What is a pattern? An introduction to a theory of grounded computation

Hiroshi Ishikawa

Nagoya City University

October 20, 2008. At INRIA Sophia Antipolis

Motivation

Example: Binary Restoration

original noise added Y_v restored X_v

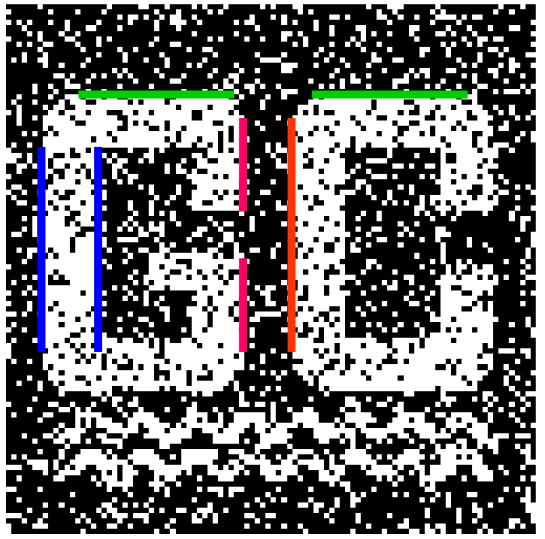
Greig, Porteous, and Seheult '89

$$E(X) = \sum_{v \in V} \lambda \, | \, Y_v - X_v \, | + \sum_{(u,v) \in E} | \, X_u - X_v \, |$$

- This is the optimal solution, given the smoothing prior.
- But it does not look perfect.

Higher-order Relationships

- Boundary smoothness
- Linearity
- Collinearity
- Parallelism



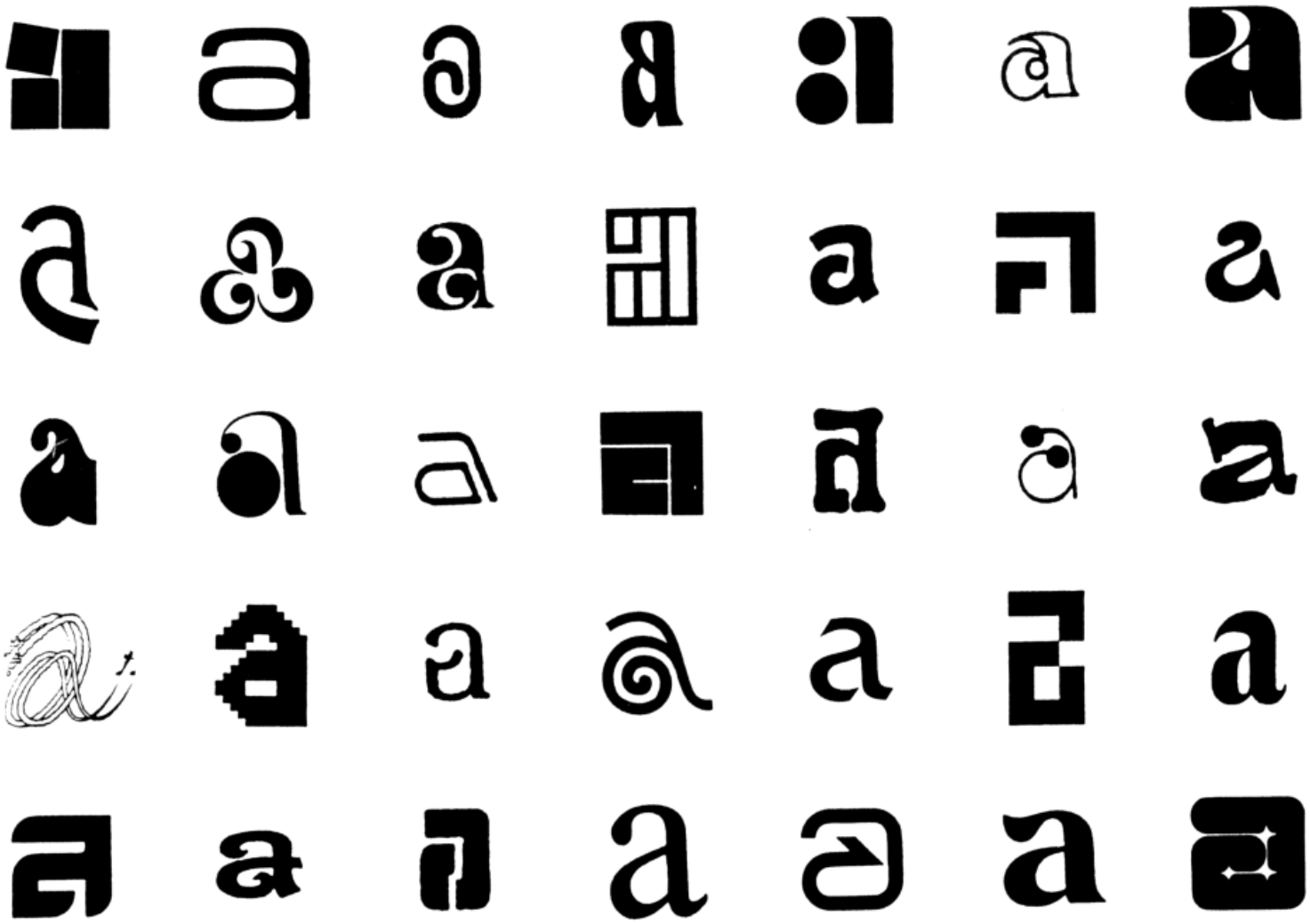
Higher-order Relationships

- Boundary smoothness
- Linearity
- Collinearity
- Parallelism
- Larger structure
- Regular structure
- Higher-level knowledge

Can we state the problem?

- We see “patterns” in these.
- How can we state the problem precisely?
- What space do these things live in?

What is a pattern?

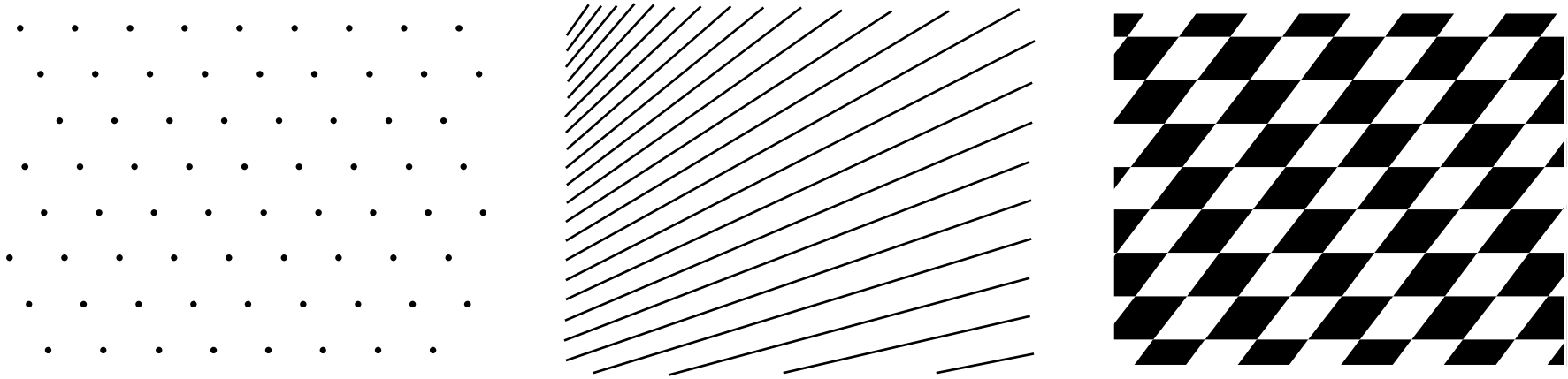


From "Fluid Concepts and Creative Analogies: Computer Models of the Fundamental Mechanisms of Thought" by Douglas R. Hofstadter and the Fluid Analogies Research Group, Basic Books, 1995. ISBN: 0465051545.

The Problem

What is a pattern?

- Define patterns in these



From dictionaries:

pat·tern

- a combination of qualities, acts, tendencies, etc., forming a consistent or characteristic arrangement: “the behavior patterns of teenagers.”
- a model or original used as an archetype.
- a perceptual structure; “a visual pattern must include not only objects but the spaces between them”

[There are many more (less abstract) meanings.

Only pertinent ones are shown here.]

In the case of strings

These strings “have patterns”

0101010101010101010101010101010101...

0101101110111101111101111111101111111...

1100100100001111111011010101000100 0...

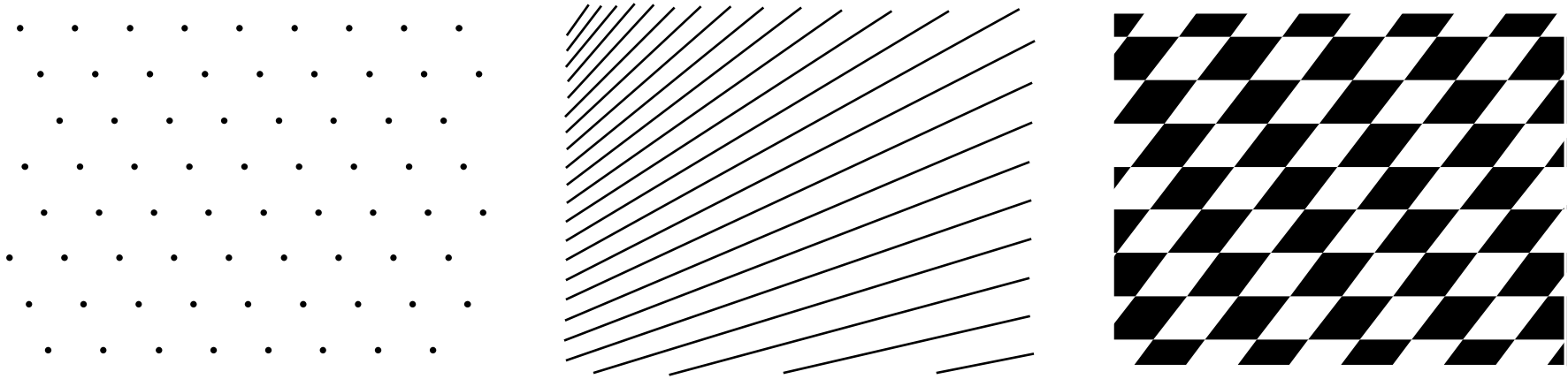
- We can say a string with much less information than its length have pattern
- That is, if there is a much shorter program to print out the string

Two descriptions

- To describe a string by another string, there are two ways:
  - Describe the string by itself (*dense*)
  - By a program that prints out the string (*sparse*)
- The latter can exploit any pattern in the string
- If the string has a pattern, the latter description can be much shorter.
- The length of the shortest such description is called its Kolmogorov Complexity
- Pattern is computation

What about objects other than strings?

