ALT 2010
October 8, 2010

# Learning Figures with the Hausdorff Metric by Fractals

Mahito SUGIYAMA[1,2], Eiju HIROWATARI[3], Hideki TSUIKI[1] & Akihiro YAMAMOTO[1]

[1]Kyoto University, [2]JSPS Research Fellow
[3]The University of Kitakyushu

# Goal and Approach

- Constructing a computational learning model for analog data with **discretization**

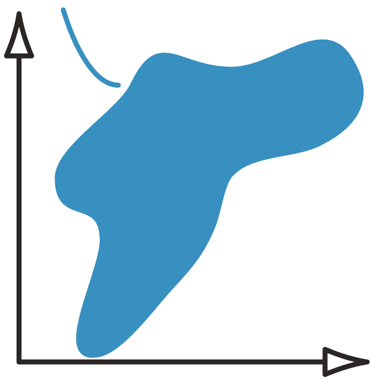
1. Gold-style learning model as a base model
2. Computable Analysis to give theoretical support for discretizing process of analog data
3. Fractals to represent (and compute) continuous objects

# Goal and Approach

- Constructing a computational learning model for analog data with **discretization**

1. Gold-style learning model as a base model
2. Computable Analysis to give theoretical support for discretizing process of analog data
3. Fractals to represent (and compute) continuous objects

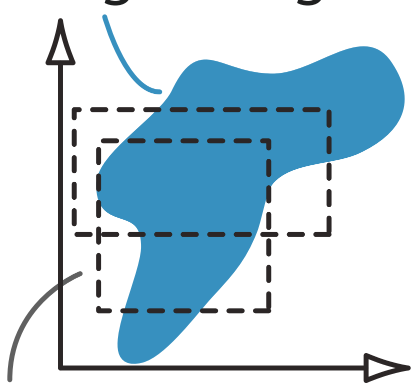
Targets: Figures (non-empty compact sets in $\mathbb{R}^n$)



# Goal and Approach

- Constructing a computational learning model for analog data with **discretization**
  1. Gold-style learning model as a base model
  2. Computable Analysis to give theoretical support for discretizing process of analog data
  3. Fractals to represent (and compute) continuous objects

Targets: Figures (non-empty compact sets in $\mathbb{R}^n$)

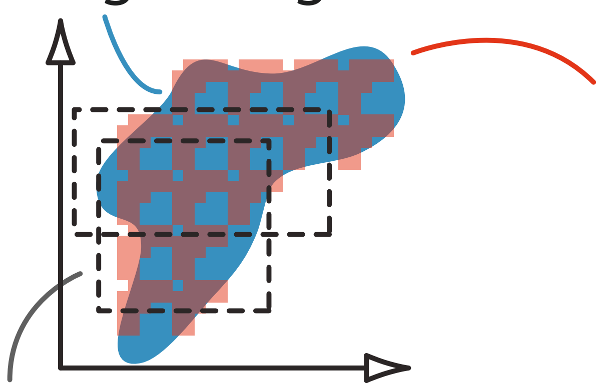


Examples for learning: Rational closed intervals

# Goal and Approach

- Constructing a computational learning model for analog data with **discretization**
  1. Gold-style learning model as a base model
  2. Computable Analysis to give theoretical support for discretizing process of analog data
  3. Fractals to represent (and compute) continuous objects

Targets: Figures (non-empty compact sets in $\mathbb{R}^n$)



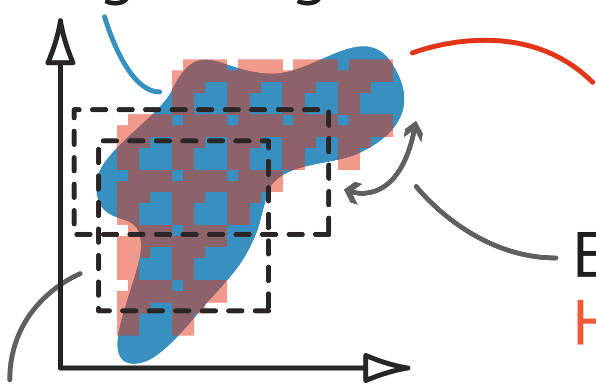
Hypotheses: Codes (Programs)
– representing fractals

Examples for learning: Rational closed intervals

# Goal and Approach

- Constructing a computational learning model for analog data with **discretization**
  1. Gold-style learning model as a base model
  2. Computable Analysis to give theoretical support for discretizing process of analog data
  3. Fractals to represent (and compute) continuous objects

Targets: Figures (non-empty compact sets in $\mathbb{R}^n$)



Hypotheses: Codes (Programs)
– representing fractals

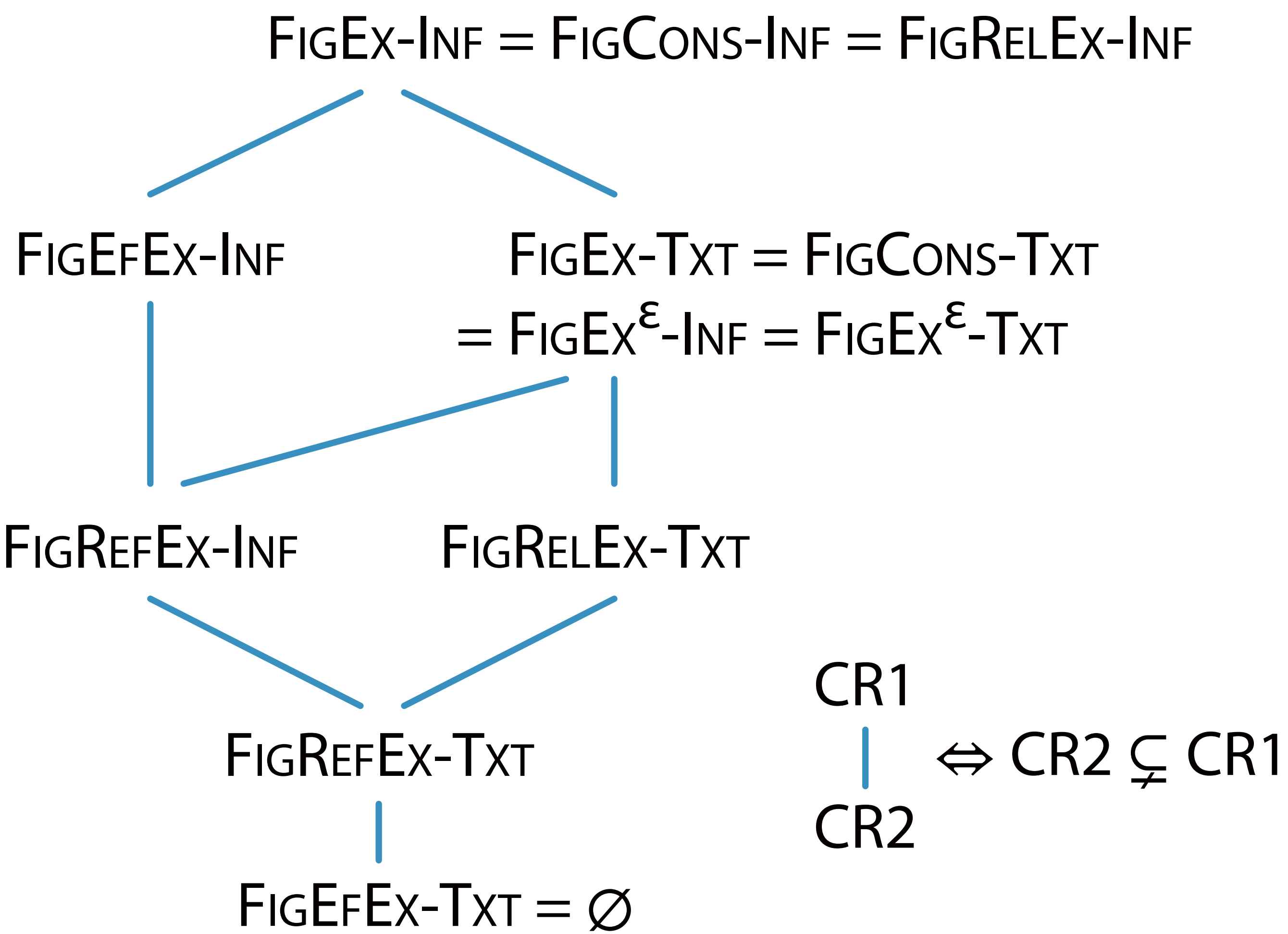
Evaluation of hypotheses:
Hausdorff metric (generalization errors)

Examples for learning: Rational closed intervals

# Main Results

1. We formulated learning of figures with self-similar sets (fractals) using the Gold-style learning model
   - Collage Theorem gives justification for self-similar sets
2. We analyzed the hierarchy of learnabilities (next slide)
3. We revealed the mathematical connection between Fractal Geometry and Computational Learning
   - The complexity of learning (sample size) is measured by using the Hausdorff dimension and the VC dimension
     - The Hausdorff dimension and the VC dimension are key concepts of Fractal Geometry and the Valiant-style learning model, respectively

# Precise Result of the Hierarchy



# Outline

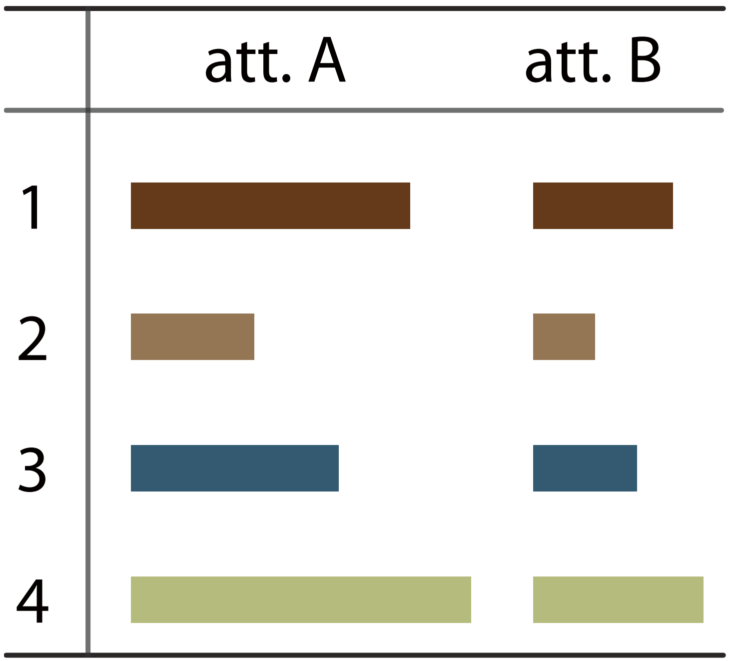
- Background
- Methods for learning figures
- Learnabilities under various learning criteria
- Characterization with $\dim_H$ and $\dim_{VC}$
- Conclusion

# Outline

- Background
- Methods for learning figures
- Learnabilities under various learning criteria
- Characterization with $\dim_H$ and $\dim_{VC}$
- Conclusion

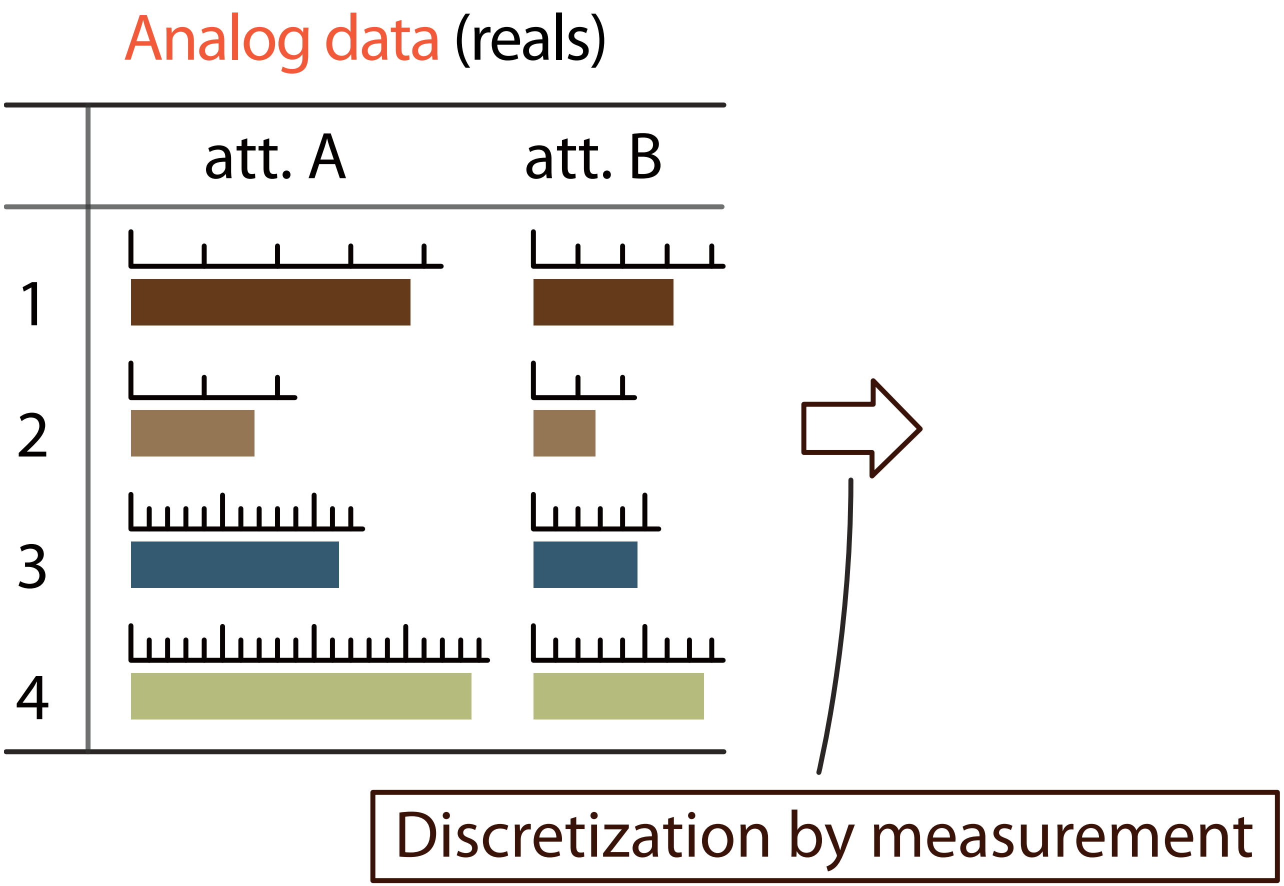
# Discretization and Learning

Analog data (reals)




