Verification of Hybrid Controlled Processing Systems based on Decomposition and Deduction

Goran Frehse · Olaf Stursberg · Sebastian Engell
Process Control Laboratory, University of Dortmund, Germany

Ralf Huuck · Ben Lukoschus*
Chair of Software Technology, University of Kiel, Germany
*visiting SRI International, Menlo Park, CA, USA

ISIC 2001 · Mexico City · September 5–7, 2001

Introduction and Motivation

Given: **hybrid process** $\leftrightarrow$ **distributed controller**

Need: proof of a **global property** of this system

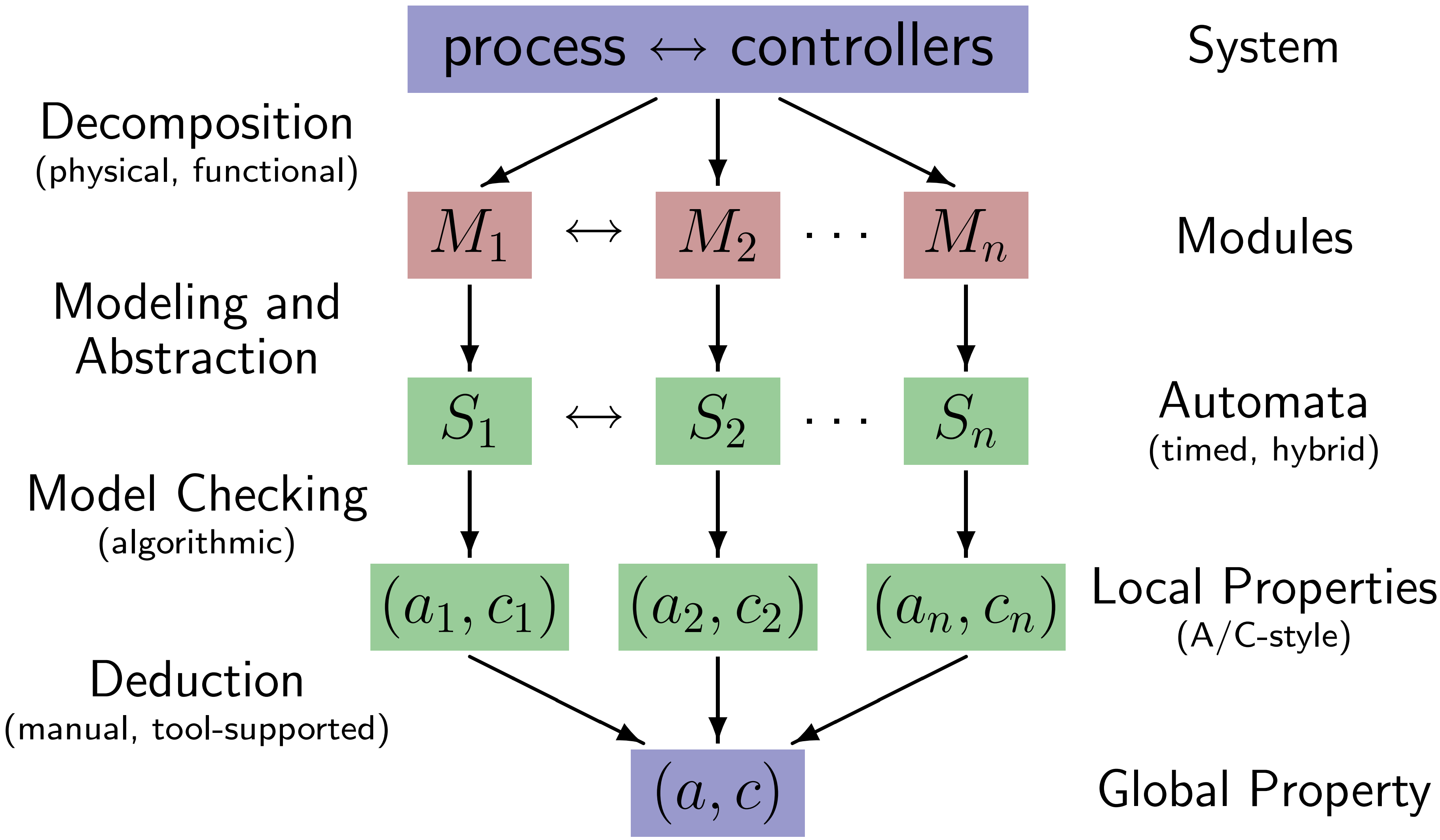
Problem: if the system is

- of **high complexity** and
- involves **parallel** and **hierarchical** structures,

verification is difficult.

