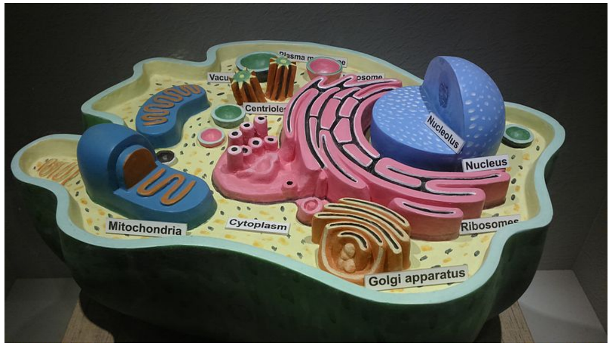
Graph Surfing by Reaction Systems

Hans-Jörg Kreowski and Grzegorz Rozenberg

set-based reaction systems introduced by
Ehrenfeucht and Rozenberg in 2007



Structure of animal cell

Author: Royroydeb

to model the functioning of living cells

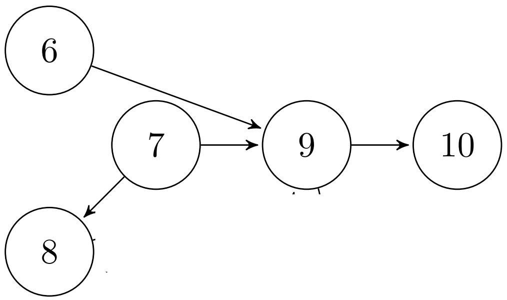
and as a new approach to computation

in this paper:
graph-based reaction systems
as a new approach to graph transformation

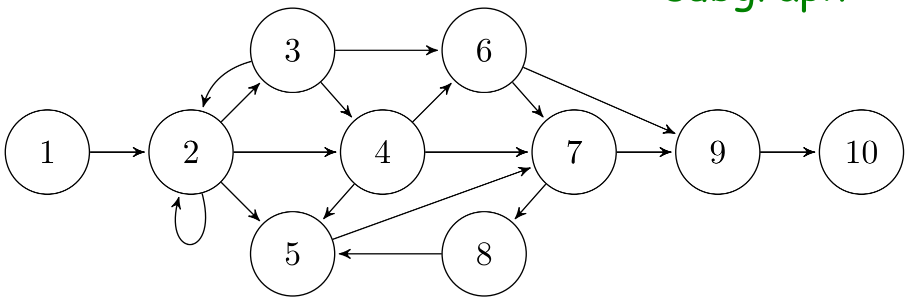
Reaction Systems

	background	state
set	B	$T \subseteq B$
graph	B	T sub B

subgraph



sub

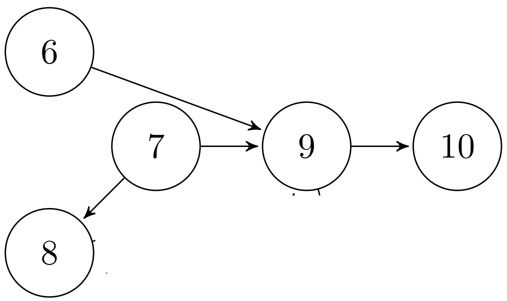


Reaction Systems

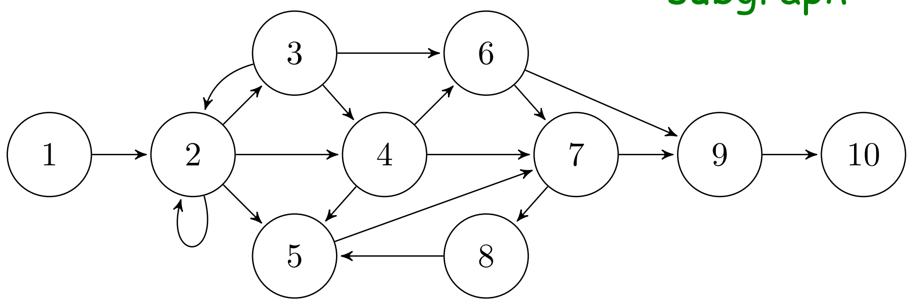
	background	state
set	B	$T \subseteq B$
graph	B	T sub B

finite

subgraph



sub



Reaction Systems

reaction b

	reactant	inhibitor	product
set	R ⊆ B	I ⊆ B	P ⊆ B
graph	R sub B	I sub B (?!)	P sub B

Reaction Systems

reaction **enabled** on state

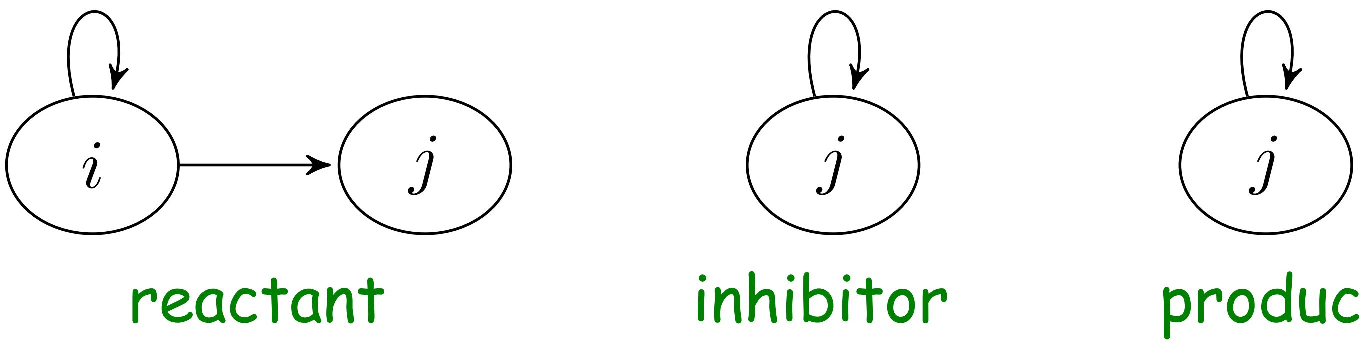
set $R \subseteq T$ and $I \cap T = \emptyset$

graph R sub T and $I \cap T = \emptyset$ (?!)

Reaction Systems

subgraph as inhibitor is inconvenient:

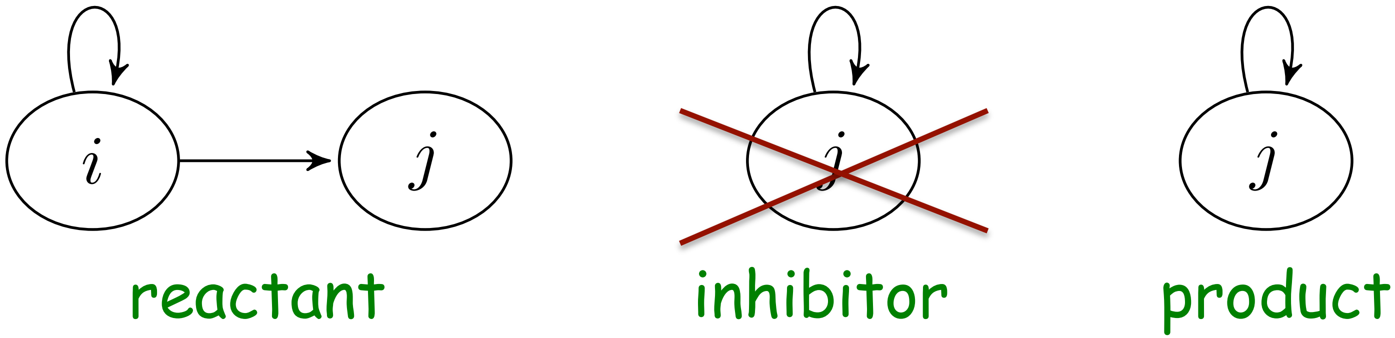
for example, loop to be moved along edge
but only if there is no loop on target



Reaction Systems

subgraph as inhibitor is inconvenient:

for example, loop to be moved along edge
but only if there is no loop on target



we allow to forbid an edge
without forbidding source and target necessarily

Reaction Systems

reaction b

	reactant	inhibitor	product
set	$R \subseteq B$	$I \subseteq B$	$P \subseteq B$
graph	R sub B	~~I sub B~~	P sub B

$I \subseteq (V_B, E_B) = U(B)$

selector — extraction

Reaction Systems

reaction **enabled** on state

set $R \subseteq T$ and $I \cap T = \emptyset$

graph R sub T and $I \cap U(T) = (\emptyset, \emptyset)$

Reaction Systems

result of reaction on state

set/graph $res_b(T) = \begin{cases} P & \text{if } b \text{ enabled on } T \\ \emptyset & \text{otherwise} \end{cases}$

result of set A of reactions on state

set/graph $res_A(T) = \bigcup_{b \in A} res_b(T)$

Reaction Systems

$$\mathcal{A} = (B, A)$$

background set/graph — set of reactions

result of $\mathcal{A}$ on state

$$res_{\mathcal{A}}(T) = res_{A}(T)$$

First case study

modeling a shortest path algorithm by the graph-based reaction system

SHORT(n)

(as parallel breadth first)

background graph of SHORT(n)

