UNIVERSITÀ DEGLI STUDI DI TRIESTE

Natural Computing Laboratory

Reaction Systems and Their Dynamics

Luca Manzoni

Università degli Studi di Trieste

Table of Contents

Why Rection Systems?

Reaction systems are a computational model inspired by *bio-chemical reactions*.

Why Rection Systems?

Reaction systems are a computational model inspired by *bio-chemical reactions*.

Why another bio-inspired model?

Why Rection Systems?

Reaction systems are a computational model inspired by *bio-chemical reactions*.

Why another bio-inspired model?

- A model abstract enough that is of theoretical interest. . .

Why Rection Systems?

Reaction systems are a computational model inspired by *bio-chemical reactions*.

Why another bio-inspired model?

- A model abstract enough that is of theoretical interest. . .
- . . .but still useful to model biological processes

Example of Application

Ion Petre et al. have studied the *the eukaryotic heat shock response*[1]

[1] Sepinoud Azimi, Bogdan Iancu, and Ion Petre. *Reaction system models for the heat shock response*. Fundamenta Informaticae, 131(3):299–312, 2014.

Example of Application

Ion Petre et al. have studied the *the eukaryotic heat shock response*[1]

The heat shock response is a defense mechanism by which the cell reacts to elevated temperatures

[1] Sepinoud Azimi, Bogdan Iancu, and Ion Petre. *Reaction system models for the heat shock response*. Fundamenta Informaticae, 131(3):299–312, 2014.

Example of Application

Ion Petre et al. have studied the *the eukaryotic heat shock response*[1]

The heat shock response is a defense mechanism by which the cell reacts to elevated temperatures

They have reformulate the existing model in terms of reaction systems and studied biologically relevant properties

[1] Sepinoud Azimi, Bogdan Iancu, and Ion Petre. *Reaction system models for the heat shock response*. Fundamenta Informaticae, 131(3):299–312, 2014.

Reactions

A *reaction* is a triple $a = (R, I, P)$ of finite sets.

Reactions

A *reaction* is a triple $a = (R, I, P)$ of finite sets.

- A set R of *reactants*

Reactions

A *reaction* is a triple $a = (R, I, P)$ of finite sets.

- A set R of *reactants*
- A set I of *inhibitors*

Reactions

A *reaction* is a triple $a = (R, I, P)$ of finite sets.

- A set R of *reactants*
- A set I of *inhibitors*
- A set P of *products*

Reactions

A *reaction* is a triple $a = (R, I, P)$ of finite sets.

- A set R of *reactants*
- A set I of *inhibitors*
- A set P of *products*

If $R, I, P \subseteq S$ then a is a reaction over S

Reaction Systems

A reaction system is a pair $\mathcal{A} = (S, A)$

Reaction Systems

A reaction system is a pair $\mathcal{A} = (S, A)$

- S is a finite set of *symbols* or *entities* called the *background set*

Reaction Systems

A reaction system is a pair $\mathcal{A} = (S, A)$

- S is a finite set of *symbols* or *entities* called the *background set*
- A is a set of reactions of over S

Reaction Systems

A reaction system is a pair $\mathcal{A} = (S, A)$

- S is a finite set of *symbols* or *entities* called the *background set*
- A is a set of reactions of over S

A *state* of $\mathcal{A}$ is a subset of S

Example of a Reaction System

Background set:

$$S = \{a, b, c, d, e\}$$

Set of reactions:

$$A = \{(\{a\}, \{b, c\}, \{a, c\}) \\ (\{a, c, e\}, \{d\}, \{d, e\})\}$$

Enabled Reactions

A reaction $a = (R, I, P)$ is enabled in a state $T \subseteq S$ when:

Enabled Reactions

A reaction $a = (R, I, P)$ is enabled in a state $T \subseteq S$ when:

- *All the reactants* are present in T:

$$R \subseteq T$$

Enabled Reactions

A reaction $a = (R, I, P)$ is enabled in a state $T \subseteq S$ when:

- *All the reactants* are present in T:

$$R \subseteq T$$

- *None of the inhibitors* is present in T:

$$I \cap T = \varnothing$$

Result Function

Let $a = (R, I, P)$ be a reaction.

The *result function* of a on $T \subseteq S$ is:

$$\mathrm{res}_a(T) = \begin{cases} P & \text{if } a \text{ is enabled in } T \\ \varnothing & \text{otherwise} \end{cases}$$

Result Function

Let $a = (R, I, P)$ be a reaction.

