CPSC 421/501 Oct 13, 2020

Start Chapter 3 — Turing machines

§3.1 Turing machines as algorithms

- Formally $(Q, \Sigma, \Gamma, \delta, q_0, q_{acc}, q_{rej})$

  plus blank symbol $\sqcup$

  Σ = input alphabet $\subset \Gamma$ tape alphabet

  (includes also $\sqcup$)

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

- Examples to recognize
  - $\{w \in \{a,b\}^* \mid \#a\text{'s in } w = \#b\text{'s}\}$
  - $\{0^n 1^n\}$

- TM descriptions
  - High level
  - Implementation level
  - Formal description (give δ)

§ 3.2 : multitape & non-deterministic Turing machines

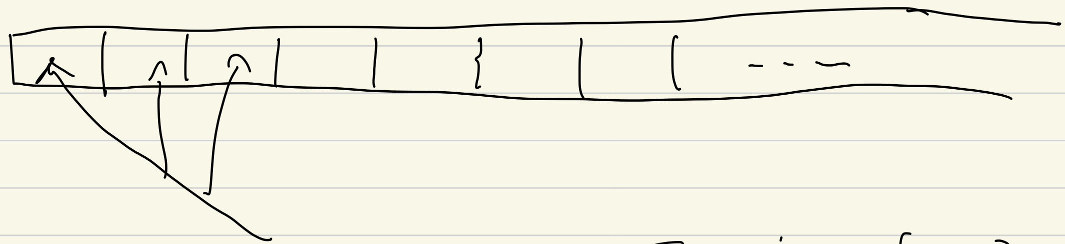
§ 3.3 : descriptions of $\left\{ \begin{array}{l} \text{graphs} \\ \text{Boolean formulas} \\ \text{etc.} \end{array} \right.$

Midterm on Nov 5, will cover material <u>up</u> to October 22 (end of next week)

Details to follow...

Turing Machines:

Part: Input/Work Tape



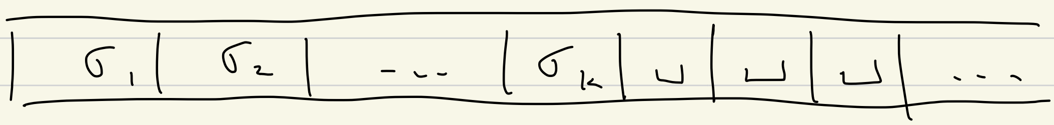
Σ alphabet for input, larger alphabet $\Gamma = \Sigma \cup \{\sqcup\} \cup \{\text{additional symbols}\}$

$\sqcup$ blank symbol

Rules for input: Say $\Sigma = \{0,1\}$

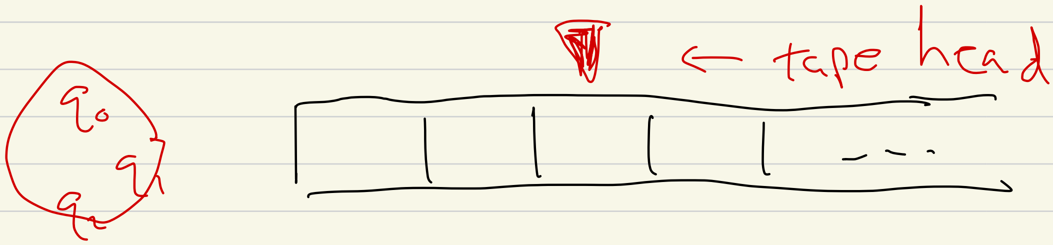
interested in $L = \{ 0^n 1^n \mid n \in \mathbb{Z} \}$

input $\omega \in \{0,1\}^*$, $\omega = \sigma_1 \ldots \sigma_k$



In addition:

