Complexity : lost in abstraction ?

PAC 2024

December 17, 2024

Baptiste Chanus

A really concrete motivation

From : what computations can we <u>effectively</u> carry ?

To : What problems can we <u>effectively</u> solve ?

1. Complexity theory : a pile of abstractions

Definition of RAM

Definition (RAM machine)

A RAM machine is a list of I of $N+1$ instructions, two registers r and w containing positive integers, an input list of values $\vec{i} = (i_0, \dots, i_m)$ and a list of registers $\vec{x} = (x_0, \dots, x_n, \dots)$ containing infinitely many zeros.

An instruction can be :

- An operation : a substitution $x_0 := x_0 * x_1$ where $* \in \{\wedge, \vee\}$ or $x_0 := \neg x_0$.
- A branch(l) : if $x_0 = 0$ then goto instruction l; else goto next instruction.
- A copy(x_r, x_w) : the content of the register x_r is updated with the value of x_w.

Example of a run

Input Tape

i_0	i_1
1	1

Run

	x_0	x_1	x_2	x_3	x_4	x_5
$init$	0	0	1	1	0	0
$copy$	1	0	1	1	0	0
$copy$	1	1	1	1	0	0
$\wedge$	1	1	1	1	0	0
$copy$	1	1	1	1	1	0
$copy$	1	1	1	1	1	0
$\vee$	1	1	1	1	1	0
$copy$	1	1	1	1	1	1
$copy$	1	1	1	1	1	1
$\neg$	0	1	1	1	1	1
$copy$	0	1	1	1	1	1
$\wedge$	0	1	1	1	1	1
$copy$	0	0	1	1	1	1
$copy$	1	0	1	1	1	1

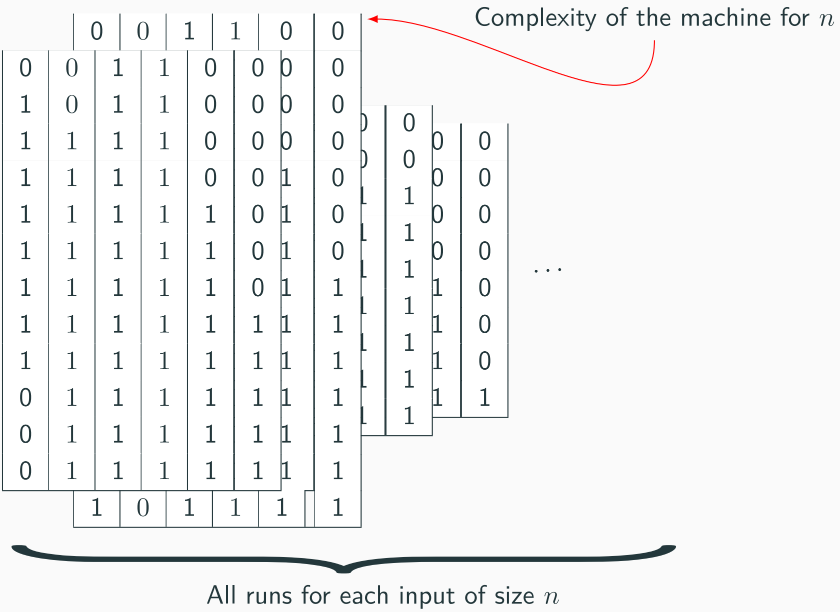
Complexity : first intuition

	x_0	x_1	x_2	x_3	x_4	x_5
$init$	0	0	1	1	0	0
$copy$	1	0	1	1	0	0
$copy$	1	1	1	1	0	0
$\wedge$	1	1	1	1	0	0
$copy$	1	1	1	1	1	0
$copy$	1	1	1	1	1	0
$\vee$	1	1	1	1	1	0
$copy$	1	1	1	1	1	1
$copy$	1	1	1	1	1	1
$\neg$	0	1	1	1	1	1
$copy$	0	1	1	1	1	1
$\wedge$	0	1	1	1	1	1
$copy$	0	0	1	1	1	1
$copy$	1	0	1	1	1	1

Time

Space

Complexity: first level of abstraction



Complexity : abstraction again

Definition (Decision Problem)

A **decision problem** L (on *string*) is a set of strings on an alphabet Σ. We write $L \subseteq \Sigma$.