The *result function* of a on $T \subseteq S$ is:

$$\mathrm{res}_a(T) = \begin{cases} P & \text{if } a \text{ is enabled in } T \\ \varnothing & \text{otherwise} \end{cases}$$

Extension to a set A of reactions:

$$\mathrm{res}_A(T) = \bigcup_{a \in A} \mathrm{res}_a(T)$$

Result Function

Let $a = (R, I, P)$ be a reaction.

The *result function* of a on $T \subseteq S$ is:

$$\text{res}_a(T) = \begin{cases} P & \text{if } a \text{ is enabled in } T \\ \varnothing & \text{otherwise} \end{cases}$$

Extension to a set A of reactions:

$$\text{res}_A(T) = \bigcup_{a \in A} \text{res}_a(T)$$

Extension to a reaction system $\mathcal{A} = (S, A)$:

$$\text{res}_{\mathcal{A}} = \text{res}_A$$

Result Function: Example

Background set: $S = \{a, b, c, d, e\}$

Reactions: $r_1 = (\{a\}, \{b, c\}, \{a, c\})$
$r_2 = (\{a, c, e\}, \{d\}, \{d, e\})$

Result Function: Example

Background set: $S = \{a, b, c, d, e\}$

Reactions: $r_1 = (\{a\}, \{b, c\}, \{a, c\})$
$r_2 = (\{a, c, e\}, \{d\}, \{d, e\})$

State: $T = \{a, b, c, e\}$

$$\{a\} \subseteq T$$
$$\{b, c\} \cap T = \{b, c\} \neq \varnothing$$

Result Function: Example

Background set: $S = \{a, b, c, d, e\}$

Reactions: $r_1 = (\{a\}, \{b, c\}, \{a, c\})$
$r_2 = (\{a, c, e\}, \{d\}, \{d, e\})$

State: $T = \{a, b, c, e\}$

$$\{a\} \subseteq T$$
$$\{b, c\} \cap T = \{b, c\} \neq \varnothing$$

$$\mathrm{res}_{r_1}(T) = \varnothing$$

Result Function: Example

Background set: $S = \{a, b, c, d, e\}$

Reactions: $r_1 = (\{a\}, \{b, c\}, \{a, c\})$
$r_2 = (\{a, c, e\}, \{d\}, \{d, e\})$

State: $T = \{a, b, c, e\}$

$$\{a, c, e\} \subseteq T$$

$$\{d\} \cap T = \varnothing$$

Result Function: Example

Background set: $S = \{a, b, c, d, e\}$

Reactions: $r_1 = (\{a\}, \{b, c\}, \{a, c\})$
$r_2 = (\{a, c, e\}, \{d\}, \{d, e\})$

State: $T = \{a, b, c, e\}$

$$\{a, c, e\} \subseteq T$$
$$\{d\} \cap T = \varnothing$$

$$\text{res}_{r_2}(T) = \{d, e\}$$

Result Function: Example

Background set: $S = \{a, b, c, d, e\}$

Reactions: $r_1 = (\{a\}, \{b, c\}, \{a, c\})$
$r_2 = (\{a, c, e\}, \{d\}, \{d, e\})$

State: $T = \{a, b, c, e\}$

$$\text{res}_A(T) = \text{res}_{r_1}(T) \cup \text{res}_{r_2}(T)$$

Result Function: Example

Background set: $S = \{a, b, c, d, e\}$

Reactions: $r_1 = (\{a\}, \{b, c\}, \{a, c\})$
$r_2 = (\{a, c, e\}, \{d\}, \{d, e\})$

State: $T = \{a, b, c, e\}$

$$\begin{aligned} \text{res}_A(T) &= \text{res}_{r_1}(T) \cup \text{res}_{r_2}(T) \\ &= \emptyset \cup \{d, e\} \end{aligned}$$

Result Function: Example

Background set: $S = \{a, b, c, d, e\}$

Reactions: $r_1 = (\{a\}, \{b, c\}, \{a, c\})$
$r_2 = (\{a, c, e\}, \{d\}, \{d, e\})$

State: $T = \{a, b, c, e\}$

$$\begin{aligned} \text{res}_A(T) &= \text{res}_{r_1}(T) \cup \text{res}_{r_2}(T) \\ &= \varnothing \cup \{d, e\} \\ &= \{d, e\} \end{aligned}$$

Table of Contents

Reaction Systems as Dynamical Systems

This is a finite dynamical system:

$$(\mathcal{A}, \mathrm{res}_{\mathcal{A}})$$

Reaction Systems as Dynamical Systems

This is a finite dynamical system:

$$(\mathcal{A}, \mathsf{res}_{\mathcal{A}})$$

where:

