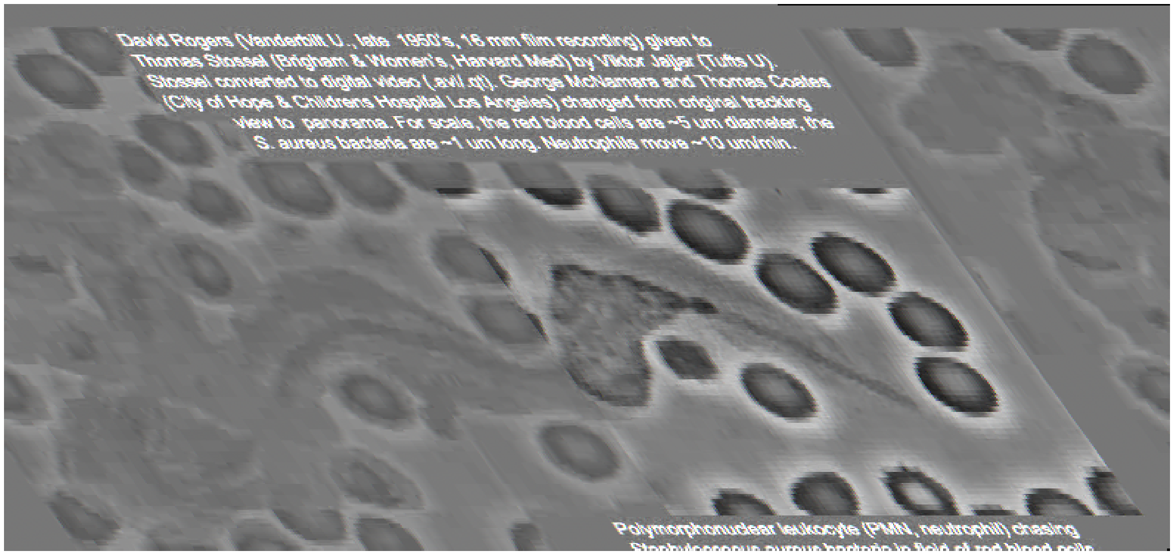
Rate-independent computation by real-valued chemistry

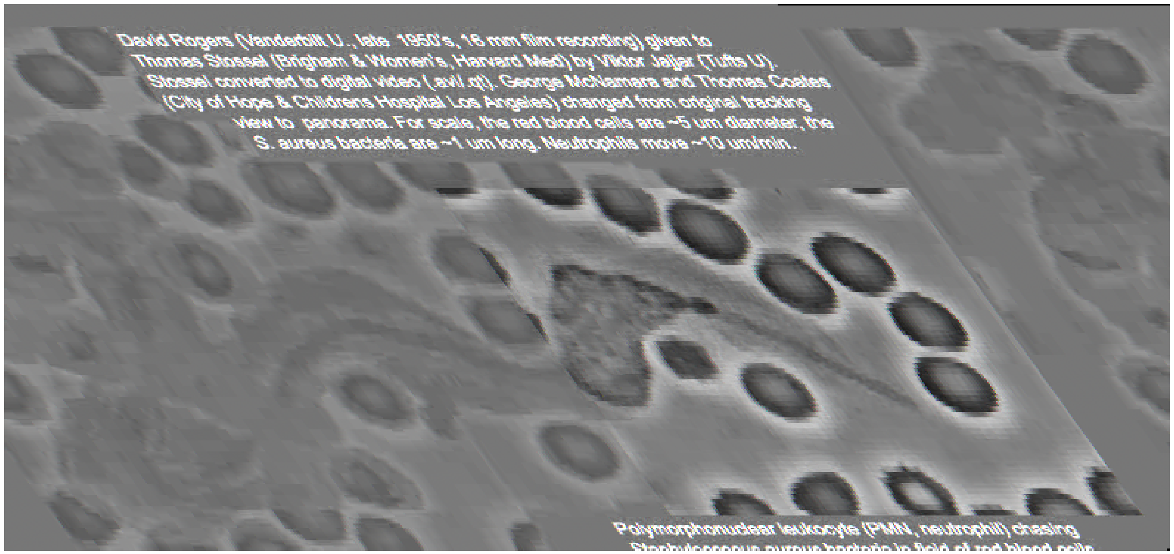
David Doty
Senior Research Fellow
California Institute of Technology

The software of life

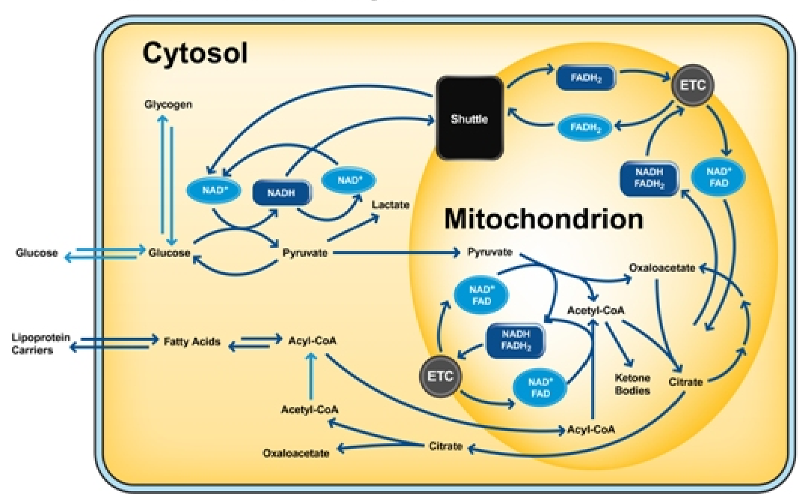


How does the cell compute?

The software of life

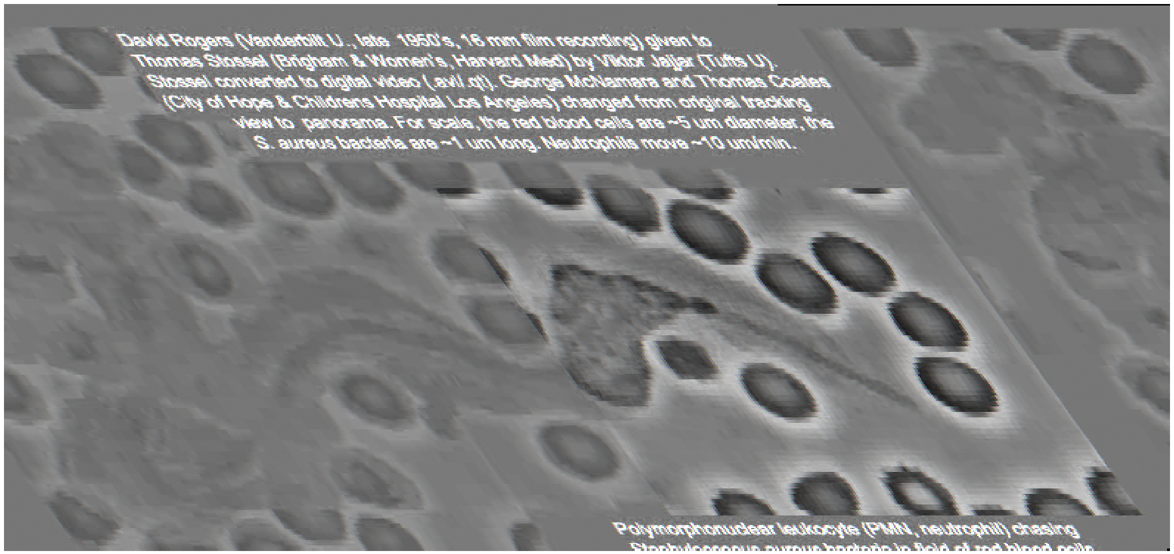


How does the cell compute?



chemistry / geometry

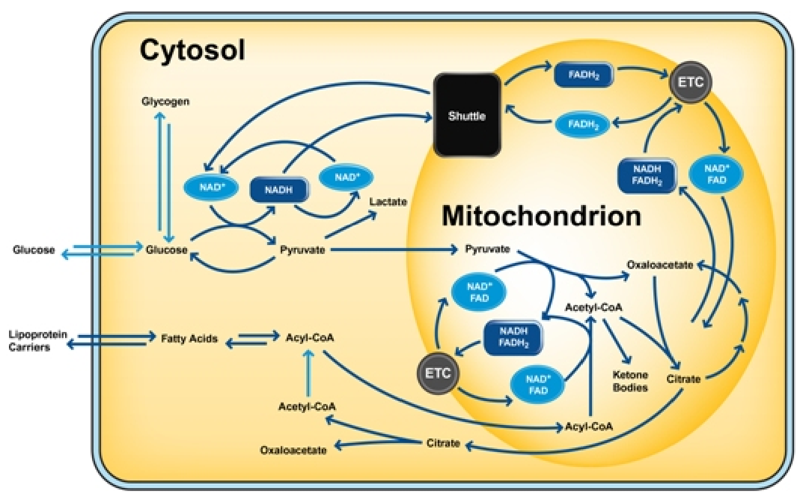
The software of life



~~How does the cell compute?~~

What is possible to compute with chemistry?

~~geometry~~



Chemical reaction networks (CRN)

Chemical reaction networks (CRN)

$$R \rightarrow P_1 + P_2$$

Chemical reaction networks (CRN)

$$R \longrightarrow P_1 + P_2$$

$$A + B \longrightarrow C$$

Chemical reaction networks (CRN)

$$R \rightarrow P_1 + P_2$$

$$A + B \rightarrow C$$

$$X + Y \rightarrow X + Z$$

Chemical reaction networks (CRN)

$$R \rightarrow P_1 + P_2$$

$$A + B \rightarrow C$$

$$X + Y \rightarrow X + Z$$

$$A + Z \rightarrow$$

(anonymous
waste product)

Chemical reaction networks (CRN)

$$R \xrightarrow{2.5} P_1 + P_2$$

$$A + B \xrightarrow{1} C$$

$$X + Y \xrightarrow{5} X + Z$$

$$A + Z \xrightarrow{0.1}$$

(anonymous
waste product)

Mass-action kinetic model of CRNs

- **species:** $\{X, Y, \ldots\}$

Mass-action kinetic model of CRNs

- **species:** $\{X, Y, \ldots\}$

- **reactions:**

$$X \xrightarrow{k_1} W + 2Y + Z$$

$$A + B \xrightarrow{k_2} X$$

Mass-action kinetic model of CRNs

- **species:** $\{X, Y, \ldots\}$
- **reactions:**

$$X \xrightarrow{k_1} W + 2Y + Z$$

$$A + B \xrightarrow{k_2} X$$

- **state:** real vector of *concentrations* $\mathbf{s} = ([X], [Y], \ldots)$

Mass-action kinetic model of CRNs

- **species:** $\{X, Y, \ldots\}$
- **state:** real vector of *concentrations* $\mathbf{s} = ([X], [Y], \ldots)$
- **reactions:**
- **rate of reaction:**

$X \xrightarrow{k_1} W + 2Y + Z$	$k_1 \bullet [X]$
$A + B \xrightarrow{k_2} X$	$k_2 \bullet [A] \bullet [B]$

Mass-action kinetic model of CRNs

- **species:** $\{X, Y, \ldots\}$
- **state:** real vector of *concentrations* $\mathbf{s} = ([X], [Y], \ldots)$
- **reactions:**
- **rate of reaction:**

$X \xrightarrow{k_1} W + 2Y + Z$	$k_1 \bullet [X]$
$A + B \xrightarrow{k_2} X$	$k_2 \bullet [A] \bullet [B]$

$$d[X]/dt \; =$$

Mass-action kinetic model of CRNs

- **species:** $\{X, Y, \ldots\}$
- **state:** real vector of *concentrations* $\mathbf{s} = ([X], [Y], \ldots)$
- **reactions:**
- **rate of reaction:**

$\boxed{X} \xrightarrow{k_1} W + 2Y + Z$	$k_1 \bullet [X]$
$A + B \xrightarrow{k_2} \boxed{X}$	$k_2 \bullet [A] \bullet [B]$

$$d[X]/dt \ =$$

Mass-action kinetic model of CRNs

- **species:** $\{X, Y, \ldots\}$
- **state:** real vector of *concentrations* $\mathbf{s} = ([X], [Y], \ldots)$
- **reactions:**
- **rate of reaction:**

$\boxed{X} \xrightarrow{k_1} W + 2Y + Z$	$k_1 \bullet [X]$
$A + B \xrightarrow{k_2} \boxed{X}$	$k_2 \bullet [A] \bullet [B]$

$$d[X]/dt \;=\; -k_1 \bullet [X] + k_2 \bullet [A] \bullet [B]$$

Mass-action kinetic model of CRNs

- **species:** $\{X, Y, \ldots\}$
- **state:** real vector of *concentrations* **s** = ([X], [Y], ...)
- **reactions:**
- **rate of reaction:**

$\boxed{X} \xrightarrow{k_1} W + 2Y + Z$	$k_1 \bullet [X]$
$A + B \xrightarrow{k_2} \boxed{X}$	$k_2 \bullet [A] \bullet [B]$

$$d[X]/dt = -k_1 \bullet [X] + k_2 \bullet [A] \bullet [B]$$

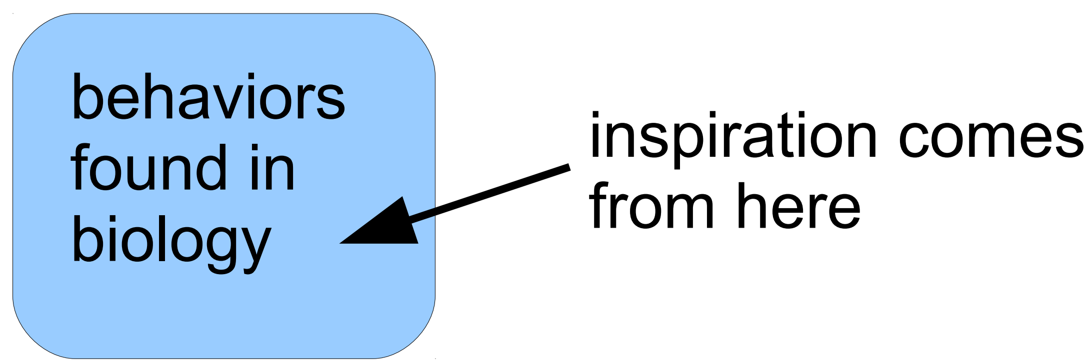
$$d[A]/dt = -k_2 \bullet [A] \bullet [B]$$

$$d[W]/dt = k_1 \bullet [X]$$

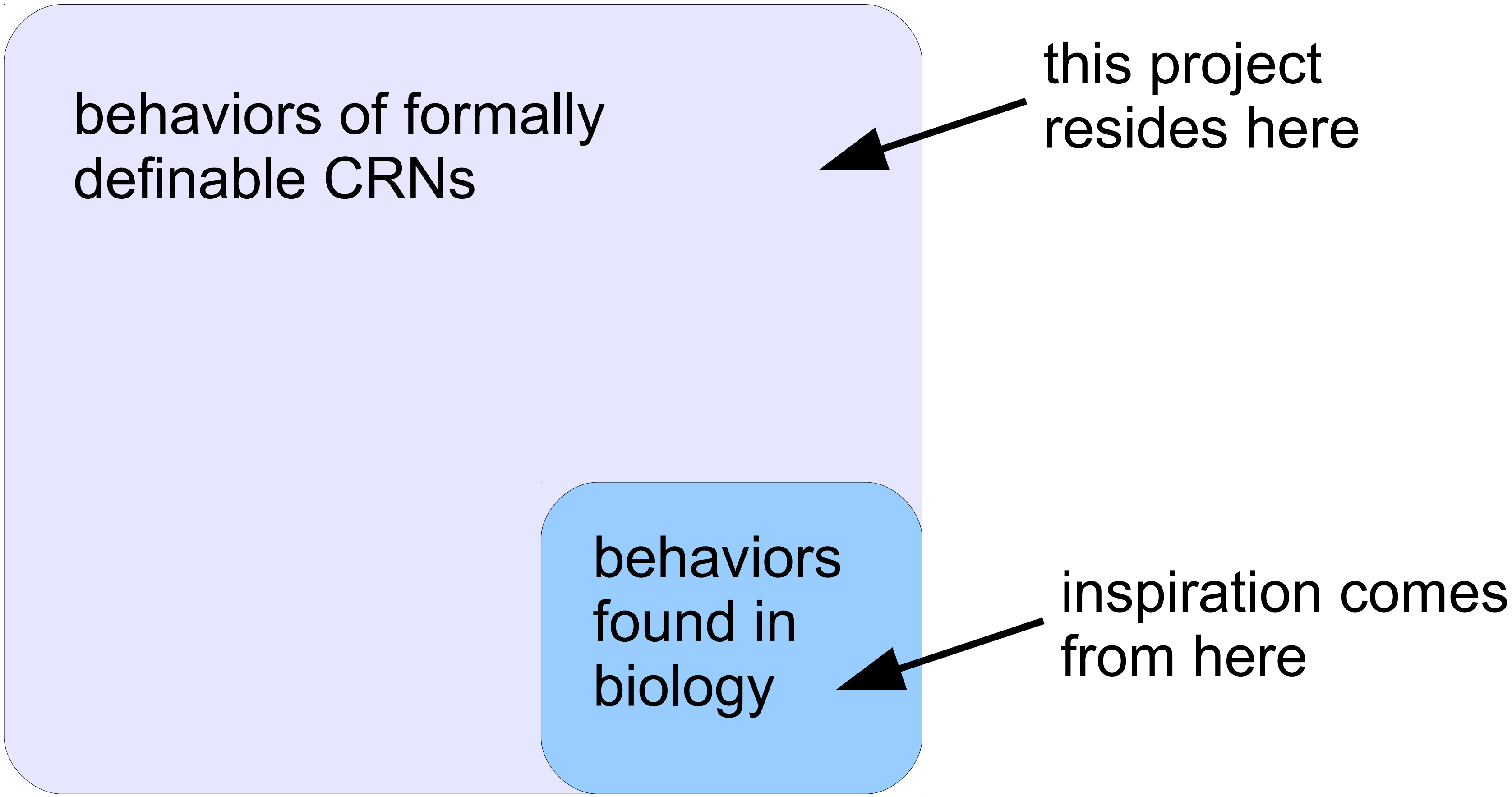
…

What behavior is possible for chemistry in principle?

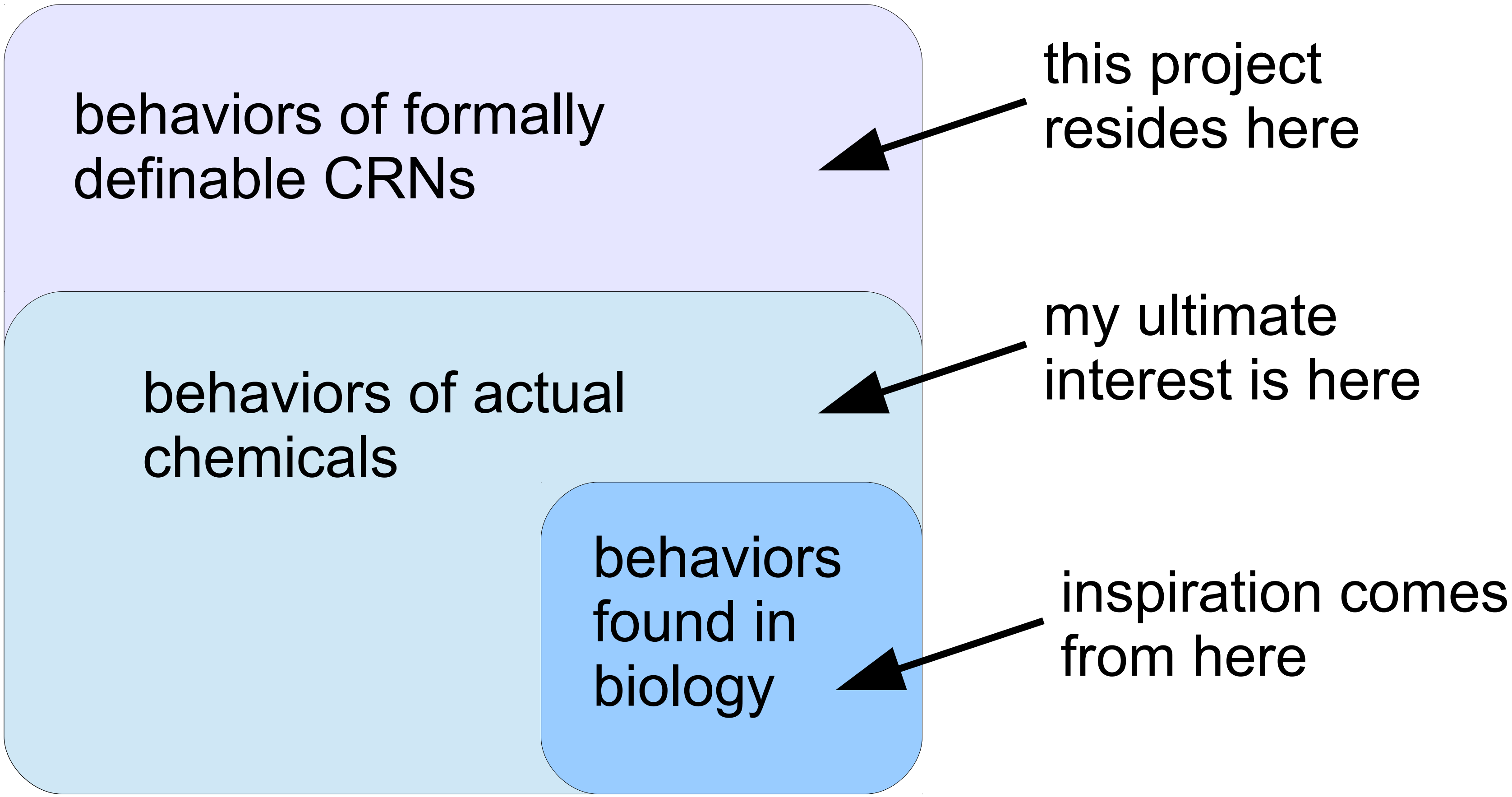
What behavior is possible for chemistry in principle?



What behavior is possible for chemistry in principle?



What behavior is possible for chemistry in principle?



Can we compute with chemistry?

