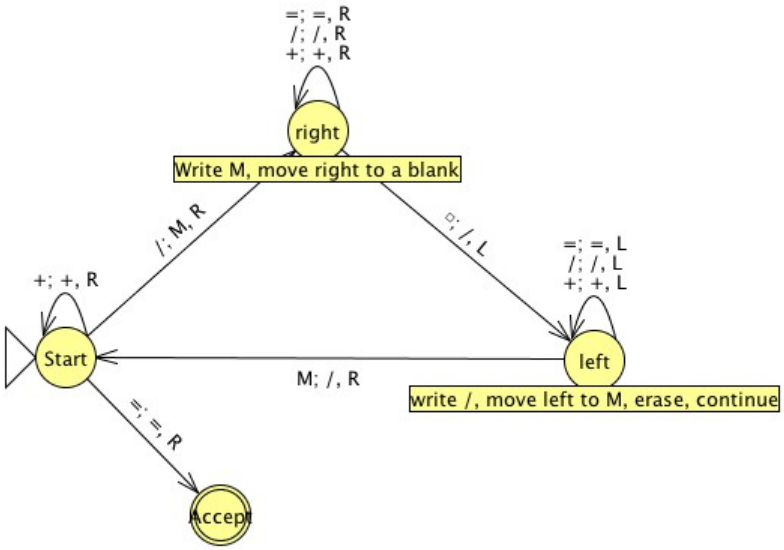
STROKE ARITHMETIC

STROKE ADDITION

We explore performing arithmetic on a turing machine. But before we do any arithmetic we have to decide on the method of representation of numbers. Do we represent them in decimal, binary, or some other way. Let's start with something very primitive, representing a non-negative integer with n-strokes, 5 = /////. A turing machine need a tape alphabet, a set of symbols that may be placed on a tape. Naturally we always have the blank symbol, symbols that will be part of the input string, symbols we need to perform the process of converting the tape to include the answer.

If we start a turing machine with a tape like, ///+//=, the machine would produce the result, ///+//=/////. The functionality of the machine demonstrates a typical turing machine process – marking an item on the tape, going to another place on the tape and doing something, then going back to the marker and repeating the process.

The ***StrokeAdder*** is illustrated in the figure to the right. It start in the indicated state, replaces the stroke with an "M" to mark the symbol that will be copied to the answer portion of the tape. For example, if the machine starts with,




```
///+//=
^
```

would write

```
M//+//=
 ^
```

and transition to the state, *right*. In *right* the machine would move to the right until it passed the = symbol, then it would write

```
M//+//=/
       ^
```

move left, go to state *left*, until it finds the M. Then it overwrites the M with a /, move right and transitions to the start *stat*e,

```
///+//=/
 ^
```

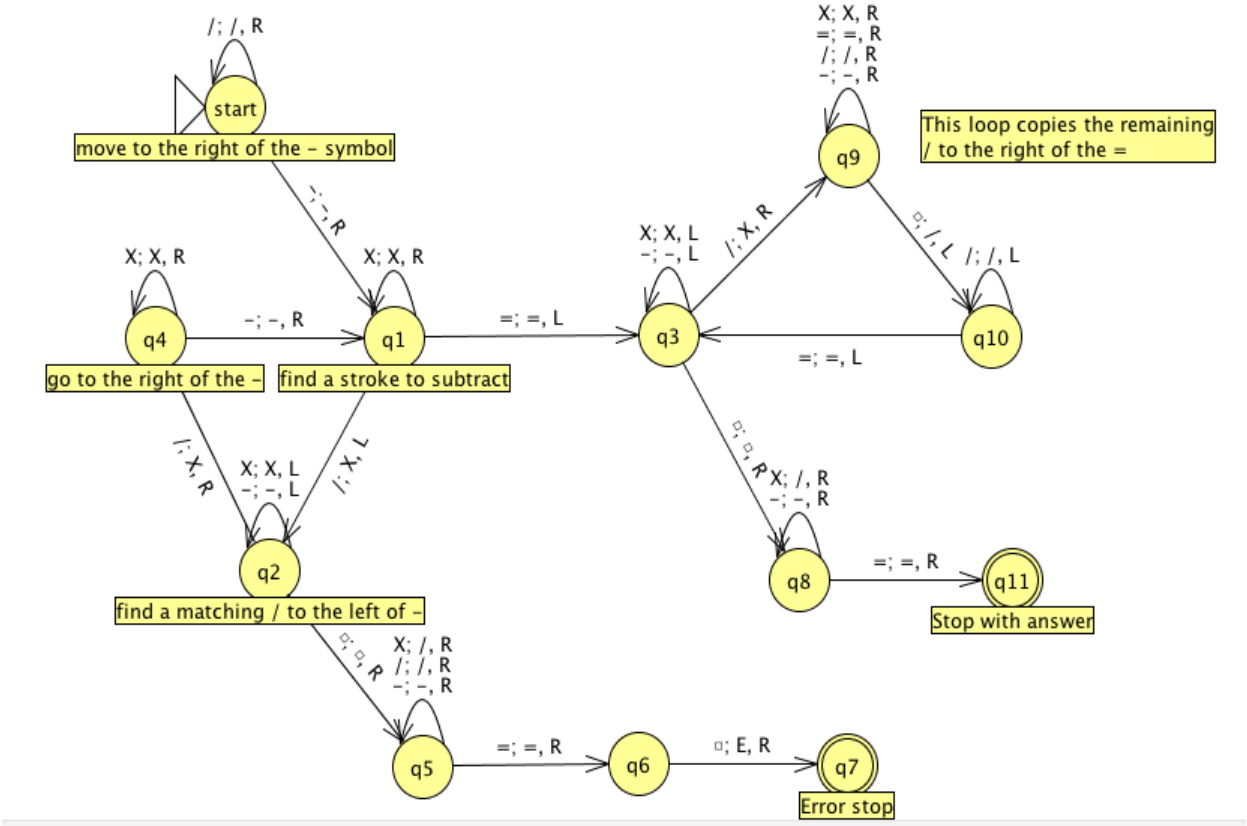
It continues going through the *start-right-left* loop, marking a / with an M, moving to the right and appending it to the string, then moving back to the left replacing the M with the /, until all the strokes to the left of the = are copied. Then the machine goes to the *Accept* state and halts.

STROKE SUBTRACTION

Now that we've seen a stroke adder, how might we construct a stroke subtractor. If you want to tryyour own, read no further and get started. There are various ways of doing this, do be biased by the

approach show below.

Assume we have an input, like ///-/= and our turing machine will process the tape and produce, ///-/=//. For completeness, the machine that we produce below also handles errors, like //-///=E.



The approach show below performs subtraction by first moving to the "-" symbol on the tape, finds a stroke to the right of the "-", then looks for a stoke to the right and marks it, state q_1. State q_2 find a stroke to the left of the "-". The q_1-q_2-q_4 loop continues pairing the strokes until it finds the "=" symbol in q_1, or does not find a match in q_2.

If it does not find a match in q_2, in returns the input string to its original form and writes an E after the "=".

If it scans the "=" in q_1 it proceeds to the q_3-q_9-q_{10} loop and copies the remaining strokes to the right of the "=", goes to state q_8 and returns the original portion of the input tape to its initial form and halts.