Lecture #7: Modelling Computation by Humans — Turing Machines

Exercises and Review

Questions for Review

1. A ***Turing machine*** was formally defined as a 6-tuple $\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{halt}})$.
   (a) What is Q here?
   (b) What is Σ here?
   (c) What is Γ here? How are Σ and Γ related?
   (d) What is the "blank" symbol, $\sqcup$? How is it related to Σ and Γ?
   (e) What is δ here?
   (f) What are q_0 and q_{halt} here? How are they related to Q?
2. Describe the way that ***information*** is stored and accessed when a Turing machine is used.
3. What is a ***configuration*** of a Turing machine M? What information does this represent or provide? How is this represented as a string?
4. Describe the ***start configuration*** of a Turing machine $\mathcal{M}$, for inputs strings $\omega_1, \omega_2, \ldots, \omega_m$, when this machine is being used to compute a function $f : (\Sigma^\star)^m \to \Sigma$.
5. What is a ***halting configuration*** of M? What is a ***non-halting*** configuration?
6. Describe how the ***transition function*** δ is applied to determine which configuration should follow another (non-halting) one.
7. Describe, as precisely as you can, what it means for a Turing machine M to ***compute*** a (partial or total) function $f : (\Sigma^\star)^m \to \Sigma^\star$. What does it mean for a (partial or total) function $f : (\Sigma^\star)^m \to \Sigma^\star$ to be ***Turing-computable***?
8. What does it mean for a language $L \subseteq \Sigma^\star$ (for an alphabet Σ) to be ***Turing-decidable***? What does it mean for L to be ***Turing-recognizable***? How are these concepts related?

9. Describe, as precisely as you can, how the notion of "Turing-computability" of a function is related to the other notions of "computability" of functions that have been introduced, in this course, before this.

Practice Problems

These problems — which continue activities in the lecture presentation — will not be discussed during the lecture, and solutions for these problems will not generally be made available. They can be used as "practice" problems that can help you to practice skill considered in the lecture presentation for Lecture #7.

1. Let $f : \mathbb{N}^k \to \mathbb{N}$ be a partial or total function, for a positive integer k. Sketch a proof that the following conditions are equivalent.

   (a) f is ***Turing-computable***, as this is defined in the lecture notes. That is, there exists a positive integer n such that if

   $$\Sigma = \{\sigma_1, \sigma_2, \ldots, \sigma_n\}$$

   is an alphabet with size n then the function $\widehat{f} : (\Sigma^\star)^k \to \Sigma^\star$ is Turing-computable — where $\widehat{f}$ is as follows: For all $x_1, x_2, \ldots, x_k \in \mathbb{N}$, if $\omega_1, \omega_2, \ldots, \omega_k \in \Sigma^\star$ such that $\#(\omega_k) = x_i$ for $1 \leq i \leq k$, then

   - if $f(x_1, x_2, \ldots, x_k)$ is defined then $\widehat{f}(\omega_1, \omega_2, \ldots, \omega_k)$ is the string $\mu \in \Sigma^\star$ such that $\#(\mu) = f(x_1, x_2, \ldots, x_k)$, and
   - if $f(x_1, x_2, \ldots, x_k)$ is undefined then $\widehat{f}(\omega_1, \omega_2, \ldots, \omega_k)$ is also undefined.

   (b) For ***every*** positive integer n, if

   $$\Sigma = \{\sigma_1, \sigma_2, \ldots, \sigma_n\}$$

   is an alphabet with size n then the function $\widehat{f} : (\Sigma^\star)^k \to \Sigma^\star$ is Turing computable — where $\widehat{f}$ is as above.

2. Confirm that the result given in the previous question implies "Claim #3" from the lecture presentation. (This should not be difficult!)

Additional problems concern the completion of proofs of claims that are included in the supplemental document, "Useful Turing Machine Variants", for this lecture.

3. Consider, first, the proof of Claim #1, which assets that if a language L is Turing-recognizable (respectively, Turing-decidable) then there exists a Turing machine with a one-way infinite tape that recognizes (respectively, decides) L.

(a) Describe the ***initialization phase*** in more detail. Your answer should include a picture of the non-blank part of $\widehat{M}$'s tape, at the end of initialization phase, for an input string $\omega = \zeta_1\zeta_2\ldots\zeta_n \in \Sigma^\star$ with length n (for $n \geq 0$), the location of the tape head, and the state that $\widehat{M}$ is in at this point.

