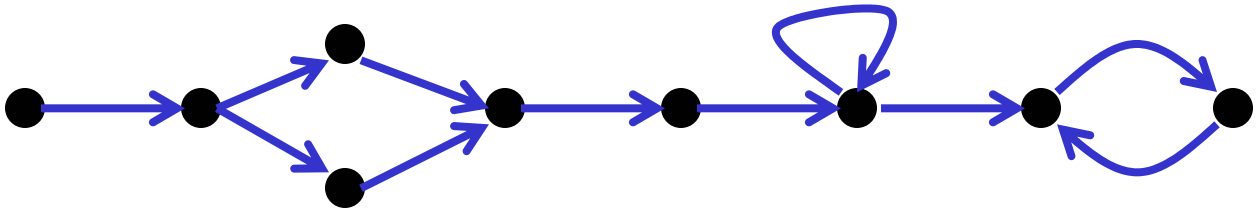
The Symbolic Approach to Hybrid Systems

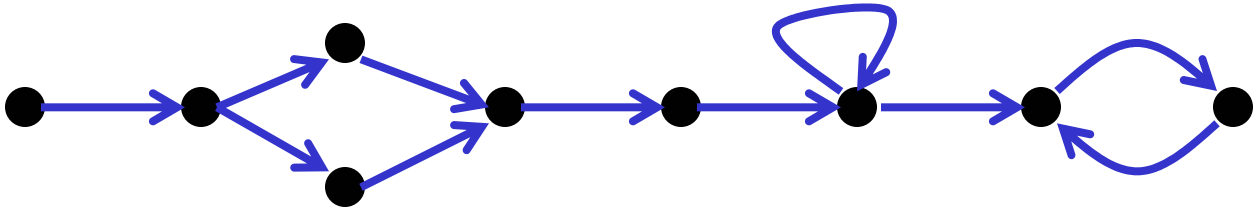
Tom Henzinger

University of California, Berkeley

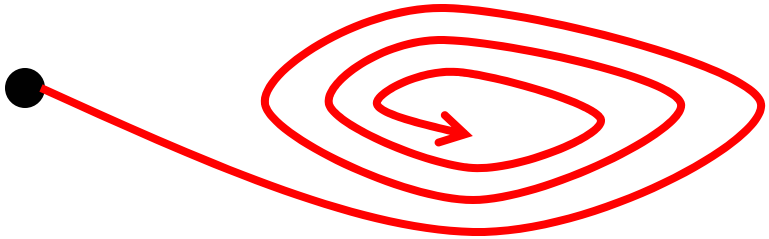
Discrete (transition) system



Discrete (transition) system

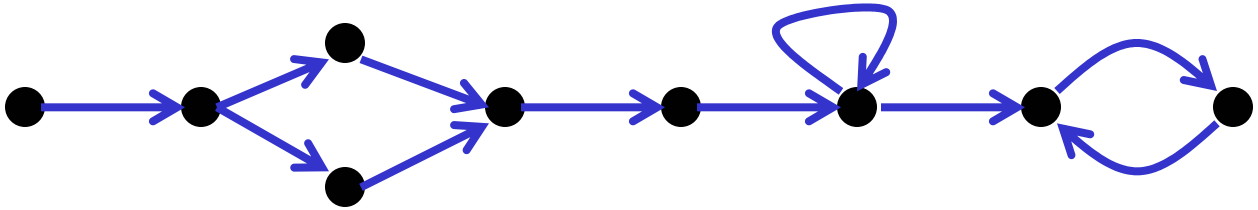


Continuous (dynamical) system

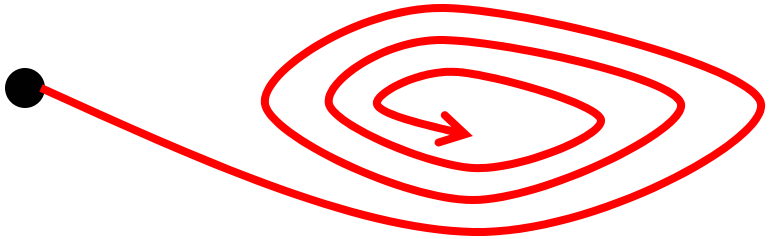


$Q = R^n$

Discrete (transition) system

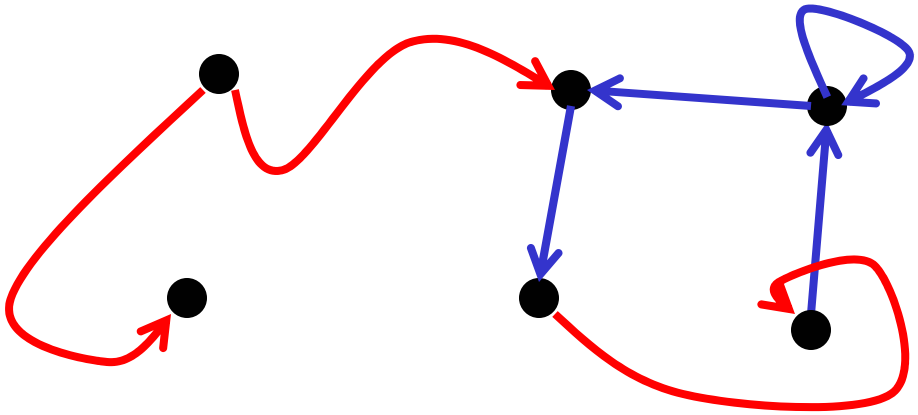


Continuous (dynamical) system



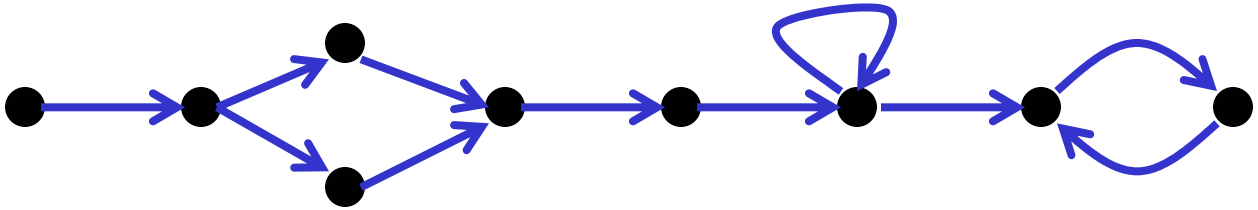
$Q = R^n$

Hybrid system

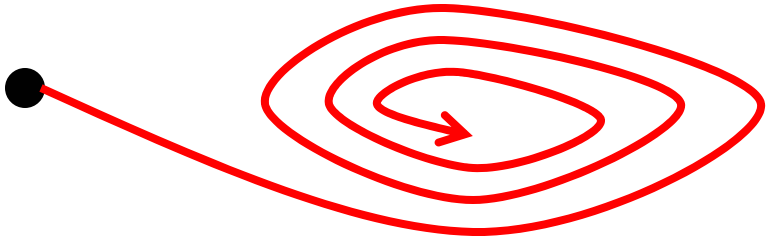


jumps
flows

Discrete (transition) system

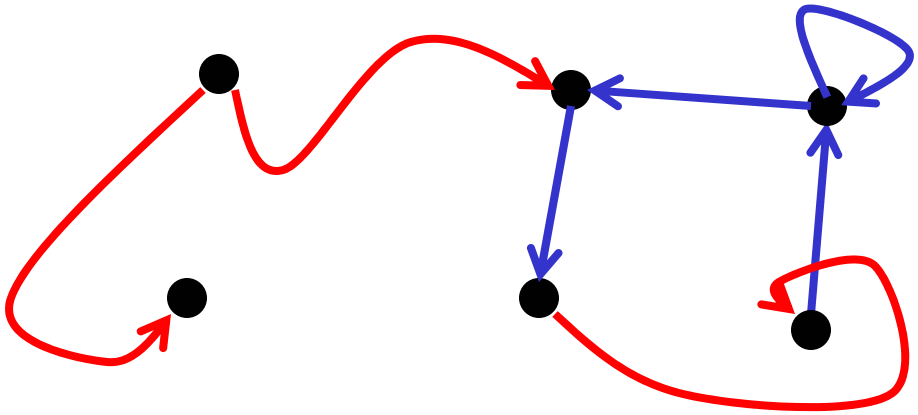


Continuous (dynamical) system



$Q = R^n$

Hybrid system



jumps
flows

-nondeterministic
-time abstract

A Thermostat

States

$x \in R$	temperature
$h \in \{ \text{on}, \text{off} \}$	heat

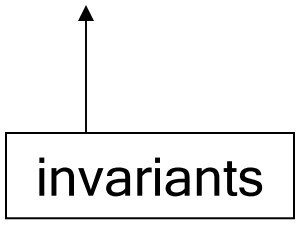
A Thermostat

States

$x \in R$	temperature
$h \in \{ on, off \}$	heat

Flows

f_1	Y h = on	$\rightarrow$	$x' = K \cdot (H-x)$
f_2	Y h = off	$\rightarrow$	$x' = -K \cdot x$



A Thermostat

States

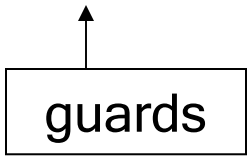
$x \in R$	temperature
$h \in \{ on, off \}$	heat

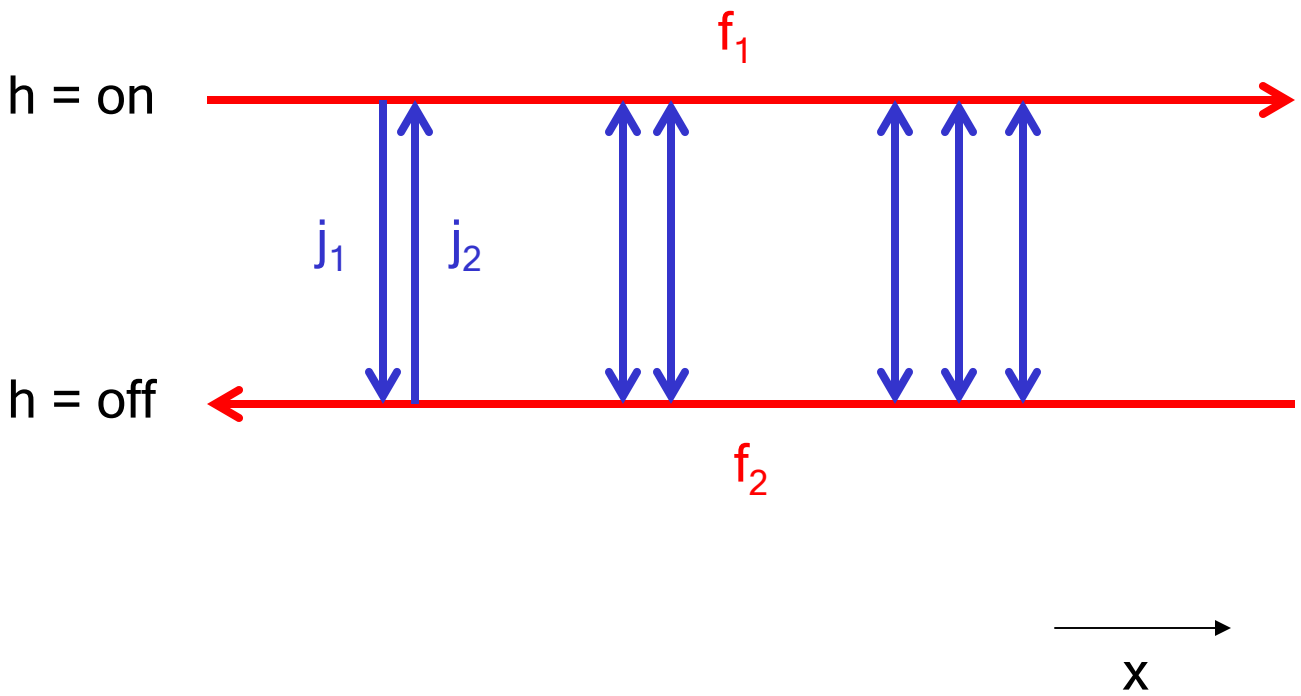
Flows

f_1	Y h = on	$\rightarrow$	$x' = K \cdot (H-x)$
f_2	Y h = off	$\rightarrow$	$x' = -K \cdot x$

Jumps

j_1	Y h = on	$\rightarrow$	h := off
j_2	Y h = off	$\rightarrow$	h := on




f1
h = on
j1
j2
h = off
f2
x

A Thermostat

States

	$x \in R$	temperature
	$h \in \{ on, off \}$	heat
	$t \in R$	timer

Flows

f_1	Y h = on	$\rightarrow$	x' = K·(H-x)
f_2	Y h = off	$\rightarrow$	x' = -K·x

Jumps

j_1	Y h = on	$\rightarrow$	h := off
j_2	Y h = off	$\rightarrow$	h := on

A Thermostat

States

	$x \in R$	temperature
	$h \in \{ on, off \}$	heat
	$t \in R$	timer

Flows

f_1	Y $h = on$	$\rightarrow$	$x' = K \cdot (H-x);\ t' = 1$
f_2	Y $h = off$	$\rightarrow$	$x' = -K \cdot x;\ t' = 1$

Jumps

j_1	Y $h = on \wedge t \geq L$	$\rightarrow$	$h := off;\ t := 0$
j_2	Y $h = off \wedge t \geq L$	$\rightarrow$	$h := on;\ t := 0$

A Thermostat

States

$x \in R$	temperature
$h \in \{ on, off \}$	heat
$t \in R$	timer

Flows

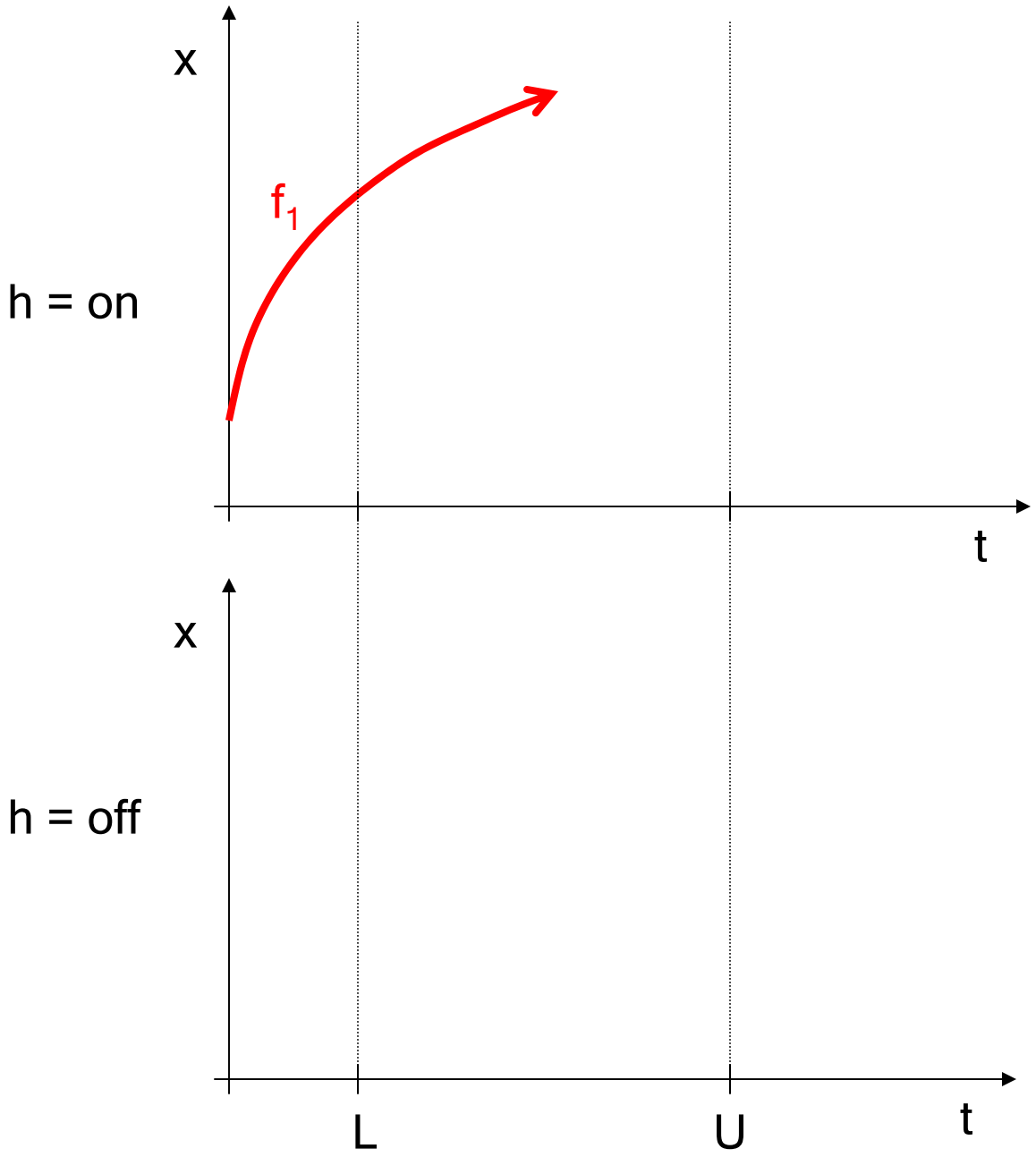
f_1 $\quad Y \;\; h = on \;\wedge\; t \leq U \;\rightarrow\; x' = K\cdot(H-x);\;\; t' = 1$

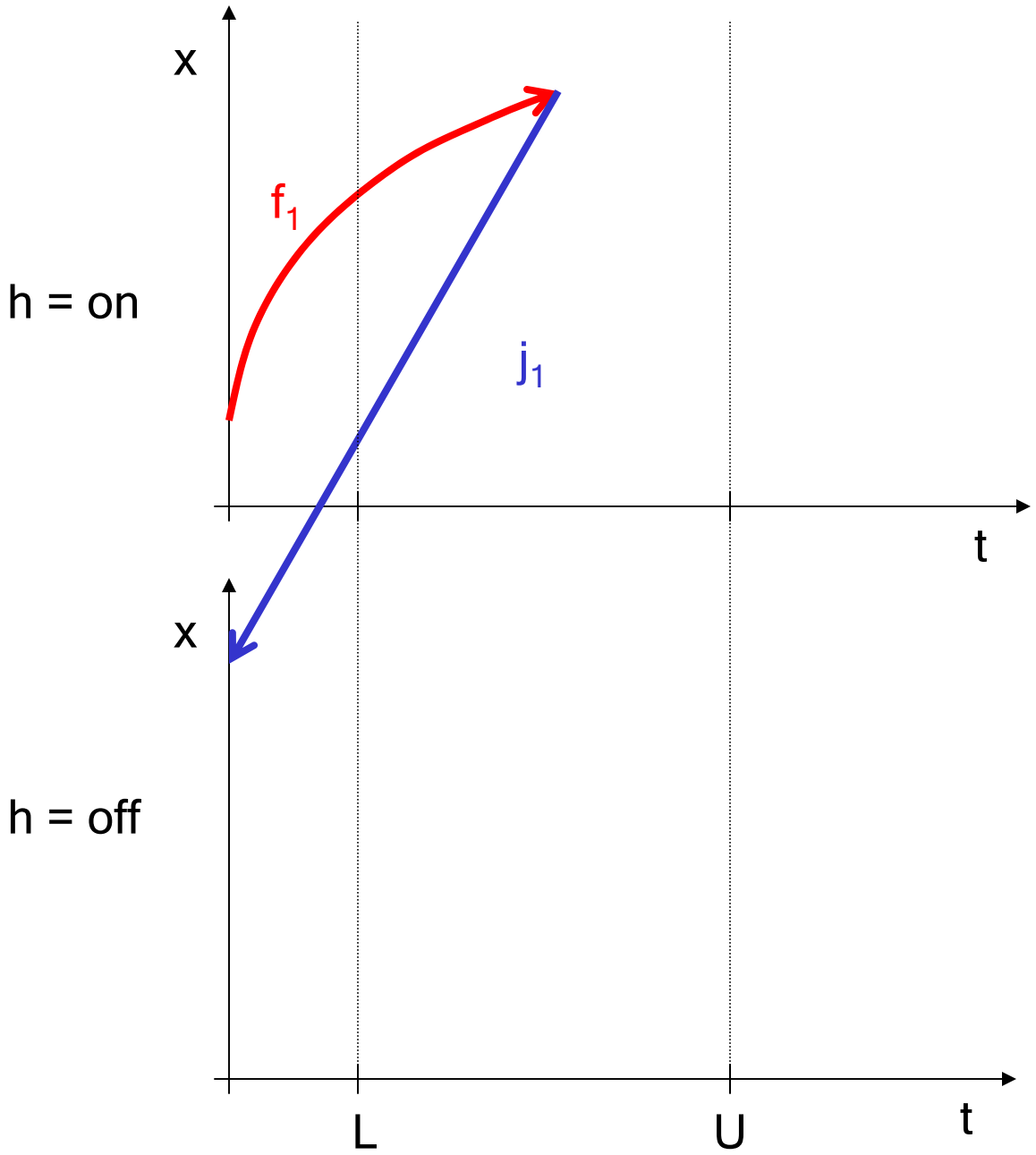
f_2 $\quad Y \;\; h = off \;\wedge\; t \leq U \;\rightarrow\; x' = -K\cdot x;\;\; t' = 1$

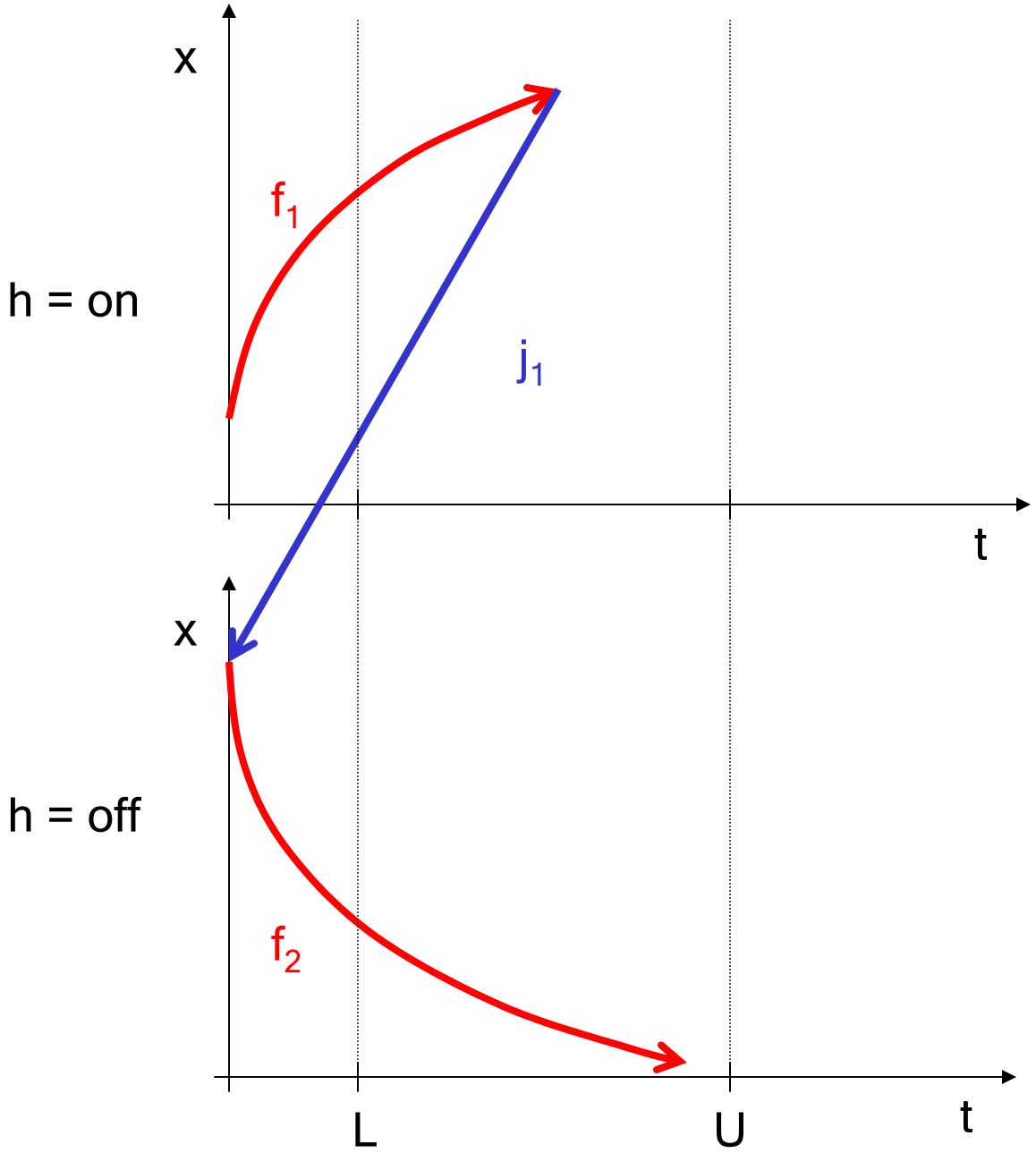
Jumps

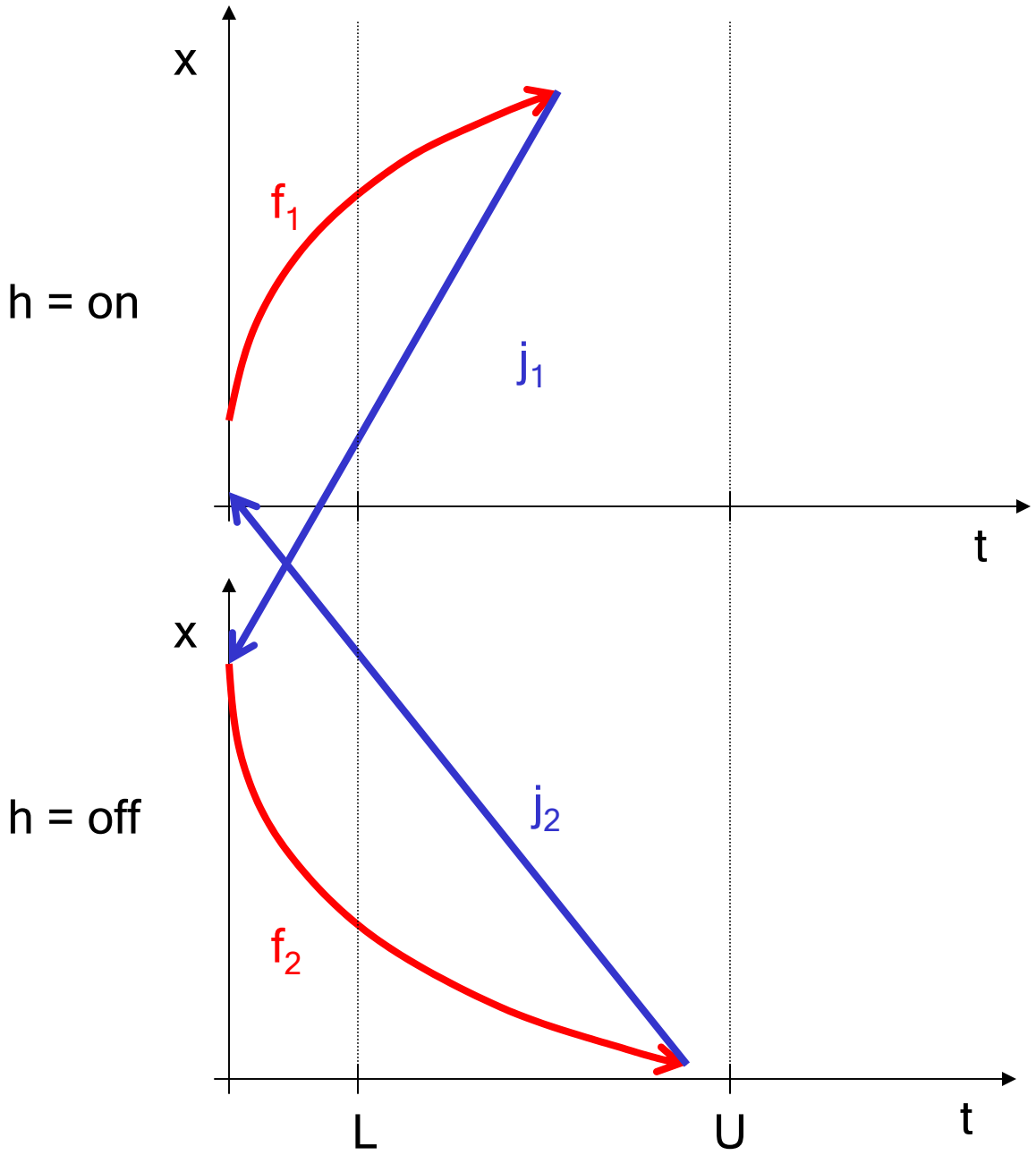
j_1 $\quad Y \;\; h = on \;\wedge\; t \geq L \;\rightarrow\; h := off;\;\; t := 0$

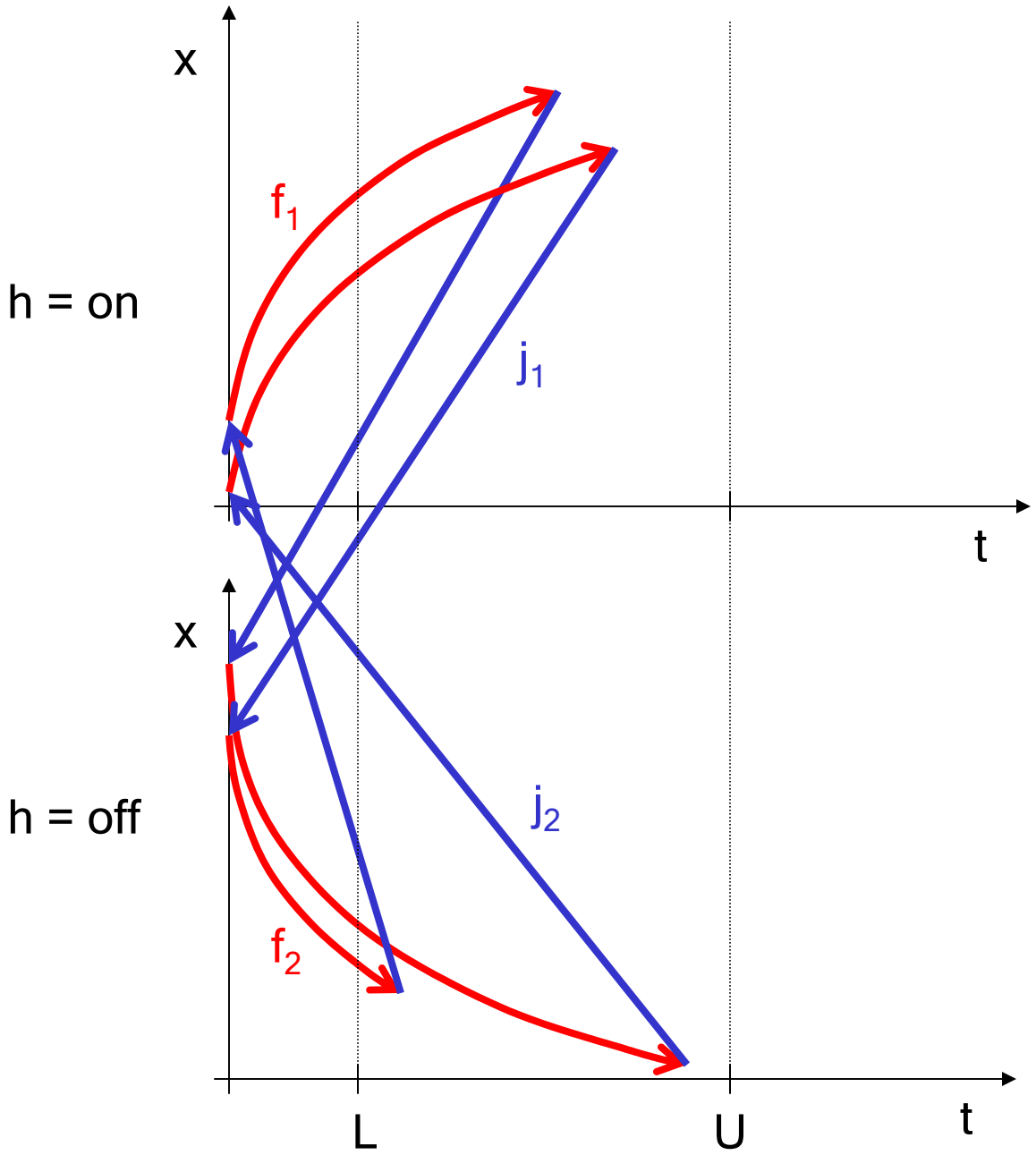
j_2 $\quad Y \;\; h = off \;\wedge\; t \geq L \;\rightarrow\; h := on;\;\; t := 0$

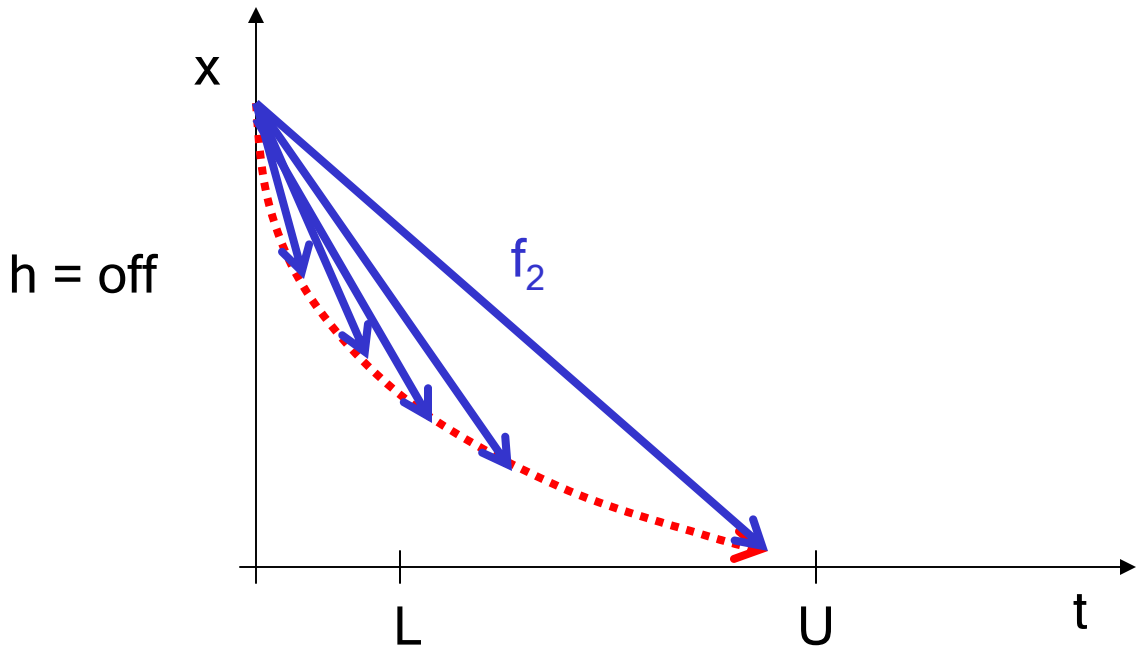

x
f_1
h = on
t
x
h = off
L
U
t


x
f_1
h = on
j_1
t
x
h = off
L
U
t


x
h = on
f_1
j_1
t
x
h = off
f_2
L
U
t


x
h = on
f_1
j_1
t
x
h = off
j_2
f_2
L
U
t


x
h = on
f1
j1
t
x
h = off
j2
f2
L
U
t



Step 1: Discretize

Transition System

Q	set of states
Σ	set of actions
post: $Q \times \Sigma \rightarrow 2^Q$	successor function

Transition System

Q	set of states
Σ	set of actions
post: $Q \times \Sigma \rightarrow 2^Q$	successor function

Thermostat

$Q = R^2 \times \{ \text{on, off} \}$

$\Sigma = \{ f_1, f_2, j_1, j_2 \}$

$$\text{post}(x, t, \text{on}, j_1) = \begin{cases} \{ (x, 0, \text{off}) \} & \text{if } t \geq L \\ \varnothing & \text{if } t < L \end{cases}$$

Transition System

Q	set of states
Σ	set of actions
post: $Q \times \Sigma \rightarrow 2^Q$	successor function

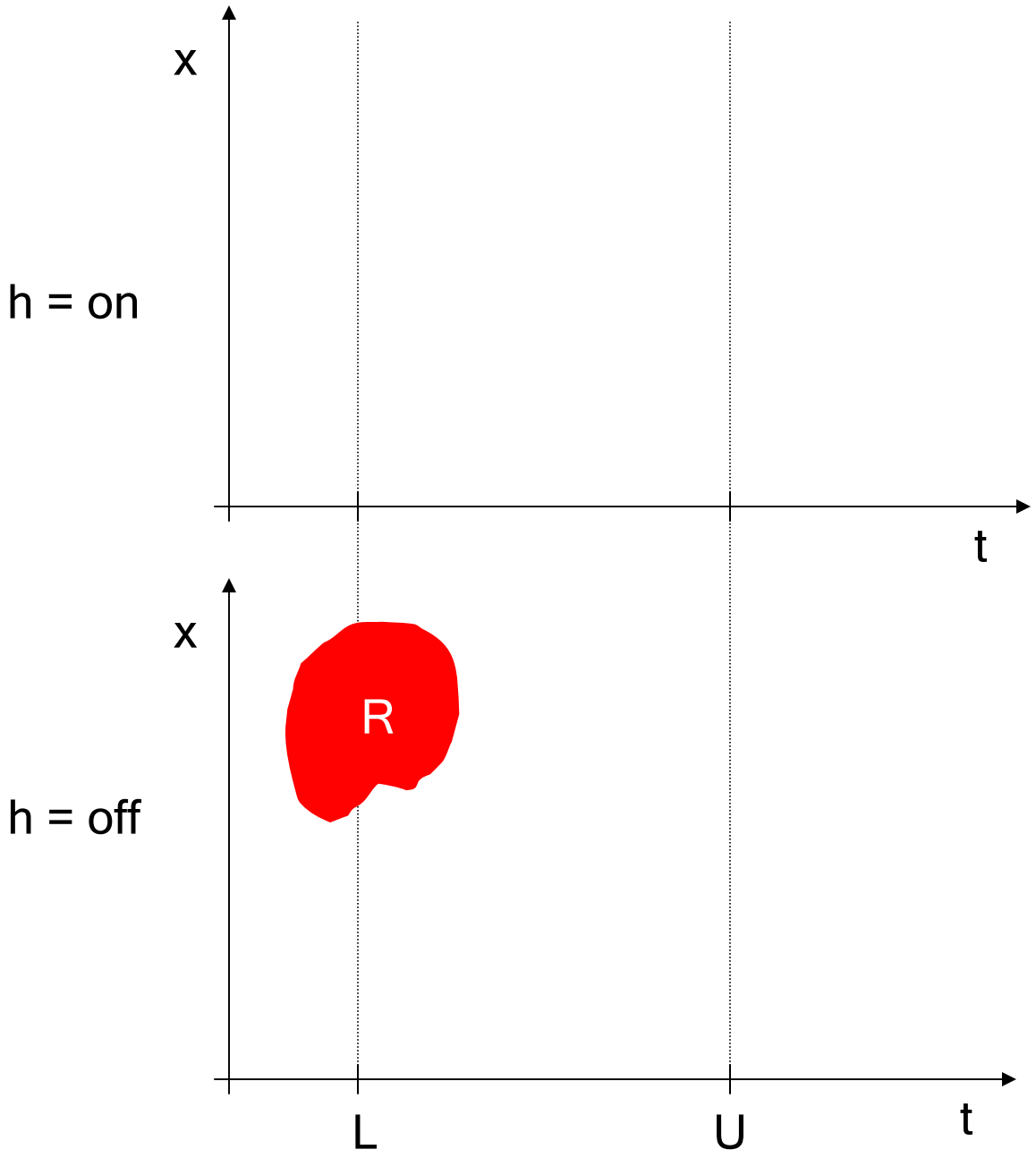
Thermostat

$Q = R^2 \times \{ \text{on, off} \}$

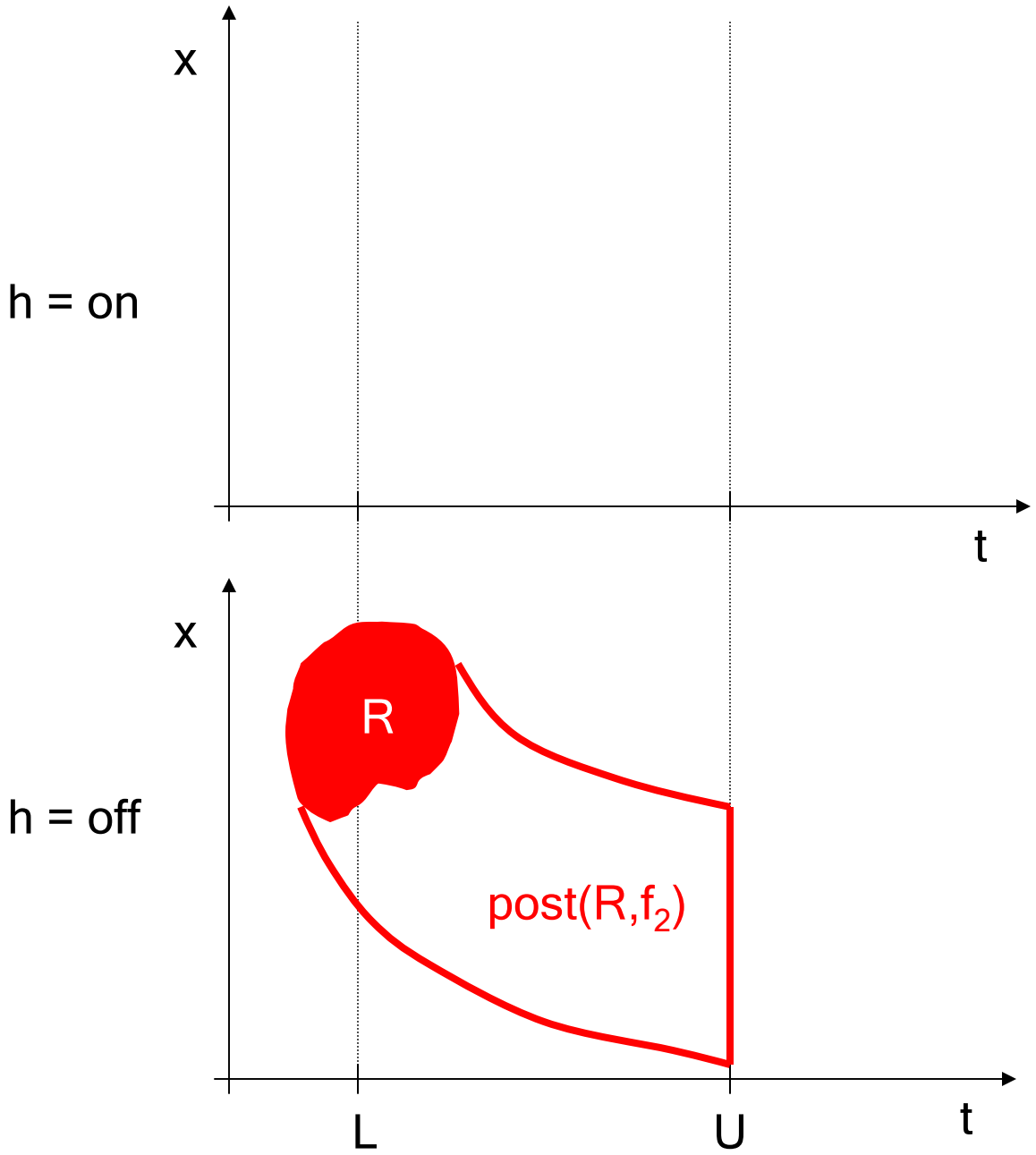
$\Sigma = \{ f_1, f_2, j_1, j_2 \}$

$$\text{post}(x, t, \text{on}, j_1) = \begin{cases} \{ (x, 0, \text{off}) \} & \text{if } t \geq L \\ \varnothing & \text{if } t < L \end{cases}$$

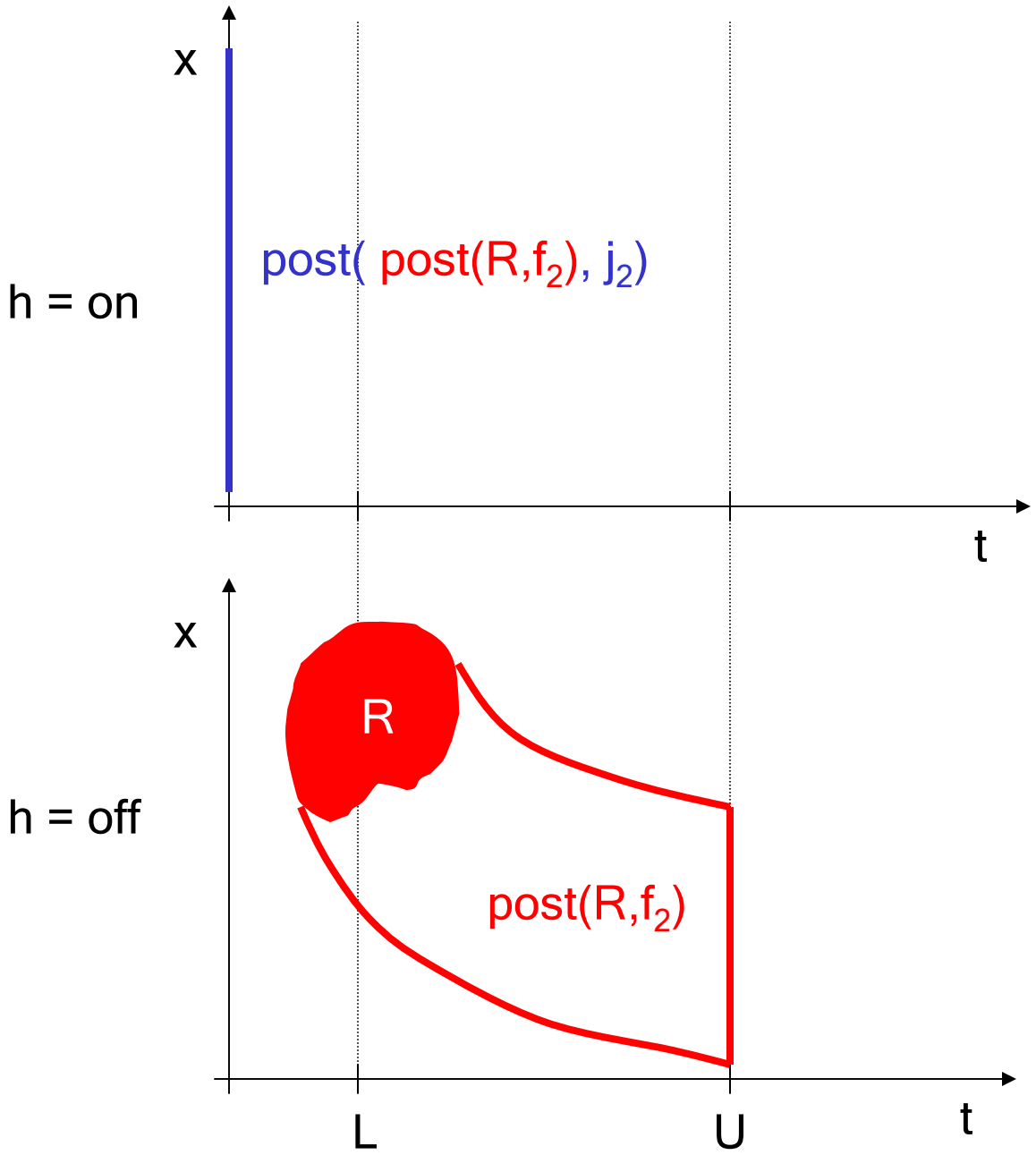
$$\text{post}(x, t, \text{on}, f_1) = \begin{cases} \text{infinite set} & \text{if } t < U \\ \{ (x, 0, \text{on}) \} & \text{if } t = U \\ \varnothing & \text{if } t > U \end{cases}$$



Step 2: Lift



Step 2: Lift



Step 2: Lift

Lifted Transition System

Q

Σ

post: $2^Q \times \Sigma \rightarrow 2^Q$ $\quad\quad$ $post(R,\sigma) = \bigcup_{q \in Q} post(q,\sigma)$

