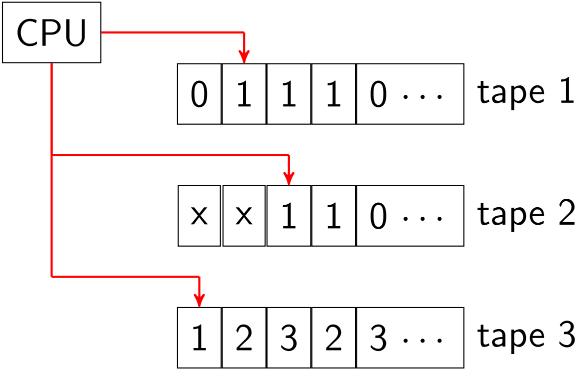
Complexity of Nondeterministic TM I

- Remember NTM is a decider if all branches halt on all inputs
- Definition of NTM time complexity $t(n)$:
  maximum # of steps the machine uses for any path from root to a leaf
- Theorem 7.11
  Let $t(n) \geq n$. For a $t(n)$ NTM (single tape)
  $\Rightarrow \exists$ a $2^{O(t(n))}$ TM (single tape)
- Assume b is the maximal number of branches at each node

Complexity of Nondeterministic TM II

- Recall our way of doing the simulation is by the following three-tape TM



Complexity of Nondeterministic TM III

- We use a breadth-first way for the simulation
- That is, after one layer is finished, we do the next
- Tape 3: a path from root to a node
- Total number of nodes in the tree:

$$O(b^{t(n)})$$

- Tape 2: run the original input w from root to one node in the tree
- Cost of running from root to one node in tape 2: $O(t(n))$

Complexity of Nondeterministic TM IV

- Cost to update contents of tape 3: $O(t(n))$
  Recall that tape 3 has
  1
  2
  3
  11
  . . .
  33
  111
  . . .
  333

Complexity of Nondeterministic TM V

- Total time:

$$\begin{aligned} &\# \text{ nodes } \times \text{ cost per node} \\ =&O(b^{t(n)}) \times O(t(n)) = 2^{O(t(n))} \end{aligned}$$

- Note that in the above equality we used

$$\begin{aligned} &b^{t(n)} \times t(n) = 2^{\log_2(b^{t(n)} t(n))} \\ =&2^{(\log_2 b) t(n) + \log_2 t(n)} = 2^{O(t(n))} \end{aligned}$$

- This is by a 3-tape TM

Complexity of Nondeterministic TM VI

- To use a single-tape TM to simulate a 3-tape one, we need

$$(2^{O(t(n))})^2 = 2^{O(t(n))}$$

cost because

$$(2^{O(t(n))})^2 \leq (2^{ct(n)})^2 = 2^{2ct(n)} = 2^{O(t(n))}$$