Modelling Reaction Systems with Petri nets

Jetty Kleijn & Grzegorz Rozenberg
Leiden University
The Netherlands

Maciej Koutny
Newcastle University
United Kingdom

Universiteit Leiden

Newcastle University

Newcastle upon Tyne, 20th June 2011

Computational systems biology

Key features of the computational (algorithmic) approach to systems biology:

- computations (operational description) rather than equations (denotational description)
- discrete state spaces (precise description of changes) rather than continuous state spaces (average behaviour)

C.Priami [Communication of the ACM, May 2009]:

- "... causality between events, the temporal ordering of interactions and the spatial distribution of components are becoming essential to addressing questions at system level ..."
- "... design principles of large software systems can help in developing an algorithmic discipline not only for systems biology but also for synthetic biology ..."
- "... the convergence of CS and biology will serve both disciplines, providing each with greater power and relevance ..."

Motivation

- Computational *understanding* of the functioning of living cells
- Formal framework of *reaction systems* (investigation of processes carried out by biochemical reactions in living cells)
- Functioning of living cell is based on interactions between individual reactions regulated by two mechanisms:

  facilitation

  inhibition
- Interactions determine *dynamic processes*
- Two axioms:

  non-permanency (cell is an open system)

  non-counting (high level of abstraction)
- Stochasticity / time / counting etc can be introduced later through *contexts* (not included in the basic model) in order not to preclude other explanations

Motivation

- This paper aims [CfP: ... standard Petri nets for qualitative modelling, if kinetics are unknown or deliberately abstracted ...] :

  faithful Petri net model of reaction systems

  qualitative approach to biological processes using Petri nets

- Long term aim:

  establishing whether Petri net based concepts (such as causal processes) and methods (such as synthesis of nets) could be transferred to reaction systems

- Not in this paper:

  direct feedback to the area of biological applications

Outline

- Basic reaction systems
- Reaction systems and low-level Petri nets
- Reaction systems and high-level Petri nets
- Set-nets
- Reaction systems and set-nets
- Concluding remarks

Reaction systems

- Reaction system $\mathcal{A}$ = (S,A) where:

  S finite background set comprising entities
  A set of reactions

- Reaction a=(R,I,P) of sets of entities:

  R reactants
  I inhibitors
  P products

- Initialised reaction system A=(S,A,C) where C is an initial state (a set of entities)

- Example: $\mathcal{A}$ =({w,x,y,z},{a,b,c},{x,z}) where:

  a = ({x},{y},{y,z})
  b = ({y},{x},{x,z})
  c = ({z},{w},{z})

Reaction systems

- Reaction $a = (R,I,P)$ is enabled at a state C if
  R is included in C
  I and C are disjoint
- The result $res_a(C)$ of a on C is P if a is enabled, and the empty set otherwise
- The result $res_A(C)$ of A on C is the union of all the reactions' results and we denote:
  $C \rightarrow res_A(C)$
- Example: $\mathcal{A} = (\{w,x,y,z\},\{a,b,c\},\{x,z\})$ where:
  $a = (\{x\},\{y\},\{y,z\})$
  $b = (\{y\},\{x\},\{x,z\})$
  $c = (\{z\},\{w\},\{z\})$
-

  $\{x,z\} \rightarrow \{y,z\}$
  $\{y,z\} \rightarrow \{x,z\}$
- *Note:* all entities present in C which are not produced disappear
  Note: there is no conflict between reactions in the "classic" sense
  Note: reactions systems are qualitative rather than quantitative

Outline

- Basic reaction systems
- Reaction systems and low-level Petri nets
- Reaction systems and high-level Petri nets
- Set-nets
- Reaction systems and set-nets
- Concluding remarks

Reaction system example

Context independent

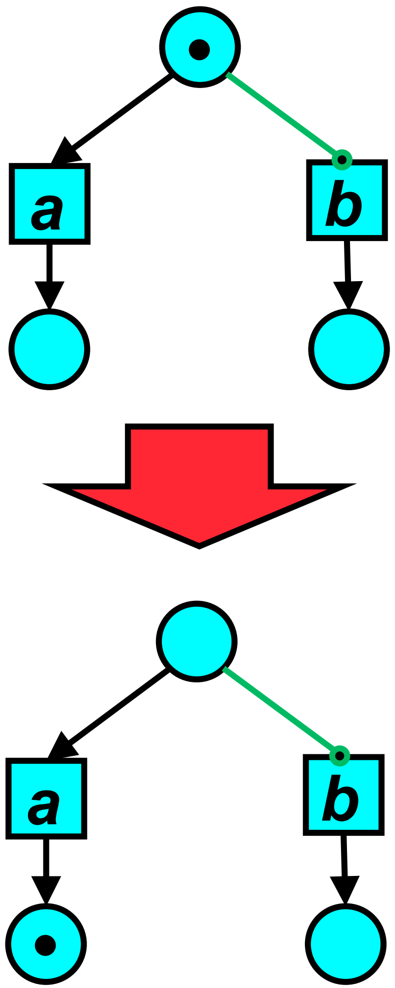
$\mathcal{A}$ = ({x,y,z}, {a,b,c})

a = ({x},{y},{y,z})
b = ({y},{x},{x,z})
c = ({z},{ },{z})

