits (Singh, Theobald, 2003)
  - Goals:
    - minimize communication
    - maximize concurrency, independence between system components
  - Not compositional!

Previous work

- Weakly endo/isochronous systems
  - Potop, Caillaud, Benveniste (2004)
  - Goals:
    - further minimize communication by exploiting intra-component concurrency
    - **Compositionality !**
  - Synchronous Mazurkiewicz traces
  - Does not handle causality and communication deadlocks

- This work: microstep weakly endo/iso systems
  - Goal: take into account causality and composition through read/write mechanisms

.

Our approach

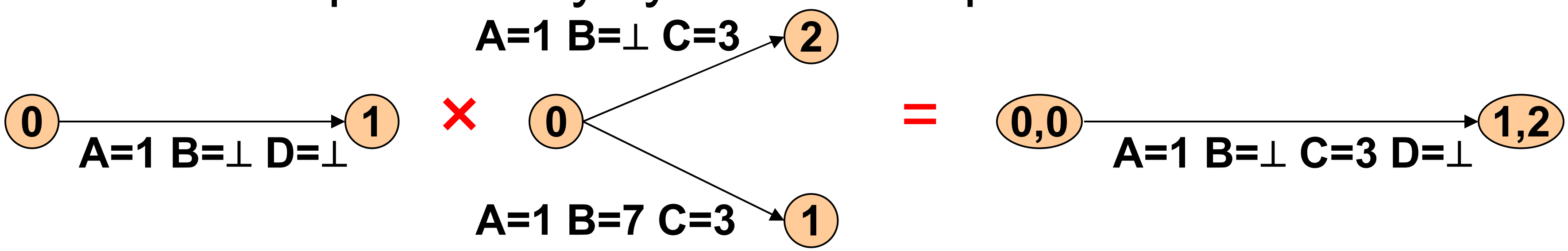
- Define a model and criteria ensuring that:
  - Creating delay-insensitive wrappers that preserve the semantics is possible without adding new signals
  - Connecting through FIFOs the resulting components produces a semantics-preserving, deadlock-free GALS implementation
- Make given components satisfy the criteria:
  - Possible solutions
    - Encode (part of) the “absent” events (Carloni et al.)
    - Add new signals
    - Decide that none is necessary due to environment constraints
- Efficient sw/hw implementation
  - Sync./async. synthesis techniques, GALS-specific communication schemes, etc.

The model: basic definitions

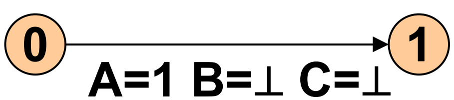
- The basics: (incomplete) automata

  $\Sigma = (S, s_0, V, \rightarrow), \ \rightarrow \subset S \times L(V) \times S, \ L(V) = \prod_{v \in V} (D_v \cup \bot)$

  – Composition by synchronized product:



  – Renaming operator: $\Sigma_1[D/C]$:



- Labels
- Finite runs:

A=1 B=⊥ C=3 ≡ A=1 C=3

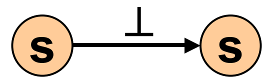
A=1 C=3 ≤ A = 1 B=7 C=3

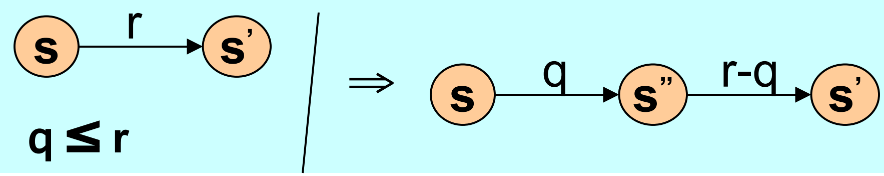
A=1 C=3 – A=1 = C=3

A=1 C=3 ; B=2 ; ;

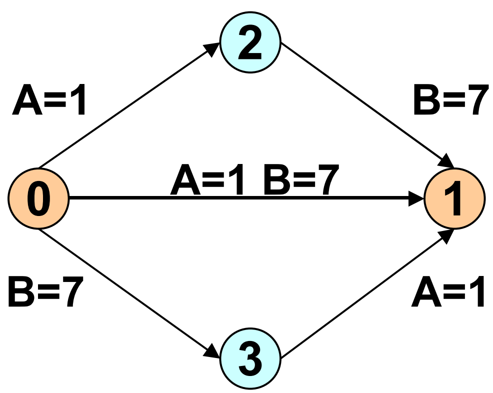
A=1 C=3 ; B=2 ; ; ≼ A=1 C=3 ; B=2 ; ; A=2;

The model: basic definitions

- Generalized concurrent transition systems(GCTS)
  - Void transitions: 