Q set of states, finite set, intuitively Q is "the program"



tells you which state

Formally: a Turing machine: 7-tuple

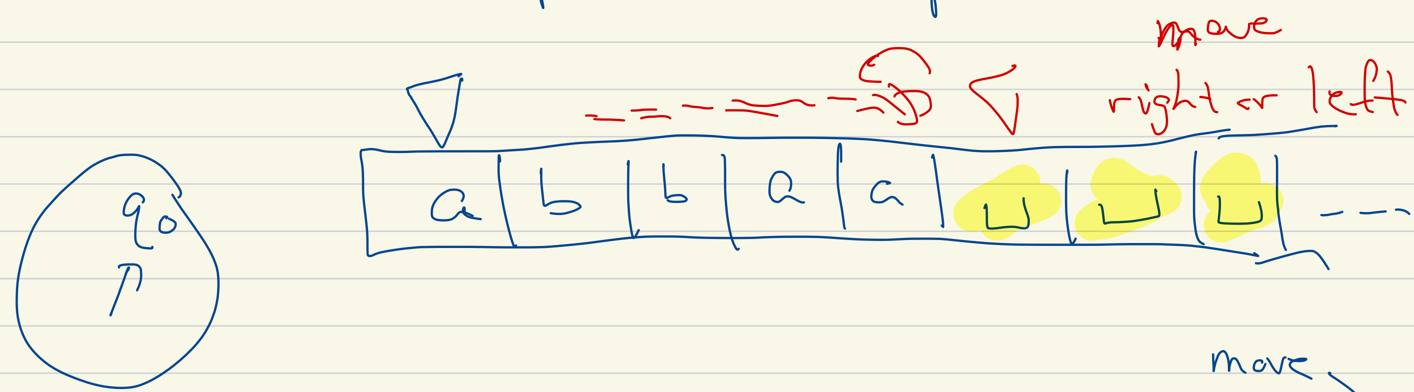
$$(Q, \Sigma, \Gamma, \delta, q_0, q_{acc}, q_{rej})$$

- Q: states
- Σ: input alphabet
- Γ: tape cell alphabet
- q_0: initial
- q_{acc}: state where you accept
- q_{rej}: state where you reject

Example: Want to recognize: $a\{a,b\}^* b$

(this is a regular language)

Turing machine approach: input



transition function: $Q \times \Gamma \; : \; Q \times \Gamma \times \{L, R\}$

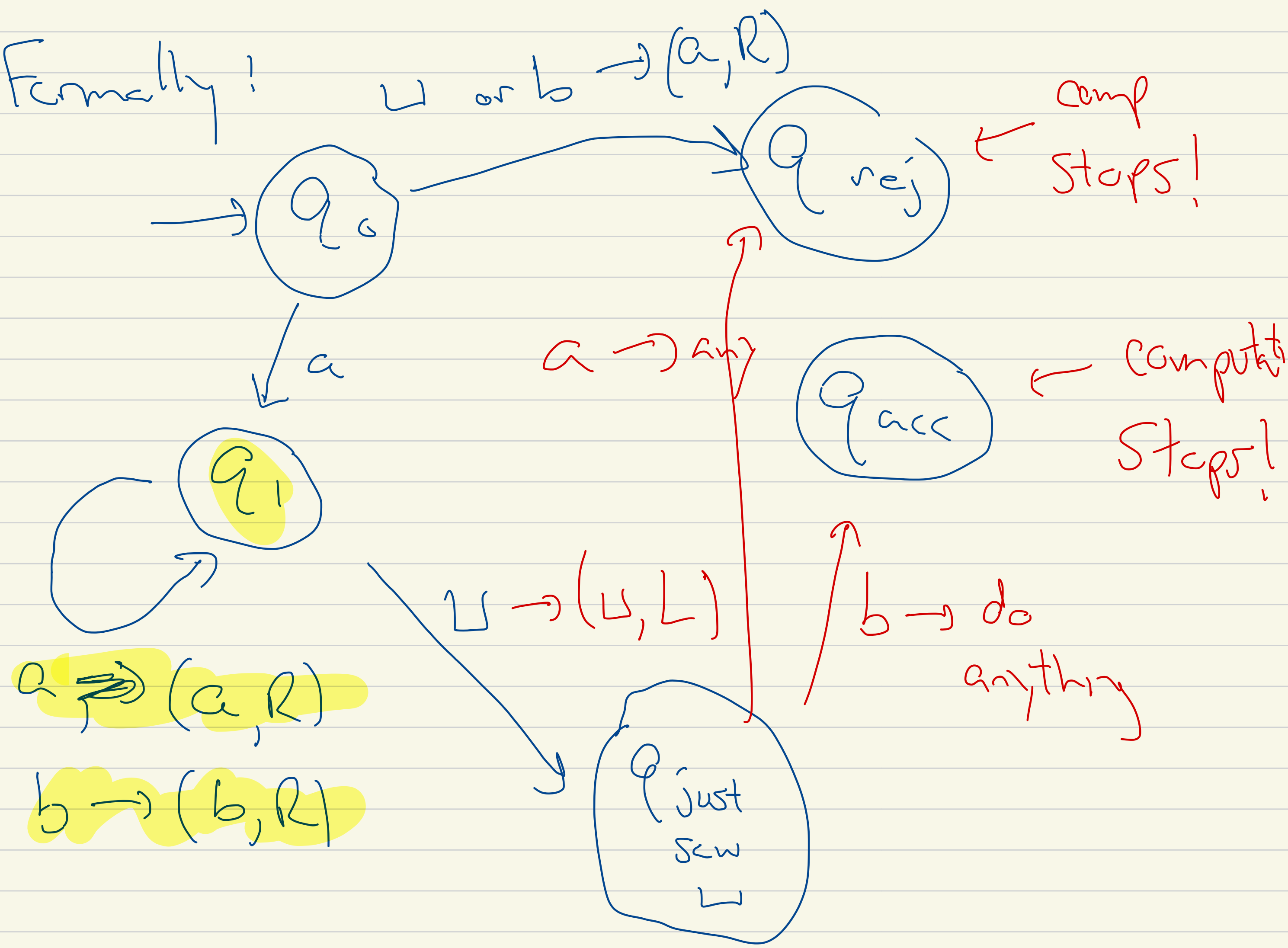
- Q: state set
- Γ: tape cell alphabet
- Q: new state
- Γ: overwrite on tape cell
- $\{L, R\}$: move

