A Distributed ASM-Theorem

Andreas Glausch

Gurevich '99:
Sequential ASM-Theorem

Every *sequential small-step algorithm* can be represented by a *Sequential Abstract State Machine (ASM)*.

Our aim: a Distributed ASM-Theorem

Every *distributed small-step algorithm* can be represented by a *Distributed ASM*.

Project funded by the
German Research Council (DFG).

Sequential Small-Step Algorithms

as defined by Yuri Gurevich

Sequential algorithms are defined by

1. the sequential time postulate,
2. the abstract state postulate, and
3. the bounded exploration postulate.

1. Sequential-time postulate

A sequential algorithm consists of

- a set of states S and
- a next-state function $\tau : S \to S$.

Each state is a structure:

- a universe U, and
- a finite collection of functions over U.

2. Abstract state postulate

- each state is a structure,
- τ preserves the universe of every state, and
- τ preserves isomorphism φ between states.

$$\begin{array}{ccc} S & \longrightarrow & \tau(S) \\ \varphi \downarrow & & \varphi \downarrow \\ R & \longrightarrow & \tau(R) \end{array}$$

3. Bounded exploration postulate

There exists a finite set of ground terms
that characterizes *all* steps.

Sequential ASM-Theorem

Sequential ASM-Theorem:

Every sequential algorithm can be represented by a *Sequential ASM*.

A Sequential ASM consists of

- a set of states
- an ASM program:

  if $cond_1$ then $t_1 := t_1'$
  if $cond_2$ then $t_2 := t_2'$
  …
  if $cond_n$ then $t_n := t_n'$

Distributed Small-Step Algorithms

Distributed algorithms are defined by

1. the states and actions postulate,
2. the isomorphism postulate,
3. the autonomy postulate, and
4. the bounded action postulate.

Towards Postulate 1…

A sequential algorithm consists of

- a set of states S, and
- a next-state function $\tau : S \to S$.

A distributed algorithm consists of

- a set of states S, and
- steps…?

global steps $S \to \tau(S)$ not suited

$\longrightarrow$ need *local* steps

Local steps

1. Represent each state as a structure.

2. Decompose a structure into local components.

3. Local step: modifies <u>some</u> components.

Local component of a structure

Example: U = {1, 2, 3}

a()
1

b()
2

x	val(x)
1	1
2	2
3	3

x	inc(x)
1	2
2	3
3	1

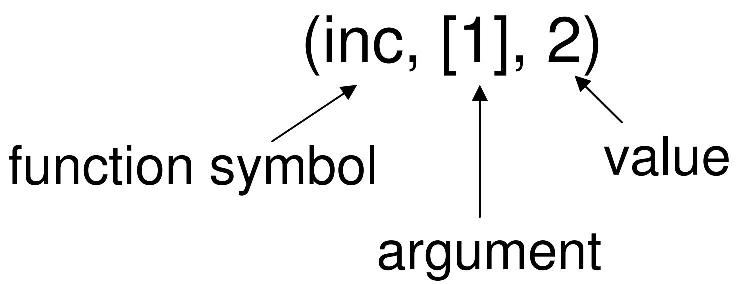
Local component of a structure

a()
1

b()
2

x	val(x)
1	1
2	2
3	3

x	inc(x)
1	2
2	3
3	1



Local component of a structure

(inc, [1], 2) ⟵ **store**

a()	b()
1	2

x	val(x)
1	1
2	2
3	3

x	inc(x)
1	2
2	3
3	1

Gurevich:

Describe local changes of a structure by stores.

Idea:

Describe the complete structure by stores.

