



HAL
open science

Method for classification of the computational problems on the basis of the multifractal division of the complexity classes

Artem Potebnia

► **To cite this version:**

Artem Potebnia. Method for classification of the computational problems on the basis of the multifractal division of the complexity classes. 3rd International Scientific-Practical Conference "Problems of Infocommunications Science and Technology" (PIC S&T 2016), IEEE, Oct 2016, Kharkiv, Ukraine. pp.1 - 4, 10.1109/INFOCOMMST.2016.7905318 . hal-01592383

HAL Id: hal-01592383

<https://hal.archives-ouvertes.fr/hal-01592383>

Submitted on 23 Sep 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Method for Classification of the Computational Problems on the Basis of the Multifractal Division of the Complexity Classes

Artem Potebnia

Computer Engineering Department
Kyiv National Taras Shevchenko University
Kyiv, Ukraine
admin@artem.bz.ua

Abstract — This paper proposes the method of the multifractal division of the computational complexity classes, which is formalized by introducing the special equivalence relations on these classes. Exposing the self-similarity properties of the complexity classes structure, this method allows performing the accurate classification of the problems and demonstrates the capability of adaptation to the new advances in the computational complexity theory.

Keywords — computational problem; P and NP classes; reduction; equivalence relation; multifractal

I. INTRODUCTION

In order to form the classification of the computational problems, they are grouped into separate classes from the perspective of their complexity. The problems belonging to each class have the same type (decision, function, counting, etc.), identical computation model and similar requirements to time and space resources [1]. The most important complexity classes are indicated based on the following concept of reduction:

Definition 1. The reduction $A \leq_{poly} B$ of the computational problem A to another problem B is presented by transformations of the problem instances and solutions, respectively denoted by f and h , which have the polynomial time complexity.

Let us consider the NP class, which contains the decision problems whose solutions can be verified in polynomial time. Being a preorder on this class, the reduction relation \leq_{poly} allows to indicate inside it the NP -complete problems that are the most difficult to solve. On the contrary, the class $P \subseteq NP$ covers all problems that can be solved in the polynomial time using the deterministic Turing machine. Nevertheless, the graph isomorphism and factoring problems have not been included in class P or set of NP -complete problems and thereby their accurate classification under this approach remains impossible.

Moreover, the closer inspection of the P class shows that it similarly contains the P -complete problems and the separate class NC , which respectively encompass the hardest and easiest problems according to the criterion of the parallel feasibility. Additionally, the P class also contains some problems that are not proven to be either P -complete or included in the NC class. In particular, one

example is the greatest common divisor problem represented in the decision form.

Therefore, the internal structure of the P class is similar to the structure of the wider NP class. This observation demonstrates the self-similarity feature of the nested complexity classes structure. Based on these considerations, this paper proposes the new method for the proper classification of the computational problems by means of the multifractal division of the complexity classes.

Within the deflation approach, the fractal division is constructed by the iterative application of the generator, which represents the decomposition rule, to the specified initiator and its smaller copies. The well-known illustrations of the fractal division are the pinwheel, half-hex, sphinx and many others. For example, in the case of the pinwheel division, the initiator is represented by the right-angled triangle with legs in a ratio of 1:2. The generator performs the decomposition of such triangle into five homothetic triangles that all are smaller copies of the initiator [2].

Notice that the multifractal division is more general than the regular one, because it allows the application of multiple generators. The remainder of this paper focuses on the investigation of the division iteration that carries out the detailing of the internal structure of the P class.

II. INTERNAL STRUCTURE OF THE P CLASS

As a warm-up observation, note that the problems belonging to the P class have different suitability for the development of the parallel solving algorithms. In particular, the problems that are fully deprived of the natural parallelism, being inherently sequential, are considered as P -complete. On the contrary, the special class $NC \subseteq P$ encompasses the problems that allow the efficient parallel implementation of the solving algorithms. However, the problem of establishing the exact relation between the NC and P classes is still open, although the assumption $NC \subset P$ is the most common. The principal requirement for the problems belonging to the NC class is formulated as the opportunity to develop the parallel solving algorithms that achieve the polylogarithmic time complexity $O(\log^k n)$ using $O(n^c)$ parallel processors, where k and c are some constants, while n is the size of input parameters [3].