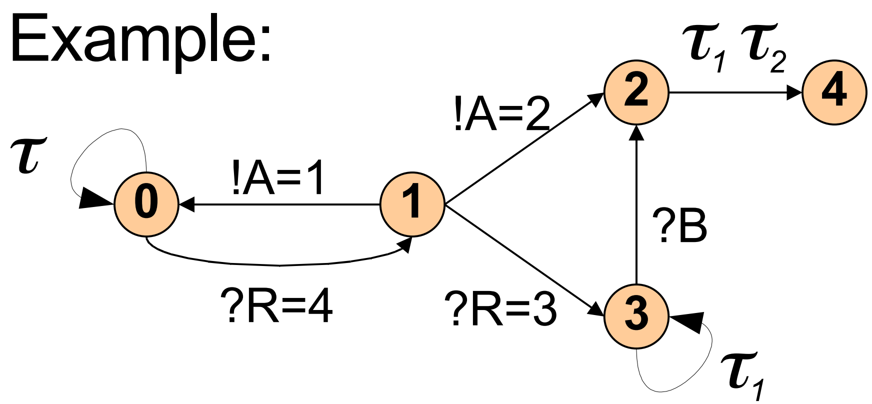
  - Down closure: 

- Example:



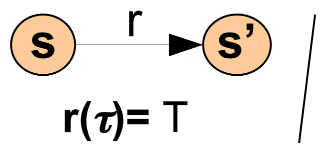
The model: I/O transition systems

- Point-to-point communication:
  - Broad/Multicast can be simulated…
  - Communication channels: c = (!c,?c) $D_{!c}=D_{?c}=D_c$
  - Dissociate emission from reception!
- Clocks: $\tau\ \tau_1$… of domain $D_{clk}=\{T\}$
- I/O transition system:
  - GCTS where all variables are channels or clocks
  - Example:



The model: synchronous systems

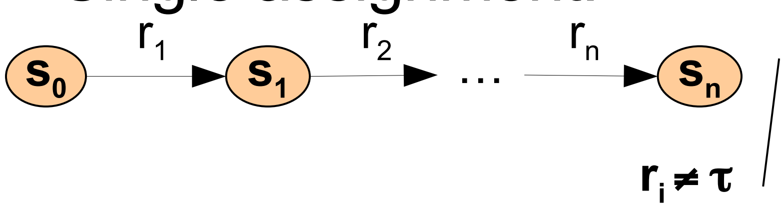
- Synchronous system: $\Sigma = (S, s_0, V, \tau, \rightarrow)$
  I/O transition system, one clock, and satisfying:
  1. Clock transitions:



$\Rightarrow$ **r** equals $\perp$ over **V**

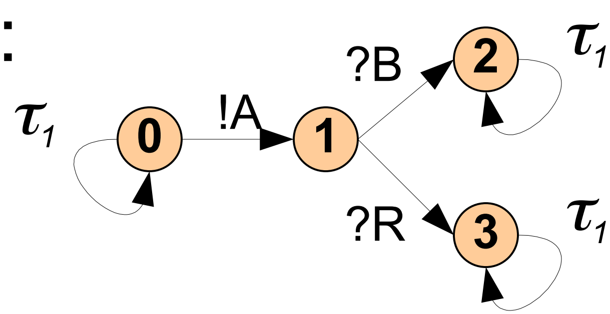
  3. Stuttering invariance:

  5. Single assignment:



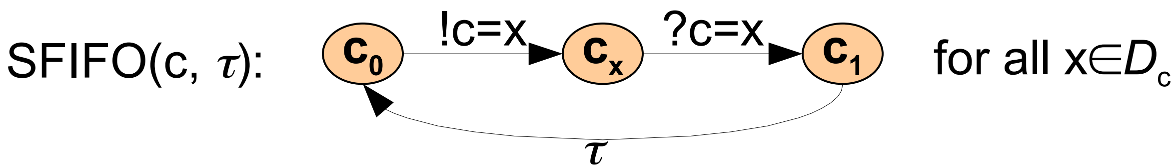
$\Rightarrow$ $\mathbf{supp(r_i) \cap supp(r_j) = \emptyset}$ for all $\mathbf{i \neq j}$

- Example:



The model : composition

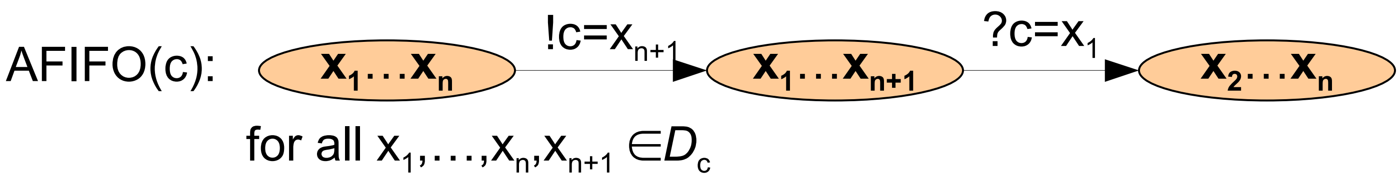
- Synchronous 1-place register:



- Synchronous composition (on clock τ) :

$$\Sigma_1|\Sigma_2 = \Sigma_1[\tau_1/\tau] \times \Sigma_2[\tau_2/\tau] \times SFIFO(c_1, \tau) \times \ldots \times SFIFO(c_n, \tau)$$