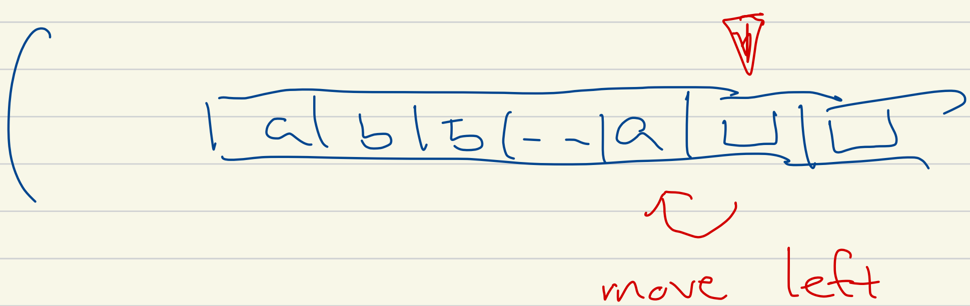
To recognize $L = a\{a,b\}^* b$

"algorithm"

1. Make sure input begins "a" (otherwise reject)
2. Move to the end of tape
3. Make sure last letter is a "b" (then accept, otherwise reject)

Formally!





$\Gamma = \{a, b, \sqcup\}$

Describe with table

	a	b	$\sqcup$
q_0	etc.	etc.	etc.
q_1	q_1, a, R	q_1, b, R	$q_{\text{just saw } \sqcup}, \sqcup, L$
$q_{\text{just saw } \sqcup}$			
q_{acc}			
q_{rej}			

$$Q \times \Gamma \times \{L, R\}$$

Now! We want to convince ourselves that Turing machines can solve any decision problem ($\Sigma^* \longrightarrow \{\text{accept}, \text{reject}\}$) that you could write in Python, C++, Javascript, etc...

Describe Turing machines!

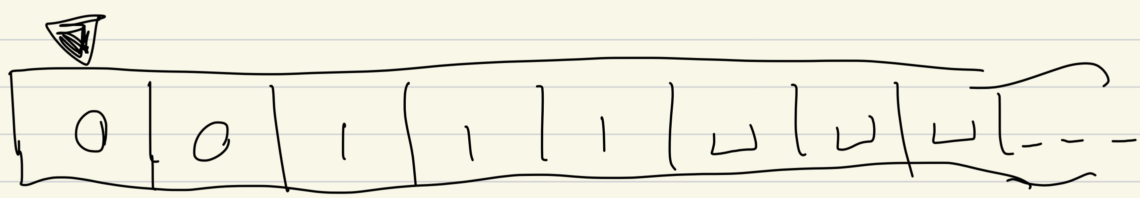
- Formal: describe δ (and explain algorithm)
- Middle: Implementation
- High-level

vague

Take $\{0^n 1^n \mid n \in \mathbb{N}\} = \{01, 0011, 0^3 1^3, \ldots\}$

give Turing machine algorithm: $\Sigma = \{0, 1\}$

given



start in q_0

input 00111 (should reject)

