Abstract Interpretation

Harry Xu

CS 253/INF 212

Spring 2013

Acknowledgements

Many slides in this file were taken from the tutorial slides that Patrick Cousot used in VMCAI'05

Abstract Interpretation

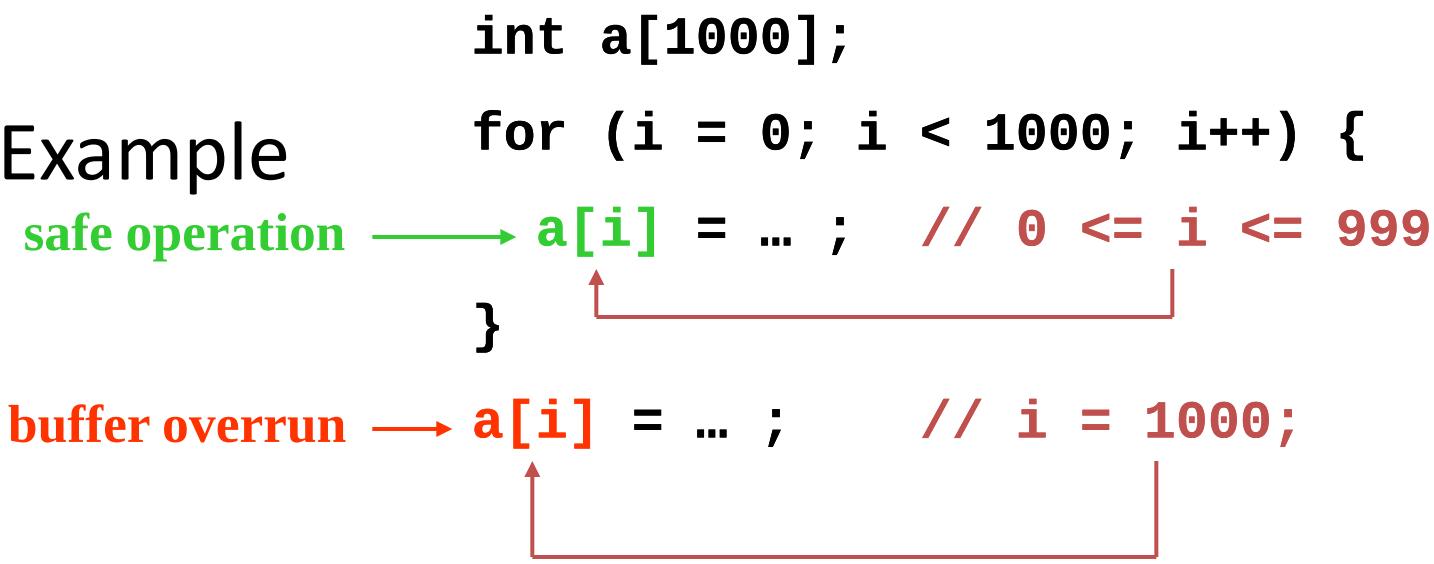
- A theory of sound approximation of the semantics of computer programs
- A partial execution of a program which gains information about its semantics (e.g., control-flow, data-flow) without performing all the calculations
- Establish a relationship between the concrete semantics and the abstract semantics

More Formally

- consists in considering an *abstract semantics*, that is to say a superset of the concrete semantics of the program;
- hence the abstract semantics covers all possible concrete cases;
- correct: if the abstract semantics is safe (does not intersect the forbidden zone) then so is the concrete semantics

Abstract Interpretation

- A methodology to derive sound static analysis with varying precision
  - Correct by construction
  - Generic
  - Easy to fine-tune
- Example



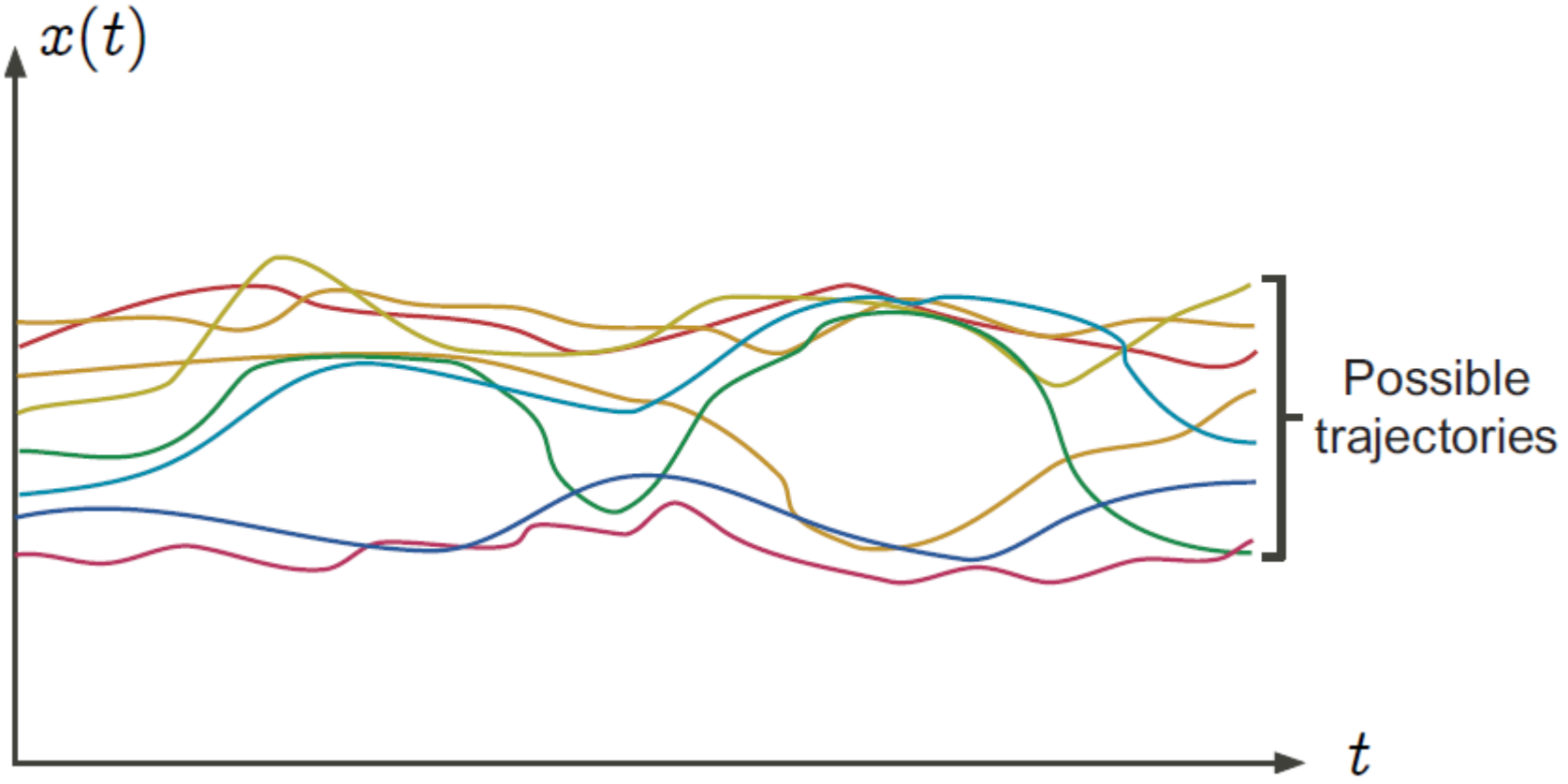
Overview

- Start with a formal specification of the program semantics (the concrete semantics)
- Construct abstract semantic equations w.r.t. a parametric approximation scheme
- Use general algorithms to solve the abstract semantic equations
- Try-and-test various instantiations of the approximation scheme in order to find the best fit

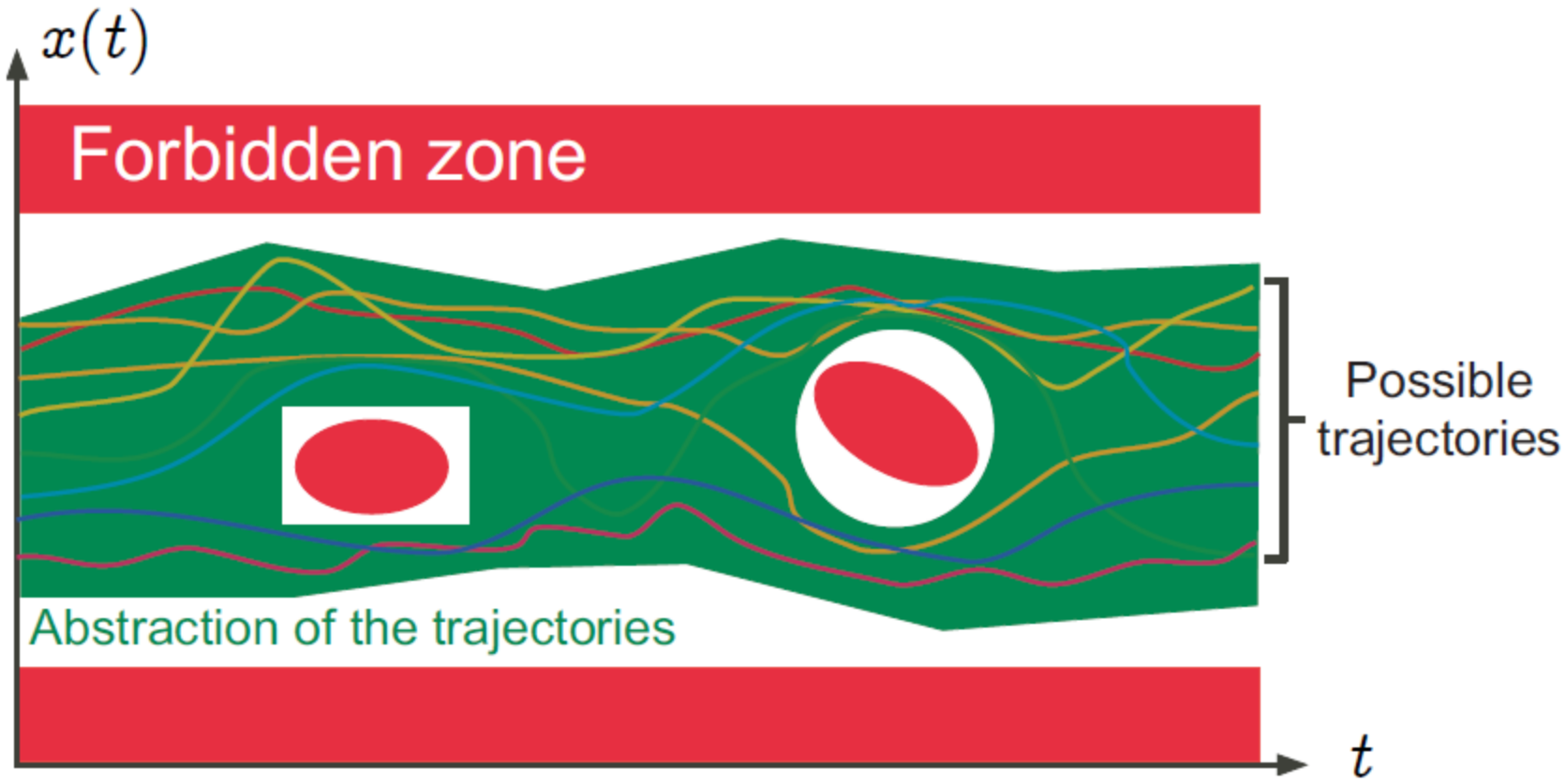
Semantics

The *concrete semantics* of a program formalizes (is a mathematical model of) the set of all its possible executions in all possible execution environments.

Graphic example: Possible behaviors



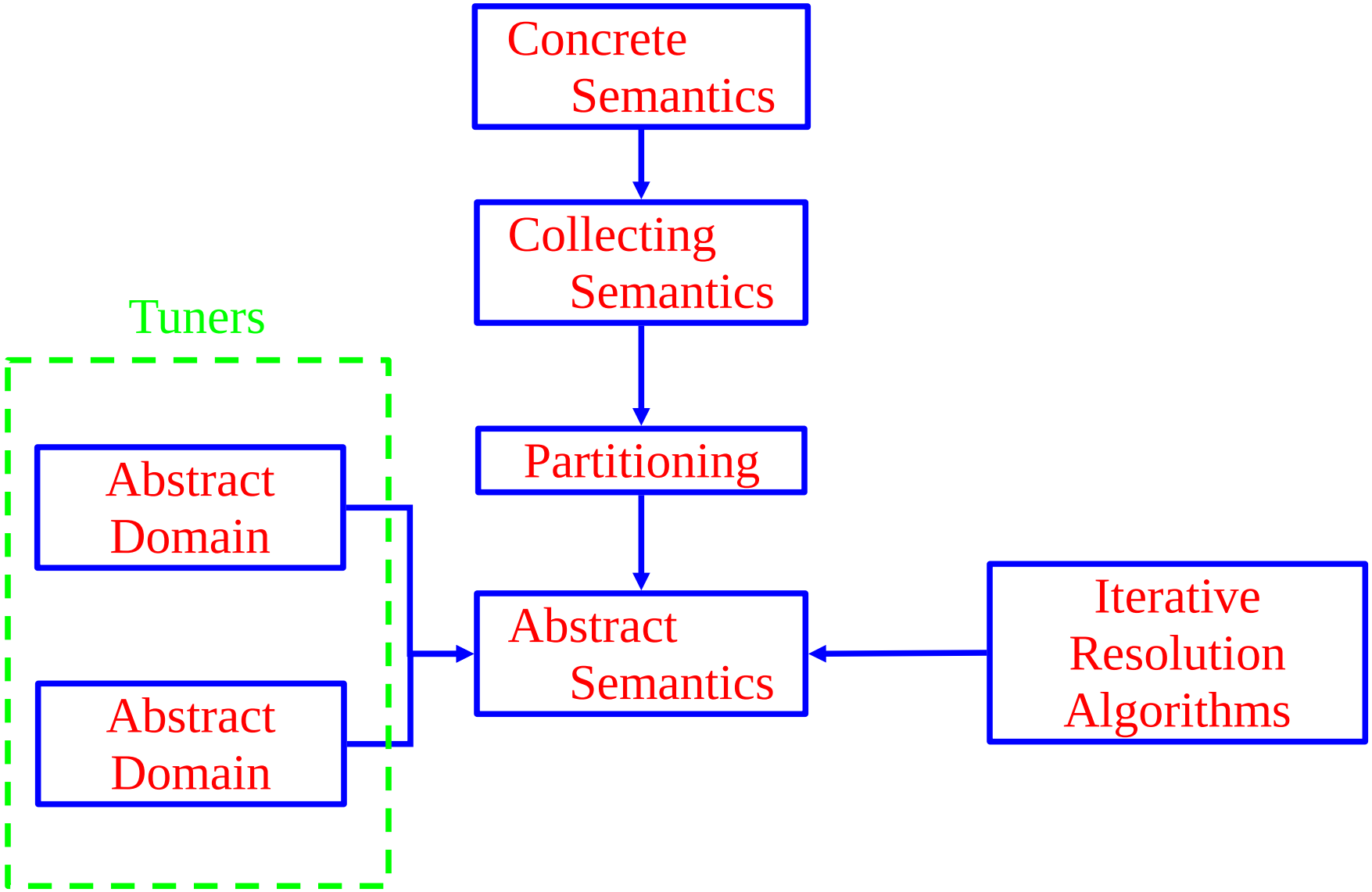
Graphic example: Abstract interpretation



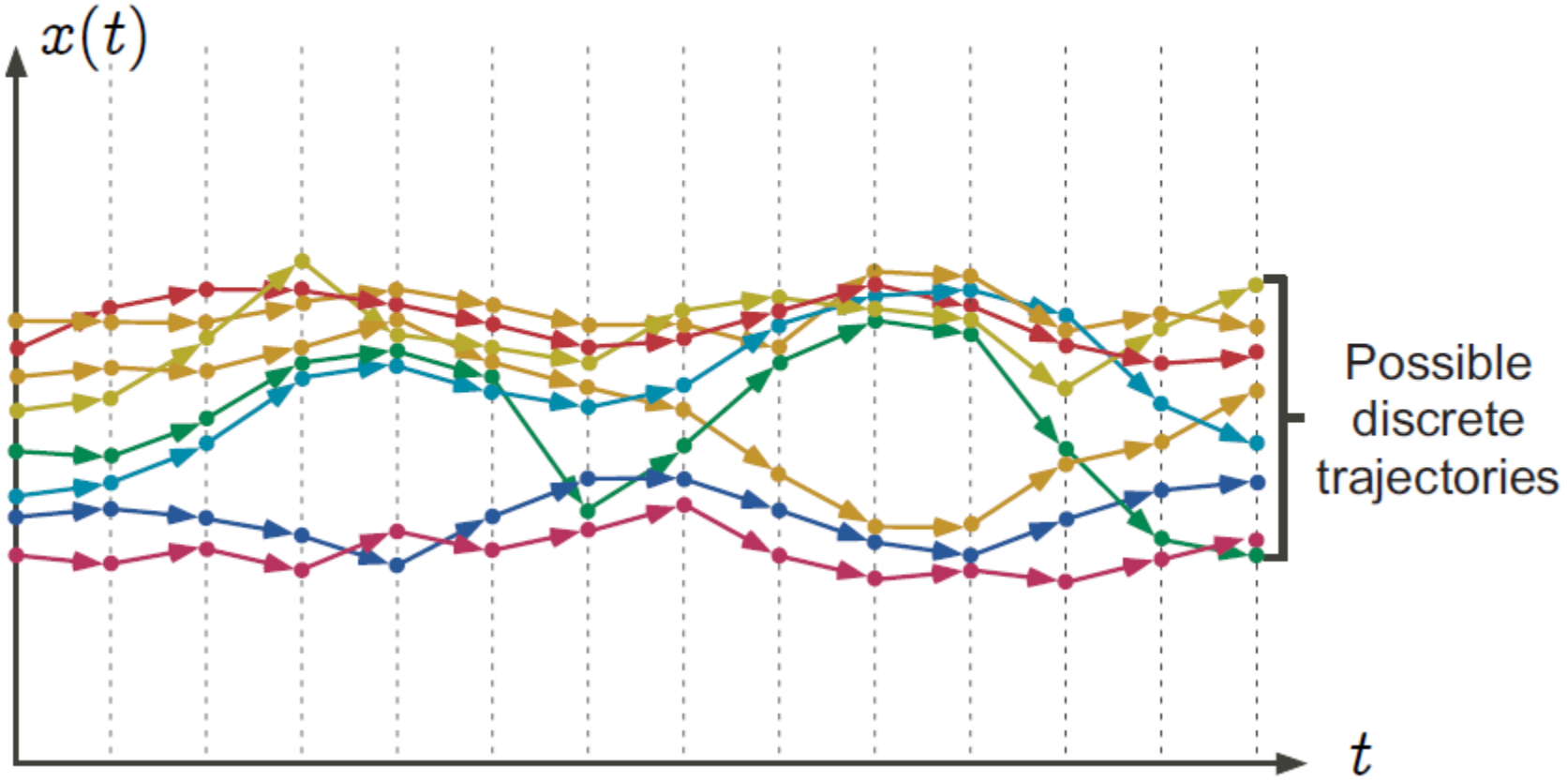
Standard semantics

- Start from a standard operational semantics that describes formally:
  - states that is data values of program variables,
  - transitions that is elementary computation steps;
- Consider traces that is successions of states corresponding to executions described by transitions (possibly infinite).

