CMSC 28100

Introduction to Complexity Theory

Spring 2024
Instructor: William Hoza

Problem set 1

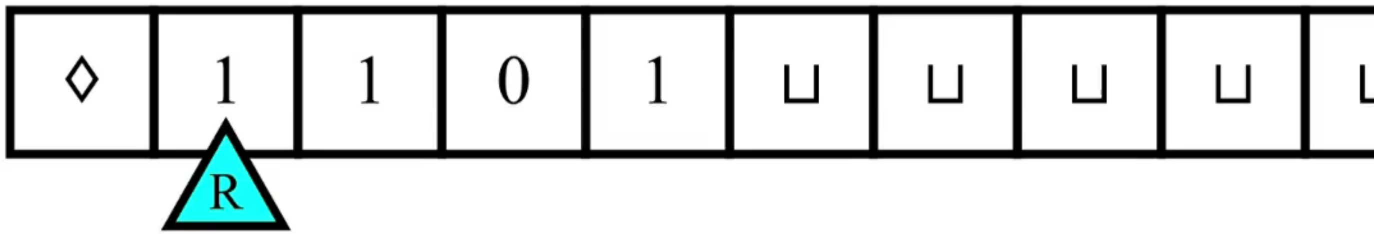
- Problem set 1 is available in Canvas
- If you aren’t officially enrolled in the course, send me an email. I’ll add you to Canvas so you can access the homework
- Office hours (Thursday, Friday, Monday) are a good place to find study partners / homework collaborators

Which problems

can be solved

through computation?

Turing machines



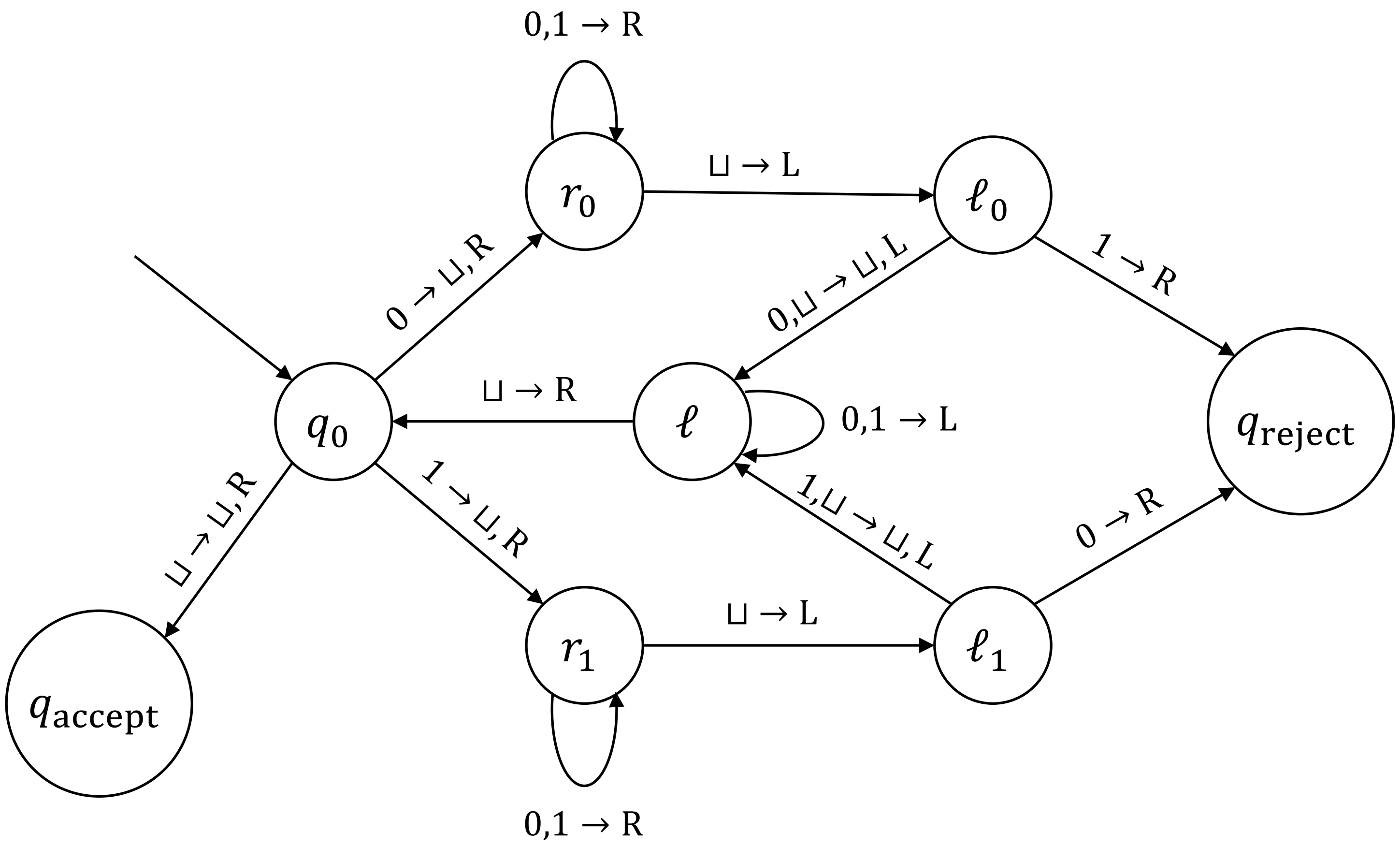
- In each step, the machine decides
  - What to write
  - Which direction to move the head (left or right)
  - The new state
- The decision is based only on the current state and the observed symbol

