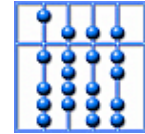


---

## **Specification of Distributed Systems**

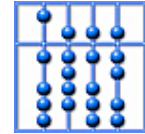
Dr. Bernhard Schätz  
Leopold-Franzens Universität Innsbruck  
Sommersemester 2005



## Overview

---

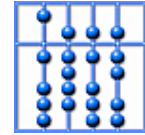
1. Introduction
2. Basics: Behavior, Interaction, Concurrency
3. Coroutines
4. Communicating Processes
5. Data Flow Models
6. State-Based Models
7. Coordination
8. Executions
9. Property Descriptions



# Overview

---

1. Introduction
2. Basics: Behavior, Interaction
  1. Modeling Computation: State Transition Systems
  2. Modeling Interaction: Labeled Transition Systems
  3. Modeling Concurrency: Synchronized Transition Systems
  4. Modeling Behavior: Streams of Observations
  5. Modeling Communication: Synchronized Behaviors
3. Coroutines
4. Communicating Processes
5. Data Flow Models
6. State-Based Models
7. Coordination
8. Executions
9. Property Descriptions



## Specifications: Concepts, Models, Notations

---

Specification (n.): A detailed listing or description of the required properties of some object proposed to be built or bought;

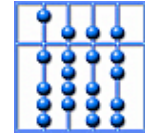
*Collaborative International Dictionary.*

To describe properties, we need:

**Concept:** Building blocks for properties, independent of a specific model

**Model:** (Mathematical) formalism, used to define concepts

**Notation:** Description technique, used to define instances of a model

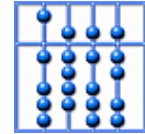


## Models

---

### Classes of models:

- **Operational model:** Specification in form of an abstract machine
  - Concepts: State, (labeled) transition, transition system
  - Verification: (Bi-)Simulation
  - Example: (Timed) Automata
- **Denotational model:** Specification in form of observable behavior
  - Concepts: Interface, event, (observation) trace,
  - Verification: Behavior Inclusion
  - Example: Trace-based models
- **Algebraic model:** Specification in form of syntactic formulae
  - Concepts: Process, action, choice,
  - Verification: Term equivalence
  - Example: (T)CSP



## 2.3 Modeling Behavior: Streams of Observations

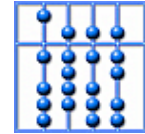
---

**Goal:** Define a model to describe the behavior of communicating reactive systems

- Relevant: Address aspects evolving with communication
- General: Cover aspects independent of specific computation techniques
- Abstract: Ignore aspects like communication bandwidth, execution speed

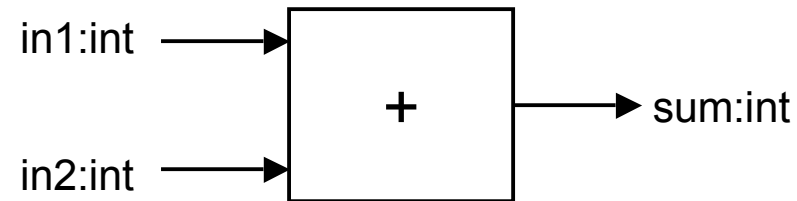
**Concept:** Interface, communication, signal/message, observation, behavior

**Model:** Sets of streams



## Concept: Interface

---

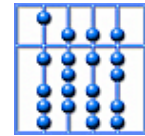


Purpose: Distinguish between system and environment

Concept: Interface, port

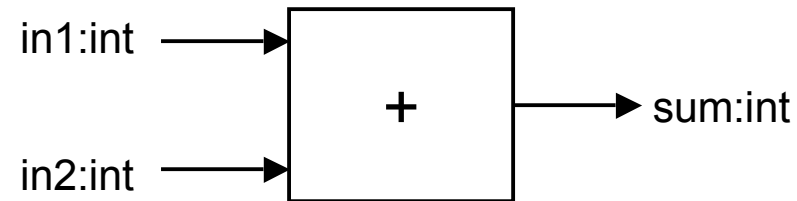
- Interface: Shared part of system and environment
- Port: Part of interface allowing data flow between system and environment

Example: Ports: in1, in2, out



## Concept: Input and Output

---



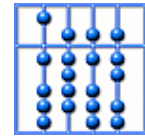
Purpose: Describing directed data flow

Concept: Input and output ( $In_1, \dots, In_m$  and  $Out_1, \dots, Out_n$ )

- Input signal: Signal controlled by environment
- Output signal: Signal controlled by system

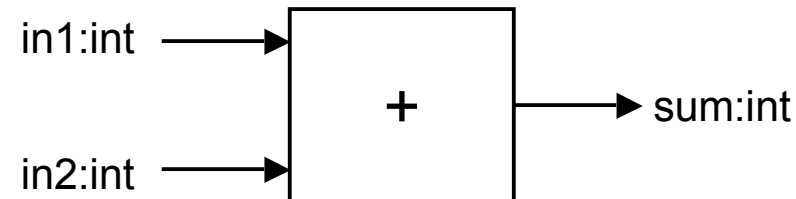
Example:

- Input ports *In*: in1(= int), in2 (= int)
- Output ports *Out*: sum (= int)



## Concept: Signal/Message

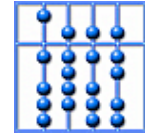
---



Purpose: Describing flow of information between system and environment

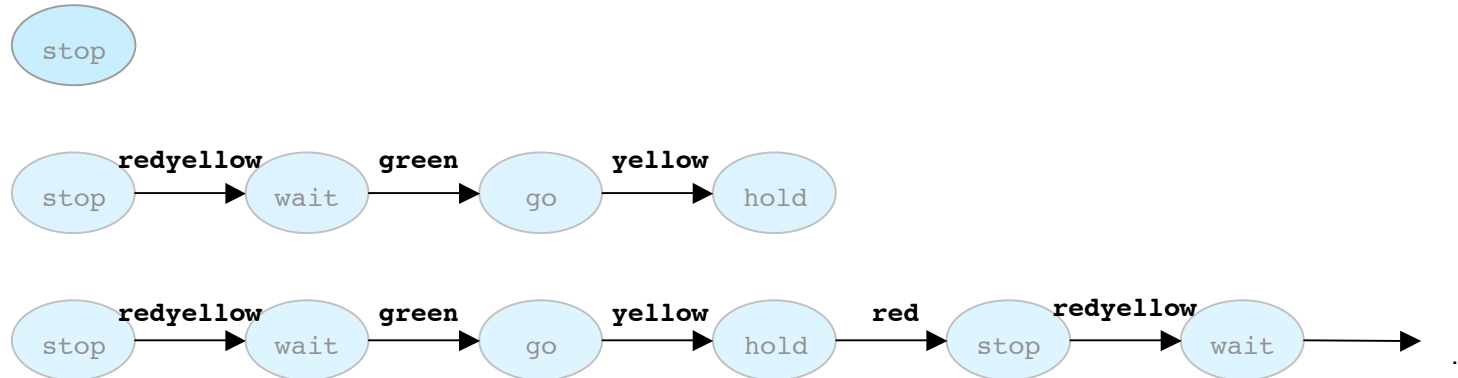
Concept: Signal vs. Message

- Signal: Time-based description of flow of information
  - Which signal is present at a given time
  - Detection of absence of a (defined) value
- Message: Event-based description of flow of information
  - Which message is sent/received
  - No detection of absence of a message



