Turing Machines

Atri Rudra

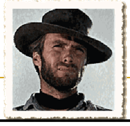
May 23

Announcements

- Change in today's office hours
  - 3:40-4:30pm
  - Instead of 2:30-2:50
- Pick up graded H/W#5
  - If you did not last class
- Pick up solutions to H/W #6

A few words about the homeworks

- First, the "bad" news
  - There is going to be an 8th assignment
  - Handed out this Friday and due the next
- Now, the "good" news
  - I will drop your homework with the lowest score
  - If you are happy with your current scores, you can choose not to turn in the last one



Today's Puzzle

- Show the following equivalence:

≡

In other words…

- Show that the following two models are equivalent
  - TM
  - PDA with two stacks

Last lecture

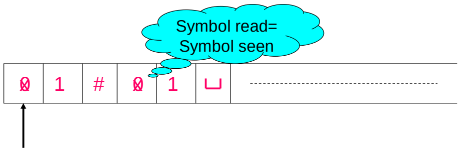
It could have been worse…

What we did manage to talk about…

- Basic “definition” of TMs
- TM= DFA + ∞ memory
  - Input is on the tape to begin with
    - Special blank character
  - Tape is read/write
    - One read/write head
    - Can move on “cell” to the right or left (unless at left most)
  - Two special states: accept, reject
    - “Effect” takes place immediately

An example

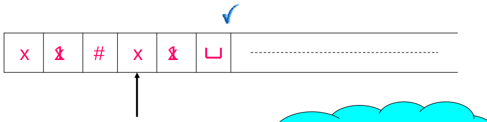
- An example $\{ w\#w \mid w\in \{0,1\}^* \}$



Remember in state: 0

An example

- An example $\{ w\#w \mid w\in \{0,1\}^* \}$
  - Repeat the same thing for the next symbol



Remember in state:

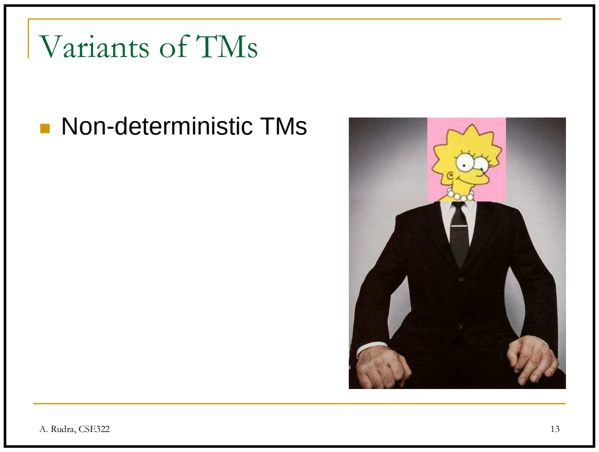
Does not take care of everything

Questions ?

A word of caution…

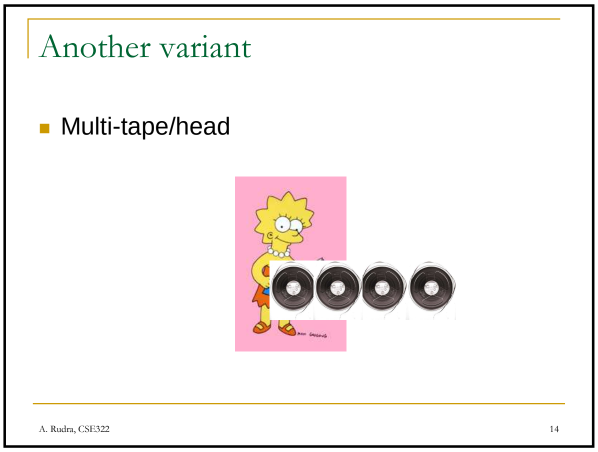
- I want to show you the “cool” stuff
  - There are problems that are “unsolvable”