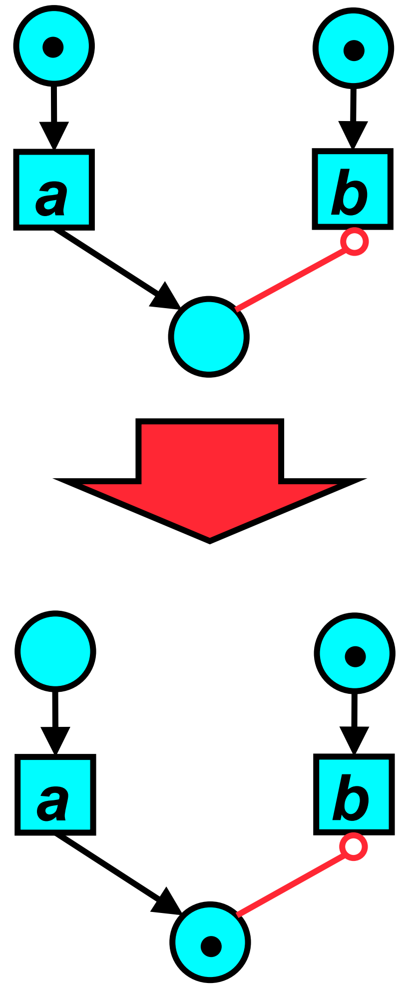
{x,z} *is our initial state*

entities	⇒	*places*
reactions		*transitions*
state		*marking*

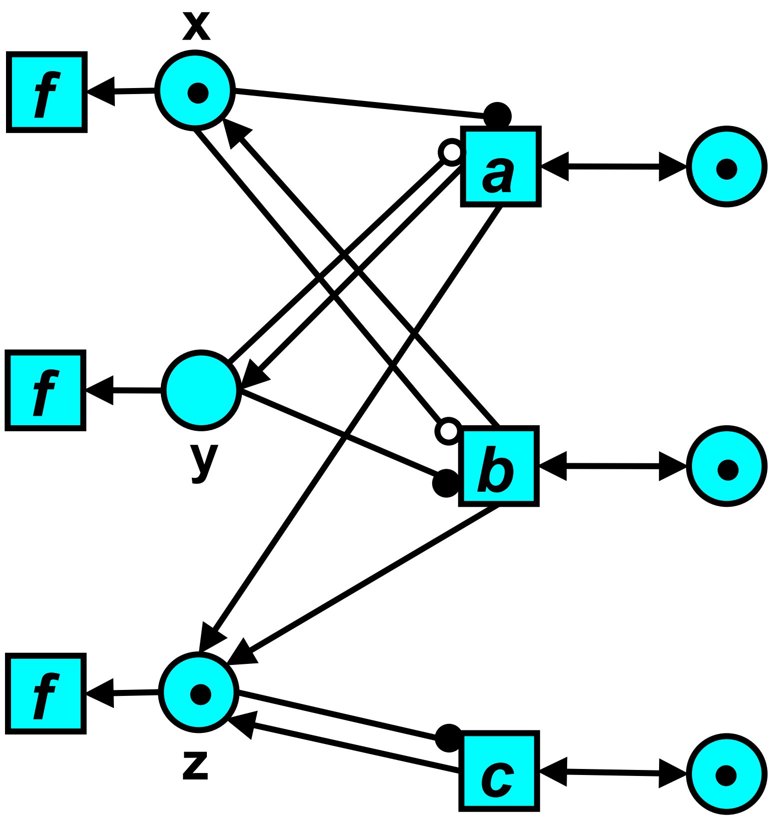
Activator / inhibitor arc



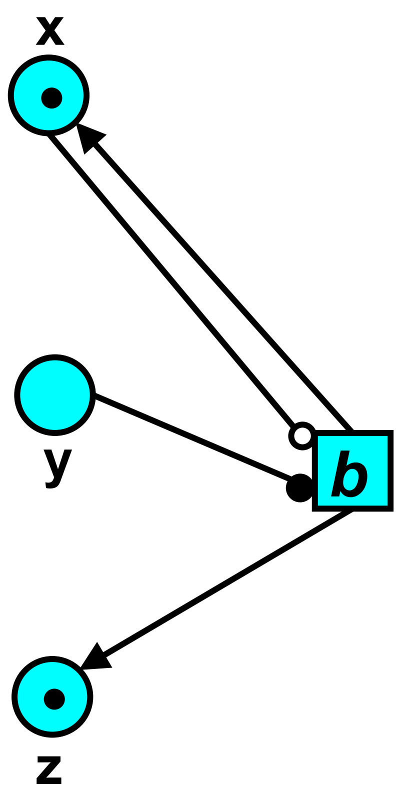
testing for
presence / absence
of tokens



Method I



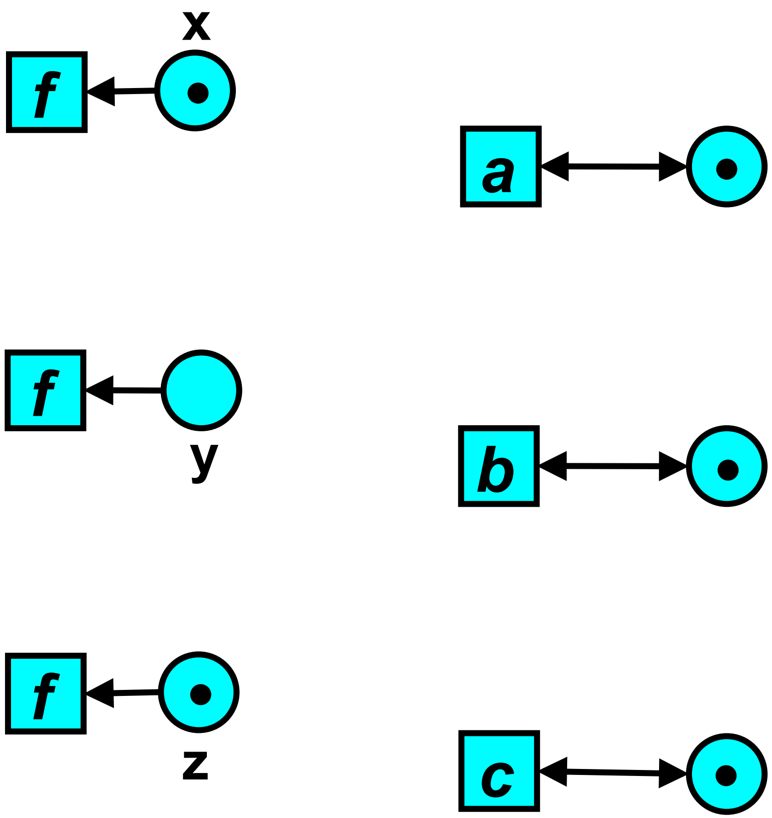
Method I (details)



b = ({y},{x},{x,z})

{x,z} is initial configuration

Method I (details)