Goal: Learning the relation between attributes A and B

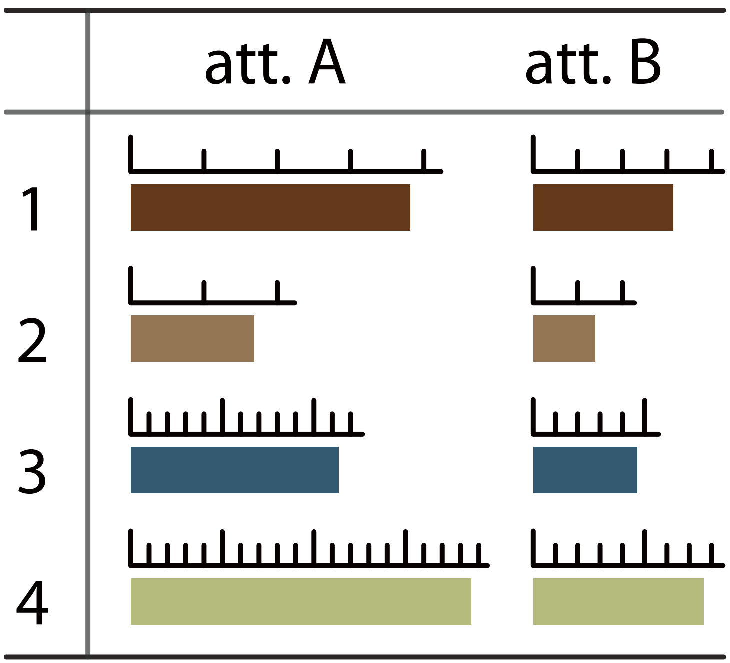
# Discretization and Learning




Goal: Learning the relation between attributes A and B

# Discretization and Learning

Analog data (reals)




Digital data (rationals)

| att. A | att. B |
|---|---|
| 1.5 | 0.6 |
| 0.6 | 0.2 |
| 1.1 | 0.4 |
| 1.8 | 0.7 |

Discretization by measurement

Goal: Learning the relation between attributes A and B

# Discretization and Learning

Analog data (reals)

| | att. A | att. B |
|---|---|---|
| 1 | 1.539582... | 0.6069... |
| 2 | 0.676711... | 0.2655... |
| 3 | 1.111577... | 0.4998... |
| 4 | 1.871501... | 0.7569... |

Digital data (rationals)

| att. A | att. B |
|---|---|
| 1.5 | 0.6 |
| 0.6 | 0.2 |
| 1.1 | 0.4 |
| 1.8 | 0.7 |

Discretization by measurement

Goal: Learning the relation between attributes A and B

# Discretization and Learning

Analog data (reals)

| | att. A | att. B |
|---|---|---|
| 1 | 1.539582… | 0.6069… |
| 2 | 0. | 655… |
| 3 | 1.111577… | 0.4998… |
| 4 | 1.871501… | 0.7569… |

Data assumed theoretically

Digital data (rationals)

| att. A | att. B |
|---|---|
| 1.5 | 0.6 |
| | |
| 1.1 | 0.4 |
| 1.8 | 0.7 |

Data used for learning

Discretization by measurement

Goal: Learning the relation between attributes A and B

# An Fatal Error Caused by Discretization

- Solve the system of linear equations [Schroder, 03]

$$40157959.0\,x + 67108865.0\,y = 1$$
$$67108864.5\,x + 112147127.0\,y = 0$$

  – Obtained by the well-known formula

$$x = \frac{b_1 a_{22} - b_2 a_{12}}{a_{11} a_{22} - a_{21} a_{12}}, \quad y = \frac{b_2 a_{11} - b_1 a_{21}}{a_{11} a_{22} - a_{21} a_{12}}$$

- By floating point arithmetic with double precision variables (IEEE 754) :

$$x = 112147127, \quad y = -67108864.5$$

- The correct solution:

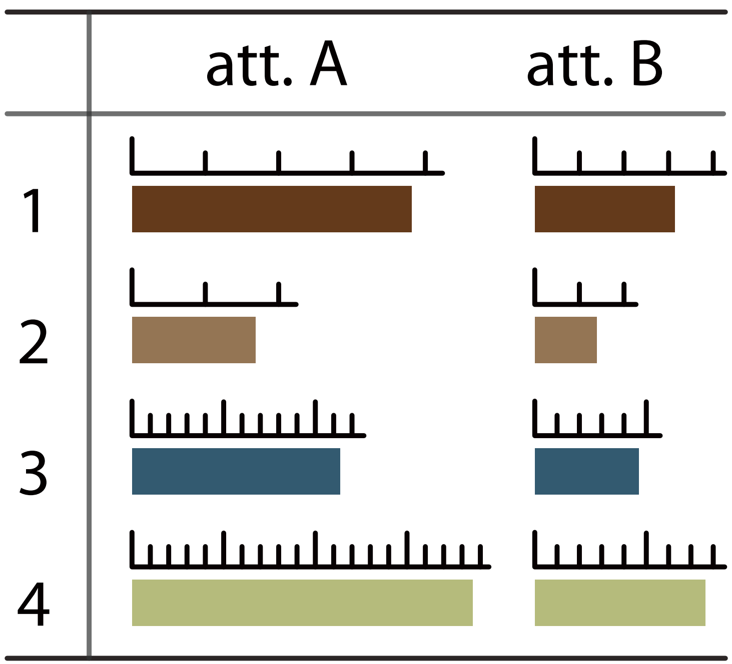
$$x = 224294254, \quad y = -134217729$$

# Our Strategy

- Use effective computing in Computable Analysis to treat discretization precess appropriately
  - While a computer reads more and more precise information of the input, it produces more and more accurate approximations of the result
- Construct an effective learning with the Gold-style learning model
  - While a learner reads more and more precise examples of the target, it produces more and more accurate hypotheses of the target
    - This accuracy corresponds to a generalization error

# Treat Data as Intervals

Continuous objects (reals)



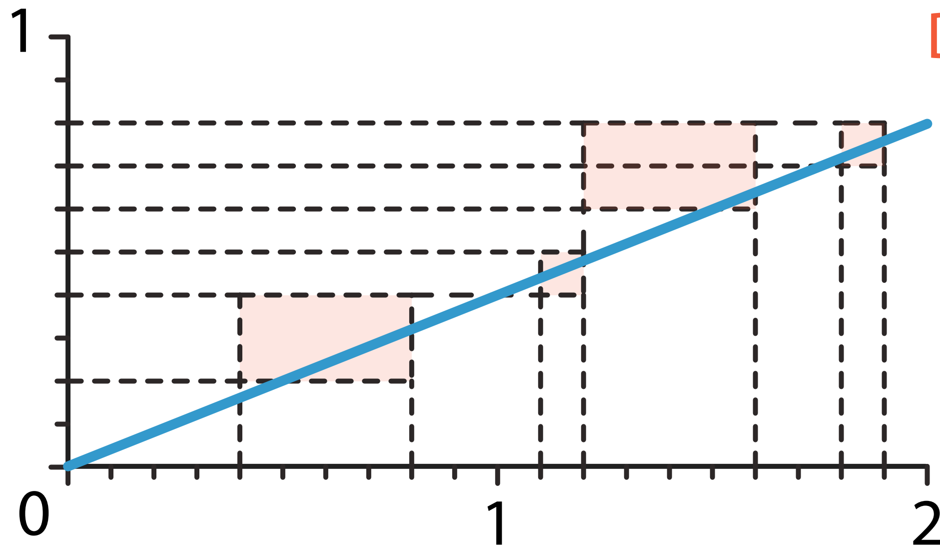

Digital data (rationals)

| att. A | att. B |
|---|---|
| 1.2~1.6 | 0.6~0.8 |
| 0.4~0.8 | 0.2~0.4 |
| 1.1~1.2 | 0.4~0.5 |
| 1.8~1.9 | 0.7~0.8 |

Discretization by measurement

Goal: Learning the relation between attributes A and B

# Leaning from Geometrical View




Digital data (rationals)

| att. A | att. B |
|---|---|
| 1.2~1.6 | 0.6~0.8 |
| 0.4~0.8 | 0.2~0.4 |
| 1.1~1.2 | 0.4~0.5 |
| 1.8~1.9 | 0.7~0.8 |

- Discretized data are intervals in $\mathbb{R}^n$
  - The width of an interval means an error of the datum
- A learner learns a figure that intersects with all intervals

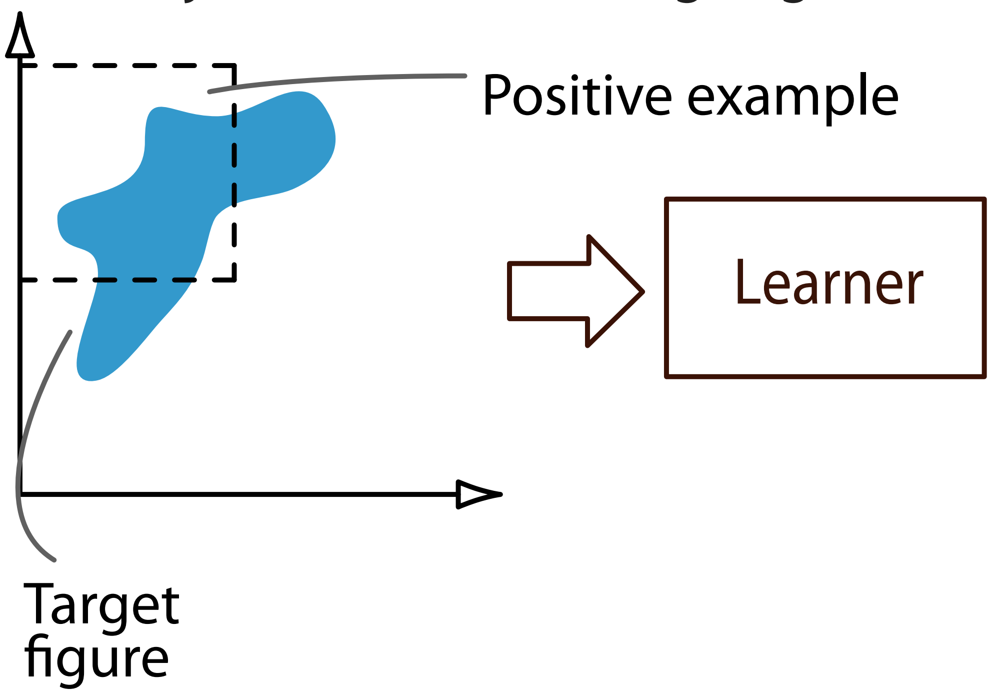
# Outline

- Background
- Methods for learning figures
- Learnabilities under various learning criteria
- Characterization with $\dim_H$ and $\dim_{VC}$
- Conclusion