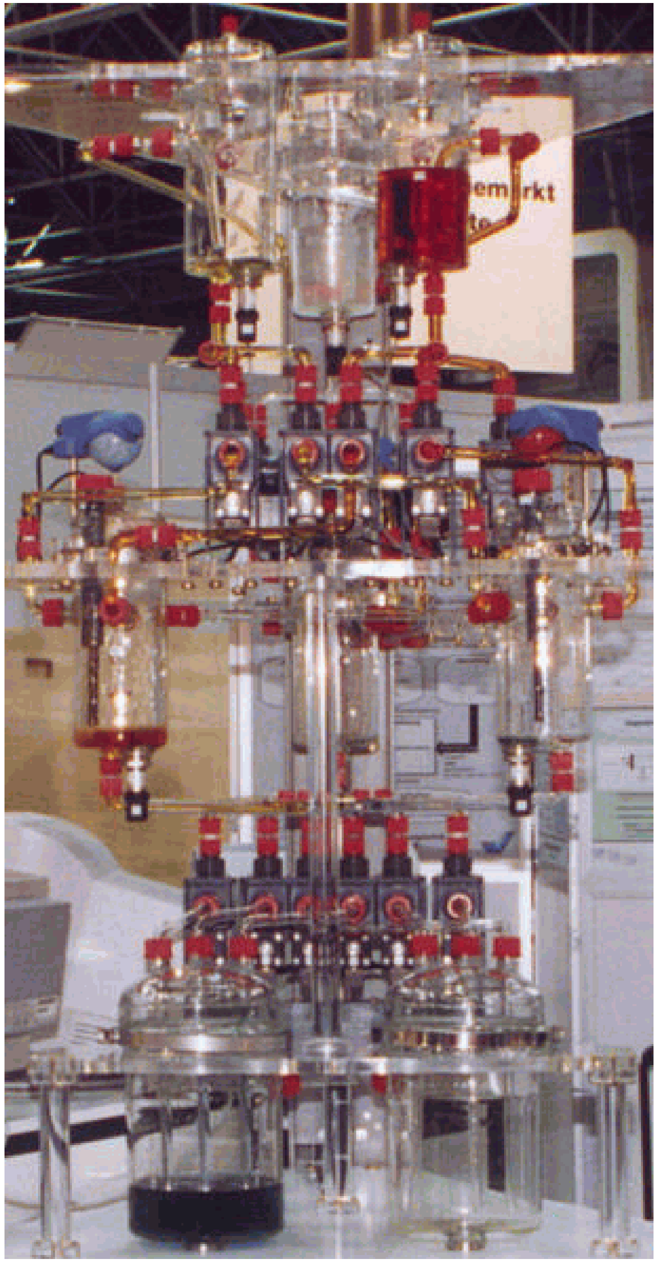
Basic idea: **"divide and conquer"**

The Approach

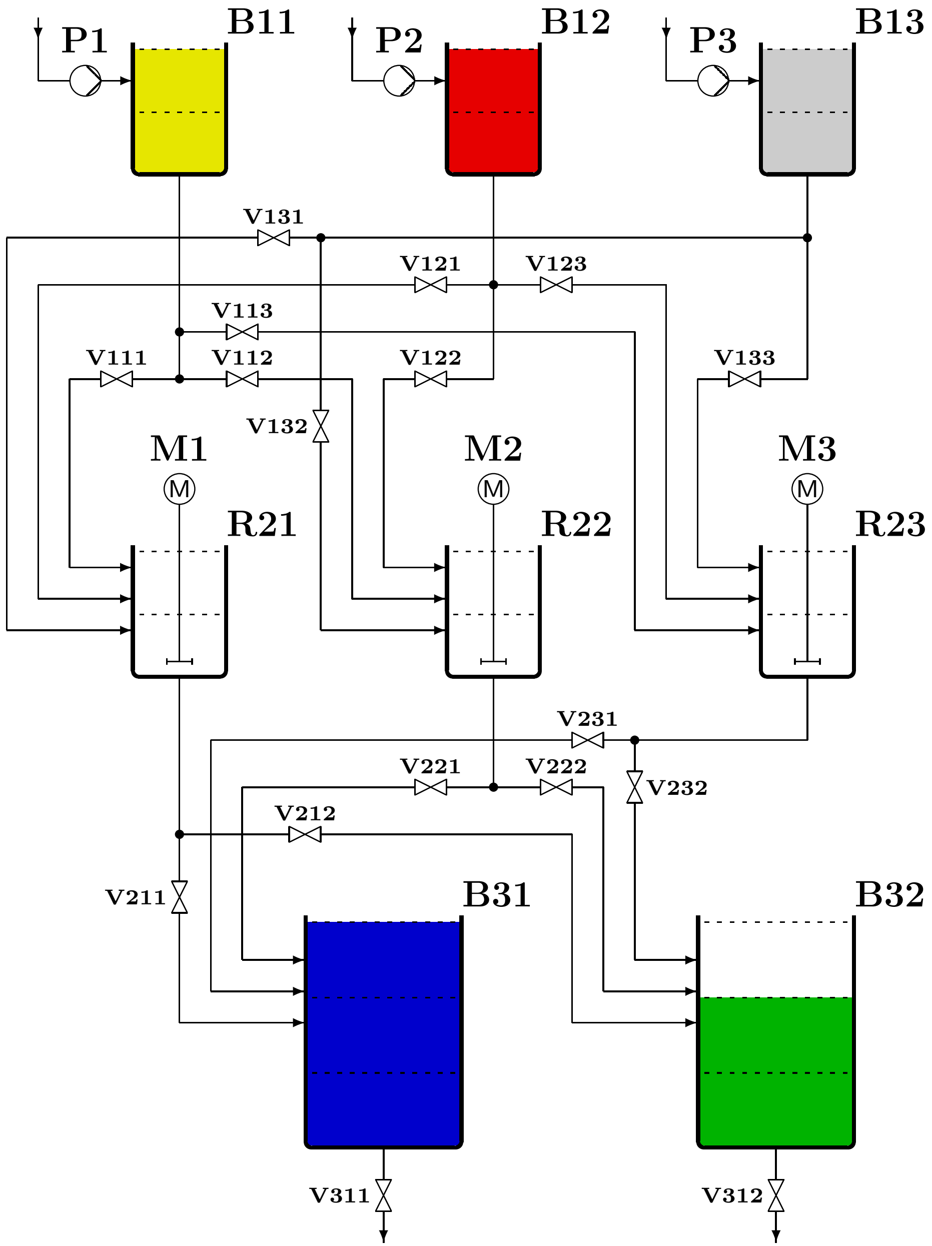


Example: A Multi-Product Batch Plant

- located at: Process Control Lab, University of Dortmund (Germany)
- chemical batch production process
- used for teaching:
  - process control
  - PLC programming
- case study in research projects:
  - modeling
  - formal verification
  - scheduling

