COMPUTATIONAL COMPLEXITY AND GRAPHS

Angel Garrido

Abstract. As we know, there exist many interesting mathematical problems in Theoretical Computer Science, as the comparison of characters between strings, or the acceptation / decision of languages by Turing Machines. Also others, as the "halting problem" or the "travel salesman problem", are undecidable. And many other remain open, only conjectured.

We can establish some results, known as *Hierarchy Theorems*. For instance, the famous classes P and NP are the first two in the tower of complexity classes, if we consider the polynomial-time hierarchy.

And we know that the proof of the conjectured inclusion: $NP \subset P$, because the relation $P \subset NP$ is obvious, results the greatest open problem in actual Mathematics. But it could be solved by a counterexample.

The Hierarchy Theorems constitutes the result of successive attempts to clarify the relation between both classes: P and NP. We will consider many other classes, modulating our requirements.

There you have a very interesting challenge for the human mind in our time, as were the old famous Hilbert's Problems for the Mathematics in the past Century.

So, we consider here some approximations to the relation between the classes P and NP. Our purpose is to contribute to clarify this question.

2000 Mathematics Subject Classification: 03D10, 68Q15, 68R10, 05C50.

Keywords and phrases: Turing Machines, Complexity Classes (hierarchies), Computability and Complexity, Graph Theory.

1. Deterministic Case

The resources measure will be considered as a function of the instance size. And we look at worst-case scenarios, as upper bound.

Given an input size, n, *the amount of resources needed could be the maximum,* over all inputs, i, of size n, of the amount of resources needed for instance i. Such *size* of the instance, i, is the length of the string that encodes it. We denote the size of i as: $|i|$.

We define a *resource bound* as the monotone nondecreasing function:

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

And we can admit more natural problem specific measures, for the input size, than its length. For instance, the number of nodes (or vertices). We can consider only the visited nodes, or all in general, when we work with a forest (unconnected graphs), a tree (connected graphs), or simply, a particular graph.

Frequently, we find infinite computation problems. Then, we will be interested in the resources needed, and how they grow when the input size increases.

The *more usual Resource Bounds* posses their proper name.

So:

$$\begin{array}{ccc} n^{O(1)} & \leftrightarrow & \textit{polynomial} \\ 2^{n^{O(1)}} & \leftrightarrow & \textit{exponential} \\ 2^{n^{o(1)}} & \leftrightarrow & \textit{subexponential} \\ O(\log n) & \leftrightarrow & \textit{logarithmic} \end{array}$$

Now, we will establish previously the bounds of allotted space and time. For this, we have these tools: *space-constructible and time-constructible functions.*

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:

$$DSPACE\left[s\left(n\right)\right]$$

This permits two initial Hierarchy Theorems:

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

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

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

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

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

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

For instance, the known problems of Graph Connectivity and Digraph Critical Path.

We denote L the class of all all decision problems that can be solved in space:

$$O\left(\log n\right)$$

From these, we can also define other space-bounded complexity classes, as:

$$PSPACE \equiv \cup_{c\succ 0} DSPACE\left[n^c\right] \; EXPSPACE \equiv \cup_{c\succ 0} DSPACE\left[2^{n^c}\right]$$

Almost all the aforementioned decision problems are examples of membership to $PSPACE$. But not in the case of Generalized Checkers, which is in $EXPSPACE$.

From the Space Hierarchy Theorem, we can deduce that:

$$L \subset PSPACE \subset EXPSPACE$$

And from the Time Hierarchy Theorem, we can find:

$$P \subset E \subset EXP$$

about the Converse, we know that:

$$DSPACE\left[s\left(n\right)\right] \subseteq \cup_{c\succ 0} DTIME\left[2^{c \cdot s(n)}\right]$$

true for any space constructible function $s\left(n\right)$, such that: $s\left(n\right) \geq \log\, n$.

Also we know, as a particular case, that:

$$L \subseteq P \subseteq PSPACE \subseteq EXP \subseteq EXPSPACE$$

some of these inclusions must be strict, but such strictness is still only conjectured.