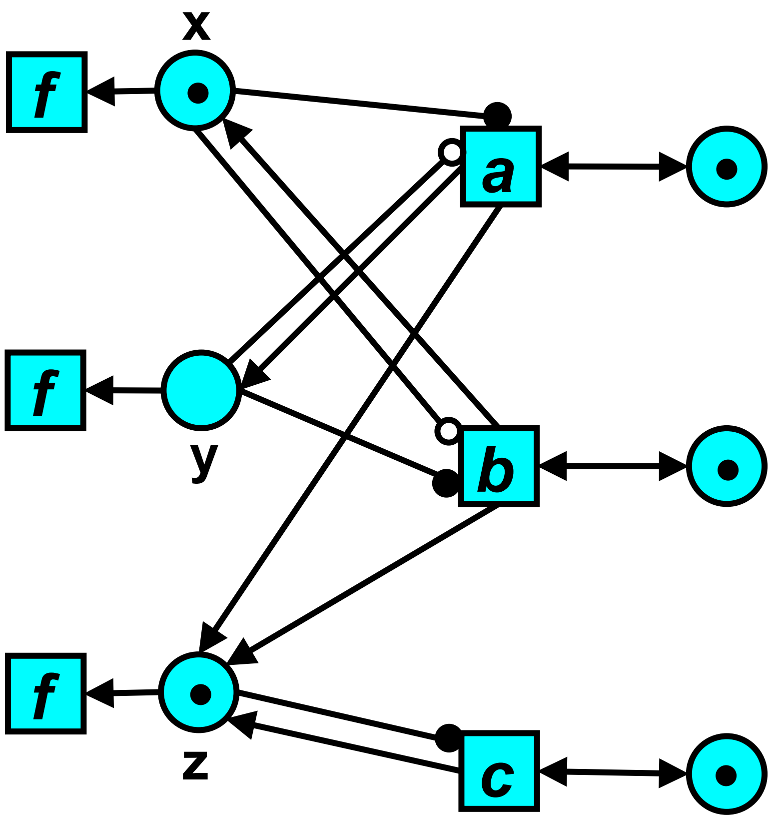


maximal concurrency

implies

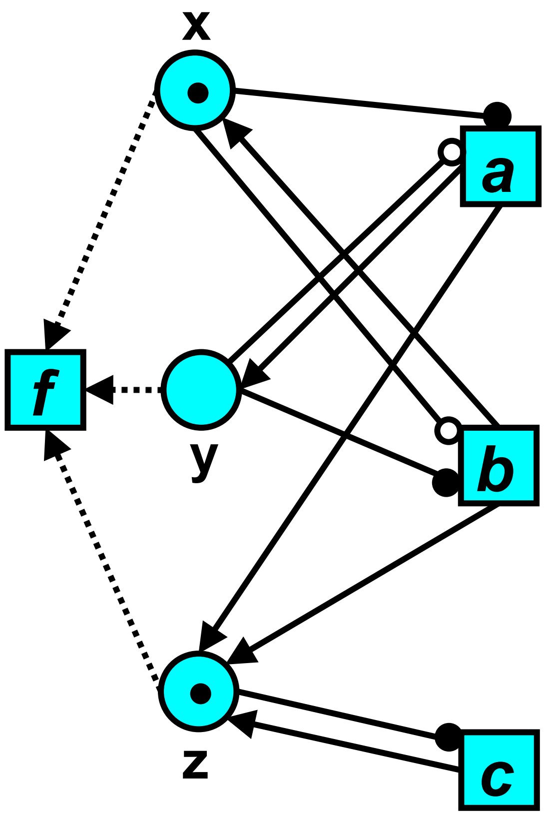
tokens in x y z are "flushed"

Method I (results)



- *the net simulates $\mathcal{A}$*
- *one execution in $\mathcal{A}$ may correspond to many executions in the net*
- *steps of transitions are multisets*

Method II



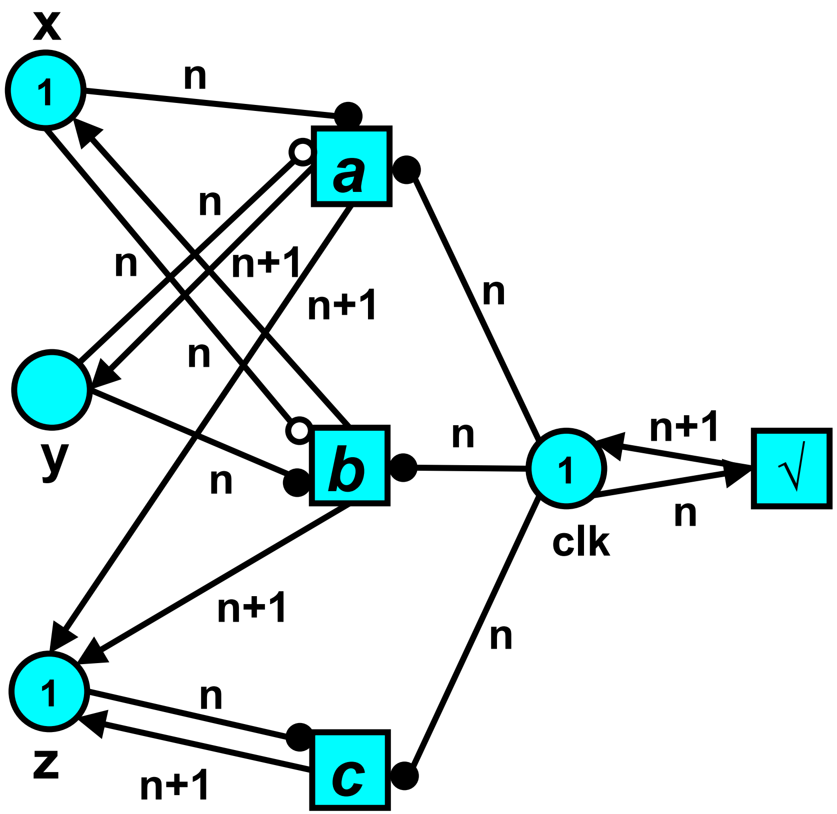
- *the net simulates $\mathcal{A}$*
- *steps are sets*
- *one execution in $\mathcal{A}$ may correspond to many executions in the net*

reset arcs empty x y z

Outline

- Basic reaction systems
- Reaction systems and low-level Petri nets
- Reaction systems and high-level Petri nets
- Set-nets
- Reaction systems and set-nets
- Concluding remarks