General Idea



Graphic example: Small-steps transition semantics



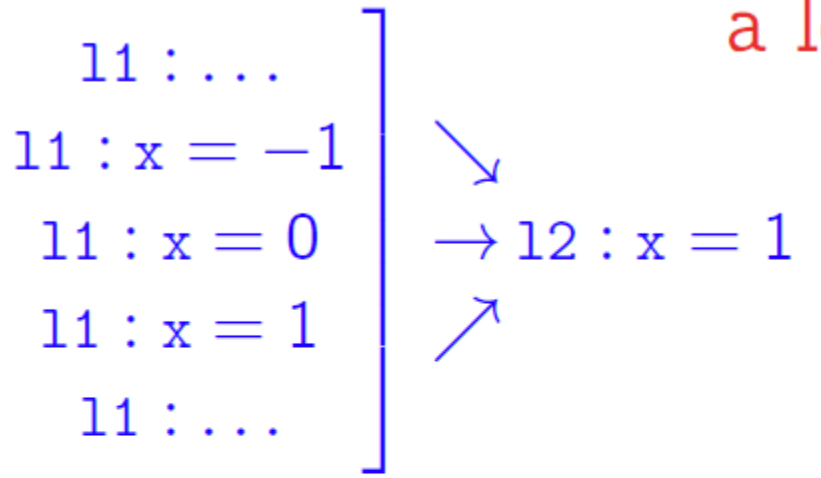
Example: Small-steps transition semantics of an assignment

```
int x;
...
l:
    x := x + 1;
l':
```

$$\{\mathtt{l} : \mathtt{x} = v \to \mathtt{l}' : \mathtt{x} = v + 1 \mid v \in [\mathtt{min\_int}, \mathtt{max\_int} - 1]\}$$
$$\cup \{\mathtt{l} : \mathtt{x} = \mathtt{max\_int} \to \mathtt{l}' : \mathtt{x} = \Omega\} \qquad \text{(runtime error)}$$

Example: Small-steps transition semantics of a loop

```
l1:
    x := 1;
l2:
    while x < 10 do
l3:
        x := x + 1
l4:
    od
l5:
```



$$l2 : x = 1 \rightarrow l3 : x = 1$$
$$l3 : x = 1 \rightarrow l4 : x = 2$$
$$l4 : x = 2 \rightarrow l3 : x = 2$$
$$l3 : x = 2 \rightarrow l4 : x = 3$$
$$\ldots$$
$$l4 : x = 10 \rightarrow l5 : x = 10$$

Example: Trace semantics of loop

```
l1:
    x := 1;
l2:
    while x < 10 do
l3:
        x := x + 1
l4:
    od
l5:
```

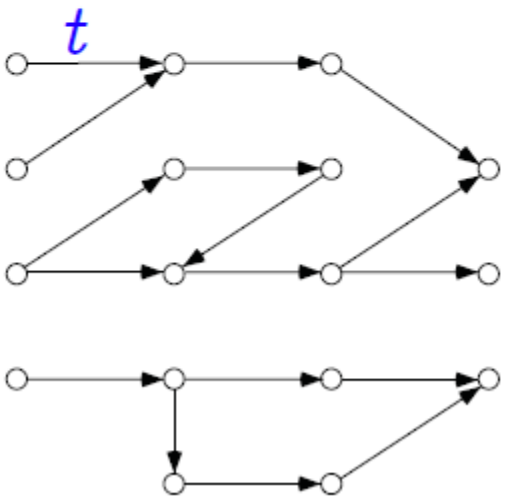
$$
\left.\begin{array}{l} \mathtt{l1}:\ldots \\ \mathtt{l1}:\mathtt{x}=-1 \\ \mathtt{l1}:\mathtt{x}=0 \\ \mathtt{l1}:\mathtt{x}=1 \\ \mathtt{l1}:\ldots \end{array}\right| \begin{array}{l} \searrow \\ \to \mathtt{l2}:\mathtt{x}=1 \to \mathtt{l3}:\mathtt{x}=1 \to \mathtt{l4}:\mathtt{x}=2 \to \\ \nearrow \end{array}
$$

$$
\mathtt{l3}:\mathtt{x}=2 \to \mathtt{l4}:\mathtt{x}=3\ldots \to \mathtt{l4}:\mathtt{x}=10 \to \mathtt{l5}:\mathtt{x}=10
$$

Transition systems

– $\langle S, \xrightarrow{t} \rangle$ where:

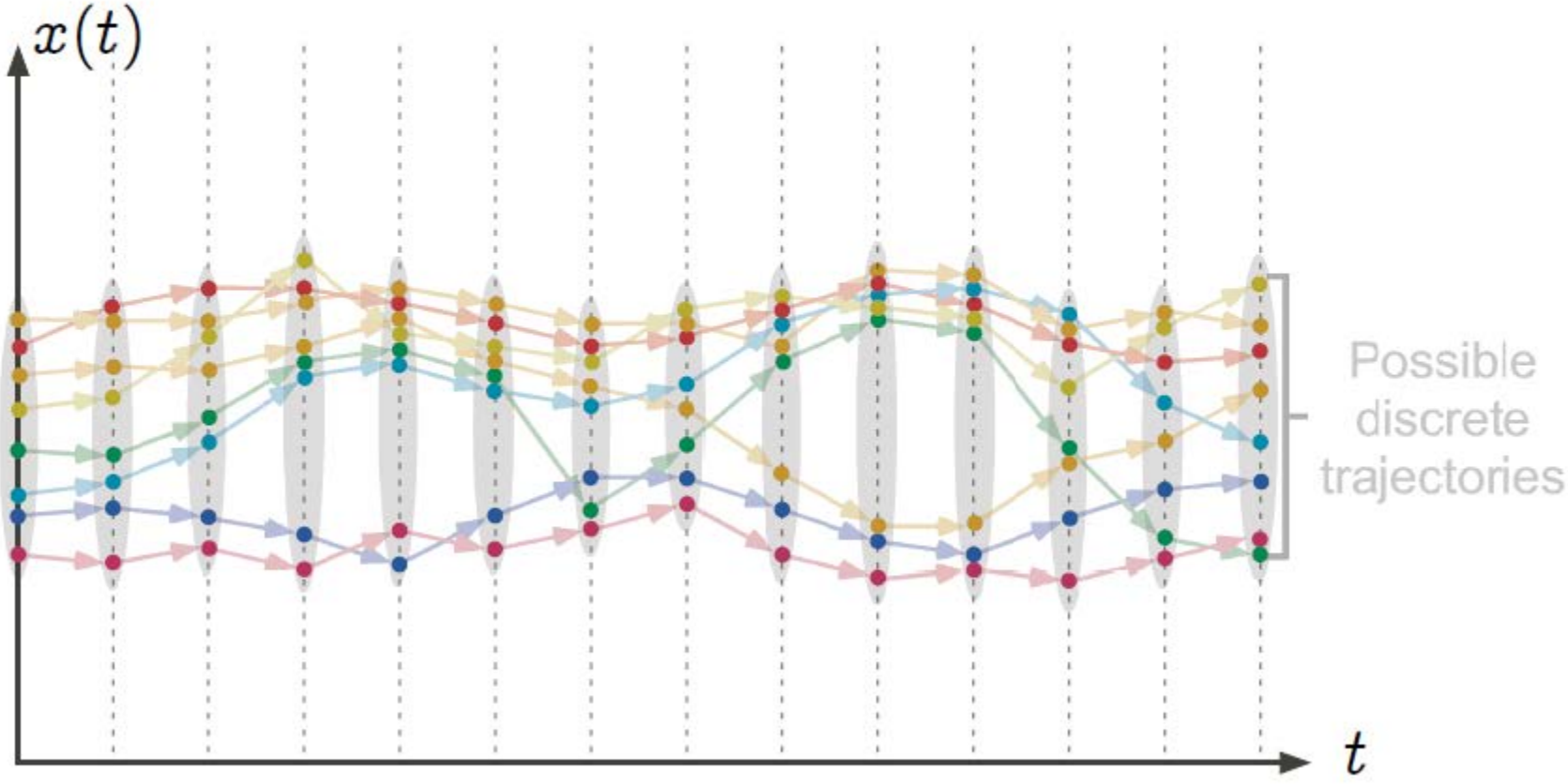
- S is a set of states/vertices/...
- $\xrightarrow{t} \in \wp(S \times S)$ is a transition relation/set of arcs/...



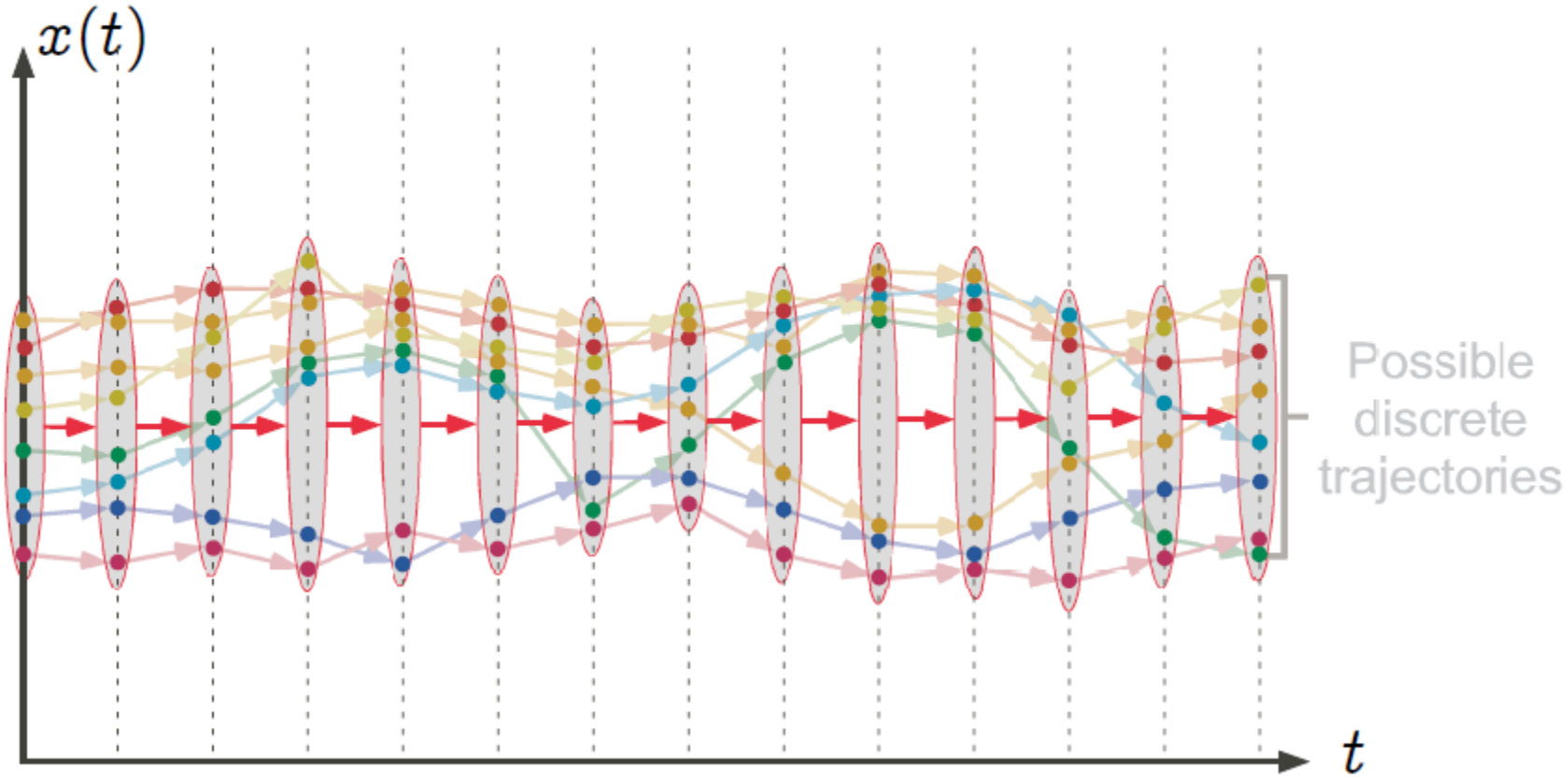
Collecting semantics

- consider all traces simultaneously;
- collecting semantics:
  - sets of states that describe data values of program variables on all possible trajectories;
  - set of states transitions that is simultaneous elementary computation steps on all possible trajectories;

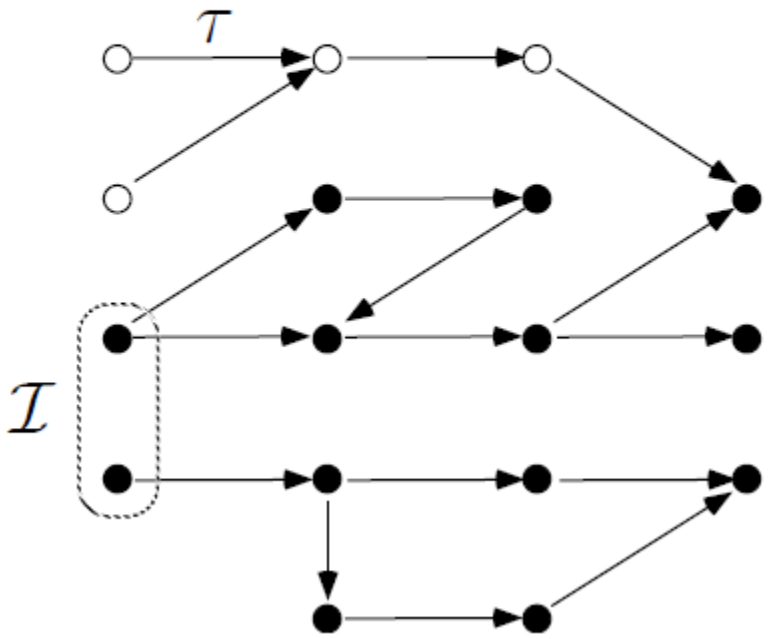
Graphic example: sets of states



Graphic example: set of states transitions



Example: Reachable states of a transition system



Reachable states in fixpoint form

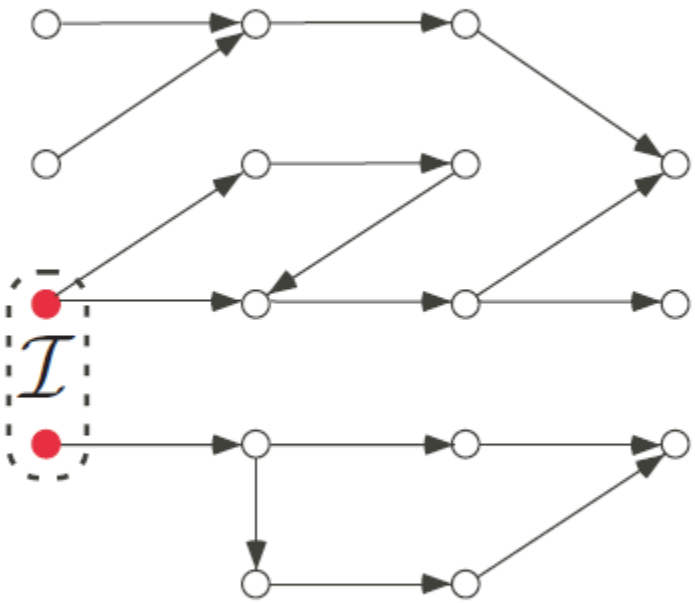
$$F(X) = \mathcal{I} \cup \{s' \mid \exists s \in X : s \xrightarrow{t} s'\}$$

$$\mathcal{R} = \mathsf{lfp}_{\emptyset}^{\subseteq} F$$

$$= \bigcup_{n=0}^{+\infty} F^n(\emptyset) \qquad \text{where} \quad \begin{aligned} f^0(x) &= x \\ f^{n+1}(x) &= f(f^n(x)) \end{aligned}$$

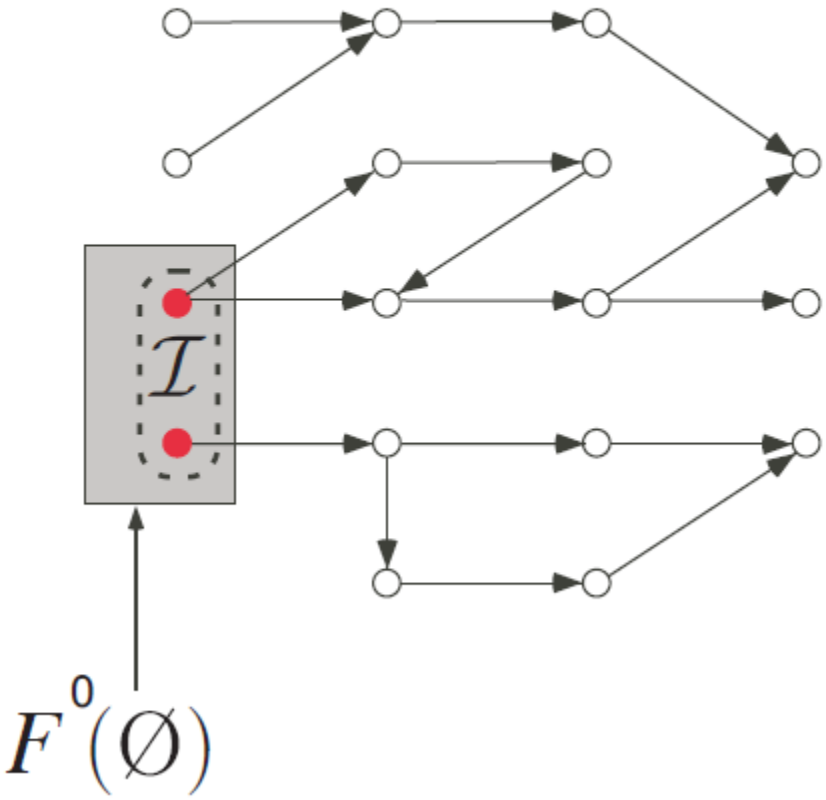
Example of fixpoint iteration

for reachable states $\mathsf{lfp}_{\emptyset}^{\subseteq} \lambda X \,.\, \mathcal{I} \cup \{s' \mid \exists s \in X : s \xrightarrow{t} s'\}$



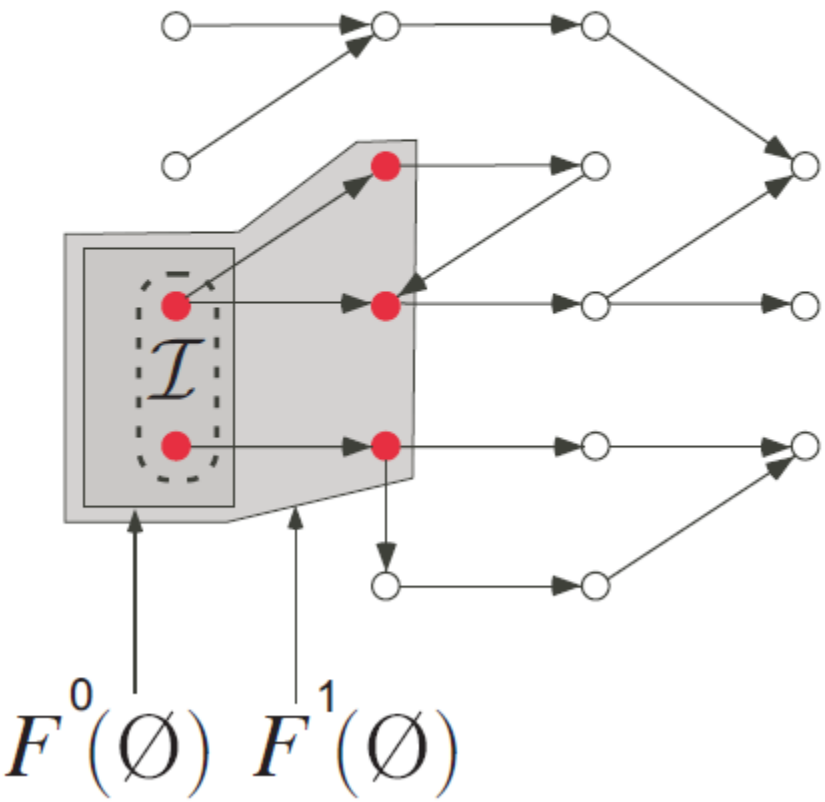
$\emptyset$

Example of fixpoint iteration for reachable states $\mathsf{lfp}^{\subseteq}_{\emptyset}\, \lambda X \cdot \mathcal{I} \cup \{s' \mid \exists s \in X : s \xrightarrow{t} s'\}$

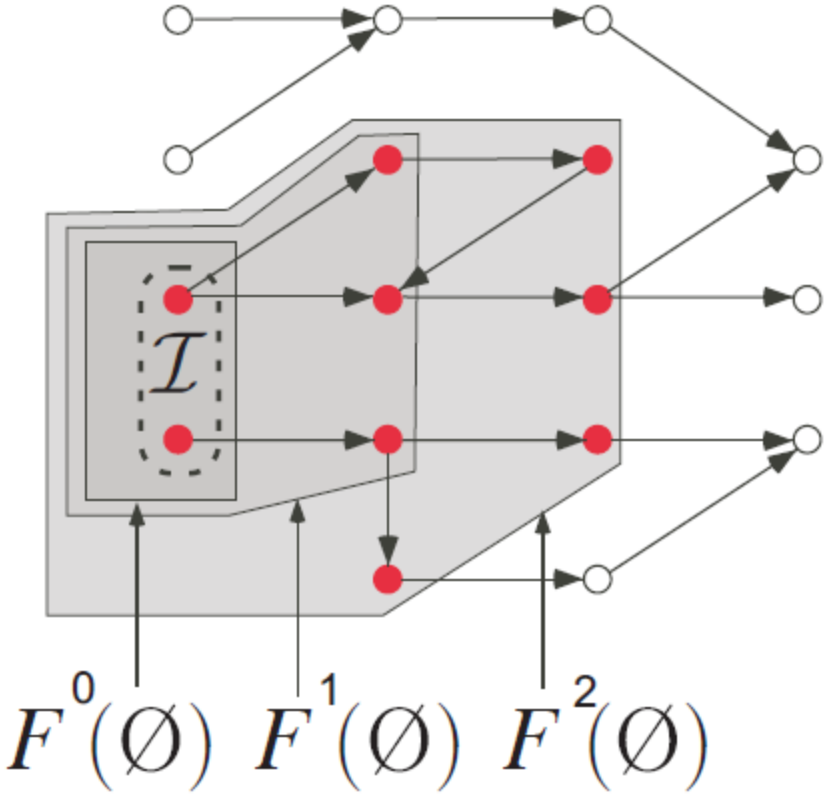


Example of fixpoint iteration

for reachable states $\mathsf{lfp}_{\emptyset}^{\subseteq} \lambda X . \mathcal{I} \cup \{s' \mid \exists s \in X : s \xrightarrow{t} s'\}$

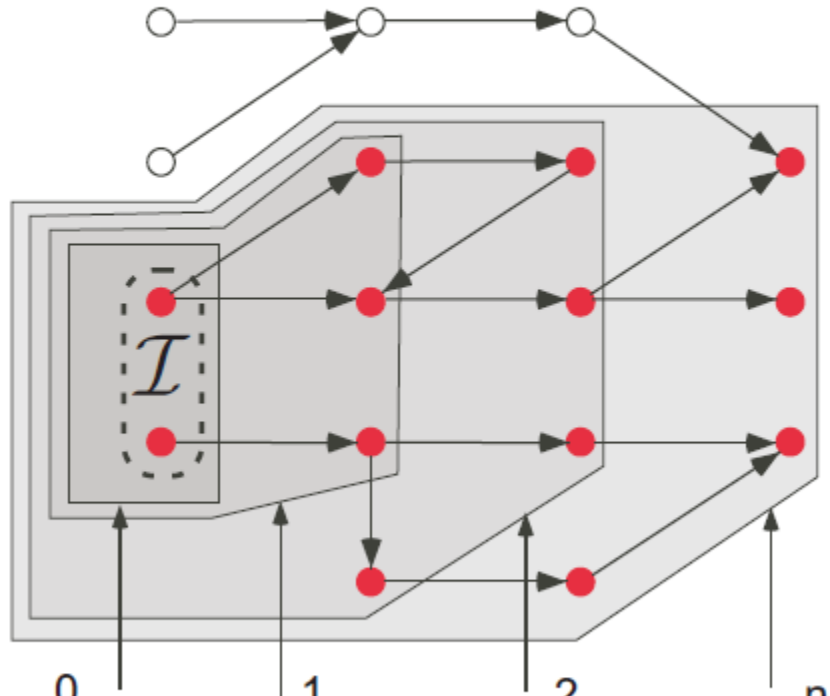


Example of fixpoint iteration for reachable states $\mathsf{lfp}_{\emptyset}^{\subseteq} \lambda X \cdot \mathcal{I} \cup \{s' \mid \exists s \in X : s \xrightarrow{t} s'\}$