## Concept: Stream

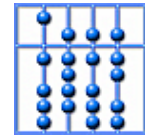
---



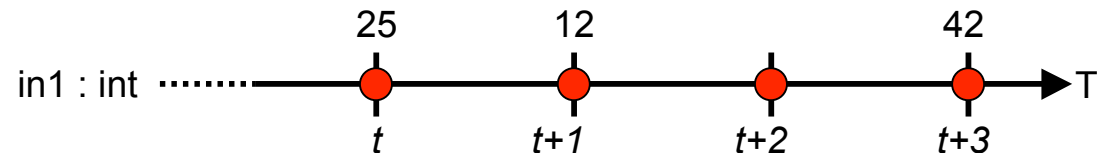
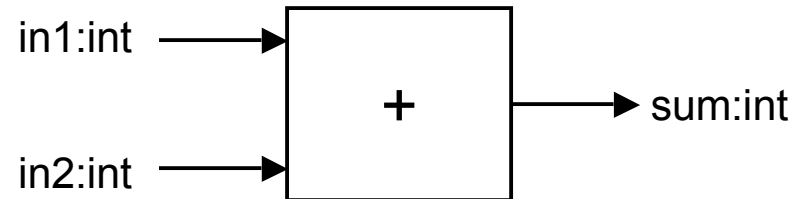
Purpose: Describing finite and infinite sequences of observations

Concept: Stream = Finite or infinite sequence of data  $D^\omega = D^* \cup D^\infty$

- Constructors: Building sequences over  $D$ 
  - Empty sequence:  $\langle \rangle$
  - Concatenation:  $d \cdot ds$  (or  $ds_1 \cdot ds_2$ )
  - Examples:  $25 \cdot 12 \cdot 42$ ,  $25 \cdot 12 \cdot 42 \cdot \langle \rangle$
- Ordering: Relating sequences over  $D$ 
  - Prefix-order:  $ds_1 \leq ds_2$
  - Identity:  $ds_1 = ds_2$  ( $ds_1 \leq ds_2$  and  $ds_2 \leq ds_1$ )
  - Examples:  $\langle \rangle \leq \langle \rangle$ ,  $25 \leq 25 \cdot 12 \cdot 42$ ,  $ds_1 \leq ds_1 \cdot ds_2$



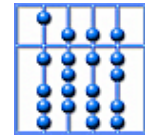
## Concept: Trace



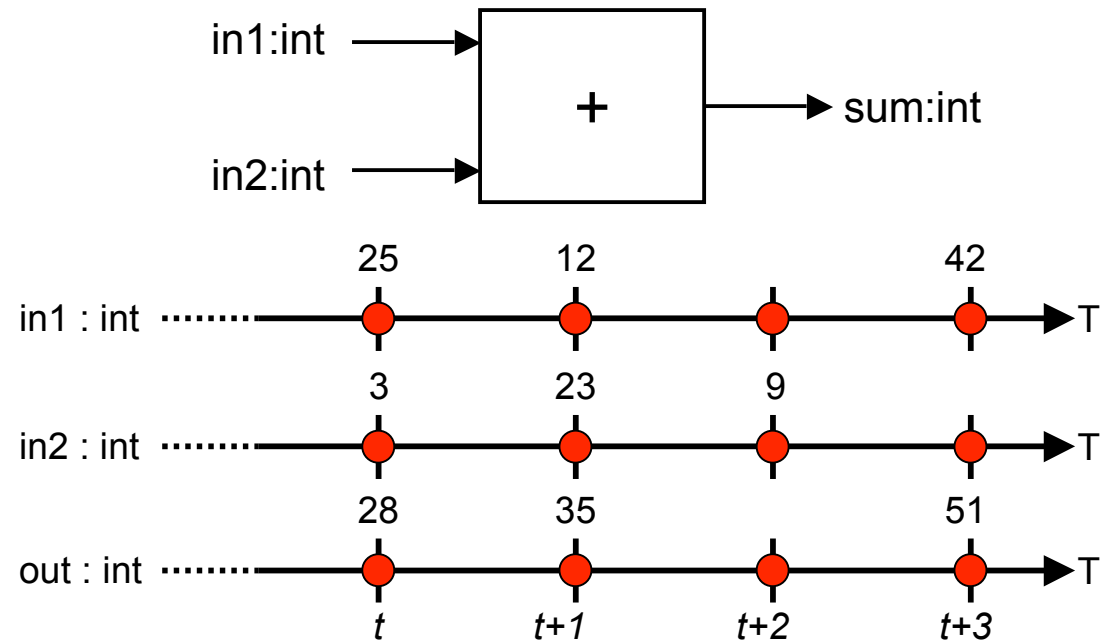
Purpose: Describing flow of signal via interface of a system

Concept: Trace = (finite or infinite) sequence of consecutive observations

- Signal trace: Time-based observation of flow of information  
Clocked Observation of port in1: ... • 25 • 12 • - • 42 • ...  $\in (\text{int} \cup \{-\})^\omega$
- Message trace: Event-based observation of flow of information  
Untimed Observation of port in1: ... • 25 • 12 • 42 • ...  $\in \text{int}^\omega$



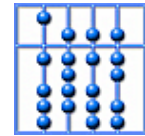
## Concept: Clocked Observation Trace



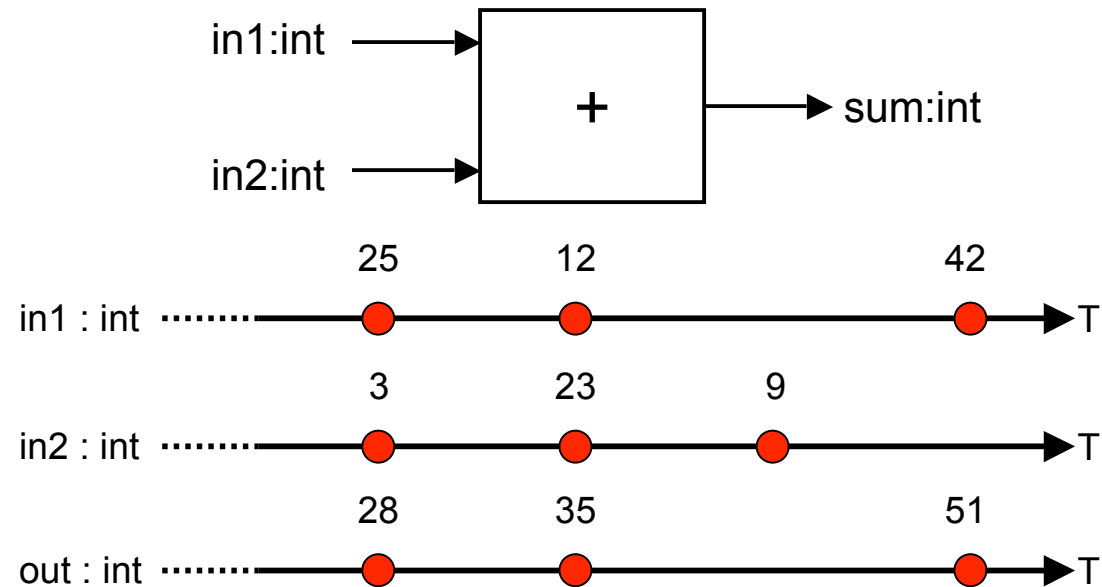
Purpose: Modeling a single (partial) execution of a system visible at its interface