Method III

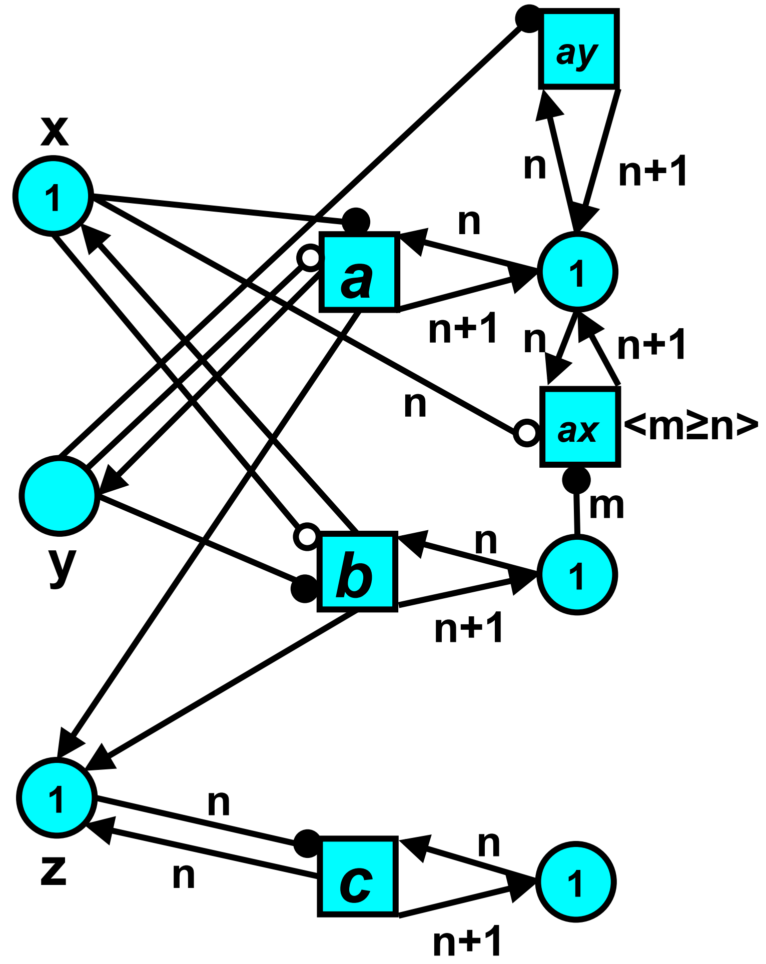


- *the net simulates $\mathcal{A}$*
- *steps are sets*
- *no "flushing"*
- *unboundedness*

tokens are time points

clock values are increased at each step

Method IV



- *the net simulates $\mathcal{A}$*
- *steps are sets*
- *no "flushing"*
- *interleaving executions can simulate maximally concurrent executions*
- *unboundedness*

individual clocks
auxiliary transitions update local clocks

Outline

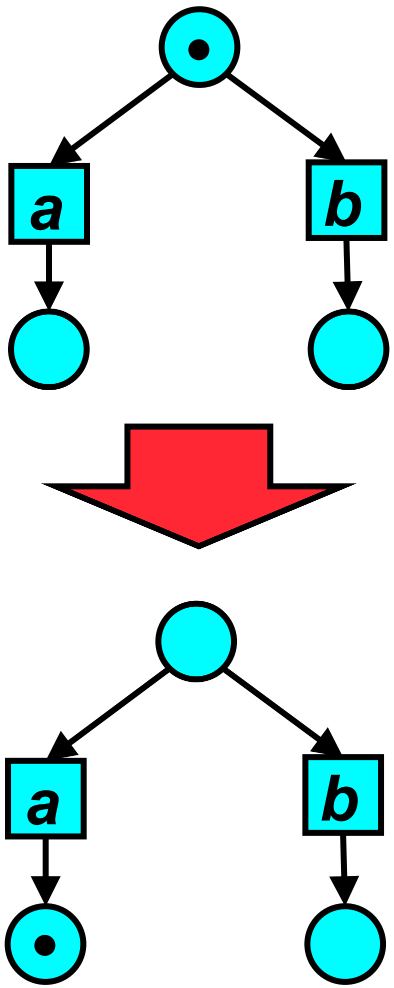
- Basic reaction systems
- Reaction systems and low-level Petri nets
- Reaction systems and high-level Petri nets
- Set-nets
- Reaction systems and set-nets
- Concluding remarks