$\Gamma = \{0, 1, \sqcup,$ any other finite number of symbols $\}$

e.g. $\{0, 1, 2, \sqcup\}$ OK

$\{0, 1, 2, 3, \ldots, \sqcup\}$

(Γ finite, Q finite)

Compare Turing machines vs. C++ programs:

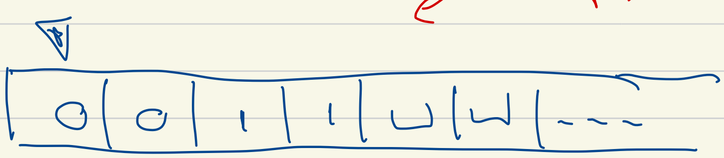
	Turing machines	C++ programs
description, computation:	simple	much more complicated
real-world performance:	unrealistically slow	better idea

Idea to recognizes $\{01, 0011, 000111, \ldots\}$

initially

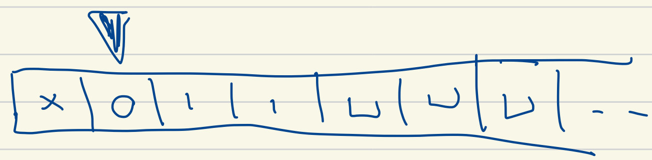
accept, but reject $00111\sqcup\sqcup\ldots$

(1) q_0

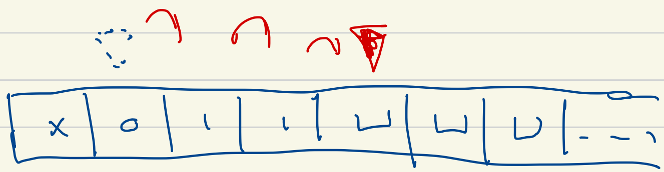


turn this 0 into x
move right

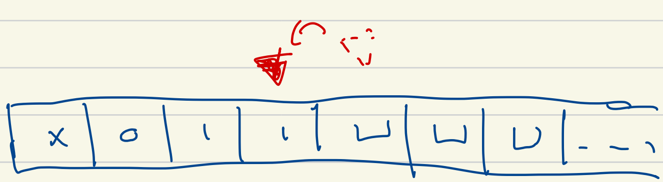
(2)



