# DESCRIPTIVE COMPLEXITY: ITS RELATIONSHIP WITH NP VS. CO-NP

Anthony Widjaja To [1]

[1] Department of Computer Science and Software Engineering,
Department of Mathematics & Statistics,
The University of Melbourne

TUTORIAL ABSTRACT

Computational complexity characterizes a problem in terms of the computational resources — such as time and space — required to compute the problem. On the other hand, descriptive complexity measures a problem in terms of the logical resources — such as the number of variables, and the types of quantifiers — required to describe the problem. It turns out that these two areas are intimately connected. For example, the complexity classes NP and co-NP are exactly the sets of finite "relational structures" describable in, respectively, existential second-order logic (ESO) and universal second-order logic (ASO). In turn, this yields a logical correspondence of the NP vs. co-NP open question: NP = co-NP iff ESO = ASO. Such a correspondence allows one to tap into tools from logic in order to resolve the NP vs. co-NP question. Moreover, since it is widely conjectured that NP does not equal co-NP, these logical characterizations may also lead to a separation of the complexity classes P and NP, and hence resolving arguably the most important open problem in computer science and contemporary mathematics.

In this tutorial, I plan to introduce the audience to the notions of "a problem describable in a logic" and "a complexity class captured by a logic". I will, then, define what we mean by ESO and ASO, and discuss the proof that NP and co-NP are captured by, respectively, ESO and ASO. We will see that this statement implies that NP = co-NP iff ESO = ASO. Finally, I will briefly discuss an approach toward separating ESO and ASO. It is assumed that the audience are familiar with the basic concepts from complexity theory such as Turing machines, nondeterminism, computational problems (i.e., languages), and the complexity classes P, NP, and co-NP. Although it is an advantage, the reader is not assumed to know the required notions from mathematical logic — such as first-order formulas, structures, and satisfaction (|=) — as they will be briefly reviewed.