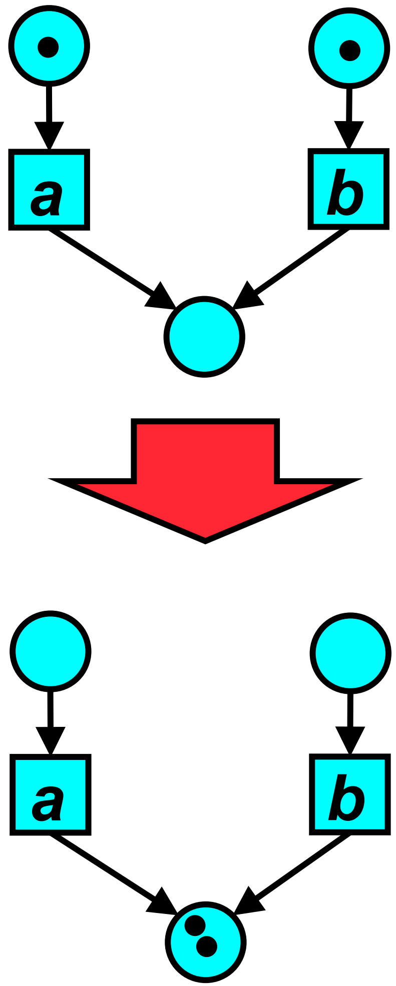
Discussion

- No method provides a direct and/or elegant representation of the behaviour of the reaction system
- No counting, just “presence” or “absence” (also in production), is difficult to manage in the standard models of Petri nets (even the most fundamental ones, including Elementary Net systems)
- Hence we propose a NEW class of nets based on set arithmetic