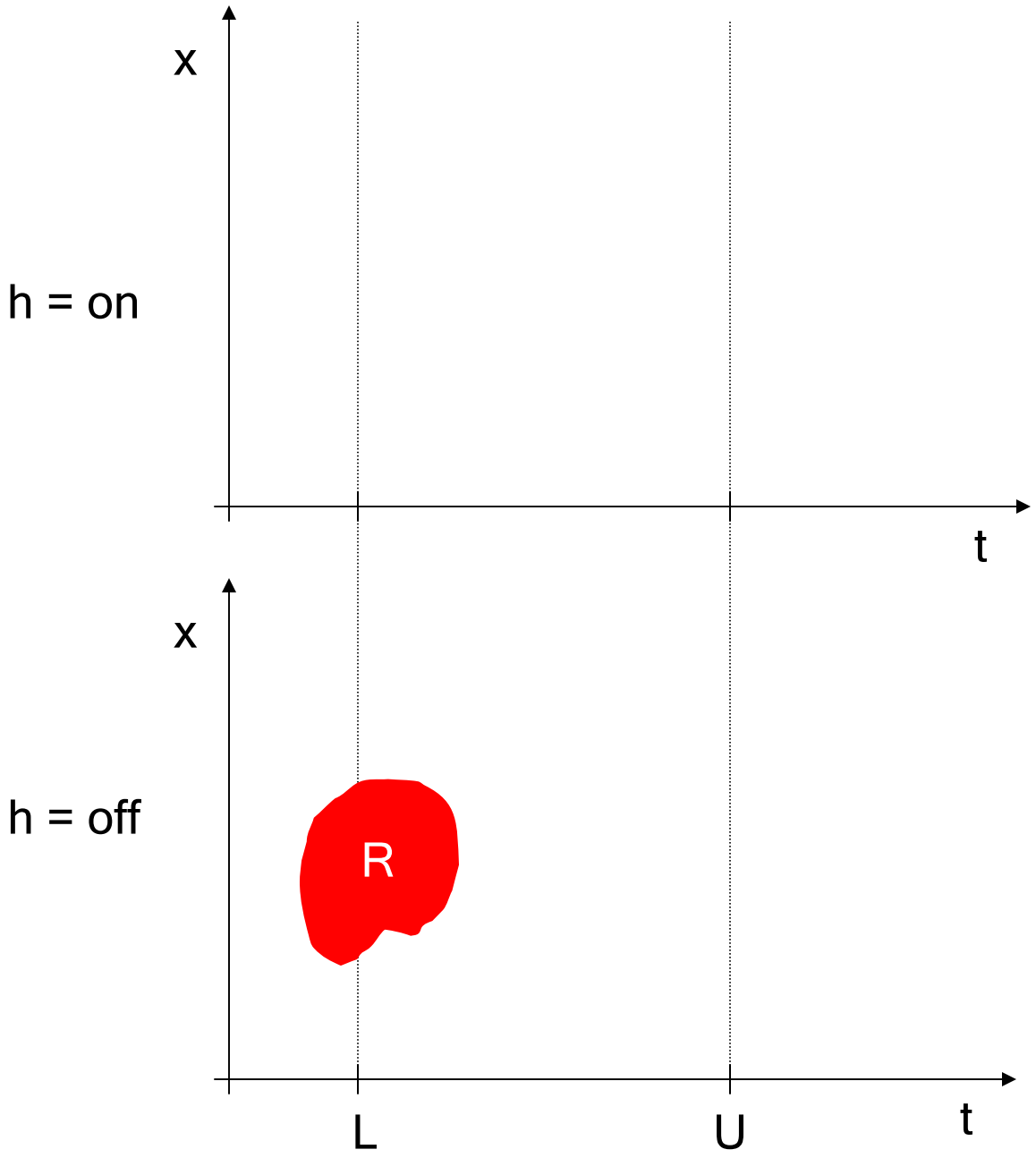
Lifted Transition System

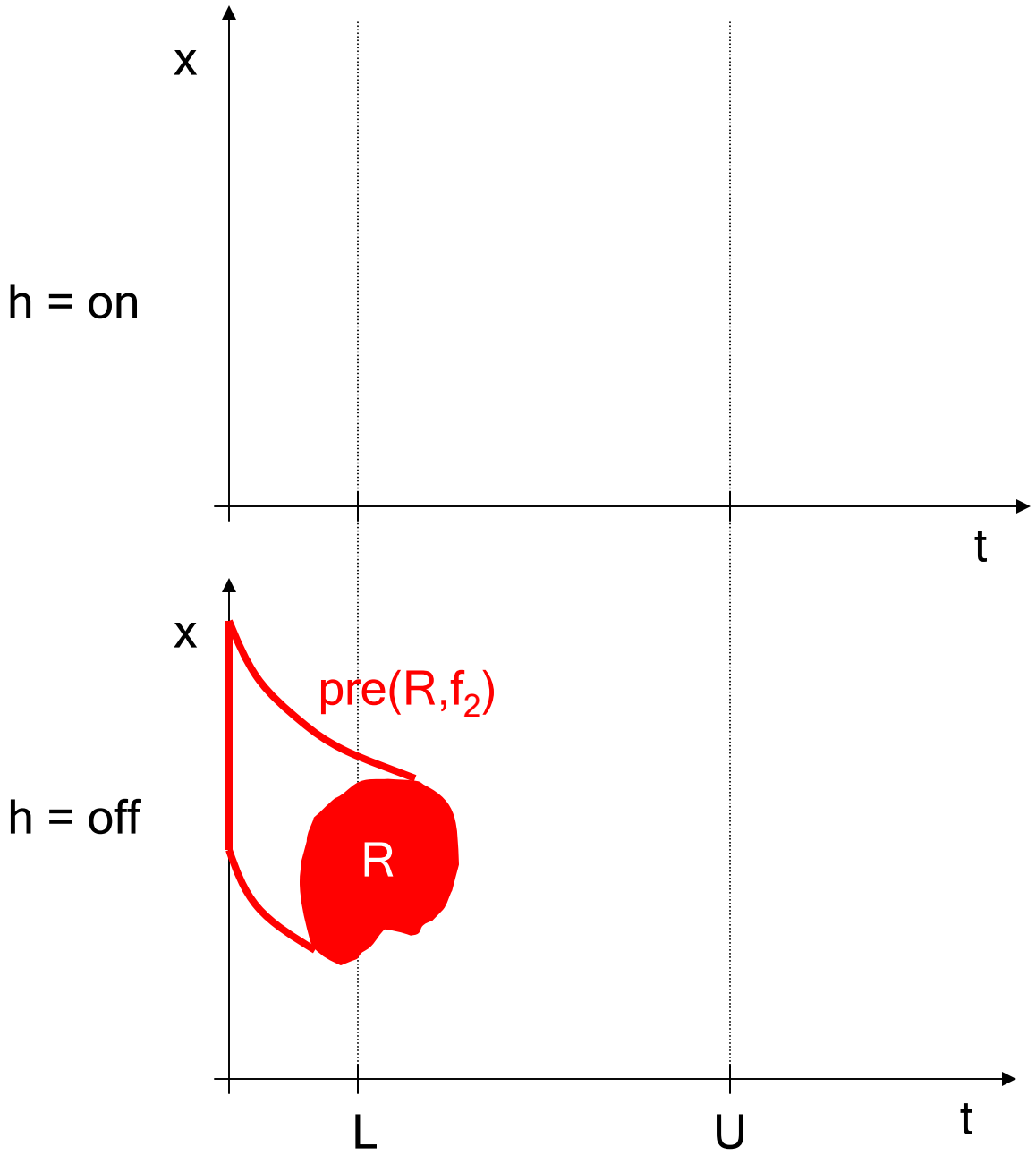
Q

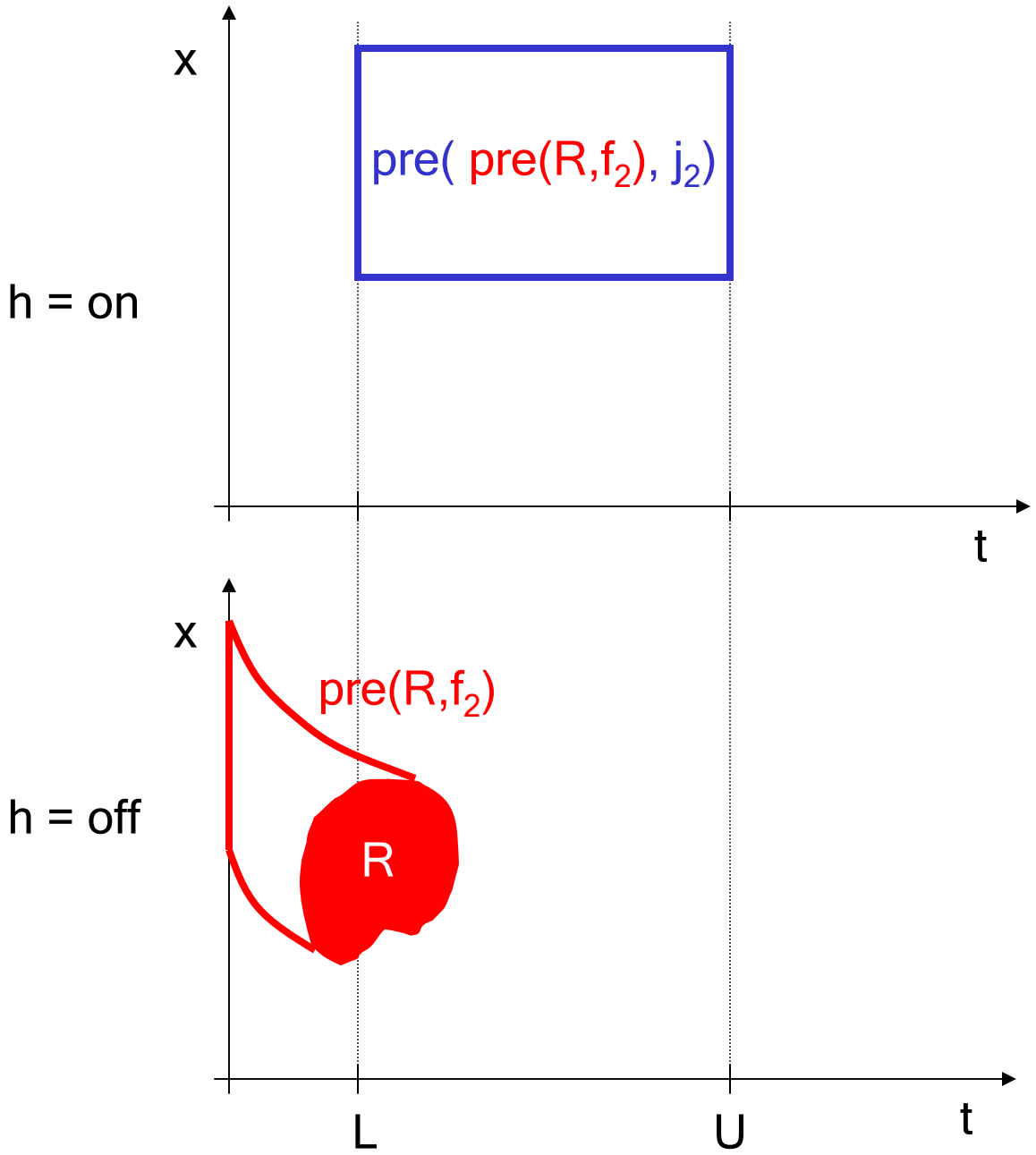
Σ

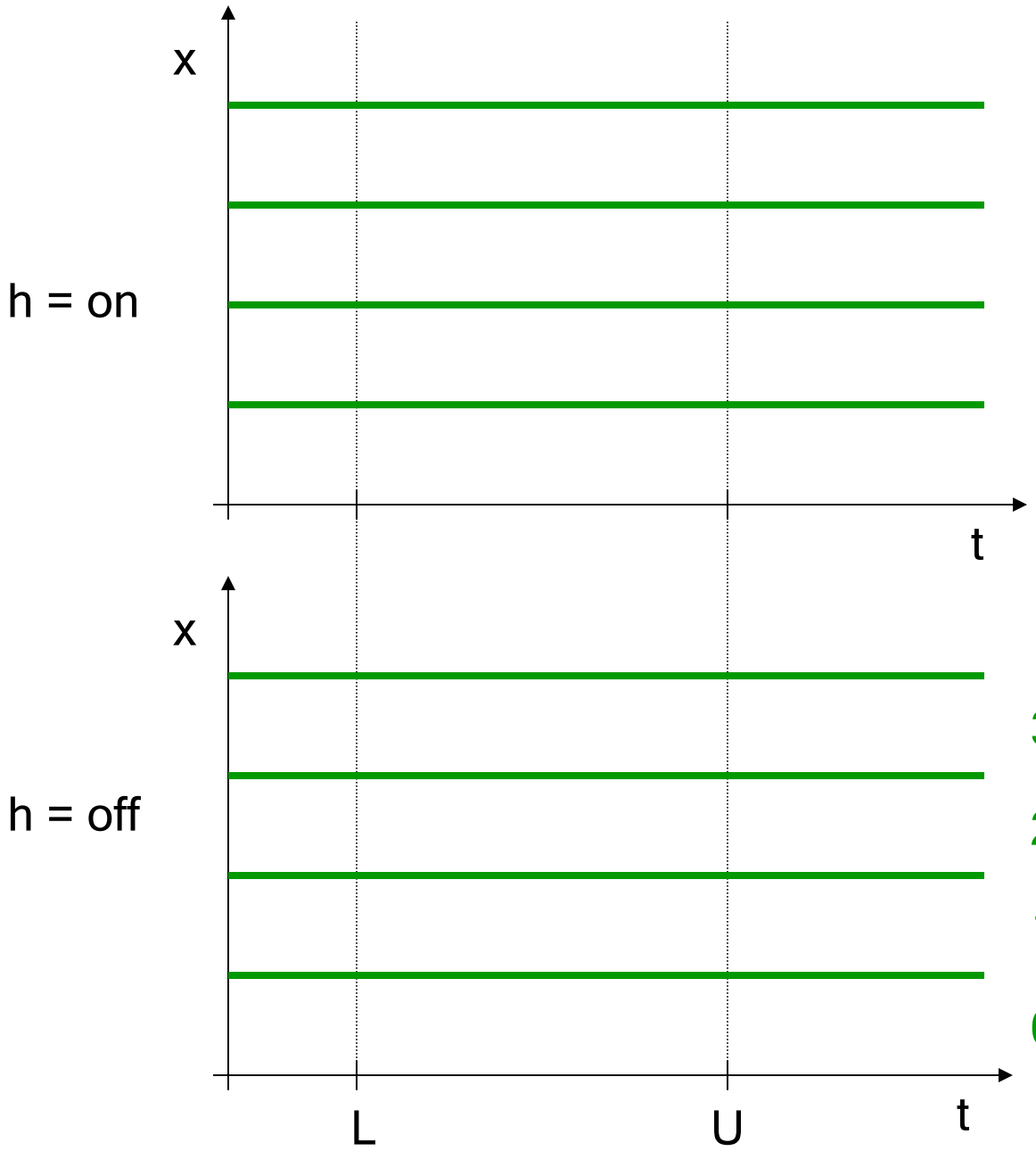
post: $2^Q \times \Sigma \rightarrow 2^Q$ $\quad\quad$ $post(R,\sigma) = \bigcup_{q \in Q} post(q,\sigma)$

pre: $2^Q \times \Sigma \rightarrow 2^Q$ $\quad\quad$ $pre(R,\sigma) = \bigcup_{q \in Q} pre(q,\sigma)$


x
h = on
t
x
h = off
R
L
U
t


x
h = on
t
x
pre(R,f2)
h = off
R
L
U
t


x
pre(pre(R,f2), j2)
h = on
t
x
pre(R,f2)
h = off
R
L
U
t



Step 3: Observe

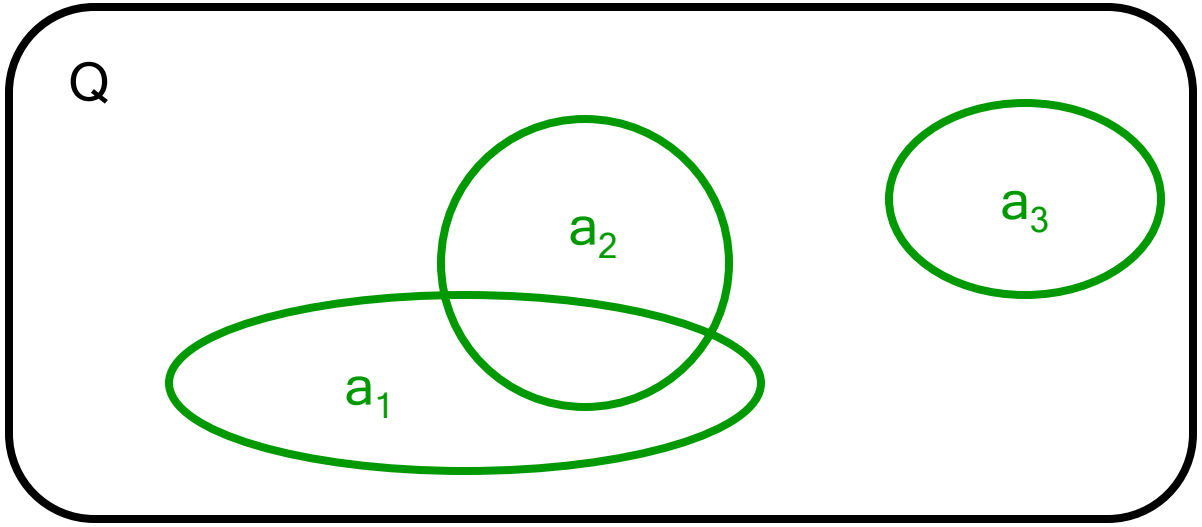
Observed Transition System

Q

Σ

pre, post: $2^Q \times \Sigma \rightarrow 2^Q$

A = { a_1, a_2, a_3, … } set of observations



Observed Transition System

Q

Σ

pre, post: $2^Q \times \Sigma \rightarrow 2^Q$

A — set of observations

Thermostat

A = { on, off } [{ x = c, c < x < c+1 | c $\in$ Z }

Symbolic Transition System

Q

Σ

pre, post

A

$\Re = \{ R_1, R_2, \ldots \}$ set of regions $R_i \subseteq Q$

Region algebra:

1. $A \subseteq \Re$
2. pre, post: $\Re \times \Sigma \to \Re$ computable
3. $\text{Å} : \Re^2 \to \Re$
   $\setminus : \Re^2 \to \Re$ computable
   $\subseteq : \Re^2 \to \{ t, f \}$

Symbolic Transition System

1. Local computation: Region Operations

Compute pre, post, Å , \ , and $\subseteq$ on regions in $\mathfrak{R}$.

2. Global computation: Symbolic Semi-Algorithms

Starting from the observations in A, compute new regions in $\mathfrak{R}$ by applying the operations pre, post, Å , \ , and $\subseteq$.

Region Algebra

If -Q is the valuations for a set X:Vals of typed variables,
-the effect of transitions can be expressed using Ops on Vals,
-the first-order theory FO(Vals,Ops) admits quantifier elimination,

then the quantifier-free fragment ZO(Vals,Ops) is a region algebra.

This is because each pre and post operation is a quantifier elimination:

$$pre(R(X)) = (\exists \underline{X})\ (Trans(X,\underline{X}) \wedge R(\underline{X}))$$

Example: Polyhedral Hybrid Automata

$Q = B^m \times R^n$

Invariants and guards:
boolean and linear constraints, e.g. $a \wedge (3x_1 + x_2 \leq 7)$

Flows: rectangular differential inclusions, e.g. $x'_1 \in [1,2]$

Jumps: boolean and linear constraints, e.g. $x_2 := 2x_1 + x_2 + 1$

Example: Polyhedral Hybrid Automata

$Q = B^m \times R^n$

Invariants and guards:
boolean and linear constraints, e.g. $a \wedge (3x_1 + x_2 \leq 7)$

Flows: rectangular differential inclusions, e.g. $x'_1 \in [1,2]$
Jumps: boolean and linear constraints, e.g. $x_2 := 2x_1 + x_2 + 1$

A = set of boolean valuations and integral polyhedra in R^n

Example: Polyhedral Hybrid Automata

$Q = B^m \times R^n$

Invariants and guards:
boolean and linear constraints, e.g. $a \wedge (3x_1 + x_2 \leq 7)$

Flows: rectangular differential inclusions, e.g. $x'_1 \in [1,2]$
Jumps: boolean and linear constraints, e.g. $x_2 := 2x_1 + x_2 + 1$

A = set of boolean valuations and integral polyhedra in R^n

$\mathfrak{R}$ = set of boolean valuations and rational polyhedra in R^n
= ... $ZO(Q,\leq,+)$

Example: Polyhedral Hybrid Automata

FO(Q,≤,+) admits quantifier elimination,
hence ZO(Q,≤,+) is a region algebra.

Jump j: $Y\ x_1 \le x_2 \rightarrow x_2 := 2x_1 - 1$

pre($1 \le x_1 \le x_2 \le 2$, j)

$= (\exists\ \underline{x}_1, \underline{x}_2)\ (x_1 \le x_2 \wedge \underline{x}_1 = x_1 \wedge \underline{x}_2 = 2x_1 - 1 \wedge 1 \le \underline{x}_1 \le \underline{x}_2 \le 2)$

Example: Polyhedral Hybrid Automata

FO(Q,≤,+) admits quantifier elimination,
hence ZO(Q,≤,+) is a region algebra.

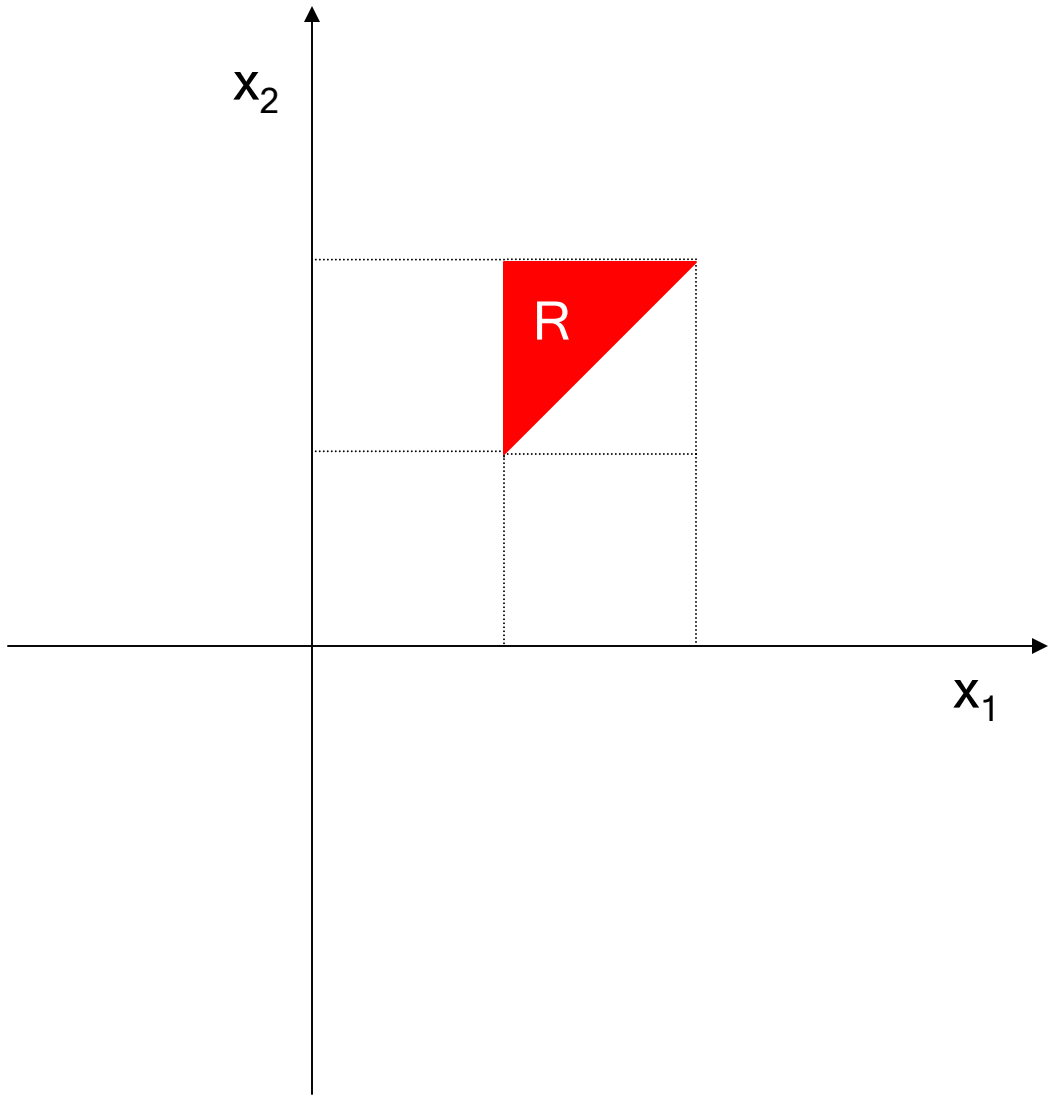
Jump j: $Y\ x_1 \le x_2 \rightarrow x_2 := 2x_1 - 1$

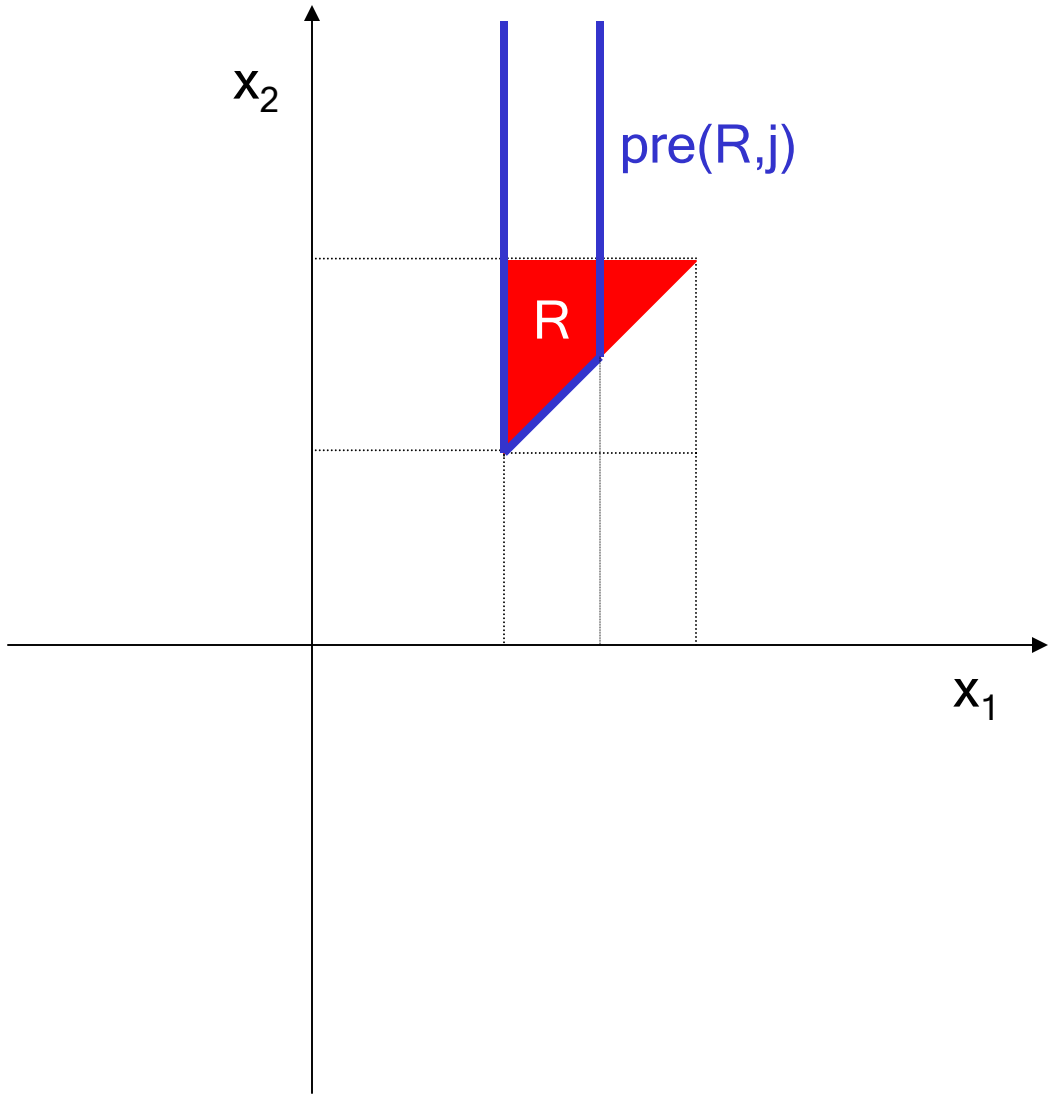
pre($1 \le x_1 \le x_2 \le 2$, j)

$= (\exists\ \underline{x}_1, \underline{x}_2)\ (x_1 \le x_2 \wedge \underline{x}_1 = x_1 \wedge \underline{x}_2 = 2x_1 - 1 \wedge 1 \le \underline{x}_1 \le \underline{x}_2 \le 2)$

$= x_1 \le x_2 \wedge 1 \le x_1 \le 2x_1 - 1 \le 2$

$= x_1 \le x_2 \wedge 1 \le x_1 \le 3/2$


x_2
R
x_1


x2
pre(R,j)
R
x1

Example: Polyhedral Hybrid Automata

Flow f: $Y\ x_1 \le x_2 \rightarrow x'_1 \in [1,2];\ x'_2 = 1$

$$\text{pre}(\ 1 \le x_1 \le x_2 \le 2,\ f\ )$$
$$= (\exists\ 1 \le k_1 \le 2)\ (\exists\ \delta \ge 0)\ (1 \le x_1 + k_1\delta \le x_2 + \delta \le 2\ \wedge$$
$$(\forall\ 0 \le \varepsilon \le \delta)\ (x_1 + k_1\varepsilon \le x_2 + \varepsilon))$$

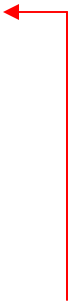
Example: Polyhedral Hybrid Automata

Flow f: $\quad Y\ x_1 \le x_2 \rightarrow x'_1 \in [1,2];\ x'_2 = 1$

pre($1 \le x_1 \le x_2 \le 2$, f)

$$= (\exists\ 1 \le k_1 \le 2)\ (\exists\ \delta \ge 0)\ (1 \le x_1 + k_1\delta \le x_2 + \delta \le 2\ \wedge$$
$$(\forall\ 0 \le \varepsilon \le \delta)\ (x_1 + k_1\varepsilon \le x_2 + \varepsilon))$$

$$= (\exists\ \delta \ge 0)\ (\exists\ \delta \le d_1 \le 2\delta)\ (1 \le x_1 + d_1 \le x_2 + \delta \le 2\ \wedge$$
$$x_1 \le x_2\ \wedge\ x_1 + d_1 \le x_2 + \delta)$$



convex guards

Example: Polyhedral Hybrid Automata

Flow f: $Y\ x_1 \le x_2 \rightarrow x'_1 \in [1,2];\ x'_2 = 1$

pre($1 \le x_1 \le x_2 \le 2$, f)

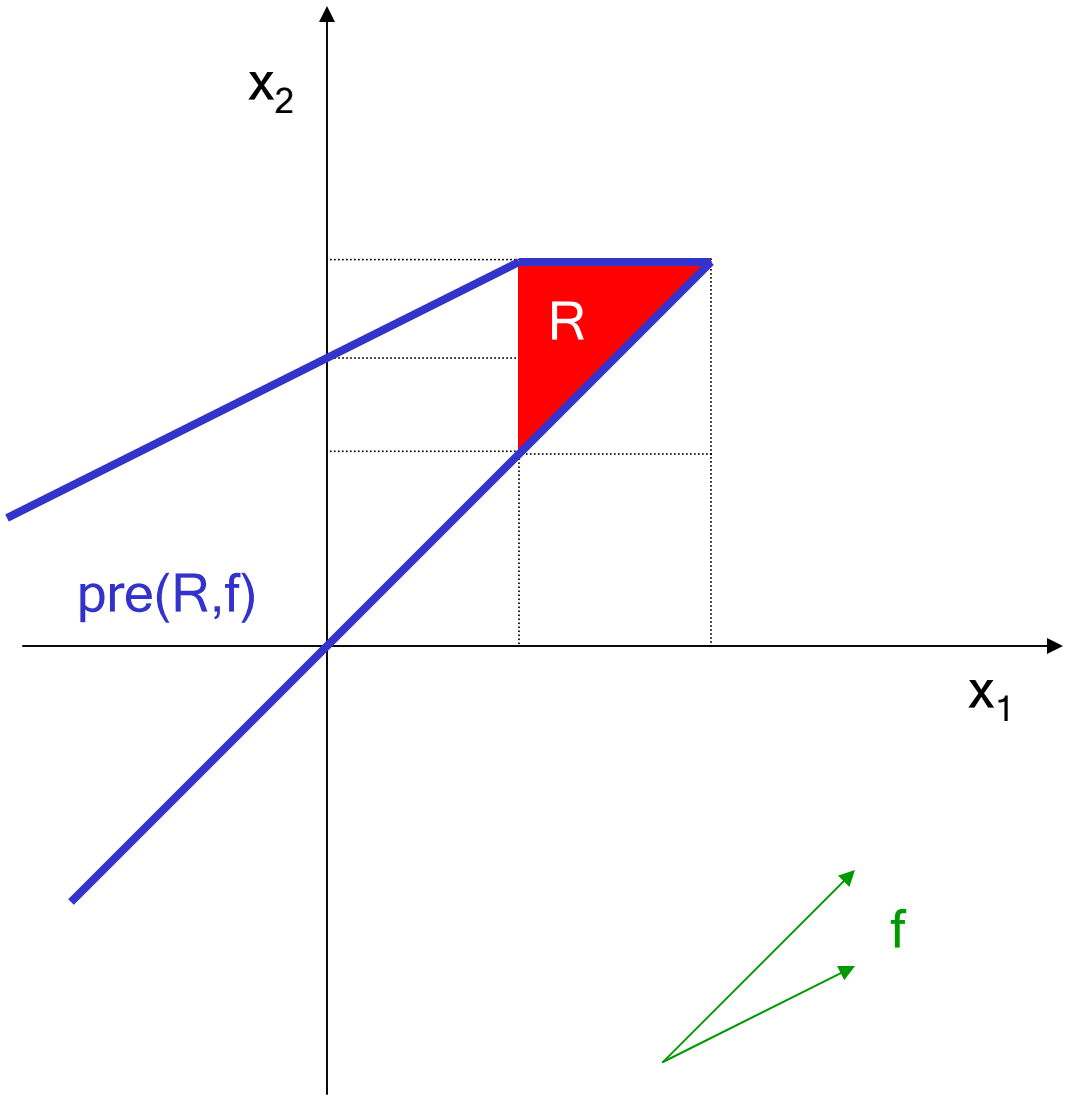
$= (\exists\ 1 \le k_1 \le 2)\ (\exists\ \delta \ge 0)\ (1 \le x_1 + k_1\delta \le x_2 + \delta \le 2\ \wedge$
$(\forall\ 0 \le \varepsilon \le \delta)\ (x_1 + k_1\varepsilon \le x_2 + \varepsilon))$

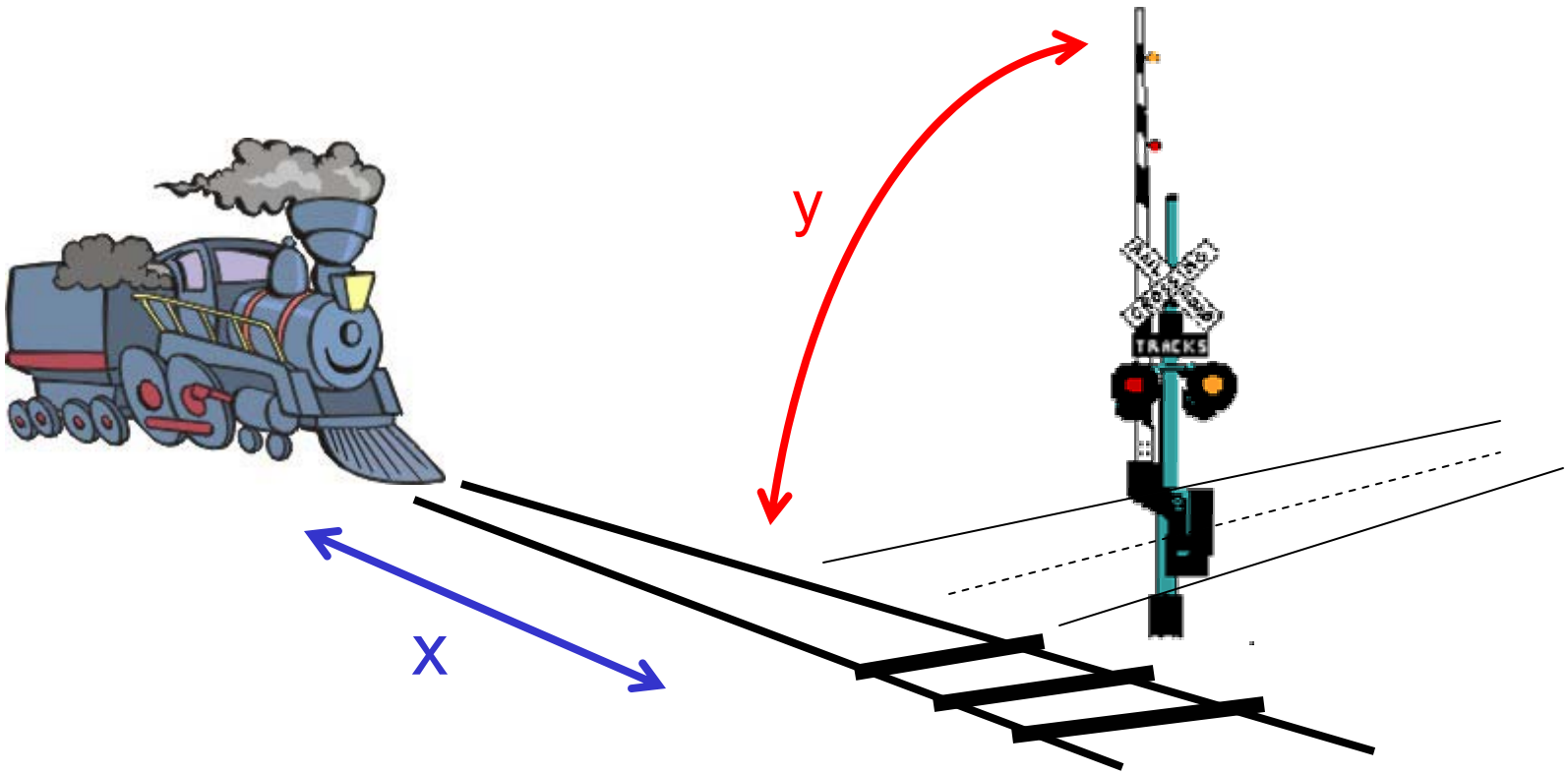
$= (\exists\ \delta \ge 0)\ (\exists\ \delta \le d_1 \le 2\delta)\ (1 \le x_1 + d_1 \le x_2 + \delta \le 2\ \wedge$
$x_1 \le x_2\ \wedge\ x_1 + d_1 \le x_2 + \delta)$

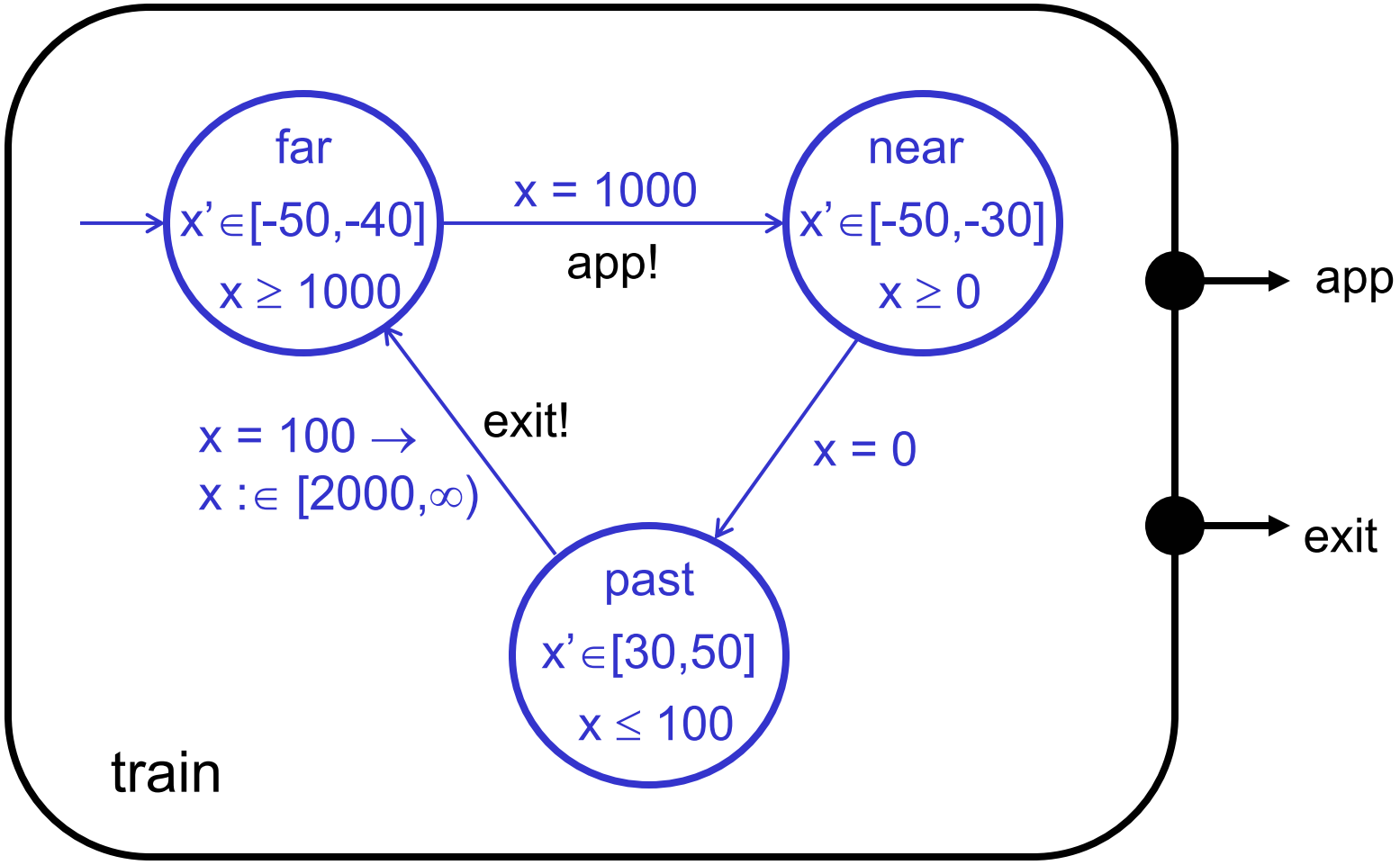
$= (\exists\ \delta \ge 0)\ (x_1 \le x_2\ \wedge\ 1 \le x_1 + 2\delta\ \wedge\ 1 \le x_2 + \delta \le 2)$

$= x_1 \le x_2\ \wedge\ x_2 \le 2\ \wedge\ 2x_2 \le x_1 + 3$

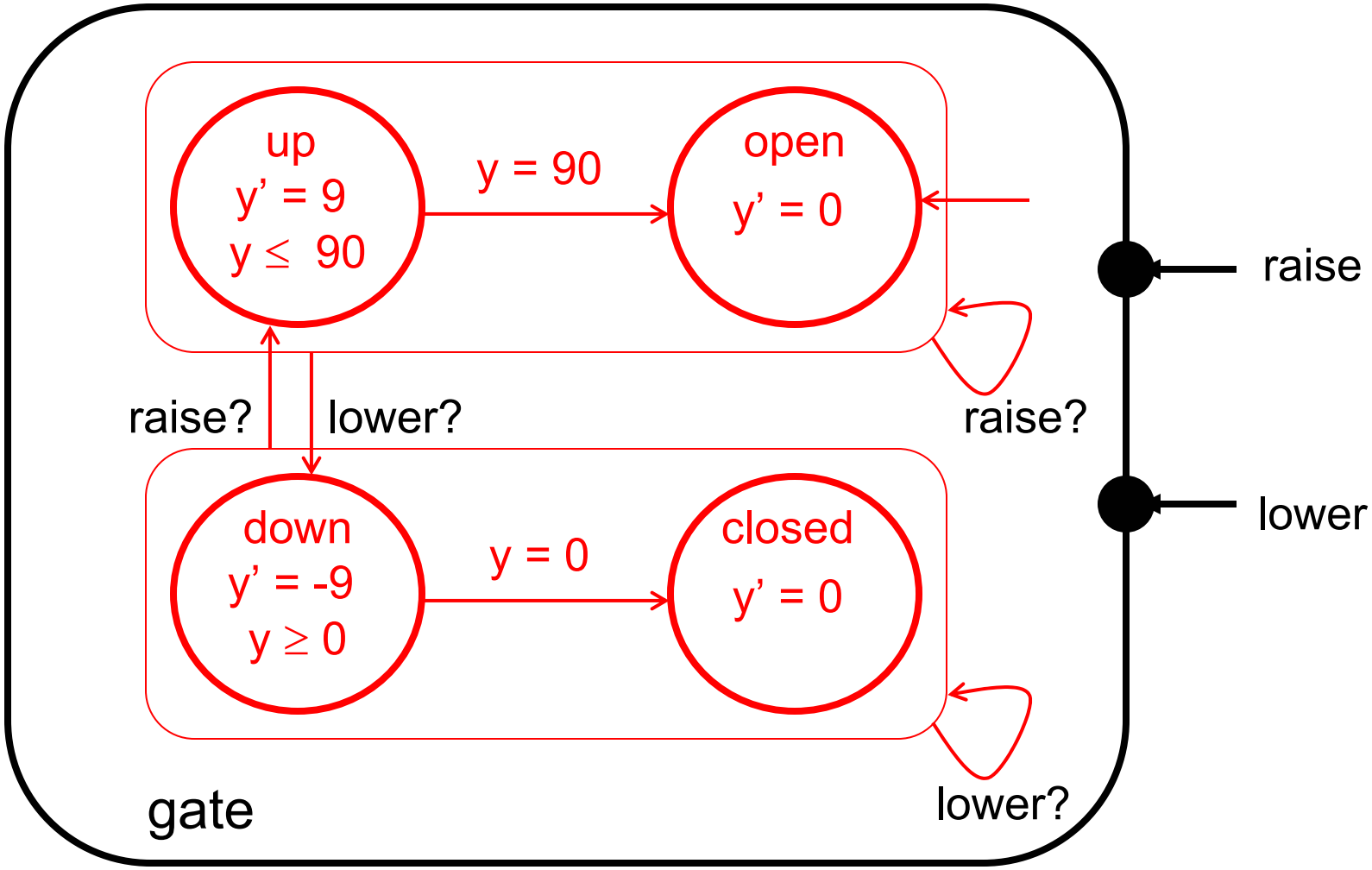
convex guards


x2
R
pre(R,f)
x1
f

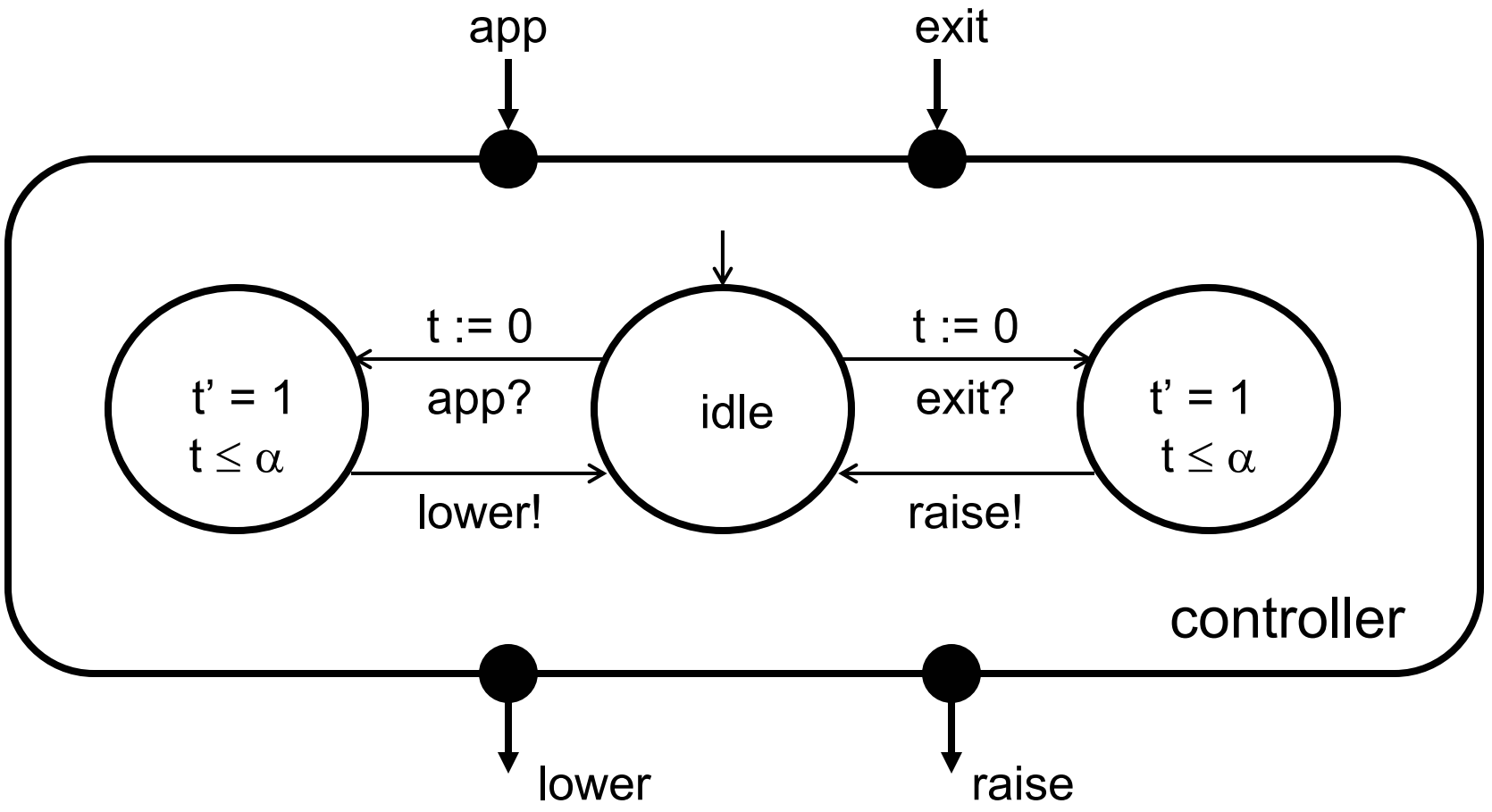

y
x
TRACKS



x: initialized rectangular variable



y: uninitialized singular variable



t: clock variable

α: design parameter

Properties

Safety: $\forall\Box\ (\ x \leq 10\ \Rightarrow\ \text{loc[gate]} = \text{closed}\ )$

For which values of α is this true?