# Summary of Learning Process

Positive examples: Rational closed intervals intersecting the learning target

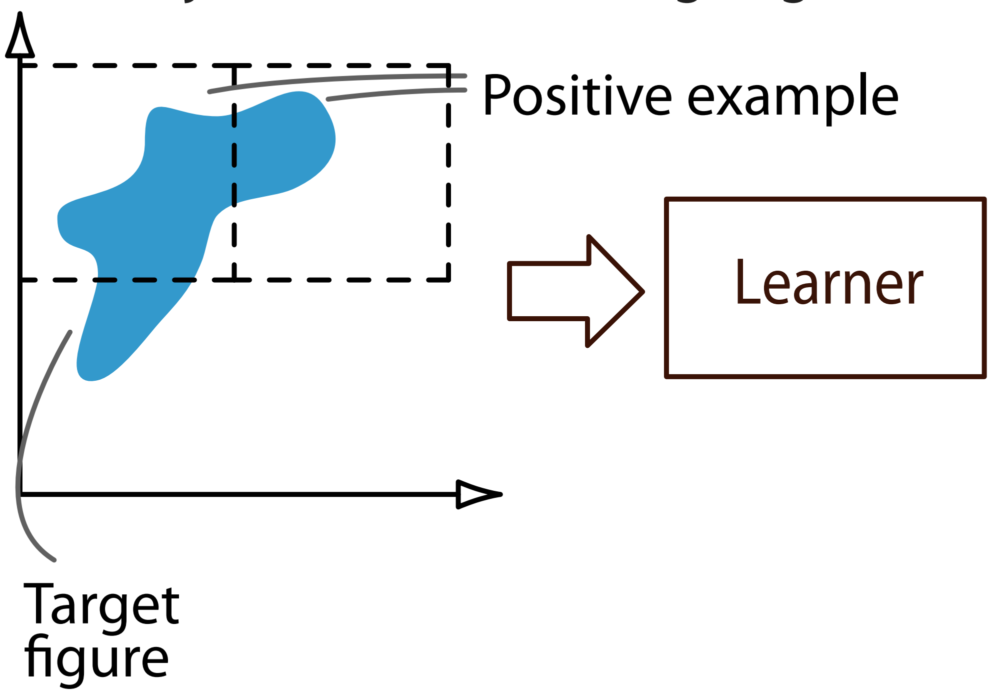
Negative examples: Rational closed intervals disjoint with the learning target



# Summary of Learning Process

Positive examples: Rational closed intervals intersecting the learning target

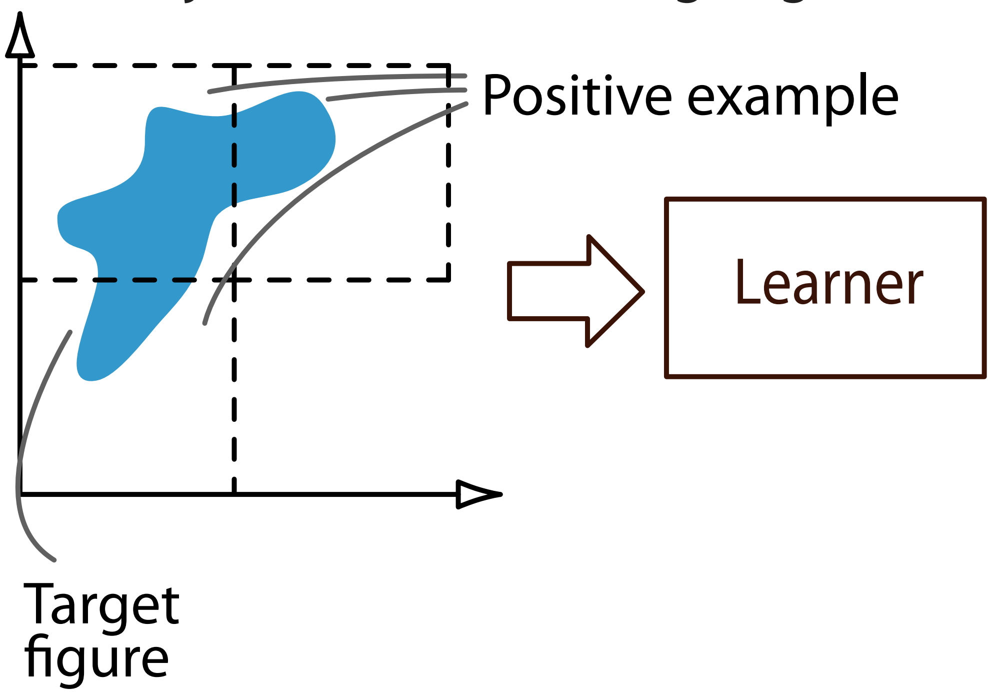
Negative examples: Rational closed intervals disjoint with the learning target



# Summary of Learning Process

Positive examples: Rational closed intervals intersecting the learning target

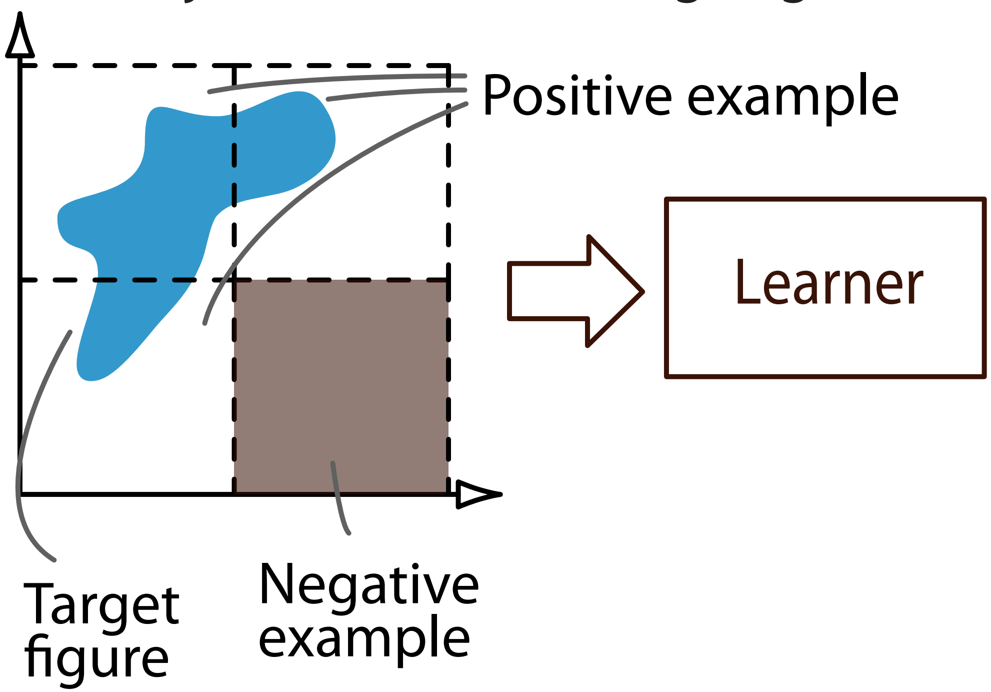
Negative examples: Rational closed intervals disjoint with the learning target



# Summary of Learning Process

Positive examples: Rational closed intervals intersecting the learning target

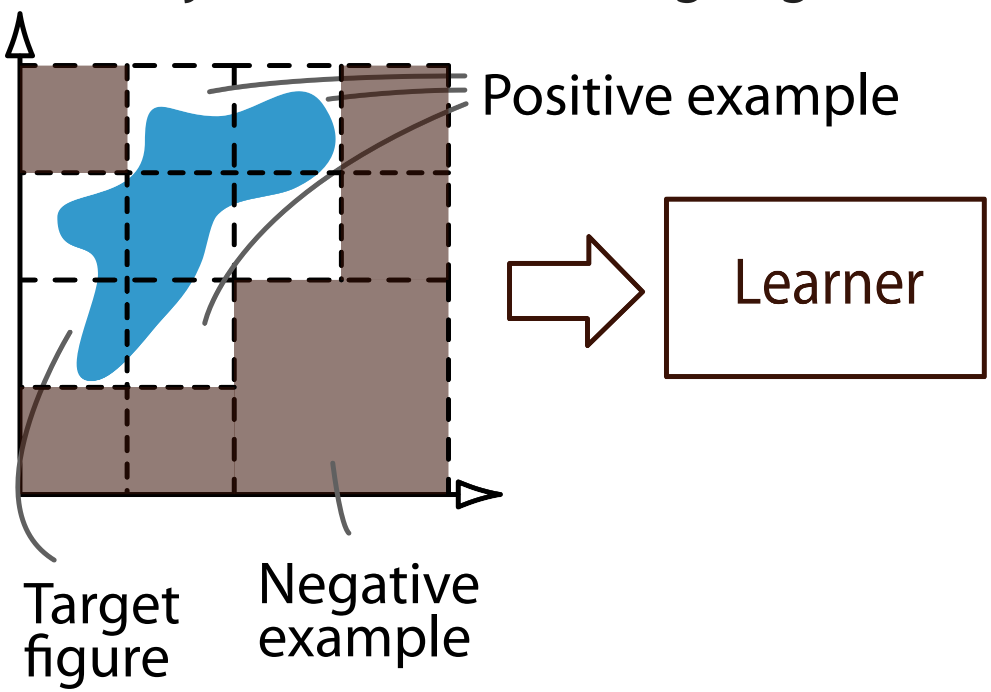
Negative examples: Rational closed intervals disjoint with the learning target



# Summary of Learning Process

Positive examples: Rational closed intervals intersecting the learning target

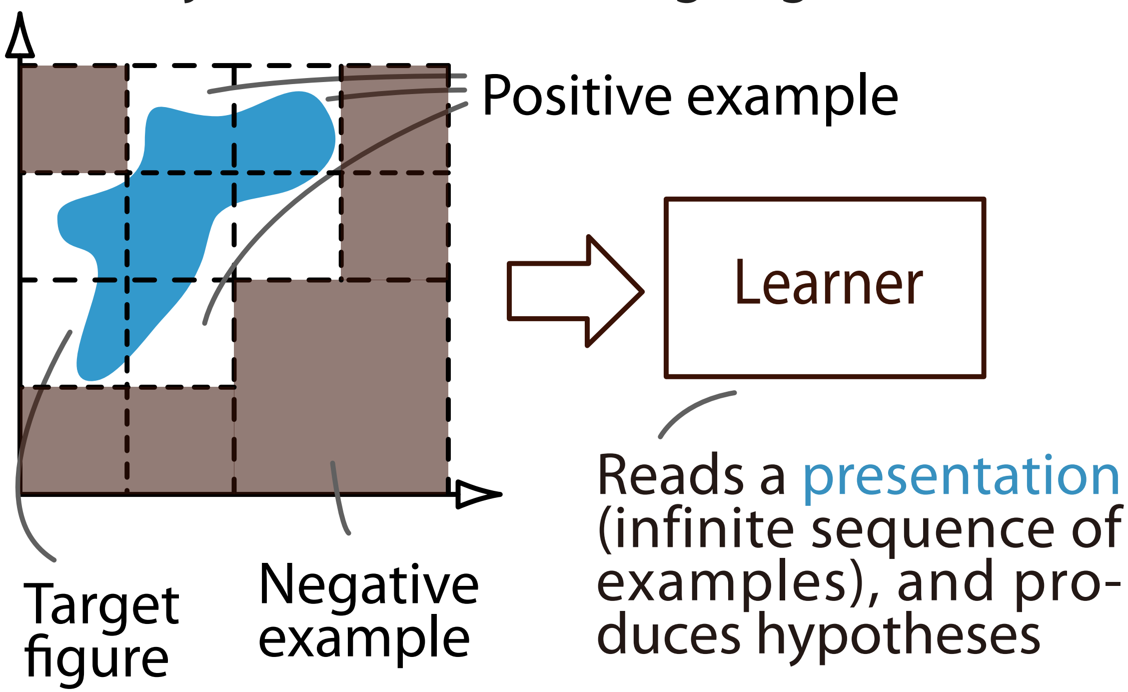
Negative examples: Rational closed intervals disjoint with the learning target



