

15-251: Great Theoretical Ideas In Computer Science

Recitation 3

Announcements

Be sure to take advantage of the following resources :

- Homework Solution Sessions - Saturday and Sunday 2:30p-3:30p in GHC 4101
- We will be releasing our grading rubrics to make the learning objectives/takeaways of homework problems explicit.

Too Many Definitions

- Informally, a Turing machine is a machine with a finite set of states, a tape (memory) that is infinite in one direction that can process inputs over some alphabet. At each step, the machine makes the following decisions (based on the state it is in and the symbol it's tape-head is currently reading) : move to some state, write some symbol at the current cell currently under the tape head, and move the tape head to the left or to the right.
- Formally, we define a Turing machine to be a 7-tuple $(Q, q_0, q_{\text{accept}}, q_{\text{reject}}, \Sigma, \Gamma, \delta)$, where Q is the set of states, q_0 is the start state, q_{accept} and q_{reject} are the final states, Σ is the input alphabet, $\Gamma \supseteq \Sigma \cup \{\sqcup\}$ is the tape alphabet, and $\delta : Q' \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$, where $Q' = Q \setminus \{q_{\text{accept}}, q_{\text{reject}}\}$ is the transition function.
- A Turing machine is called a decider if for all inputs $x \in \Sigma^*$, it halts and either accepts or rejects x .
- A language $L \subseteq \Sigma^*$ is called decidable if there exists a decider Turing machine M such that $L = L(M)$.
- Let L and K be languages, where K is decidable. We say that solving L reduces to solving K (or simply, L reduces to K , denoted $L \leq K$), if we can decide L by using a decider for K as a subroutine (helper function).

Closure Ceremony

Suppose that L_1 and L_2 are decidable languages. Show that the languages $L_1 \cdot L_2$ and L_1^* are decidable as well.¹

Freeze All Automata Functions

Prove that the following languages are decidable by reducing it to **EMPTY_{DFA}**.

(a) **NO – ODD – ONES** = $\{\langle D \rangle : D \text{ does not accept any string containing an odd number of 1's}\}$

(b) **INF_{DFA}** = $\{\langle D \rangle : D \text{ is a DFA with } L(D) \text{ infinite}\}$.

Hint: Consider a DFA with k states that accepts some string with more than k characters.

Not Just Your Regular Old TM

Suppose we change the definition of a TM so that the transition function has the form

$$\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{R, S\}$$

where the meaning of S is “stay”. That is, at each step, the tape head can move one cell to the right or stay in the same position. Suppose M is a TM of this new kind, and suppose also that M is a decider. Show that $L(M)$ is a regular language.

(Extra) Only \$19.99! Call now!

Dr. Hyper Turing Machines Inc LLC is selling a whole host of new Turing machines, each for \$19.99:

- Bi-infinite TMs - with a tape that stretches infinitely in both directions!
- Infinitely-scalable TMs - choose however many tapes heads you like!
- Quad-core TMs - now with 4 tapes (each with its own tape head)

Normal TMs usually go for \$9.99 these days. Your friend (who's not very Turing-savvy) is in the market for a new Turing machine and just texted you asking you for purchasing advice. Your instincts tell you that maybe most of this is marketing hype. But some of those improvements do sound pretty compelling... Your friend doesn't use their TM for all that much - mostly just browsing the web and checking email. What should you recommend them to do?

(Bonus) Tick Tock Clock

Write a Turing Machine that does the following : given an input string $s \in \{0, 1\}^*$, the Turing machine should finish with a binary representation of $|s|$ on the tape (and nothing else). The TM should run in time at most $c_1 |s| \log |s| + c_2$ steps, where c_1, c_2 are some constants.

¹Exercise : show that $L_1 \cup L_2$ and $L_1 \cap L_2$ are also decidable.