

Computational Requirements for Nano-machines

Melanie Badura

Universität zu Lübeck

Lübeck

melanie.badura@student.uni-luebeck.de

ABSTRACT

This paper is a short paper for "Computational Requirements for Nano-Machines: There is limited Space at the bottom".[1]

CCS CONCEPTS

• **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

KEYWORDS

ACM proceedings

Machine	Problems		Origin
AC^0 :	<i>ADD</i>	<i>ODD/EVEN</i>	
	<i>SUB</i>	<i>DIV₂</i>	
	<i>SIGN</i>	<i>MOD₂</i>	
	<i>INC</i>	<i>LOG₂</i>	
	<i>AND/OR</i>	<i>INV</i>	
NC^1 :	<i>MULT</i>	<i>MIN/MAX</i>	
	<i>DIV</i>	<i>PARITY</i>	
	<i>EXP</i>	<i>REG</i>	
	<i>MAJOR</i>	<i>MOD</i>	
	<i>THRES</i>	<i>AVG</i>	
L :	<i>Label</i>	<i>D_{FS}</i>	
	<i>Logmem</i>	<i>B_{FS}</i>	
	<i>REACH</i>	<i>MEDIAN</i>	

Table 1: Complexity classes of nano-machines

ACM Reference Format:

Melanie Badura. 2021. Computational Requirements for Nano-machines. In *Proceedings of AzuNet Seminar (CR for Nano'21)*. ACM, New York, NY, USA, 2 pages. https://doi.org/10.475/123_4

1 INTRODUCTION

For years, there has been talking of using nano-machines to create solutions in medicine and other subjects. Such a machine should be able to communicate and sense/act. Computational power is also a big issue. Because Nano-machines are small, one question is how to implement these capabilities. While many researchers already deal with communication technology (cite one here), computational capability is usually left out. In this paper, there is an attempt to provide a general analysis of the computational capability of nano-machines. Since the capabilities of nano-machines vary widely, from nanoparticles with no computational capabilities to microprocessors (11 cites). Nano-machines divide into three groups according to complexity theory, by analyzing the tasks that nano-machines handle.

2 MITTELTEIL?

Most problems that a nano-machine has to solve need basic arithmetic. Pattern matching and parity solve other problems. Protocols, such as forwarding and routing, solve the communication problem. These protocols are solved in different ways so that a nano-machine can handle several types of messages. Communication is a perfect example for the use of basic arithmetics, pattern matching, storage needs. They need memory for the storage of routing information and for other values with which they must work. Nano-machines should also be able to perform more complex operations like implementing a neural network. They use graph algorithms for this.

All these problems divide into complexity classes. As the name suggests, the problems divide into classes that have

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

CR for Nano'21, July 2021, Lübeck, Germany

© 2021 Copyright held by the owner/author(s).

ACM ISBN 123-4567-24-567/08/06.

https://doi.org/10.475/123_4

approximately the same complexity. The reduction of operations chooses the complexity class. For example, subtract can be reduced to add. A nano-machine that solves the problems of the highest complexity class can also use the operations of the lower classes. However, this does not work the other way around. In table 1, the three different classes are seen. An L-machine can solve problems like a Turing machine. The class AC^0 describes boolean circuits with polynomial-size and a constant depth, whereas the class NC^1 may have a logarithmic depth and two inputs per gate.

Problems can lower their class if researchers can find new reductions for them. However, problems exist that need more powerful machines or global knowledge about the network. Thus problems, like addressing, routing broadcasting/forwarding, could not be divided into the three complexity classes.

Insights into the feasibility of Nanao-machines provides the classification of the applications according to the computational power. Researchers may consult Table 1 to implement a protocol or an algorithm in a nano-machine. The algorithm must decompose into its basic operations. Afterward, the operation that is the most complex gives the decision for a complexity class. Just because a nano-machine may perform only a few easy operations does not mean that it is simple to build, but the choice of complexity class can help.

Considering the AC^0 class, if a nano-machine can add, it may do all other operations of the complexity class. Let us take the complexity class NC^1 for a practical example. In the body, a nano-machine observes the concentration of a marker with *AVG*. Since *THRES* is also in the same class, this nano-machine can also check if a defined value exceeds. This without increasing the complexity nor the memory.

Falls du auf zu wenig zeilen kommst, ein reduction beispiel auf jeden fall machen. Und vllt nochmal auf origin eingehen.

3 CONCLUSION

This paper tries to close the gap of the insufficient specified computational requirements. It does so by analyzing different medical scenarios. (I don't really do that) The analyzed problems divide into three complexity classes: AC^0, NC^1, L . A nano-machine can then solve all the problems of his class and the class problems under him.

However, additional components, such as storage or clocking, may mean that the classes do not represent the actual implementation of the nano-machines. Furthermore, we have not addressed the combination of operations, so the power of the complexity class may exceed. In addition, the definition of additional complexity classes may specify the operations in more detail.

REFERENCES

- [1] Florian-Lennert Adrian Lau, Florian Büther, and Bennet Gerlach. 2017. Computational requirements for nano-machines: there is limited space at the bottom. In *Proceedings of the 4th ACM International Conference on Nanoscale Computing and Communication*. 1–6.