Notice that the following formal definition of the NC class uses the concept of a Boolean circuit in order to identify the property of the natural parallelism in the algorithms.

Definition 2. The complexity class NC is specified as a family of subsets NC^k , i.e.

$$NC = \bigcup_{k=0}^{\infty} NC^k.$$

Each subset NC^k covers the problems decidable by the Boolean circuits of the depth $O(\log^k n)$, which are composed of the polynomial number of gates $n^{O(1)}$ having at most two inputs.

Since the NC class is the subset of the P class, it is fully composed only of the decision problems. In this regard, its function extension $FNC \subseteq FP$ is specified in order to encompass the function problems that have the high parallel feasibility. Note that the formal criteria for the problem inclusion in the NC and FNC classes are the same. The only difference between the problems belonging to these classes occurs only in the form of the solution representation. In particular, the circuits corresponding to the FNC class problems have multiple outputs in contrast to the circuits for the NC class problems, which are limited to the presence of only one output. All the problems within the NC class are linked by the special NC -reductions in order to determine their relative parallel feasibility.

Definition 3. The NC -reduction of the decision problem A to another decision problem B is denoted by $A \leq_{poly}^{NC} B$ and represents the special case of the general reduction relation where the mapping f could be computed in the polylogarithmic time using the polynomial number of processors, i.e. $f \in FNC$.

Similar to the general reduction relation, the NC -reduction is a preorder on the P class, which means that a pair (P, \leq_{poly}^{NC}) is a preordered set of problems.

Establishing the relation $A \leq_{poly}^{NC} B$ imposes the following simple restrictions on the inclusion of the A and B problems in the NC class: If $B \in NC$, then $A \in NC$ and if $A \notin NC$, then $B \notin NC$.

III. IDENTIFICATION OF THE P-COMPLETE PROBLEMS BASED ON THE NC-REDUCTION RELATION

Definition 4. The problem L is considered P -complete if it belongs to the P class and is linked to all other problems $L' \in P$ by the NC -reduction $L' \leq_{poly}^{NC} L$.

The process of solving the P -complete problems is associated with the presence of the fundamental data dependencies (such as the read-after-write (RAW), write-after-read (WAR), write-after-write (WAW) situations), which results in an inability to develop the efficient parallel algorithms. Most commonly, the P -completeness of the problem $L \in P$ is determined by specifying the relation $L^* \leq_{poly}^{NC} L$ to the known P -complete problem L^* .

In the paper [4], Ladner has proven that the Circuit Value Problem (CVP) serves as one of the basic P -complete problems, similarly to the satisfiability problem (SAT) in the NP -completeness theory. Instances of the CVP problem are presented in the form of sequences $C = \langle C_0, \dots, C_n \rangle$ composed of the input values $C_i = 0$, $C_i = 1$ and the basis functions $C_i = \overline{C}_j$, $C_i = C_j \wedge C_k$, $C_i = C_j \vee C_k$, applied to the previous elements of C , i.e. $j, k < i$. At the same time, the element C_n designates the output gate of the circuit C , while its calculated value represents the solution of the CVP problem.

Moreover, the CVP problem has several variations, such as TopCVP, MCVP, NANDCVP, NORCVP and PCVP, which also are P -complete and differ in the restrictions imposed on the construction of the problem instances C . Together, these variations provide a powerful basis for establishing the reductions to all other problems.