(b) Describe the simulation of a single ***move*** of M — specifying how $\widehat{M}$'s tape should be updated and how its state should change. You will need to consider several cases that concern whether M's tape is the the left of its initial position, to the right of its position, or *at* its position — and you will also need to consider simulations of moves in which M's tape head would move left, move right, or stay where it is.

It will probably also be necessary to consider *special cases* — in which the non-blank part of $\widehat{M}$'s tape is increased by one cell, and in which the non-blank part of $\widehat{M}$'s tape is shortened by (at least) on cell.

(c) State a claim — concerning the relationship between configurations of M and $\widehat{M}$, on the same input string $\omega \in \Sigma^\star$ after some number, k, of moves of M have been made, and these moves have been simulated by $\widehat{M}$. The claim should be something that is reasonably easy to establish, if you have confirmed the correctness of your answers for parts (a) and (b) — and it should imply both parts of Claim #1 in the supplemental document.

(d) Describe how this simulation can be modified so that it concerns Turing machines that compute functions (so that it can be used to prove part of Claim #3 in the supplemental document). When answering this part of the question you will need to describe a "cleanup phase" that ends with M's output being written on $\widehat{M}$'s tape in the expected way.

4. Now consider the simulation used prove Claim #2 in the supplemental document.

(a) While the simulation used to prove Claim #2, in the supplement, is more fully described, there is still a bit more to do. State a claim — concerning the relationship between configurations of M and $\widehat{M}$ on the same input string $\omega \in \Sigma^\star$ after some number, k, of moves of M have been made, and these have been simulated by $\widehat{M}$. The claim should be something that is reasonably easy to establish, — and it should imply both parts of Claim #1 in the supplemental document.

(b) Describe how this simulation can be modified so that it concerns Turing machines that compute functions (so that it can be used to prove the rest of Claim #3 in the supplemental document). When answering this part of the question you will need to describe a "cleanup phase" that ends with M's output being written on $\widehat{M}$'s tape in the expected way.

5. Next, consider the simulation used in the proof of Claim #5 in the supplemental document. Suppose that — at the beginning of a move of M — the non-blank part of M's i^{th} tape stores symbols

$$\alpha_{i,\ell_i}, \alpha_{i,\ell_i+1}, \ldots, \alpha_{i,r_i-1}, \alpha_{i,r_i}$$

(in contiguous cells), where $\ell_i \leq r_i$ and the following properties are satisfied:

- Either $\alpha_{i,\ell_i} \neq \sqcup$ or M's i^{th} tape head points to the cell storing α_{i,ℓ_i}.
- Either $\alpha_{i,r_i} \neq \sqcup$ or M's i^{th} tape head points to the cell storing α_{i,r_i}.
- For $\ell_i \leq j \leq r_i$, if $j > 0$ then $\alpha_{i,j}$ is stored in a cell that is j cells to the *right* of the initial position of M's i^{th} tape head. If $j < 0$ then $\alpha_{i,j}$ is stored in a cell that is $|j|$ cells to the *left* of the initial position of M's i^{th} tape head. Finally, if $j = 0$ then $\alpha_{i,j}$ is stored a the initial location of M's i^{th} tape head.

Let $\ell = \min(\ell_1, \ell_2, \ldots, \ell_k)$ and let $r = \max(r_1, r_2, \ldots, r_k)$. For $1 \leq i \leq k$ and $\ell \leq j \leq r$, let $\alpha_{i,j}$ be "$\sqcup$" if either $j < \ell_i$ or $j > r_i$ — so that all symbols "$\alpha_{i,j}$"' are defined in a way that is consistent with the contents of M's tapes.

(a) Using the above notation, as needed, state a claim — corresponding to the relationship between configurations of M and $\widehat{M}$ on the same input string $\omega \in \Sigma^\star$ after some number, t, of moves of M have been made, and these have been simulated by $\widehat{M}$. The claim should be something that is reasonably easily to establish (if steps in the simulation are given in sufficient detail) — and it should imply both parts of Claim #5 in the supplemental document.

(b) Let ℓ and r be as described above. How small can ℓ be, and how large can r be, after M has made t moves?

(c) Describe how this simulation can be modified so that it concerns Turing machines that compute functions (so that it can be used to prove Claim #6 in the supplemental document). When answering this part of the question you will need to describe a "cleanup phase" that ends with M's output being written on $\widehat{M}$'s tape in the expected way.

6. Finally, consider the simulation of a nondeterministic Turing machine used in of the proof of Claim #8 in the supplemental document. Recall that this simulation is implementing a ***breadth-first search*** of the computation tree, so that — while the simulation is in progress — $\widehat{M}$'s Tape #1 will store the sequence of all configurations that are at level t of the computation tree for the input string, for some non-negative integer t.