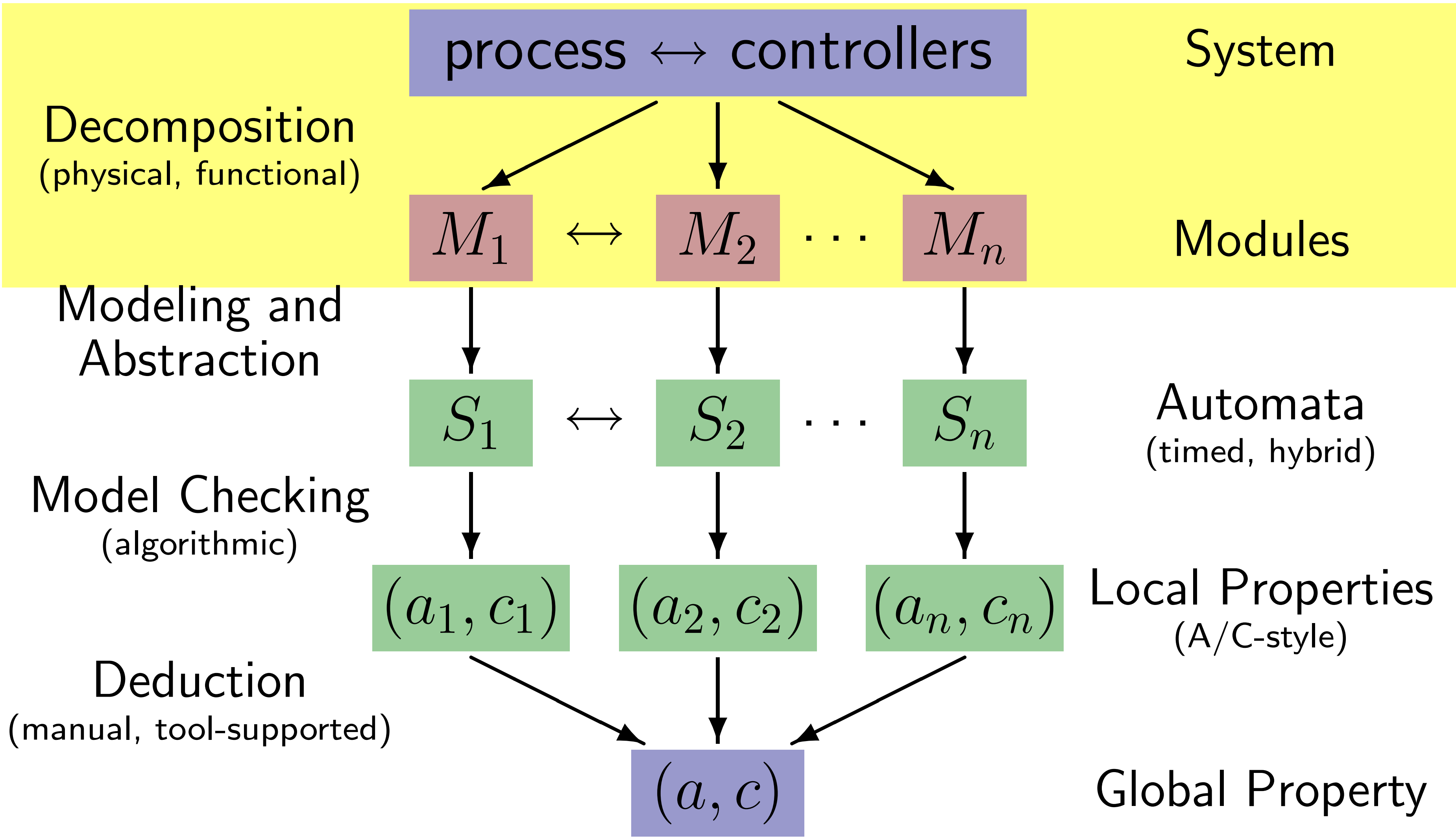


Example: A Multi-Product Batch Plant

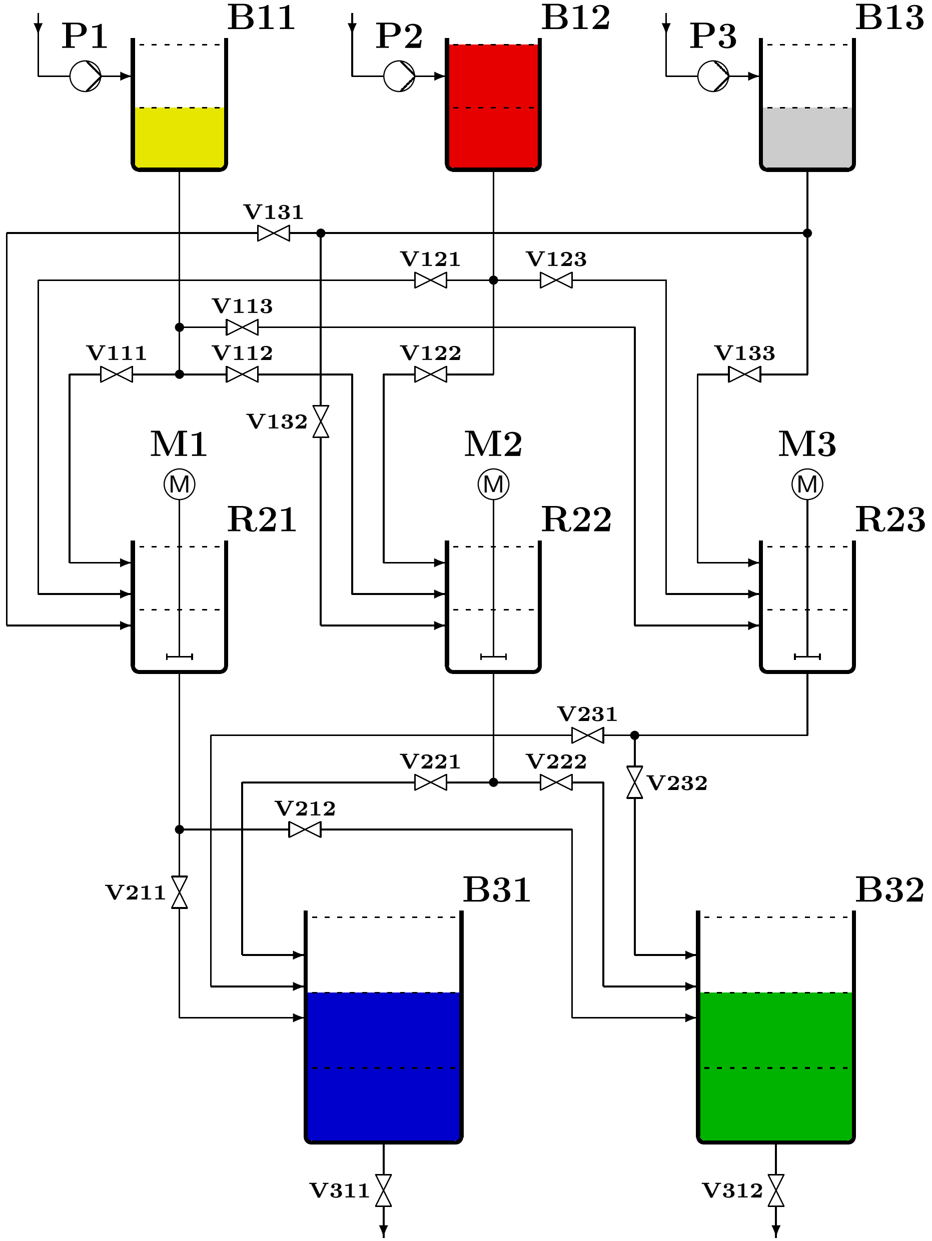


- 2 products:
  blue, green
- 3 basic substances:
  yellow, red, white
- 3 reactors for production of blue, green
- PLC-based distributed control system