"Not every crazy CRN you scribble on paper describes actual chemicals!"

Can we compute with chemistry?

"Not every crazy CRN you scribble on paper describes actual chemicals!"

Response to objection: Soloveichik et al. [*PNAS* 2010] showed a physical implementation of <u>every</u> CRN, using *DNA strand displacement*

Can we compute with chemistry?

"Not every crazy CRN you scribble on paper describes actual chemicals!"

Response to objection: Soloveichik et al. [*PNAS* 2010] showed a physical implementation of <u>every</u> CRN, using *DNA strand displacement*

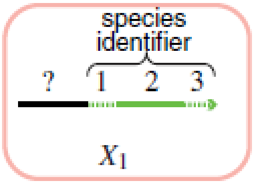
$$X_1 + X_2 \rightarrow X_3$$

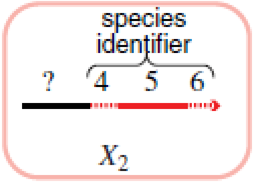
Can we compute with chemistry?

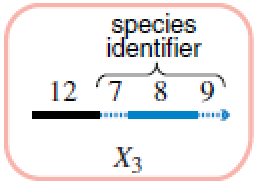
"Not every crazy CRN you scribble on paper describes actual chemicals!"

Response to objection: Soloveichik et al. [*PNAS* 2010] showed a physical implementation of <u>every</u> CRN, using *DNA strand displacement*

$$X_1 + X_2 \rightarrow X_3$$





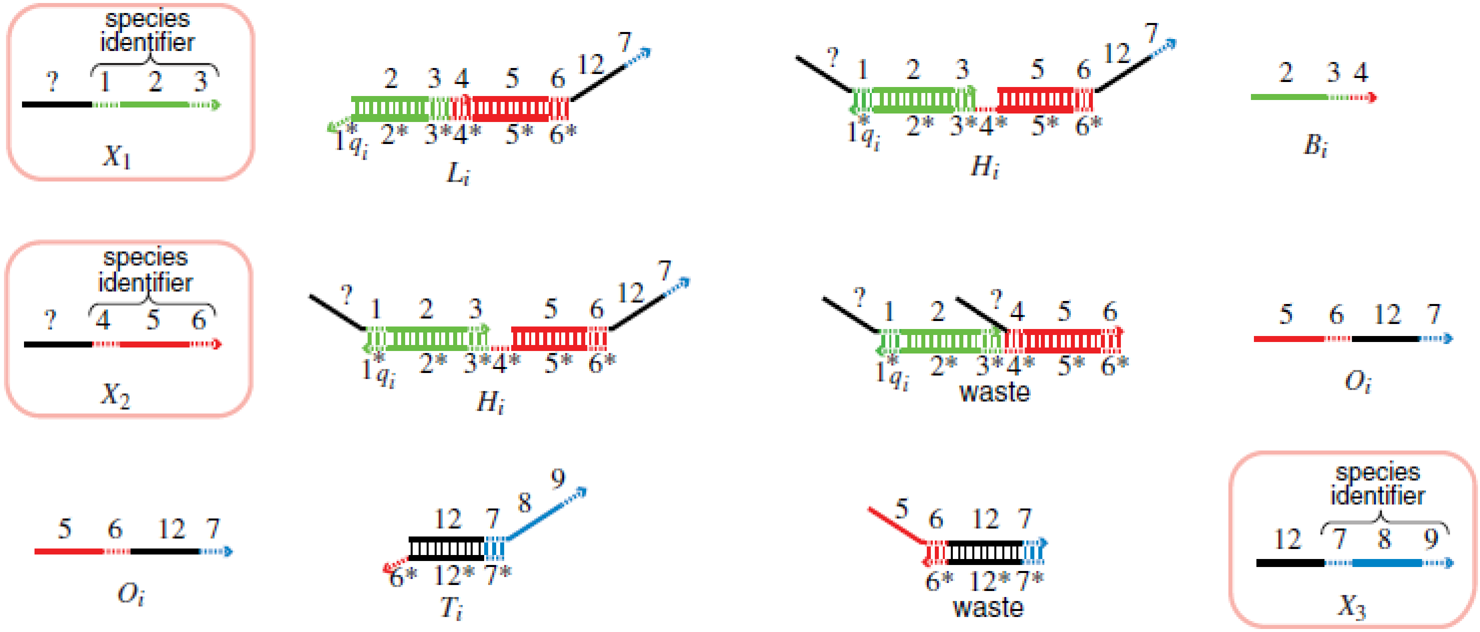


Can we compute with chemistry?

"Not every crazy CRN you scribble on paper describes actual chemicals!"

Response to objection: Soloveichik et al. [*PNAS* 2010] showed a physical implementation of <u>every</u> CRN, using *DNA strand displacement*

$$X_1 + X_2 \rightarrow X_3$$

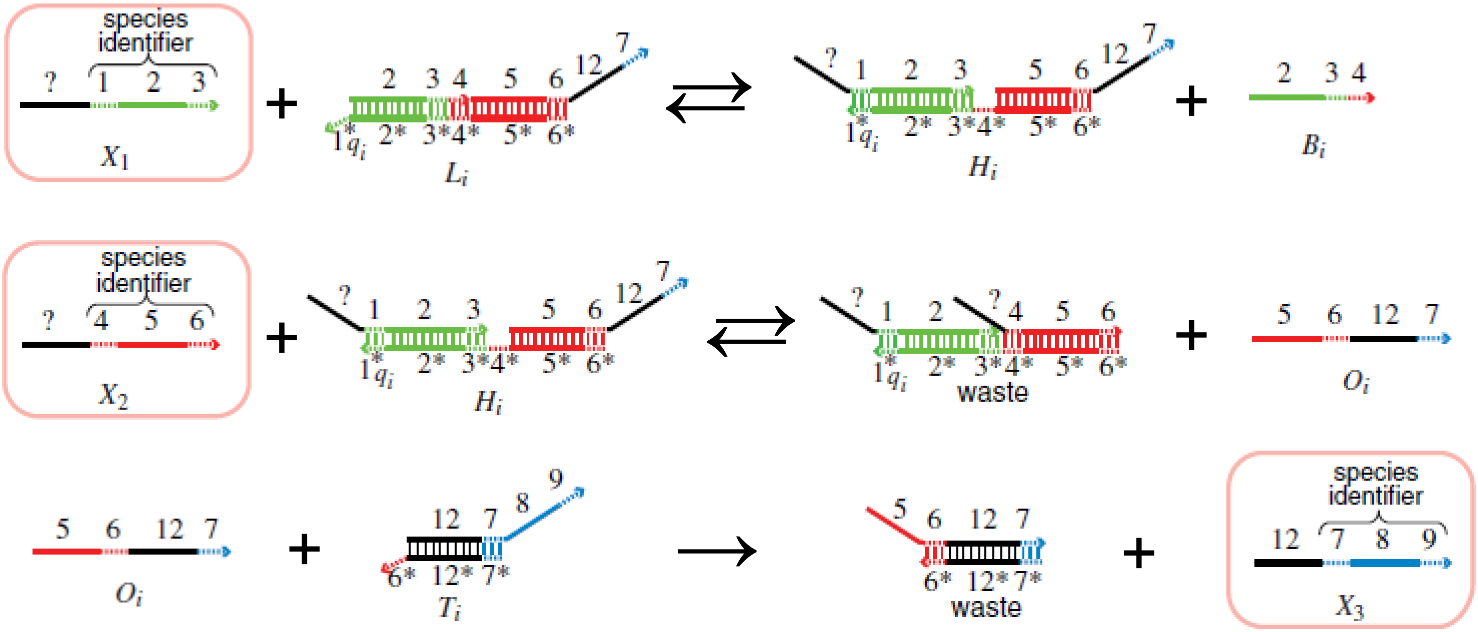


Can we compute with chemistry?

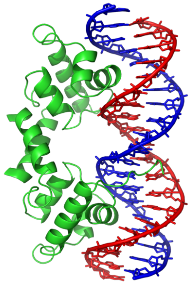
“Not every crazy CRN you scribble on paper describes actual chemicals!”

Response to objection: Soloveichik et al. [*PNAS* 2010] showed a physical implementation of <u>every</u> CRN, using *DNA strand displacement*

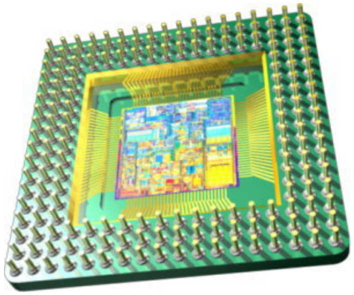
$X_1 + X_2 \rightarrow X_3$



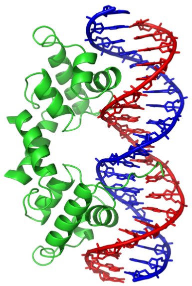
Why compute with chemistry?



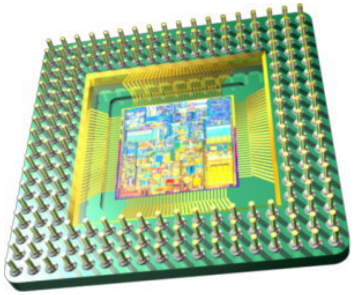
versus



Why compute with chemistry?

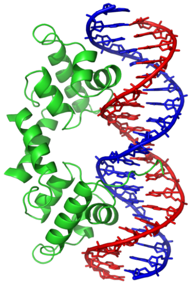


versus

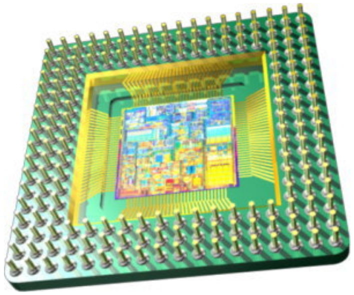


speed?

Why compute with chemistry?



versus

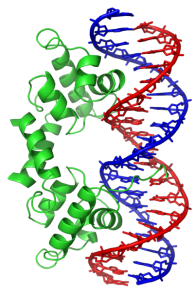


slower

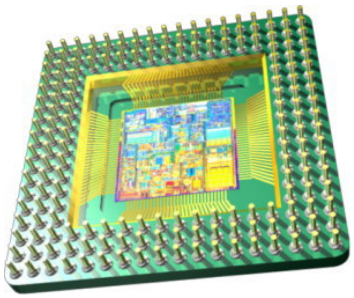
speed?

faster

Why compute with chemistry?



versus

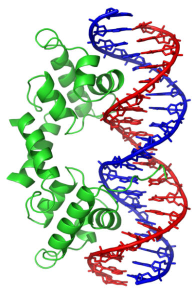


slower

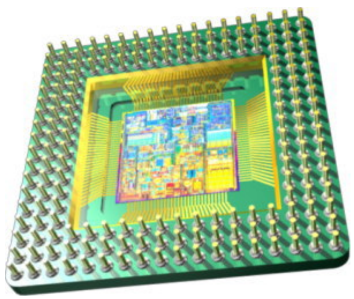
~~speed?~~

faster

Why compute with chemistry?



versus



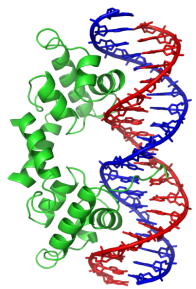
slower

~~speed?~~

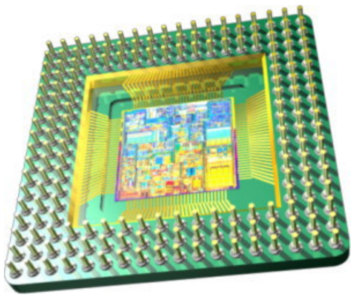
faster

component size?

Why compute with chemistry?



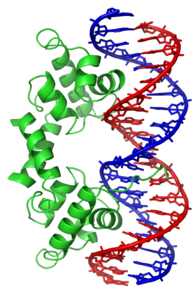
versus



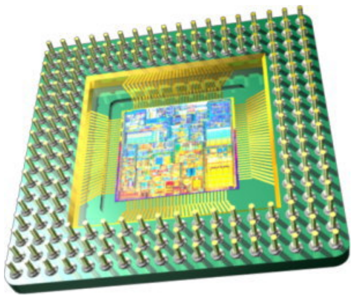
slower ~~speed?~~ faster

≈ 10-100 nm component size?

Why compute with chemistry?

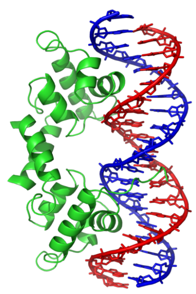


versus

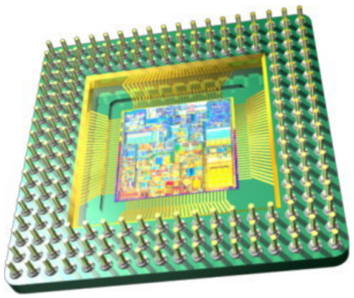


slower	~~speed?~~	faster
≈ 10-100 nm	component size?	≈ 10-100 nm

Why compute with chemistry?



versus



slower

~~speed?~~

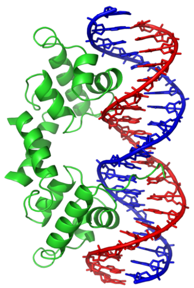
faster

≈ 10-100 nm

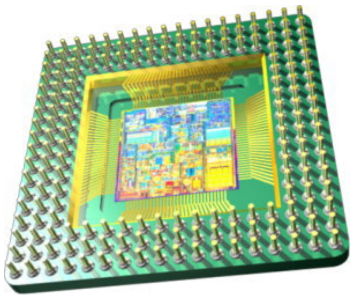
~~component size?~~

≈ 10-100 nm

Why compute with chemistry?

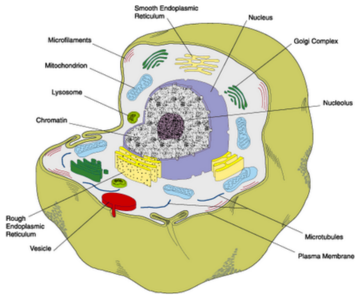


versus



slower	~~speed?~~	faster
≈ 10-100 nm	~~component size?~~	≈ 10-100 nm
yes	✓ Compatible with biological or other “wet environments”?	not easily

cells



“smart drug” to detect microRNAs levels of cell and release appropriate drug in response

bioreactors

“chemical controller” to increase yield of metabolically produced biofuels/drugs/etc.

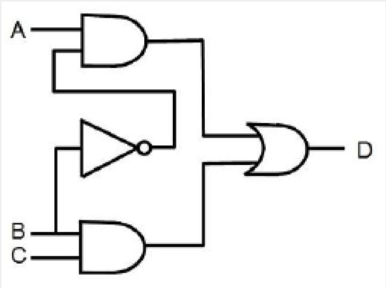
What does it mean to compute with chemistry?

CRNs have a wide range of behaviors:

What does it mean to compute with chemistry?

CRNs have a wide range of behaviors:

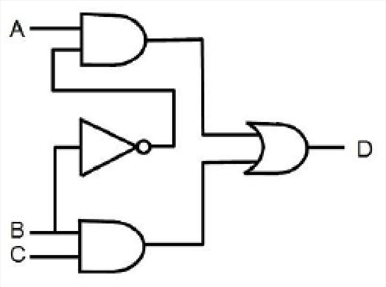
Boolean logic



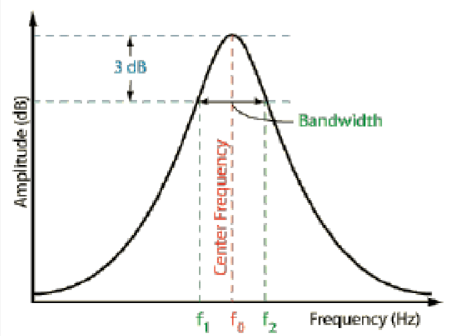
What does it mean to compute with chemistry?

CRNs have a wide range of behaviors:

Boolean logic



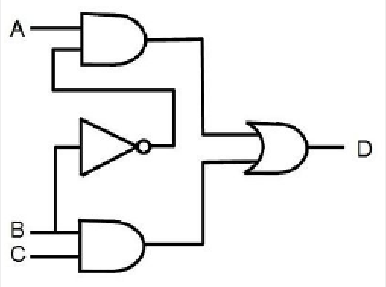
signal processing



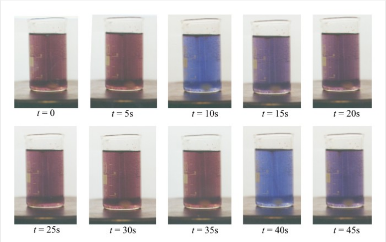
What does it mean to compute with chemistry?

CRNs have a wide range of behaviors:

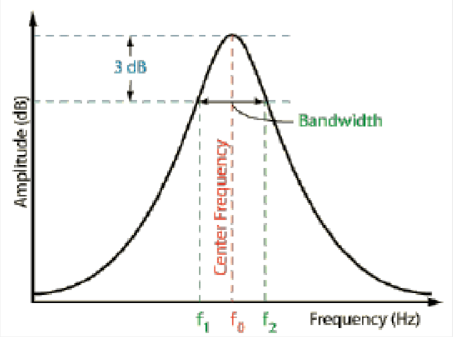
Boolean logic



oscillation



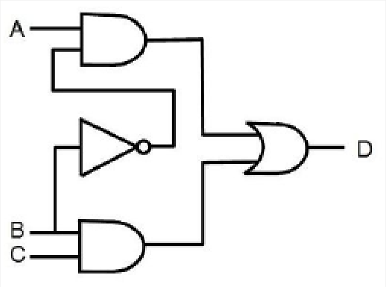
signal processing



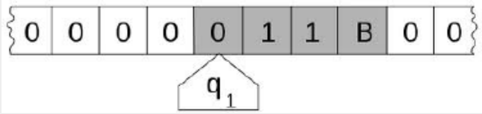
What does it mean to compute with chemistry?

CRNs have a wide range of behaviors:

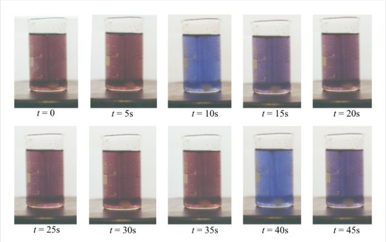
Boolean logic



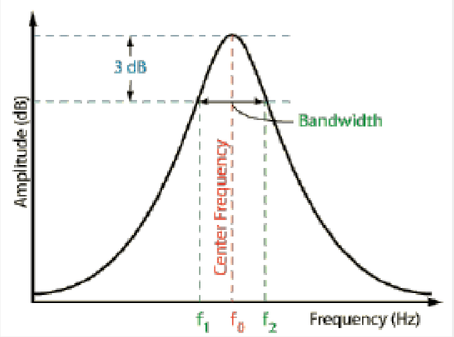
discrete algorithms



oscillation



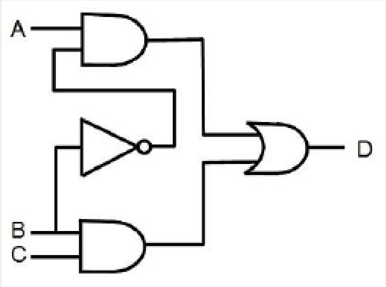
signal processing



What does it mean to compute with chemistry?