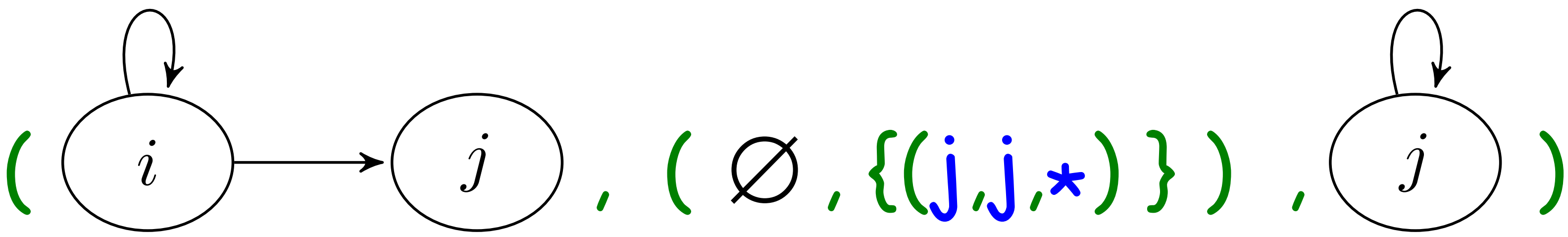
the complete directed graph with

nodes 1,2, ... ,n and

edges (i,j,*) for i,j = 1, ... ,n

where * is a special label invisible in drawings

reactions of SHORT(n) for all i,j = 1,...,n

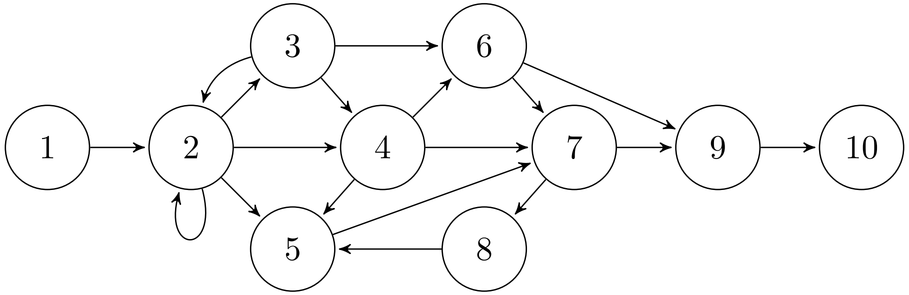


loop is moved along edge if there is no loop on target

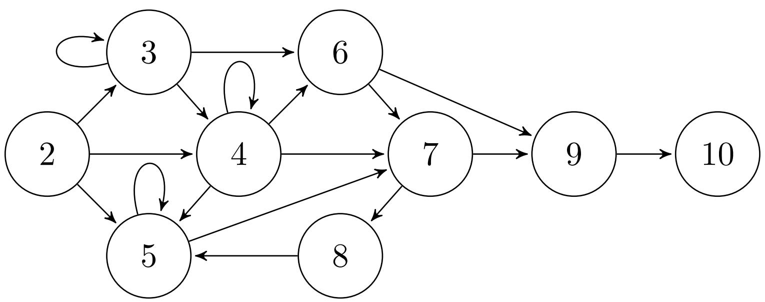
edge is sustained if there is no loop at target

sample sequence of states in SHORT(10)

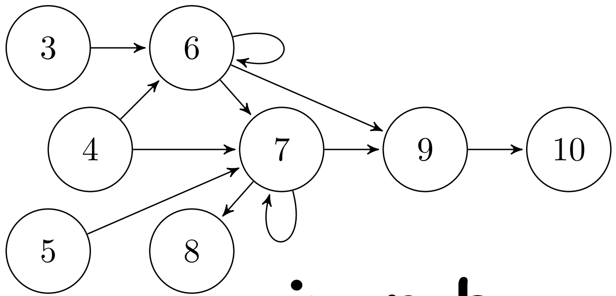
T_0 =



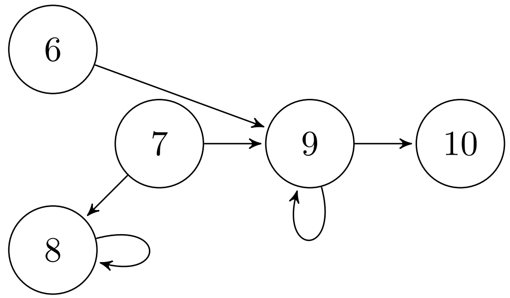
T_1 =



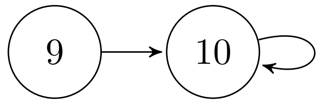
T_2 =



T_3 =



T_4 =



T_5 = $\emptyset$

given by a sequence of reactions

Theorem

Let T_0 be a state with a single loop at node i_0, and let $T_0, \ldots, T_m$ be a sequence of states in SHORT(n) such that $T_i = res_{SHORT(n)}(T_{i-1})$ for $i = 1, \ldots, m$.

Then the node j has a loop in T_k for some k if and only if k is the length of a shortest path from i_0 to j in T_0 .

Reaction Systems

$$\mathcal{A} = (\,B\,,\,A\,)$$

background set/graph set of reactions

result of $\mathcal{A}$ on state

$$res_{\mathcal{A}}(T) = res_{A}(T)$$

deterministic # maximum parallel # no conflict

nothing sustains if not (re)produced

Reaction Systems

forcing sustainability

Let S sub B, and let A contain the (uninhibited) reactions

$$(v , (\varnothing, \varnothing) , v) \text{ for } v \in V_S, \text{ and}$$

$$(e^\bullet, (\varnothing, \varnothing) , e^\bullet) \text{ for } e \in E_S.$$

Then $S \cap T \subseteq res_A(T)$.

Second case study

modeling a finite state automaton $\mathcal{F}$ and its recognition of strings by the graph-based reaction system

$$\mathcal{A}(\mathcal{F})$$

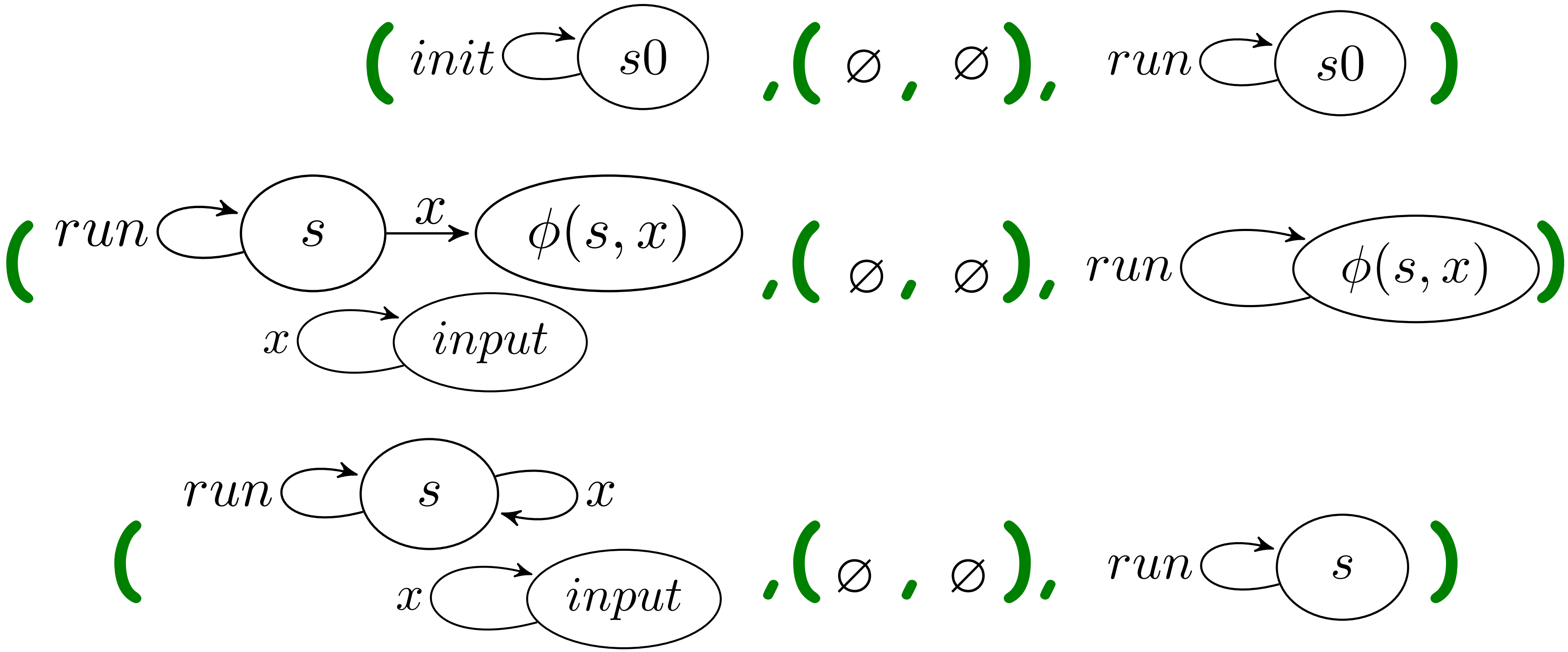
(using mainly the state graph)

Let $\mathcal{F} = (Q, I, \phi, s_0, F)$ be a finite state automaton.

Then the background graph of $\mathcal{A}(\mathcal{F})$ consists of the state graph $gr(\mathcal{F})$ of $\mathcal{F}$ plus a run-loop at each node plus an extra node input with an x-loop for each x in I.

φ

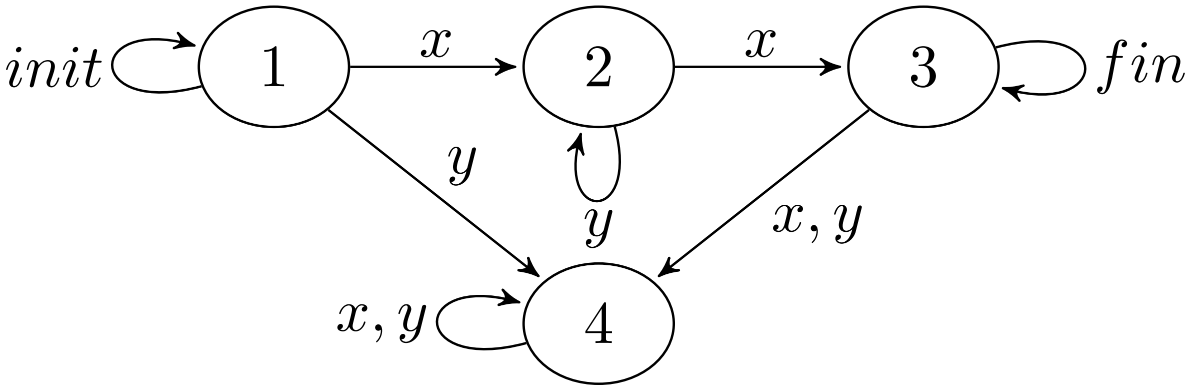
The reactions of $\mathcal{A}(\mathcal{F})$ are:



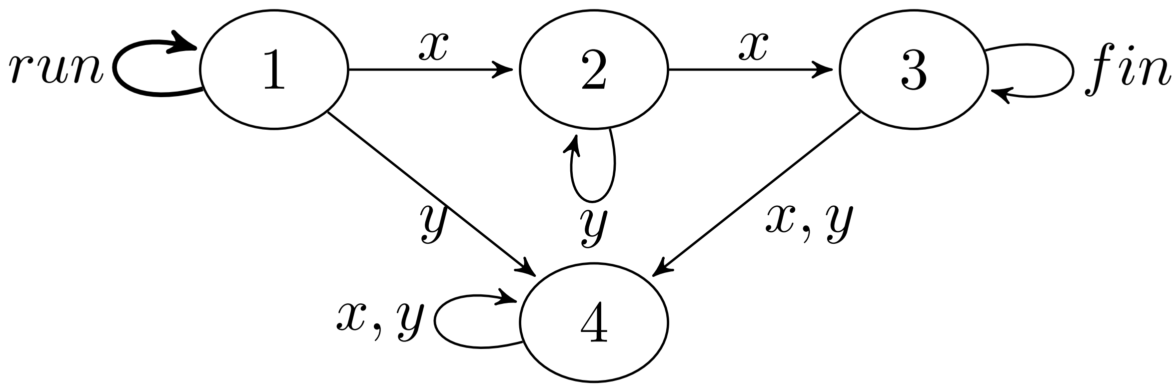
plus the reactions sustaining the state graph gr($\mathcal{F}$) up to the init-loop

result graphs

(0)

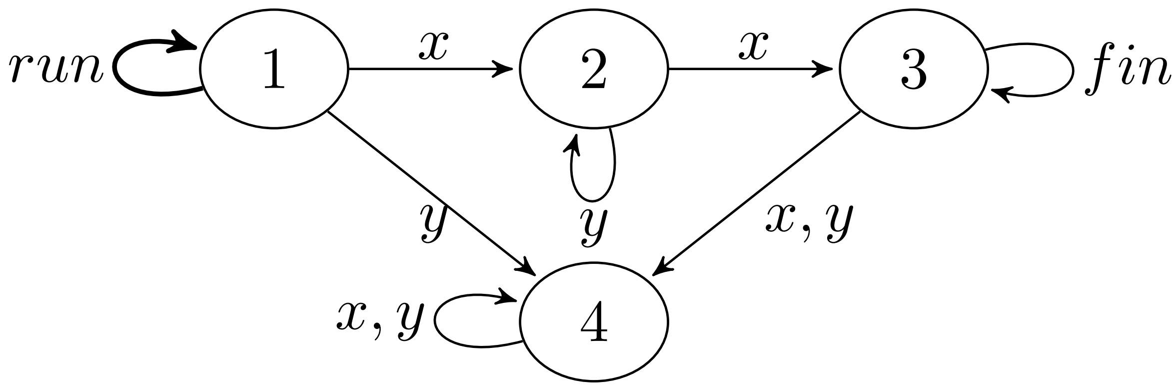


(1)

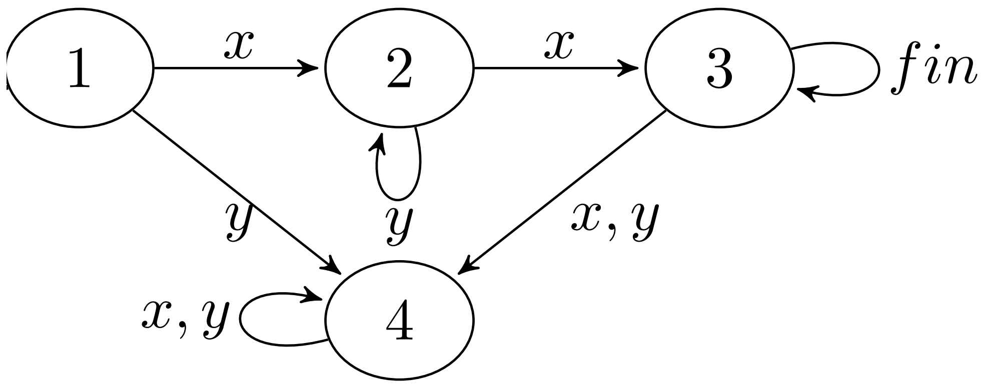


result graphs

(1)

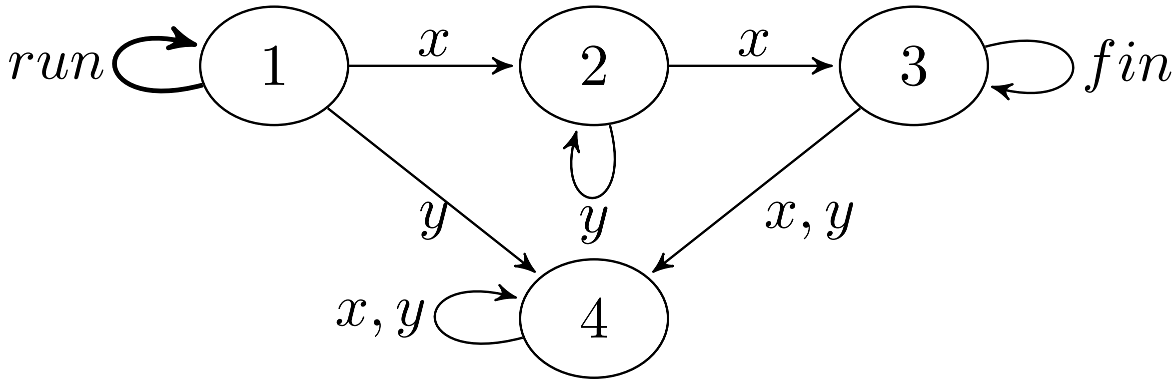


(2?)



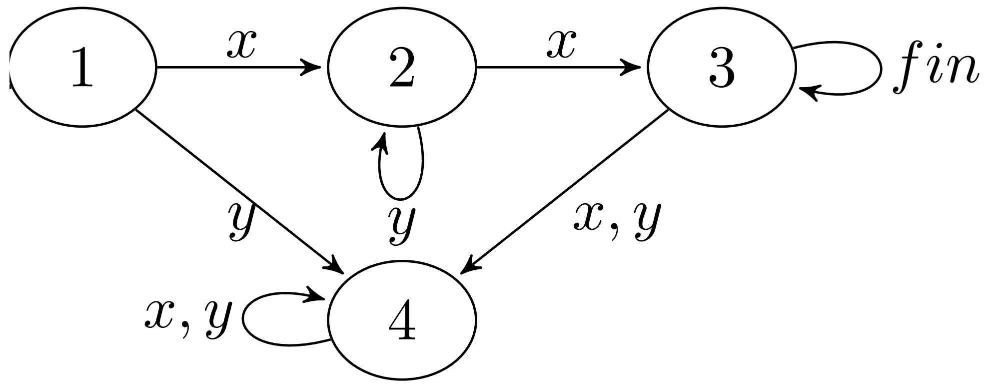
result graphs

(1)



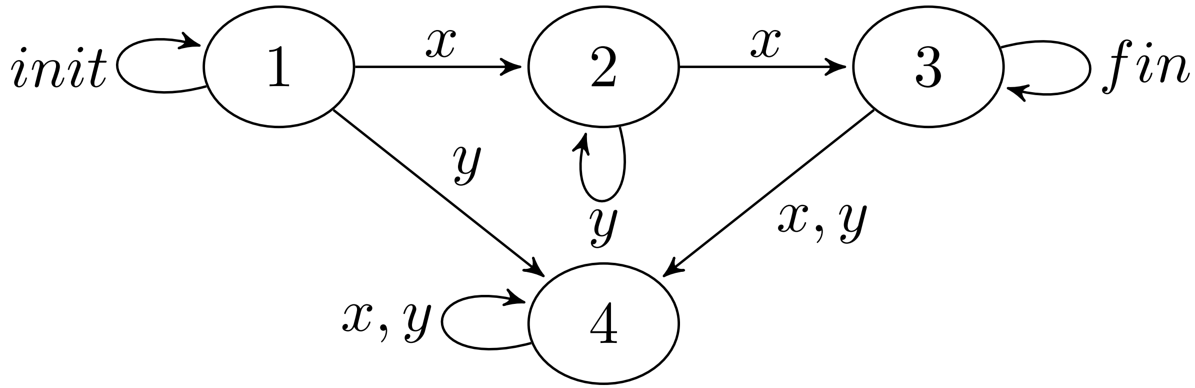
(2?)

stable
from
here on

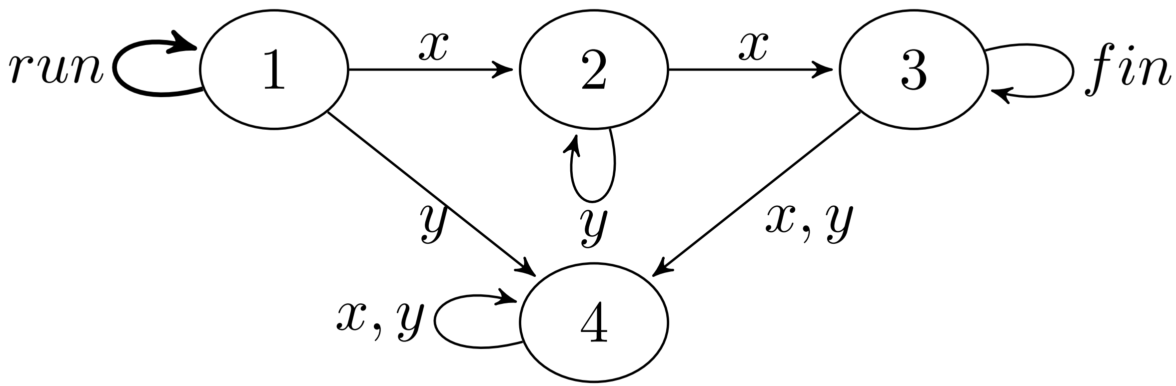


result graphs

(0)