# Summary of Learning Process

Positive examples: Rational closed intervals intersecting the learning target

Negative examples: Rational closed intervals disjoint with the learning target

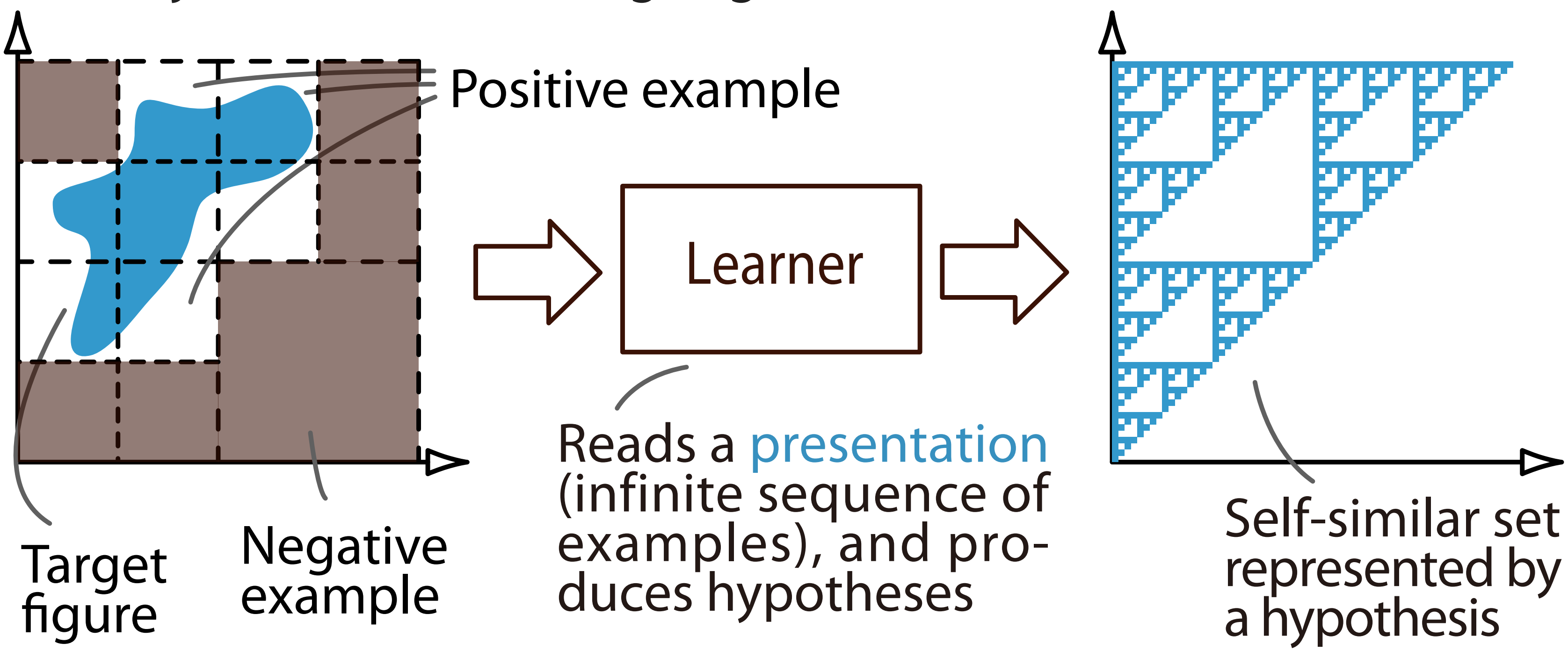


# Summary of Learning Process

Positive examples: Rational closed intervals intersecting the learning target

Negative examples: Rational closed intervals disjoint with the learning target

Hypotheses: Codes that represent self-similar sets (self-similar programs)

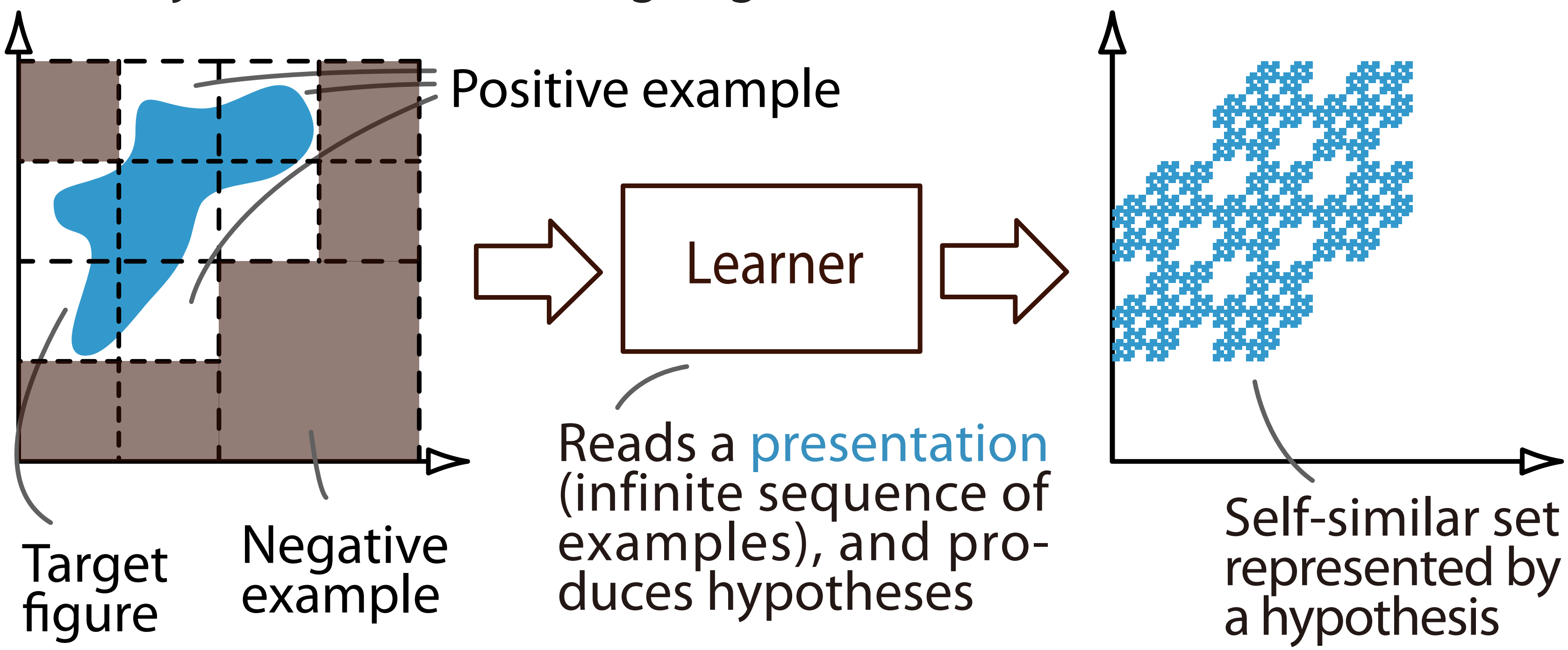


# Summary of Learning Process

Positive examples: Rational closed intervals intersecting the learning target

Negative examples: Rational closed intervals disjoint with the learning target

Hypotheses: Codes that represent self-similar sets (self-similar programs)

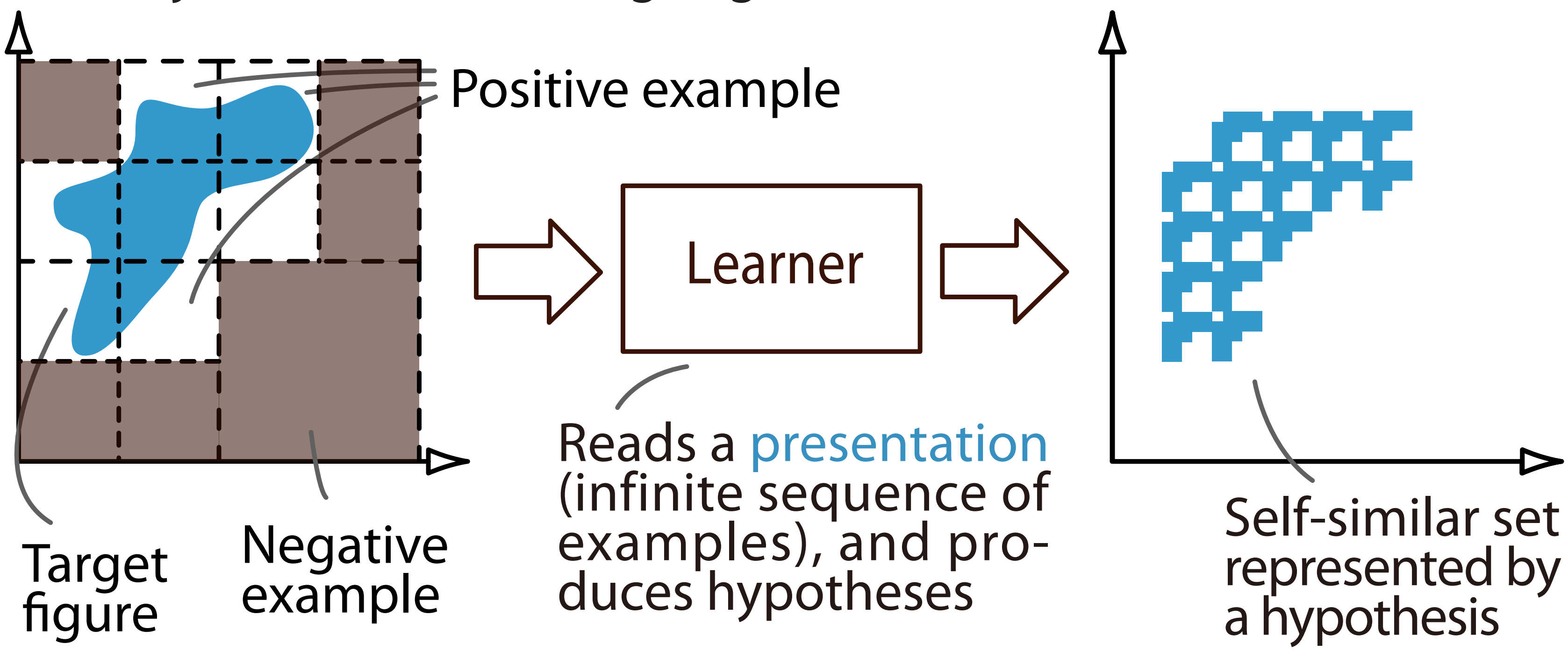


# Summary of Learning Process

Positive examples: Rational closed intervals intersecting the learning target

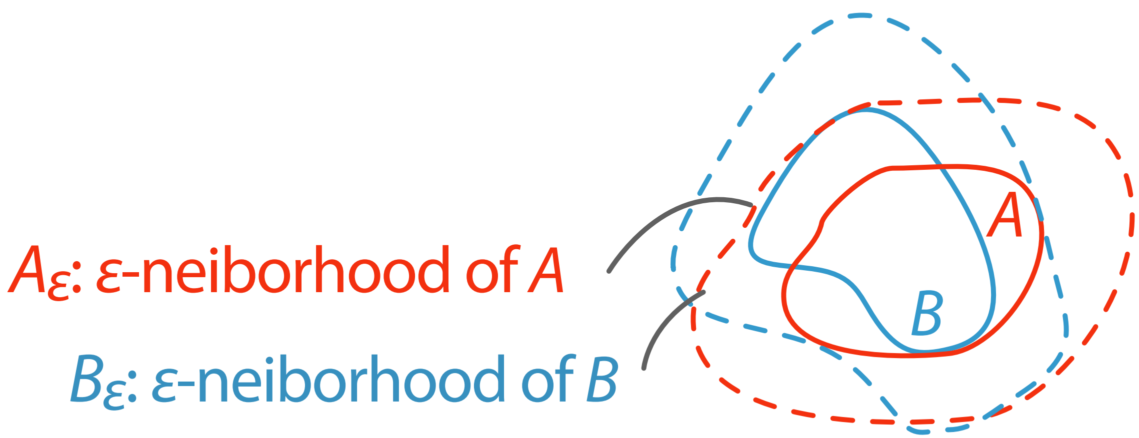
Negative examples: Rational closed intervals disjoint with the learning target

Hypotheses: Codes that represent self-similar sets (self-similar programs)



# Generalization Errors

- We measure "goodness" of a hypothesis by a generalization error
  - We use the Hausdorff metric (distance between figures)
- The Hausdorff distance between figures $A$ and $B$ (denoted by $d_{\mathrm{H}}(A, B)$) is the minimum $\varepsilon$ satisfying $A \subset B_\varepsilon$ and $B \subset A_\varepsilon$

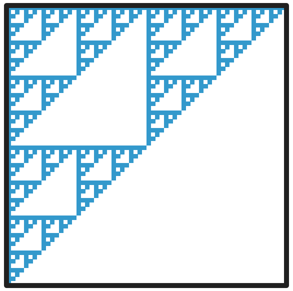


# Hypotheses Represent Self-Similar Sets

- We use logic programs to represent self-similar sets

$$\begin{cases} \varphi_1 \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \\ \varphi_2 \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{2} \end{bmatrix} \\ \varphi_3 \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \end{bmatrix} \end{cases} \Rightarrow \begin{cases} Path(\lambda) \\ Path(0x) \leftarrow Path(x) \\ Path(1x) \leftarrow Path(x) \\ Path(3x) \leftarrow Path(x) \end{cases}$$

Sierpiński triangle



- Any figure can be approximated by some self-similar set (Corollary of Collage Theorem) [Falconer, 03]
  - For all figure $K$ and $\delta > 0$, there exists a self-similar set $V$ such that $d_H(K, V) < \delta$