Decompose the structure into *stores*

a()
1

b()
2

x	val(*x*)
1	1
2	2
3	3

x	inc(*x*)
1	2
2	3
3	1

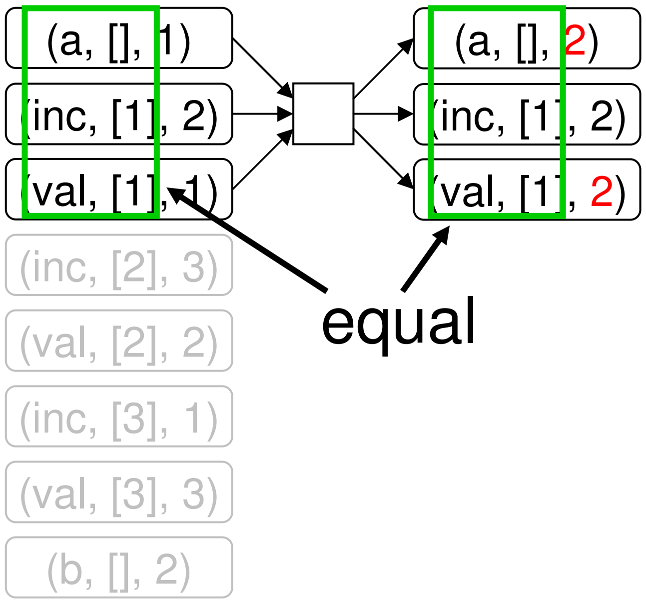
→
←

(a, [], 1)
(inc, [1], 2)
(val, [1], 1)
(inc, [2], 3)
(val, [2], 2)
(inc, [3], 1)
(val, [3], 3)
(b, [], 2)

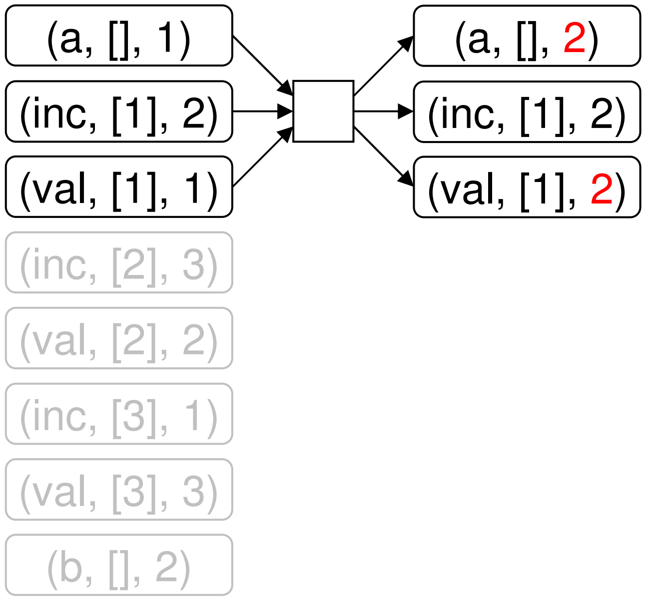
Action : locally bounded change of the state

(a, [], 1)
(inc, [1], 2)
(val, [1], 1)
(inc, [2], 3)
(val, [2], 2)
(inc, [3], 1)
(val, [3], 3)
(b, [], 2)

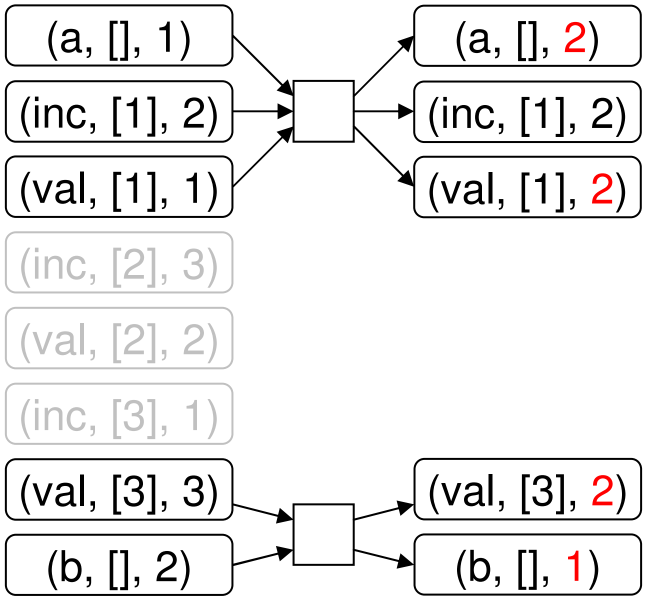
Action : locally bounded change of the state