Infinite set but finite information

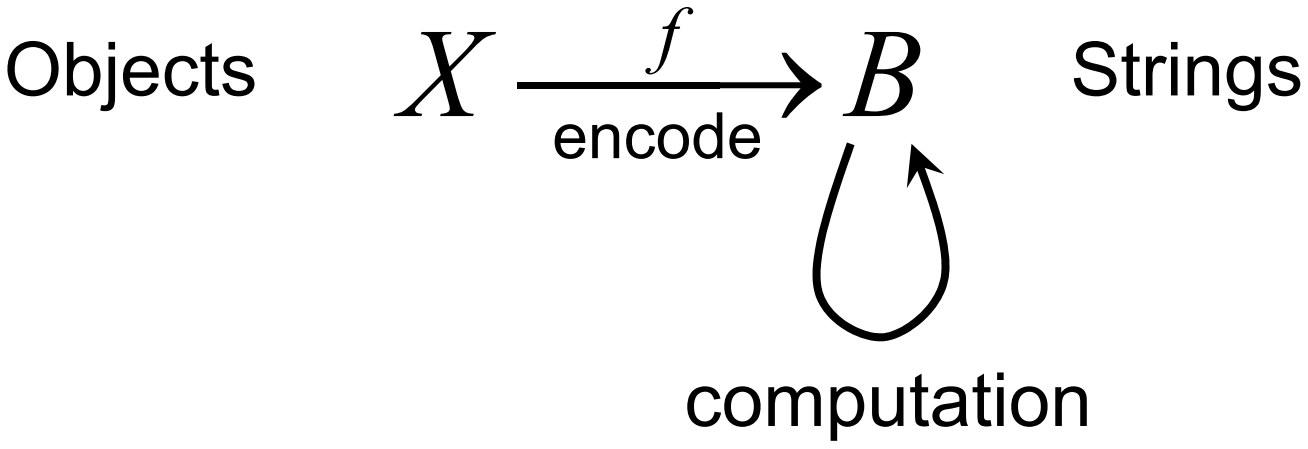


- Can we define a measure of information in these?

The Thesis of Computer Science

"Information can be encoded"

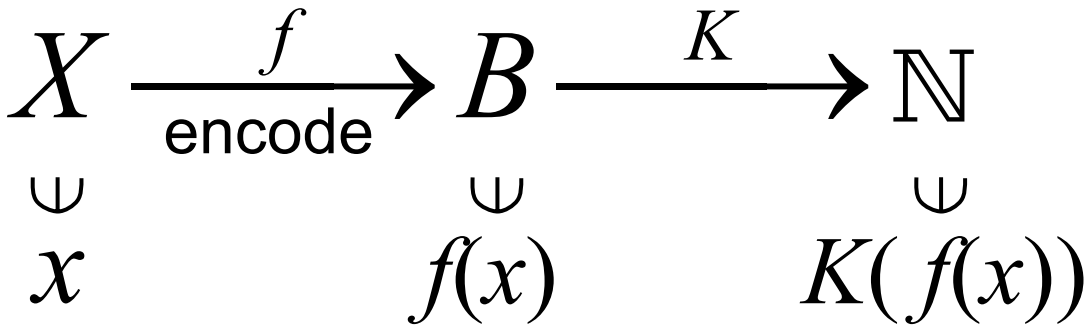
- Computation is only defined in the world of symbols and strings
- Everything else can be encoded into strings
  - Ex. Real numbers → binary expansion



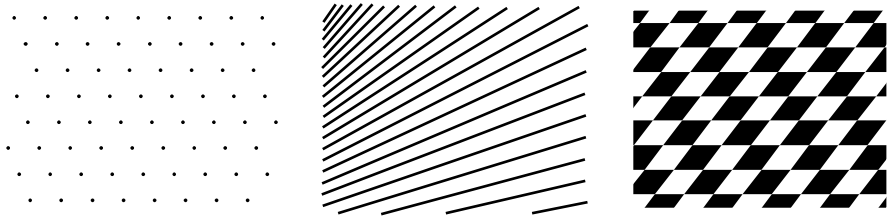
Encoding the Universe

To define an information measure in general objects:

- Fix an encoding (an enumeration) of the objects
- Define the information in an object x by the Kolmogorov Complexity $K(f(x))$ of its code string



Example: Shapes on Euclidean Plane



Encode:

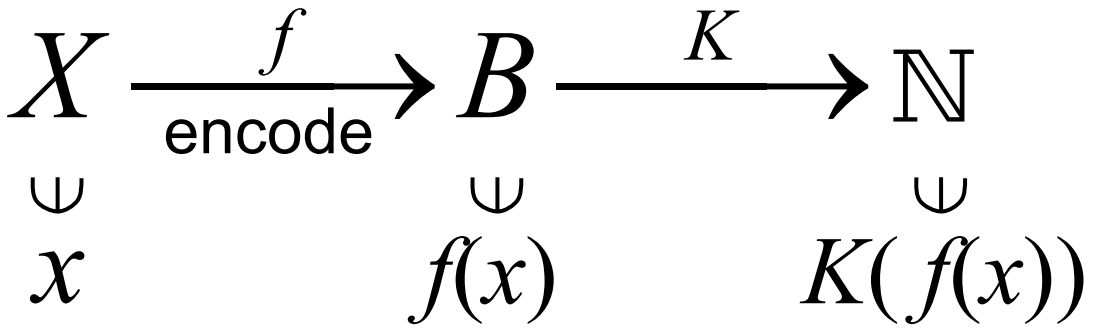
- Real number: binary expansion
- Point: a pair of reals
- Line: two points
- Circle: a point and a real
- …..

$$\begin{array}{ccccc} X & \xrightarrow[\text{encode}]{f} & B & \xrightarrow{K} & \mathbb{N} \\ \cup & & \cup & & \cup \\ x & & f(x) & & K(f(x)) \end{array}$$

Example: Shapes on Euclidean Plane

But then:

- How far must we go?
  - We can never exhaust all possible shapes
  - This is only one example
- We are presupposing what the patterns are
- The encoding f is doing the actual computation



Example: Shapes on Euclidean Plane

So let's encode only the points

- Point → 2 reals → alternate binary expansion

  $(0.x_0x_1x_2x_3..., 0.y_0y_1y_2y_3...) \rightarrow x_0y_0x_1y_1x_2y_2x_3y_3....$

- Finite number of points → rotate among points

  $a \rightarrow a_0a_1a_2a_3..., \quad b \rightarrow b_0b_1b_2b_3..., \quad c \rightarrow c_0c_1c_2c_3...,$

  $\{a,b,c\} \rightarrow a_0b_0c_0a_1b_1c_1a_2b_2c_2a_3b_3c_3...$

- Countable number of points → rotate among increasingly many points

  $\{a,b,c,d,...\} \rightarrow a_0a_1b_0a_2b_1c_0a_3b_2c_1d_0...$

- We can only encode a countable subset

Example: Shapes on Euclidean Plane

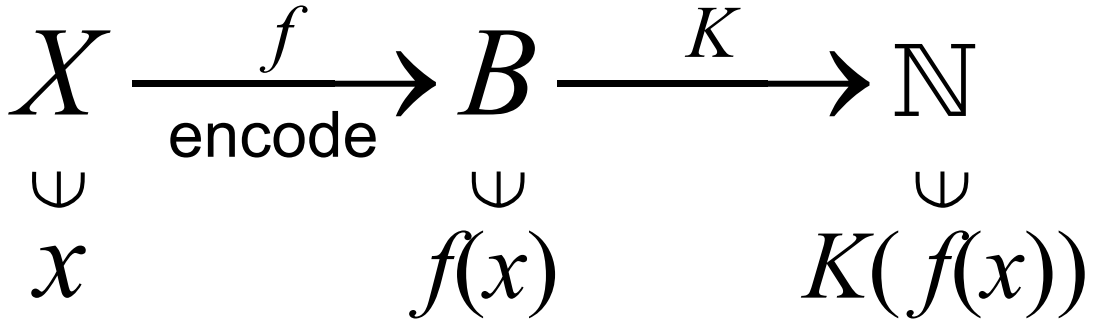
So let's only consider countable sets.

