

## IMPROVING THE CLASSIFICATION OF COMPLEXITIES

ANGEL GARRIDO

ABSTRACT. When we estimate the temporal complexity of an algorithm that we suppose useful to solve a certain problem, some difficulties can appear. As we know, classical problems exist, and many other remain open, only conjectured. So, we consider the last attempts to see the relation between the classes P and NP. Our purpose is to contribute to clarify this question.

2000 *Mathematics Subject Classification*: Automata Theory, Complexity Theory, Temporal Complexity.

### 1. INTRODUCTION

Can a certain problem be resolvable? Is it possible to reach such solution in a reasonable amount of time? Because it can be necessary too much time.

The Theory of Complexity attempts to classify all the known solvable problems, according to their degrees of difficulty, so as the power of the tools we may use to solve them. We can establish some results, known as Hierarchy Theorems. For instance, the famous classes P and NP are the first in the tower of complexity classes, if we consider the polynomial-time hierarchy.

### 2. HIERARCHY THEOREMS (DETERMINISTIC CASE)

Such aforementioned Hierarchy Theorems constitute the result of successive attempts to clarify such relation between the class  $P$  and  $NP$ . We will consider many other classes, modulating our requirements.

We need, previously, some definitions. Supposing known our reference model, the  $TM$ , in its respective varieties: deterministic, non-deterministic and alternating.

As we say, the most relevant resources would be the space and the time. The space considered as the number of different cells of the work tape that are accessed during the computation. And the time as the number of transformations (possibly, on the same position of memory) that intervening during the process, until its final state.

Now, we need bounds of the allotted space and time. For this, we have the constructible functions.

The function:

$$s : N \rightarrow [0, +\infty) = R_+ \cup \{0\}$$

is *space-constructible*, if there exists a *TM* that runs in space exactly  $s(n)$ , on every input of size  $n$ .

Analogously, the function:

$$t : N \rightarrow [0, +\infty) = R_+ \cup \{0\}$$

is *time-constructible*, if there exists a *TM* that runs in time exactly  $t(n)$ , on every input of size  $n$ .

The class of languages that can be decided in time  $O[t(n)]$ , on a *TM*, is denoted:

$$DTIME[t(n)]$$

And the class of languages that can be decided in space  $O[s(n)]$ , on a *TM*, is:

$$DSPACE[s(n)]$$

*Space Hierarchy Theorem*

Let  $s_i : N \rightarrow [0, +\infty) = R_+ \cup \{0\}$ ,  $i = 1, 2$ , be such  $s_2$  is space-constructible and  $s_2 \in \omega(s_1)$ . Then, we have the strict inclusion:

$$DSPACE[s_1(n)] \subset DSPACE[s_2(n)]$$

*Time Hierarchy Theorem*

Let  $t_i : N \rightarrow [0, +\infty) = R_+ \cup \{0\}$ ,  $i = 1, 2$ , be such  $t_2$  is time-constructible and  $t_2 \in \omega(t_1)$ . Then, we have the strict inclusion:

$$DTIME[t_1(n)] \subset DTIME[t_2(n)]$$

The computations on a sequential machine ( $TM$ ) can be considered *space efficient*, if they take no more than logarithmic work space. For instance, deciding whether a graph is acyclic can be done in logarithmic space.

And our computations on a sequential machine ( $TM$ ) can be considered *time efficient*, if they take no more than polynomial time.

### 3. HIERARCHY THEOREMS (NON-DETERMINISTIC CASE)

We can also consider a *non-deterministic Turing Machine, NTM*, which can be viewed as a formalization of a proof system. So, works interacting between an all-powerful prover and a computationally limited verifier. Such  $NTM$  permits the characterization of the complexity for more general computational problems.