Concept: Clocked observation trace

- Clocked Observation for all interface ports of a system:  
 $( \dots \cdot 25 \cdot 12 \cdot - \cdot 42, \dots \cdot 3 \cdot 23 \cdot 9 \cdot - , \dots \cdot 25 \cdot 12 \cdot - \cdot 51 ) \in (\text{int} \cup \{-\})^\omega \times (\text{int} \cup \{-\})^\omega \times (\text{int} \cup \{-\})^\omega$



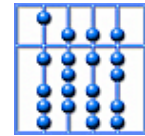
## Concept: Untimed Structured Observation Trace



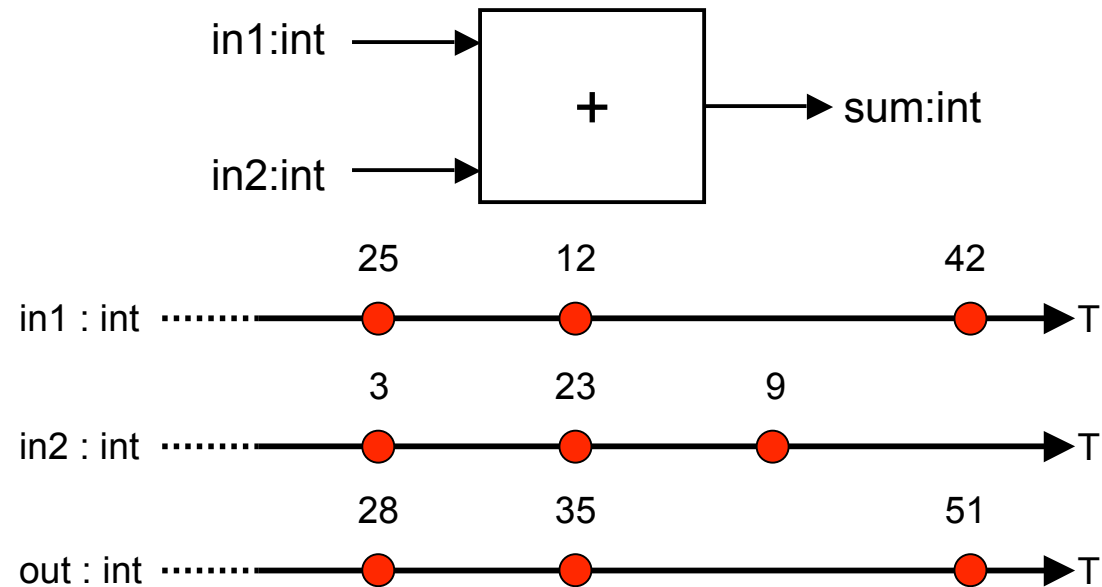
Purpose: Modeling a single (partial) execution of a system visible at its (structured) interface

Concept: Untimed observation trace

- Untimed Observation for all interface ports of a system:  
( ... • 25 • 12 • 42, ... • 3 • 23 • 9, ... • 25 • 12 • 51)  $\in \text{int}^\omega \times \text{int}^\omega \times \text{int}^\omega$



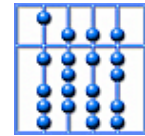
## Concept: Untimed Observation Trace



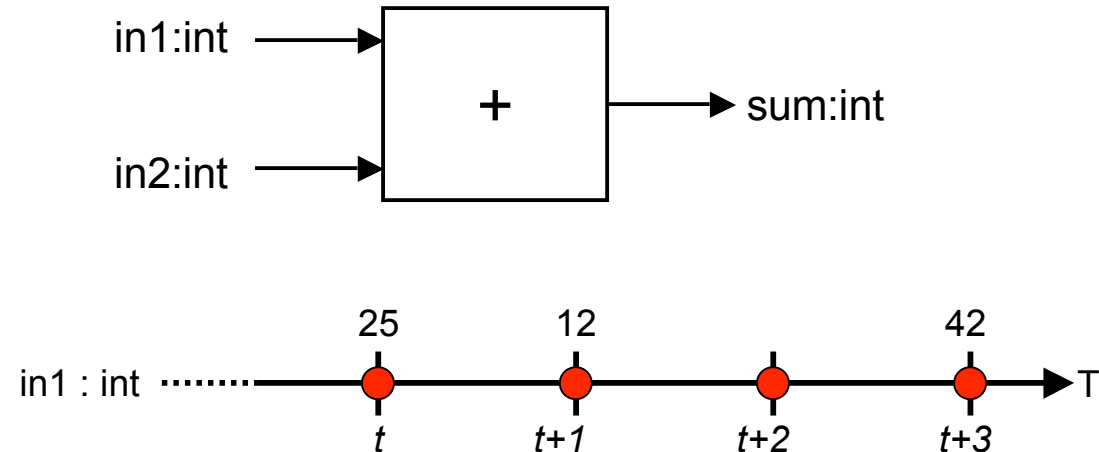
Purpose: Modeling a single (partial) execution of a system visible at its (unstructured) interface

Concept: Untimed observation trace

- Untimed Observation for all interface ports of a system:  
 $( \dots \cdot (in1,25) \cdot (in2,12) \cdot (in2,3) \cdot (in2,23) \cdot (out,28) \cdot (in2,9) \cdot (in1,12) ) \in ((In \cup Out) \times int)^\omega$



