Describility in Formal Systems Concerning the $\mathcal{P} \stackrel{?}{=} \mathcal{NP}$ Problem

Claus Akira Horodynski Matsushigue

Departamentos de Matemática
IME — Universidade de São Paulo (USP)
IGCE — Universidade Estadual Paulista (UNESP)
São Paulo, Brasil

—Brasília, February, 2004—

Presenting the $\mathcal{P} \stackrel{?}{=} \mathcal{NP}$ Problem

- The most important complexity classes are:

$$\boldsymbol{\mathcal{L}} \subseteq \mathcal{NL}\text{=co-}\mathcal{NL} \subseteq \boldsymbol{\mathcal{P}} \subseteq (\mathcal{NP}\cap\text{co-}\mathcal{NP}) \subseteq \boldsymbol{\mathcal{NP}} \subseteq$$

$$\mathcal{PH} \subseteq \boldsymbol{\mathcal{PSPACE}}\text{=}\mathcal{NPSPACE} \subseteq \boldsymbol{\mathcal{EXP}} \subseteq \mathcal{NEXP}.$$

- We also know that

$$\mathcal{L} \neq \mathcal{PSPACE}$$

and

$$\mathcal{P} \neq \mathcal{EXP}.$$

- Notice that all the above results are obtained in a very simple way at the beginning of Complexity Theory studies.

- However, that is all that we know!

- Moreover, all of the remaining inclusions are supposed as being proper (viz., they are not equalities either).

- The classical problems in Structural Complexity Theory are:

$$\mathcal{P} \stackrel{?}{=} \mathcal{NP}$$

$$\mathcal{NP} \stackrel{?}{=} \text{co–}\mathcal{NP}$$

$$\mathcal{L} \stackrel{?}{=} \mathcal{NL}$$

and the collapse of the polynomial hierarchy

(or $\mathcal{PH} \stackrel{?}{=} \mathcal{PSPACE}$).

- The question $\mathcal{P} \stackrel{?}{=} \mathcal{NP}$ may be considered as the main open problem in Computer Science.

- Due to its difficulty, it also becomes a very important problem in Mathematics.

- However, the statement "$\mathcal{P} = \mathcal{NP}$" is perhaps *independent* of traditional formal logical systems, like the Zermelo–Frænkel Set Theory with the axiom of choice, ZFC.

  It means that both "$\mathcal{P} = \mathcal{NP}$" and "$\mathcal{P} \neq \mathcal{NP}$" may be unprovable in ZFC.

Supposing this independence...

- Foundational techniques seem to be a powerful way to deal with questions on Structural Complexity Theory.

  Moreover, if the statement "$\mathcal{P} = \mathcal{NP}$" is indeed independent of formal systems:

  - they are the unique way for which independence can be proved, and
  - other mathematical techniques will be totally useless.

- We shall present three important differences between the statement "$\mathcal{P} = \mathcal{NP}$" and the classical statements of Mathematics that are known to be independent of ZFC (such as CH).

  At this point, the **first two** differences appear:

(i)
- ∗ Classical statements of Mathematics that are known to be independent of ZFC refer to *(ideal) infinite elements.*
- ∗ In the opposite direction, the computational complexity objects are *finite.*
  Moreover, all of them can be dealt with in the Hilbert's (finitistic) metamathematical conception.

(ii) Observe that the security of electronic communications (made by financial institutions like banks) are based on the supposition that $\mathcal{P} \neq \mathcal{NP}$ holds.

The statement "$\mathcal{P} = \mathcal{NP}$" has, in fact, a lot of very important practical consequences in the *real world*.

Then imagining if we deal with a practical problem as

Theorem ($\mathcal{P} \neq \mathcal{NP}$): ....

in the same way as happens in Mathematics

Theorem (CH): ....

In this sense, the *relativistic* treatment given to other mathematical independent statements are inconceivable if "$\mathcal{P} = \mathcal{NP}$" is independent.

Structural Complexity Theory

- An *alphabet* Γ is a finite set of *letters* (with at least 2 elements).

- A *word* x is a finite (possibly empty) sequence of letters. The set of all words will be denoted by $\Gamma^* := \Gamma^{<\omega}$.

- Notice that we can easily associate the set of all words with the set of natural numbers:

$$\Gamma^* \quad \rightsquigarrow \quad \mathbb{N}.$$

- A *decision problem* is an arbitrary subset of words, i.e.,

$$L \subseteq \Gamma^*,$$

so we can think it as a subset of $\mathbb{N}$, i.e.,

$$L \in \mathcal{P}(\mathbb{N}).$$

- A *complexity class* is a set of decision problems, i.e.,

$$\mathcal{C} \subseteq \mathcal{P}(\Gamma^*),$$

so we can think it as being

$$\mathcal{C} \in \mathcal{P}(\mathcal{P}(\mathbb{N})).$$

- The $\mathcal{P}$, $\mathcal{NP}$, co-$\mathcal{NP}$, $\mathcal{PH}$, etc. are examples of complexity classes.

