

## Equivalence of two-tape and one-tape Turing machines

Recall that a multi-tape TM, such as a two-tape one, is a TM with a single transition function that maps from the state and the  $k$  symbols read on the  $k$  tapes to a new state and  $k$  operations of writing to a tape or moving left or right on it.

**Theorem:** Two-tape TMs and one-tape TMs accept the same set of languages.

**Proof:**

1. (1  $\rightarrow$  2) Given a one-tape TM  $M$ , construct a two-tape TM that is identical to the one-tape one except that its transition function has provision for mappings from inputs and to actions on the second tape, and all these mappings are null.
2. (2  $\rightarrow$  1) Given two-tape TM  $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{acc}, q_{rej})$ , construct TM  $M' = (Q', \Sigma', \Gamma', \delta', q_0', q_{acc}, q_{rej})$ , as follows.  
Each tape cell stores a four-part symbol, two parts of which are the symbols on the two-tape machine's tape, and two of which are markers  $\mu$  for the tape head locations.
3. Hence the set of composite symbols readable on the tape of  $M'$  is:  
$$\Sigma' \cup \Gamma' = (\Sigma \cup \Gamma)_2 \times \{ \mu \}_2.$$
4. At each computation step,  $M'$  scans the tape for the composite symbol containing  $M$ 's tape-1 marker, scans the tape for  $M$ 's tape-2 marker, generating the same actions as  $M$  would generate but executing them by changing components of the composite symbols on the single tape of  $M'$ . In this way  $M'$  simulates  $M$  in lock step.

---

Source: Hopcroft, Ullman, Motwani, 2007, pp. 345-346.