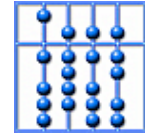
## Concept: Complete Trace



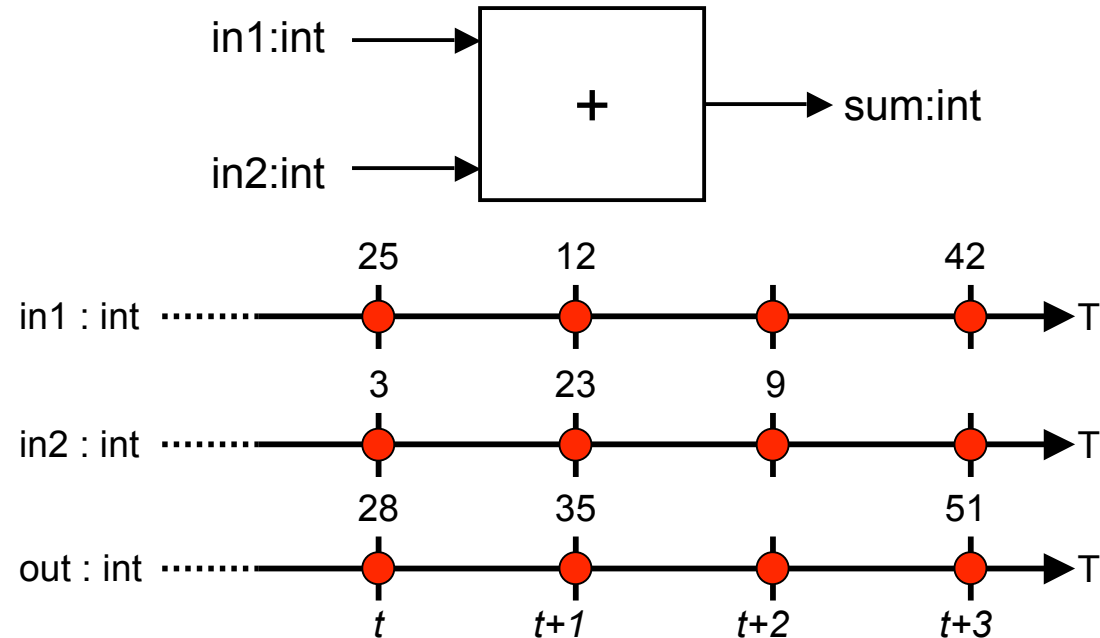
Purpose: Modeling a single **complete** execution of a system visible at its interface

Concept: Complete observation trace

- Complete trace: System has reached a stable state
- Interpretation:
  - Timed Observation Trace: All time steps (= infinite sequence)
  - Untimed Observation Trace: All outputs occurred (= possible finite sequence)



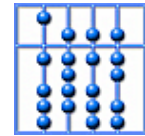
## Concept: Complete Clocked Observation Trace



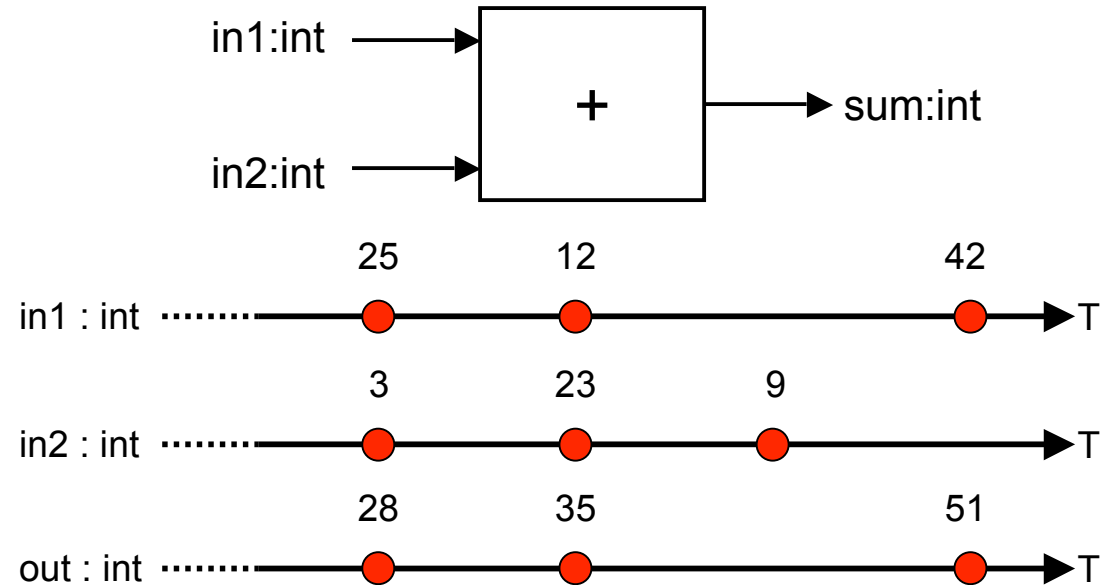
Purpose: Modeling a single complete execution of a system visible at its interface

Concept: Complete clocked observation trace

- Clocked Observation for all interface ports of a system:  
 $( \dots \cdot 25 \cdot 12 \cdot - \cdot 42 \cdot \dots, \dots \cdot 3 \cdot 23 \cdot 9 \cdot - \cdot \dots, \dots \cdot 25 \cdot 12 \cdot - \cdot 51 \cdot \dots ) \in (\text{int} \cup \{-\})^\infty \times (\text{int} \cup \{-\})^\infty \times (\text{int} \cup \{-\})^\infty$



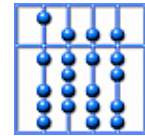
## Concept: Complete Untimed Structured Observation Trace



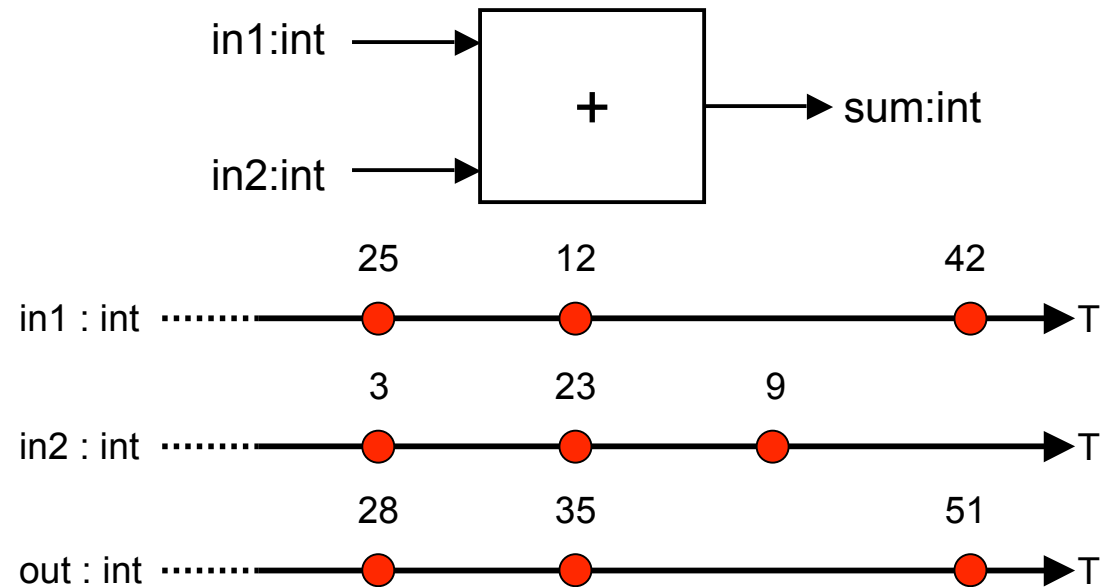
Purpose: Modeling a single complete execution of a system visible at its (structured) interface