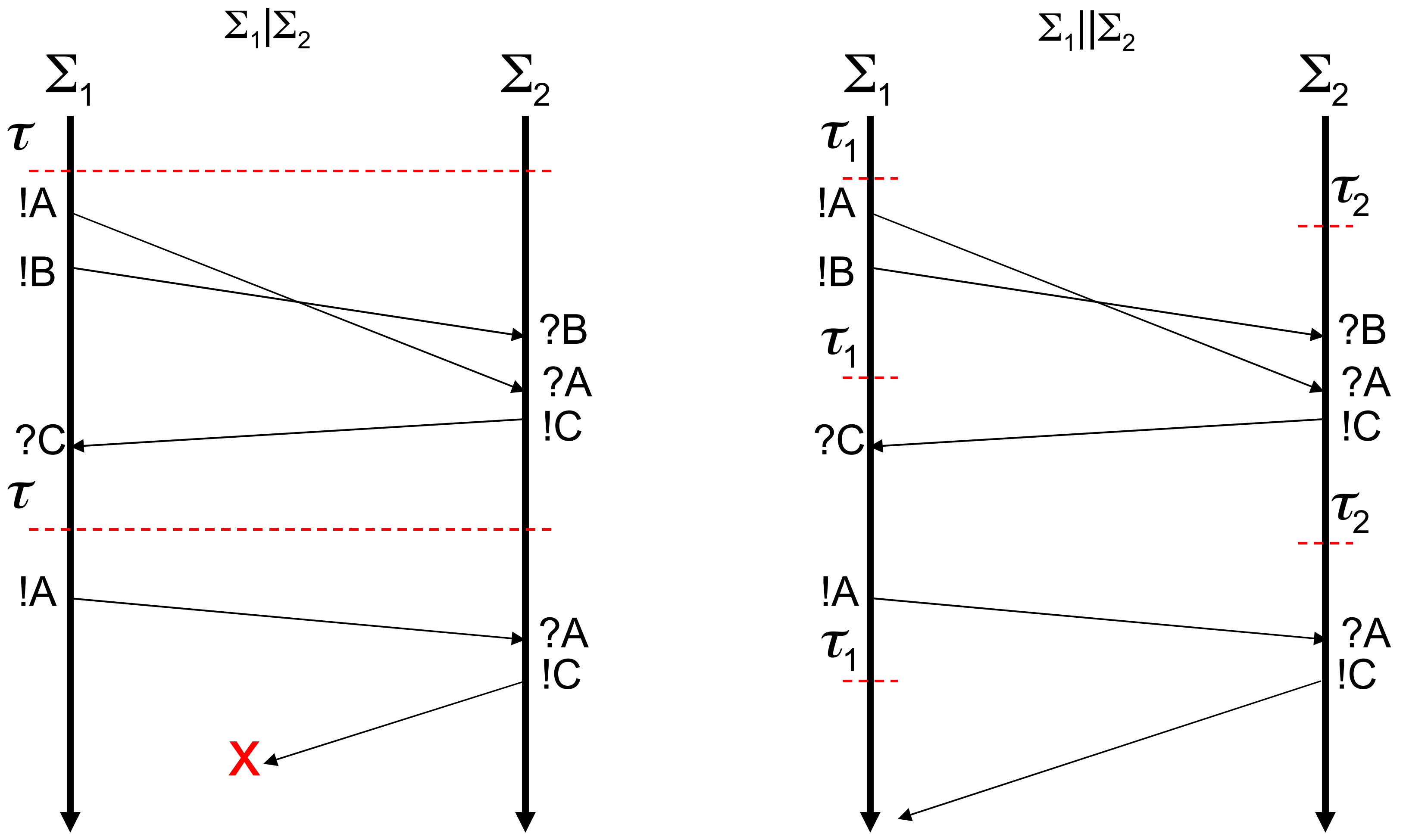
- Asynchronous FIFO:



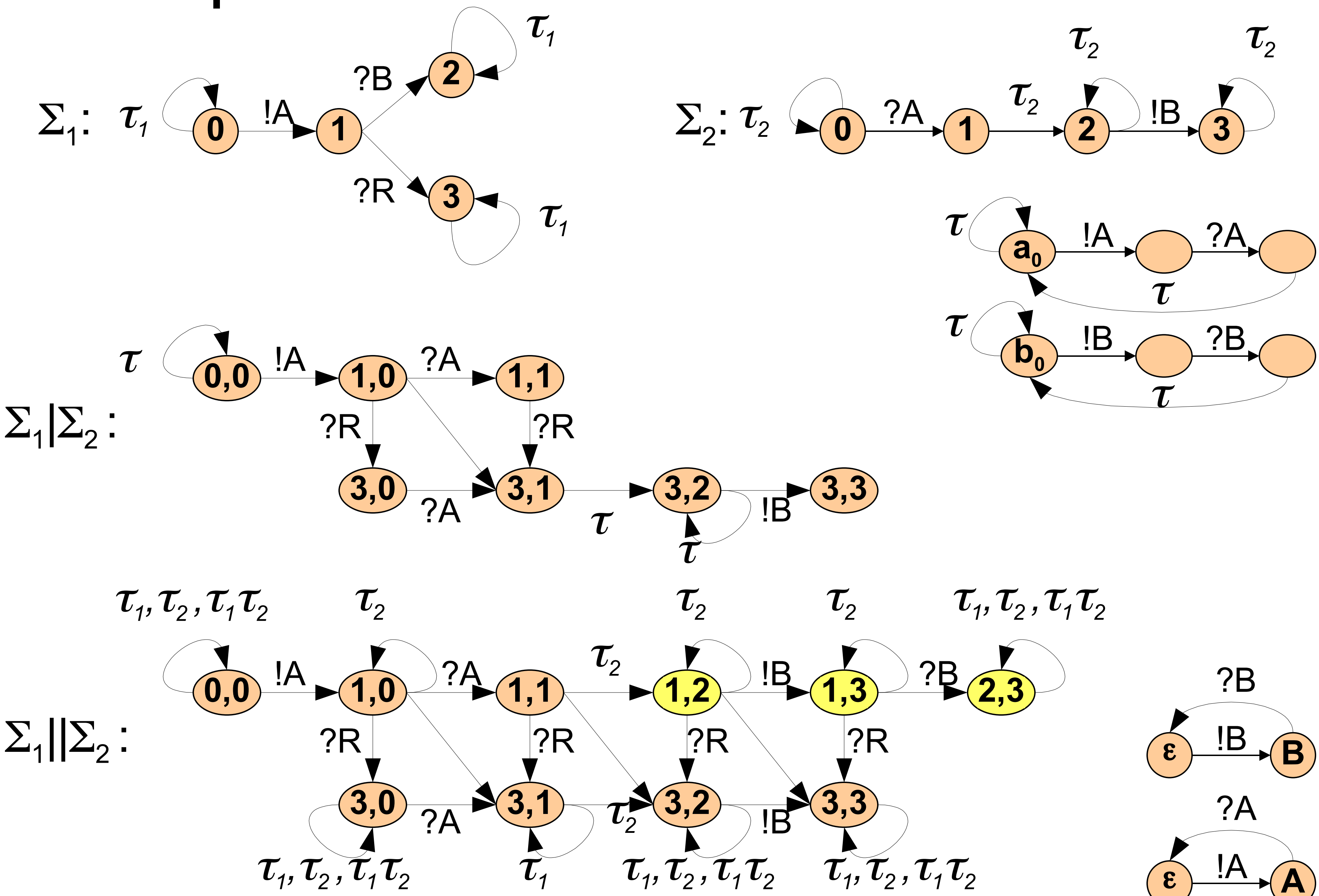
- Asynchronous composition:

$$\Sigma_1||\Sigma_2 = \Sigma_1 \times \Sigma_2 \times AFIFO(c_1) \times \ldots \times AFIFO(c_n)$$

The model : composition



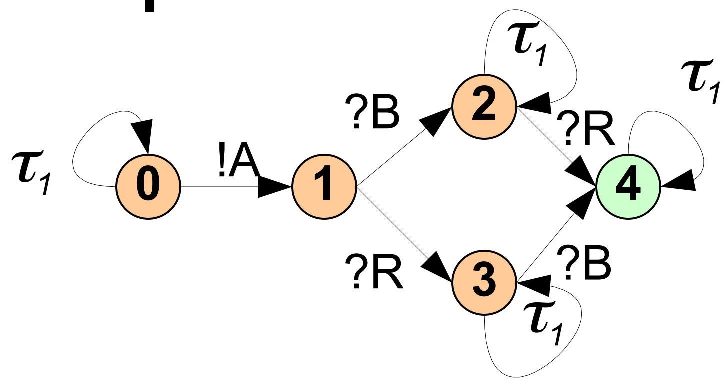
Example



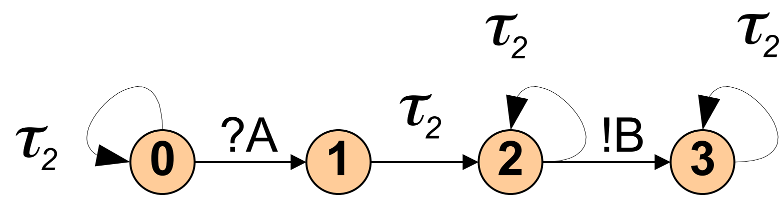
Σ1: τ1
0
!A
1
?B
2
τ1
?R
3
τ1
Σ2: τ2
0
?A
1
τ2
2
τ2
!B
3
τ2
τ
a0
!A
?A
τ
τ
b0
!B
?B
τ
Σ1|Σ2 :
τ
0,0
!A
1,0
?A
1,1
?R
?R
3,0
?A
3,1
τ
3,2
τ
!B
3,3
Σ1||Σ2 :
τ1, τ2, τ1τ2
τ2
τ2
τ2
τ1, τ2, τ1τ2
0,0
!A
1,0
?A
1,1
τ2
1,2
!B
1,3
?B
2,3
?R
?R
?R
?R
3,0
?A
3,1
τ2
3,2
!B
3,3
τ1, τ2, τ1τ2
τ1
τ1, τ2, τ1τ2
τ1, τ2, τ1τ2
?B
ε
!B
B
?A
ε
!A
A