Stop me if I am going too fast



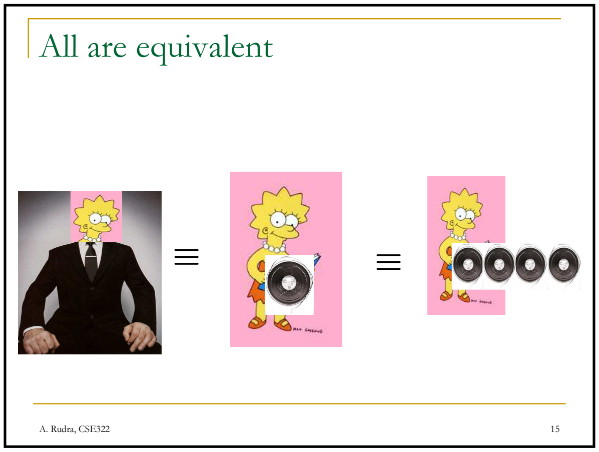
Variants of TMs

- Non-deterministic TMs

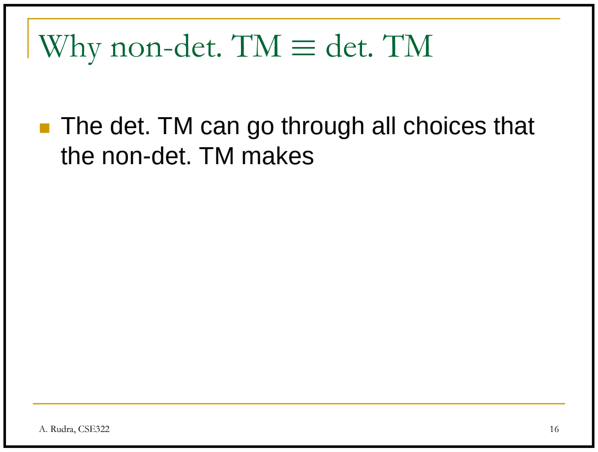


Another variant

- Multi-tape/head

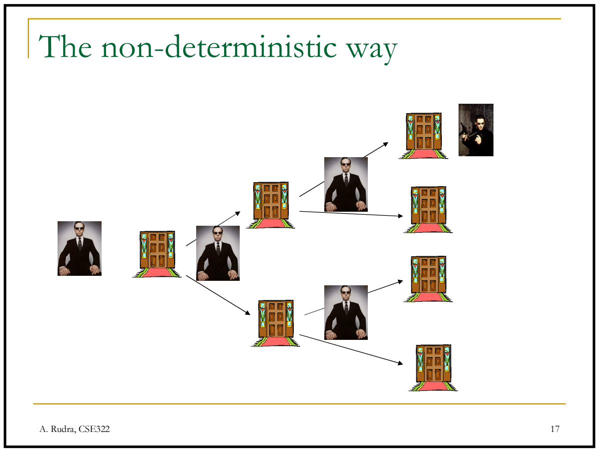


All are equivalent

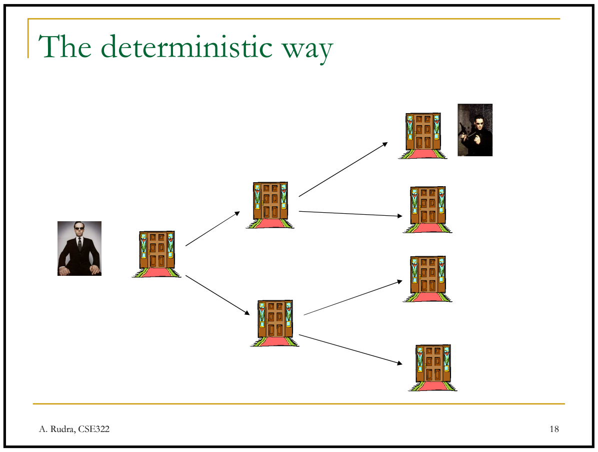


Why non-det. TM ≡ det. TM

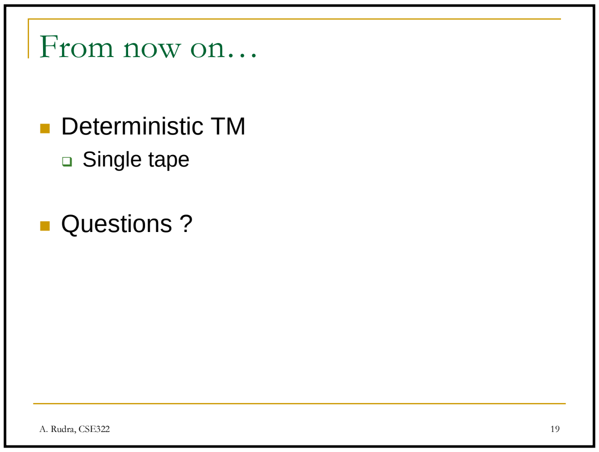
- The det. TM can go through all choices that the non-det. TM makes