Concept: Complete untimed structured observation trace

- Untimed Observation for all interface ports of a system:  
( ... • 25 • 12 • 42, ... • 3 • 23 • 9, ... • 25 • 12 • 51)  $\in \text{int}^\omega \times \text{int}^\omega \times \text{int}^\omega$



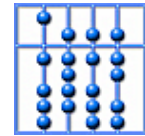
## Concept: Complete Untimed Observation Trace



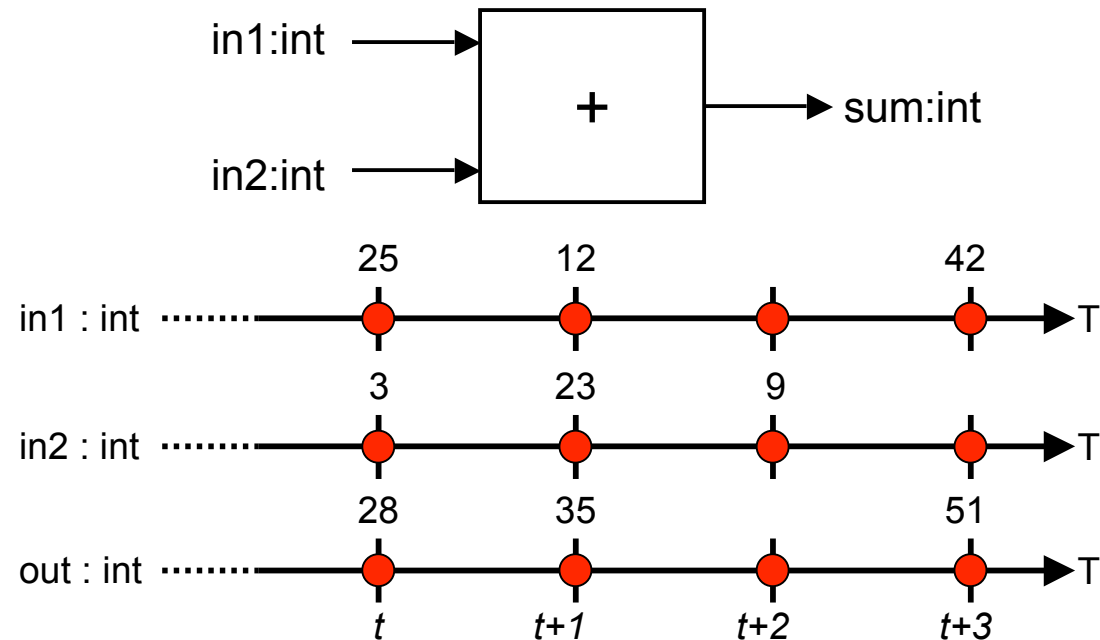
Purpose: Modeling a single (partial) execution of a system visible at its (unstructured) interface

Concept: Complete untimed observation trace

- Untimed Observation for all interface ports of a system:  
( ... • (in1,25) • (in2,12) • (in2,3) • (in2,23) • (out,28) • (in2,9) • (in1,12) • (out,35) •  
(in1,42) • (out,51) ) ∈ ((In ∪ Out) × int)<sup>ω</sup>



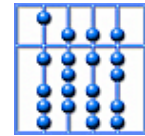
## Concept: Behavior



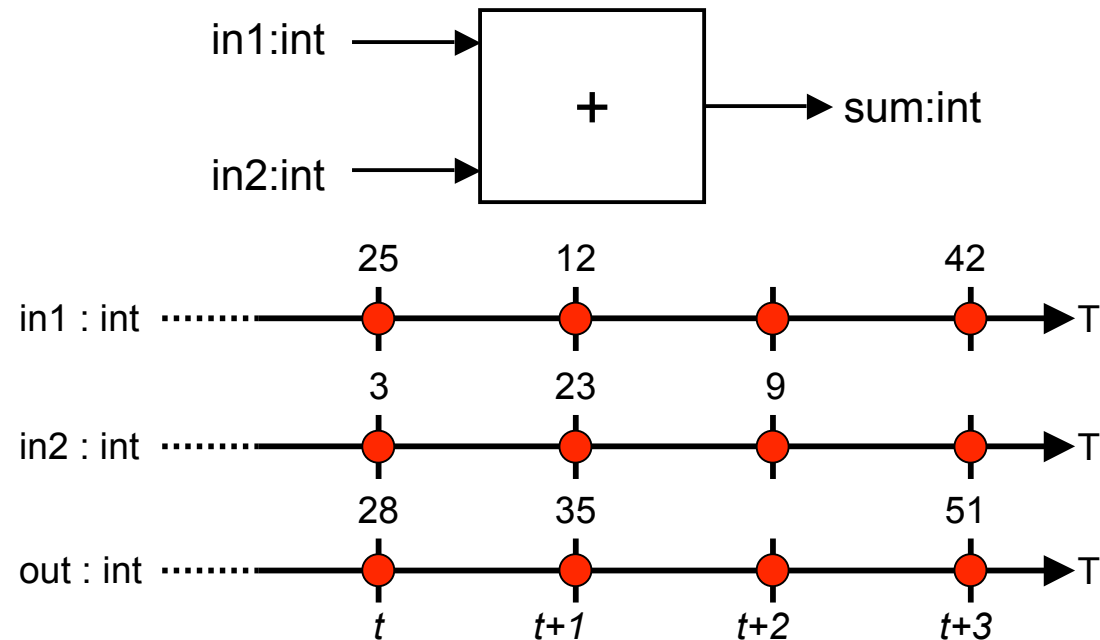
Purpose: Modeling all possible executions of a system visible at its interface

Concept: Behavior = set of observations traces

- Clocked behavior: Set of possible signal flows of a system
- Untimed behavior: Set of possible message exchanges of a system



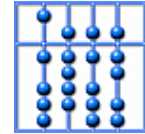
## Concept: Clocked Behavior



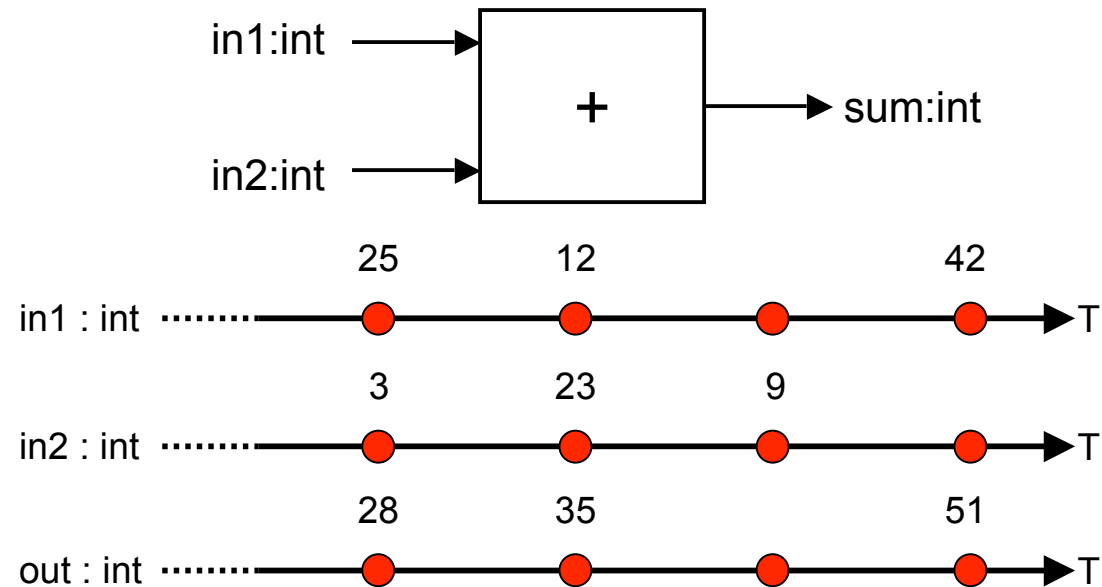
Purpose: Modeling all possible executions of a system visible at its interface