- $\mathcal{A}$ is a reaction system
- $\mathsf{res}_{\mathcal{A}}$ is its result function

Reaction Systems as Dynamical Systems

This is a finite dynamical system:

$$(\mathcal{A}, \mathrm{res}_{\mathcal{A}})$$

where:

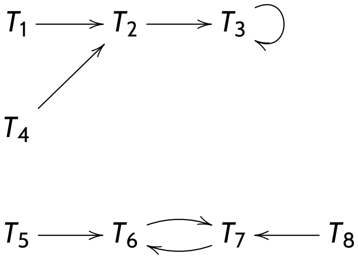
- $\mathcal{A}$ is a reaction system
- $\mathrm{res}_{\mathcal{A}}$ is its result function

State sequence or *orbit* starting from $T \subseteq S$:

$$\left(T, \mathrm{res}_{\mathcal{A}}(T), \mathrm{res}_{\mathcal{A}}^2(T), \mathrm{res}_{\mathcal{A}}^3(T), \ldots\right)$$

Some Terminology

If $\text{res}_{\mathcal{A}}(T_i) = T_j$ then there is an arrow from T_i to T_j:



Some Dynamical Properties 1/3

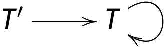
Some Dynamical Properties 1/3

- **Fixed Point.** $\text{res}_{\mathcal{A}}(T) = T$:

Some Dynamical Properties 1/3

- **Fixed Point.** $\text{res}_{\mathcal{A}}(T) = T$:

- **Fixed Point Attractor.** “A fixed point with something going in”



Some Dynamical Properties 1/3

- **Fixed Point.** $\text{res}_{\mathcal{A}}(T) = T$:

$$T \circlearrowright$$

- **Fixed Point Attractor.** “A fixed point with something going in”

$$T' \longrightarrow T \circlearrowright$$

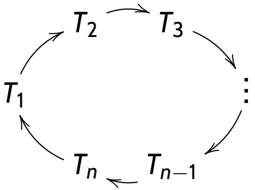
- **Global Fixed Point Attractor.** “A fixed point where everything goes in”

$$\forall T' \subseteq S \qquad T' \longrightarrow T'' \longrightarrow \ldots \longrightarrow T \circlearrowright$$

Some Dynamical Properties 2/3

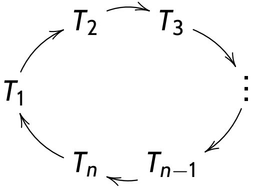
Some Dynamical Properties 2/3

- **Cycle.** Every finite dynamical system has a cycle

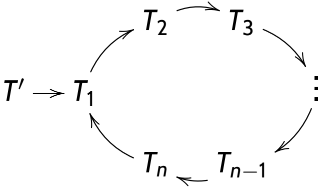


Some Dynamical Properties 2/3

- **Cycle.** Every finite dynamical system has a cycle



- **Attractor Cycle.** “A cycle with something going in”



Some Dynamical Properties 3/3

Some Dynamical Properties 3/3

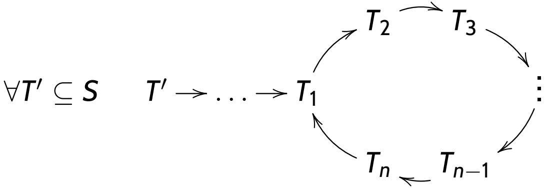
- **Global Attractor Cycle.** “A cycle reachable from every state”

$$\forall T' \subseteq S \qquad T' \to \dots \to T_1$$

$$T_1 \to T_2 \to T_3 \to \cdots \to T_{n-1} \to T_n \to T_1$$

Some Dynamical Properties 3/3

- **Global Attractor Cycle.** "A cycle reachable from every state"

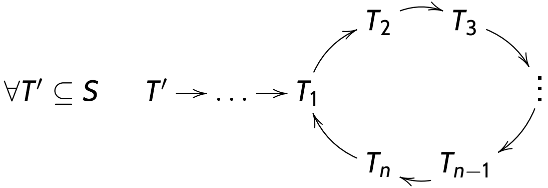


- **Gardens of Eden.** "A state with nothing going in"
  A state with no *preimages*

$$T' \overset{\text{never}}{\dashrightarrow} T$$

Some Dynamical Properties 3/3

- **Global Attractor Cycle.** "A cycle reachable from every state"



- **Gardens of Eden.** "A state with nothing going in"
  A state with no *preimages*

$$T' \overset{\text{never}}{\dashrightarrow} T$$

  Recall that: *garden of Eden* $\iff$ *attractor cycle*

Table of Contents

Fixed Points Problems

Given a reaction system $\mathcal{A} = (S, A)$:

Fixed Points Problems

Given a reaction system $\mathcal{A} = (S, A)$:

- does $\mathcal{A}$ have a fixed point?