1. Require the coordinates of the points to be rational
   - Then any set of such points are countable
   - But many sets don't have many rational points

     example: the line $y = \sqrt{2}x$
2. Only consider any countable sets
   - Even a single point can have an infinite information

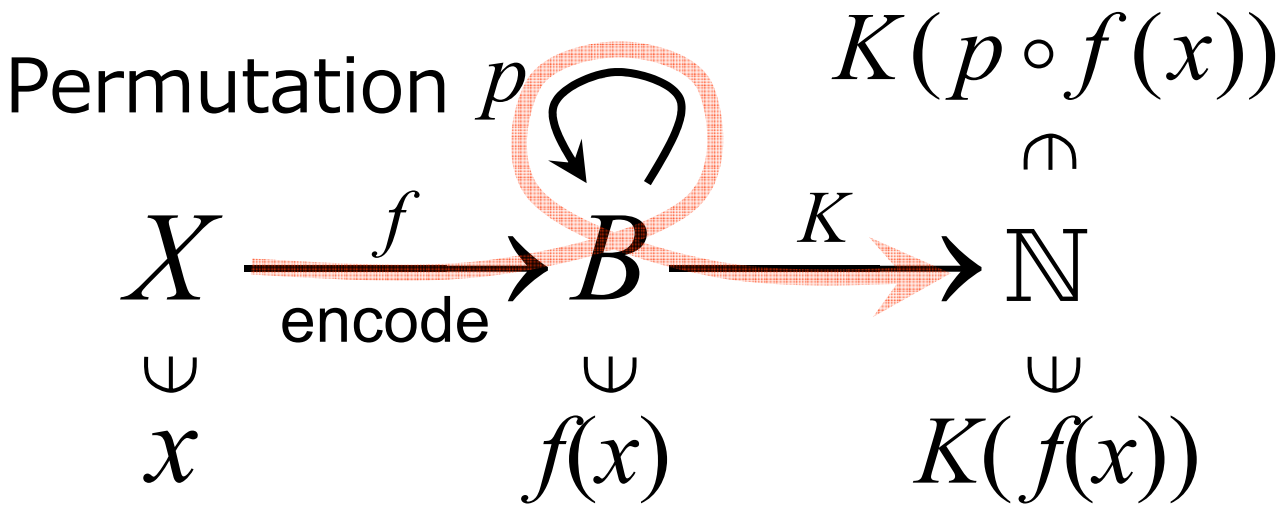
Fundamental Problems

- Hard to define general enough encoding f
- We need to be able to define the information amount in specific parts *a priori*
  - Ex: Points to have the same amount of information



The Arbitrariness Trap

- In CS, the encoding f is just said to "exist".
- The arbitrariness of f makes the information measure arbitrary, too
- So Computer Science does have to think about encoding seriously



A Solution

Remember the two descriptions

- To describe a string by another string, there are two ways:
  - Describe the string by itself (dense)
  - By a program that prints out the string (sparse)
- Similarly, we will represent an object in two ways:
  - Represent it by a subset (dense representation)

    Ground representation
  - By a new representation (sparse representation) that produces the dense representation

    Representation by diagrams and cross sections

Ground Representation

- Something like bitmap, representing raw data
- Abstract a bit → subsets
  - Subsets of Euclidean plane
  - Image: a subset of $\mathbb{R}^2 \times Color$

    i.e., image function $I(p)$ on $D \subset \mathbb{R}^2$

    $\Leftrightarrow \ \{(p, I(p)) \mid p \in D\} \subset \mathbb{R}^2 \times Color$
  - Objects in 3D space (at a level of abstraction) :

    $\text{Car} \subset \mathbb{R}^3 \times Material$

    $Material = \{\text{iron, rubber, glass, plastic, } \ldots\}$

Ground Representation

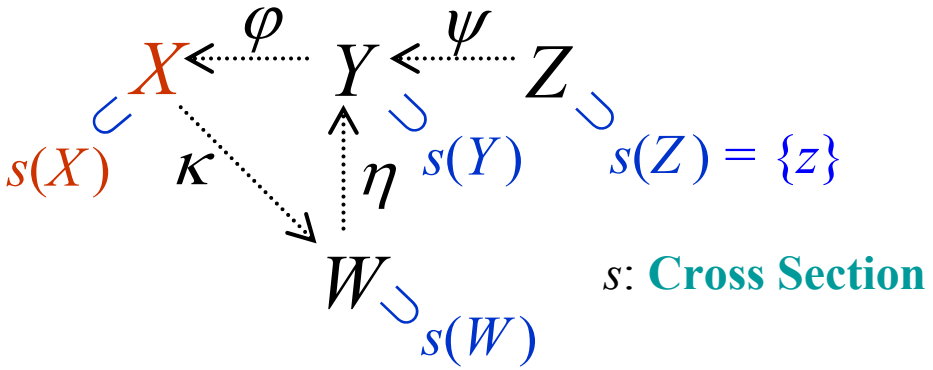
- Raw signal
- *A priori*
- Almost everything can be easily represented
- Does not reflect the structure in the object, even if there is any
- Most that can be represented is random noise

Representation

- Specify sets and *power maps* between them (*diagram*)

Ex.

$$\varphi : \mathcal{P}(Y) \to \mathcal{P}(X)$$
$$\psi : \mathcal{P}(Z) \to \mathcal{P}(Y)$$
$$\eta : \mathcal{P}(W) \to \mathcal{P}(Y)$$
$$\kappa : \mathcal{P}(X) \to \mathcal{P}(W)$$



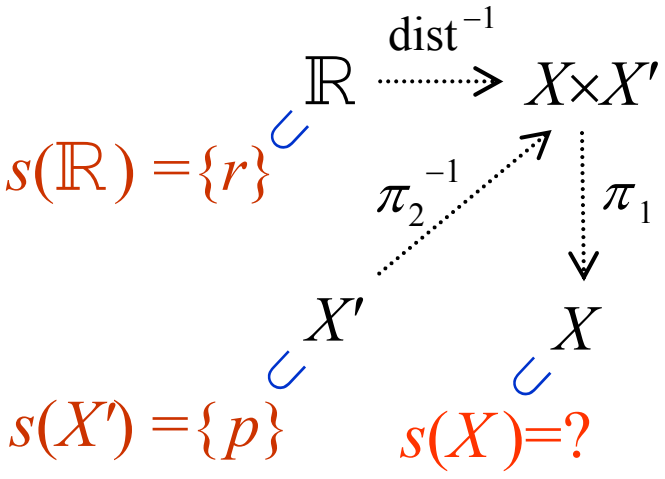
- Consider an assignment s to each set of its subset that satisfies

$$s(S) = \bigcap_{\mu \in \mathrm{in}(S)} \mu(s(\mathrm{dm}(\mu)))$$

$$s(X) = \varphi(s(Y))$$
$$s(Y) = \psi(s(Z)) \cap \eta(s(W))$$
$$s(W) = \kappa(s(X))$$

- Specify some of the s(.)
- Designate one set (say X) whose assigned subset ($s(X)$) is the dense representation of what is represented.

The Simplest Examples



X, X' : Euclidean plane

dist: $X \times X \to \mathbb{R}$ distance function

π_1, π_2: Projection

$$s(S) = \bigcap_{\mu \in \mathrm{in}(S)} \mu(s(\mathrm{dm}(\mu)))$$

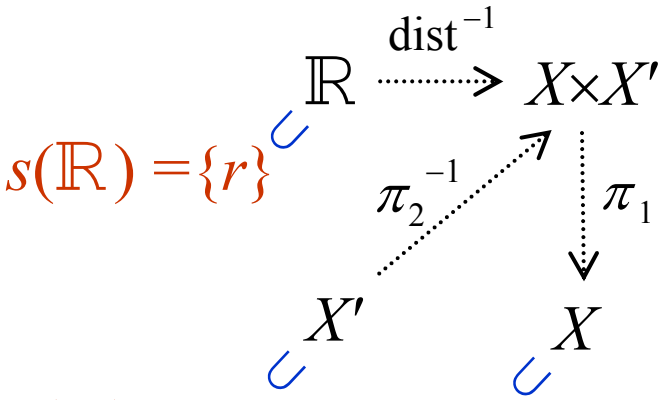
$f : A \to B$

$\to$

$f : \mathcal{P}(A) \to \mathcal{P}(B) \qquad A \supset S \mapsto \{f(x) \mid x \in S\} \subset B$

$f^{-1} : \mathcal{P}(B) \to \mathcal{P}(A) \qquad B \supset S \mapsto \{x \in A \mid f(x) \in S\} \subset A$

The Simplest Examples

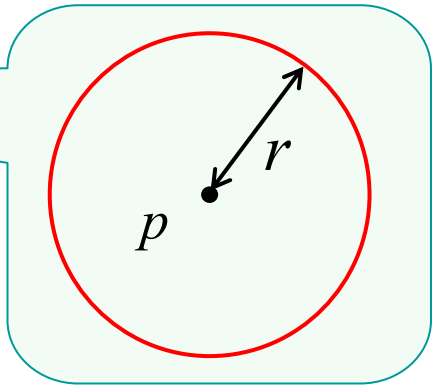


X, X' : Euclidean plane

dist: $X \times X \to \mathbb{R}$ distance function

π_1, π_2: Projection

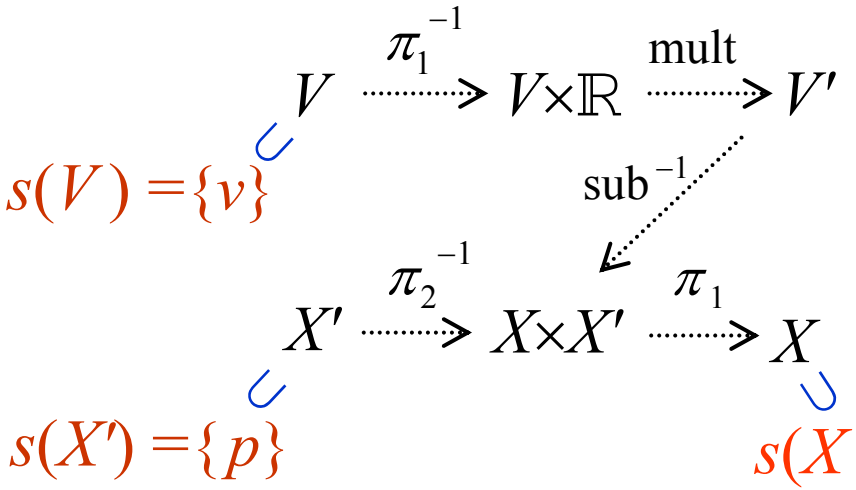
$$s(S) = \bigcap_{\mu \in \mathrm{in}(S)} \mu(s(\mathrm{dm}(\mu)))$$



$$s(X \times X') = \mathrm{dist}^{-1}(s(\mathbb{R})) \cap \pi_2^{-1}(s(X'))$$
$$= \{(x, y) \mid \mathrm{dist}(x, y) \in s(\mathbb{R}), \pi_2(x, y) = p\}$$
$$= \{(x, p) \mid \mathrm{dist}(x, p) = r\}$$

$$s(X) = \pi_1(s(X \times X'))$$
$$= \{x \mid \mathrm{dist}(x, p) = r\}$$

The Simplest Examples



X, X' : Euclidean plane

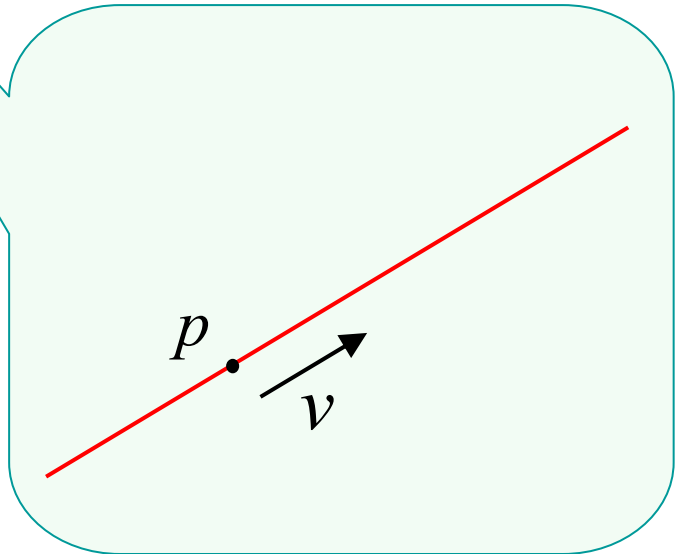
V, V' : 2D vector space

mult: $V\times\mathbb{R}\to V \quad (v,c)\mapsto cv$

sub: $X\times X\to V \quad (x,y)\mapsto x-y$

π_1, π_2: Projection

$$s(V\times\mathbb{R}) = \{(v,c)\mid c\in\mathbb{R}\}$$
$$s(V') = \{cv\mid c\in\mathbb{R}\}$$
$$s(X\times X') = \{(x,p)\mid x-p\in s(V')\}$$
$$s(X) = \{x\mid x=p+cv, c\in\mathbb{R}\}$$