Example of fixpoint iteration

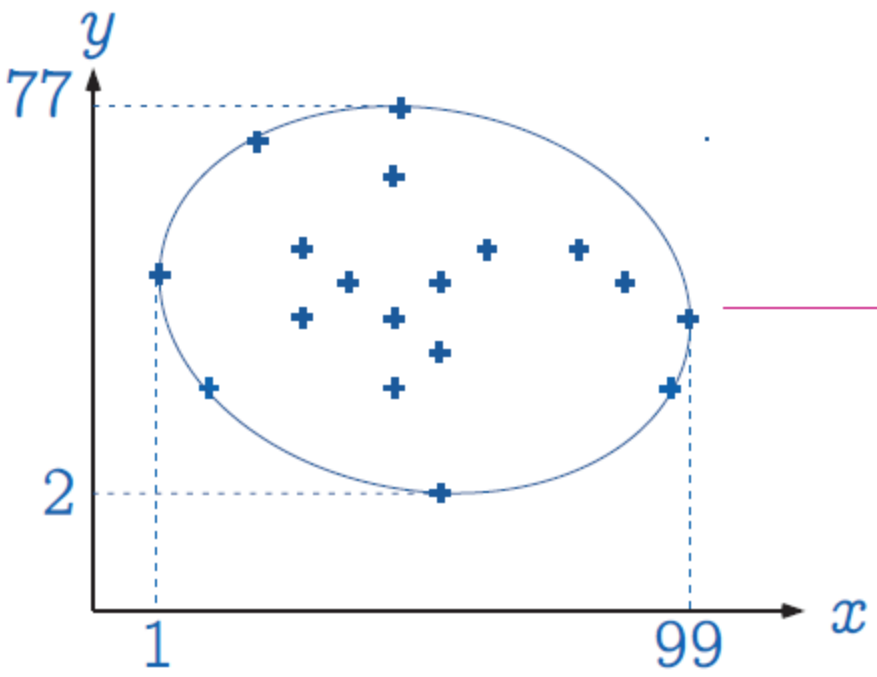
for reachable states $\mathsf{lfp}_{\emptyset}^{\subseteq} \lambda X \cdot \mathcal{I} \cup \{s' \mid \exists s \in X : s \xrightarrow{t} s'\}$



Abstracting sets (i.e. properties)

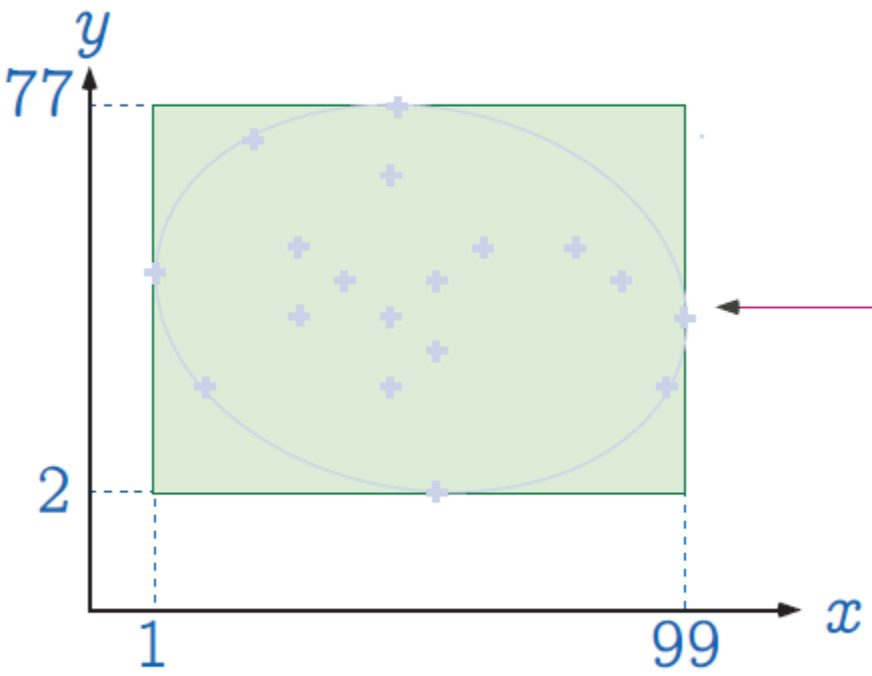
- Choose an abstract domain, replacing sets of objects (states, traces, ...) S by their abstraction $\alpha(S)$
- The abstraction function α maps a set of concrete objects to its abstract interpretation;
- The inverse concretization function γ maps an abstract set of objects to concrete ones;
- Forget no concrete objects: (abstraction from above) $S \subseteq \gamma(\alpha(S))$.

Interval abstraction α



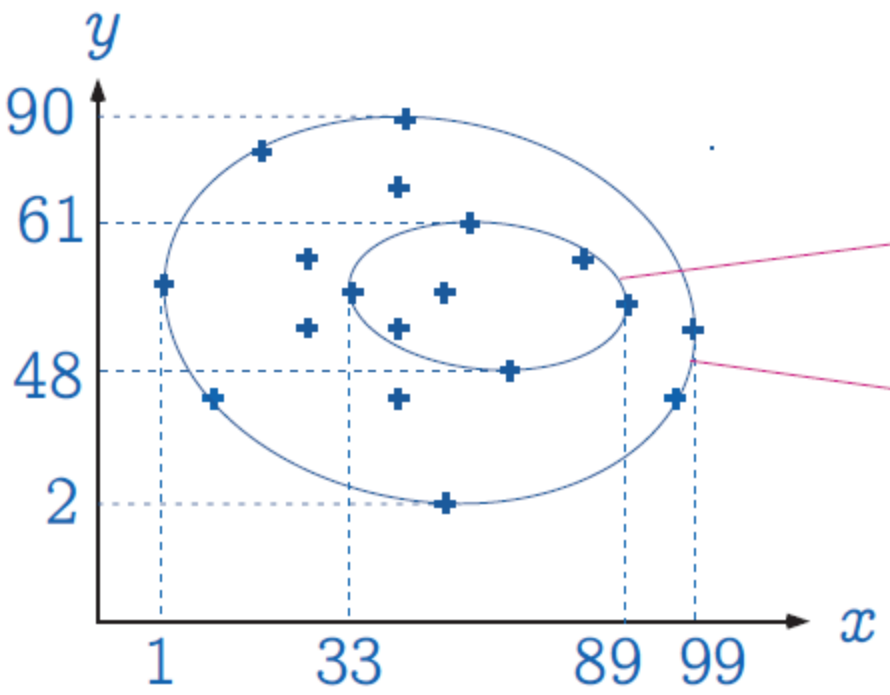
$\{x : [1, 99], y : [2, 77]\}$

Interval concretization γ



$\{x : [1, 99], y : [2, 77]\}$

The abstraction α is monotone



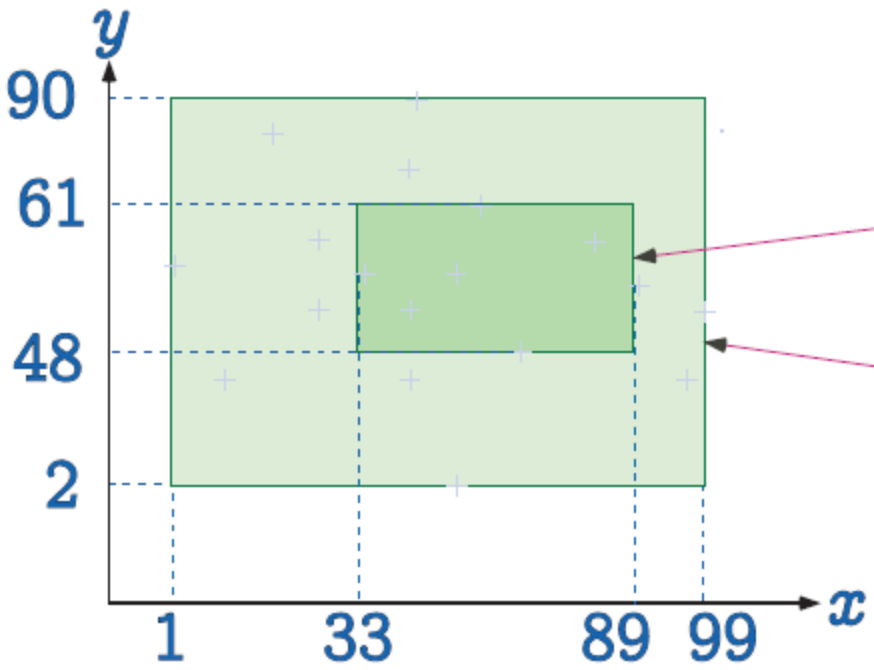
$\{x : [33, 89], y : [48, 61]\}$

$\sqsubseteq$

$\{x : [1, 99], y : [2, 90]\}$

$$X \subseteq Y \Rightarrow \alpha(X) \sqsubseteq \alpha(Y)$$

The concretization γ is monotone



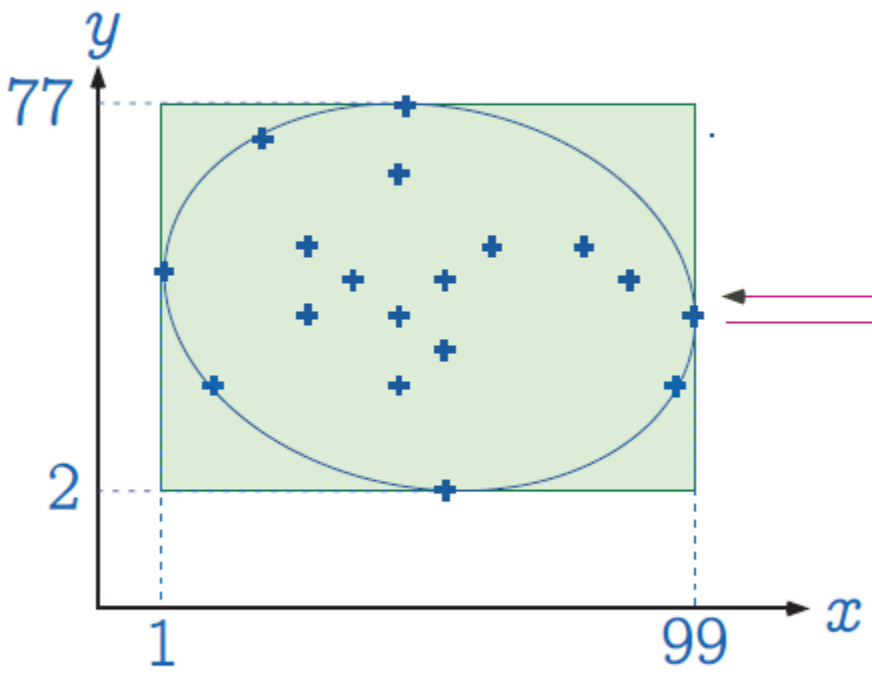
$\{x : [33, 89], y : [48, 61]\}$

$\sqsubseteq$

$\{x : [1, 99], y : [2, 90]\}$

$$X \sqsubseteq Y \Rightarrow \gamma(X) \subseteq \gamma(Y)$$

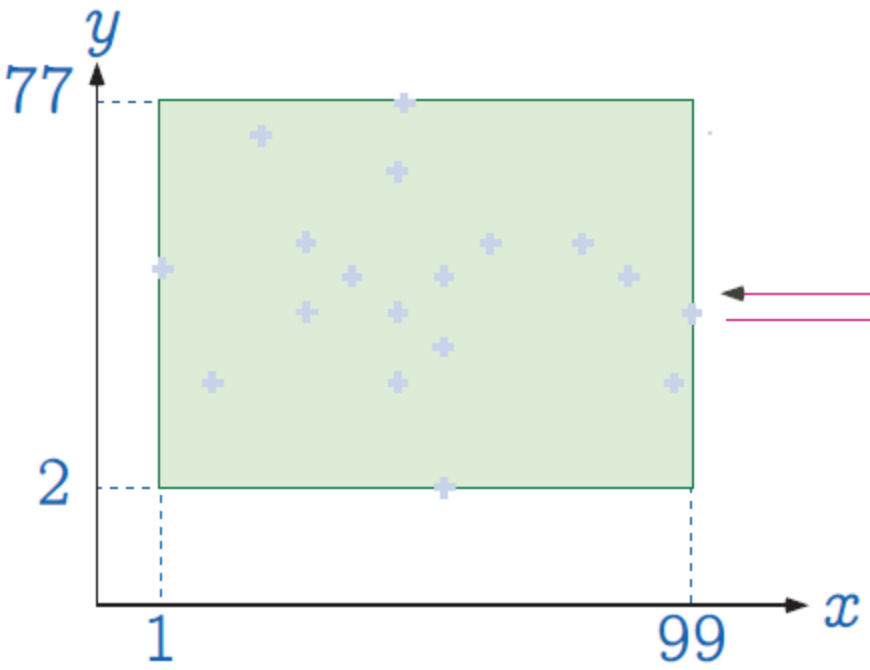
The $\gamma \circ \alpha$ composition is extensive



$\{x : [1, 99], y : [2, 77]\}$

$X \subseteq \gamma \circ \alpha(X)$

The $\alpha \circ \gamma$ composition is reductive



$$\{x : [1, 99], y : [2, 77]\}$$
$$=/\sqsubseteq$$
$$\{x : [1, 99], y : [2, 77]\}$$

$$\alpha \circ \gamma(Y) =/\sqsubseteq Y$$

Which Collecting Semantics?

- Buffer overrun, division by zero, arithmetic overflows: state properties
- Deadlocks, un-initialized variables: finite trace properties
- Loop termination: finite and infinite trace properties

Correspondance between concrete and abstract properties

– The pair $\langle \alpha, \gamma \rangle$ is a Galois connection:

$$\langle \wp(S), \subseteq \rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle \mathcal{D}, \sqsubseteq \rangle$$

– $\langle \wp(S), \subseteq \rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle \mathcal{D}, \sqsubseteq \rangle$ when α is onto (equivalently $\alpha \circ \gamma = 1$ or γ is one-to-one).

Galois connection

$$\langle \mathcal{D}, \subseteq \rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle \overline{\mathcal{D}}, \sqsubseteq \rangle$$

iff $\quad \forall x, y \in \mathcal{D} : x \subseteq y \Longrightarrow \alpha(x) \sqsubseteq \alpha(y)$

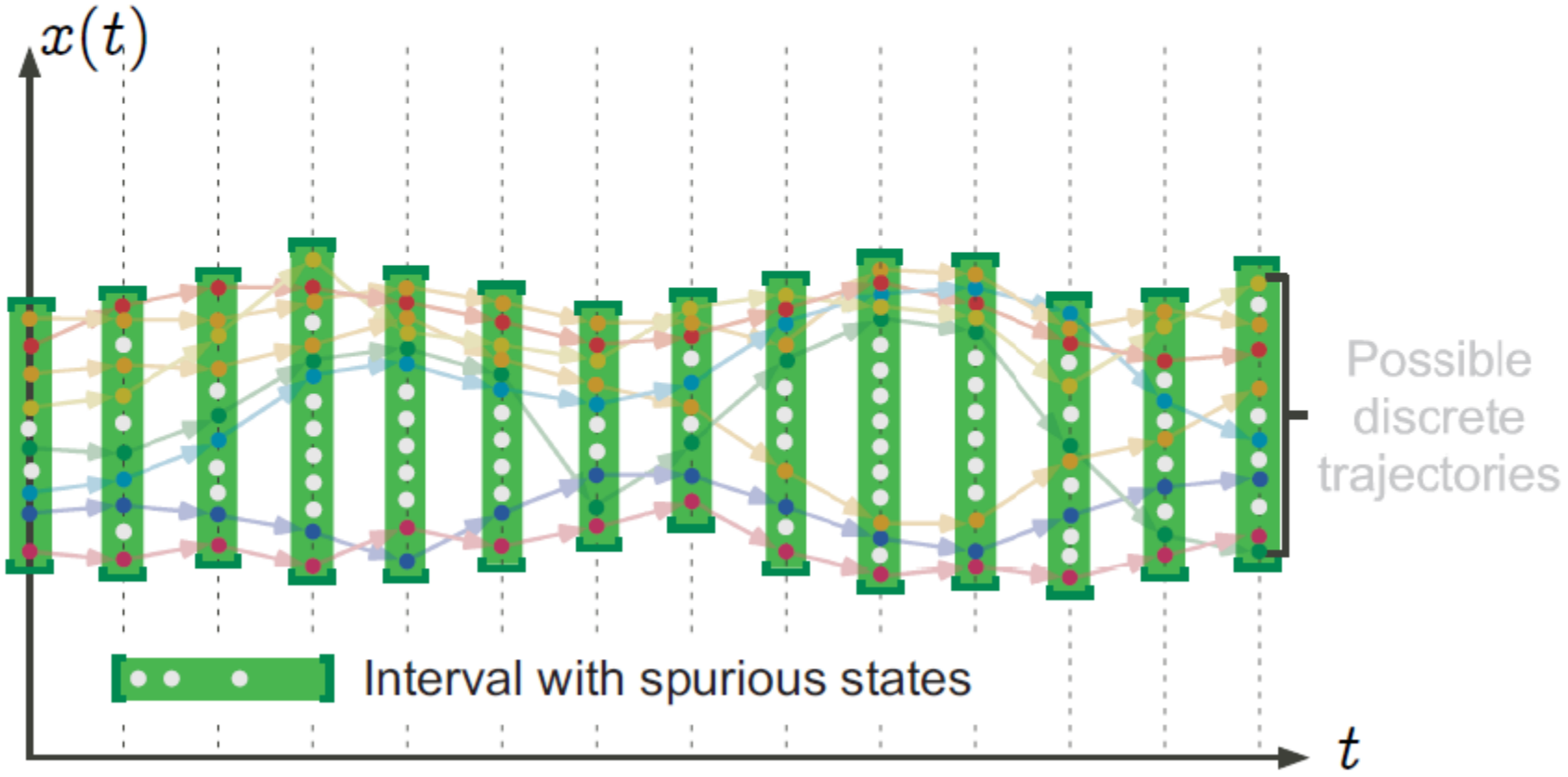
$\wedge\ \forall \overline{x}, \overline{y} \in \overline{\mathcal{D}} : \overline{x} \sqsubseteq \overline{y} \Longrightarrow \gamma(\overline{x}) \subseteq \gamma(\overline{y})$

$\wedge\ \forall x \in \mathcal{D} : x \subseteq \gamma(\alpha(x))$

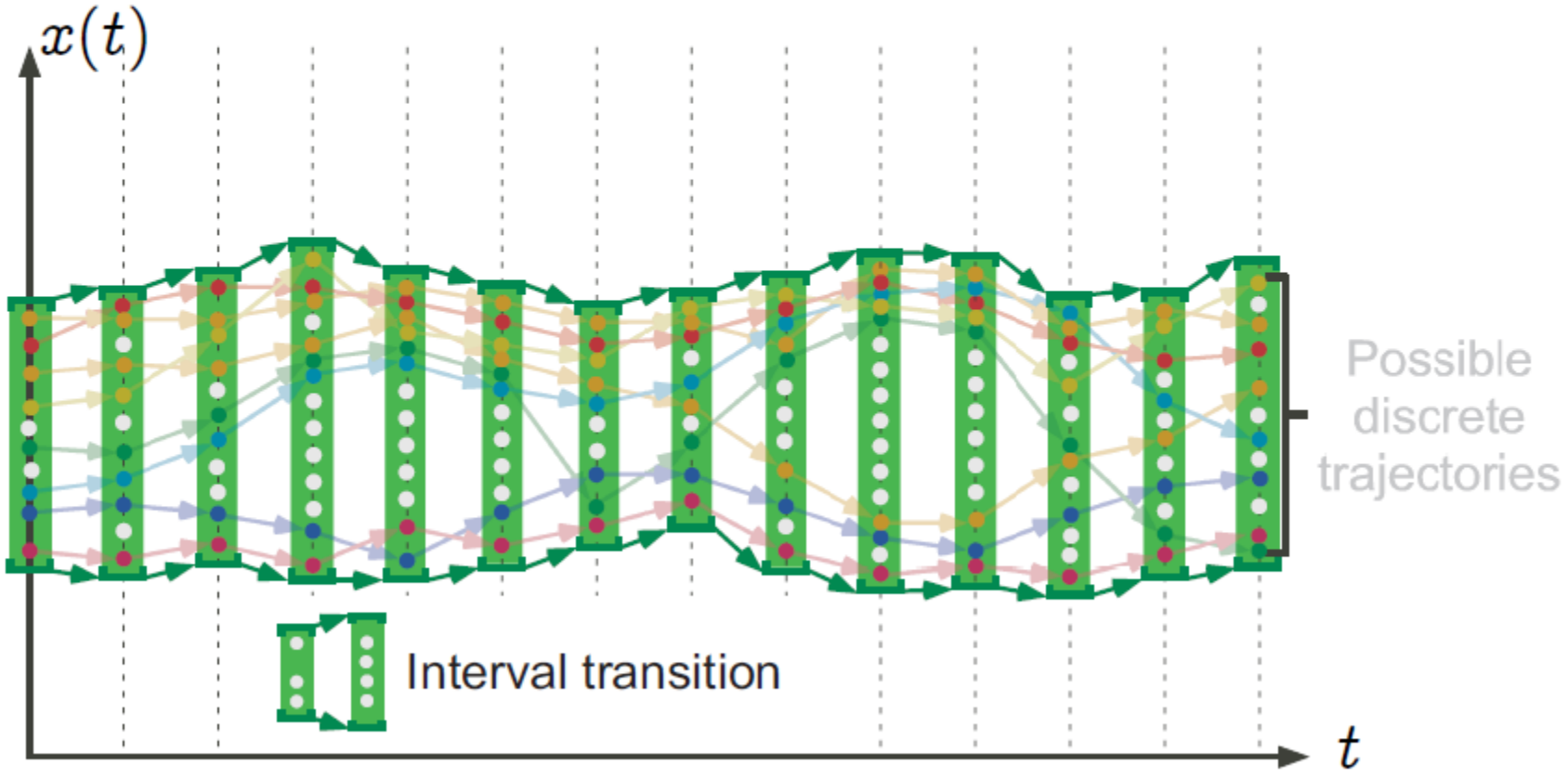
$\wedge\ \forall \overline{y} \in \overline{\mathcal{D}} : \alpha(\gamma(\overline{y})) \sqsubseteq \overline{x}$

