CPSC 351 — Tutorial Exercise #12
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 #12, or in Tutorial Exercise #12.

Practice Problems

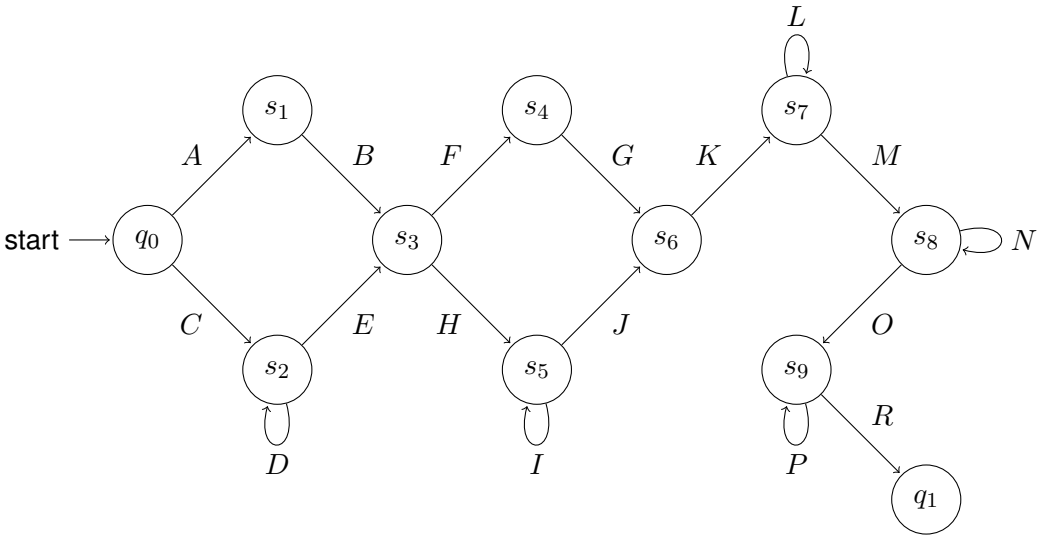
Completing a Turing Machine for Binary Addition

Recall that the initial questions in Tutorial #12 concerned a component of a 3-tape Turing machine — which can be used as part of a Turing machine for binary addition — that could (sometimes) be used to complete the computation when the empty string was to be returned. This component included states r_0, r_1, r_2, r_3, r_4, and r_5, along with the Turing machine's halting state, q_{reject}.

Let us consider, now, another component of a "binary addition" Turing machine, that is as shown in Figure 1 on page 2 — where $\delta(q_0, \sqcup, \sqcup, \sqcup) = (q_{\text{halt}}, (\sqcup, \mathtt{S}), (\sqcup, \mathtt{S}), (\sqcup, \mathtt{S}))$. In order to keep the diagram simple, transitions, to the component that was considered in the tutorial exercises are not shown — and these are as follows:

- $\delta(q_0, \#, \sqcup, \sqcup) = (r_0, (\dot{\#}, \mathtt{R}), (\sqcup, \mathtt{S}), (\sqcup, \mathtt{S}))$.
- For all $\sigma \in \{0, 1, \sqcup\}$, $\delta(s_1, \sigma, \sqcup, \sqcup) = (r_0, (\sigma, \mathtt{S}), (\sqcup, \mathtt{S}), (\sqcup, \mathtt{S}))$.
- $\delta(s_2, \sqcup, \sqcup, \sqcup) = (r_0, (\sqcup, \mathtt{S}), (\sqcup, \mathtt{S}), (\sqcup, \mathtt{S}))$.
- For $\sigma \in \{\#, \sqcup\}$, $\delta(s_3, \sigma, \sqcup, \sqcup) = (r_0, (\sigma, \mathtt{S}), (\sqcup, \mathtt{S}), (\sqcup, \mathtt{S}))$.
- For all $\sigma \in \{0, 1, \#\}$, $\delta(s_4, \sigma, \sqcup) = (r_0, (\sigma, \mathtt{S}), (\sqcup, \mathtt{S}), (\sqcup, \mathtt{S}))$.
- $\delta(s_5, \#, \sqcup, \sqcup) = (r_0, (\#, \mathtt{R}), (\sqcup, \mathtt{S}), (\sqcup, \mathtt{S}))$.

Some of these transitions might be followed ω either has at least two copies of "#" or does not include any copies of this symbol. Others are followed when $\omega = \mu\#\nu$ for $\mu \in \{0, 1\}^\star$ and $\nu \in \{0, 1, \#\}^\star$, but at least one of μ or ν begins with "0" and has length at least two.