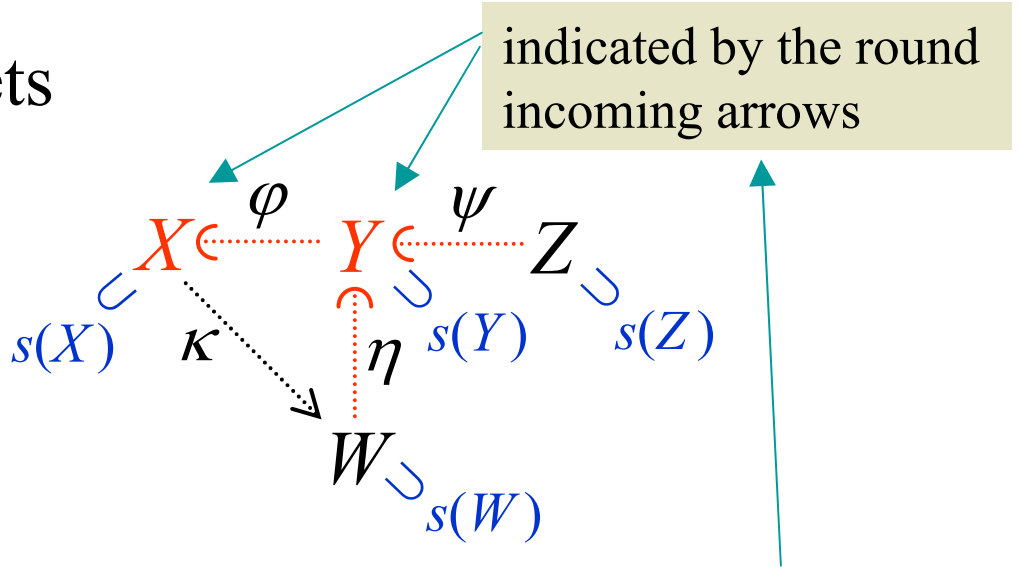


A little extension

- Choose some of the sets

  Ex.

$$\varphi : \mathcal{P}(Y) \to \mathcal{P}(X)$$
$$\psi : \mathcal{P}(Z) \to \mathcal{P}(Y)$$
$$\eta : \mathcal{P}(W) \to \mathcal{P}(Y)$$
$$\kappa : \mathcal{P}(X) \to \mathcal{P}(W)$$



- Assume the assignment s satisfies for the chosen sets

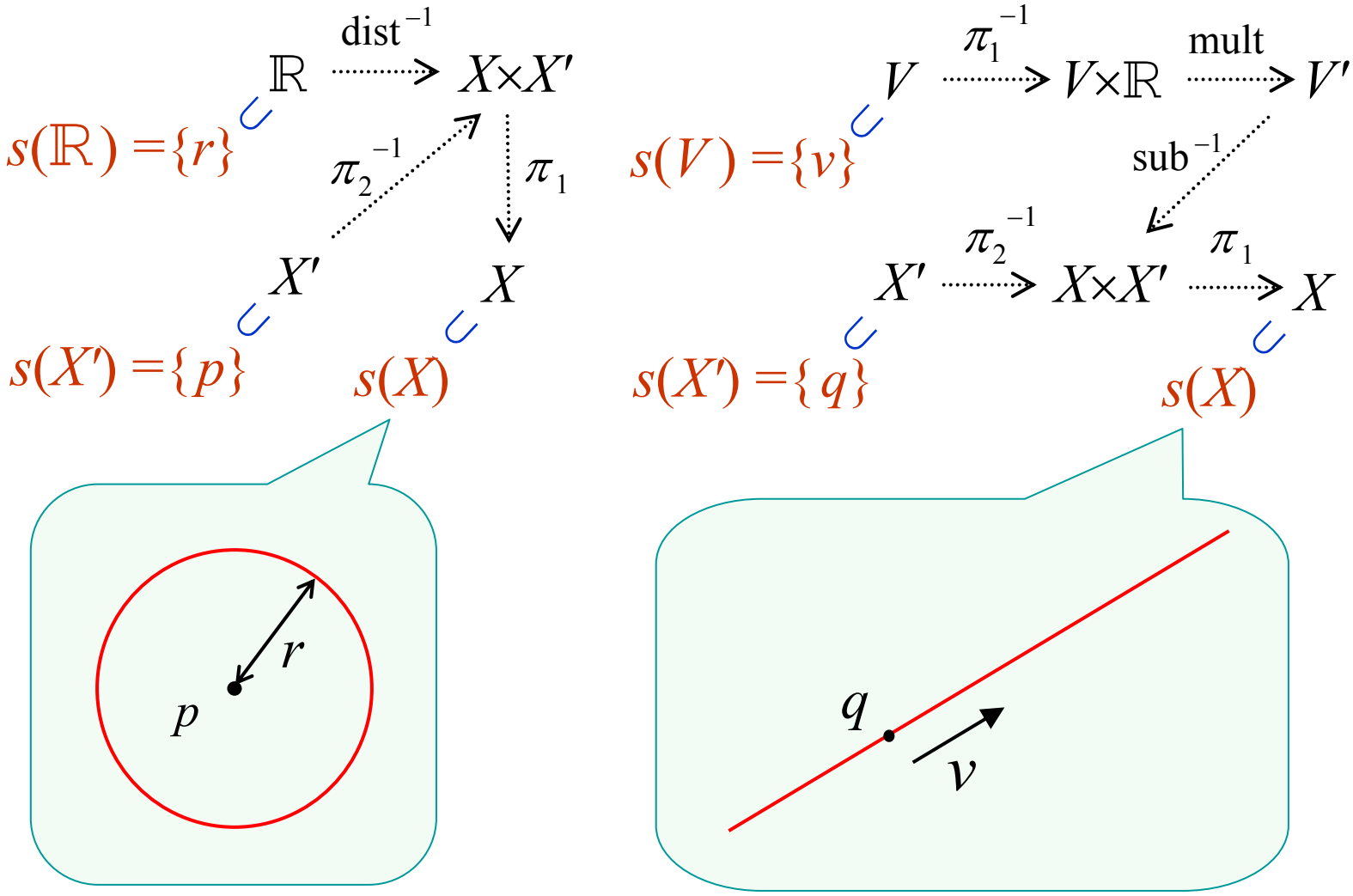
$$s(S) = \bigcup_{\mu \in \mathrm{in}(S)} \mu(s(\mathrm{dm}(\mu)))$$

$$s(X) = \varphi(s(Y))$$
$$s(Y) = \psi(s(Z)) \cup \eta(s(W))$$
$$s(W) = \kappa(s(X))$$

