CPSC 351 — Tutorial Exercise #7
Simulations and Turing Machine Variants

This exercise is intended to give you practice using ***simulations*** to relate the computational power of machine models — especially variants of Turing machines. Thus it gives you practice applying material introduced in Lecture #7.

Turing Machines That Clean Up After Themselves

A ***Turing machine that cleans up after itself*** is a "standard" Turing machine (that is, the kind of Turing machine introduced in Lecture #6)

$$M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})$$

which satisfies an additional property: M's tape head is always erased (that is, filled with copies of "$\sqcup$"), with its tape head pointing to the leftmost cell of the tape, when a computation ends. Thus M has only one ***accepting configuration***, which is represented by the string "q_{accept}", and M has only one ***rejecting configuration***, which is represented by the string"q_{reject}".

This first problem should easy to solve, and it solutions should only be a few lines.

1. Let $L \subseteq \Sigma^\star$, for an alphabet Σ.

   (a) Prove that if there exists a Turing machine M that cleans up after itself, such that $L = L(M)$, then L is Turing-recognizable (that is, it is the language of a "standard" Turing machine).

   (b) Prove that if there exists a Turing machine M that cleans up after itself, such that M decides L, then L is Turing-decidable (that it, there exists a "standard" Turing machine that decides L).

The goal of the rest of this exercise is to develop a proof of the following.

Claim. *Let $L \subseteq \Sigma^\star$, for an alphabet Σ.*

(a) If L is Turing-recognizable then there exists a Turing machine $\widehat{M}$ that cleans up after itself such that $L = L(\widehat{M})$.

(b) If L is Turing-decidable then there exists a Turing machine $\widehat{M}$ that cleans up after itself such that $\widehat{M}$ decides L.

With that noted, let $L \subseteq \Sigma^\star$, for an alphabet Σ, and let

$$M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})$$

such that $L = L(M)$. The above claim can be proved by describing a Turing machine

$$M = (\widehat{Q}, \Sigma, \widehat{\Gamma}, \widehat{\delta}, \widehat{q}_0, \widehat{q}_{\text{accept}}, \widehat{q}_{\text{reject}})$$

that cleans up after itself, and proving that $L = \widehat{M}$ as well — proving, in addition, that if M decides L then $\widehat{M}$ decides L too.

Representing a Configuration of M

Let $\#_L$, and $\#_R$ be "new" symbols, that is, symbols that do not belong to $Q \cup \Gamma$, and let

$$\widehat{\Gamma} = \Gamma \cup \{\#_L, \#_R\}.$$

Suppose that

$$\omega = \alpha_1 \alpha_2 \ldots \alpha_n$$

is a string in $\Sigma^\star$ — where n is a non-negative integer and $\alpha_1, \alpha_2, \ldots, \alpha_n \in \Sigma^\star$ (so that n is the length of ω).

A configuration of M, that is reached during M's execution on input ω, will be represented (during a simulation by $\widehat{M}$) by itself — except that the symbols will be shifted over by one position to the right, so that the new symbol "$\#_L$" appears on the leftmost cell of the tape — and so that a single copy of "$\#_R$" also appears on the tape, to mark the farthest position that has either ever been reached,[1] or that currently stores a non-blank symbol in Γ.

Suppose, in particular, that a configuration of M is represented by a string

$$\mu_1 \, q \, \mu_2 \tag{1}$$

where $q \in Q$ and $\mu_1, \mu_2 \in \Gamma^\star$ — such that either $\mu_2 = \lambda$ of the rightmost symbol of μ_2 is a non-blank symbol in Γ.

[1] This condition is being used to make the representation of a configuration unique, but also to make it reasonably easy to describe the simulation.

- If μ_2 is not the empty string then, for a non-negative integer ℓ, suppose that the farthest position on the tape that M's tape head has occupied, up until this point, is ℓ cells *to the right of the rightmost symbol in* μ_2. Then $\widehat{M}$'s representation of the configuration at line (1) is a configuration represented by the string

$$\#_L \, \mu_1 \, q \, \mu_2 \, \sqcup^{\ell} \, \#_R \tag{2}$$

- If μ_2 is the empty string then, for a non-negative integer ℓ suppose that the farthest position on the tape that M's tape head has occupied, up until this point, is ℓ cells *to the right of its current position*. Then $\widehat{M}$'s representation of the configuration at line (1) is a configuration represented by the string

$$\#_L \, \mu_1 \, q \, \sqcup^{\ell+1} \, \#_R \tag{3}$$

For each string $\mathcal{C}$ representing a configuration of M, let $\varphi(\mathcal{C})$ be the string representing $\widehat{M}$'s *representation of this configuration*, as described above.

Initialization: A First Assumption

2. When progressing from M's start configuration on a non-empty string, $\omega \in \Sigma^\star$, it is necessary to move the symbols in M over to the right by one position — so each symbol might be (briefly) erased from the tape.

   Describe — briefly and informally — how $\widehat{M}$ can "remember" each symbol, between the time that it has been erased and the time that it has been copied to a new position.

For this exercise, let us ***assume*** that states have been included in $\widehat{Q}$, and transitions have been defined, such that, for $\omega \in \Sigma^\star$, the following properties are satisfied:

- In $\widehat{M}$, $\widehat{q}_0 \vdash^\star \#_L \, q_0 \, \sqcup \, \#_R$.
- If $\omega \neq \lambda$ then, in $\widehat{M}$, $\widehat{q}_0 \, \omega \vdash^\star \#_L \, q_0 \, \omega \, \#_R$.

