

Title	Computational Complexity of Puzzles and Games (Invited Talk)
Author(s)	Uehara, Ryuhei
Citation	Lecture Notes in Computer Science, 9135: XXVI
Issue Date	2015-06
Type	Journal Article
Text version	author
URL	http://hdl.handle.net/10119/15306
Rights	This is the author-created version of Springer, Ryuhei Uehara, Lecture Notes in Computer Science, 9135, 2015, XXVI. The original publication is available at www.springerlink.com , http://dx.doi.org/10.1007/978-3-662-47666-6
Description	42nd International Colloquium, ICALP 2015, Kyoto, Japan, July 6-10, 2015, Proceedings, Part II

Computational Complexity of Puzzles and Games

Ryuhei Uehara

School of Information Science
Japan Advanced Institute of Science and Technology, Japan
uehara@jaist.ac.jp

A computation consists of algorithm of basic operations. When you consider an algorithm, you assume, say, the standard RAM model, that has “usual” arithmetic operations. On the other hand, when you consider an algorithm on a DNA computer, your basic operations are duplication and inversion on a string. Then you need to consider completely different algorithms, and their computational complexity also changes. That is, when we discuss computational complexity of a problem, it strongly depends on the set of basic operations you use. When you enjoy a puzzle, you have to find an algorithm by combining reasonable basic operations to its goal. (Some puzzles require to find the basic operations themselves, but we do not consider such puzzles in this talk.) From the viewpoint of theoretical computer science, puzzles give us some insight to computation and computational complexity classes in various way.

Some puzzles and games give reasonable characterizations to computational complexity classes. For example, “pebble game” is a classic model that gives some complexity classes in a natural way, and “constraint logic” is recent model that succeeds to solve a long standing open problem due to Martin Gardner that asks the computational complexity of sliding block puzzles. Such puzzles gives us “typical” and characterization and “intuitive” understanding for some computational complexity classes.

On the other hand, there are some puzzles and games that give nontrivial interesting aspects of computational complexity classes. For example, consider “14-15 puzzle” which is classic well known sliding puzzle. By parity, we can determine if one arrangement can be slid to the other in linear time. Moreover, we can always find a way for sliding between them in quadratic time. However, interestingly, finding the optimal solution is NP-complete in general. I also introduce a relatively new notion of the reconfiguration problem. This series of new problems will give some new notion of computational complexity classes.