Exercise 7.26.

Let *G* be the nondeterministic TM described in Example 7.30, which begins with a blank tape, writes an arbitrary string x on the tape, and halts with tape $\underline{\Delta}x$.

Let *NB*, *PB*, *Copy*, *Delete*, and *Equal* be the TMs described in Examples 7.17, 7.18, 7.20, and 7.24.

Consider the NTM

$$\mathit{NB} \rightarrow G \rightarrow \mathit{Copy} \rightarrow \mathit{NB} \rightarrow \mathit{Delete} \rightarrow \mathit{PB} \rightarrow \mathit{PB} \rightarrow \mathit{Equal}$$

which is nondeterministic because *G* is.

What language does it accept?

Exercise 7.27.

Using the idea in Exercise 7.26, draw a transition diagram for an NTM that accepts the language

$$\{1^n \mid \; n = k^2 \text{ for some } k \geq 0\}.$$

A slide from lecture 4:

Example 7.30. The Language of Prefixes of Elements of L.

Let $L = L(T)$. Then

$$P(L) = \{x \in \Sigma^* \mid \; xy \in L \text{ for some } y \in \Sigma^*\}$$

$$\mathit{NB} \rightarrow \mathit{G} \rightarrow \mathit{Delete} \rightarrow \mathit{PB} \rightarrow T$$

Exercise 7.28.

Suppose L is accepted by a TM T.

For each of the following languages, describe informally how to construct a nondeterministic TM that will accept that language.

a. The set of all suffixes of elements of L.

b. The set of all substrings of elements of L.

Exercise 7.29.

Suppose L_1 and L_2 are subsets of Σ^* and T_1 and T_2 are TMs accepting L_1 and L_2, respectively.

Describe how to construct a nondeterministic TM to accept L_1L_2.

Exercise 7.30.

Suppose T is a Turing machine accepting a language L. Describe how to construct a nondeterministic TM accepting L^*.

Exercise 7.32.

Describe informally how to construct a TM T that enumerates the set of palindromes over $\{0, 1\}$ in canonical order.

In other words, T loops forever, and for every positive integer n, there is some point at which the initial portion of T's tape contains the string

$$\Delta\Delta 0\Delta 1\Delta 00\Delta 11\Delta 000\Delta \ldots \Delta x_n$$

where x_n is the nth palindrome in canonical order, and this portion of the tape is never subsequently changed.

Exercise 7.33.

Suppose you are given a Turing machine T (you have the transition diagram), and you are watching T processing an input string. At each step you can see the configuration of the TM: the state, the tape contents, and the tape head position.

a. Suppose that for some n, the tape head does not move past square n while you are watching. If the pattern continues, will you be able to conclude at some point that the TM is in an infinite loop? If so, what is the longest you might need to watch in order to draw this conclusion?

b. Suppose that in each move you observe, the tape head moves right. If the pattern continues, will you be able to conclude at some point that the TM is in an infinite loop? If so, what is the longest you might need to watch in order to draw this conclusion?

Exercise 7.34.

In each of the following cases, show that the language accepted by the TM T is regular.

a. There is an integer n such that no matter what the input string is, T never moves its tape head to the right of square n.

b. For every $n \geq 0$ and every input of length n, T begins by making $n+1$ moves in which the tape head is moved right each time, and thereafter T does not move the tape head to the left of square $n+1$.

A slide from lecture 4:

Definition 8.1. Accepting a Language and Deciding a Language

A Turing machine T with input alphabet Σ accepts a language $L \subseteq \Sigma^*$,
if $L(T) = L$.

T *decides* L,
if T computes the characteristic function $\chi_L : \Sigma^* \to \{0, 1\}$

A language L is *recursively enumerable*,
if there is a TM that accepts L,

and L is *recursive*,
if there is a TM that decides L.

A slide from lecture 4:

Theorem 8.7. If L is a recursively enumerable language, and its complement L' is also recursively enumerable, then L is recursive
(and therefore, by Theorem 8.6, L' is recursive).

Proof. . .

Exercise 8.4.

Suppose $L_1, L_2, \ldots, L_k$ form a partition of Σ^*: in other words, their union is Σ^* and any two of them are disjoint.

Show that if each L_i is recursively enumerable, then each L_i is recursive.

Exercise 8.8.

Suppose L is recursively enumerable but not recursive. Show that if T is a TM accepting L, there must be infinitely many input strings for which T loops forever.