A casual introduction to Abstract Interpretation

NYU, 29–30 March 2012

Patrick Cousot

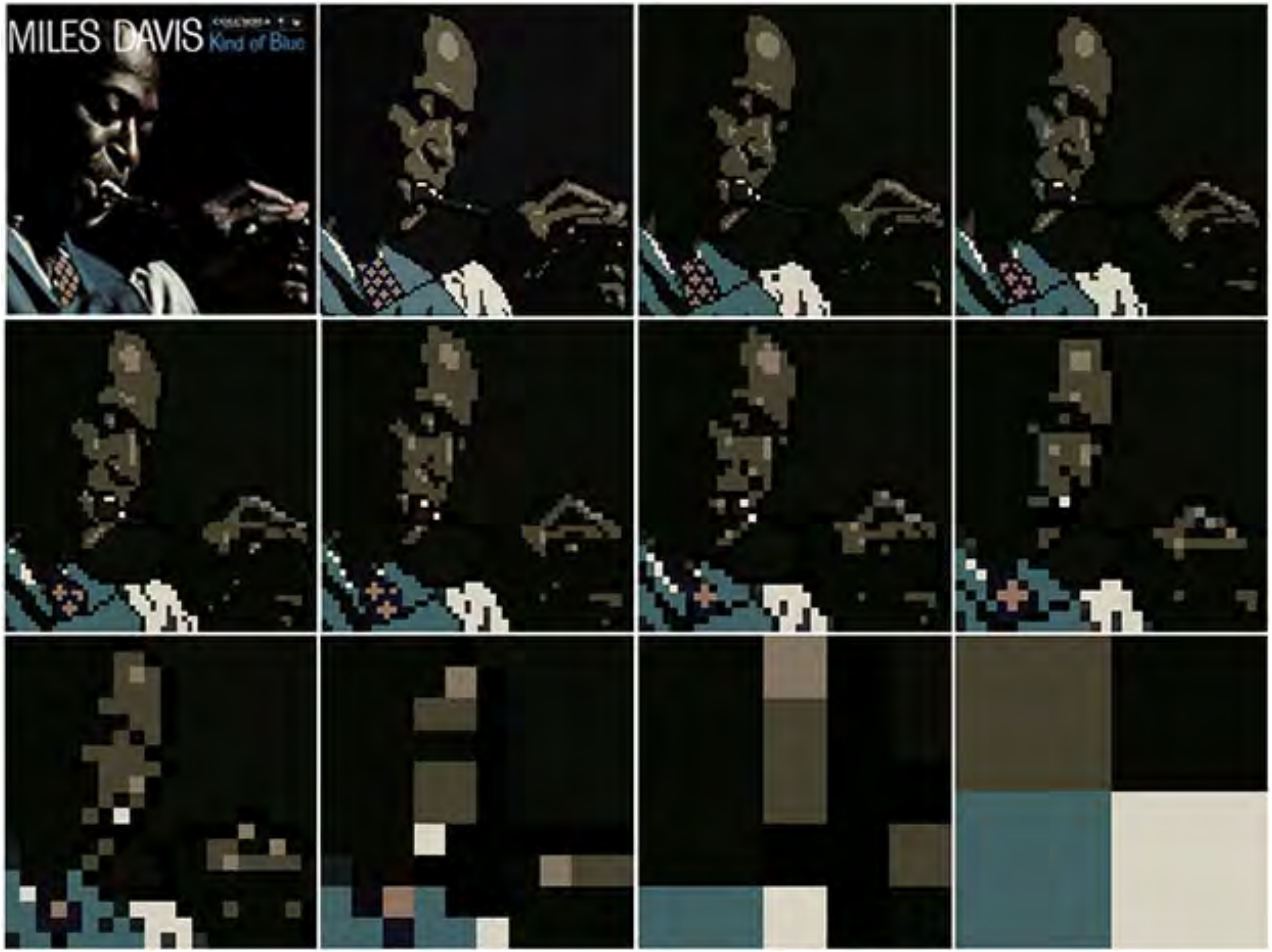
cs.nyu.edu/~pcousot di.ens.fr/~cousot

Examples of Abstractions

P. Cousot & R. Cousot. A gentle introduction to formal verification of computer systems by abstract interpretation. In *Logics and Languages for Reliability and Security*, J. Esparza, O. Grumberg, & M. Broy (Eds), NATO Science Series III: Computer and Systems Sciences, © IOS Press, 2010, Pages 1—29.

Abstractions of Dora Maar by Picasso

Pixelation of a photo by Jay Maisel



www.petapixel.com/2011/06/23/how-much-pixelation-is-needed-before-a-photo-becomes-transformed/

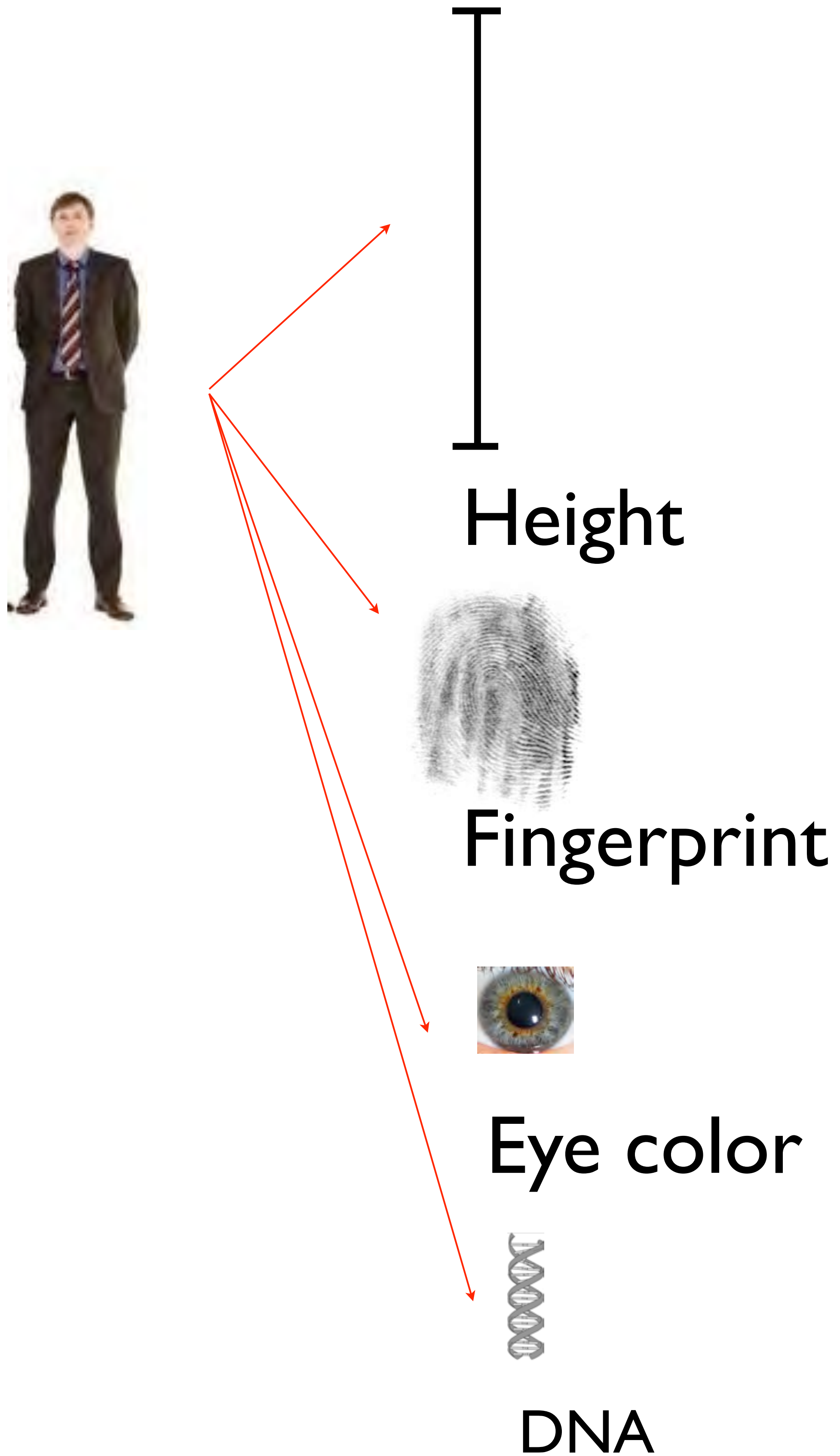
Image credit: *Photograph by Jay Maisel*

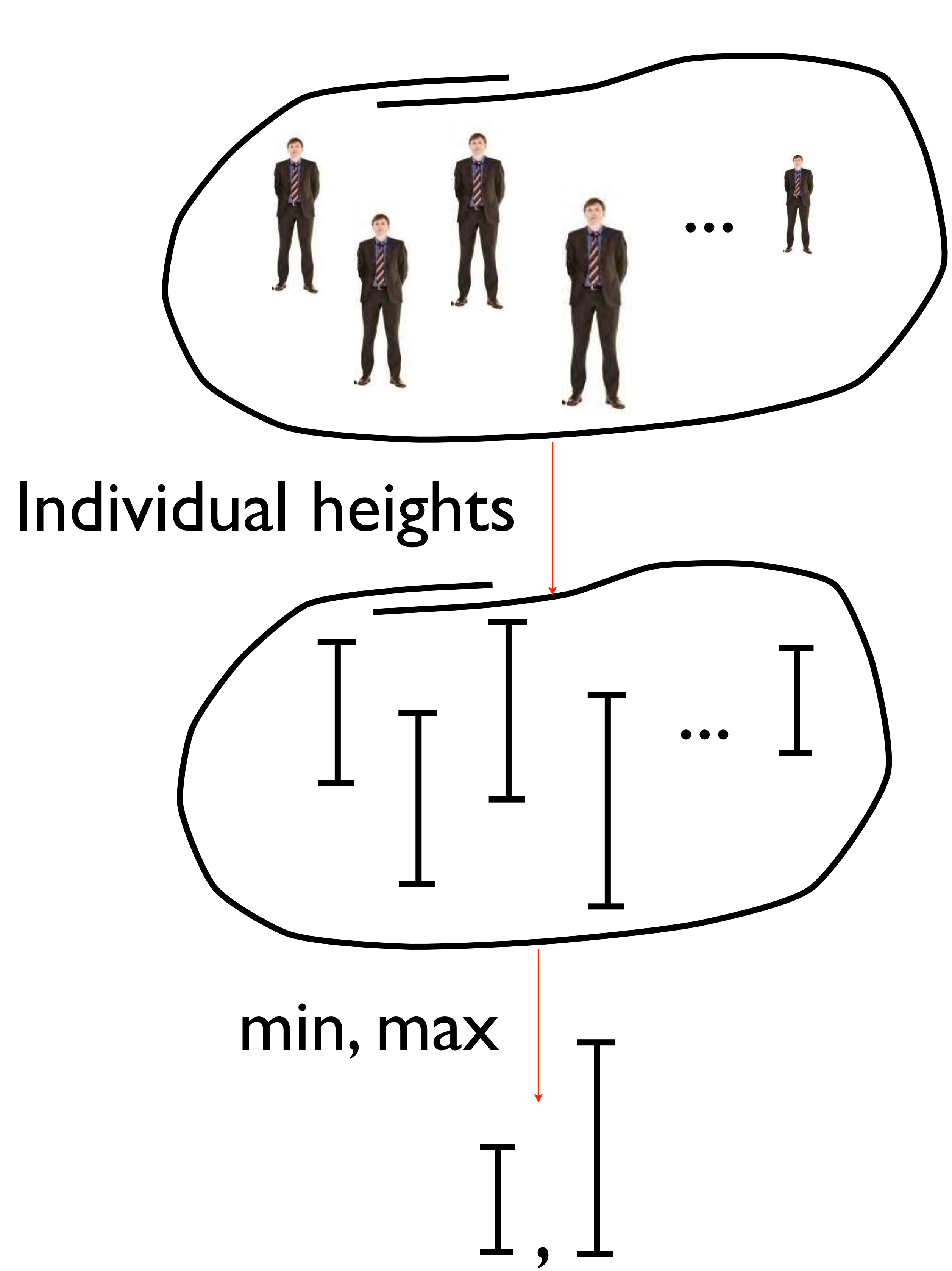
An old idea...

20 000 years old picture in a spanish cave:

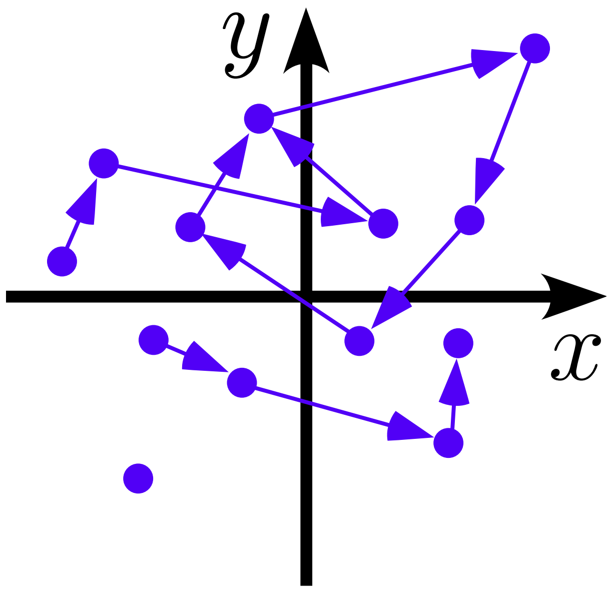
The concrete is not always well-known!

Abstractions of a man / crowd

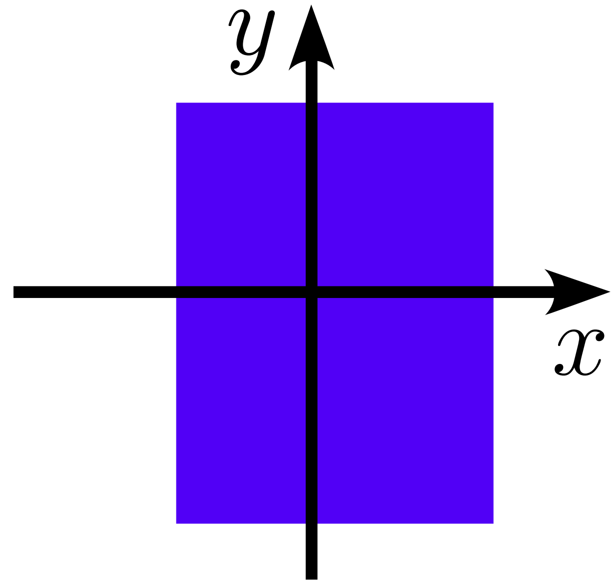




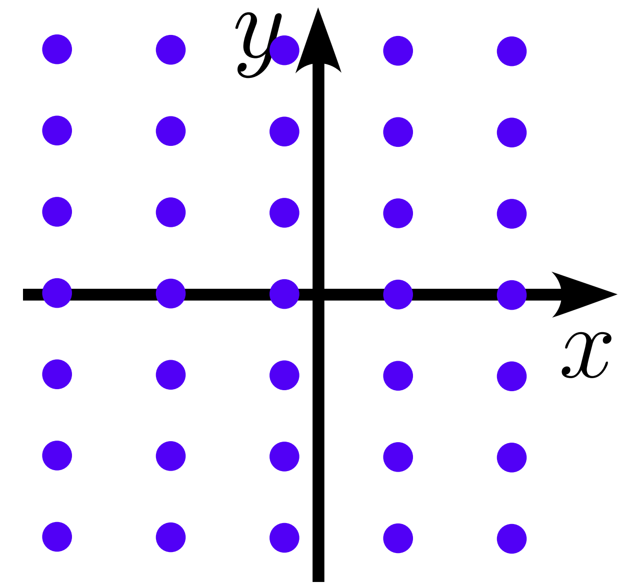
Numerical abstractions in Astrée



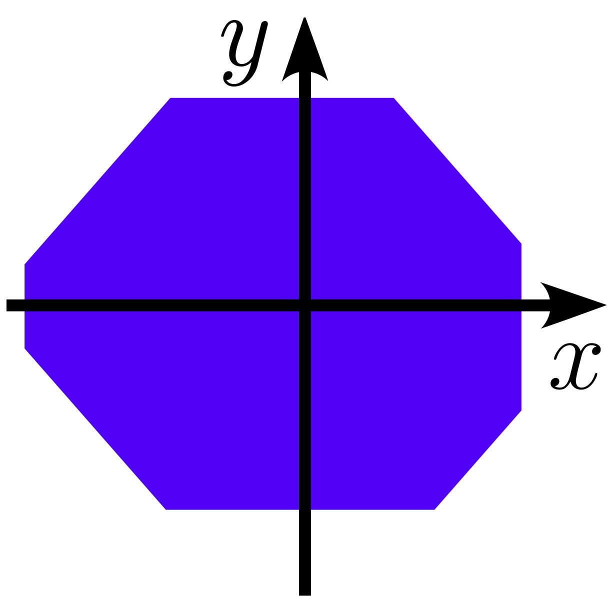
Collecting semantics:
partial traces



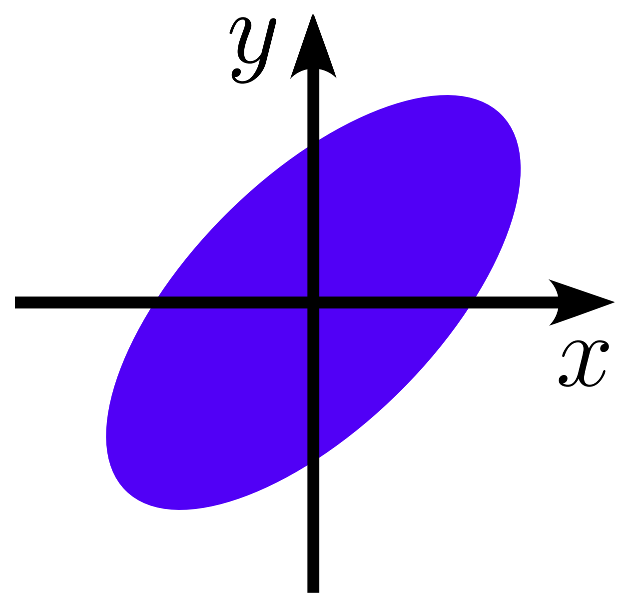
Intervals:
$\mathtt{x} \in [a, b]$



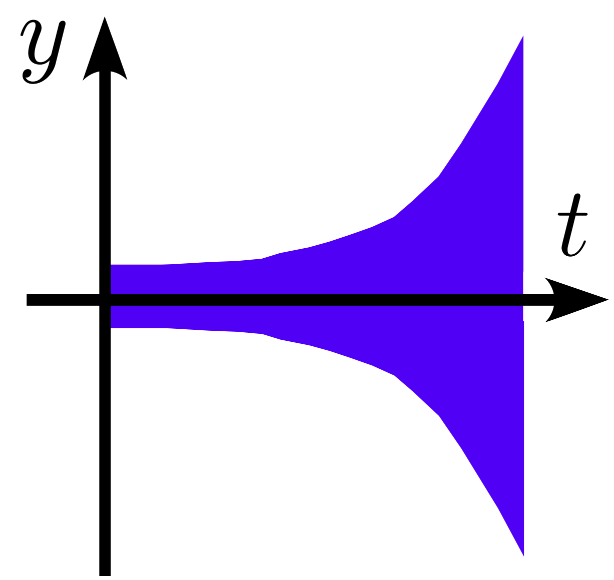
Simple congruences:
$\mathtt{x} \equiv a[b]$



Octagons:
$\pm\mathtt{x} \pm \mathtt{y} \leqslant a$



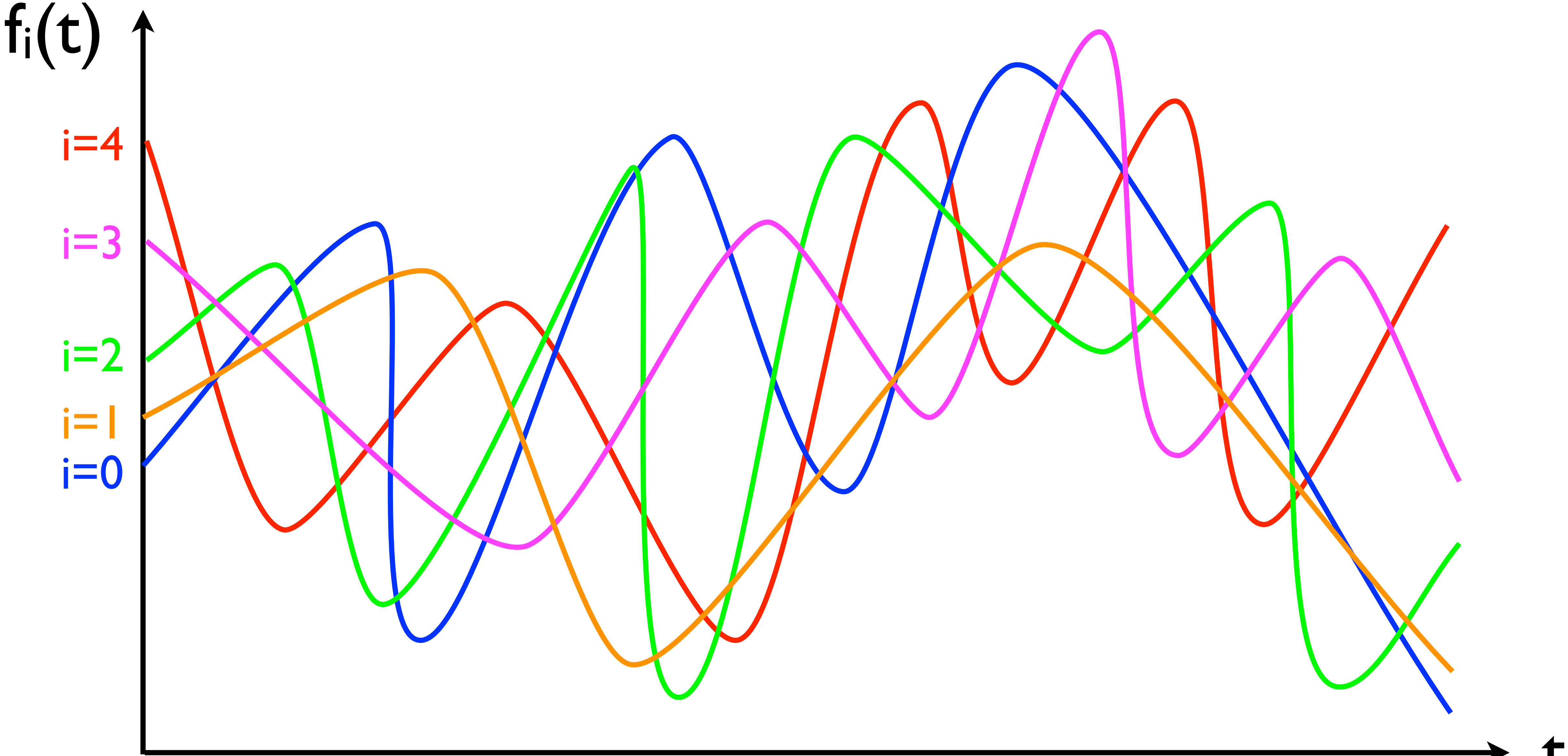
Ellipses:
$\mathtt{x}^2 + b\mathtt{y}^2 - a\mathtt{x}\mathtt{y} \leqslant d$



