CPSC 351 — Tutorial Exercise #11
Additional Practice Problems

These problems will not be discussed during the tutorial, and solutions for these problems will not be made available. They can be used as "practice" problems that can help you practice skills considered in the lecture presentation for Lecture #11, or in Tutorial Exercise #11.

Practice Problems

Turing Machines That Clean Up After Themselves — Completing the Simulation

Suppose that "Turing machines that clean up after themselves" are as defined in Tutorial Exercise #11. Let $L \subseteq \Sigma^\star$, for an alphabet Σ, let

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

such that $L = L(M)$. By solving the problems on Tutorial Exercise #11 you began the development of 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, such that $L = L(\widehat{M})$ as well — and, furthermore, such that $\widehat{M}$ decides L if M does.

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.

- 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 — Proving that an Assumption is Satisfied

Along with $\widehat{M}$'s start state, $\widehat{q}_0$, let us include "new" states $\widehat{q}_1$ and $\widehat{q}_2$ in $\widehat{Q}$, defining transitions as follows:

- Let $\widehat{\delta}(\widehat{q}_0, \sqcup) = (\widehat{q}_1, \#_L, \mathtt{R})$.
- For every symbol $\sigma \in \widehat{\Gamma}$, let $\widehat{\delta}(\widehat{q}_1, \sigma) = (\widehat{q}_2, \sigma, \mathtt{R})$.

[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.

- For every symbol $\sigma \in \widehat{\Gamma}$, let $\widehat{\delta}(\widehat{q}_2, \sigma) = (q_0, \#_R, \mathtt{L})$.

Note that, at this point, all transitions out of $\widehat{q}_1$ and $\widehat{q}_2$ (but not all transitions out of $\widehat{q}_0$) have been defined.

1. Prove that, in $\widehat{M}$, $\widehat{q}_0 \vdash^\star \#_L\, q_0 \sqcup \#_R$.

To continue, let us include new states $\widehat{q}_{3,\sigma}$ in $\widehat{Q}$, for every *non-blank* symbol $\sigma \in \widehat{\Gamma}$, as well as a new state $\widehat{q}_4$. Let us define new transitions as follows:

- For every *non-blank* symbol $\sigma \in \widehat{\Gamma}$, let $\widehat{\delta}(\widehat{q}_0, \sigma) = (\widehat{q}_{3,\sigma}, \#_L, \mathtt{R})$.
- Let $\widehat{\delta}(\widehat{q}_{3,\sigma}, \sqcup) = (\widehat{q}_4, \sigma, \mathtt{R})$.
- For every symbol $\sigma \in \widehat{\Gamma}$, let $\widehat{\delta}(\widehat{q}_4, \sigma) = (q_0, \#_R, \mathtt{L})$.

Note that, at this point, all transitions out of $\widehat{q}_0$ (along with all transitions out of $\widehat{q}_1$, $\widehat{q}_2$, and $\widehat{q}_4$ — but not all transitions out of $\widehat{q}_3$) have been defined.

2. Suppose that $\omega = \sigma \in \Sigma$, so that ω is a string in $\Sigma^\star$ with length one. Prove that, in $\widehat{M}$, $\widehat{q}_0\, \omega \vdash^\star \#_L\, q_0\, \omega\, \#_R$.

Now let us include new states $\widehat{q}_{5,\sigma}$ in $\widehat{Q}$, for every *non-blank* symbol $\sigma \in \widehat{\Gamma}$, as well as new states $\widehat{q}_6$ and $\widehat{q}_7$. Let us define new transitions as follows:

- For all *non-blank* symbols $\alpha, \beta \in \widehat{\Gamma}$, let $\widehat{\delta}(\widehat{q}_{3,\alpha}, \beta) = (\widehat{q}_{5,\beta}, \alpha, \mathtt{R})$.
- For all *non-blank* symbols $\alpha, \beta \in \widehat{\Gamma}$, let $\widehat{\delta}(\widehat{q}_{5,\alpha}, \beta) = \widehat{q}_{5,\beta}, \alpha, \mathtt{R})$.
- For all *non-blank* symbols $\alpha \in \widehat{\Gamma}$, let $\widehat{\delta}(\widehat{q}_{5,\alpha}, \sqcup) = (\widehat{q}_6, \alpha, \mathtt{R})$.
- For all symbols $\sigma \in \widehat{\Gamma}$, let $\widehat{\delta}(\widehat{q}_6, \sigma) = (\widehat{q}_7, \#_L, \mathtt{L})$.
- For all symbols $\sigma \in \widehat{\Gamma}$ such that $\sigma \neq \#_L$, let $\widehat{\delta}(\widehat{q}_7, \sigma) = (\widehat{q}_7, \sigma, \mathtt{L})$.
- Finally, let $\widehat{\delta}(\widehat{q}_7, \#_L) = (q_0, \#_L, \mathtt{R})$.

