$$\delta : Q \times \Gamma^2 \longrightarrow Q \times \Gamma^2 \times \{L, R\}$$

- from Sipser 3.3 (adapted)

Considering the Turing Machine (TM) with one head and two tapes (T_m), then our T_m is a 7-tuple $(Q, \Sigma, \Gamma^2, \delta, q_0, q_{accept}, q_{reject})$ where Q, Σ, Γ are all finite sets and:

1. Q is the set of states;
2. Σ is the input alfabet ~~where~~ not containing the blank symbol '␣';
3. $\Gamma^2 = \underbrace{\Gamma_1}_{\text{tape 1}} \cup \underbrace{\Gamma_2}_{\text{tape 2}}$ = tape alphabet, where '␣' $\in \Gamma$ and $\Sigma \subseteq \Gamma$;
4. $\delta : Q \times \Gamma^2 \to Q \times \Gamma^2 \times \{L, R\}$ - the transition function;
5. $q_0 \in Q$ is the start state;
6. $q_{accept} \in Q$ is the accept state;
7. $q_{reject} \in Q$ is the reject state, where $q_{reject} \neq q_{accept}$.

Our T_m computes as follows: Initially ~~It~~ ~~receives~~

T_m receives its input $w = w_1 w_2 \ldots w_n \in \Sigma^*$ on the leftmost n squares of the tape 1. Then we copy the string w on the tape 2 so we can verify them later. The rest of the tape is blank (first '␣' marks the ~~right~~ end of the input). Once T_m has started, the computation proceeds according to the rules described by the δ function. If T_m tries to move its head to the left off the left-hand end of the tape, the head stays in the same place for that move (index 0). The computation continues until it enters either the accept or the reject state at which point it halts. If neither occurs T_m goes on forever.

The second tape can ~~[illegible]~~ be used at every step (checking if the k-square on t_1 is identical to the k-square on t_2) ~~[illegible]~~ but we'll probably use it as a raw input and never modify it, so if we encounter an error, we check the first K squares that ~~are~~ computed on tape 1 on the tape 2 and read from tape 2, then read from tape 1 the symbol that we should 've wrote.