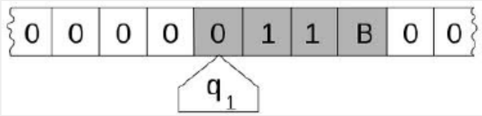
CRNs have a wide range of behaviors:

Boolean logic

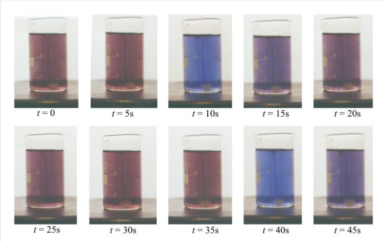


analog computing

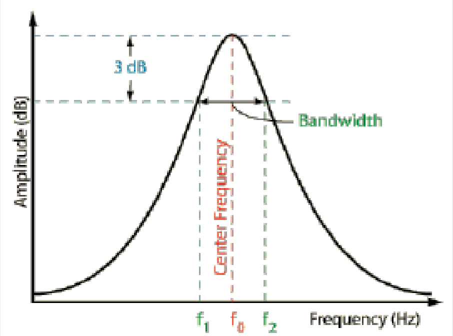
discrete algorithms



oscillation

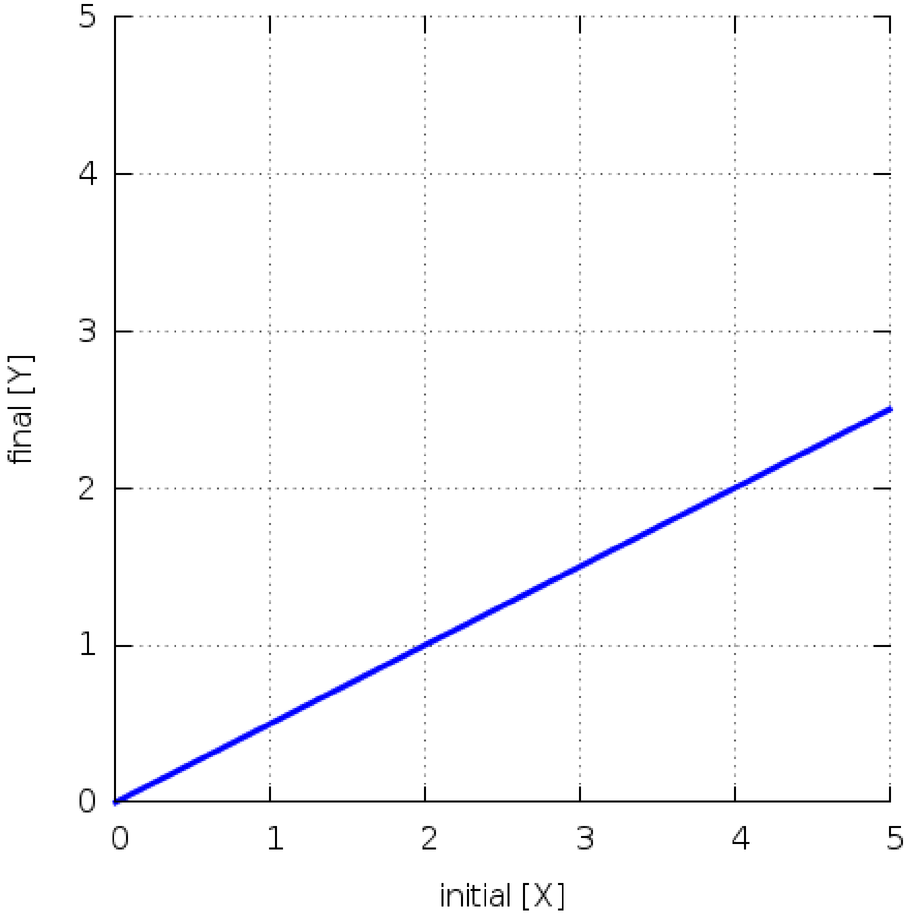


signal processing



CRN computation

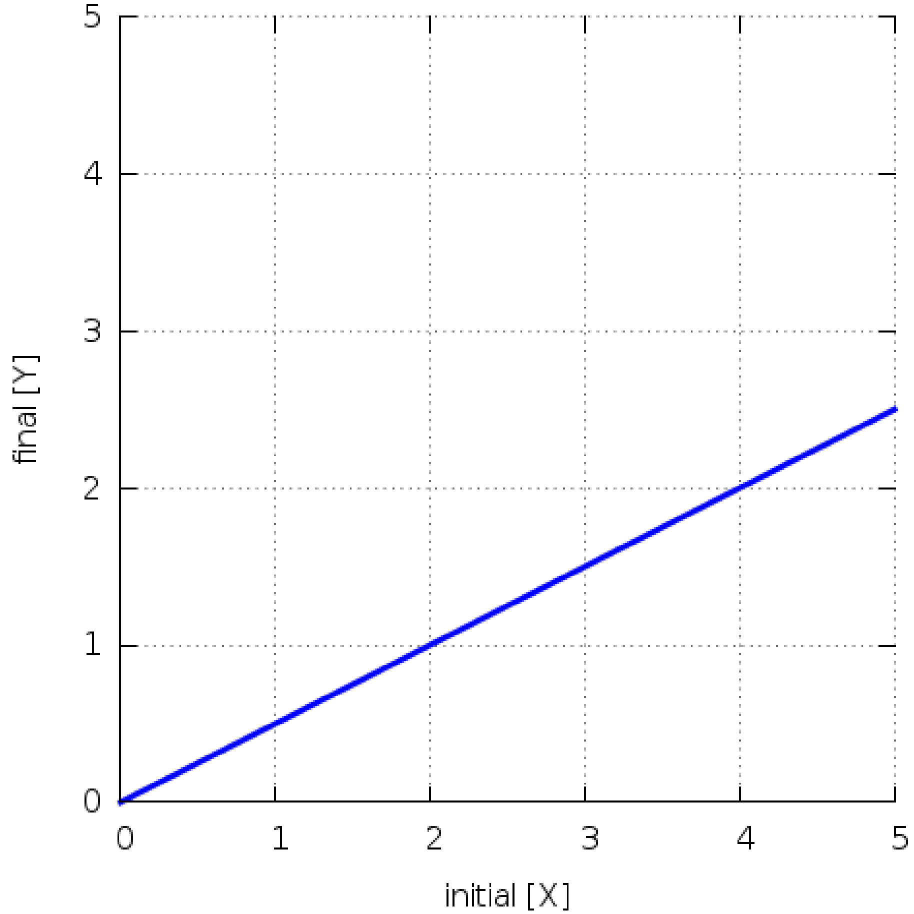
$f(x) = x/2$



starting with some concentration x of X, want to end with $x/2$ of Y

CRN computation

$f(x) = x/2$

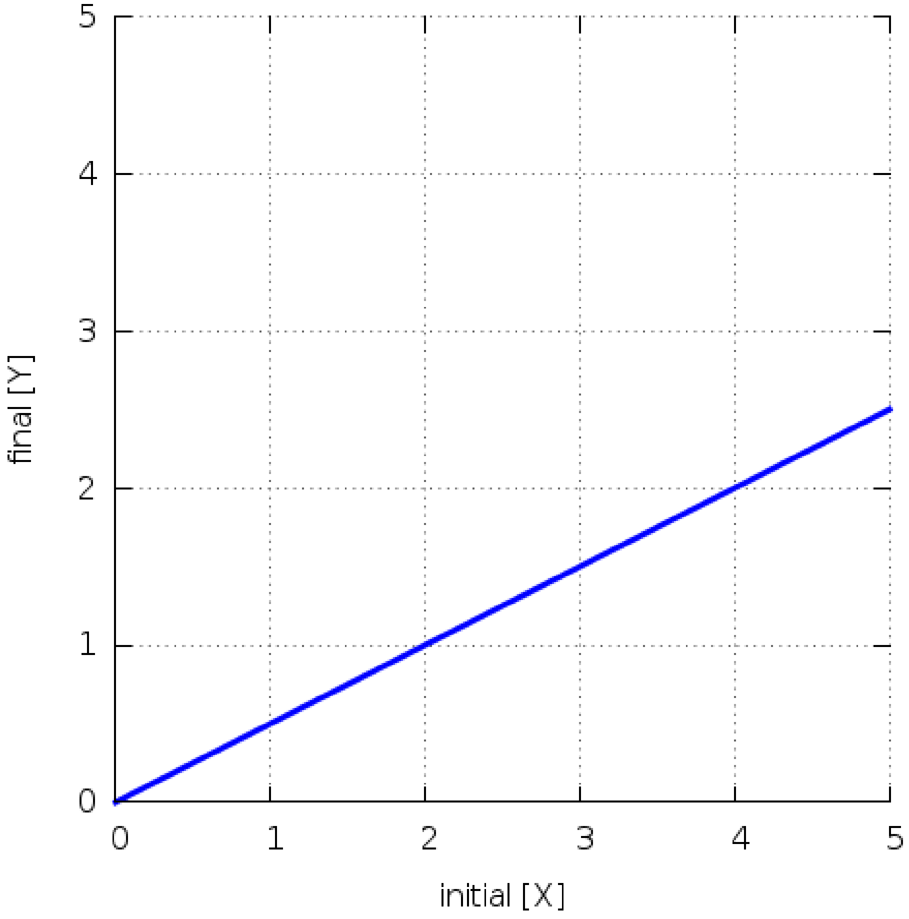


starting with some concentration x of X, want to end with $x/2$ of Y

$$X \xrightarrow{1} Y$$
$$Y \xrightarrow{1} X$$

CRN computation

$$f(x) = x/2$$



starting with some concentration x of X, want to end with $x/2$ of Y

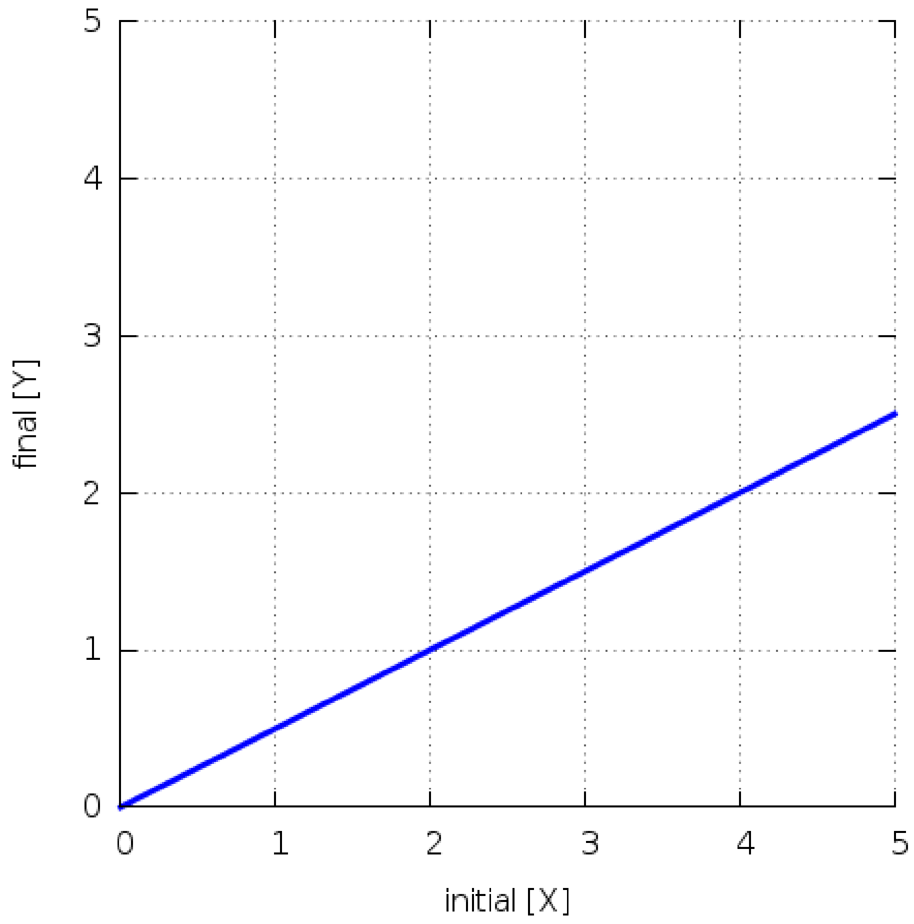
$$X \xrightarrow{1} Y$$
$$Y \xrightarrow{1} X$$

$d[Y]/dt = [X] - [Y]$
$= 0$ when $[Y]$ stops changing

CRN computation

$f(x) = x \bullet 2/3$

$\cancel{f(x) = x/2}$



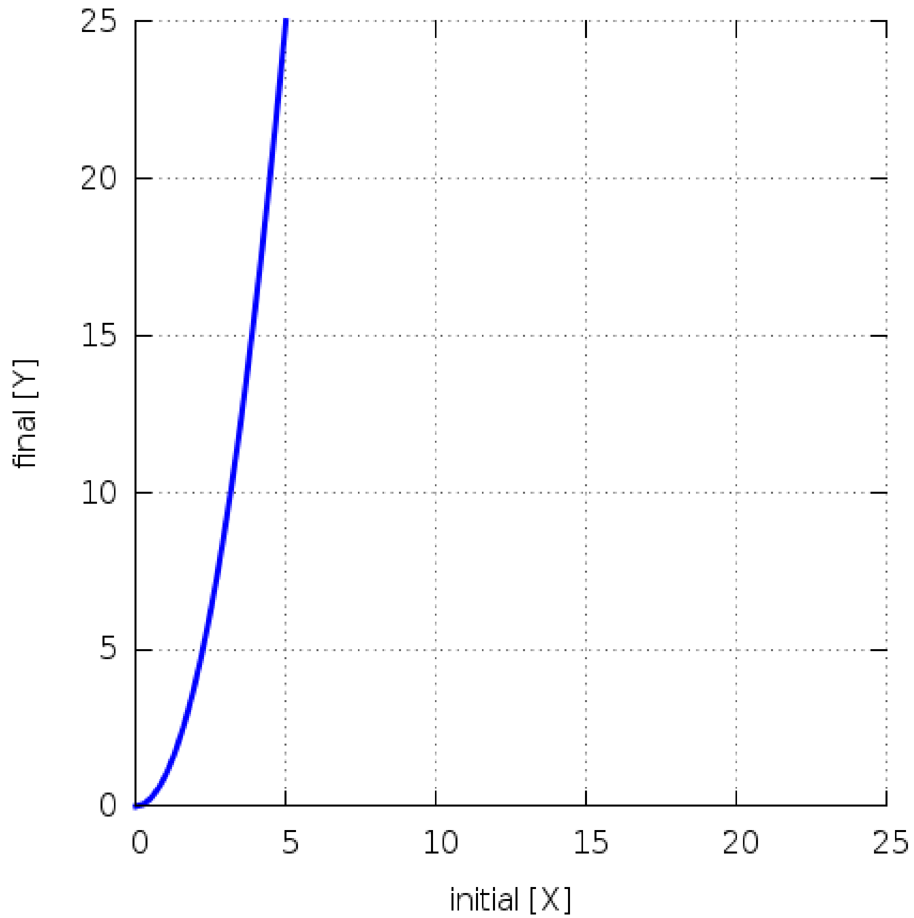
starting with some concentration *x* of *X*, want to end with *x*/2 of *Y*

$$X \xrightarrow{1} Y$$
$$Y \xrightarrow{2} X$$

$d[Y]/dt = [X] - 2[Y]$
$= 0$ when $[Y]$ stops changing

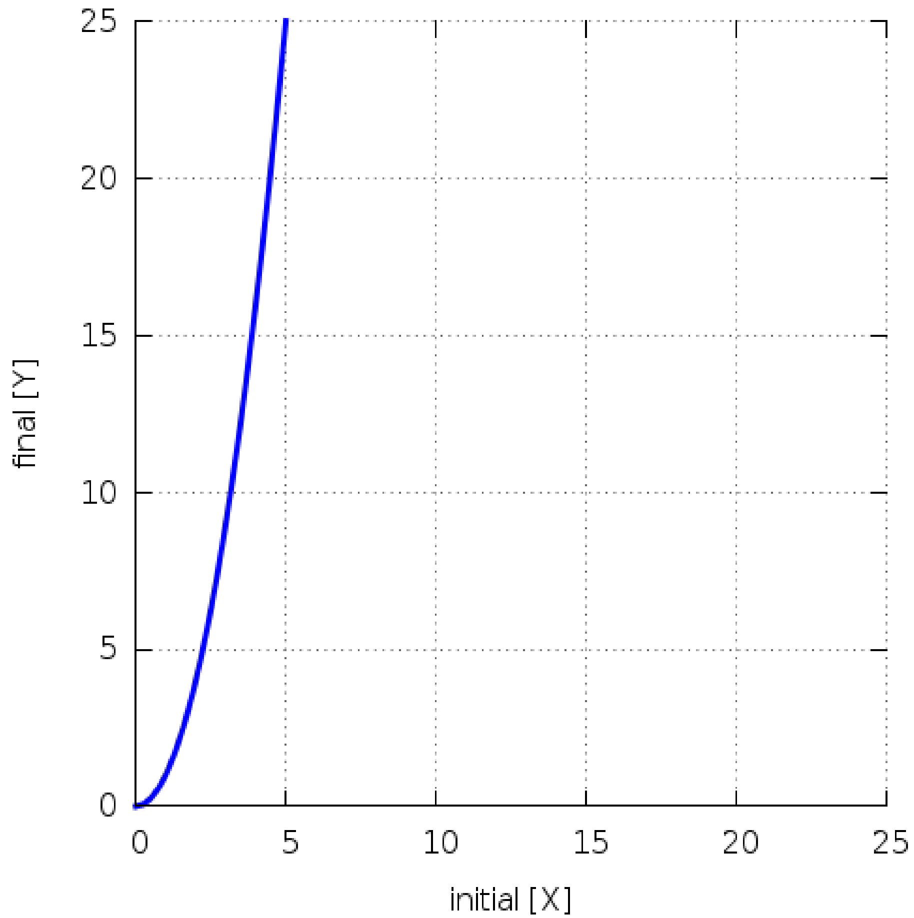
CRN computation

$$f(x) = x^2$$



CRN computation

$$f(x) = x^2$$

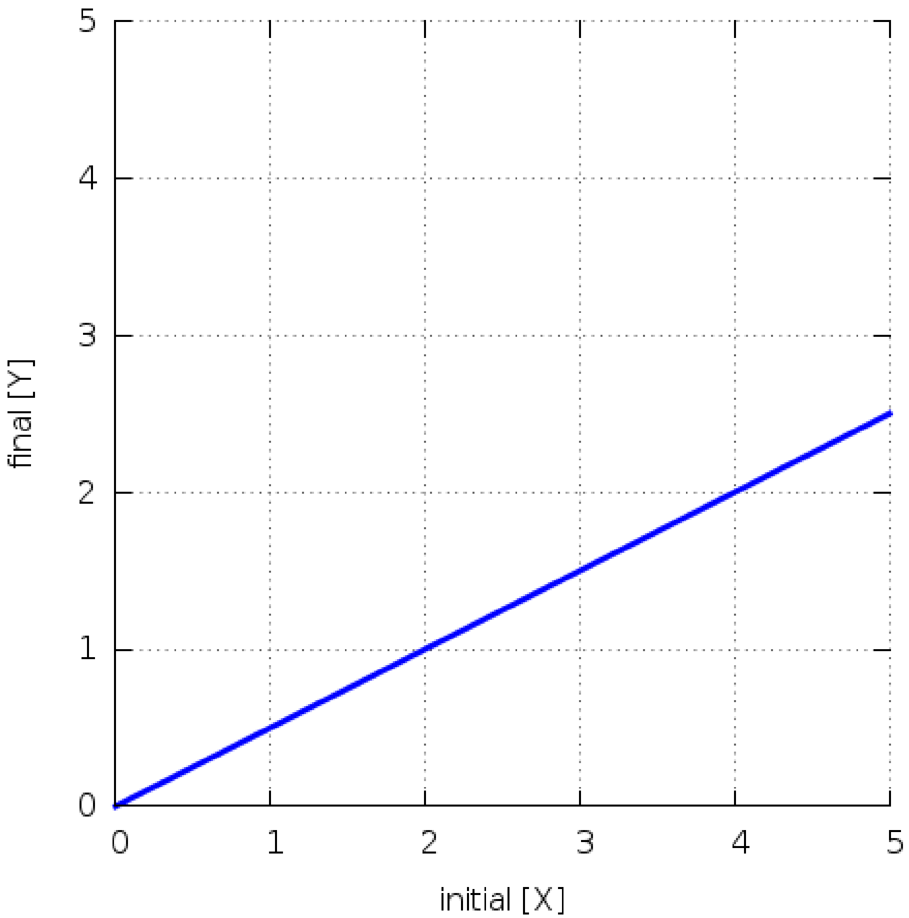


$$X + X \xrightarrow{1} X + X + Y$$
$$Y \xrightarrow{1}$$

$$d[Y]/dt = [X]^2 - [Y]$$

CRN computation

$f(x) = x/2$

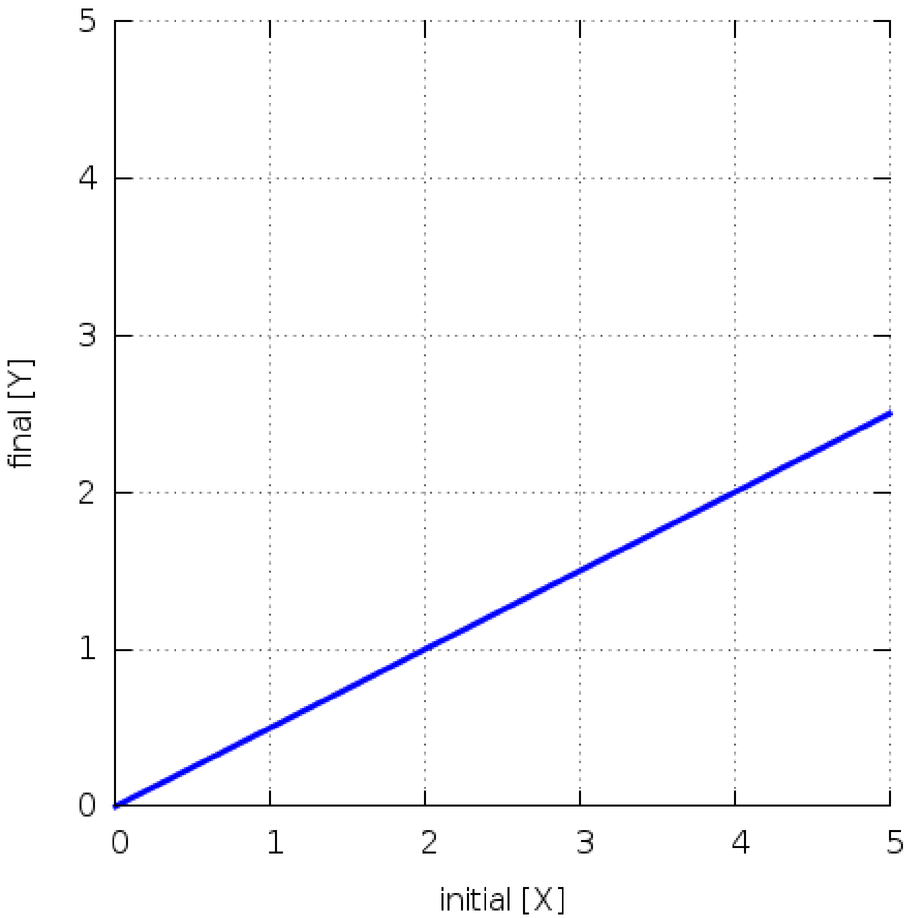


$$X \xrightarrow{1} Y$$
$$X \xrightarrow{1}$$

CRN computation

$f(x) = x/3$

~~$f(x) = x/2$~~



$X \xrightarrow{1} Y$

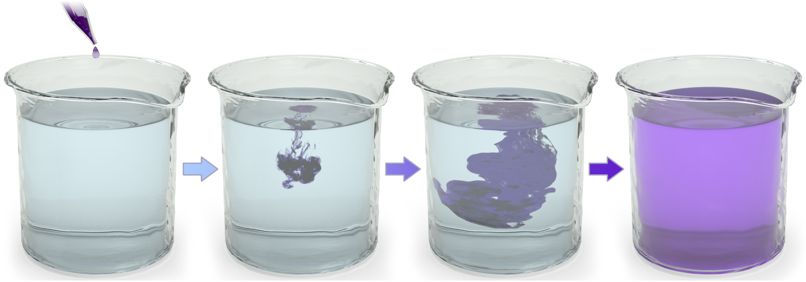
$X \xrightarrow{2}$

Rate-independent CRN computation

What can CRNs can compute when we don't know/can't control the rates?

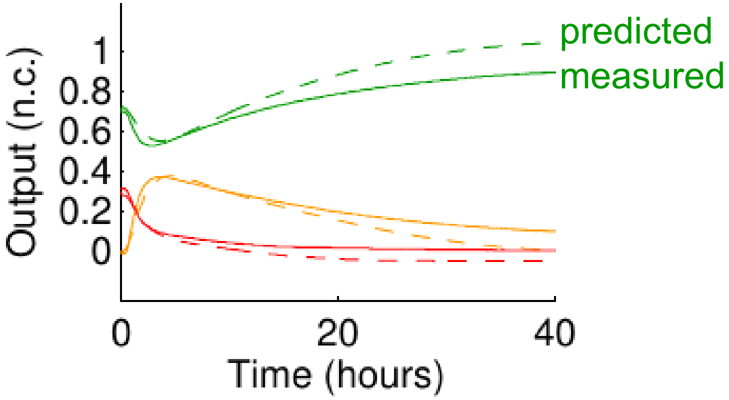
Rate-independent CRN computation

What can CRNs can compute when
we don't know/can't control the rates?