Concept: Behavior  $B$  = set of infinite signal traces

- Clocked behavior:  $B \subseteq (In_1 \cup \{-\})^\infty \times \dots \times (In_m \cup \{-\})^\infty \times (Out_1 \cup \{-\})^\infty \times \dots \times (Out_n \cup \{-\})^\infty$
- Example:  $\{ (- \cdot - \cdot \dots, - \cdot - \cdot \dots, - \cdot - \cdot \dots), \dots, (25 \cdot 3 \cdot \dots, - \cdot - \cdot \dots, - \cdot - \cdot \dots), \dots, (25 \cdot 12 \cdot \dots, 3 \cdot 23 \cdot \dots, 28 \cdot 35 \cdot \dots), \dots \}$



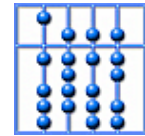
## Concept: Untimed Structured Behavior



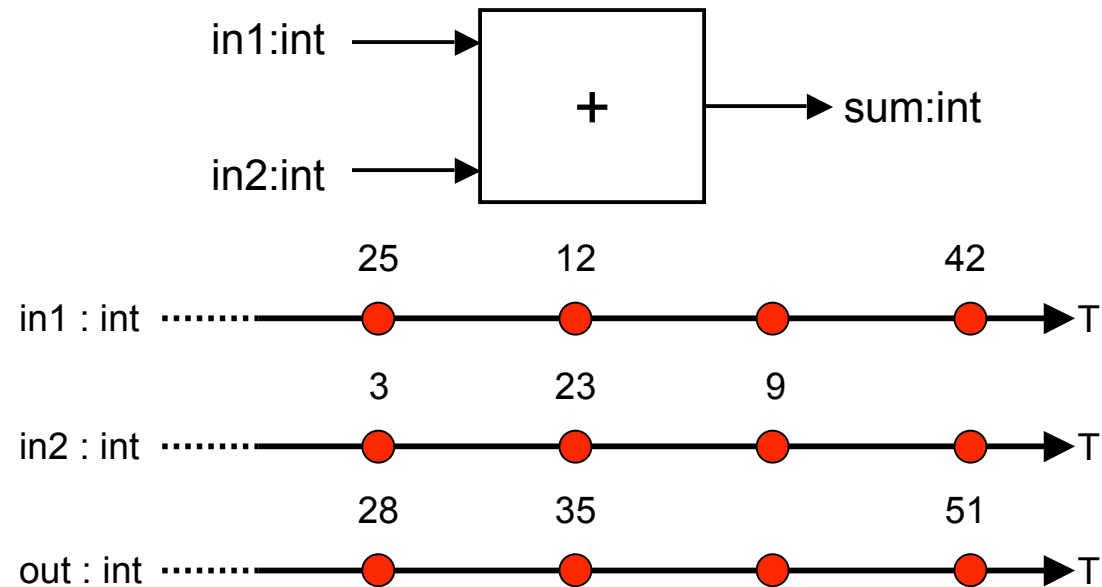
Purpose: Modeling all possible executions of a system visible at its interface

Concept: Behavior  $B$  = set of finite or infinite observations traces

- Untimed behavior:  $B \subseteq (In_1)^\omega \times \dots \times (In_m)^\omega \times (Out_1)^\omega \times \dots \times (Out_n)^\omega$
- Example:  $\{ (<>, <>, <>), (25, <>, <>), (25 \cdot 3, <>, <>), \dots, (25 \cdot 12 \cdot \dots, 3 \cdot 23 \cdot \dots, 28 \cdot 35 \cdot \dots), \dots \}$



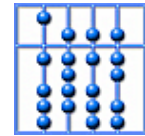
## Concept: Untimed Behavior



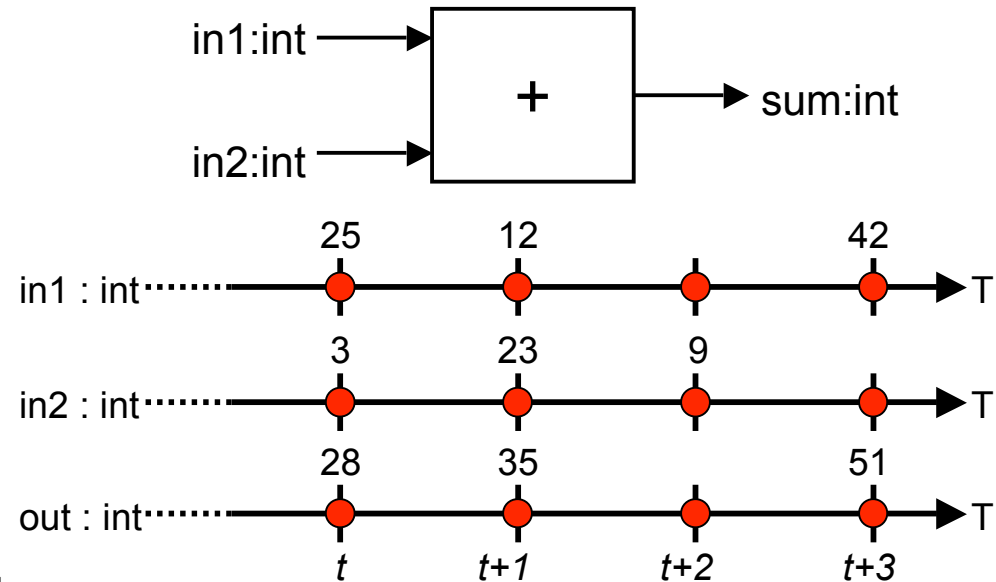
Purpose: Modeling all possible executions of a system visible at its unstructured interface