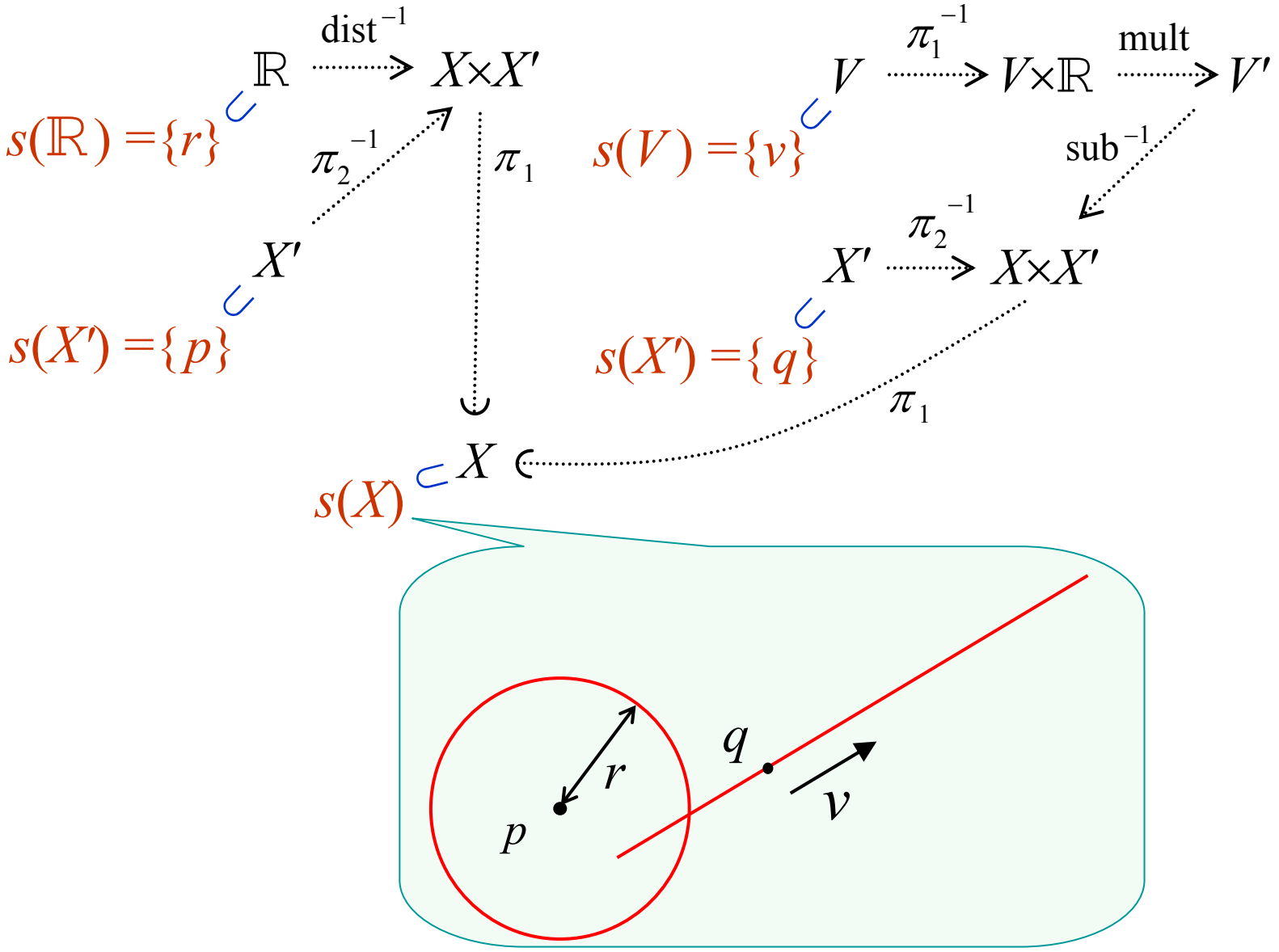
instead of

$$s(S) = \bigcap_{\mu \in \mathrm{in}(S)} \mu(s(\mathrm{dm}(\mu)))$$

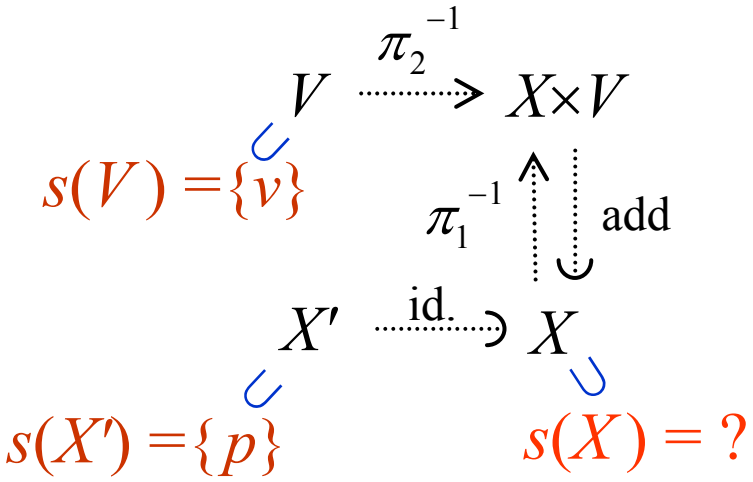
Combination



Combination



Recursive definition



X, X' : Euclidean plane

V: 2D vector space

add: $X \times V \to X \quad (x, v) \mapsto x + v$

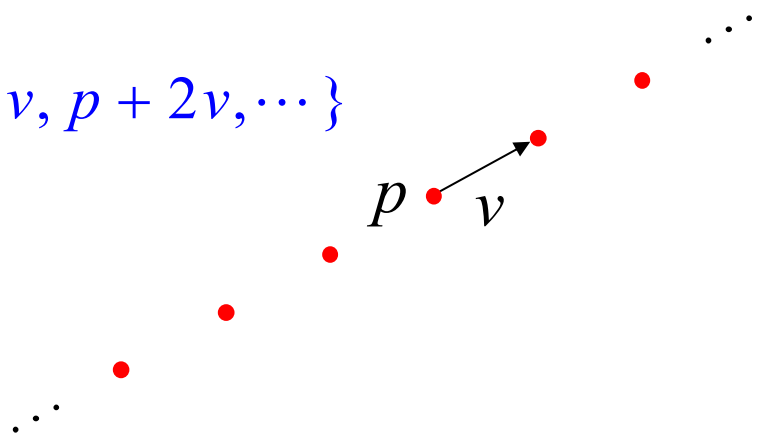
π_1, π_2: Projection

$$s(X \times V) = \{(x, v) \mid x \in s(X)\}$$

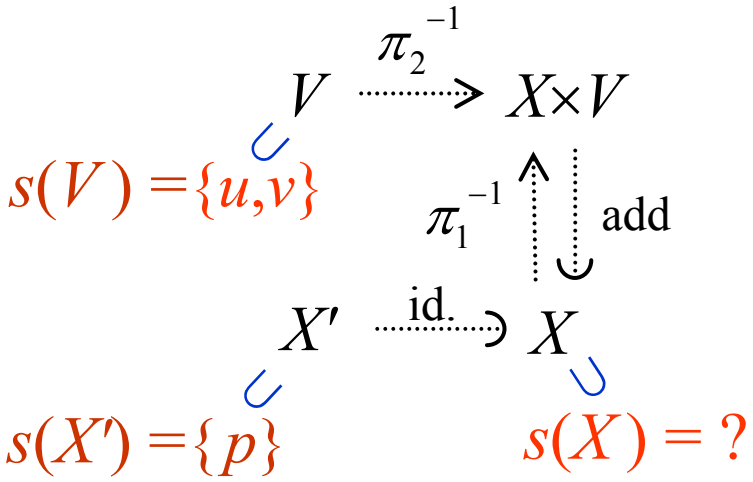
$$s(X) = s(X') \cup \text{add}(\, s(X \times V))$$

$$= \{p\} \cup \{x + v \mid x \in s(X)\}$$

$$\supset \{\cdots, p - 2v, p - v, p, p + v, p + 2v, \cdots\}$$



Recursive definition

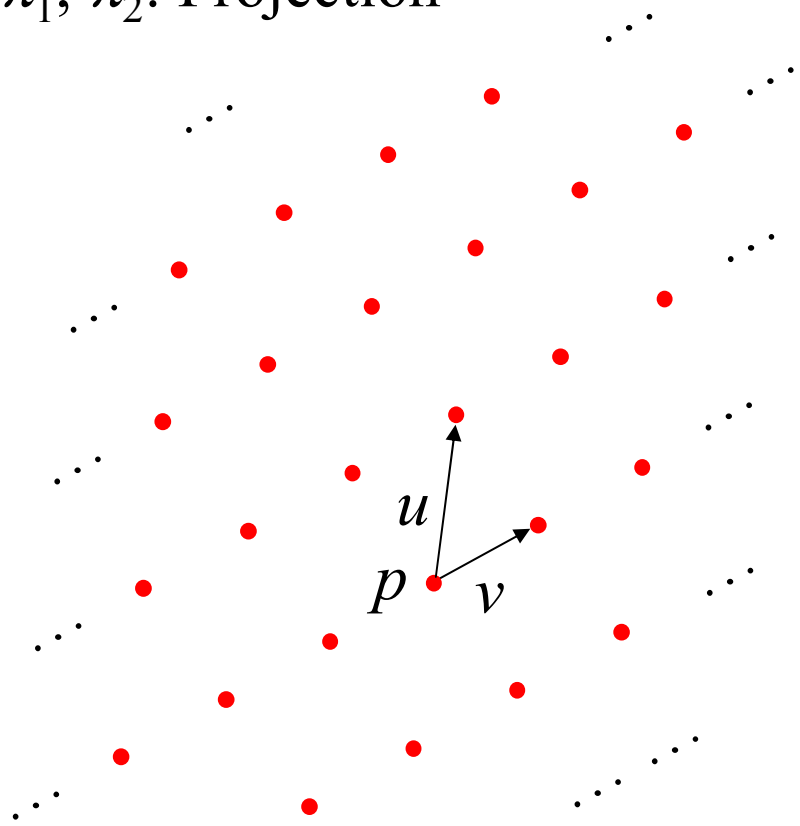


X, X' : Euclidean plane

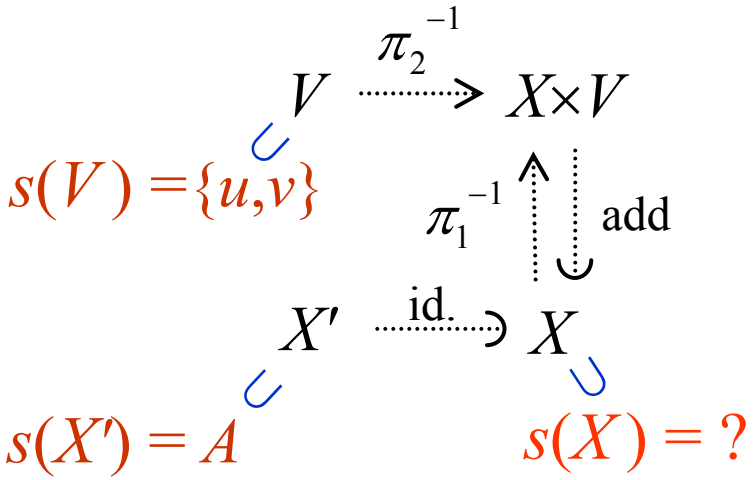
V: 2D vector space

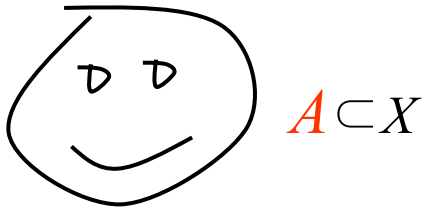
add: $X \times V \to X$ $\quad (x, v) \mapsto x + v$