Rate-independent CRN computation

What can CRNs can compute when we don't know/can't control the rates?



How will the system evolve?

$$E + R \rightarrow E + P$$

How will the system evolve?

$$E + R \rightarrow E + P$$

mass-action rate: $k \cdot [E] \cdot [R]$

How will the system evolve?

$$E + R \rightarrow E + P$$

mass-action rate: $k \cdot [E] \cdot [R]$

Michaelis-Menten rate: $k_{cat} \cdot [E] \cdot [R] \; / \; (K_m + [R])$

How will the system evolve?

$$E + R \rightarrow E + P$$

mass-action rate: $k \cdot [E] \cdot [R]$

Michaelis-Menten rate: $k_{cat} \cdot [E] \cdot [R] \;/\; (K_m + [R])$

Hill rate: $k_{cat} \cdot [E] \cdot [R]^n \;/\; (K_m + [R]^n)$

How ~~will~~ might the system evolve?

$$E + R \rightarrow E + P$$

~~mass-action rate:~~ $\cancel{k \cdot [E] \cdot [R]}$

~~Michaelis-Menten rate:~~ $\cancel{k_{cat} \cdot [E] \cdot [R] \ / \ (K_m + [R])}$

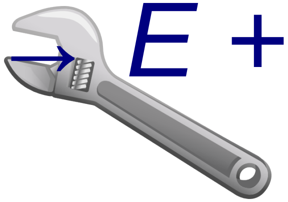
~~Hill rate:~~ $\cancel{k_{cat} \cdot [E] \cdot [R]^n \ / \ (K_m + [R]^n)}$

"if $\mathbf{s}(E) > 0$ and $\mathbf{s}(R) \geq x$, then it is possible to reach from state $\mathbf{s}$ to state $\mathbf{s} + \{x\ P\} - \{x\ R\}$"

How ~~will~~ the system evolve?
might

system evolves
adversarially

$E + R \to E + P$



~~mass-action rate:~~ ~~$k \cdot [E] \cdot [R]$~~

~~Michaelis-Menten rate:~~ ~~$k_{cat} \cdot [E] \cdot [R] \;/\; (K_m + [R])$~~

~~Hill rate:~~ ~~$k_{cat} \cdot [E] \cdot [R]^n \;/\; (K_m + [R]^n)$~~

“if **s**(E) > 0 and **s**(R) ≥ x, then it is <u>possible</u> to reach from state **s** to state **s** + {x P} – {x R}”

Worst-case analysis: Bad science but good engineering

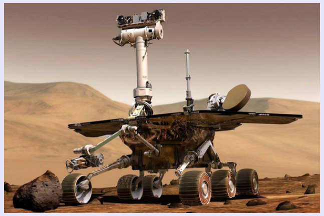
Worst-case analysis: Bad science but good engineering

Science		Engineering
understand nature	goal	build systems that work in nature

Worst-case analysis:
Bad science but good engineering

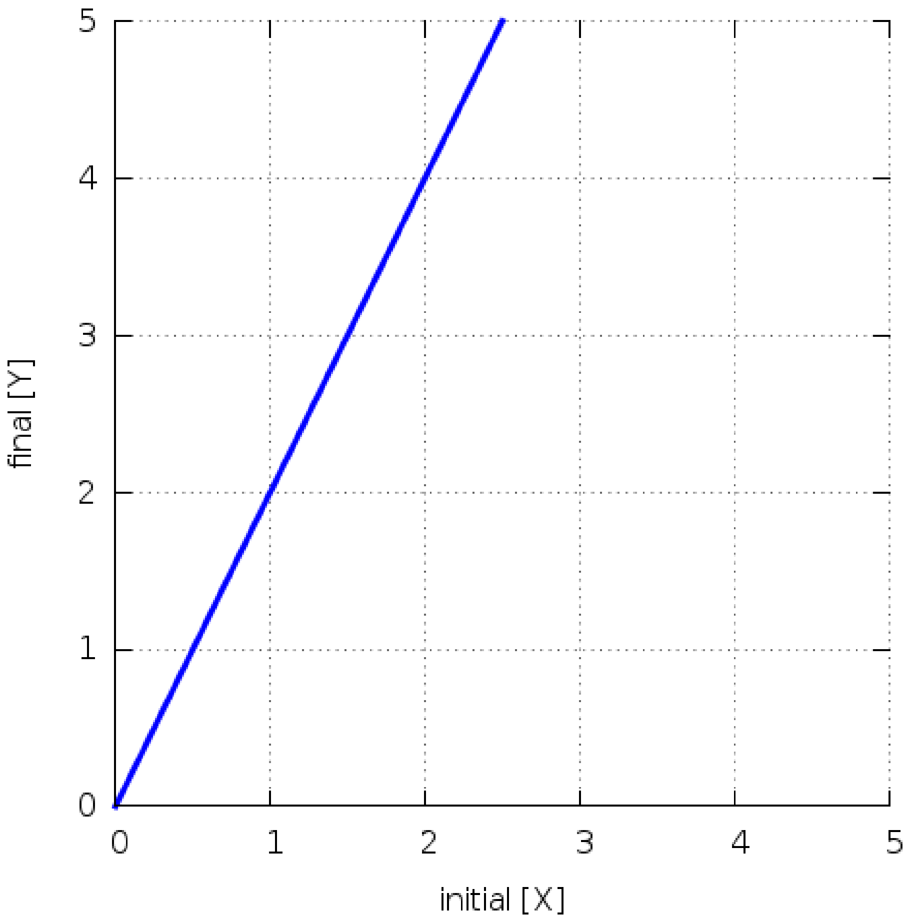
Science		Engineering
understand nature	goal	build systems that work in nature
predict behavior of nature	role of models	constrain how nature can affect system

Worst-case analysis:
Bad science but good engineering

Science		Engineering
understand nature	goal	build systems that work in nature
predict behavior of nature	role of models	constrain how nature can affect system
“How many bits *actually get flipped* in a message from Mars Rover?”	example 	“How many bit flips can my error-correcting code *tolerate*?”

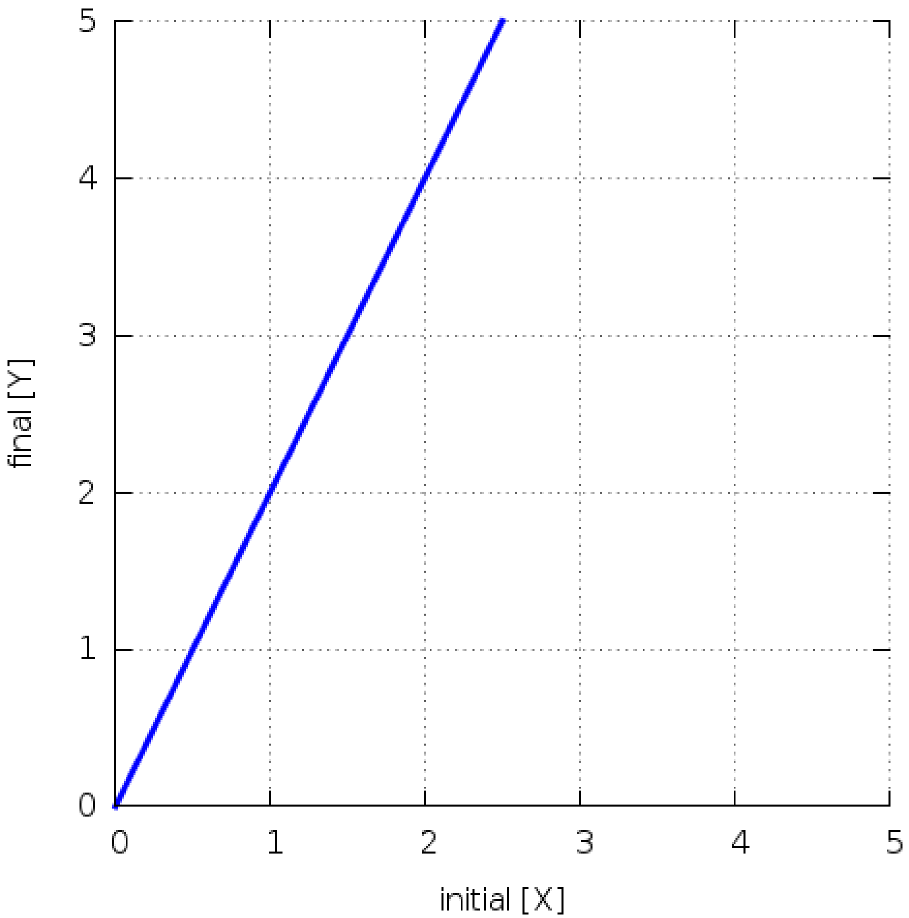
Multiplication by a constant

$$f(x) = 2x$$



Multiplication by a constant

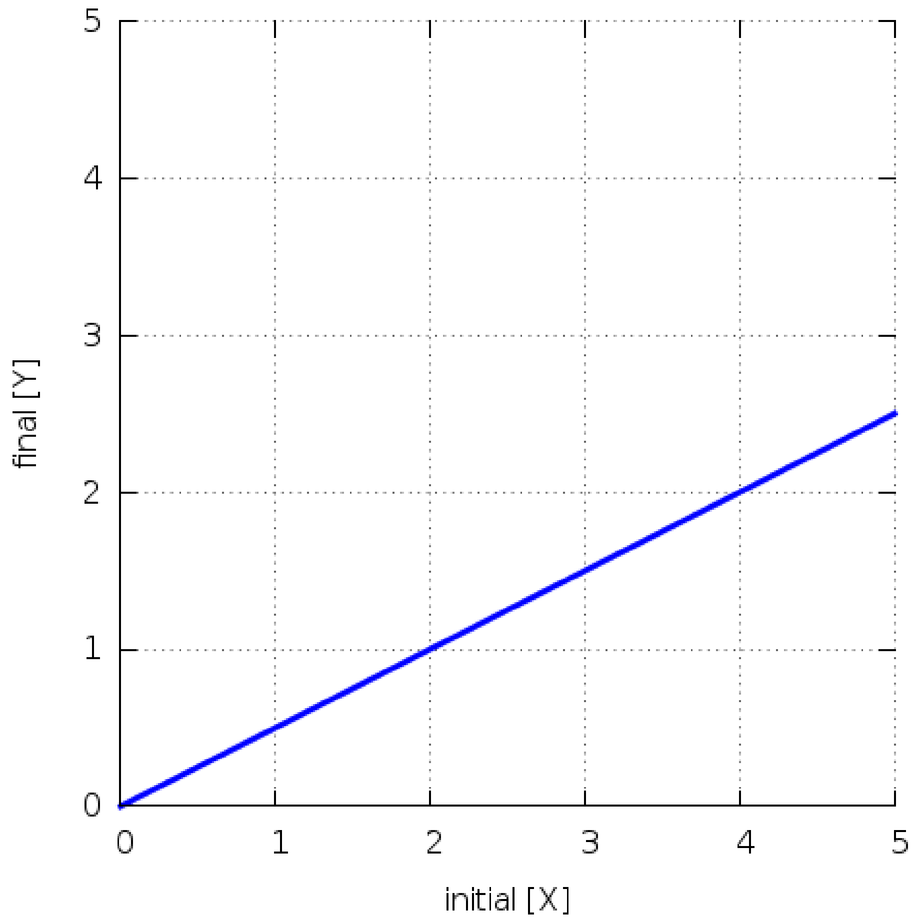
$$f(x) = 2x$$



$$X \rightarrow Y + Y$$

Division by a constant

$$f(x) = x/2$$



rate-*dependent* CRNs computing x/2

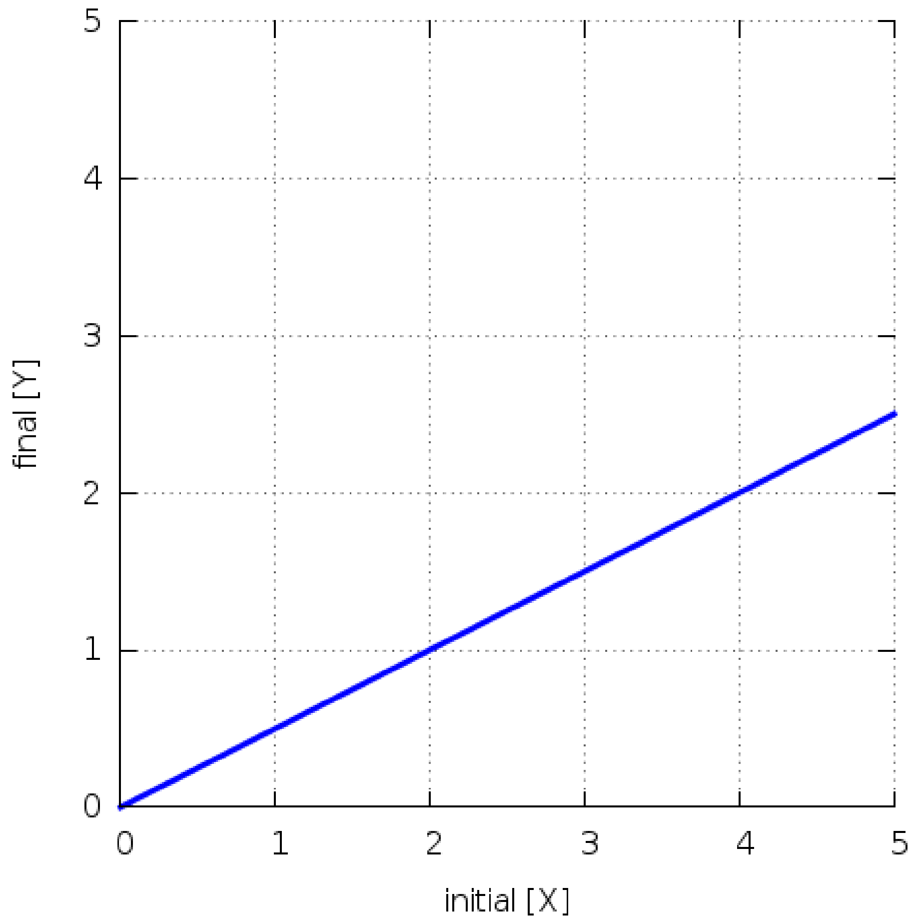
$$X \xrightarrow{1} Y \qquad Y \xrightarrow{1} X$$

$$X \xrightarrow{1} Y \qquad X \xrightarrow{1}$$

rate-*independent* CRN computing x/2?

Division by a constant

$f(x) = x/2$



rate-*dependent* CRNs computing x/2

$$X \xrightarrow{1} Y \qquad Y \xrightarrow{1} X$$

$$X \xrightarrow{1} Y \qquad X \xrightarrow{1}$$

rate-*independent* CRN computing x/2?

$$X + X \rightarrow Y$$

CRNs can compute in nonintuitive ways

$$f(x) = x/2$$

$$X \to Y + A$$

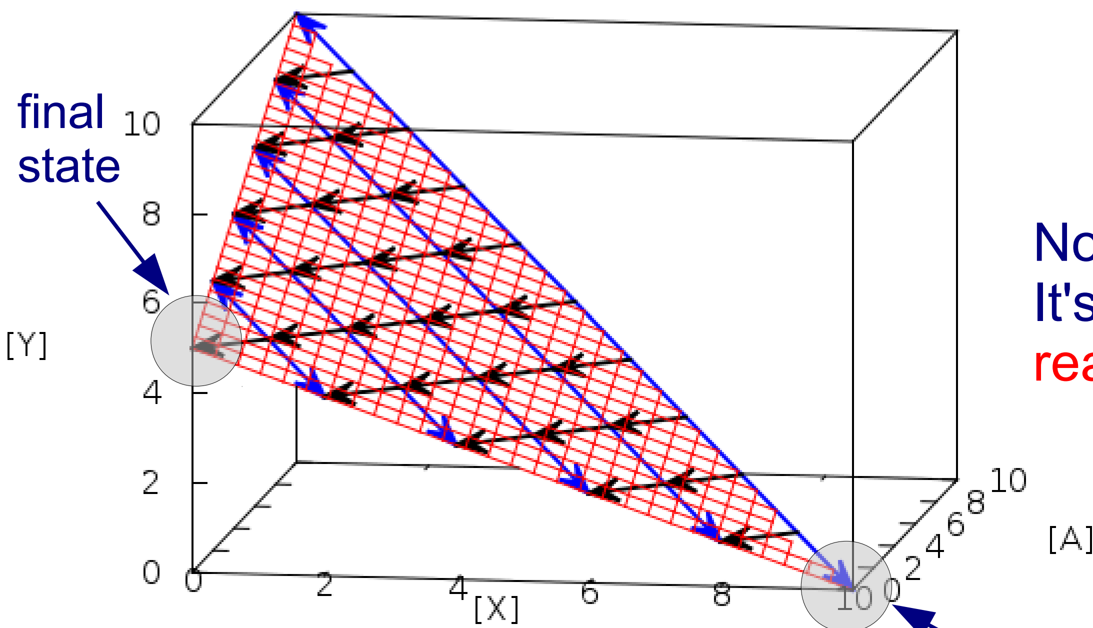
$$Y + A \to X$$

$$X + A \to$$

CRNs can compute in nonintuitive ways

$f(x) = x/2$

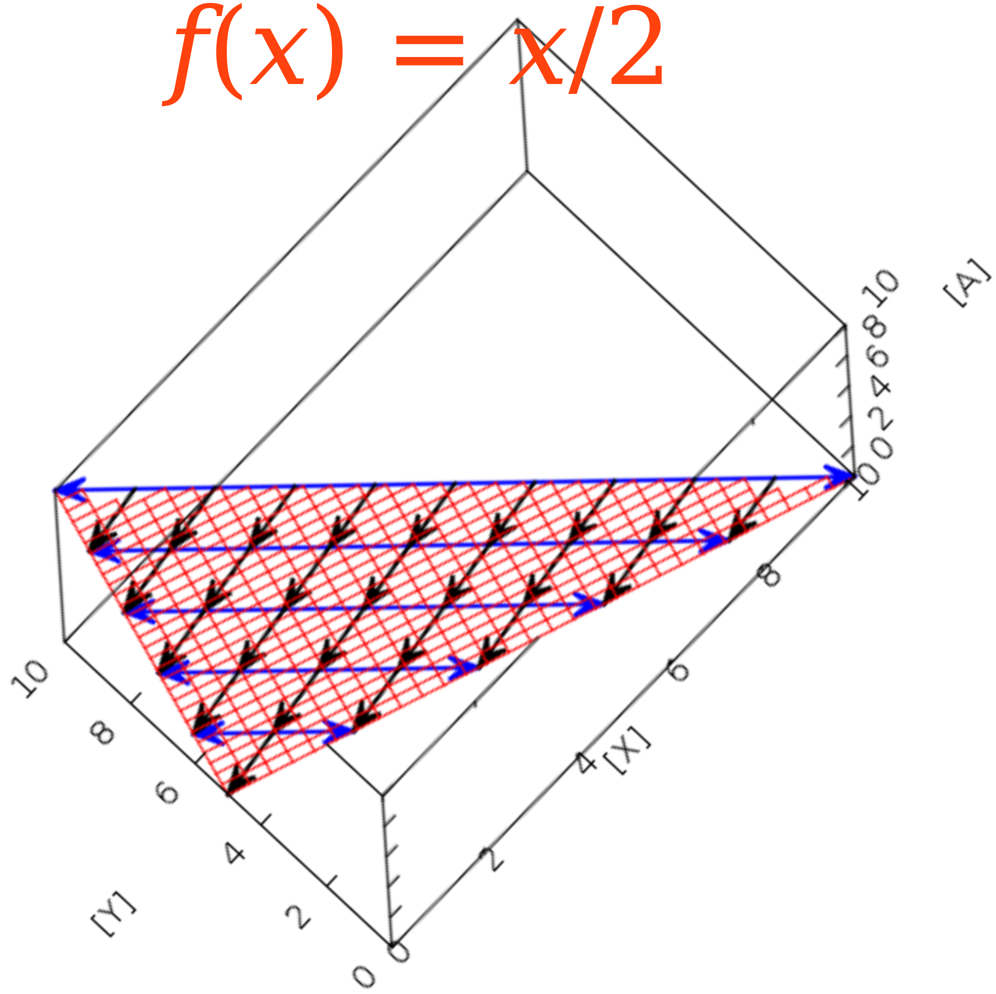
$$X \to Y + A$$
$$Y + A \to X$$
$$X + A \to$$



Not a plot of *f*!
It's the space of
reachable states.