Example

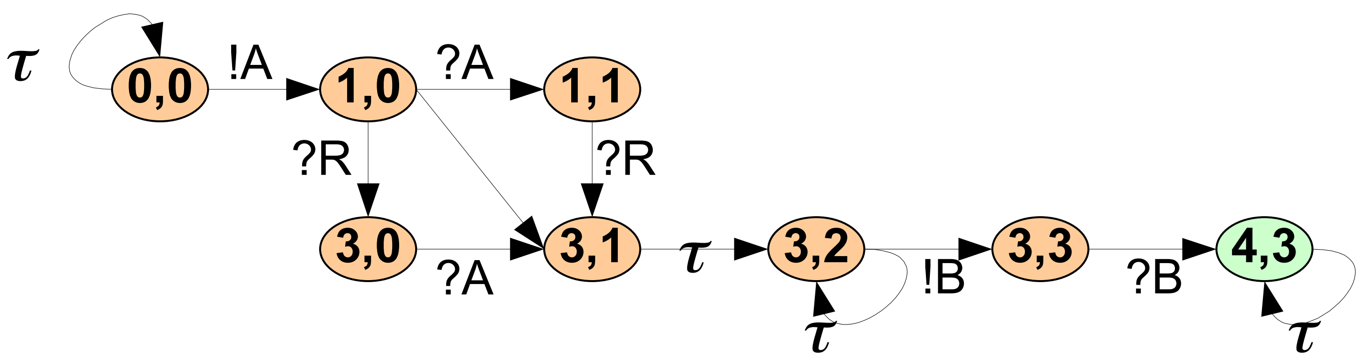
Σ_1:



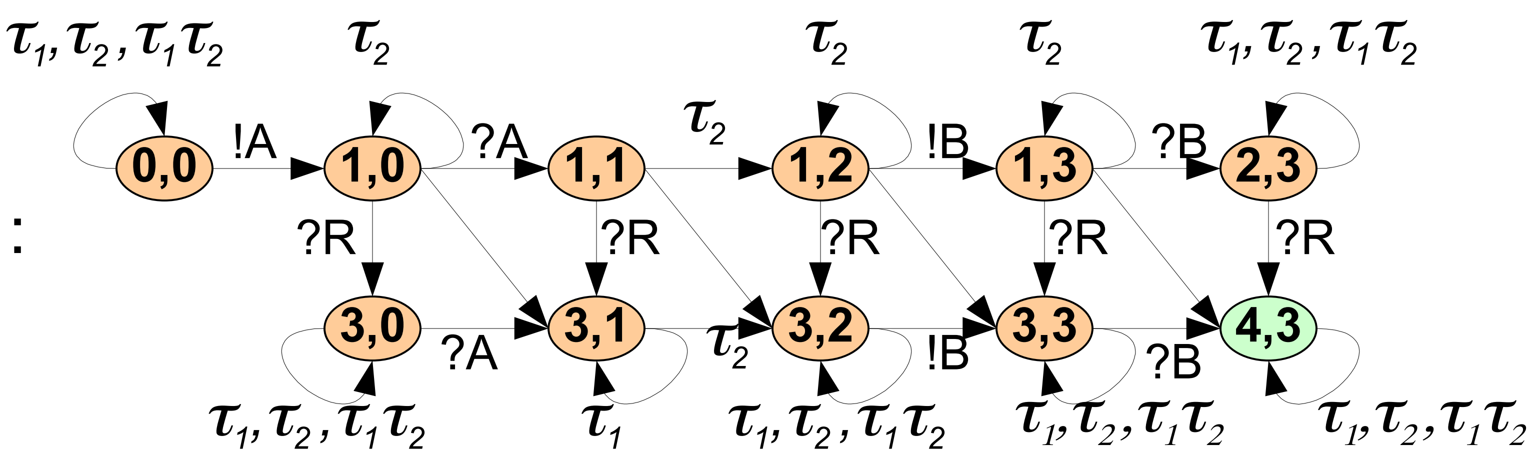
Σ_2:



$\Sigma_1|\Sigma_2$:



$\Sigma_1||\Sigma_2$:



Correctness

- Some notations:

  !A=1 ; τ_1; ?A=1 ; τ_2 ; !C=3 ; $\sim$!A=1 ?A=1 ; $\tau_1 \tau_2$; !C=3 ; τ_2 ;

  !A=1 ; τ_1; τ_2 ; !C=3 ; $\leq$!A=1 ?A=1 ; τ_1 τ_2 ; !C=3 ; τ_2 ;

- Formal correctness criterion

> $\Sigma_1||\ldots||\Sigma_n$ is correct w.r.t. $\Sigma_1|\ldots|\Sigma_n$ if
> for all $s \in RSS(\Sigma_1|\ldots|\Sigma_n)$ and all $\phi \in Traces_{\Sigma 1||\ldots||\Sigma n}(s)$
> there exist $\alpha \in Traces_{\Sigma 1||\ldots||\Sigma n}(s)$ and $\beta \in Traces_{\Sigma 1|\ldots|\Sigma n}(s)$
> such that $\phi \leq \alpha$ and $\alpha \sim \beta$