Definition (Complexity of a Problem)

The complexity of a problem is the complexity that uses the least amount of (a given) ressources to solve the problem.

Remark (Complexity Class)

A **complexity class** is a set of problems that are solvable in *bounded ressources* (time, space...) in a given model of computations (Turing Machines, RAM).

2. Descriptive complexity : diving deeper

Descriptive complexity : abstraction's final boss ?

Remark (Machine-less complexity)

Could we find a way to define complexity independantly of any model of computation ?

Descriptive complexity : languages and computations

Logical description

- Finite structures over finite signatures
- Logical ressources for expressivity (higher order quantifiers, operators)

Decision algorithm

- Models of computations (Turing machines, circuits)
- Computational ressources (Time, space)

Descriptive complexity : languages and computations

Logical description

- Finite structures over finite signatures
- Logical ressources for expressivity (higher order quantifiers, operators)

Decision algorithm

- Models of computations (Turing machines, circuits)
- Computational ressources (Time, space)

Descriptive complexity

$$\boxed{x \models F} \iff \boxed{x \in L}$$

Descriptive complexity : languages and computations

Logical description

- Finite structures over finite signatures
- Logical ressources for expressivity (higher order quantifiers, operators)

Decision algorithm

- Models of computations (Turing machines, circuits)
- Computational ressources (Time, space)

Descriptive complexity

Input

$$x \models F \iff x \in L$$

Logical description

Computational description

Boolean Theories

Definition

A **Boolean theory** $\mathbb{T}$ is a triple

$$(\mathrm{Sort}(\mathbb{T}), \mathrm{Rel}(\mathbb{T}), \mathrm{Ax}(\mathbb{T}))$$

A Boolean theory $\mathbb{T}$ is **finite** if $\mathrm{Sort}(\mathbb{T})$, $\mathrm{Rel}(\mathbb{T})$ and $\mathrm{Ax}(\mathbb{T})$ are all finite.

Example : $\mathbb{S}\mathrm{tr}$

Definition

$$\mathrm{Sort}(\mathbb{S}\mathrm{tr}) = \{N\}$$
$$\mathrm{Rel}(\mathbb{S}\mathrm{tr}) = \{\leq \rightarrowtail N \times N, X \rightarrowtail N\}$$
$$\mathrm{Ax}(\mathbb{S}\mathrm{tr}) = \{\text{“} \leq \text{is a total order''}\}$$

$$\begin{array}{cccccccccc}
(N, \leq) & : & \mathsf{a} & \leq & \mathsf{b} & \leq & \mathsf{c} & \leq & \mathsf{d} \\
& & \uparrow & & \uparrow & & \uparrow & & \uparrow \\
X & : & 0 & & 1 & & 0 & & 1
\end{array}$$

Other example : $\mathbb{G}\mathrm{rph}$

Definition

$$\begin{aligned}
\mathrm{Sort}(\mathbb{G}\mathrm{rph}) &= \{V\} \\
\mathrm{Rel}(\mathbb{G}\mathrm{rph}) &= \{E \rightarrowtail V \times V\} \\
\mathrm{Ax}(\mathbb{G}\mathrm{rph}) &= \emptyset
\end{aligned}$$

Extension of a theory

Definition

$\mathbb{T}$ extends $\mathbb{T}'$ iff :

- $\mathrm{Sort}(\mathbb{T}') \subseteq \mathrm{Sort}(\mathbb{T})$
- $\mathrm{Rel}(\mathbb{T}') \subseteq \mathrm{Rel}(\mathbb{T})$
- $\mathrm{Ax}(\mathbb{T}') \subseteq \mathrm{Ax}(\mathbb{T})$