CRNs can compute in nonintuitive ways

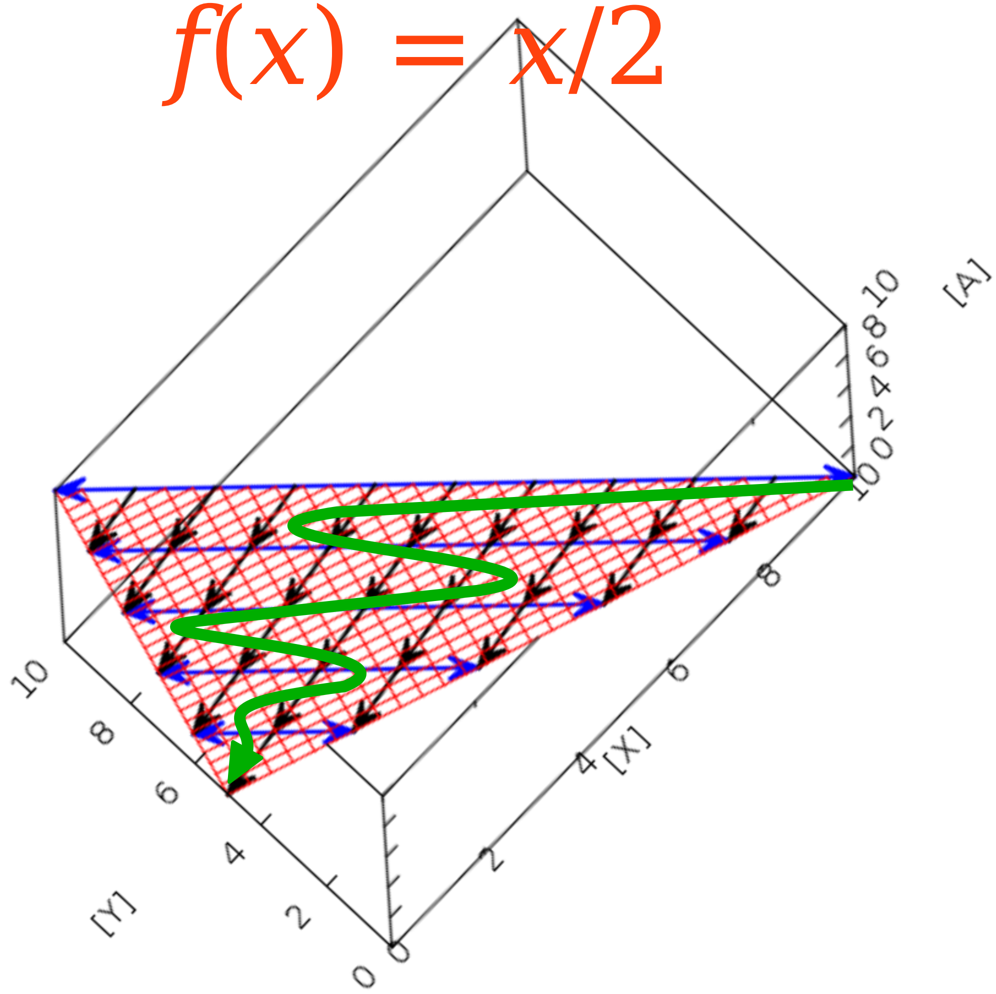
$$f(x) = x/2$$



$$X \rightarrow Y + A$$
$$Y + A \rightarrow X$$
$$X + A \rightarrow$$

CRNs can compute in nonintuitive ways

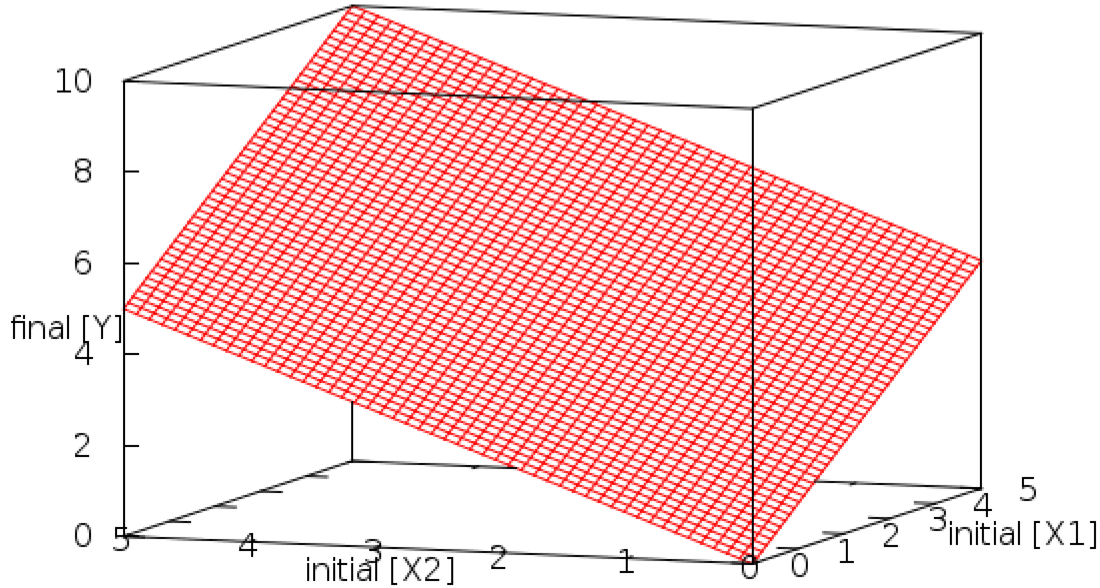
$f(x) = x/2$



$X \rightarrow Y + A$
$Y + A \rightarrow X$
$X + A \rightarrow$

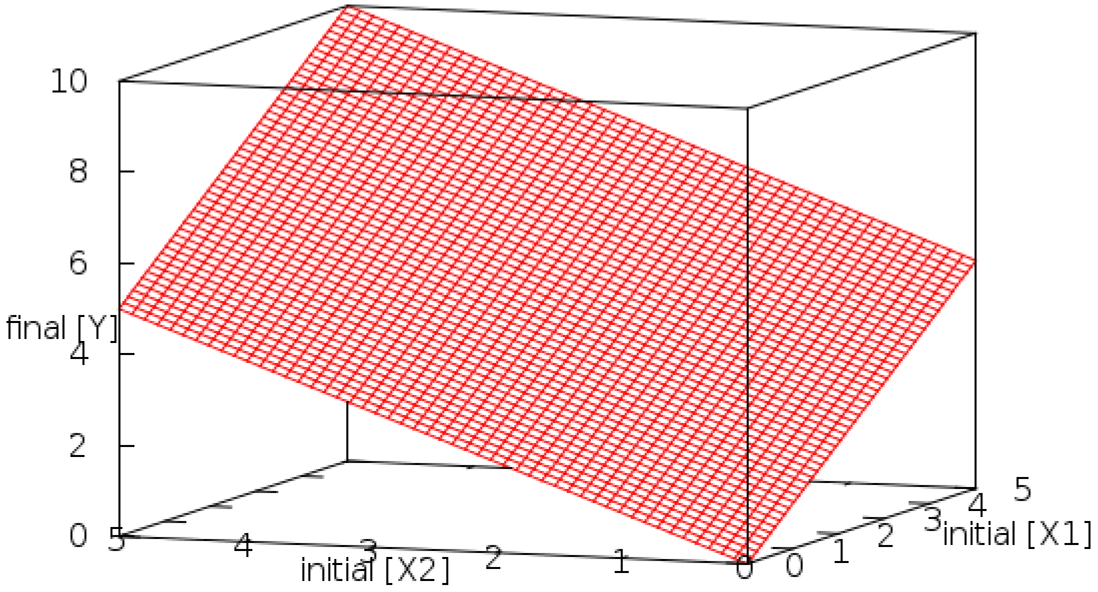
Multiple inputs

$$f(x_1, x_2) = x_1 + x_2$$



Multiple inputs

$$f(x_1,x_2) = x_1 + x_2$$

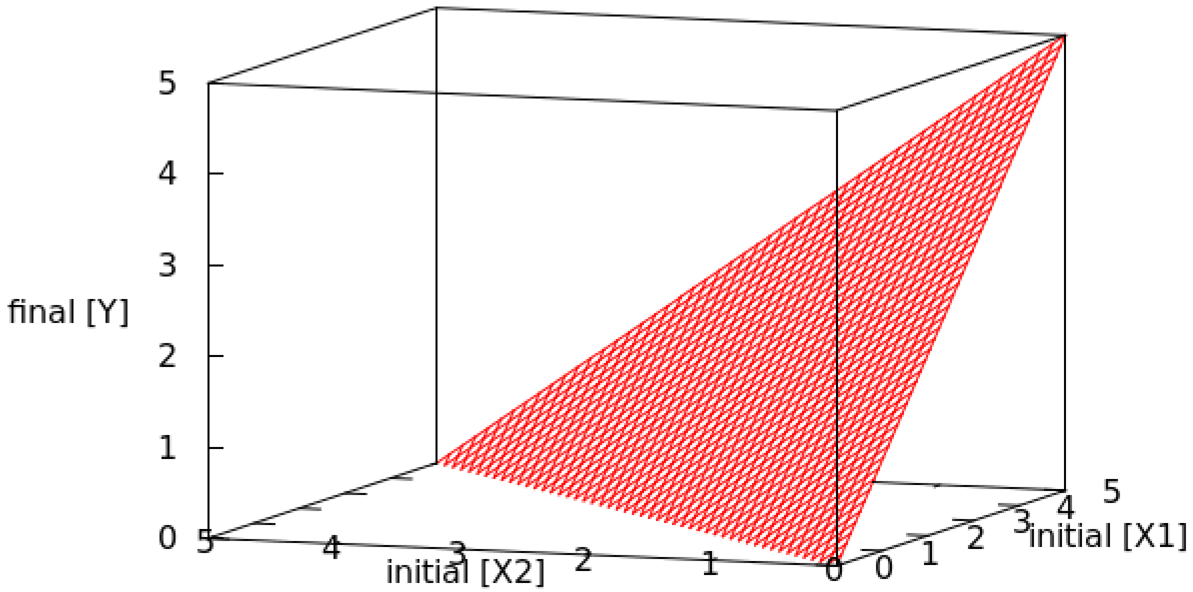


$X_1 \rightarrow Y$

$X_2 \rightarrow Y$

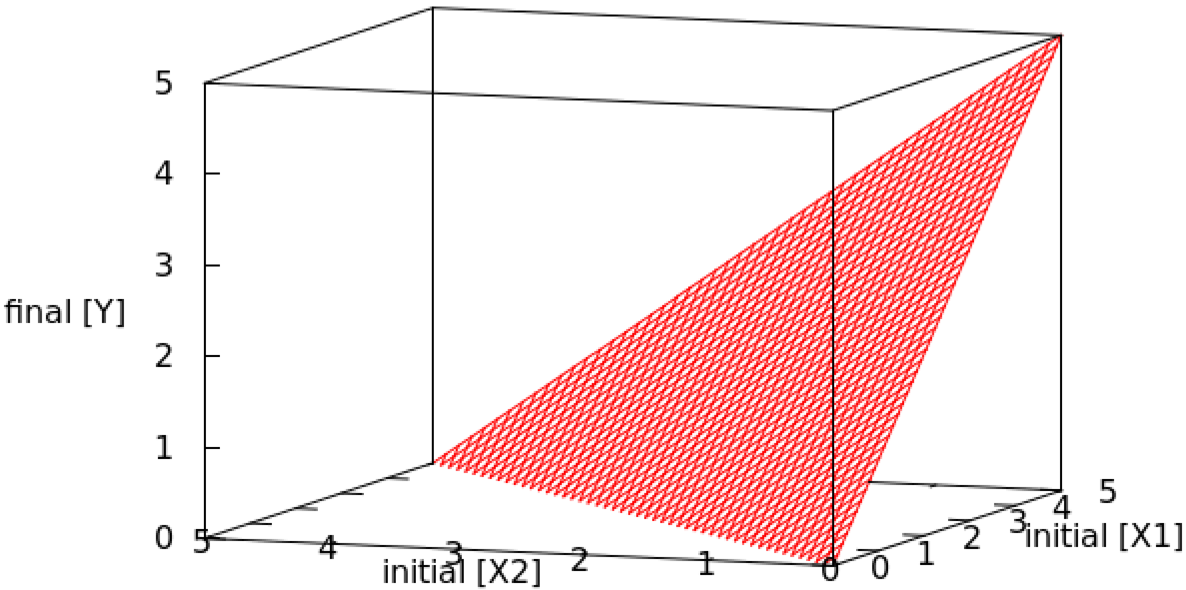
Subtraction

$$f(x_1,x_2) = x_1 - x_2$$



Subtraction

$$f(x_1,x_2) = x_1 - x_2$$



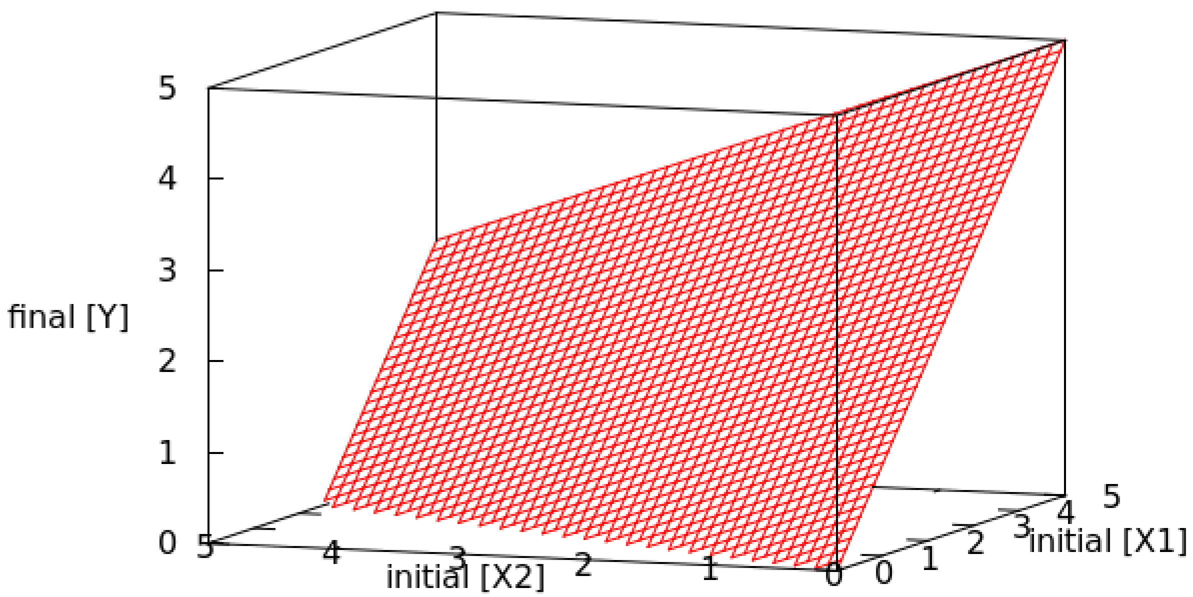
$$X_1 \rightarrow Y$$

$$X_2 + Y \rightarrow$$

Sometimes necessary to consume output

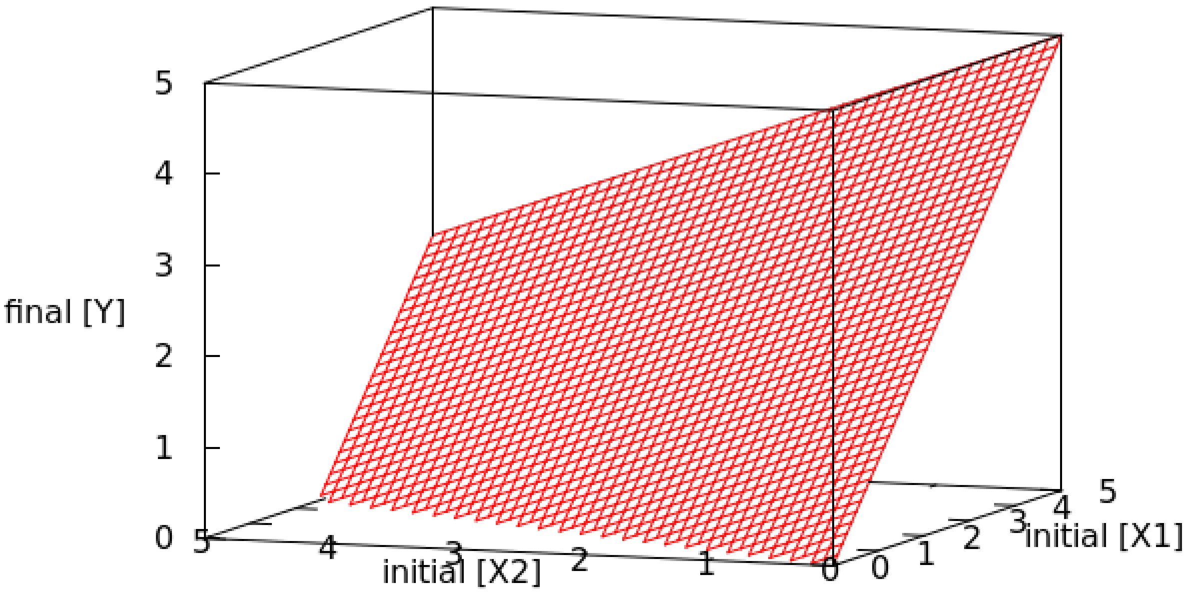
Combining previous ideas

$$f(x_1,x_2) = x_1 - x_2/2$$



Combining previous ideas

$f(x_1,x_2) = x_1 - x_2/2$



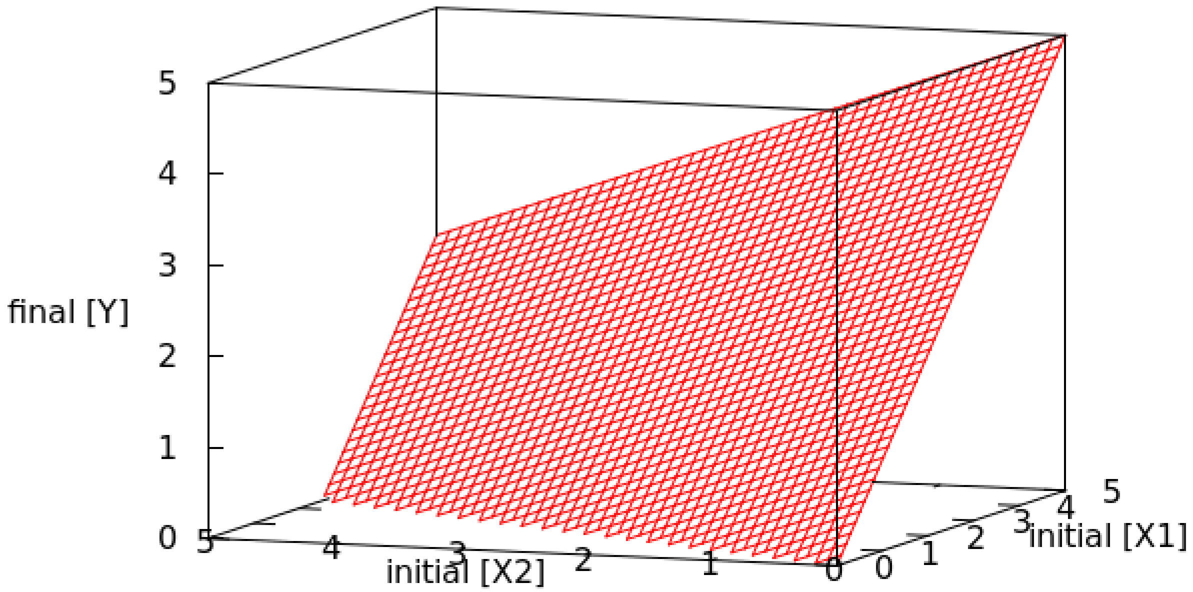
$X_1 \rightarrow Y$

$X_2 + X_2 + Y \rightarrow$

Combining previous ideas

$f(x_1,x_2) = x_1 - x_2/2$

Only linear functions possible?