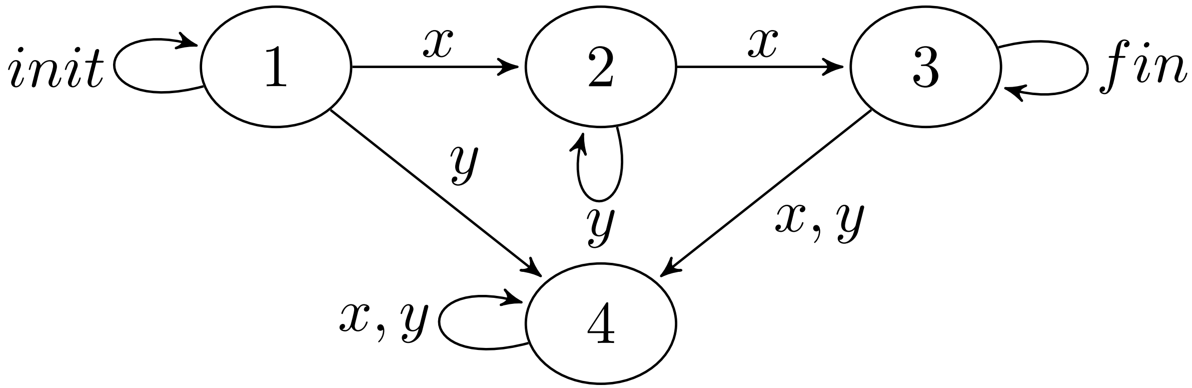


(1)

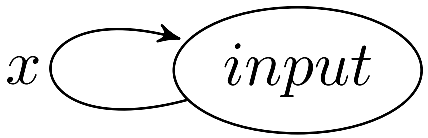


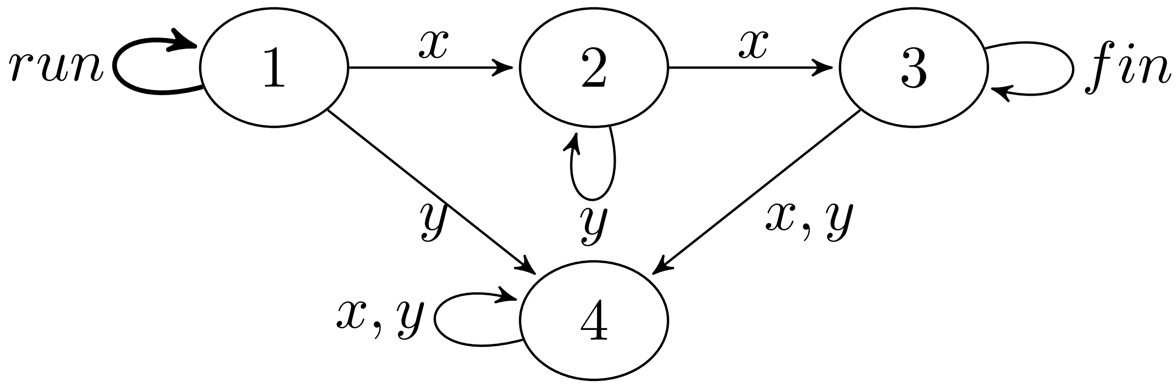
context graphs result graphs

(0)



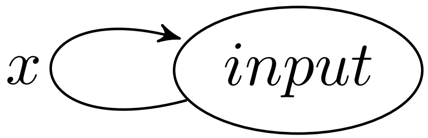
(1)

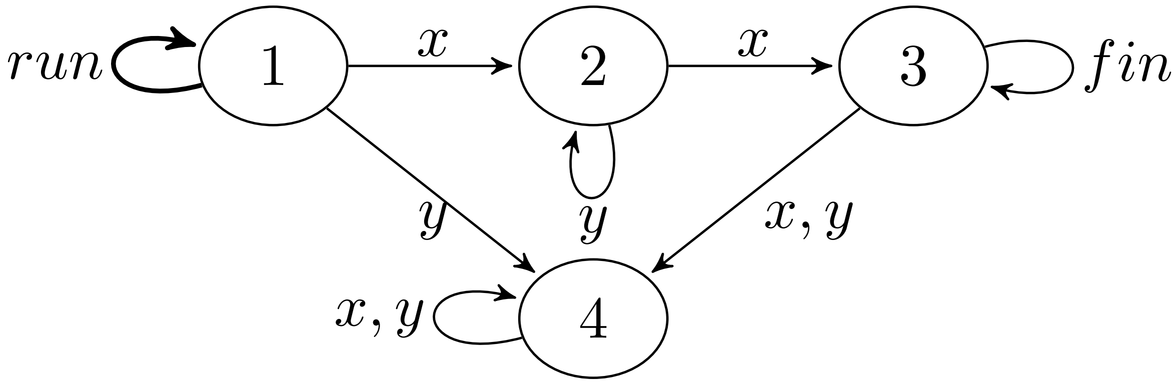




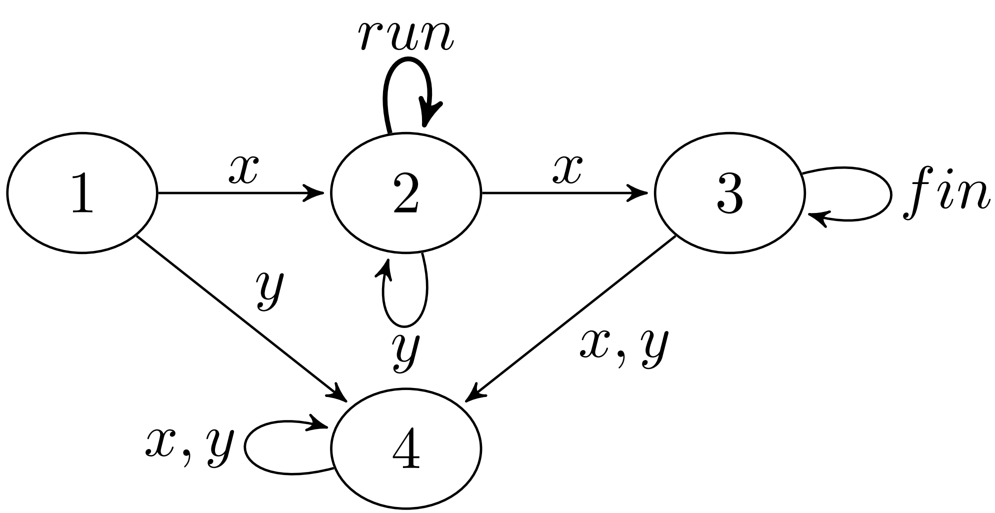
context graphs   result graphs

(1)



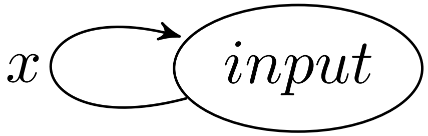


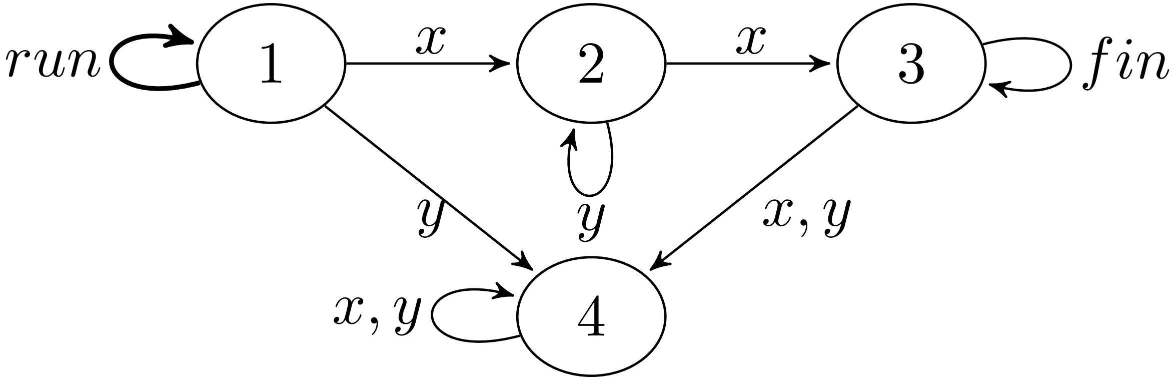
(2)



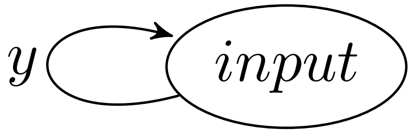
context graphs   result graphs

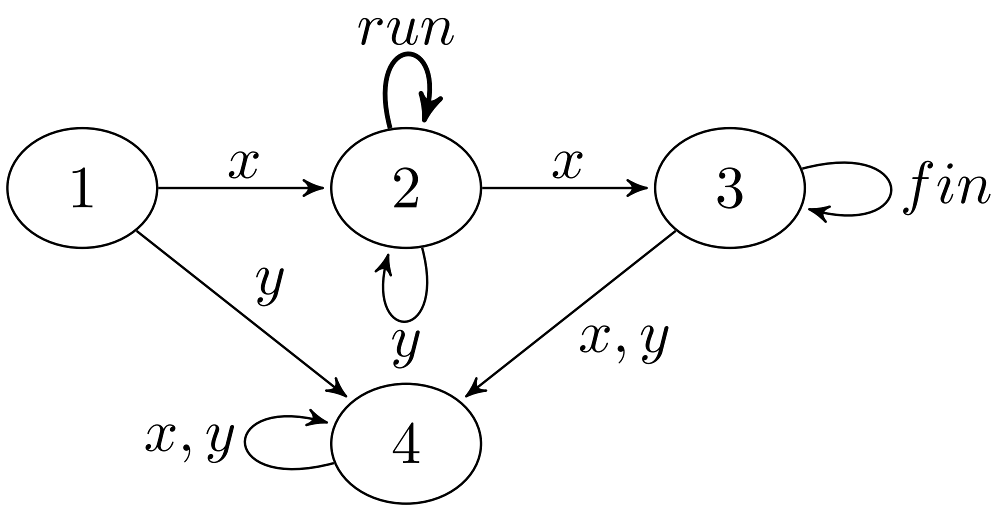
(1)