Definition

$\mathbb{T}$ is a relational extension of $\mathbb{T}'$ iff :

- $\mathbb{T}'$ is an extension of $\mathbb{T}$
- $\mathrm{Sort}(\mathbb{T}') = \mathrm{Sort}(\mathbb{T})$

Extension of a theory

Definition

$\mathbb{T}$ extends $\mathbb{T}'$ iff :

- $\mathrm{Sort}(\mathbb{T}') \subseteq \mathrm{Sort}(\mathbb{T})$
- $\mathrm{Rel}(\mathbb{T}') \subseteq \mathrm{Rel}(\mathbb{T})$
- $\mathrm{Ax}(\mathbb{T}') \subseteq \mathrm{Ax}(\mathbb{T})$

Definition

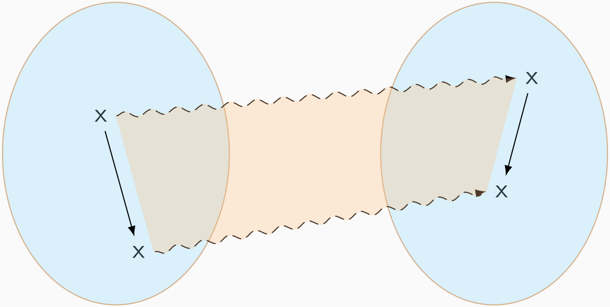
$\mathbb{T}$ is a relational extension of $\mathbb{T}'$ iff :

- $\mathbb{T}'$ is an extension of $\mathbb{T}$
- $\mathrm{Sort}(\mathbb{T}') = \mathrm{Sort}(\mathbb{T})$

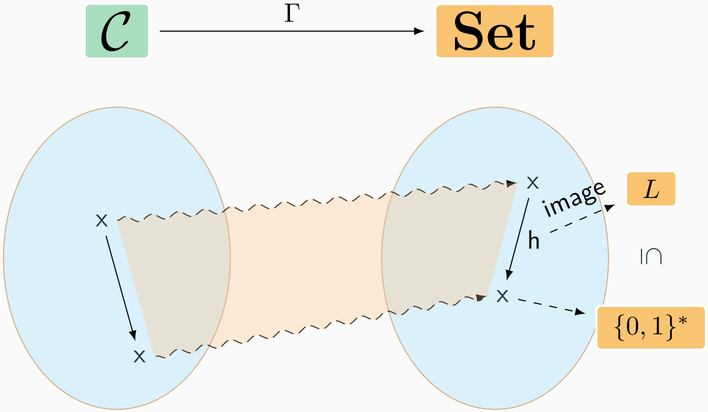
Note that there is a natural notion of projection from the extension to the base theory. (i.e. the one that forgets the extra information)

A drawing is worth more than a thousand word

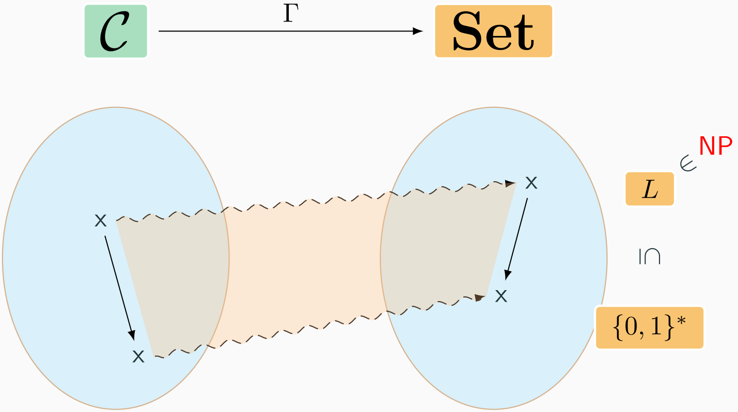
$$\mathcal{C} \xrightarrow{\Gamma} \mathbf{Set}$$