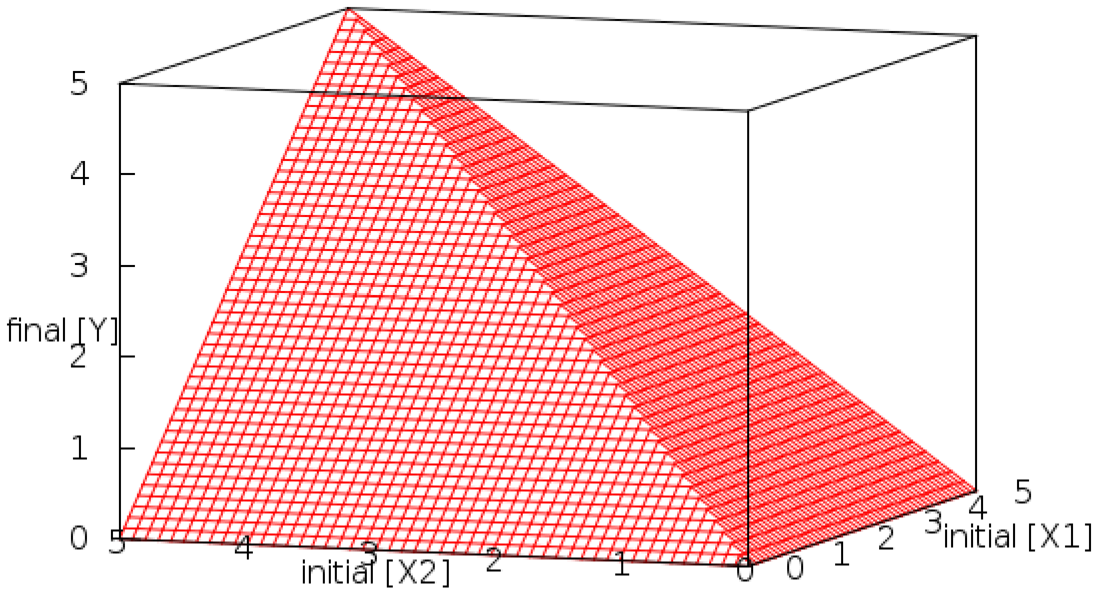


$X_1 \rightarrow Y$

$X_2 + X_2 + Y \rightarrow$

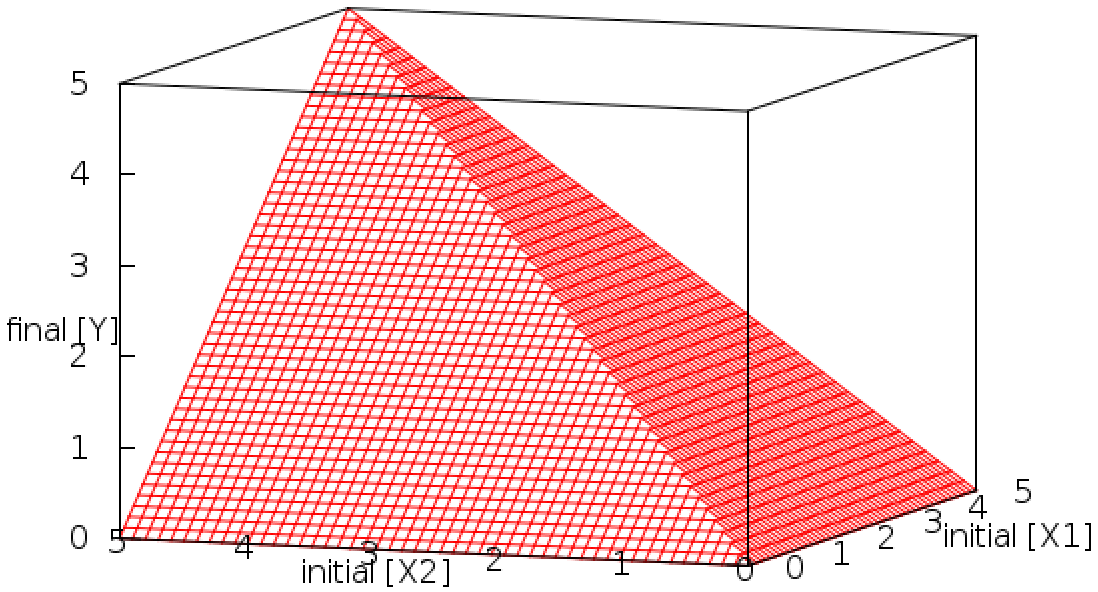
Nonlinear functions

$$f(x_1,x_2) = \min\{x_1,x_2\}$$



Nonlinear functions

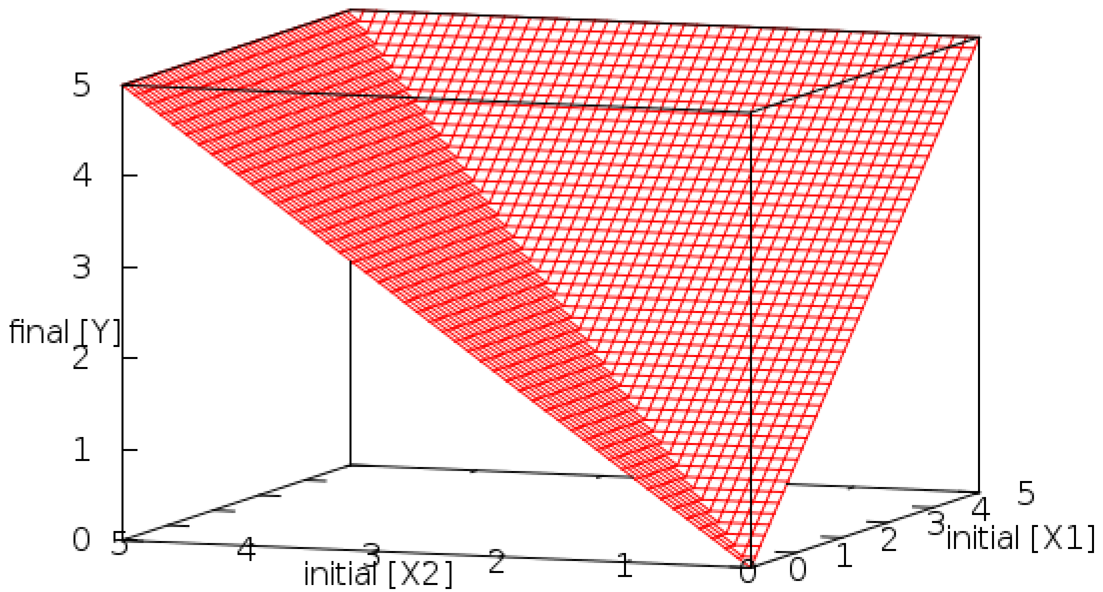
$$f(x_1,x_2) = \min\{x_1,x_2\}$$



$$X_1 + X_2 \rightarrow Y$$

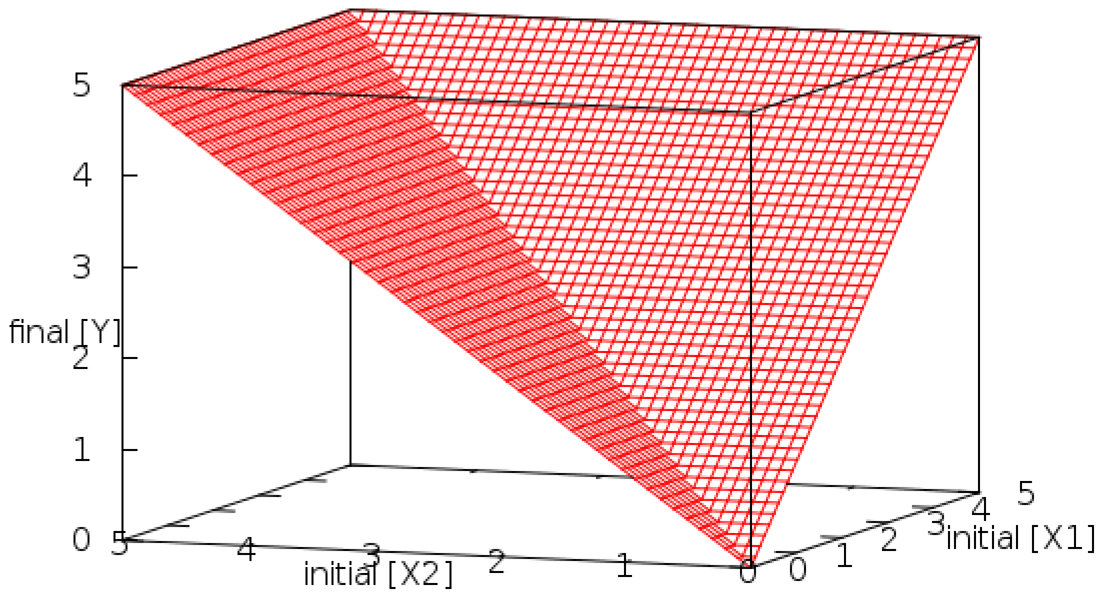
Last example

$f(x_1,x_2) = \max\{x_1,x_2\}$



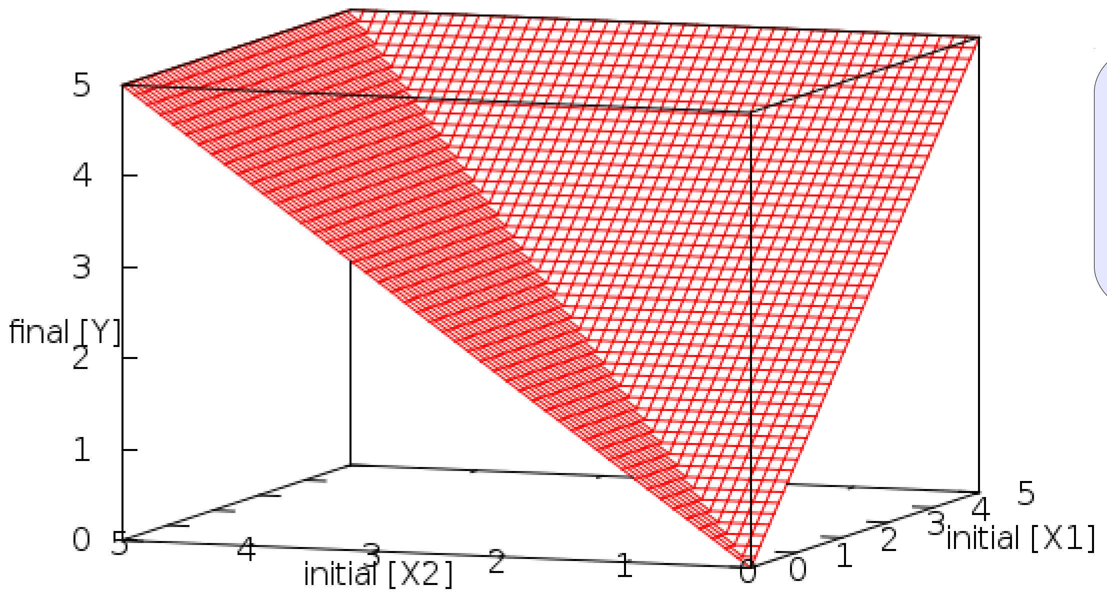
Last example

$$f(x_1,x_2) = \max\{x_1,x_2\} = x_1 + x_2 - \min\{x_1,x_2\}$$



Last example

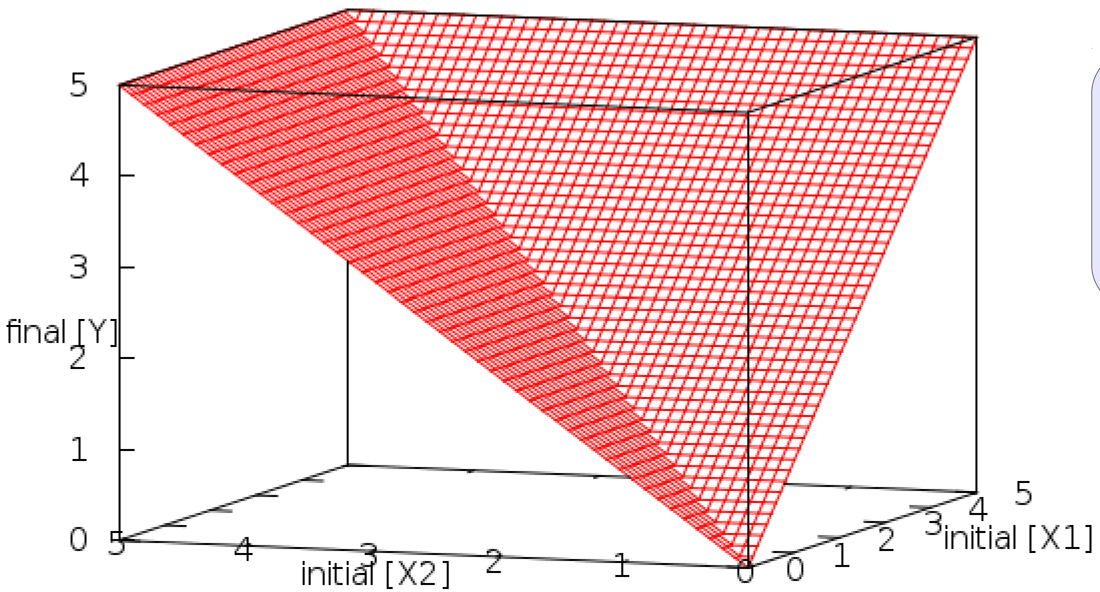
$$f(x_1,x_2) = \max\{x_1,x_2\} = \boxed{x_1 + x_2} - \min\{x_1,x_2\}$$



sum $\quad X_1 \rightarrow Y$
$\quad\quad X_2 \rightarrow Y$

Last example

$$f(x_1,x_2) = \max\{x_1,x_2\} = \boxed{x_1 + x_2} - \min\{x_1,x_2\}$$



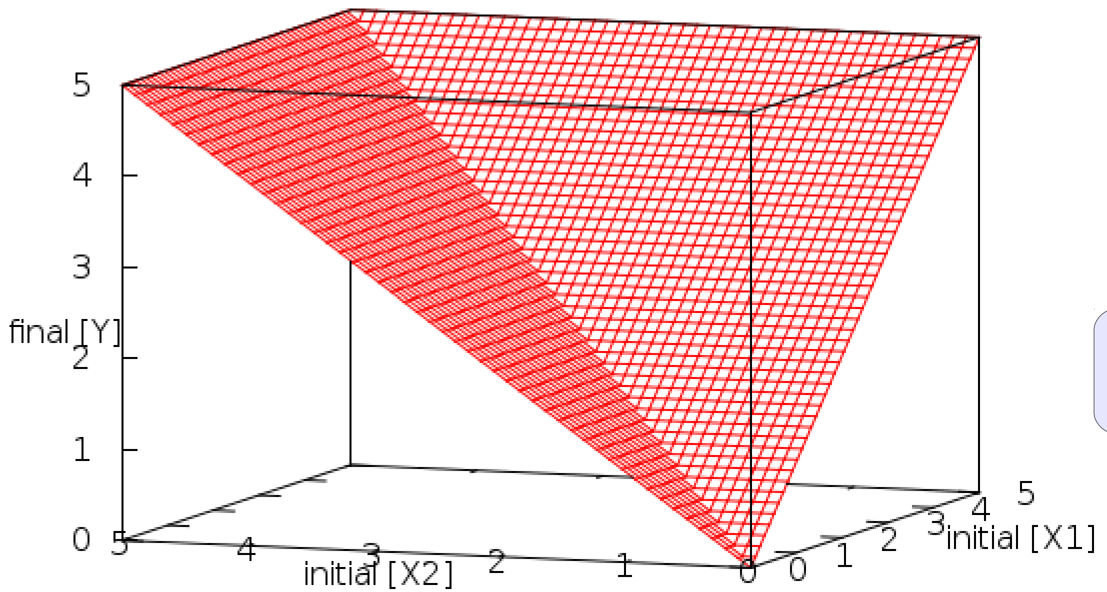
store input values

sum

$$X_1 \rightarrow Y + X_1'$$
$$X_2 \rightarrow Y + X_2'$$

Last example

$$f(x_1,x_2) = \max\{x_1,x_2\} = x_1 + x_2 - \boxed{\min\{x_1,x_2\}}$$



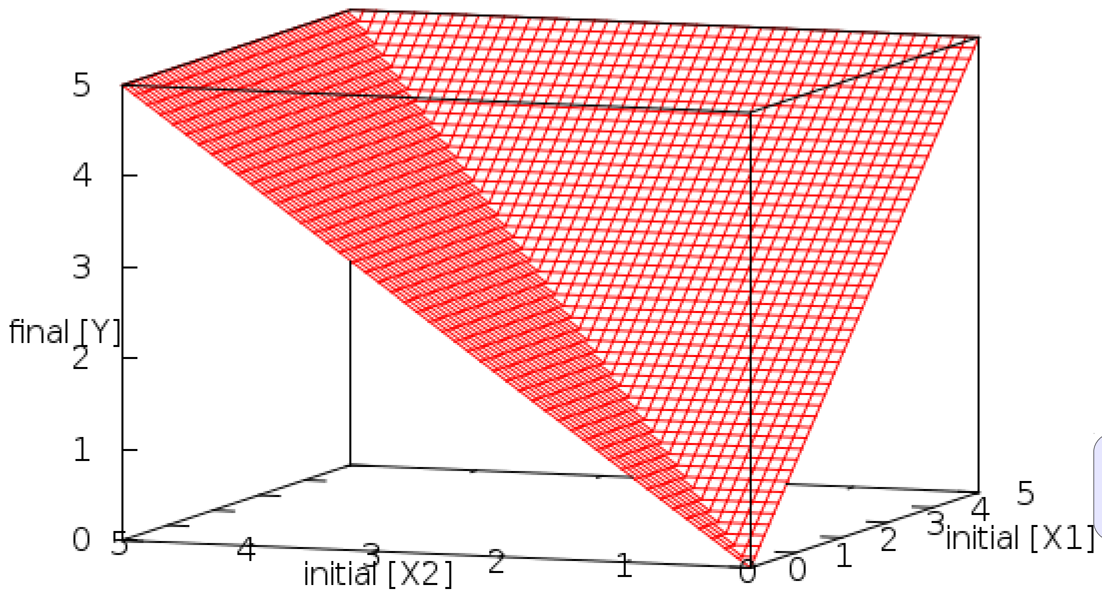
sum $X_1 \rightarrow Y + X_1'$

$X_2 \rightarrow Y + X_2'$

min $X_1' + X_2' \rightarrow K$

Last example

$$f(x_1,x_2) = \max\{x_1,x_2\} = x_1 + x_2 \boxed{-} \min\{x_1,x_2\}$$



sum $X_1 \rightarrow Y + X_1'$

$X_2 \rightarrow Y + X_2'$

min $X_1' + X_2' \rightarrow K$

subtract $K + Y \rightarrow$

Other functions?

$f(x) = x^2$?

$f(x_1, x_2) = x_1 \bullet x_2$?

$f(x) = 2^x$?

$f(x) = x \bullet \sqrt{2}$?

Other functions?

$f(x) = x^2$?

$f(x_1,x_2) = x_1 \bullet x_2$?

$f(x) = 2^x$?

$f(x) = x \bullet \sqrt{2}$?

Rate-independent CRN computation (formal definition)

task: compute function $y = f(x_1,\ldots,x_k)$, $\quad y,x_1,\ldots,x_k \in \mathbb{R}_{>0}$

Rate-independent CRN computation (formal definition)

task: compute function $y = f(x_1,\ldots,x_k)$, $y,x_1,\ldots,x_k \in \mathbb{R}_{>0}$

- **initial state**: concentrations $[X_1],\ldots,[X_k]$

Rate-independent CRN computation (formal definition)

task: compute function $y = f(x_1,\ldots,x_k)$, $y,x_1,\ldots,x_k \in \mathbb{R}_{>0}$

- **initial state**: concentrations $[X_1],\ldots,[X_k]$
- **output**: $[Y]$

Rate-independent CRN computation (formal definition)

task: compute function $y = f(x_1,\ldots,x_k)$, $\quad y,x_1,\ldots,x_k \in \mathbb{R}_{>0}$

- **initial state**: concentrations $[X_1],\ldots,[X_k]$
- **output**: $[Y]$
- **output-stable state**: all states reachable from it have same $[Y]$

Rate-independent CRN computation (formal definition)

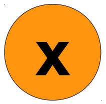
task: compute function $y = f(x_1,\ldots,x_k)$, $y,x_1,\ldots,x_k \in \mathbb{R}_{>0}$

- **initial state**: concentrations $[X_1],\ldots,[X_k]$
- **output**: $[Y]$
- **output-stable state**: all states reachable from it have same $[Y]$
- **rate-independent computation**: if a state **c** is reachable from the initial state **x**, a correct output-stable state **o** is reachable from **c**

Rate-independent CRN computation (formal definition)

task: compute function $y = f(x_1,\ldots,x_k)$, $\quad y,x_1,\ldots,x_k \in \mathbb{R}_{>0}$

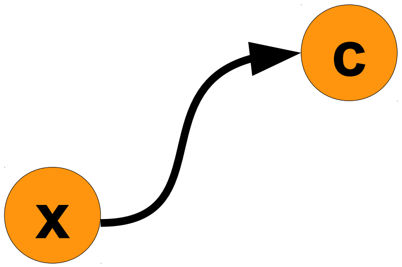
- **initial state**: concentrations $[X_1],\ldots,[X_k]$
- **output**: $[Y]$
- **output-stable state**: all states reachable from it have same $[Y]$
- **rate-independent computation**: if a state **c** is reachable from the initial state **x**, a correct output-stable state **o** is reachable from **c**



Rate-independent CRN computation (formal definition)

task: compute function $y = f(x_1,\ldots,x_k)$, $y,x_1,\ldots,x_k \in \mathbb{R}_{>0}$

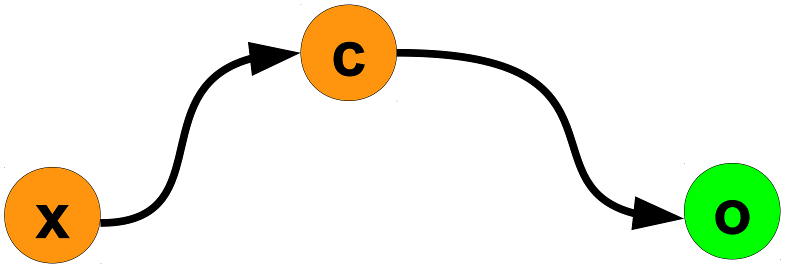
- **initial state**: concentrations $[X_1],\ldots,[X_k]$
- **output**: $[Y]$
- **output-stable state**: all states reachable from it have same $[Y]$
- **rate-independent computation**: if a state **c** is reachable from the initial state **x**, a correct output-stable state **o** is reachable from **c**



Rate-independent CRN computation (formal definition)

task: compute function $y = f(x_1,\ldots,x_k)$, $y,x_1,\ldots,x_k \in \mathbb{R}_{>0}$

- **initial state**: concentrations $[X_1],\ldots,[X_k]$
- **output**: $[Y]$
- **output-stable state**: all states reachable from it have same $[Y]$
- **rate-independent computation**: if a state **c** is reachable from the initial state **x**, a correct output-stable state **o** is reachable from **c**

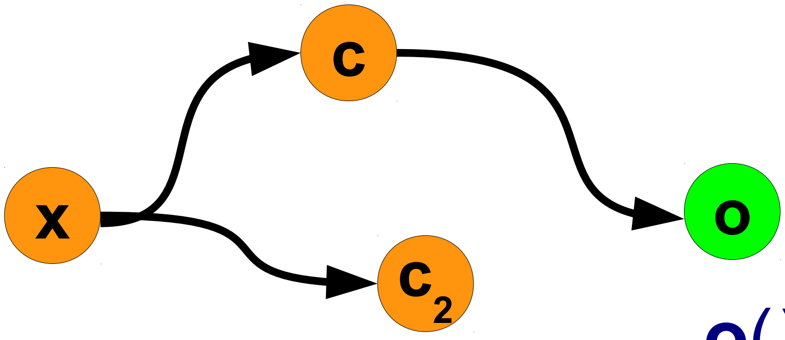


$\mathbf{o}(Y) = f(x_1,\ldots,x_k)$

Rate-independent CRN computation (formal definition)

task: compute function $y = f(x_1,...,x_k)$, $y,x_1,...,x_k \in \mathbb{R}_{>0}$

- **initial state**: concentrations $[X_1],...,[X_k]$
- **output**: $[Y]$
- **output-stable state**: all states reachable from it have same $[Y]$
- **rate-independent computation**: if a state **c** is reachable from the initial state **x**, a correct output-stable state **o** is reachable from **c**

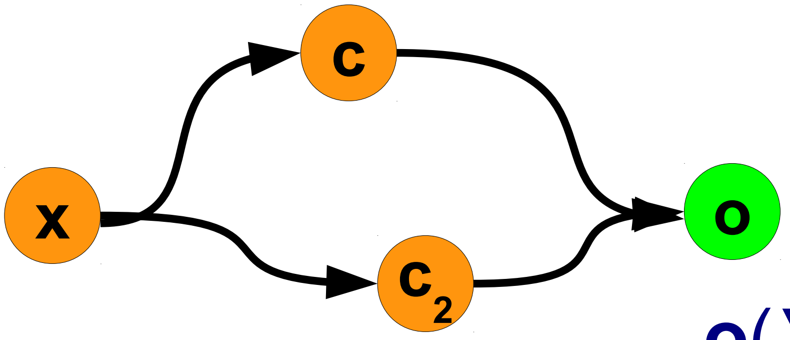


$\mathbf{o}(Y) = f(x_1,..,x_k)$

Rate-independent CRN computation (formal definition)

task: compute function $y = f(x_1,\ldots,x_k)$, $y,x_1,\ldots,x_k \in \mathbb{R}_{>0}$