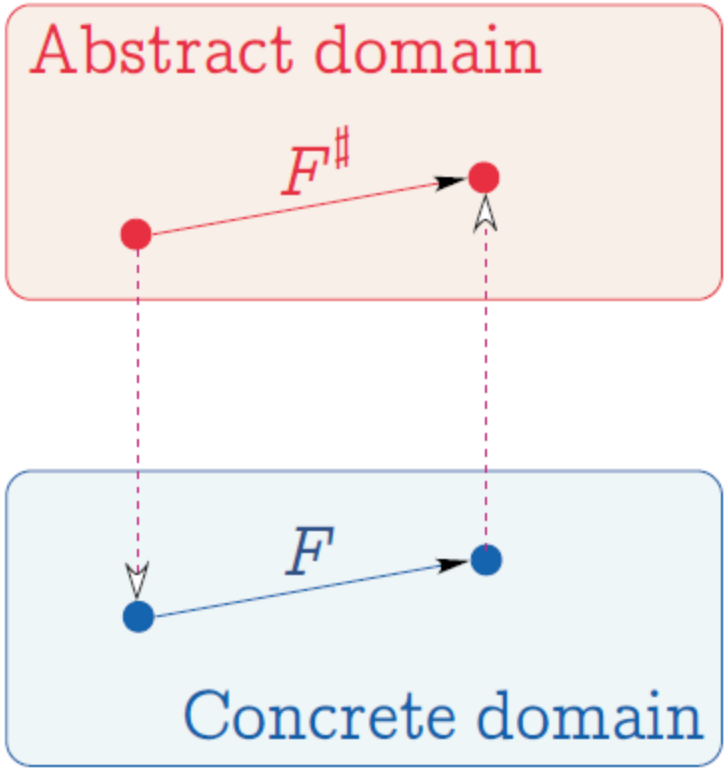
iff $\quad \forall x \in \mathcal{D}, \overline{y} \in \overline{\mathcal{D}} : \alpha(x) \sqsubseteq y \Longleftrightarrow x \subseteq \gamma(y)$

Graphic example: Interval abstraction



Graphic example: Abstract transitions





Function abstraction

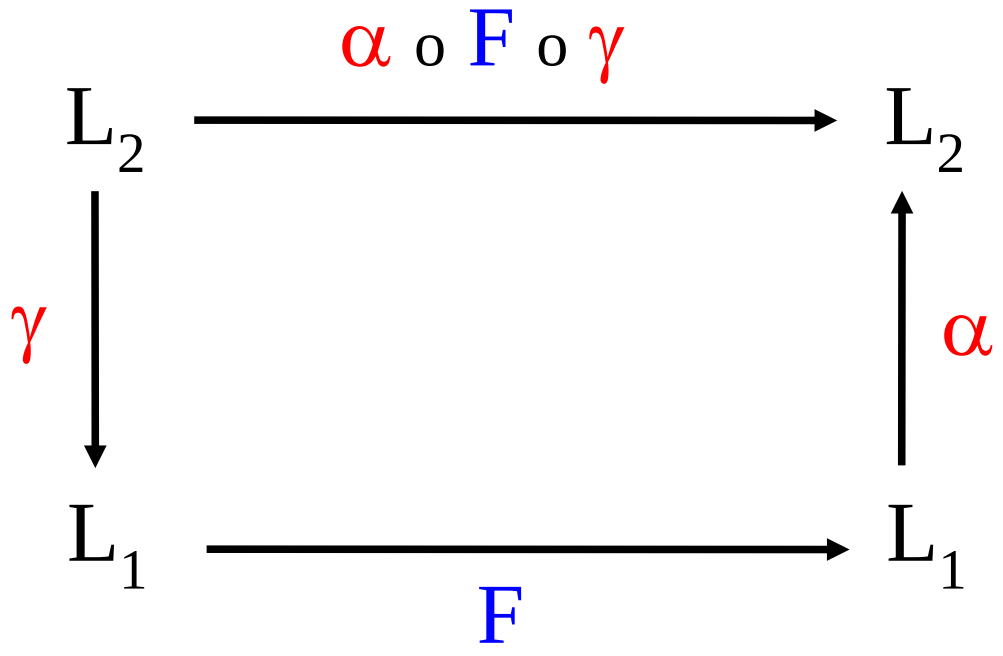
$$F^\sharp = \alpha \circ F \circ \gamma$$

i.e. $F^\sharp = \rho \circ F$

$$\langle P, \subseteq \rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle Q, \sqsubseteq \rangle \Rightarrow$$

$$\langle P \overset{\text{mon}}{\longmapsto} P, \dot{\subseteq} \rangle \underset{\lambda F \,.\, \alpha \circ F \circ \gamma}{\overset{\lambda F^\sharp \,.\, \gamma \circ F^\sharp \circ \alpha}{\leftrightarrows}} \langle Q \overset{\text{mon}}{\longmapsto} Q, \dot{\sqsubseteq} \rangle$$

Fixpoint Approximation

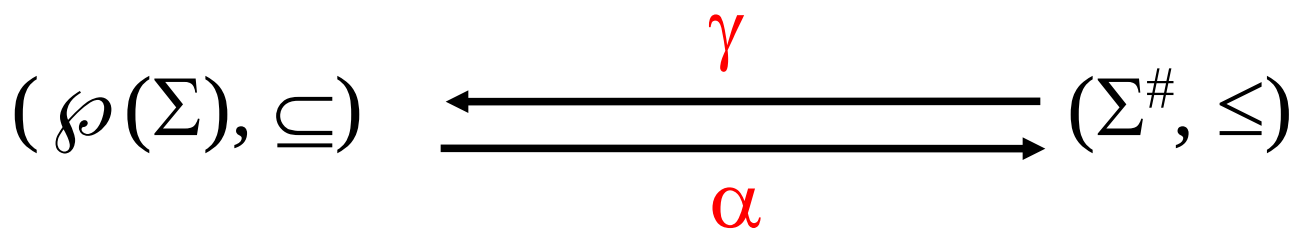


Theorem:

$$\text{lfp } F \subseteq \gamma\,(\text{lfp } \alpha \circ F \circ \gamma)$$

Abstracting the Collecting Semantics

- Find a Galois connection:



- Find a function: $\alpha \circ F \circ \gamma \leq F^{\#}$

Partitioning ⇨ Abstract sets of environments

Example: Interval transition semantics of assignments

```
int x;
...
l:
    x := x + 1;
l':
```

$$\{\mathtt{l} : \mathtt{x} \in [\ell, h] \rightarrow \mathtt{l}' : \mathtt{x} \in [l + 1, \min(h + 1, \mathtt{max\_int})] \cup \{\Omega \mid h = \mathtt{max\_int}\} \mid \ell \leq h\}$$

where $[\ell, h] = \emptyset$ when $h < \ell$.

Example: Set of traces to trace of intervals abstraction

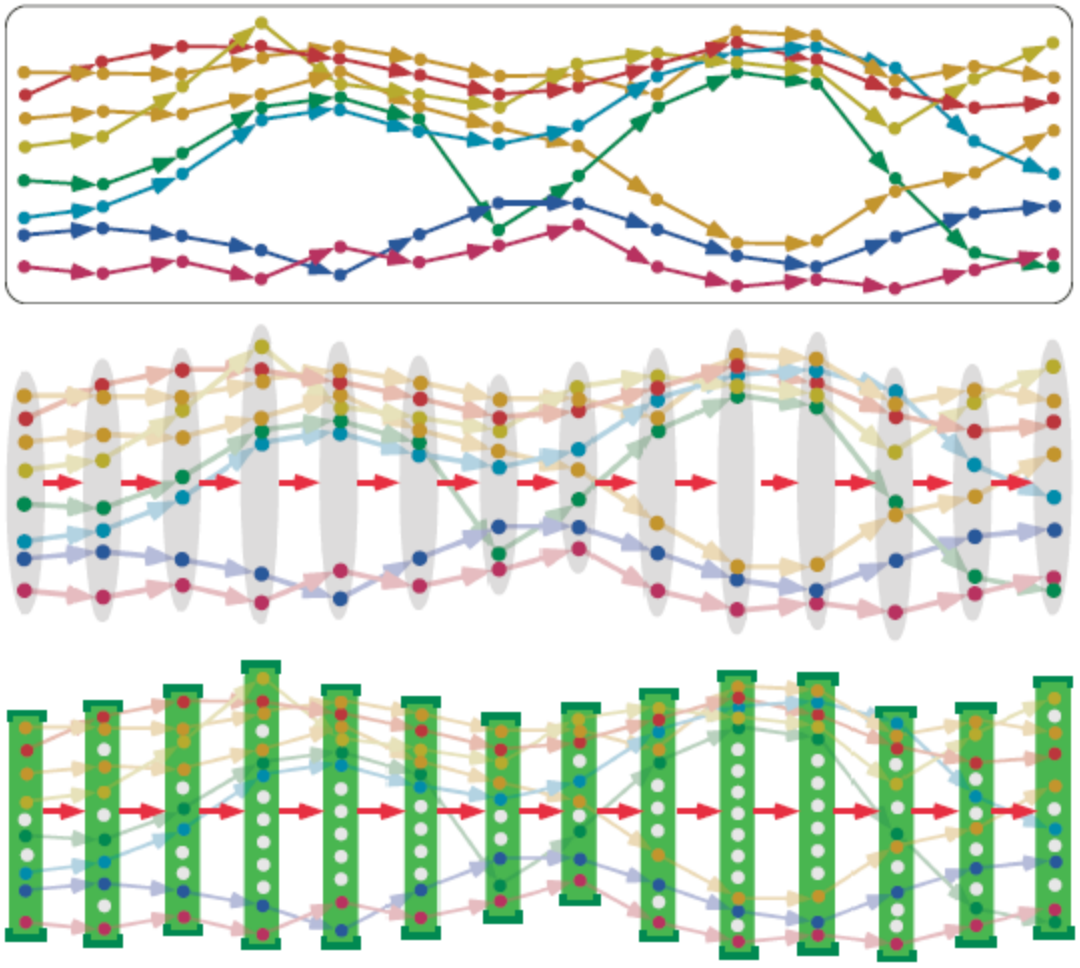
Set of traces:

$\alpha_1 \downarrow$

Trace of sets:

$\alpha_2 \downarrow$

Trace of intervals



Example: Set of traces to reachable states abstraction

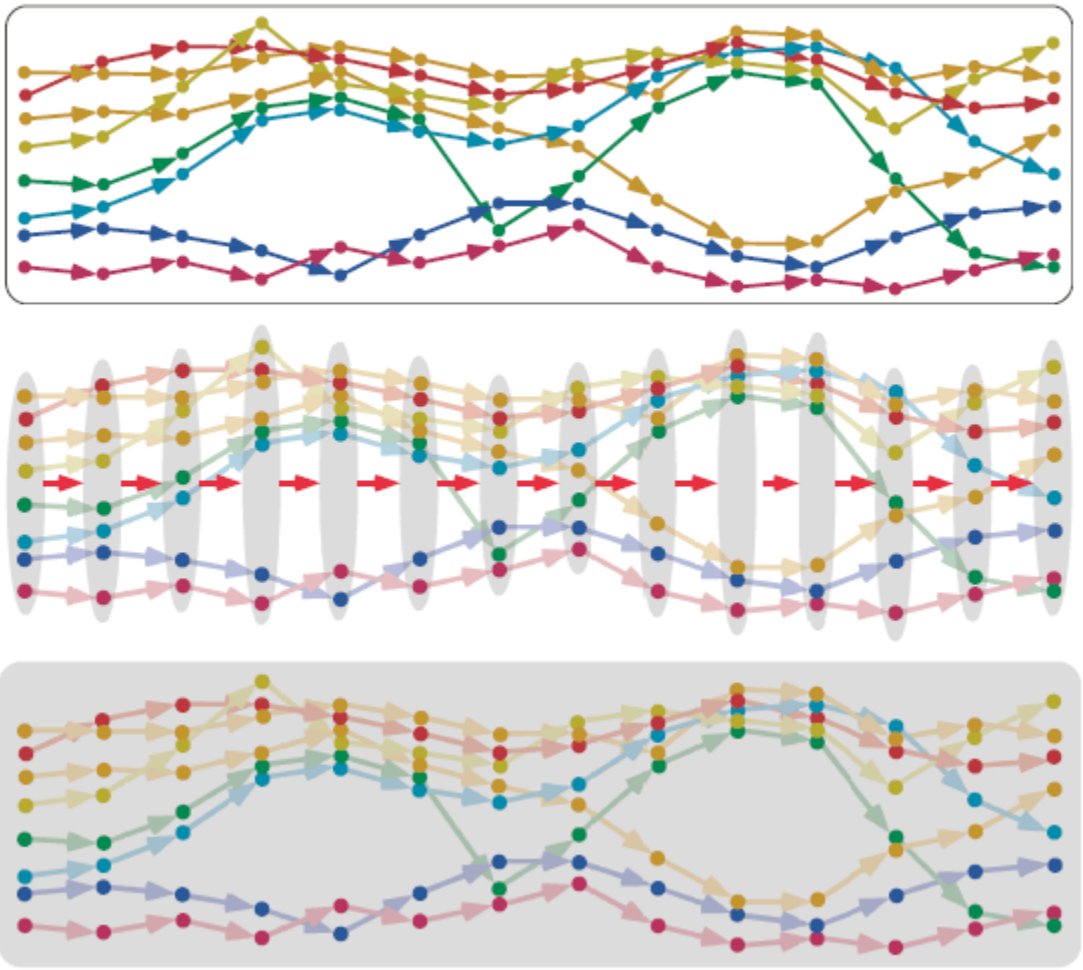
Set of traces:

$\alpha_1 \downarrow$

Trace of sets:

$\alpha_3 \downarrow$

Reachable states



Composition of Galois Connections

The composition of Galois connections:

$$\langle L, \leq \rangle \underset{\alpha_1}{\overset{\gamma_1}{\leftrightarrows}} \langle M, \sqsubseteq \rangle$$

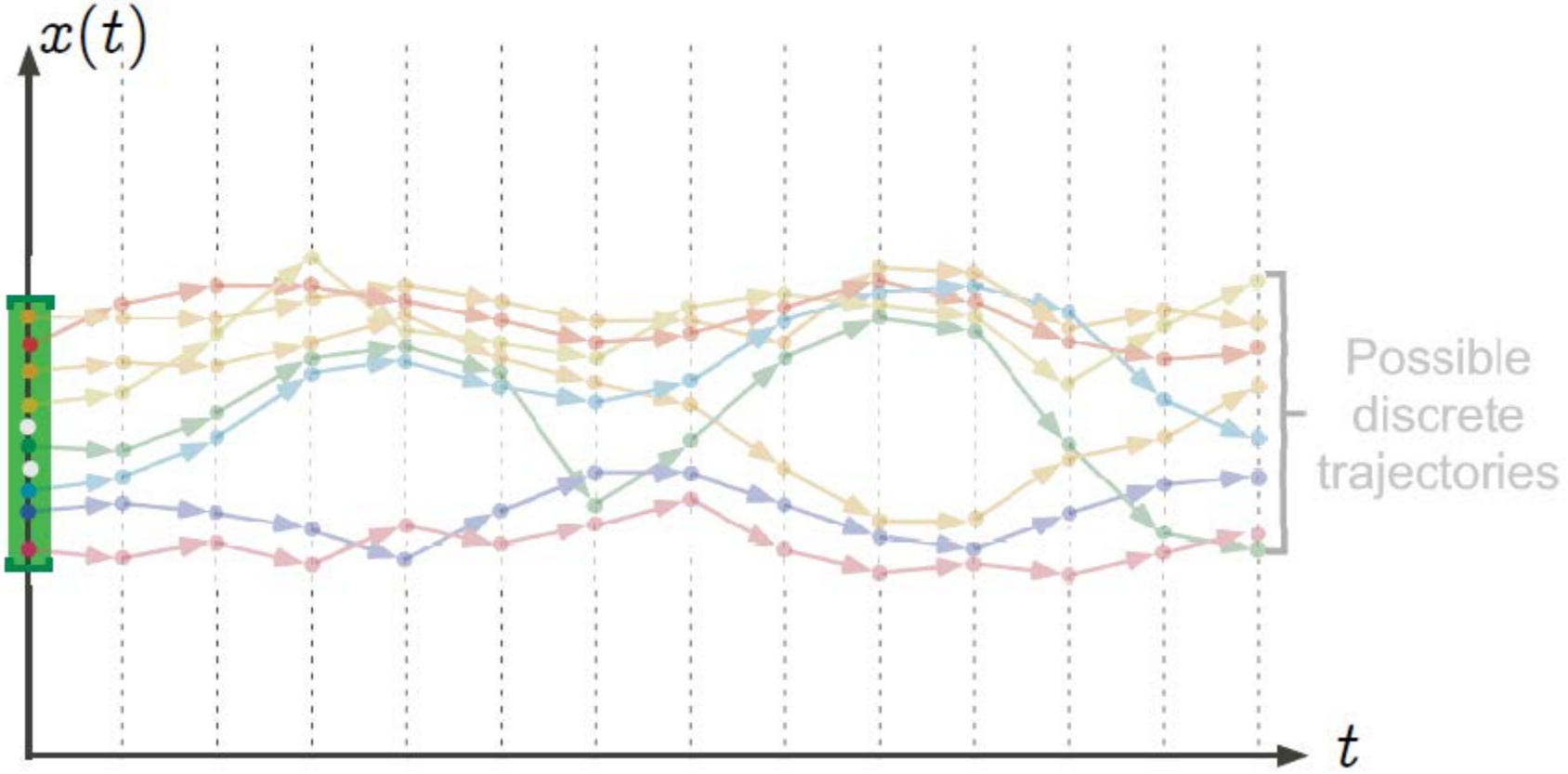
and:

$$\langle M, \sqsubseteq \rangle \underset{\alpha_2}{\overset{\gamma_2}{\leftrightarrows}} \langle N, \preceq \rangle$$

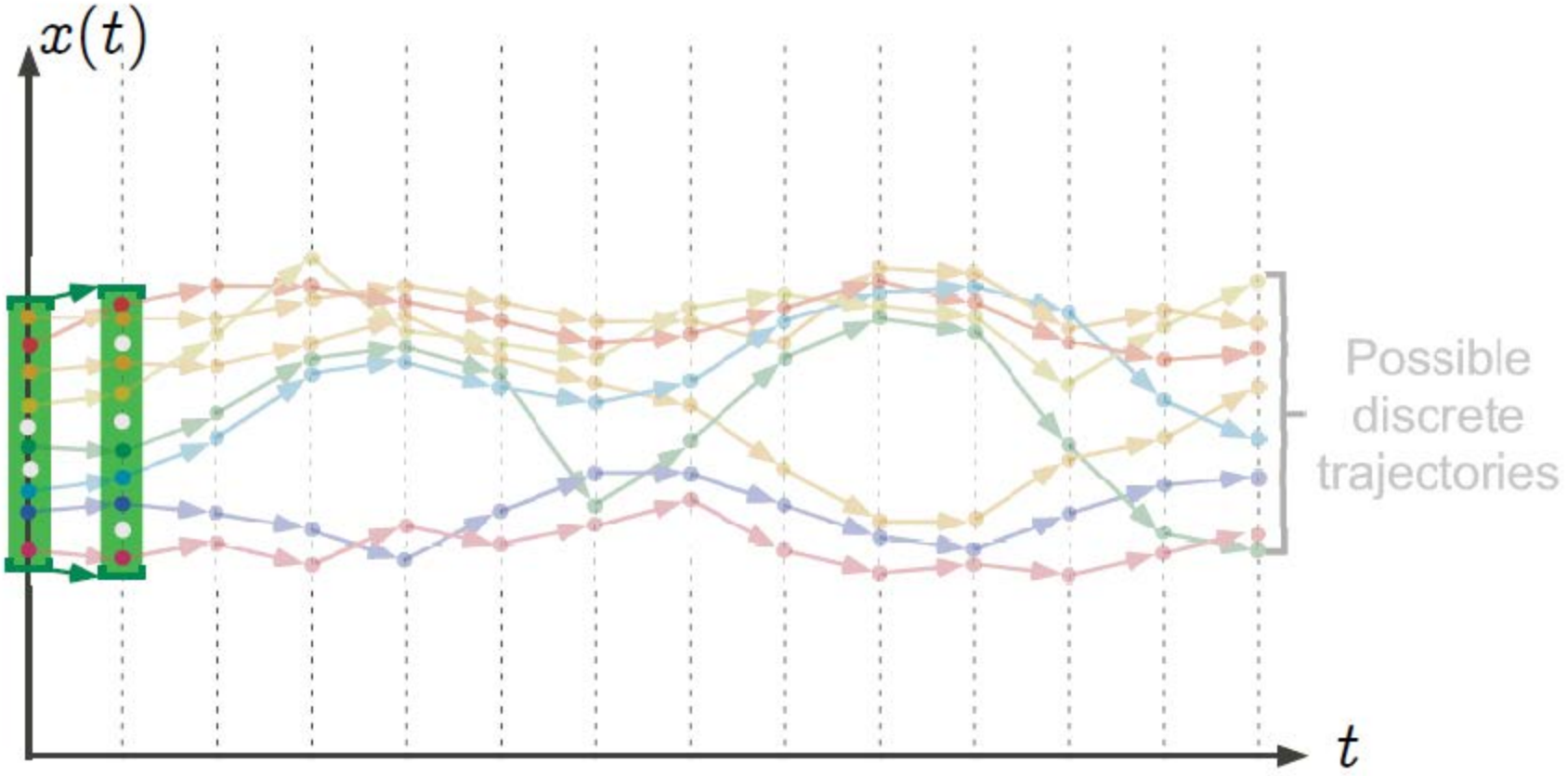
is a Galois connection:

$$\langle L, \leq \rangle \underset{\alpha_2 \circ \alpha_1}{\overset{\gamma_1 \circ \gamma_2}{\leftrightarrows}} \langle N, \preceq \rangle$$