The non-deterministic way

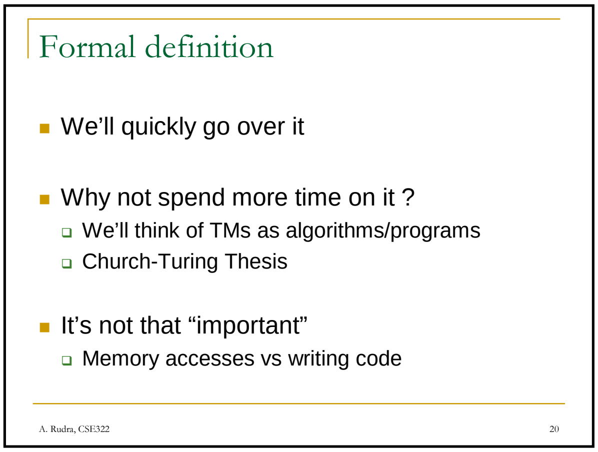


The deterministic way



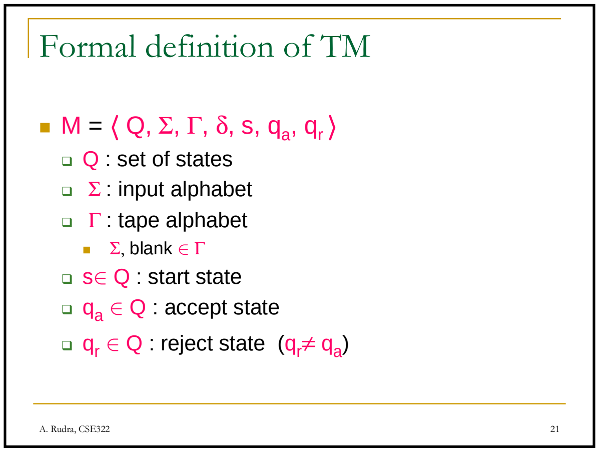
From now on…

- Deterministic TM
  - Single tape
- Questions ?



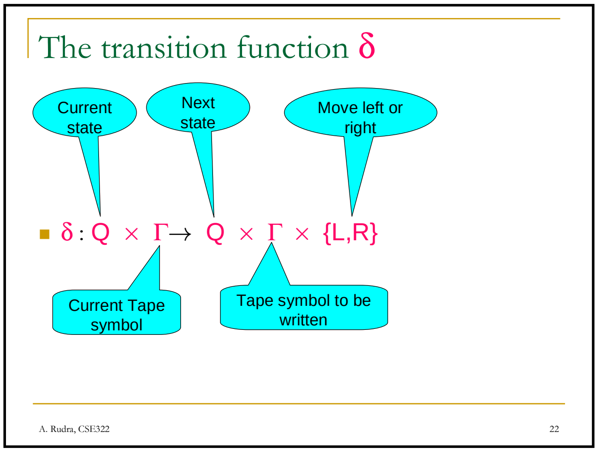
Formal definition

- We'll quickly go over it
- Why not spend more time on it ?
  - We'll think of TMs as algorithms/programs
  - Church-Turing Thesis
- It's not that "important"
  - Memory accesses vs writing code



Formal definition of TM

- $M = \langle Q, \Sigma, \Gamma, \delta, s, q_a, q_r \rangle$
  - Q : set of states
  - Σ : input alphabet
  - Γ : tape alphabet
    - Σ, blank $\in \Gamma$
  - $s \in Q$: start state
  - $q_a \in Q$: accept state
  - $q_r \in Q$: reject state ($q_r \neq q_a$)



The transition function δ

Current state

Next state

Move left or right

- $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L,R\}$

Current Tape symbol

Tape symbol to be written