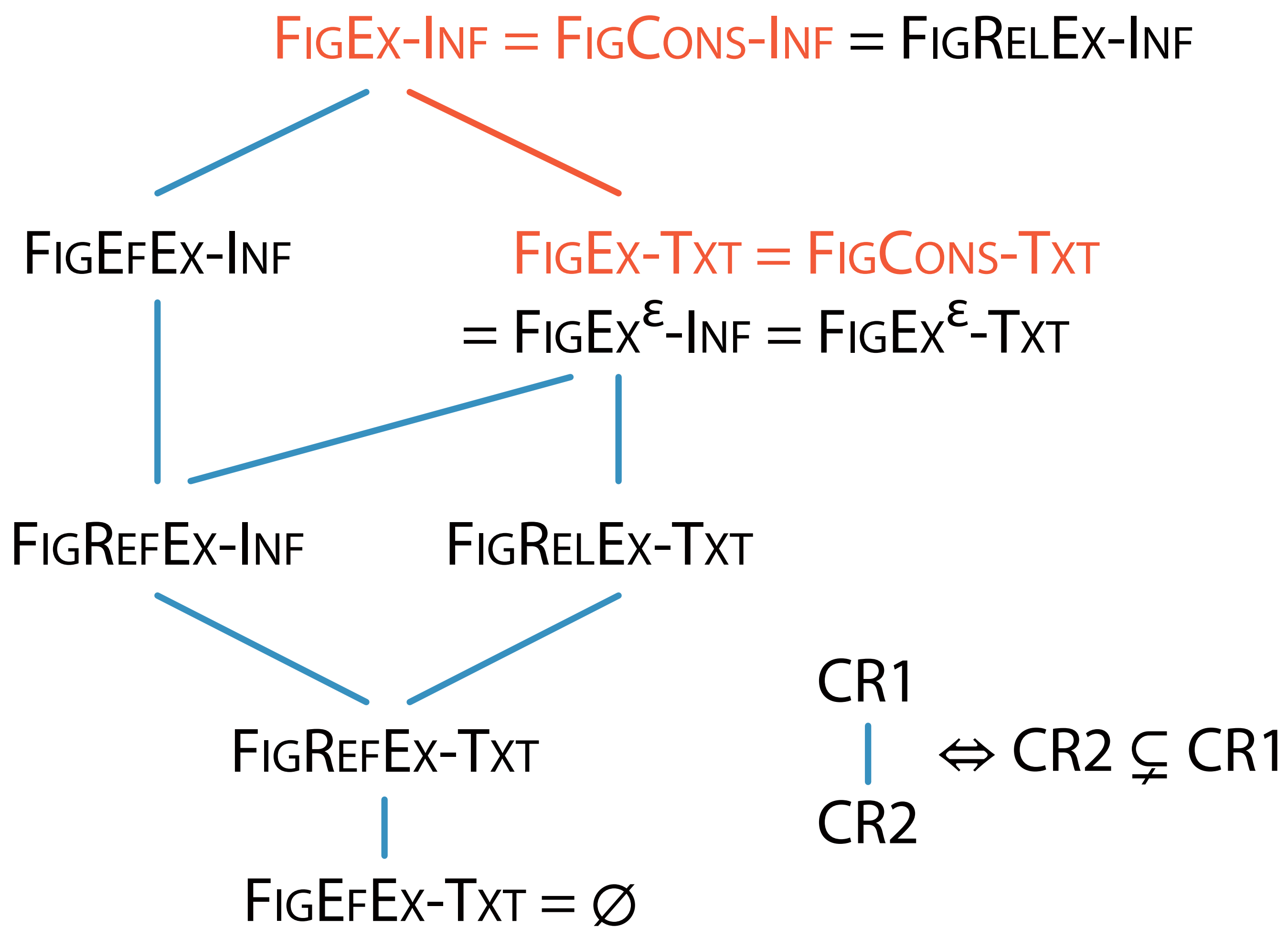
# Outline

- Background
- Methods for learning figures
- Learnabilities under various learning criteria
- Characterization with $\dim_{\mathrm{H}}$ and $\dim_{\mathrm{VC}}$
- Conclusion

# Learning Self-Similar Sets in the Limit

- We formulate learning of self-similar sets based on the Gold-style learning model
  - A target is always represented by some program
- A learner FigEx-Inf-learns (FigEx-Txt-learns) a set of figures $\mathscr{F} \subseteq \mathscr{K}^*$ $\iff$ For all $K \in \mathscr{F}$ and informants (texts), its output converges to a hypothesis $P$, where $\mathbf{GE}(K, P) = 0$
  - $\mathscr{K}^*$: The set of figures, $\mathbf{GE}(K, P) := d_{\mathrm{H}}(K, \kappa(P))$
    - $\kappa(P)$ denotes the set represented by a program $P$
  - Notation: $\mathscr{F}$ is **CR**-learnable $\iff$ $\mathscr{F} \in \mathbf{CR}$
- We also consider consistent learning (FigCons-Inf- and FigCons-Txt-learning), where every hypothesis is consistent with received examples so far

# The Hierarchy of Learnabilities

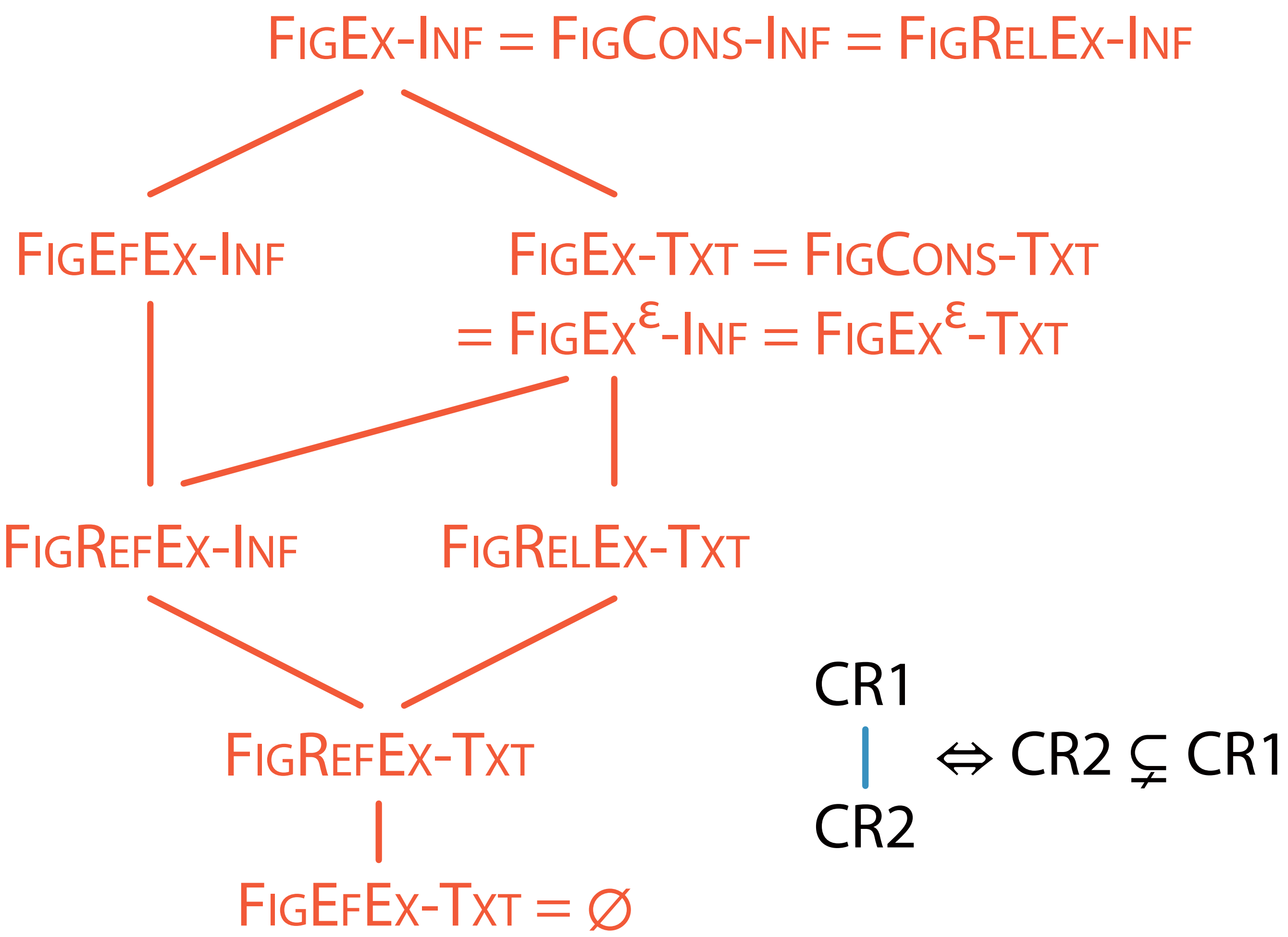


# Approach to All Figures

- In FigEx-learning, the space $\mathscr{F}$ for learning (concept space) is given (*a priori*)
  - When a target figure $K \notin \mathscr{F}$, nothing is guaranteed
- Here we give some guarantee to such cases
  - We treat not only self-similar sets, but also figures
  - The similar model has been studied in learning of languages [Mukouchi and Arikawa, 95]

1. Refutable learning: a learner stops (if a target $K \notin \mathscr{F}$)
2. Reliable learning: hypotheses do not converge (if $K \notin \mathscr{F}$)
3. Effective learning: generalization errors converge to zero
4. Learning with generalization error bounds: hypotheses converge under the error bounds

# The Hierarchy of Learnabilities



# Conclusion So Far

- Learning of figures was realized in computational manner using the Gold-style learning model
  - Discretization process was treated by using the effective computing model in Computable Analysis
  - Generalization error of a hypothesis was measured by the Hausdorff metric
- Learnabilities of figures were analyzed under existing and new learning criteria

# Conclusion So Far

- Learning of figures was realized in computational manner using the Gold-style learning model
  - Discretization process was treated by using the effective computing model in Computable Analysis
  - Generalization error of a hypothesis was measured by the Hausdorff metric
- Learnabilities of figures were analyzed under existing and new learning criteria
- We show a mathematical connection between Fractal Geometry and Computational Learning using the Hausdorff dimension and the VC dimension

# Outline

- Background
- Methods for learning figures
- Learnabilities under various learning criteria
- Characterization with $\dim_H$ and $\dim_{VC}$
- Conclusion

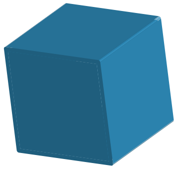
# The Hausdorff Dimension ($\dim_H$)

- The Hausdorff dimension is a central concept of fractals
  - This indicates how much space a sets occupies near to each of its points
  - Defined by the Hausdorff measure
- Extension of usual (topological) dimension

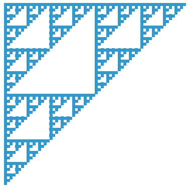
Point  = 0

Line  = 1

Plane  = 2

Cube  = 3

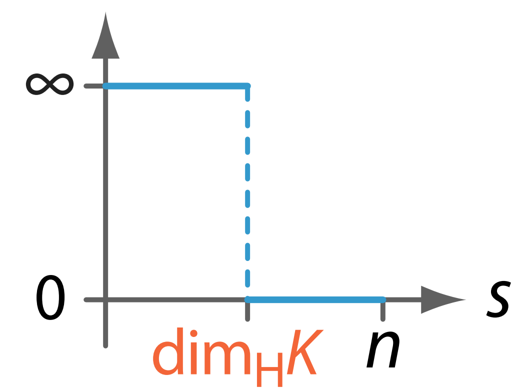
Sierpiński triangle



= log3 / log2
= 1.584...