Fixed Points Problems

Given a reaction system $\mathcal{A} = (S, A)$:

- does $\mathcal{A}$ have a fixed point?
- does $\mathcal{A}$ have a fixed point attractor?

Fixed Points Problems

Given a reaction system $\mathcal{A} = (S, A)$:

- does $\mathcal{A}$ have a fixed point?
- does $\mathcal{A}$ have a fixed point attractor?
- does $\mathcal{A}$ have a fixed point that is a global attractor?

Cycles Problems

Given a reaction system $\mathcal{A} = (S, A)$:

Cycles Problems

Given a reaction system $\mathcal{A} = (S, A)$:

- does $\mathcal{A}$ have an attractor cycle?

Cycles Problems

Given a reaction system $\mathcal{A} = (S, A)$:

- does $\mathcal{A}$ have an attractor cycle?
- does $\mathcal{A}$ have a global attractor cycle?

Existence of a Fixed Point

Let $\varphi = (x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee x_3)$

Existence of a Fixed Point

Let $\varphi = (x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee x_3)$

We will build a reaction system with a fixed point *iff* φ is satisfiable

Existence of a Fixed Point

Let $\varphi = (x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee x_3)$

We will build a reaction system with a fixed point *iff* φ is satisfiable

Background set: $S = \{x_1, x_2, x_3, \clubsuit, \spadesuit\}$

Encoding the Assignments

$$x_1 = \text{True}$$
$$x_2 = \text{False}$$
$$x_3 = \text{True}$$

Encoding the Assignments

$$x_1 = \text{True}$$
$$x_2 = \text{False} \qquad \Rightarrow \{x_1, x_3\}$$
$$x_3 = \text{True}$$

Encoding the Assignments

$$\left.\begin{aligned} x_1 &= \text{True} \\ x_2 &= \text{False} \\ x_3 &= \text{True} \end{aligned}\right. \qquad \Rightarrow \{x_1, x_3\}$$

Idea: if T is a satisfying assignment then:

$$T \circlearrowright$$

else

$$T \longrightarrow T \cup \{\spadesuit\} \rightleftarrows T \cup \{\clubsuit\}$$

The Reactions

Preserve the assignment:

$$(\{x_i\}, \varnothing, \{x_i\})$$

The Reactions

Preserve the assignment:

$$(\{x_i\}, \varnothing, \{x_i\})$$

Create a cycle with ♠ and ♣:

$$(\{\spadesuit\}, \varnothing, \{\clubsuit\})$$
$$(\{\clubsuit\}, \{\spadesuit\}, \{\spadesuit\})$$

The Reactions

Preserve the assignment:

$$(\{x_i\}, \emptyset, \{x_i\})$$

Create a cycle with ♠ and ♣:

$$(\{\spadesuit\}, \emptyset, \{\clubsuit\})$$
$$(\{\clubsuit\}, \{\spadesuit\}, \{\spadesuit\})$$

Evaluate a clause (e.g., $x_1 \vee \neg x_2 \vee x_3$):

$$(\{x_2\}, \{x_1, x_3, \spadesuit, \clubsuit\}, \{\spadesuit\})$$

A Non-Satisfying Assignment

Evaluation of

$$\varphi = (x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee x_3)$$

with the assignment $x_1 = \text{False}, x_2 = \text{True}, x_2 = \text{False}$

A Non-Satisfying Assignment

Evaluation of

$$\varphi = (x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee x_3)$$

with the assignment $x_1 = \text{False}, x_2 = \text{True}, x_2 = \text{False}$

$$\{x_2\} \longrightarrow \{\quad , \quad\}$$

A Non-Satisfying Assignment

Evaluation of

$$\varphi = (x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee x_3)$$

with the assignment $x_1 = \text{False}, x_2 = \text{True}, x_2 = \text{False}$

$$\{x_2\} \longrightarrow \{x_2, \quad \}$$

$$(\{x_2\}, \varnothing, \{x_2\})$$

A Non-Satisfying Assignment

Evaluation of

$$\varphi = (x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee x_3)$$

with the assignment $x_1 = \text{False}, x_2 = \text{True}, x_2 = \text{False}$

$$\{x_2\} \longrightarrow \{x_2, \spadesuit\}$$

$$(\{x_2\}, \{x_1, x_3, \spadesuit, \clubsuit\}, \{\spadesuit\})$$

A Non-Satisfying Assignment

Evaluation of

$$\varphi = (x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee x_3)$$

with the assignment $x_1 = \text{False}, x_2 = \text{True}, x_2 = \text{False}$

$$\begin{array}{ccc} \{x_2\} & \longrightarrow & \{x_2, \spadesuit\} \\ & & \downarrow \\ & & \{\quad,\quad\} \end{array}$$