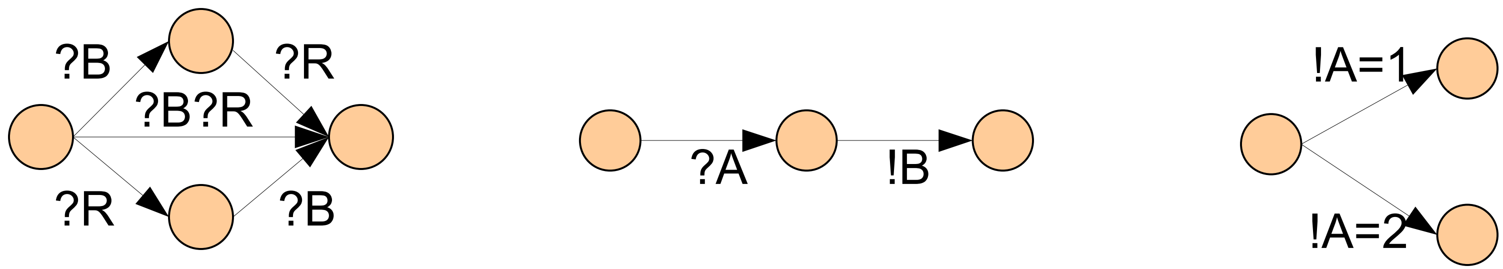
- Intuition: every trace of $\Sigma_1||\ldots||\Sigma_n$ can be completed to one that is equivalent to a synchronous trace

Microstep weak endochrony

- Compositional delay-insensitivity criterion (signal absence information is not needed)
- Axioms (part 1):

A1: Determinism

A2: In every state, non-clock transitions sharing no common variable are independent



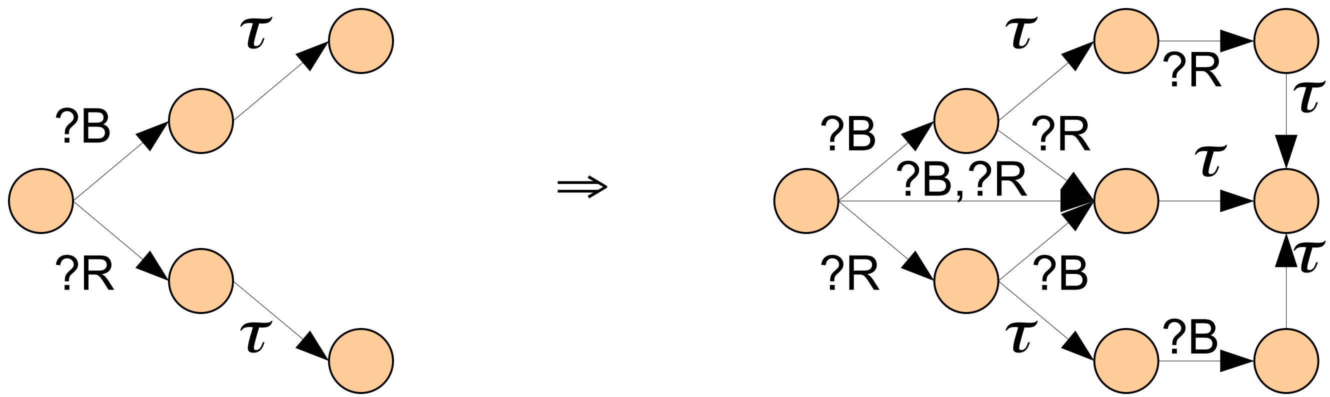
Microstep weak endochrony

- Axioms (continued):

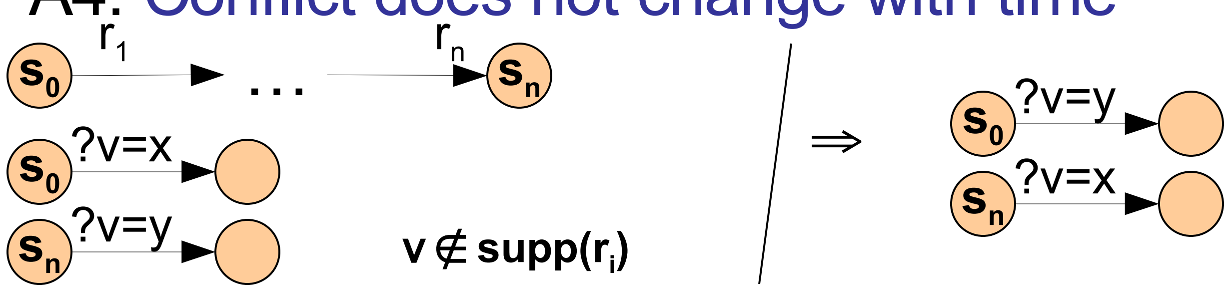
A1: Determinism

A2: In every state, non-clock transitions sharing no common variable are independent