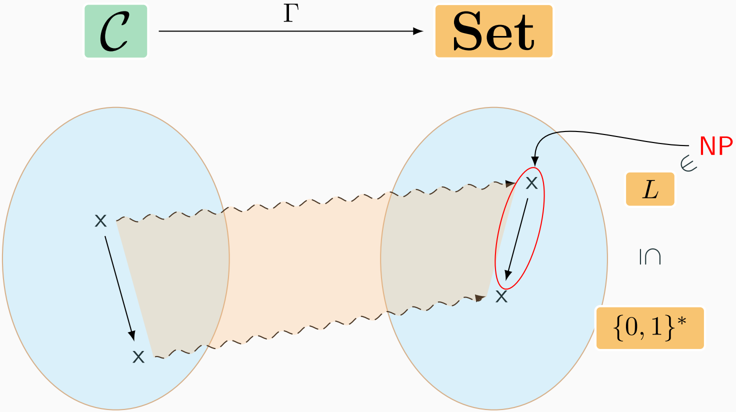
A drawing is worth more than a thousand word



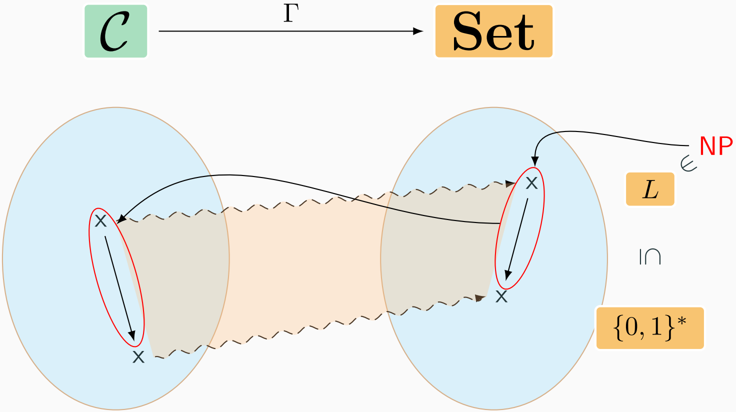
A drawing is worth more than a thousand word



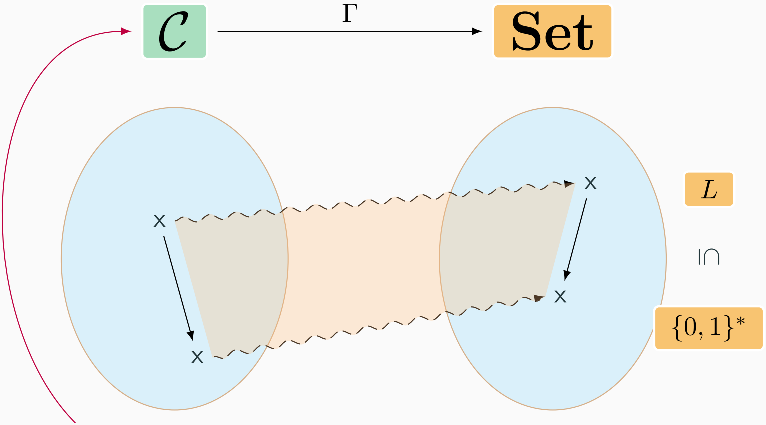
A drawing is worth more than a thousand word