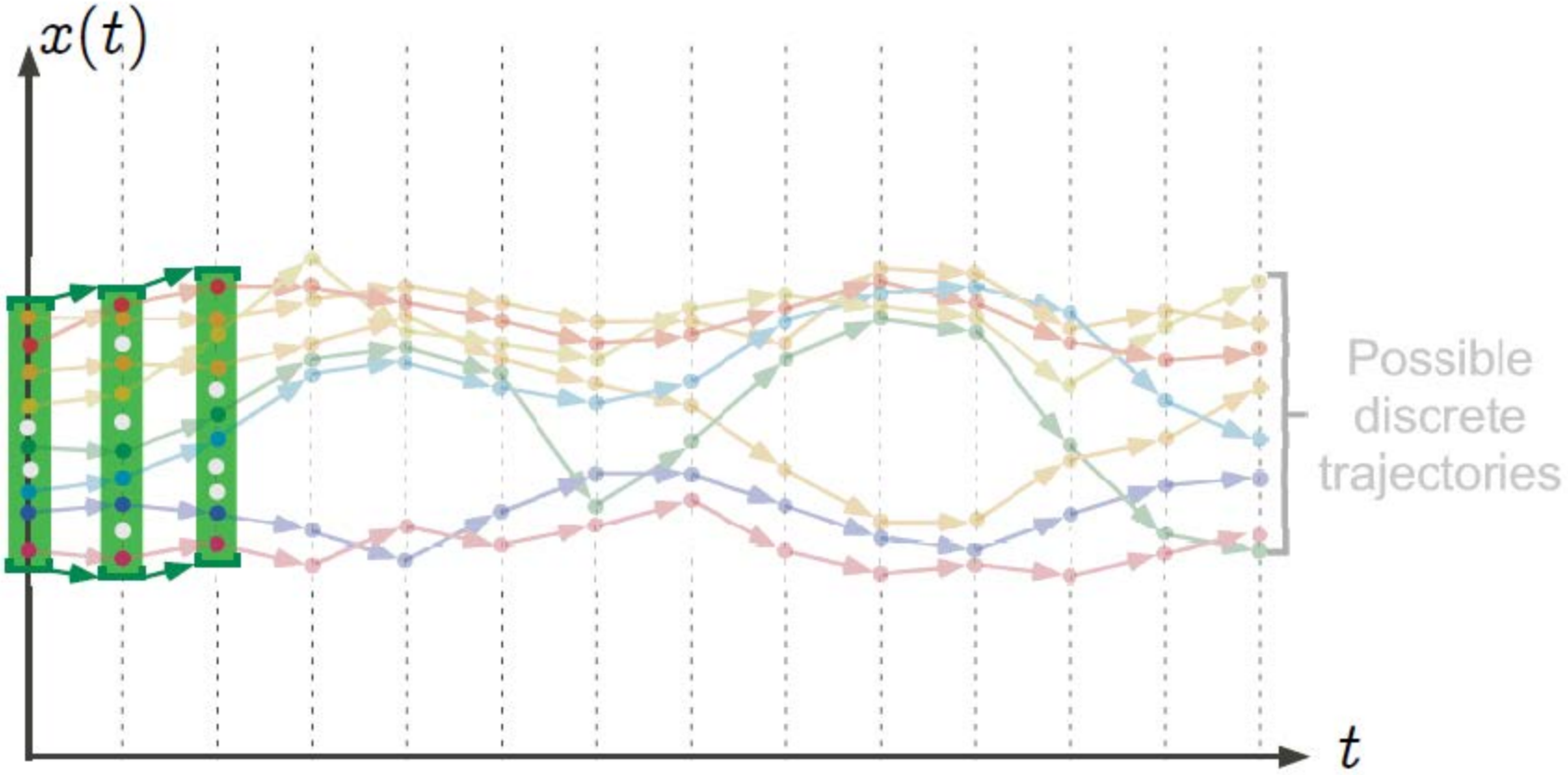
Graphic example: traces of intervals in fixpoint form



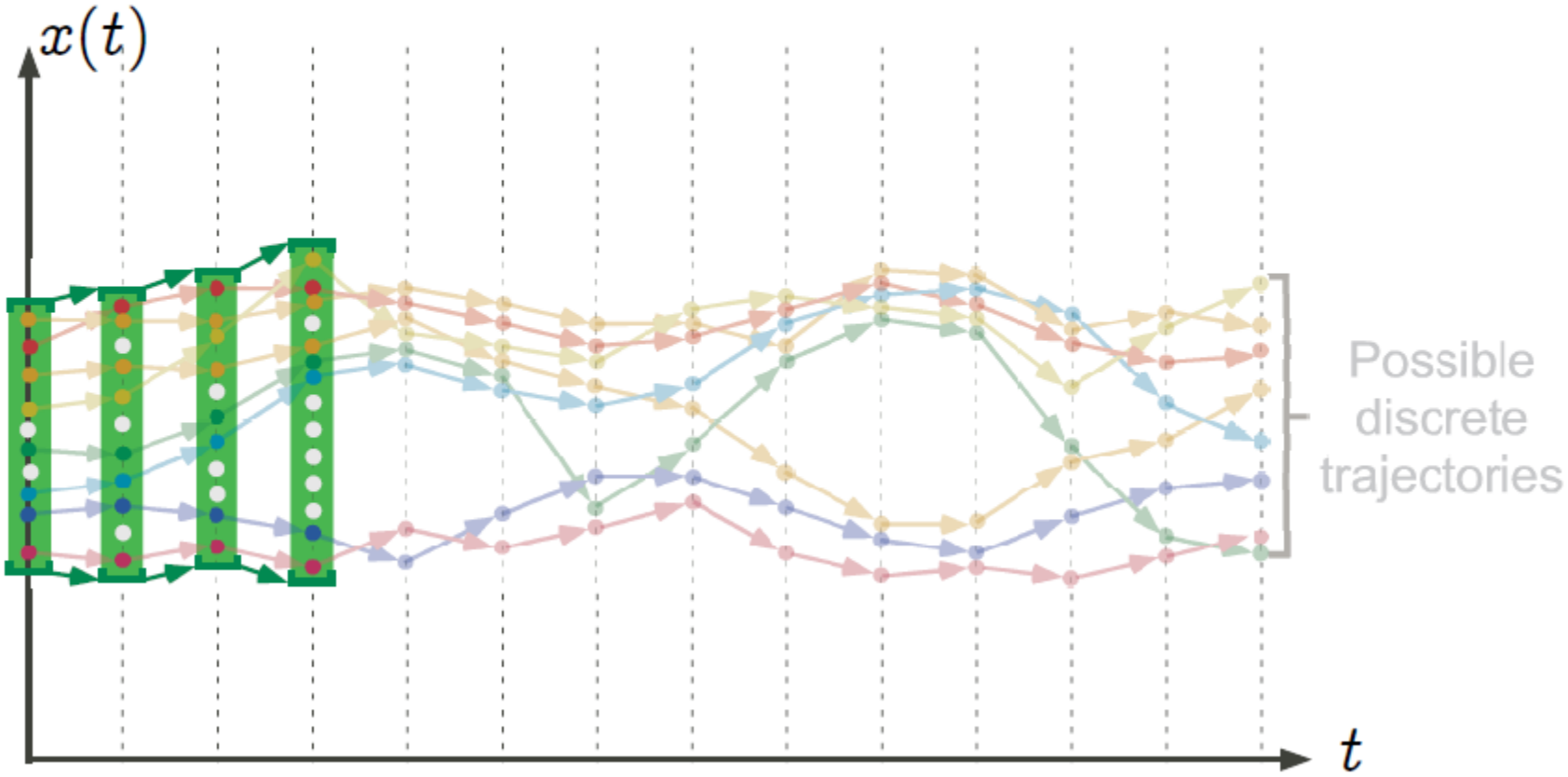
Graphic example: traces of intervals in fixpoint form



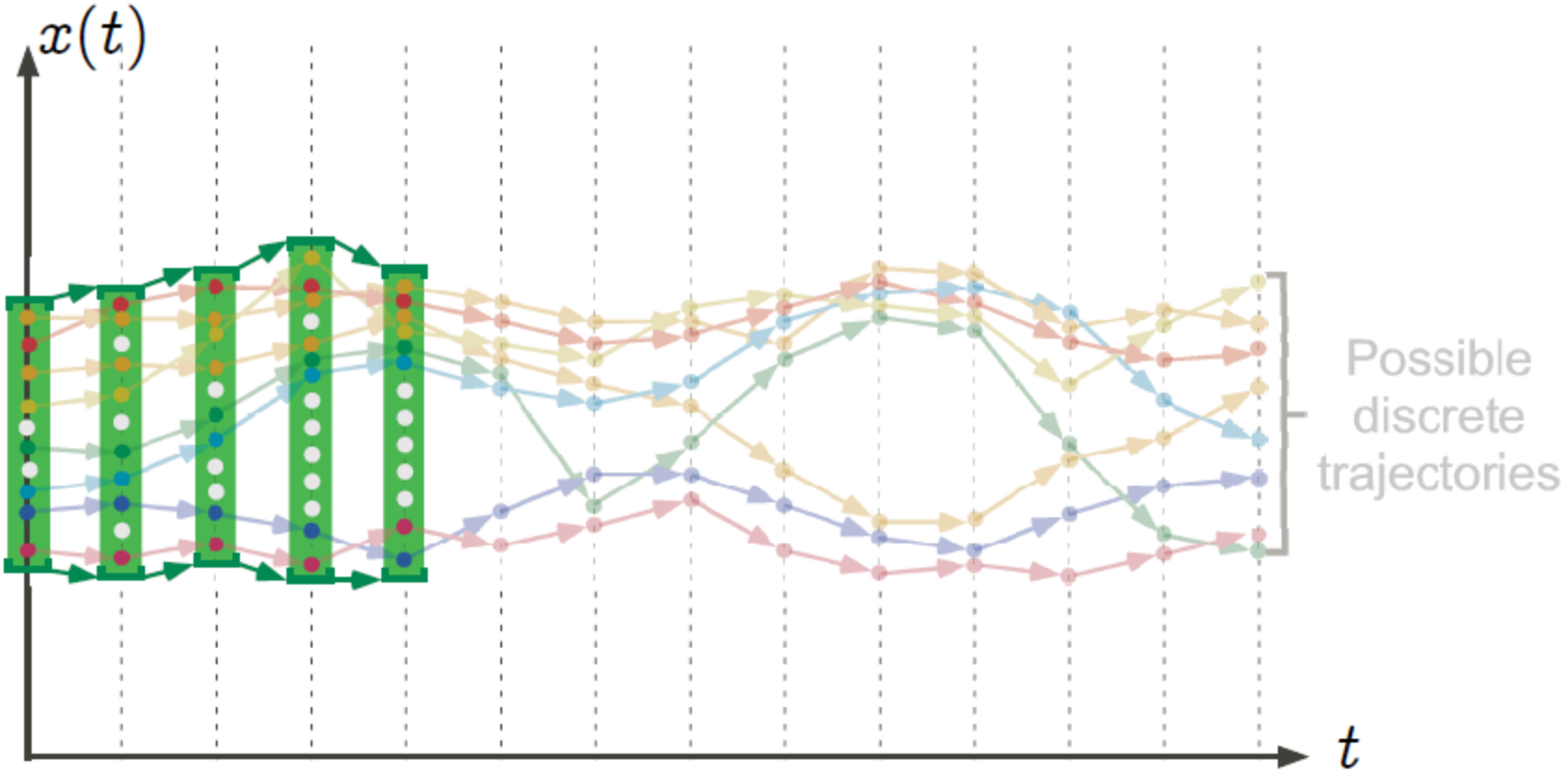
Graphic example: traces of intervals in fixpoint form



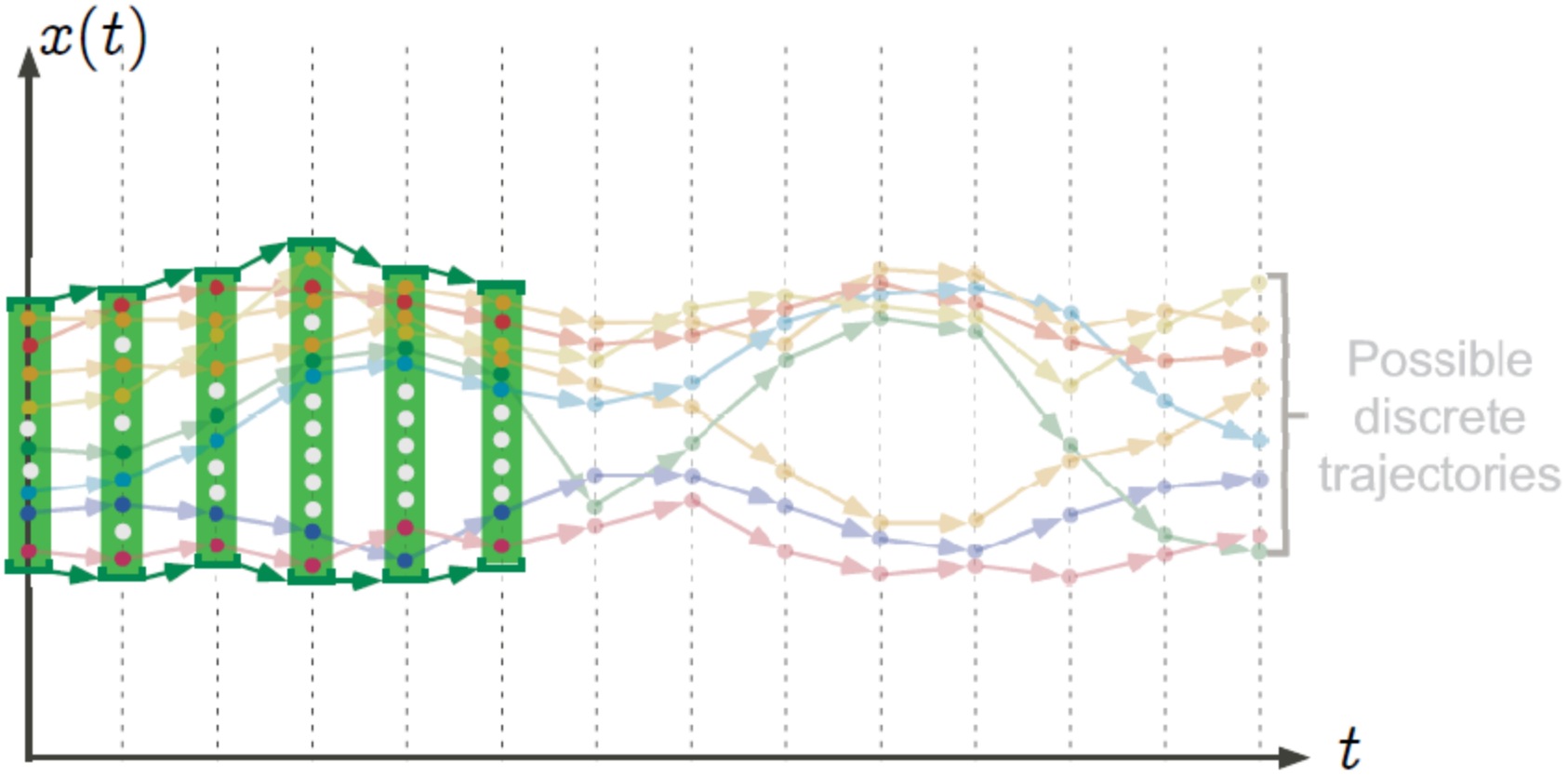
Graphic example: traces of intervals in fixpoint form



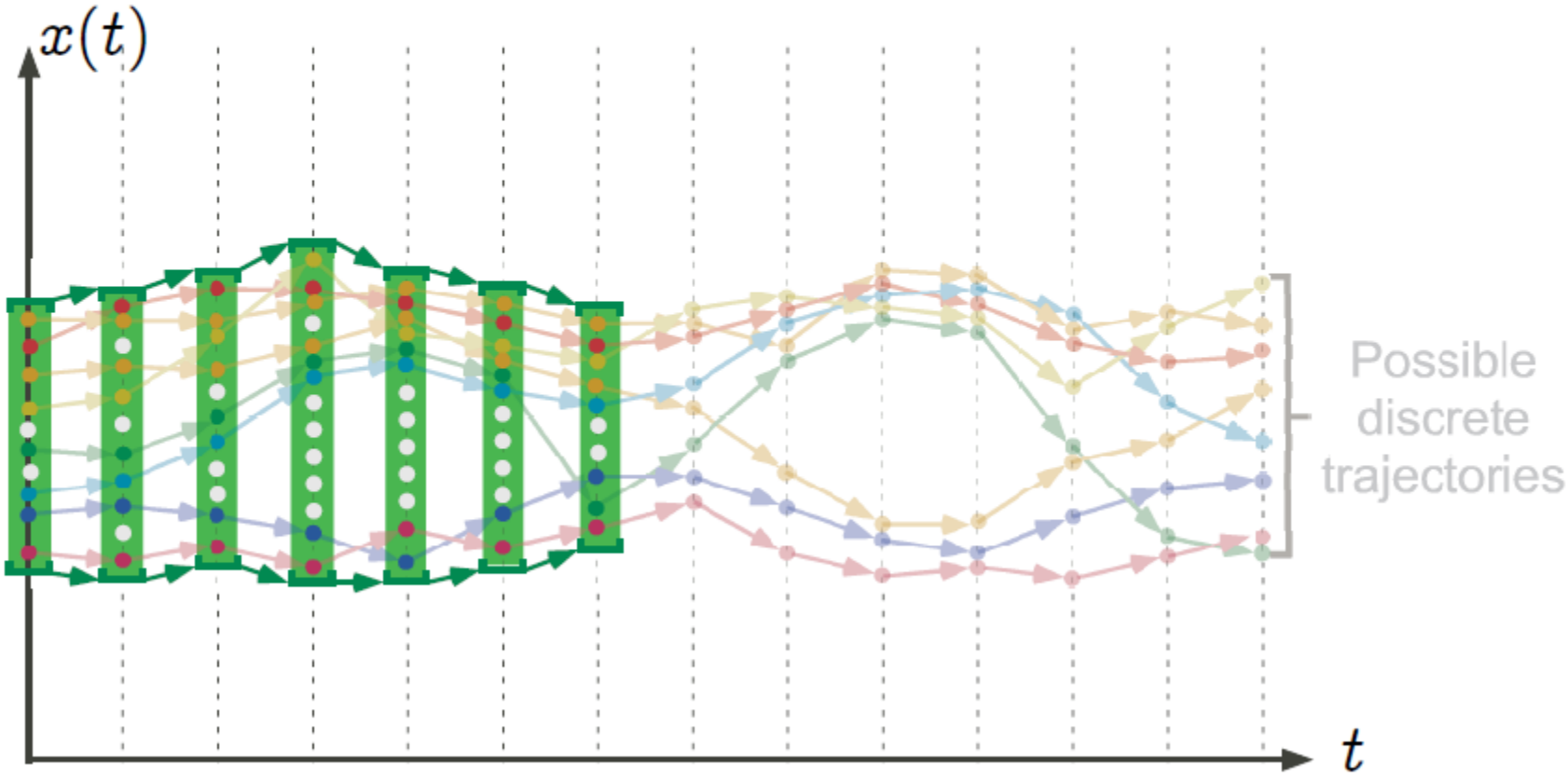
Graphic example: traces of intervals in fixpoint form



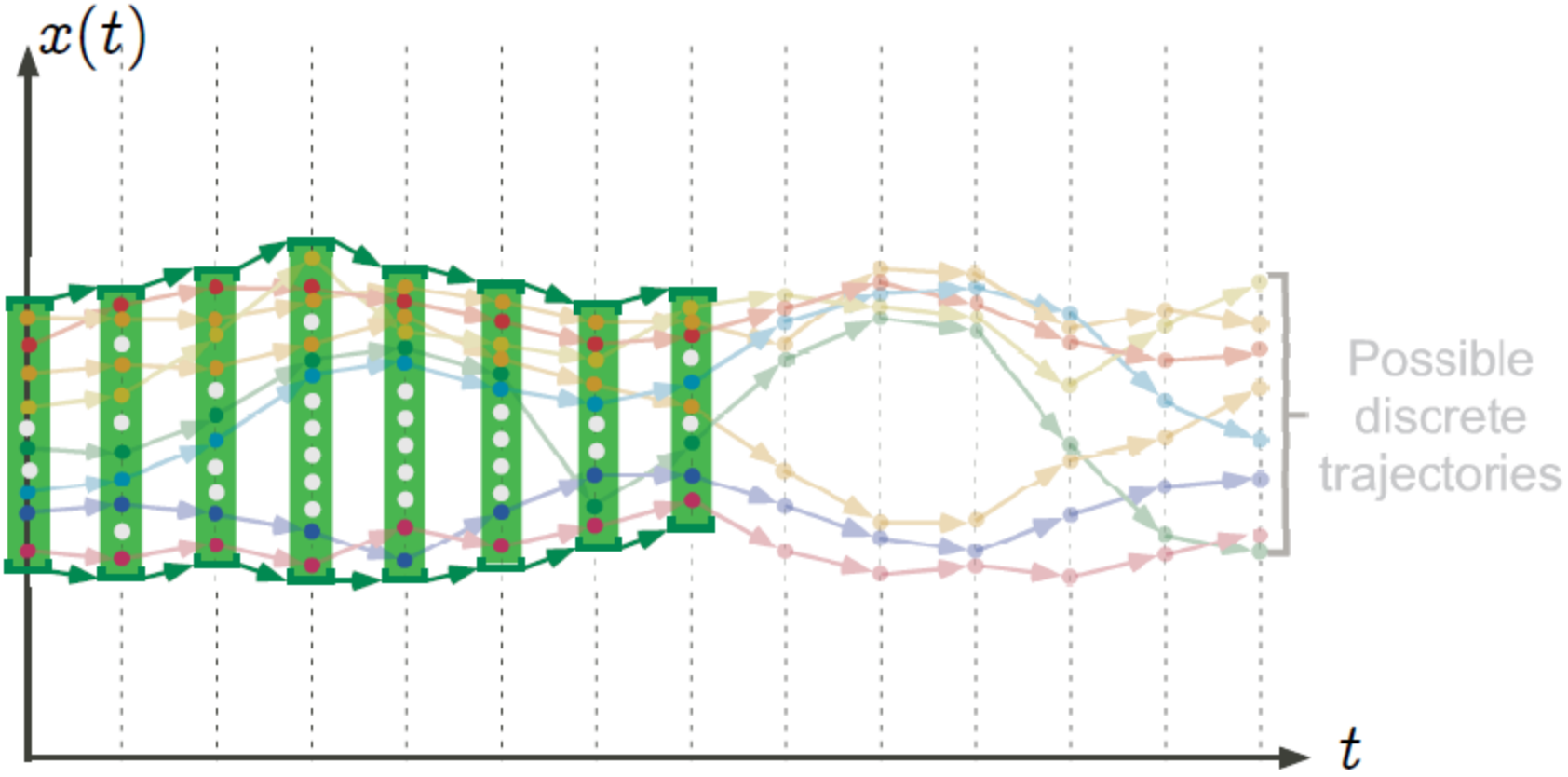
Graphic example: traces of intervals in fixpoint form