# The Hausdorff Dimension ($\dim_H$)

- The Hausdorff dimension is a central concept of fractals
  - This indicates how much space a sets occupies near to each of its points
  - Defined by the Hausdorff measure
- Hausdorff measures generalize ideas of length, area, …
  - Defined by using “covering” of a set
- $s$-dimensional Hausdorff measure of $K := \lim_{\varepsilon \to 0} \mathscr{H}^s_\varepsilon(K)$
  - Countable set $\boldsymbol{U}$ is a $\varepsilon$-cover of $K \iff \forall U \in \boldsymbol{U}.\, |U| \leqslant \varepsilon$, and $X \subset \bigcup_{U \in \boldsymbol{U}} U$
  - $\mathscr{H}^s_\varepsilon(K) = \inf\{\sum_{U \in \boldsymbol{U}} |U|^s \mid \boldsymbol{U}$ is a $\varepsilon$-cover of $K\}$

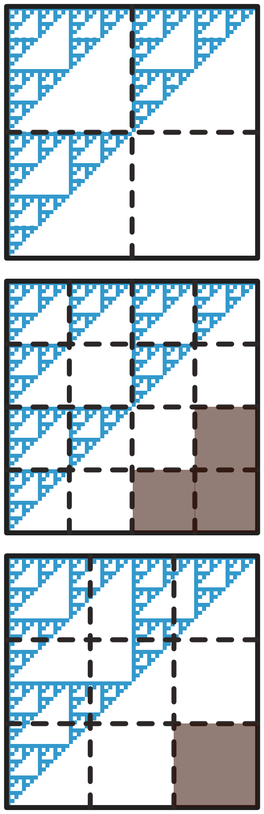


# Characterization with $\dim_H$

- General case:
  If level $k$ is large enough, for every target figure $K$ and for any $s < \dim_H K$, the figure $K$ can be covered by $N$ intervals, where $N \geqslant b^{ks}$
- Special case:
  Moreover, if a target figure $K$ is represented by some self-similar program $P$, then $K$ can be covered by $N$ intervals, where $N \geqslant b^{k \dim_H K}$
  - We use base-$b$ partition in both cases

# Characterization with $\dim_H$

- Example:
  - $K$: The Sierpiński triangle ($\dim_H K = 1.584\ldots$)
  - $N(K)$: # of level-$k$ positive examples
- With 2-dimensional base-2 partition
  - Level 1: $3 \leqslant N(K)$ $(2^{\dim_H K} = 3)$
  - Level 2: $9 \leqslant N(K)$ $(4^{\dim_H K} = 9)$
- With 2-dimensional base-3 partition
  - Level 1: $6 \leqslant N(K)$ $(3^{\dim_H K} = 5.70\ldots)$

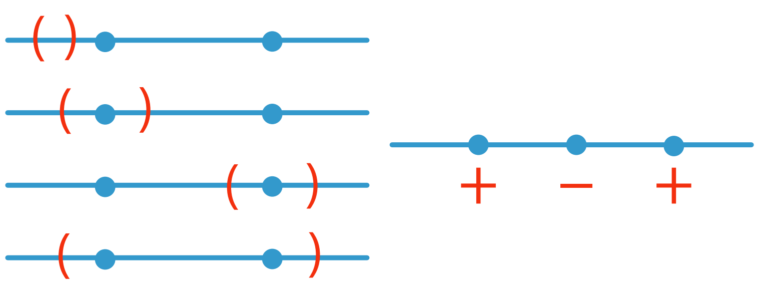


# The VC Dimension ($\dim_{\mathrm{VC}}$)

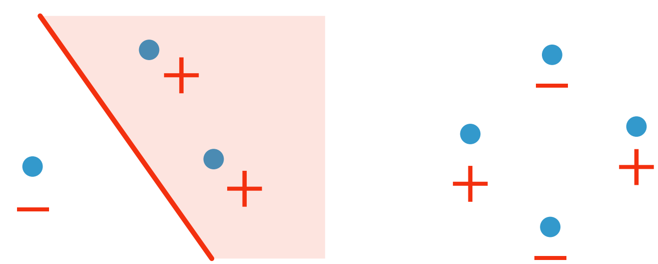
- The VC dimension is a parameter of separability (complexity) of a class
  - How many points can be separated?
  - In the Valiant-style (PAC) learning model, the sample size is characterized by the VC dimension

***I***: The class of open intervals in the real line $\mathbb{R}$

***H***: The class of half spaces in 2-dimensional real-space $\mathbb{R}^2$



$\dim_{\mathrm{VC}} \boldsymbol{I} = 2$



$\dim_{\mathrm{VC}} \boldsymbol{H} = 3$

# Characterization with $\dim_H$ and $\dim_{VC}$

- The VC dimension of the set of level $k$ programs $\mathscr{P}^k$ is equal to the cardinality of the number of level $k$ intervals

# Characterization with $\dim_{\mathrm{H}}$ and $\dim_{\mathrm{VC}}$

- The VC dimension of the set of level $k$ programs $\mathscr{P}^k$ is equal to the cardinality of the number of level $k$ intervals
- General case:
  If level $k$ is large enough, for every target figure $K$ and for any $s < \dim_{\mathrm{H}} K$, the figure $K$ can be covered by $N$ intervals, where $N \geqslant b^{ks}$
- Special case:
  Moreover, if a target figure $K$ is represented by some self-similar program $P$, then $K$ can be covered by $N$ intervals, where $N \geqslant b^{k \dim_{\mathrm{H}} K}$
  - We use base-$b$ partition in both cases

# Characterization with $\dim_H$ and $\dim_{VC}$

- The VC dimension of the set of level $k$ programs $\mathscr{P}^k$ is equal to the cardinality of the number of level $k$ intervals
- General case:
  If level $k$ is large enough, for every target figure $K$ and for any $s < \dim_H K$, the figure $K$ can be covered by $N$ intervals, where $N \geqslant (\dim_{VC} \mathscr{P}^k)^{s/n}$
- Special case:
  Moreover, if a target figure $K$ is represented by some self-similar program $P$, then $K$ can be covered by $N$ intervals, where $N \geqslant (\dim_{VC} \mathscr{P}^k)^{\dim_H K/n}$
  - We use base-$b$ partition in both cases

# Outline

- Background
- Methods for learning figures
- Learnabilities under various learning criteria
- Characterization with $\dim_H$ and $\dim_{VC}$
- Conclusion

# Conclusion

- Learning of figures was realized in computational manner using the Gold-style learning model
  - Discretization process was treated by using the effective computing model in Computable Analysis
  - Generalization error of a hypothesis was measured by the Hausdorff metric
- Learnabilities of figures were analyzed under existing and new learning criteria
- A novel mathematical connection between Fractal Geometry and Computational Learning was shown using the Hausdorff dimension and the VC dimension

# Appendix

# Background

- Machine learning from analog data
  - The discrete Fourier analysis is a typical method
    - But only the direction of the time axis is discretized
  - We discretized all axes and give a fully computational learning model
- What kind of representation system is appropriate?
  - Recursive algorithms are key to bridge continuous and discrete
    - FFT is used in the discrete Fourier analysis
  - Fractals are geometric concepts of recursiveness
    - They are recursive algorithms to generate fractals
- Formulate “Learning figures by fractals”

# Computational and Statistical Learning

analyzed in detail started from the Gold-style learning model — Computational learning

great success in Knowledge Discovery — Statistical learning

| | Computational learning | Statistical learning |
|---|---|---|
| Targets | Discrete (languages) | Continuous (real-valued functions) |
| Representations (Hypotheses) | Symbol-level (algorithms) | Signal-level (multilayer perceptrons) |
| Evaluation | Generalization error (Characteristic func.) | Generalization error (KL divergence) |

# Computational and Statistical Learning

analyzed in detail started from the Gold-style learning model

great success in Knowledge Discovery

introduce geometric concept

| | Computational learning | Statistical learning |
|---|---|---|
| Targets | Discrete (languages) | Continuous (real-valued functions) |
| Representations (Hypotheses) | Symbol-level (algorithms) | Signal-level (multilayer perceptrons) |
| Evaluation | Generalization error (Characteristic func.) | Generalization error (KL divergence) |

# Computational and Statistical Learning

analyzed in detail started from
the Gold-style learning model

great success in
Knowledge Discovery

introduce geometric
concept

| | Computational learning | | Statistical learning |
|---|---|---|---|
| Targets | Discrete (languages) | ⇨ Continuous (Figures) | tinuous ed functions) |
| Representations (Hypotheses) | Symbol-level (algorithms) | ⇨ Symbol-level (Self-similar sets) | el ptrons) |
| Evaluation | Generalization error (Characteristic func.) | ⇨ Generalization error (Hausdorff metric) | C |

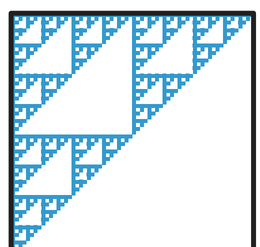
# Self-Similar Sets

- Self-similar sets are in a class of fractals
  - defined as a fixed point of a finite set of contractions
- $\varphi : X \to X$ is a contraction $\iff d(\varphi(x), \varphi(y)) \le c\,d(x, y)$
  $(0 < c < 1$, $d$ is a metric on $X)$

$$\begin{cases} \varphi_1 \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \tfrac{1}{2} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \\ \varphi_2 \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \tfrac{1}{2} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \tfrac{1}{2} \end{bmatrix} \\ \varphi_3 \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \tfrac{1}{2} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \tfrac{1}{2} \\ \tfrac{1}{2} \end{bmatrix} \end{cases}$$

$$\Phi(K) = \bigcup_{i=1}^{3} \varphi_i(K)$$

$\Phi^0(K) = K$ $\quad$ $\Phi^1(K) = \Phi(K)$ $\quad$ $\Phi^2(K) = \Phi(\Phi(K))$



$$\mathrm{Fix}(\Phi) = \bigcap_{k=0}^{\infty} \Phi^k(K)$$

(Sierpiński triangle)

# Self-Similar Programs

- Represent $\Phi$ as a logic program (self-similar program)

$$\mathrm{SP}(W) = \{\mathit{Path}(\lambda)\} \cup \{\mathit{Path}(wx) \leftarrow \mathit{Path}(x) \mid w \in W\}$$

- Bottom-up construction of the least Herbrand model of SP($W$) corresponds to computing Fix($\Phi$) effectively
- Example:

$$\mathrm{SP}(\{0,1,3\}) = \left\{ \begin{array}{l} \mathit{Path}(\lambda), \mathit{Path}(0x) \leftarrow \mathit{Path}(x), \\ \mathit{Path}(1x) \leftarrow \mathit{Path}(x), \mathit{Path}(3x) \leftarrow \mathit{Path}(x) \end{array} \right\}$$

  - Bottom-up construction:

$$\{\mathit{Path}(\lambda)\}, \{\mathit{Path}(0), \mathit{Path}(1), \mathit{Path}(3)\},$$

$$\left\{ \begin{array}{l} \mathit{Path}(0\,0), \mathit{Path}(0\,1), \mathit{Path}(0\,3), \mathit{Path}(1\,0), \mathit{Path}(1\,1) \\ \mathit{Path}(1\,3), \mathit{Path}(3\,0), \mathit{Path}(3\,1), \mathit{Path}(3\,3) \end{array} \right\}, \ldots$$

# Collage Theorem

- Hausdorff distance between a figure $K$ and a self-similar set $V$ can be bounded (Collage Theorem) [Barnsley, 93]

$$d_H(K, V) \leqslant \frac{d_H(K, \bigcup_{\varphi \in C} \varphi(K))}{1 - c}$$

(*V* is a self-similar set for $C$, $c$ is a contractivity factor of $C$)

- Any figure can be approximated (in the meaning of the Hausdorff metric) by some self-similar set arbitrarily closely [Falconer, 03]

> For any figure $K$ and $\delta > 0$, there exists a self-similar set $V$ satisfying $d_H(K, V) < \delta$