Decomposition

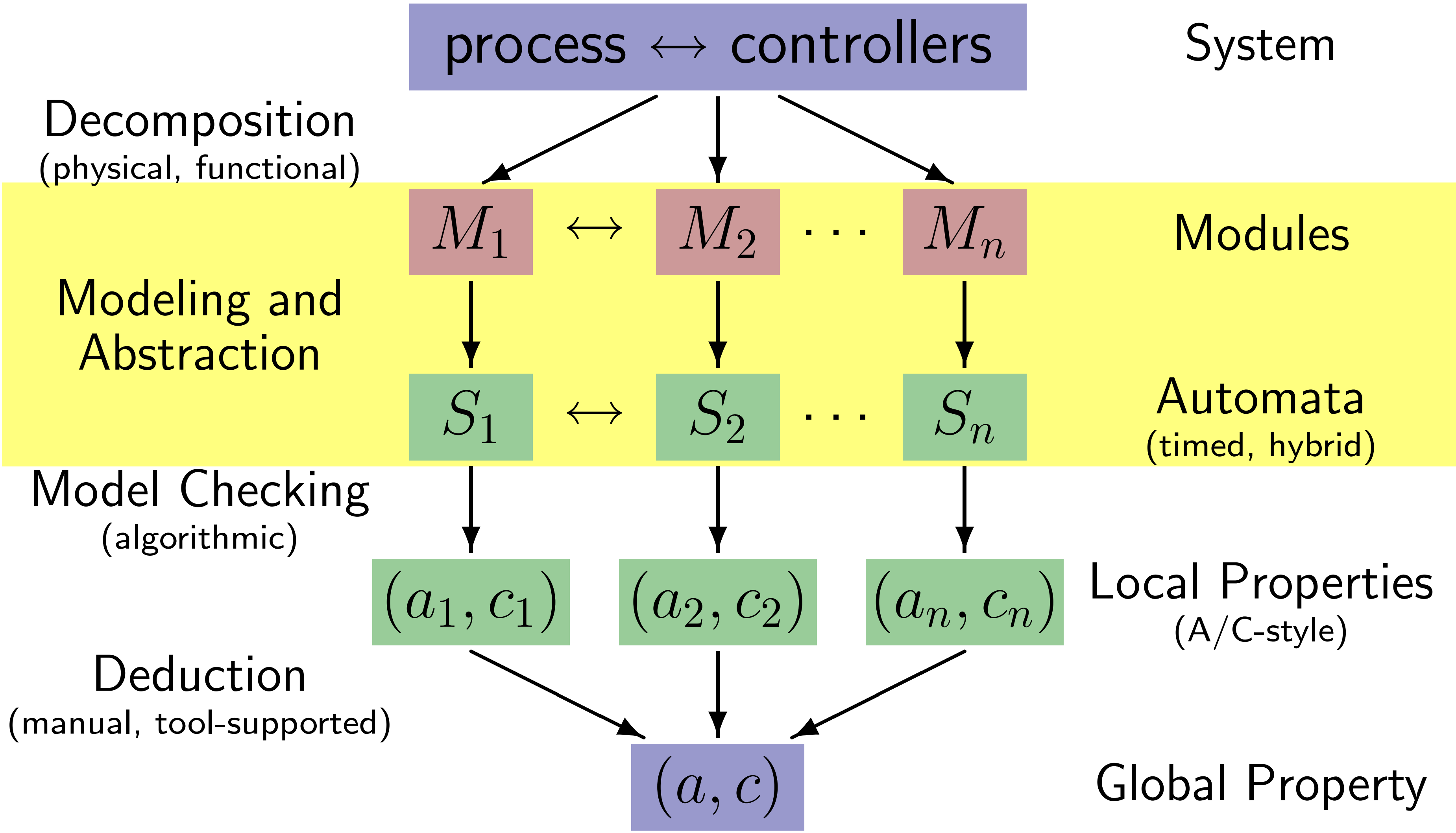


Decomposition



- Plant Hardware
  - tanks, pumps
  - reactors, mixers
  - valves, pipes
  - sensors
- Control Software
  - raw material delivery
  - production
  - resource management
  - emergency shutdown, maintenance, . . .

Modeling and Abstraction



Modeling and Abstraction

Modeling framework:

communicating linear hybrid automata (CLHA)