Properties

Safety: $\forall\Box\ (\ x \leq 10\ \Rightarrow \text{loc[gate]} = \text{closed}\ )$

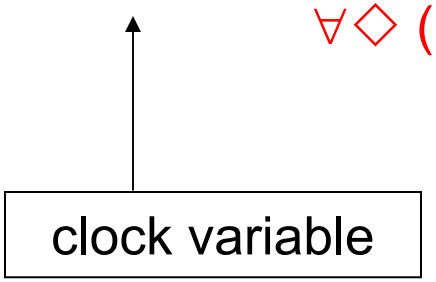
Liveness: $\forall\Box\ \forall\Diamond\ (\ \text{loc[gate]} = \text{open}\ )$

Properties

Safety: $\forall\Box\ (\ x \leq 10\ \Rightarrow \text{loc[gate]} = \text{closed}\ )$

Liveness: $\forall\Box\ \forall\Diamond\ (\ \text{loc[gate]} = \text{open}\ )$

Real time: $\forall\Box\ z := 0.\ (\ z' = 1\ \Rightarrow$
$\forall\Diamond\ (\ \text{loc[gate]} = \text{open}\ \wedge\ z \leq 60\ ))$



Properties

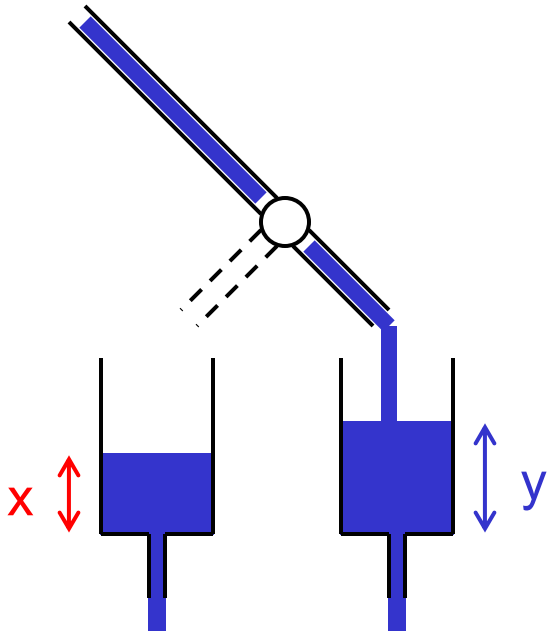
Safety: $\forall\Box\ (\ x \leq 10\ \Rightarrow \text{loc[gate]} = \text{closed}\ )$

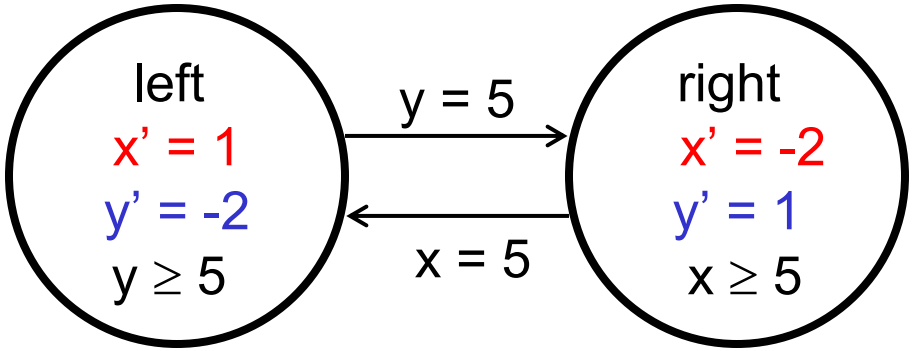
Liveness: $\forall\Box\ \forall\Diamond\ (\ \text{loc[gate]} = \text{open}\ )$

Real time: $\forall\Box\ z := 0.\ (\ z' = 1\ \Rightarrow\ \forall\Diamond\ (\ \text{loc[gate]} = \text{open}\ \wedge\ z \leq 60\ ))$

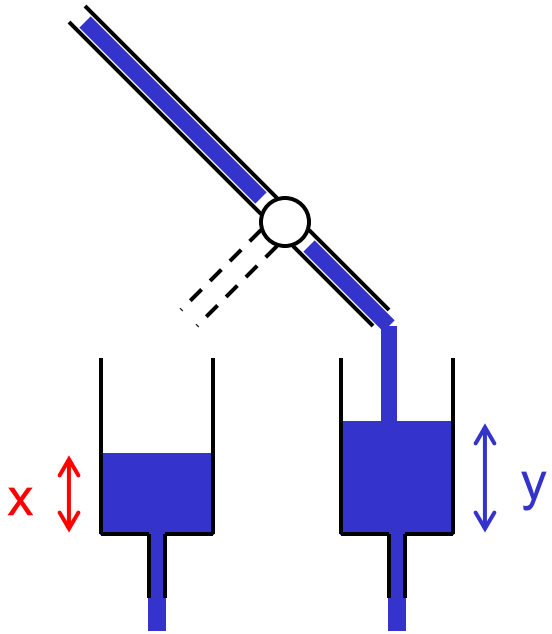
Nonzeno: $\forall\Box\ z := 0.\ (\ z' = 1\ \Rightarrow\ \exists\Diamond\ (\ z = 1\ ))$

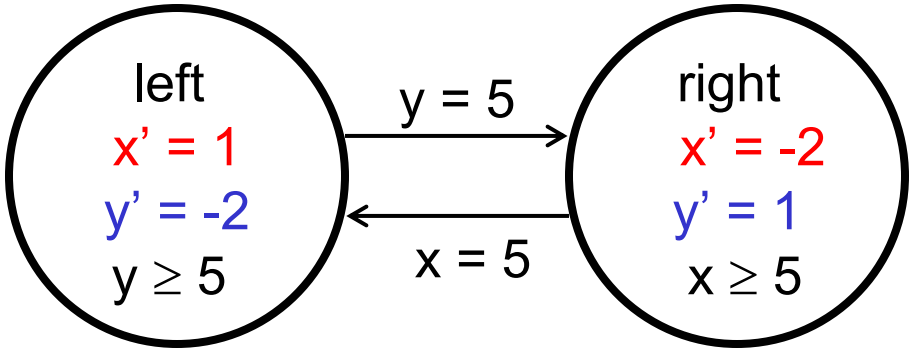
A Zeno System

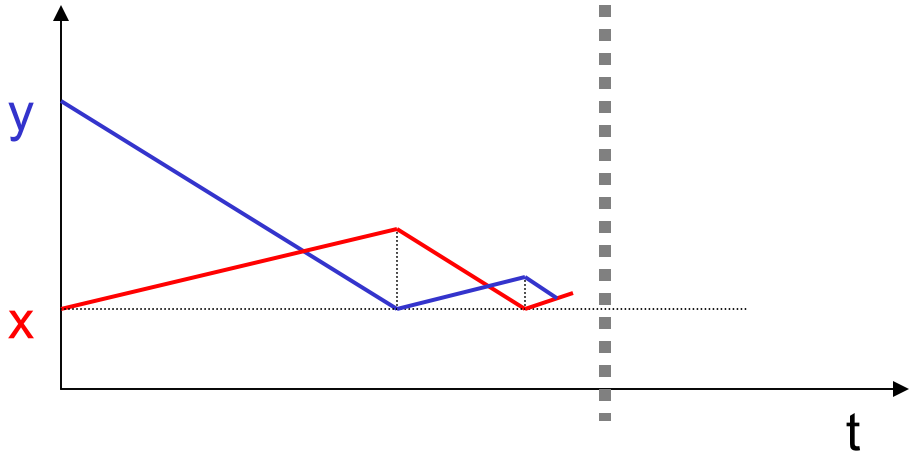


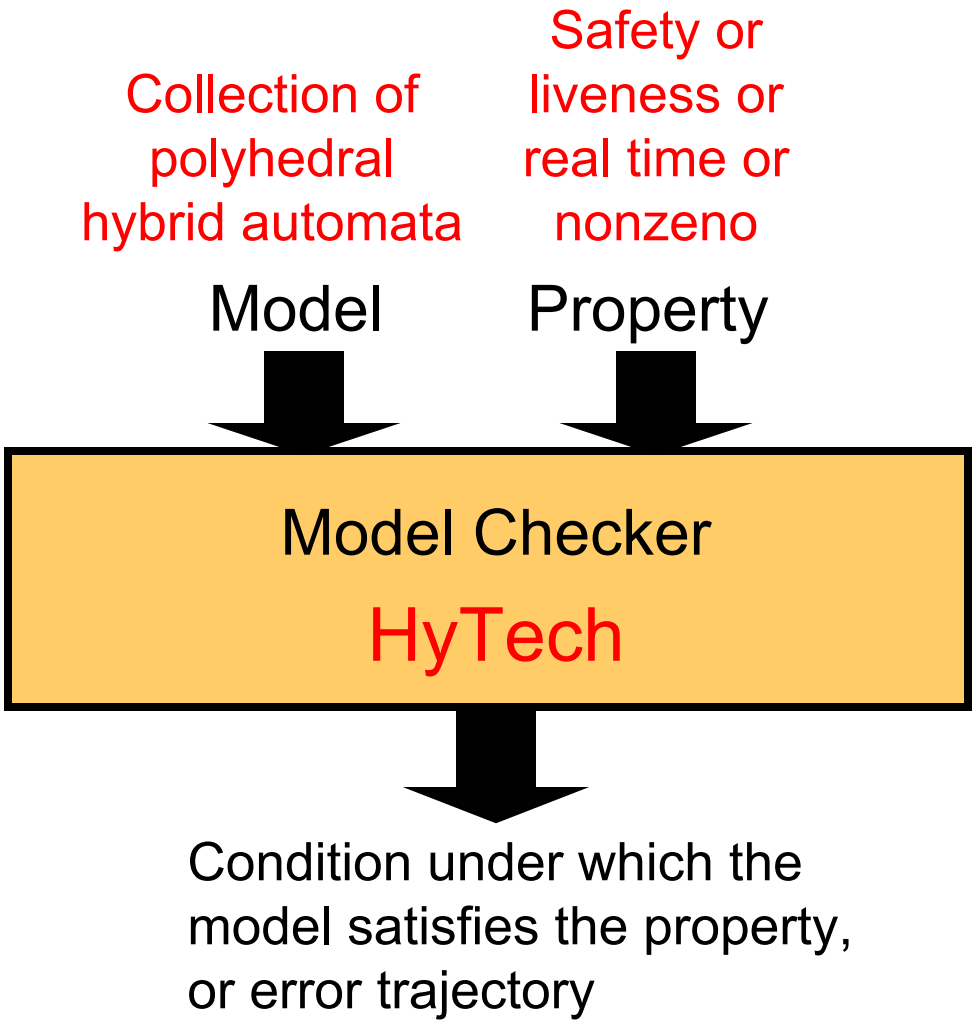


A Zeno System

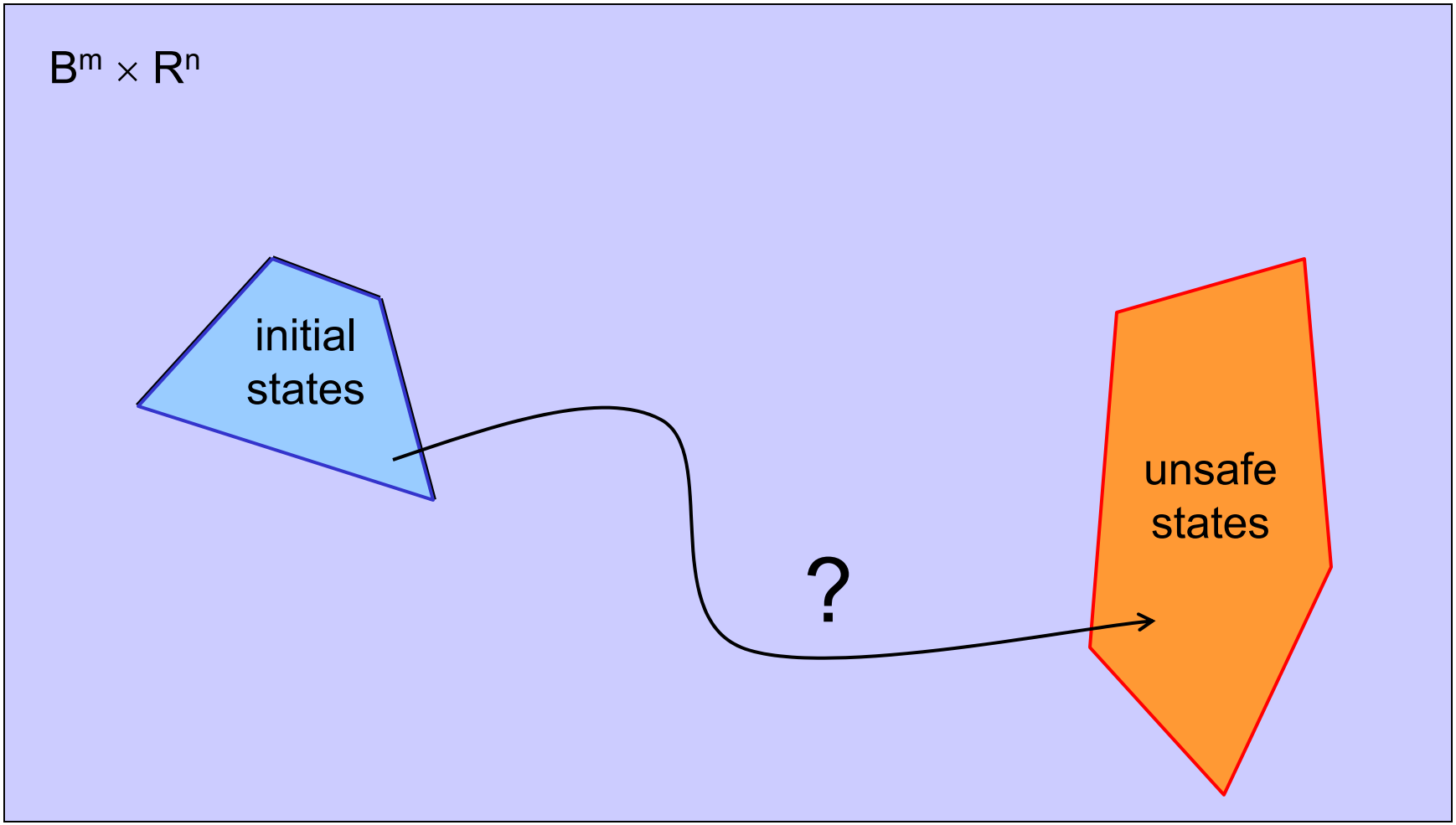




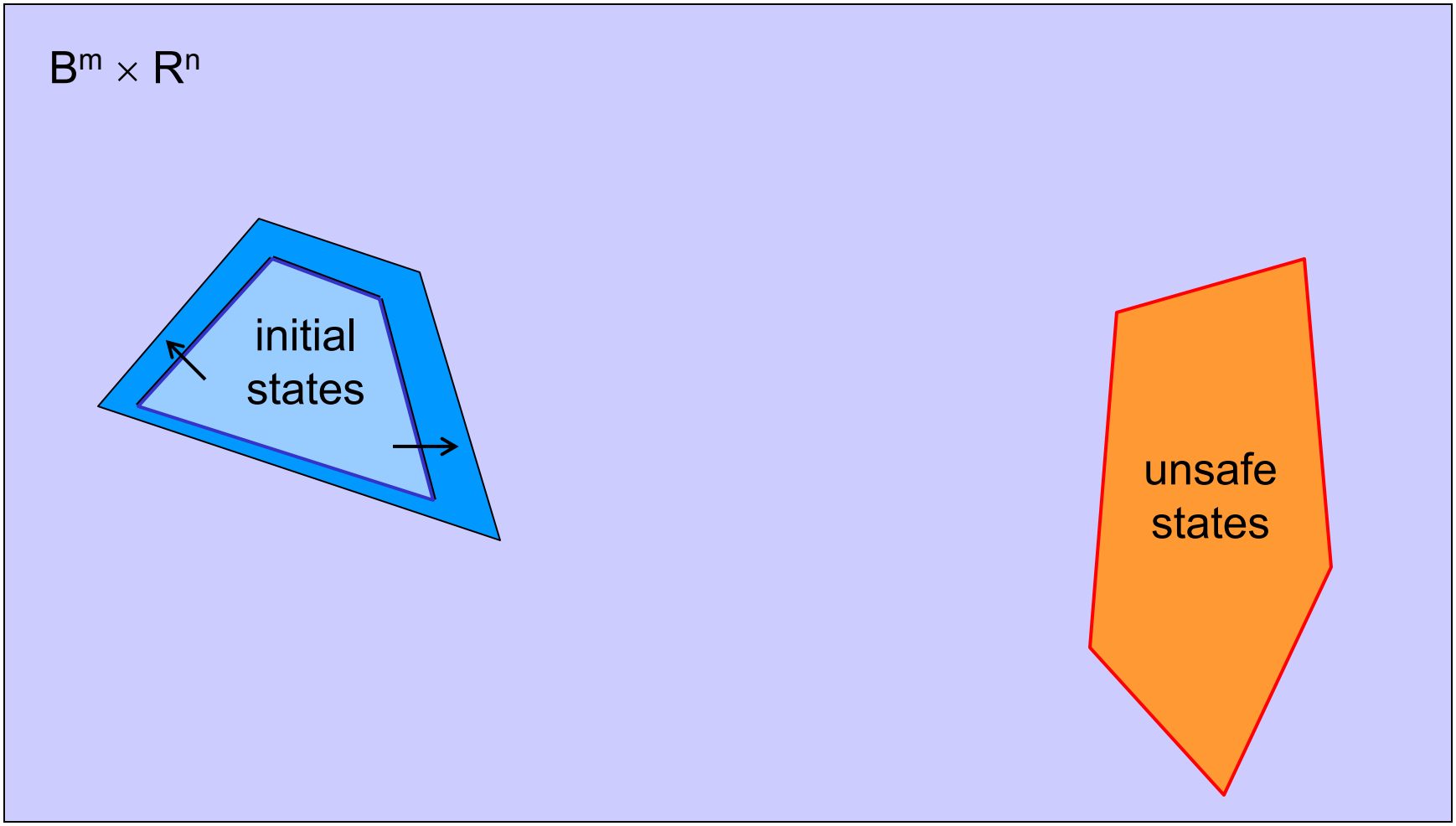



Collection of polyhedral hybrid automata
Safety or liveness or real time or nonzeno
Model
Property
Model Checker
HyTech
Condition under which the model satisfies the property, or error trajectory

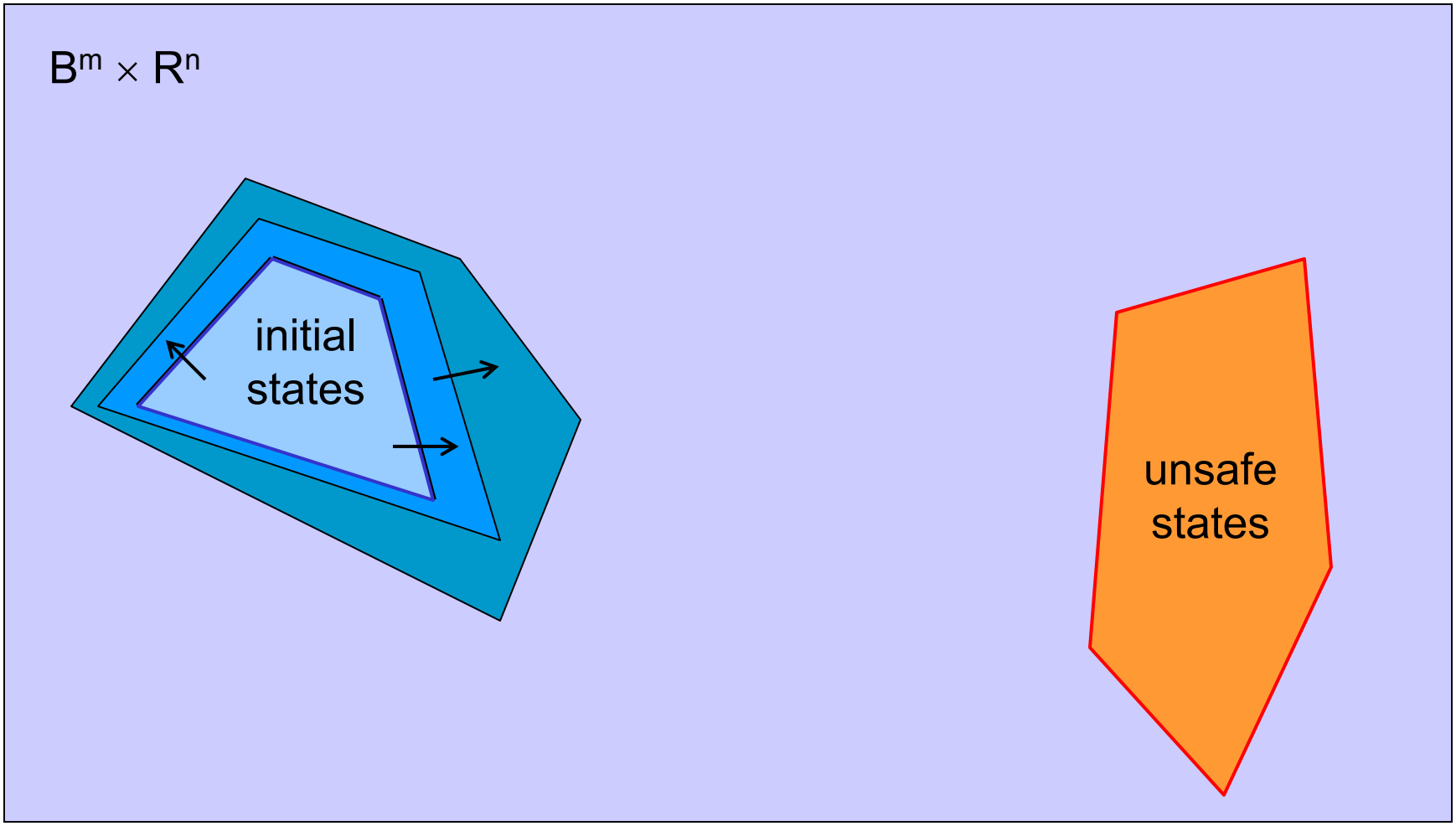
Model Checking for Safety



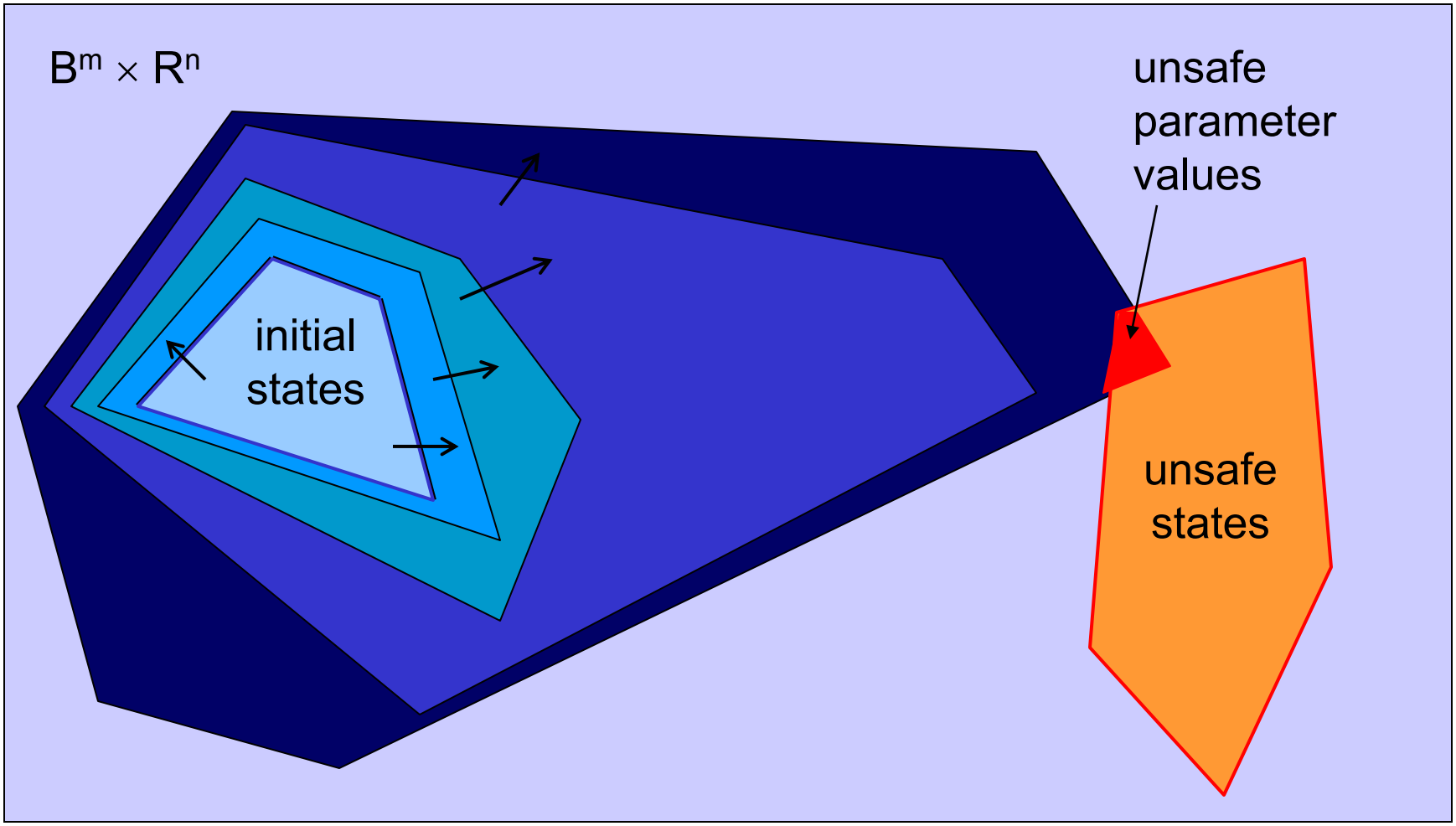
Model Checking for Safety



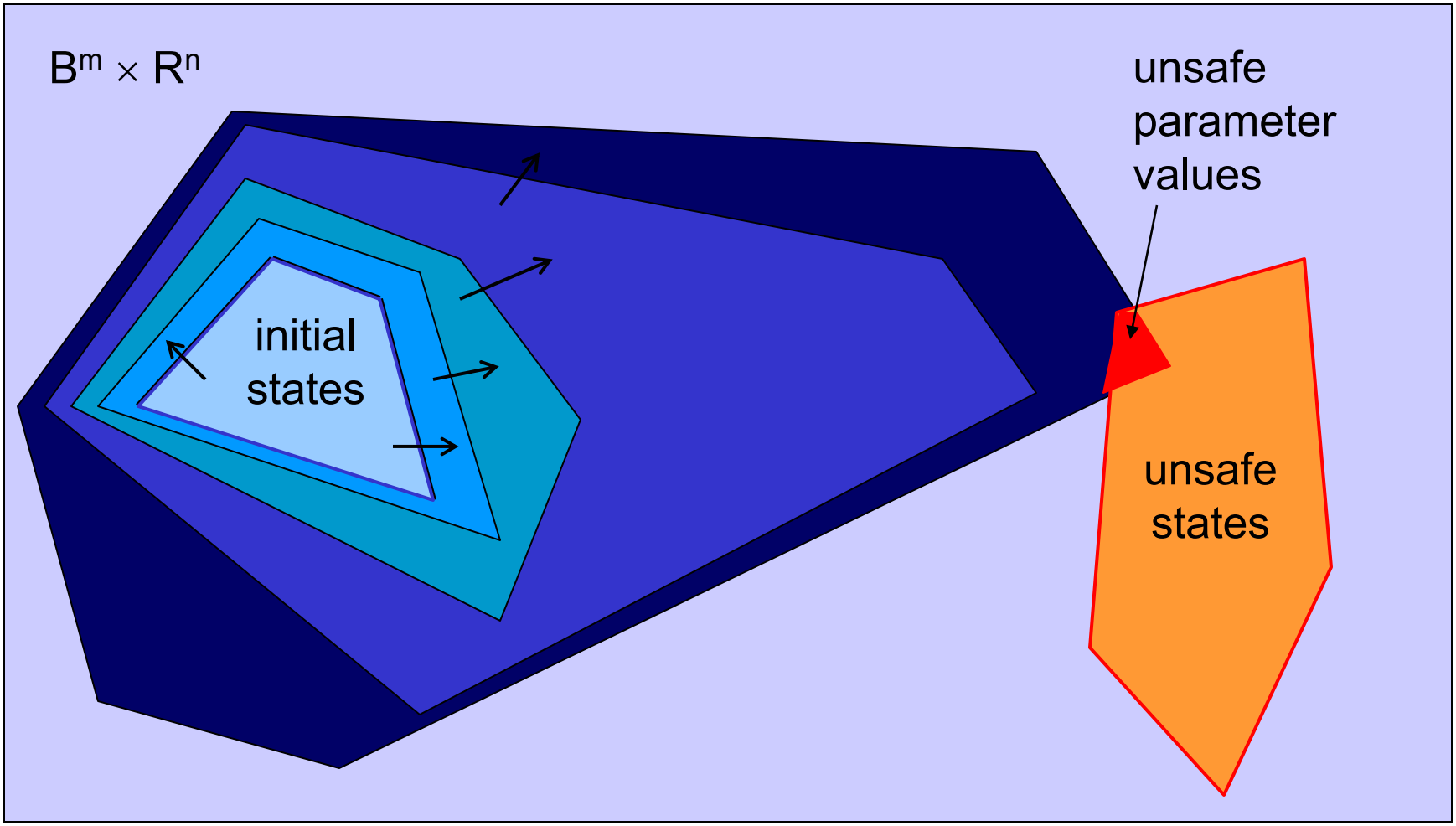
Model Checking for Safety



Model Checking for Safety



Model Checking for Safety



This is guaranteed to terminate if all variables are initialized (e.g. clocks).

The Result

hytech-out.cgi

Applications of HyTech and Derivations:
polyhedral overapproximation of dynamics

-automotive engine control [Wong-Toi et al.]
-chemical plant control [Preussig et al.]
-flight control [Honeywell; Rockwell-Collins]
-air traffic control [Tomlin et al.]
-robot control [Corbett et al.]

Successor Tools:

1. More expressive region algebras, e.g. FO(R,≤,+,·) still permits quantifier elimination [Pappas et al.]

2. More robust approximations, e.g. ellipsoid instead of polyhedral regions [Varaiya et al.]

3. Abstract region operations, e.g. interval numerical methods [HyperTech]; also [Krogh et al.][Maler et al.]

Symbolic Semi-Algorithms

Starting from the observations in A, compute new regions in $\mathfrak{R}$ by applying the operations pre, post, Å , \ , and $\subseteq$.

Termination?

Four Verification Questions

V1: Reachability $\exists\Diamond$ b

Is an invariant always true? $\exists\Diamond$ unsafe

Four Verification Questions

V1:	Reachability	$\exists\Diamond$ b
	Is an invariant always true?	$\exists\Diamond$ unsafe
V2:	Repeated Reachability	$\exists\Box\Diamond$ b
	Linear temporal logic (LTL)	
	Liveness	$\exists\Box\Diamond$ unfair

Four Verification Questions

V1:	Reachability	$\exists\Diamond\ b$
	Is an invariant always true?	$\exists\Diamond$ unsafe
V2:	Repeated Reachability	$\exists\Box\Diamond\ b$
	Linear temporal logic (LTL)	
	Liveness	$\exists\Box\Diamond$ unfair
V3:	Nested Reachability	$\exists\Diamond\ (b \wedge \exists\Diamond\ b_1 \wedge \exists\Diamond\ b_2)$
	Half branching temporal logic ($\exists$CTL, $\forall$CTL)	

Four Verification Questions

V1: Reachability $\exists\Diamond\ b$

Is an invariant always true? $\exists\Diamond$ unsafe

V2: Repeated Reachability $\exists\Box\Diamond\ b$

Linear temporal logic (LTL)

Liveness $\exists\Box\Diamond$ unfair

V3: Nested Reachability $\exists\Diamond\ (b \wedge \exists\Diamond\ b_1 \wedge \exists\Diamond\ b_2)$

Half branching temporal logic ($\exists$CTL, $\forall$CTL)

V4: Negated Reachability $\forall\Box\ (b \rightarrow \exists\Diamond\ c)$

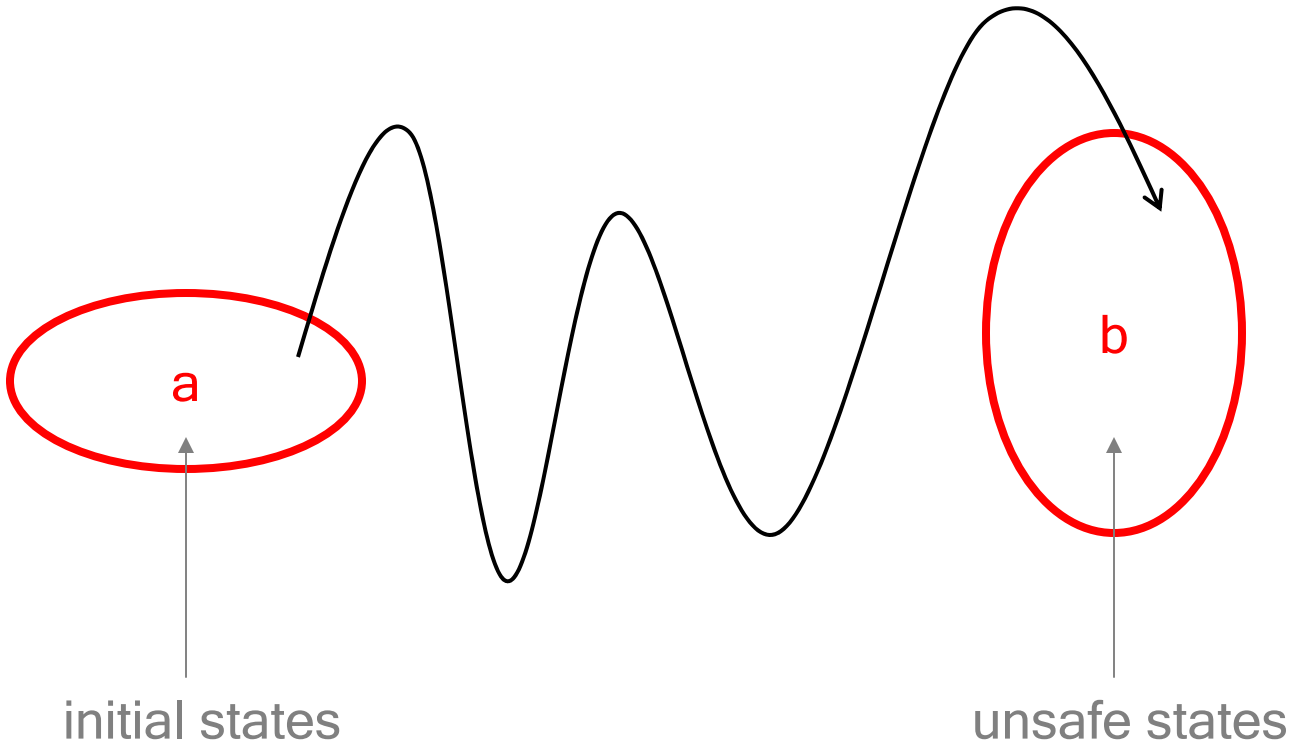
Full branching temporal logic (CTL)

Nonzenoness $\forall\Box\ (\text{tick} \rightarrow \exists\Diamond\ \text{tick})$

V1: Symbolic Reachability

$a \wedge \exists \Diamond b$

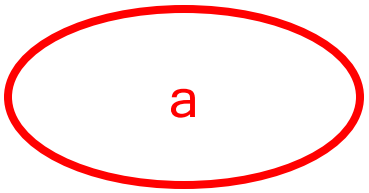
Given $a,b \in A$, is there a trajectory from a to b?

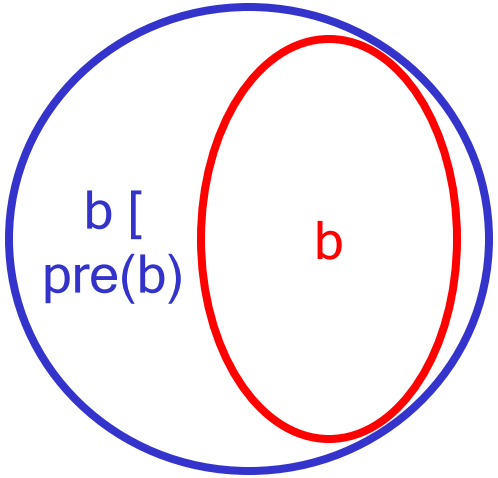


V1: Symbolic Reachability

$a \wedge \exists\Diamond\, b$

Given $a,b \in A$, is there a trajectory from a to b?



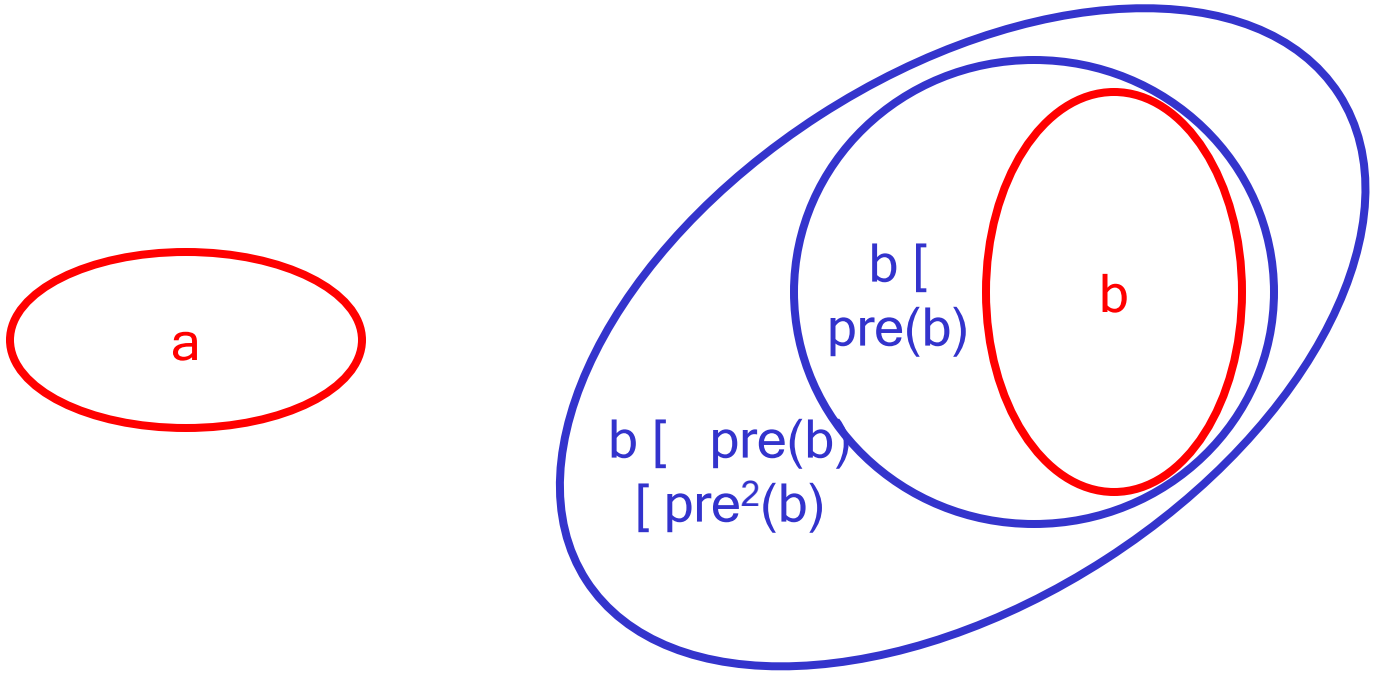


$pre(b) = \bigcup_{\sigma \in \Sigma} pre(b,\sigma)$

V1: Symbolic Reachability

$a \wedge \exists \Diamond b$

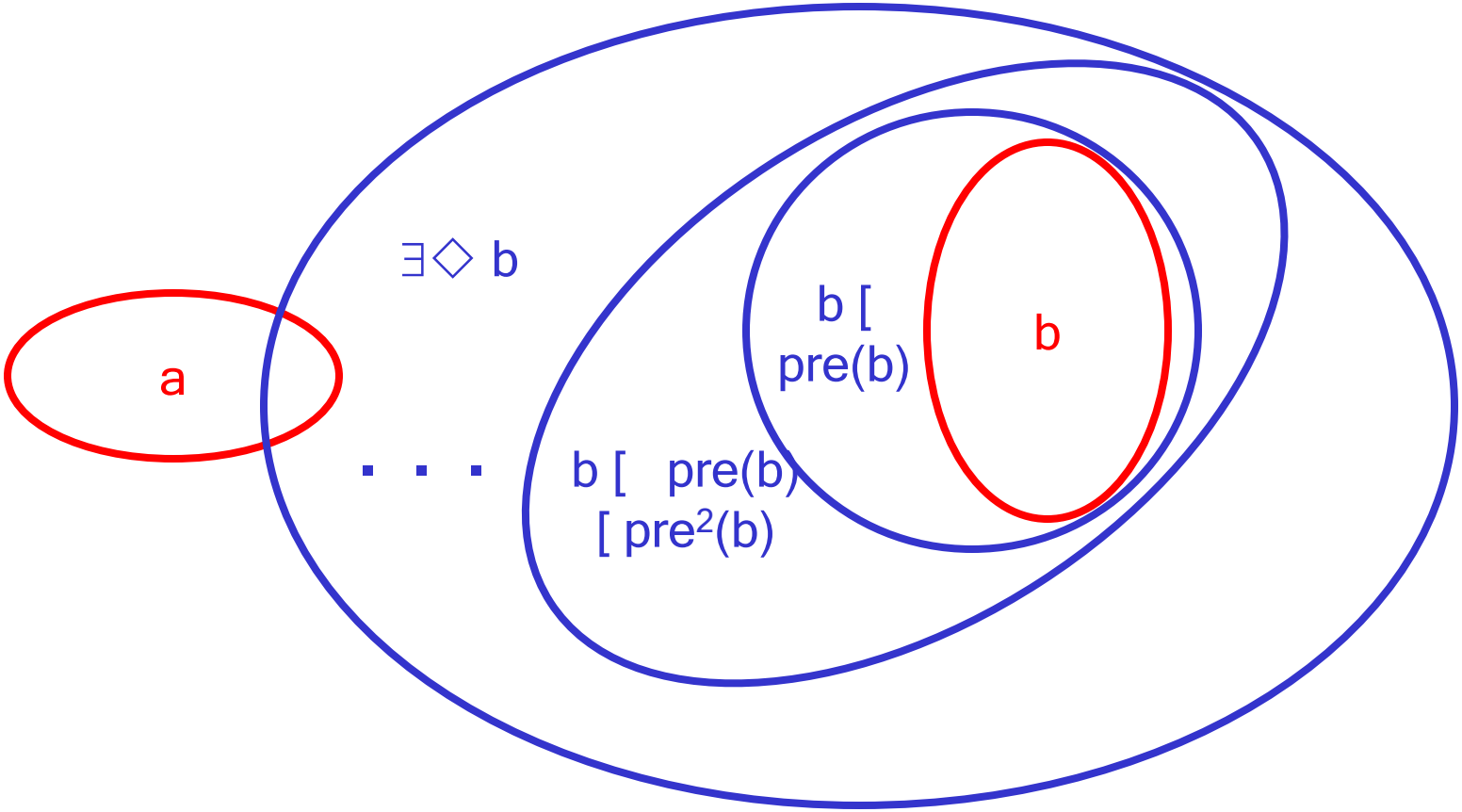
Given $a,b \in A$, is there a trajectory from a to b?



V1: Symbolic Reachability

$a \wedge \exists\Diamond\, b$

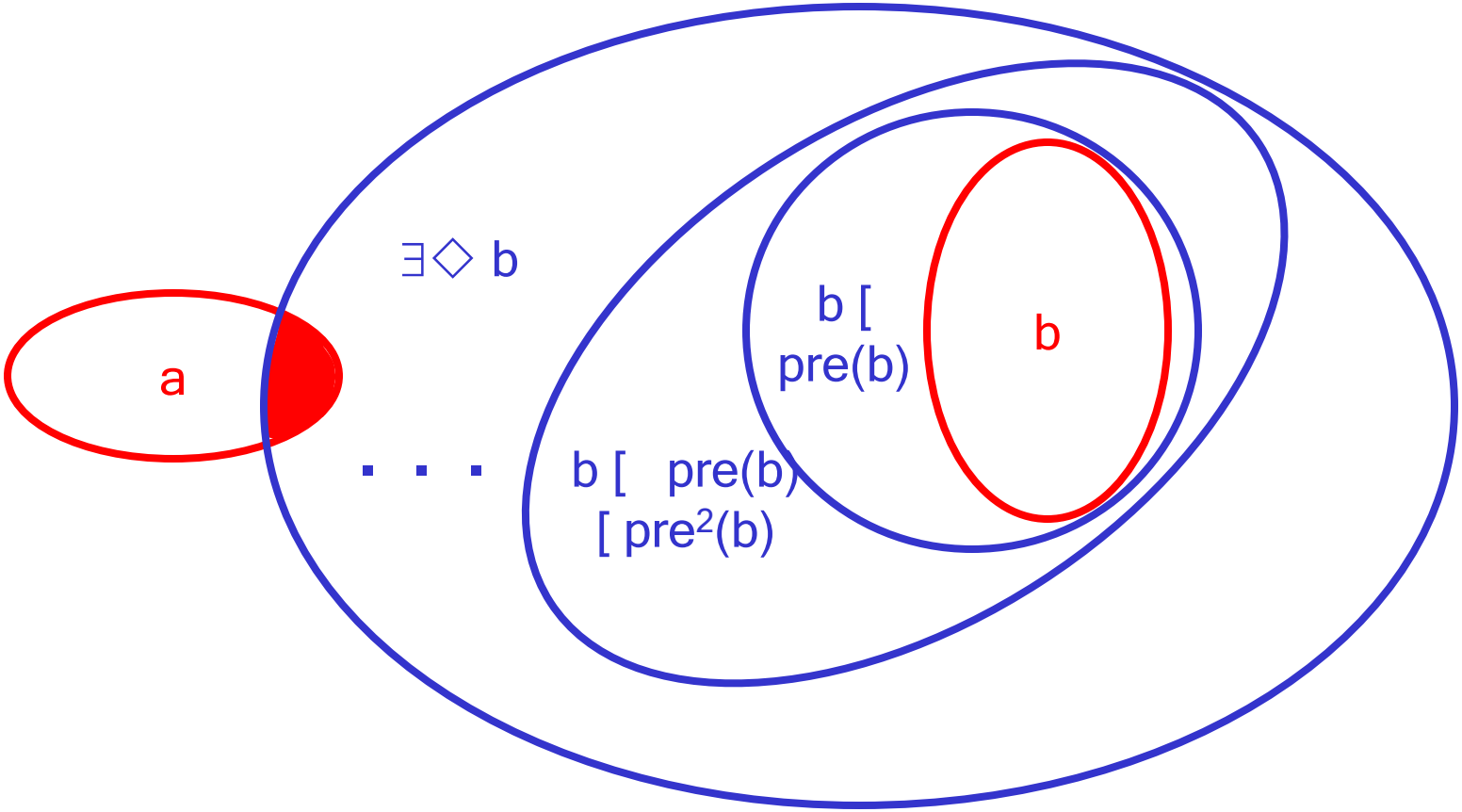
Given $a,b \in A$, is there a trajectory from a to b?



V1: Symbolic Reachability

$a \wedge \exists\Diamond\, b$

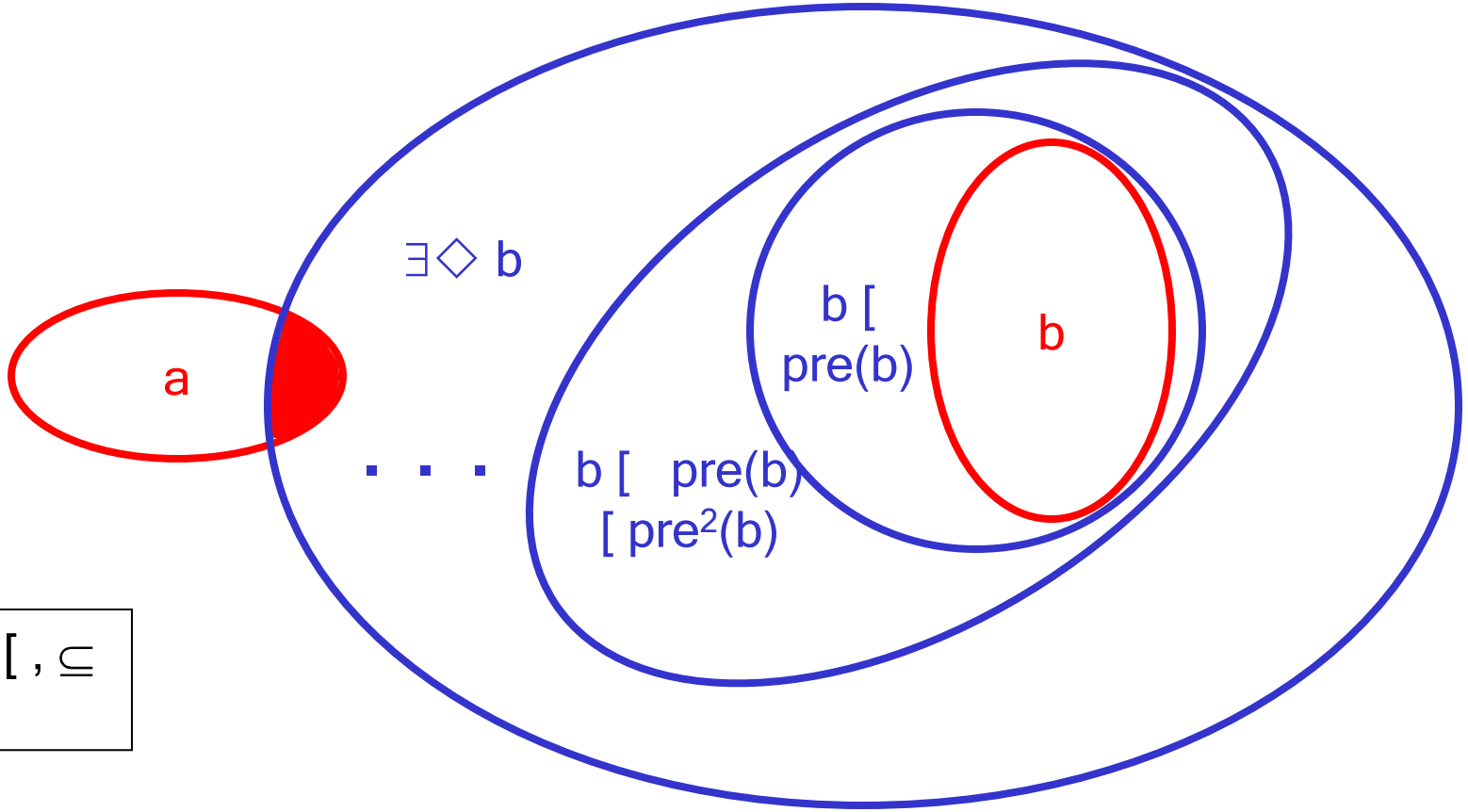
Given $a,b \in A$, is there a trajectory from a to b?



V1: Symbolic Reachability

$a \wedge \exists\Diamond\, b$

Given $a,b \in A$, is there a trajectory from a to b?

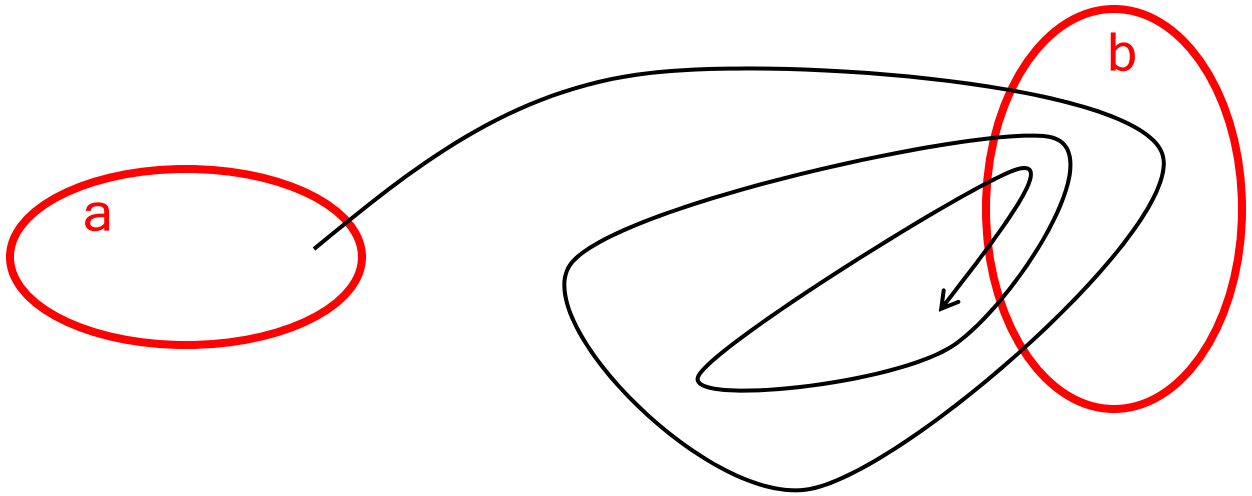


1. pre, [, ⊆
2. Å a

V2: Symbolic Repeated Reachability

$a \wedge \exists \Box \Diamond b$

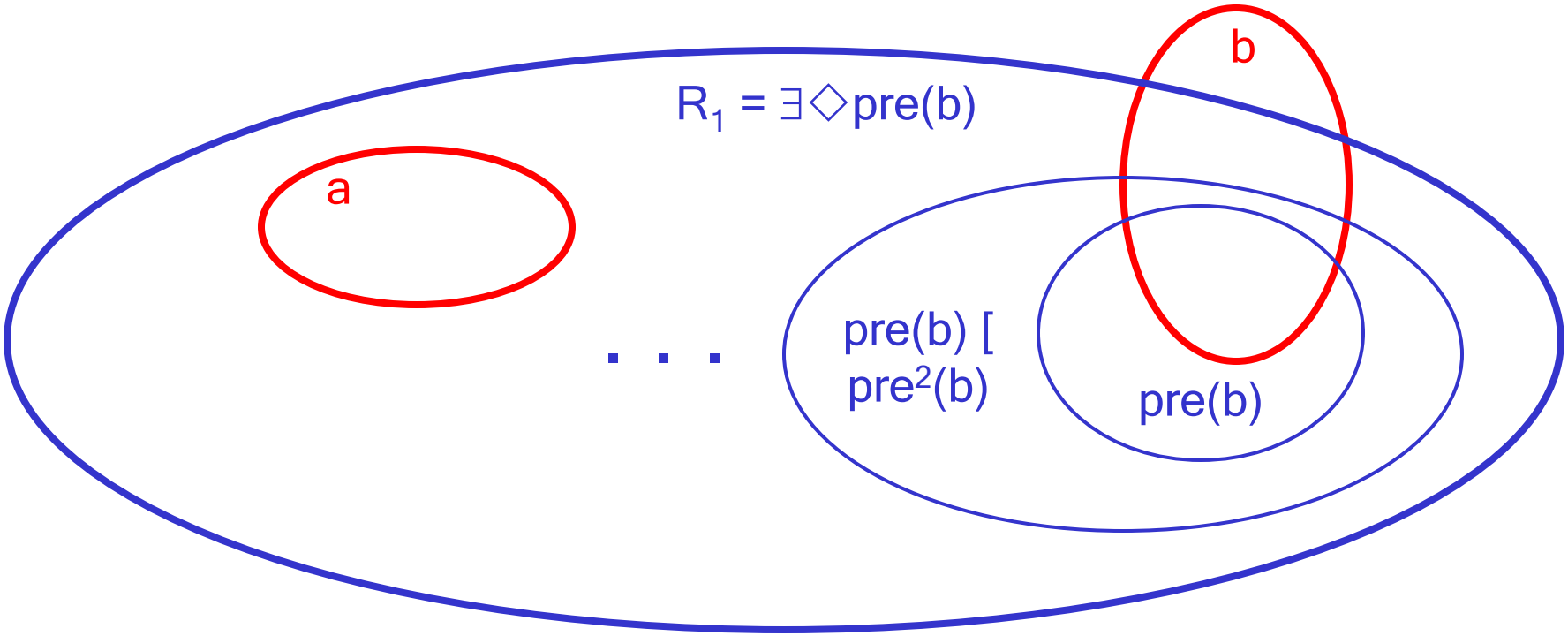
Given $a,b \in A$, is there an infinite trajectory from a that visits b infinitely often?