Exponentials:
$-a^{bt} \leqslant \mathtt{y}(t) \leqslant a^{bt}$

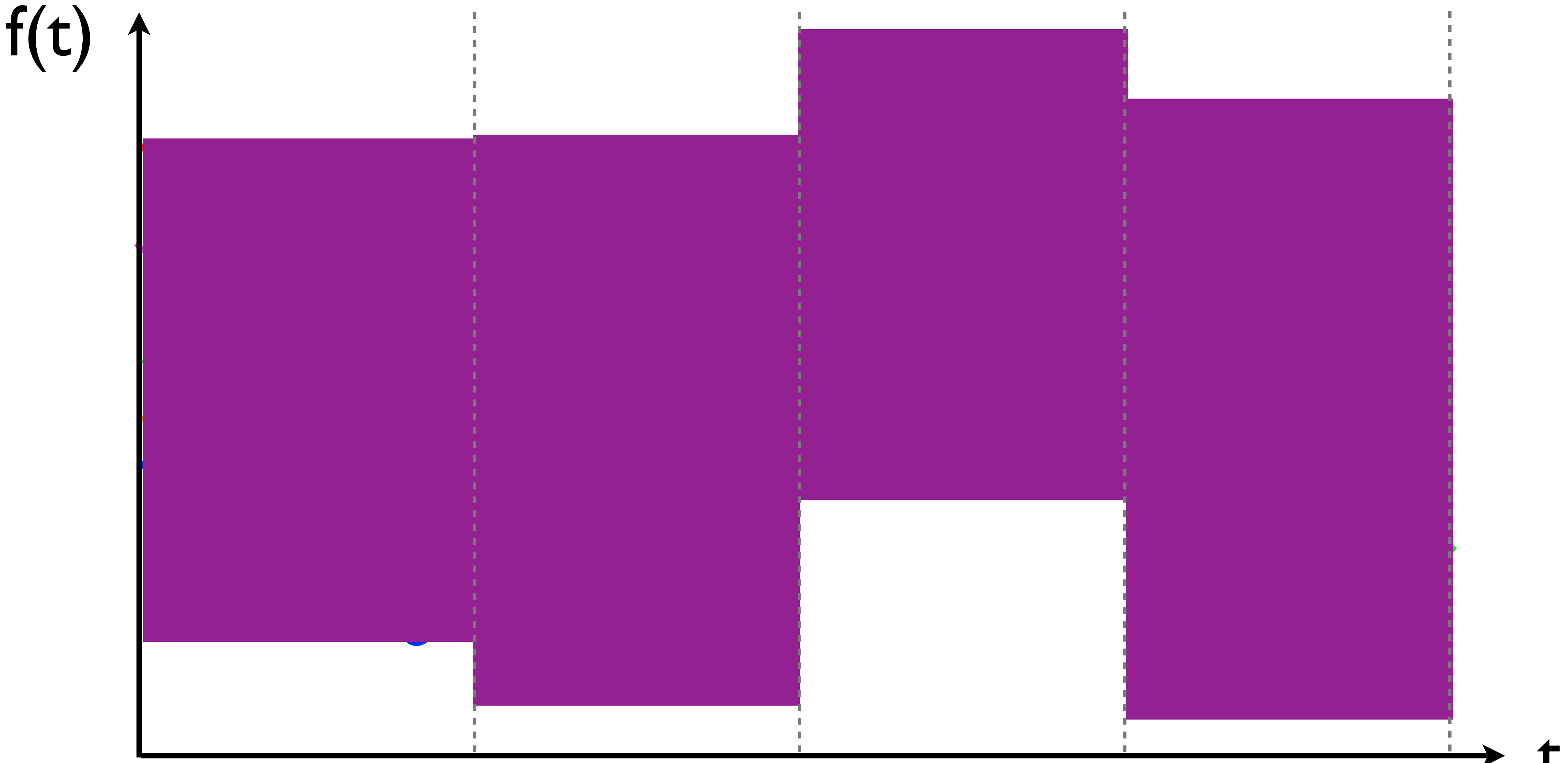
A slightly more detailled example

Set of functions

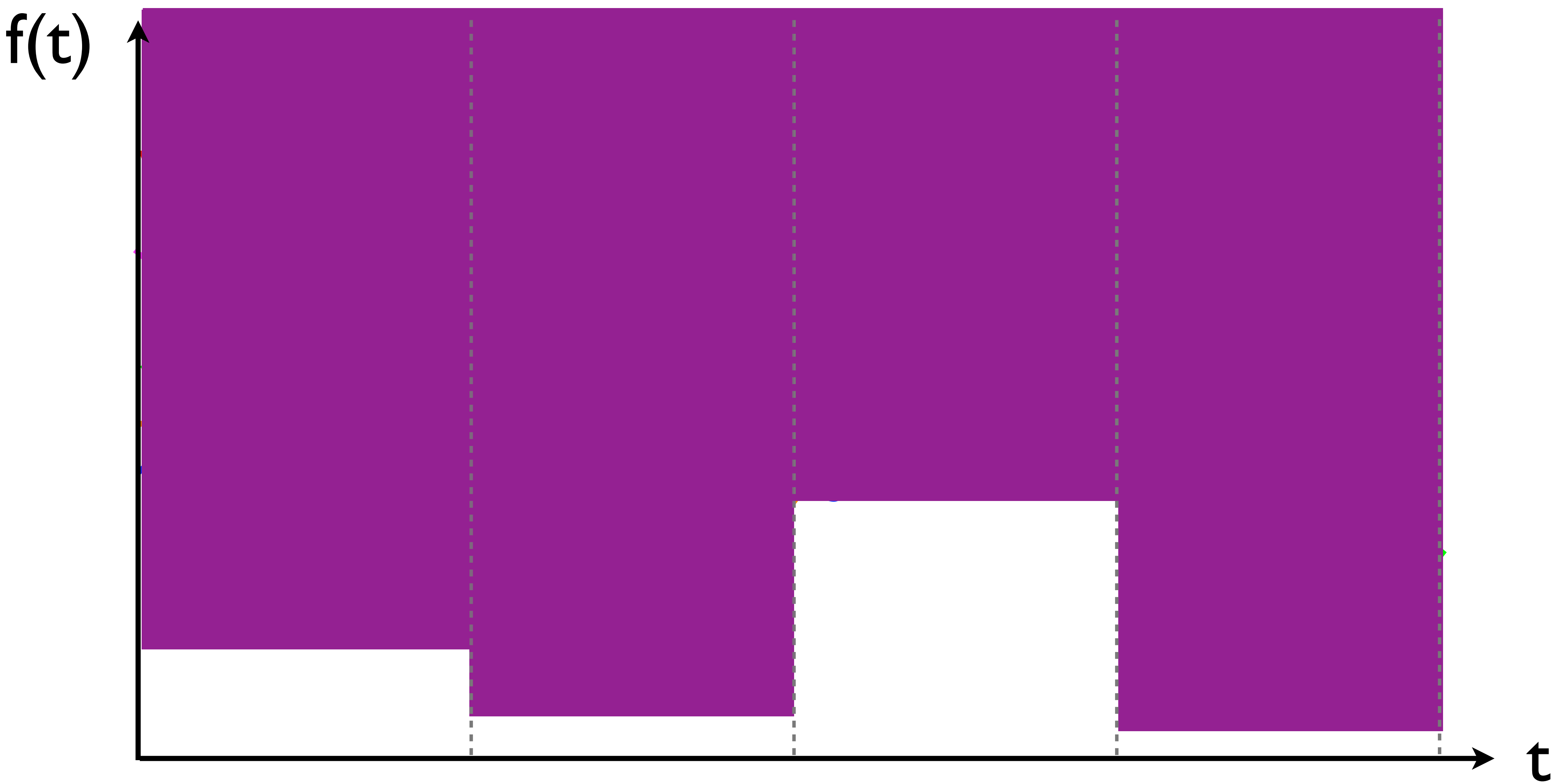


How to approximate { f_1, f_2, f_3, f_4 } ?

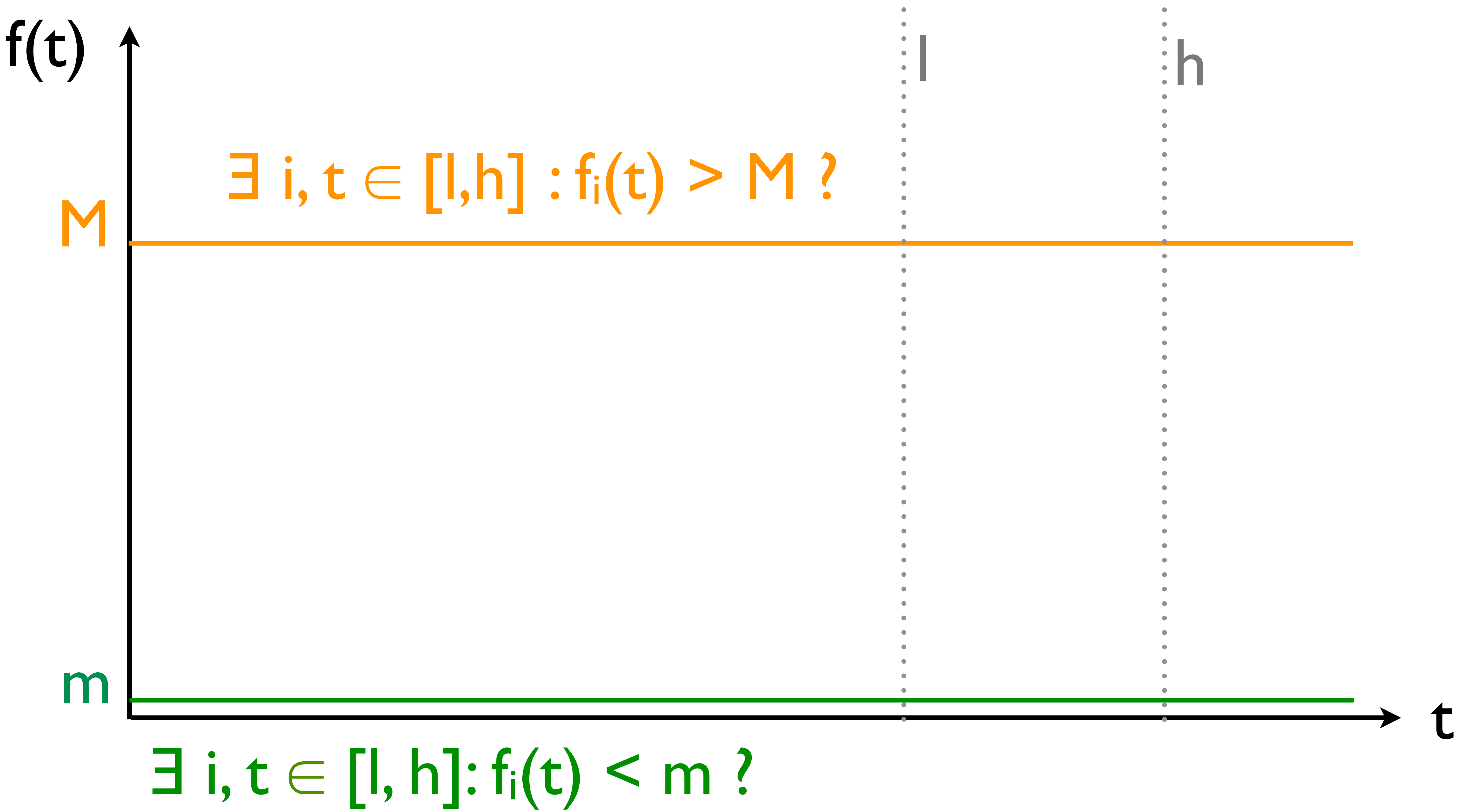
Set of functions abstraction



A less precise abstraction

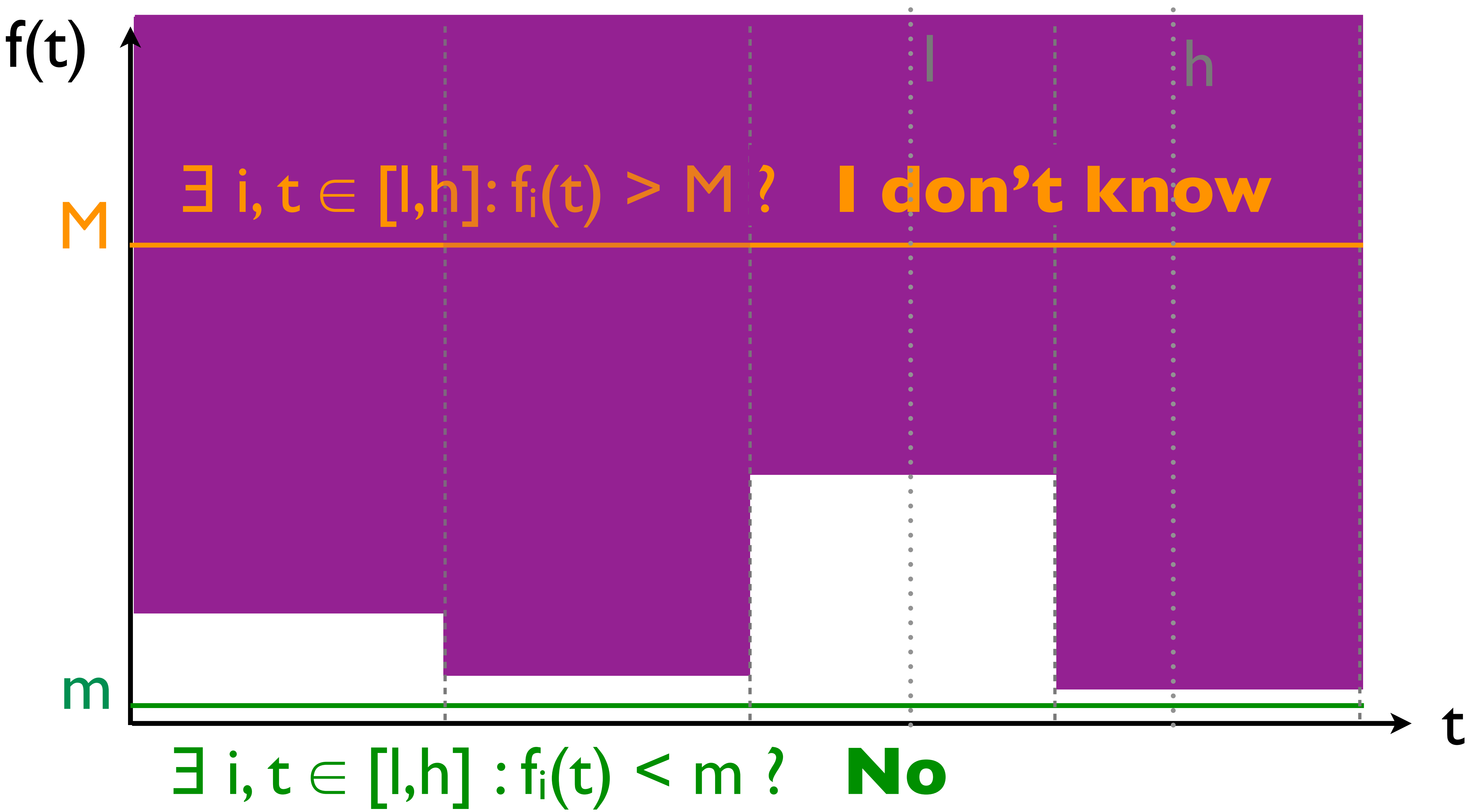


Concrete questions on the f_i



Min/max questions on the f_i

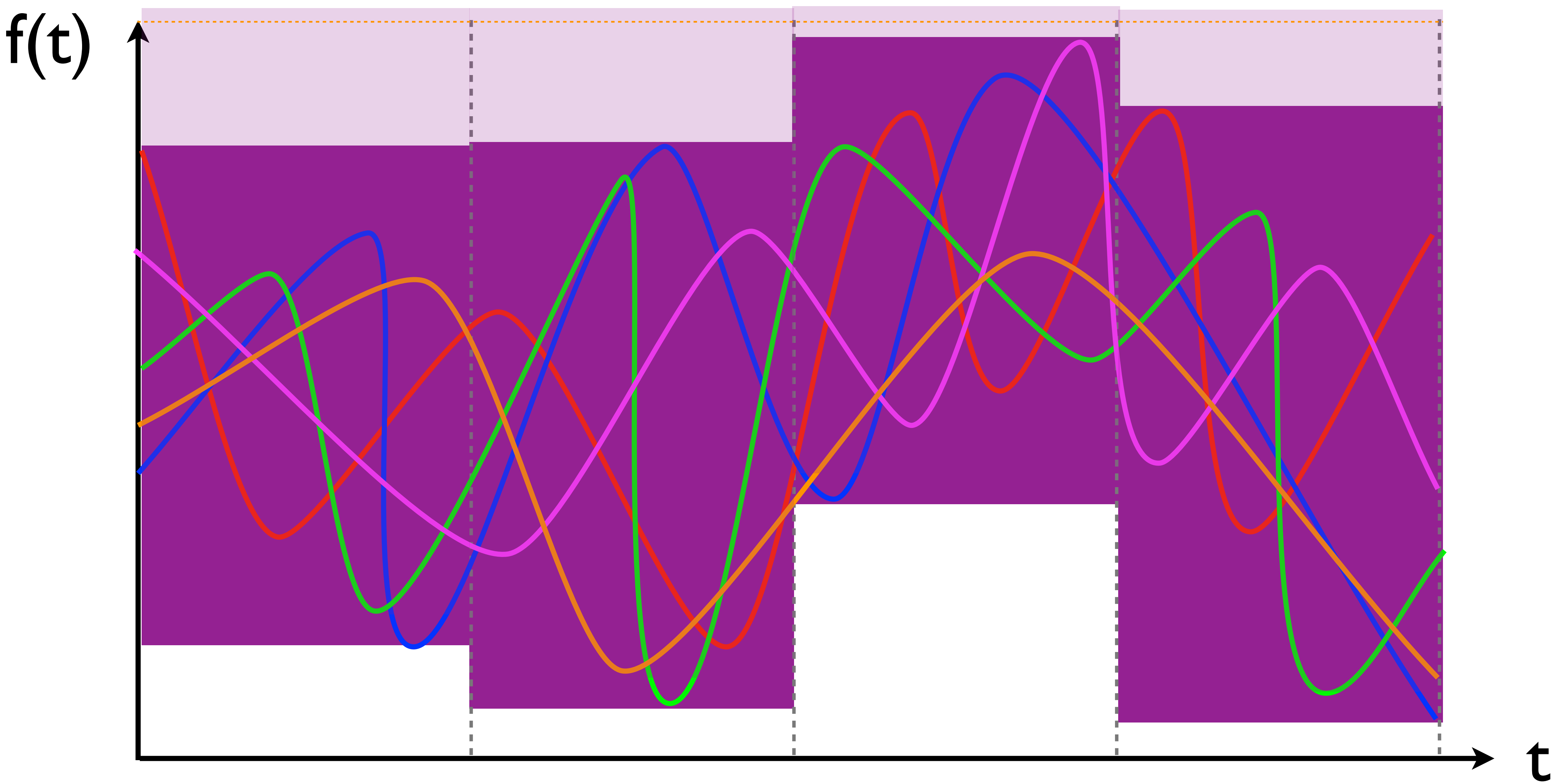
Concrete questions answered in the abstract



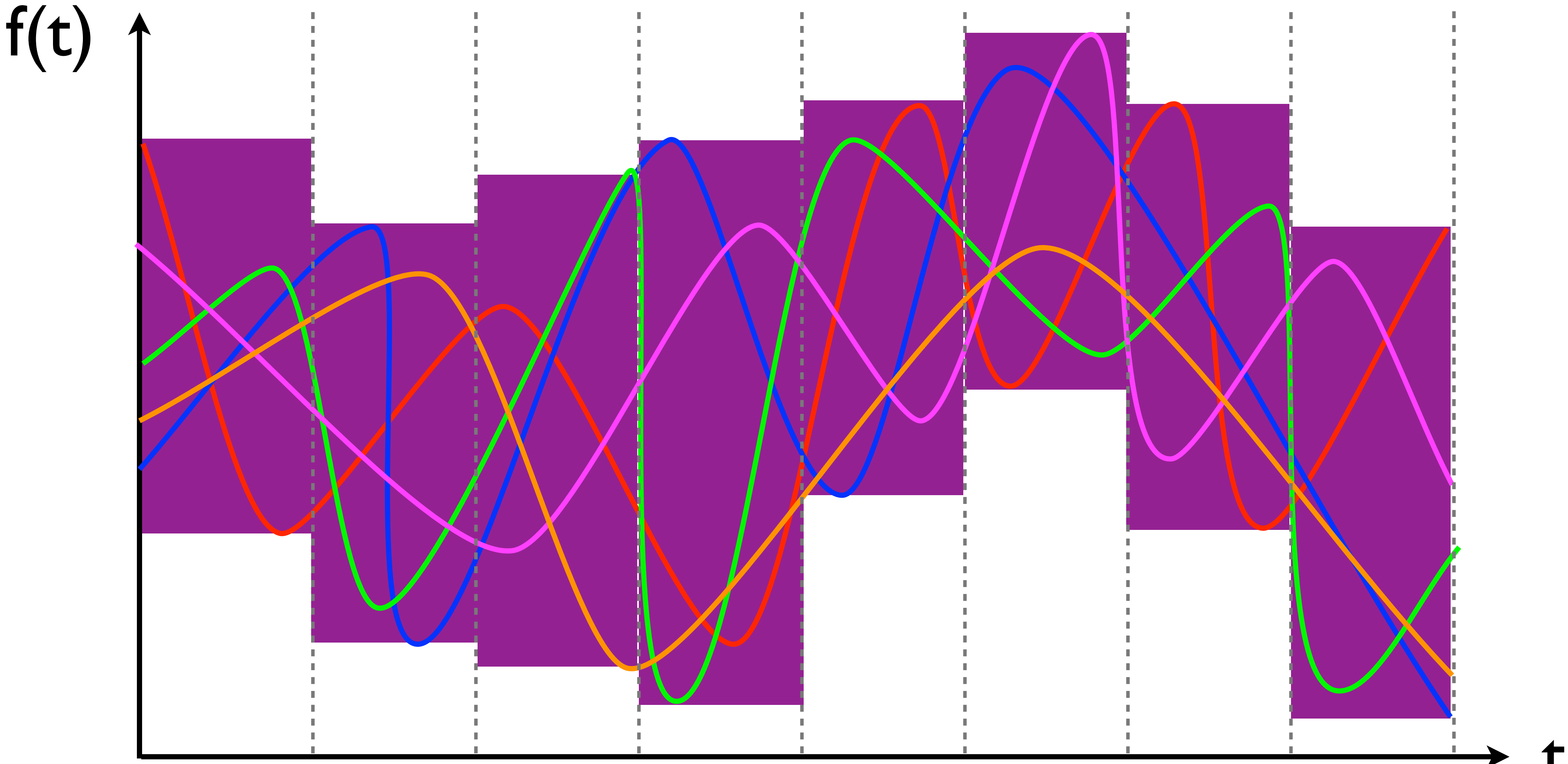
Min/max questions on the f_i

Soundness of the abstraction

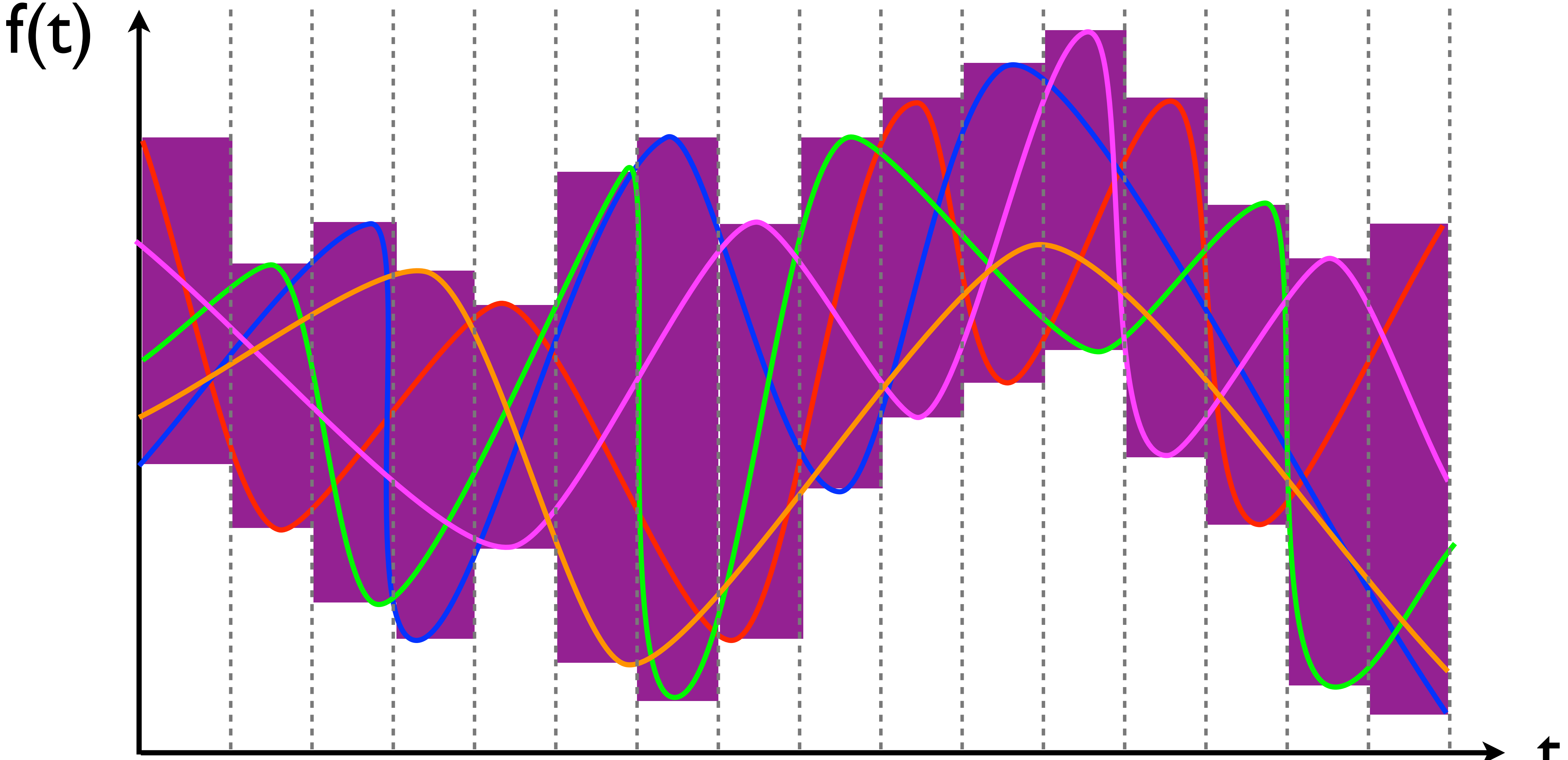
- No concrete case is ever forgotten:



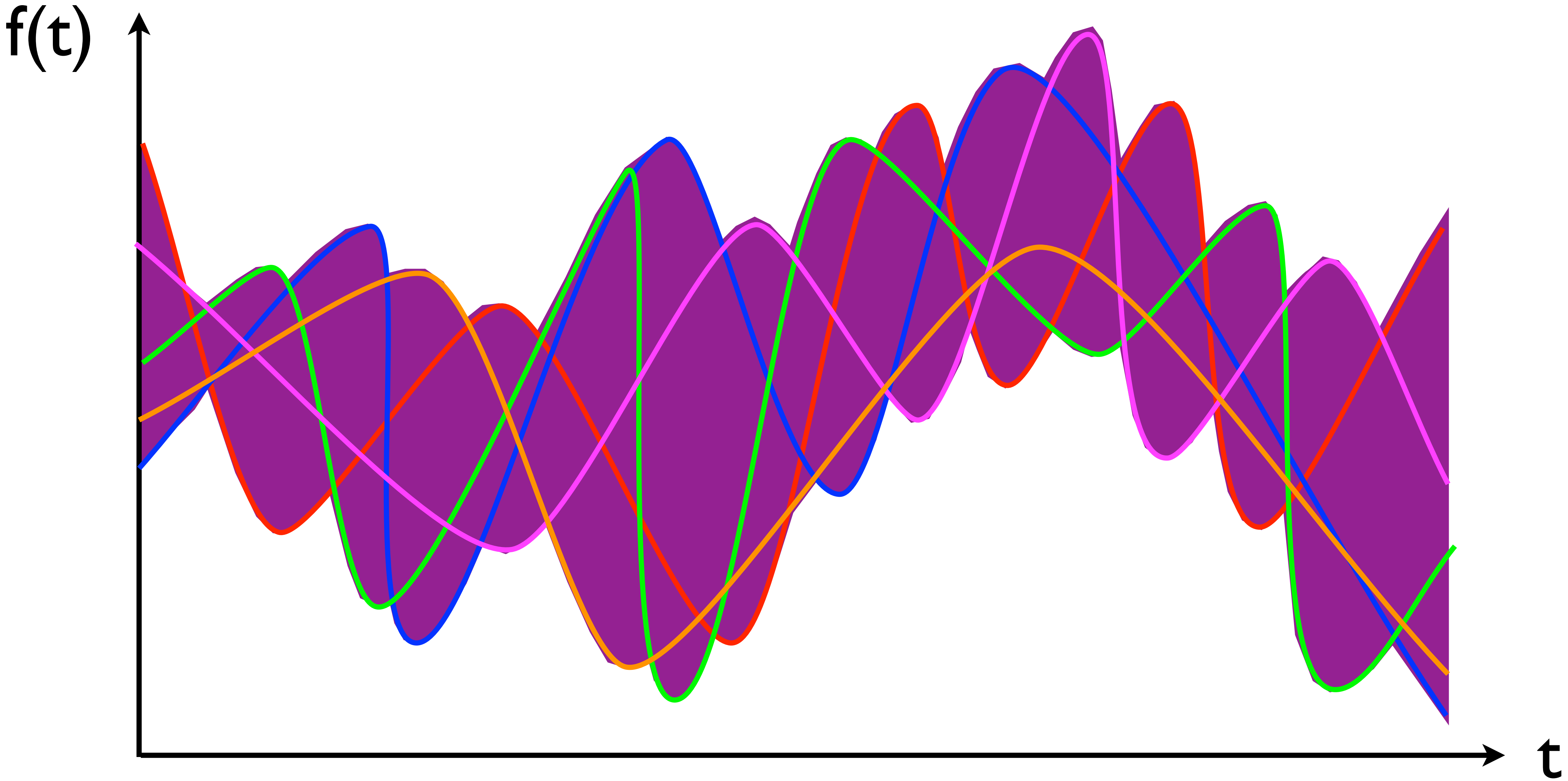
A more precise/refined abstraction



An even more precise/refined abstraction

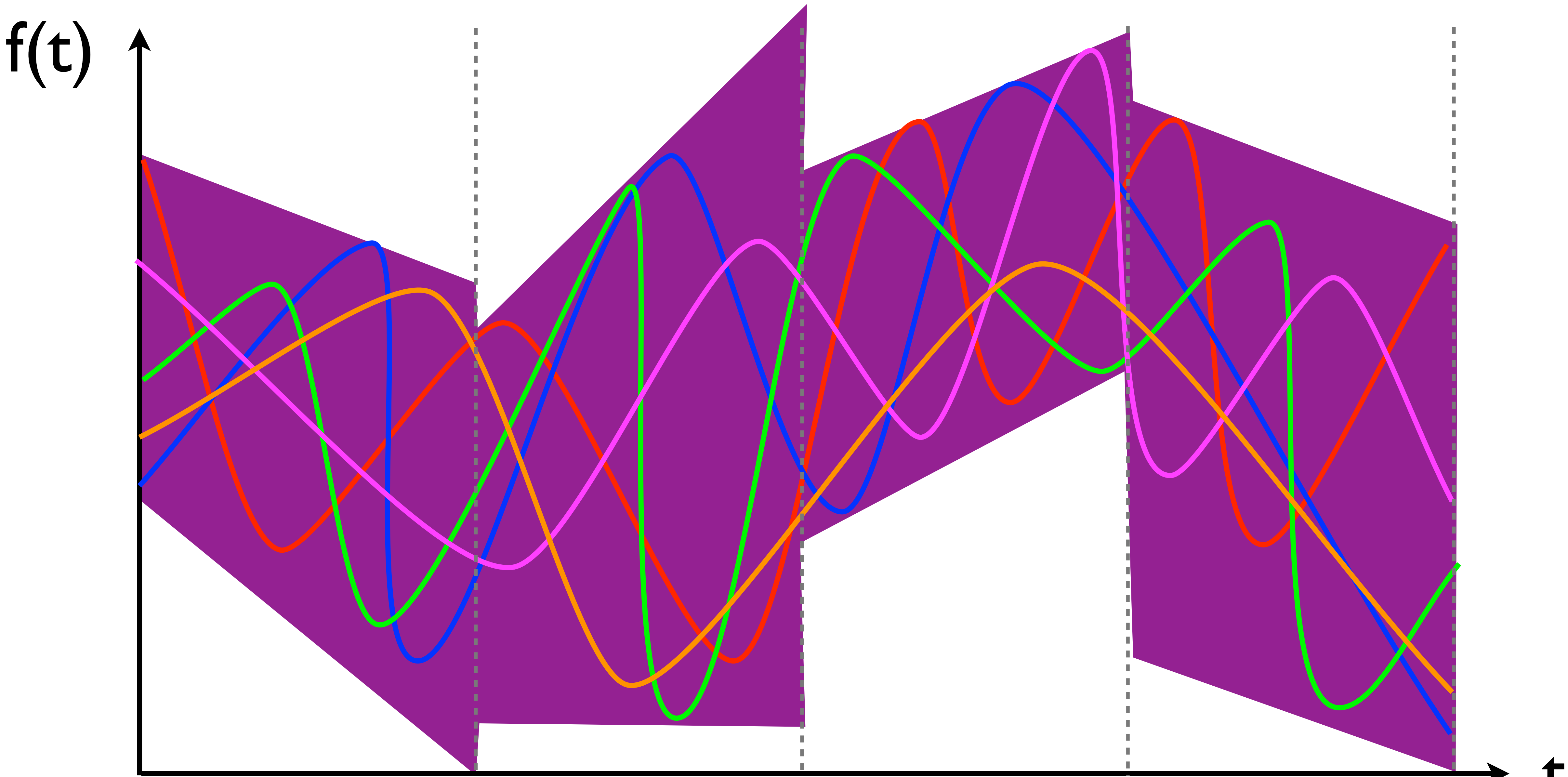


Passing to the limit



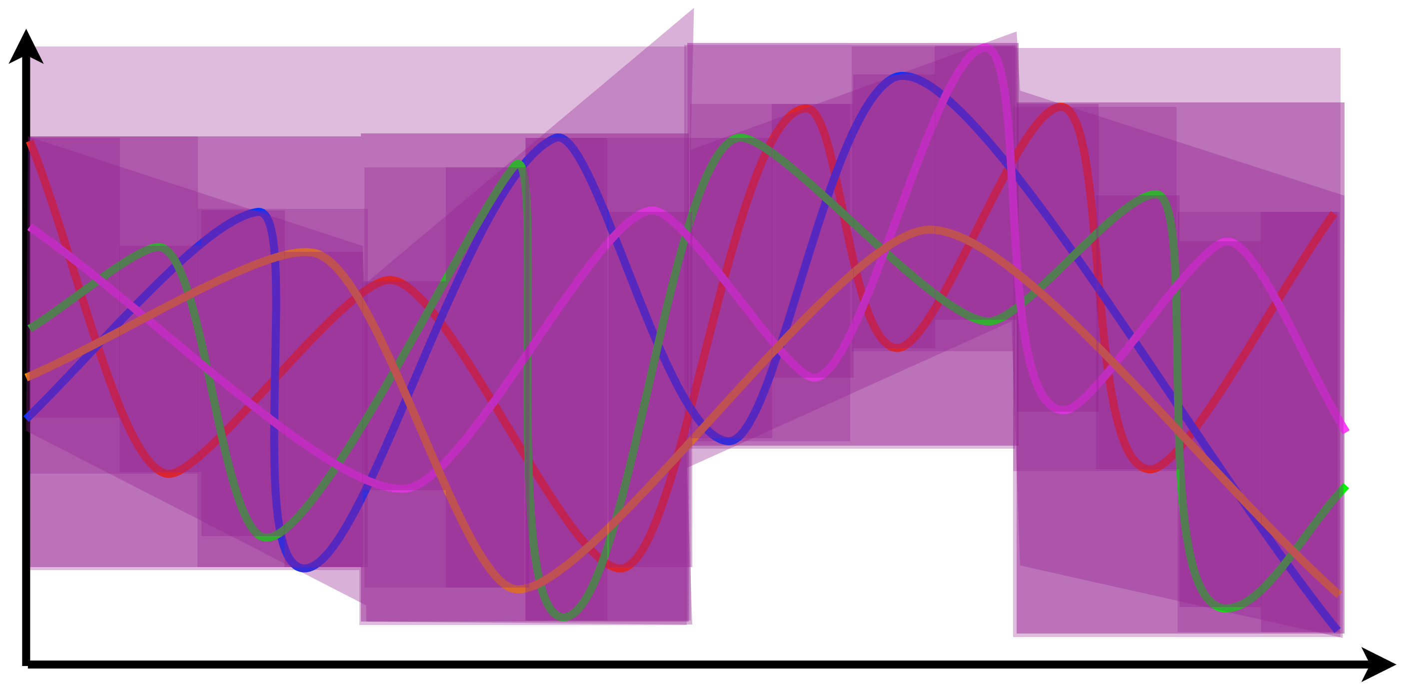
Sound and *complete* abstraction for min/max questions on the f_i

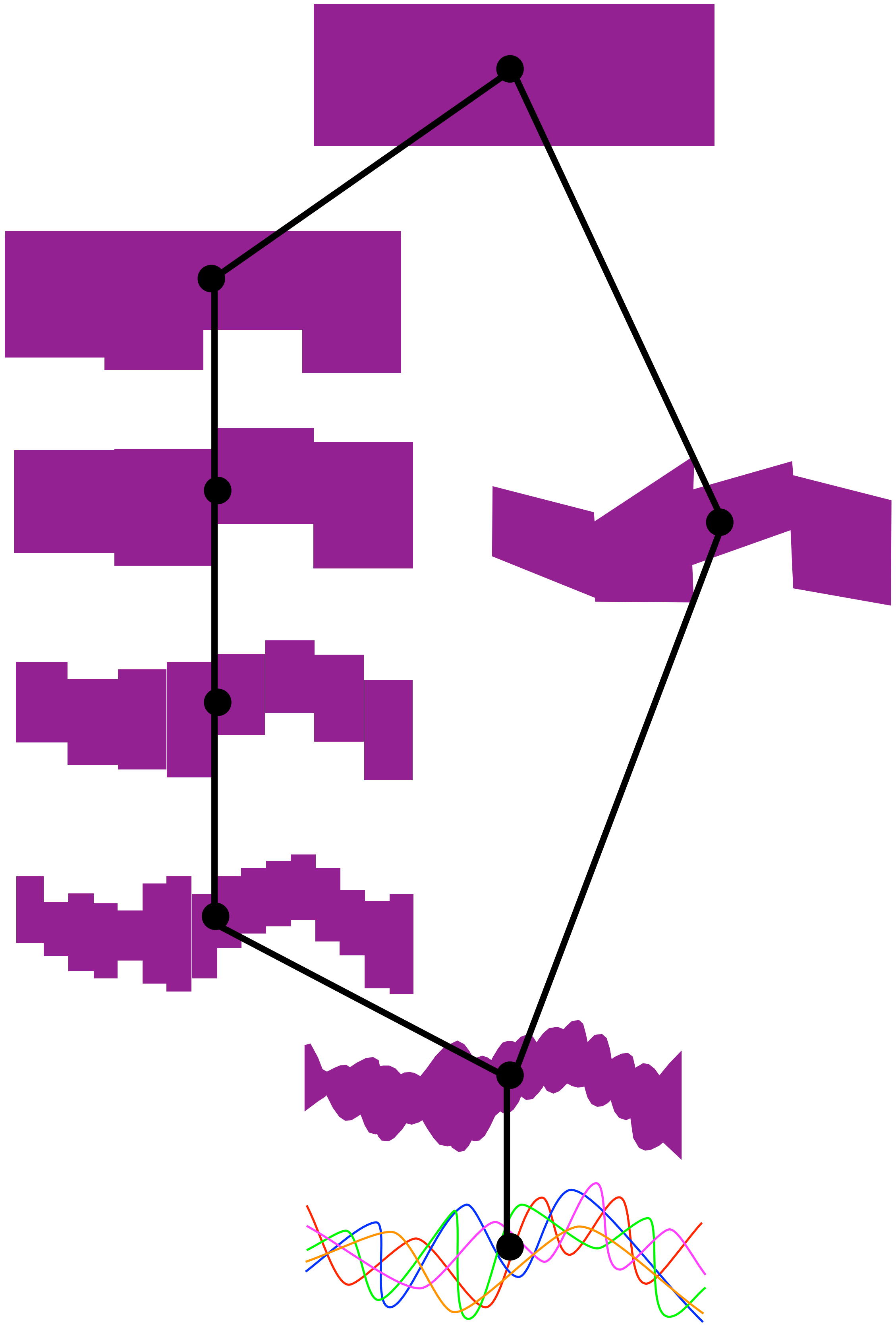
A non-comparable abstraction



Sound and <u>*in*</u>*complete* abstraction for min/max questions on the f_i

The hierarchy of abstractions





Elements of Abstract Interpretation Theory Explained with ...

Patrick Cousot & Radhia Cousot. Static Determination of Dynamic Properties of Programs. In B. Robinet, editor, *Proceedings of the second international symposium on Programming,* Paris, France, pages 106—130, April 13-15 1976, Dunod, Paris.

Patrick Cousot, Radhia Cousot: Abstract Interpretation: A Unified Lattice Model for Static Analysis of Programs by Construction or Approximation of Fixpoints. POPL 1977: 238-252

Patrick Cousot, Radhia Cousot: Systematic Design of Program Analysis Frameworks. POPL 1979: 269-282

Patrick Cousot. Méthodes itératives de construction et d'approximation de points fixes d'opérateurs monotones sur un treillis, analyse sémantique des programmes. *Thèse És Sciences Mathématiques*, Université Joseph Fourier, Grenoble, France, 21 March 1978

Patrick Cousot. Semantic foundations of program analysis. In S.S. Muchnick & N.D. Jones, editors, *Program Flow Analysis: Theory and Applications,* Ch. 10, pages 303—342, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, U.S.A., 1981.

Elements of Abstract Interpretation Theory Explained with ... Flowers

Patrick Cousot & Radhia Cousot. Static Determination of Dynamic Properties of Programs. In B. Robinet, editor, *Proceedings of the second international symposium on Programming*, Paris, France, pages 106—130, April 13-15 1976, Dunod, Paris.

Patrick Cousot, Radhia Cousot: Abstract Interpretation: A Unified Lattice Model for Static Analysis of Programs by Construction or Approximation of Fixpoints. POPL 1977: 238-252

Patrick Cousot, Radhia Cousot: Systematic Design of Program Analysis Frameworks. POPL 1979: 269-282

Patrick Cousot. Méthodes itératives de construction et d'approximation de points fixes d'opérateurs monotones sur un treillis, analyse sémantique des programmes. *Thèse És Sciences Mathématiques*, Université Joseph Fourier, Grenoble, France, 21 March 1978

Patrick Cousot. Semantic foundations of program analysis. In S.S. Muchnick & N.D. Jones, editors, *Program Flow Analysis: Theory and Applications,* Ch. 10, pages 303—342, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, U.S.A., 1981.

The concrete world

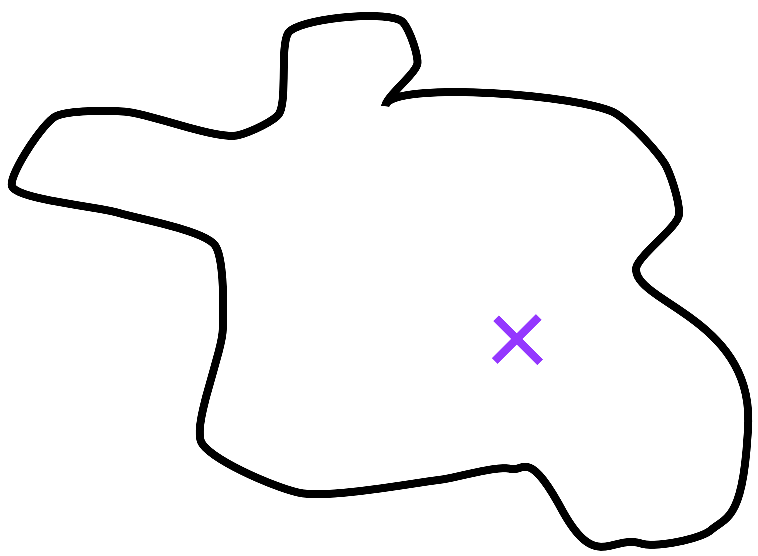
A mini graphical language

- Objects $o \in O$
- Operations on objects $O^n \longrightarrow O$, $n \geq 0$
- Logical operations on objects $O^n \longrightarrow \mathit{Booleans}$, $n \geq 0$

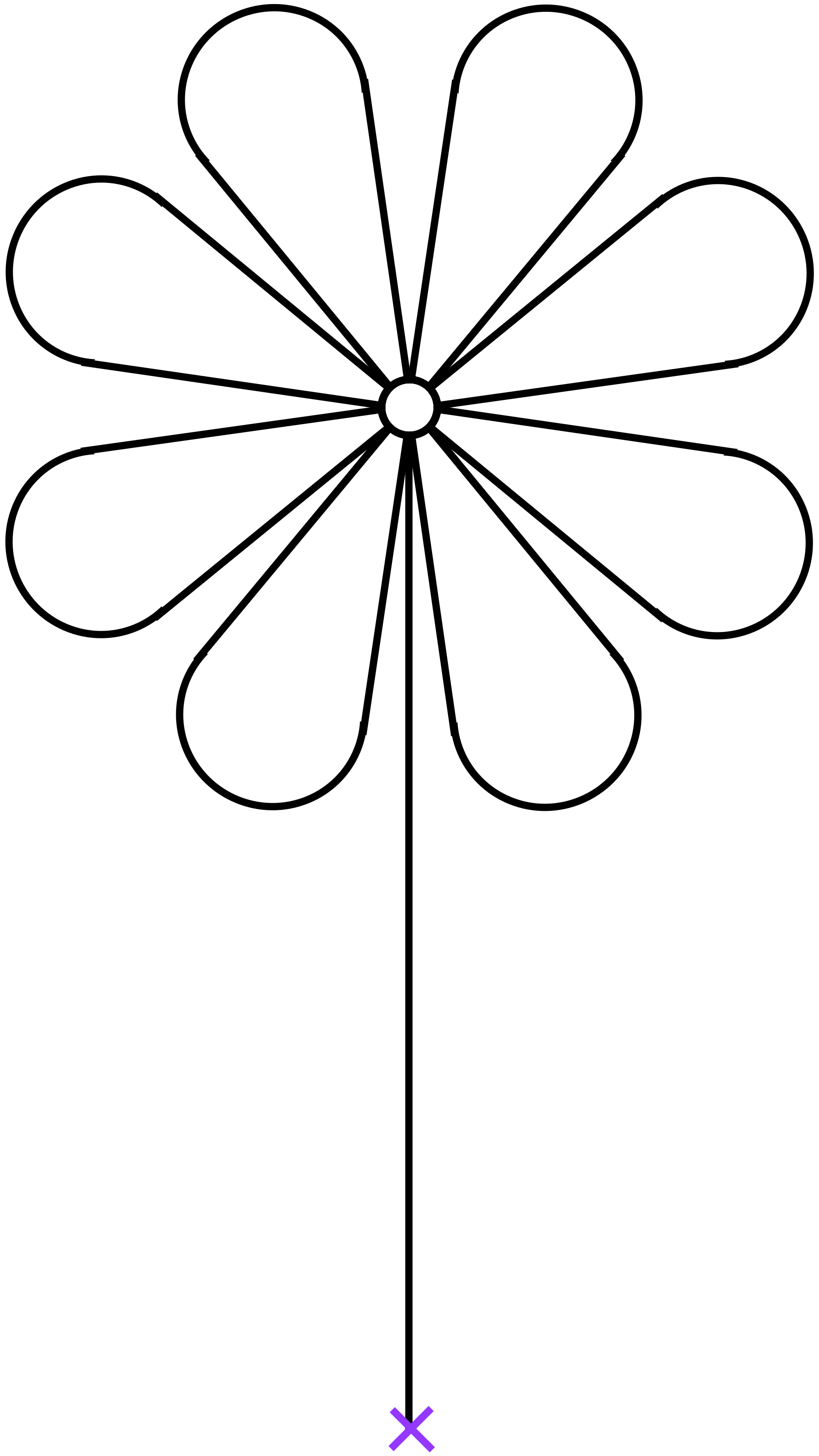
Objects

- An object $o \in \mathcal{O}$ is defined by
  - An origin (a reference point ×)
  - A set of (infinitely small) black pixels (on a white background)

- Example 1 of object:



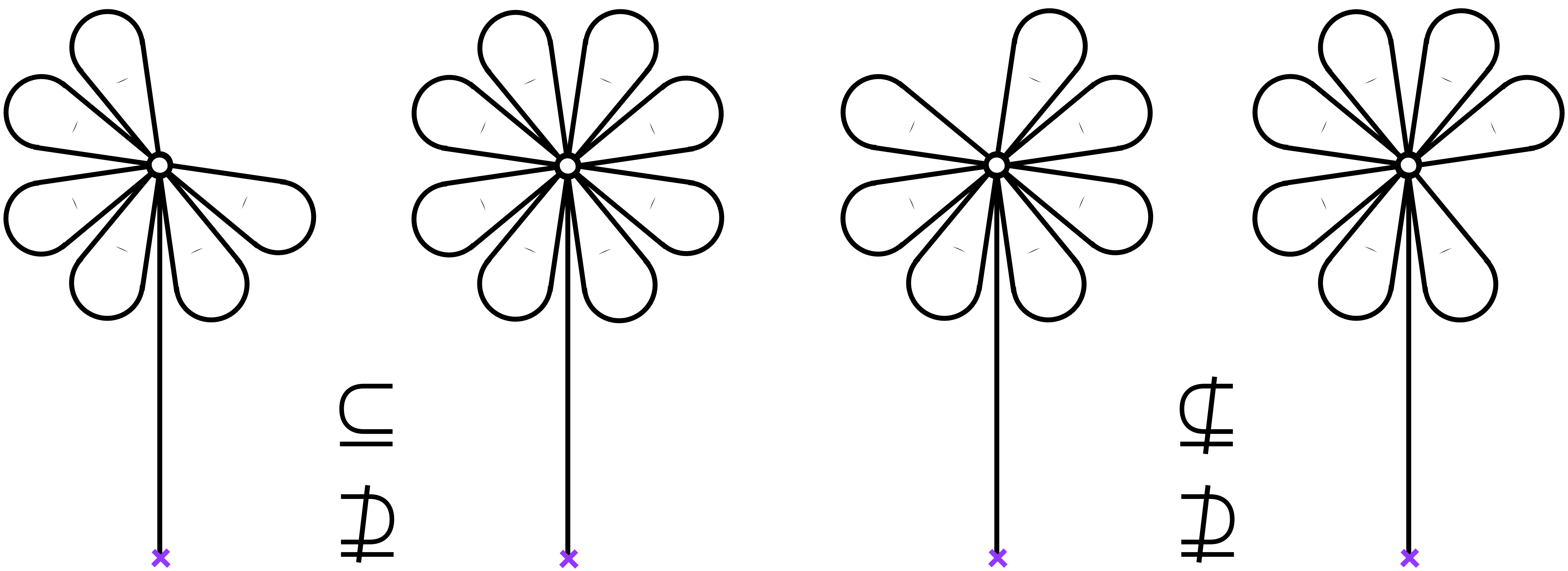
An example II of object: a flower



Logical operations on objects

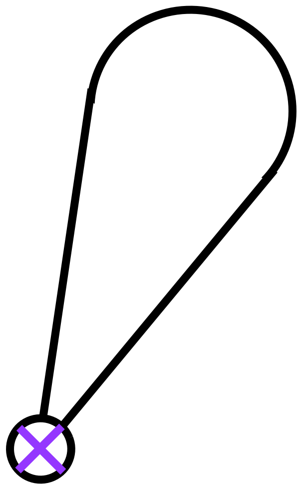
- Inclusion $\subseteq$

- Examples:



Constant objects

- A petal is an example of constant object

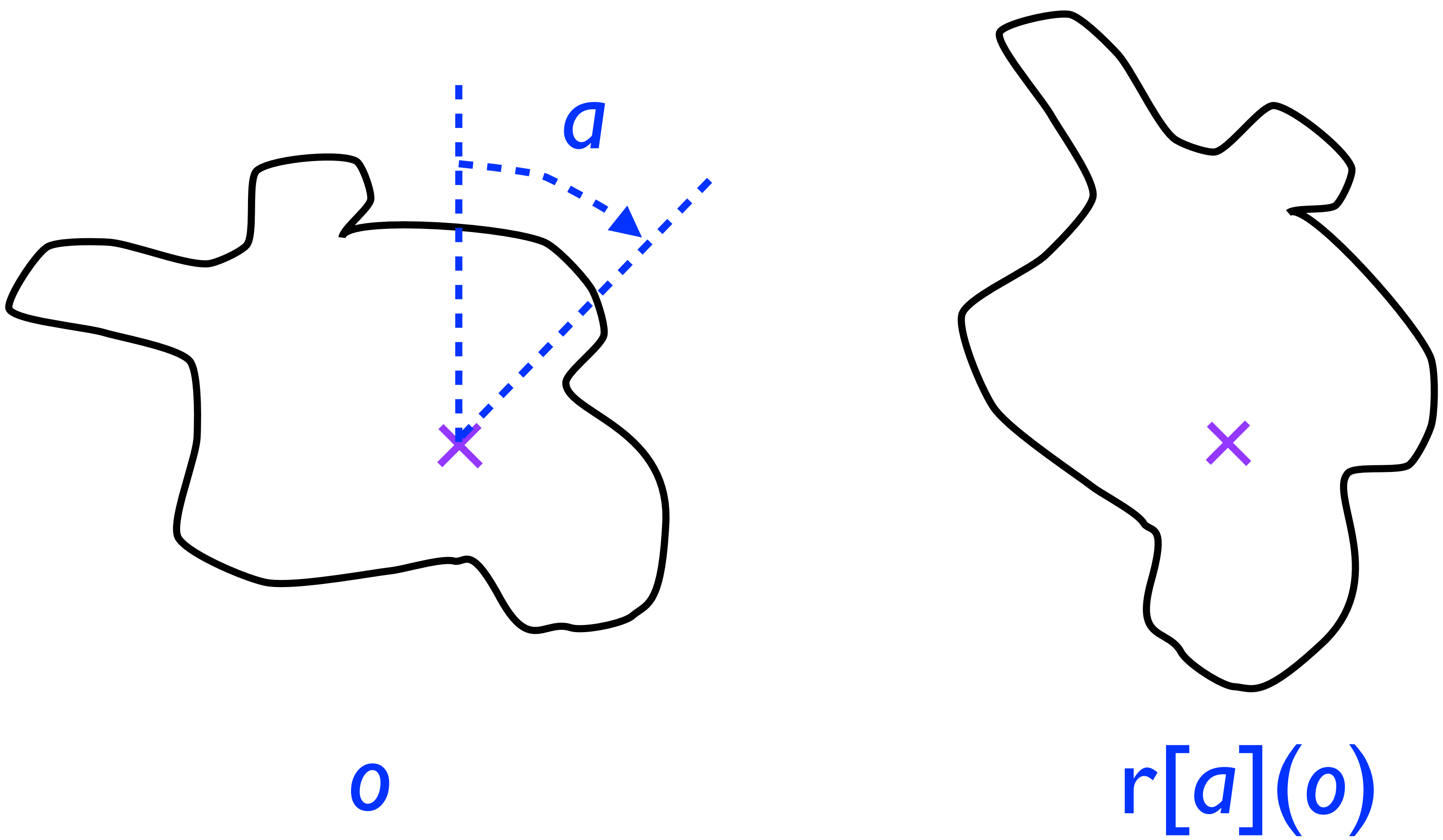
petal $=$ 

Operation on objects: rotation

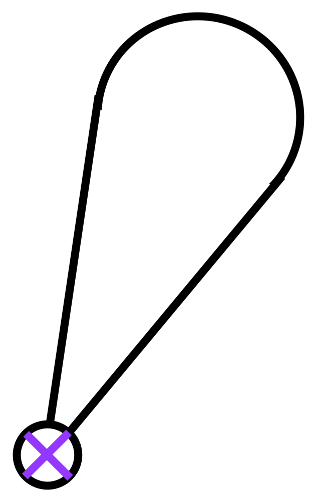
- rotation

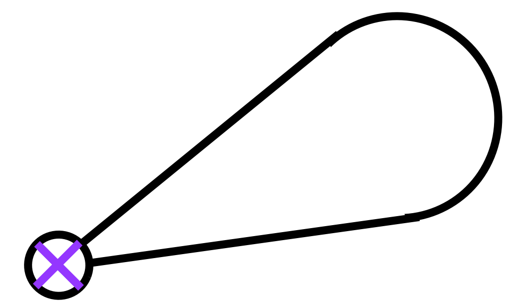
$$r[a](o)$$

rotates the object o clockwise by angle a degrees around its origin

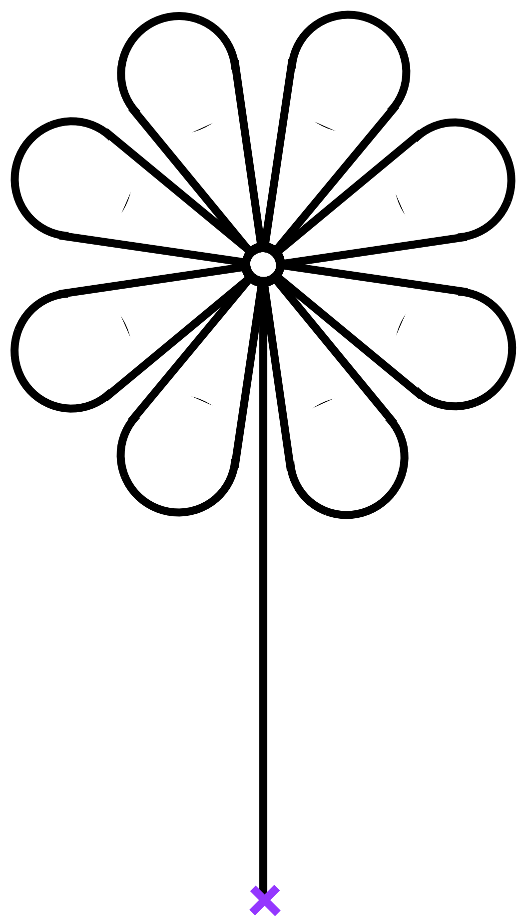


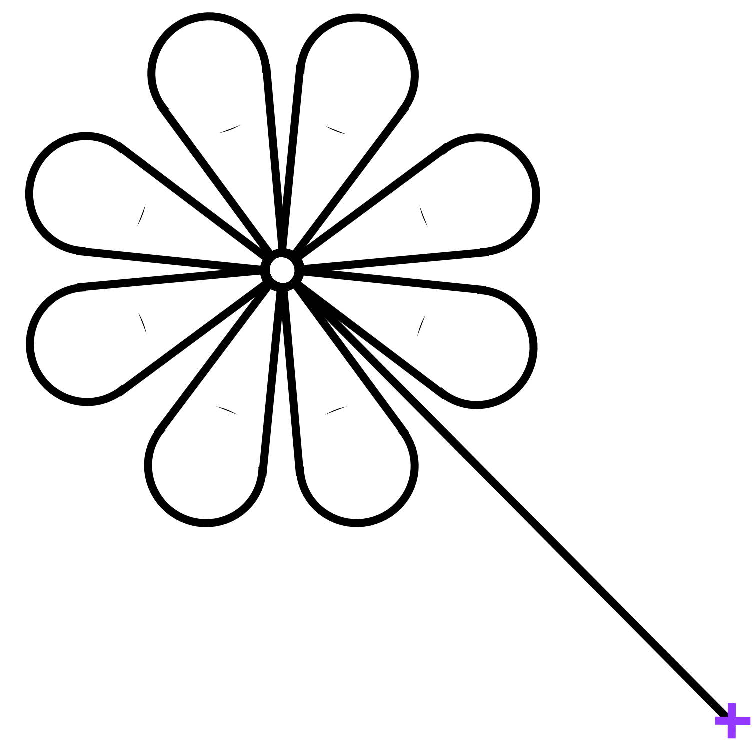
Example 1 of rotation

petal = 

r[45](petal) = 

Example II of rotation

flower = 

r[-45](flower) = 

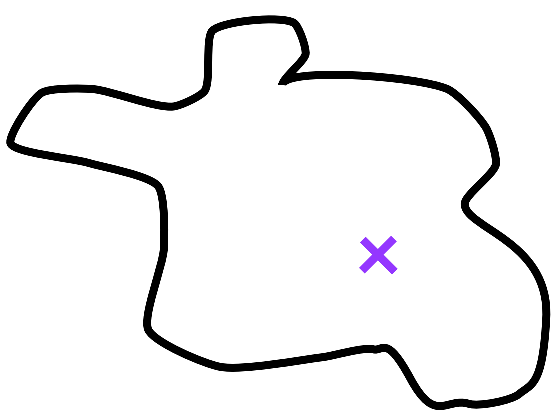
Operation on objects: add a stem

- Add a stem

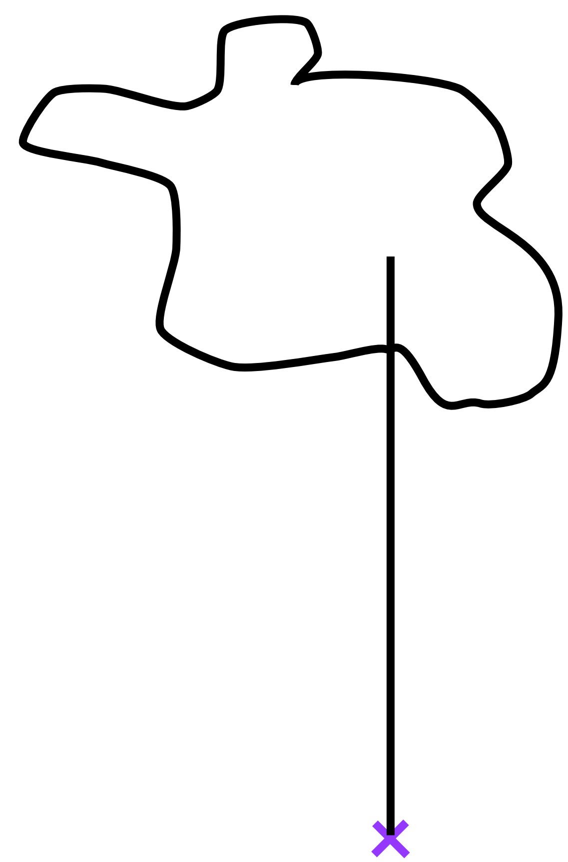
  stem(o)

  adds a stem to object o (up to the origin of object o, with new origin at the root of the stem)

o =



stem(o) =

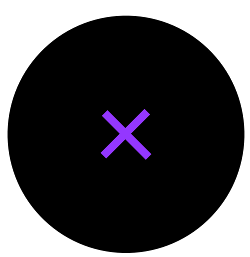


Operation on objects: union

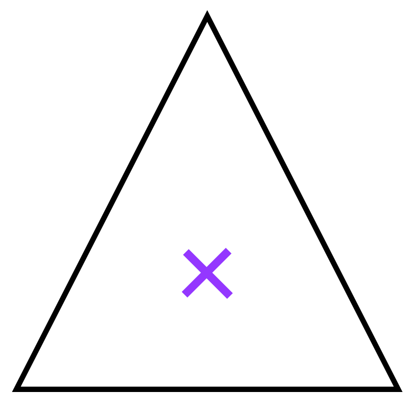
- The union $o_1 \cup o_2$ of objects o_1 and o_2 is the superposition of the pixels of o_1 and o_2 at their origins

- Example:

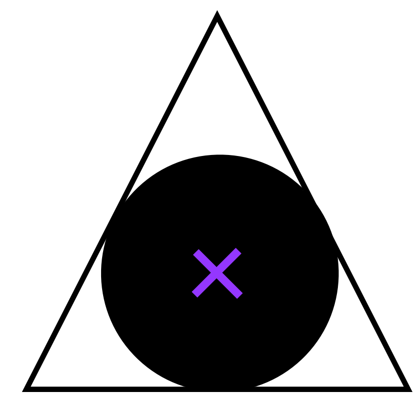
$o_1 =$



$o_2 =$

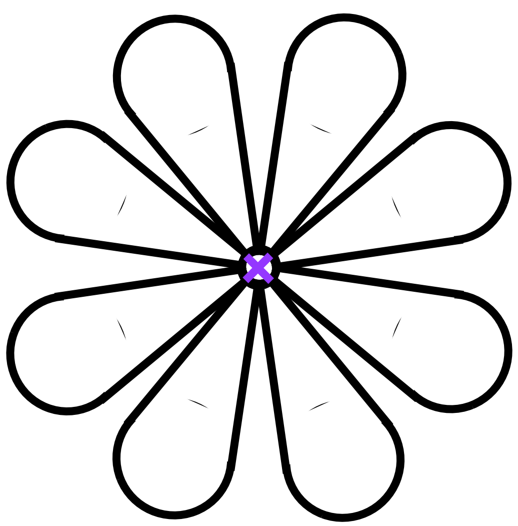


$o_1 \cup o_2 =$

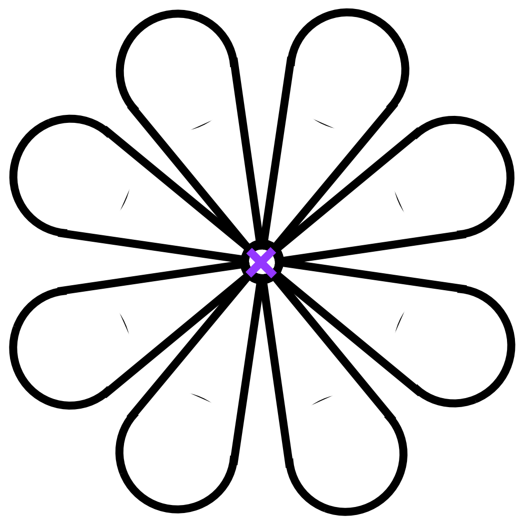


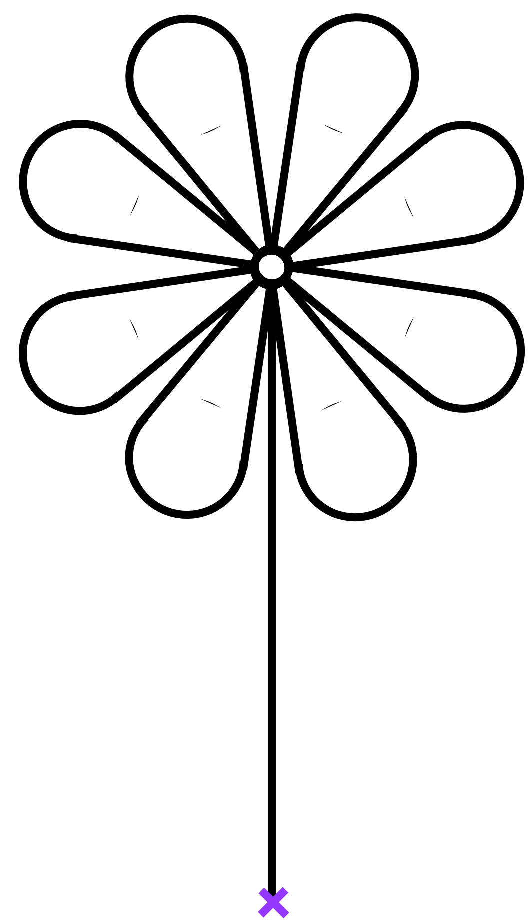
Example: corolla

- corolla = petal U r[45](petal) U r[90](petal) U r[135](petal) U r[180](petal) U r[225](petal) U r[270](petal) U r[315](petal)

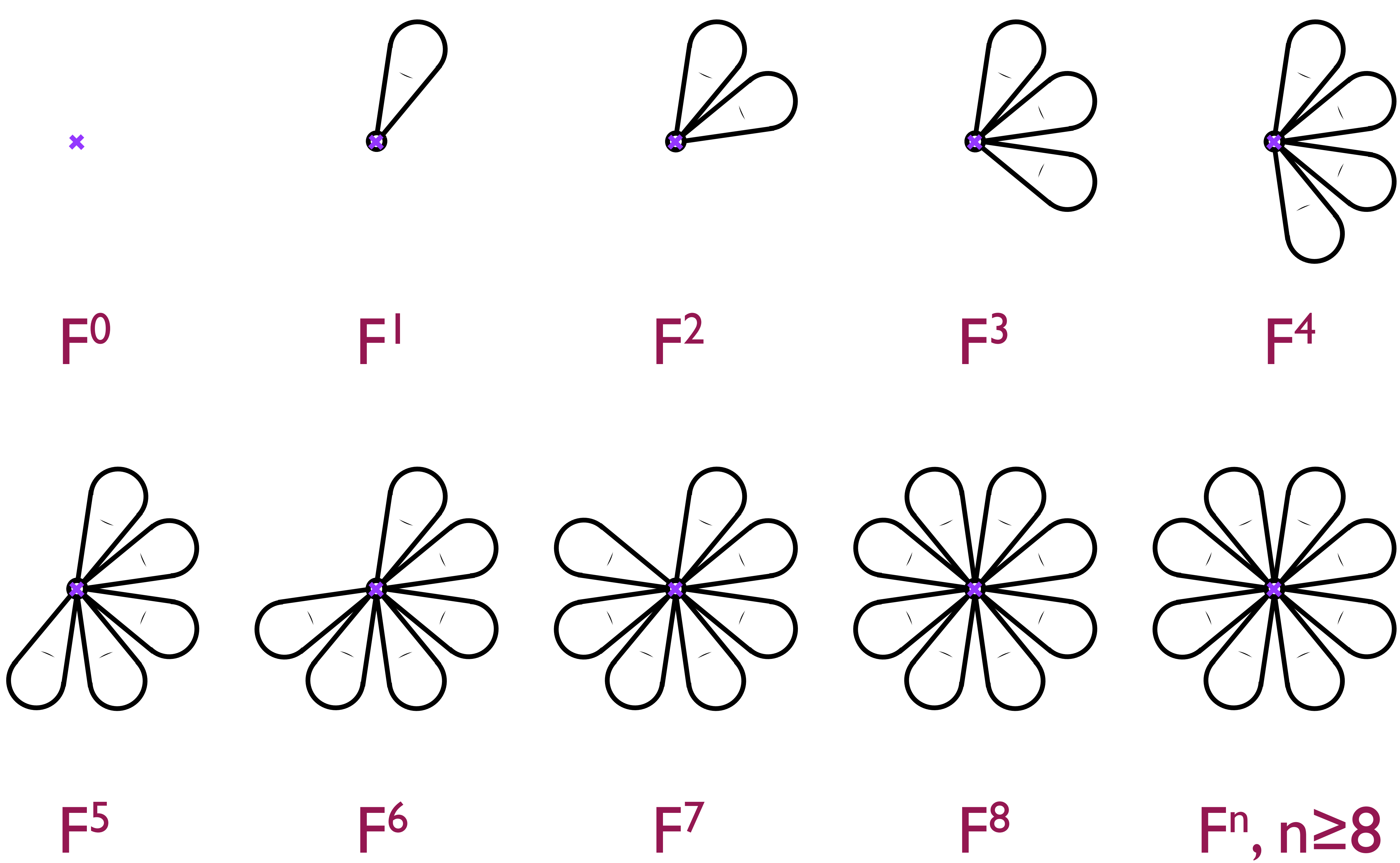


flower

corolla = 

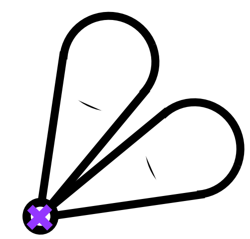
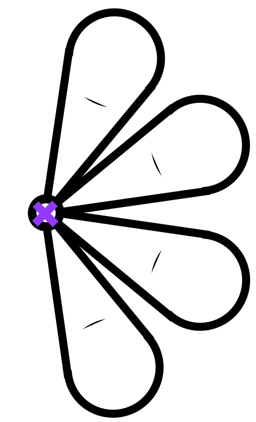
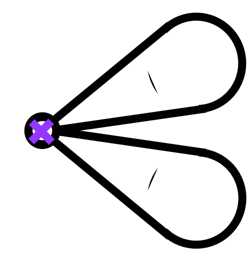
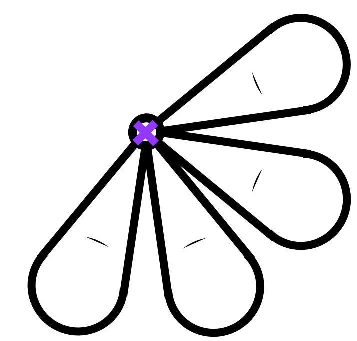
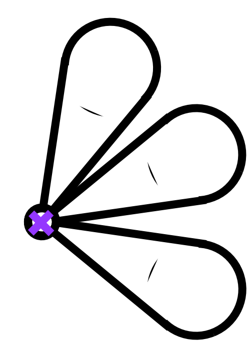
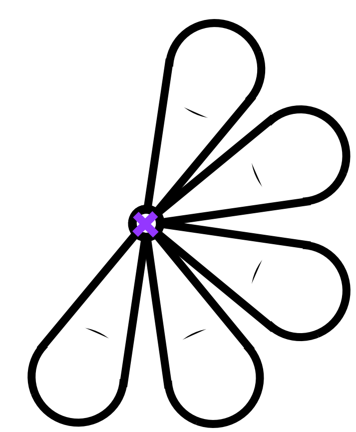
flower = stem(corolla) = 

Building a corolla iteratively



Corolla transformer

- F(X) = r[45]X U petal

- Example:
  - X =  
  - r[45]X =  
  - r[45]X U petal =  

Iterates of a transformer to a fixpoint

- The iterates F^n, $n \geq 0$, of F from the empty set $\emptyset$ are

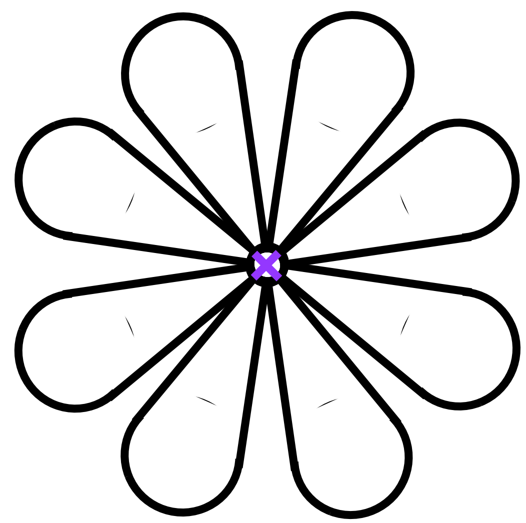
$$F^0 = \emptyset$$
$$F^1 = F(F^0)$$
$$F^2 = F(F^1)$$
$$\ldots$$
$$F^{n+1} = F(F^n)$$
$$\ldots$$
$$F^\omega = \bigcup_{n \geq 0} F^n = \mathrm{lfp}\ F$$

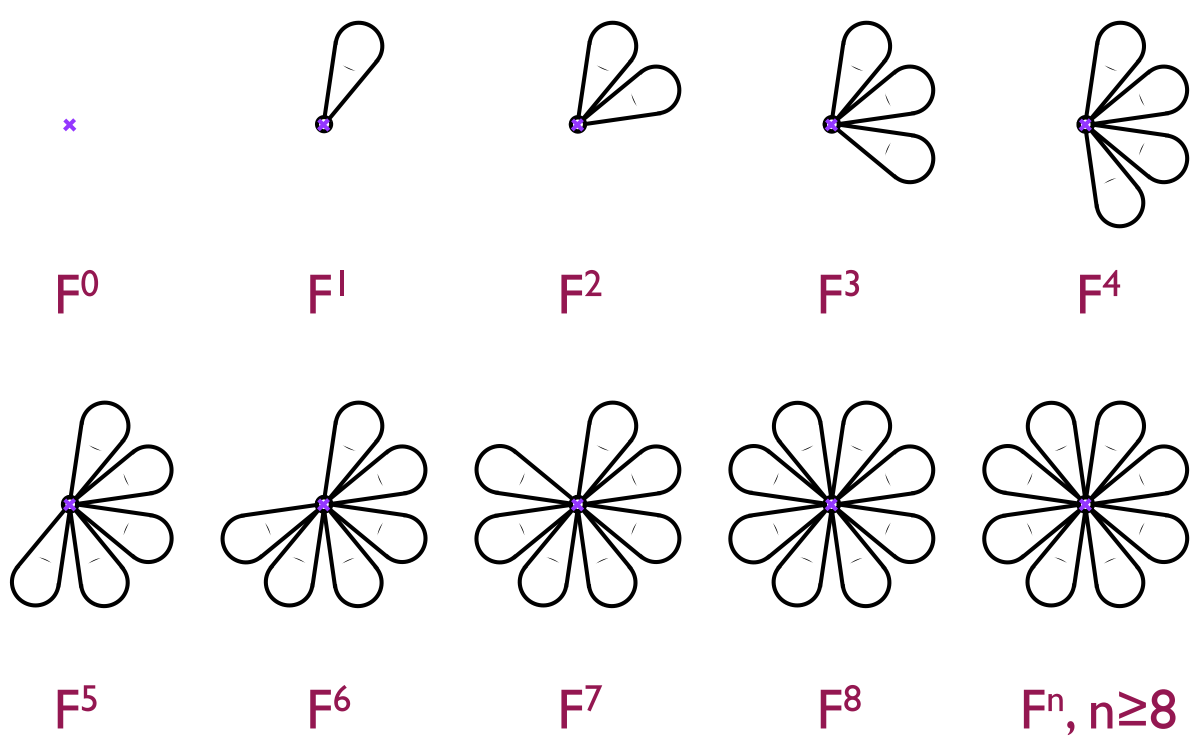
(assuming F continuous)

- Least fixpoint: $F(\mathrm{lfp}\ F) = \mathrm{lfp}\ F$, and $F(x)=x$ implies $\mathrm{lfp}\ F \subseteq x$

Patrick Cousot & Radhia Cousot. Constructive versions of Tarski's fixed point theorems. In *Pacific Journal of Mathematics*, Vol. 82, No. 1, 1979, pp. 43—57.