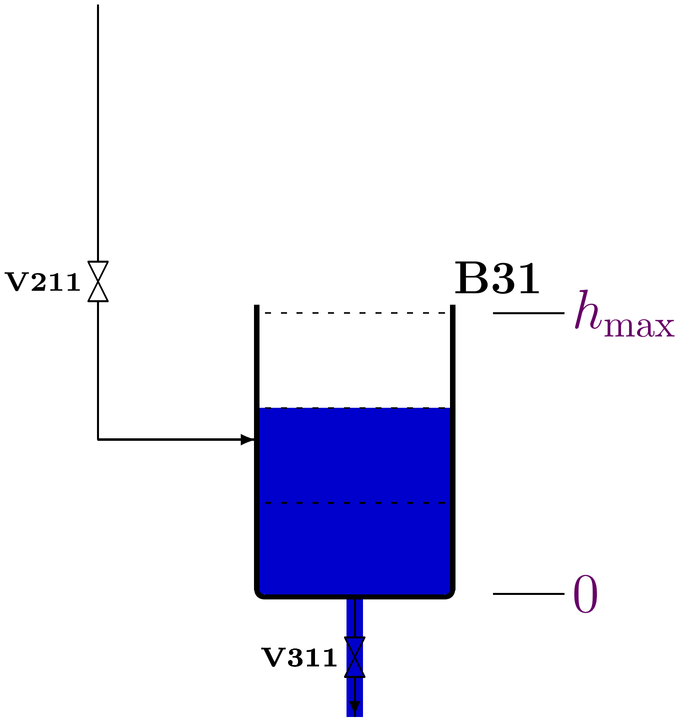
CLHA are LHA with

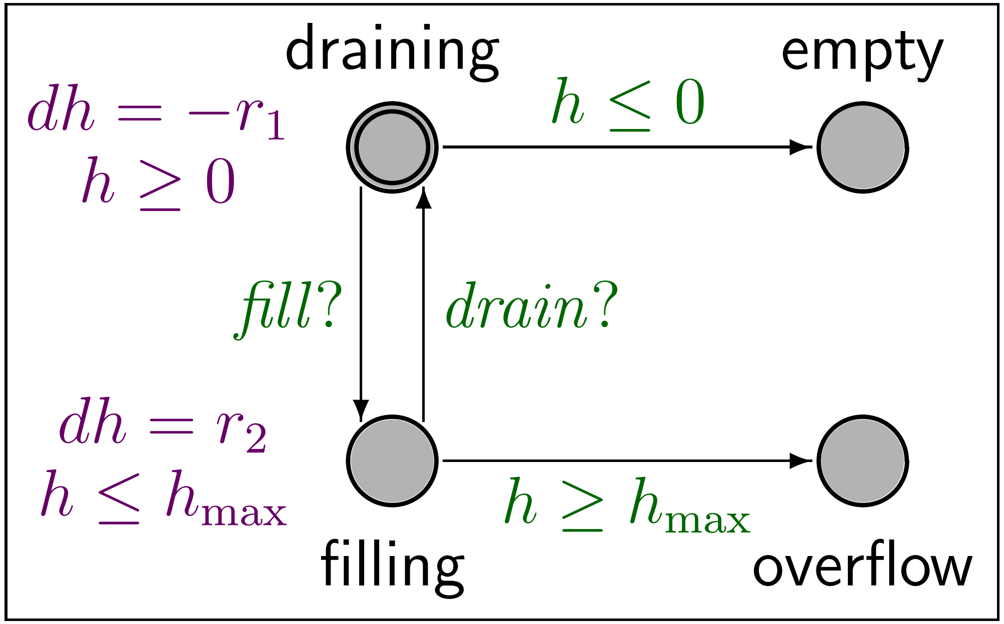
- continuous input/output variables
- labels for directed and undirected communication:
    - *send*
    - *receive*
    - *synchronization*

