Questions for EZQuiz 6

- State our textbook's version of the Turing thesis. Solution: "Any computation that can be carried out by mechanical means can be performed by some Turing machine." (p244)
- Define *algorithm*. Solution: A function f has an algorithm if there exists a Turing machine that halts with tape contents $f(w)$ when given input w, for all w in the domain of f. [This is a simplified version of definition 9.5, page 246.]
- List five different types of Turing machines that have equivalent computational power. Solution: single semi-infinite tape, single infinite tape, multi-track, multi-tape, nondeterministic.
- Define *universal Turing machine*. Solution: "an automaton that, given as input a description of any Turing machine M and a string w, can simulate the computation of M on w." (p266)
- Give a definition of *recursively enumerable languages*. Solution: definition 11.1, page 276.
- Give a definition of *recursive languages*. Solution: definition 11.2, page 276.
- Give a concrete description of a language that is recursively enumerable, but not recursive. Solution: Let $M_1, M_2, \ldots$ be an enumeration of Turing machines with the input alphabet $\{a\}$. Let L be the language that contains a^i if and only if $a^i \in L(M_i)$. Then it can be shown that L is recursively enumerable, but not recursive. [This language is used in the proofs of theorems 11.3 and 11.5, pp279–281.]
- State the relationship between recursively enumerable languages and grammars. Solution: Every unrestricted grammar generates an r. e. language, and every r. e. language is generated by an unrestricted grammar. [This combines theorem 11.6 (p283) and theorem 11.7 (p287).]
- Prove that the halting problem is undecidable. Solution: Suppose not. Then there exists a Turing machine H that solves the halting problem. Specifically, H outputs "y" on input $w_M w$ if M halts on input w, and otherwise outputs "n". By making some simple changes to H, we can produce D, which enters an infinite loop on input w_M if M halts on input w_M, and otherwise halts. Consider the behavior of D on input w_D. By the definition of D, this computation halts if and only if it does not halt. This contradiction tells us that neither D nor H can exist, proving the desired result. [This is an abbreviated and altered version of the proof of theorem 12.1, page 300.]
- Give two examples of practical problems that are undecidable (the halting problem does not count). Solution: (i) determining whether programs can crash; (ii) determining whether context-free grammars are ambiguous.