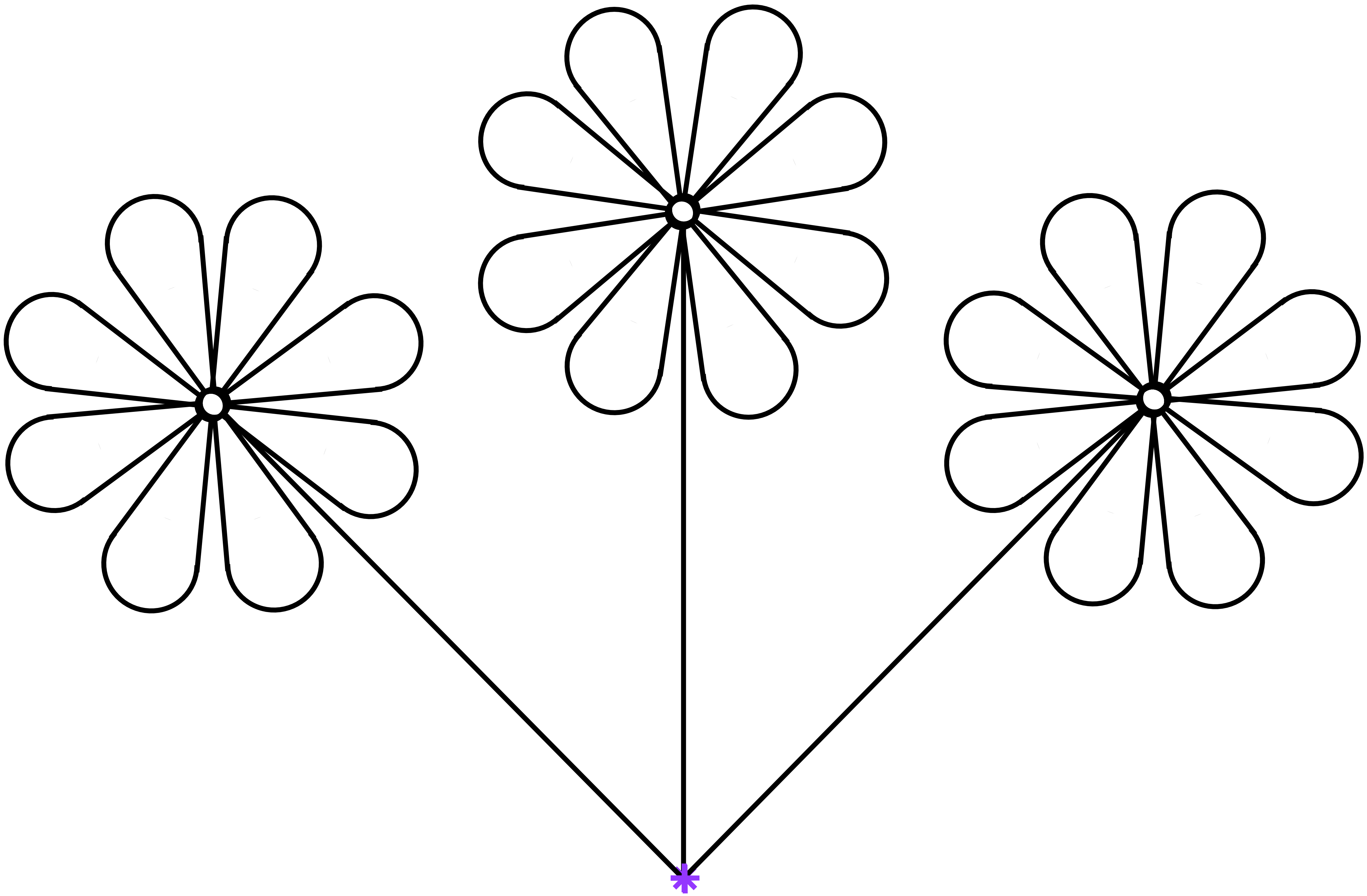
Fixpoint corolla

- F(X) = r[45]X ∪ petal
- corolla = lfp F = 
- Proof: the *iterates* are



Concrete bouquet

- bouquet = r[-45](flower) U flower U r[45](flower)



The abstract world

Over-approximation

- An over-approximation of an object o is an object $\overline{o}$ with
  - same origin
  - <u>more</u> pixels

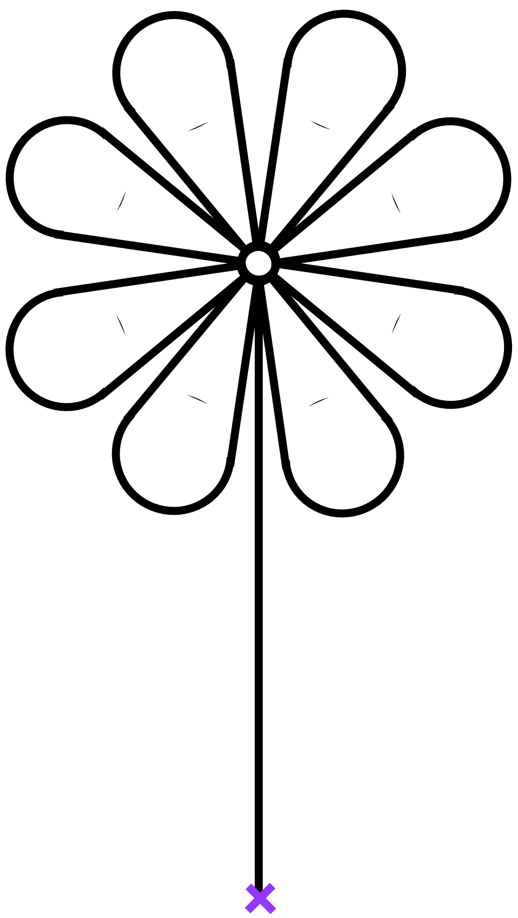
- The dual[(I)] is an under-approximation, with <u>less</u> pixels

---

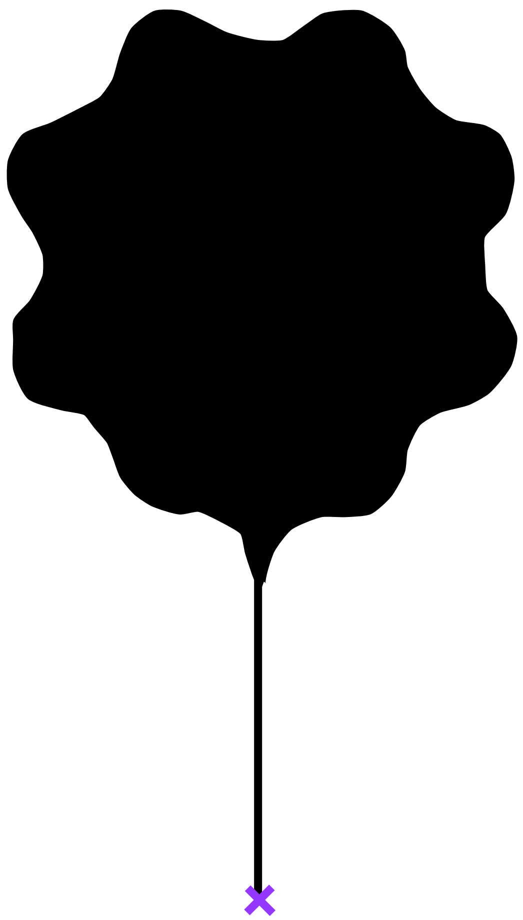
(I) Patrick Cousot. Méthodes itératives de construction et d'approximation de points fixes d'opérateurs monotones sur un treillis, analyse sémantique des programmes. *Thèse És Sciences Mathématiques*, Université Joseph Fourier, Grenoble, France, 21 March 1978

Patrick Cousot. Semantic foundations of program analysis. In S.S. Muchnick & N.D. Jones, editors, *Program Flow Analysis: Theory and Applications*, Ch. 10, pages 303—342, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, U.S.A., 1981.

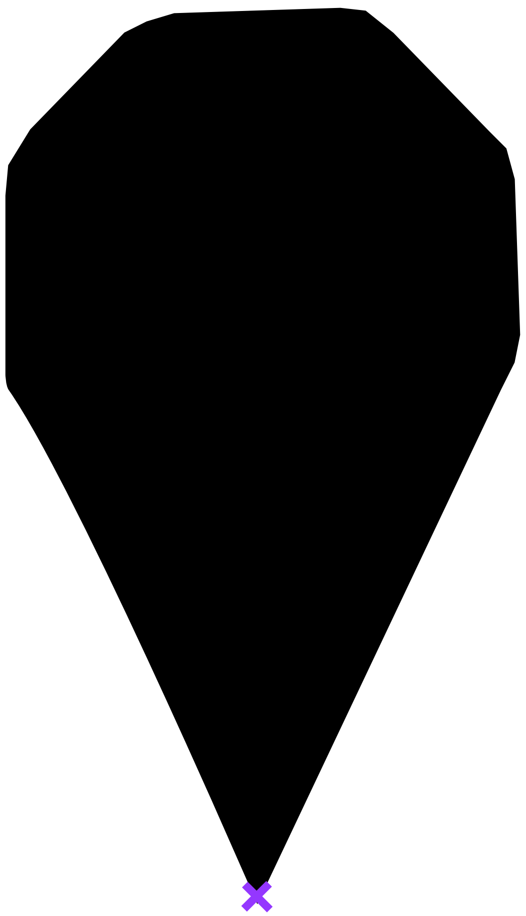
Examples of over-approximations of flowers



$\subseteq$

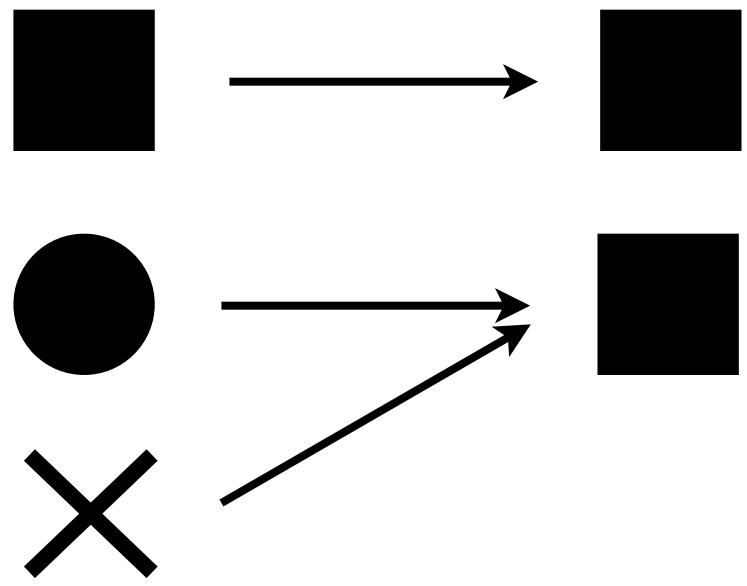


$\subseteq$



Abstraction

- An abstraction of an object o is a mathematical/computer representation of an over-approximation of this object o
- The abstraction is sometimes exact else is a strict over-approximation
  - Examples abstraction by plain squares

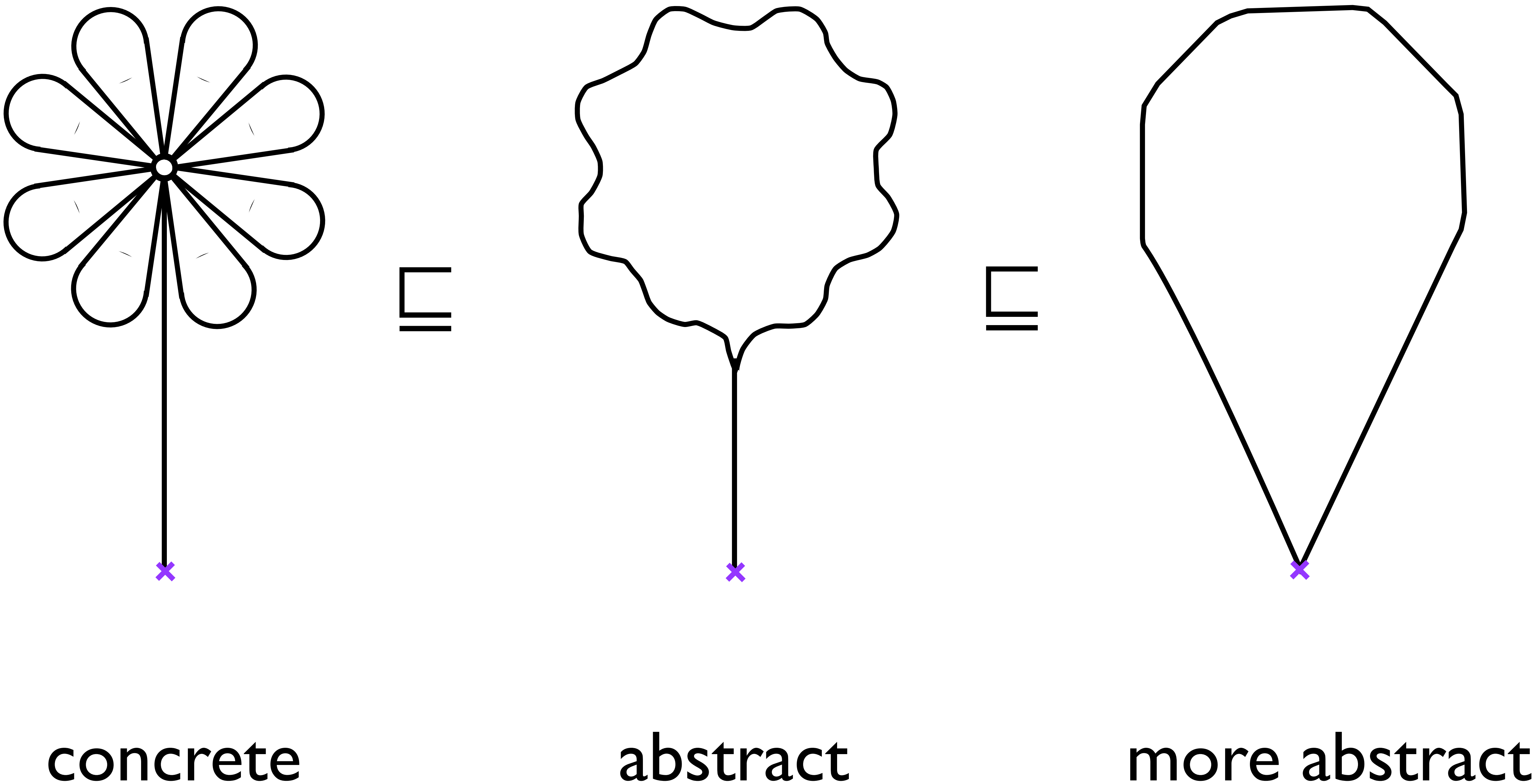


exact

strict over-approximation

Examples of abstractions of flowers

- Encode a concrete over-approximation by its outline



A Touch of Abstract Interpretation Theory

Patrick Cousot, Radhia Cousot: Abstract Interpretation: A Unified Lattice Model for Static Analysis of Programs by Construction or Approximation of Fixpoints. POPL 1977: 238-252
Patrick Cousot, Radhia Cousot: Systematic Design of Program Analysis Frameworks. POPL 1979: 269-282

Abstract domain

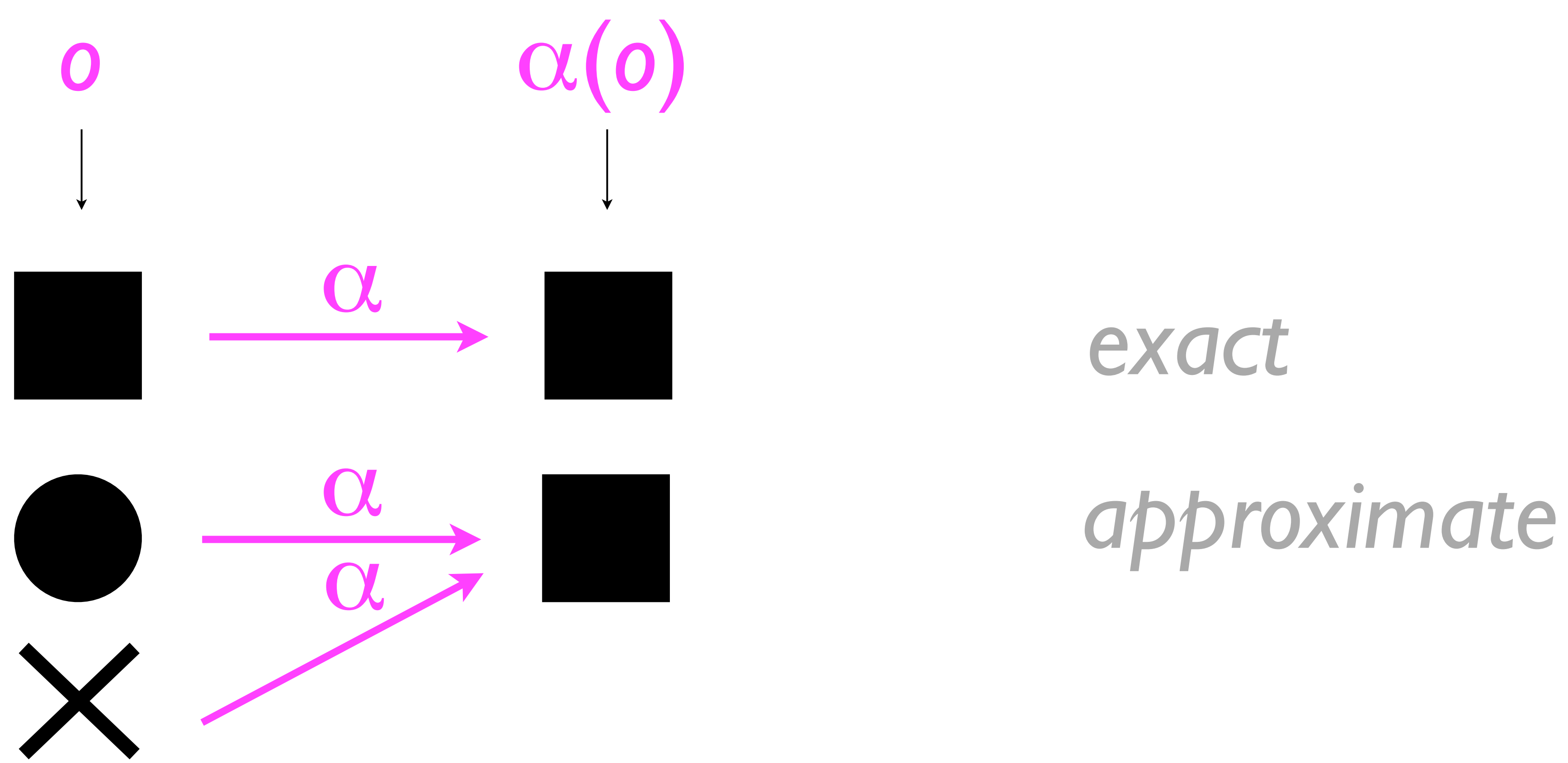
- An abstract domain is
  - a set of abstract objects $\overline{O}$ (abstracting concrete objects)
  - a set of abstract operations (abstracting the concrete operations) $\overline{O}^n \longrightarrow \overline{O}$, $n \geq 0$
  - a set of logical abstract operations

    $\overline{O}^n \longrightarrow \mathit{Booleans}$, $n \geq 0$

Abstraction function

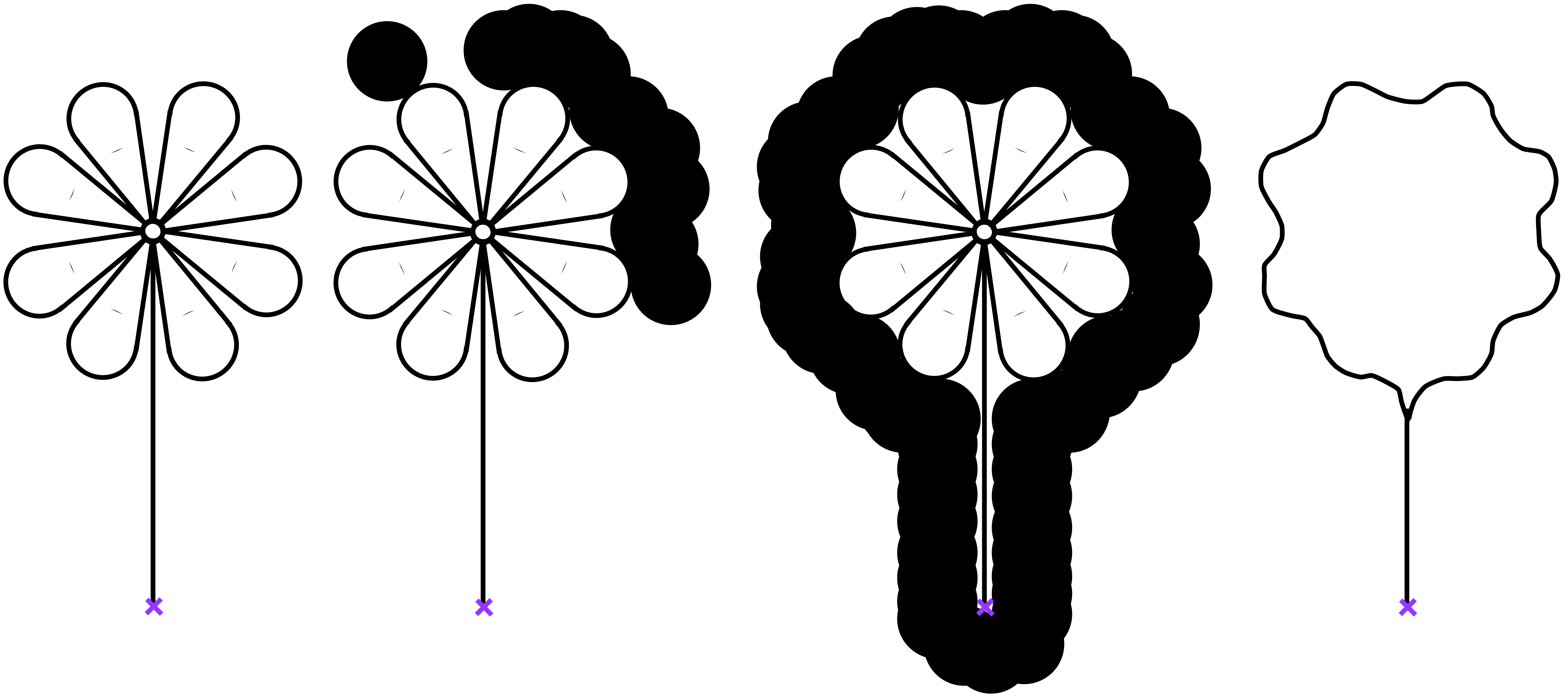
- The abstraction function $\alpha \in \mathcal{O} \longrightarrow \overline{\mathcal{O}}$ maps concrete objects $o \in \mathcal{O}$ to their approximation by an abstract object $\alpha(o) \in \overline{\mathcal{O}}$