V2: Symbolic Repeated Reachability

$a \wedge \exists\Box\Diamond\, b$

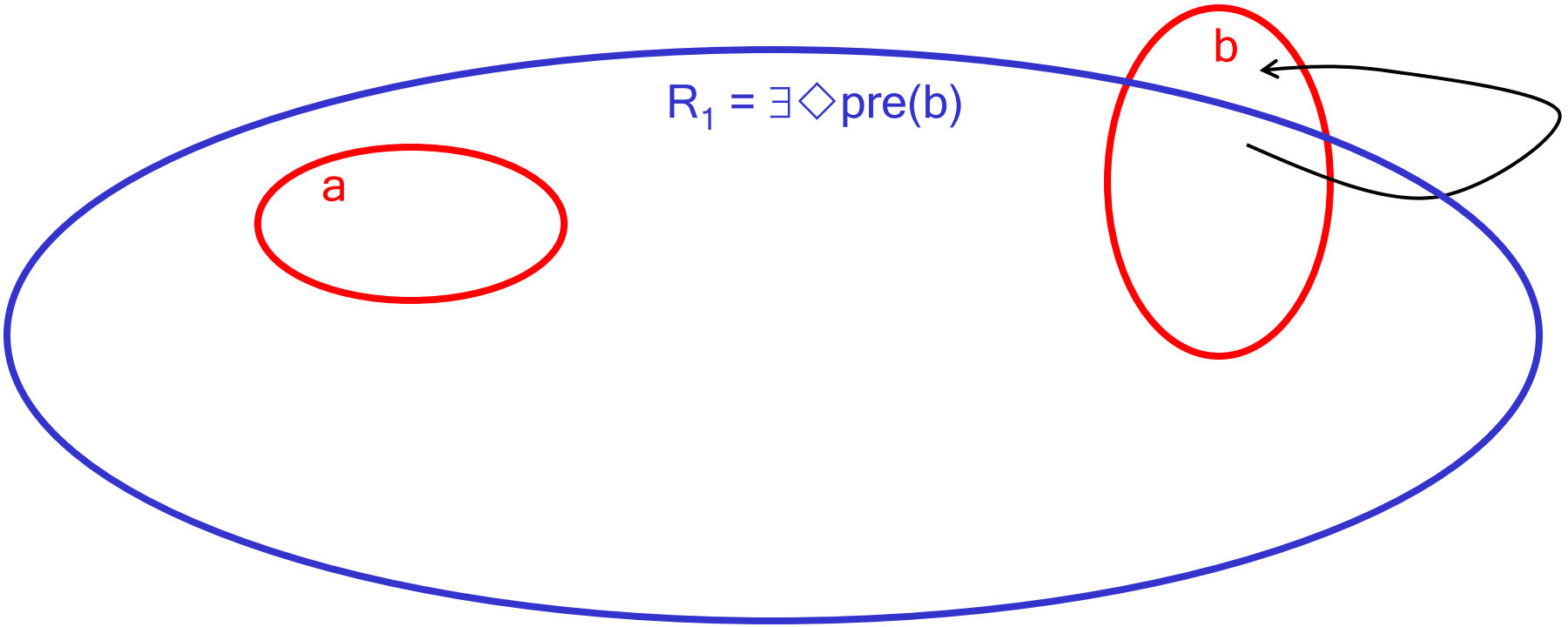
Given $a,b \in A$, is there an infinite trajectory from a that visits b infinitely often?



V2: Symbolic Repeated Reachability

$a \wedge \exists\Box\Diamond\, b$

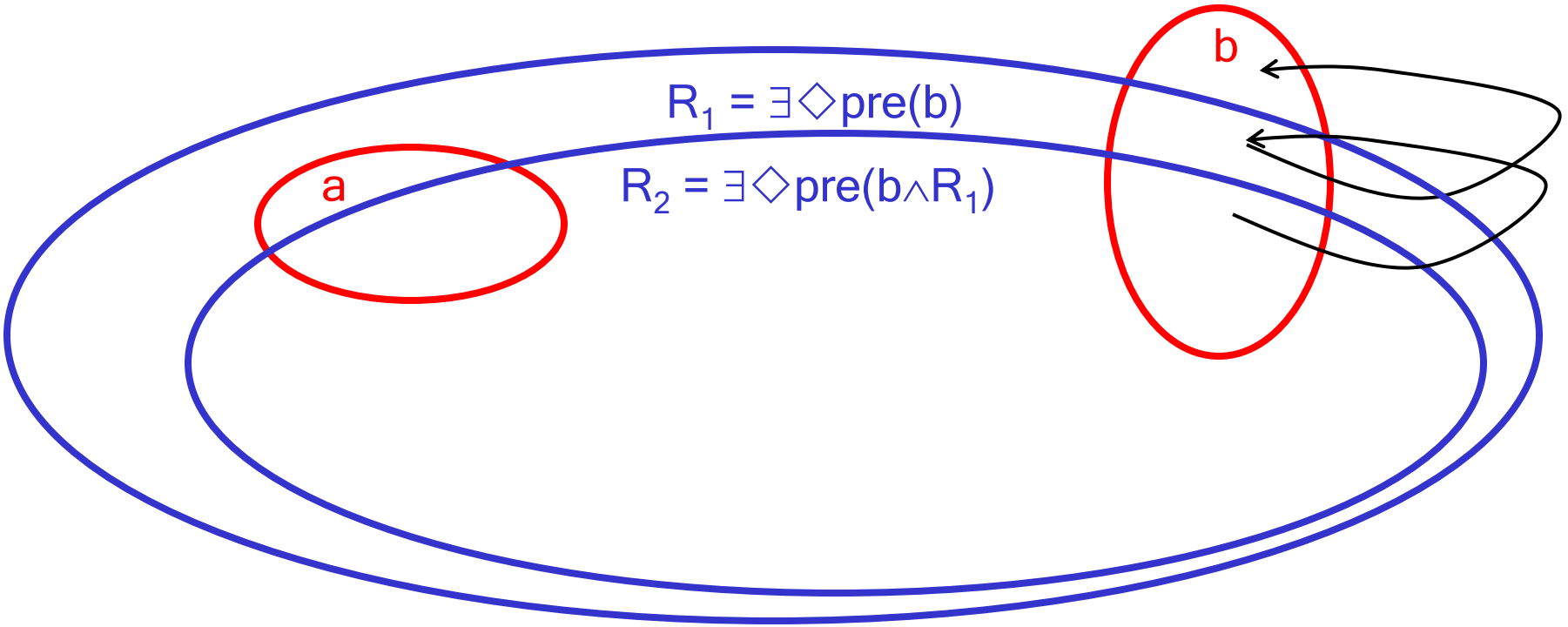
Given $a,b \in A$, is there an infinite trajectory from a that visits b infinitely often?



V2: Symbolic Repeated Reachability

$a \wedge \exists\Box\Diamond\, b$

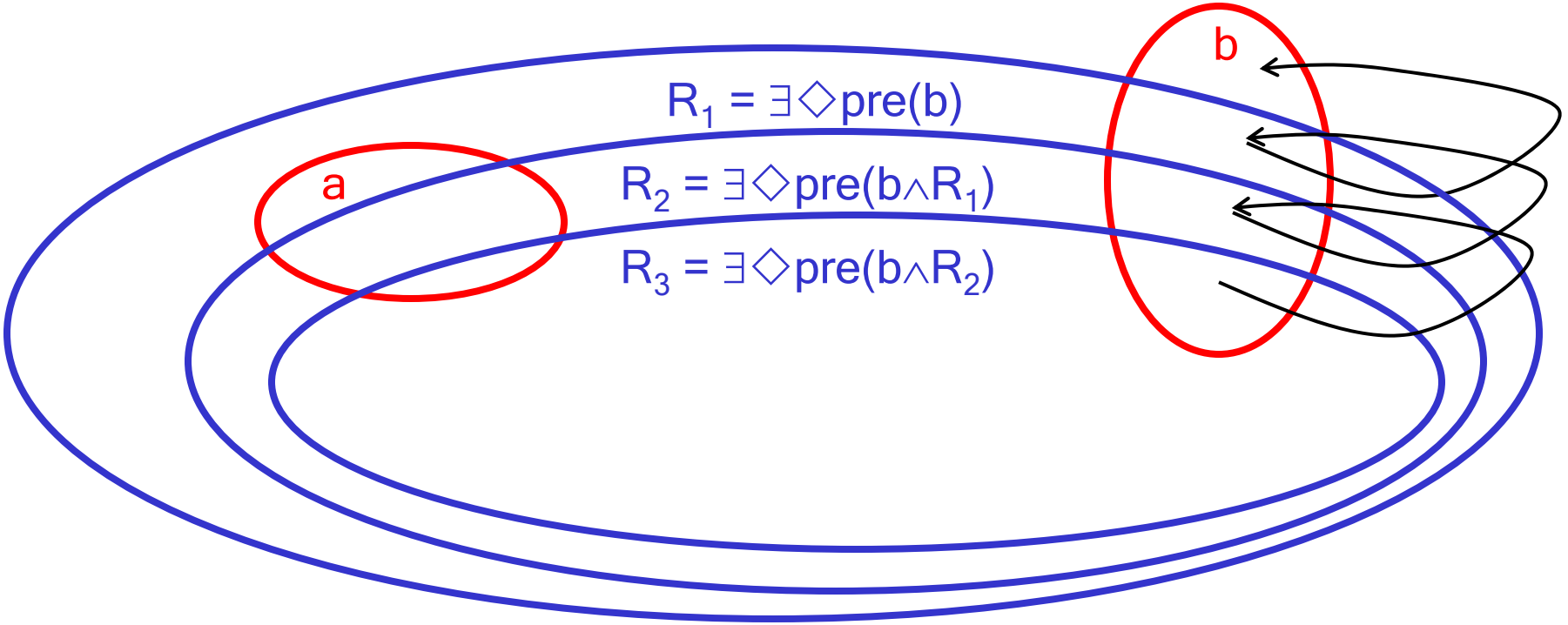
Given $a,b \in A$, is there an infinite trajectory from a that visits b infinitely often?



V2: Symbolic Repeated Reachability

$a \wedge \exists\Box\Diamond\, b$

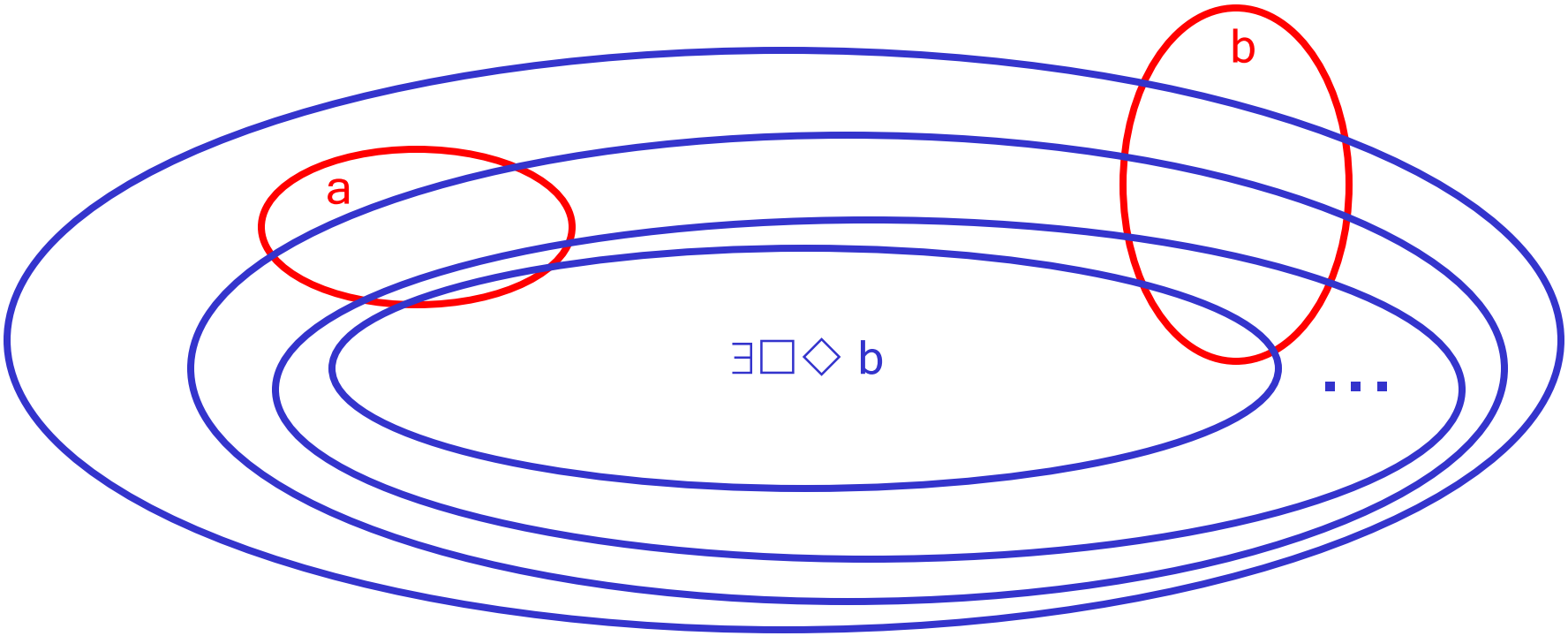
Given $a,b \in A$, is there an infinite trajectory from a that visits b infinitely often?



V2: Symbolic Repeated Reachability

$a \wedge \exists\Box\Diamond\, b$

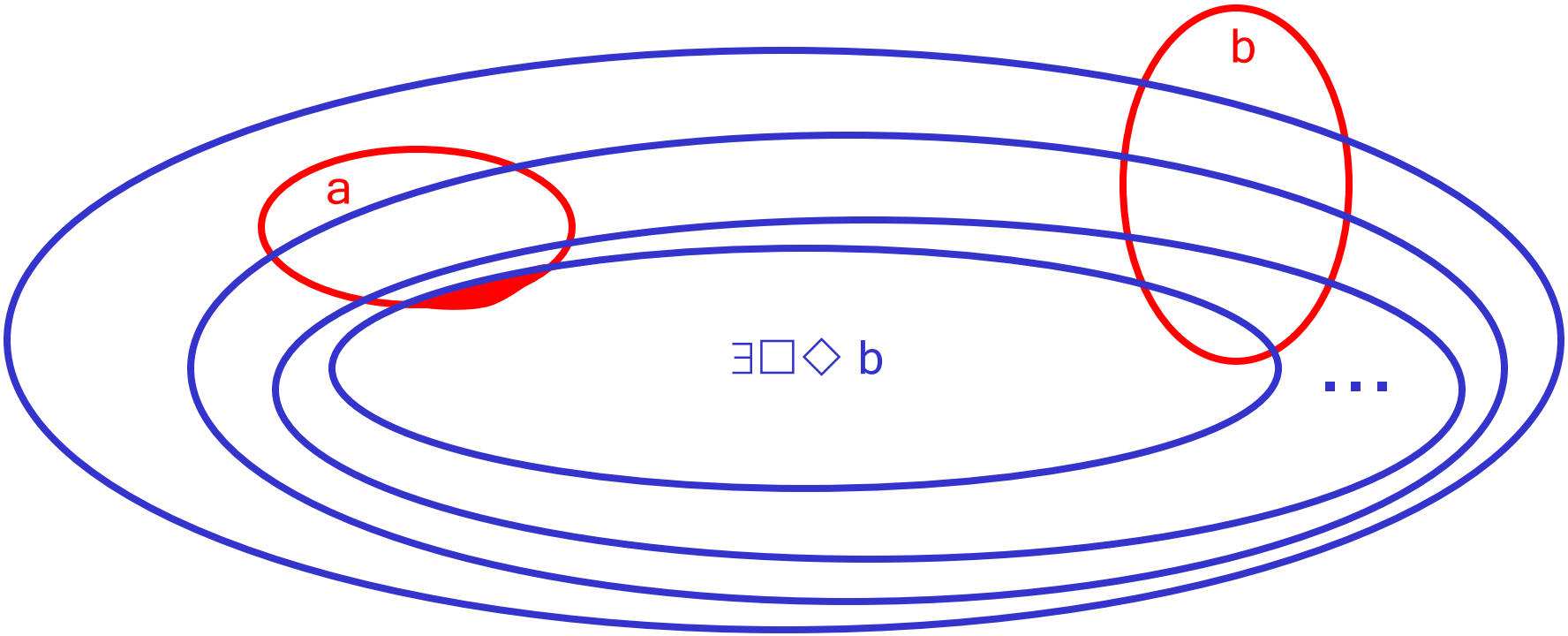
Given $a,b \in A$, is there an infinite trajectory from a that visits b infinitely often?



V2: Symbolic Repeated Reachability

$a \wedge \exists\Box\Diamond\, b$

Given $a,b \in A$, is there an infinite trajectory from a that visits b infinitely often?

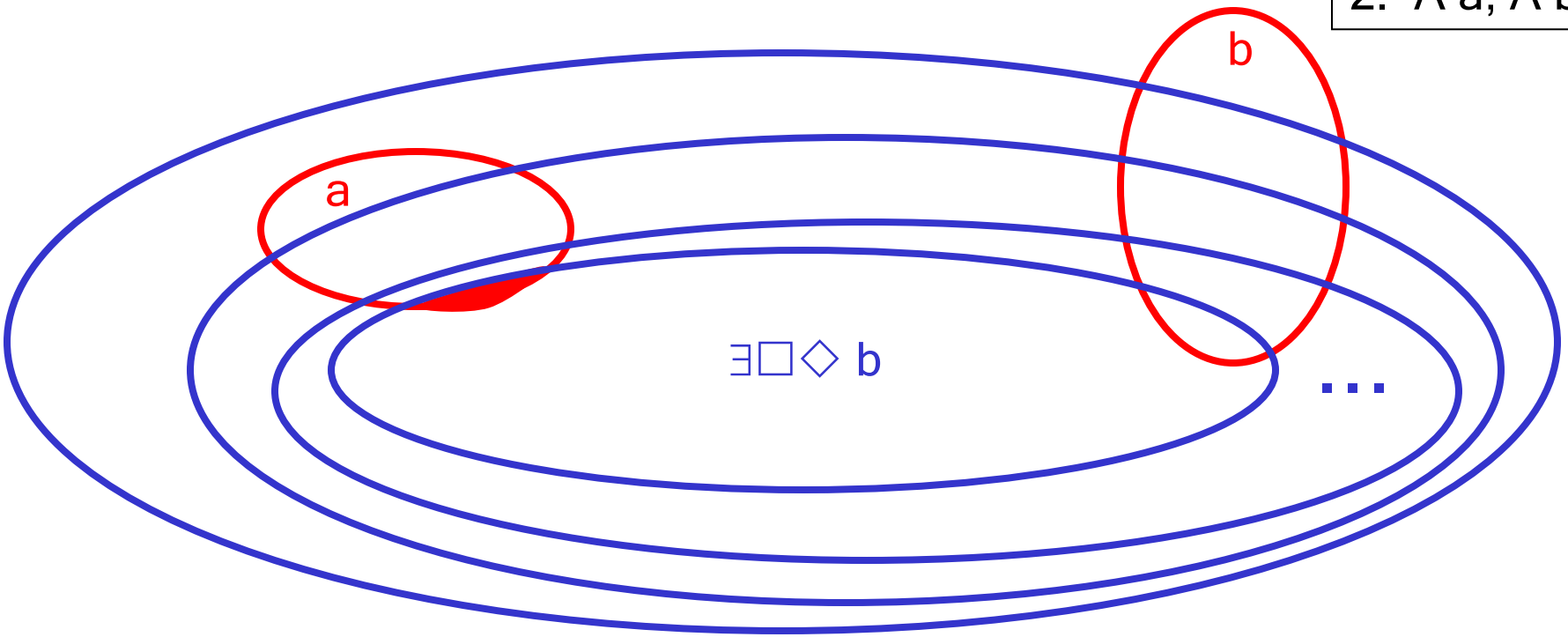


V2: Symbolic Repeated Reachability

$a \wedge \exists\Box\Diamond\, b$

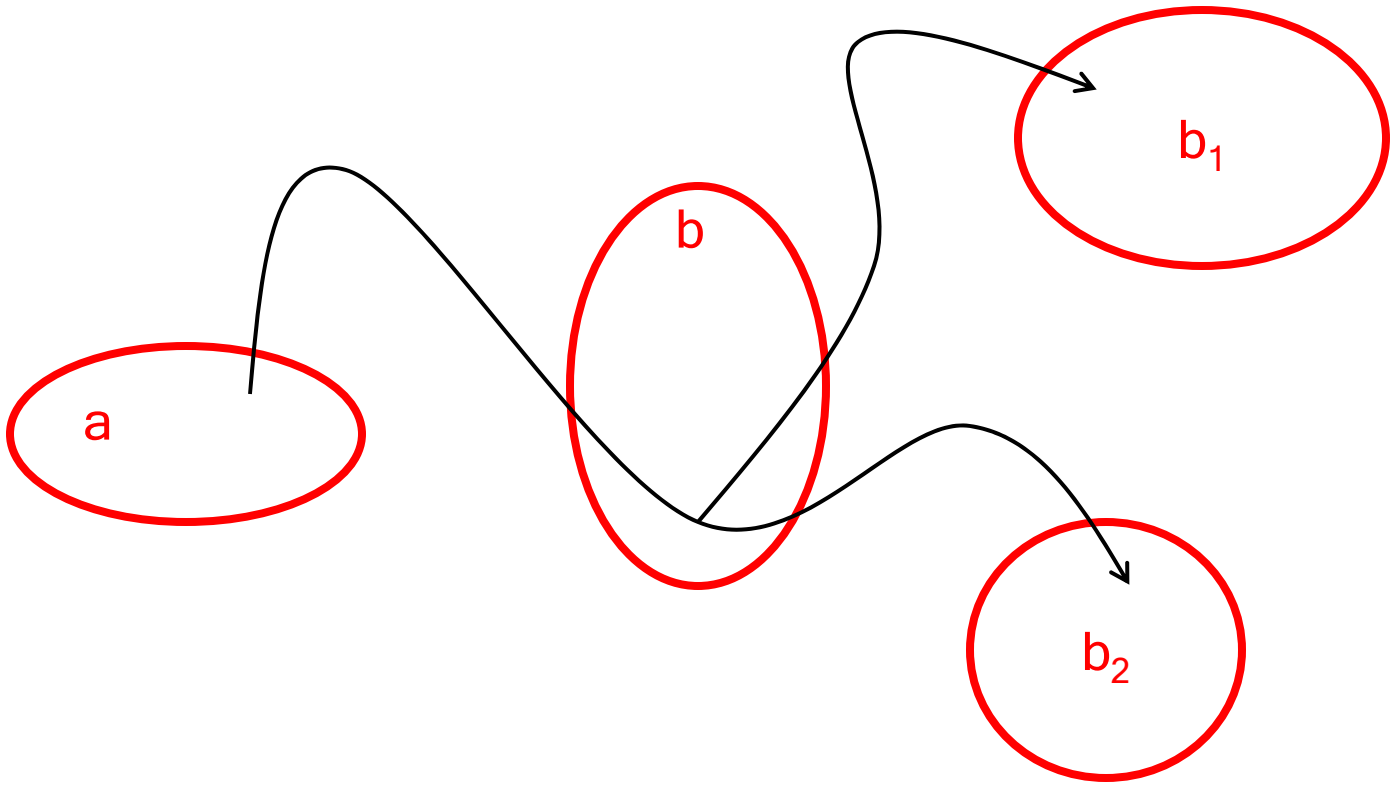
Given $a,b \in A$, is there an infinite trajectory from a that visits b infinitely often?

1. pre, [, $\subseteq$
2. Å a, Å b



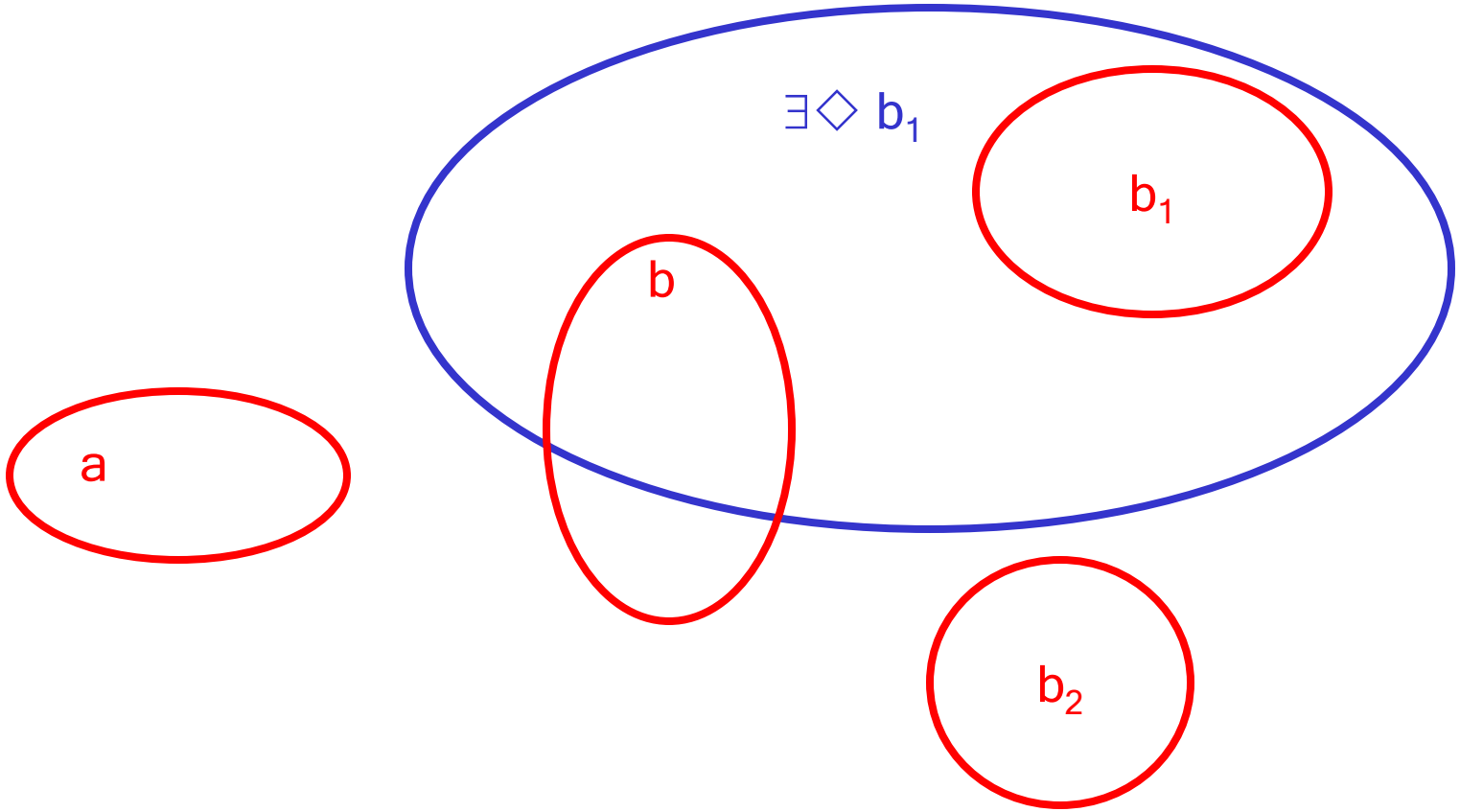
V3: Symbolic Nested Reachability

$a \wedge \exists\Diamond\ (b \wedge \exists\Diamond\ b_1 \wedge \exists\Diamond\ b_2)$



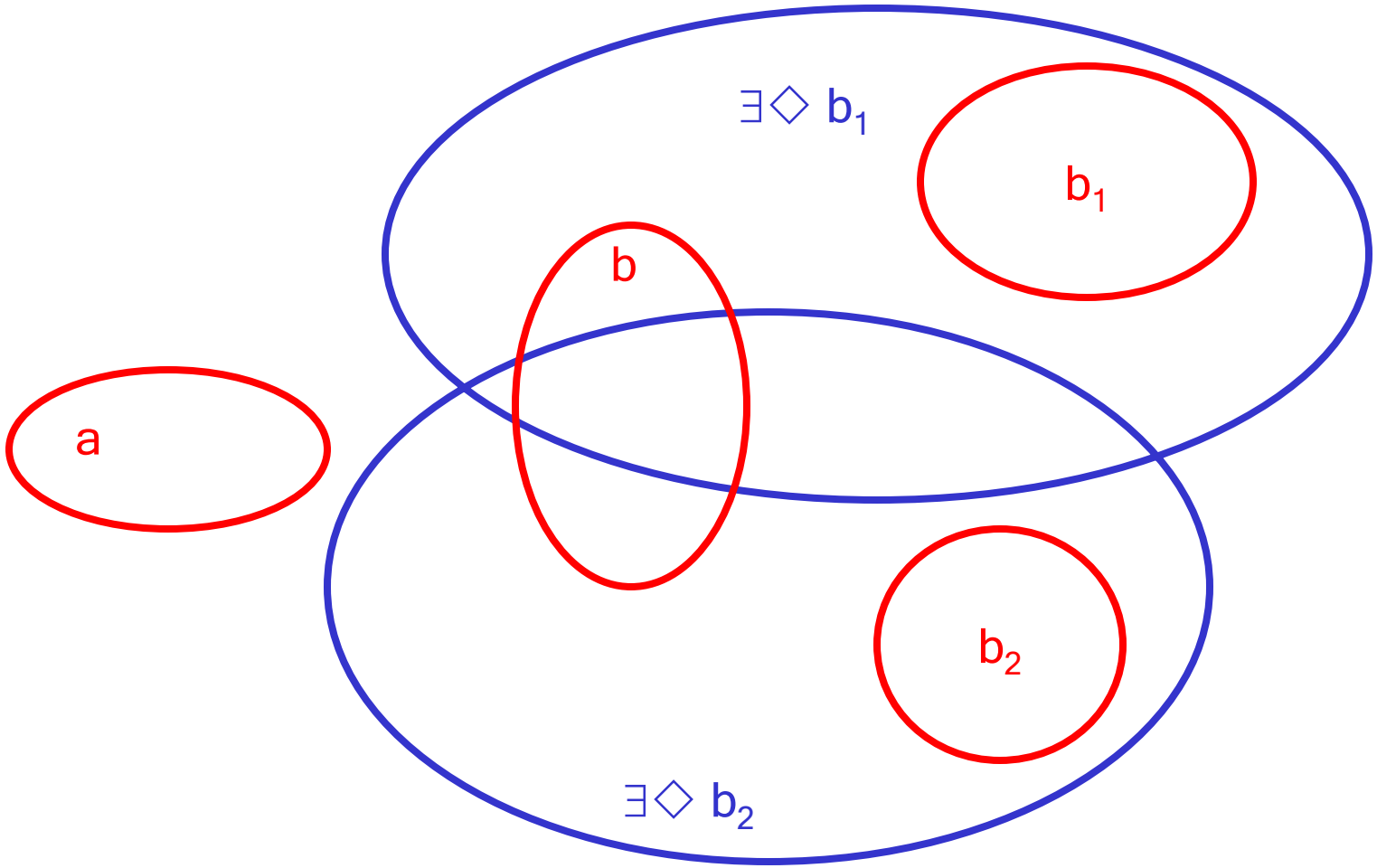
V3: Symbolic Nested Reachability

$a \wedge \exists\Diamond\ (b \wedge \exists\Diamond\ b_1 \wedge \exists\Diamond\ b_2)$



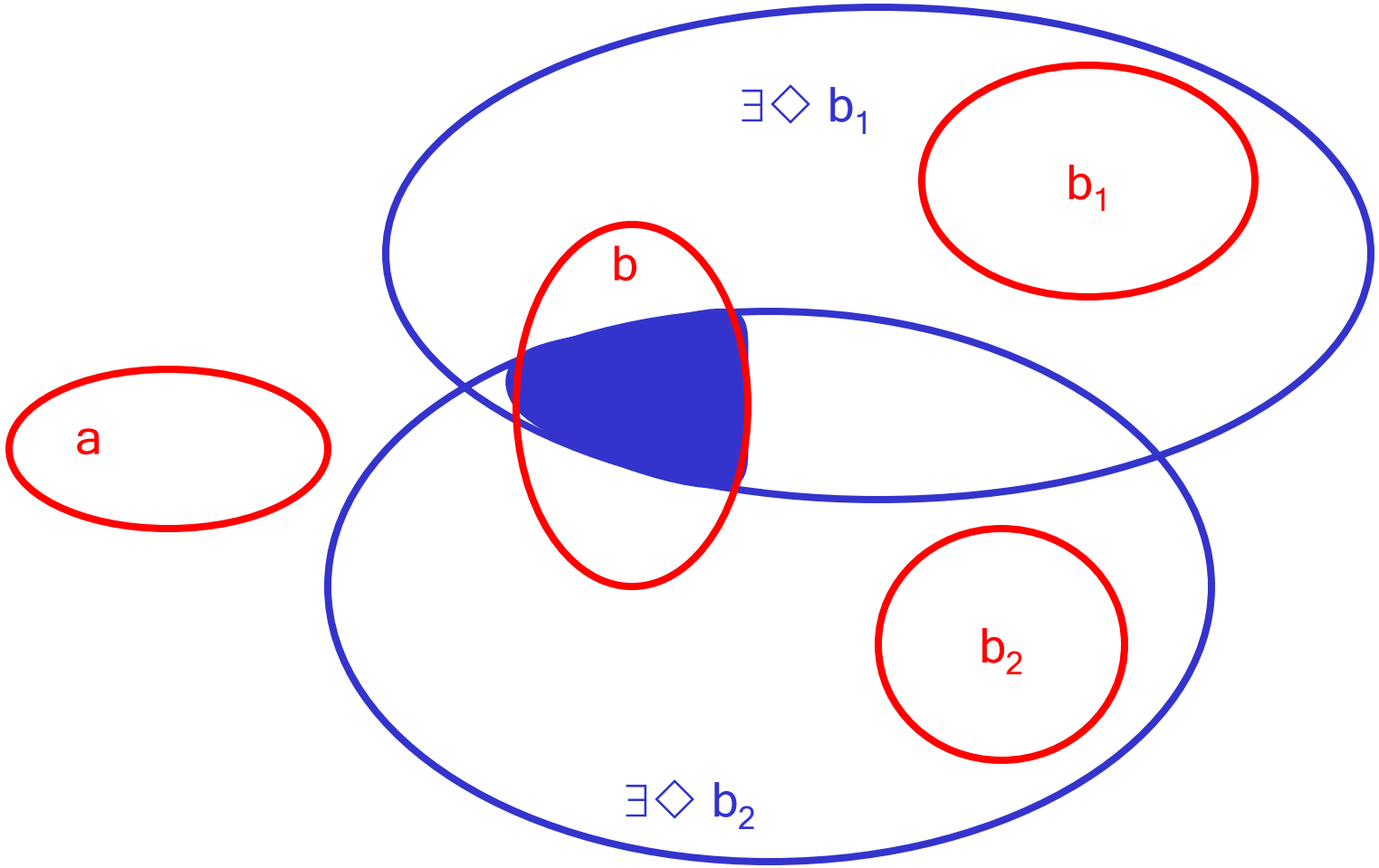
V3: Symbolic Nested Reachability

$a \wedge \exists\Diamond\ (b \wedge \exists\Diamond\ b_1 \wedge \exists\Diamond\ b_2)$



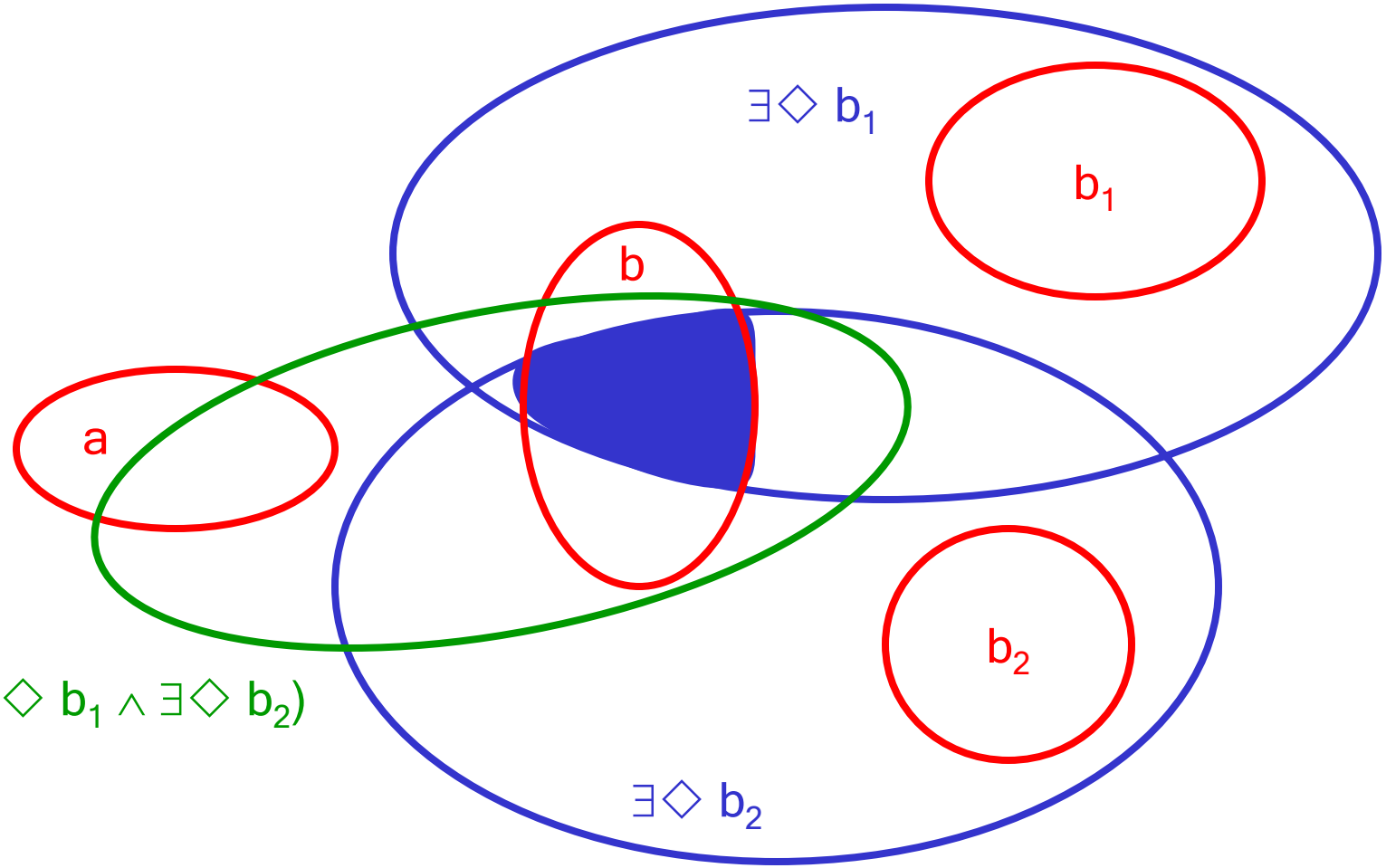
V3: Symbolic Nested Reachability

$a \wedge \exists\Diamond\ (b \wedge \exists\Diamond\ b_1 \wedge \exists\Diamond\ b_2)$



V3: Symbolic Nested Reachability

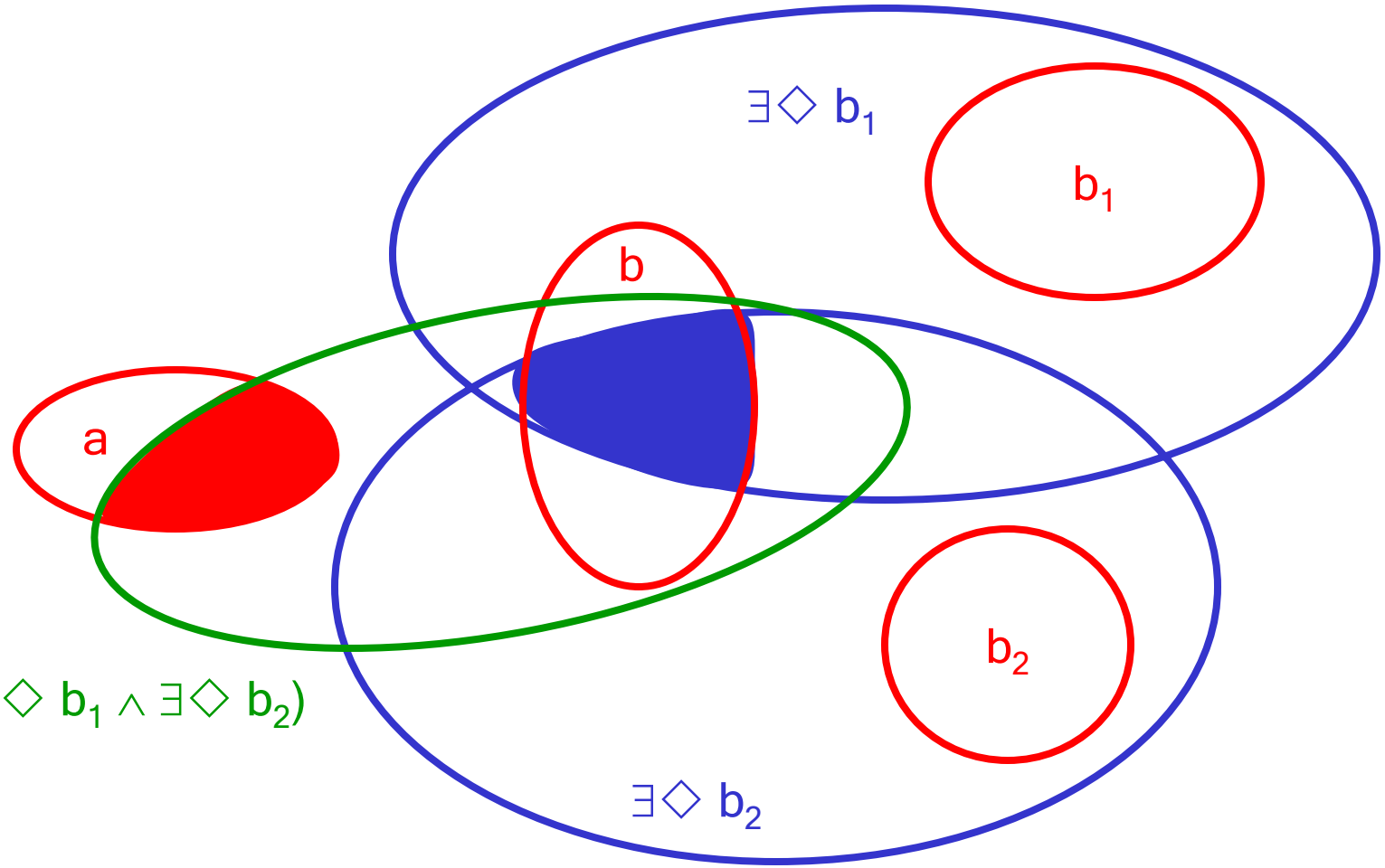
$a \wedge \exists\Diamond\ (b \wedge \exists\Diamond\ b_1 \wedge \exists\Diamond\ b_2)$



$\exists\Diamond\ (b \wedge \exists\Diamond\ b_1 \wedge \exists\Diamond\ b_2)$

V3: Symbolic Nested Reachability

$a \wedge \exists\Diamond\ (b \wedge \exists\Diamond\ b_1 \wedge \exists\Diamond\ b_2)$

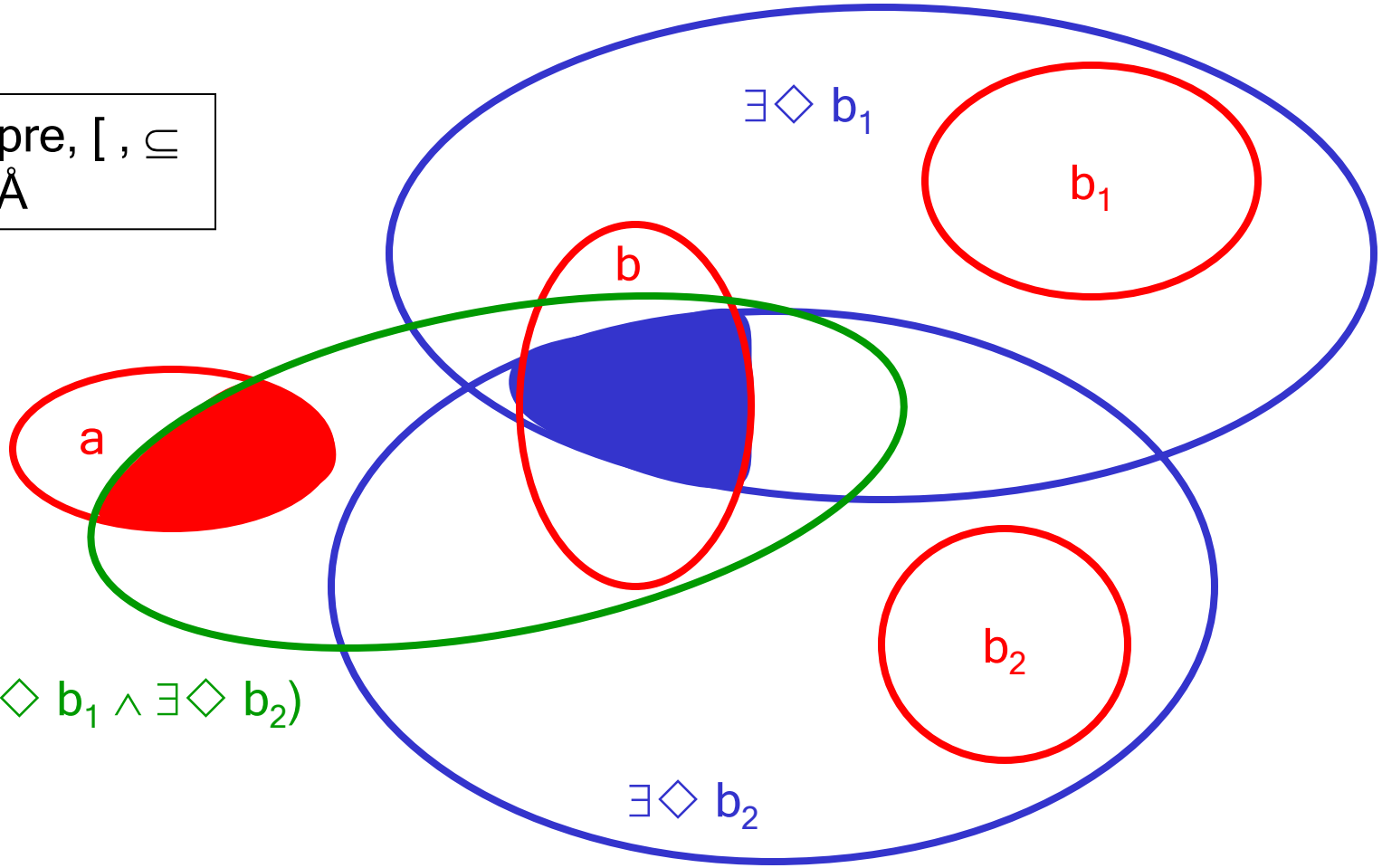


$\exists\Diamond\ (b \wedge \exists\Diamond\ b_1 \wedge \exists\Diamond\ b_2)$

V3: Symbolic Nested Reachability

$a \wedge \exists\Diamond\ (b \wedge \exists\Diamond\ b_1 \wedge \exists\Diamond\ b_2)$

1. pre, [, $\subseteq$
2. Å

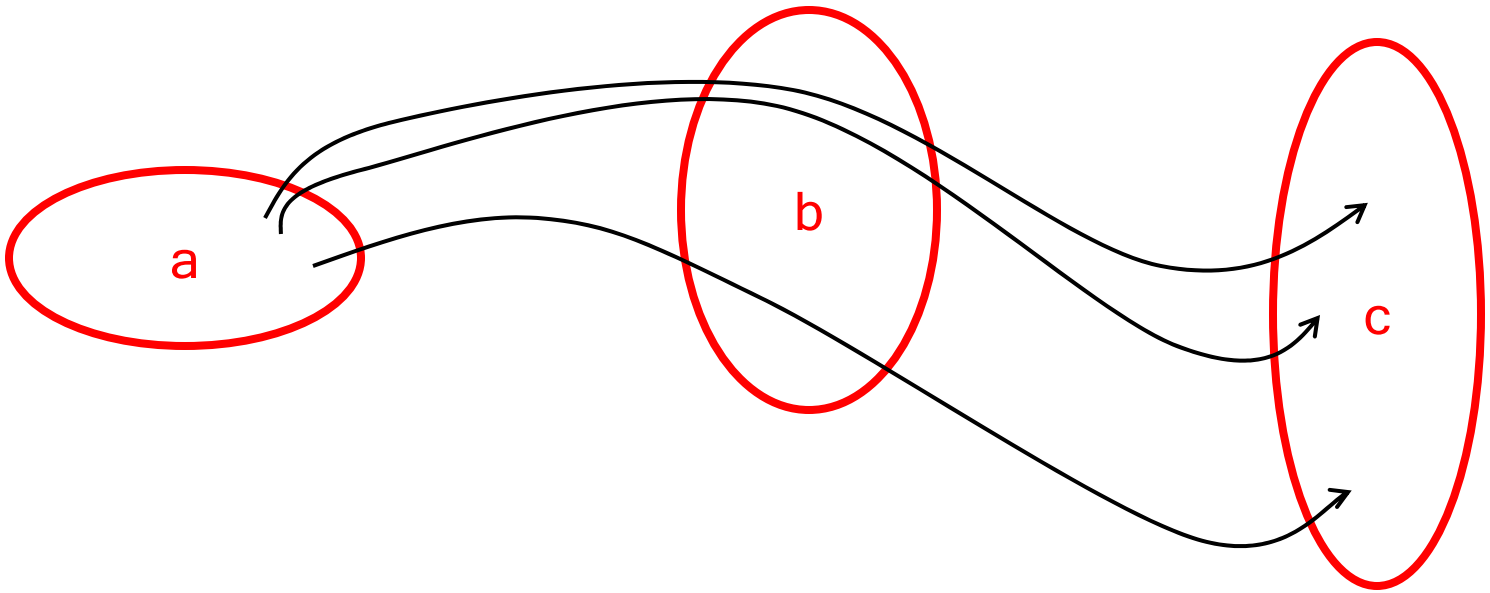


$\exists\Diamond\ (b \wedge \exists\Diamond\ b_1 \wedge \exists\Diamond\ b_2)$

V4: Symbolic Negated Reachability

$a \wedge \forall\Box\, (b \rightarrow \exists\Diamond\, c)$

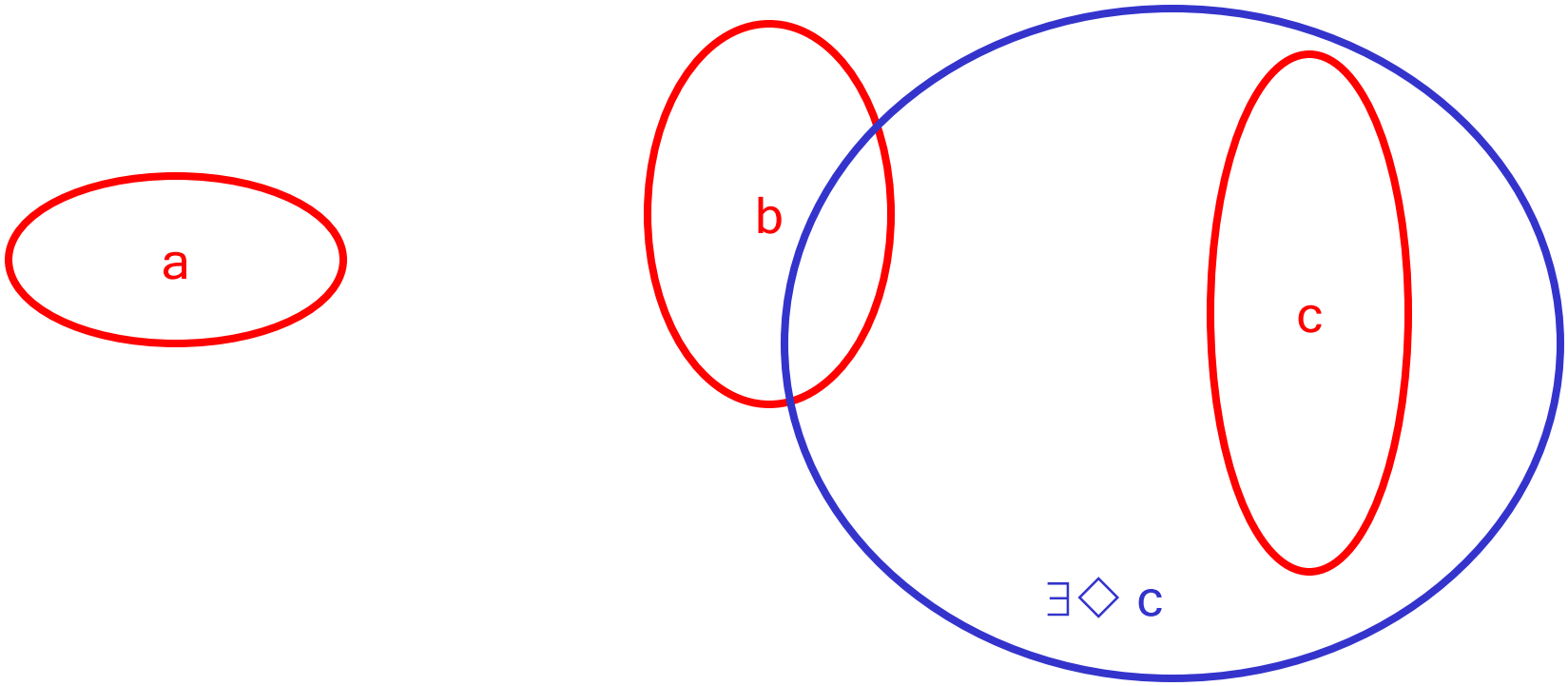
Given a,b,c$\in$A, can every trajectory from a to b be extended to c?



