More on Turing machines[1] :

- computing functions with DTMs: every computation terminates in a unique final state; leaves tape head at the leftmost cell; typically, initial state is never reentered

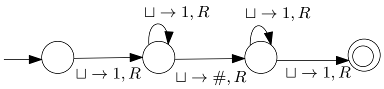
  concatenating two input strings separated by a blank, each with alphabet $\{a, b\}$

  given an integer j in unary, compute $3j$ with a modular design —

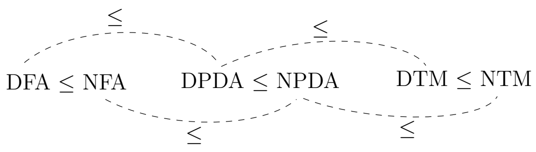
  - modules correspond to – replace last 1 with # (let the resulting string be w); $w\#w$ while leaving tape head at first cell; move to # on the right; $w\#w\#w$ while leaving tape head at first #; $w\#ww$ while leaving tape head at first #; move tape head to first cell; www while leaving tape head at first cell

- NTM example: $L = \{\text{positive integer } r \text{ in unary} \mid r \text{ is composite}\}$

  input is 1^{r+1}; non-deterministically choose two positive integers, say p and q (refer to automata below); with tape containing $1^{r+1}\#1^{p+1}\#1^{q+1}$; make sure $p < r$ and $q < r$; multiply p and q so that the input has $1^{r+1}\#1^{p*q+1}$; compare 1^{r+1} with 1^{p*q+1} and accept if they are same; otherwise reject



- with respect to the power of recognizing a language, it is obvious that,



  however, shortly, we will establish $DFA = NFA < PDA < DTM = NTM$

[1] Prepared by R. Inkulu, Dept of Computer Science, IITG, India. http://www.iitg.ac.in/rinkulu/