*Non – deterministic Space Hierarchy Theorem*

Let  $s_i : N \rightarrow [0, +\infty) = R_+ \cup \{0\}$ ,  $i = 1, 2$ , be such  $s_2$  is space-constructible and  $s_2 \in \omega(s_1)$ . Then, we have the strict inclusion:

$$NSPACE[s_1] \subset NSPACE[s_2]$$

*Non – deterministic Time Hierarchy Theorem*

Let  $t_i : N \rightarrow [0, +\infty) = R_+ \cup \{0\}$ ,  $i = 1, 2$ , be such  $t_2$  is time-constructible and  $t_2 \in \omega(t_1(n+1))$ . Then, we have the strict inclusion:

$$NTIME[t_1] \subset NTIME[t_2]$$

where  $NTIME[t(n)]$  will be the class of decidable languages by a  $NTM$  in time  $O[t(n)]$ . And  $NSPACE[s(n)]$ , the class of decidable languages by a  $NTM$  in space  $O[s(n)]$ .

From such classes, we can construct:

$$NP \equiv \cup_{c>0} NTIME[n^c]$$

as the class of languages with short efficiently verifiable membership proofs. As typical  $NP$  languages, we can show the known  $ISO$  and  $SAT$ .

*Open questions?* Certainly; for instance:

- the subexponential membership proofs to the closure of  $SAT$ .
- $NP \neq coNP$ . That is, do not exist polynomial size proofs for  $coNP$  languages.  
Such conjecture is harder than the classical:  $NP \neq P$ .
- Whether we refine the sequence of inclusions, until to reach:

$$L \subseteq NL \subseteq P \subseteq NP \subseteq PSPACE \subseteq EXP \subseteq NEXP \subseteq EXPSPACE$$

some of such inclusions are strict, but not all. Only is conjectured where is  $\neq$ .

#### 4. HIERARCHY THEOREMS (ALTERNATION CASE)

An  $ATM$  is a generalization of a  $NTM$ , intended to capture the complexity of certain languages. These can be defined by quantified logical formulas, where alternate existential ( $\exists$ ) and universal ( $\forall$ ) quantifiers. Remember that in  $NTM$  we can only handle formulas with  $\exists$ . The use of  $ATM$  is for decision problems. They possess an additional resource: the number of changes, in the machine, from  $\exists$  to  $\forall$ , and vice versa. into a computational path. Such change or switch is called an alternation.

*Alternating Space Hierarchy Theorem*

Let  $s_i : N \rightarrow [0, +\infty) = R_+ \cup \{0\}$ ,  $i = 1, 2$ , be such  $s_2$  is space constructible and  $s_2 \in \omega(s_1)$ . Then, we have the strict inclusion:

$$ASPACE[s_1] \subset ASPACE[s_2]$$

*Alternating Time Hierarchy Theorem*

Let  $t_i : N \rightarrow [0, +\infty) = R_+ \cup \{0\}$ ,  $i = 1, 2$ , be such  $t_2$  is time constructible and  $t_2 \in \omega(t_1)$ . Then, we have the strict inclusion:

$$ATIME[t_1] \subset ATIME[t_2]$$

**Final note**

We could introduce the logically equivalent structure of circuits, with their corresponding gates. This can be for the next paper, with some more advanced questions.

REFERENCES

- [1] Ambos-Spies, K., et al., "Genericity and Measure for exponential time", *Theoretical Computer Science*, 168 (1), 1996, pp 3-19.
- [2] Fortnow, L., "Counting complexity", *Complexity Theory Retrospective II*, Springer-Verlag, 1997, pp 81-107.
- [3] Garrido, A.: "Complexity of Algorithms", *Proceed. of Eps-MsO*, Athens, 2005.
- [4] Lutz, J., "The quantitative structure of exponential time", *Complexity Theory Retrospective II*, Springer-Verlag, 1997, pp 225-260.
- [5] Zák, S., "A Turing machine time hierarchy", *Theoretical Computer Science*, 26, 1983, pp 327-33.

Ángel Garrido Bullón  
Department of Fundamental Mathematics.  
Faculty of Sciences UNED Senda del Rey, 9.  
28040-Madrid (Spain)  
email: *algbmv@telefonica.net*