- **initial state**: concentrations $[X_1],\ldots,[X_k]$
- **output**: $[Y]$
- **output-stable state**: all states reachable from it have same $[Y]$
- **rate-independent computation**: if a state **c** is reachable from the initial state **x**, a correct output-stable state **o** is reachable from **c**

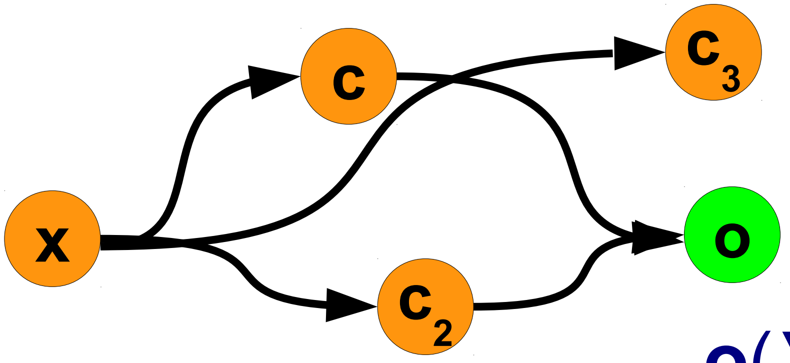


$\mathbf{o}(Y) = f(x_1,\ldots,x_k)$

Rate-independent CRN computation (formal definition)

task: compute function $y = f(x_1,\ldots,x_k)$, $\quad y,x_1,\ldots,x_k \in \mathbb{R}_{>0}$

- **initial state**: concentrations $[X_1],\ldots,[X_k]$
- **output**: $[Y]$
- **output-stable state**: all states reachable from it have same $[Y]$
- **rate-independent computation**: if a state **c** is reachable from the initial state **x**, a correct output-stable state **o** is reachable from **c**

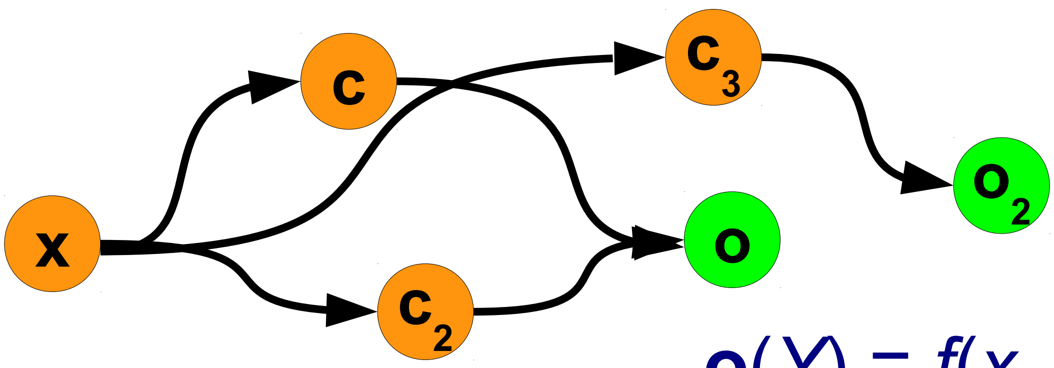


$\mathbf{o}(Y) = f(x_1,\ldots,x_k)$

Rate-independent CRN computation (formal definition)

task: compute function $y = f(x_1,\ldots,x_k)$, $\quad y,x_1,\ldots,x_k \in \mathbb{R}_{>0}$

- **initial state**: concentrations $[X_1],\ldots,[X_k]$
- **output**: $[Y]$
- **output-stable state**: all states reachable from it have same $[Y]$
- **rate-independent computation**: if a state **c** is reachable from the initial state **x**, a correct output-stable state **o** is reachable from **c**

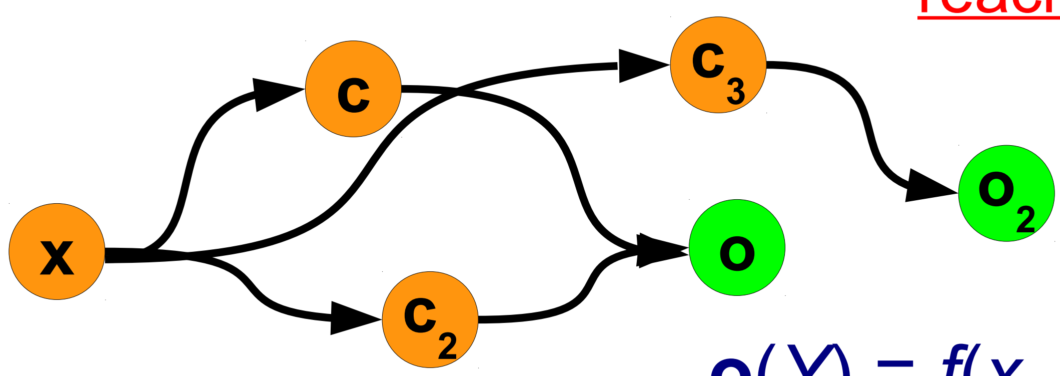


$\mathbf{o}(Y) = f(x_1,..,x_k) = \mathbf{o}_2(Y)$

Rate-independent CRN computation (formal definition)

task: compute function $y = f(x_1,\ldots,x_k)$, $\quad y,x_1,\ldots,x_k \in \mathbb{R}_{>0}$

- **initial state**: concentrations $[X_1],\ldots,[X_k]$
- **output**: $[Y]$
- **output-stable state**: all states reachable from it have same $[Y]$
- **rate-independent computation**: if a state **c** is reachable from the initial state **x**, a correct output-stable state **o** is ~~reachable~~ from **c**
  reached by mass-action kinetics



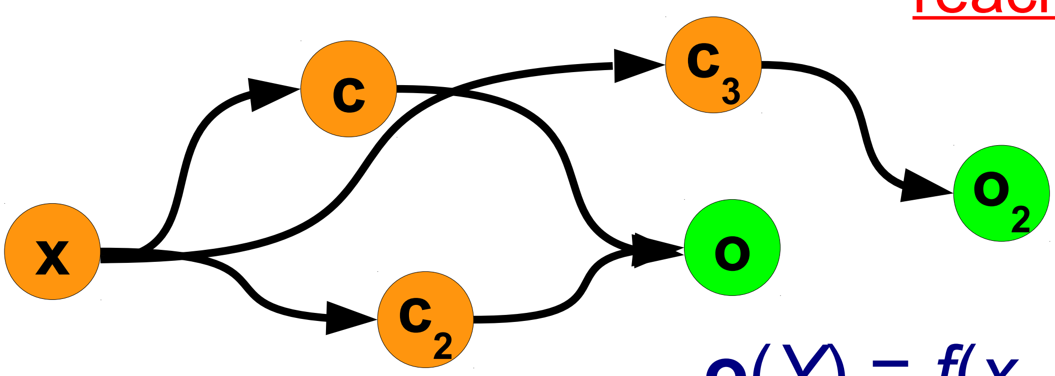
$\mathbf{o}(Y) = f(x_1,\ldots,x_k) = \mathbf{o}_2(Y)$

Rate-independent CRN computation (formal definition)

task: compute function $y = f(x_1,\ldots,x_k)$, $\quad y,x_1,\ldots,x_k \in \mathbb{R}_{>0}$

- **initial state**: concentrations $[X_1],\ldots,[X_k]$
- **output**: $[Y]$
- **output-stable state**: all states reachable from it have same $[Y]$
- **rate-independent computation**: if a state **c** is reachable from the initial state **x**, a correct output-stable state **o** is ~~reachable~~ from **c** reached by ~~mass-action~~ kinetics any reasonable*

*future work



$\mathbf{o}(Y) = f(x_1,\ldots,x_k) = \mathbf{o}_2(Y)$

Theorem: f is computable by a rate-independent CRN if and only if f is continuous and piecewise rational linear.

Chen, D, Soloveichik, *Innovations in Theoretical Computer Science*, 2014

Theorem: f is computable by a rate-independent CRN if and only if f is continuous and piecewise rational linear.

$$f(x_1,\ldots,x_k) = q_1 x_1 + \ldots + q_k x_k$$

rational coefficients

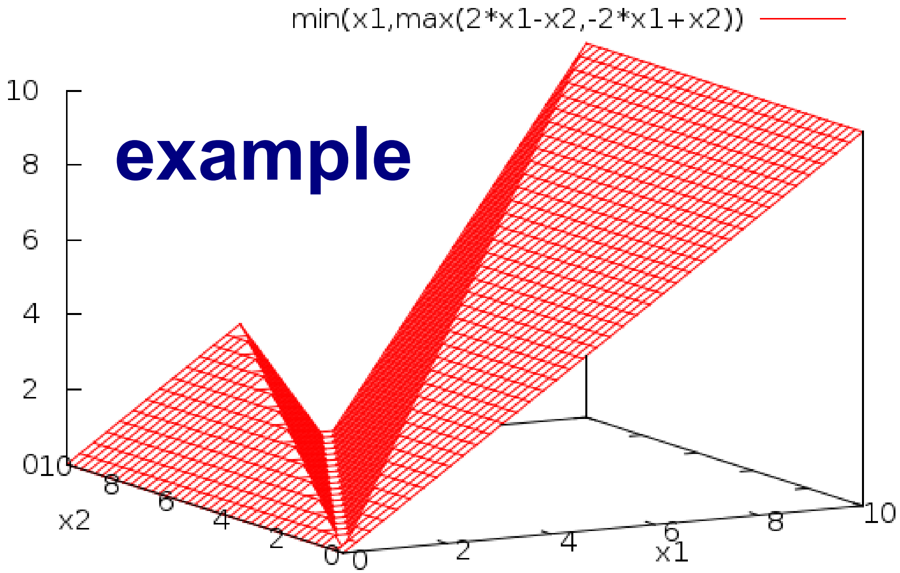
Chen, D, Soloveichik,
Innovations in Theoretical Computer Science, 2014

Theorem: f is computable by a rate-independent CRN if and only if f is continuous and piecewise rational linear.

$$f(x_1,\ldots,x_k) = q_1 x_1 + \ldots + q_k x_k$$

rational coefficients

Chen, D, Soloveichik, *Innovations in Theoretical Computer Science*, 2014



Comparison to related questions

	integer-valued (stochastic) CRN	real-valued (mass-action) CRN
rate independent		continuous piecewise linear [Chen, D, Soloveichik, *ITCS 2014*]
rate dependent		

Comparison to related questions

	integer-valued (stochastic) CRN	**real-valued (mass-action) CRN**
rate independent	semilinear [Chen, D, Soloveichik, *DNA 2012, NaCo 2013*] [Angluin, Aspnes, Eisenstat, *PODC 2006*]	continuous piecewise linear [Chen, D, Soloveichik, *ITCS 2014*]
rate dependent		

Comparison to related questions

	integer-valued (stochastic) CRN	**real-valued (mass-action) CRN**
rate independent	semilinear [Chen, D, Soloveichik, *DNA 2012, NaCo 2013*] [Angluin, Aspnes, Eisenstat, *PODC 2006*]	continuous piecewise linear [Chen, D, Soloveichik, *ITCS 2014*]
rate dependent	Turing computable (with small probability of error) [Soloveichik, Cook, Winfree, Bruck, *NaCo 2008*] [Angluin, Aspnes, Eisenstat, *DISC 2006*]	

Comparison to related questions

	integer-valued (stochastic) CRN	**real-valued (mass-action) CRN**
rate independent	semilinear [Chen, D, Soloveichik, *DNA 2012, NaCo 2013*] [Angluin, Aspnes, Eisenstat, *PODC 2006*]	continuous piecewise linear [Chen, D, Soloveichik, *ITCS 2014*]
rate dependent	Turing computable (with small probability of error) [Soloveichik, Cook, Winfree, Bruck, *NaCo 2008*] [Angluin, Aspnes, Eisenstat, *DISC 2006*]	???

Comparison to related questions

	integer-valued (stochastic) CRN	real-valued (mass-action) CRN
rate independent	semilinear [Chen, D, Soloveichik, *DNA 2012, NaCo 2013*] [Angluin, Aspnes, Eisenstat, *PODC 2006*]	continuous piecewise linear [Chen, D, Soloveichik, *ITCS 2014*]
rate dependent	Turing computable (with small probability of error) [Soloveichik, Cook, Winfree, Bruck, *NaCo 2008*] [Angluin, Aspnes, Eisenstat, *DISC 2006*]	???

population protocols
(model of distributed computing)

Every continuous piecewise rational linear function is computable

Every continuous piecewise rational linear function is computable

Theorem (Ovchinnikov 2002): Let f be a continuous piecewise linear function made of linear functions $f_1, \ldots, f_p$. Then there are subsets $S_1, \ldots, S_q$ of $\{1, \ldots, p\}$ such that $f(\mathbf{x}) = \max_{i \in \{1 \ldots q\}} \min_{j \in S_i} f_j(\mathbf{x})$

Every continuous piecewise rational linear function is computable

Theorem (Ovchinnikov 2002): Let f be a continuous piecewise linear function made of linear functions $f_1, \ldots, f_p$. Then there are subsets $S_1, \ldots, S_q$ of $\{1, \ldots, p\}$ such that $f(\mathbf{x}) = \max_{i \in \{1 \ldots q\}} \min_{j \in S_i} f_j(\mathbf{x})$

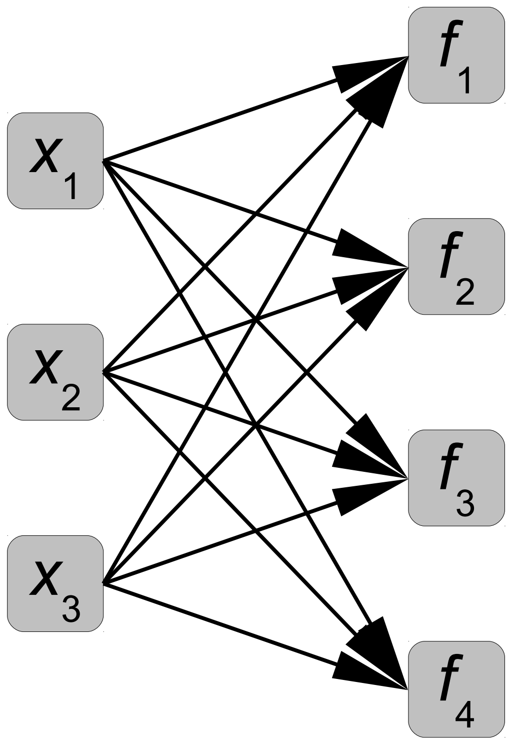
x_1

x_2

x_3

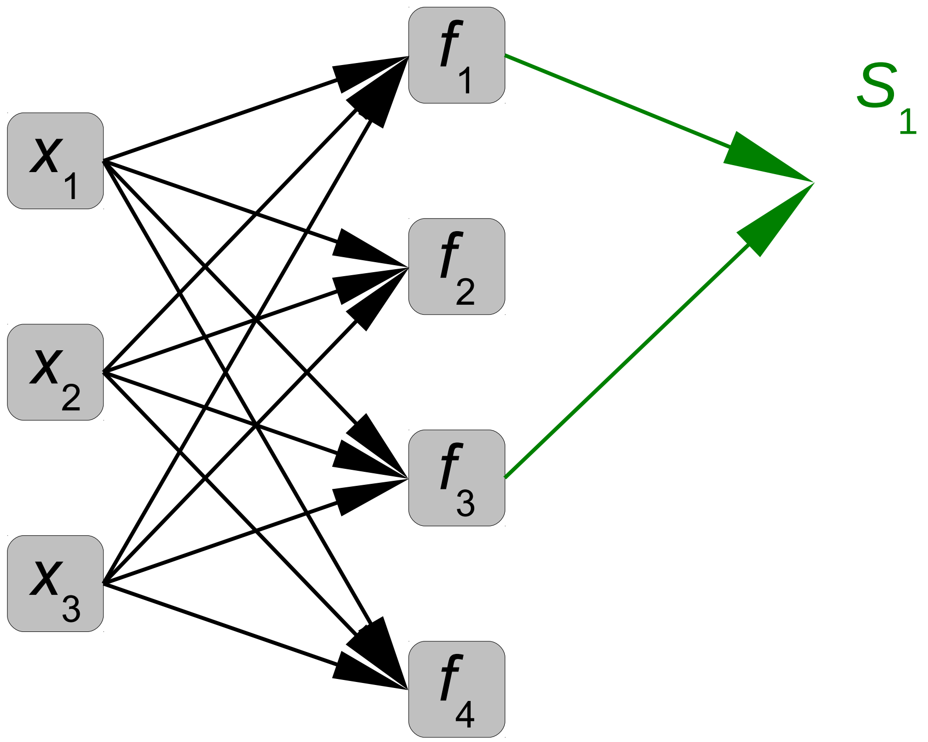
Every continuous piecewise rational linear function is computable

Theorem (Ovchinnikov 2002): Let f be a continuous piecewise linear function made of linear functions $f_1, \ldots, f_p$. Then there are subsets $S_1, \ldots, S_q$ of $\{1, \ldots, p\}$ such that $f(\mathbf{x}) = \max_{i \in \{1 \ldots q\}} \min_{j \in S_i} f_j(\mathbf{x})$



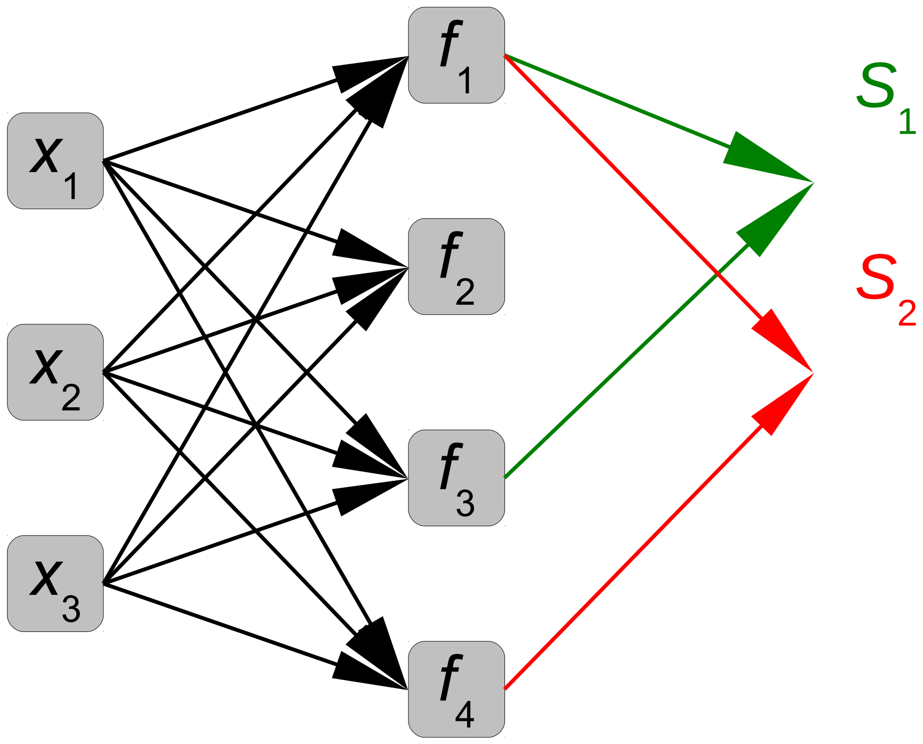
Every continuous piecewise rational linear function is computable

Theorem (Ovchinnikov 2002): Let f be a continuous piecewise linear function made of linear functions $f_1, \dots, f_p$. Then there are subsets $S_1,\dots,S_q$ of $\{1,\dots,p\}$ such that $f(\mathbf{x}) = \max_{i \in \{1 \dots q\}} \min_{j \in S_i} f_j(\mathbf{x})$



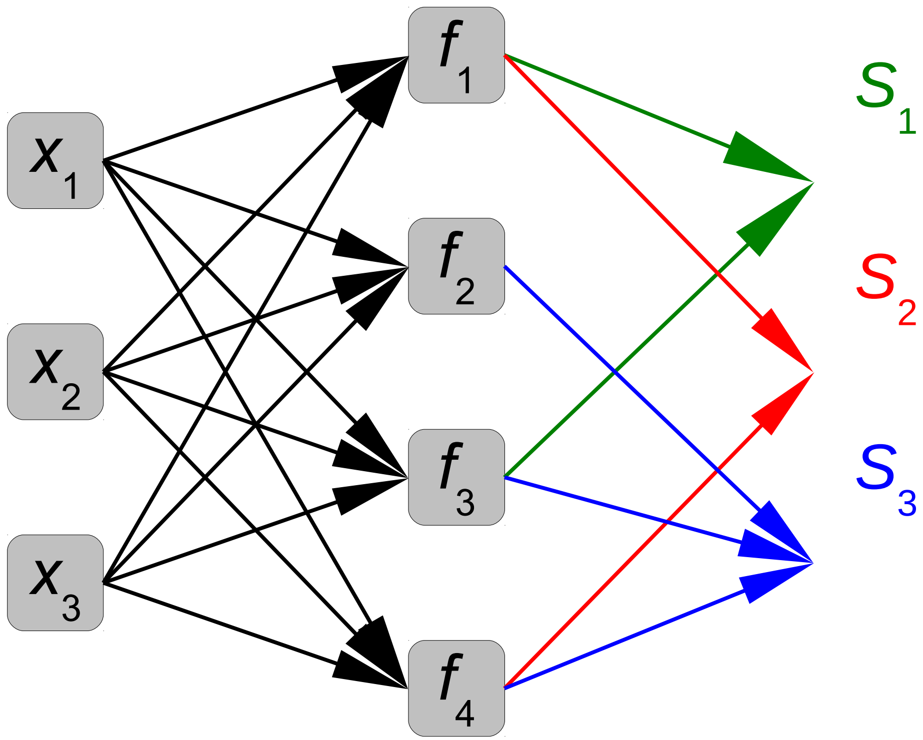
Every continuous piecewise rational linear function is computable

Theorem (Ovchinnikov 2002): Let f be a continuous piecewise linear function made of linear functions $f_1,\ldots, f_p$. Then there are subsets $S_1,\ldots,S_q$ of $\{1,\ldots,p\}$ such that $f(\mathbf{x}) = \max_{i \in \{1\ldots q\}} \min_{j \in S_i} f_j(\mathbf{x})$