V4: Symbolic Negated Reachability

$a \wedge \forall\Box\,(b \rightarrow \exists\Diamond\, c)$

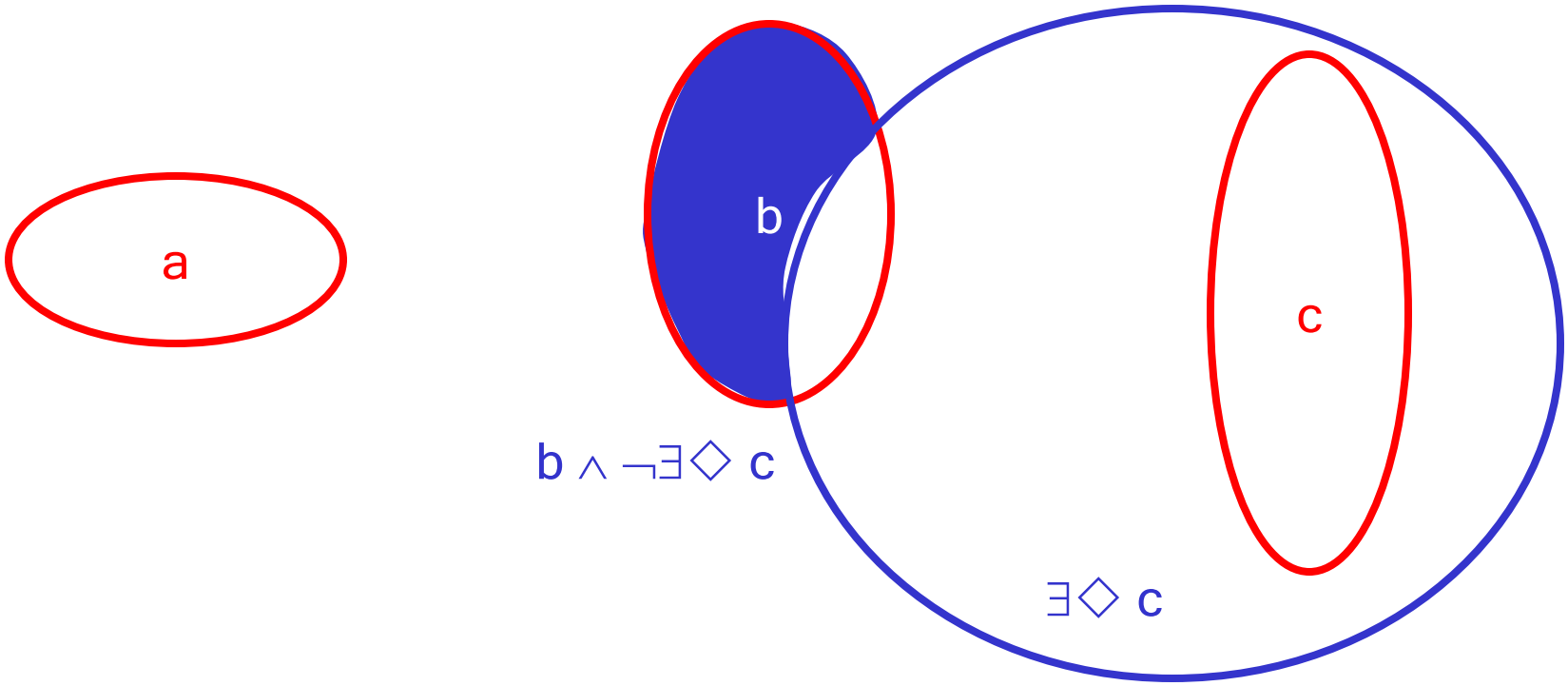
Given a,b,c$\in$A, can every trajectory from a to b be extended to c?



V4: Symbolic Negated Reachability

$a \wedge \forall\Box\, (b \rightarrow \exists\Diamond\, c)$

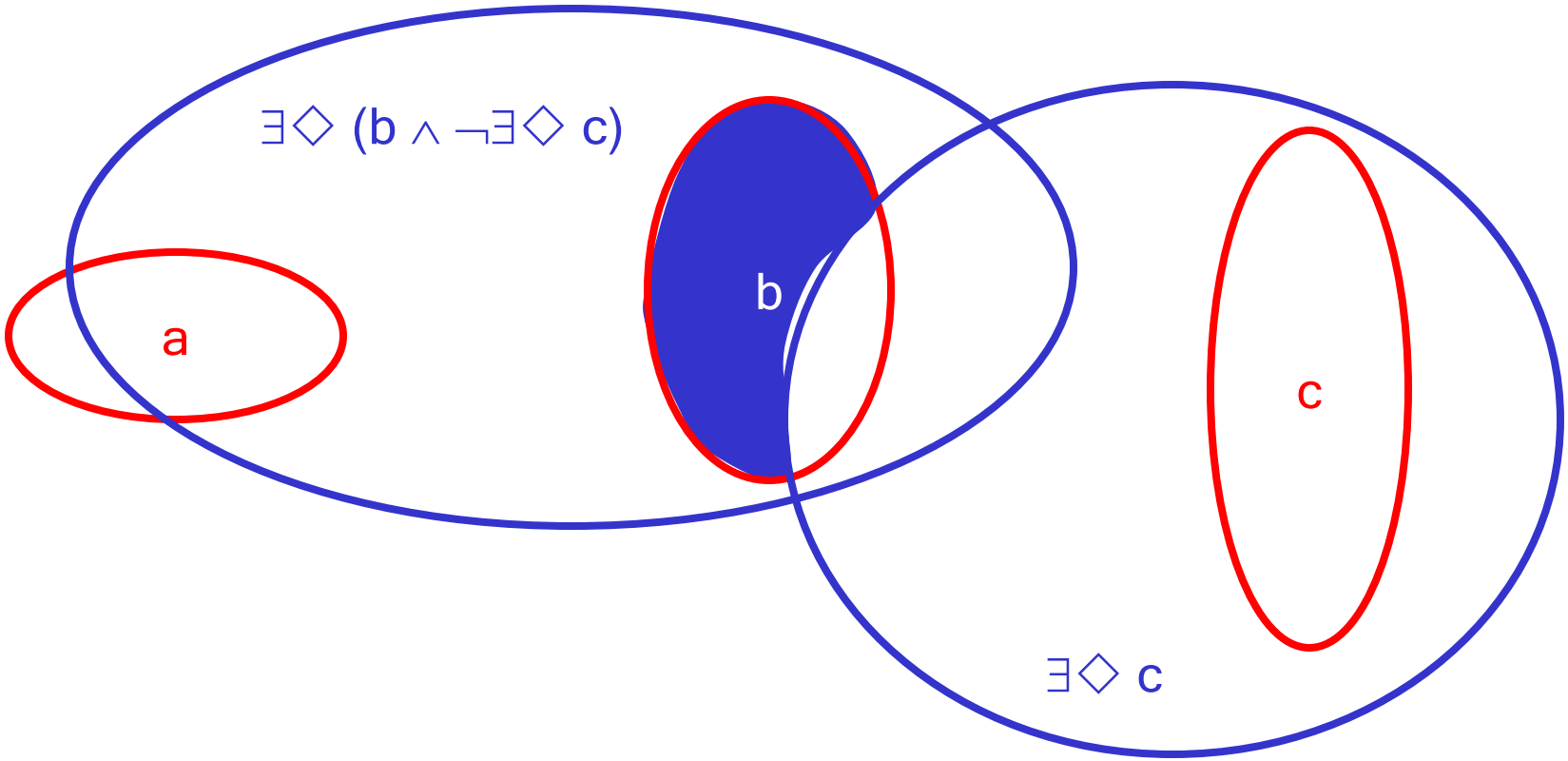
Given a,b,c∈A, can every trajectory from a to b be extended to c?



V4: Symbolic Negated Reachability

$a \wedge \forall\Box\, (b \rightarrow \exists\Diamond\, c)$

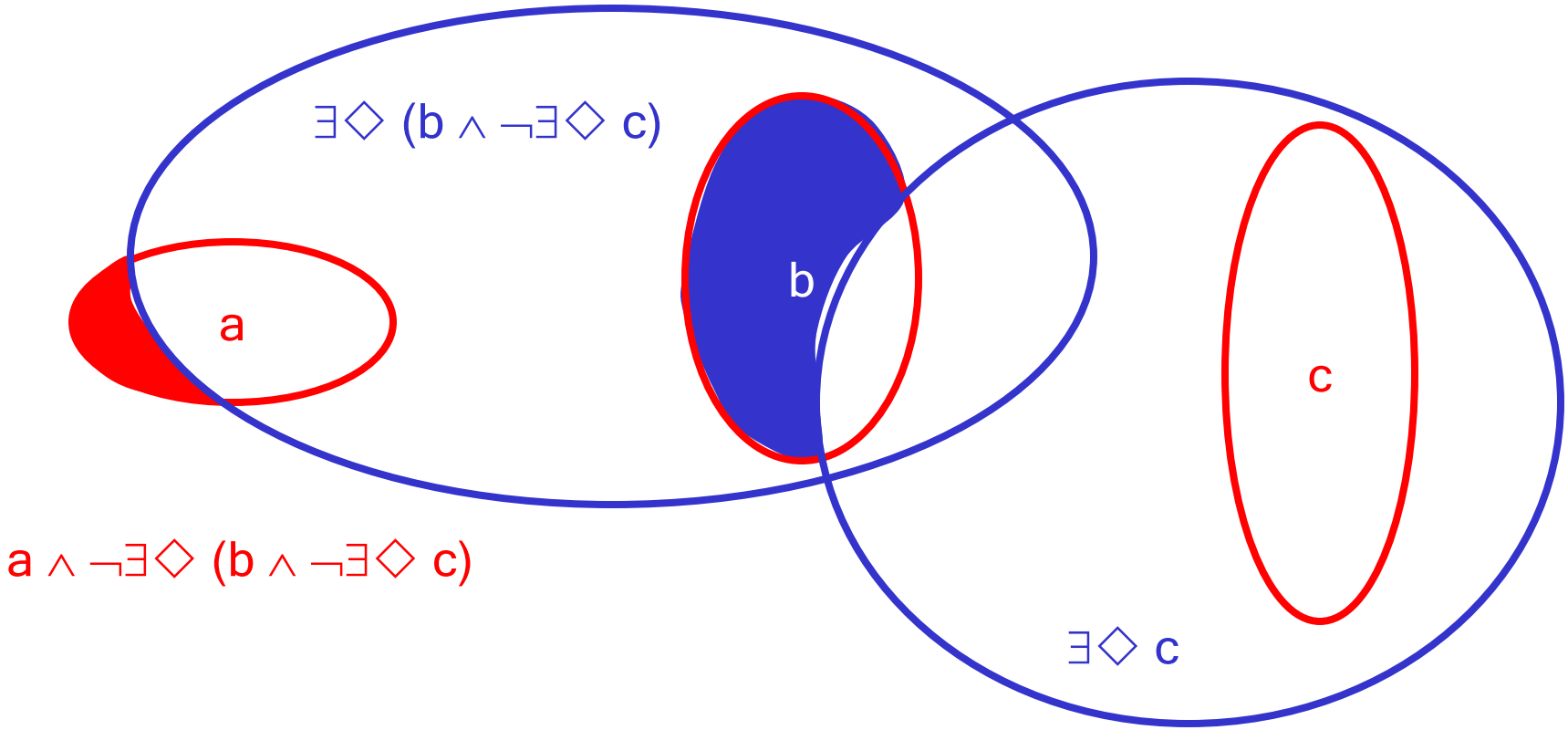
Given a,b,c∈A, can every trajectory from a to b be extended to c?



V4: Symbolic Negated Reachability

$a \wedge \forall\Box\, (b \rightarrow \exists\Diamond\, c)$

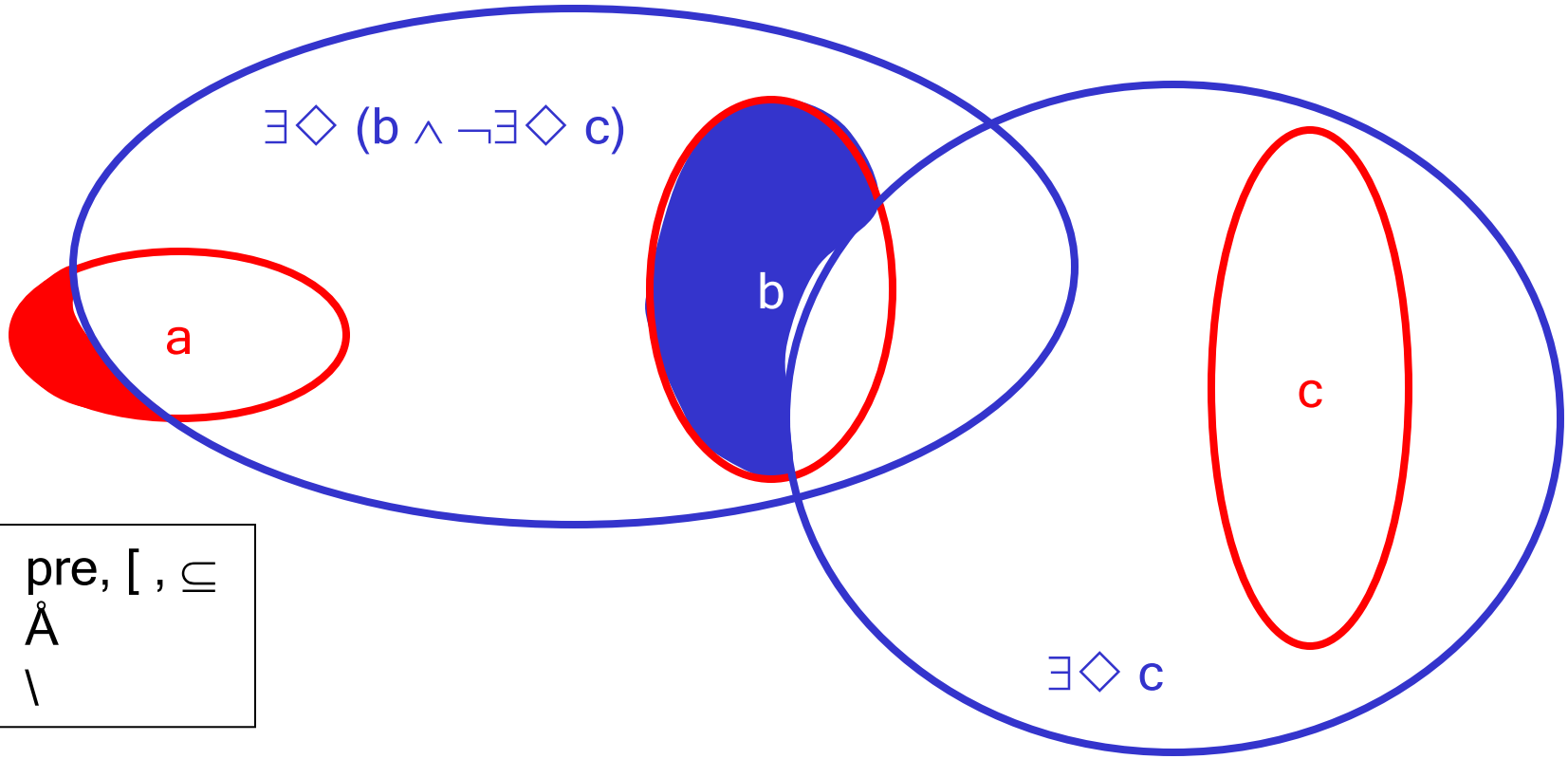
Given a,b,c∈A, can every trajectory from a to b be extended to c?



V4: Symbolic Negated Reachability

$a \wedge \forall\Box\,(b \rightarrow \exists\Diamond\, c)$

Given a,b,c∈A, can every trajectory from a to b be extended to c?



1. pre, [, ⊆
2. Å
3. \

Four Symbolic Semi-Algorithms

A1: Close A under pre

$\mathfrak{I}_0 := A$
for i=1,2,3,… do
$\mathfrak{I}_i := \mathfrak{I}_{i-1} [\{ pre(R) \mid R \in \mathfrak{I}_i \}$

until $\mathfrak{I}_i = \mathfrak{I}_{i-1}$

Four Symbolic Semi-Algorithms

A1: Close A under pre

A2: Close A under pre, Å a

$\mathfrak{I}_0$:= A
for i=1,2,3,… do
$\mathfrak{I}_i := \mathfrak{I}_{i-1} [\{ pre(R) \mid R \in \mathfrak{I}_i \}$
$[\{ R Å a \mid R \in \mathfrak{I}_i Æ a2A \}$
until
$\mathfrak{I}_i = \mathfrak{I}_{i-1}$

Four Symbolic Semi-Algorithms

A1: Close A under pre

A2: Close A under pre, Å a

A3: Close A under pre, Å

$\mathfrak{I}_0 := A$
for i=1,2,3,… do
$\mathfrak{I}_i := \mathfrak{I}_{i-1} [\{ pre(R) \mid R \in \mathfrak{I}_i \}$
$[\{ R_1 Å R_2 \mid R_1, R_2 \in \mathfrak{I}_i \}$

until $\mathfrak{I}_i = \mathfrak{I}_{i-1}$

Four Symbolic Semi-Algorithms

A1: Close A under pre

A2: Close A under pre, Å a

A3: Close A under pre, Å

A4: Close A under pre, Å , \

$\mathfrak{I}_0$:= A
for i=1,2,3,… do
$\mathfrak{I}_i := \mathfrak{I}_{i-1} [\{ pre(R) \mid R \in \mathfrak{I}_i \}$
$[\{ R_1 Å R_2 \mid R_1, R_2 \in \mathfrak{I}_i \}$
$[\{ R_1 \setminus R_2 \mid R_1, R_2 \in \mathfrak{I}_i \}$
until $\mathfrak{I}_i = \mathfrak{I}_{i-1}$

Four Symbolic Semi-Algorithms

A1: Close A under pre

A2: Close A under pre, Å a

A3: Close A under pre, Å

A4: Close A under pre, Å , \

A_k terminates $(1 \leq k \leq 4) \Rightarrow$
symbolic model checking of V_k terminates.

Four State Equivalences

E1: Reach Equivalence

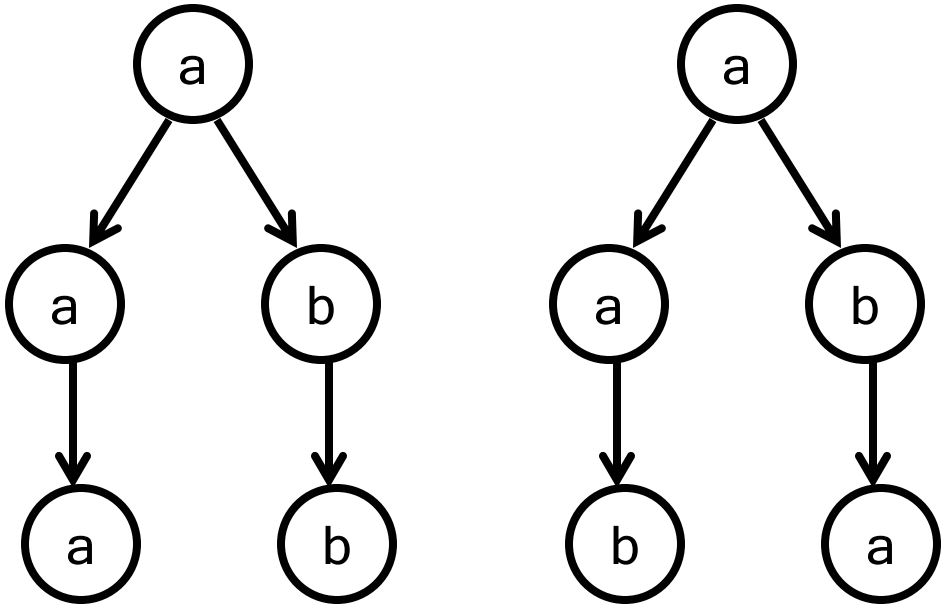
$q_1 \cong_1 q_2$ iff if $a \in A$ can be reached from q_1 in d steps, then a can be reached from q_2 in d steps, and vice versa.

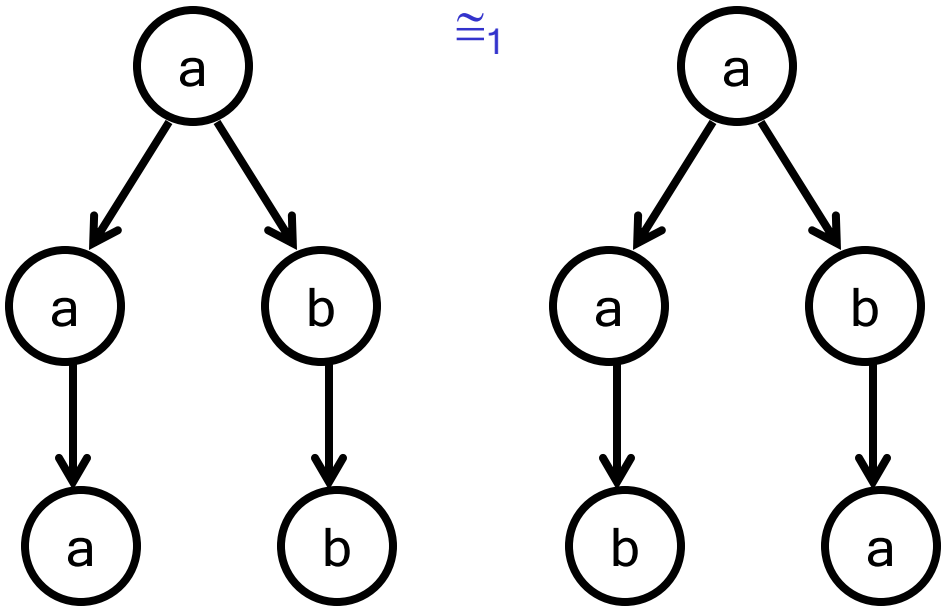
Four State Equivalences

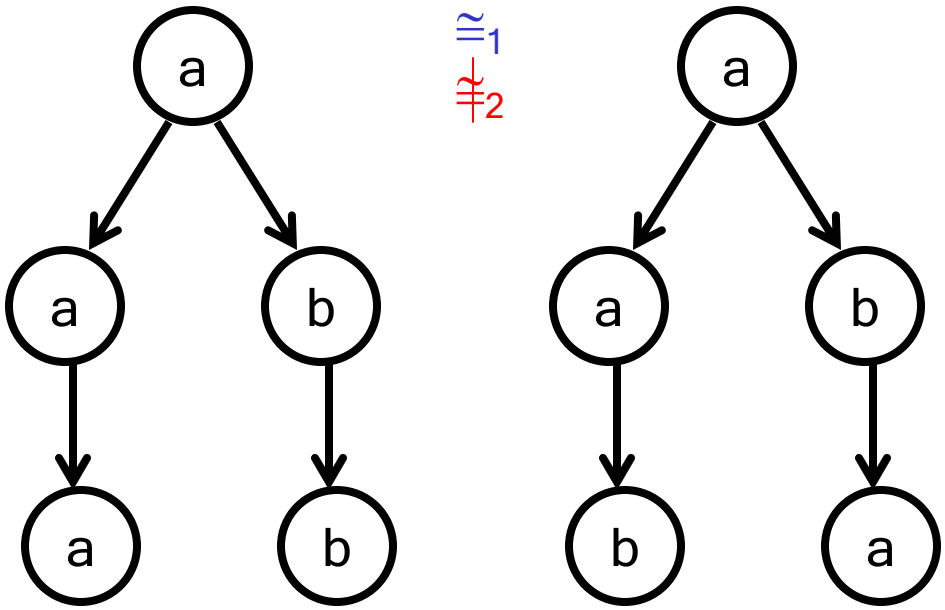
E1: Reach Equivalence

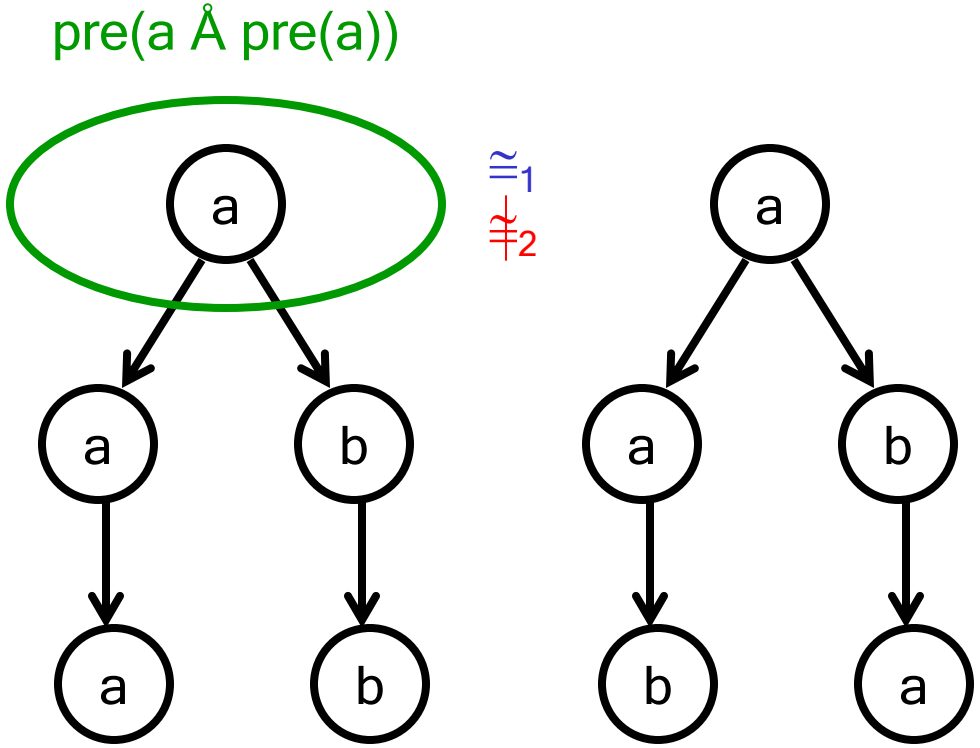
E2: Trace Equivalence

$q_1 \cong_2 q_2$ iff if every finite trace from q_1 is a finite trace from q_2, and vice versa.


a
a
b
a
b
a
a
b
b
a


a
a
b
a
b
≅1
a
a
b
b
a


a
a
b
a
b
≅1
≇2
a
a
b
b
a


pre(a Å pre(a))
a
a
b
a
b
≅1
≇2
a
a
b
b
a

Four State Equivalences

E1: Reach Equivalence

E2: Trace Equivalence

E3: Similarity

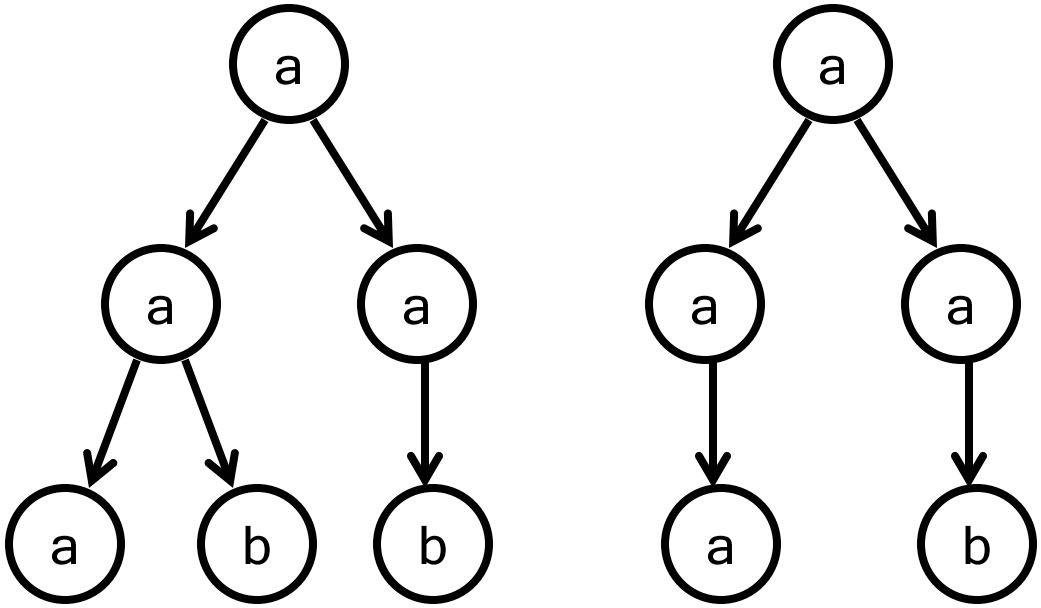
$q_1 \cong_3 q_2$ iff if q_1 simulates q_2, and vice versa.

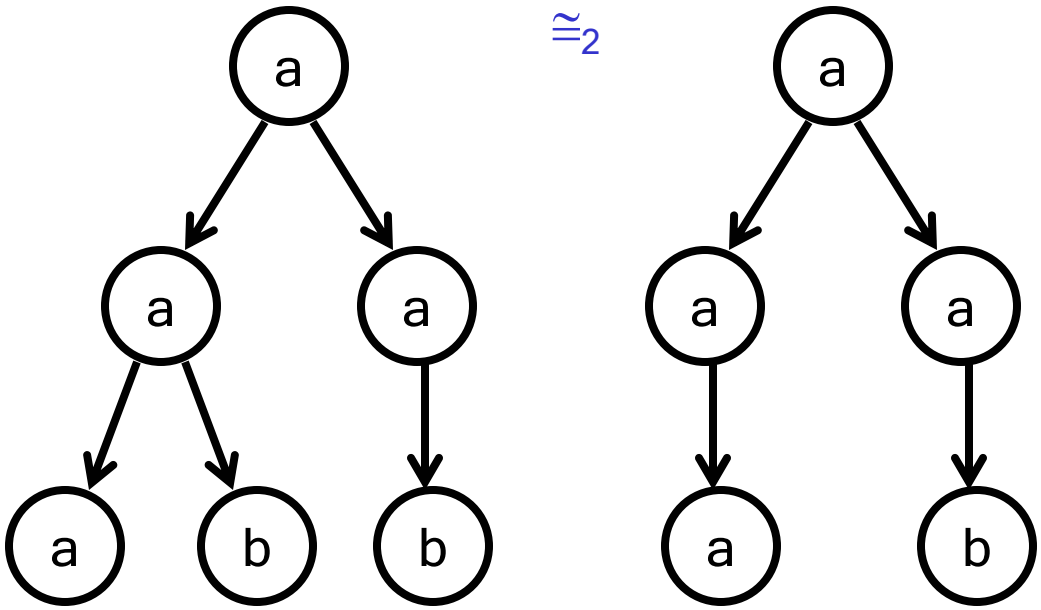
q_1 is simulated by q_2

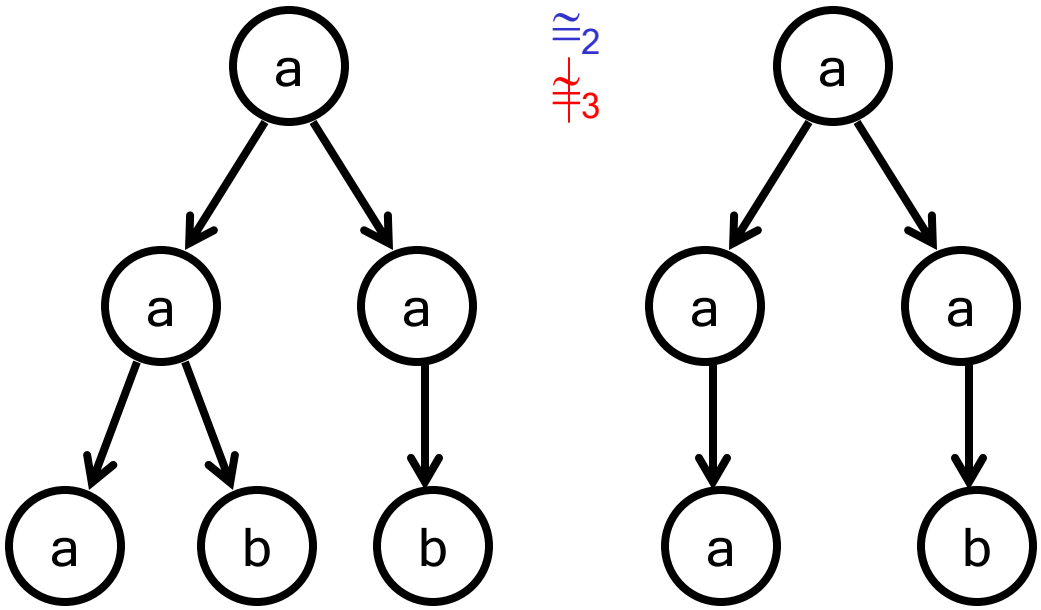
iff

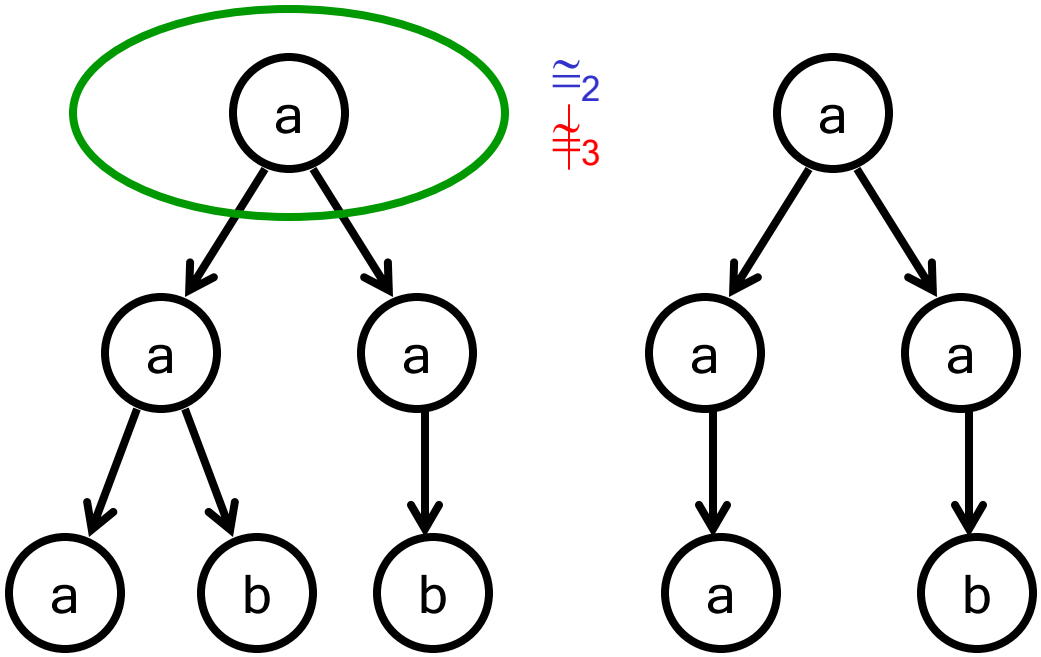
there is a simulation relation S such that

1. $S(q_1,q_2)$
2. if S(p,q) then
   a. (8 a2A) (p2 a iff q2 a)
   b. (8 p') (if p2 pre(p') then
      (9 q') (q2 pre(q') Æ S(p',q')))


a
a
a
a
b
b
a
a
a
a
b


a
a
a
a
b
b
≅2
a
a
a
a
b


a
a
a
a
b
b
≅2
≇3
a
a
a
a
b


pre(pre(a) Å pre(b))
≅2
≇3
a
a
a
a
b
b
a
a
a
a
b

Four State Equivalences

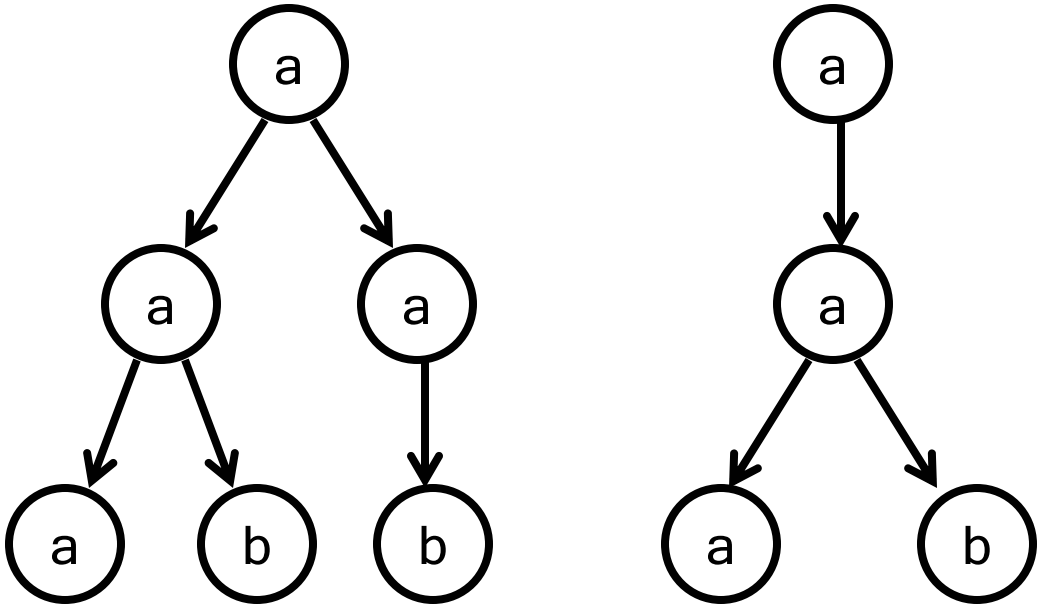
E1: Reach Equivalence

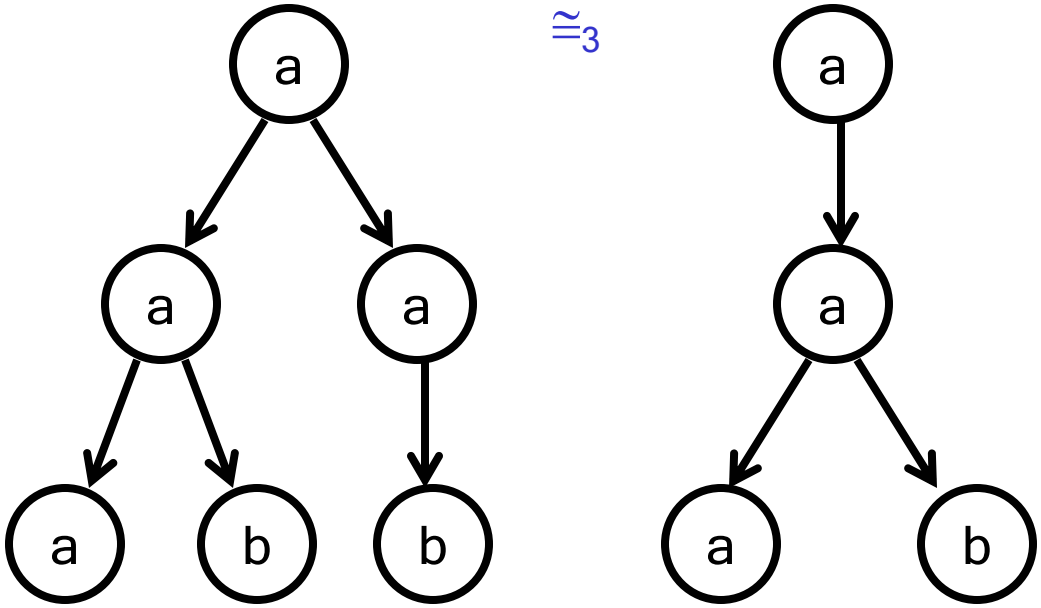
E2: Trace Equivalence

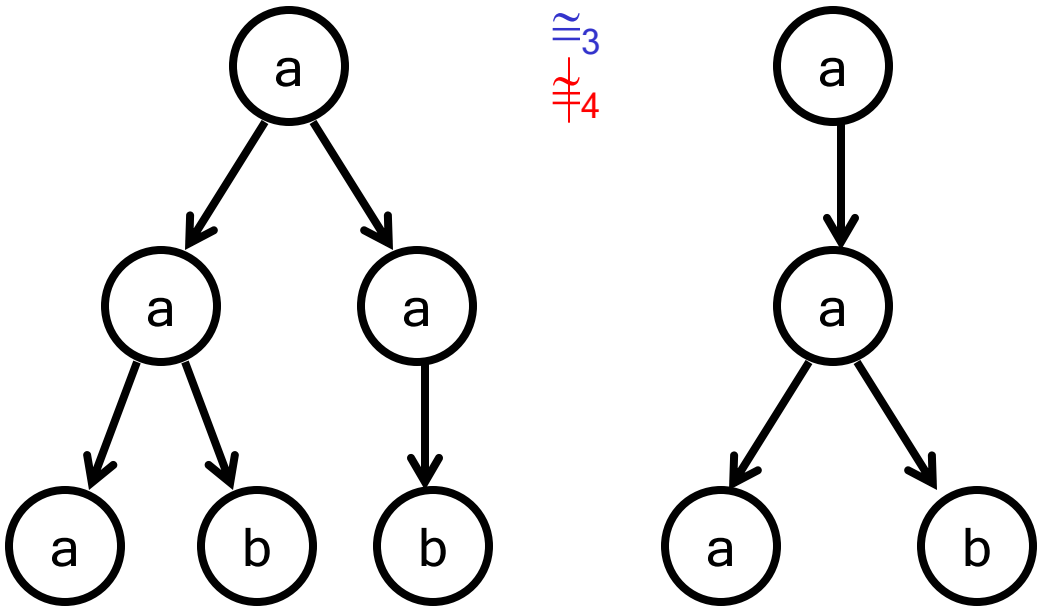
E3: Similarity

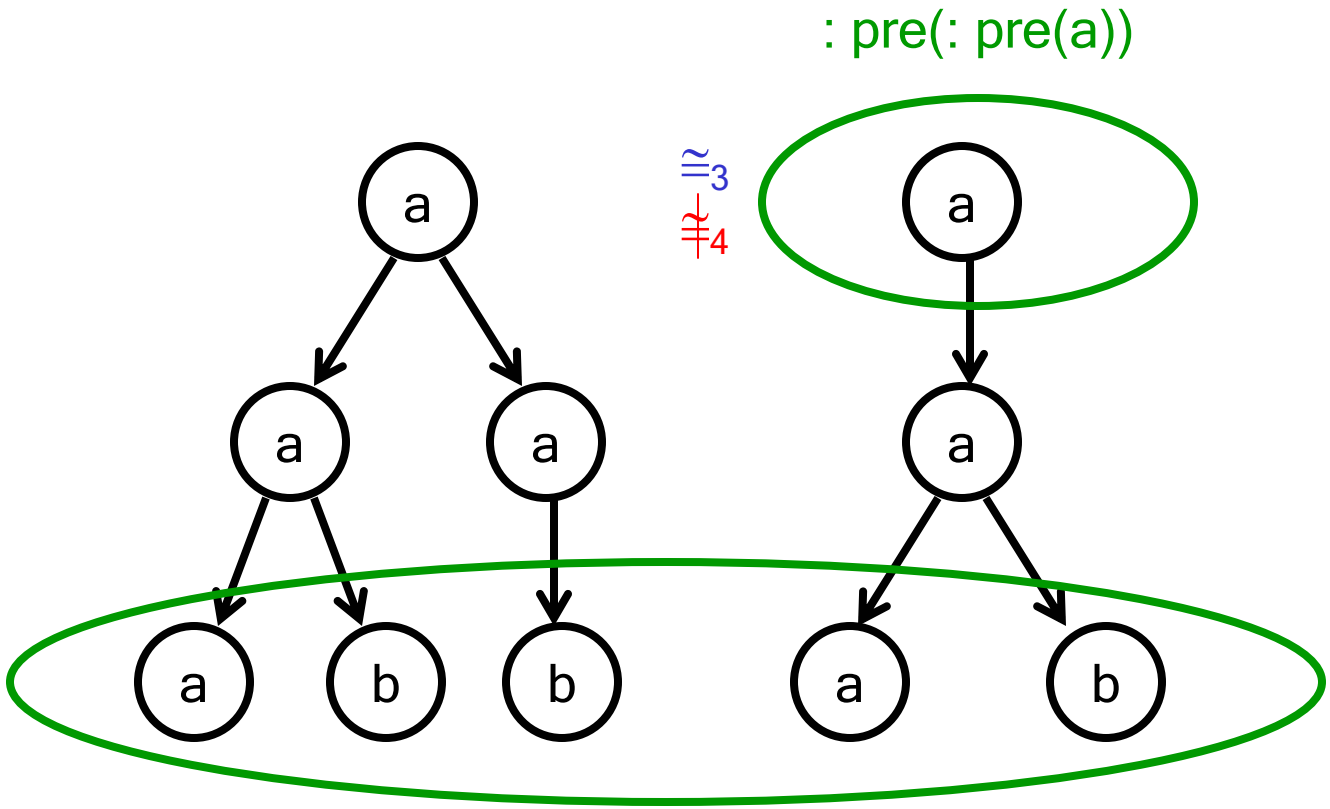
E4: Bisimilarity

$q_1 \cong_4 q_2$ iff if q_1 simulates q_2 via a symmetric simulation relation (this is called a bisimulation relation).


a
a
a
a
b
b
a
a
a
b


a
a
a
a
b
b
≅3
a
a
a
b


a
a
a
a
b
b
≅3
≇4
a
a
a
b


: pre(: pre(a))
≅3
≇4
a
a
a
a
a
a
b
b
a
b

Specification Logics:

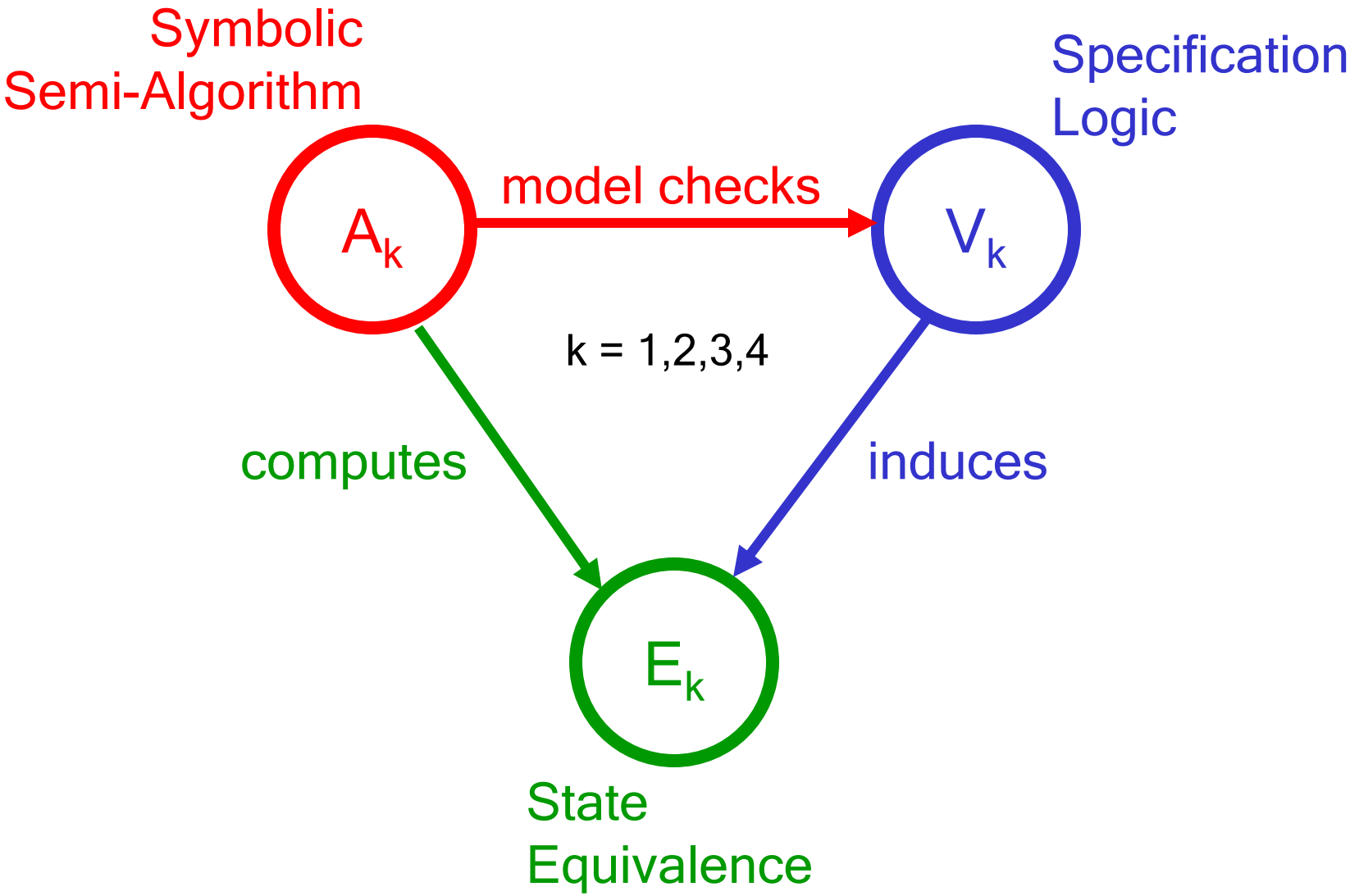
V1	Reachability (Safety)
V2	Linear Temporal Logic (Liveness)
V3	Existential/Universal Branching Temporal Logic
V4	Branching Temporal Logic (Nonzenoness)