A Non-Satisfying Assignment

Evaluation of

$$\varphi = (x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee x_3)$$

with the assignment $x_1 = \text{False}, x_2 = \text{True}, x_2 = \text{False}$

$$\begin{array}{ccc} \{x_2\} & \longrightarrow & \{x_2, \spadesuit\} \\ & & \downarrow \\ & & \{x_2, \quad\} \end{array}$$

$$(\{x_2\}, \varnothing, \{x_2\})$$

A Non-Satisfying Assignment

Evaluation of

$$\varphi = (x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee x_3)$$

with the assignment $x_1 = \text{False}, x_2 = \text{True}, x_2 = \text{False}$

$$\begin{array}{ccc} \{x_2\} & \longrightarrow & \{x_2, \spadesuit\} \\ & & \downarrow \\ & & \{x_2, \clubsuit\} \end{array}$$

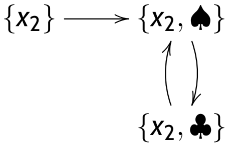
$$(\{\spadesuit\}, \varnothing, \{\clubsuit\})$$

A Non-Satisfying Assignment

Evaluation of

$$\varphi = (x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee x_3)$$

with the assignment $x_1 = \text{False}, x_2 = \text{True}, x_2 = \text{False}$



A Satisfying Assignment

Evaluation of

$$\varphi = (x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee x_3)$$

with the assignment $x_1 = \text{True}, x_2 = \text{True}, x_2 = \text{False}$

A Satisfying Assignment

Evaluation of

$$\varphi = (x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee x_3)$$

with the assignment $x_1 = \text{True}, x_2 = \text{True}, x_2 = \text{False}$

$$\{x_1, x_2\} \longrightarrow \{\quad,\quad\}$$

A Satisfying Assignment

Evaluation of

$$\varphi = (x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee x_3)$$

with the assignment $x_1 = \text{True}, x_2 = \text{True}, x_2 = \text{False}$

$$\{x_1, x_2\} \longrightarrow \{x_1, \quad\}$$

$$(\{x_1\}, \varnothing, \{x_1\})$$

A Satisfying Assignment

Evaluation of

$$\varphi = (x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee x_3)$$

with the assignment $x_1 = \text{True}, x_2 = \text{True}, x_2 = \text{False}$

$$\{x_1, x_2\} \longrightarrow \{x_1, x_2\}$$

$$(\{x_2\}, \varnothing, \{x_2\})$$

NP-complete problems

This shows the **NP**-hardness of finding if a fixed point exists

NP-complete problems

This shows the **NP**-hardness of finding if a fixed point exists

With similar techniques we can find:

NP-complete problems

This shows the **NP**-hardness of finding if a fixed point exists

With similar techniques we can find:

- Finding if a fixed point exists is **NP**-complete

NP-complete problems

This shows the **NP**-hardness of finding if a fixed point exists

With similar techniques we can find:

- Finding if a fixed point exists is **NP**-complete
- Finding if a fixed point attractor exists is **NP**-complete

NP-complete problems

This shows the **NP**-hardness of finding if a fixed point exists

With similar techniques we can find:

- Finding if a fixed point exists is **NP**-complete
- Finding if a fixed point attractor exists is **NP**-complete
- Finding if an attractor cycle exists is **NP**-complete

Global Attractors

For global attractors we need another approach:

Global Attractors

For global attractors we need another approach:

A Turing Machine + A binary counter

Global Attractors

For global attractors we need another approach:

A Turing Machine + A binary counter

- The Turing Machine has a polynomially-sized tape

Global Attractors

For global attractors we need another approach:

A Turing Machine + A binary counter

- The Turing Machine has a polynomially-sized tape
- The binary counter force the machine in a fixed point after a finite number of steps. . .