Modeling and Abstraction

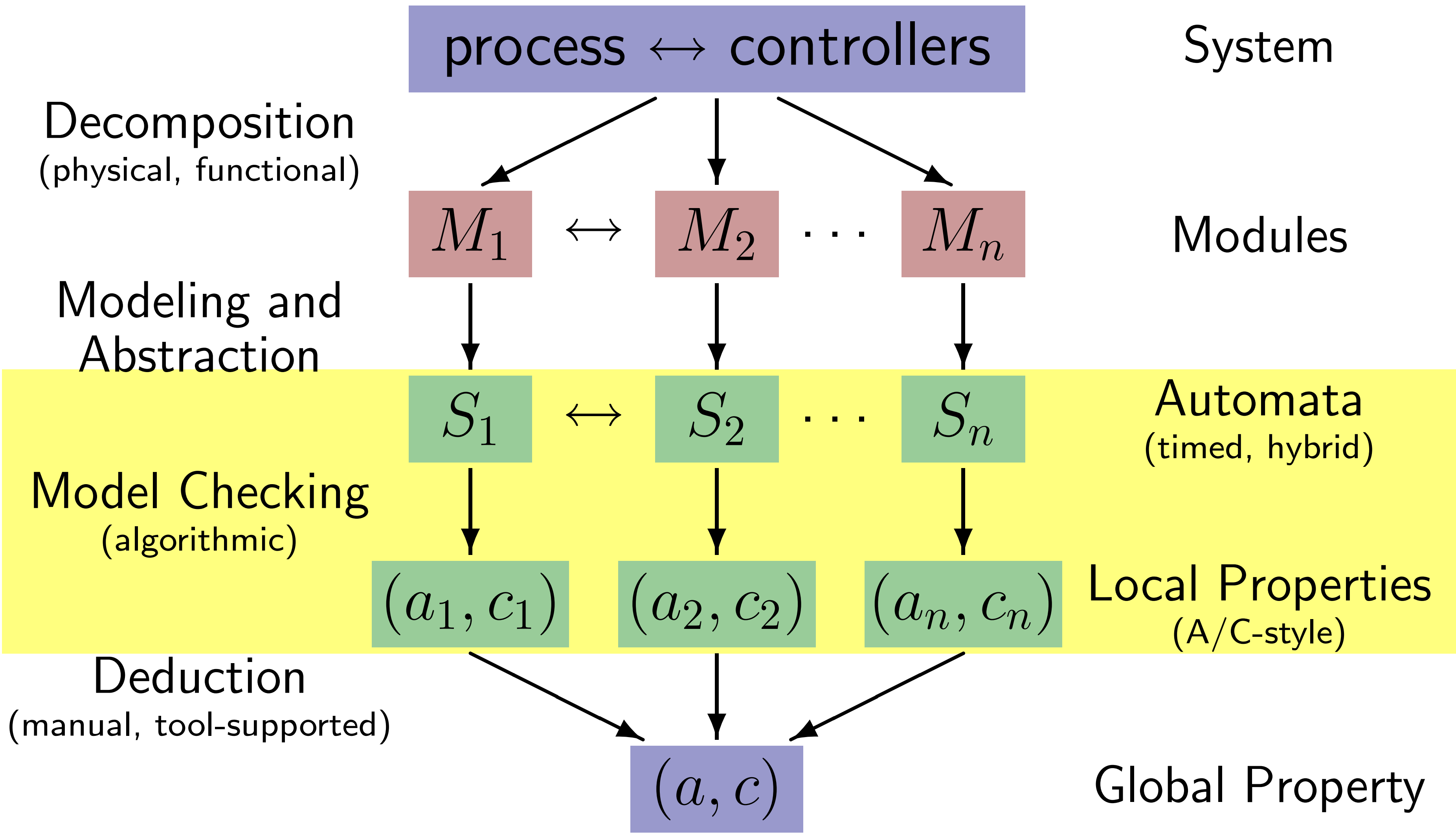
CLHA model of Tank $\mathrm{B31}$

- draining $(\mathrm{V211}$ closed$)$: level sinks with rate $r_1 = 1\,\mathrm{cm\,s^{-1}}$
- filling $(\mathrm{V211}$ open$)$: level rises with rate $r_2 = 2\,\mathrm{cm\,s^{-1}}$
- desired level: $0 < h < h_{\max}$





Model Checking



The Assumption/Commitment (A/C) paradigm

assumption a expected behavior of the environment

commitment c guaranteed behavior of the module

The Semantics of an A/C Formula (a, c)

$S \models (a, c) \iff$ "if the environment of module S fulfills a, then module S fulfills c"

Example: A/C Property of Tank $\mathrm{B31}$

