TURING BLOCKS

The transition from multi-tape Turing machines to Block Turing Machine in JFLAP is easy along as you have the right focus. View the "Blocks" approach as focusing on "what", and the multi-tape Turing machine as focusing on "how". The first bid difference is that the multi-tape machine transitions associate with a triple, (symbolViewed, symbolWrote, headMotion). The transitions on the block machine simply associate with the symbolViewed. That is, if you are in a block and you view the symbol assoicated with the transition, then go to the next block.

The library of predefined blocks is very complete and contains a reasonable set of the normal types of operations one may want to string together to construct a Turing machine:

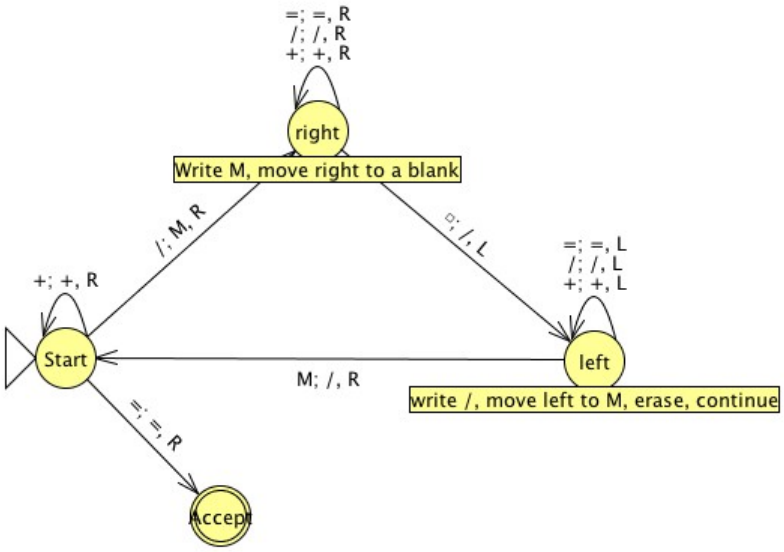
1. Start Block: Self explanatory.
2. Final Block: Self Explanatory.
3. Move Block: Move left or right as indicated.
4. Write Block: Write a symbol on the tape.
5. Move Until Block: Move left or right until you find a symbol on the tape.
6. Move Until Not Block: Move left or right until you by-pass a specific symbol.
7. Shift Block:
8. Copy Block:

Simply stated, you will not necessarily wind up with a turing machine with fewer states. Rather, you will wind up with a turing machine that is easier to read. To illustrate, consider the StrokeAdder (UnaryAdder). The machine starts with a input string in the format, for example,

//+///=,

with the read/write head scanning the leftmost symbol, and terminates with the tape containing,

//+///=/////.



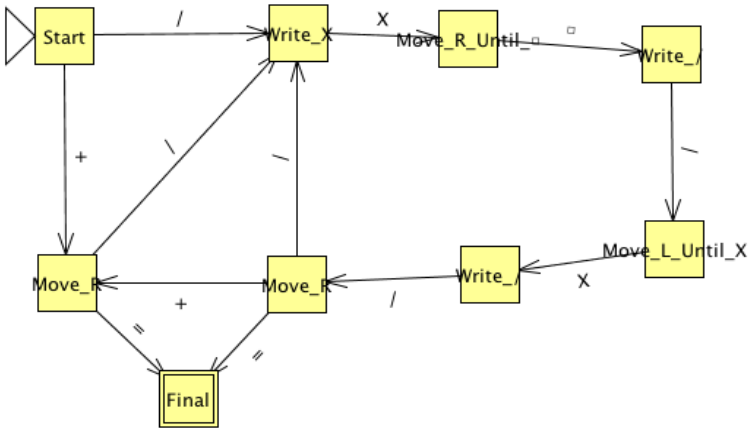
Basically, the states, *start, right, left*, mark a stroke with an "M" in the transition from *Start* to *right*, moves right to find a blank space, then writes a stroke in the blank space, goes to the *left* state looks for the "M", replaces it by a "/" and goes back to *start*.

HOW OR WHAT!

A description of the operation of a turing machine focuses on **how** the problem is solved. Describing the operation of a block turing machine focuses on a higher level step-wise description of **what** we wish to accomplish at each step in the algorithm.

A machine constructed with blocks begins in the block state, **Start**:

- **Start**: If a "/" is scanned, go to the Write_X block. Since an "X" was written marking the tape position, go to the Move_R_Until_empty block, at the empty space, **Write_/**, with the "/" written, transition to the **Move_L_until_X** block. Upon seeing the "X", **Write_/**, upon seeing the "/"go to **Move_R** and continue with the bullet below.



  In **Start** if a "+" is scanned, the first number was zero, in which case the read/write head does a **Move_R**. Here it depends upon whether the read/write head is scanning a "/" or a "=". If it sees a "/", it goes to **Write_X** to mark the tape and goes through the loop described above and process the second number. If it see the "=", the second value was zero, and the algorithm goes to the **Final** state.

- **Move_R**: The next action from this block depends upon what is being scanned, the "=", the "+", or a "/". If the "=" is scanned, the process is finished and it goes to the **Final** block. If the "+" is scanned it it goes to the other **Move_R** block, see above, and begins scanning the representation of the second number. If it scans a "/", it continues processing the representation of the first number.

In general, once you make the transition to block turing machines you will never want to go back.