- Thus, Structural Complexity Theory compares complexity classes, which are elements of $\mathcal{P}(\mathcal{P}(\mathbb{N}))$.

- Observe that ZFC cannot decide either what is the cardinality of $\mathcal{P}(\mathbb{N})$ $\quad$ (viz., $|2^\omega| = ?$), $\quad$ with $|2^\omega| \in \mathcal{P}(\mathcal{P}(\mathbb{N}))$.

- But the difference is that $|2^\omega|$ requires infinite concepts to be described. On the other hand, the complexity class $\mathcal{P}$ is totally described by finite concepts.

  Although Complexity issues seem to be more simple, however, this may not be always the case.

- By having made a precise definition of *Turing Machines* (and using the association of words with the natural numbers), we can codify (and enumerate) every Turing Machine:

$$M_0, M_1, M_2, \ldots.$$

- Using it, we also make an enumeration of the *decision problems accepted* by at least one Turing Machine

$$L_0, L_1, L_2, \ldots.$$

taking

$$\begin{aligned} L_i &:= \mathtt{L}(M_i) \\ &:=: \{ x \in \Gamma^* \parallel M_i(x) \text{ has acceptance } \} \\ &:=: \{ x \in \Gamma^* \parallel M_i \text{ accepts the input } x \}. \end{aligned}$$

Notice that there are repetitions in this last enumeration.

Unprovabilities and Indemonstrabilities

- For us, a *formal (logical) system* is thought as being a logical system in the formal framework idealized by Hilbert in his Formalist Program.

  Then, first of all, the underlying logic of a formal system is Classical First Order Logic.

- Also, formal systems will be (recursively) axiomatizable theories.

- However, the metatheory can be sometimes taken stronger than the Hilbert's (finitistic) Metamathematics, as Cohen's forcing technique.

- From now on, let T be a (consistent) formal system.

(1) Then there exists $e < \omega$ in such a way that $L_e \in \mathcal{PH}$ (viz., it holds in the metatheory) and $\ulcorner L_e \in \mathcal{PH}\urcorner^T$ is independent of T, i.e.,

$$T \nvdash \ulcorner L_e \in \mathcal{PH}\urcorner^T$$

and

$$T \nvdash \neg \ulcorner L_e \in \mathcal{PH}\urcorner^T.$$

This also happens with all important complexity classes, as well as every level of the hierarchy of $\mathcal{PH}$ (including $\mathcal{P}$, $\mathcal{NP}$ and co–$\mathcal{NP}$) and the space bounded complexity classes (such as $\mathcal{L}$, $\mathcal{NL}$ and $\mathcal{PSPACE}$).

(2) There exists $e < \omega$ such that M_e has logarithmic non–deterministic space and

$$\ulcorner M_e \text{ has logarithmic non–deterministic space} \urcorner^T$$

is independent of T. This also holds for every combination of:

– (logarithmic), polynomial and exponential;

– (non–)deterministic; and

– time/space bounds.

(3) For all of them, it also is possible to obtain e in an *effective* (viz., constructive) way in Hilbert's Metamathematics.

(4) There exists $e < \omega$ for which M_e runs in non–deterministic time f and

$$\ulcorner M_e \text{ runs in non–deterministic time } f \urcorner^T$$

is independent of T.

The same happens if we change the bound to

$$\text{(non–)deterministic space } h.$$

Here, the restrictions to the functions f and g are very small.

(5) Again we can obtain these results finding e in an *effective* way in Hilbert's Metamathematics.

- All of the above results which do *not* concern the *effective existence* were previously proved by Hájek '77 and '79 in a polynomial deterministic time formulation.
  Notice that, as his study was not provided with the *non–deterministic* computations, his results did not work in the $\mathcal{P} \stackrel{?}{=} \mathcal{NP}$ problem.

- The above *effective existence* results are based on Hartmanis–Hopcroft '76.
  In order to obtain them, we must prove in a very suitable setting the SMN–Theorem (and the Universal Turing Machine Theorem), taking into account non–deterministic space/time bounds.

- The next results follow some ideas introduced by Grant '80.

(6) For every decision problem L there exists $e < \omega$ for which $L_e = L$ and $\ulcorner L_e = L \urcorner^T$ is unprovable in T.

(7) For every decision problem L there exists $e < \omega$ for which $L_e = L$ and $\ulcorner L_e \in \mathcal{NP} \urcorner^T$ is independent of T.