P/T-nets

conflict

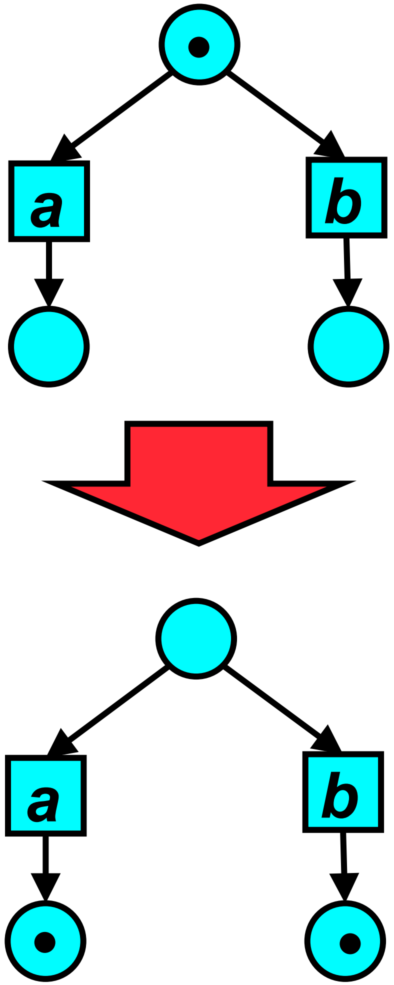


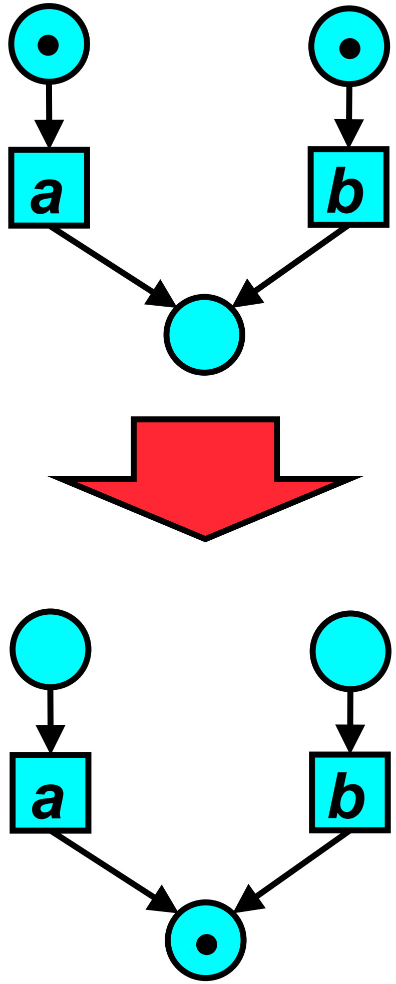


counting

Set-nets

no conflict



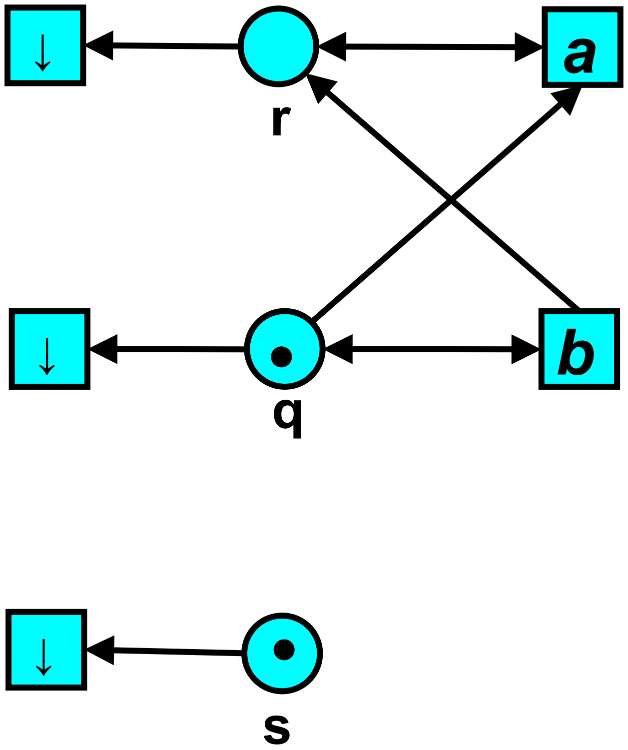


no counting

Outline

- Basic reaction systems
- Reaction systems and low-level Petri nets
- Reaction systems and high-level Petri nets
- Set-nets
- Reaction systems and set-nets
- Concluding remarks