π_1, π_2: Projection



Hierarchical definition





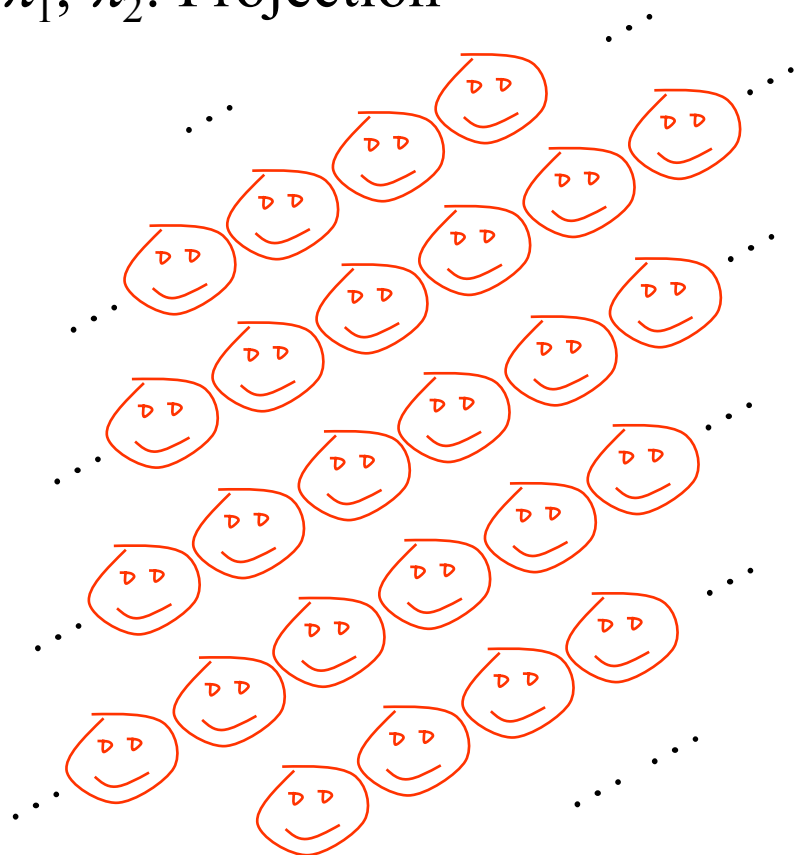
$A \subset X$

X, X' : Euclidean plane

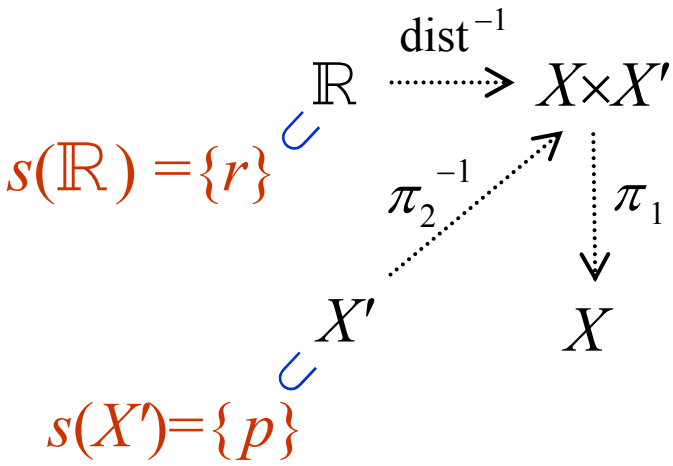
V: 2D vector space

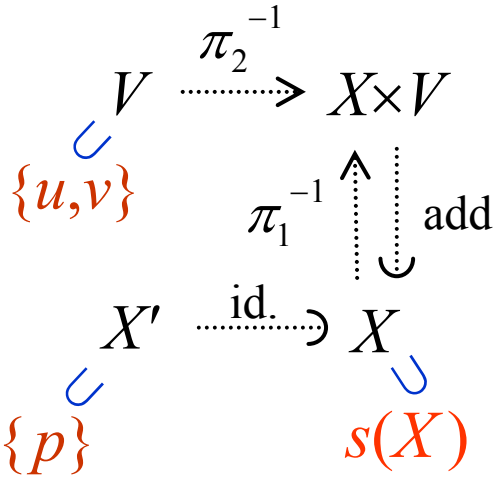
add: $X \times V \to X$ $\quad (x, v) \mapsto x + v$