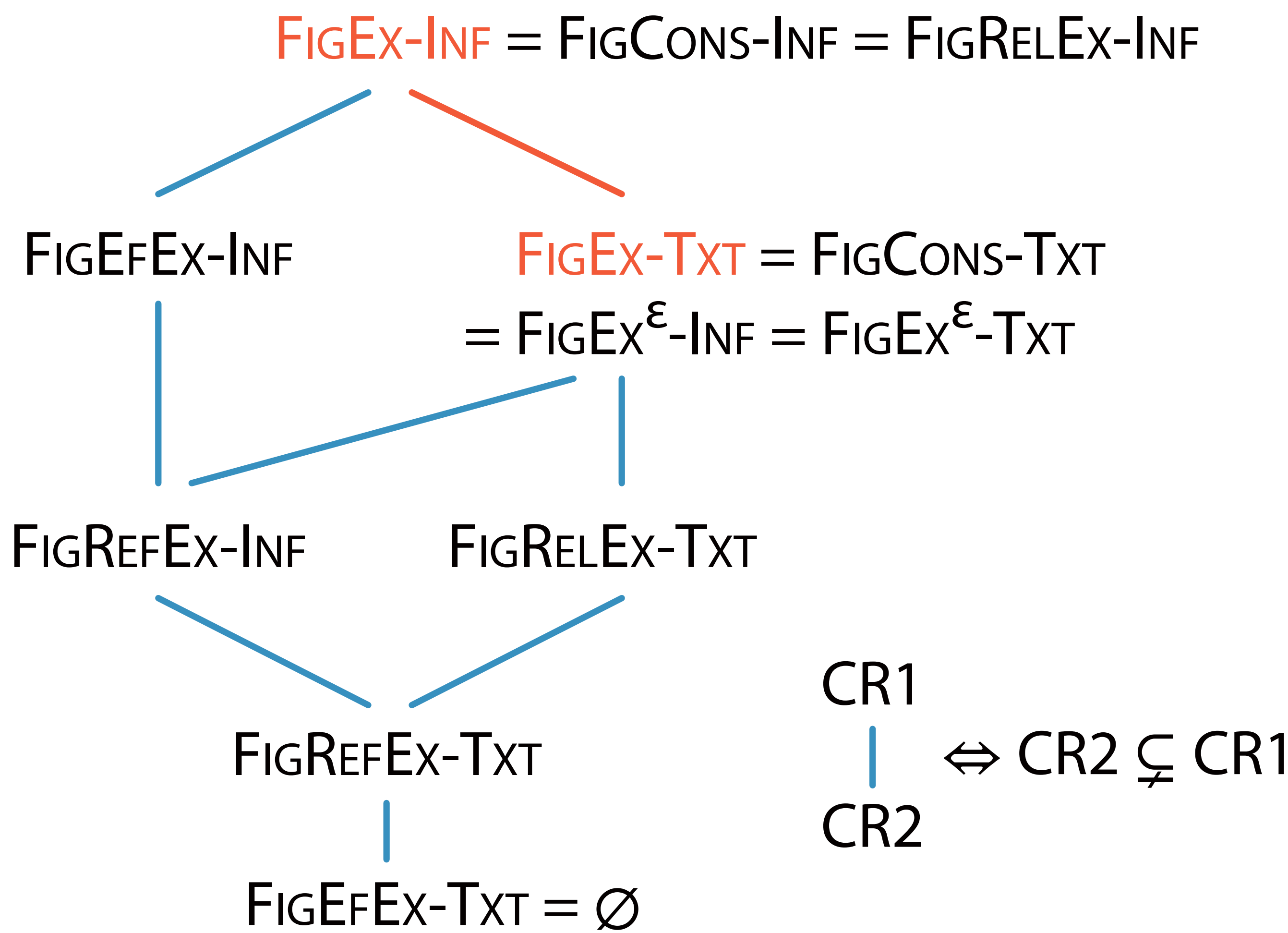
# Learning in the Limit

- Introduce a criterion corresponding to **Ex**-learning
- A learner **M** FigEx-Inf-learns (FigEx-Txt-learns) a set of figures $\mathscr{F} \subseteq \mathscr{K}^*$ $\iff$ For all $K \in \mathscr{F}$ and informants (texts) of $K$, $\mathbf{M}(\sigma_K)$ converges to a hypothesis $P$ such that $\mathbf{GE}(K, P) = 0$
  - $\mathbf{GE}(K, P)$ is a generalization error, define by $d_{\mathrm{H}}(K, \kappa(P))$
    - $d_{\mathrm{H}}$ is the Hausdorff distance
  - Hypotheses converge $\iff$ every hypothesis is same from some point
- $\mathscr{F}$ is CR-learnable if some learner **M** **CR**-learns $\mathscr{F} \subseteq \mathscr{K}^*$
  - **CR** denotes the class of **CR**-learnable sets of figures
    - $\mathscr{F}$ is **CR**-learnable $\iff$ $\mathscr{F} \in \mathbf{CR}$

# Analysis of Learnability in the Limit

- The set $\kappa(\mathscr{P}^*)$ is **FigEx-Inf**-learnable ($\mathscr{K}^*$ is not **FigEx-Inf**-learnable)
  - $\kappa(\mathscr{P}^*)$ is recursively enumerable
  - For all $P \in \mathscr{P}^*$ and $w$, whether $\rho(w) \in \mathscr{Q}(\kappa(P))$ can be decidable in finite time
  - Use the strategy of "generate and test"
- The set $\kappa(\mathscr{P}^*)$ is not **FigEx-Txt**-learnable
- The set $\kappa(\mathscr{P}_N)$ ($N$ is finite set of natural numbers) is **FigEx-Txt**-learnable
  - If a learner knows the number of contractions *a priori*, it can learn from texts

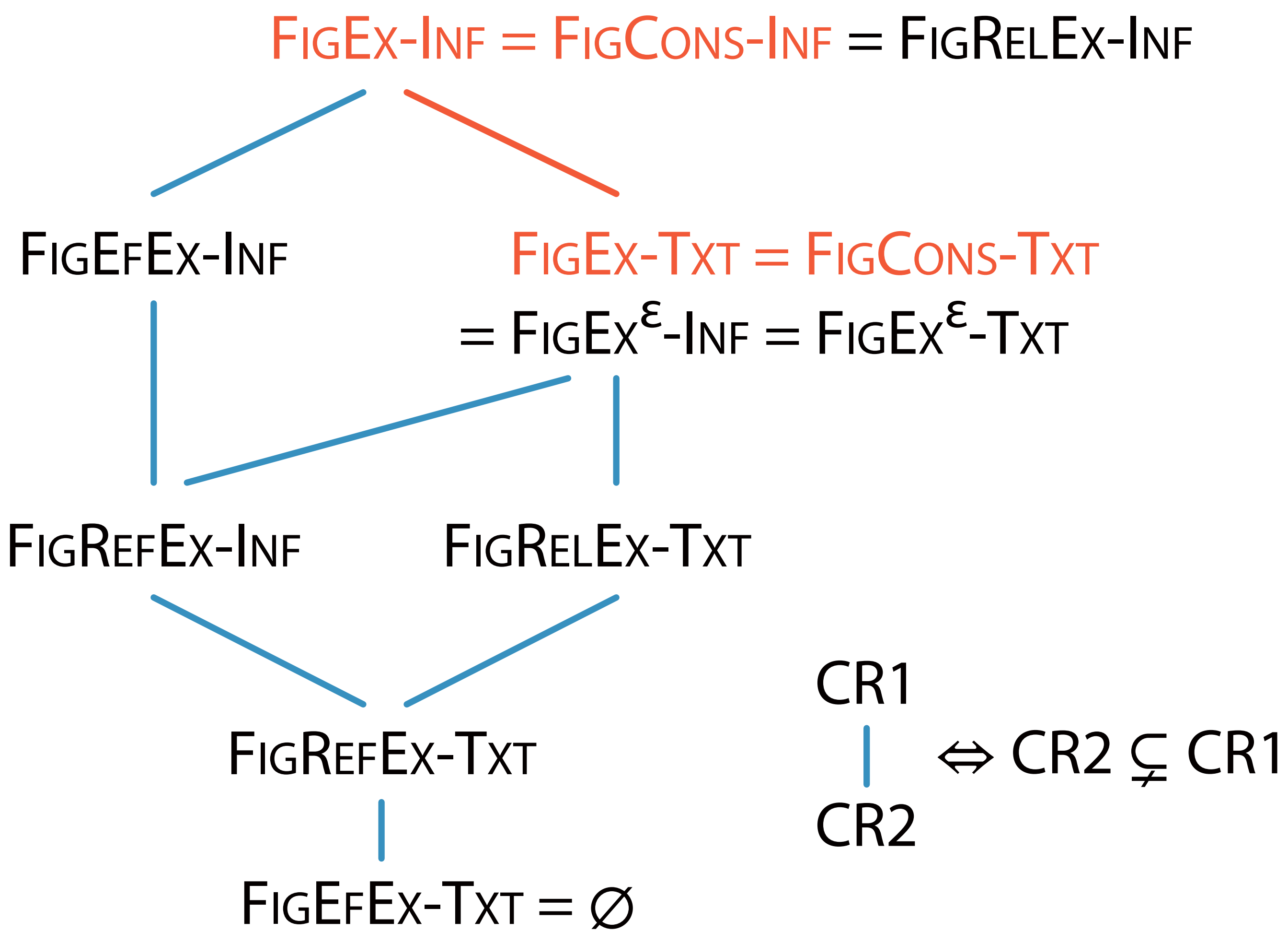
# The Hierarchy of Learnabilities



# Consistent Learning

- A learner **M** FigCons-Inf-learns (FigCons-Txt-learns) a set of figures $\mathscr{F} \subseteq \mathscr{K}^*$ $\iff$ **M** **FigEx-Inf**-learns (**FigEx-Txt**-learns) $\mathscr{F}$, and every hypothesis is consistent with received examples so far
  - $\mathscr{F}$ is **FigCons-Inf**-learnable $\Rightarrow$ $\mathscr{F}$ is **FigEx-Inf**-learnable
  - $\mathscr{F}$ is **FigCons-Txt**-learnable $\Rightarrow$ $\mathscr{F}$ is **FigEx-Txt**-learnable
- **FigEx-Inf** = **FigCons-Inf**
  - If $\mathscr{F} \in$ **FigEx-Inf**, **M** always outputs a consistent hypothesis
- **FigEx-Txt** = **FigCons-Txt**
  - If $\mathscr{F} \in$ **FigEx-Txt**, **M** always outputs a consistent hypothesis

# The Hierarchy of Learnabilities



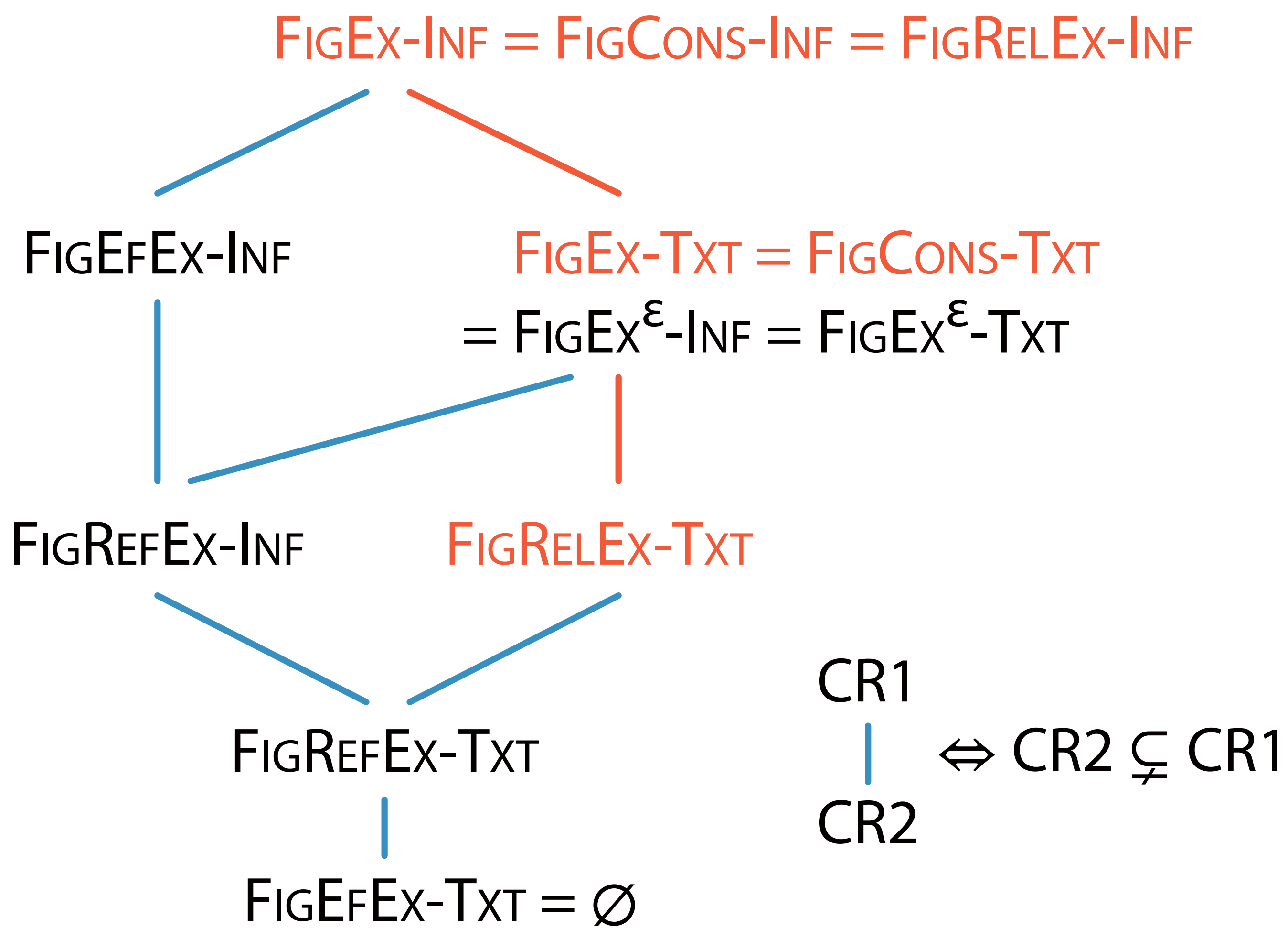
# Extension of Learning in the Limit

- In **FigEx-Inf**- (and **FigEx-Txt**-) learning, $\mathscr{F}$ is given as a concept space *a priori*
  - When a target figure $K \notin \mathscr{F}$, nothing is guaranteed
- Here we give some guarantee to such cases, where a target figure $K \notin \mathscr{F}$
  - More difficult than **FigEx-Inf**- and **FigEx-Txt**-learning
- We consider the following criteria: 1) refutable learning, 2) reliable learning, 3) effective learning, and 4) learning with generalization error bounds
  - If a target $K \notin \mathscr{F}$, 1) a learner stops, 2) hypotheses do not converge, 3) generalization errors converge to zero, and 4) converges under the error bounds