(8) For every decision problem L there exist $e, d < \omega$ such that $L_e = L_d = L$ and both

$$\ulcorner M_e \text{ has polynomial non–deterministic time} \urcorner^T$$

and

$$\ulcorner M_d \text{ does not have polynomial non–deterministic time} \urcorner^T$$

are unprovable in T.

- Once again, the last two results are valid for all usual complexity classes, as well as for the other usual (non–)deterministic time/space bounds.

Internalizing Complexity statements

- Notice that the above independence results are based on a description $\ulcorner R \urcorner^T$ of a statement R in the language of a formal system (viz., an axiomatizable theory) T.

  But we have not discussed until now what this description actually is.

- An important task performed by Gödel, in order to prove his famous Incompleteness Theorems, was to internalize the Hilbert's Metamathematics inside (a little bit strong) formal systems (actually the formal systems which can interpret -or extend- Robinson Arithmetics PA^{-}).

  This is the so-called *arithmetization of Metamathematics.*

- The main idea of it is to *internalize* into a theory notions which we know outside it (i.e., in the metatheory). Thus it is done in the following direction:

$$\text{from semantics} \quad \Longrightarrow \quad \text{to syntax.}$$

- The internalization of $R(x)$ (a unary statement on $\mathbb{N}$, i.e., $R \subseteq \mathbb{N}$) in a theory T is defined in two ways:

  – in the *representation* case, for all $i \in \mathbb{N}$:

$$\begin{aligned} R(i) \text{ holds} \quad &\Longrightarrow \quad T \vdash \ulcorner R(i) \urcorner^{T} \\ R(i) \text{ does not hold} \quad &\Longrightarrow \quad T \vdash \neg \ulcorner R(i) \urcorner^{T}; \end{aligned}$$

  – in the *expressiveness* case, for all $i \in \mathbb{N}$:

$$R(i) \text{ holds} \quad \Longleftrightarrow \quad T \vdash \ulcorner R(i) \urcorner^{T}.$$

- However, the first independence results presented here follow directly from the following classification in the Arithmetical Hierarchy:

  - They are Σ_3–complete sets:

$$\{i < \omega \| L_i \in \mathcal{P}\},$$

$$\{i < \omega \| L_i \in \mathcal{NP}\}$$

    and

$$\{i < \omega \| L_i \in \mathcal{PH}\};$$

– They are Σ_2–complete sets:

$$\{\ i < \omega \parallel M_i \text{ has polynomial non–deterministic time } \}$$

and

$$\{\ i < \omega \parallel M_i \text{ has logarithmic non–deterministic space } \};$$

– They are Π_1–complete sets:

$$\{\ i < \omega \parallel M_i \text{ runs in non–deterministic time } f\ \}$$

and

$$\{\ i < \omega \parallel M_i \text{ runs in non–deterministic space } h\ \}.$$

- Just the following relations are important now:

$$
\begin{array}{rcl}
\Sigma_1\text{–sets} & \Longleftrightarrow & \text{recursively enumerable sets;} \\
\Pi_1\text{–sets} & \Longleftrightarrow & \text{complementary of the recursively} \\
& & \text{enumerable sets;} \\
\Delta_1\text{–set} & \Longleftrightarrow & \text{recursive sets;}
\end{array}
$$

And so on. (Here, $\Delta_1 := \Sigma_1 \cap \Pi_1$).

- The Σ_3–, Σ_2– and Π_1–complete set *are not* recursively enumerable (since they are Π_1–hard).

- However, in a consistent axiomatizable theory T, every representable and expressible subset of $\mathbb{N}$ is recursive and recursively enumerable, respectively.

- So, it *is not possible* to internalize in (consistent) formal systems (viz., consistent axiomatizable theories) conceptions that *are not* recursively enumerable.

- Therefore, it *is impossible to internalize* all the statements above in (consistent) formal systems.

- Remark:
  All of these Arithmetical Hierarchy classifications are made in the metatheory, i.e., independently whatever a consistent axiomatizable theory T is taken.

- First of all, notice that every *deterministic time* version of the above results was obtained by Hájek '77 and '79, following ideas used by Hartmanis and Hopcroft '76.

- But they made use of descriptions as $\ulcorner L_x \in \mathcal{P} \urcorner^T$ in an indiscriminate way, not taking into account what their meaning is inside a theory.

- Of course, all of these statements may be codified in formal systems which extend Robinson's Arithmetic PA^-.

  But the question is: what can their meaning be inside these theories, remembering that they cannot be internalized?

Certification

- We relax the notion of internalization, inverting its direction by:

$$\text{from syntax} \quad \Longrightarrow \quad \text{to semantics.}$$

- In fact, this is what we hope to get from formal systems in order to obtain (outside) results which are proved inside a formal system.