Note that, at this point, all transitions for states $\widehat{q}_0$, $\widehat{q}_1$, $\widehat{q}_2$, $\widehat{q}_4$, $\widehat{q}_6$, $\widehat{q}_7$, and $\widehat{q}_{3,\sigma}$ and $\widehat{q}_{5,\sigma}$, for all $\sigma \in \widehat{\Gamma}$, have been defined.

3. Let $\omega \in \Sigma^\star$ such that $|\omega| \geq 2$. Prove that, in $\widehat{M}$, $\widehat{q}_0\, \omega \vdash^\star \#_L\, q_0\, \omega\, \#_R$.

   This problem is more challenging than the first two, because it requires more than just a trace of execution. If you attempt it then please consider tracing $\widehat{M}$'s execution on several strings with lengths two, three, and four — so that you better understand what the new states are being used for. You should then be able to write down and prove a sequence of claims (somewhat resembling claims that you proved when solving problems on Tutorial Exercise #10, which can be proved in similar ways) that can be used to establish the above result.

Note that, if you solved problems #1–#3, above, then you have shown that the ***first assumption***, concerning "Initialization", that was used in Tutorial Exercise #8, is satisfied.

Completing the Simulation

Now let include states $q_{\text{accept},\mathtt{R}}$, $q_{\text{accept},\mathtt{L}}$, $q_{\text{reject},\mathtt{R}}$, $q_{\text{reject},\mathtt{L}}$, $\widehat{q}_{\text{accept}}$ and $\widehat{q}_{\text{reject}}$ in $\widehat{Q}$ — noting that $\widehat{q}_{\text{accept}}$ and $\widehat{q}_{\text{reject}}$ are the accepting and rejecting states of $\widehat{M}$, respectively.

Suppose that we continue the definition of the transition function $\widehat{\delta}$, as follows.

- For every symbol $\sigma \in \Gamma$ (so that $\sigma \notin \{\#_L, \#_R\}$), $\widehat{\delta}(q_{\text{accept}}, \sigma) = q_{\text{accept},\mathtt{R}}, \sigma, \mathtt{R})$.
- For every symbol $\sigma \in \widehat{\Gamma}$ such that $\sigma \neq \#_R$, $\widehat{\delta}(q_{\text{accept},\mathtt{R}}, \sigma) = (q_{\text{accept},\mathtt{R}}, \sigma, \mathtt{R})$.
- $\widehat{\delta}(q_{\text{accept},\mathtt{R}}, \#_R) = (q_{\text{accept},\mathtt{L}}, \sqcup, \mathtt{L})$.
- For every symbol $\sigma \in \widehat{\Gamma}$ such that $\sigma \neq \#_L$, $\widehat{\delta}(q_{\text{accept},\mathtt{L}}, \sigma) = (q_{\text{accept},\mathtt{L}}, \sqcup, \mathtt{L})$.
- $\widehat{\delta}(q_{\text{accept},\mathtt{L}}, \#_L) = (\widehat{q}_{\text{accept}}, \sqcup, \mathtt{L})$.

4. Prove that, for every accepting configuration $\mathcal{C}_A$ of M, $\varphi(\mathcal{C}_A) \vdash^{\star} \widehat{q}_{\text{accept}}$ (during an execution of $\widehat{M}$).

5. Complete the definition of $\widehat{\delta}$ by introducing transitions for states q_{reject}, $q_{\text{reject},\mathtt{R}}$ and $q_{\text{reject},\mathtt{L}}$, and use these to prove that, for every rejecting configuration $\mathcal{C}_R$ of M, $\varphi(\mathcal{C}_R) \vdash^{\star} \widehat{q}_{\text{reject}}$ (during an execution of $\widehat{M}$).

Note that, if you have solved all these problems, then you have shown that the ***assumptions*** concerning initialization and cleanup, given in Tutorial Exercise #11, can be satisfied. You have, therefore, completed a proof that the sets of recognizable and decidable languages are not changed if they are defined using Turing machines that clean up after themselves, instead of standard Turing machines.

Other Turing Machine Variants

6. A ***Turing Machine That Never Erases*** is a Turing machine

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

— as defined in Lecture #10 — which satisfies one additional condition: For every state $q \in Q \setminus \{q_{\text{accept}}, q_{\text{reject}}\}$, and for every symbol $\sigma \in \Gamma$, if

$$\delta(q, \sigma) = (r, \tau, d)$$

for $r \in Q$, $\tau \in \Gamma$, and $d \in \{\mathtt{L}, \mathtt{R}\}$, then $\tau \neq \sqcup$. That is, M never writes a copy of "$\sqcup$" onto its tape.

Prove that the sets of recognizable languages are not changed, if they are defined using "Turing Machines That Never Erase" instead of standard Turing machines.

7. One might also imagine ***Turing Machines That Clean Up After Themselves*** **and** ***That Never Erase*** — which are standard Turing machines that satisfy the additional requirements for both "Turing Machines That Clean Up After Themselves" and "Turing Machines That Never Erase".

Would the sets of recognizable and decidable languages be changed, if they were defined using *these* new kinds of Turing machines?

Indeed, do these new kinds of Turing machines even exist?