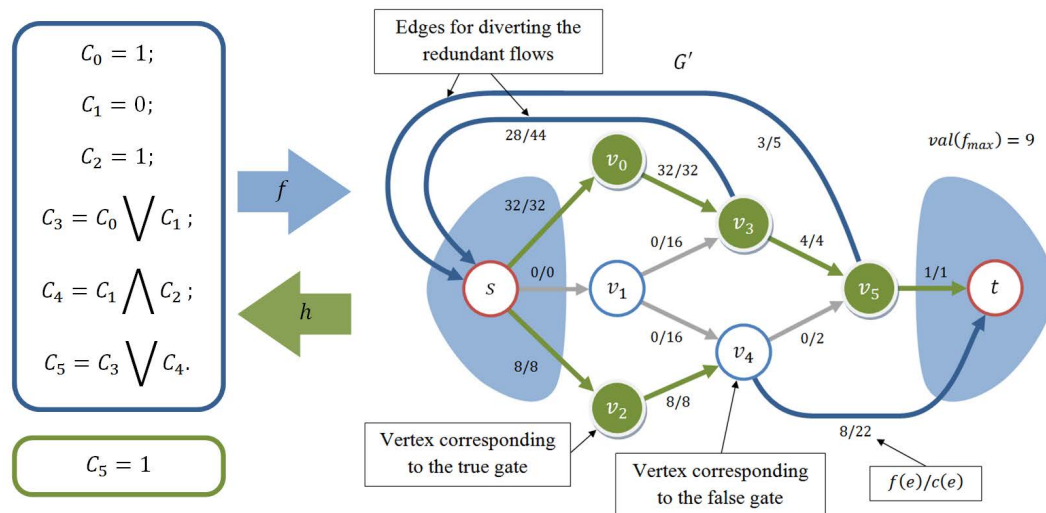


Fig. 1. Example of the NC -reduction $MCVP \leq_{poly}^{NC} MFP^d$

As an illustration, let us show the P -completeness of the maximum flow problem MFP^d (formulated in the decision form as the determination of the flow value parity) by specifying the reduction $MCVP \leq_{poly}^{NC} MFP^d$. Instances of the MCVP problem are represented by the monotone circuits C constructed over the incomplete basis $\{\wedge, \vee\}$ without application of the negation elements $C_i = \overline{C_j}$. The transformation of instances f converts such monotone circuits C into networks $G' = (V', E')$ composed of the vertices $V' = \{v_0, v_1, \dots, v_n\} \cup \{s, t\}$, where s and t designate, respectively, the source and sink nodes of the network. The edges of the network $e' \in E'$ and their capacity values $c(e')$ are formed by implementing the following operations:

1. The input gates $C_i = 1$ of the circuit instance C are reflected in the structure of G' by attaching the arcs (s, v_i) with capacities $c(s, v_i) = 2^{n-j}$.
2. The network is expanded by adding the edge (v_j, v_i) with $c(v_j, v_i) = 2^{n-j}$ for every argument C_j of the function implemented by the C_i element.
3. The network vertex v_n that corresponds to the output gate of the circuit is linked to the sink node by attaching the arc (v_n, t) with $c(v_n, t) = 1$.
4. For each gate $C_i = C_j \wedge C_k$, the network structure is modified by forming edges (v_i, t) , whose capacity values are limited only by the condition of the flows conservation.
5. In a similar manner, the elements $C_i = C_j \vee C_k$ are reflected in the network structure by adding the backward connections to the source node (v_i, s) .

Notice that the edges (v_i, t) and (v_i, s) excepting (v_n, t) serve for diverting the redundant flows and balancing the overall network. Moreover, the described algorithm f performs the construction of the network G' by processing the circuit elements in the independent manner, which means that $f \in FNC$. Let us combine the intermediate vertices of the network $V = V' \setminus \{s, t\}$ into the T and F subsets such that the nodes $v \in T$ correspond to the true gates of the circuit, while the vertices $v \in F$ correspond to the false ones. The network G' holds the standard flow formed by filling the flows passing through the edges (v_i, v) up to their capacity and assigning zero values to the flows that correspond to the edges (v_f, v) , where $v_t \in T$, $v_f \in F$ and $v \in V$. In addition, the specification of the standard flow requires setting $f(v_n, t) = 1$ if $C_n = 1$.