- We define as a *certification* of a unary statement on $\mathbb{N}$, $R(x)$, in a theory T when we have that, for all $i \in \mathbb{N}$:

$$\begin{aligned} T \vdash \ulcorner R(i) \urcorner^T \quad &\Longrightarrow \quad R(i) \text{ holds} \\ T \vdash \neg \ulcorner R(i) \urcorner^T \quad &\Longrightarrow \quad R(i) \text{ does not hold.} \end{aligned}$$

- Remarks:

The statements (in general) do not have certifications in inconsistent theories.

Also in complete theories, a certification plays the same role as representation and expressiveness.

- In this way, it is important:

  All our above results hold for:

  – all formal systems T (i.e., axiomatizable theories), that internalize the natural numbers $\mathbb{N}$; and

  – all certifications $\ulcorner \cdot \urcorner^T$ of the respective statement in T.

- We also define a *pseudo–certification* of a sentence statement φ in a theory T by:

$$\begin{aligned} T \vdash \ulcorner\varphi\urcorner^{T} &\quad\Longrightarrow\quad \varphi \text{ holds} \\ T \vdash \neg\ulcorner\varphi\urcorner^{T} &\quad\Longrightarrow\quad \varphi \text{ does not hold.} \end{aligned}$$

- Remarks:

  The *ideal/theoretical* conception of the (pseudo–)certification becomes evident now.

  Probably, to prove that any formula is a (pseudo–)certification of a statement is as hard as to prove directly this statement.

  So we are always supposing that a (pseudo–)certification is syntactically described in such a way that it has the semantic intentionality which we want.

- An example of a pseudo–certification of the $\mathcal{P} = \mathcal{NP}$ statement may be based on the conception

$$\text{“}\forall x(L_x \in \mathcal{P} \leftrightarrow L_x \in \mathcal{NP})\text{”}$$

(where we must know how to certify the statements “$L_x \in \mathcal{P}$” and “$L_x \in \mathcal{NP}$” in T).

In this way,

$$\ulcorner \mathcal{P} = \mathcal{NP} \urcorner^{T} \quad := \quad \forall x(\ulcorner L_x \in \mathcal{P} \urcorner^{T} \leftrightarrow \ulcorner L_x \in \mathcal{NP} \urcorner^{T}).$$

- When a decision problem L is $\mathcal{NP}$–complete, we have that

$$\mathcal{P} = \mathcal{NP} \iff L \in \mathcal{P}.$$

In some sense, the $\mathcal{NP}$–complete problems are the hardest $\mathcal{NP}$ decision problems to be solved.

There are a lot of $\mathcal{NP}$–complete problems and they are, in general, the most important problems in either practical or theoretical point of view.

- So, if L_i is $\mathcal{NP}$–complete, we may consider

$$\ulcorner \mathcal{P} = \mathcal{NP} \urcorner^{T} \quad := \quad \ulcorner L_i \in \mathcal{P} \urcorner^{T}$$

as being a pseudo–certification in T.

The decision problem of the boolean satisfiable formulas, SAT, is a $\mathcal{NP}$–complete.

(9) Therefore, by (7), there is a way to interpret SAT (i.e., an $e < \omega$ such that $L_e = \mathrm{SAT}$) for which

$$\ulcorner \mathcal{P} = \mathcal{NP} \urcorner^{T}$$

is *independent* of T.

Formal Logical Systems

- The above results do not point only to independence, but also that these statements *cannot even be axiomatized in a "stable" manner.*

  That is, whichever way we try to axiomatize these statements, independence persists.

- At this point, the **third** important difference between the statement "$\mathcal{P} = \mathcal{NP}$" and the classical statements of Mathematics that are known to be independent of ZFC appears:

(iii) ∗ Since the usual mathematical statements refer to (ideal) infinite notions, they can be only expressed in a syntactical manner and it is done in only one way.
So, they actually always are well–defined syntactically and there is no point involving this matter.

∗ On the other hand, there are several ways to codify each Computational Complexity statement inside a formal system. Then, in order to perform the axiomatization, there is no way to establish which of those codifications is *the canonical* one.

- Therefore, it would escape, in a frontal way, from the Hilbert Program if we have to pay attention to the syntactical description of the statements when we are dealing with mathematical issues.

Conclusion

- In some sense, we have more Computational Complexity information outside a formal logical system (viz., in the metatheory) than inside it (viz., in the theory).

- Therefore, it is basically a question of discussing how small is the capacity of formal logical systems (viz., axiomatizable theories of Classical First Order Logic) to describe correctly Computational Complexity issues.