Defining Turing machines rigorously

- **Def**: A Turing machine is a 9-tuple $M = (Q, \Sigma, \Gamma, \Diamond, \sqcup, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})$ such that
  - Q is a finite set (the set of "states")
  - Σ and Γ are alphabets (the "input alphabet" and the "tape alphabet")
  - We have $\Sigma \cup \{\Diamond, \sqcup\} \subseteq \Gamma$ and $\sqcup, \Diamond \notin \Sigma$
  - δ is a function $\delta: Q \times \Gamma \to Q \times \Gamma \times \{\mathrm{L}, \mathrm{R}\}$ (the "transition function")
  - If $\delta(q, \Diamond) = (q', b', D)$, then $b' = \Diamond$ and $D = \mathrm{R}$
  - If $\delta(q, b) = (q', b', D)$ and $b \neq \Diamond$, then $b' \neq \Diamond$
  - $q_0, q_{\text{accept}}, q_{\text{reject}} \in Q$ and $q_{\text{accept}} \neq q_{\text{reject}}$.

⚠ Warning: The definition in the textbook is slightly different. Sorry! (The two models are equivalent.)

State diagram



- Each node represents a state
- An arc from q to q' labeled "$b \to b', D$" means $\delta(q, b) = (q', b, D)$
- The label "$b \to D$" is shorthand for "$b \to b, D$"
- An arc labeled "$a, b \to \cdots$" represents two arcs ("$a \to \cdots$" and "$b \to \cdots$")

Defining TM computation rigorously

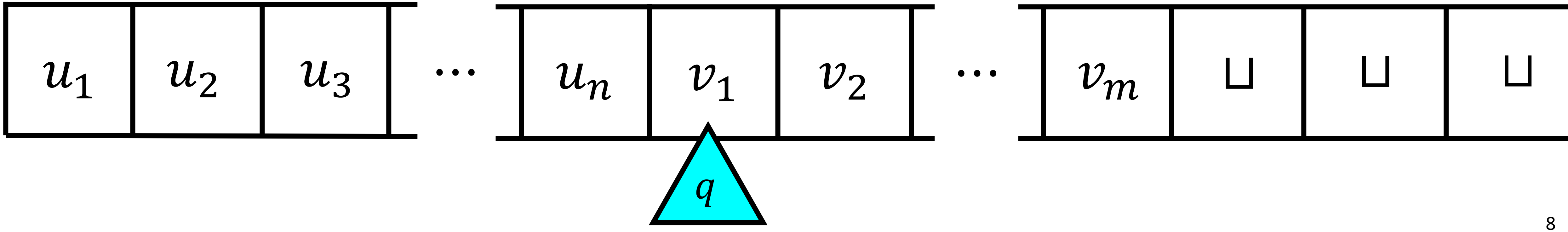
- The transition function δ describes the local evolution of the computation
- Now let's precisely describe the global evolution of the computation

Configurations of a Turing machine

- Let $M = (Q, \Sigma, \Gamma, \diamondsuit, \sqcup, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})$ be a Turing machine
- A configuration of M is a triple (u, q, v) where $u \in \Gamma^*$, $q \in Q$, and $v \in \Gamma^*$.