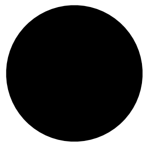
- Example 1 of abstraction function by plain squares:



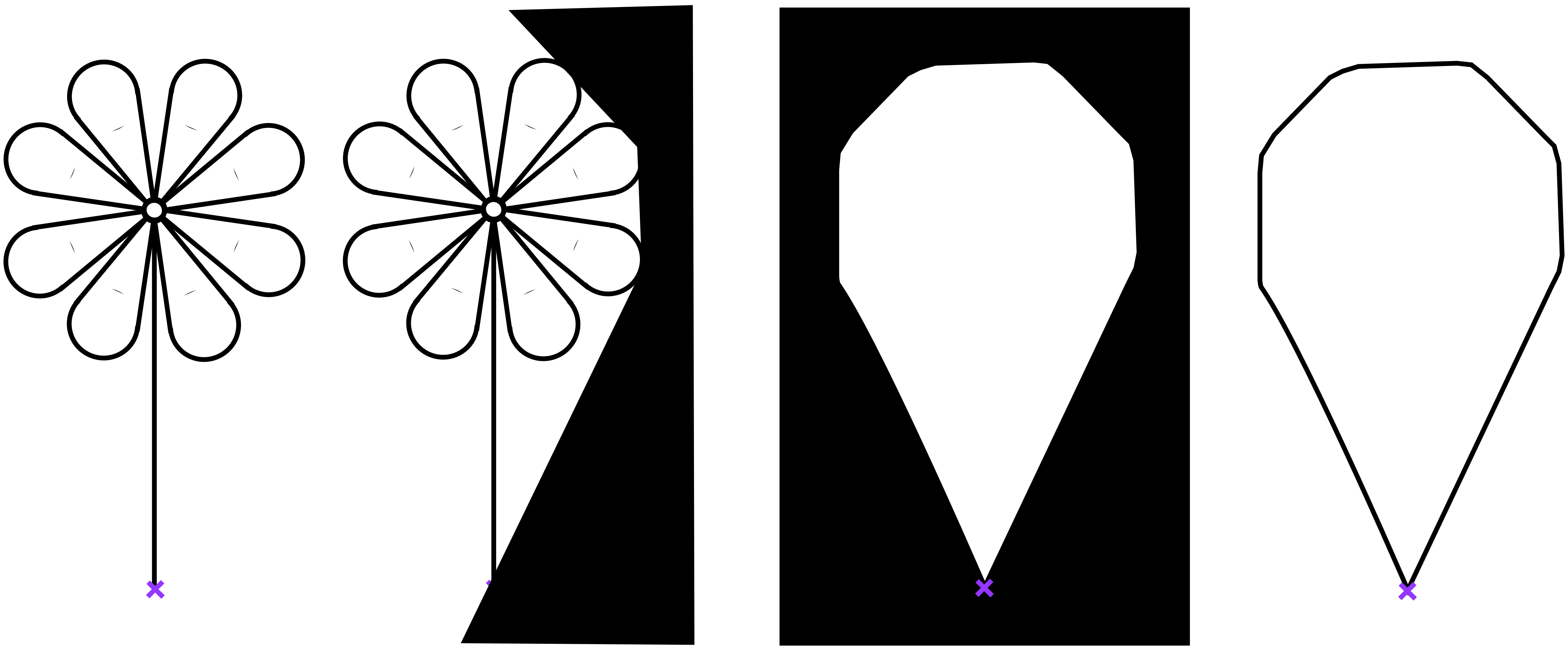
Patrick Cousot, Radhia Cousot: Systematic Design of Program Analysis Frameworks. POPL 1979: 269-282
Patrick Cousot, Radhia Cousot, Laurent Mauborgne: The Reduced Product of Abstract Domains and the Combination of Decision Procedures. FOSSACS 2011: 456-472

Example II of abstraction function



Outlining brush: 

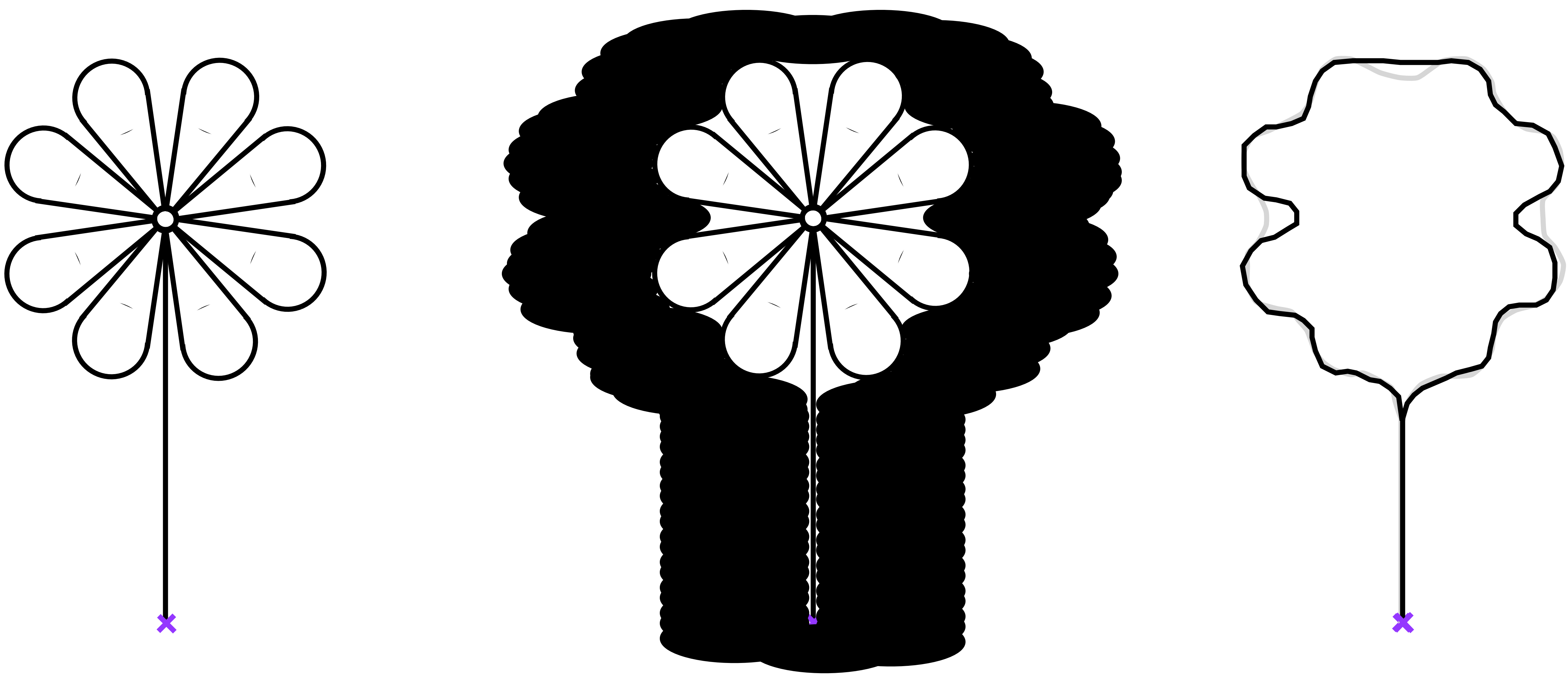
Example III of abstraction function



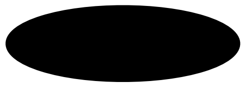
Outlining brush of infinite diameter

The hierarchy of abstractions

- Larger brush diameter: more abstract
- Different brush shapes: may be non-comparable abstractions



Oval outlining brush:

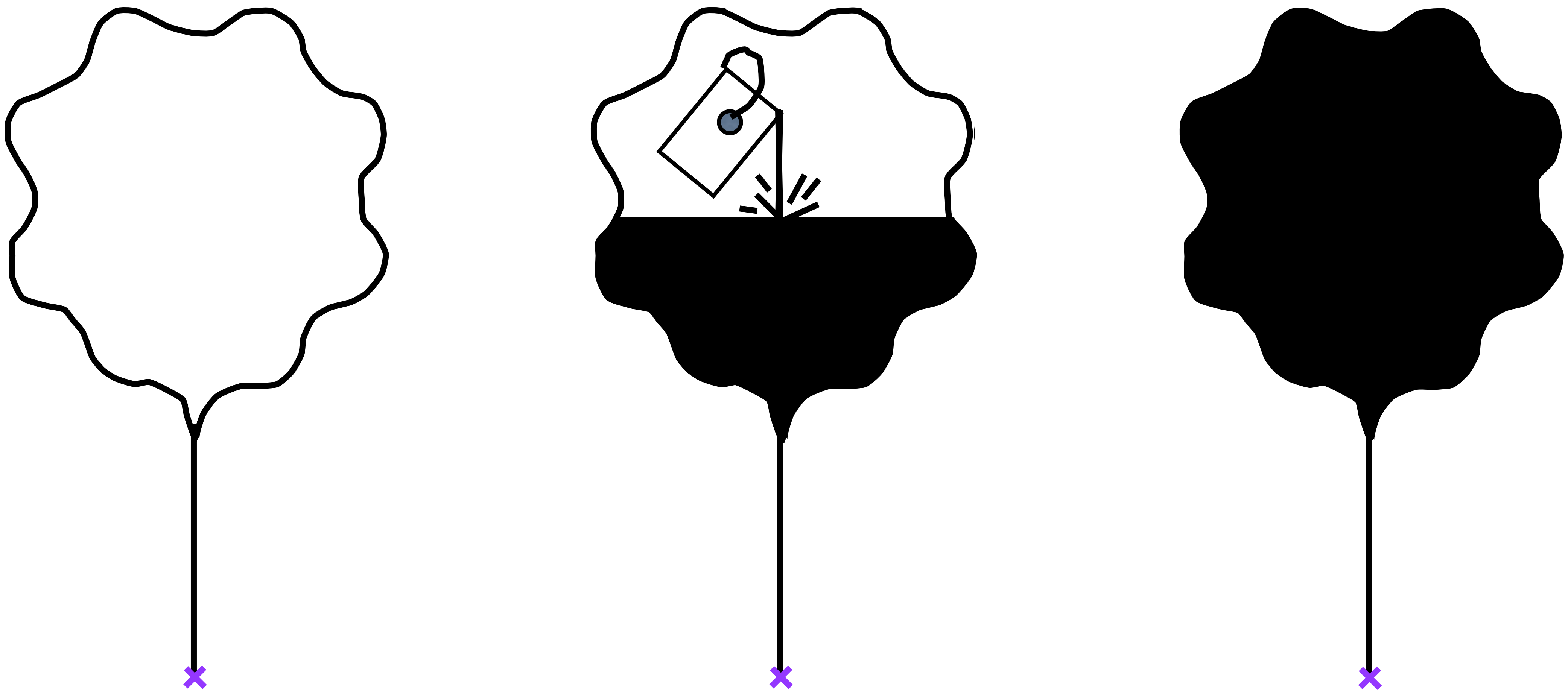


Concretization function

- A concretization function $\gamma \in \overline{O} \longrightarrow O$ maps an abstract object $\bar{o} \in \overline{O}$ to the concrete objects $\gamma(\bar{o}) \in O$ that is represents/approximates
- $\gamma(\bar{o})$ is the concrete meaning/semantics of $\bar{o}$

Patrick Cousot, Radhia Cousot: Systematic Design of Program Analysis Frameworks. POPL 1979: 269-282
Patrick Cousot, Radhia Cousot, Laurent Mauborgne: The Reduced Product of Abstract Domains and the Combination of Decision Procedures. FOSSACS 2011: 456-472

Example of concretization



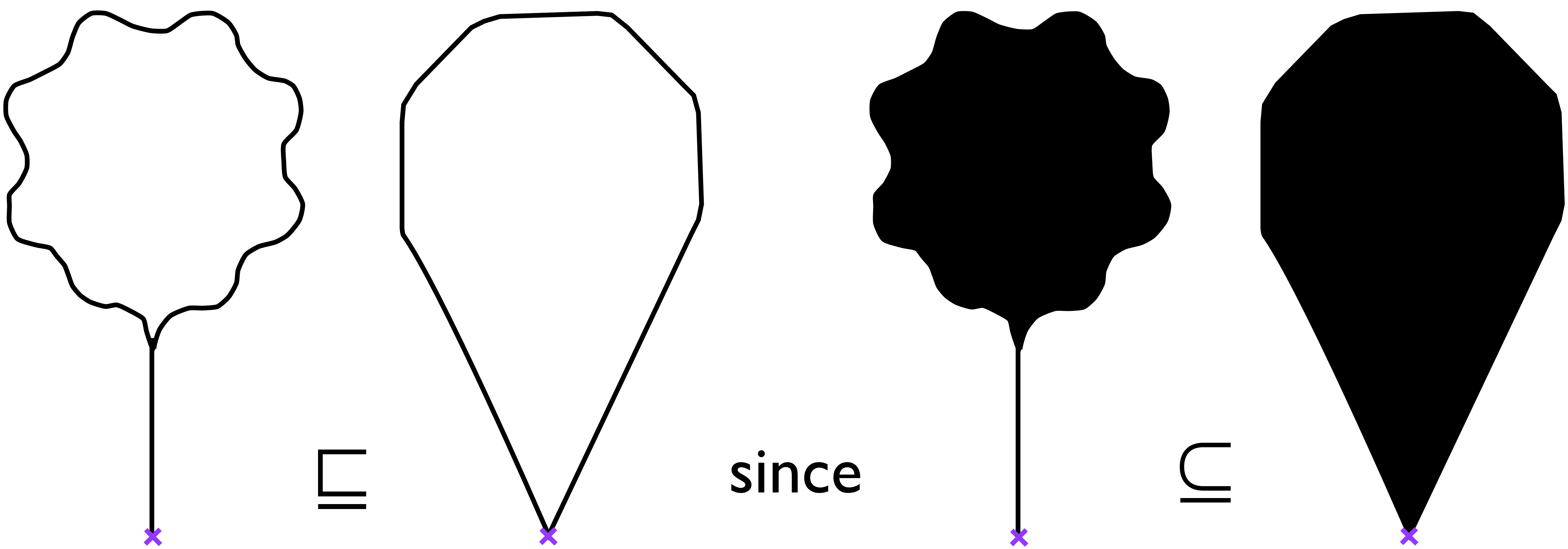
$\bar{o}$ $\xrightarrow{\gamma}$ $\gamma(\bar{o})$

Abstract logical operation: abstract inclusion

- The abstract flower inclusion is defined as

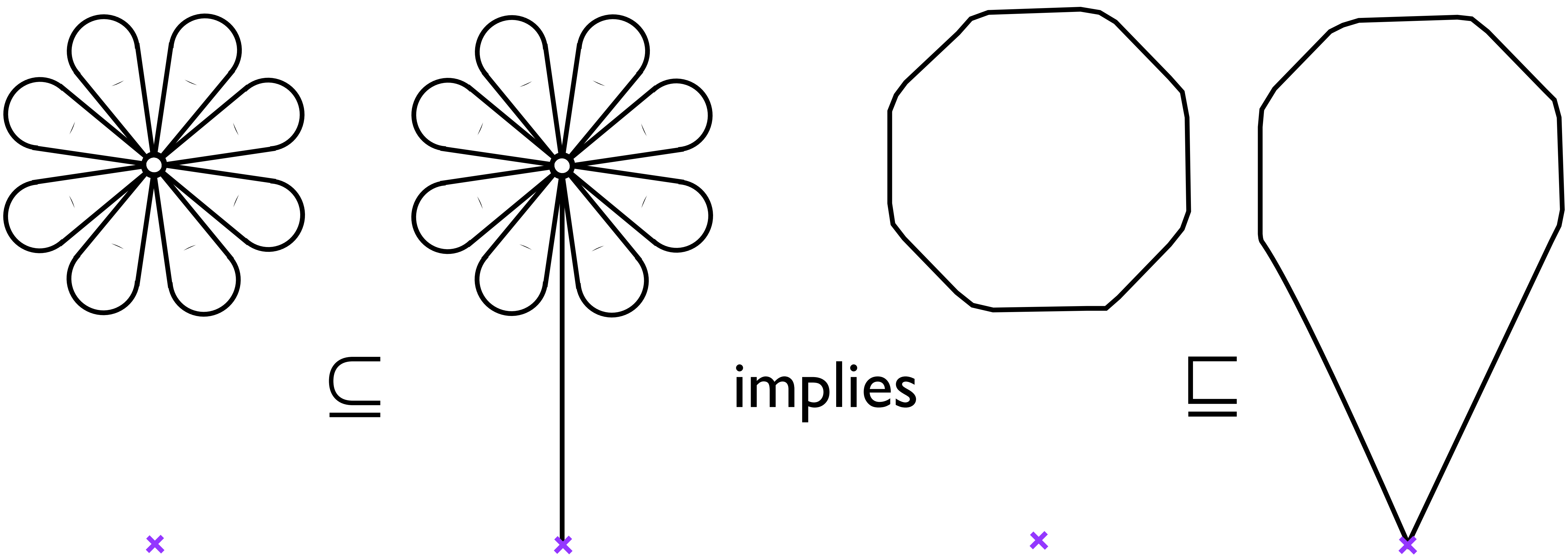
$$\bar{o}_1 \sqsubseteq \bar{o}_2 \quad \text{if and only if} \quad \gamma(\bar{o}_1) \subseteq \gamma(\bar{o}_2)$$

- Example:



Galois connection 1/4

- α is increasing

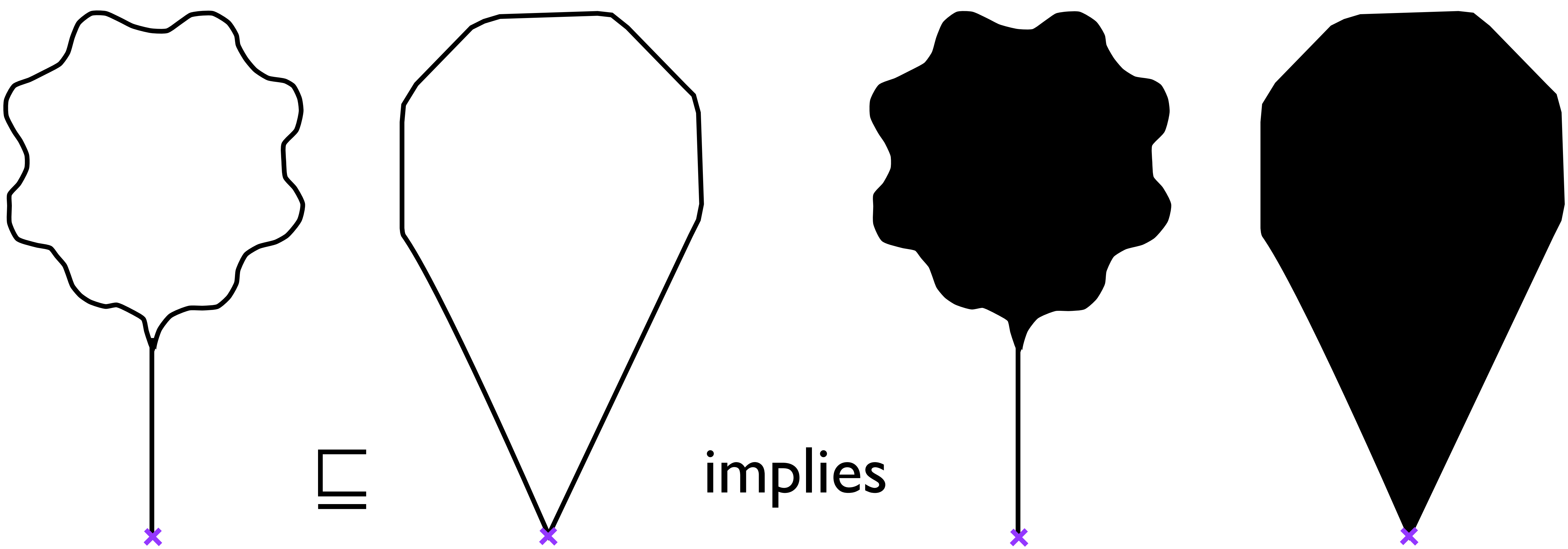


The larger the concrete, the larger the abstract

Patrick Cousot, Radhia Cousot: Systematic Design of Program Analysis Frameworks. POPL 1979: 269-282
Patrick Cousot, Radhia Cousot, Laurent Mauborgne: The Reduced Product of Abstract Domains and the Combination of Decision Procedures. FOSSACS 2011: 456-472

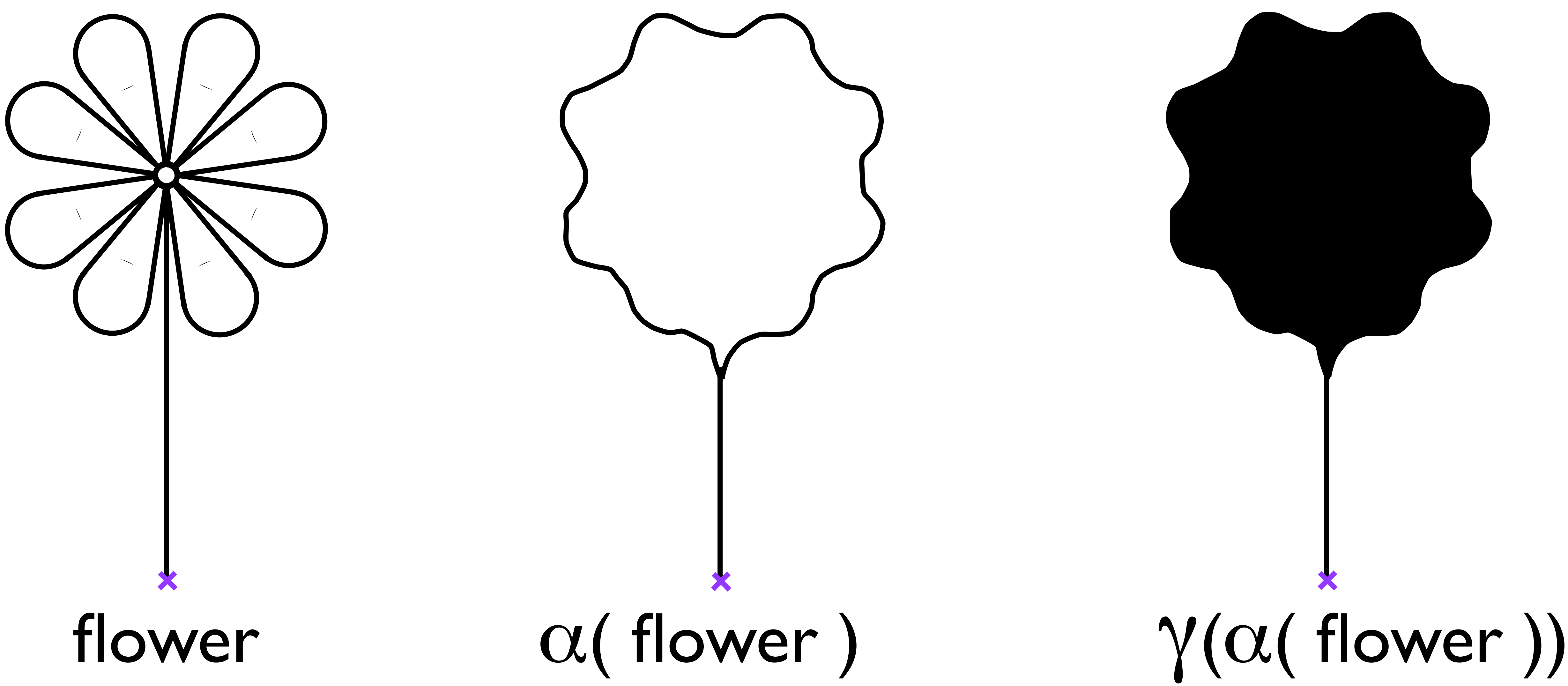
Galois connection 2/4

- γ is increasing
- Proof: by definition of $\sqsubseteq$, $\overline{o}_1 \sqsubseteq \overline{o}_2$ implies $\gamma(\overline{o}_1) \subseteq \gamma(\overline{o}_2)$