(3) continue moving right until hit ⊔

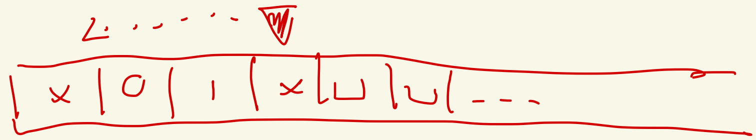


(4) move one left



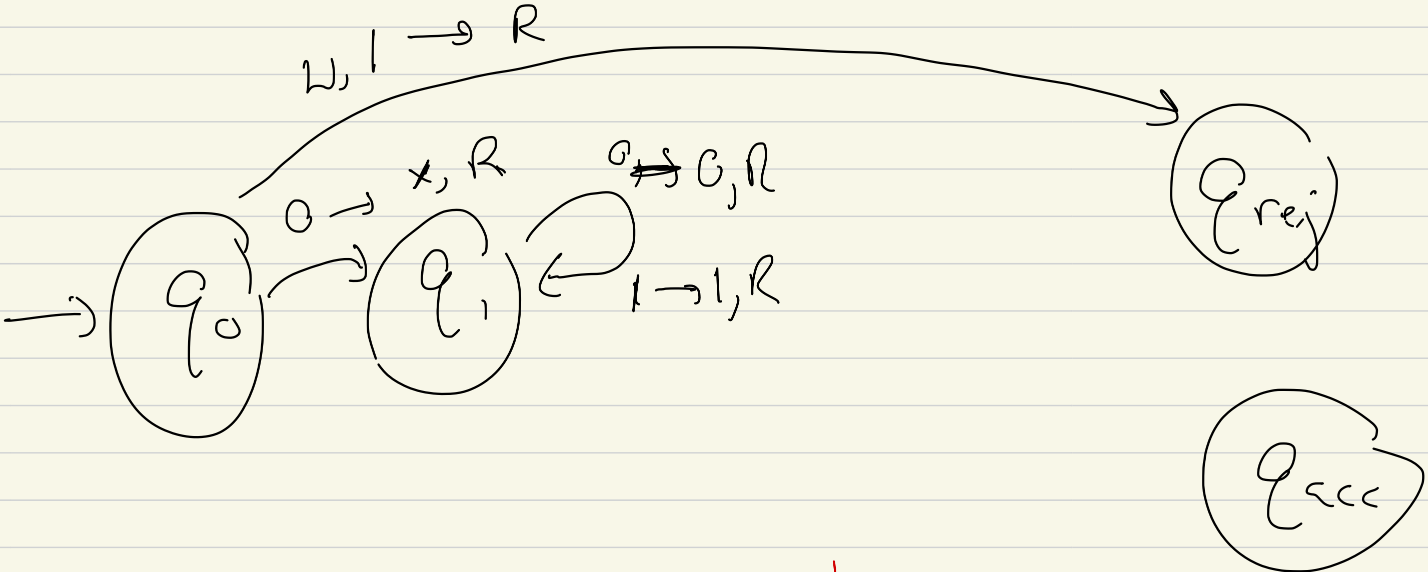
(5) write an x (or y or ⊔)

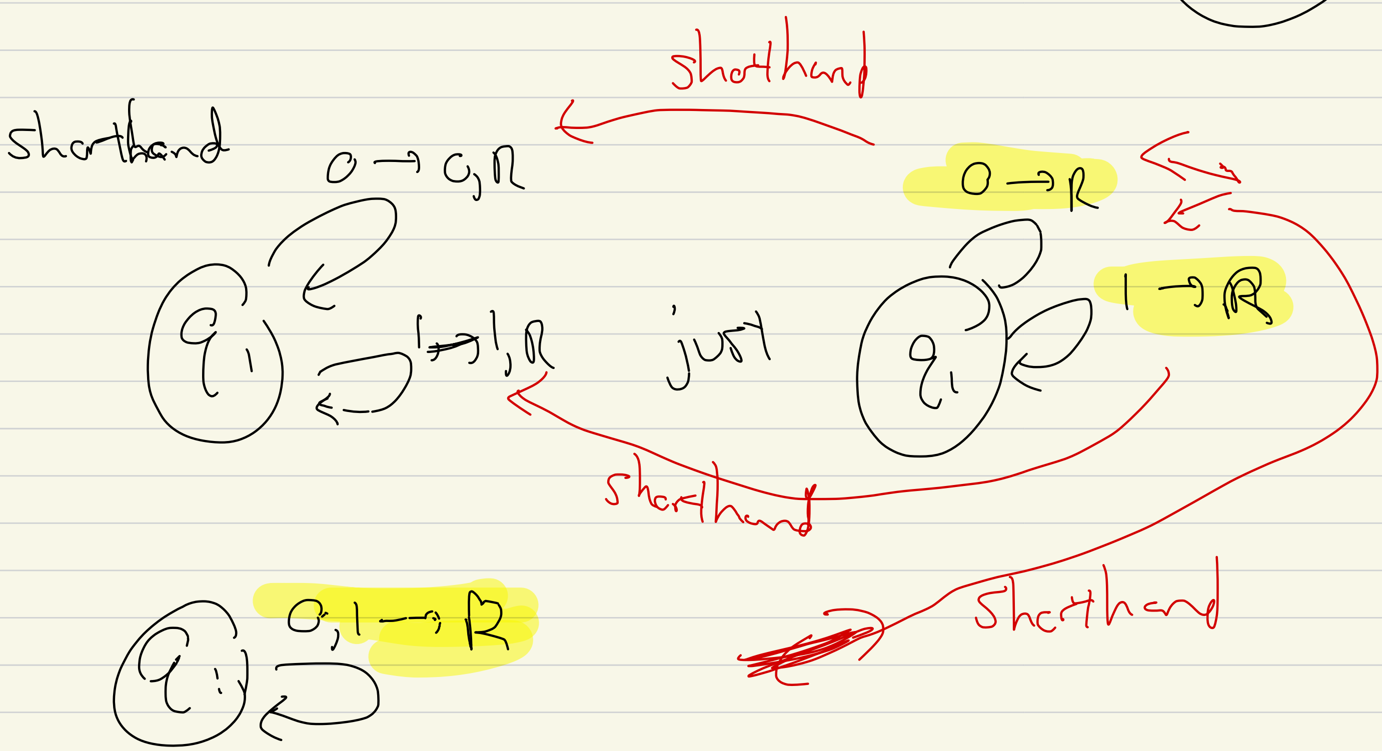
⑥ write back to first x, then sort of start again