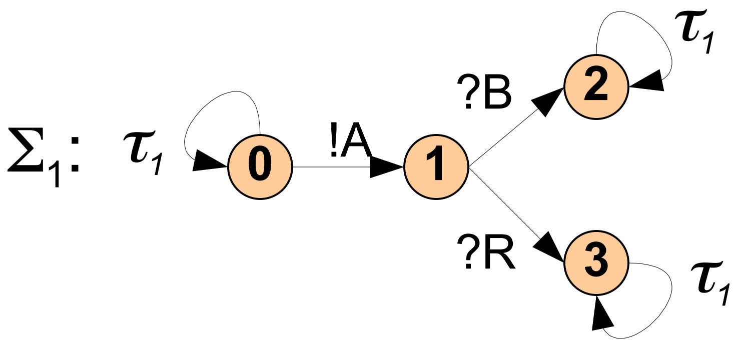
A3: Non-contradictory reactions can be united



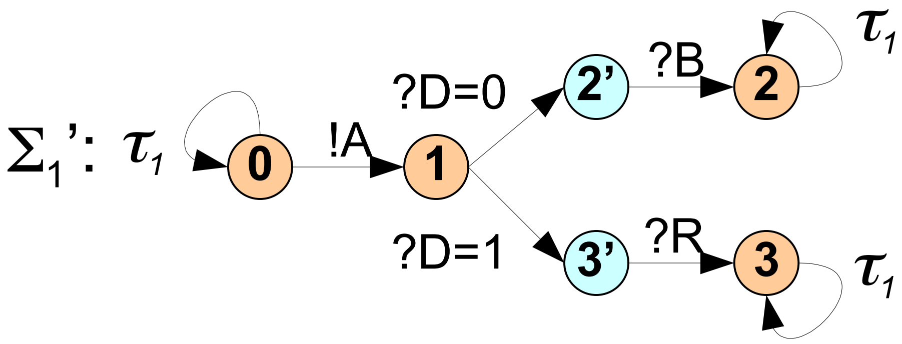
A4: Conflict does not change with time



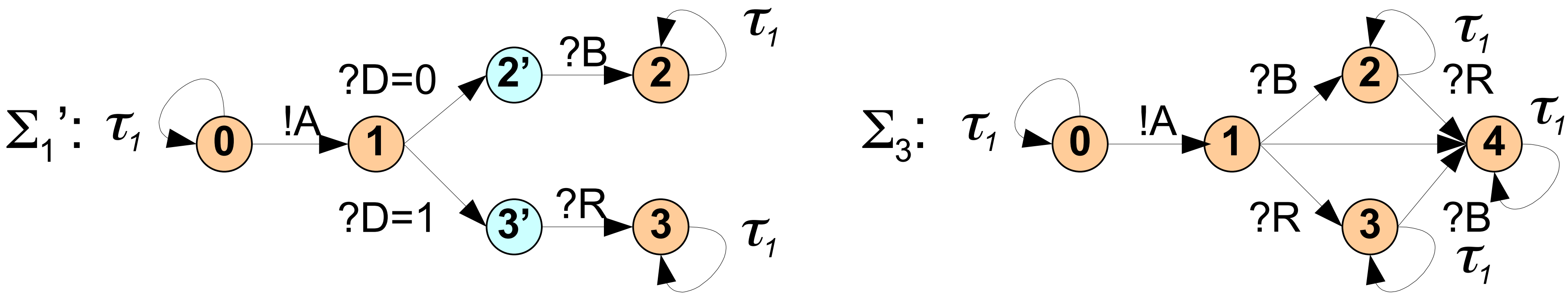
Example



Example



Example



Σ_1':
τ_1
0
!A
1
?D=0
?D=1
2'
3'
?B
?R
2
3
τ_1
τ_1
Σ_3:
τ_1
0
!A
1
?B
?R
2
3
4
τ_1
?R
τ_1
?B
τ_1