In other words, let us ***assume*** that $\widehat{M}$ goes from its initial configuration on input ω, to its representation of M's initial configuration on input ω, using a finite number of steps — for every string $\omega \in \Sigma^\star$.

Simulating a Step of M

To continue, note that (if configurations are to be represented as described above) each state in Q must also be a state in $\widehat{Q}$ — that is, $Q \subseteq \widehat{Q}$.

Suppose, now, that M is in a non-halting configuration represented by a string $\mathcal{C}$. Suppose, as well, that the next step of M make use of a transition

$$\delta(q, \sigma) = (r, \tau, d) \tag{4}$$

(for $r \in Q$, $\tau \in \Gamma$, and $d \in \{\mathtt{L}, \mathtt{R}\}$) to move from the configuration represented by $\mathcal{C}$ to another configuration, which is represented by a string $\mathcal{C}'$.

To start things, off, let us require that

$$\widehat{\delta}(q, \sigma) = \delta(q, \sigma) = (r, \tau, d) \tag{5}$$

for every state $q \in Q$ such that $q \notin \{q_{\text{accept}}, q_{\text{reject}}\}$, and for every symbol $\sigma \in \Gamma$.

If the transition $\widehat{\delta}(q, \sigma)$ is applied, and $\widehat{M}$'s tape head points to symbol in Γ (and not to either "$\#_L$" or "$\#_R$") then $\widehat{M}$ is now in its representation of the configuration (shown as) $\mathcal{C}'$, as desired. ***You may use this fact without proving it.***

However, it is possible that $\widehat{M}$'s tape head is now pointing to "$\#_L$", or to "$\#_R$"', instead.

3. Suppose that, after the above transition has been applied, $\widehat{M}$'s tape head points to a copy of "$\#_L$" — so that it must now be pointing to the leftmost cell on $\widehat{M}$'s tape.

   (a) What does this mean about the configuration represented by $\mathcal{C}$ — that is, the configuration that M was in immediately before this step? What does it mean about the transition that is now being applied?

   (b) How is the configuration that $\widehat{M}$ is in, after its transition is applied, related to $\widehat{M}$'s representation of the configuration shown as $\mathcal{C}'$ — that is, the configuration that M is in *after* this step? (You should find that they are almost the same.)

   (c) Describe how transitions $\widehat{\delta}(q, \#_L)$ can be defined, for $q \in Q$, so that $\widehat{M}$'s representation of the configuration shown as $\mathcal{C}'$ can be reached, by taking one more step of $\widehat{M}$.

4. Suppose instead that, after the above transition has been applied, $\widehat{M}$'s tape head points to a copy of "$\#_R$" so that it must be pointing the rightmost cell on $\widehat{M}$'s tape that does not store a copy of "$\sqcup$".

   (a) What does this mean about the configuration represented by $\mathcal{C}$ — that is, the configuration that M was in, immediately before this step? What does it mean about the transition that that is now being applied?

   (b) How is the configuration that $\widehat{M}$ is in, after this transition applied, related to $\widehat{M}$'s representation of the configuration shown as $\mathcal{C}'$ — that is, the configuration that M is in *after* this step? (Once again, you should find that they are almost the same).

(c) Let us introduce new states once again — by setting

$$\dot{Q} = \{\dot{q} \mid q \in Q\}$$

(renaming state, as necessary, so that $\dot{Q} \cap \Gamma = \emptyset$ and Q is also disjoint from the states that have been included in $\widehat{Q}$, before this) — and including all of the states, that are in $\dot{Q}$, in the set $\widehat{Q}$ of states of $\widehat{M}$.

Describe how transitions $\widehat{\delta}(q, \#_R)$ and transitions $\widehat{\delta}(\dot{q}, \sqcup)$ can be defined, for all states $q \in Q$, so that $\widehat{M}$'s representation of the configuration shown as $\mathcal{C}'$ can be reached by taking another *two* steps of $\widehat{M}$.

Completing the Simulation

5. Now consider the following.

   Claim. *Let $\omega \in \Sigma^\star$ and let Turing machines M and $\widehat{M}$ be as described above.*

   (a) If M accepts ω then $\widehat{M}$ moves its start state, for ω, to its representation of an accepting configuration of M, using a finite number of steps.

   (b) If M rejects ω then $\widehat{M}$ moves from its start state, for ω, to its representation of a rejecting configuration of M, using a finite number of steps.

   (c) If M loops on ω then $\widehat{M}$ loops on ω too.

   Describe — reasonably ***briefly*** — how you could prove this claim.

6. In a ***cleanup*** stage, $\widehat{M}$ should move from its representation of any accepting configuration of M to the configuration represented by the string "$\widehat{q}_{\text{accept}}$", and it should move from its representation of any rejecting configuration of M to the configuration represented by the string "$\widehat{q}_{\text{reject}}$".

   Describe — reasonably ***briefly*** — how $\widehat{M}$ can do this. How are the new symbols, "$\#_L$" and $\#_R$, useful here?

For the rest of this exercise you may ***assume*** that if $\mathcal{C}_A$ is an accepting configuration of M then $\varphi(\mathcal{C}_A) \vdash^\star \widehat{q}_{\text{accept}}$, when using $\widehat{M}$, and that if $\mathcal{C}_{\text{R}}$ is a rejecting configuration of M then $\varphi(\mathcal{C}_R) \vdash^\star \widehat{q}_{\text{reject}}$, when using $\widehat{M}$.

7. ***Briefly*** explain how it follows, from the above, that the sets of "recognizable languages" and "decidable" languages would not be changed if they were defined using "Turing machines that clean up after themselves" instead of standard Turing machines.