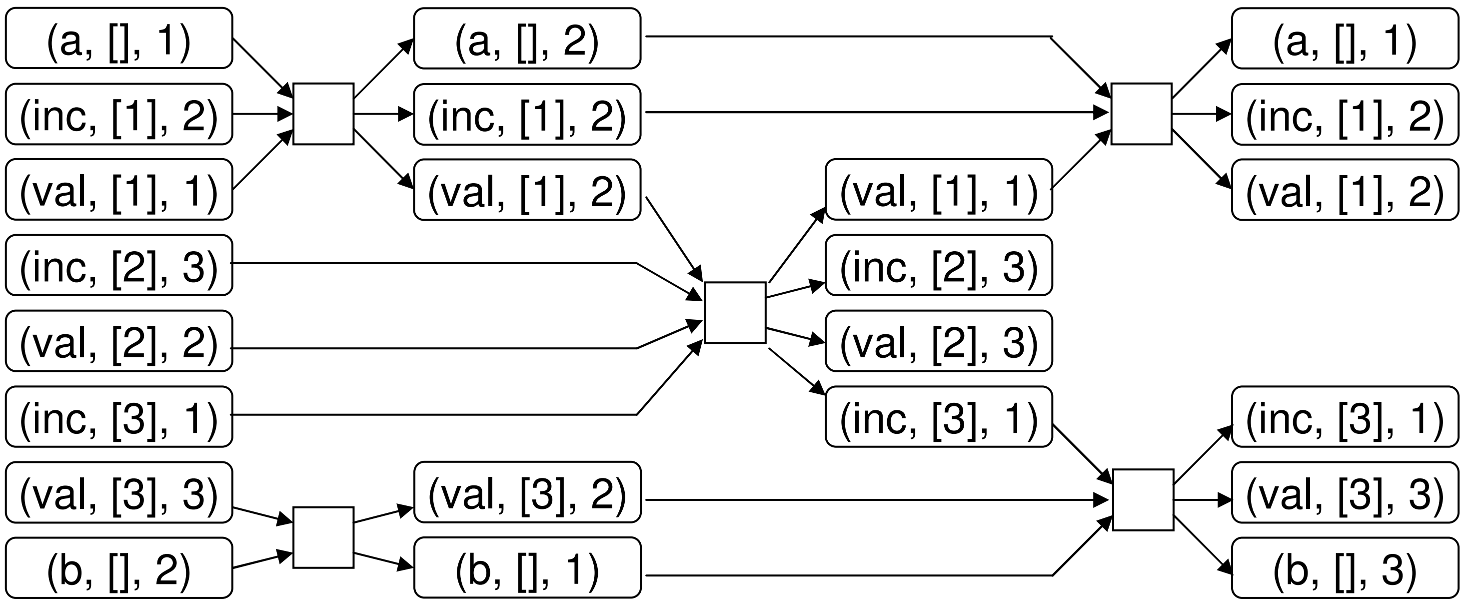
Actions may occur concurrently



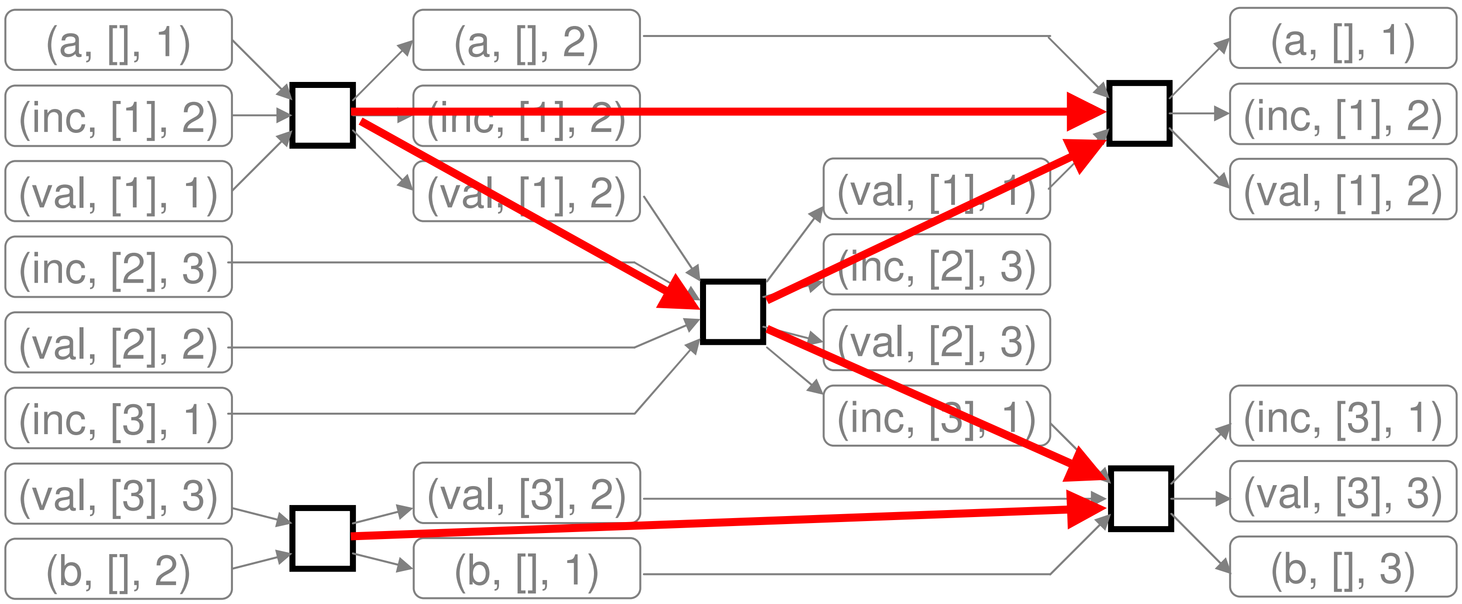