a "*fill*" happens before $h \leq 0$ and "*drain*" before $h \geq h_{\mathrm{max}}$

c Tank $\mathrm{B31}$ does not run empty and does not overflow

Model Checking

Verifying $\mathrm{B31} \models (a, c)$

Model checkers usually do not support A/C directly, but:

- a can be expressed as another automaton A (sending "fill" and "drain" at the right time)
- c can be expressed as the reachability property "the states empty and overflow are never reached"

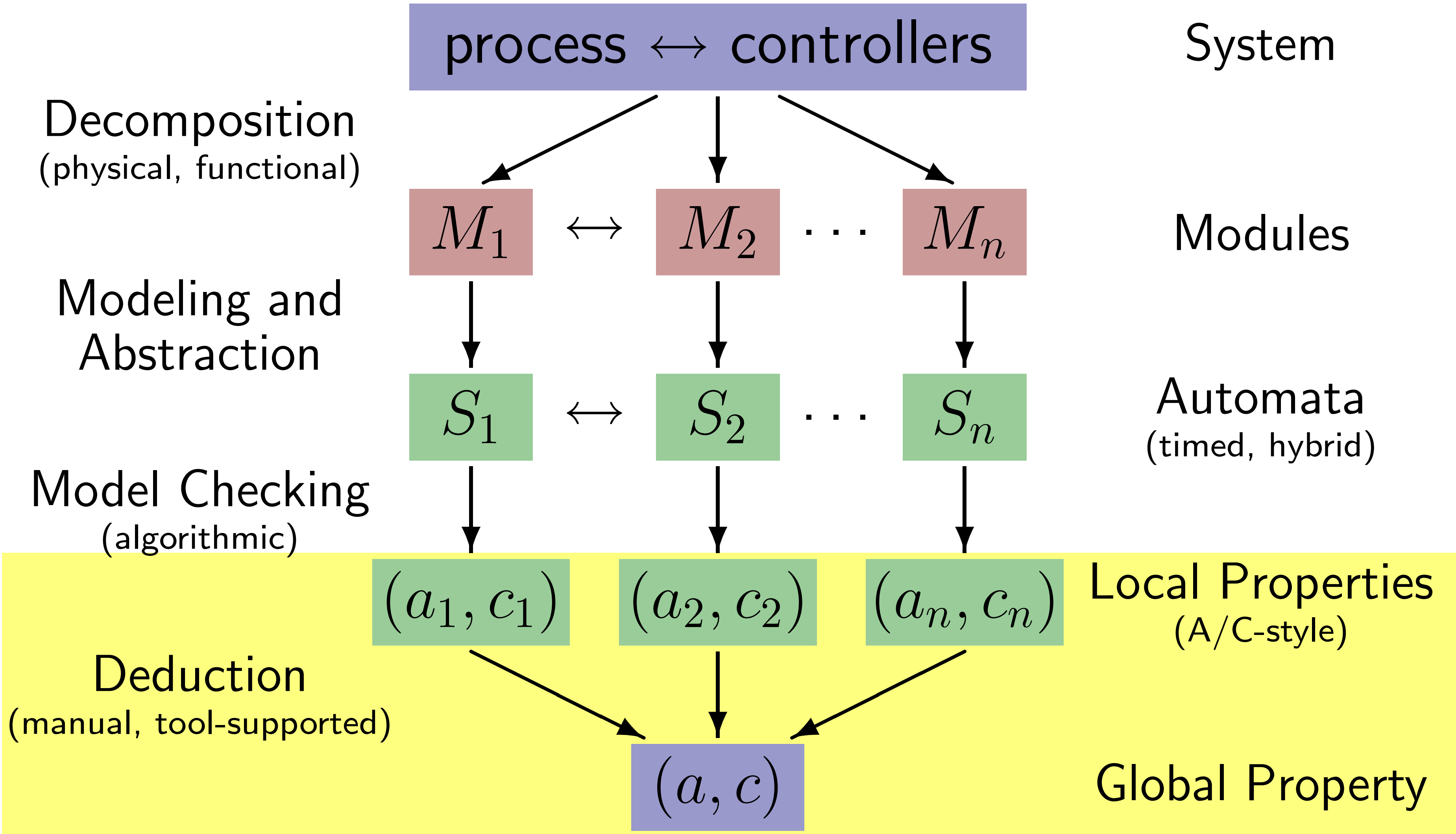
Now use a hybrid model checker to show

$$\mathrm{B31}||A \models \neg reach(\text{empty}) \wedge \neg reach(\text{overflow})$$