Global Attractors

For global attractors we need another approach:

A Turing Machine + A binary counter

- The Turing Machine has a polynomially-sized tape
- The binary counter force the machine in a fixed point after a finite number of steps. . .
- . . .unless the TM has already rejected the input

Global Attractors: Results

- Finding if there exists a global fixed point attractor is **PSPACE**-complete

Global Attractors: Results

- Finding if there exists a global fixed point attractor is **PSPACE**-complete
- Finding if there exists a global attractor cycle is **PSPACE**-complete

Global Attractors: Results

- Finding if there exists a global fixed point attractor is **PSPACE**-complete
- Finding if there exists a global attractor cycle is **PSPACE**-complete
- Reachability between two configurations is **PSPACE**-complete

Table of Contents

Bounding Reactants and Inhibitors

$\mathcal{RS}(r, i)$:
All Reaction Systems whose reactions

- have at most *r* reactants
- and at most *i* inhibitors

Bounding Reactants and Inhibitors

$\mathcal{RS}(r, i)$:
All Reaction Systems whose reactions

- have at most *r* reactants
- and at most *i* inhibitors

We have the classes:

Bounding Reactants and Inhibitors

$\mathcal{RS}(r, i)$:
All Reaction Systems whose reactions

- have at most r reactants
- and at most i inhibitors

We have the classes:

- $\mathcal{RS}(\infty, 0)$ is all Reaction Systems without inhibitors

Bounding Reactants and Inhibitors

$\mathcal{RS}(r, i)$:
All Reaction Systems whose reactions

- have at most *r* reactants
- and at most *i* inhibitors

We have the classes:

- $\mathcal{RS}(\infty, 0)$ is all Reaction Systems without inhibitors
- $\mathcal{RS}(0, \infty)$ is all Reaction Systems without reactants

Bounding Reactants and Inhibitors

$\mathcal{RS}(r, i)$:
All Reaction Systems whose reactions

- have at most r reactants
- and at most i inhibitors

We have the classes:

- $\mathcal{RS}(\infty, 0)$ is all Reaction Systems without inhibitors
- $\mathcal{RS}(0, \infty)$ is all Reaction Systems without reactants
- $\mathcal{RS}(\infty, \infty)$ is all Reaction Systems

Classification

$\mathcal{RS}(\infty, \infty)$ Every function $2^S \to 2^S$

Classification

$\mathcal{RS}(\infty, \infty)$ Every function $2^S \to 2^S$

$\mathcal{RS}(0, \infty)$ Antitone functions: $T \subseteq T' \to \text{res}_{\mathcal{A}}(T) \supseteq \text{res}_{\mathcal{A}}(T')$

Classification

$\mathcal{RS}(\infty, \infty)$ Every function $2^S \to 2^S$

$\mathcal{RS}(0, \infty)$ Antitone functions: $T \subseteq T' \to \text{res}_{\mathcal{A}}(T) \supseteq \text{res}_{\mathcal{A}}(T')$

$\mathcal{RS}(\infty, 0)$ Monotone functions: $T \subseteq T' \to \text{res}_{\mathcal{A}}(T) \subseteq \text{res}_{\mathcal{A}}(T')$

Classification

$\mathcal{RS}(\infty, \infty)$	Every function $2^S \to 2^S$
$\mathcal{RS}(0, \infty)$	Antitone functions: $T \subseteq T' \to \text{res}_{\mathcal{A}}(T) \supseteq \text{res}_{\mathcal{A}}(T')$
$\mathcal{RS}(\infty, 0)$	Monotone functions: $T \subseteq T' \to \text{res}_{\mathcal{A}}(T) \subseteq \text{res}_{\mathcal{A}}(T')$
$\mathcal{RS}(1, 0)$	Functions such that $\text{res}_{\mathcal{A}}(T \cup U) = \text{res}_{\mathcal{A}}(T) \cup \text{res}_{\mathcal{A}}(U)$

Classification

$\mathcal{RS}(\infty, \infty)$	Every function $2^S \to 2^S$
$\mathcal{RS}(0, \infty)$	Antitone functions: $T \subseteq T' \to \text{res}_{\mathcal{A}}(T) \supseteq \text{res}_{\mathcal{A}}(T')$
$\mathcal{RS}(\infty, 0)$	Monotone functions: $T \subseteq T' \to \text{res}_{\mathcal{A}}(T) \subseteq \text{res}_{\mathcal{A}}(T')$
$\mathcal{RS}(1, 0)$	Functions such that $\text{res}_{\mathcal{A}}(T \cup U) = \text{res}_{\mathcal{A}}(T) \cup \text{res}_{\mathcal{A}}(U)$
$\mathcal{RS}(0, 0)$	All constant functions