2. 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, it 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.

Also we can view, in a NTM, *a computation as a path,* in its configuration directed graph (digraph).

Its vertices or nodes will be all possible configurations of the digraph.

Whereas the links, or arcs between them, correspond to the transitions amongst configurations.

The non-determinism character refers to the possibility of choice between some different alternatives, according with the information provided by the prover.

Apart from the aforementioned resources, of space and time, we may consider, for instance, the amount of non-determinism the TM needs.

The *hierarchy theorems* are now:

Space Hierarchy Theorem. *Let* $s_i : N \to [0, +\infty) = R_+ \cup \{0\}$, $i = 1$, 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* $t_i : N \to [0, +\infty) = R_+ \cup \{0\}$, $i = 1$, 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 languages which are decidable by a* NTM *in time* $O[t(n)]$.

And $NSPACE[s(n)]$, the class of languages which are decidable by a NTM in space $O[s(n)]$.

From such classes, we construct:

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

as the class of languages with short efficiently verifiable membership proofs.

As typical *NP languages*, we can show the known *ISO* and *SAT*.

Relative to *ISO*, an isomorphism is equivalent to an efficiently verifiable short proof.

And respect to *SAT*, a satisfying truth-value assignment. The *SAT language* solves the satisfiability problem. Remember the equivalence between solving a problem and deciding a language.

The *SAT* is one of hardest languages, into the class NP.

Also, in this step, we would define the *non-deterministic corresponding classes:*

$$NE \equiv \cup_{c \succ 0} NTIME\left[2^{c(n)}\right] \; NEXP \equiv \cup_{c \succ 0} NTIME\left[2^{n^c}\right]$$

From the Hierarchy Theorems, we have:

$$NP \subset NE \subset NEXP$$

It is also true that:

$$DTIME\left[t\left(n\right)\right] \subset NTIME\left[t\left(n\right)\right]$$

that is, *the nondeterministic class contains its deterministic counterpart.*

What happens to the converse inclusion? We know that:

$$NTIME\left[t\left(n\right)\right] \subseteq \cup_{c\succ 0} DTIME\left[2^{c\cdot t(n)}\right]$$

Into the deterministic time-bounded classes, the smallest known containing NP is EXP.

Many more questions, and very relevant ones, remain in progressive advance, as whether the non-deterministic classes are closed under complementarity.

Because of temporal limits, we can give an interesting result:

$$coNTIME\left[t\left(n\right)\right] \subseteq \cup_{c\succ 0} NTIME\left[2^{c\cdot t(n)}\right]$$

3. Open Question

Certainly, many questions remain open.

For instance:

· the subexponential membership proofs of the closure of SAT.

· $NP \neq coNP$. It does not exist polynomial size proofs for $coNP$ languages.

Such conjecture is harder than the classical: $NP \neq P$.

· If we refine the sequence of inclusions, until reaching:

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

some of such inclusions are strict, but not all.

Only is conjectured where is $\neq$.

4. Alternating Case

It is also very interesting the case of an *alternating TM*.

The *alternating TM*, ATM, is a generalization of the non-deterministic TM, to control the complexity of languages like the Graph Ramsey Triple.

We introduce now the class of languages accepted by an ATM in polynomial time, with no more than k alternations and an existential initial state, denoted:Σ_{k+1}^{p}.

And also the analogous definition, but with a universal, instead of existential, initial state, as: Π_{k+1}^{p}.

Such classes form two hierarchies, inter-related, with the properties:

$$\Sigma_{k}^{p} = co\ \Pi_{k}^{p}$$

$$\Sigma_{k}^{p} \cup \Pi_{k}^{p} \subseteq \Sigma_{k+1}^{p} \cap \Pi_{k+1}^{p}$$

and if:

$$\Sigma_{k}^{p} = \Pi_{k}^{p}$$

we obtain:

$$\cup_{l} \Sigma_{l}^{p} = \Sigma_{k}^{p}$$

From these last two properties, we have that the subsequent hierarchy of classes:

$$\Sigma_{0}^{p} \subseteq \Sigma_{1}^{p} \subseteq \Sigma_{2}^{p} \subseteq \ldots \subseteq \Sigma_{k}^{p} \subseteq \Sigma_{k+1}^{p} \subseteq \ldots \qquad (\circ)$$

has the upward *collapse property*:

If two subsequent levels coincide: $\Sigma_{k}^{p} = \Sigma_{k+1}^{p}$, then the whole hierarchy collapses to such level:

$$\Sigma_{i}^{p} = \Sigma_{k}^{p}, \forall i \in N^{*} = N \cup \{0\}$$

The aforementioned hierarchy , as $(\circ)$, is called *Polynomial Time Hierarchy*. The union of the classes in such hierarchy is denoted by PH.

But if we use exponential-time ATM, instead of polynomial-time ones, we similarly obtain the *Exponential Time Hierarchy*. It is denoted by $EXPH$. We have:

$$EXPH = \Sigma_k^{\exp} \cup \Pi_k^{\exp}$$

Observe that:

$$\begin{gathered}\Sigma_0^p = \Pi_0^p = P \\ \Sigma_1^p = NP \\ \Pi_1^p = coNP\end{gathered}$$

In a similar way:

$$\begin{gathered}\Sigma_0^{\exp} = \Pi_0^{\exp} = EXP \\ \Sigma_1^{EXP} = NEXP \\ \Pi_1^{\exp} = coNEXP\end{gathered}$$

It is conjectured that one more alternation perhaps permits to decide more languages, within the same bounds.

In particular, it can be conjectured that the polynomial size hierarchy does not collapse. An assumption stronger than the aforementioned: $NP \neq coNP$.

If the equality holds, then such hierarchy collapses at its first level.

Finally, we will formulate the analogous results, for the case of *alternation*:

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

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

5. Final Note

For temporal reasons, we do not introduce the logically equivalent structure of circuits, with their gates. Interesting issue, because gives access to the simulation of nondeterministic time bounded computations by small circuits.

A possible question: Whether non-deterministic polynomial time has polynomial size circuits or not?

Our answer must be that such situation is not possible, unless the polynomial size hierarchy collapses to its second level.

As you know:

$$\textit{If } NP \subseteq P/poly \Rightarrow \Sigma_2^p = \Pi_2^p$$

With super-polynomial time and a pair of alternations, the polynomial size surely is not sufficient.

Through the same technique, we can see that exponential-time with a couple of alternations requires circuits of exponential-size.

References

[1] Ambos-Spies, K., et al.: *Genericity and Measure for exponential time*, *Theoretical Computer Science*, 168 (1): 3-19, 1996.

[2] Borodin, A.: *On relating time and space to size and depth*, *SIAM Journal on Computing,* 6: 733-744, 1977.

[3] Buhrman, H., et al.: *Separating complexity classes using autoreductibility*, *SIAM Journal on Computing,* 29(5): 1497-1520, 2000.

[4] Fortnow, L.: *Counting complexity*, in L. Hemaspaandra and A. Selman editors, *Complexity Theory Retrospective II,* pp. 81-107, Springer-Verlag, 1997.

[5] Garrido, A.: *Complexity of Algorithms*, *Intern. Conf. Eps-MsO*, Vol. I, Patras Publ., Athens, 2005.

[6] Garrido, A.: *Analysis of Complexity*, *Intern. Conf. IC-SCCE,* Vol. I, Patras Publ., Athens, 2006.

[7] Garrido, A.: *Improving the Complexity Classes*, *Acta Universitas Apulensis (AUA)*, Vol. N. ^{o} 10, pp. 19-22, Alba Iulia University, Editura Aeternitas, Cluj-Napoca, 2006.

[8] Stockmeyer, L.: *The polynomial-time hierarchy*, *Theoretical Computer Science,* 3: pp. 1-22, 1976.

Author:

Ángel Garrido Bullón
Department of Fundamental Mathematics
National University of Distance Learning (UNED)
Senda del Rey, 9, 28040-Madrid, Spain
email: algbmv@telefonica.net