Method V



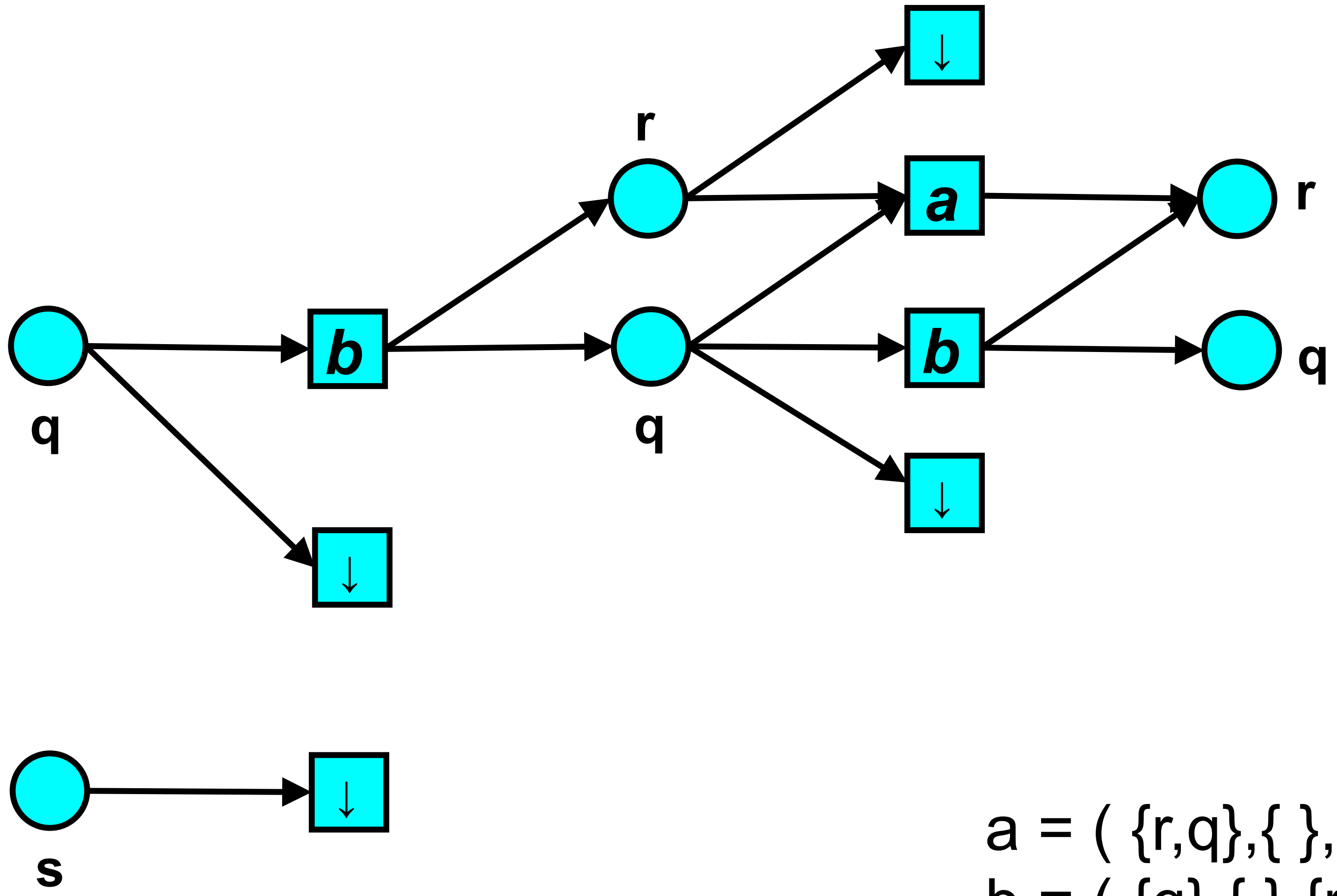
•*the net simulates $\mathcal{A}_1$*

•*one-to-one correspondence of the dynamic behaviours*

•*steps of transitions are sets*

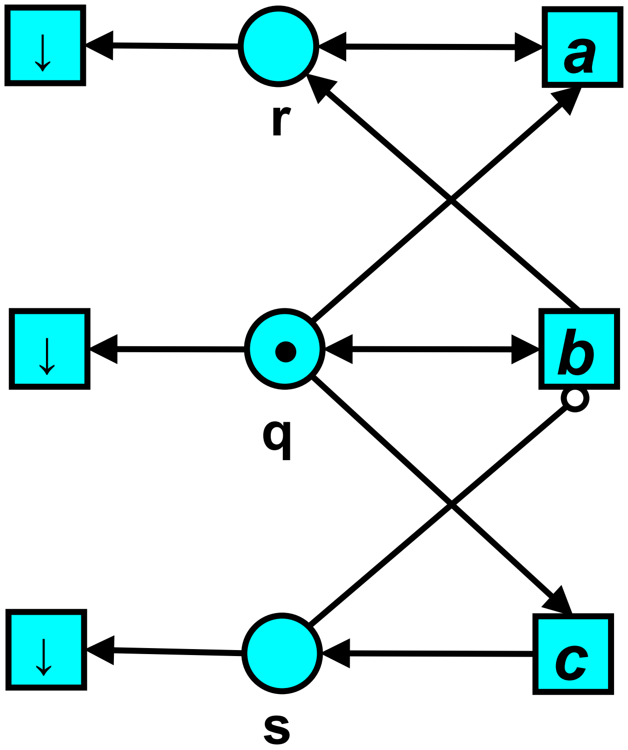
$\mathcal{A}_1$: a = ({r,q},{ },{r})
b = ({q},{ },{r,q})

Set-net processes of $\mathcal{A}_1$



a = ({r,q},{ },{r})
b = ({q},{ },{r,q})

Method V (extended)



$\mathcal{A}2$: a = ({r,q},{ },{r})
b = ({q},{s},{r,q})
c = ({q},{ },{s})

Outline

- Basic reaction systems
- Reaction systems and low-level Petri nets
- Reaction systems and high-level Petri nets
- Set-nets
- Reaction systems and set-nets
- Concluding remarks

Concluding remarks

- Reaction systems *inspired* new class of Petri nets
- Set-nets are a *fundamental* Petri net model
- Set-nets can bring Petri net *analytical methods* to reaction systems

Concluding remarks

- Reaction systems *inspired* new class of Petri nets
- Set-nets are a *fundamental* Petri net model
- Set-nets can bring Petri net *analytical methods* to reaction systems
- BUT one first needs to *re-cast* Petri net methods in terms of set-nets

Concluding remarks

- Reaction systems *inspired* new class of Petri nets
- Set-nets are a *fundamental* Petri net model
- Set-nets can bring Petri net *analytical methods* to reaction systems
- BUT one first needs to *re-cast* Petri net methods in terms of set-nets

... for example ...

Concluding remarks

- Two types of subnets:

  siphon - once empty remains empty
  trap – once marked remains marked

- *"A Structural Theorem [Hack 1970s]"*

  If every siphon contains a marked trap then a P/T-net is deadlock-free

  Numerous applications/extensions/structure theory

- *"Another Structural Theorem"*

  If every siphon contains a marked trap then a set-net is deadlock-free

Concluding remarks

Future work:

process semantics (causality)

analytical techniques

synthesis (initially, region based as in ART'11)

...

Thank you!