Journal of Computer and System Sciences 82 (2016) 1268–1282

Contents lists available at ScienceDirect

Journal of Computer and System Sciences

www.elsevier.com/locate/jcss

Generic case completeness ☆

Alexei Miasnikov, Alexander Ushakov *

Department of Mathematics, Stevens Institute of Technology, Hoboken, NJ, USA

ARTICLE INFO

Article history:
Received 1 September 2014
Received in revised form 13 May 2016
Accepted 13 May 2016
Available online 7 June 2016

Keywords:
Generic-case complexity
Completeness
Randomized problems
Bounded halting problem

ABSTRACT

In this note we introduce a notion of a generically (strongly generically) **NP**-complete problem and show that the randomized bounded version of the halting problem is strongly generically **NP**-complete.

Contents

☆ The work was partially supported by NSF grant DMS-1318716.
* Corresponding author.
E-mail addresses: amiasnik@stevens.edu (A. Miasnikov), aushakov@stevens.edu (A. Ushakov).

http://dx.doi.org/10.1016/j.jcss.2016.05.002
0022-0000/

1. Introduction

We introduce and study problems that are generically in **NP**, i.e., decision problems that have partial errorless nondeterministic decision algorithms that solve the problem in polynomial time on "most" inputs. We define appropriate reductions in this class and show that there are some complete problems there, called *strongly generically* **NP**-*complete* problems. In particular, the randomized bounded version of the halting problem is one of them.

Rigorous formulation of notions of generic algorithms and generic complexity appeared first in group theory [17,18] as a response to several challenges that algorithmic algebra faced at that time. First, it was well understood that many hard, even undecidable, algorithmic problems in groups can be easily solved on most instances (see [17,18,8,21] for a thorough discussion). Second, the study of random objects and generic properties of objects has become the mainstream of geometric group theory, following the lead of graph and number theory (see [9–11,23,1,4,3]). It turned out that "random", "typical" objects have many nice properties that lead to simple and efficient algorithms. However a rigorous formalization of this approach was lagging behind. Algorithmic algebra was still focusing mostly on the worst-case complexity with minor inroads into average case complexity. Third, with the rapid development of algebraic cryptography the quest for natural algorithmic problems, which are hard on most inputs, became one of the main subjects in complexity theory (see discussion in [21]). It was realized that the average case complexity does not fit well here. Indeed, by definition, one cannot consider average case complexity of undecidable problems, which are in the majority in group theory; the proofs of average case results are usually difficult and technical [12,25], and, most importantly, there are problems that are provably hard on average but easy on most inputs (see [8,21] for details). In fact, Gurevich showed in [12] that the average case complexity is not about "most" or "typical" instances, but that it grasps the notion of "trade-off" between the time of computation on hard inputs and how many of such hard instances are there. Nowadays, generic algorithms form an organic part of computational algebra and play an essential role in practical computations.

In a surprising twist generic algorithms and ideas of generic complexity were recently adopted in abstract computability (recursion theory). There is interesting and active research there concerning absolutely undecidable problems, generic Turing degrees, coarse computability, etc., relating generic computation with deep structural properties of Turing degrees [20,16,2, 14,6,5].

We decided to relativize these ideas to lower complexity classes. Here we consider the class **NP**. Motivation to study generically hardest problems in the class **NP** comes from several areas of mathematics and computer science. First, as we have mentioned above, average case complexity, even when it is high, does not give information on the hardness of the problem at hand on the typical or generic inputs. Therefore, to study hardness of the problem on most inputs one needs to develop a theory of generically complete problems in the class **NP**. This is interesting in its own right, especially when much of activity in modern mathematics focuses on generic properties of mathematical objects and how to deal with them. On the other hand, in modern cryptography, there is a quest for cryptoprimitives which are computationally hard to break on most inputs. It would be interesting to analyze which **NP**-problems are hard on most inputs, i.e., which of them are generically **NP**-complete. Note, there are **NP**-complete problems that are generically polynomial [21]. All this requires a robust theory of generic **NP**-completeness. As the first attempt to develop such a theory we study here the class of all generically **NP**-problems, their reductions, and the complete problems in the class. Most of the time, our exposition follows the seminal Gurevich's paper [12] on average complexity. We conclude with several open problems that seem to be important for the theory.

Here we briefly describe the structure of the paper and mention the main results. In Section 2, we recall some notions and introduce notation from the classical decision problems. In Section 3, we discuss distributional decision problems (when the set of instances of the problem comes equipped with some measure), then define the generic complexity and problems decidable generically (strongly generically) in polynomial time. In Section 4, we define generic polynomial time reductions. In Section 5, we show that the distributional bounded halting problem for Turing machines is strongly generically **NP**-complete. Notice that though generic Ptime randomized algorithms are usually much easy to come up with (than say Ptime on average algorithms), the reductions in the class of generic **NP**-problems are still as technical as reductions in the class of **NP**-problems on average. In fact, the reductions in both classes are similar. Essentially, these are reductions among general randomized problems and the main technical, as well as theoretical, difficulty concerns the transfer of the measure when reducing one randomized problem to another one. It seems this difficulty is intrinsic to reductions in randomized computations and does not depend on whether we consider generic or average complexity. In Section 6 we discuss some open problems that seem to be important for the development of the theory of generic **NP**-completeness.

2. Preliminaries

In this section we introduce notation to follow throughout the paper.

2.1. Decision problems

Informally, a *decision problem* is an arbitrary yes-or-no question for an (infinite) set of *inputs* (or *instances*) I, i.e., an unary predicate P on I. The problem is termed *decidable* if P is computable, and the main classical question is whether a given problem is decidable or not. In complexity theory the predicate P usually is given by its true set $L = \{x \in I \mid P(x) = 1\}$,