Weak non-blocking property

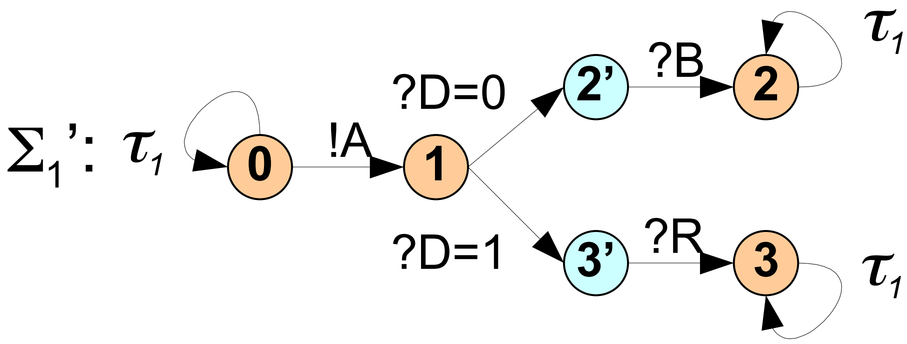
- Weak non-blocking

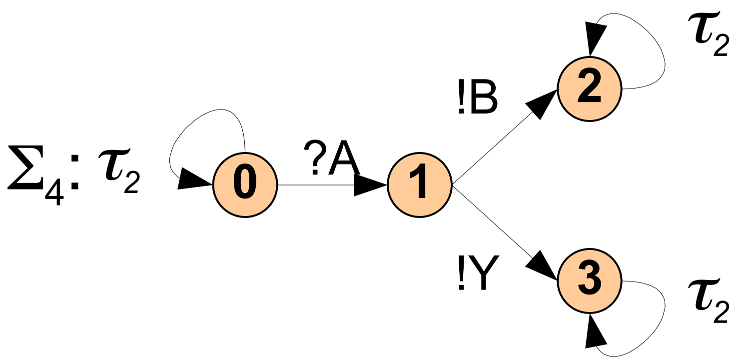
$\Sigma_1,\ldots,\Sigma_n$ are weakly non-blocking iff
for all $s \in RSS(\Sigma_1|\ldots|\Sigma_n)$ and all $\phi \in Traces_{\Sigma 1|\ldots|\Sigma n}(s)$
maximal and containing no clock transition, there exists
$\alpha \in Traces_{\Sigma 1|\ldots|\Sigma n}(s)$ non-void such that
$\alpha \preccurlyeq \phi$ and $\alpha;\tau \in Traces_{\Sigma 1|\ldots|\Sigma n}(s)$

- Semantics preservation criterion

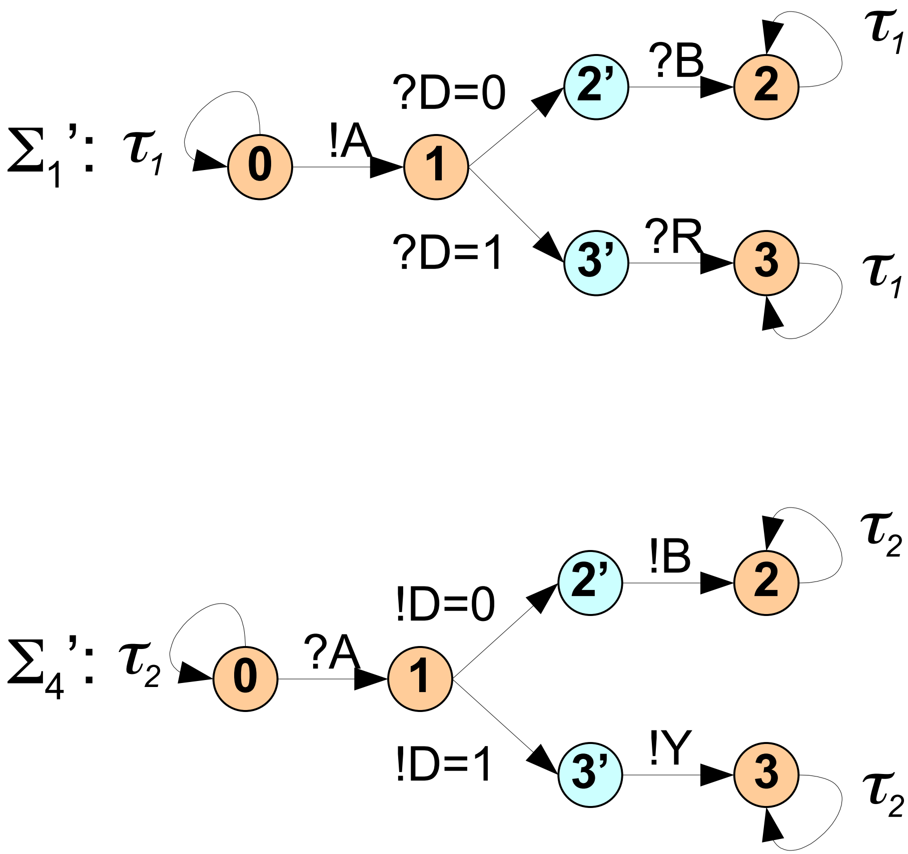
If $\Sigma_1,\ldots,\Sigma_n$ are weak non-blocking and weak
endochronous, then $\Sigma_1||\ldots||\Sigma_n$ is correct w.r.t. $\Sigma_1|\ldots|\Sigma_n$

Example


Σ_1':
τ_1
0
!A
1
?D=0
2'
?B
2
τ_1
?D=1
3'
?R
3
τ_1


Σ_4: τ_2
0
?A
1
!B
2
τ_2
!Y
3
τ_2

Example



Σ_1':
τ_1
0
!A
1
?D=0
?D=1
2'
?B
2
τ_1
3'
?R
3
τ_1
Σ_4':
τ_2
0
?A
1
!D=0
!D=1
2'
!B
2
τ_2
3'
!Y
3
τ_2

Conclusion

- Decidable criteria for GALS implementation of synchronous specifications
  - Covers causality and read/write communication
  - Compositionality, concurrency
- Future: Synthesis
  - Make synchronous automata weakly endo/isochronous. Optimality issues.
  - Heuristics for actual synchronous languages and specifications. Scaling issues (large specifications).
  - GALS circuits using asynchronous logic
  - Deal with mode changing latency
- What about timed models ?