Actions may occur concurrently



Actions yield *distributed runs*



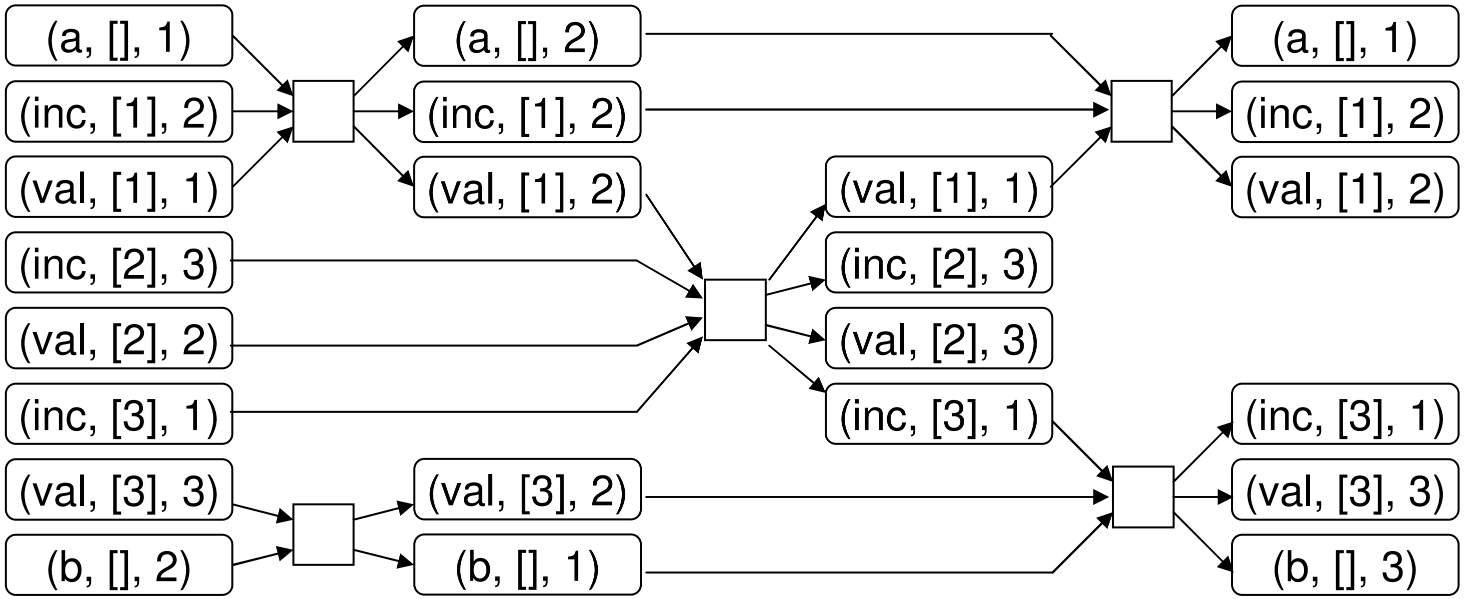