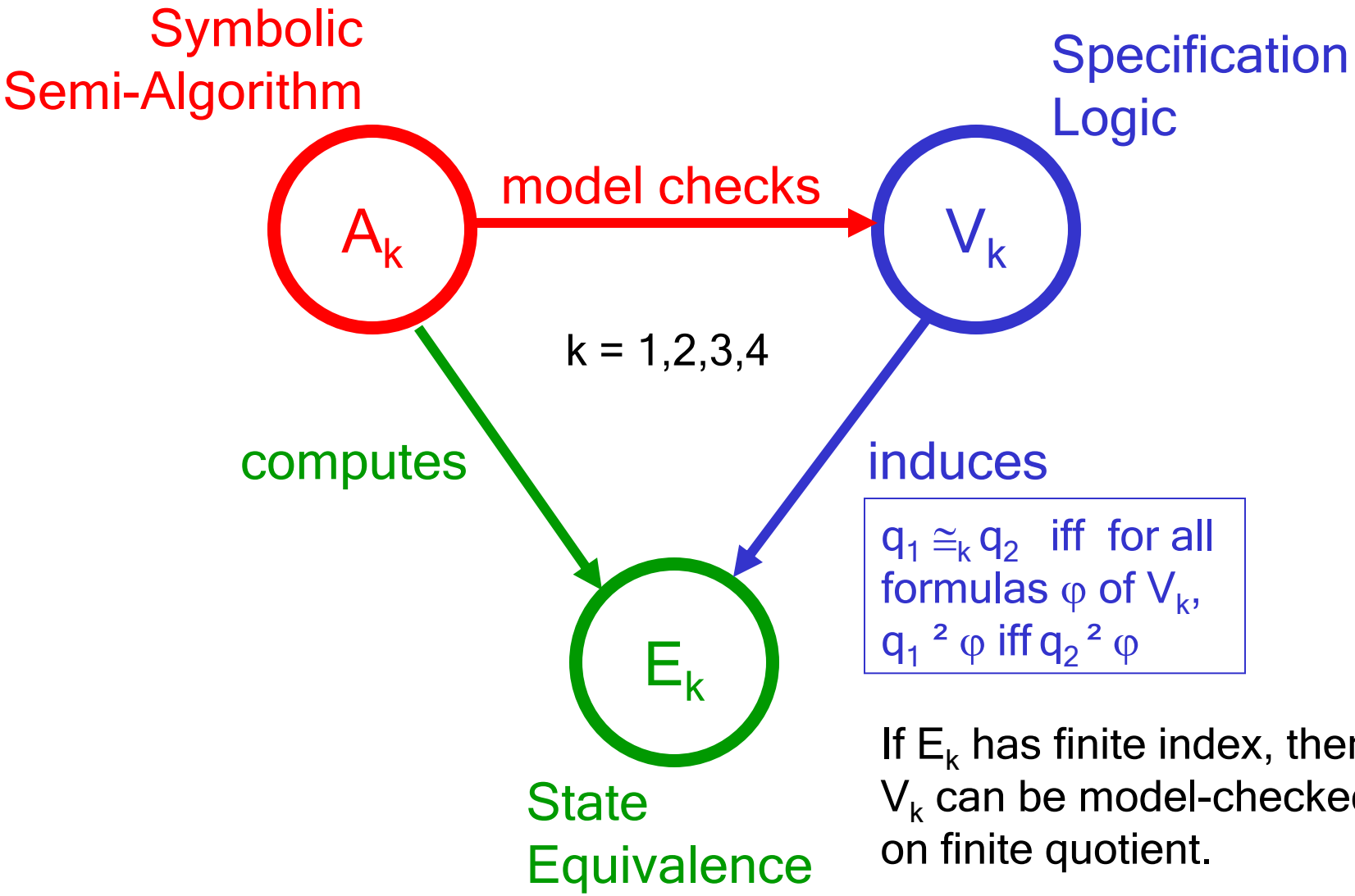
State Equivalences:

E1	Reach Equivalence
E2	Trace Equivalence
E3	Similarity
E4	Bisimilarity

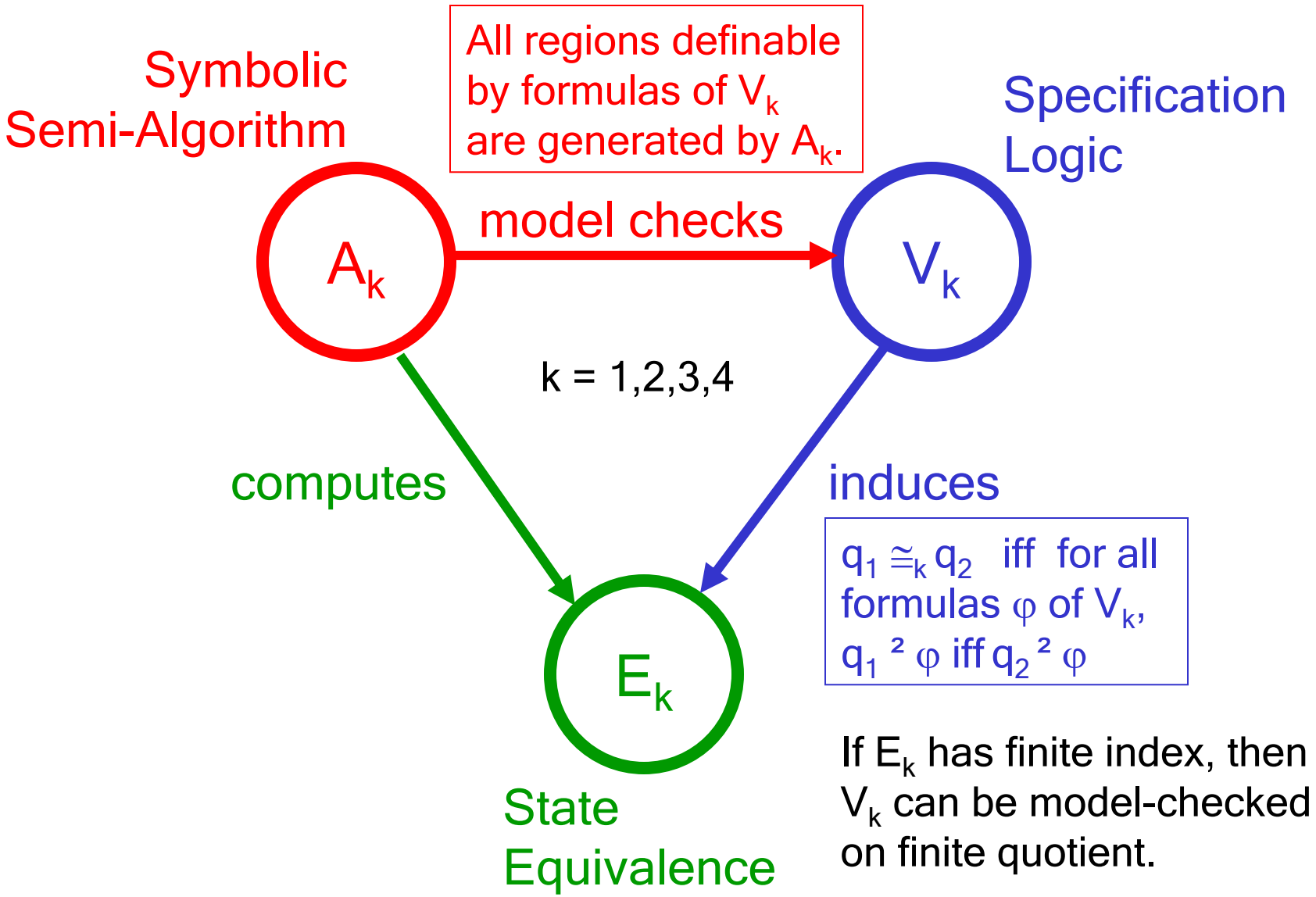
Symbolic Semi-Algorithms:

A1	Closure under pre
A2	Closure under pre, Å a
A3	Closure under pre, Å
A4	Closure under pre, Å , \


Symbolic
Semi-Algorithm
Specification
Logic
A_k
model checks
V_k
k = 1,2,3,4
computes
induces
E_k
State
Equivalence

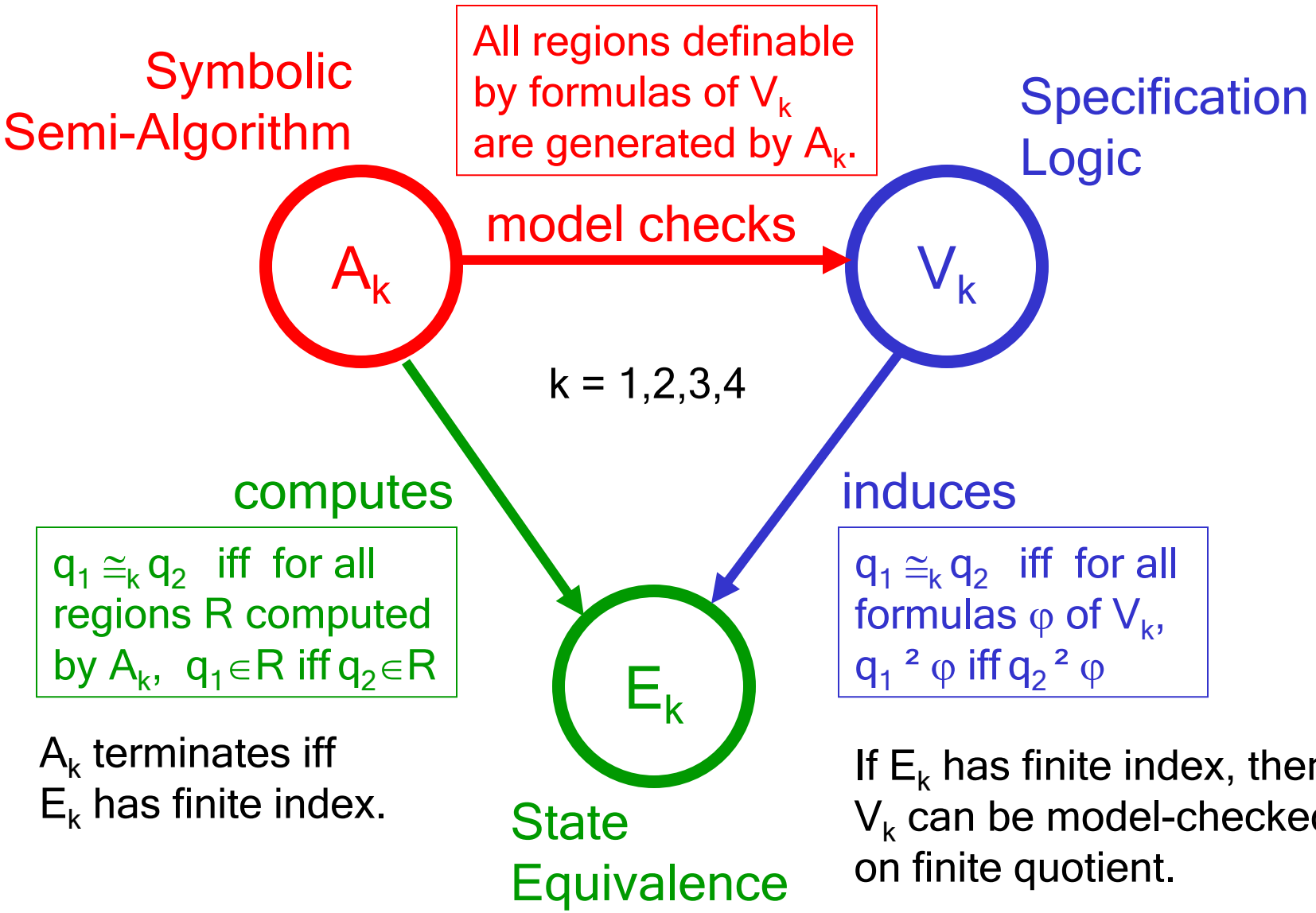

Symbolic
Semi-Algorithm
Specification
Logic
A_k
model checks
V_k
k = 1,2,3,4
computes
induces
$q_1 \cong_k q_2$ iff for all formulas φ of V_k, $q_1 \models \varphi$ iff $q_2 \models \varphi$
E_k
State
Equivalence
If E_k has finite index, then V_k can be model-checked on finite quotient.

A_k terminates iff symbolic model checking terminates for all formulas of V_k.



If E_k has finite index, then V_k can be model-checked on finite quotient.

A_k terminates iff symbolic model checking terminates for all formulas of V_k.



Four Classes of Symbolic Transition Systems

STS1: pre closure terminates $\Leftrightarrow$ Finite reach equiv $\Rightarrow$
Safety ($\exists\Diamond$) decidable

STS2: (pre, Å a) closure terminates $\Leftrightarrow$ Finite trace equiv $\Rightarrow$
Liveness (LTL) decidable

STS3: (pre, Å) closure terminates $\Leftrightarrow$ Finite similarity $\Rightarrow$
Existential/universal branching ($\exists$CTL, $\forall$CTL) decidable

STS4: (pre, Å , \ ) closure terminates $\Leftrightarrow$ Finite bisimilarity $\Rightarrow$
Nonzenoness (CTL) decidable

Four Classes of Symbolic Transition Systems

STS1: pre closure terminates $\Leftrightarrow$ Finite reach equiv $\Rightarrow$
Safety ($\exists\Diamond$) decidable
Well-structured transition systems of Abdulla, Finkel et al.

STS2: (pre, Å a) closure terminates $\Leftrightarrow$ Finite trace equiv $\Rightarrow$
Liveness (LTL) decidable
Initialized rectangular hybrid automata

STS3: (pre, Å) closure terminates $\Leftrightarrow$ Finite similarity $\Rightarrow$
Existential/universal branching ($\exists$CTL, $\forall$CTL) decidable
2D initialized rectangular hybrid automata

STS4: (pre, Å , \ ) closure terminates $\Leftrightarrow$ Finite bisimilarity $\Rightarrow$
Nonzenoness (CTL) decidable
Initialized singular hybrid automata

Example: Singular Hybrid Automata

$Q = B^m \times R^n$

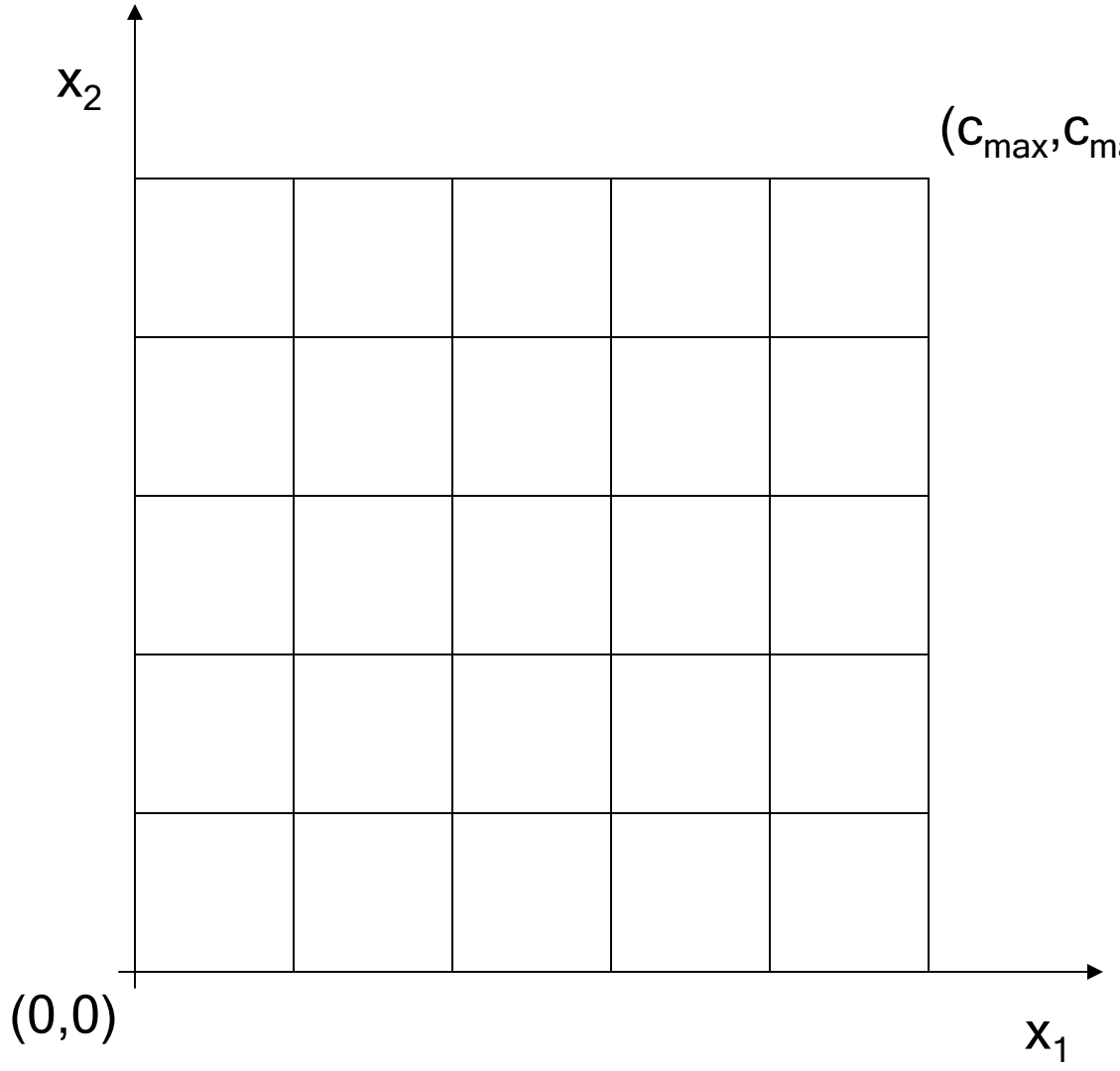
Invariants and guards:

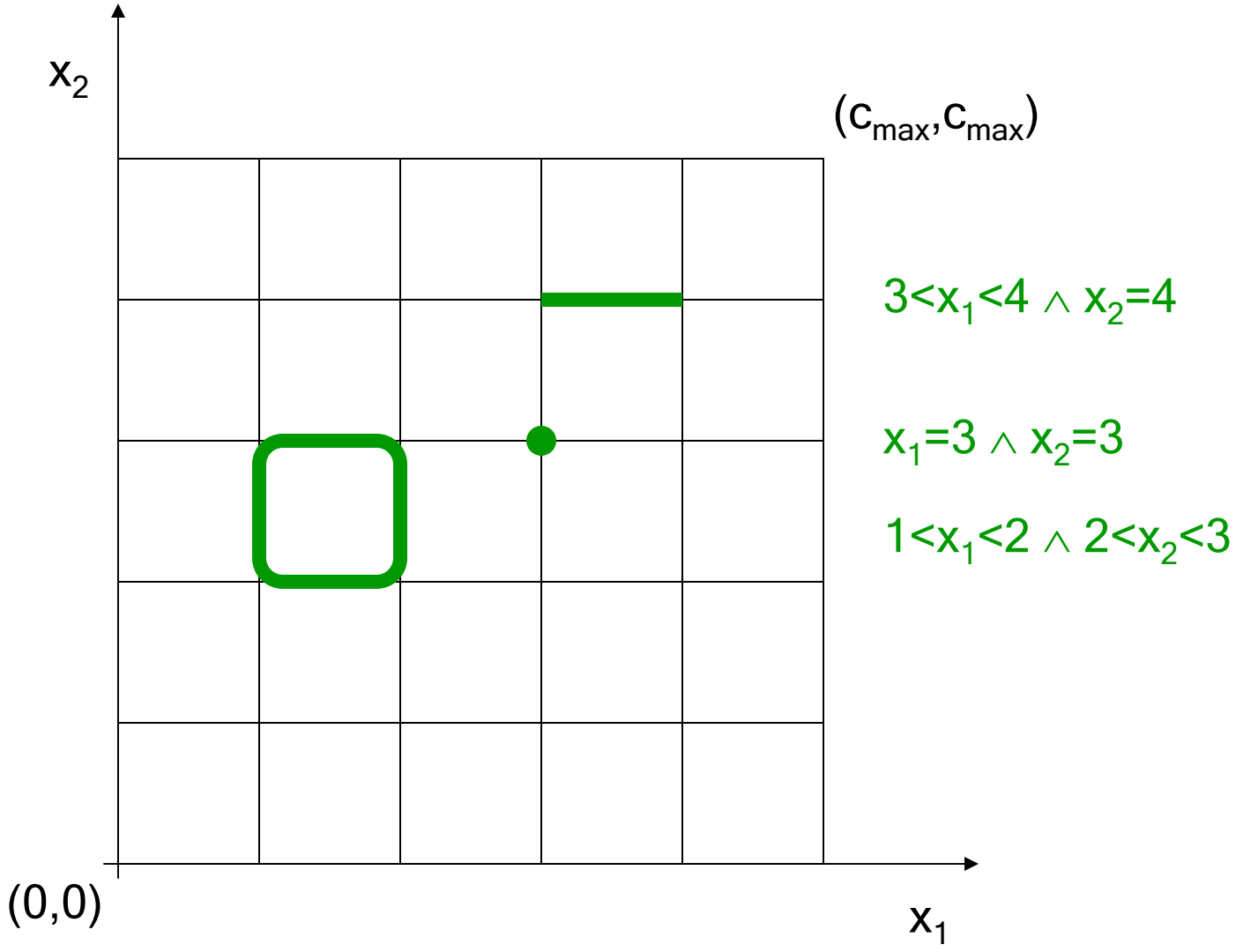
integral bounds, e.g. $x_1 < 7 \ \wedge \ 1 \leq x_2 \leq 2$

Flows: constant slopes, e.g. $x'_1 = 1; \ x'_2 = 2$

Jumps: integral assignments, e.g. $x_1 := 0; \ x_2 := 5$

$A = \{ x_i = c, c < x_i < c+1 \mid 1 \leq i \leq n, c \in N, c < c_{max} \}$


x_2
(c_{max}, c_{max})
(0,0)
x_1


x_2
(c_{max},c_{max})
$3<x_1<4 \wedge x_2=4$
$x_1=3 \wedge x_2=3$
$1<x_1<2 \wedge 2<x_2<3$
(0,0)
x_1

Example: Singular Hybrid Automata

$Q = B^m \times R^n$

Invariants and guards:

integral bounds, e.g. $x_1 < 7 \ \wedge \ 1 \leq x_2 \leq 2$

Flows: constant slopes, e.g. $x'_1 = 1; \ x'_2 = 2$

Jumps: integral assignments, e.g. $x_1 := 0; \ x_2 := 5$

$A = \{ x_i = c, c < x_i < c+1 \mid 1 \leq i \leq n, c \in N, c < c_{max} \}$

Initialized: assignment when slope changes.

Special Case: Timed Automata

$Q = B^m \times R^n$

Invariants and guards:

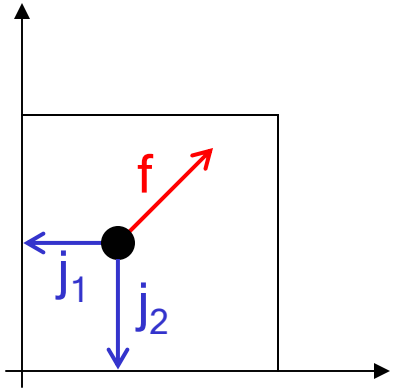
integral bounds, e.g. $x_1 < 7 \ \wedge \ 1 \leq x_2 \leq 2$

Flows: clocks, e.g. $x'_1 = 1; \ x'_2 = 1$

Jumps: integral assignments, e.g. $x_1 := 0; \ x_2 := 5$

$A = \{ x_i = c, c < x_i < c+1 \mid 1 \leq i \leq n, c \in N, c < c_{max} \}$

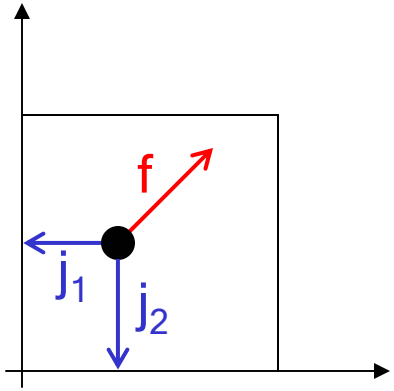
Always initialized.



f: $x_1' = 1; x_2' = 1$

j_1: $x_1 := 0$

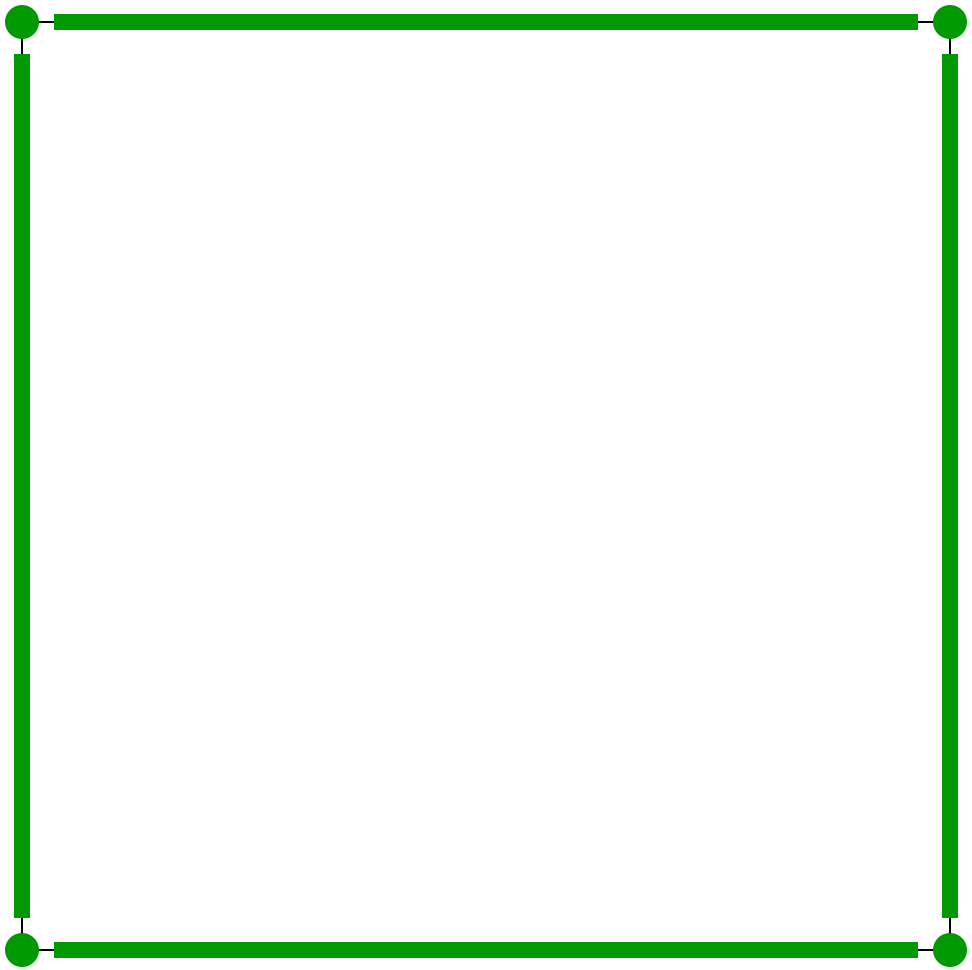
j_2: $x_2 := 0$

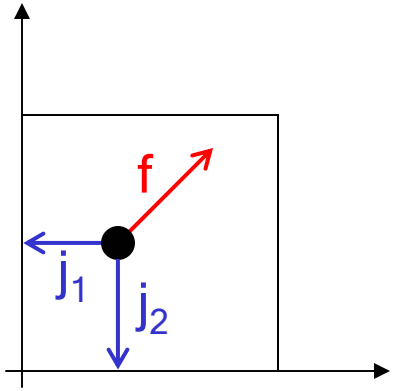


f: $x_1' = 1; x_2' = 1$

j_1: $x_1 := 0$

j_2: $x_2 := 0$

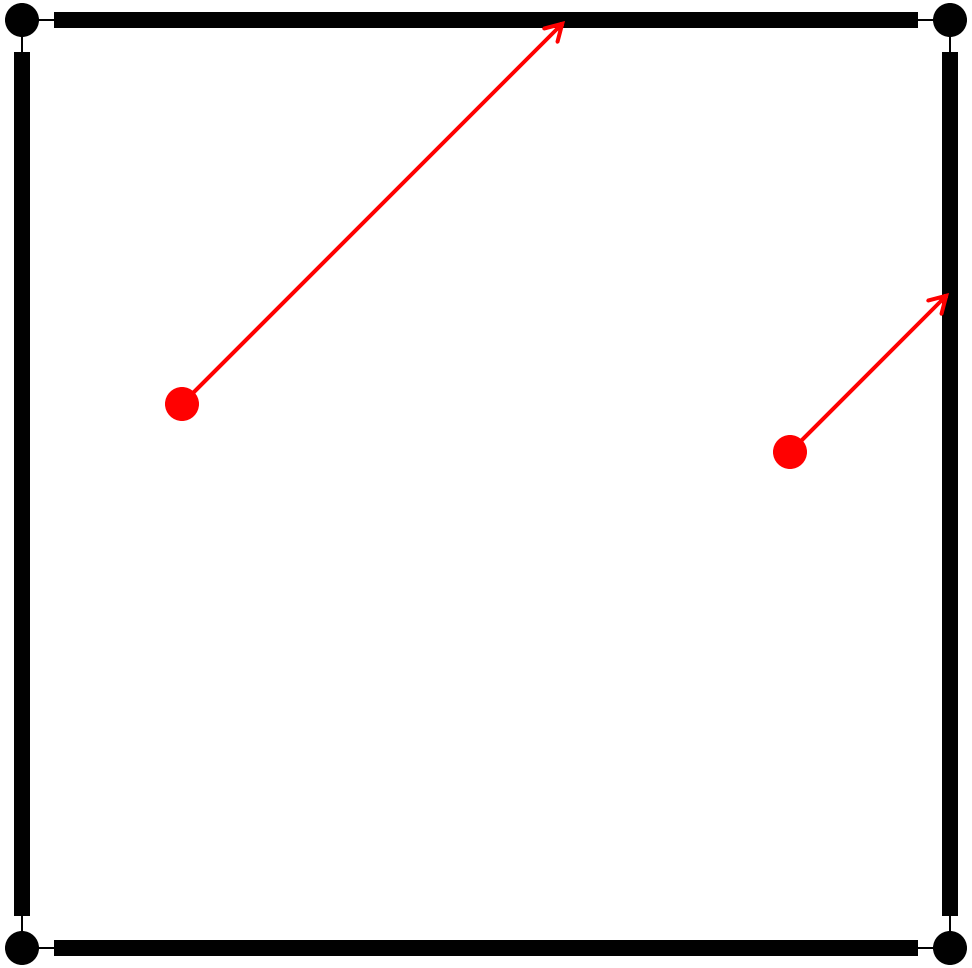


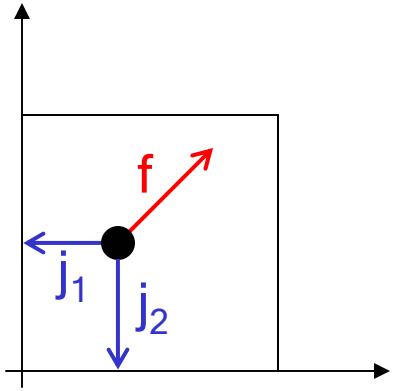


f: $x_1' = 1; x_2' = 1$

j_1: $x_1 := 0$

j_2: $x_2 := 0$

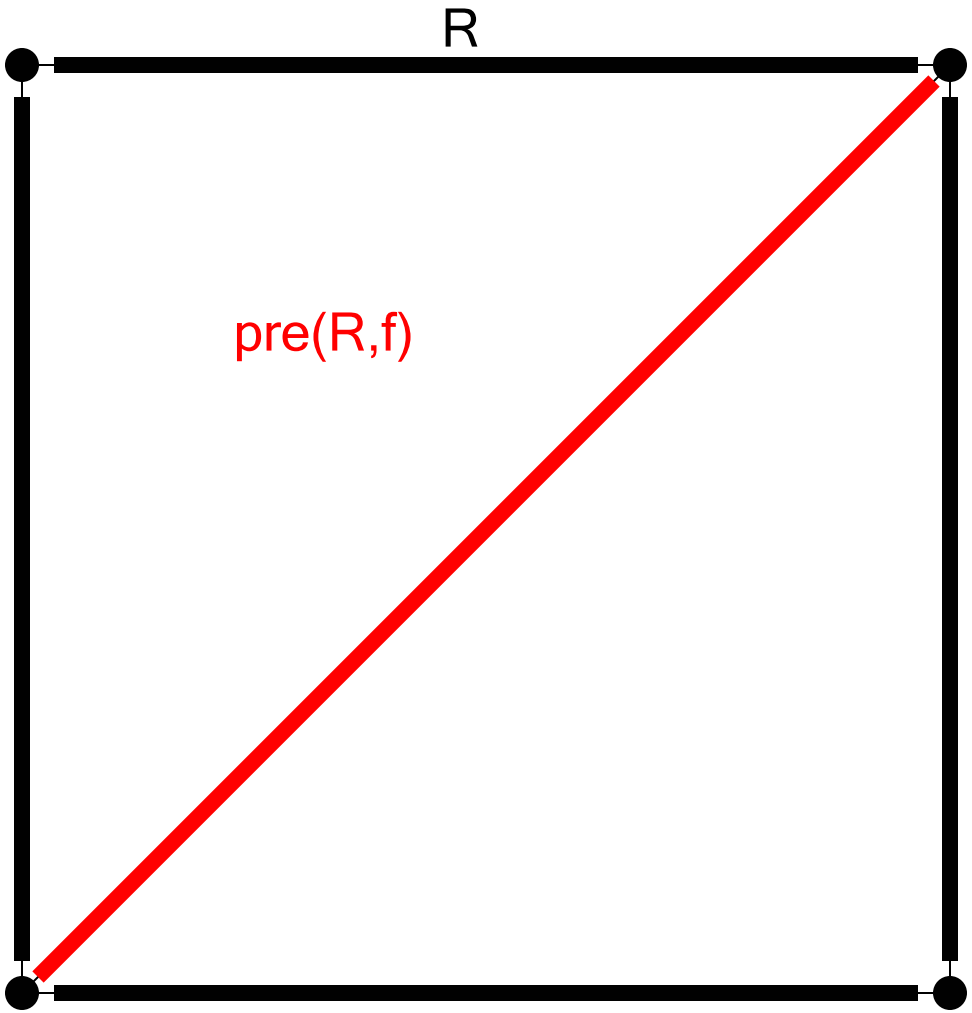


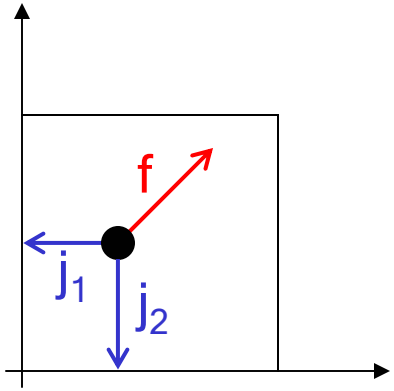


f: $x_1' = 1; x_2' = 1$

j_1: $x_1 := 0$

j_2: $x_2 := 0$

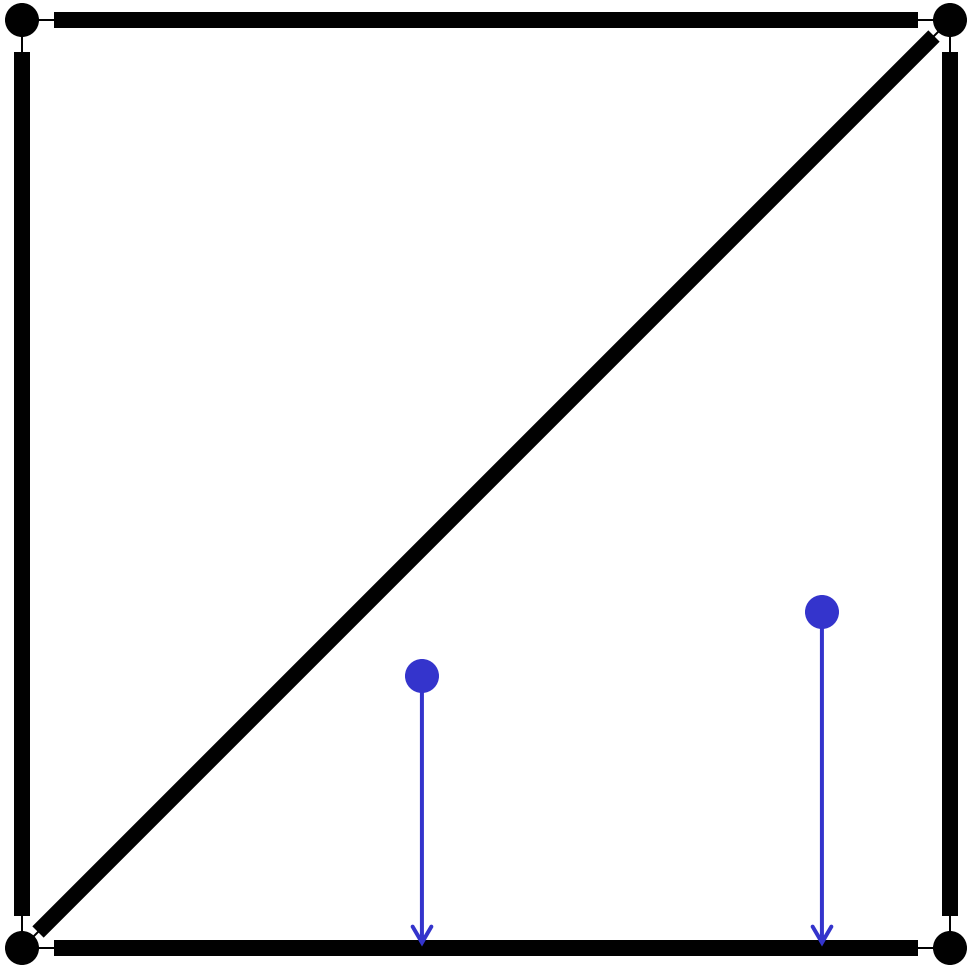


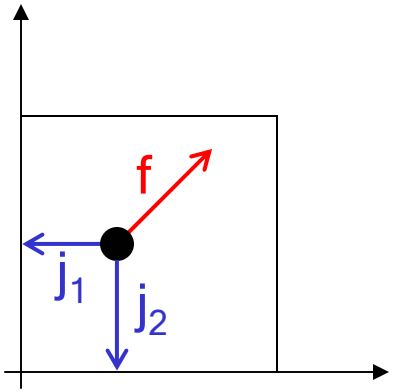


f: $x_1' = 1; x_2' = 1$

j_1: $x_1 := 0$

j_2: $x_2 := 0$

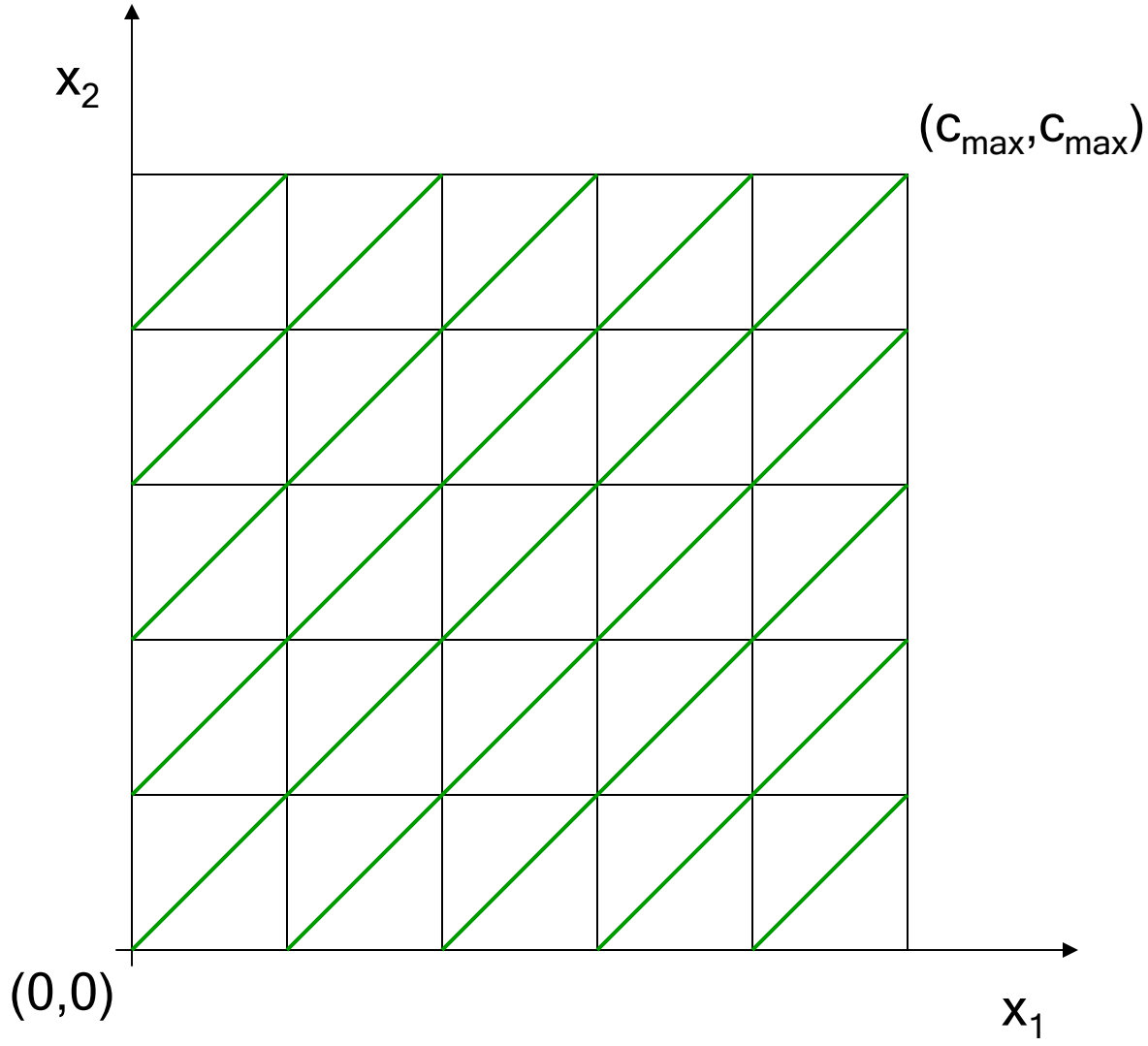




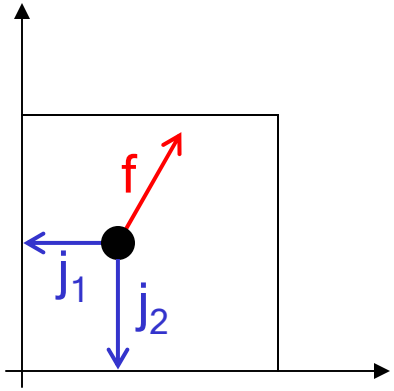
f: $x_1' = 1; x_2' = 1$

j_1: $x_1 := 0$

j_2: $x_2 := 0$



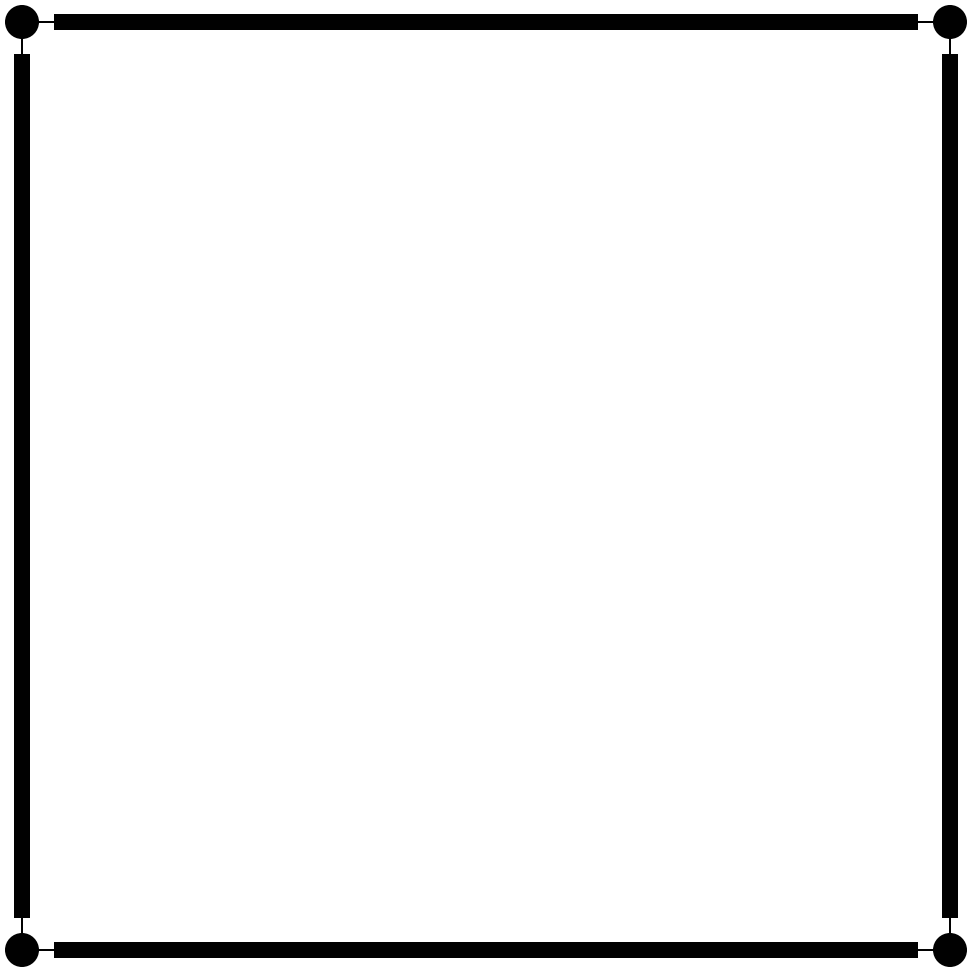
Finite bisimulation.