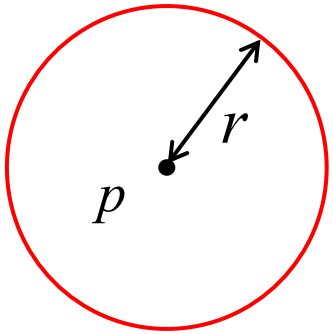
π_1, π_2: Projection

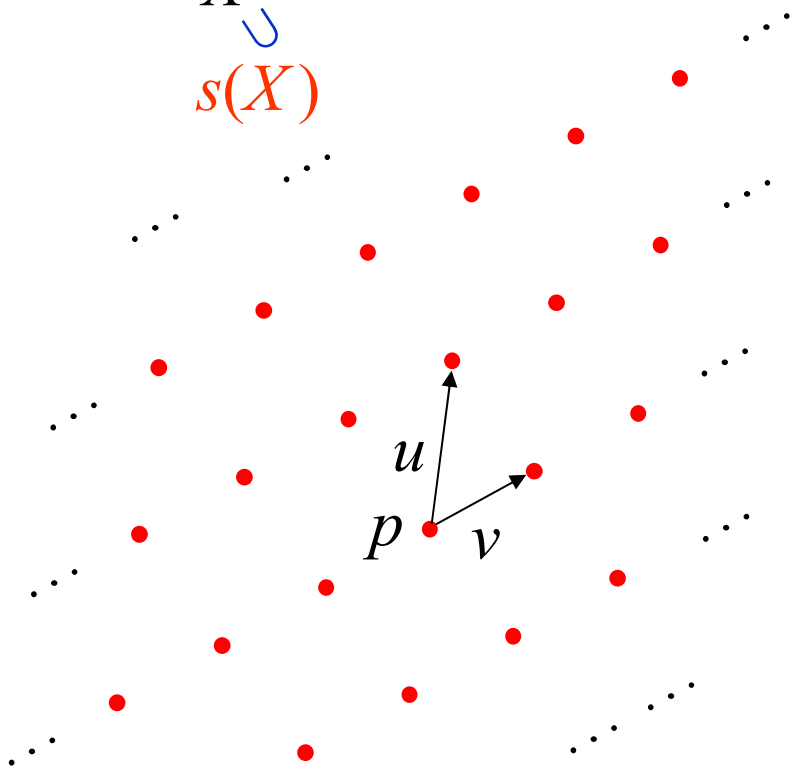


Hierarchical definition

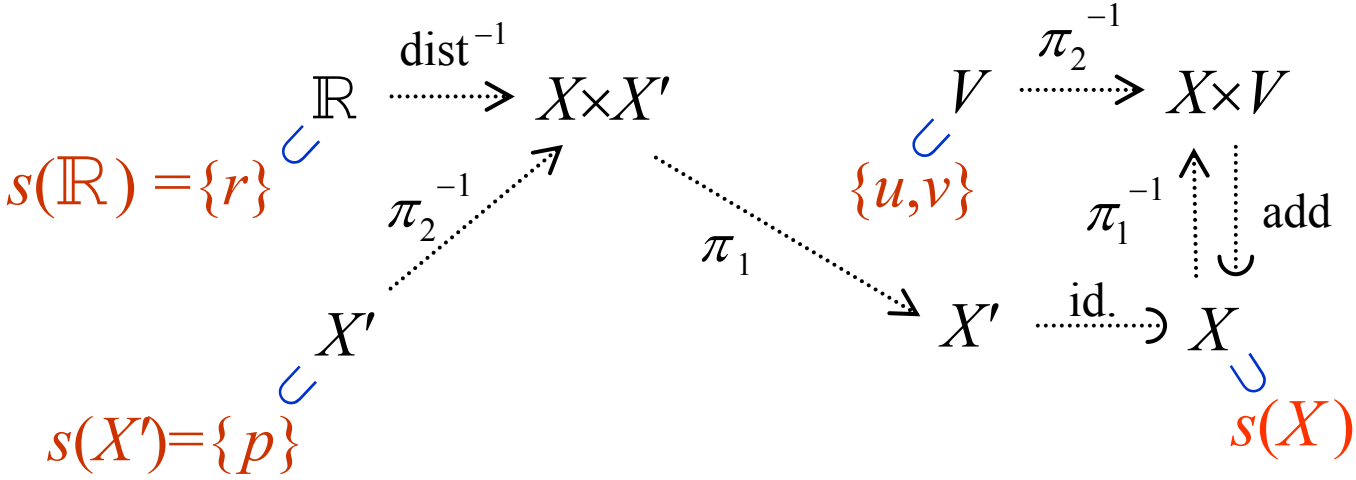


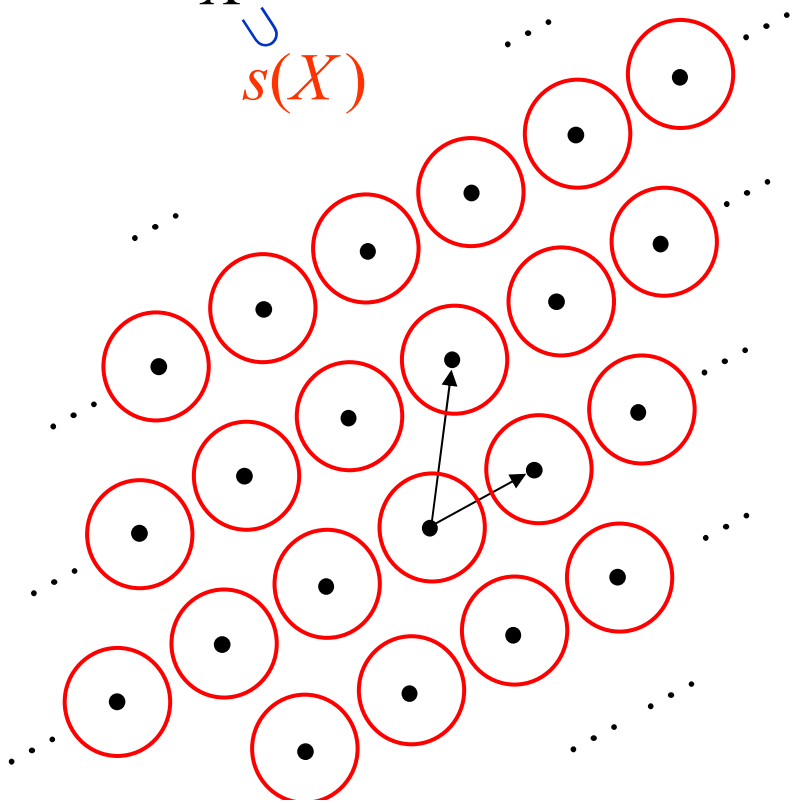






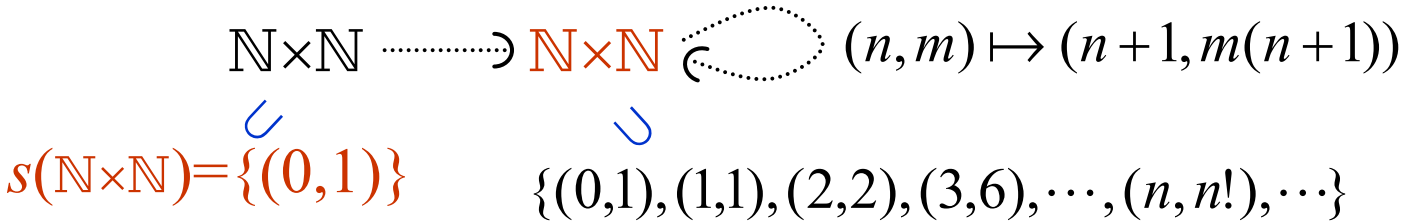
Hierarchical definition



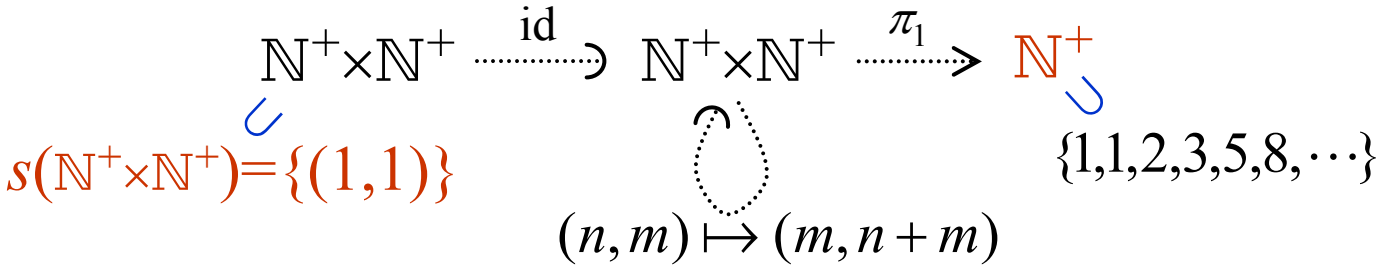


Computation

- Ex: Factorials



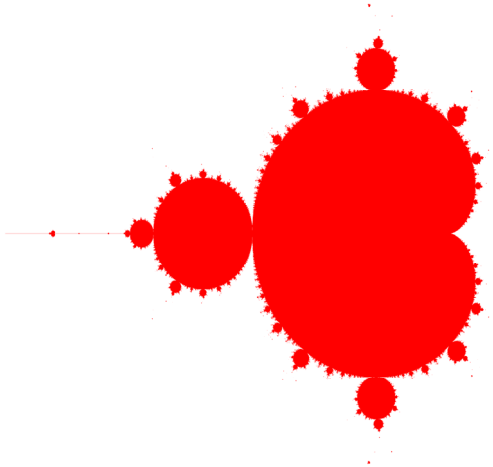
- Ex: Fibonacci numbers

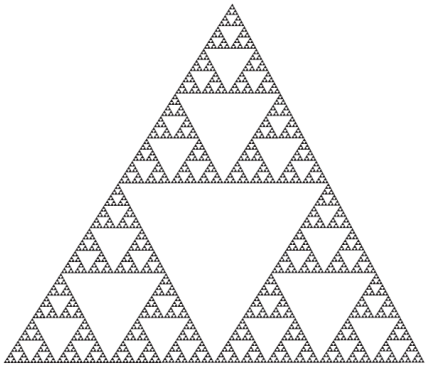


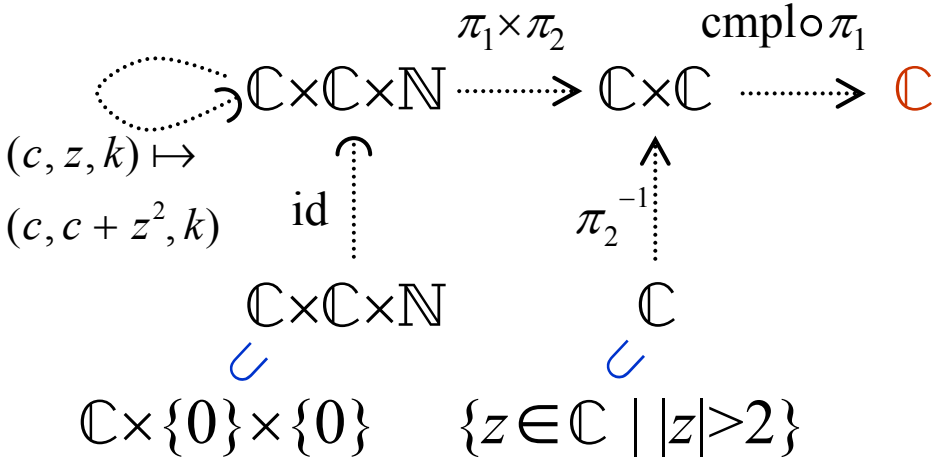
- Can emulate any Turing machine

Computation

- Can embed computation in any space







Information Measure

- **Define** the complexity of any subset **by** the size of the minimum diagram that represents it
- The size of a diagram is measured by the total size of maps generated from a fixed set of structure maps
  - id: $X \to X$, $\pi_i: X_1 \times \ldots \times X_n \to X_i$, $\omega : X \to \{0\}$
  - $f : X \to Y, g : Y \to Z \Rightarrow g \circ f : X \to Z$
  - $f_i : X \to Y_i \Rightarrow f_1 \times \ldots \times f_n : X \to Y_1 \times \ldots \times Y_n$

Information Measure

- The information measure of a subset $A \subset X$ relative to the set of maps $\mathcal{M}$ is denoted by

$$I(A|\mathcal{M})$$

- But any subset can be represented by the trivial representation

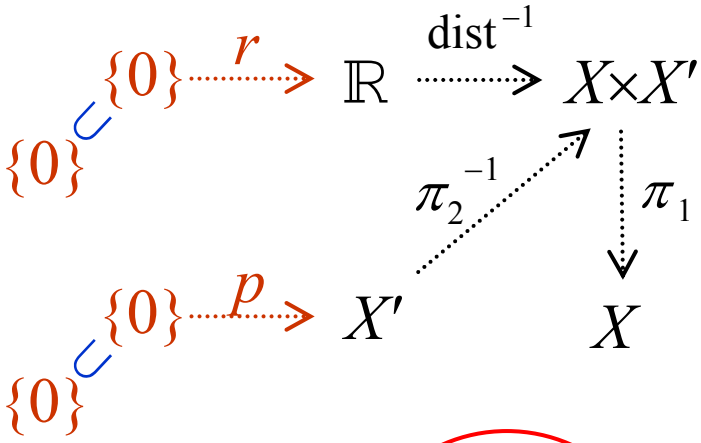
$$s(X)=A \curvearrowleft X$$

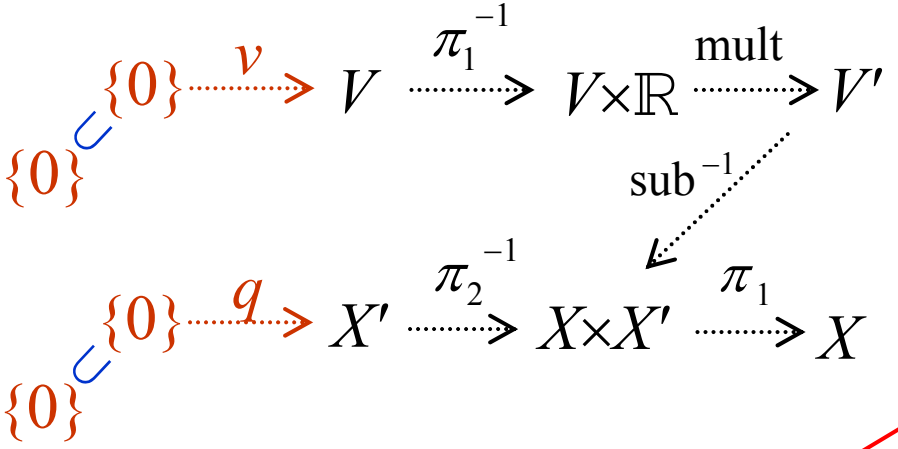
by defining the given set as the fixed partial cross section part of the representation.

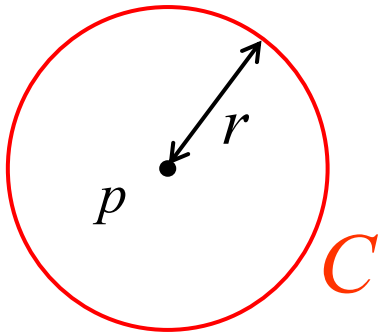
- No map is needed, so the size is 0.

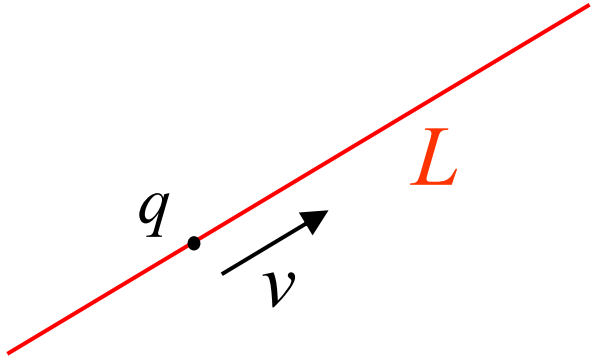
Information Measure

Consider constant $x \in X$ as a map $x : \{0\} \to X$, only allow the fixed partial cross section $s(\{0\}) = \{0\}$









$I(C \mid \{r, p, \text{dist}\}) \leq 5$

$I(L \mid \{v, q, \text{mult}, \text{sub}\}) \leq 7$

Information Measure

We choose the maps and constants that characterize the structure of the space in question.