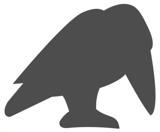




(2)

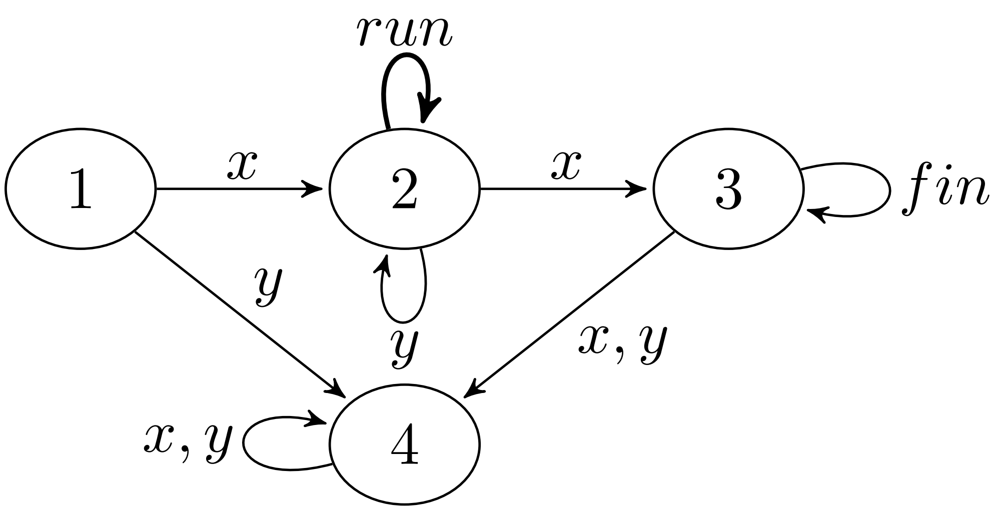




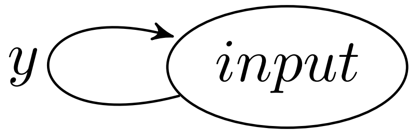


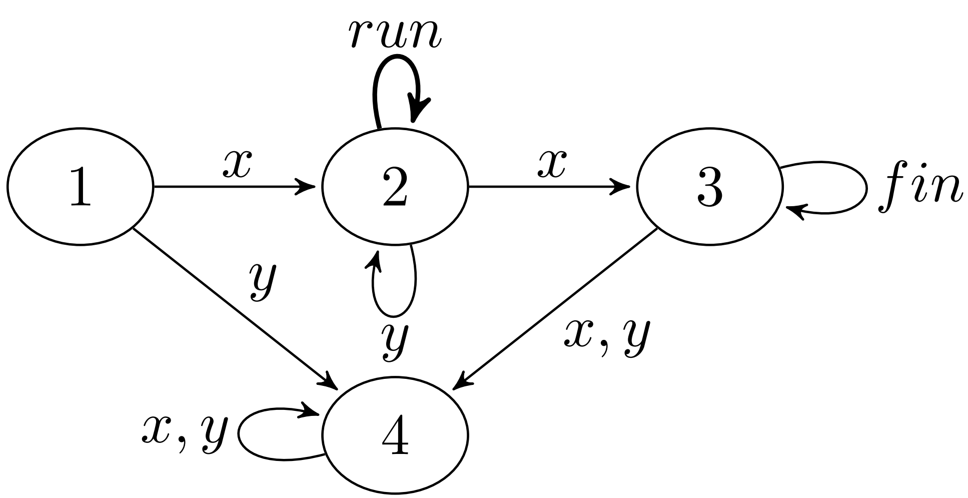
context graphs

result graphs



(2)

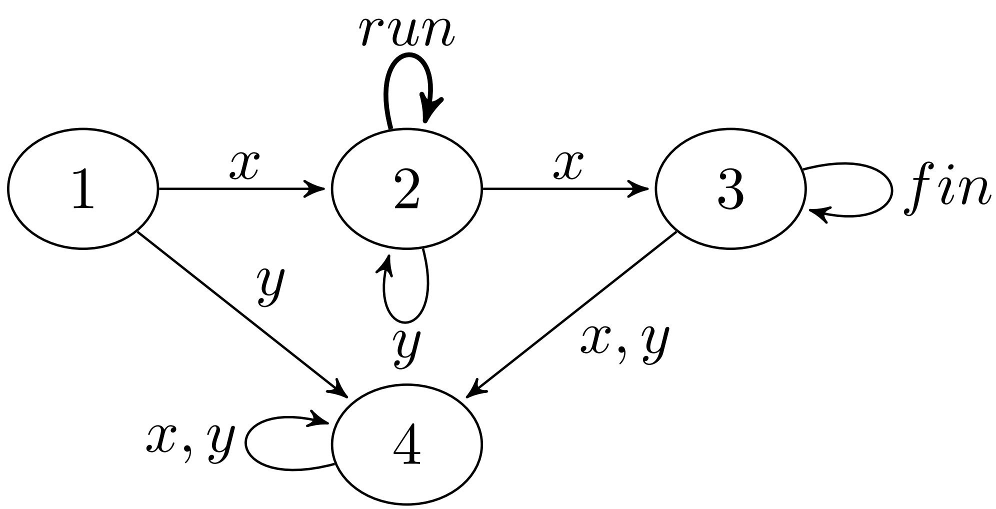




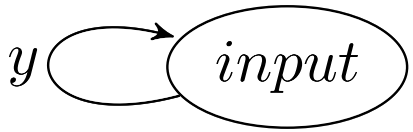
(3)

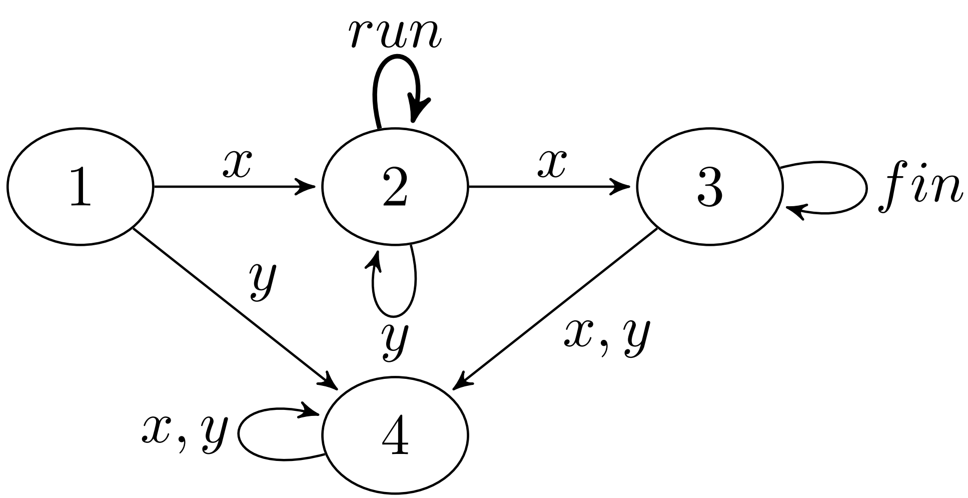
context graphs

result graphs

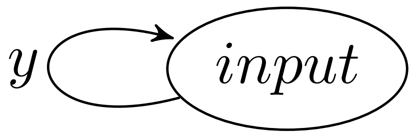


(2)





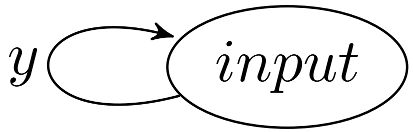
(3)

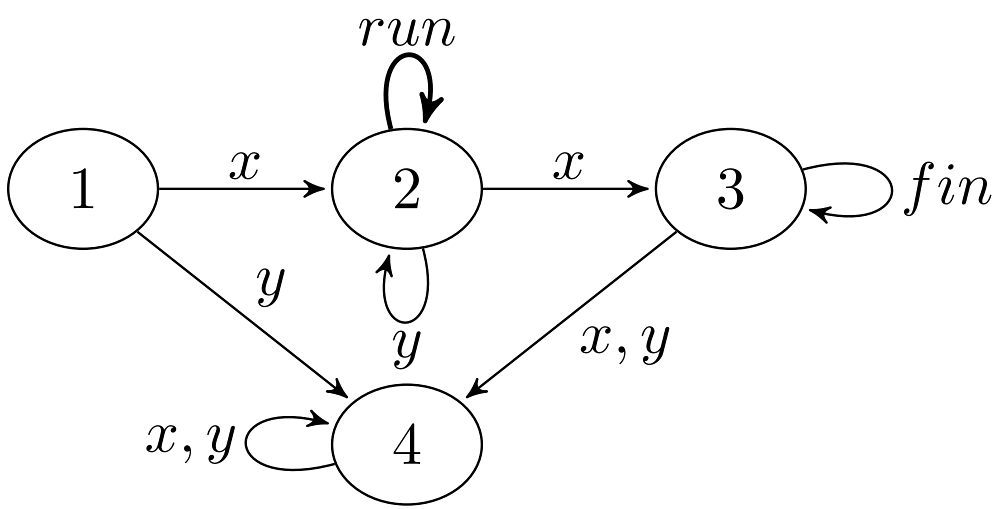


context graphs

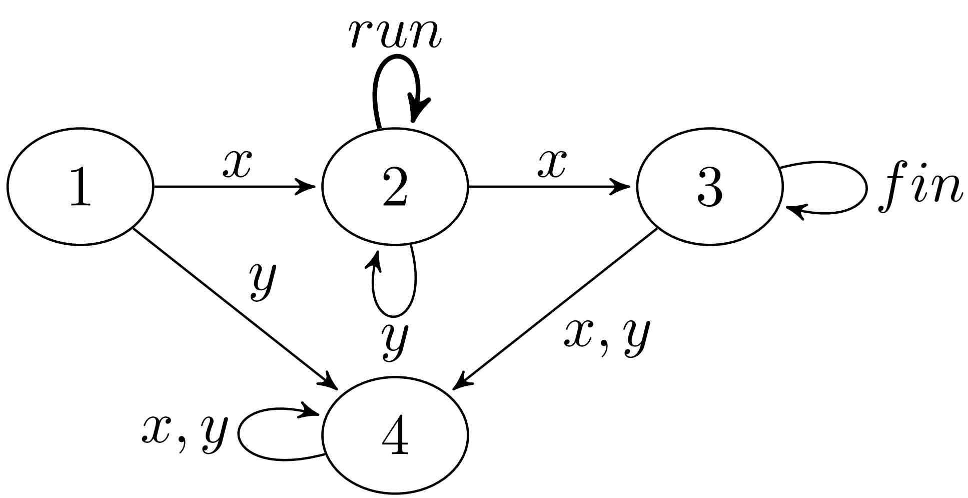
result graphs

(3)