A: $\delta(q_0, \mathtt{0}, \sqcup, \sqcup) = (s_1, (\dot{\mathtt{0}}, \mathtt{R}), (\dot{\mathtt{0}}, \mathtt{R}), (\sqcup, \mathtt{S}))$
B: $\delta(s_1, \#, \sqcup, \sqcup) = (s_3, (\#, \mathtt{R}), (\sqcup, \mathtt{S}), (\sqcup, \mathtt{S}))$
C: $\delta(q_0, \mathtt{1}, \sqcup, \sqcup) = (s_2, (\dot{\mathtt{1}}, \mathtt{R}), (\dot{\mathtt{1}}, \mathtt{R}), (\sqcup, \mathtt{S}))$
D For all $\sigma \in \{\mathtt{0}, \mathtt{1}\}$, $\delta(s_2, \sigma, \sqcup, \sqcup) = (s_2, (\sigma, \mathtt{R}), (\sigma, \mathtt{R}), (\sqcup, \mathtt{S}))$
E $\delta(s_2, \#, \sqcup, \sqcup) = (s_3, (\#, \mathtt{R}), (\sqcup, \mathtt{S}), (\sqcup, \mathtt{S}))$
F: $\delta(s_3, \mathtt{0}, \sqcup, \sqcup) = (s_4, (\mathtt{0}, \mathtt{R}), (\sqcup, \mathtt{S}), (\dot{\mathtt{0}}, \mathtt{R}))$
G $\delta(s_4, \sqcup, \sqcup, \sqcup) = (s_6, (\sqcup, \mathtt{S}), (\sqcup, \mathtt{S}), (\sqcup, \mathtt{S}))$
H: $\delta(s_3, \mathtt{1}, \sqcup, \sqcup) = (s_5, (\mathtt{1}, \mathtt{R}), (\sqcup, \mathtt{S}), (\dot{\mathtt{1}}, \mathtt{R}))$
I: For all $\sigma \in \{\mathtt{0}, \mathtt{1}\}$, $\delta(s_5, \sigma, \sqcup, \sqcup) = (s_5, (\sigma, \mathtt{R}), (\sqcup, \mathtt{S}), (\sigma, \mathtt{R}))$
J: $\delta(s_5, \sqcup, \sqcup, \sqcup) = (s_6, (\sqcup, \mathtt{S}), (\sqcup, \mathtt{S}), (\sqcup, \mathtt{S}))$
K: $\delta(s_6, \sqcup, \sqcup, \sqcup) = (s_7, (\sqcup, \mathtt{L}), (\sqcup, \mathtt{S}), (\sqcup, \mathtt{S}))$
L: For all $\sigma \in \{\mathtt{0}, \mathtt{1}, \#\}$, $\delta(s_7, \sigma, \sqcup, \sqcup) = (s_7, (\sqcup, \mathtt{L}), (\sqcup, \mathtt{S}), (\sqcup, \mathtt{S}))$
M: For all $\sigma \in \{\dot{\mathtt{0}}, \dot{\mathtt{1}}\}$, $\delta(s_7, \sigma, \sqcup, \sqcup) = (s_8, (\sqcup, \mathtt{S}), (\sqcup, \mathtt{L}), (\sqcup, \mathtt{S}))$
N: For all $\sigma \in \{\mathtt{0}, \mathtt{1}\}$, $\delta(s_8, \sqcup, \sigma, \sqcup) = (s_8, (\sqcup, \mathtt{S}), (\sigma, \mathtt{L}), (\sqcup, \mathtt{S}))$
O: For all $\sigma \in \{\dot{\mathtt{0}}, \dot{\mathtt{1}}\}$, $\delta(s_8, \sqcup, \sigma, \sqcup) = (s_9, (\sqcup, \mathtt{S}), (\sigma, \mathtt{S}), (\sqcup, \mathtt{L}))$
P: For all $\alpha \in \{\dot{\mathtt{0}}, \dot{\mathtt{1}}\}$ and for all $\beta \in \{\mathtt{0}, \mathtt{1}\}$, $\delta(s_9, \sqcup, \alpha, \beta) = (s_9, (\sqcup, \mathtt{S}), (\alpha, \mathtt{S}), (\beta, \mathtt{L}))$
R: For all $\alpha, \beta \in \{\mathtt{0}, \mathtt{1}\}$, $\delta(s_9, \sqcup, \dot{\alpha}, \dot{\beta}) = (q_1, (\sqcup, \mathtt{S}), (\alpha, \mathtt{S}), (\beta, \mathtt{S}))$

Figure 1: Component of a 3-Tape Turing Machine

1. Show that $q_0\,\omega \;\sharp\; q_0 \;\sharp\; q_0 \vdash^\star q_{\text{halt}} \;\sharp\; q_{\text{halt}} \;\sharp\; q_{\text{halt}}$, for $\omega \in \{\mathtt{0}, \mathtt{1}, \#\}^\star$, in each of the following cases.

   (a) $\omega = \lambda$.

(b) ω begins with "#".

(c) $\omega = 0$.

(d) $\omega = 0\alpha$, for a string $\alpha \in \{0, 1, \#\}^\star$ that begins with either "0" or "1".

(e) ω begins with "1" but does not include any copies of "#".

(f) $\omega = \mu\,\#$, where μ is the unpadded binary representation of some non-negative integer.

(g) $\omega = \mu\#0\alpha$, where μ is the unpadded binary representation of some non-negative integer, and α is a non-empty string in $\{0, 1, \#\}^\star$.