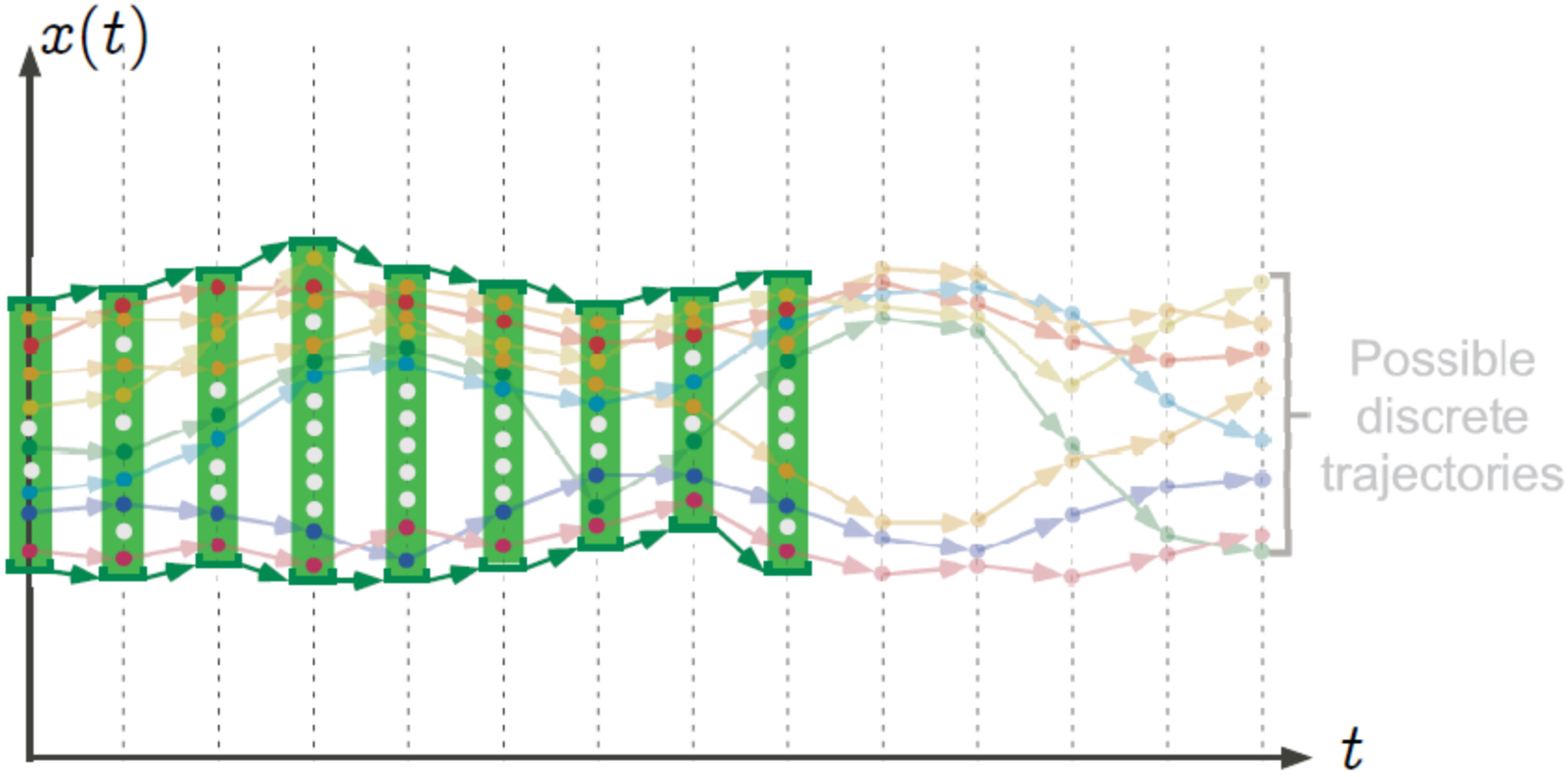
Graphic example: traces of intervals in fixpoint form



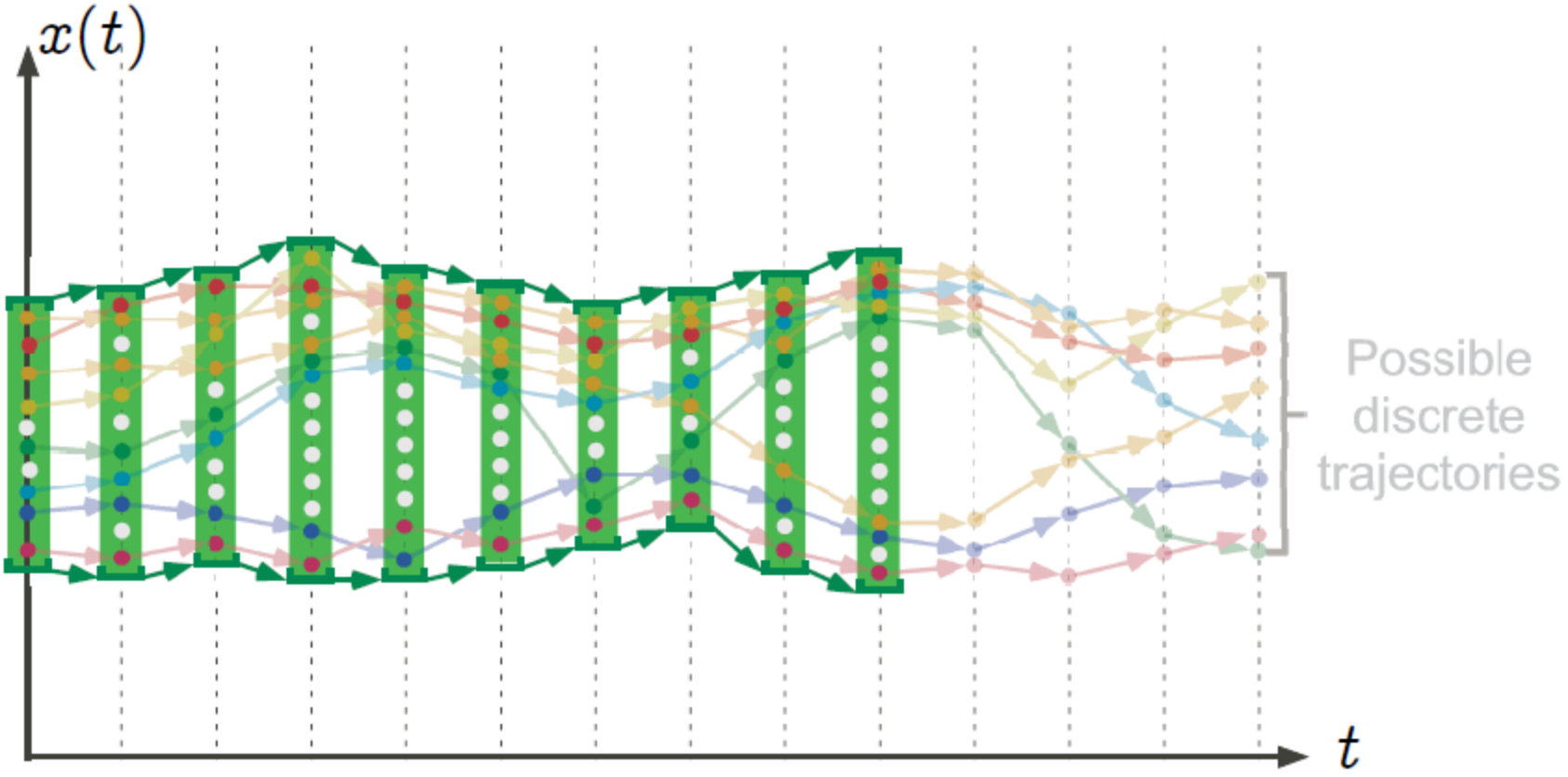
Graphic example: traces of intervals in fixpoint form



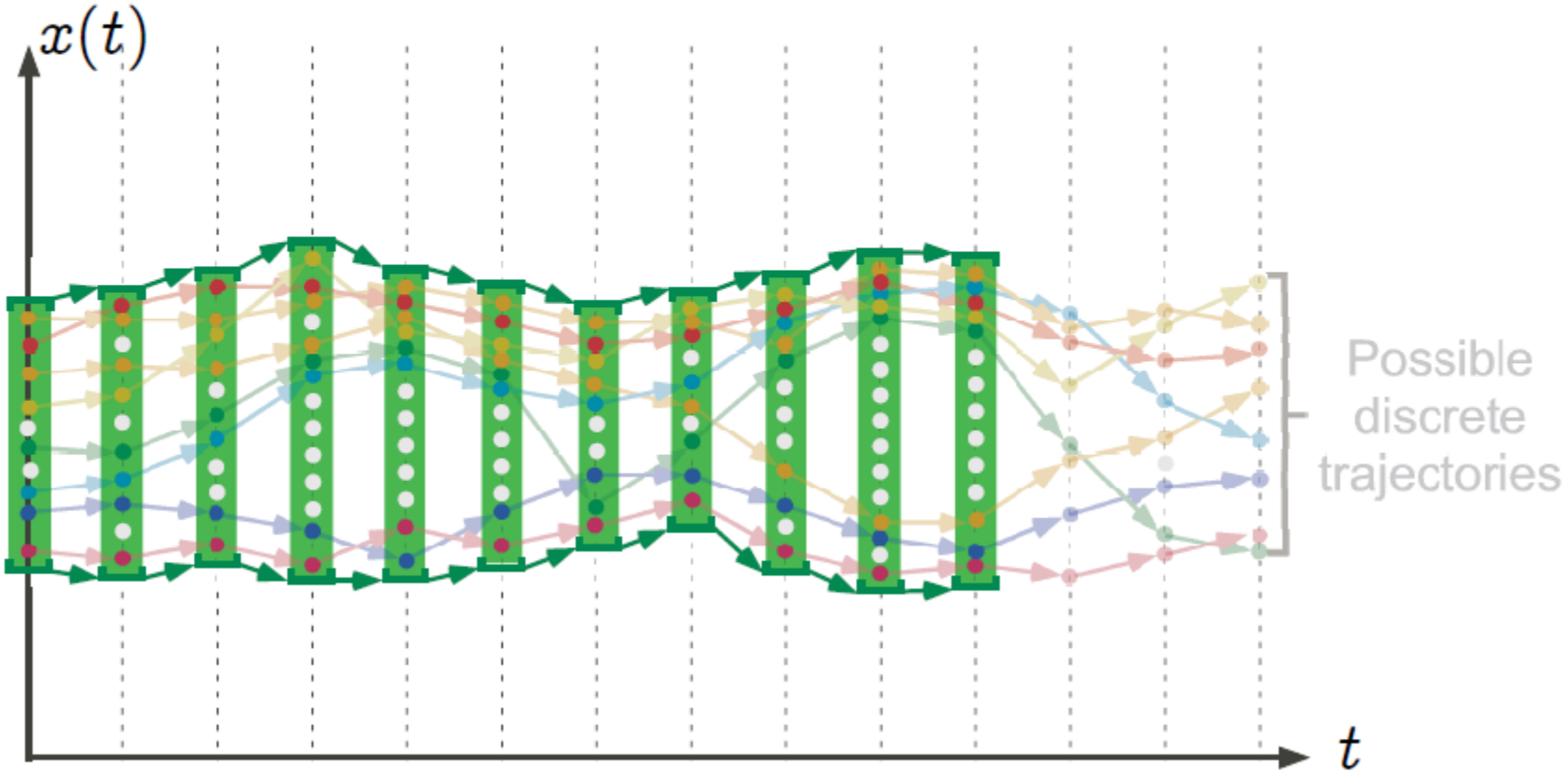
Graphic example: traces of intervals in fixpoint form



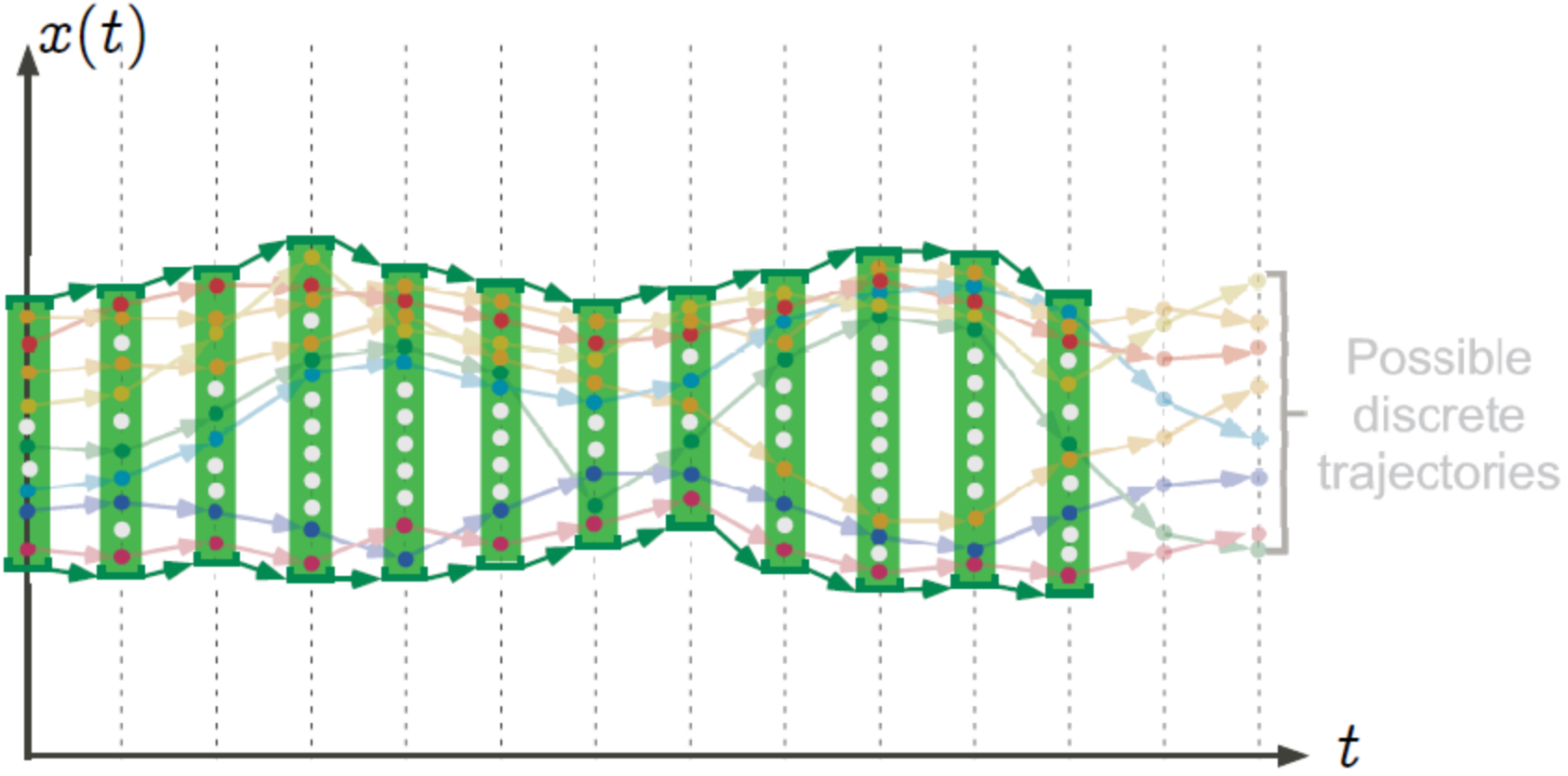
Graphic example: traces of intervals in fixpoint form



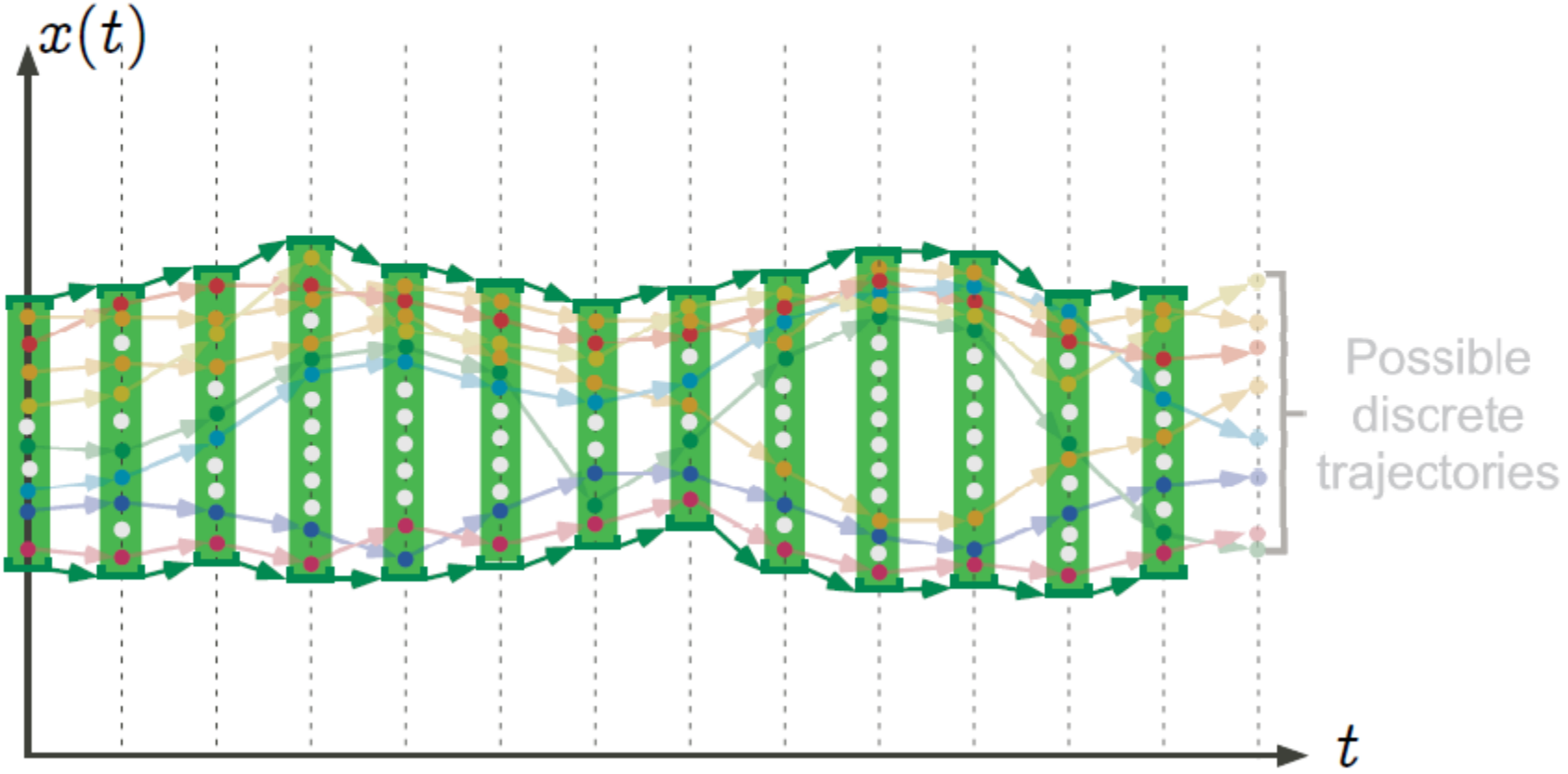
Graphic example: traces of intervals in fixpoint form