  Interpretation:

  - The tape currently contains $uv \sqcup\sqcup\sqcup\sqcup \cdots$
  - The machine is currently in state q and the head is currently located in cell $|u| + 1$



Configuration shorthand

- Instead of (u, q, v), we often write uqv
- We think of uqv as a string over the alphabet $\Gamma \cup Q$
- This shorthand can only be used if $Q \cap \Gamma = \emptyset$, which we can assume without loss of generality by renaming states if necessary

Equivalent configurations

- Note: uqv and $uqv\ \sqcup$ are technically two distinct configurations…
- However, they represent the exact same scenario
- We can say that they are "equivalent"
- (A configuration is a finite string, even though the tape is infinitely long)

The initial configuration

- Let $w \in \Sigma^*$ be an input
- The initial configuration of M on w is $\Diamond q_0 w$

The “next” configuration

- Let uqv be any configuration of M such that uv begins with $\diamondsuit$
- We define $\mathrm{NEXT}(uqv)$ as follows:
  - Break uqv into individual symbols: $uqv = u_1 u_2 \dots u_{n-1} u_n q v_1 v_2 v_3 \dots v_m$
  - Let b be the symbol that M is “currently observing”
    - $b = v_1$, unless $m = 0$, in which case $b = \sqcup$
  - If $\delta(q, b) = (q', b', \mathrm{R})$, then $\mathrm{NEXT}(uqv) = u_1 u_2 \dots u_{n-1} u_n b' q' v_2 v_3 \dots v_m$
  - If $\delta(q, b) = (q', b', \mathrm{L})$, then $\mathrm{NEXT}(uqv) = u_1 u_2 \dots u_{n-1} q' u_n b' v_2 v_3 \dots v_m$
    - This is well-defined ($u \neq \epsilon$), because M must move right if $b = \diamondsuit$

Halting configurations

- An accepting configuration is a configuration of the form $uq_{\text{accept}}v$
- A rejecting configuration is a configuration of the form $uq_{\text{reject}}v$
- A halting configuration is an accepting or rejecting configuration

Computation history

- Let $w \in \Sigma^*$ be an input
- Let C_0 be the initial configuration of M on w, i.e., $C_0 = \Diamond q_0 w$
- Inductively, for each $i \in \mathbb{N}$, let $C_{i+1} = \mathrm{NEXT}(C_i)$
- The computation history of M on w is the sequence $C_0, C_1, \ldots, C_T$, where C_T is the first halting configuration in the sequence
- If there is no such C_T, then the computation history is $C_0, C_1, C_2, \ldots$ (infinite)

Accepting, rejecting, and looping

- If the computation history of M on w ends with an accepting configuration, then we say that M accepts w
- If the computation history of M on w ends with a rejecting configuration, then we say that M rejects w
- In either of those cases, we say that M halts on w. If the computation history of M on w is infinite, then we say that M loops on w

Time

- Suppose the computation history of M on w is $C_0, C_1, \dots, C_T$
- We say that T is the running time of M on w
- If M loops on w, then its running time on w is ∞
- We say that M halts on w within T steps if the running time of M on w is at most T

Space

Which of the following statements is <u>false</u>?

A: Space used $\leq$ running time + 2

B: If M halts on w within $|w|$ steps, then M halts on ww

C: If M halts on w, then M uses a finite amount of space on w

D: If M uses a finite amount of space on w, then M halts on w

Respond at PollEv.com/whoza or text “whoza” to 22333

- Let $C_0, C_1, \ldots$ be the (finite o
- Write $C_i = (u_i, q_i, v_i)$ where $u_i \in \Gamma^*$, $q_i \in Q$, $v_i \in \Gamma^*$
- The space used by M on w is $\max_i |u_i| + 1$, i.e., it’s the maximum S such that during the computation of M on w, the head visits cell S
- (Can be ∞)