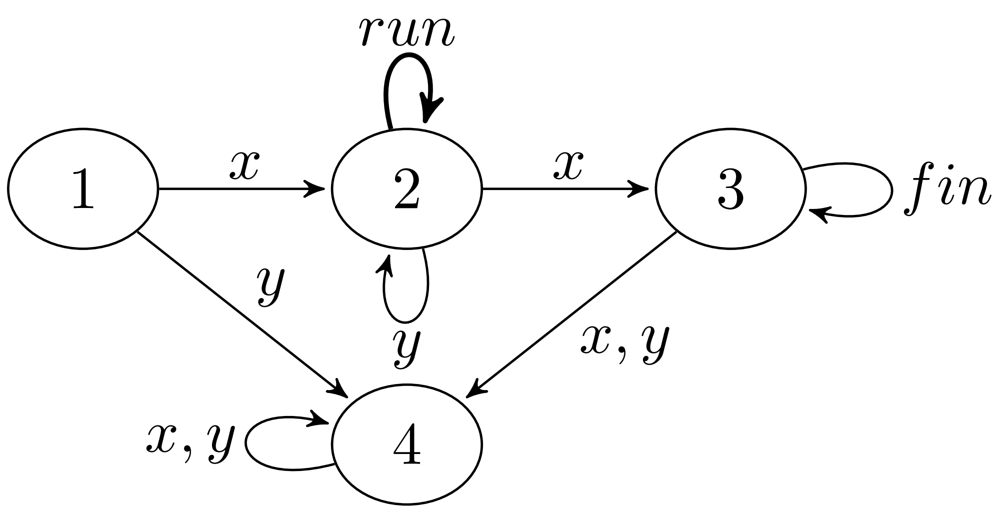


(4)

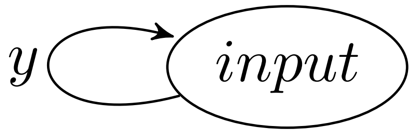


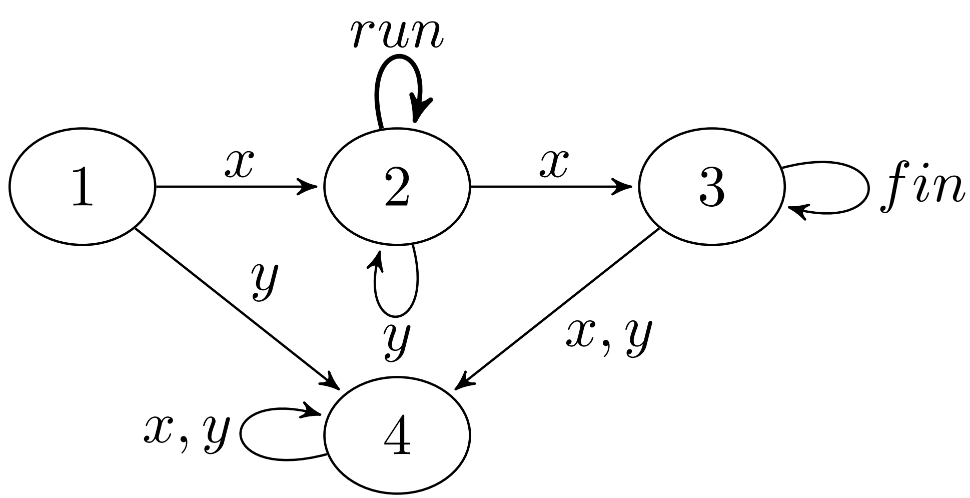
context graphs

result graphs

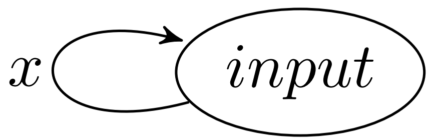


(3)





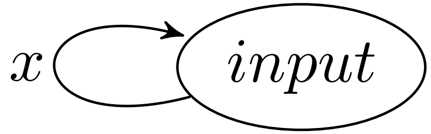
(4)

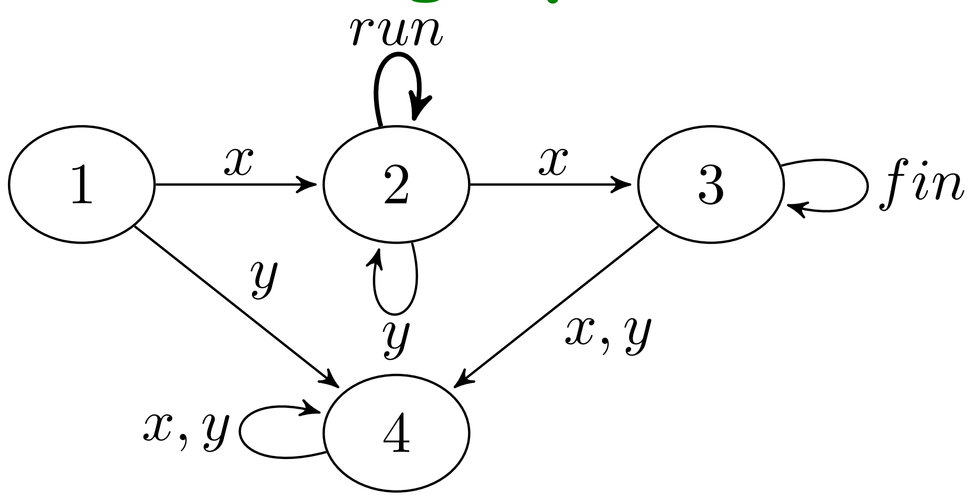


context graphs

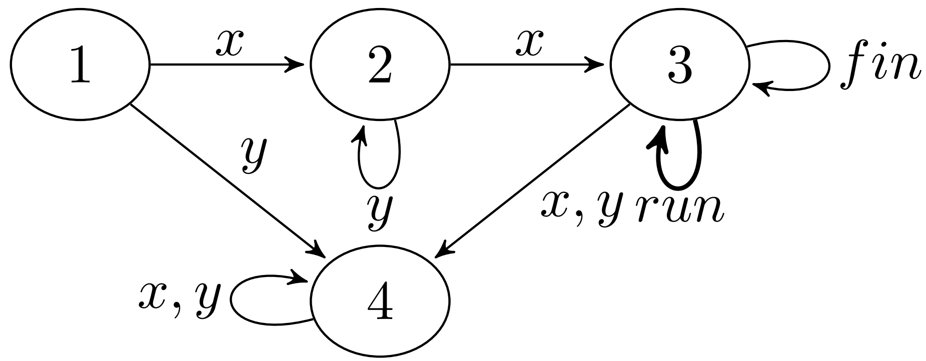
result graphs

(4)

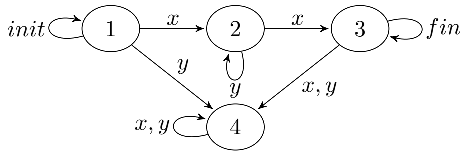




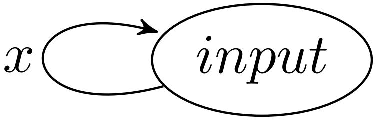
(5)

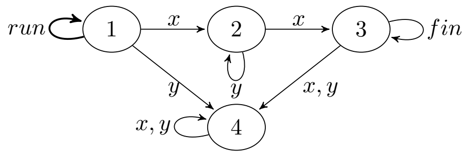


(0) $\varnothing$

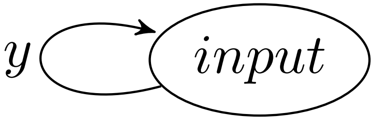


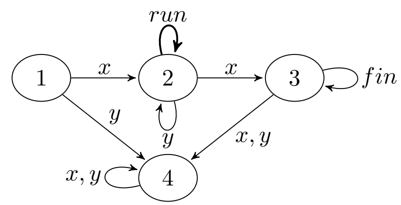
(1)



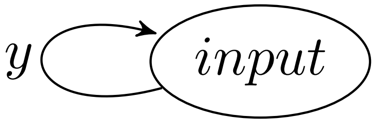


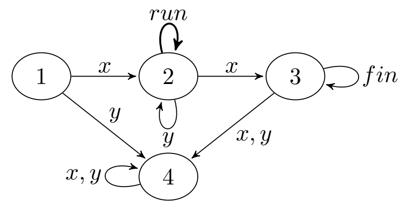
(2)



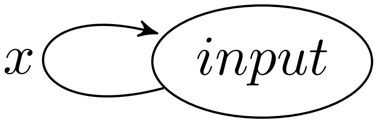


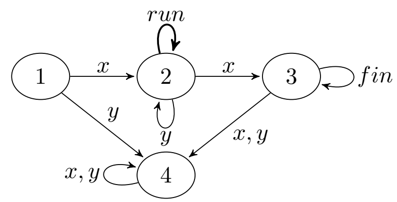
(3)

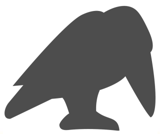




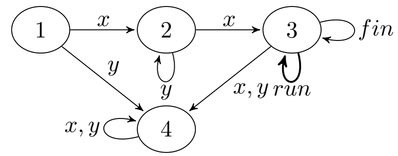
(4)