A is much smaller than the full environment of $\mathrm{B31}$

$\Rightarrow$ **model checking becomes feasible**

Deduction


process ↔ controllers
System
Decomposition
(physical, functional)
M_1 ↔ M_2 ... M_n
Modules
Modeling and
Abstraction
S_1 ↔ S_2 ... S_n
Automata
(timed, hybrid)
Model Checking
(algorithmic)
(a_1, c_1) (a_2, c_2) (a_n, c_n)
Local Properties
(A/C-style)
Deduction
(manual, tool-supported)
(a, c)
Global Property

Deduction

Given

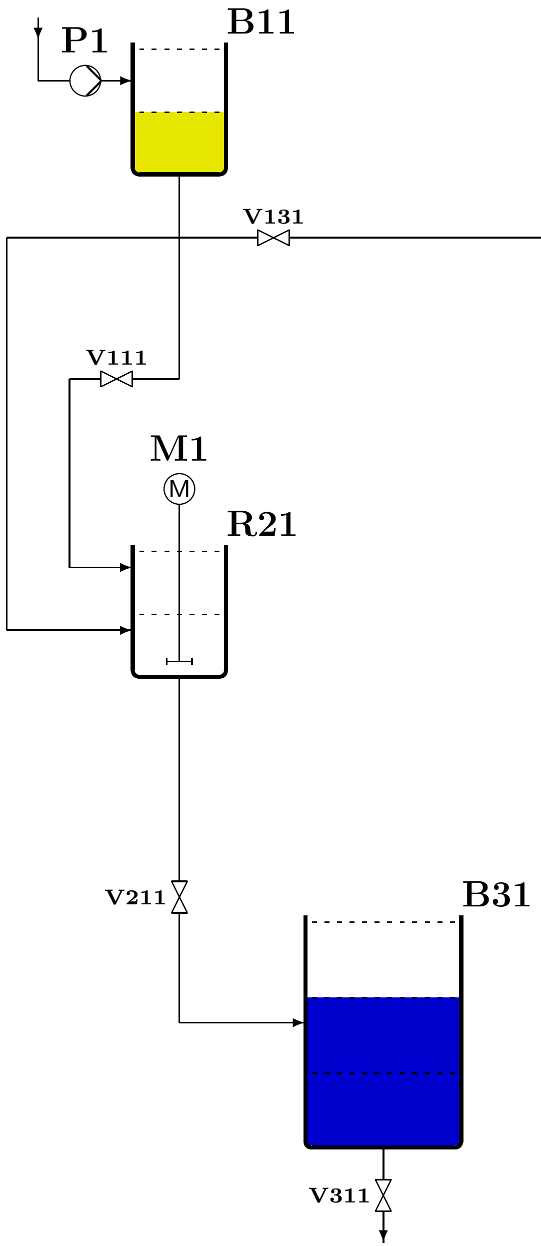
- the **local properties** $S_1 \models (a_1, c_1)$, ..., $S_n \models (a_n, c_n)$
- additional conditions B

we use **deductive analysis** to derive

- a **global property** (a, c) of the system.

A theorem prover (e.g., PVS) can be used to support the analysis.

Deduction



a_{B11}	$\mathrm{P1}$ can deliver yellow to $\mathrm{B11}$
c_{B11}	yellow is available for $\mathrm{R21}$
a_{B13}	$\mathrm{P3}$ can deliver white to $\mathrm{B13}$
c_{B13}	white is available for $\mathrm{R21}$
a_{R21}	yellow and white are available for $\mathrm{R21}$
c_{R21}	$\mathrm{R21}$ contains blue in time
a_{Ctrl}	$\mathrm{R21}$ contains blue in time
c_{Ctrl}	"*fill*" happens before $h \leq 0$
	...
a	$\mathrm{P1}$ and $\mathrm{P3}$ can deliver raw materials
c	$\mathrm{B31}$ does not run empty

Computation Results

Verifying a part of the multi-product batch plant

Method	Memory	Time
conventional	70 MB	600 sec.
A/C (17 specs)	$17\times$ <1 MB	$17\times$ <10 sec.

Related Work

- HUNGAR (1993)
  A/C and data abstraction for CSP programs
- DINGEL, FILKORN (1995)
  A/C and data abstraction for infinite state systems
- XU, SWARUP (1998)
  A/C in Hoare logic and duration calculus
- DE ALFARO, ALUR, GROSU, HENZINGER, KANG (2000)
  A/G and refinement for reactive modules
- HENZINGER, MINEA, PRABHU (2001)
  A/G for hierarchical hybrid systems
- AMLA, EMERSON, NAMJOSHI, TREFLER (2001)
  A/G for synchronous transition diagrams
- SHANKAR (2000) The SAL framework