Department of Mathematics
Spring 2023: Math 6720 Topics in Logic (CRN 68213)

Turing Degrees and Applications

MW 2:20–3:35pm
Monroe Hall (2115 G Street, Room 250)

Prof. Valentina Harizanov
harizanv@gwu.edu
https://home.gwu.edu/~harizanv/

Description. Computability theory is the mathematical theory of algorithms (or decision procedures), which explores power and limitations of computation. Classical computability theory formalized the intuitive notion of an algorithm and provided a theoretical basis for digital computers. It also demonstrated the limitations of algorithms and showed that most sets of natural numbers and problems they encode are not decidable (Turing computable). Important results of modern computability theory include the classification of the computational difficulty of sets and problems.

The Turing degree provides an important measure of the level of such difficulty. There are uncountably many Turing degrees and they are partially ordered. We will study the structure of Turing degrees. It is an upper semilattice but not a lattice, and there are minimal degrees. The main methods of computability theory that we plan to use are various coding techniques, finite and infinite extension methods, and the priority methods. We use a priority argument when we want to satisfy mutually conflicting requirements by fitting together opposite strategies. We first decompose such requirements into infinitely many simpler sub-requirements, assign priorities to the sub-requirements, and then attempt to satisfy them in order of their priority, possibly injuring some lower priority ones.

We will further use Turing degrees to investigate the complexity of problems in modern computable mathematics, for example, of structures and their isomorphism types.

Textbook

R. Soare, *Turing Computability* (*Theory and Applications*), Springer, 2016.

V. Harizanov, Computability-theoretic complexity of countable structures, *Bulletin of Symbolic Logic* 8 (2002), pp. 457–477.

Other course material from various sources will be provided in class.

Required background
MATH 2971 or an equivalent, or CS 2312 or CS 3313 or an equivalent. Familiarity with the notion of an algorithm.

Math 6720 can be taken for credit repeatedly. Advanced undergraduate students may also take this course for credit.

Grading
Based on homework assignments, take-home exams, projects, and their presentations.