Reachability in Resource-Constrained Reaction Systems

By adapting the simulation of Turing Machine we can prove that

Reachability in Resource-Constrained Reaction Systems

By adapting the simulation of Turing Machine we can prove that

- Reachability is **PSPACE**-complete for $\mathcal{RS}(0, \infty)$

Reachability in Resource-Constrained Reaction Systems

By adapting the simulation of Turing Machine we can prove that

- Reachability is **PSPACE**-complete for $\mathcal{RS}(0, \infty)$
- Reachability is **PSPACE**-complete for $\mathcal{RS}(\infty, 0)$

Reachability in Resource-Constrained Reaction Systems

By adapting the simulation of Turing Machine we can prove that

- Reachability is **PSPACE**-complete for $\mathcal{RS}(0, \infty)$
- Reachability is **PSPACE**-complete for $\mathcal{RS}(\infty, 0)$

However for $\mathcal{RS}(1, 0)$ it is **NL**-hard and in **NP**.
We solved the similar problem of *sup-reachability*

Influence Graph

$$
\begin{aligned}
S &= \{a, b, c\} \\
A &= \{(\{a\}, \varnothing, \{b\}) \\
&\quad (\{b\}, \varnothing, \{c\}) \\
&\quad (\{a\}, \varnothing, \{c\}) \\
&\quad (\{c\}, \varnothing, \{c\})\}
\end{aligned}
$$

Influence Graph

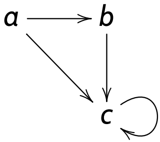
$$
\begin{aligned}
S &= \{a, b, c\} \\
A &= \{(\{a\}, \varnothing, \{b\}) \\
&\quad (\{b\}, \varnothing, \{c\}) \\
&\quad (\{a\}, \varnothing, \{c\}) \\
&\quad (\{c\}, \varnothing, \{c\})\}
\end{aligned}
$$

We can interpret S as a set of *vertices* and A as a set of *edges*:

Influence Graph

$$S = \{a, b, c\}$$
$$A = \{(\{a\}, \varnothing, \{b\}) \\ (\{b\}, \varnothing, \{c\}) \\ (\{a\}, \varnothing, \{c\}) \\ (\{c\}, \varnothing, \{c\})\}$$

We can interpret S as a set of *vertices* and A as a set of *edges*:



Sup-Reachability in $\mathcal{RS}(1, 0)$

Let $\varphi = (x_1 \vee \neg x_2) \wedge (x_1 \vee x_2) \wedge (\neg x_1 \vee \neg x_2 \vee x_3) \wedge (x_2 \vee x_3)$

Sup-Reachability in $\mathcal{RS}(1, 0)$

Let $\varphi = (x_1 \vee \neg x_2) \wedge (x_1 \vee x_2) \wedge (\neg x_1 \vee \neg x_2 \vee x_3) \wedge (x_2 \vee x_3)$

For each variable x_i:

Sup-Reachability in $\mathcal{RS}(1, 0)$

Let $\varphi = (x_1 \vee \neg x_2) \wedge (x_1 \vee x_2) \wedge (\neg x_1 \vee \neg x_2 \vee x_3) \wedge (x_2 \vee x_3)$

For each variable x_i:

- Create a cycle of length p_i (the i-th prime) in the influence graph

Sup-Reachability in $\mathcal{RS}(1, 0)$

Let $\varphi = (x_1 \vee \neg x_2) \wedge (x_1 \vee x_2) \wedge (\neg x_1 \vee \neg x_2 \vee x_3) \wedge (x_2 \vee x_3)$

For each variable x_i:

- Create a cycle of length p_i (the i-th prime) in the influence graph
- A point of the cycle generates all the clauses that $x_i = \text{True}$ forces to be true

Sup-Reachability in $\mathcal{RS}(1, 0)$

Let $\varphi = (x_1 \vee \neg x_2) \wedge (x_1 \vee x_2) \wedge (\neg x_1 \vee \neg x_2 \vee x_3) \wedge (x_2 \vee x_3)$

For each variable x_i:

- Create a cycle of length p_i (the i-th prime) in the influence graph
- A point of the cycle generates all the clauses that $x_i =$ True forces to be true
- All the other points generates all the clauses that $x_i =$ False forces to be true