A drawing is worth more than a thousand word

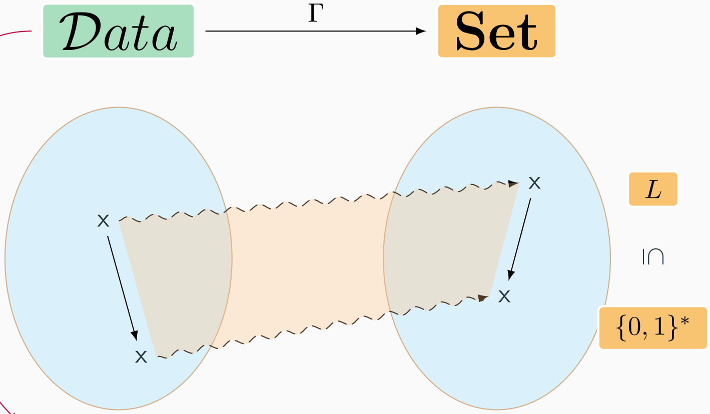


A drawing is worth more than a thousand word



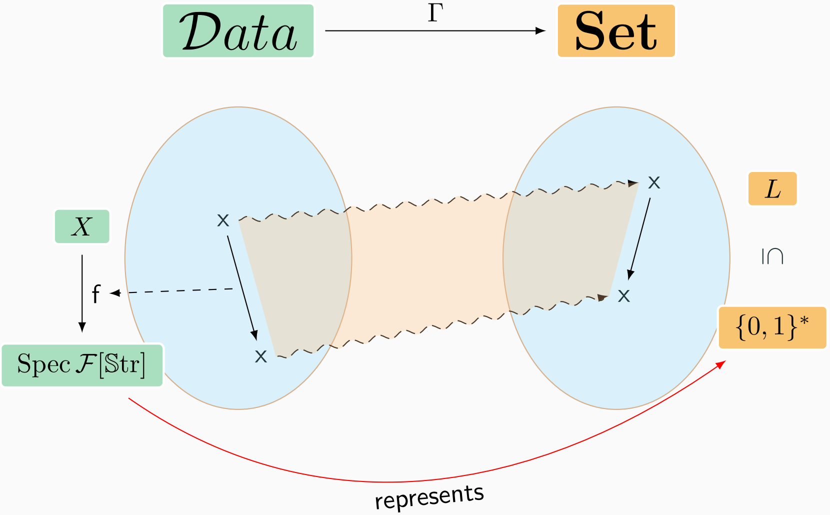
What is that ?

A drawing is worth more than a thousand word

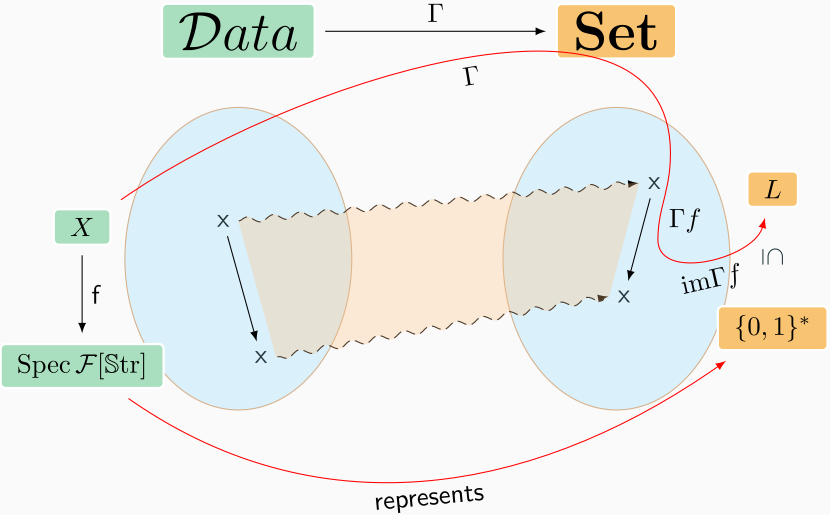


Data specifications

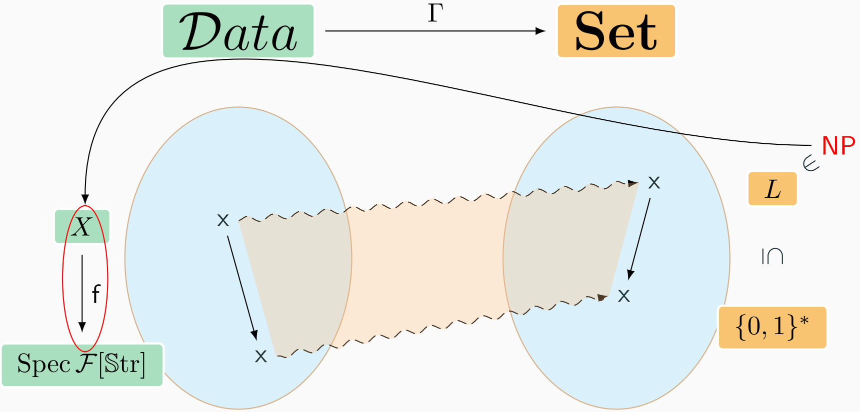
A drawing is worth more than a thousand word