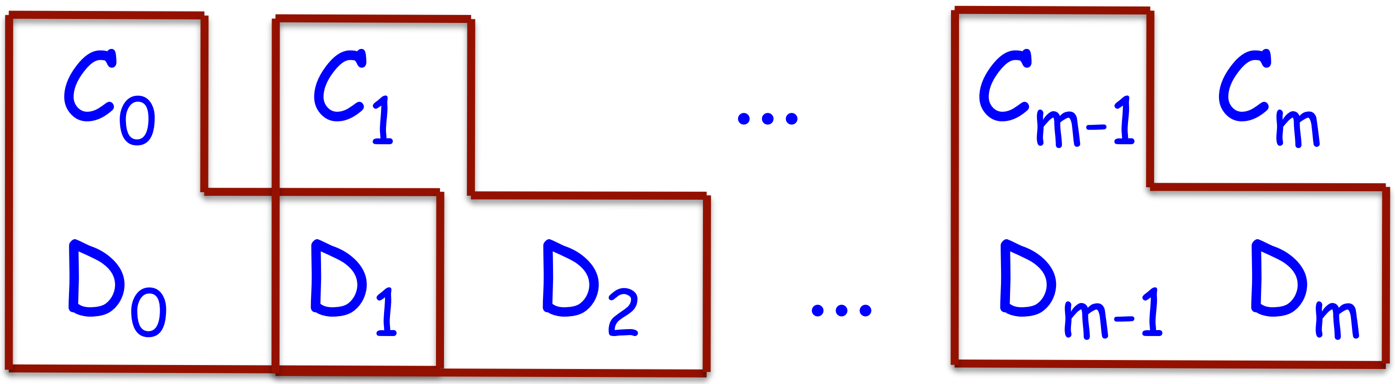




(5) $\varnothing$



An **interactive process** is a pair $\pi = (\gamma, \delta)$ where $\gamma = C_0, \ldots, C_m$ and $\delta = D_0, \ldots, D_m$ are two sequences of graphs for some m such that $D_i = res_{\mathcal{A}}(C_{i-1} \cup D_{i-1})$ for $i = 1, \ldots, m$.



An interactive process is a pair $\pi = (\gamma, \delta)$ where $\gamma = C_0, \ldots, C_m$ and $\delta = D_0, \ldots, D_m$ are two sequences of graphs for some m such that $D_i = res_{\mathcal{A}}(C_{i-1} \cup D_{i-1})$ for $i = 1, \ldots, m$.

γ is the **context sequence** of π,
δ its **result sequence**, and
$\tau = T_0, \ldots, T_m$ with $T_i = C_i \cup D_i$ for $i = 0, \ldots, m$ is its **state sequence**.

An interactive process is a pair $\pi = (\gamma, \delta)$ where $\gamma = C_0, \ldots, C_m$ and $\delta = D_0, \ldots, D_m$ are two sequences of graphs for some m such that $D_i = res_{\mathcal{A}}(C_{i-1} \cup D_{i-1})$ for $i = 1, \ldots, m$.

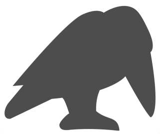
π is the **context-independent** if C_i sub D_i for $i = 0, \ldots, m$. In this case, $\delta = \tau$, and $\gamma = \varnothing, \ldots, \varnothing$ wlog.

consider the interactive processes $\pi(x_1 \dots x_n)$ given by context sequences of the form

$$\emptyset, \mathrm{loop}(\mathrm{input}, x_1), \dots, \mathrm{loop}(\mathrm{input}, x_n), \emptyset$$

for some n and $x_i \in \Sigma$ for $i = 1, \dots, n$, and the state graph of $\mathcal{F}$ as initial result graph

then the language $L(\mathcal{A}(\mathcal{F}))$ specified by $\mathcal{A}(\mathcal{F})$ contains all strings $x_1 \dots x_n$ such that the last result graph of $\pi(x_1 \dots x_n)$ has a node with a run- and a fin-loop



Theorem: $L(\mathcal{A}(\mathcal{F})) = L(\mathcal{F})$

Third case study

modeling cellular automata that form one of the oldest paradigms of massively parallel computation with a rich stock of theory and applications

cellular automata

A cellular automaton is a system

$$\mathcal{C} = (COL, w, k, \phi, CELL, \mathcal{N})$$

- COL: finite set of colors
- w: special default color
- k: neighbor-hood size
- ϕ: transition function from COL^{k+1} to COL subject to the condition $\phi(w, \ldots, w) = w$ for the default color w
- $CELL$: set of cells
- $\mathcal{N}$: neighborhood specification consisting of a neighborhood function $\mathcal{N}_i : CELL \to CELL$ for each $i = 1, \ldots, k$

cellular automata

A cellular automaton is a system

$$\mathcal{C} = (COL\ ,\ w\ ,\ k\ ,\ \phi,\ CELL\ ,\ \mathcal{N})$$

- COL: finite set of colors
- w: special default color
- k: neighbor-hood size
- ϕ: transition function from COL^{k+1} to COL subject to the condition $\phi(w, \ldots ,w) = w$ for the default color w
- $CELL$: set of cells (may be infinite)
- $\mathcal{N}$: neighborhood specification consisting of a neighborhood function $\mathcal{N}_i: CELL \rightarrow CELL$ for each $i = 1,\ldots,k$

cellular automata

A configuration of C is a function
α: $CELL \rightarrow COL$ such that
the set of active cells
$act(\alpha) = \{v \in CELL \mid \alpha(v) \neq w\}$ is finite.

$\in$

Given a configuration α, one gets
a uniquely determined successor configuration
α' : $CELL \rightarrow COL$ defined by
$\alpha'(v) = \phi(\alpha(v), \alpha(N_1(v)), \ldots, \alpha(N_k(v)))$
for each $v \in CELL$.

Example

transition function:

(blue|red,c_1,...,c_6) $\mapsto$ black in all cases

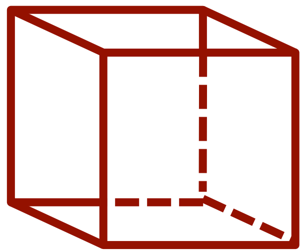
(black,c_1,...,c_6) $\mapsto$ blue if one c_i is red
and the others black

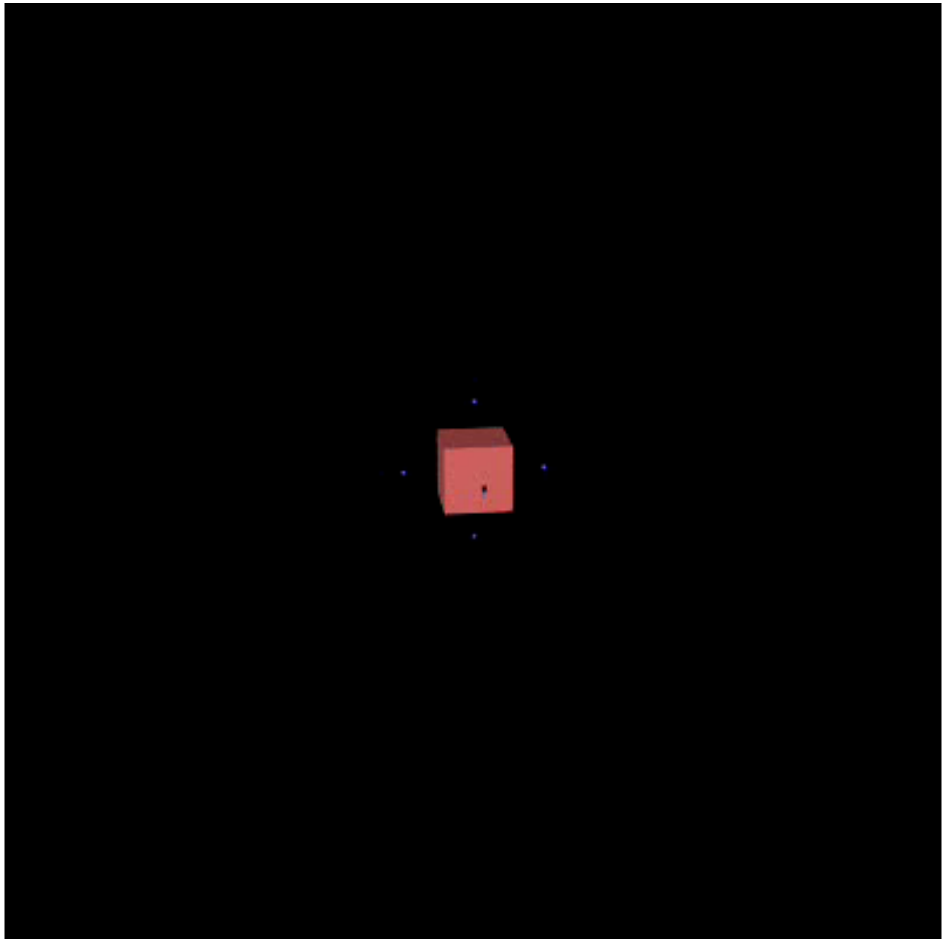
(black,c_1,...,c_6) $\mapsto$ red if one c_i is blue
and the others black

(black,c_1,...,c_6) $\mapsto$ black in all other cases

cells: unit cubes in the Euclidean 3d space with integer coordinates

neighbors: the cubes that share a side





transformation of cellular automata into graph-based reaction systems

Let $\mathcal{C} = (COL, w, k, \phi, CELL, \mathcal{N})$ be a cellular automaton.
Then, for each finite subset $Z \subseteq CELL$,
the cells of interest, a graph-based reaction system

$$\mathcal{A}(\mathcal{C},Z) = (B(\mathcal{C},Z) = (V,\Sigma,E), A(\mathcal{C},Z))$$