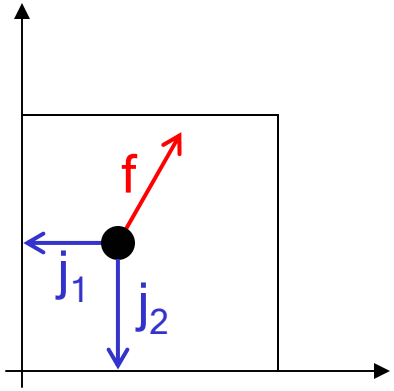


f: $x_1' = 1; x_2' = 2$

j_1: $x_1 := 0$

j_2: $x_2 := 0$

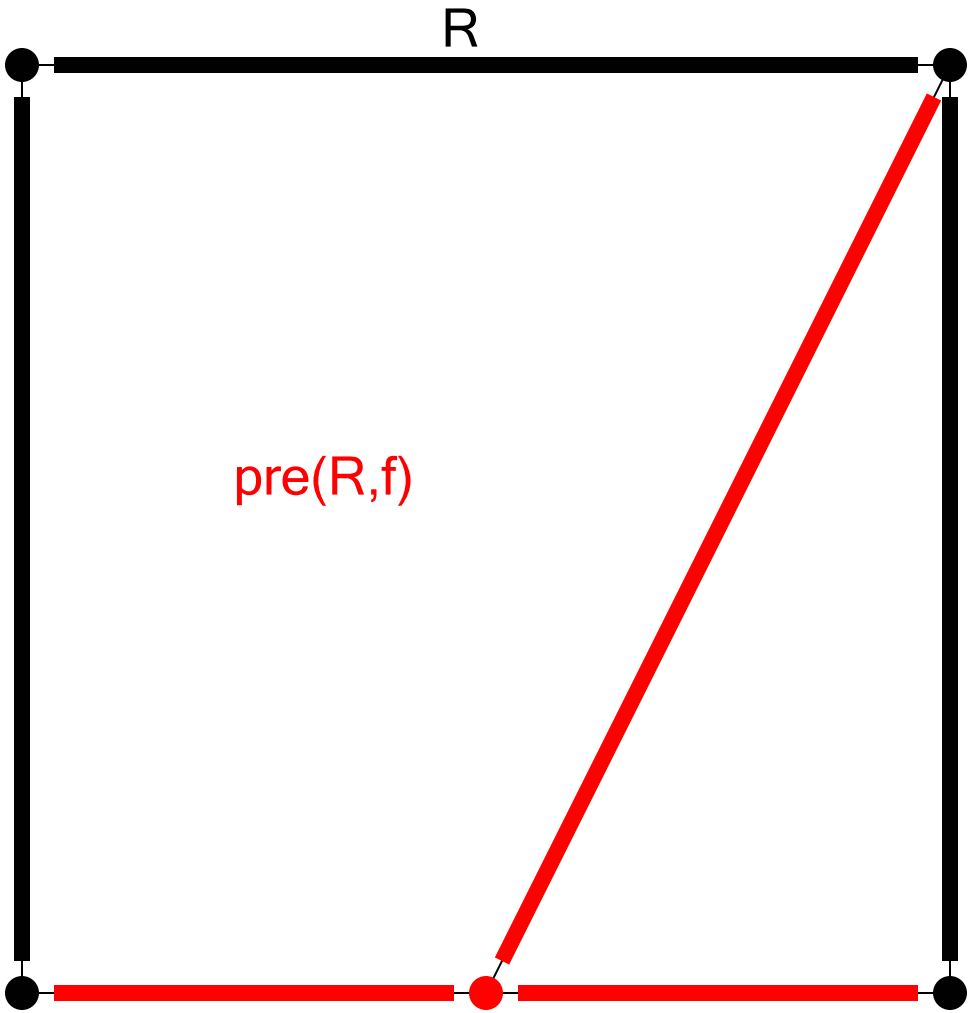


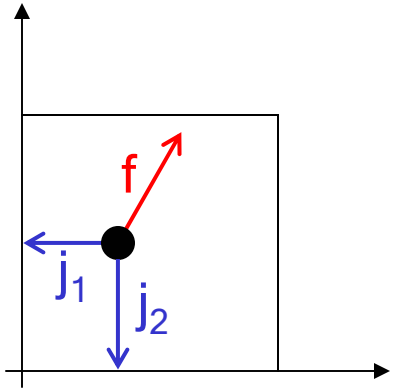


f: $x_1' = 1; x_2' = 2$

j_1: $x_1 := 0$

j_2: $x_2 := 0$

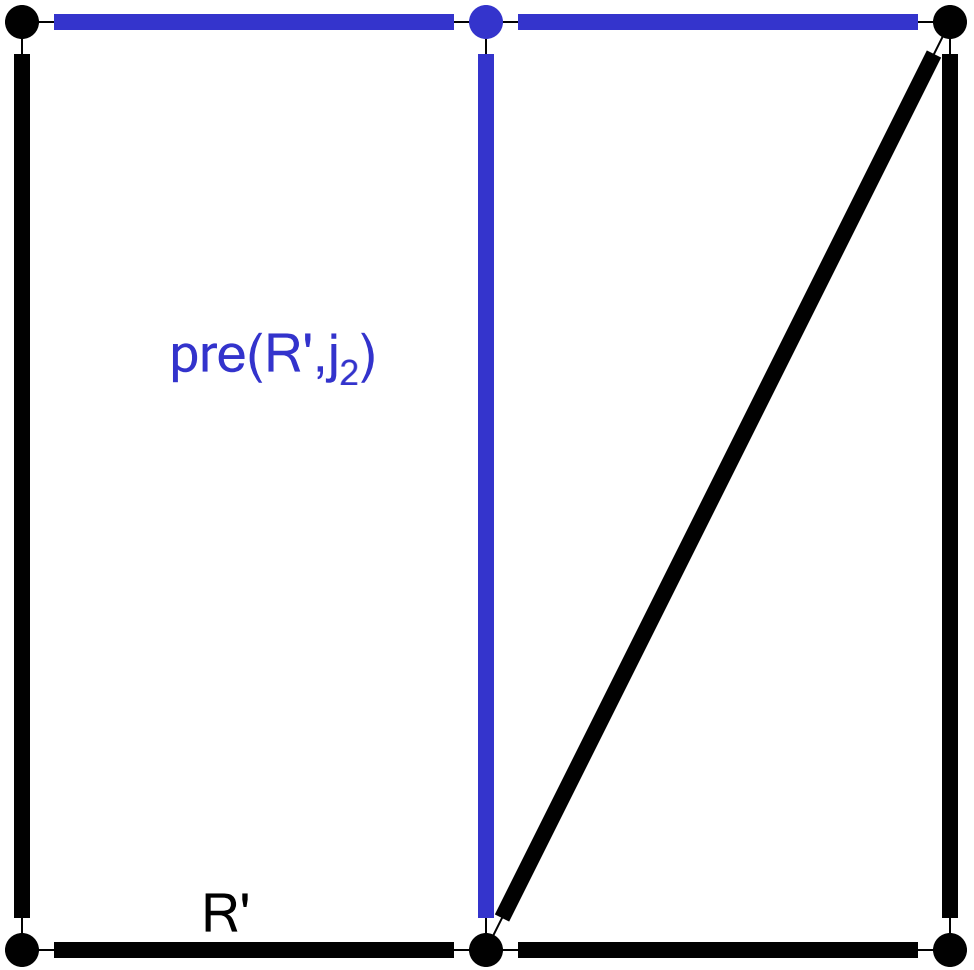


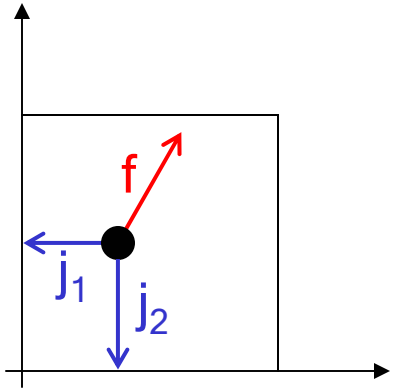


f: $x_1' = 1; x_2' = 2$

j_1: $x_1 := 0$

j_2: $x_2 := 0$

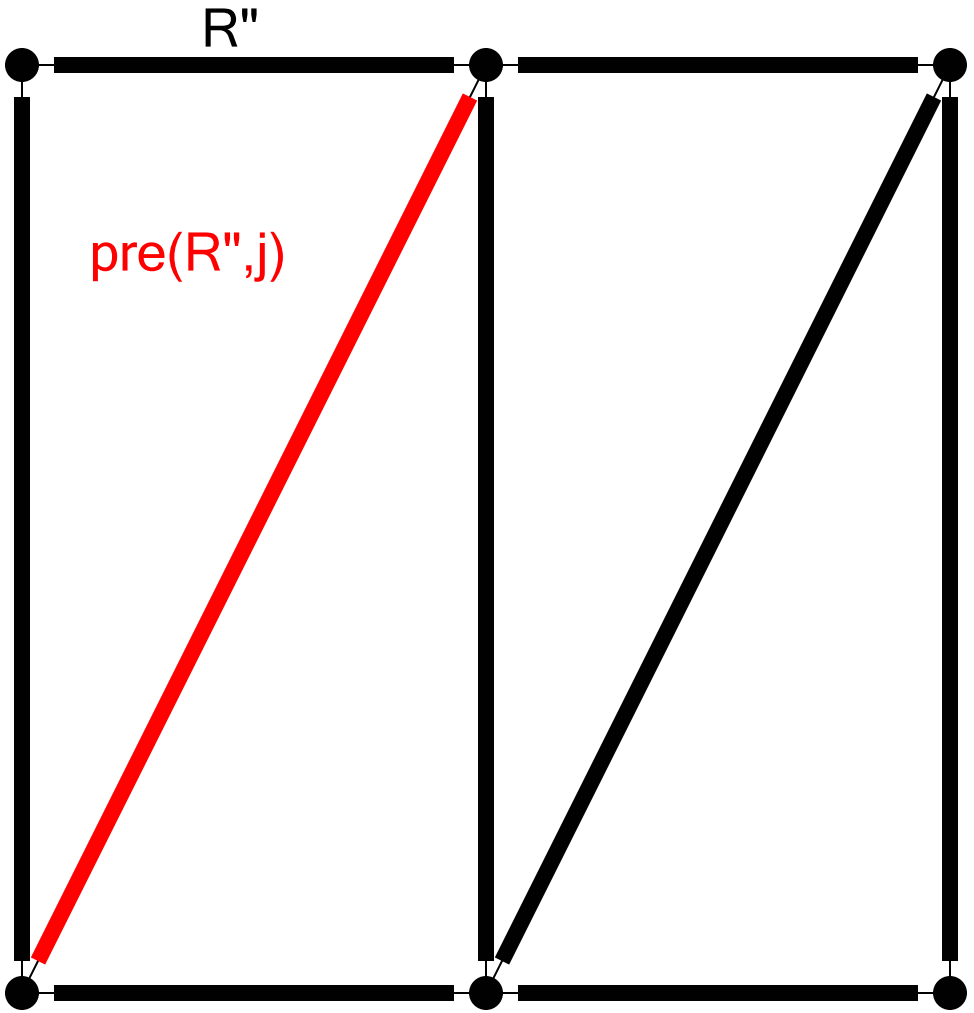


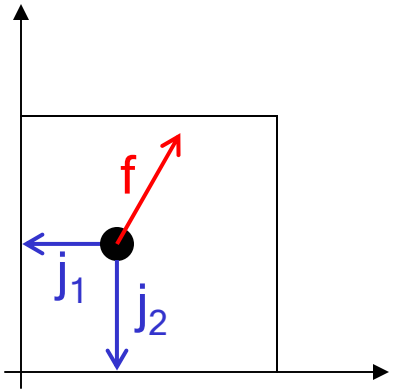


f: $x_1' = 1; x_2' = 2$

j_1: $x_1 := 0$

j_2: $x_2 := 0$

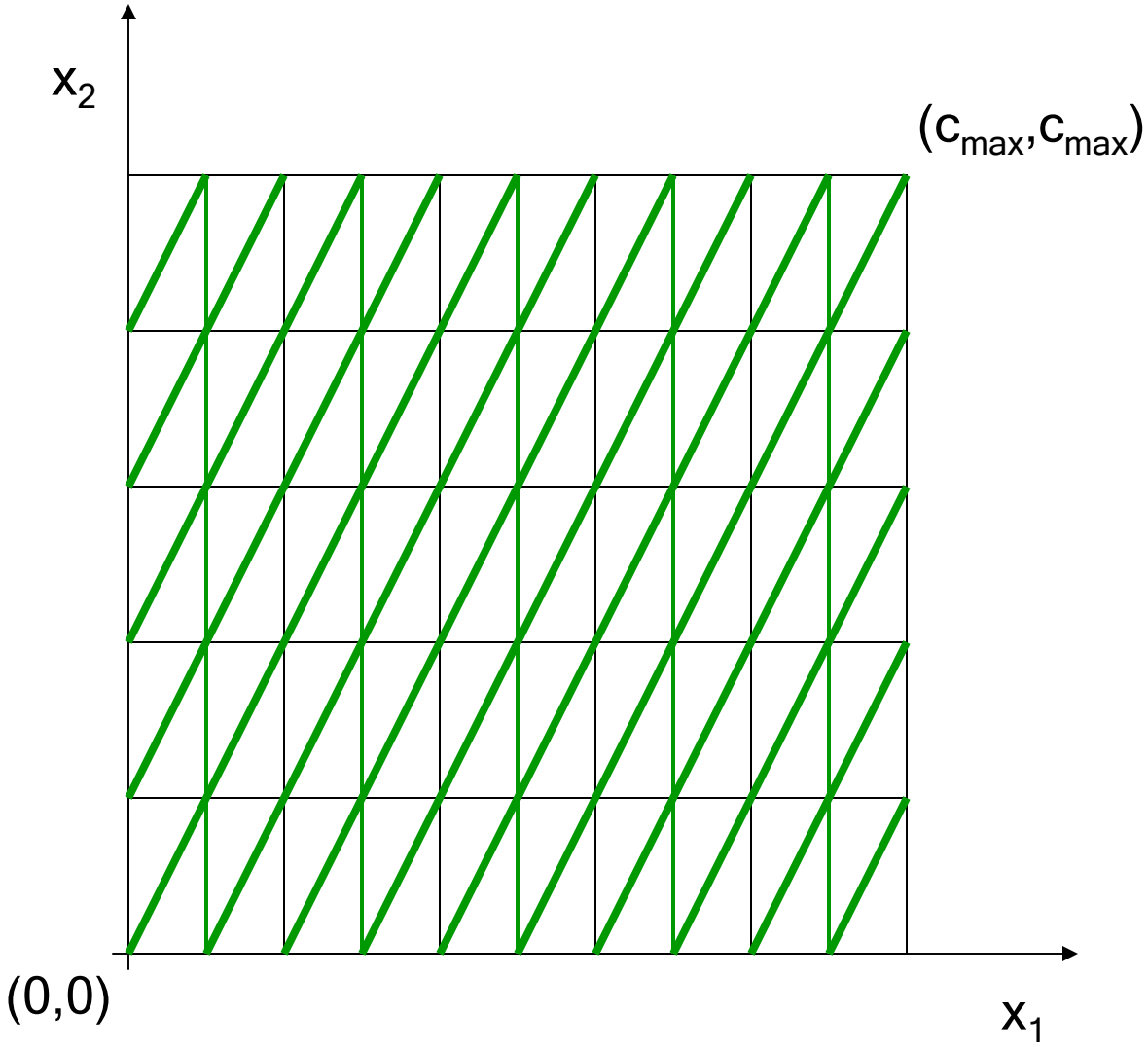




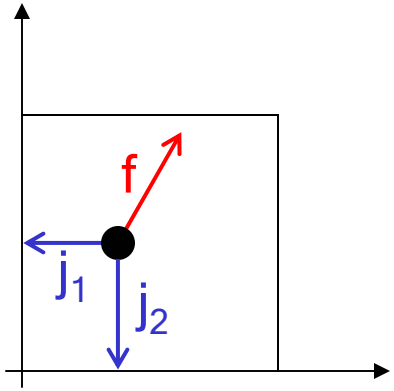
f: $x_1' = 1; x_2' = 2$

j_1: $x_1 := 0$

j_2: $x_2 := 0$



Finite bisimulation.

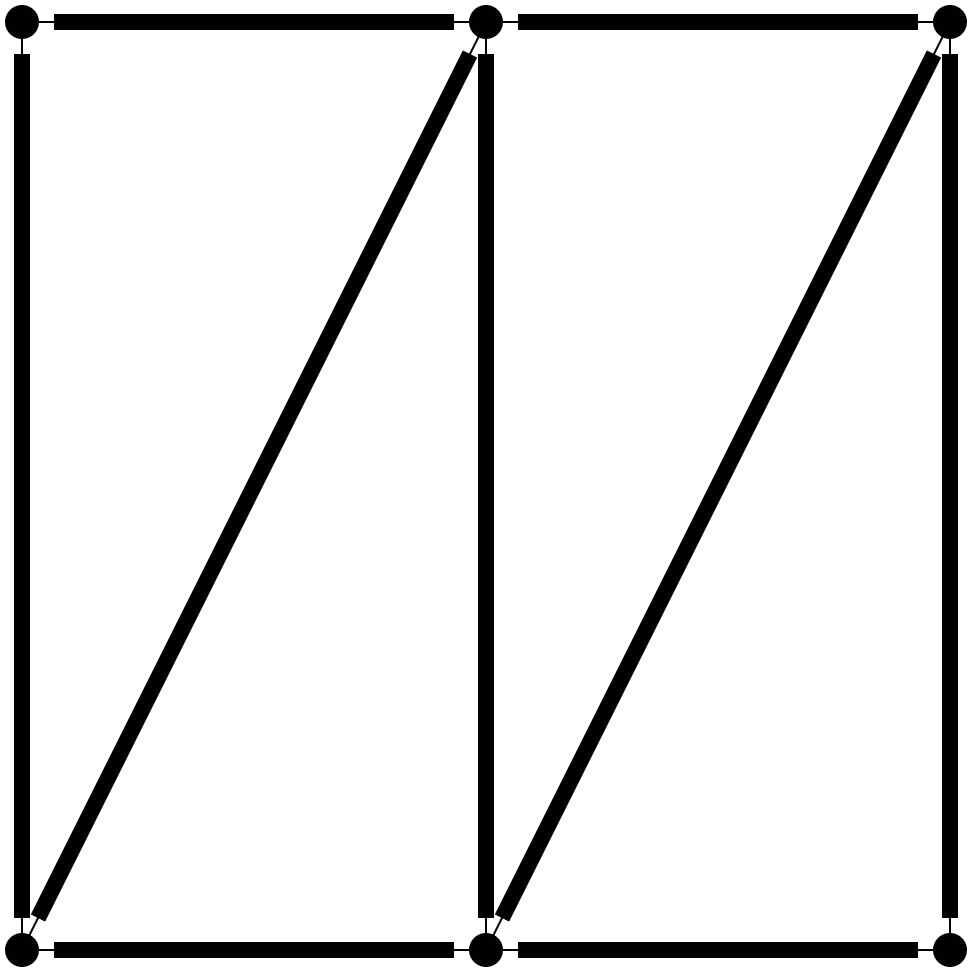


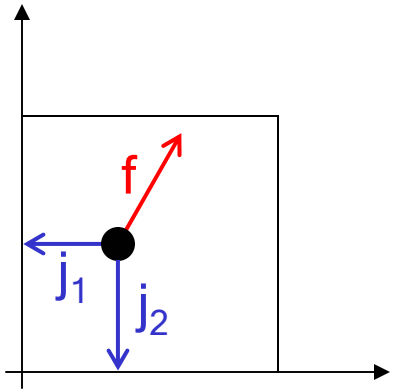
f: $x_1' = 1; x_2' = 2$

j_1: $x_1 := 0$

j_2: $x_2 := 0$

Observation, invariant, or guard: $x_1 > x_2$



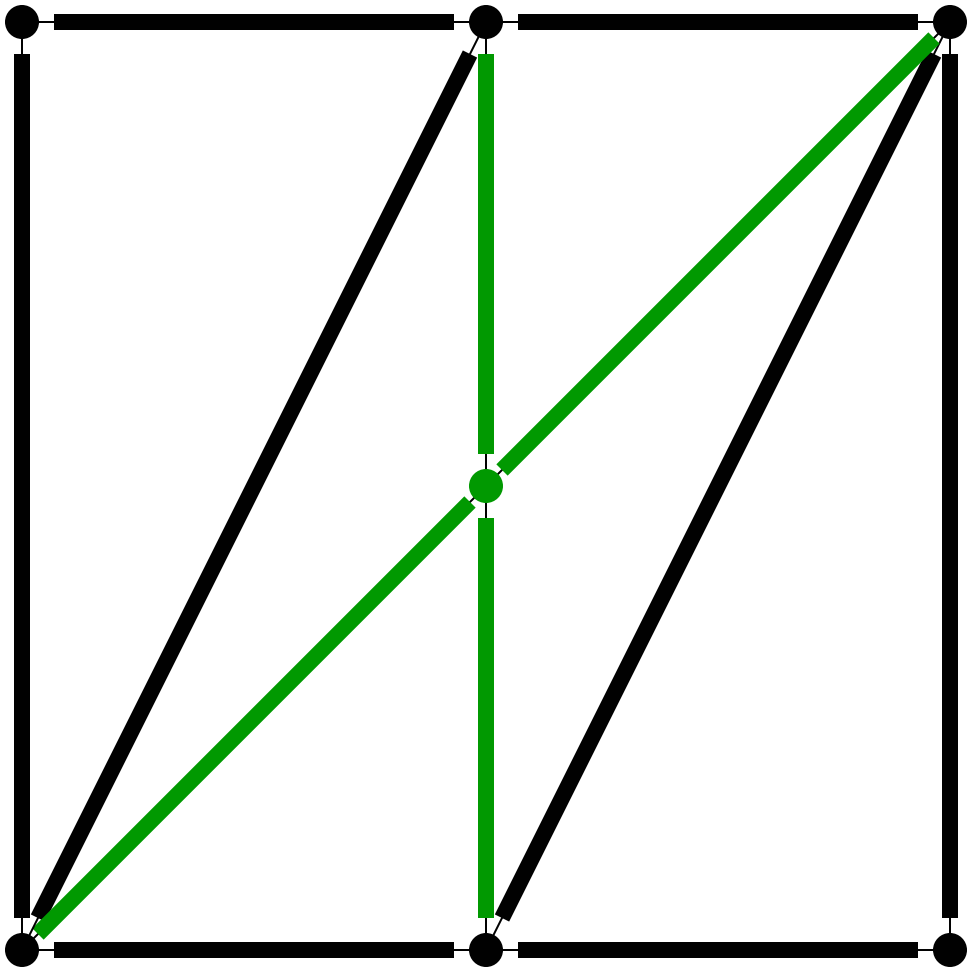


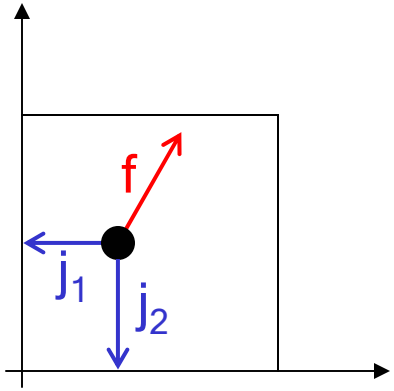
f: $x_1' = 1; x_2' = 2$

j_1: $x_1 := 0$

j_2: $x_2 := 0$

Observation, invariant, or guard: $x_1 > x_2$



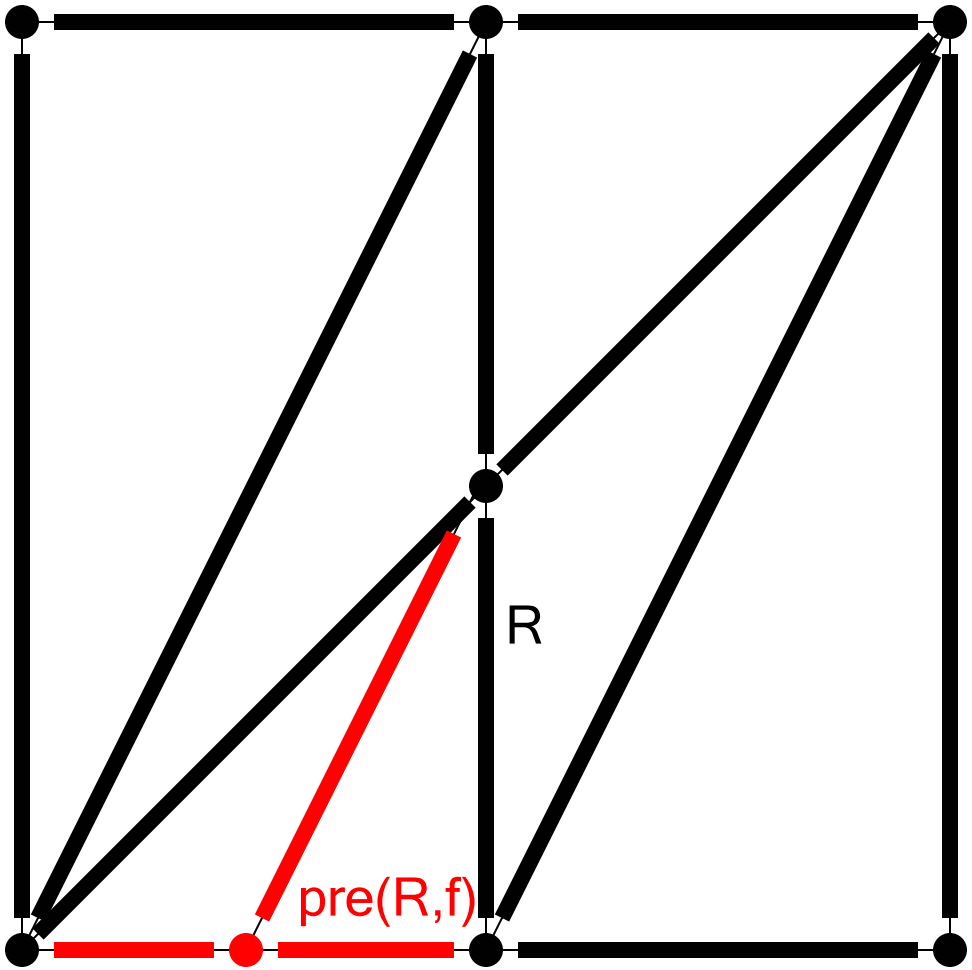


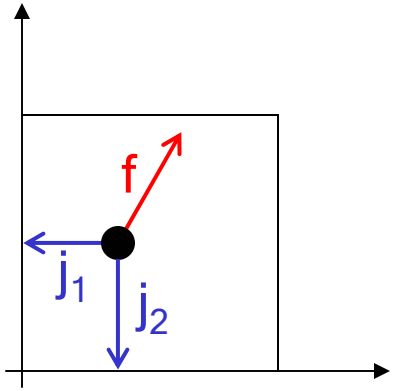
f: $x_1' = 1; x_2' = 2$

j_1: $x_1 := 0$

j_2: $x_2 := 0$

Observation, invariant, or guard: $x_1 > x_2$



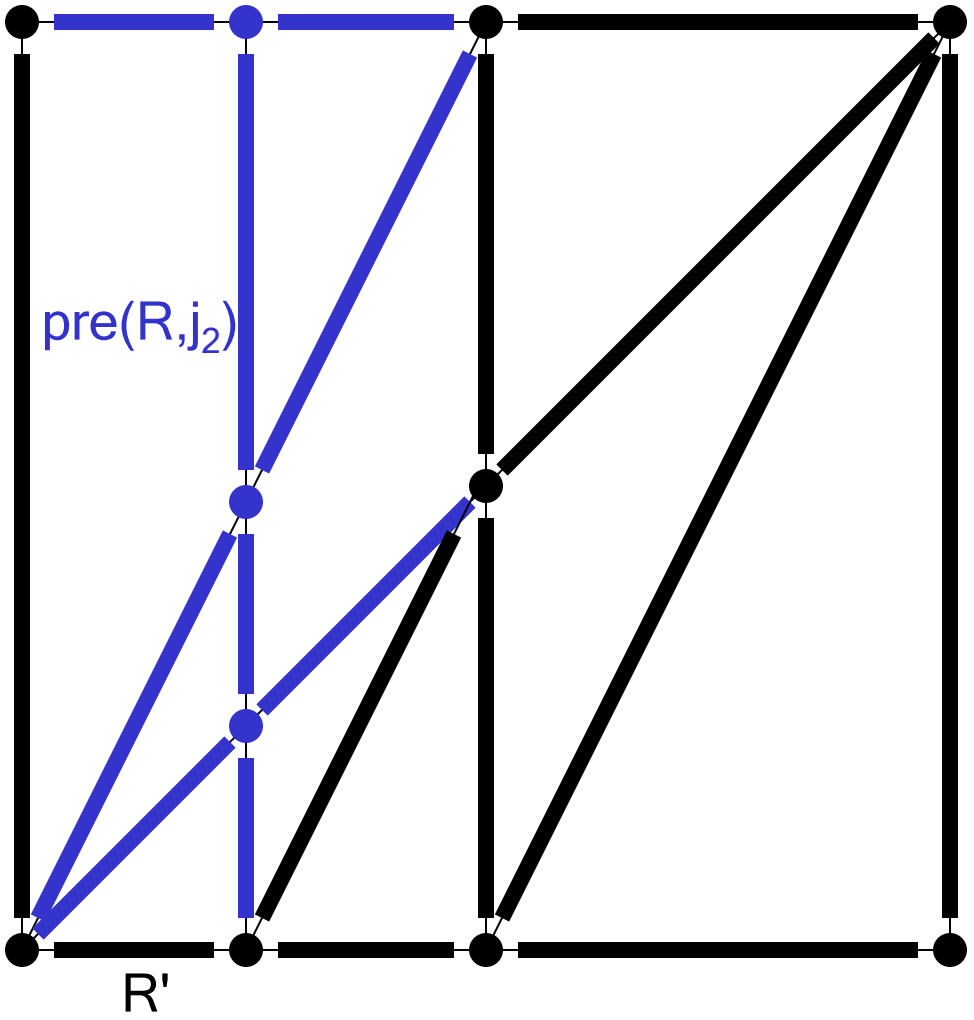


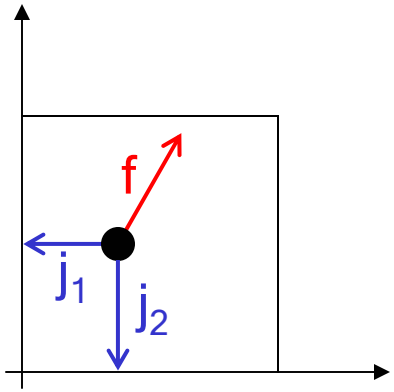
f: $x_1' = 1; x_2' = 2$

j_1: $x_1 := 0$

j_2: $x_2 := 0$

Observation, invariant, or guard: $x_1 > x_2$



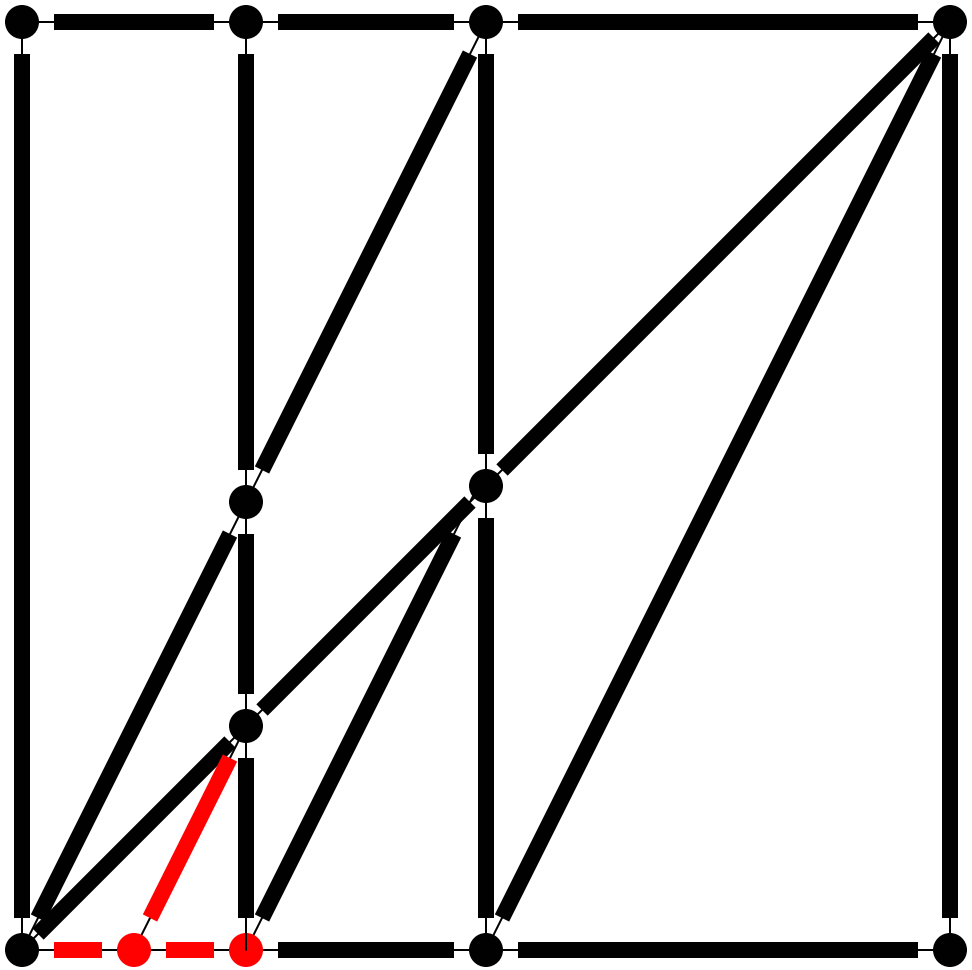


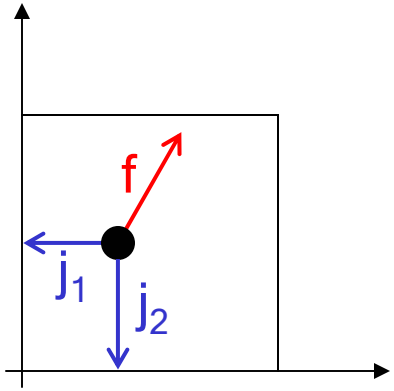
f: $x_1' = 1; x_2' = 2$

j_1: $x_1 := 0$

j_2: $x_2 := 0$

Observation, invariant, or guard: $x_1 > x_2$



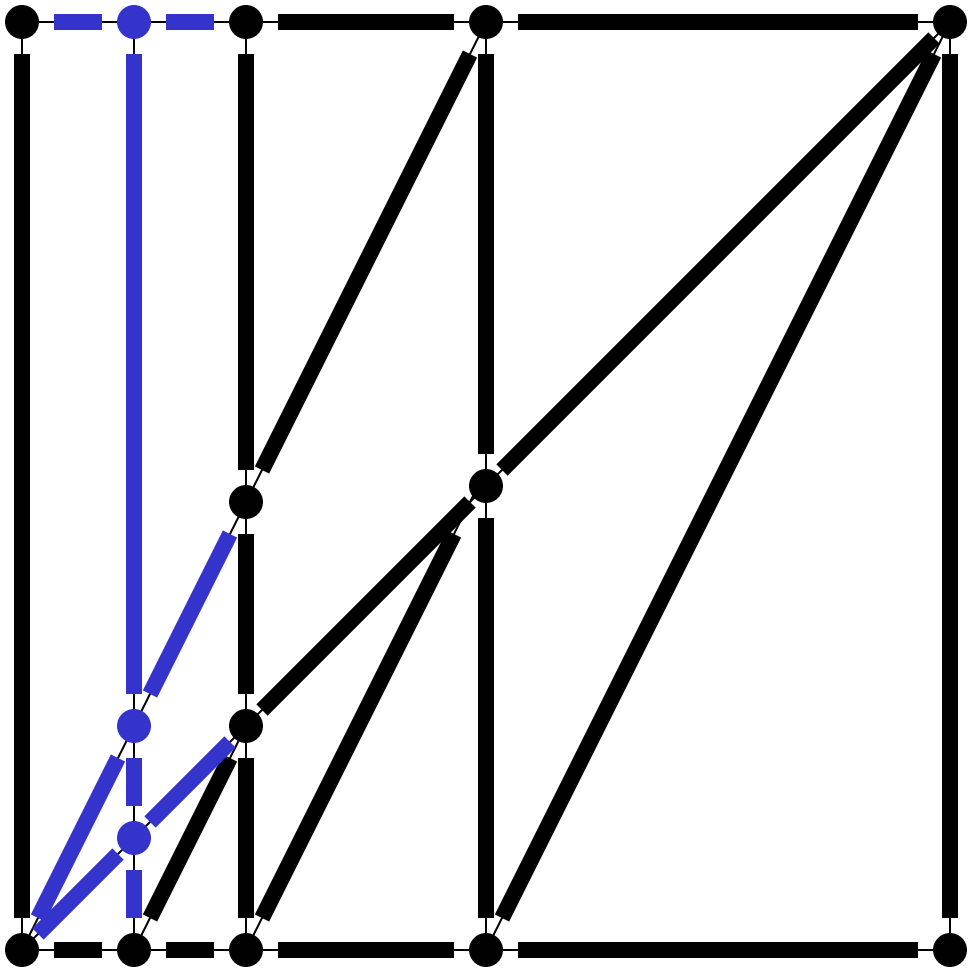


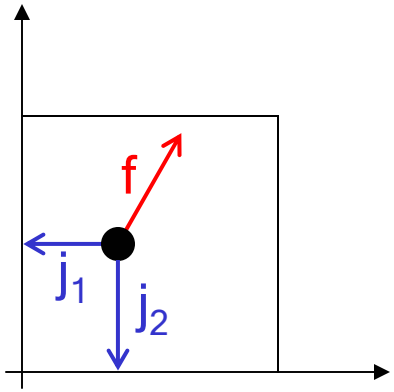
f: $x_1' = 1; x_2' = 2$

j_1: $x_1 := 0$

j_2: $x_2 := 0$

Observation, invariant, or guard: $x_1 > x_2$



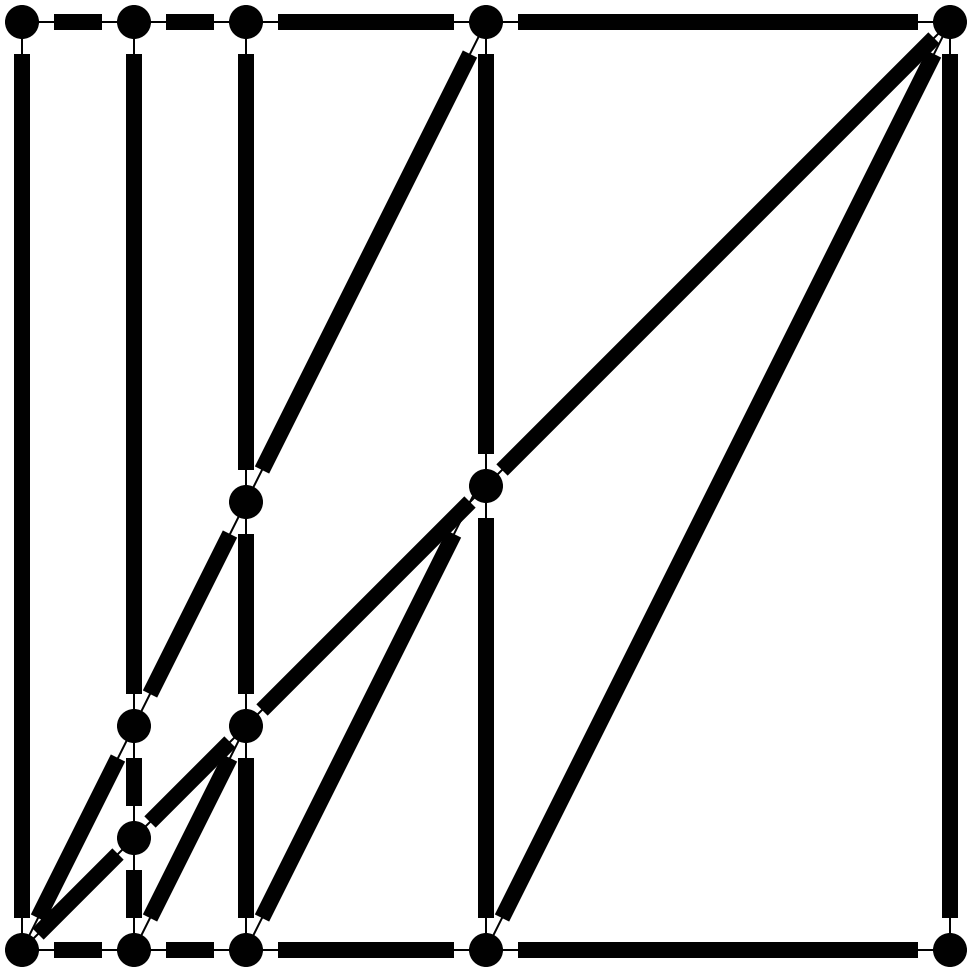


f: $x_1' = 1; x_2' = 2$

j_1: $x_1 := 0$

j_2: $x_2 := 0$

Observation, invariant, or guard: $x_1 > x_2$



Infinite bisimulation.

Example: Rectangular Hybrid Automata

$Q = B^m \times R^n$

Invariants and guards:

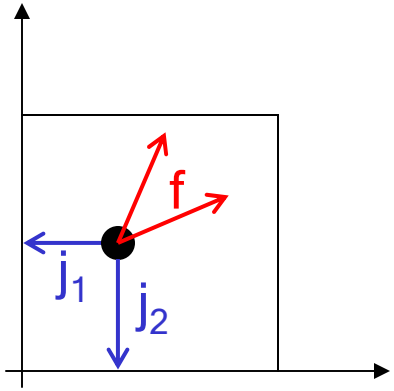
integral bounds, e.g. $x_1 < 7 \ \wedge \ 1 \leq x_2 \leq 2$

Flows: bounded slopes, e.g. $x'_1 \in [1,2]; \ x'_2 = 1$

Jumps: integral assignments, e.g. $x_1 := 0; \ x_2 := 5$

$A = \{ x_i = c, c < x_i < c+1 \mid 1 \leq i \leq n, c \in N, c < c_{max} \}$

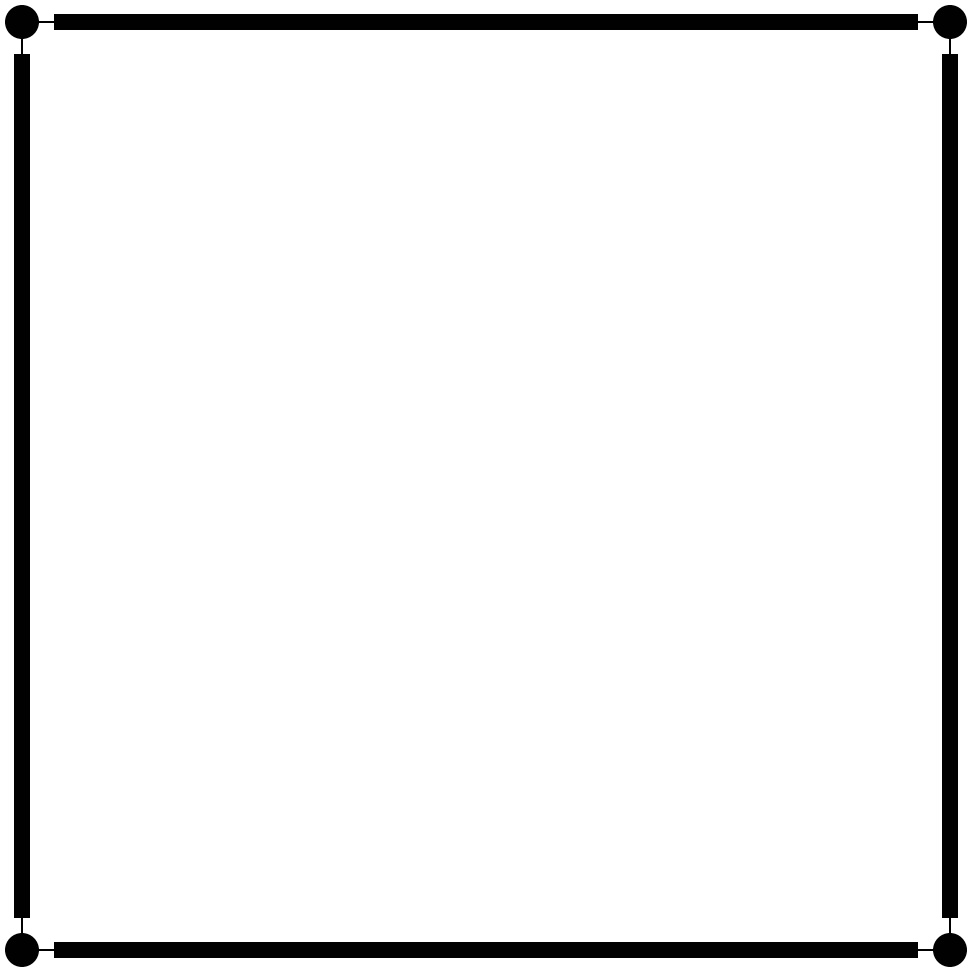
Initialized: assignment when slope bounds change.