Sup-Reachability in $\mathcal{RS}(1, 0)$

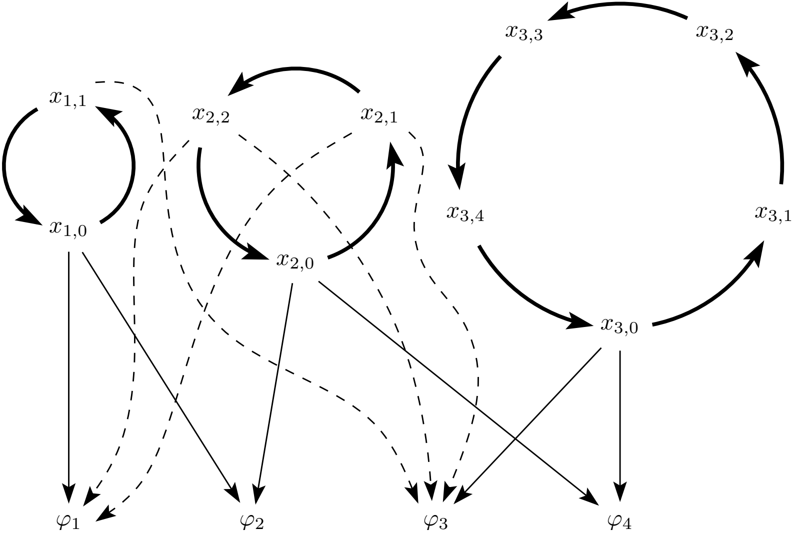
Let $\varphi = (x_1 \vee \neg x_2) \wedge (x_1 \vee x_2) \wedge (\neg x_1 \vee \neg x_2 \vee x_3) \wedge (x_2 \vee x_3)$

For each variable x_i:

- Create a cycle of length p_i (the i-th prime) in the influence graph
- A point of the cycle generates all the clauses that $x_i =$ True forces to be true
- All the other points generates all the clauses that $x_i =$ False forces to be true

The set of all clauses appears iff φ is satisfiable

Reachability Influence Graph



Sup-Reachability Complexity

The previous construction shows the **NP**-hardness of the sup-reachability problem

Sup-Reachability Complexity

The previous construction shows the **NP**-hardness of the sup-reachability problem

To show the containment in **NP**:

Sup-Reachability Complexity

The previous construction shows the **NP**-hardness of the sup-reachability problem

To show the containment in **NP**:

- Let G be the adjacency matrix of the influence graph

Sup-Reachability Complexity

The previous construction shows the **NP**-hardness of the sup-reachability problem

To show the containment in **NP**:

- Let G be the adjacency matrix of the influence graph
- Let X be the characteristic vector of the state $T_X \subseteq S$

Sup-Reachability Complexity

The previous construction shows the **NP**-hardness of the sup-reachability problem

To show the containment in **NP**:

- Let G be the adjacency matrix of the influence graph
- Let X be the characteristic vector of the state $T_X \subseteq S$
- Let Y be the characteristic vector of the state $T_y \subseteq S$

Sup-Reachability Complexity

The previous construction shows the **NP**-hardness of the sup-reachability problem

To show the containment in **NP**:

- Let G be the adjacency matrix of the influence graph
- Let X be the characteristic vector of the state $T_X \subseteq S$
- Let Y be the characteristic vector of the state $T_y \subseteq S$
- Let $\geq$ be the element-wise comparison of two vectors

Sup-Reachability Complexity

The previous construction shows the **NP**-hardness of the sup-reachability problem

To show the containment in **NP**:

- Let G be the adjacency matrix of the influence graph
- Let X be the characteristic vector of the state $T_X \subseteq S$
- Let Y be the characteristic vector of the state $T_y \subseteq S$
- Let $\geq$ be the element-wise comparison of two vectors

then we only need to guess a time step $t \in \mathbb{N}$ and check if

$$G^t X \geq Y$$

Table of Contents

Other Research Areas

Other Research Areas

- Combinatorial properties of Reaction Systems

Other Research Areas

- Combinatorial properties of Reaction Systems
- Long sequences and cycle in resource-constrained Reaction Systems

Other Research Areas

- Combinatorial properties of Reaction Systems
- Long sequences and cycle in resource-constrained Reaction Systems
- Dynamical Properties in resource-constrained Reaction Systems

Other Research Areas

- Combinatorial properties of Reaction Systems
- Long sequences and cycle in resource-constrained Reaction Systems
- Dynamical Properties in resource-constrained Reaction Systems
- Modeling of biological systems

Other Research Areas

- Combinatorial properties of Reaction Systems
- Long sequences and cycle in resource-constrained Reaction Systems
- Dynamical Properties in resource-constrained Reaction Systems
- Modeling of biological systems
- Combination of multiple Reaction Systems

Thank you
for your attention