(a) Describe the ***initialization phase*** for this simulation, when the input is a string $\omega \in \Sigma^\star$. You should find that this is quite simple!

(b) Now consider the simulation of a step of M. This should begin with the contents of $\widehat{M}$'s Tape #1 being copied onto Tape #2, erasing all but the first two symbols, "♠♠",

that are on the first tape — with both tape heads moved back to the third cell on each tape.

Each of the configurations on Tape #2 should now be used to produce zero or more configurations that will be added onto the end of Tape #1. This process will (almost certainly) begin with a sweep to right over the representation of the configuration (which is now on Tape #2) to determine the state, $q \in Q$ that it includes, and the symbol, $\sigma \in \Gamma$, that would be visible on the tape at this point — which can be stored using $\widehat{M}$'s finite control. $\widehat{M}$'s tape head, on Tape #2, could then be moved back to the beginning of the representation of this configuration.

Recall that we are simulating a single ***fixed*** nondeterministic Turing machine M — so that M's possible moves are fixed, and can be used to define $\widehat{M}$'s set of states and transition function). As a result, $\widehat{M}$ can include a fixed "submachine" for each state $q \in Q \setminus \{q_{\text{accept}}, q_{\text{reject}}\}$ and for each symbol $\sigma \in \Gamma$ whose transition $\delta(q, \sigma)$ should be applied. Furthermore, the submachine for q and σ will include a fixed (and finite) number of simpler ***submachines*** — one for each 3-tuple (r, τ, D) (for $r \in Q$, $\tau \in \Gamma$ and $D \in \{\mathtt{L}, \mathtt{R}\}$) that is in the *set* $\delta(q, \sigma)$.

Figure out what these submachines will do! Consider each of the following cases.

(i) $\delta(q, \sigma)$ includes a 3-tuple $(r, \tau, \mathtt{L})$ for $r \in Q$ and $\tau \in \Gamma$.
(ii) $\delta(q, \sigma)$ includes a 3-tuple $(r, \tau, \mathtt{R})$ for $r \in Q$ and $\tau \in \Gamma$.
(iii) $\delta(q, \sigma)$ includes a 3-tuple $(r, \tau, \mathtt{S})$ for $r \in Q$ and $\tau \in \Gamma$.

Note that if the transition $\delta(q, \sigma)$ is now being used then $|\delta(q, \sigma)|$ configurations will be added to the end of Tape #1 to replace the configuration on Tape #2 that is now being processed.

During this process, ***rejecting*** configurations should simply be erased from Tape #2's tape as they are discovered.

If an ***accepting configuration*** is ever discovered then $\widehat{M}$ should ***accept***.

On the other hand, if this has not happened and — at the end of a stage — Tape #1's non-blank portion only stores

♠♠

then $\widehat{M}$ should ***reject*** — because the computation tree was finite, has now been completely searched, and it did not include an accepting configuration.

Otherwise, after all the transitions on Tape #2 have been processed, this tape should be erased (and the tape head should be moved back to the left). $\widehat{M}$' should change state to store M's current state, so that the simulation of the next step can begin.

(c) Since M is a fixed nondeterministic Turing machine there exists a positive constant c (depending only on M) such that

$$|\delta(q, \sigma)| \leq c$$

for every state $q \in Q$ and symbol $\sigma \in \Gamma$. Use this to prove the following

Claim: Let $t \geq 0$. Then, after t stages of the simulation of M, the following properties are satisfied.

(a) The length of each substring representing a configuration of M, on Tape #1, has length at most linear in $\max(n, t)$ if the input string $\omega \in \Sigma^\star$ has length n.

(b) The number of configurations represented on Tape #1 is at most c^t.

(c) The length of the non-blank portion of Tape #1 is at most linear in $\max(n, t) \times c^t$.

(d) Use this to prove the following claim as well.

The number of steps used by $\widehat{M}$ to carry out the t^{th} stage of the simulation is at most linear in $\max(n, t) \times c^t$.

(e) Use the above to prove that if M accepts a string $\omega \in \Sigma^\star$ then $\widehat{M}$ accepts ω as well.

(f) Use the above to prove that if M rejects a string $\omega \in \Sigma^\star$ then $\widehat{M}$ rejects ω too.

(g) Finally, prove that if M loops a string $\omega \in \Sigma^\star$ then $\widehat{M}$ also loops on this string.