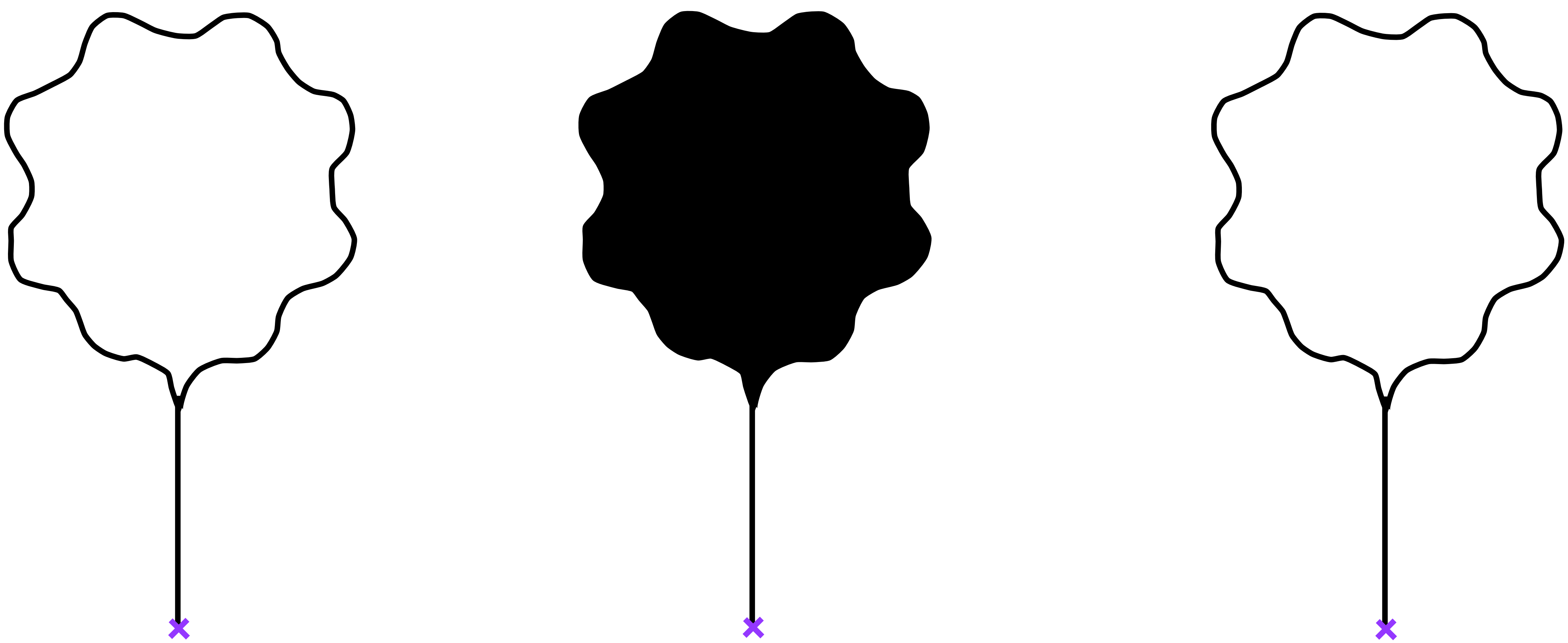
Galois connection 3/4

- For all concrete objects $x \in \mathcal{O}$, $x \subseteq \gamma \circ \alpha(x)$
- Intuition: soudness (over-approximation)



Galois connection 4/4

- For all abstract objects $y \in \overline{O}$, $\alpha \circ \gamma(y) = y$
- Intuition: α returns the most precise abstraction



$\text{fl}\overline{\text{ower}}$ $\qquad$ $\gamma(\text{fl}\overline{\text{ower}})$ $\qquad$ $\alpha(\gamma\big(\text{fl}\overline{\text{ower}}))$

Galois connection: all in one

- Notation:

$$\langle O, \subseteq \rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle \bar{O}, \sqsubseteq \rangle$$

- Equivalent definition

$$\forall o \in O, \bar{o} \in \bar{O}: \quad \alpha(o) \sqsubseteq \bar{o} \quad \textit{iff} \quad o \subseteq \gamma(\bar{o})$$

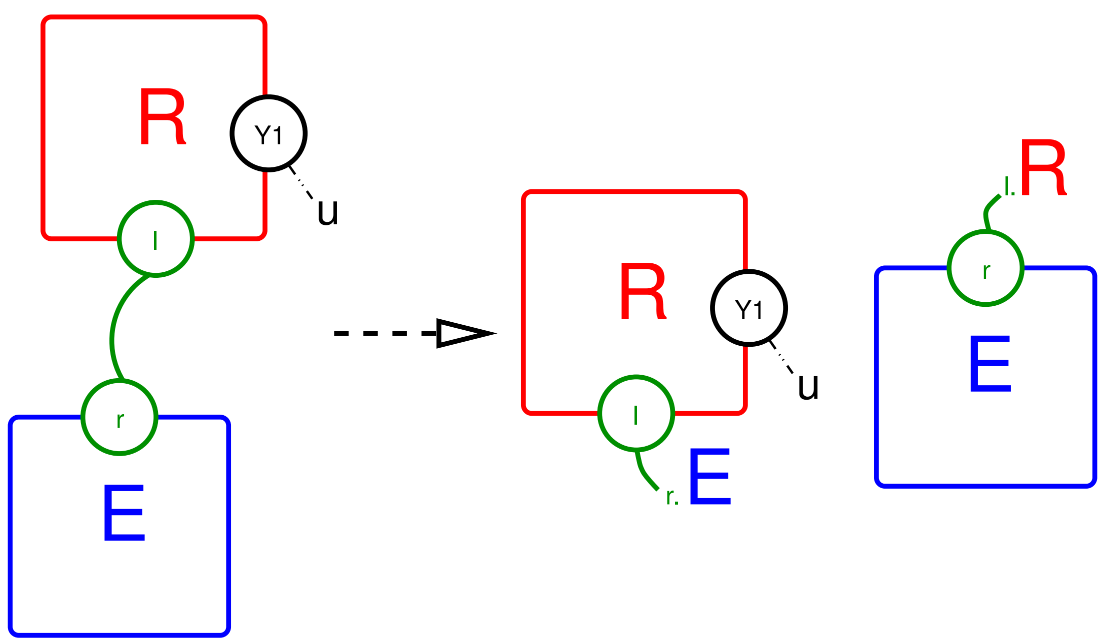
and

$\Longrightarrow$ soundness
$\Longleftarrow$ best abstraction

α surjective (otherwise $\alpha \circ \gamma(y) \sqsubseteq y$)

Patrick Cousot, Radhia Cousot: Systematic Design of Program Analysis Frameworks. POPL 1979: 269-282
Patrick Cousot, Radhia Cousot, Laurent Mauborgne: The Reduced Product of Abstract Domains and the Combination of Decision Procedures. FOSSACS 2011: 456-472

Example of biological abstraction

- Let *Species* be the set of all chemical species ($\mathrm{C}, \mathrm{c}_1, \mathrm{c}'_1, \ldots, \mathrm{c}_k, \mathrm{c}'_k, \ldots \in$ *Species*).
- Let *Local_view* be the set of all local views
- Let $\alpha \in \wp(\mathit{Species}) \rightarrow \wp(\mathit{Local\_view})$ be the function that maps any set of complexes into the set of their local views.
- 

$\alpha(\{\mathrm{R(Y1{\sim}u,l!1), E(r!1)}\})$

$= \{\mathrm{R(Y1{\sim}u,l!r.E); E(r!l.R)}\}$

- The function α defines a Galois connexion: $\wp(\mathit{Species}) \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \wp(\mathit{Local\_view})$
- (The function γ maps a set of local views into the set of complexes that can be built with these local views).

Jérôme Feret. Reachability Analysis of Biological Signalling Pathways by Abstract Interpretation. In Proceedings of the International Conference of Computational Methods in Sciences and Engineering (ICCMSE'2007), Corfu, Greece, 25--30 september, T.E. Simos(Ed.), 2007, American Institute of Physics conference proceedings 963.(2), pp 619--622.

Vincent Danos, Jérôme Feret, Walter Fontana, Jean Krivine: Abstract Interpretation of Cellular Signalling Networks. VMCAI 2008: 83-97

Specification of abstract operations

- $\text{cte} \triangleq \alpha(\text{cte})$ constant

  $\text{op}_1(x) \triangleq \alpha(\text{op}_1(\gamma(x)))$ unary

  $\text{op}_2(x, y) \triangleq \alpha(\text{op}_2(\gamma(x), \gamma(y)))$ binary

  ....

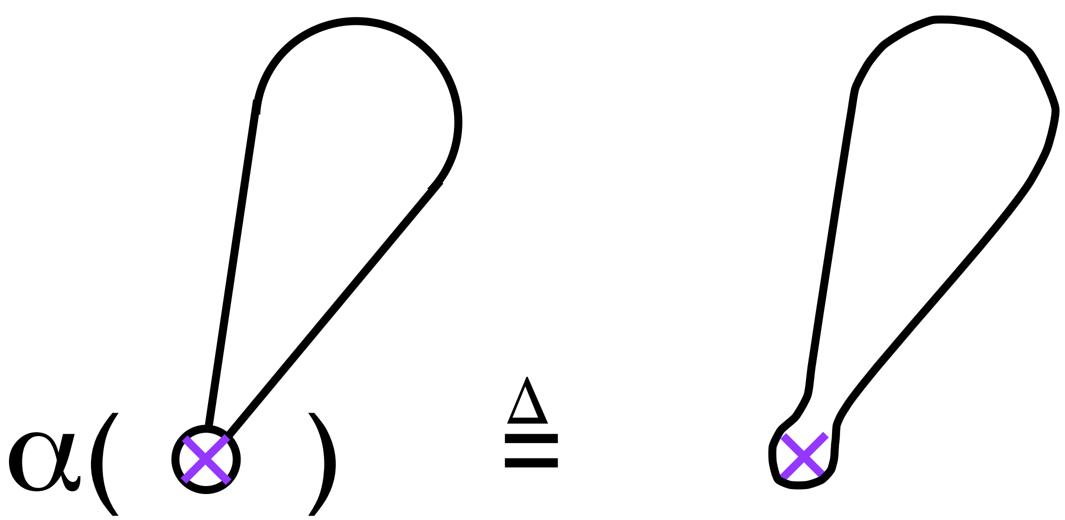
  $\text{op}_n(x_1,\ldots,x_n) \triangleq \alpha(\text{op}_n(\gamma(x_1), \ldots,\gamma(x_n)))$ *n*-ary

- Can be less precise

  $\alpha(\text{op}_n(\gamma(x_1), \ldots,\gamma(x_n))) \sqsubseteq \text{op}_n(x_1,\ldots,x_n)$

Abstract constants

- Abstract petal

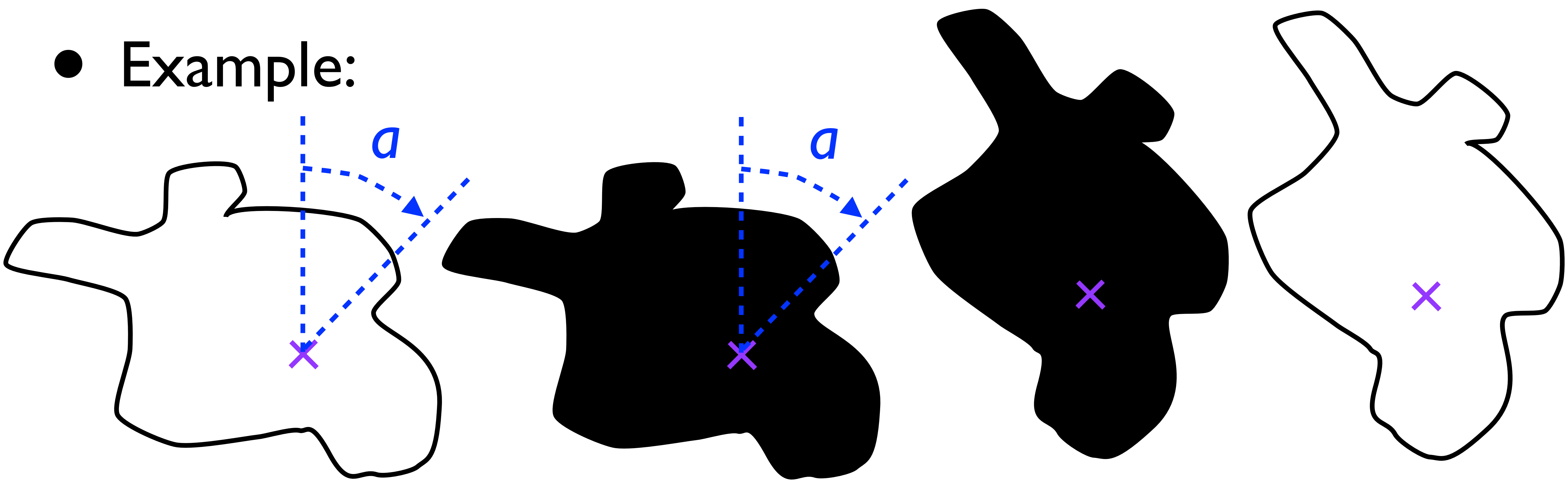


Abstract rotation

- Abstract rotation

$$\begin{array}{rcll} \bar{r}[a](\bar{o}) & \stackrel{\Delta}{=} & \alpha(r[a](\gamma(\bar{o}))) & \text{definition} \\ & = & \alpha(\gamma(r[a](\bar{o}))) & \text{rotation preserves shape} \\ & = & r[a](\bar{o}) & \text{identity} \end{array}$$

- Example:



$\bar{o}$ $\quad\gamma(\bar{o})\quad$ $r[a](\gamma(\bar{o}))$ $\quad\alpha(r[a](\gamma(\bar{o})))$

A commutation theorem on rotation

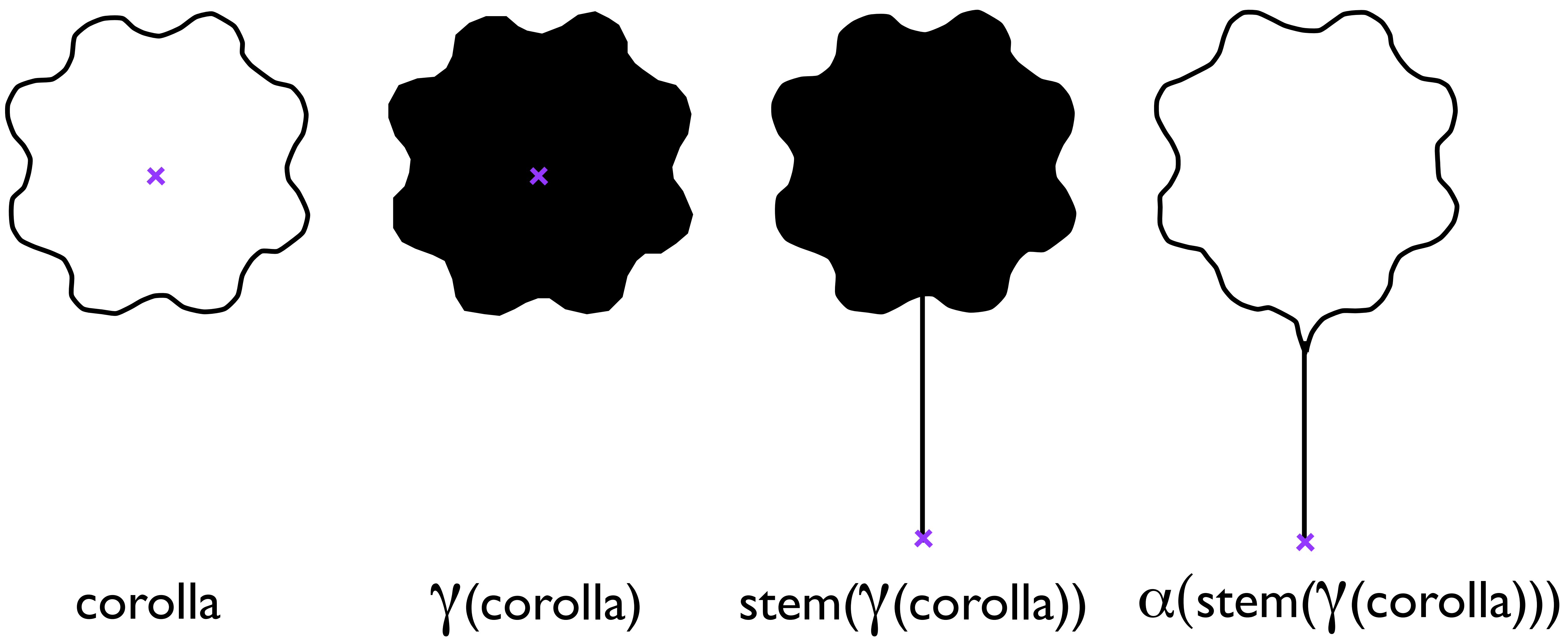
- $\alpha(r[a](y)) = \bar{r}[a](\alpha(y))$ $\quad\forall y \in \bar{O}$
- Proof:

$$
\begin{aligned}
& \alpha(r[a](y)) && \alpha \circ \gamma \text{ is the identity} \\
=\ & \alpha(\gamma(\alpha(r[a](y)))) && \alpha \text{ preserves rotation} \\
=\ & \alpha(\gamma(r[a](\alpha(y)))) && \gamma \text{ preserves rotation} \\
=\ & \alpha(r[a](\gamma(\alpha(y)))) && \\
\stackrel{\Delta}{=}\ & \bar{r}[a](\alpha(y)) && \text{definition abstract rotation}
\end{aligned}
$$

Abstract stems

- $\overline{\text{stem}}(y) \triangleq \alpha(\text{stem}(\gamma(y)))$

- Example:



Abstract union

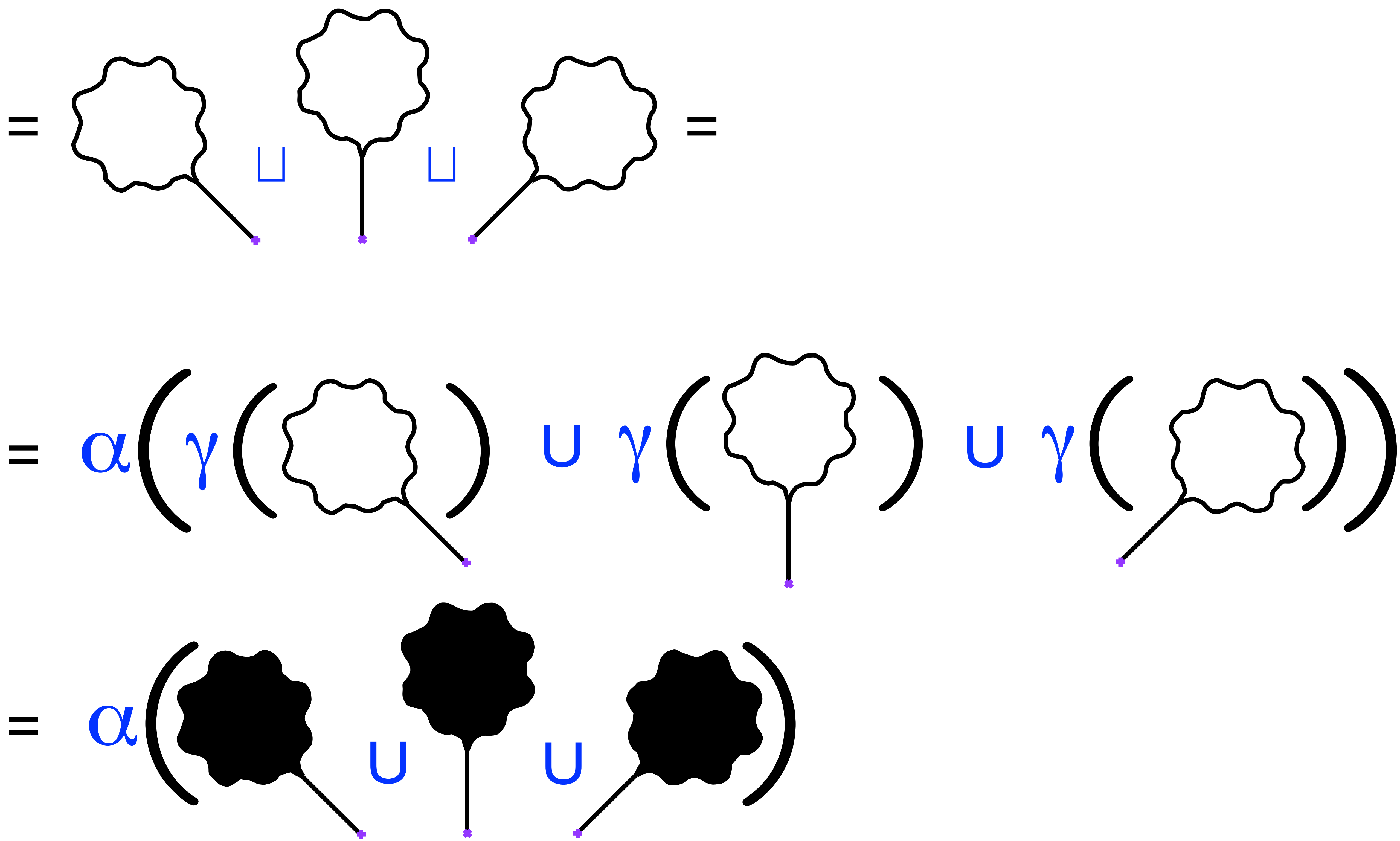
- $x \sqcup y \triangleq \alpha(\gamma(x) \cup \gamma(y))$

- Join abstraction theorem:

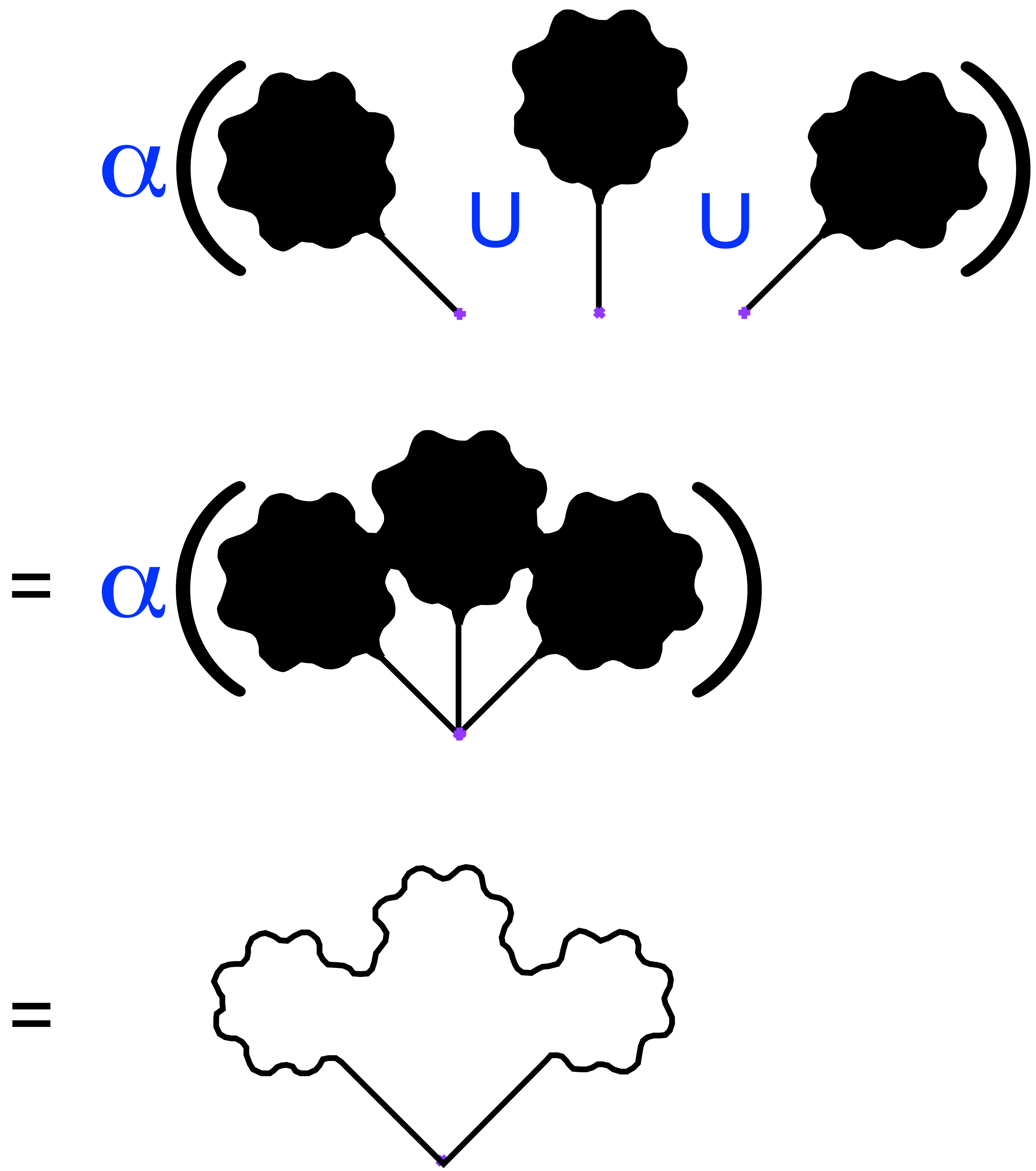
$\alpha(x) \sqcup \alpha(y) = \alpha(x \cup y)$ Galois connection