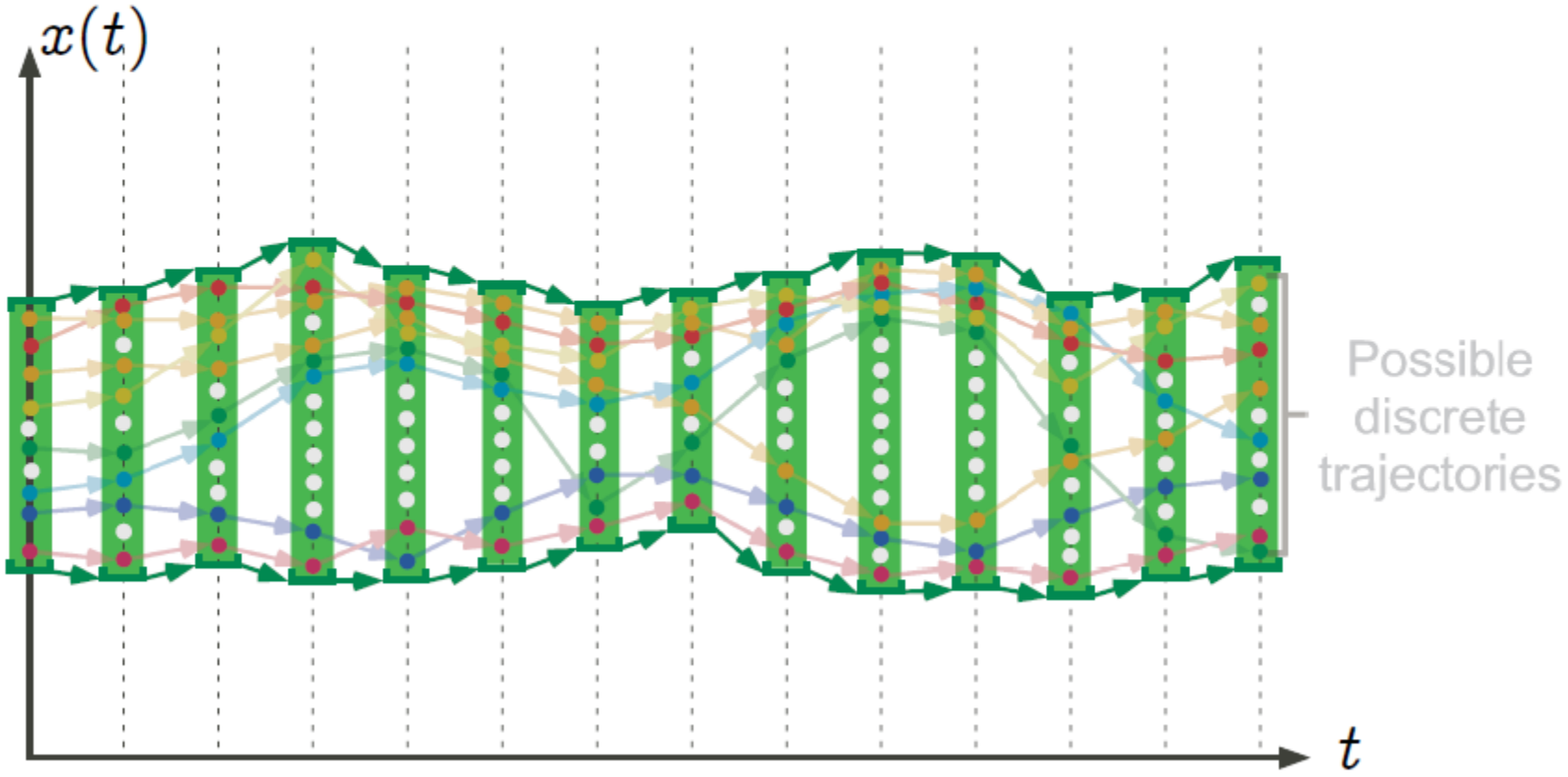
Graphic example: traces of intervals in fixpoint form



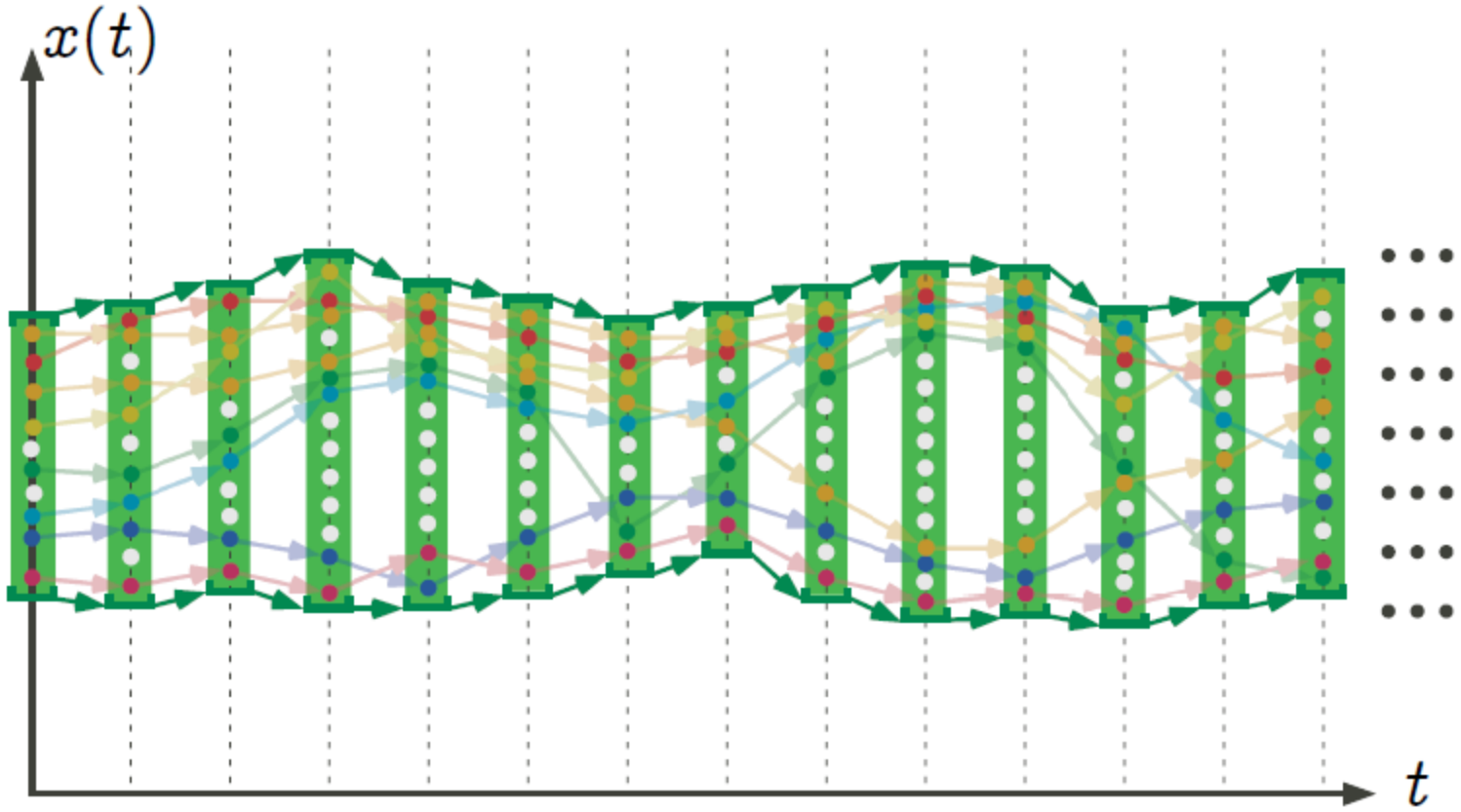
Graphic example: traces of intervals in fixpoint form



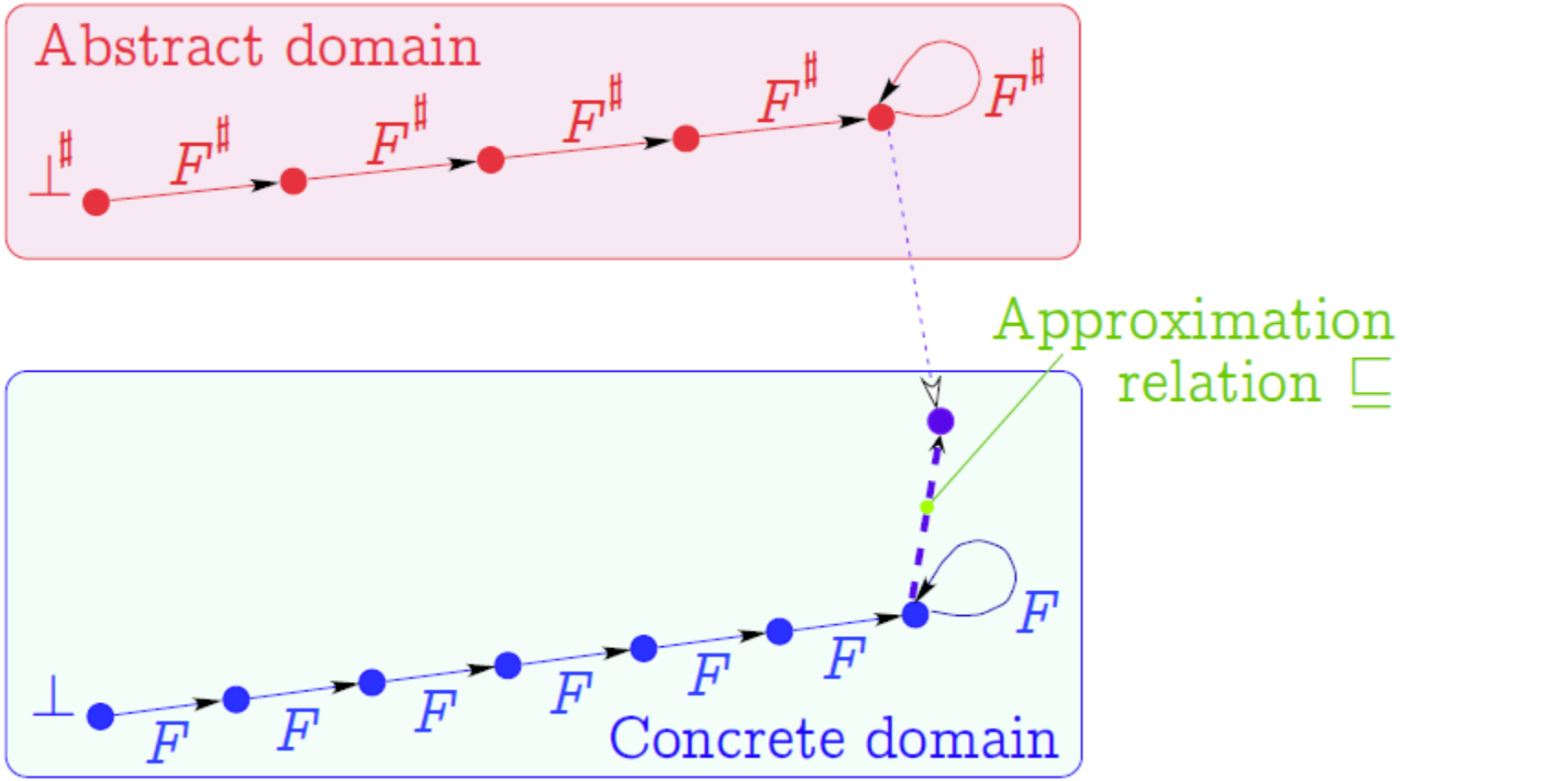
Graphic example: traces of intervals in fixpoint form



Graphic example: traces of intervals in fixpoint form



Approximate fixpoint abstraction



$\alpha(\mathsf{lfp}\ F) \sqsubseteq \mathsf{lfp}\ F^\sharp$

approximate/exact fixpoint abstraction

Exact Abstraction:

$$\alpha(\mathsf{lfp}\ F) = \mathsf{lfp}\ F^\sharp$$

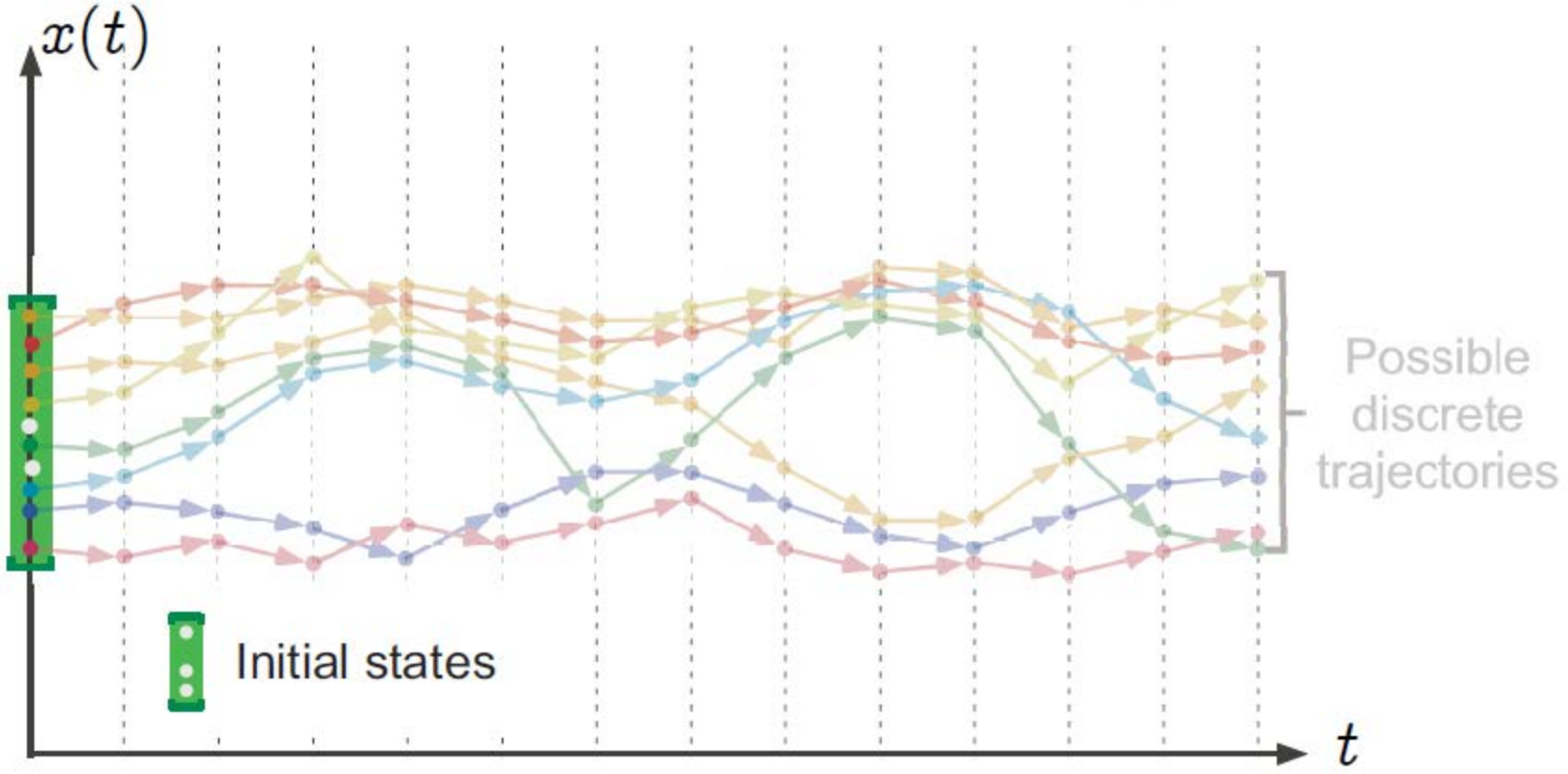
Approximate Abstraction:

$$\alpha(\mathsf{lfp}\ F) \sqsubseteq^\sharp \mathsf{lfp}\ F^\sharp$$

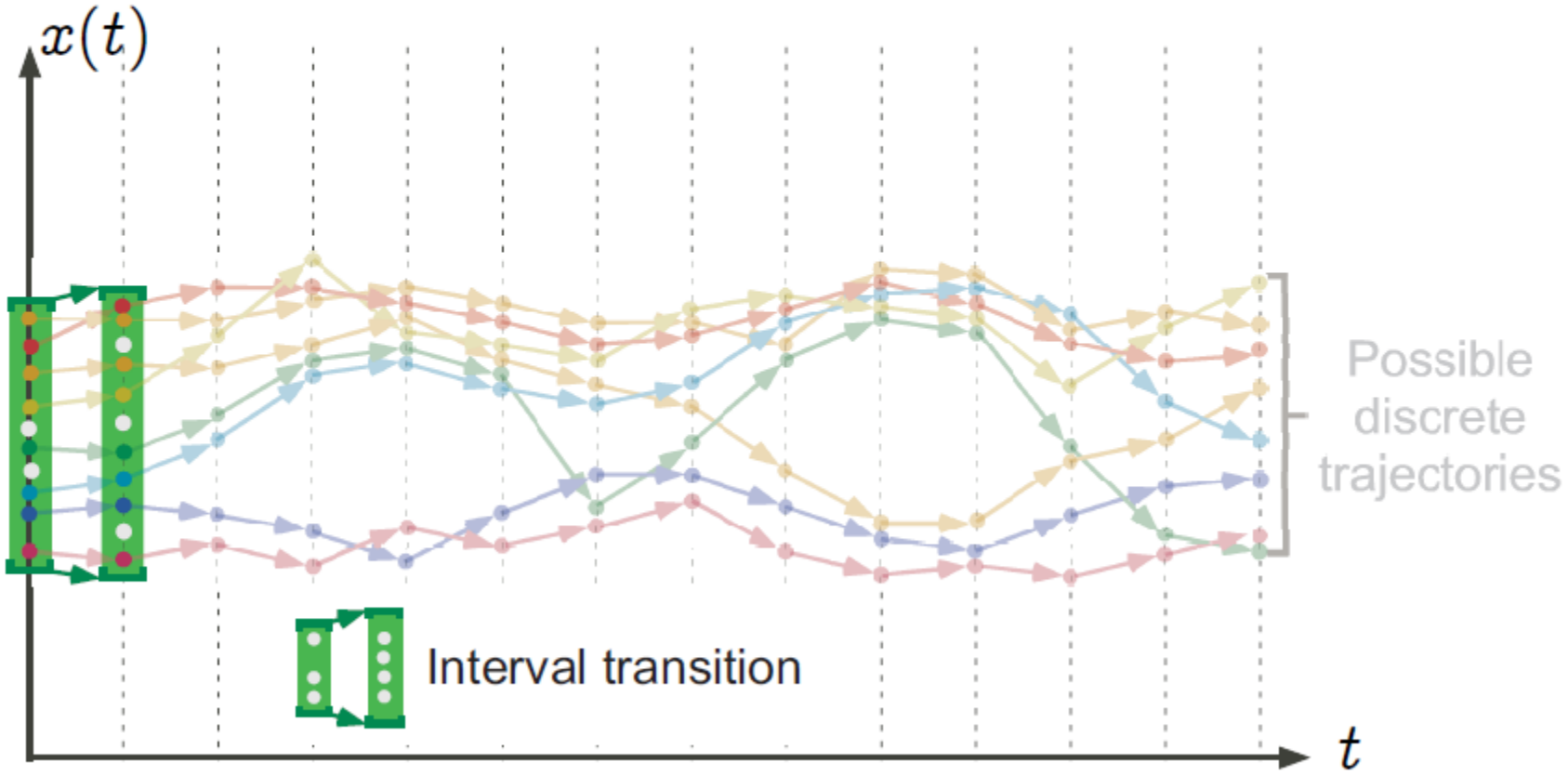
Widening and Narrowing

- Help the fixpoint iteration quickly converge and stabilize
- Conceptually similar to join and meet in dataflow analysis

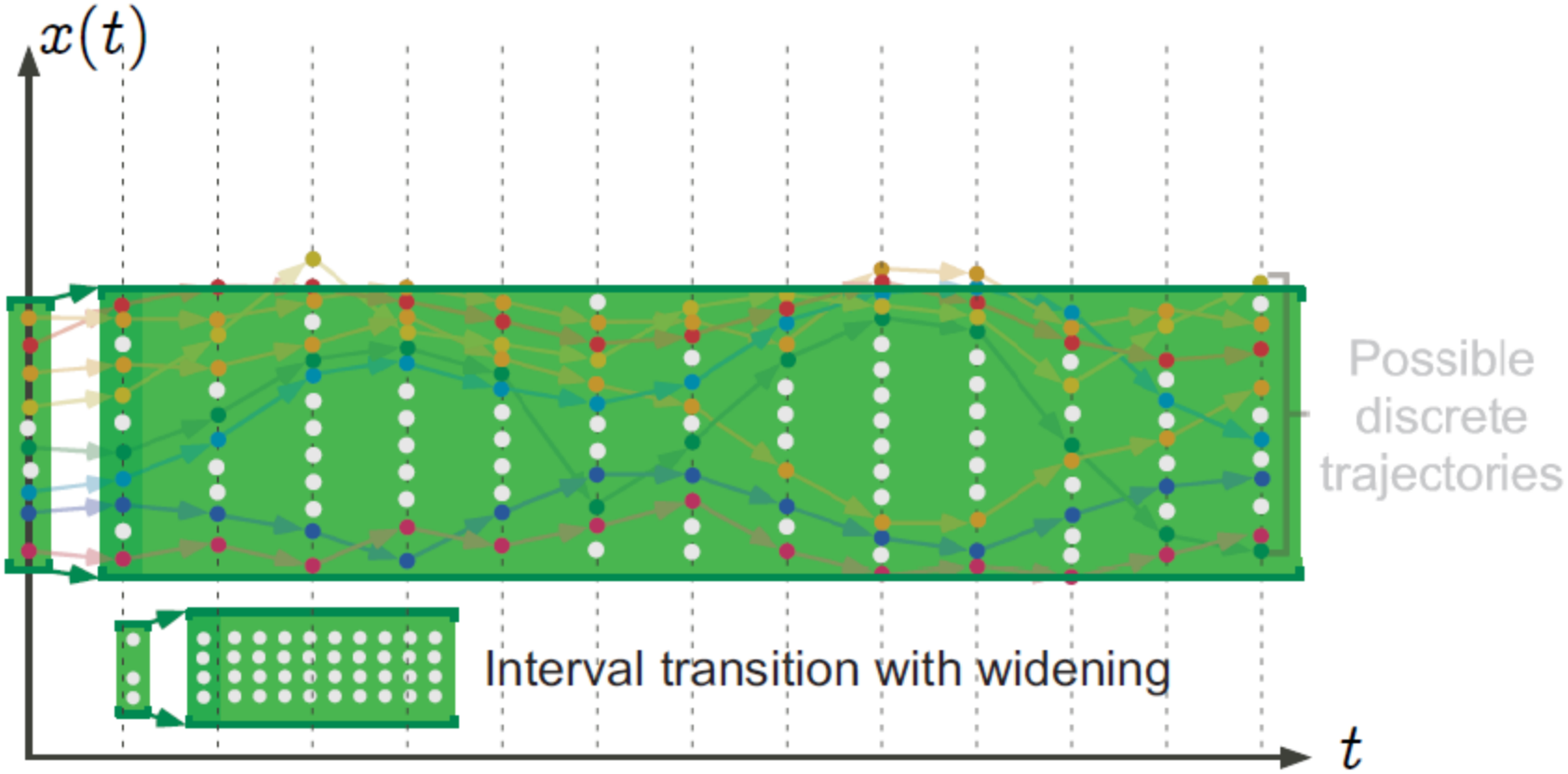
Graphic example: upward iteration with widening