# Reliable Learning

- A learner **M** FigRelEx-Inf-learns (FigRelEx-Txt-learns) a set of figures $\mathscr{F} \subseteq \mathscr{K}^* \iff$ **M** **FigEx-Inf**-learns (**FigEx-Txt**-learns) $\mathscr{F}$, and if a target $K \in \mathscr{K}^* \setminus F$, then for all informants (texts) $\sigma_K$, $\mathbf{M}(\sigma_K)$ does not converge to any hypothesis
- FigEx-Inf = FigRelEx-Inf
  - If $K \in \mathscr{K}^* \setminus F$, then for all $P \in \mathscr{P}^*$, there exists an example that is not consistent with $P$
- $\kappa(\mathscr{P}_N)$ is **FigRelEx-Txt**-learnable only if $N = \{1\}$
  - Example: Let $N = \{2\}$ and $K = \{(0,0), (1/2, 1/2), (1, 1)\}$. Then $K \subset \kappa(\mathrm{SP}(\{0, 3\}))$, and outputs converges to this program
- In learning of languages, a class $\mathscr{L}$ is reliably inferable from texts if and only if $\mathscr{L}$ contains no infinite concept

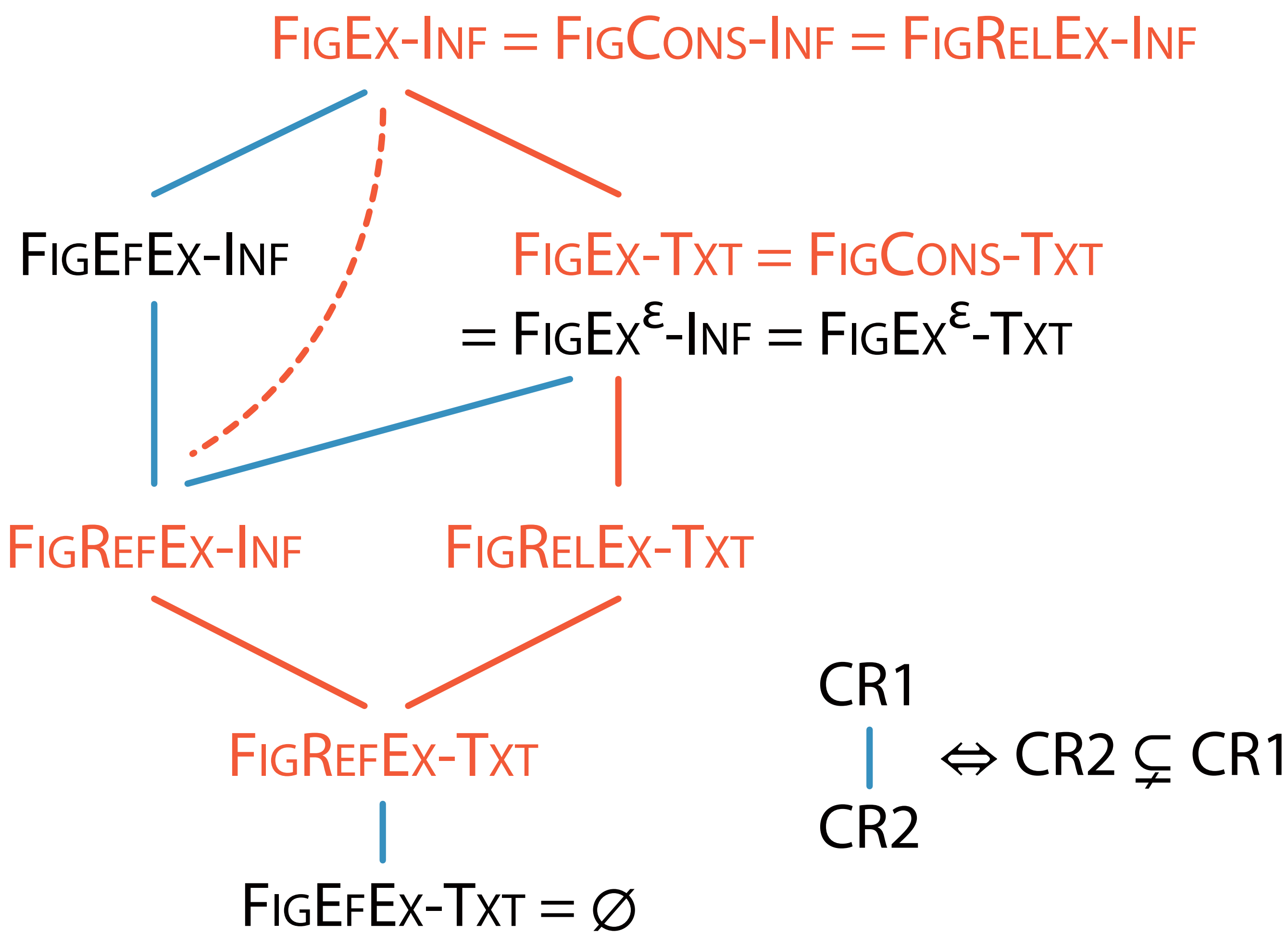
# The Hierarchy of Learnabilities



# Refutable Learning

- A learner **M** FigRefEx-Inf-learns (FigRefEx-Txt-learns) a set of figures $\mathscr{F} \subseteq \mathscr{K}^*$ $\iff$ **M** **FigEx-Inf-** (**FigEx-Txt-**) learns $\mathscr{F}$, and if a target $K \in \mathscr{K}^* \setminus F$, then for all informants (texts), **M** stops and outputs a special symbol $\perp$
- $\kappa(\mathscr{P}_m)$ ($m \in \mathbb{N}$) is not **FigRefEx-Inf**-learnable
  - **FigRelEx-Txt** $\not\subseteq$ **FigRefEx-Inf**
- **FigRefEx-Inf** $\not\subseteq$ **FigRelEx-Txt** holds
- **FigRefEx-Txt** $\subseteq$ **FigRelEx-Txt** from Definition, and trivially **FigRefEx-Txt** $\neq$ **FigRelEx-Txt**
- **FigRefEx-Txt** $\subset$ **FigRefEx-Inf** also holds
- **FigRefEx-Txt** $\neq \emptyset$

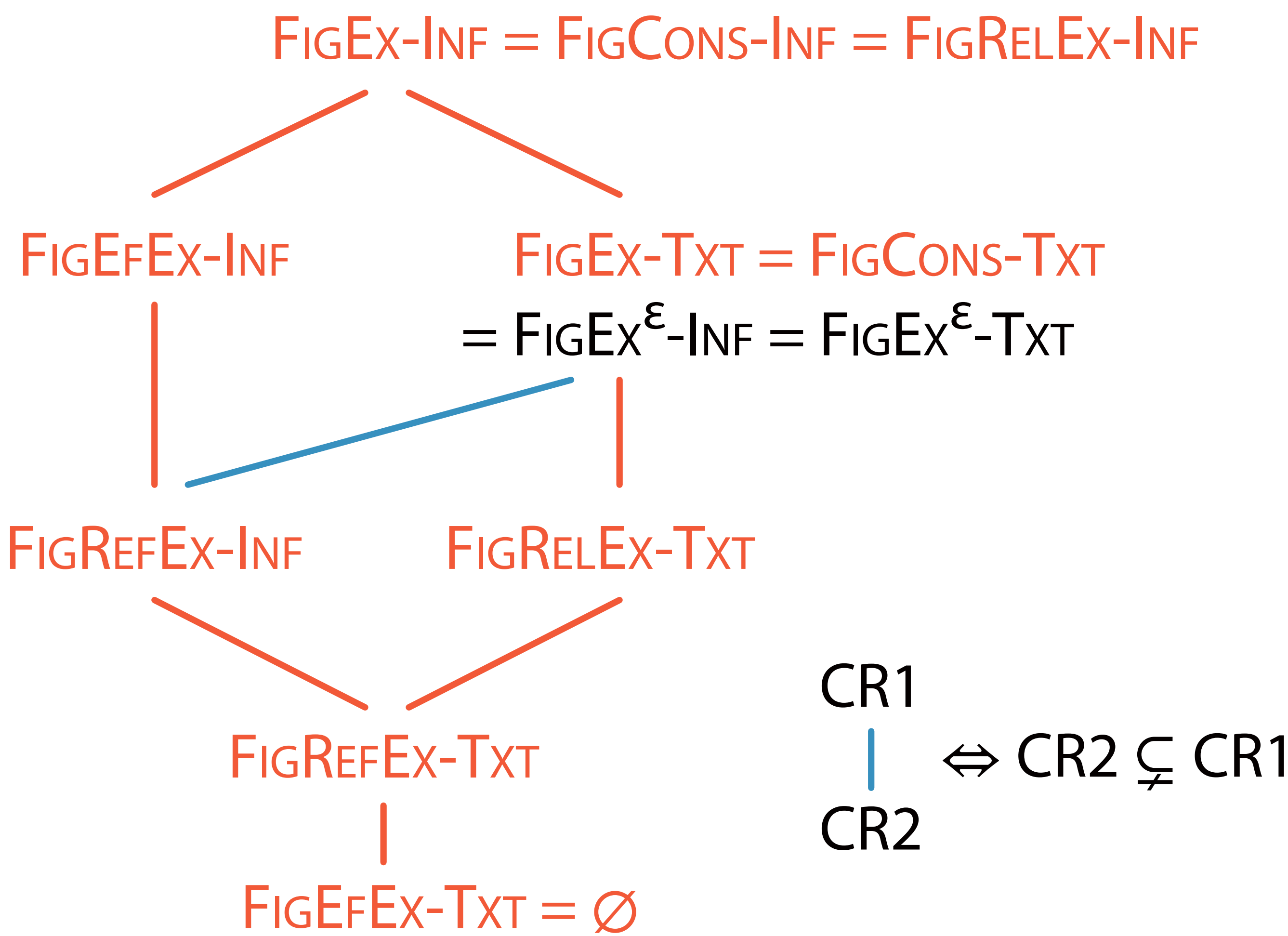
# The Hierarchy of Learnabilities



# Effective Learning

- A learner **M** FigEfEx-Inf-learns (FigEfEx-Txt-learns) a set of figures $\mathscr{F} \subseteq \mathscr{K}^* \iff$ **M** FigEx-Inf- (FigEx-Txt-) learns $\mathscr{F}$, and if a target $K \in \mathscr{K}^* \setminus F$, then for all informants (texts), there exists some monotone function $\varepsilon: \mathbb{N} \to \mathbb{R}^+$, where $\lim_{i\to\infty} \varepsilon(i) = 0$ and $\mathbf{GE}(K, \mathbf{M}(\sigma_K)(i)) \leq \varepsilon(i)$
- $\kappa(\mathscr{P})$ is FigEfEx-Inf-learnable, and $\kappa(\mathscr{P}_m)$ ($m \in \mathbb{N}$) is not FigEfEx-Inf-learnable
  - FigEfEx-Inf $\not\subseteq$ FigEx-Txt and FigEx-Txt $\not\subseteq$ FigEfEx-Inf
- FigRefEx-Inf $\subseteq$ FigEfEx-Inf from Definition, and trivially FigRefEx-Inf $\neq$ FigEfEx-Inf
- FigEfEx-Txt $= \varnothing$
  - FigEfEx-Txt $\subset$ FigRefEx-Txt (different from learning from informants)

# The Hierarchy of Learnabilities



# Approximative Learning

- A learner **M** FIGEX$^{\varepsilon}$-INF-learns (FIGEX$^{\varepsilon}$-TXT-learns) a set of figures $\mathscr{F} \subseteq \mathscr{K}^* \iff$ For all $K \in \mathscr{K}^*$ and informants (texts) $\sigma_K$, $\mathbf{M}(\sigma_K)$ converges to $P$ such that $\mathbf{GE}(K, P) = 0$ if $K \in \mathscr{F}$, and to $Q$ such that $\mathbf{GE}(K, Q) \leq \varepsilon$ if $K \in \mathscr{K}^* \setminus F$
- For all $\varepsilon \in \mathbb{R}^+$, FIGEX$^{\varepsilon}$-TXT $=$ FIGEX-TXT
- For all $\varepsilon \in \mathbb{R}^+$, FIGEX$^{\varepsilon}$-TXT $=$ FIGEX$^{\varepsilon}$-INF
  - Since if $\mathscr{F} \in$ **FIGEX$^{\varepsilon}$-INF**, then $F \subseteq \mathscr{P}_{\leq k}$
- **FIGREFEX-INF** $\subset$ **FIGEX$^{\varepsilon}$-INF**

# The Hierarchy of Learnabilities