CS1200: Intro. to Algorithms and their Limitations	Anshu & Vadhan

Sender–Receiver Exercise 7: Reading for Receivers

Harvard SEAS - Fall 2024 *Nov. 21, 2024*

The goals of this exercise are:

- to develop your skills at understanding, distilling, and communicating proofs and the conceptual ideas in them,
- to practice reductions for proving unsolvability, and gain more intuition for what kinds of problems about programs are unsolvable.

Sections 1 and 3 are also in the reading for receivers. Your goal will be to communicate the *proof* of Theorem 1.1 (i.e. the content of Section 2) to the receivers. Section 3 contains questions for you and your receiver to think about if you finish the exercise early; there is no need to prepare anything in advance for that.

1 The Result

In class, we mentioned Rice's Theorem, which says that all nontrivial problems about the input–output behavior of programs (i.e. about the program's semantics) are unsolvable.

Here we will see an example of a computational problem that is not about the input–output behavior of programs but is nevertheless unsolvable:

Input	: A RAM program P
Output	: `yes` if P has running time $O(n)$, `no` otherwise

Computational Problem IsLinearTime

The statement "P has running time $O(n)$" means that there are constants c and n_0 such that for all $n \geq n_0$ and all inputs x of length at most n, $P(x)$ halts within $c \cdot n$ steps. Note that the constants c and n_0 are allowed to depend on P.

Theorem 1.1. *IsLinearTime is unsolvable.*

2 The Proof

The approach to proving unsolvability.

The reduction.

Correctness of the reduction.

3 Food for Thought

If you and your partner(s) finish early, here are some additional questions or issues you can think about:

1. We constructed Q_P so that Q_P has running time $O(n)$ if and only if P halts on ε. Can you think of how to construct Q_P so that Q_P has running time $O(n)$ if and only if P *doesn't* halt on ε? If you used such a construction, how would the reduction A change?

2. So far, our intuition for unsolvability has been that it comes from the possibility that RAM programs don't halt. However, the programs Q_P constructed in the above reduction always halt in time $O(n^2)$. Thus, the same reduction proves unsolvability of the following variant of IsLinearTime, where we *promise* that the input program halts in time $O(n^2)$:

Input	**:** A RAM program P with running time $O(n^2)$
Output	**:** `yes` if P has running time $O(n)$, `no` otherwise

Computational Problem IsLinearTimePromise

Try to develop some of your own intuition for what makes a problem like this, on always-halting programs, unsolvable.