CPSC 421/501

Ch. 3 : Turing machines

$$M = (Q, \Sigma, \Gamma, \delta, q_0, q_{acc}, q_{rej})$$

Γ: tape alphabet

q_{acc}: accepting state

q_{rej}: rejecting state

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

Much closer to a { Python, C++, Javascript, etc. } algorithm. (memory is not random access).

Define: { accept, reject, loops / doesn't halt }, { recognize, decide },

universal machine, delightful machines,

...

(return to "Uncomputability in CPSC 421/501").

Examples:

$\Sigma = \{a\}$, $L = \{a\}$

$\Sigma = \{a, b\}$, $L_k = \{ s \mid$ k^{th} to last symbol of s in an "a" $\}$

$\Sigma = \{a, b\}$, $L = \{$ PALINDROME Σ, $\{a^n b^n \mid n \in \mathbb{N}\}$, etc. $\}$

i.e. no problem set due Oct 30

Admin:

No problem set to appear this Thursday; instead look at midterm study guide 2025, to appear tomorrow.

Bonus problem for HW 7:

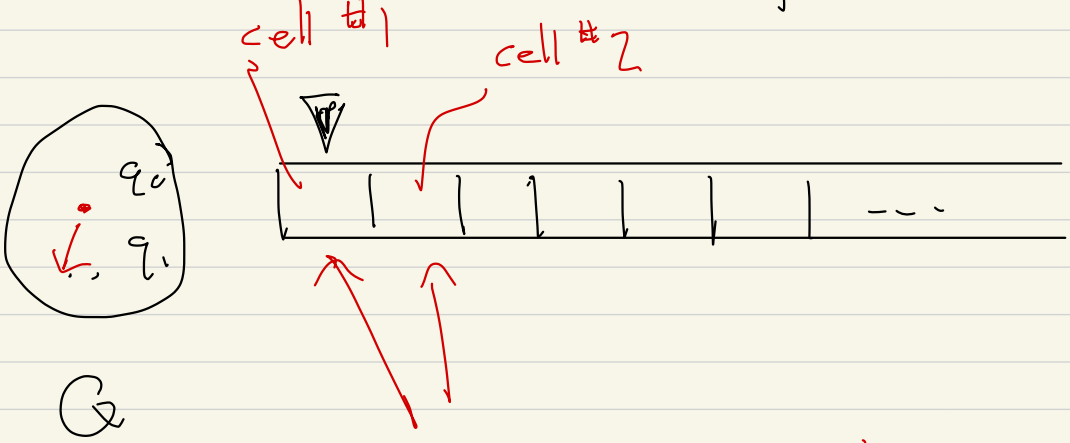
now due Tuesday, (Oct 28)

M	W	F
	22	
27	29	31

I'll take questions on study materials ←

$L = \{a\}$; L_k ... ; $\{a^n b^n \mid n \in \mathbb{N}\}$

What is a T.M. Turing machine



Q

like lines in a program (Python, Javascript)

cell = tape cell, and each holds an element of Γ

Σ = input alphabet, e.g. $\Sigma = \{a\}$, $\Sigma = \{a, b\}$

Γ = tape alphabet, consist of

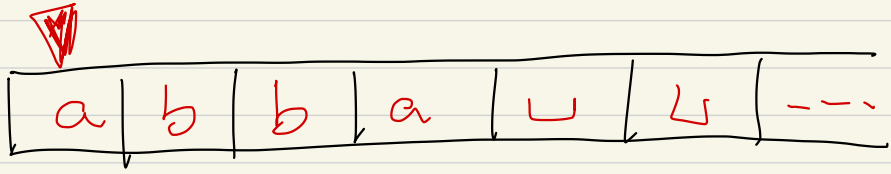
$$\Sigma \cup \{ \sqcup \} \cup \{ \quad \}$$

blank symbol

any additional set of symbols (finitely many)