For instance, for a Euclidean space X, it might be

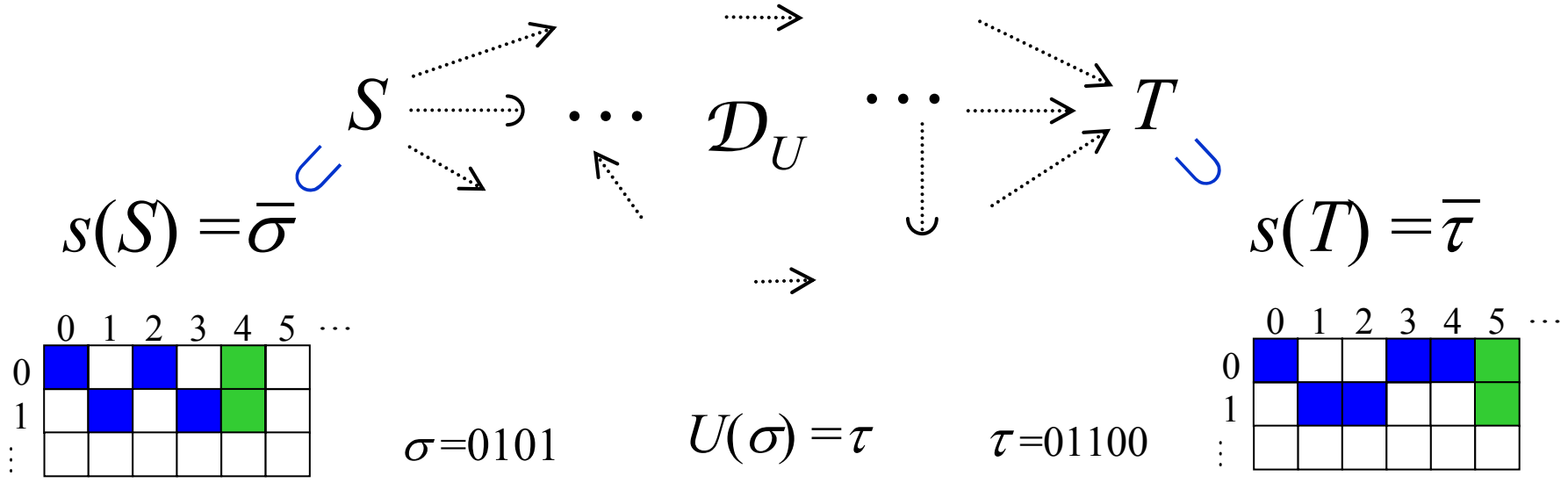
$$\mathcal{M}_{\mathrm{E}} = \{\text{dist, add, sub, mult}\} \cup X \cup V \cup \mathbb{R} \cup \{x : \{0\} \to X \mid x \in X\}$$

Then,

- $I(C \mid \mathcal{M}_{\mathrm{E}}) \leq 5,\ I(L \mid \mathcal{M}_{\mathrm{E}}) \leq 7$
- For any finite set $A \subset X$, $I(A \mid \mathcal{M}_{\mathrm{E}}) \leq |A|$

Relation with Kolmogorov Complexity

Theorem For any Turing machine U, there exists a diagram $\mathcal{D}_U$ generated by $\mathcal{M}_{\mathbb{N}}$ including the sets $S = \mathbb{N}\times\mathbb{N}$ and $T = \mathbb{N}\times\mathbb{N}$ such that, for any $\sigma \in b^*$ and any cross section s with $s(S) = \overline{\sigma}$, $s(T) = \varnothing$ if $U(\sigma) = \uparrow$ and $s(T) = \overline{\tau}$ if $U(\sigma) = \tau \in b^*$.

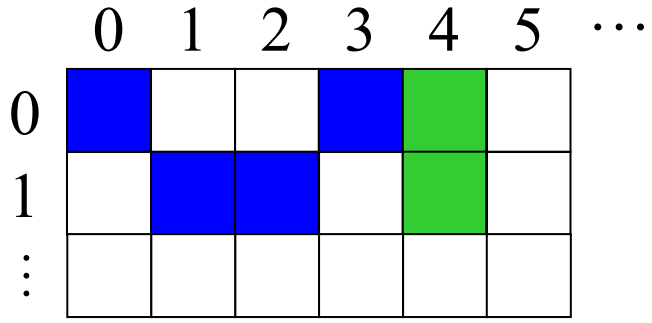


Relation with Kolmogorov Complexity

For $b = \{0,1\}$, $\sigma \in b^*$, define $\overline{\sigma} \subset \mathbb{N}\times\mathbb{N}$ as

$\overline{\sigma} = \{(i,\sigma[i]) \mid i = 0, 1,\cdots,|\sigma|-1\} \cup \{(|\sigma|,0), (|\sigma|,1)\}$

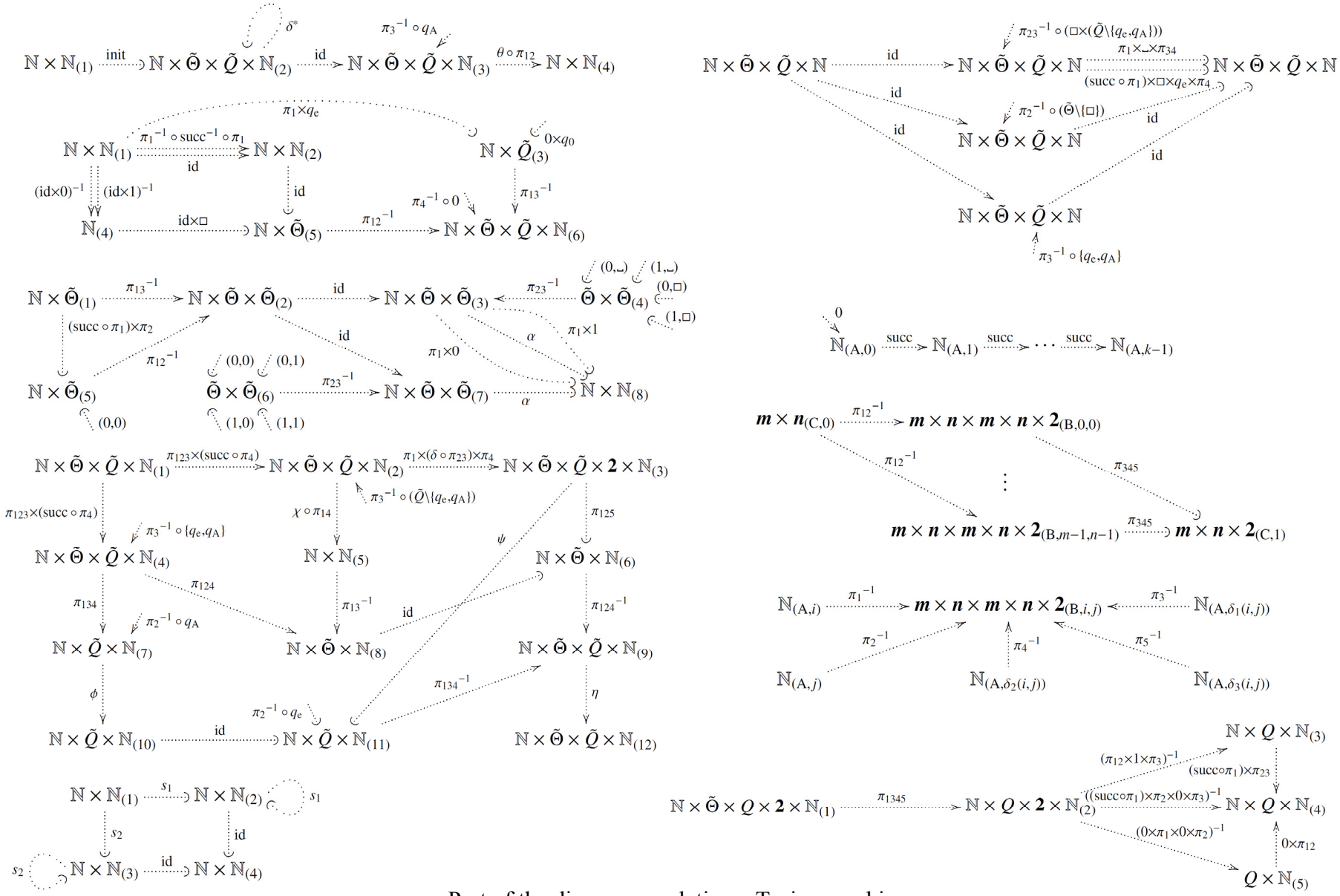
Ex:If σ=0110, $\overline{\sigma}$={(0,0),(1,1),(2,1),(3,0),(4,0),(4,1)}



$\mathcal{M}_{\mathbb{N}} = \{0, \text{succ}\}$

- $0:\{0\} \rightarrow \mathbb{N}$, $0(0) = 0$
- succ: $\mathbb{N} \rightarrow \mathbb{N}$ $[\text{succ}(n) = n+1]$

Relation with Kolmogorov Complexity



Part of the diagram emulating a Turing machine

Relation with Kolmogorov Complexity

Theorem There exists a Turing machine that, given an encoded diagram $\mathcal{D}_\sigma$ representing $\bar{\sigma}$ for $\sigma \in b^*$, outputs σ and terminates.

Since $\mathcal{D}_\sigma$ can be encoded with size $\sim I(\bar{\sigma} \mid \mathcal{M}_{\mathbb{N}})$, $K(\sigma)$ is dominated by a constant multiple of $I(\bar{\sigma} \mid \mathcal{M}_{\mathbb{N}})$

Automatic handling of structures

- Implement basic components approximately: spaces and maps
  - real numbers, spaces, …
- Structural varieties are generated combinatorially
- The structure does not depend on the approximation.
  - Data exchange between systems does not change the structure itself

Pattern Discovery as Optimization

- Given a signal-level data, find ever smaller diagrams representing it.
- The upper bound is the trivial representation
- Regular parts can be represented more compactly

Concluding Remarks

- Patterns and computations are defined directly
- Can mix sparse and dense representation
- Hierarchical and recursive representation possible
- Connects the representation to the represented.
  No arbitrariness in encoding across structures.
- The representation is a formalization of definition;
  i.e., a meta-definition.
- We can treat consistently and automatically what
  we usually define in natural languages