Abstract bouquet (cont'd)

- $\overline{\text{bouquet}} = \bar{r}[-45](\overline{\text{flower}}) \sqcup \overline{\text{flower}} \sqcup \bar{r}[45](\overline{\text{flower}})$



Abstract bouquet (cont'd)



A theorem on abstract bouquets

- $\overline{\text{bouquet}}$

$= \bar{r}[-45](\overline{\text{flower}}) \sqcup \overline{\text{flower}} \sqcup \bar{r}[45](\overline{\text{flower}})$

flower = α(flower)

$= \bar{r}[-45](\alpha(\text{flower})) \sqcup \alpha(\text{flower}) \sqcup \bar{r}[45](\alpha(\text{flower}))$

rotation commutation theorem

$= \alpha(r[-45](\text{flower})) \sqcup \alpha(\text{flower}) \sqcup \alpha(r[45](\text{flower}))$

join abstraction theorem

$= \alpha(r[-45](\text{flower}) \cup \text{flower} \cup r[45](\text{flower}))$

definition concrete bouquet

$= \alpha(\text{bouquet})$

Abstract corolla transformer

- Corolla transformer commutation theorem:
- $\alpha(F(x))$

$= \alpha(\text{petal} \cup r[45](x))$ definition F

$= \alpha(\text{petal}) \sqcup \alpha(r[45](x))$ join abstraction theorem

$= \overline{\text{petal}} \sqcup \alpha(r[45](x))$ definition abstract petal

$= \overline{\text{petal}} \sqcup \bar{r}[45](\alpha(x))$ rotation commut. theorem

$= \bar{F}(\alpha(x))$

by defining $\bar{F}(y) = \overline{\text{petal}} \sqcup \bar{r}[45](y)$

Abstract transformer

- An abstract transformer $\overline{F}$ is
  - *Sound* iff

$$\forall P \in \mathcal{P} : \alpha \circ F(P) \sqsubseteq \overline{F} \circ \alpha(P)$$

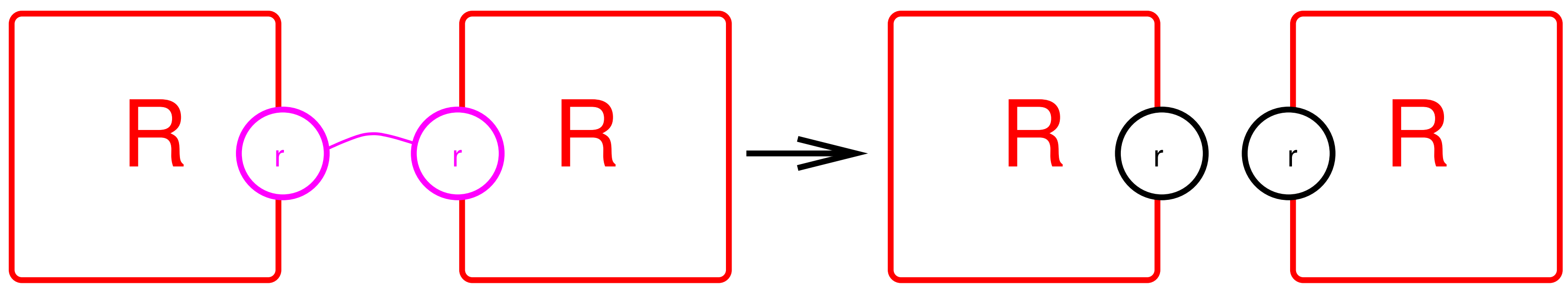
  - *Complete* iff

$$\forall P \in \mathcal{P} : \alpha \circ F(P) = \overline{F} \circ \alpha(P)$$

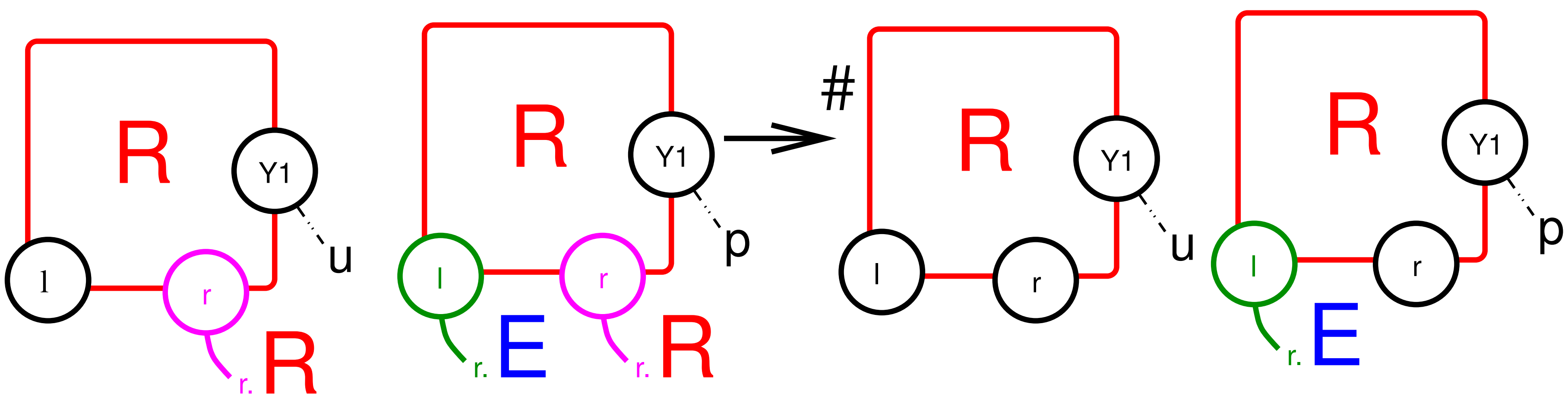
Patrick Cousot, Radhia Cousot: Abstract Interpretation: A Unified Lattice Model for Static Analysis of Programs by Construction or Approximation of Fixpoints. POPL 1977: 238-252
Patrick Cousot, Radhia Cousot: Systematic Design of Program Analysis Frameworks. POPL 1979: 269-282

Example of biological transformer

- Concrete rule:



- Abstract rule:



Jérôme Feret. Reachability Analysis of Biological Signalling Pathways by Abstract Interpretation. In Proceedings of the International Conference of Computational Methods in Sciences and Engineering (ICCMSE'2007), Corfu, Greece, 25--30 september, T.E. Simos(Ed.), 2007, American Institute of Physics conference proceedings 963.(2), pp 619--622.

Vincent Danos, Jérôme Feret, Walter Fontana, Jean Krivine: Abstract Interpretation of Cellular Signalling Networks. VMCAI 2008: 83-97

Fixpoint abstraction

- For an increasing and sound abstract transformer, we have a *fixpoint approximation*

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

- For an increasing, sound, and complete abstract transformer, we have an *exact fixpoint abstraction*

$$\alpha(\mathsf{lfp}^{\leqslant} F) = \mathsf{lfp}^{\sqsubseteq} \overline{F}$$

Patrick Cousot, Radhia Cousot: Systematic Design of Program Analysis Frameworks. POPL 1979: 269-282

Abstract corolla

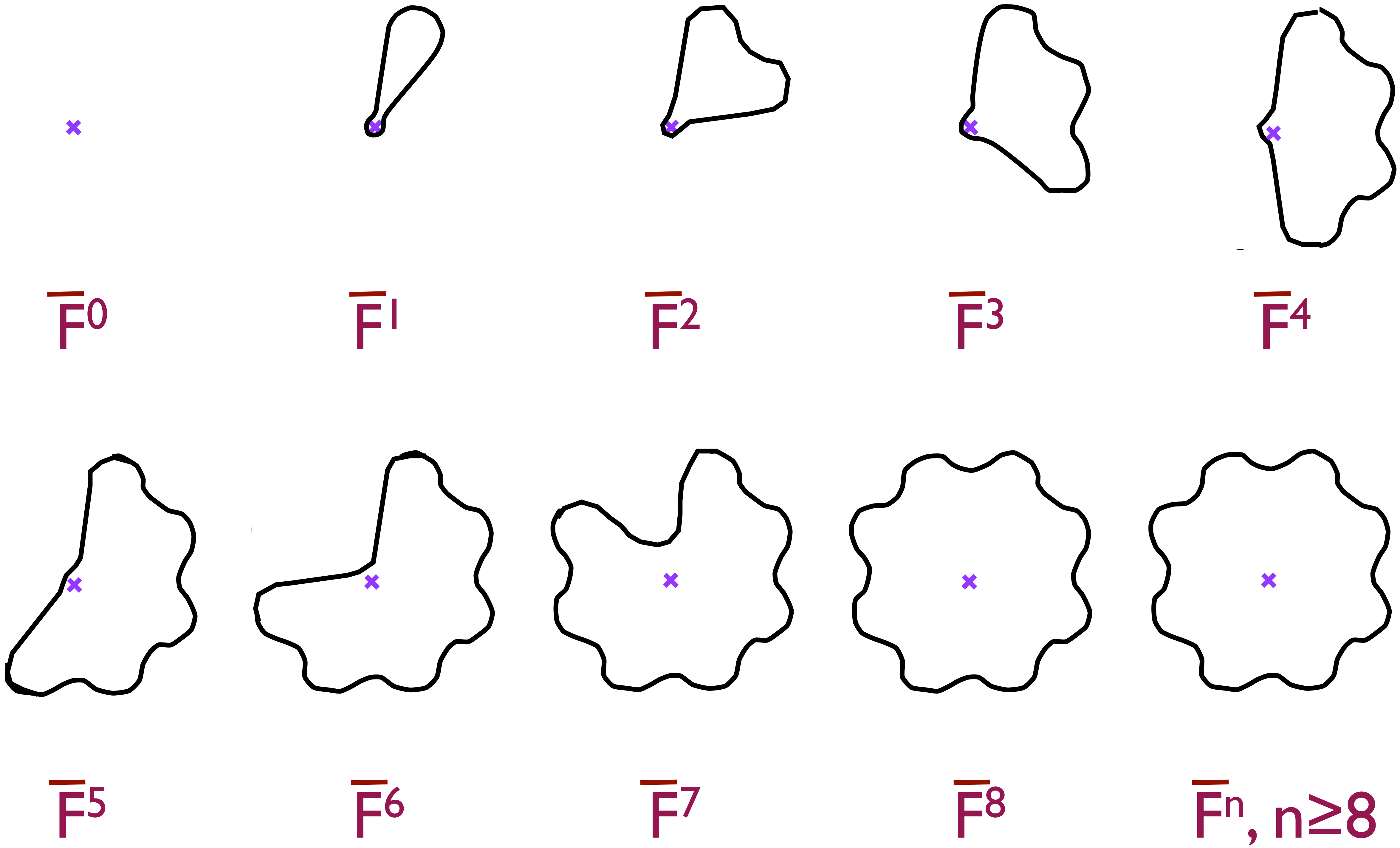
- $\overline{\text{corolla}} = \alpha(\text{corolla}) = \alpha(\text{lfp}^{\subseteq}\ \text{F}) = \text{lfp}^{\sqsubseteq}\ \overline{\text{F}}$

  since $\text{F}(x) = \text{petal} \cup \text{r}[45](x)$

  and $\overline{\text{F}}(y) = \overline{\text{petal}} \sqcup \overline{\text{r}}[45](y)$

  do commute: $\alpha(\text{F}(x)) = \overline{\text{F}}(\alpha(x))$

Iterates for the abstract corolla



Example of biological fixpoint

- Concrete reachability transformer:

$$\mathbb{F} : \begin{cases} \wp(\mathit{Species}) & \rightarrow \wp(\mathit{Species}) \\ X & \mapsto X \cup \left\{ c'_j \;\middle|\; \begin{array}{l} \exists R_k \in \mathcal{R}, c_1, \ldots, c_m \in X, \\ c_1, \ldots, c_m \rightarrow_{R_k} c'_1, \ldots, c'_n \end{array} \right\} \end{cases}$$

- Reachable species from $\mathit{Species}_0$

$$\mathsf{lfp}^{\subseteq}\ \lambda X.\ \mathit{Species}_0 \ \cup \mathbb{F}(X)$$

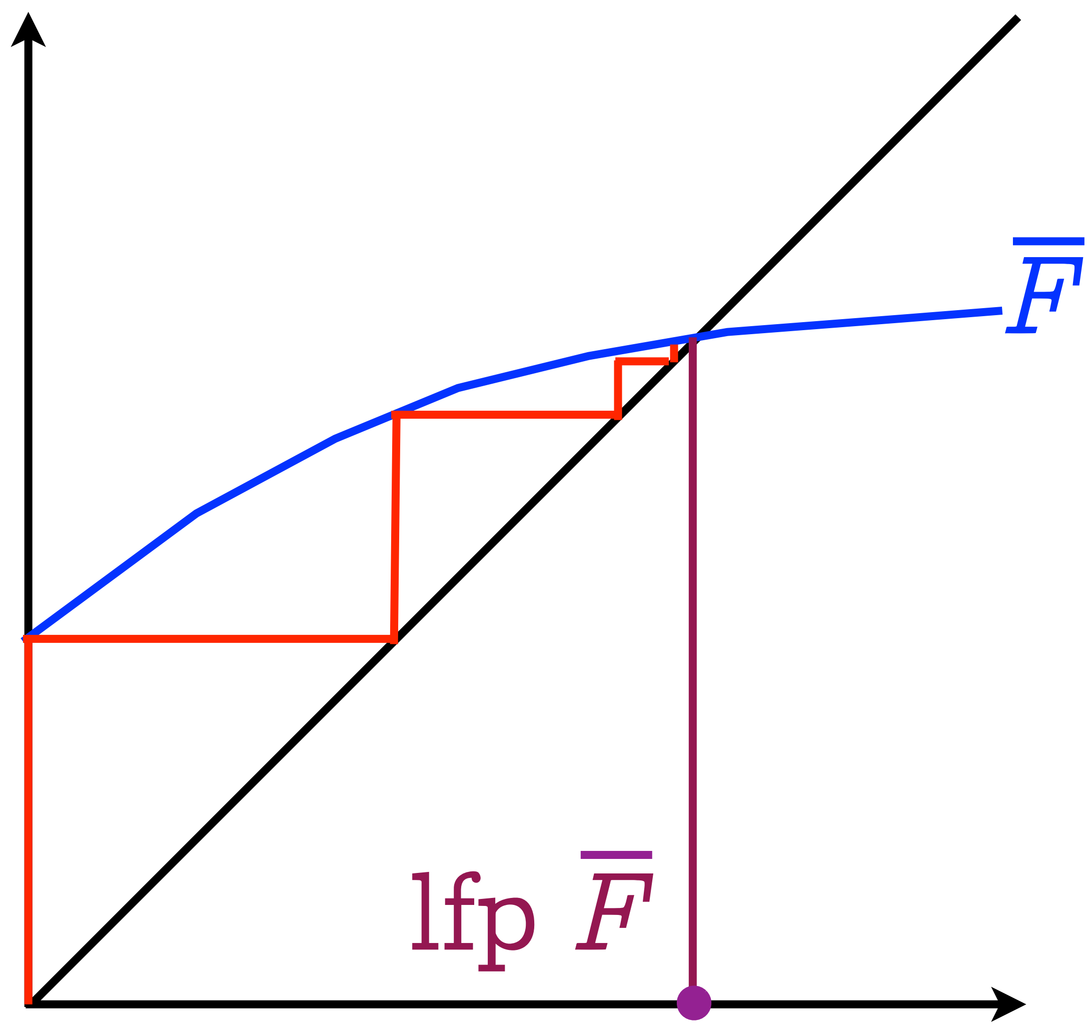
- Abstract reachability transformer:

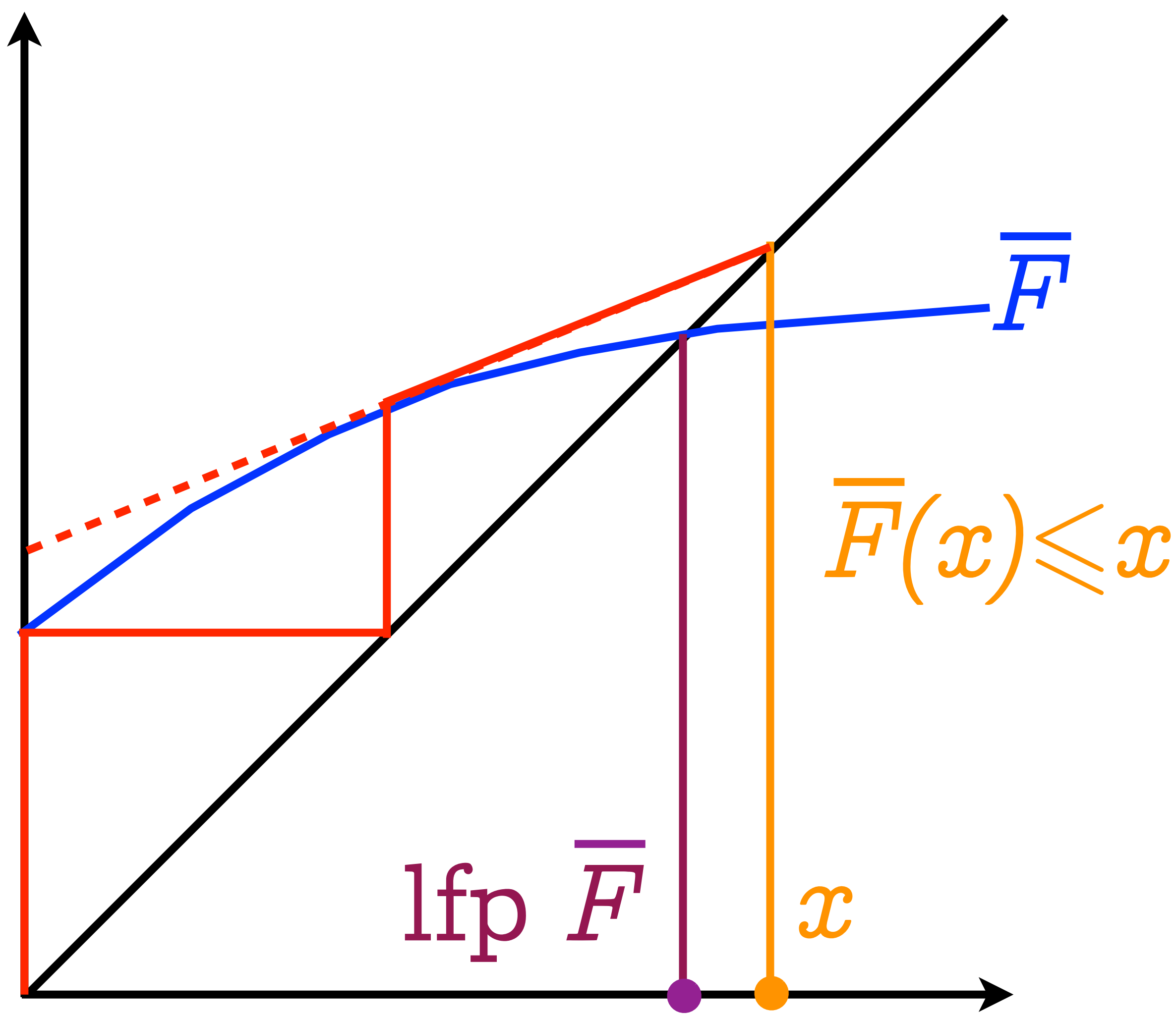
$$\mathbb{F}^\sharp : \begin{cases} \wp(\mathit{Local\_view}) & \rightarrow \wp(\mathit{Local\_view}) \\ X & \mapsto X \cup \left\{ \mathit{lv}'_j \;\middle|\; \begin{array}{l} \exists R_k \in \mathcal{R}, \mathit{lv}_1, \ldots, \mathit{lv}_m \in X, \\ \mathit{lv}_1, \ldots, \mathit{lv}_m \rightarrow^\sharp_{R_k} \mathit{lv}'_1, \ldots, \mathit{lv}'_n \end{array} \right\} \end{cases}$$

Jérôme Feret. Reachability Analysis of Biological Signalling Pathways by Abstract Interpretation. In Proceedings of the International Conference of Computational Methods in Sciences and Engineering (ICCMSE'2007), Corfu, Greece, 25--30 september, T.E. Simos(Ed.), 2007, American Institute of Physics conference proceedings 963.(2), pp 619--622.

Vincent Danos, Jérôme Feret, Walter Fontana, Jean Krivine: Abstract Interpretation of Cellular Signalling Networks. VMCAI 2008: 83-97

Convergence acceleration with widening





Infinite iteration

Accelerated iteration with widening
(e.g. with a widening based on the derivative as in Newton-Raphson method)

Abstraction of the graphical language

- Any graphical program can be abstracted by replacing the concrete objects/operations by abstract ones
- The soundness follows by induction on the syntax of programs

Applications of Abstract Interpretation in Computer Science

See Software Horror Stories (`www.cs.tau.ac.il/~nachumd/horror.html`)

Software

- Ait: static analysis of the worst-case execution time of control/command software (www.absint.com/ait/)
- Astrée: proof of absence of runtime errors in embedded synchronous real time control/command software (www.absint.com/astree/), AstréeA for asynchronous programs (www.astreea.ens.fr/)
- C Global Surveyor, NASA, static analyzer for flight software of NASA missions (www.cmu.edu/silicon-valley/faculty-staff/venet-arnaud.html)
- Checkmate: static analyzer of multi-threaded Java programs (www.pietro.ferrara.name/checkmate/)
- CodeContracts Static Checker, Microsoft (msdn.microsoft.com/en-us/devlabs/dd491992.aspx)
- Fluctuat: static analysis of the precision of numerical computations (www-list.cea.fr/labos/gb/LSL/fluctuat/index.html)

Software

- Infer: Static analyzer for C/C^{++} (monoidics.com/)
- Julia: static analyzer for Java and Android programs (www.juliasoft.com/juliasoft-android-java-verification.aspx?Id=201177234649)
- Predator: static analyzer of C dynamic data structures using separation logic (www.fit.vutbr.cz/research/groups/verifit/tools/predator/)
- Terminator: termination proof (www.cs.ucl.ac.uk/staff/p.ohearn/Invader/Invader/Invader_Home.html)
- etc

Libraries:

- Apron numerical domains library (apron.cri.ensmp.fr/library/)
- Parma Polyhedral Library (bugseng.com/products/ppl/)
- etc

Hardware

- (Generalized) symbolic trajectory evaluation (Intel)

Intel's Successes with Formal Methods

John Harrison

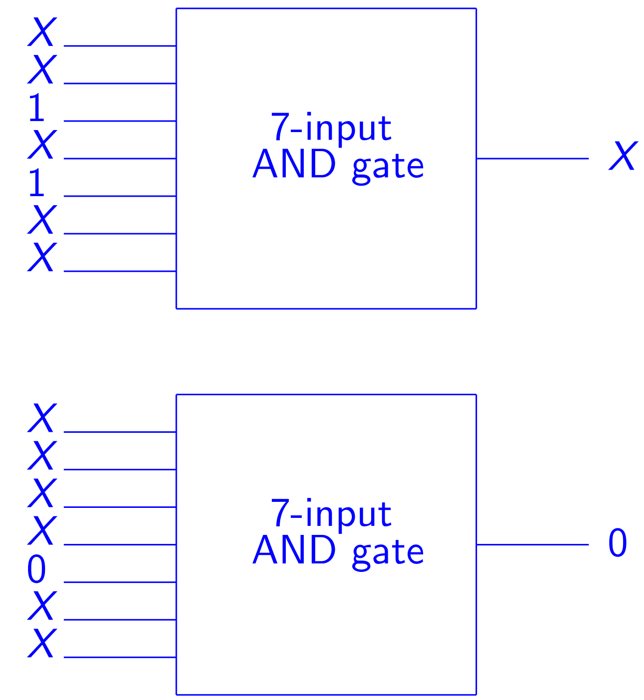
Intel Corporation

15 March 2012

Tsinghua software day, March 15, 2012, Tsinghua University, Beijing, China

Example of ternary simulation

If some inputs are undefined, the output often is too, but not always:



$$\begin{aligned} 0 &= \{0\} \\ 1 &= \{1\} \\ X &= \{0, 1\} \end{aligned}$$

Jin Yang and Carl-Johan H. Seger, *Generalized Symbolic Trajectory Evaluation — Abstraction in Action*, Formal Methods in Computer-Aided Design, Lecture Notes in Computer Science, 2002, Volume 2517/2002, 70–87.

Jin Yang; Seger, C.-J.H.; *Introduction to generalized symbolic trajectory evaluation*, IEEE Transactions onVery Large Scale Integration (VLSI) Systems 11(3), June 2003, 345–353.

System biology

- See SBFM'2012 !

Conclusion

Conclusion

If the simulation/analysis/checking of your model does not scale up, consider using (sound (and complete)) abstractions