Input, $\Sigma = \{a, b\}$, is presented on the tape as

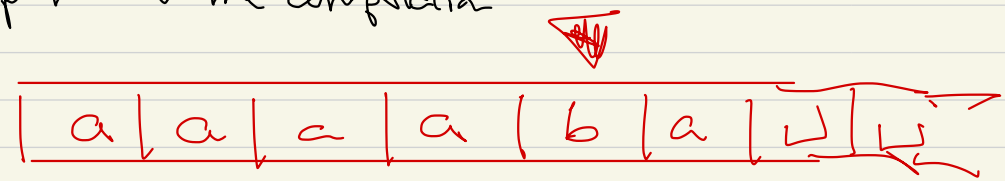
$S = abba$



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

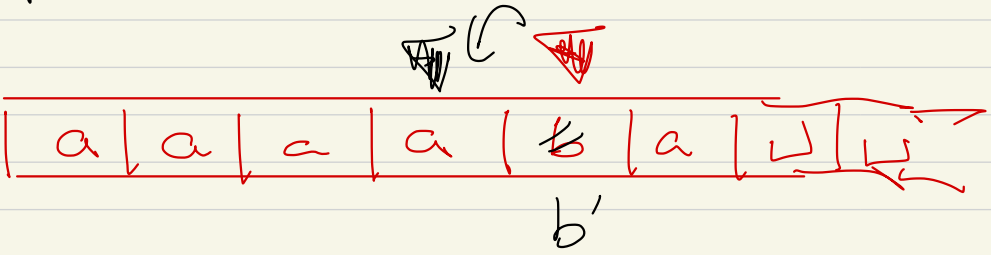
current state, symb — new — movement

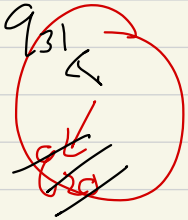
step 12 of the computation



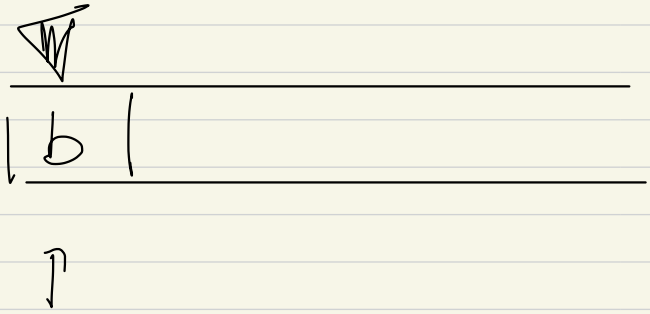
if $\delta(q_{20}, b) = (q_{31}, b', L)$

step 13





Question: what if



and move L

Limited Def: A L (Left movement) at cell 1 makes program invalid

this is our convention

Wider Def: -- ~ we just say oh cell 1. Problem: we won't know we are at cell 1.

Rem: (MJJ): If you can't move left, only right, then you can only recognize regular languages

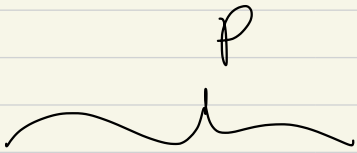
[This is an exercise in [Sip]]

Rem: [Auden] just allowing L (and R) movement makes T.m.'s as powerful as Python C++ programs !

So P
poly time on a T.m
= --- Python alg

NP
nondet poly time T.m.
= --- Python alg

Conceptually



Reg Lang $\subset$ Poly time T.m

$=$ " " Python

$\subset$ Non-det poly time T.m

— — Python

NP

$\subset$ non-deterministically $\subset$ non-recognizable

Class Ends

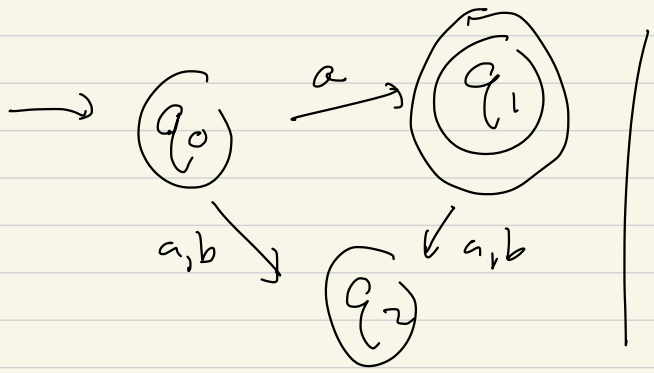
Problem 1(b)(iii) HW 7

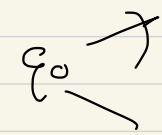
(J.S.)

Section 1.3 → Section 1.2.

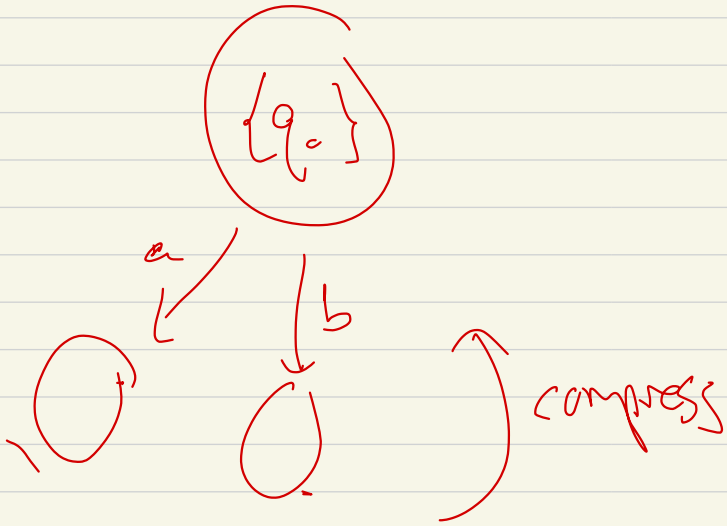
↑ regular expressions

↑ DFA, NFA





16 states



"needless" ← unreachable is better

might be ambiguous

Clarify