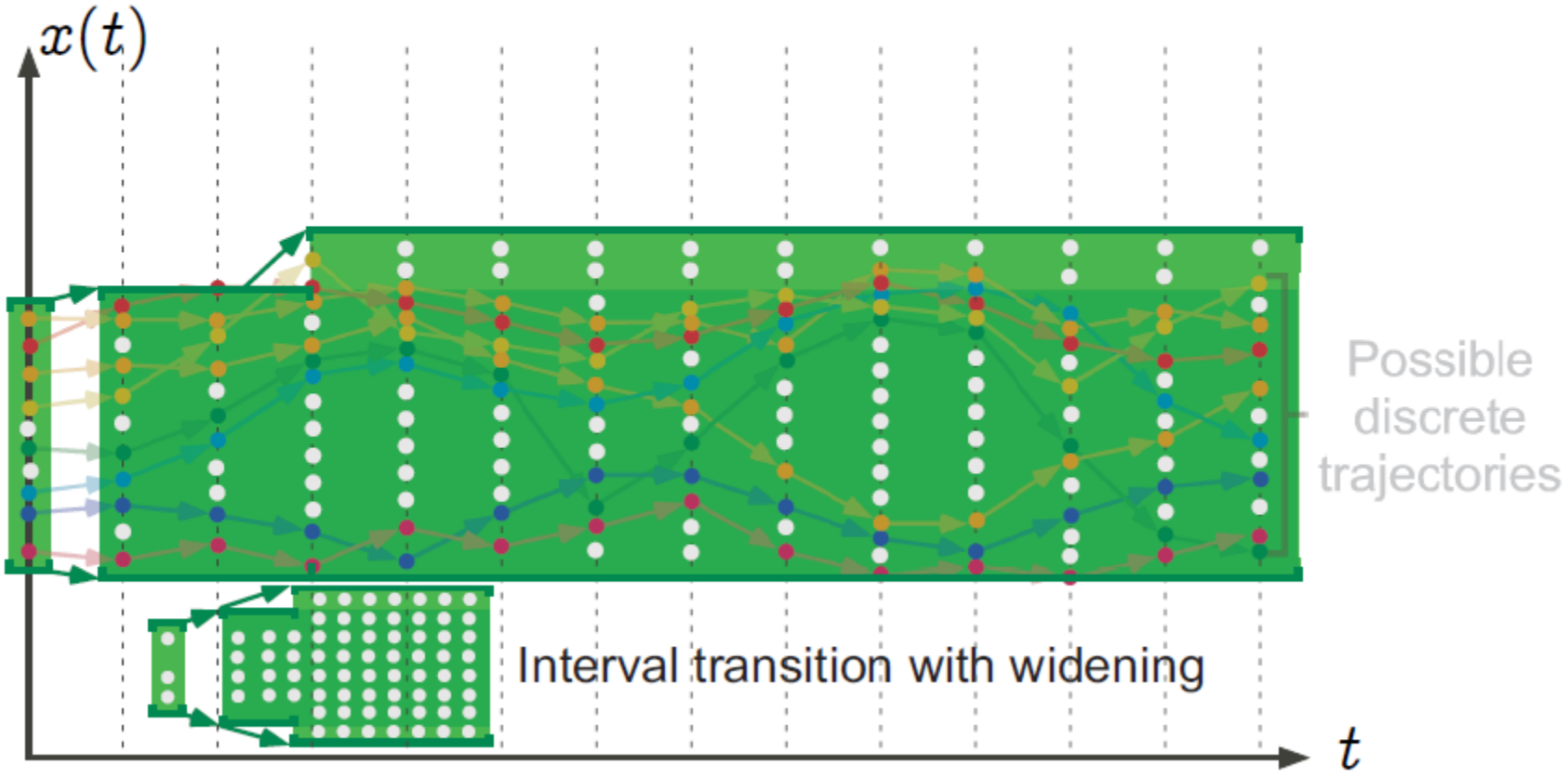
Graphic example: upward iteration with widening



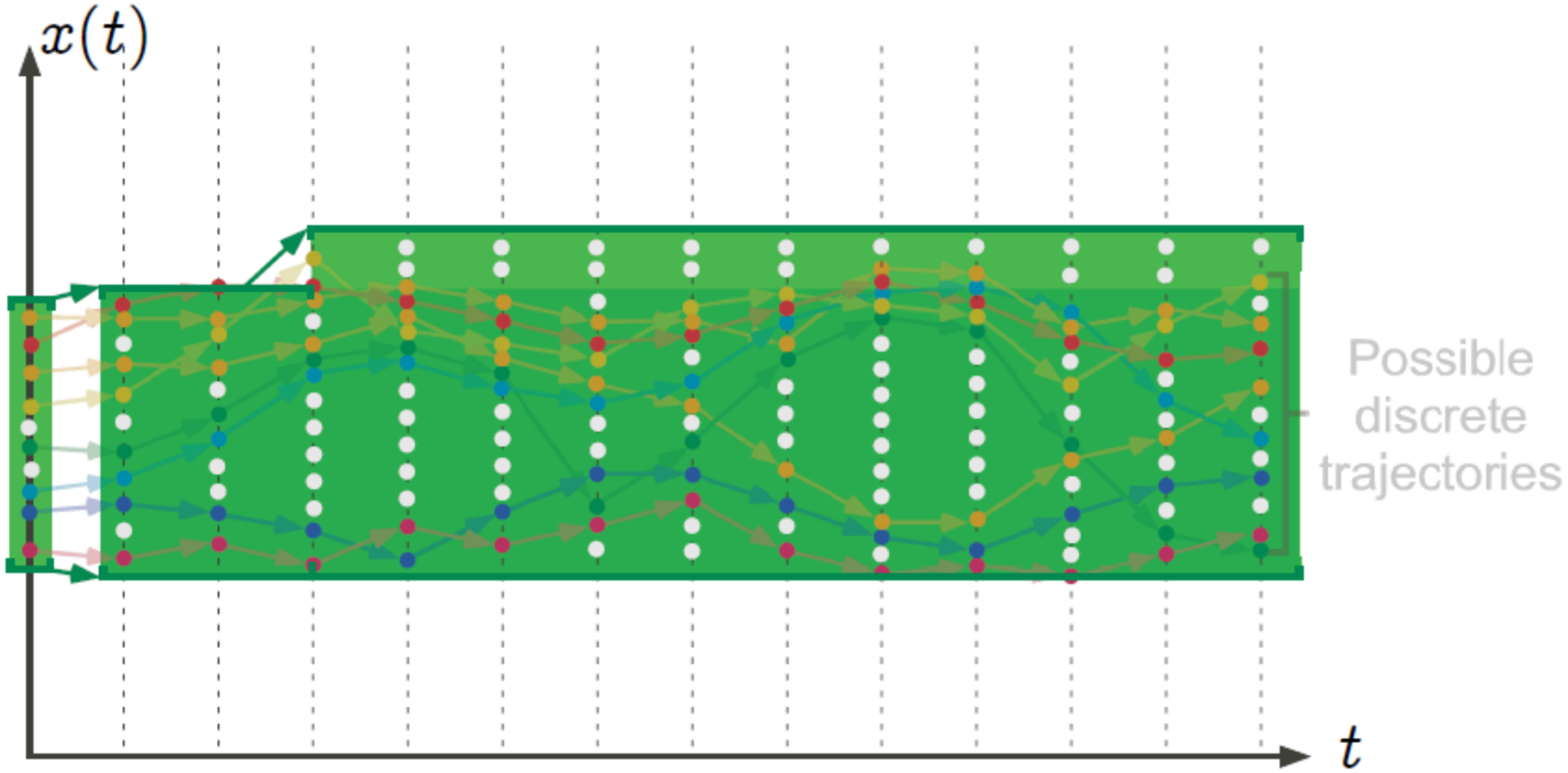
Graphic example: upward iteration with widening



Graphic example: upward iteration with widening



Graphic example: stability of the upward iteration



Widening operator

A widening operator $\nabla \in \overline{L} \times \overline{L} \mapsto \overline{L}$ is such that:

- Correctness:
  - $\forall x, y \in \overline{L} : \gamma(x) \sqsubseteq \gamma(x \nabla y)$
  - $\forall x, y \in \overline{L} : \gamma(y) \sqsubseteq \gamma(x \nabla y)$
- Convergence:
  - for all increasing chains $x^0 \sqsubseteq x^1 \sqsubseteq \ldots$, the increasing chain defined by $y^0 = x^0, \ldots, y^{i+1} = y^i \nabla x^{i+1}, \ldots$ is not strictly increasing.

Fixpoint approximation with widening

The upward iteration sequence with widening:

- $\hat{X}^0 = \overline{\bot}$ (infimum)
- $\hat{X}^{i+1} = \hat{X}^i$ if $\overline{F}(\hat{X}^i) \sqsubseteq \hat{X}^i$
  $= \hat{X}^i \nabla F(\hat{X}^i)$ otherwise

is ultimately stationary and its limit $\hat{A}$ is a sound upper approximation of $\mathsf{lfp}^{\overline{\bot}}\ \overline{F}$:

$$\mathsf{lfp}^{\overline{\bot}}\ \overline{F} \sqsubseteq \hat{A}$$

Interval widening

- $\overline{L} = \{\bot\} \cup \{[\ell, u] \mid \ell, u \in \mathbb{Z} \cup \{-\infty\} \wedge u \in \mathbb{Z} \cup \{\} \wedge \ell \leq u\}$
- The widening extrapolates unstable bounds to infinity:

$$\bot \nabla X = X$$

$$X \nabla \bot = X$$

$$[\ell_0, u_0] \nabla [\ell_1, u_1] = [\text{if } \ell_1 < \ell_0 \text{ then } -\infty \text{ else } \ell_0,$$
$$\text{if } u_1 > u_0 \text{ then } +\infty \text{ else } u_0]$$

Not monotone. For example $[0, 1] \sqsubseteq [0, 2]$ but $[0, 1] \nabla [0, 2] = [0, +\infty] \not\sqsubseteq [0, 2] = [0, 2] \nabla [0, 2]$

Example: Interval analysis (1975)

Program to be analyzed:

```
   x := 1;
1:
   while x < 10000 do
2:
        x := x + 1
3:
   od;
4:
```

Example: Interval analysis (1975)

Equations (abstract interpretation of the semantics):

```
   x := 1;
1:
   while x < 10000 do
2:
        x := x + 1
3:
   od;
4:
```

$$\begin{cases} X_1 = [1,1] \\ X_2 = (X_1 \cup X_3) \cap [-\infty, 9999] \\ X_3 = X_2 \oplus [1,1] \\ X_4 = (X_1 \cup X_3) \cap [10000, +\infty] \end{cases}$$

Example: Interval analysis (1975)

Resolution by chaotic increasing iteration:

```
   x := 1;
1:
   while x < 10000 do
2:
        x := x + 1
3:
   od;
4:
```

$$\begin{cases} X_1 = [1,1] \\ X_2 = (X_1 \cup X_3) \cap [-\infty, 9999] \\ X_3 = X_2 \oplus [1,1] \\ X_4 = (X_1 \cup X_3) \cap [10000, +\infty] \end{cases}$$

$$\begin{cases} X_1 = \emptyset \\ X_2 = \emptyset \\ X_3 = \emptyset \\ X_4 = \emptyset \end{cases}$$

Example: Interval analysis (1975)

Increasing chaotic iteration:

```
   x := 1;
1:
   while x < 10000 do
2:
        x := x + 1
3:
   od;
4:
```

$$\begin{cases} X_1 = [1,1] \\ X_2 = (X_1 \cup X_3) \cap [-\infty, 9999] \\ X_3 = X_2 \oplus [1,1] \\ X_4 = (X_1 \cup X_3) \cap [10000, +\infty] \end{cases}$$

$$\begin{cases} X_1 = [1,1] \\ X_2 = \emptyset \\ X_3 = \emptyset \\ X_4 = \emptyset \end{cases}$$

Example: Interval analysis (1975)

Increasing chaotic iteration:

```
   x := 1;
1:
   while x < 10000 do
2:
        x := x + 1
3:
   od;
4:
```

$$\begin{cases} X_1 = [1,1] \\ X_2 = (X_1 \cup X_3) \cap [-\infty, 9999] \\ X_3 = X_2 \oplus [1,1] \\ X_4 = (X_1 \cup X_3) \cap [10000, +\infty] \end{cases}$$

$$\begin{cases} X_1 = [1,1] \\ X_2 = [1,1] \\ X_3 = \emptyset \\ X_4 = \emptyset \end{cases}$$

Example: Interval analysis (1975)

Increasing chaotic iteration:

```
   x := 1;
1:
   while x < 10000 do
2:
        x := x + 1
3:
   od;
4:
```

$$\begin{cases} X_1 = [1,1] \\ X_2 = (X_1 \cup X_3) \cap [-\infty, 9999] \\ X_3 = X_2 \oplus [1,1] \\ X_4 = (X_1 \cup X_3) \cap [10000, +\infty] \end{cases}$$

$$\begin{cases} X_1 = [1,1] \\ X_2 = [1,1] \\ X_3 = [2,2] \\ X_4 = \emptyset \end{cases}$$

Example: Interval analysis (1975)

Increasing chaotic iteration:

```
   x := 1;
1:
   while x < 10000 do
2:
        x := x + 1
3:
   od;
4:
```

$$\begin{cases} X_1 = [1,1] \\ X_2 = (X_1 \cup X_3) \cap [-\infty, 9999] \\ X_3 = X_2 \oplus [1,1] \\ X_4 = (X_1 \cup X_3) \cap [10000, +\infty] \end{cases}$$

$$\begin{cases} X_1 = [1,1] \\ X_2 = [1,2] \\ X_3 = [2,2] \\ X_4 = \emptyset \end{cases}$$

Example: Interval analysis (1975)

Increasing chaotic iteration: **convergence !**

```
   x := 1;
1:
   while x < 10000 do
2:
        x := x + 1
3:
   od;
4:
```

$$\begin{cases} X_1 = [1,1] \\ X_2 = (X_1 \cup X_3) \cap [-\infty, 9999] \\ X_3 = X_2 \oplus [1,1] \\ X_4 = (X_1 \cup X_3) \cap [10000, +\infty] \end{cases}$$

$$\begin{cases} X_1 = [1,1] \\ X_2 = [1,2] \\ X_3 = [2,3] \\ X_4 = \emptyset \end{cases}$$

Example: Interval analysis (1975)

Increasing chaotic iteration: **convergence !!**

```
   x := 1;
1:
   while x < 10000 do
2:
        x := x + 1
3:
   od;
4:
```

$$\begin{cases} X_1 = [1,1] \\ X_2 = (X_1 \cup X_3) \cap [-\infty, 9999] \\ X_3 = X_2 \oplus [1,1] \\ X_4 = (X_1 \cup X_3) \cap [10000, +\infty] \end{cases}$$

$$\begin{cases} X_1 = [1,1] \\ X_2 = [1,3] \\ X_3 = [2,3] \\ X_4 = \emptyset \end{cases}$$

Example: Interval analysis (1975)

Increasing chaotic iteration: **convergence !!!**

```
    x := 1;
1:
    while x < 10000 do
2:
        x := x + 1
3:
    od;
4:
```

$$\begin{cases} X_1 = [1,1] \\ X_2 = (X_1 \cup X_3) \cap [-\infty, 9999] \\ X_3 = X_2 \oplus [1,1] \\ X_4 = (X_1 \cup X_3) \cap [10000, +\infty] \end{cases}$$

$$\begin{cases} X_1 = [1,1] \\ X_2 = [1,3] \\ X_3 = [2,4] \\ X_4 = \emptyset \end{cases}$$

Example: Interval analysis (1975)

Increasing chaotic iteration: **convergence !!!!**

```
   x := 1;
1:
   while x < 10000 do
2:
       x := x + 1
3:
   od;
4:
```

$$\begin{cases} X_1 = [1,1] \\ X_2 = (X_1 \cup X_3) \cap [-\infty, 9999] \\ X_3 = X_2 \oplus [1,1] \\ X_4 = (X_1 \cup X_3) \cap [10000, +\infty] \end{cases}$$

$$\begin{cases} X_1 = [1,1] \\ X_2 = [1,4] \\ X_3 = [2,4] \\ X_4 = \emptyset \end{cases}$$

Example: Interval analysis (1975)

Increasing chaotic iteration: **convergence !!!!!**

```
    x := 1;
1:
    while x < 10000 do
2:
        x := x + 1
3:
    od;
4:
```

$$\begin{cases} X_1 = [1,1] \\ X_2 = (X_1 \cup X_3) \cap [-\infty, 9999] \\ X_3 = X_2 \oplus [1,1] \\ X_4 = (X_1 \cup X_3) \cap [10000, +\infty] \end{cases}$$

$$\begin{cases} X_1 = [1,1] \\ X_2 = [1,4] \\ X_3 = [2,5] \\ X_4 = \emptyset \end{cases}$$

Example: Interval analysis (1975)

Increasing chaotic iteration: **convergence !!!!!!**

```
   x := 1;
1:
   while x < 10000 do
2:
        x := x + 1
3:
   od;
4:
```

$$\begin{cases} X_1 = [1,1] \\ X_2 = (X_1 \cup X_3) \cap [-\infty, 9999] \\ X_3 = X_2 \oplus [1,1] \\ X_4 = (X_1 \cup X_3) \cap [10000, +\infty] \end{cases}$$

$$\begin{cases} X_1 = [1,1] \\ X_2 = [1,5] \\ X_3 = [2,5] \\ X_4 = \emptyset \end{cases}$$

Example: Interval analysis (1975)

Increasing chaotic iteration: **convergence !!!!!!!**

```
    x := 1;
1:
    while x < 10000 do
2:
        x := x + 1
3:
    od;
4:
```

$$\begin{cases} X_1 = [1,1] \\ X_2 = (X_1 \cup X_3) \cap [-\infty, 9999] \\ X_3 = X_2 \oplus [1,1] \\ X_4 = (X_1 \cup X_3) \cap [10000, +\infty] \end{cases}$$

$$\begin{cases} X_1 = [1,1] \\ X_2 = [1,5] \\ X_3 = [2,6] \\ X_4 = \emptyset \end{cases}$$

Example: Interval analysis (1975)

Convergence speed-up by widening:

```
    x := 1;
1:
    while x < 10000 do
2:
        x := x + 1
3:
    od;
4:
```

$$\begin{cases} X_1 = [1,1] \\ X_2 = (X_1 \cup X_3) \cap [-\infty, 9999] \\ X_3 = X_2 \oplus [1,1] \\ X_4 = (X_1 \cup X_3) \cap [10000, +\infty] \end{cases}$$

$$\begin{cases} X_1 = [1,1] \\ X_2 = [1,+\infty] \quad \Leftarrow \text{widening} \\ X_3 = [2,6] \\ X_4 = \emptyset \end{cases}$$

Example: Interval analysis (1975)

Decreasing chaotic iteration:

```
   x := 1;
1:
   while x < 10000 do
2:
        x := x + 1
3:
   od;
4:
```

$$\begin{cases} X_1 = [1,1] \\ X_2 = (X_1 \cup X_3) \cap [-\infty, 9999] \\ X_3 = X_2 \oplus [1,1] \\ X_4 = (X_1 \cup X_3) \cap [10000, +\infty] \end{cases}$$

$$\begin{cases} X_1 = [1,1] \\ X_2 = [1, +\infty] \\ X_3 = [2, +\infty] \\ X_4 = \emptyset \end{cases}$$

Example: Interval analysis (1975)

Decreasing chaotic iteration:

```
   x := 1;
1:
   while x < 10000 do
2:
        x := x + 1
3:
   od;
4:
```

$$\begin{cases} X_1 = [1,1] \\ X_2 = (X_1 \cup X_3) \cap [-\infty, 9999] \\ X_3 = X_2 \oplus [1,1] \\ X_4 = (X_1 \cup X_3) \cap [10000, +\infty] \end{cases}$$

$$\begin{cases} X_1 = [1,1] \\ X_2 = [1,9999] \\ X_3 = [2,+\infty] \\ X_4 = \emptyset \end{cases}$$

Example: Interval analysis (1975)

Decreasing chaotic iteration:

```
   x := 1;
1:
   while x < 10000 do
2:
       x := x + 1
3:
   od;
4:
```

$$\begin{cases} X_1 = [1,1] \\ X_2 = (X_1 \cup X_3) \cap [-\infty, 9999] \\ X_3 = X_2 \oplus [1,1] \\ X_4 = (X_1 \cup X_3) \cap [10000, +\infty] \end{cases}$$

$$\begin{cases} X_1 = [1,1] \\ X_2 = [1, 9999] \\ X_3 = [2, +10000] \\ X_4 = \emptyset \end{cases}$$

Example: Interval analysis (1975)

Final solution:

```
   x := 1;
1:
   while x < 10000 do
2:
        x := x + 1
3:
   od;
4:
```

$$\begin{cases} X_1 = [1,1] \\ X_2 = (X_1 \cup X_3) \cap [-\infty, 9999] \\ X_3 = X_2 \oplus [1,1] \\ X_4 = (X_1 \cup X_3) \cap [10000, +\infty] \end{cases}$$

$$\begin{cases} X_1 = [1,1] \\ X_2 = [1,9999] \\ X_3 = [2,+10000] \\ X_4 = [+10000,+10000] \end{cases}$$

Example: Interval analysis (1975)

Result of the interval analysis:

```
    x := 1;
1: {x = 1}
    while x < 10000 do
2: {x ∈ [1, 9999]}
        x := x + 1
3: {x ∈ [2, +10000]}
    od;
4: {x = 10000}
```

$$\begin{cases} X_1 = [1,1] \\ X_2 = (X_1 \cup X_3) \cap [-\infty, 9999] \\ X_3 = X_2 \oplus [1,1] \\ X_4 = (X_1 \cup X_3) \cap [10000, +\infty] \end{cases}$$

$$\begin{cases} X_1 = [1,1] \\ X_2 = [1,9999] \\ X_3 = [2,+10000] \\ X_4 = [+10000,+10000] \end{cases}$$

Example: Interval analysis (1975)

Checking absence of runtime errors with interval analysis:

```
   x := 1;
1: {x = 1}
   while x < 10000 do
2: {x ∈ [1, 9999]}
       x := x + 1          ⟵ no overflow
3: {x ∈ [2, +10000]}
   od;
4: {x = 10000}
```