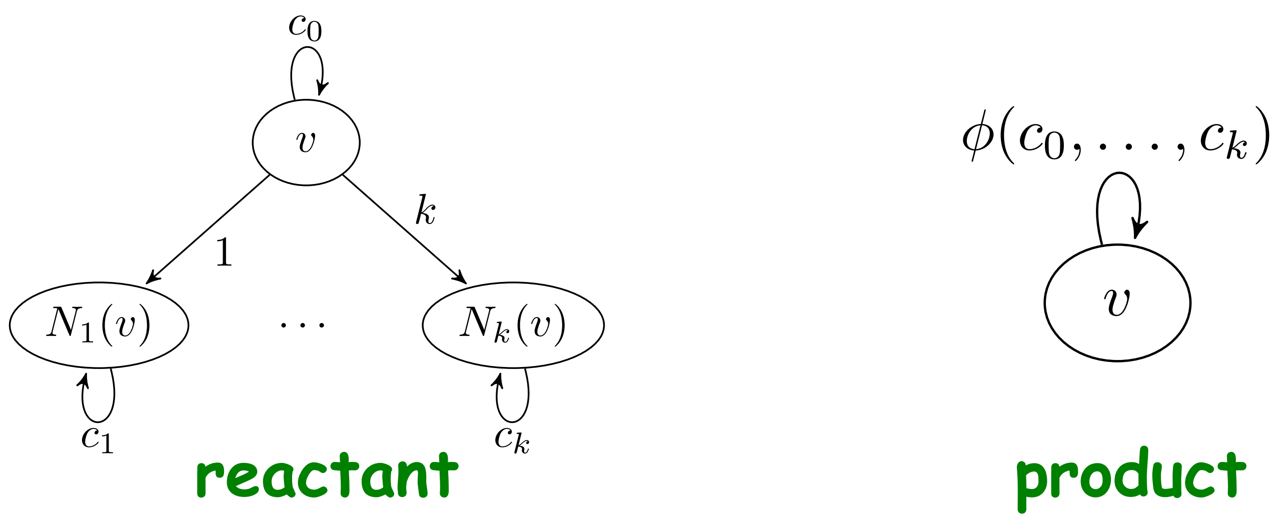
is defined as follows:

$$V = Z \cup \{\mathcal{N}_i(v) \mid v \in Z,\ i = 1,\ldots,k\},$$

$$\Sigma = COL \cup \{1,\ldots,k\} \cup \{*\}, \text{ and}$$

$$E = \{(v,\mathcal{N}_i(v),i) \mid v \in Z,\ i = 1,\ldots,k\} \cup \{(v,v,c) \mid v \in Z,\ c \in COL\} \cup \{(v,v,w) \mid v \in V \setminus Z\}.$$

The set of reactions A(C,Z) consists of the following uninhibited reactions



for all $v \in Z$ and $c_0, \dots, c_k \in COL$
plus sustaining rules for the neighborhood structure of all $v \in Z$ and the w-loops of all $v \in V \setminus Z$

Let $\mathcal{C}$ = (COL, w, k, ϕ, CELL, $\mathcal{N}$) be a cellular automaton *and* $\mathcal{A}(\mathcal{C}, Z)$ be the corresponding reaction system with respect to some finite $Z \subseteq$ CELL.

Let T_{lab} be a well-formed state, i.e. all nodes, all neighborhood edges as well as one loop at each node labeled due to some labeling function lab: Z → COL and w-loops at all $v \in V \backslash Z$.

Note: The successor state of a well-formed state is well-formed.

Let $T_{lab0} \to \cdot \; \cdot \; \cdot \to T_{labn}$ be the state sequence of the context-independent process starting in some T_{lab0}

Let $\mathcal{C}$ = (COL, w, k, ϕ, CELL, $\mathcal{N}$) be a cellular automaton *and* $\mathcal{A}(\mathcal{C}, Z)$ be the corresponding reaction system with respect to some finite $Z \subseteq$ CELL.

For lab: $Z \to$ COL, let α(lab) be the configugation with α(lab)(v) = lab(v) for v $\in$ Z and α(lab)(v) = w otherwise

Let $\alpha(lab_0) = \alpha_0 \to \alpha_1 \to \ldots \to \alpha_n$ be a computation in $\mathcal{C}$ subject to the condition $act(\alpha_i) \subseteq Z$ for $i = 1,\ldots,n$.

Theorem: Then $\alpha_i = \alpha(lab_i)$ for $i = 1,\ldots,n$.

Discussion

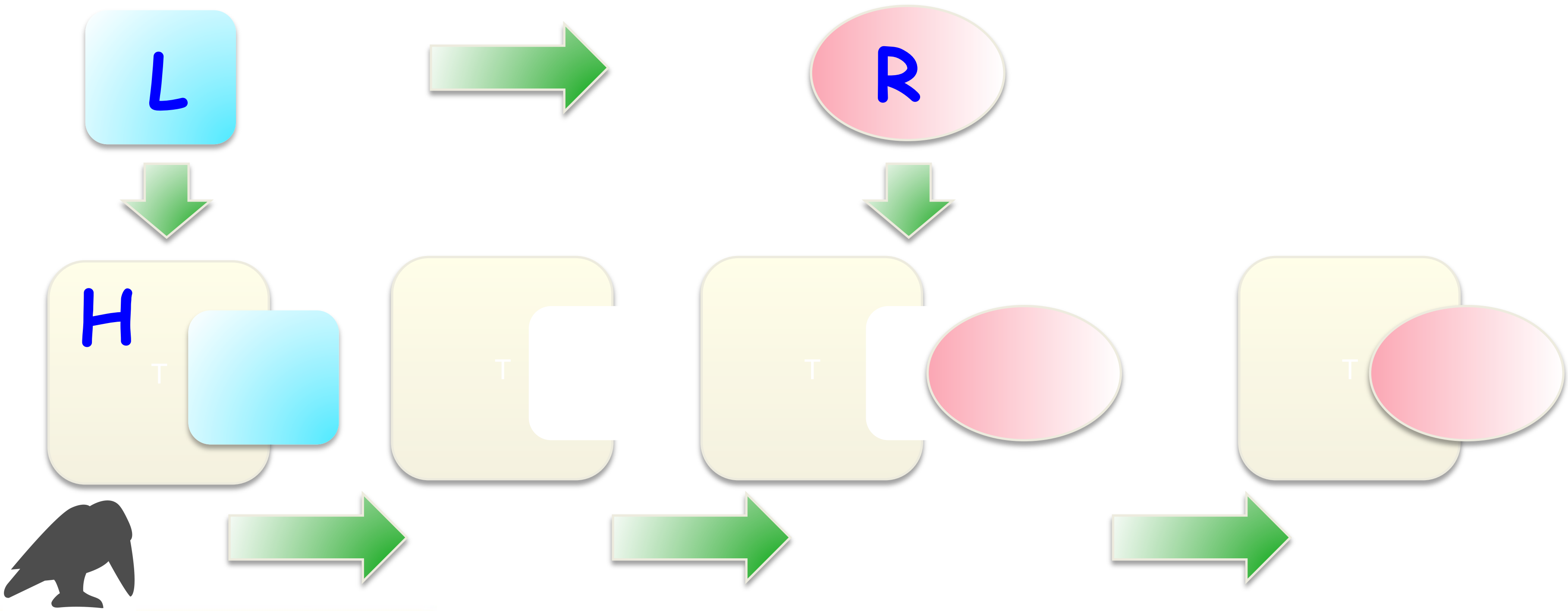
graph-based reaction systems generalize set-based ones

providing a graph-processing and visual level to the framework

the three case studies demonstrate the modeling capacity including the chance of proving correctness

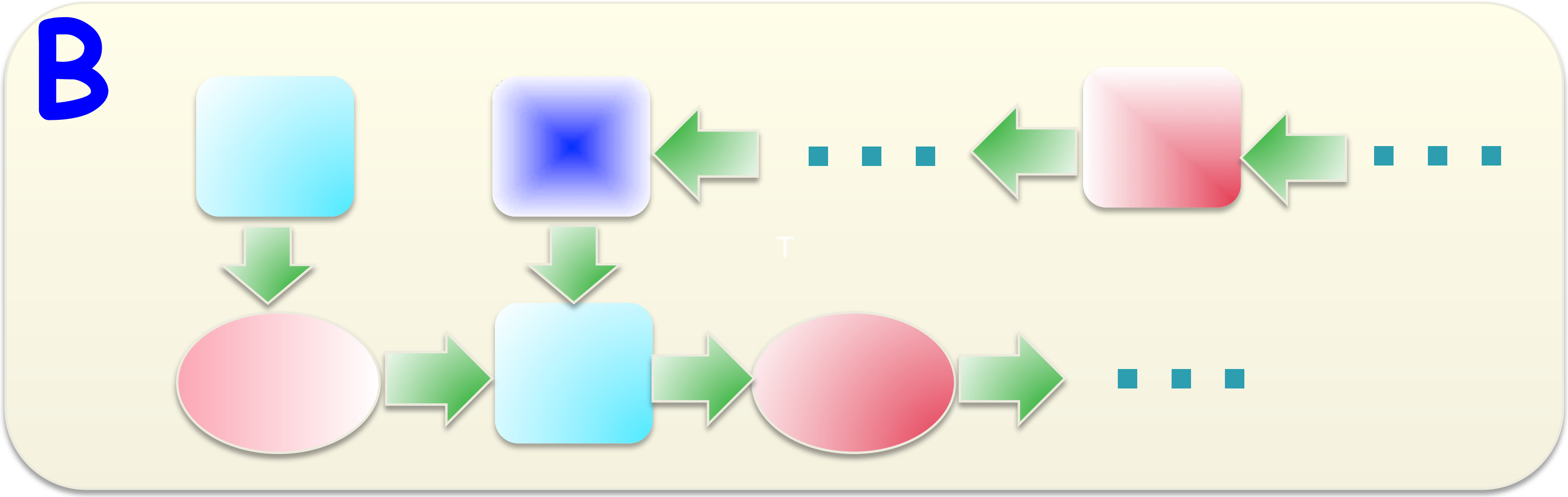
Discussion

most known approaches to graph transformation follow the match-cut-add-paste methodology



Discussion

in contrast to this, graph-based reaction systems provide a „surfing" methodoly



Discussion

surfing on the background entity yields sequences of states

this has become a favorite research topic in the set case recently

we expect this to be a very promising future research topic in the graph case
because graphs offer more structure than sets

thank you for your attention