Lectures and Reading. Whereas fine machine details of DFAs and NFAs are enduringly important, those of Turing machines will matter only (a) to establish the equivalence of TMs and high-level programming languages (ultimately through the “Universal RAM Simulator” hand-out https://cse.buffalo.edu/~regan/cse396/UTMRAMsimulator.pdf) and later (b) to discuss differences between linear and quadratic time. The notes by Debray recognize this by progressing swiftly to emphasize the language-class concepts, for which there are numerous synonyms and usage differences to note:

- **Decidable** and **recursive** are synonyms to describe the class of languages \( L = L(M) \) for deterministic Turing machines that are **total**, i.e., that halt for all inputs. The class is usually called **REC** but I may also write **DEC**

- The languages \( L(M) \) “of” or “accepted by” Turing machines \( M < ( \text{that need not be total} ) \) are synonymously called all of the following: **recursively enumerable** (r.e.), **computably enumerable** (c.e.), **Turing-acceptable**, and **(Turing-)recognizable**. The term “r.e.” is historical since the concept originally came from systems of recursion proved equivalent to TMs, but “c.e.” is gaining favor. The class, however, is only called **RE**.

- The complement of an r.e. language is called **co-r.e.** or **co-c.e.** and the class of them is denoted by **co-RE**. Note that the classes are “inclusive downward,” so both **RE** and **co-RE** have all the decidable languages as well as “their own” undecidable ones.

The other difference from standard courses is that lectures will define the basic complexity concepts of time and space usage right away. So besides reading up through section 8 of the notes, please jump ahead to section 13 in “Part 3” (pages 39 bottom through 43 in the second batch of Debray’s notes given out), and also the Allender-Loui-Regan Chapter 27 notes through page 6, in both cases stopping before the language SAT is introduced. Most in particular, I will define mapping reductions \( \leq_m \) and polynomial-time mapping reductions \( \leq^p_m \), mainly because virtually all intro examples of the former are actually the latter. This will continue into weeks 6 and 7; the use of reductions for undecidability proofs will not be on the first exam since it is not on this assignment.

Assignment 3 (really “4”), due Wed. 10/3 in class

(1) Define \( L = \{ 0^k10^m10^k : k, m \geq 1 \} \). Note the resemblance to the language called “\( L'_\infty \)” in the Wed. 9/19 lecture.

(a) **Sketch** the design of a 2-tape Turing machine \( M \) that accepts \( L \) in time \( O(n) \). Note that the length \( n = |x| \) equals \( 2k + m + 2 \) when \( x \in L \) and that the number \( m \) of zeroes in the middle doesn’t really matter provided there is at least one. Here “sketch” means to give enough machine detail so that the correctness of the machine and its running in linear time are evident; you may but need not give a complete arc-node diagram (with comments).
(b) Describe in words a single-tape TM $M'$ that accepts $L$ in time $O(n \log n)$. **Hint:** Whatever the number $k$ of 0s before the first 1 (if any) in a given input $x$, we know $k \leq n = |x|$. Thus we can count up to $n$ using a “cartouche” of $O(\log n)$ bits. Treat the tape as having two “tracks” where the second track is used to maintain and move the “cartouche” alongside the reading of $x$ like a caterpillar. Here an arc-node diagram would be complicated and crufty, so use accurate words.

(c) Show that $L$ cannot be recognized by a 1-tape TM $M'' = (Q, \Sigma, \Gamma, \delta, \ldots)$ in $O(n)$ time. This is stronger than saying $L$ is not regular. It suffices to consider the behavior of $M''$ on inputs $x$ of the form $0^k10^k10^\ell$, that is, where the “$m$” part happens to equal $k$ (though $M''$ need not care about that) but the $\ell$ might be anywhere from $k$ to (say) $2k$. The upshot is that if $M''$ runs within $cn$ time for some $c$, it can cross the middle $0^k$ “desert” at most $4c$ times (or at most $2c$ times if you count going back-and-forth). Argue as best you can—formal use of Kolmogorov complexity is optional as I intend to skim/skip section 12—that since the only information $M''$ can transport across the desert each time is $O(\log |Q| + |\Gamma|) = O(1)$ bits, it cannot distinguish among $\Theta(k) = \Theta(n)$ possible values for $\ell$ in finitely many crossings. (The grading will not be strict here; this is mainly a teaching tool to convey how Myhill-Nerode and streaming ideas run up against information complexity. 18 + 18 + 18 = 54 pts.)

(2) Let $D$ be any undecidable language over $\Sigma = \{0, 1\}$ (or any alphabet $\Sigma$ you like; it doesn’t matter). Define languages $A$ and $B$ such that:

- $A \cap B$ is finite.
- $A \cup B = \Sigma^*$.
- $A \cdot B = \Sigma^*$.
- $A$ and $B$ are undecidable—in fact, $A \equiv_m D$ and $B \equiv_m \tilde{D}$.

Note how this means the union, intersection, and concatenation of undecidable languages can not only be decidable, they can be regular. (18 pts., for 72 total on the set)

---

1 The word “cartouche” refers to how hieroglyphs naming Egyptian royals were drawn with a box around them.