Turing Machine Exercises

Use the primitives R, L, R_a, L_a, R_b, L_b, $R_\square$, $L_\square$, R_0, L_0, R_1, L_1, $R_{\overline{a}}$, $L_{\overline{a}}$, $R_{\overline{b}}$, $L_{\overline{b}}$, $R_{\overline{\square}}$, $L_{\overline{\square}}$, $R_{\overline{0}}$, $L_{\overline{0}}$, $R_{\overline{1}}$, $L_{\overline{1}}$, a, b, 0, 1, $\square$, A, S, Shl, Shr, N_L, N_R, W_E and W_B, and the tape alphabet of $\{a, b, 0, 1, \square\}$ where,

A — Adds one in binary, the read/write head begins and ends on the leftmost digit. So applying it to $\underline{1}00101$ produces $\underline{1}00110$. Also the number grows to the left, so $\square\underline{1}11$ produces $\underline{1}000$.

S — Subtracts one in binary, the read/write head begins and ends on the leftmost digit. So applying it to $\underline{1}00110$ produces $\underline{1}00101$. Also the number shrinks on the left, so $\underline{1}000$ produces $\square\underline{1}11$.

Shl — Shifts a word one space to the left. So $\square\underline{a}ba$ produces $\underline{a}ba\square$.

Shr — Shifts a word one space to the right. So $\underline{a}ba\square$ produces $\square\underline{a}ba$.

N_L — Moves the read/write head to the beginning of the next word to the left.

N_R — Moves the read/write head to the beginning of the next word to the right.

W_E — Moves the read/write head to the end of the word. If the read/write head is on a space the head does not move.

W_B — Moves the read/write head to the beginning of the word. If the read/write head is on a space the head does not move.

1. Construct a Turing machine, in diagram form, that takes two input numbers on the tape, n_1 and n_2, in binary form, and returns the binary form of $n_1 n_2$. For example, an input of $101\square1011$ will produce an output of 110111. For this exercise, you may also use the Ad machine we constructed in class, that adds two binary numbers. That is, if the tape holds $1101\square1011\underline{0}$ the result will be $\square\square\square\square10001\underline{1}$.

2. Construct a Turing machine, in diagram form, that will take an input of a single word from $\{a, b\}^*$ and write the number of characters in binary on front of the word. The original word is not altered by the computation. For example, if the input tape is $\square\underline{a}bbbababbaa\square$ the Turing machine produces $\square1011\square\underline{a}bbbababbaa\square$. You may use other tape symbols if you would like, in this case the R_x, $R_{\overline{x}}$, L_x and $L_{\overline{x}}$ machines work as usual.

3. Construct a Turing machine, in diagram form, that will leave only a 1 on the tape if the current word on the tape is in the language $L = \{w \mid n_a(w) = n_b(w)\}$ and will leave only a 0 on the tape if the word is not in the language.

4. Construct a Turing machine, in diagram form, that takes an input number, n, in binary form, and outputs the word $a^n b^n$. For example, an input of 1011 produces $aaaaaaaaaaabbbbbbbbbbb$.

5. Construct a Turing machine, in diagram form, that takes an input number, n, in binary form, and returns the binary form of 2^n. For example, an input of 101 will produce an output of 100000.