A drawing is worth more than a thousand word



A drawing is worth more than a thousand word



Fagin's Theorem (our version)

Theorem (Fagin (Boolean sauce))

NP is equal to the relational extensions of $\mathbb{S}\mathrm{tr}$.

3. Abstracting again ?

Why not after all ?

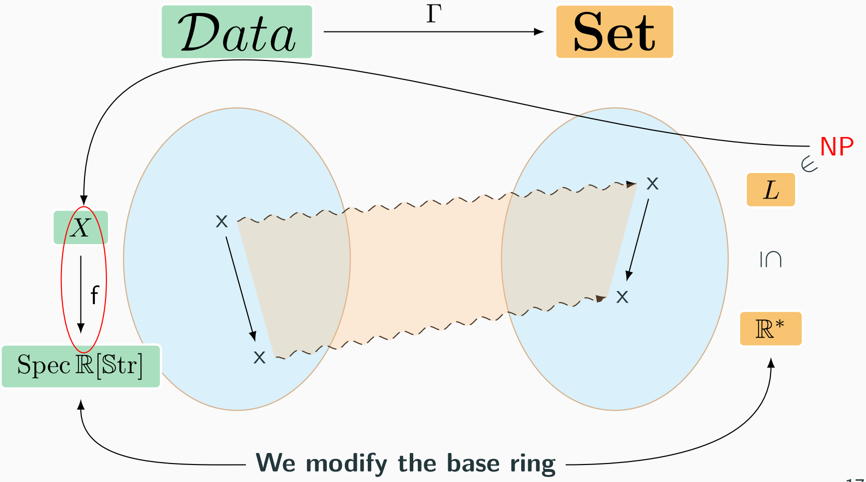
Definition (RAM machine)

A RAM machine on real numbers is a list of I of $N+1$ instructions, two registers r and w containing positive integers, an input list of values $\vec{i} = (i_0, \dots, i_m)$ and a list of registers $\vec{x} = (x_0, \dots, x_n, \dots)$ containing infinitely many zeros.

An instruction can be :

- An operation : a substitution $x_0 := x_0 * x_1$ where $* \in \{+, -, \times, /\}$.
- A branch(l) : if $x_0 = 0$ then goto instruction l; else goto next instruction.
- A copy(x_r, x_w) : the content of the register x_r is updated with the value of x_w.

Why not after all ?



Trust me...

Theorem

The Fagin's theorem still holds over $\mathbb{S}\mathrm{tr}_{\mathbb{R}}$.

Is this loss ?

Our computers cannot manipulate real numbers.
We were supposed to talk about <u>effective</u> computations !

The twist !

Remark

Could we use the concept of computation outside the scope of engineering ?

Examples in physics

The Second Law of Quantum Complexity. Brown, Adam R., et Leonard Susskind. (2018).

Example

A relation between the volume of a black hole and quantum circuit complexity.

Examples in physics

MIP=RE.* Ji, Zhengfeng, Anand Natarajan, Thomas Vidick, John Wright, et Henry Yuen (2022).

Example

Explicit constructions of counter-examples to Connes' Embedding Conjecture using a computability result.

MIP=RE.* Ji, Zhengfeng, Anand Natarajan, Thomas Vidick, John Wright, et Henry Yuen (2022).

Example

Explicit constructions of counter-examples to Connes' Embedding Conjecture using a computability result.

Remark

Computability is complexity with infinite ressources...

The end.

Could complexity be used in other disciplines ?