Actions yield *distributed runs*



⟶ partial order of action occurrences ("Lipari Guide")

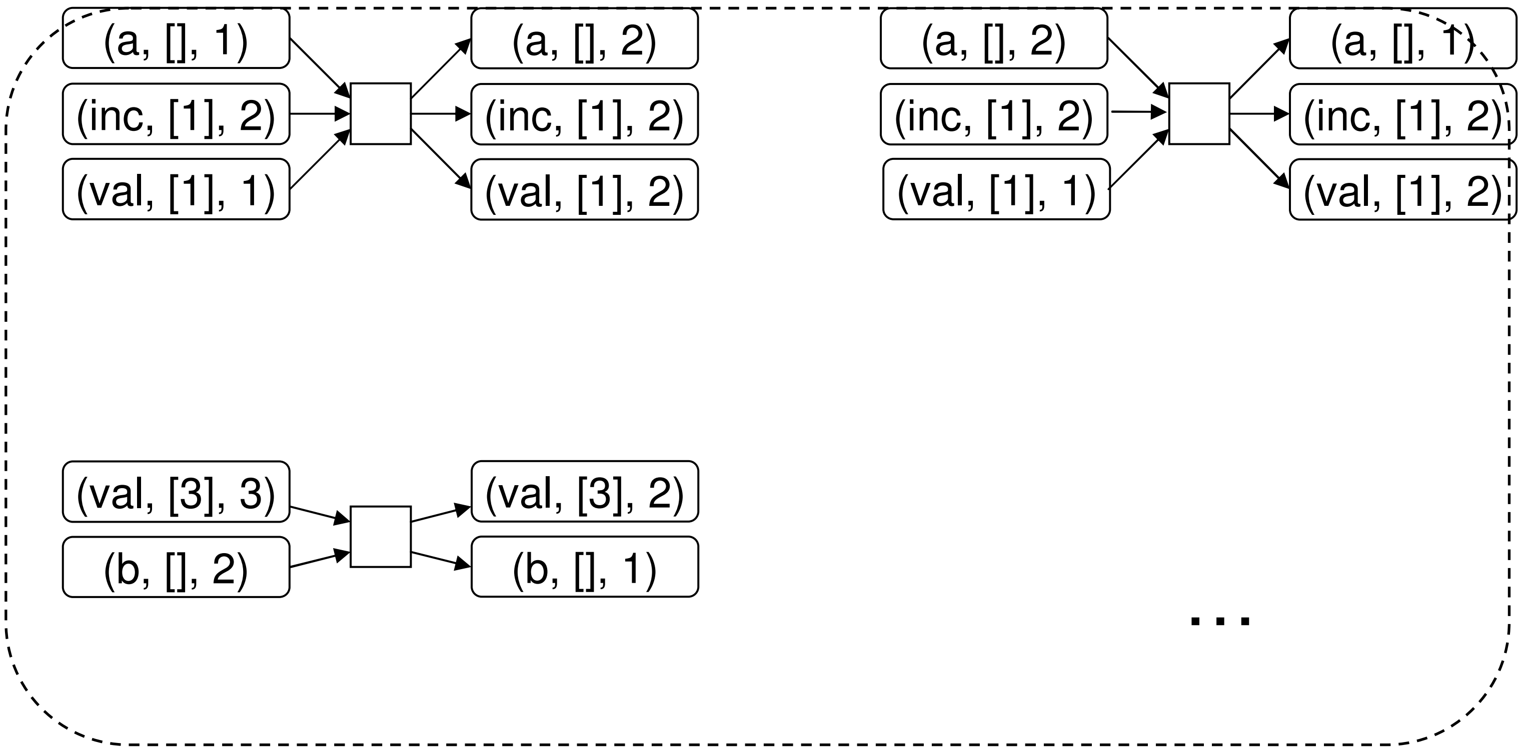
Idea:

Describe distributed behaviour by a **set of actions.**



Idea:

Describe distributed behaviour by a **set of actions.**



1. States and actions postulate

A distributed algorithm consists of

- a set of states S, and
- steps…?

1. States and actions postulate

A distributed algorithm consists of

- a set of states S, and
- a set of actions.

Sequential time postulate

A sequential algorithm consists of

- a set of states S and
- a next-state function $\tau : S \rightarrow S$.

2. Isomorphism postulate

At isomorphic states, a distributed algorithm performs isomorphic actions.

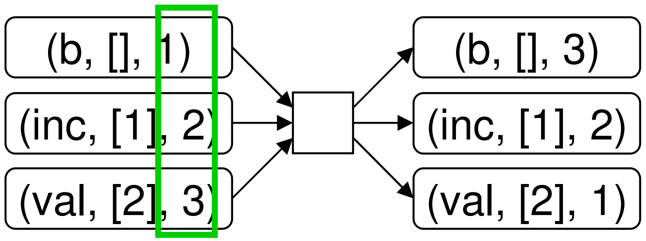
Abstract state postulate

- …
- τ preserves isomorphism φ between states.

3. Autonomy postulate

Every action *a* is *autonomous*:

a only uses elements of the universe that are *provided* to *a*.



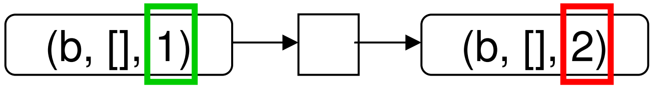
autonomous

Elements are provided by values of functions

3. Autonomy postulate

Every action *a* is *autonomous*:

a only uses elements of the universe that are *provided* to *a*.

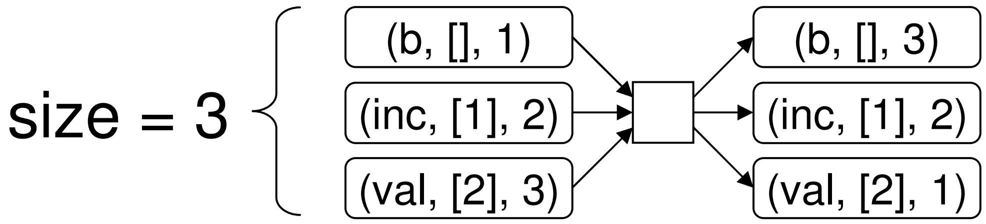


not autonomous!

1 is provided   2 is not provided

4. Bounded actions postulate

There exists an upper bound k∈ℕ for the size of all actions.



4. Bounded actions postulate

There exists an upper bound $k \in \mathbb{N}$ for the size of all actions.

Bounded exploration postulate:

There exists a finite set of terms that characterizes *all* steps.

A *distributed algorithm*

1. consists of states and actions,
2. respects isomorphism,
3. acts autonomously, and
4. has bounded actions.

A Distributed ASM-Theorem

Distributed ASM-Theorem:

Every distributed algorithm can be represented by a distributed ASM.