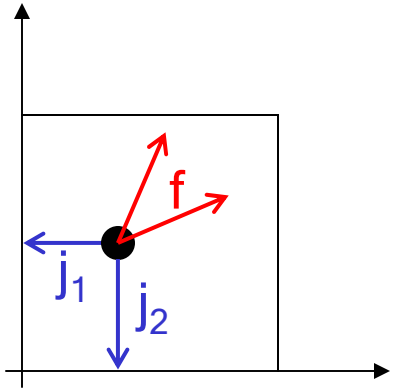


f: $x_1' \in [1,2]; x_2' \in [1,2]$

j_1: $x_1 := 0$

j_2: $x_2 := 0$

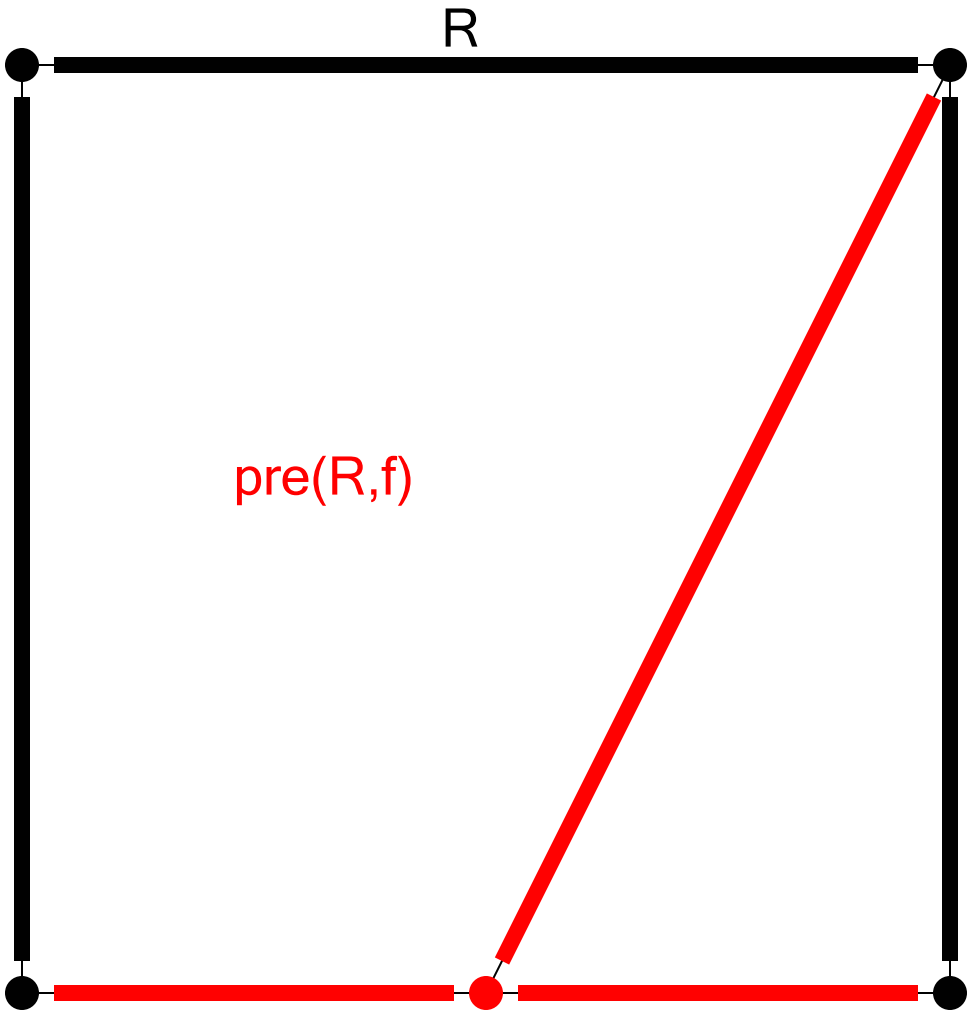


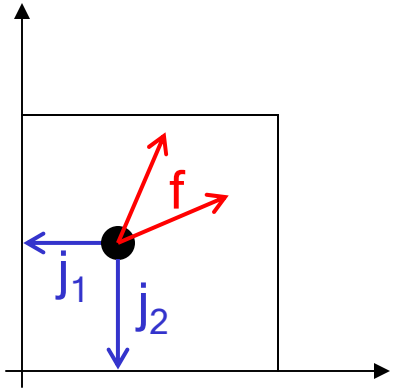


f: $x_1' \in [1,2]$; $x_2' \in [1,2]$

j_1: $x_1 := 0$

j_2: $x_2 := 0$

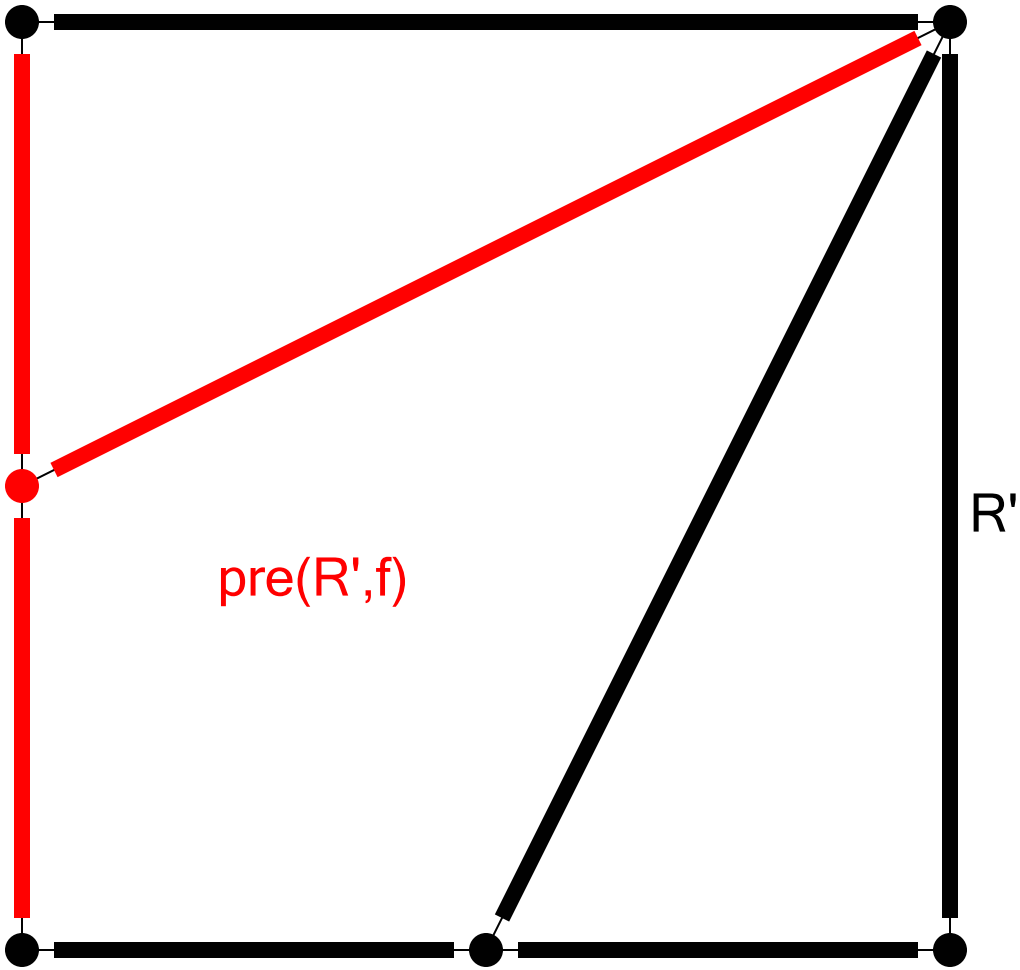


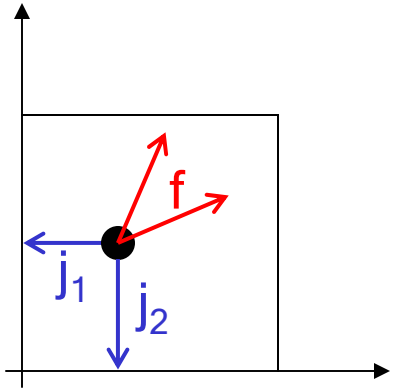


f: $x_1' \in [1,2]$; $x_2' \in [1,2]$

j_1: $x_1 := 0$

j_2: $x_2 := 0$

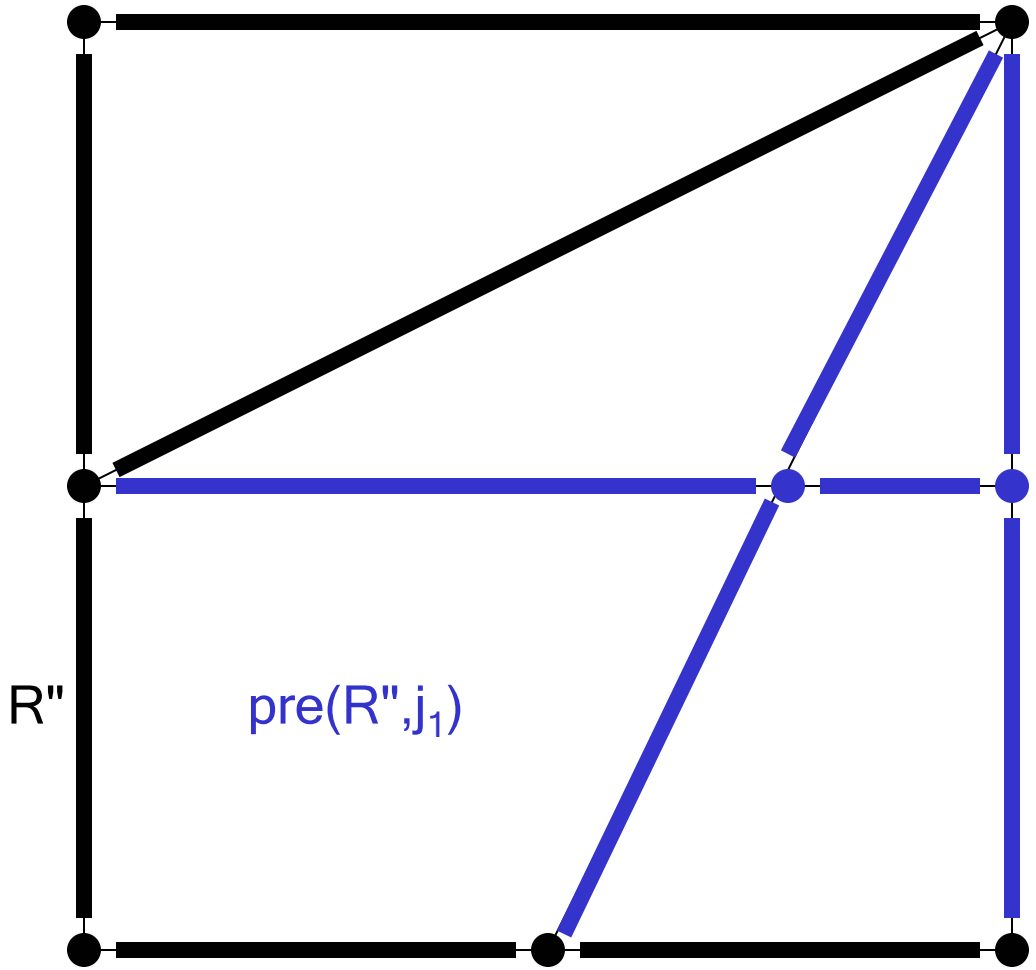


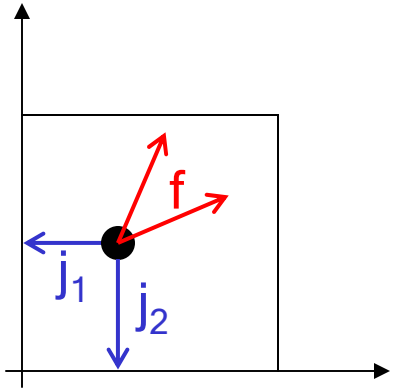


f: $x_1' \in [1,2]$; $x_2' \in [1,2]$

j_1: $x_1 := 0$

j_2: $x_2 := 0$

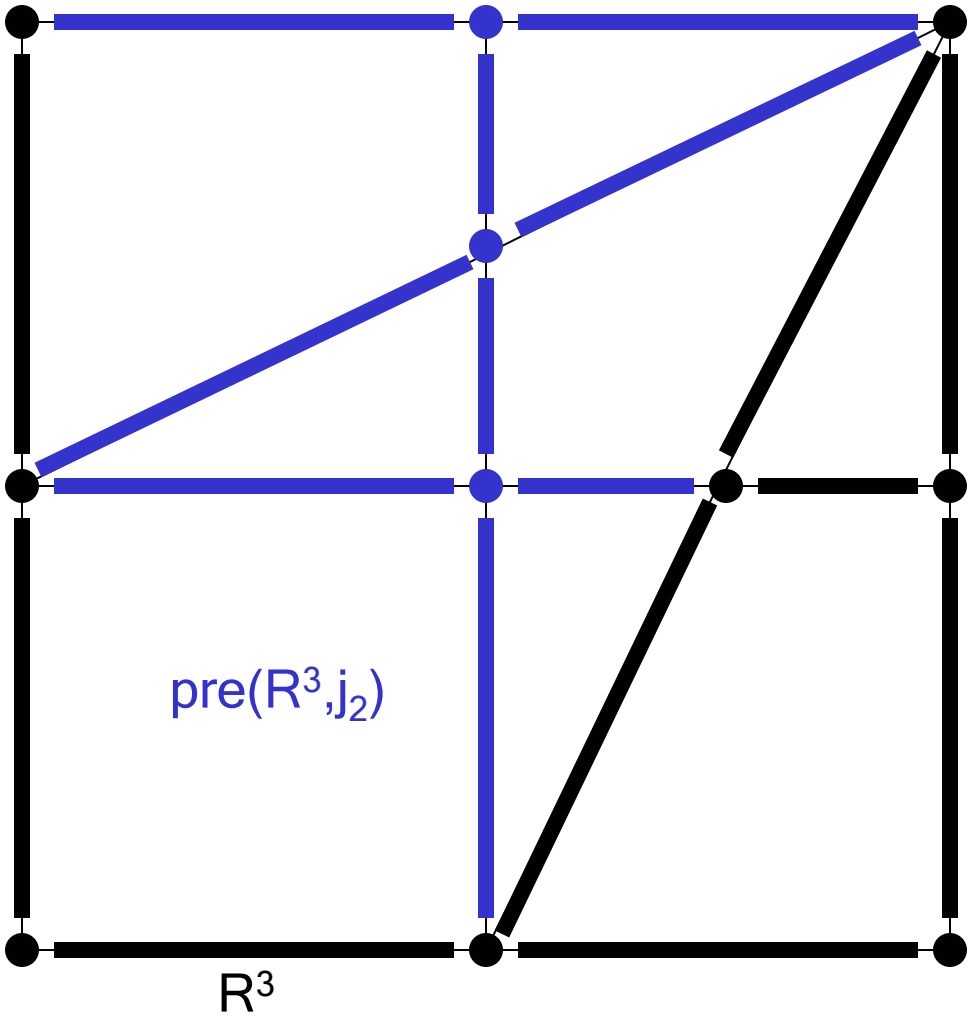


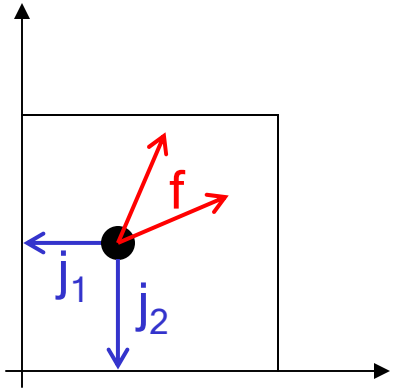


f: $x_1' \in [1,2]$; $x_2' \in [1,2]$

j_1: $x_1 := 0$

j_2: $x_2 := 0$

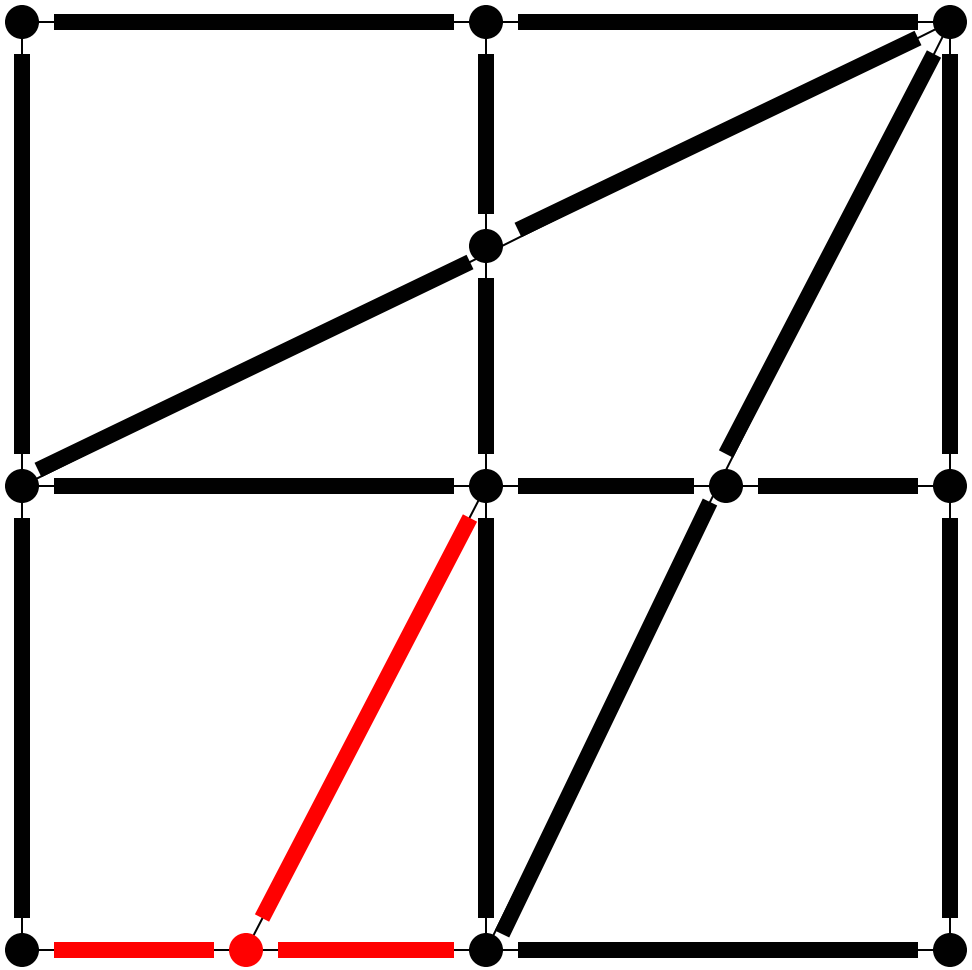


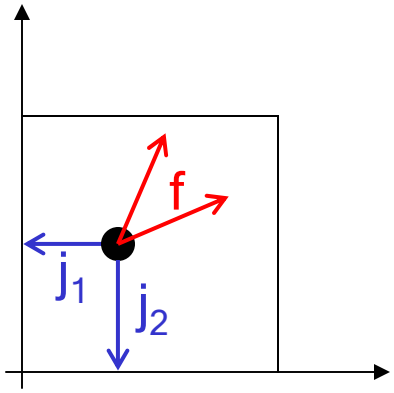


f: $x_1' \in [1,2]$; $x_2' \in [1,2]$

j_1: $x_1 := 0$

j_2: $x_2 := 0$

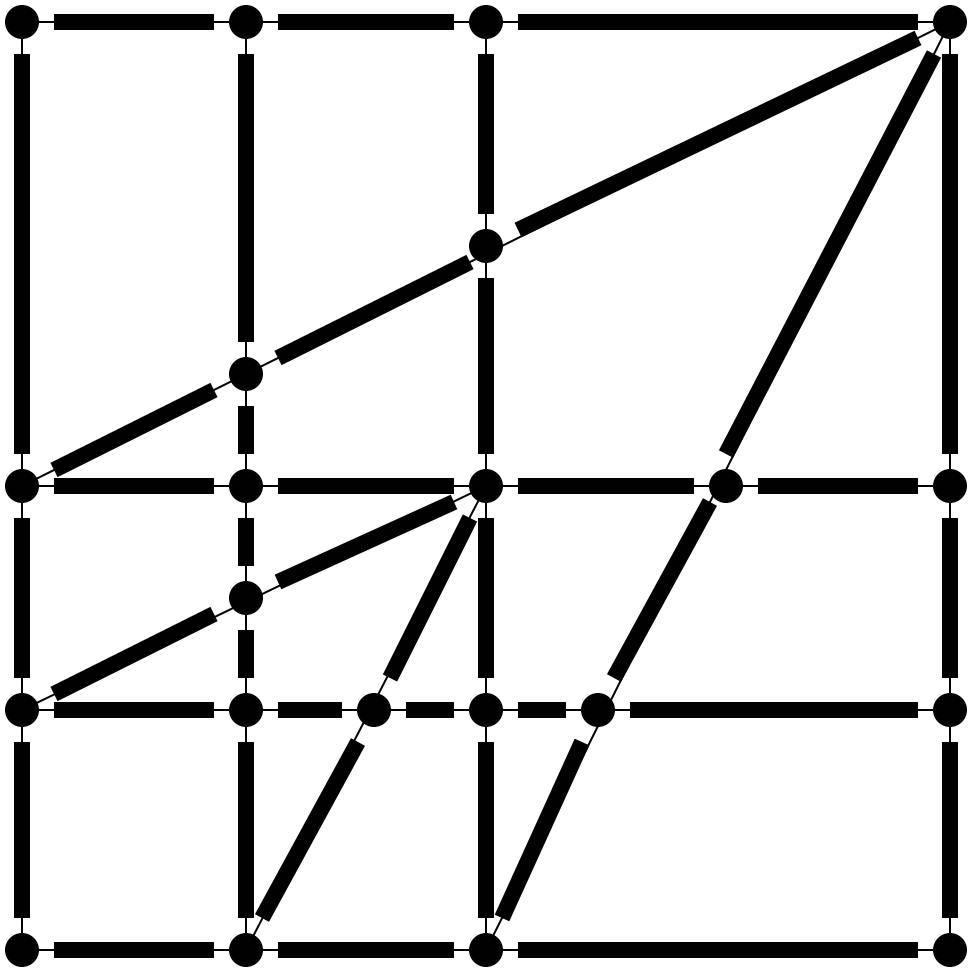




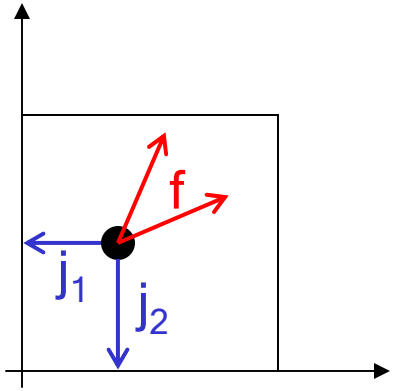
f: $x_1' \in [1,2]$; $x_2' \in [1,2]$

j_1: $x_1 := 0$

j_2: $x_2 := 0$



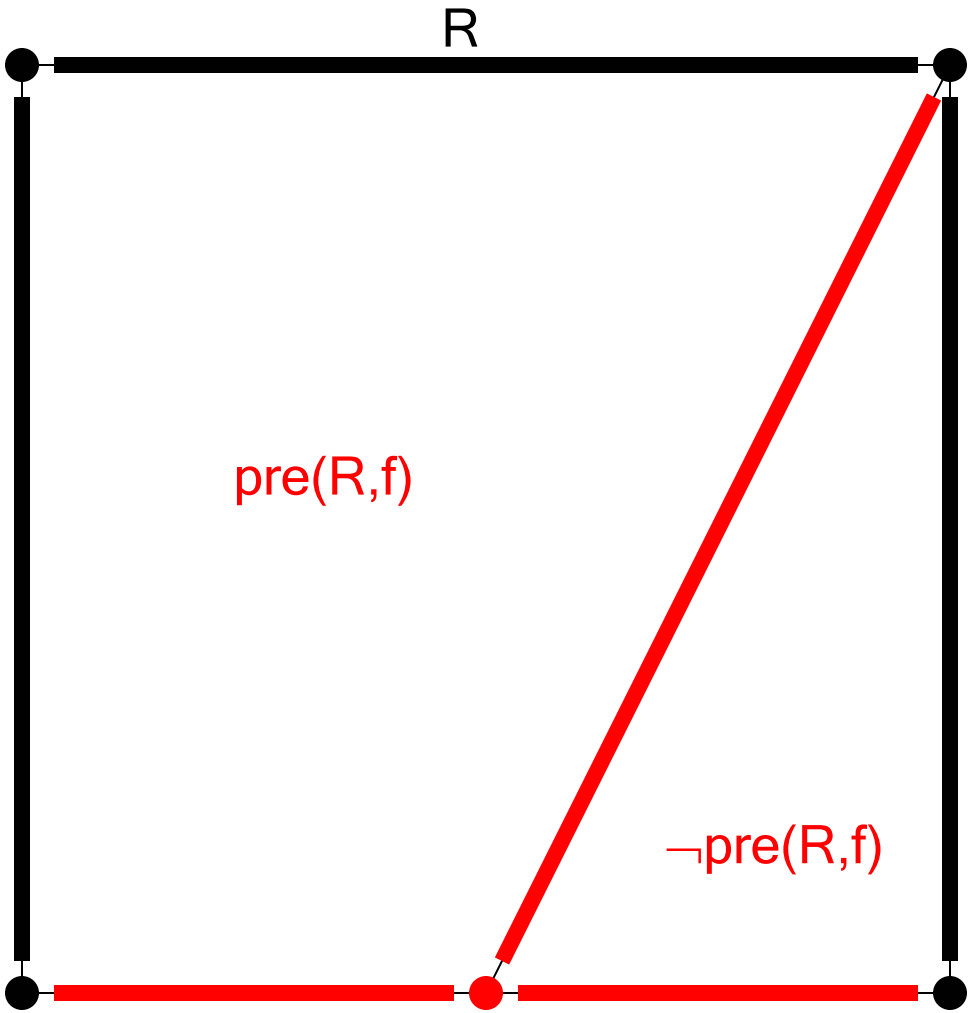
Infinite bisimulation.

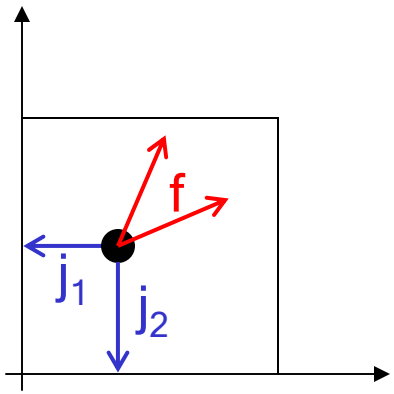


f: $x_1' \in [1,2]$; $x_2' \in [1,2]$

j_1: $x_1 := 0$

j_2: $x_2 := 0$

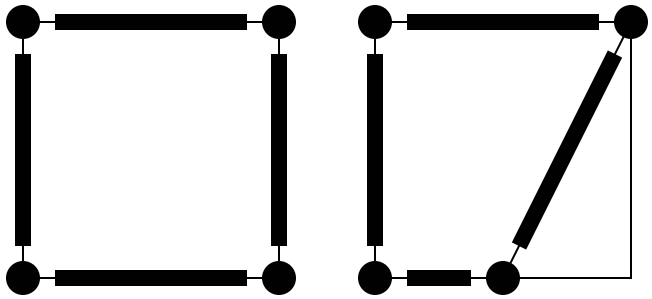


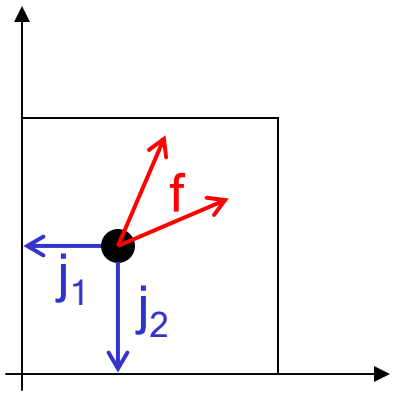


f: $x_1' \in [1,2]$; $x_2' \in [1,2]$

j_1: $x_1 := 0$

j_2: $x_2 := 0$

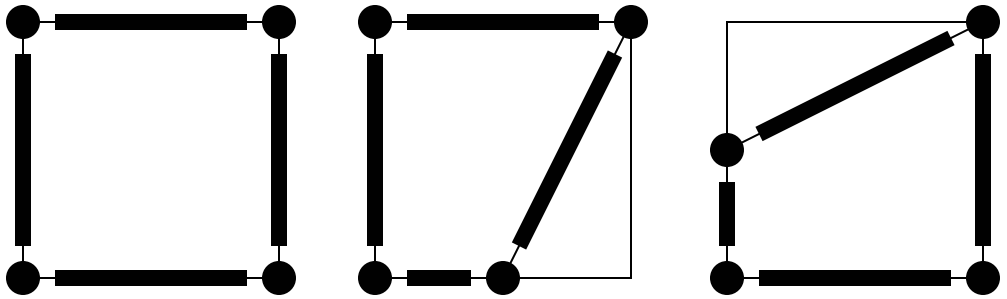


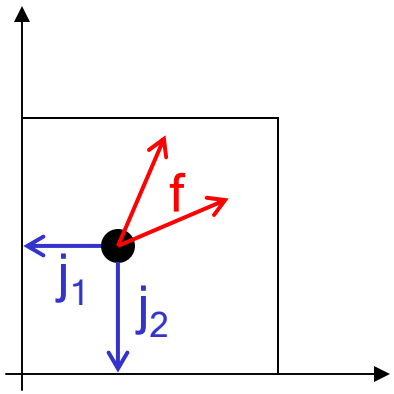


f: $x_1' \in [1,2]$; $x_2' \in [1,2]$

j_1: $x_1 := 0$

j_2: $x_2 := 0$

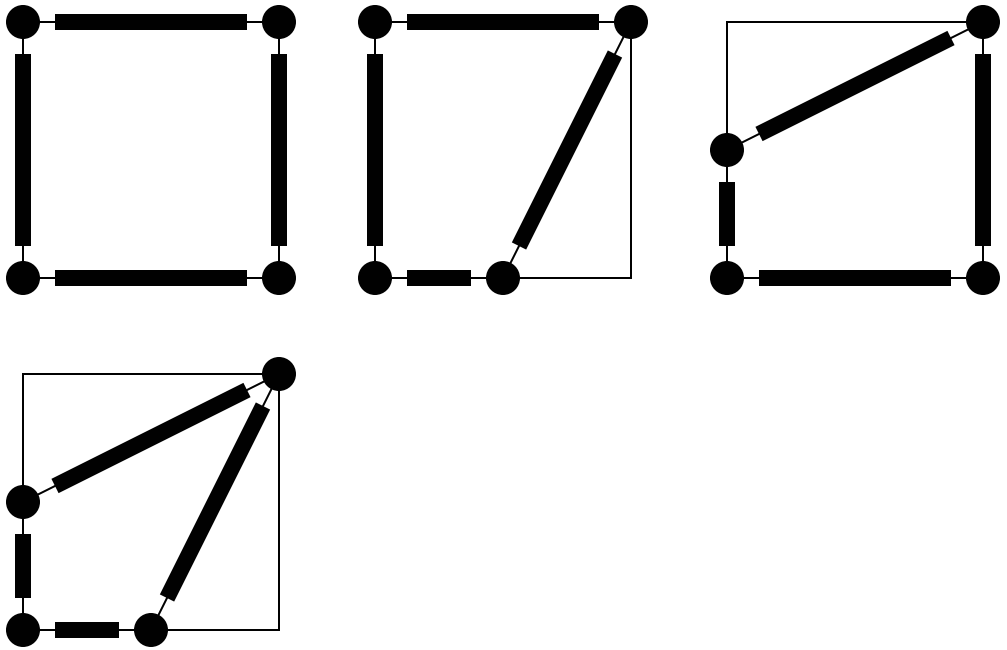


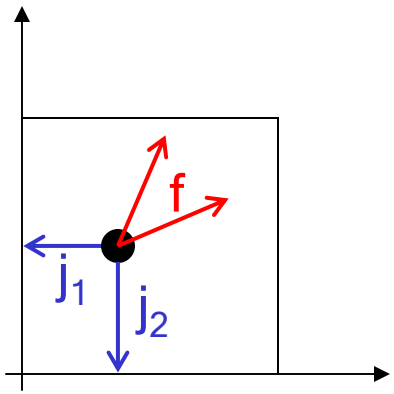


f: $x_1' \in [1,2]$; $x_2' \in [1,2]$

j_1: $x_1 := 0$

j_2: $x_2 := 0$

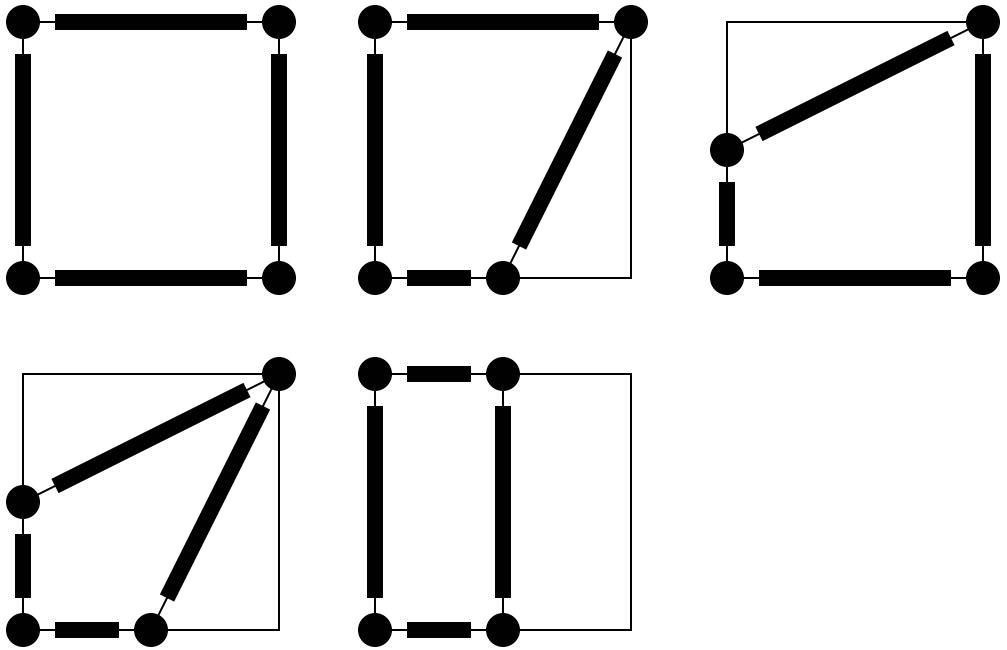


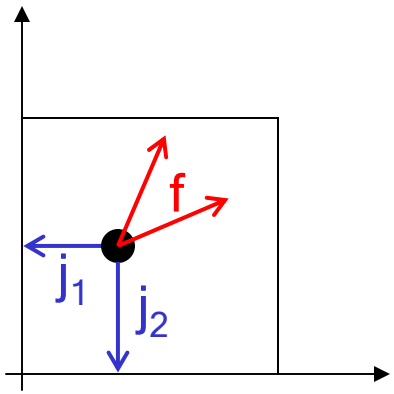


f: $x_1' \in [1,2]$; $x_2' \in [1,2]$

j_1: $x_1 := 0$

j_2: $x_2 := 0$

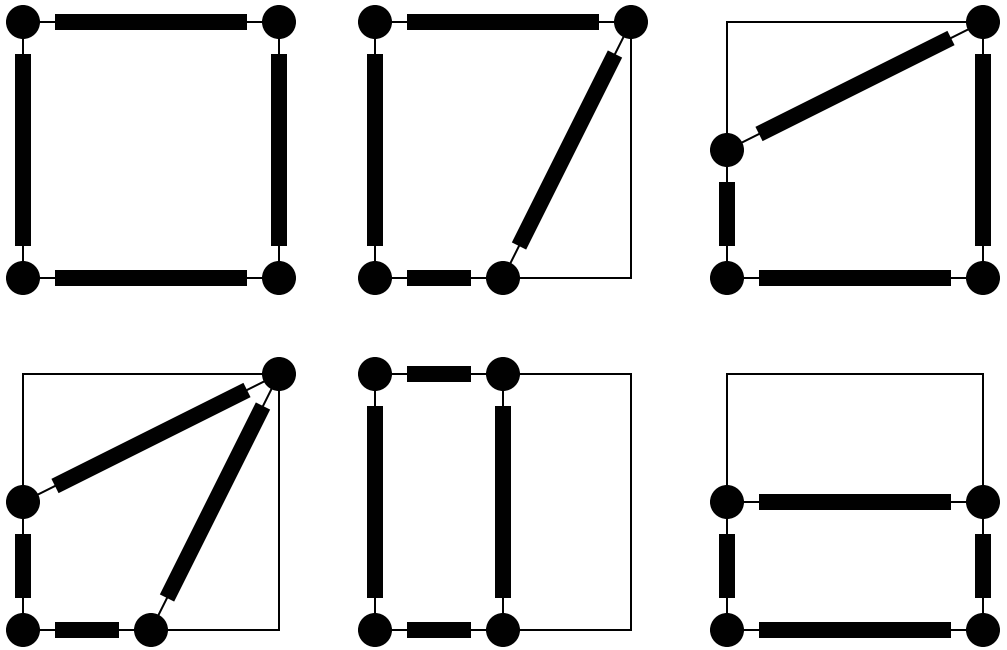


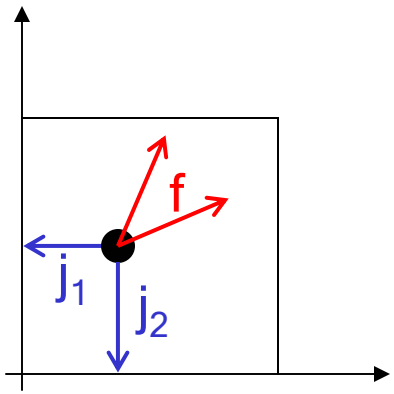


f: $x_1' \in [1,2]$; $x_2' \in [1,2]$

j_1: $x_1 := 0$

j_2: $x_2 := 0$

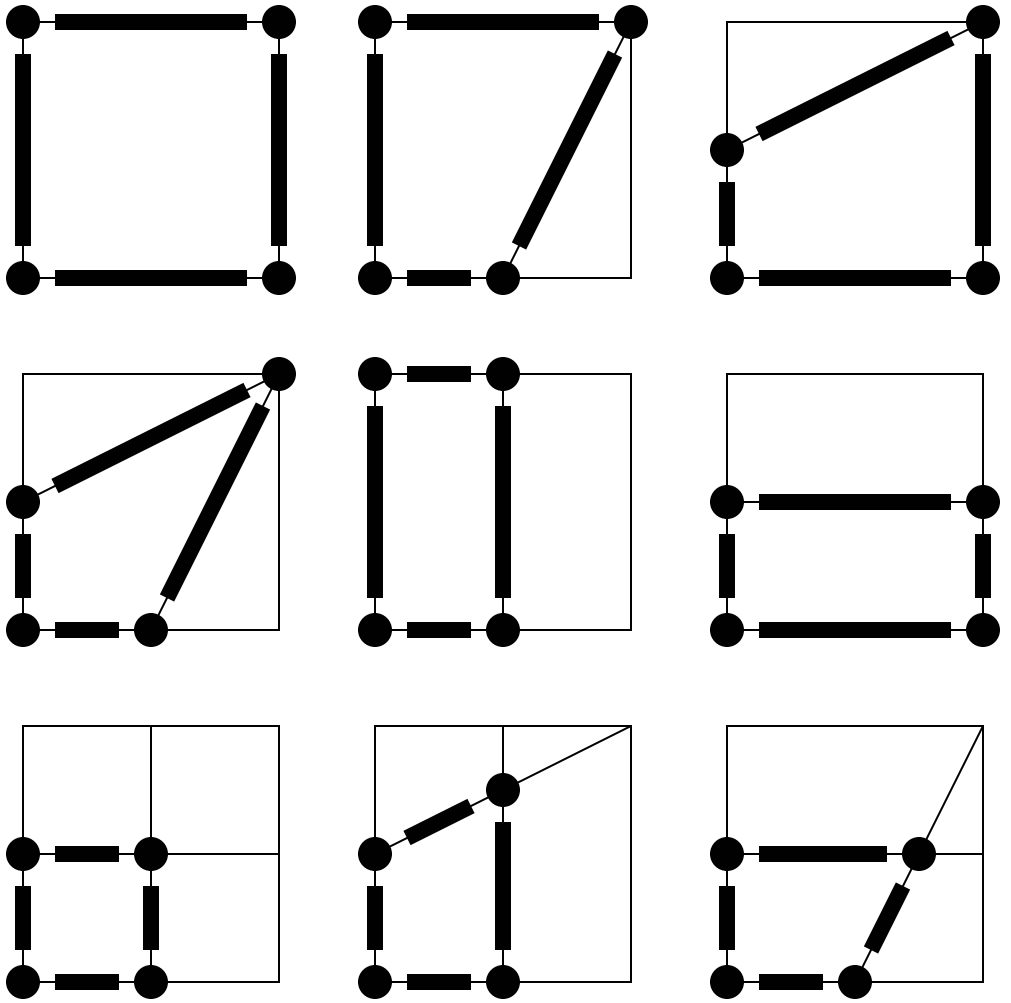


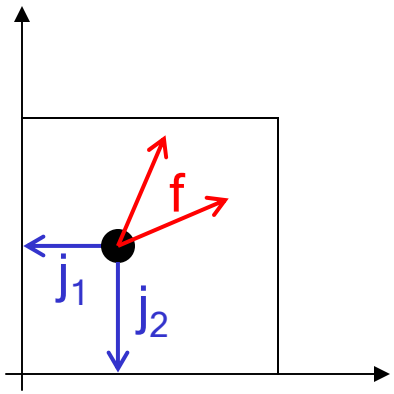


f: $x_1' \in [1,2]$; $x_2' \in [1,2]$

j_1: $x_1 := 0$

j_2: $x_2 := 0$

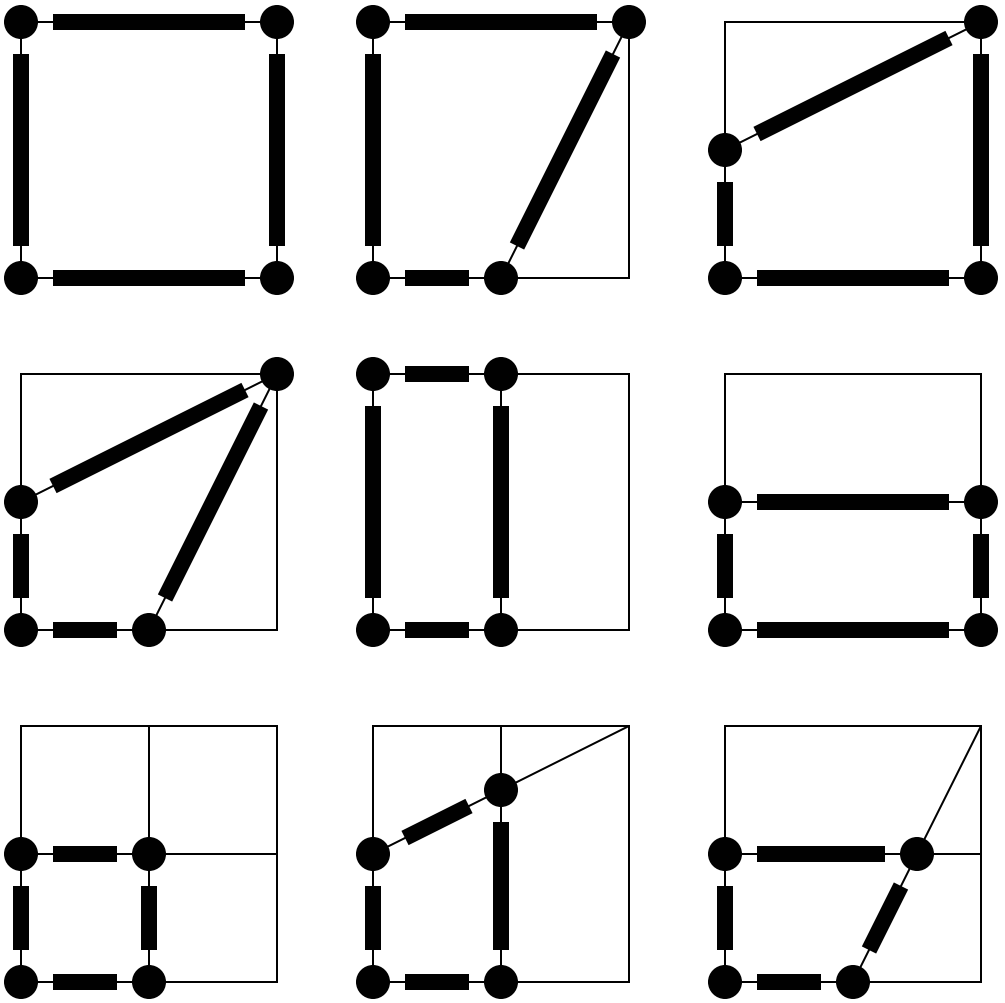


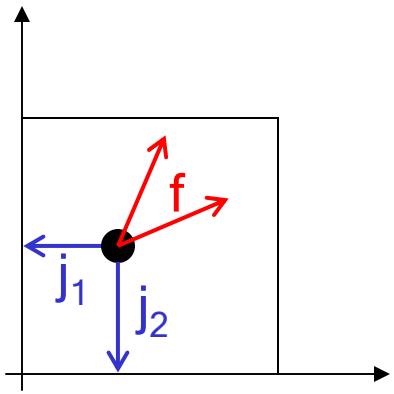


f: $x_1' \in [1,2]$; $x_2' \in [1,2]$

j_1: $x_1 := 0$

j_2: $x_2 := 0$

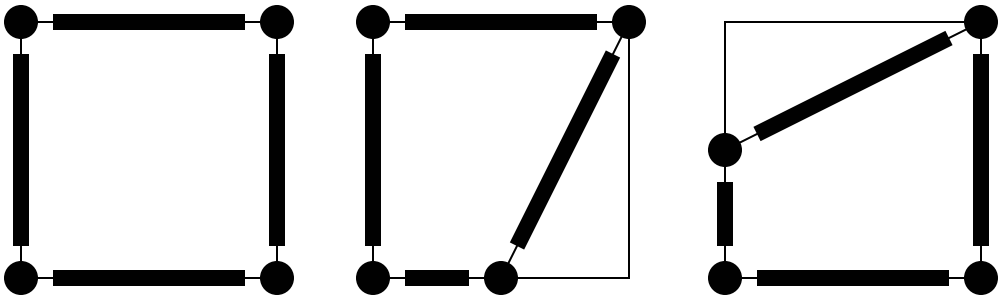


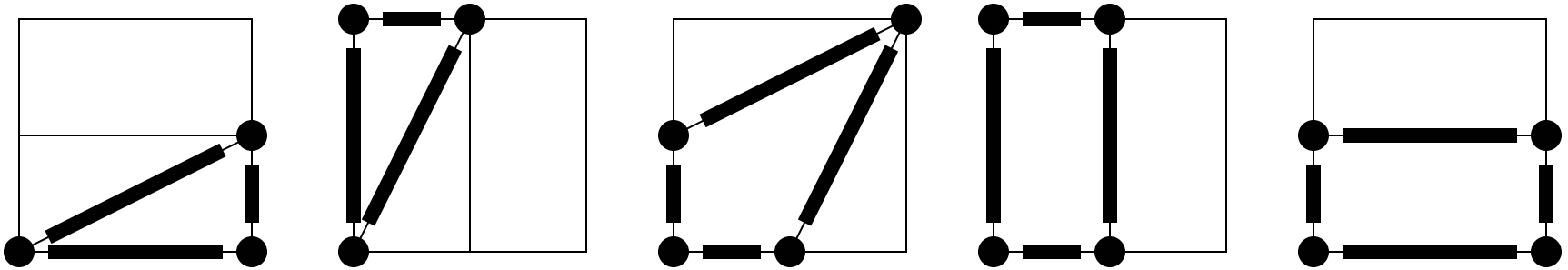


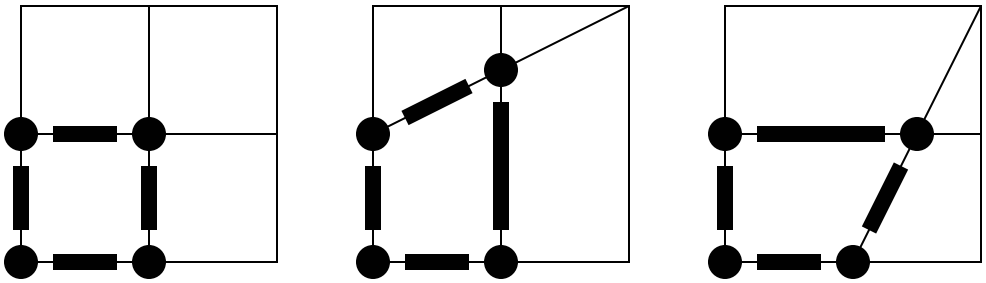
f: $x_1' \in [1,2]$; $x_2' \in [1,2]$

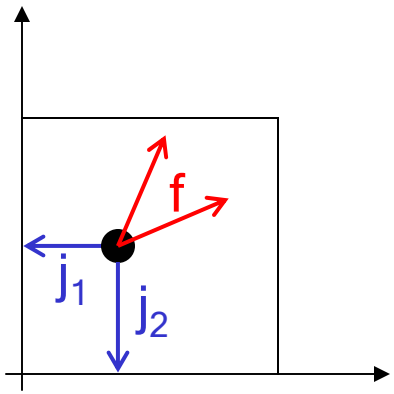
j_1: $x_1 := 0$

j_2: $x_2 := 0$





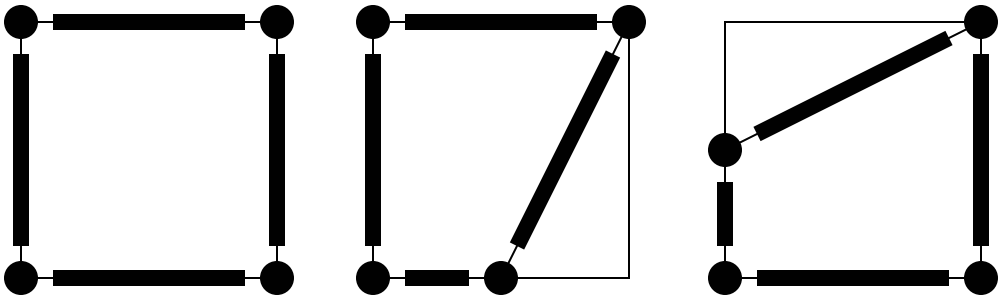


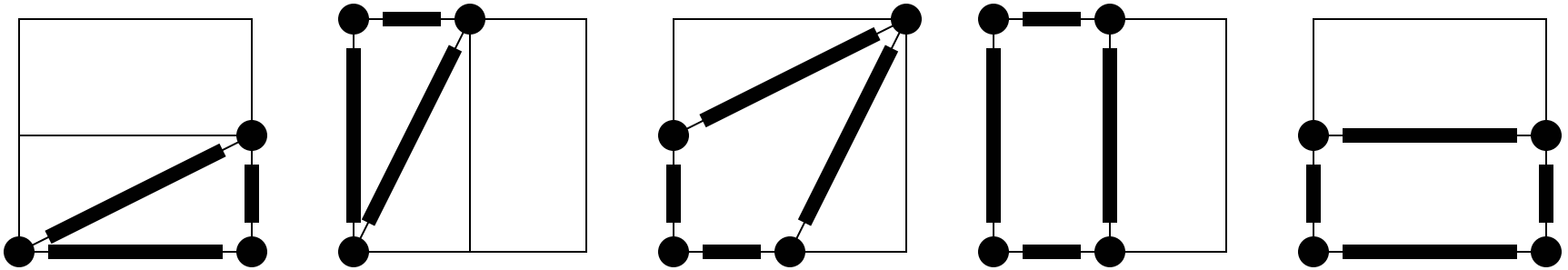


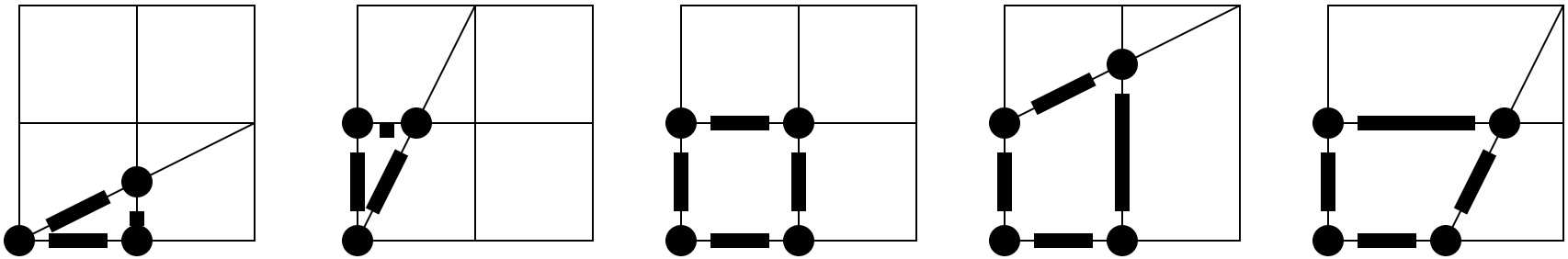
f: $x_1' \in [1,2]$; $x_2' \in [1,2]$

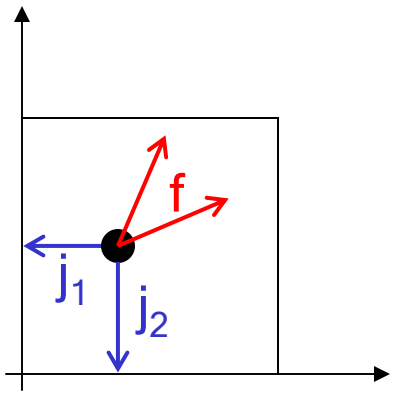
j_1: $x_1 := 0$

j_2: $x_2 := 0$





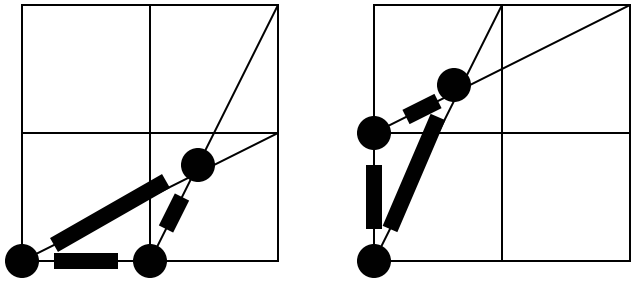


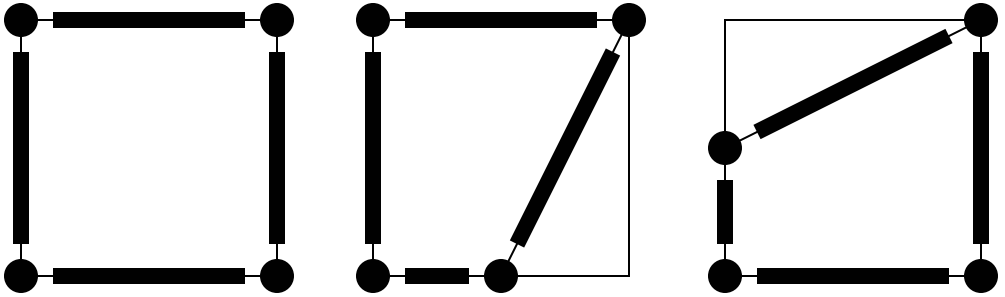


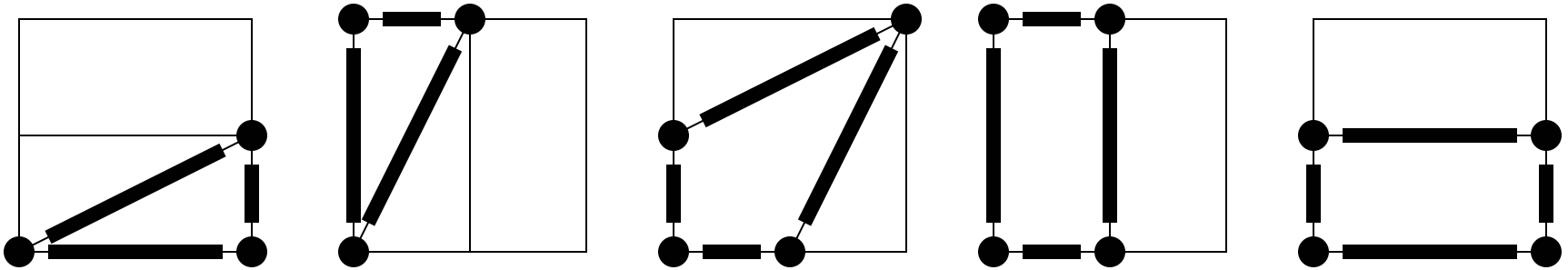
f: $x_1' \in [1,2]$; $x_2' \in [1,2]$

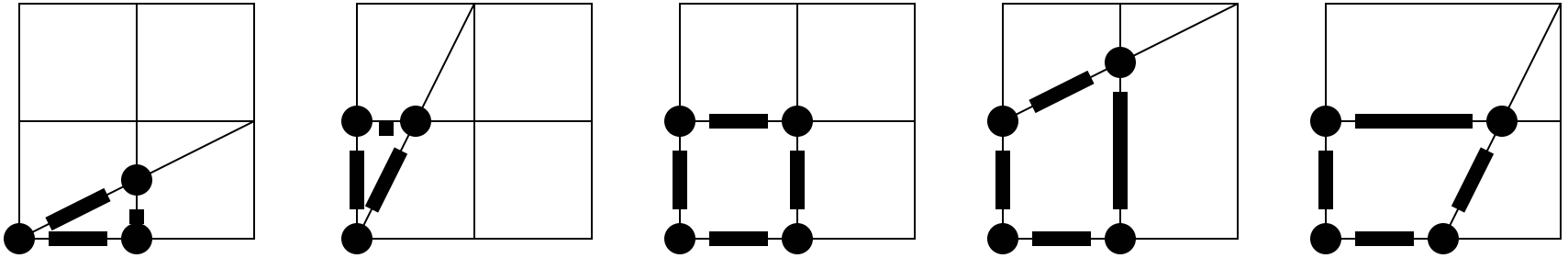
j_1: $x_1 := 0$

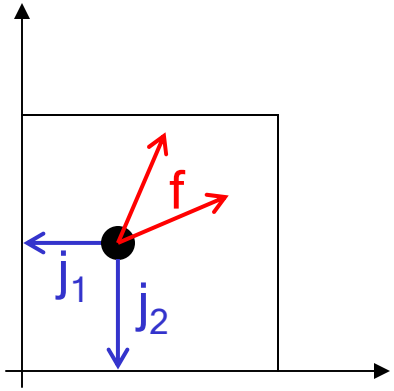
j_2: $x_2 := 0$







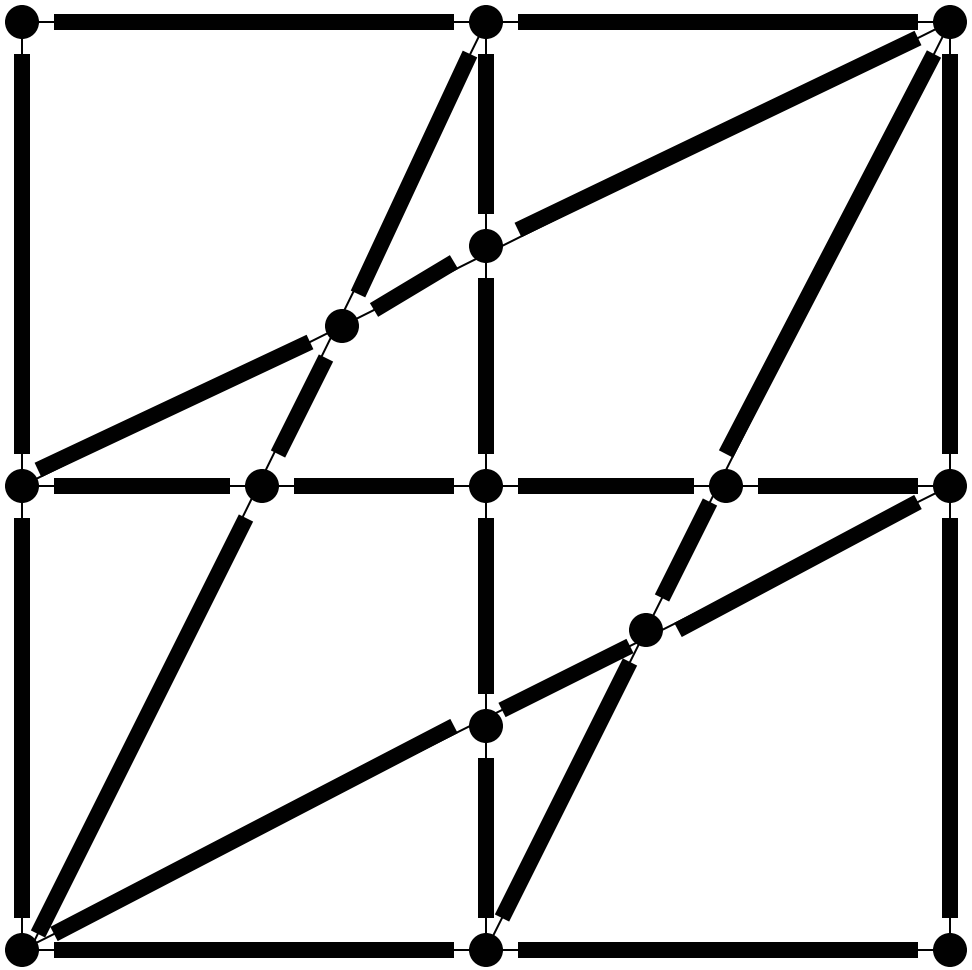


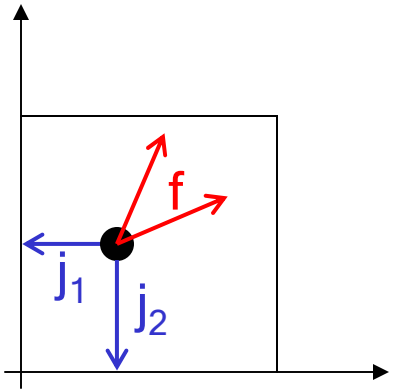


f: $x_1' \in [1,2]$; $x_2' \in [1,2]$

j_1: $x_1 := 0$

j_2: $x_2 := 0$

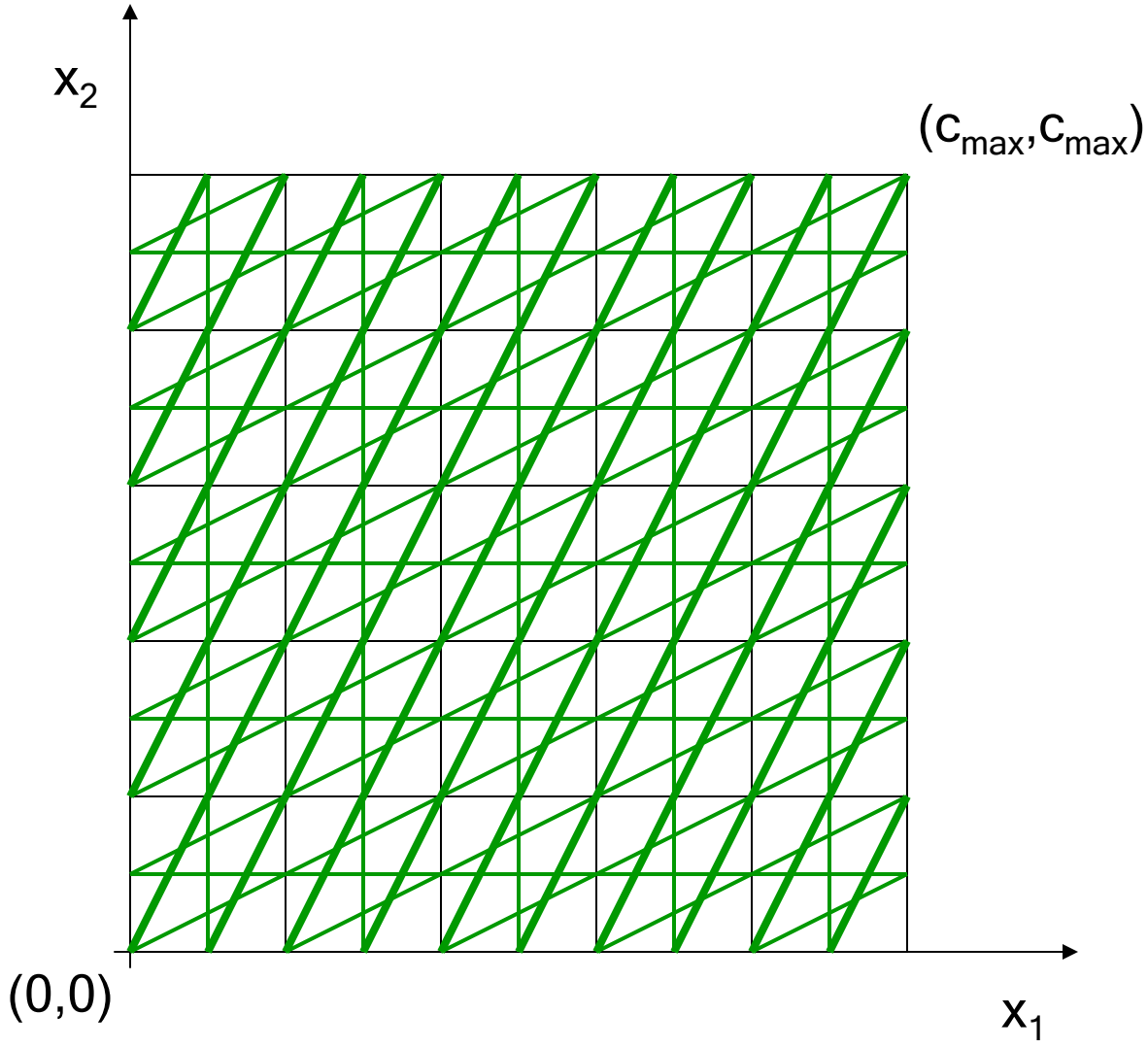




f: $x_1' \in [1,2]$; $x_2' \in [1,2]$

j_1: $x_1 := 0$

j_2: $x_2 := 0$



Finite simulation.

Summary

1. Separate local (region algebra) from global (symbolic semi-algorithms) concerns
2. Appeal to quantifier elimination for the computability of region operations (e.g. polyhedral hybrid systems)
3. Appeal to finite abstractions for the termination of symbolic semi-algorithms (e.g. rectangular hybrid systems)