(h) ω begins with $\mu\#1$, where μ is the unpadded binary representation of some non-negative integer, but ω includes at least two copies of "#".

Note that if none of the above cases hold, then $\omega = \mu\,\#\,\nu$, where $\mu, \nu \in \{0, 1\}$ are the unpadded binary representations of a pair of non-negative integers.

2. Suppose, next, that $\omega = \mu\,\#\,\nu$, where μ and ν are the unpadded binary representations of a pair of non-negative integers. For a non-empty string

$$\zeta = \alpha_1\alpha_2\ldots\alpha_\ell$$

with length $\ell \geq 1$ (and with $\alpha_1, \alpha_2, \ldots, \alpha_\ell \in \{0, 1\}$, let

$$\zeta_M = \dot{\alpha}_1\alpha_2\ldots\alpha_\ell$$

— so that the leftmost symbol, α, has been replaced in ζ_M by the corresponding "dotted" symbol, $\dot{\alpha}_1$, and no other symbols have been changed.

(a) Show that $q_0\,\mu\,\#\,\nu\,\sharp\,q_0\,\sharp\,q_0 \vdash^\star \mu_M\,\#\,s_3\,\nu\,\sharp\,\mu_M\,s_3\,\sharp\,s_3$.

(b) Show that $\mu_M\,\#\,s_3\,\nu\,\sharp\,\mu_M\,s_3\,\sharp\,s_3 \vdash^\star \mu_M\,\#\,\nu\,s_6\,\sharp\,\mu_M\,s_6\,\sharp\,\nu_M\,s_6$.

(c) Show that $\mu_M\,\#\,\nu\,s_6\,\sharp\,\mu_M\,s_6\,\sharp\,\nu_M\,s_6 \vdash^\star s_8\,\sharp\,\mu_M\,s_8\,\sharp\,\nu_M\,s_8$.

(d) Show that $s_8\,\sharp\,\mu_M\,s_8\,\sharp\,\nu_M\,s_8 \vdash^\star s_9\,\sharp\,s_9\,\mu_M\,\sharp\,\nu_M\,s_9$.

(e) Show that $s_9\,\sharp\,s_9\;\;\mu_9\,\sharp\,\nu_M\,s_9 \vdash^\star q_1\,\sharp\,q_1\,\mu\,\sharp\,q_1\,\nu$.

Note that it now follows that $q_0\,\mu\,\#\,\nu \vdash^\star q_1\,\sharp\,q_1\,\mu\,\sharp\,q_1\,\nu$.

3. Use the above to show that the component of a Turing machine given above, and at the beginning of Tutorial Exercise #12, correctly implements step 1 of the algorithm for the addition of binary numbers that is discussed in the presentation for Lecture #12.

Additional Problems

4. Modify the component of a 3-tape Turing machines shown at the beginning of Tutorial Exercise #12, and above, in order to produce a similar component that implements step #1 of the algorithm shown in Figure 2 of Tutorial Exercise #12, instead of step #1 of the algorithm shown in Figure 1 of the notes for Lecture #12 — and explain, briefly, why your answer is correct.

5. Once again, let $\Sigma_{\text{binary}} = \{0, 1\}$ and let $\Sigma_{\text{pair}} = \{0, 1, \#\}$. Consider a function

$$f_{\text{mult}} : \Sigma^\star_{\text{pair}} \to \Sigma^\star_{\text{binary}}$$

such that the following properties are satisfied, for every string $\omega \in \Sigma^\star_{\text{pari}}$:

- If $\omega = \mu \# \nu$, where $\mu, \nu \in \Sigma^\star_{\text{binary}}$ are the unpadded binary representations of a pair of non-negative integers n and m, respectively, then $f_{\text{mult}}(\omega)$ is the unpadded binary representation of their product, $n \times m$.
- Otherwise $f_{\text{mult}}(\omega) = \lambda$, the empty string.

(a) Write an algorithm (similar to the algorithms that have been considered in the presentation for Lecture #12 and earlier on, in this exercise) to compute this function — assuming that you may now include a step

$$i := i + j$$

for non-negative integers i and j as a step in your algorithm.

(b) Design a multi-tape Turing machine that computes the function f_{mult} and sketch a proof that your algorithm is correct.

Ideally, if you see how the above problems can be solved, then you can also see (at least roughly) how the following additional problems could be solved too:

- Given a string $\omega \in \Sigma^\star_{\text{pair}}$, design a Turing machine to decide whether $\omega = \mu \# \nu$, for a pair of strings $\mu, \nu \in \Sigma^\star_{\text{binary}}$ that are the unpadded binary representations of integers n and m such that $n < m$.

- Design Turing machines to compute functions

$$f_{\text{quotient}} : \Sigma^\star_{\text{pair}} \to \Sigma^\star_{\text{binary}} \qquad \text{and} \qquad f_{\text{remainder}} : \Sigma^\star_{\text{pair}} \to \Sigma^\star_{\text{binary}}$$

in order to implement integer division with remainder (with non-negative integers as inputs) as well.