In order to explore in detail the standard flow, let us consider the (s, t) -cut $\langle T \cup \{s\}, F \cup \{t\} \rangle$ of the network G' . Note that the overall flow in such cut is equal to its

total capacity, because only completely filled flows pass through the specified partition. This means that the standard flow is the maximum in the network G' . Excluding the arcs used for balancing the network, only flow passing through the edge (v_n, t) can have even or odd value depending on the output of the element C_n . This observation provides the basis for establishing the mapping h between the solutions of the CVP and MFP^d problems. Fig. 1 demonstrates the example of the reduction $MCVP \leq_{poly}^{NC} MFP^d$, which clearly illustrates all above considerations.

IV. DIVISION OF THE COMPLEXITY CLASSES INTO THE EQUIVALENCE CLASSES

The specification of both $A \leq_{poly}^{NC} B$ and $B \leq_{poly}^{NC} A$ reductions means that the A and B problems are equivalent under the \leq_{poly}^{NC} relation and have the same parallel feasibility in terms of the NC class theory. In order to describe such case in detail, let us introduce the special subrelation $\sim_{poly}^{NC} \subseteq \leq_{poly}^{NC}$.

Definition 5. The binary relation \sim_{poly}^{NC} over the P class is defined as the intersection of the original \leq_{poly}^{NC} relation with the inverse relation $\leq_{poly}^{NC -1}$, i.e.

$$\sim_{poly}^{NC} = \leq_{poly}^{NC} \cap \leq_{poly}^{NC -1}$$

In this expression, the relation resulting from the operation of the relations intersection is composed of all pairs of problems belonging to both relations that are the arguments of the operation, i.e.

$$\sim_{poly}^{NC} = \left\{ (A, B) \mid (A, B) \in \leq_{poly}^{NC} \text{ and } (A, B) \in \leq_{poly}^{NC -1} \right\}$$

The inverse relation, in turn, is obtained from the original one by switching the order of problems in each pair, i.e.

$$\leq_{poly}^{NC -1} = \left\{ (A, B) \mid (B, A) \in \leq_{poly}^{NC} \right\}$$

Theorem 1. The relation \sim_{poly}^{NC} is an equivalence relation on the P class.

▲ In order to prove this theorem, we need to show that the relation \sim_{poly}^{NC} holds the properties of the reflexivity, symmetry, and transitivity. Above all, notice that the properties of the reflexivity and transitivity are stable under the operation of the inverse relation formation. This implies that both \leq_{poly}^{NC} and $\leq_{poly}^{NC -1}$ relations are reflexive and transitive and, consequently, are preorders on the P class. The following part of the proof is organized into three stages, which show that the intersection \sim_{poly}^{NC} preserves the reflexivity and transitivity properties of both relations and additionally is symmetric.

Since the relations \leq_{poly}^{NC} and $\leq_{poly}^{NC^{-1}}$ are reflexive, it follows that $(A, A) \in \leq_{poly}^{NC}$ and $(A, A) \in \leq_{poly}^{NC^{-1}}$ for any problem $A \in P$. Then, by the definition of the intersection, we have that $(A, A) \in \sim_{poly}^{NC}$, which means that the relation \sim_{poly}^{NC} is reflexive.

Assume there exist problems A, B and C such that $(A, B) \in \sim_{poly}^{NC}$ and $(B, C) \in \sim_{poly}^{NC}$. From the definition of the intersection, we receive that $(A, B) \in \leq_{poly}^{NC}$, $(B, C) \in \leq_{poly}^{NC}$, $(A, B) \in \leq_{poly}^{NC^{-1}}$ and $(B, C) \in \leq_{poly}^{NC^{-1}}$.

Moreover, $(A, C) \in \leq_{poly}^{NC}$ and $(A, C) \in \leq_{poly}^{NC^{-1}}$, because both relations are transitive. Finally, applying the definition of the intersection again, we obtain that $(A, C) \in \sim_{poly}^{NC}$. This clearly shows the transitivity of the \sim_{poly}^{NC} relation.

