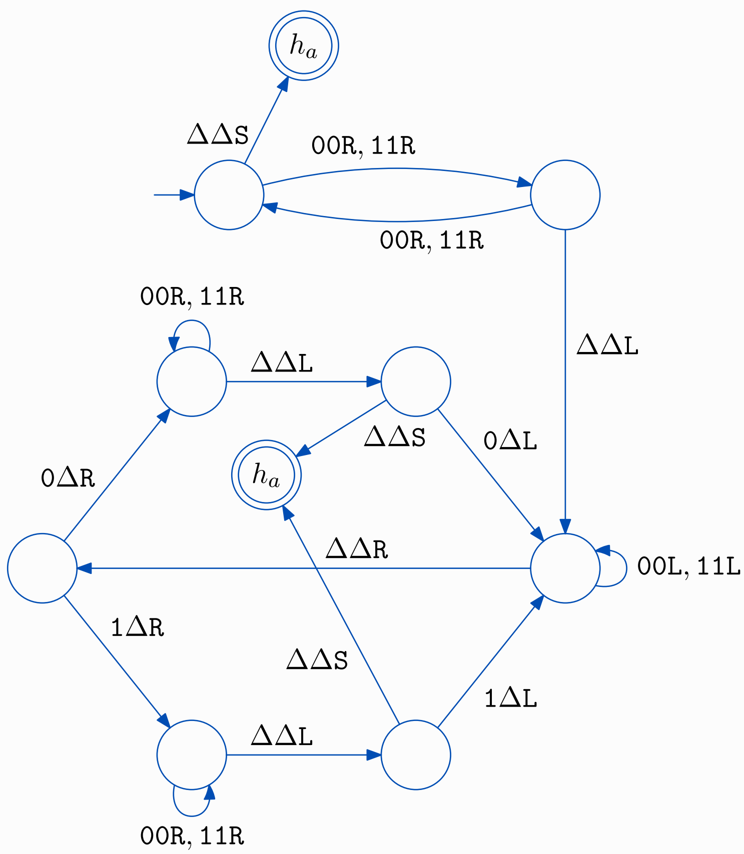
Warmup 4: TMs and their Languages

1. Let X be the set of all binary strings that either have even length or are odd-length palindromes. Draw a TM for X.



2. Give an **example** of (or state that does not exist):

   (a) a language that is recursive but not r.e.

   By definition does not exist

   (b) a language that is regular but whose complement is recursive

   Any regular language

   (c) a language accepted by a TM that does not have a context-free grammar

   $0^n1^n2^n$

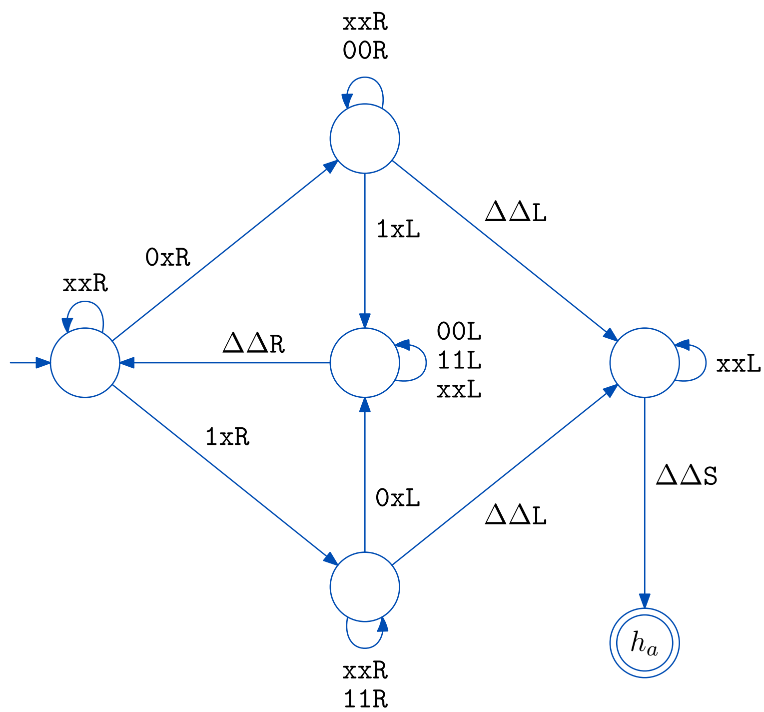
   (d) a language accepted by a nondeterministic TM but not by a deterministic TM.

   Does not exist

3. State Church's thesis.

There is a TM for problem if and only if there is an effective procedure (algorithm) for it

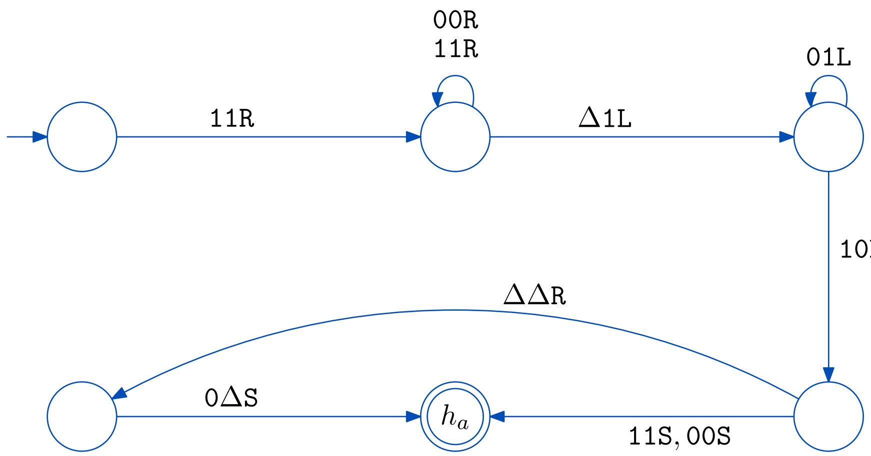
4. Consider the following TM with input alphabet {0, 1}:



(a) Give two strings of length 3 accepted by the TM.
(b) Give two strings of length 3 NOT accepted by the TM.
(c) Describe in succinct-ish English the language of this TM. Be precise.

All binary strings where the number of 0's and the number of 1's differ by exactly one.

5. Consider the following transducer.



(a) What is the output if the input is `11` ?

101

(b) What is the output if the input is `1000` ?

1111

This transducer computes an arithmetic function treating the input (and output) as the binary of a positive integer. What function does it compute?

$x \to 2x - 1$

6. Consider a TM that is not allowed to write the same character to a cell that it just read. (For example, `00R` would not be allowed.) Show that this version has the same power as a standard TM.

E.g. For each offending transition, such as `00R`, add extra state and replace with a move that writes a special character and keeps head stationary while going to the extra state; then add transition that writes `0` and moves right.