Concept: Behavior  $B$  = set of finite or infinite observations traces

- Untimed behavior:  
$$B \subseteq ((\{in_1\} \times In_1) \cup \dots \cup (\{in_m\} \times In_m) \cup (\{out_1\} \times Out_1) \cup \dots \cup (\{out_n\} \times Out_n))^{\omega}$$
- Example:  $\{ \langle \rangle, ((in1,25)), ((in1,25) \cdot (in2,3)), \dots, ((in1,25) \cdot (in1,12) \cdot (in2,3) \cdot (out,28) \cdot (in1,42) \cdot (in2,23) \cdot (out,35)), \dots \}$



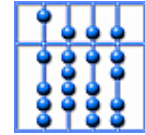
## 2.3 Summary: Modeling Behavior



Concepts:

- Interface: Shared part of system and environment
- Signal: Time-based description of flow of information
- Message: Event-based description of flow of information
- (Observation) Trace: Sequence of interactions at interface
- Behavior: Set of (observation) traces

Model: Sets of timed/untimed traces



## 2.4 Modeling Communication: Synchronized Behaviors

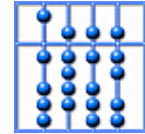
---

**Goal:** Define a model to describe the behavior of communicating concurrent reactive systems

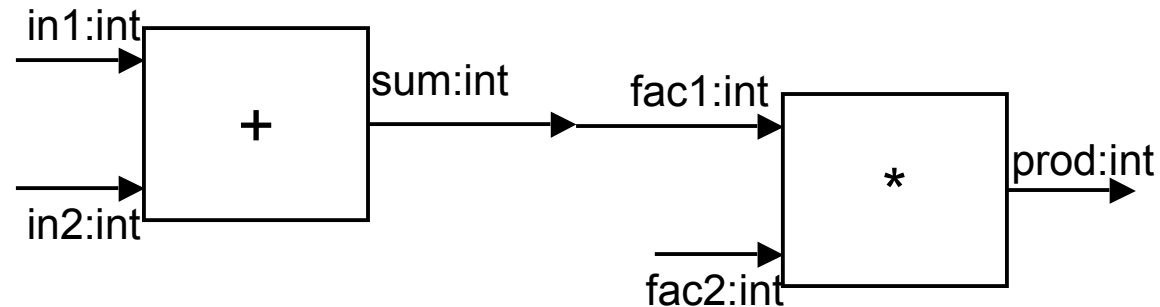
- Relevant: Address aspects evolving with communication
- General: Cover aspects independent of specific computation techniques
- Abstract: Ignore aspects like communication bandwidth, execution speed

**Concept:** Connected interface, shared/independent interaction, concurrent execution

**Model:** Synchronized Behavior



## Concept: Connected Interface



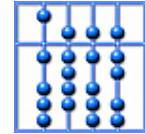
Purpose: Describing communicating systems

Concept: Connected interfaces  $(In1, Out1)$  and  $(In2, Out2)$

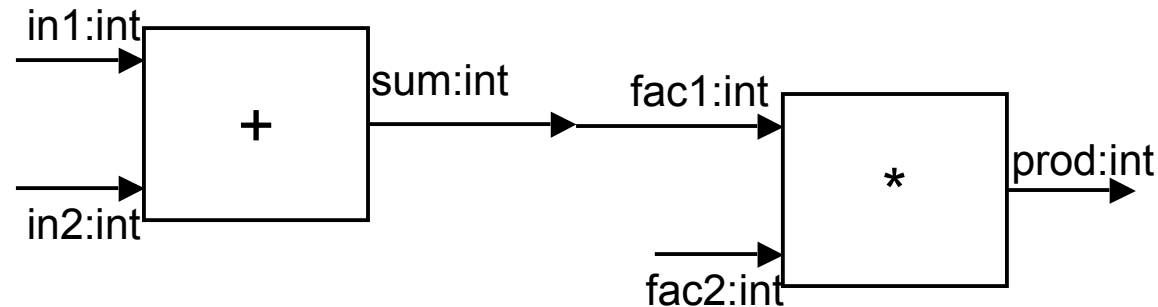
- Inputs:  $(In1 \cup In2) \setminus (Out1 \cup Out2)$
- Outputs:  $(Out1 \cup Out2) \setminus (In1 \cup In2)$
- Internal communication:  $(In1 \cap Out2) \setminus (In2 \cap Out1)$

Example:

- Inputs: in1, in2, fac2
- Outputs: prod
- Internal communication: sum/fac1



## Concept: Composed Clocked Behavior



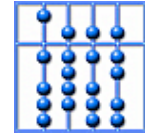
Purpose: Describing communicating systems

Concept: Composed Timed Behavior  $B1 \parallel B2$  with shared ports *Shared1* and *Shared2*

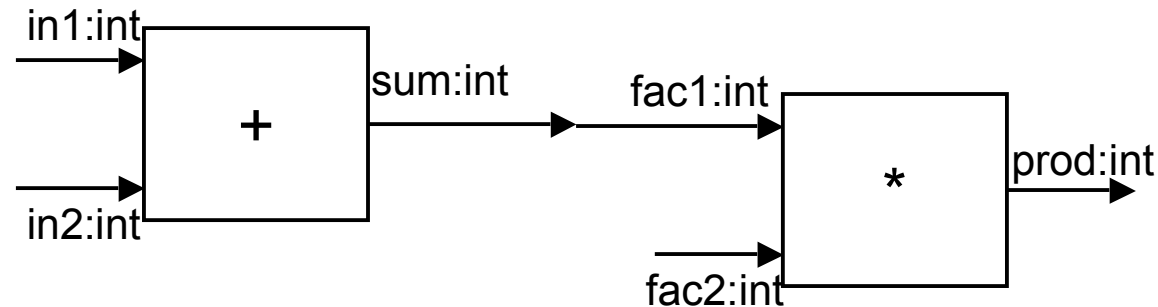
- $B1: (in1, shared1, out1, shared2) \in (In2 \oplus \{-\})^\infty, Out1 \oplus \{-\})^\infty$
- $B2: (in2, shared2, out2, shared1) \in (In2 \oplus \{-\})^\infty, Out2 \oplus \{-\})^\infty$
- $B3: \{ (in1, in2, out1, out2) \mid \exists shared1, shared2: (in1, shared1, out1, shared2) \in B1 \wedge (in2, shared2, out2, shared1) \in B2 \}$

Examples:

- $(- \cdot - \cdot \dots, - \cdot - \cdot \dots, - \cdot - \cdot \dots - \cdot - \cdot \dots)$  for  $(- \cdot - \cdot \dots, - \cdot - \cdot \dots)$ ,  $(- \cdot - \cdot \dots, - \cdot - \cdot \dots)$
- $(25 \cdot 3 \cdot \dots, - \cdot - \cdot \dots, 2 \cdot -1 \cdot \dots, - \cdot - \cdot \dots)$  for  $(25 \cdot 3 \cdot \dots, - \cdot - \cdot \dots, - \cdot - \cdot \dots)$ ,  $(- \cdot - \cdot \dots, 2 \cdot -1 \cdot \dots, - \cdot - \cdot \dots)$
- $(25 \cdot 12 \cdot \dots, 3 \cdot 23 \cdot \dots, 2 \cdot -1 \cdot \dots, 56 \cdot -35 \cdot \dots)$  for  $(25 \cdot 12 \cdot \dots, 3 \cdot 23 \cdot \dots, 28 \cdot 35 \cdot \dots)$ ,  $(28 \cdot 35 \cdot \dots, 2 \cdot -1 \cdot \dots, 56 \cdot -35 \cdot \dots)$