Suppose that $(A, B) \in \sim_{poly}^{NC}$ for some A and B problems. As in the previous stage, considering the definition of the intersection, we have that $(A, B) \in \leq_{poly}^{NC}$ and $(A, B) \in \leq_{poly}^{NC^{-1}}$. Applying the operation of the inverse relation formation, we receive that $(B, A) \in \leq_{poly}^{NC^{-1}}$ and $(B, A) \in \left(\leq_{poly}^{NC^{-1}} \right)^{-1}$. Obviously, the last expression is equivalent to $(B, A) \in \leq_{poly}^{NC}$. Therefore, from the definition of the intersection, we obtain that $(B, A) \in \sim_{poly}^{NC}$, which shows that the relation \sim_{poly}^{NC} is symmetric and finally completes the proof. ▼

In the light of the above theorem, we can conclude that the relation \sim_{poly}^{NC} splits the P class into a collection of disjoint equivalence classes, forming a quotient set

$$P / \sim_{poly}^{NC} = \left\{ [E]_{\sim_{poly}^{NC}} \mid E \in P \right\}.$$

Here each equivalence class $[E]_{\sim_{poly}^{NC}}$ is denoted by some representative problem E and covers all problems that are related to E by \sim_{poly}^{NC} . Therefore, the problems that have the same parallel feasibility in terms of the NC class theory are grouped in one equivalence class. Fig. 2 presents the simplified example of splitting the sample class P containing five problems into the equivalence classes of the subrelation \sim_{poly}^{NC} on the basis of the pre-defined relation \leq_{poly}^{NC} .

In a similar manner, we can specify the equivalence relation $\sim_{poly} = \leq_{poly} \cap \leq_{poly}^{-1}$ over the wider NP class, thereby forming the corresponding quotient set

NP / \sim_{poly} . Therefore, we can conclude that based on the set NP / \sim_{poly} , the generator performs the formation of the subsequent iteration of the fractal division by replacing the equivalence class $[E]_{\sim_{poly}^{NC}}$ that contains the easiest problems with the quotient set P / \sim_{poly}^{NC} .

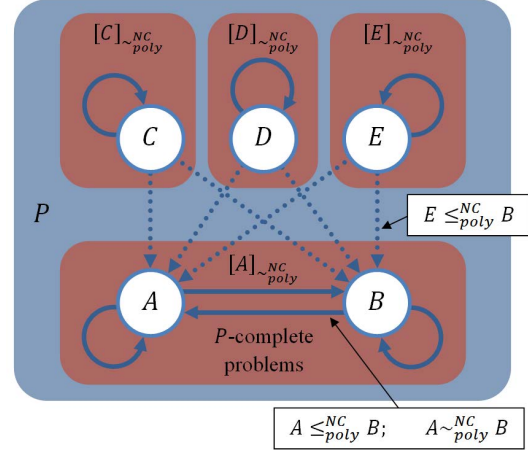


Fig. 2. Simplified example of specifying the collection of the equivalence classes inside the sample class P

V. CONCLUSIONS

The method for classification of the computational problems based on the equivalence classes of the relations \sim_{poly}^{NC} and \sim_{poly} allows for a much more detailed description of the internal structure of the complexity classes. In particular, such approach is sufficiently flexible to address the situations where $NC = P$ (P / \sim_{poly}^{NC} contains just one equivalence class) or $NC \subset P$ (in this case, the P class can be divided into any number of subsets by introducing the corresponding equivalence classes).

REFERENCES

- [1] A. Potebnia and S. Pogorilyy. "Investigation of transformations and landscapes for combinatorial optimization problems," in *2016 13th International Conference on Modern Problems of Radio Engineering, Telecommunications and Computer Science (TCSET)*, pp.510-514, 23-26 Feb. 2016.
- [2] M. Senechal. *Quasicrystals and Geometry*. New York: Cambridge University Press, 1996.
- [3] A. Potebnia and S. Pogorilyy. "Fast output-sensitive approach for minimum convex hulls formation," in *Recent Advances in Computational Optimization: Results of the Workshop on Computational Optimization WCO 2015*, Vol. 655. Cham: Springer International Publishing, 2016, pp.1-20.
- [4] R. Ladner. "The circuit value problem is log space complete for P," *SIGACT News*, vol. 7, pp.18-20, 1975.