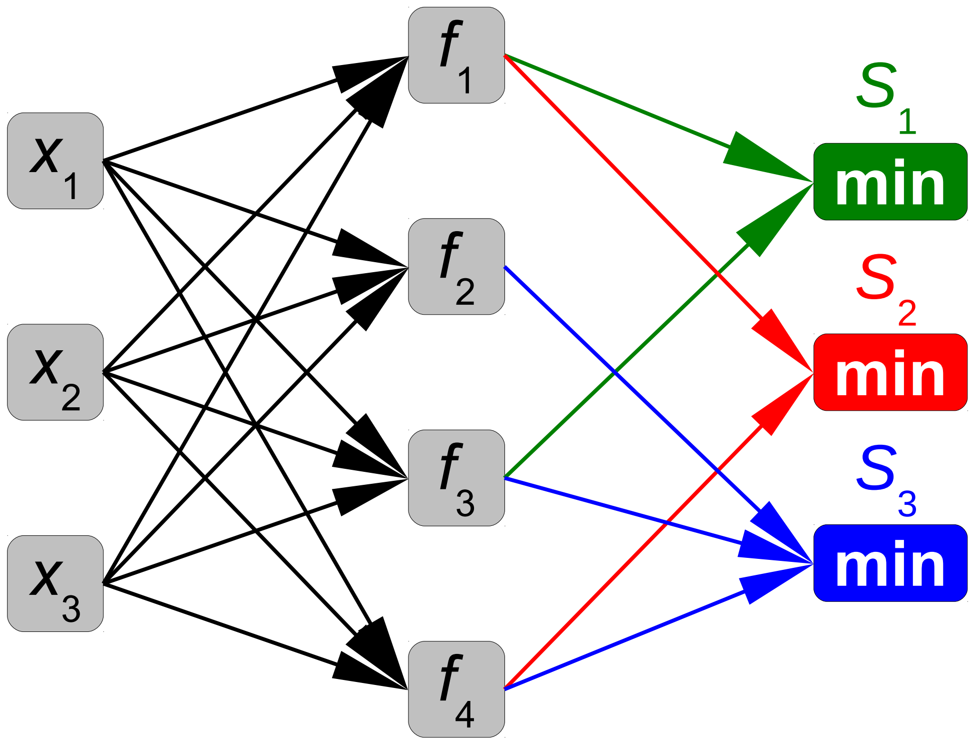
Every continuous piecewise rational linear function is computable

Theorem (Ovchinnikov 2002): Let f be a continuous piecewise linear function made of linear functions $f_1, \ldots, f_p$. Then there are subsets $S_1,\ldots,S_q$ of $\{1,\ldots,p\}$ such that $f(\mathbf{x}) = \max_{i \in \{1\ldots q\}} \min_{j \in S_i} f_j(\mathbf{x})$



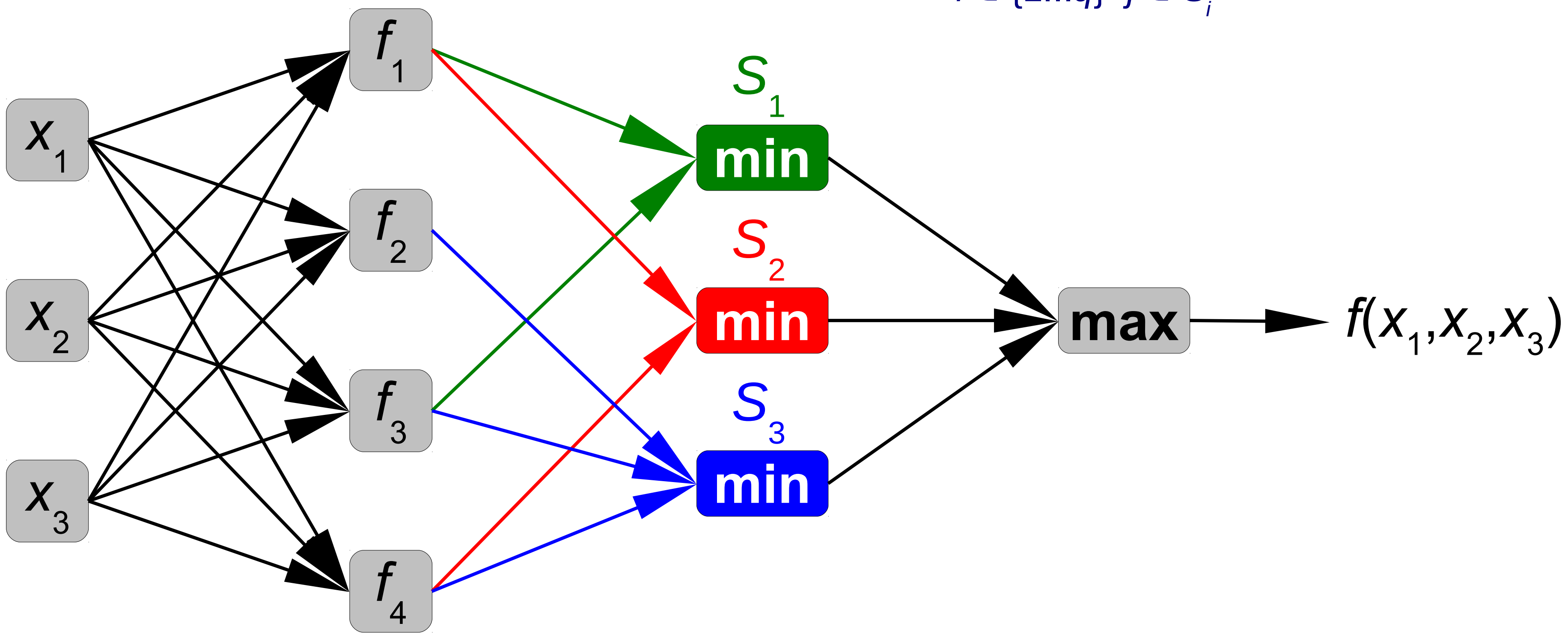
Every continuous piecewise rational linear function is computable

Theorem (Ovchinnikov 2002): Let f be a continuous piecewise linear function made of linear functions $f_1,\ldots, f_p$. Then there are subsets $S_1,\ldots,S_q$ of $\{1,\ldots,p\}$ such that $f(\mathbf{x}) = \max_{i \in \{1\ldots q\}} \min_{j \in S_i} f_j(\mathbf{x})$



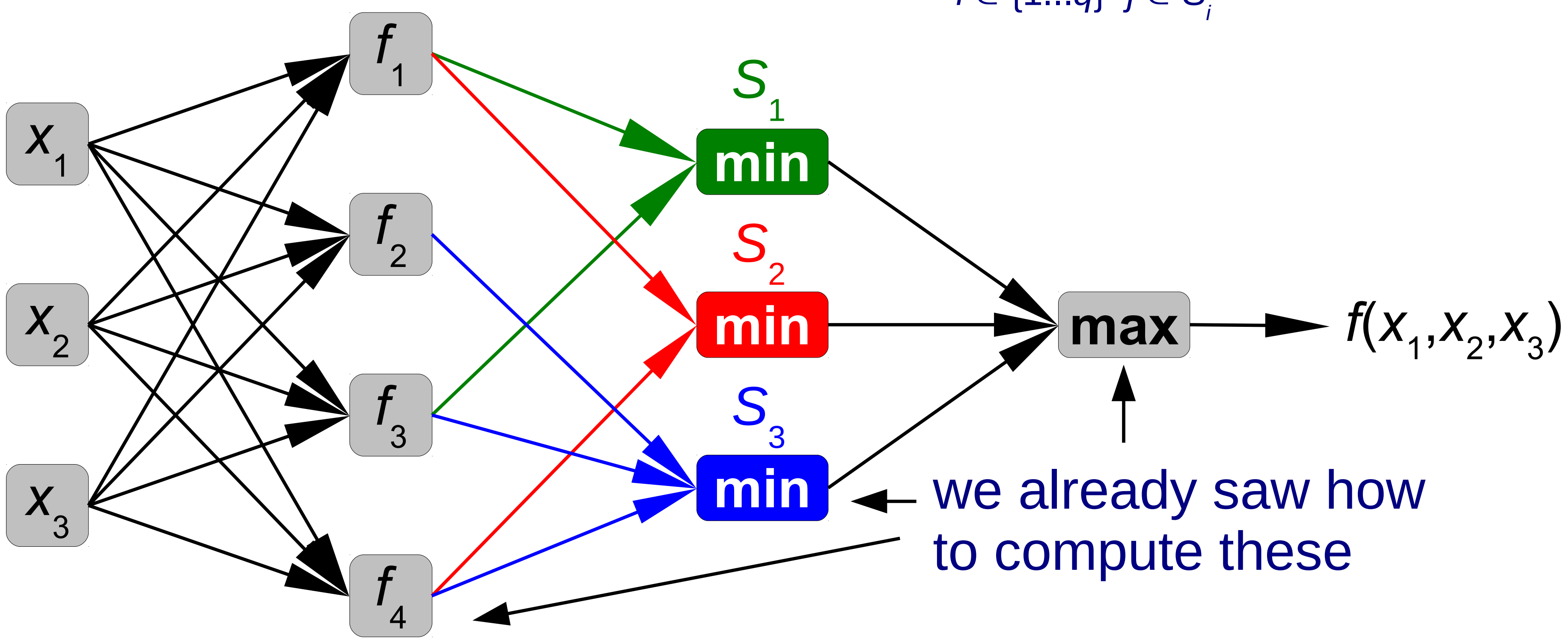
Every continuous piecewise rational linear function is computable

Theorem (Ovchinnikov 2002): Let f be a continuous piecewise linear function made of linear functions $f_1,\ldots, f_p$. Then there are subsets $S_1,\ldots,S_q$ of $\{1,\ldots,p\}$ such that $f(\mathbf{x}) = \max_{i \in \{1\ldots q\}} \min_{j \in S_i} f_j(\mathbf{x})$



Every continuous piecewise rational linear function is computable

Theorem (Ovchinnikov 2002): Let f be a continuous piecewise linear function made of linear functions $f_1,\ldots, f_p$. Then there are subsets $S_1,\ldots,S_q$ of $\{1,\ldots,p\}$ such that $f(\mathbf{x}) = \max_{i \in \{1\ldots q\}} \min_{j \in S_i} f_j(\mathbf{x})$



Every computable function is continuous and piecewise rational linear

Every computable function is continuous and piecewise rational linear

siphon: subset of species Ω whose *absence is forward-invariant*, i.e., if **s**(Ω)=**0**, and **s'** is reachable from **s**, then **s'**(Ω)=**0**.

Every computable function is continuous and piecewise rational linear

siphon: subset of species Ω whose *absence is forward-invariant*, i.e., if **s**(Ω)=**0**, and **s'** is reachable from **s**, then **s'**(Ω)=**0**.

Lemma: If [*Y*] has stabilized, it's because some siphon is absent ("drained"). We call this an **output siphon**.

Every computable function is continuous and piecewise rational linear

siphon: subset of species Ω whose *absence is forward-invariant*, i.e., if **s**(Ω)=**0**, and **s'** is reachable from **s**, then **s'**(Ω)=**0**.

Lemma: If [*Y*] has stabilized, it's because some siphon is absent ("drained"). We call this an **output siphon**.

Example: min(x_1,x_2)

$$X_1 + X_2 \rightarrow Y$$

Every computable function is continuous and piecewise rational linear

siphon: subset of species Ω whose *absence is forward-invariant*, i.e., if **s**(Ω)=**0**, and **s'** is reachable from **s**, then **s'**(Ω)=**0**.

Lemma: If [*Y*] has stabilized, it's because some siphon is absent ("drained"). We call this an **output siphon**.

Example: min(x_1,x_2)

$$x_1 > x_2 \quad \Rightarrow \quad \Omega = \{X_2\} \text{ drains}$$

$$X_1 + X_2 \rightarrow Y$$

Every computable function is continuous and piecewise rational linear

siphon: subset of species Ω whose *absence is forward-invariant*, i.e., if **s**(Ω)=**0**, and **s'** is reachable from **s**, then **s'**(Ω)=**0**.

Lemma: If [Y] has stabilized, it's because some siphon is absent ("drained"). We call this an **output siphon**.

Example: min(x_1,x_2)

$$X_1 + X_2 \to Y$$

$$x_1 > x_2 \quad \Rightarrow \quad \Omega = \{X_2\} \text{ drains}$$

$$x_1 < x_2 \quad \Rightarrow \quad \Omega = \{X_1\} \text{ drains}$$

Every computable function is continuous and piecewise rational linear

We will show:

- On inputs **x** draining Ω, $f(\mathbf{x})$ = rational linear function f_Ω of **x**, so *f* is piecewise rational linear since there are finitely many Ω.

Every computable function is continuous and piecewise rational linear

We will show:

- On inputs **x** draining Ω, $f(\mathbf{x})$ = rational linear function f_Ω of **x**, so *f* is piecewise rational linear since there are finitely many Ω.
- Set of Ω-draining inputs is *closed*, so *f* is continuous because

Every computable function is continuous and piecewise rational linear

We will show:

- On inputs **x** draining Ω, $f(\mathbf{x})$ = rational linear function f_{Ω} of **x**, so f is piecewise rational linear since there are finitely many Ω.
- Set of Ω-draining inputs is *closed*, so f is continuous because

$$\lim_{\mathbf{w}\to\mathbf{x}} f(\mathbf{w}) \qquad\qquad = f(\mathbf{x})$$

Every computable function is continuous and piecewise rational linear

We will show:

- On inputs **x** draining Ω, $f(\mathbf{x})$ = rational linear function f_Ω of **x**, so f is piecewise rational linear since there are finitely many Ω.
- Set of Ω-draining inputs is *closed*, so f is continuous because

$$\lim_{\mathbf{w}\to\mathbf{x}} f(\mathbf{w}) = \lim_{\substack{\mathbf{w}\to\mathbf{x} \\ \mathbf{w} \text{ drains } \Omega}} f_\Omega(\mathbf{w}) \qquad = f(\mathbf{x})$$

Every computable function is continuous and piecewise rational linear

We will show:

- On inputs **x** draining Ω, $f(\mathbf{x})$ = rational linear function f_Ω of **x**, so f is piecewise rational linear since there are finitely many Ω.
- Set of Ω-draining inputs is *closed*, so f is continuous because

$$\lim_{\mathbf{w}\to\mathbf{x}} f(\mathbf{w}) = \lim_{\substack{\mathbf{w}\to\mathbf{x}\\ \mathbf{w}\text{ drains }\Omega}} f_\Omega(\mathbf{w}) = f_\Omega(\mathbf{x}) = f(\mathbf{x})$$

Every computable function is continuous and piecewise rational linear

We will show:

- On inputs **x** draining Ω, $f(\mathbf{x})$ = rational linear function f_Ω of **x**, so f is piecewise rational linear since there are finitely many Ω.
- Set of Ω-draining inputs is *closed*, so f is continuous because

$$\lim_{\mathbf{w}\to\mathbf{x}} f(\mathbf{w}) = \lim_{\substack{\mathbf{w}\to\mathbf{x} \\ \mathbf{w}\text{ drains }\Omega}} f_\Omega(\mathbf{w}) = \boxed{f_\Omega(\mathbf{x}) = f(\mathbf{x})}$$

Every computable function is continuous and piecewise rational linear

We will show:

- On inputs $\mathbf{x}$ draining Ω, $f(\mathbf{x})$ = rational linear function f_Ω of $\mathbf{x}$, so f is piecewise rational linear since there are finitely many Ω.
- Set of Ω-draining inputs is *closed*, so f is continuous because

$$\lim_{\mathbf{w}\to\mathbf{x}} f(\mathbf{w}) = \lim_{\substack{\mathbf{w}\to\mathbf{x} \\ \mathbf{w} \text{ drains } \Omega}} f_\Omega(\mathbf{w}) = f_\Omega(\mathbf{x}) = f(\mathbf{x})$$

Intuition:

"<u>state $\mathbf{x}$ can reach state $\mathbf{o}$, where $\mathbf{o}(\Omega) = 0$ and $\mathbf{o}(Y) = y$</u>" $\Leftrightarrow$ $\mathbf{Az} \geq \mathbf{0}$, where $\mathbf{z}$ contains $\mathbf{x}$ and y as unknowns, $\mathbf{A}$ constant (and rational).

Every computable function is continuous and piecewise rational linear

We will show:

- On inputs **x** draining Ω, $f(\mathbf{x})$ = rational linear function f_Ω of **x**, so *f* is piecewise rational linear since there are finitely many Ω.
- Set of Ω-draining inputs is *closed*, so *f* is continuous because

$$\lim_{\mathbf{w}\to\mathbf{x}} f(\mathbf{w}) = \lim_{\substack{\mathbf{w}\to\mathbf{x} \\ \mathbf{w} \text{ drains } \Omega}} f_\Omega(\mathbf{w}) = f_\Omega(\mathbf{x}) = f(\mathbf{x})$$

Intuition:

"<u>state **x** can reach state **o**, where **o**(Ω) = 0 and **o**(*Y*) = *y*</u>" ⇔ $\mathbf{A}\mathbf{z} \geq \mathbf{0}$, where **z** contains **x** and *y* as unknowns, **A** constant (and rational).

The set of **z** satisfying $\mathbf{A}\mathbf{z} \geq \mathbf{0}$ is closed and convex.

Every computable function is continuous and piecewise rational linear

We will show:

- On inputs **x** draining Ω, *f*(**x**) = rational linear function f_Ω of **x**, so *f* is piecewise rational linear since there are finitely many Ω.
- Set of Ω-draining inputs is *closed*, so *f* is continuous because

$$\lim_{\mathbf{w}\to\mathbf{x}} f(\mathbf{w}) = \lim_{\substack{\mathbf{w}\to\mathbf{x} \\ \mathbf{w}\text{ drains }\Omega}} f_\Omega(\mathbf{w}) = f_\Omega(\mathbf{x}) = f(\mathbf{x})$$

Intuition:

"state **x** can reach state **o**, where **o**(Ω) = 0 and **o**(*Y*) = *y*" ⇔ $\mathbf{Az} \geq \mathbf{0}$, where **z** contains **x** and *y* as unknowns, **A** constant (and rational).

The set of **z** satisfying $\mathbf{Az} \geq \mathbf{0}$ is closed and convex.

Every computable function is continuous and piecewise rational linear

We will show:

- On inputs $\mathbf{x}$ draining Ω, $f(\mathbf{x})$ = rational linear function f_Ω of $\mathbf{x}$, so f is piecewise rational linear since there are finitely many Ω.
- Set of Ω-draining inputs is *closed*, so f is continuous because

$$\lim_{\mathbf{w}\to\mathbf{x}} f(\mathbf{w}) = \lim_{\substack{\mathbf{w}\to\mathbf{x} \\ \mathbf{w} \text{ drains } \Omega}} f_\Omega(\mathbf{w}) = f_\Omega(\mathbf{x}) = f(\mathbf{x})$$

Intuition:

"state $\mathbf{x}$ can reach state $\mathbf{o}$, where $\mathbf{o}(\Omega) = 0$ and $\mathbf{o}(Y) = y$" $\Leftrightarrow$ $\mathbf{Az} \geq \mathbf{0}$, where $\mathbf{z}$ contains $\mathbf{x}$ and y as unknowns, $\mathbf{A}$ constant (and rational).

The set of $\mathbf{z}$ satisfying $\mathbf{Az} \geq \mathbf{0}$ is closed and convex.

Every computable function is continuous and piecewise rational linear

We will show:

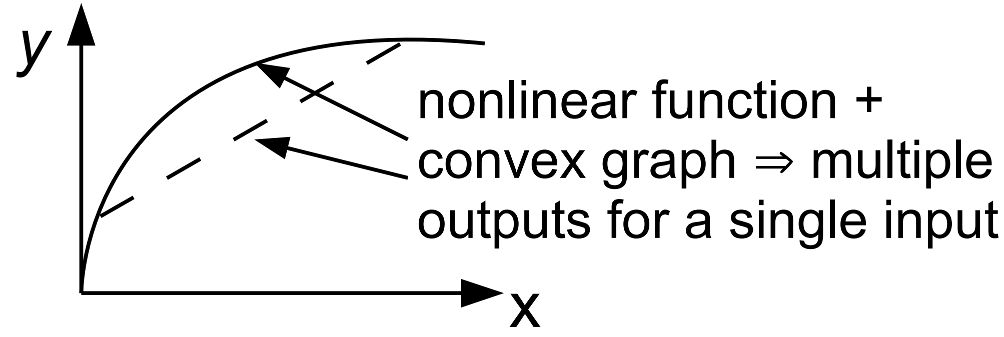
- On inputs $\mathbf{x}$ draining Ω, $f(\mathbf{x})$ = rational linear function f_Ω of $\mathbf{x}$, so f is piecewise rational linear since there are finitely many Ω.
- Set of Ω-draining inputs is *closed*, so f is continuous because

$$\lim_{\mathbf{w}\to\mathbf{x}} f(\mathbf{w}) = \lim_{\substack{\mathbf{w}\to\mathbf{x} \\ \mathbf{w}\text{ drains }\Omega}} f_\Omega(\mathbf{w}) = f_\Omega(\mathbf{x}) = f(\mathbf{x})$$

Intuition:

"<u>state **x** can reach state **o**, where **o**(Ω) = 0 and **o**(*Y*) = *y*</u>" ⇔ $\mathbf{Az} \geq \mathbf{0}$, where **z** contains **x** and *y* as unknowns, **A** constant (and rational).

The set of **z** satisfying $\mathbf{Az} \geq \mathbf{0}$ is closed and convex.



Acknowledgments

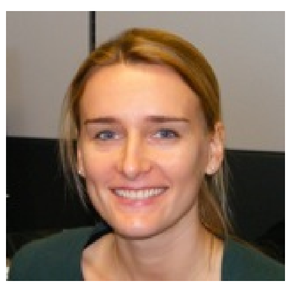
Ho-Lin Chen

David Soloveichik

UCSF
center for systems & synthetic biology
an NIGMS national systems biology center

American Institute of Mathematics

Damien Woods



Elisa Franco

Anne Shiu

German Enciso

Bernd Sturmfels

Manoj Gopalkrishnan

Thank you!