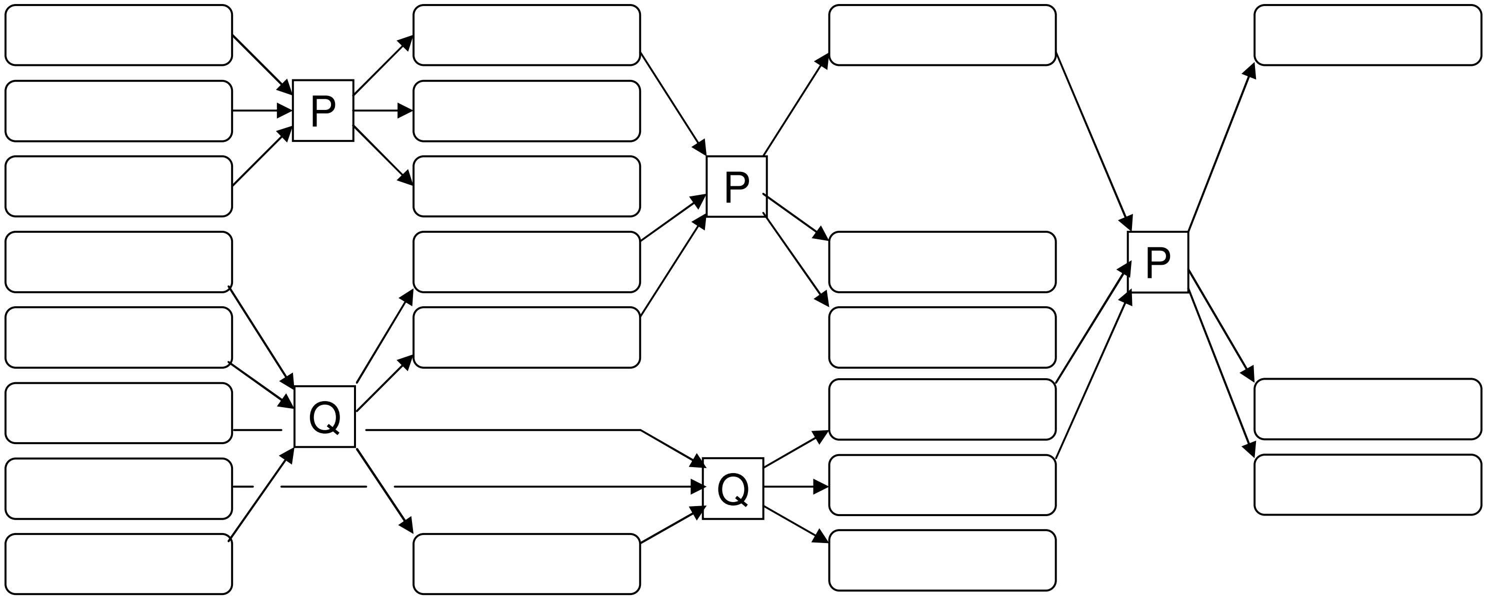
A Distributed ASM consists of

- a set of states,
- a finite set of ASM programs $\{P_1, \ldots, P_n\}$.

$$P_i \begin{cases} \text{if } cond_1 \text{ then } t_1 := t_1' \\ \text{if } cond_2 \text{ then } t_2 := t_2' \\ \ldots \\ \text{if } cond_n \text{ then } t_n := t_n' \end{cases} \longrightarrow \text{set of actions}$$

A Distributed ASM yields distributed runs

ASM programs: P, Q



A Distributed ASM consists of

- a set of states
- a finite set of ASM programs $\{P_1, \dots, P_n\}$.

A Sequential ASM consists of

- a set of states
- an ASM program.

Summary

- distributed algorithms generalize sequential algorithms

  Idea: replace global steps by local actions

- sequential ASM-Theorem is generalized to a distributed ASM-Theorem

Future work

- allow actions to share stores
- integrate agent concepts from the “Lipari Guide”:
  - self-symbol
  - adding/removing agents
- synchronous actions (handshake communication)

Thank you!