## Concept: Composed Untimed Structured Behavior



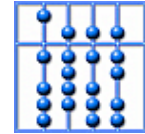
Purpose: Describing communicating systems with structured interface

Concept: Composed Timed Behavior  $B1 \parallel B2$  with shared ports *Shared1* and *Shared2*

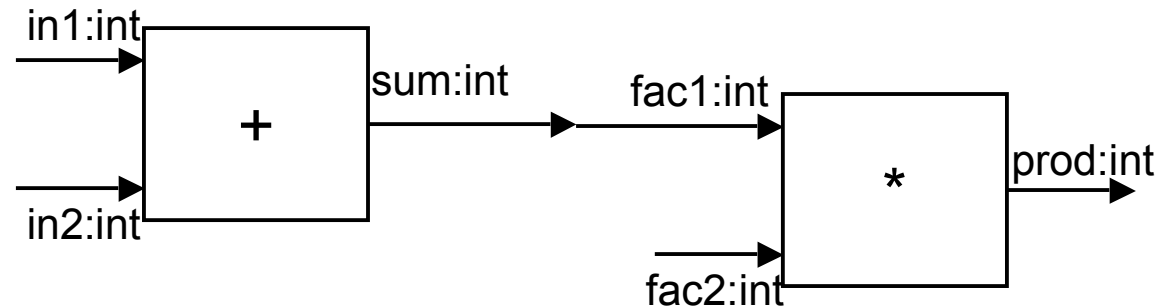
- $B1: (in1, shared1, out1, shared2) \in (In1^{\omega}, Out1^{\omega})$
- $B2: (in2, shared2, out2, shared1) \in (In2^{\omega}, Out2^{\omega})$
- $B3: \{ (in1, in2, out1, out2) \mid \exists shared1, shared2: (in1, shared1, out1, shared2) \in B1 \wedge (in2, shared2, out2, shared1) \in B2 \}$

Examples:

- $(\langle \rangle, \langle \rangle, \langle \rangle)$  for  $(\langle \rangle, \langle \rangle, \langle \rangle)$ ,  $(\langle \rangle, \langle \rangle, \langle \rangle)$
- $(25 \cdot 3, \langle \rangle, 2 \cdot -1, \langle \rangle)$  for  $(25 \cdot 3, \langle \rangle, \langle \rangle)$ ,  $(\langle \rangle, 2 \cdot -1, \langle \rangle)$
- $(25 \cdot 12 \cdot \dots, 3 \cdot 23 \cdot \dots, 2 \cdot -1 \cdot \dots, 56 \cdot -35 \cdot \dots)$  for  $(25 \cdot 12 \cdot \dots, 3 \cdot 23 \cdot \dots, 28 \cdot 35 \cdot \dots)$ ,  $(28 \cdot 35 \cdot \dots, 2 \cdot -1 \cdot \dots, 56 \cdot -35 \cdot \dots)$



## Concept: Composed Untimed Behavior



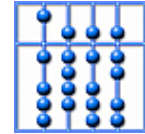
Purpose: Describing communicating systems with unstructured interface

Concept: Composed Timed Behavior  $B1 \parallel B2$  with shared ports  $(In1 \cap Out2)$  and  $(In2 \cap Out1)$

- $B1: b1 \in (In1 \cup Out1)^\omega$
- $B2: b2 \in (In2 \cup Out2)^\omega$
- $B3: \{ b \odot ((In1 \setminus Out2) \cup (In2 \setminus Out1)) \mid \exists b \in (In1 \cup Out1 \cup In2 \cup Out2) : b \odot (In1 \cup Out1) \in B1 \wedge b \odot (In2 \cup Out2) \in B2 \}$

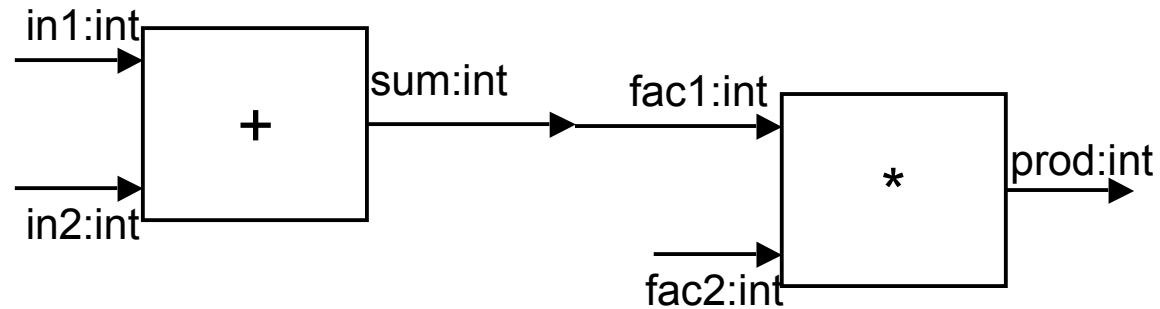
Examples:

- $\langle \rangle$  for  $\langle \rangle, \langle \rangle$
- $(in1,25) \cdot (fac2,2) \cdot (fac2,-1) \cdot (in1,3)$  for  $(in1,25) \cdot (in1,3), (fac2,2) \cdot (fac2,-1)$
- $(in1,25) \cdot (fac2,2) \cdot (fac2,-1) \cdot (in2,3) \cdot (in2,12) \cdot (prod,56) \cdot (in1,23) \cdot (prod,-35)$  for  $(in1,25) \cdot (in2,3) \cdot (sum,28) \cdot (in2,12) \cdot (in1,3) \cdot (sum,15), (fac2,2) \cdot (fac2,-1) \cdot (sum,28) \cdot (prod,56) \cdot (sum,15) \cdot (prod,-35)$



## 2.4 Summary: Modeling Communication

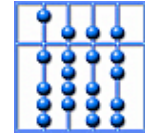
---



Concepts:

- Connected Interfaces: Shared parts of components
- Independent interaction: Disjoint observations about traces
- Synchronized Interaction: Common observations about traces

Model: Composed sets of (observation) traces



## 2.4 Questions

---

1. Exercise: Describe the behavior of the adder, doubler, and initializer for values  $-2, \dots, 2$  and traces up to length 3.
2. Exercise: Calculate the behavior of the integrator from those behaviors.
3. What is the difference between the traces of a LTS of a system and a stream-based behavior describing the same system?
4. When does a behavior describe a receptive (i.e. input-enabled system)?
5. When does a behavior describe a deterministic system?
6. When does a behavior describe a receptive and deterministic system?
7. Allowing only untimed behavior, what kind of send/receive actions are possible?
8. What kind of behavior corresponds to systems obeying Questions 4 and 5