[Break 10:20 am, back in 5 min]

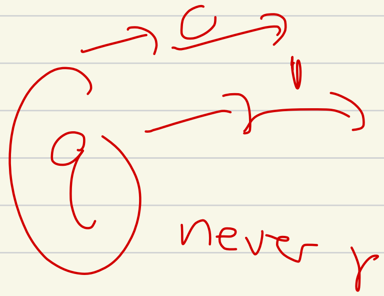
State diagram: $L = \{01, 0011, \ldots\}$, use textbook shorthand





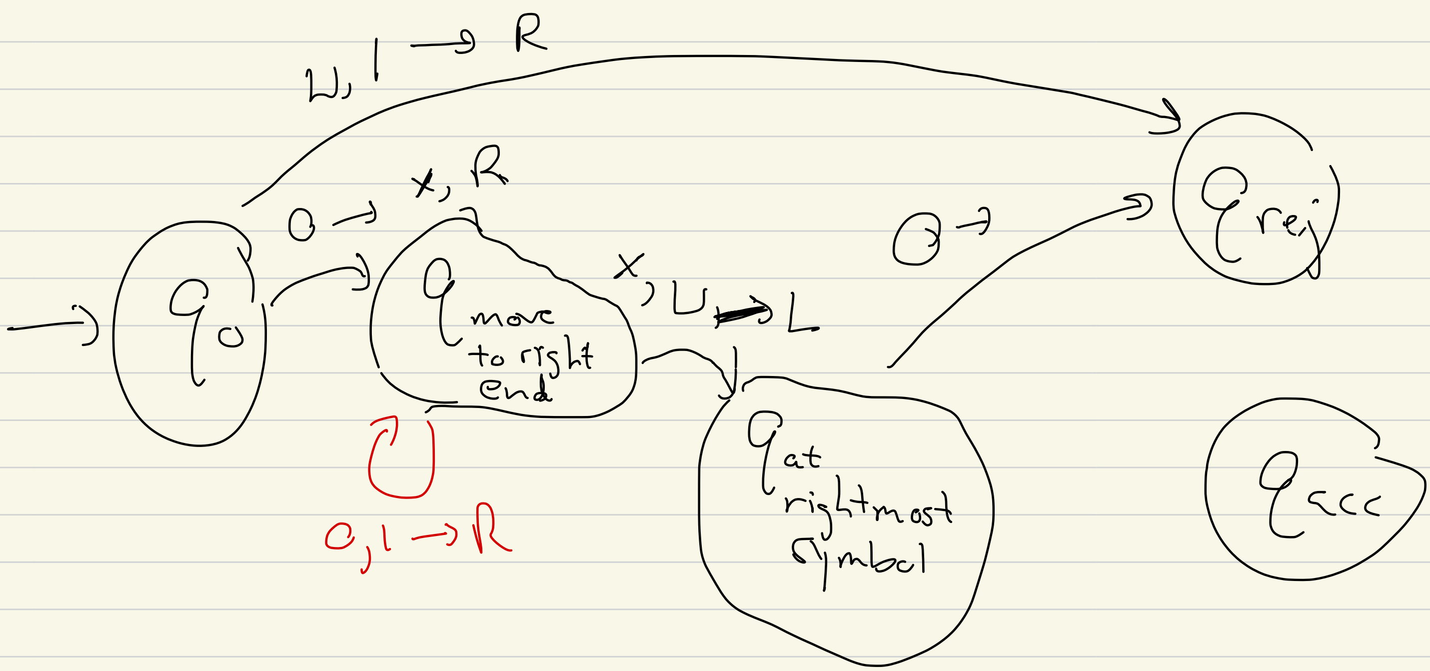
shorthand: if $(q,\gamma) \in Q \times \Gamma$

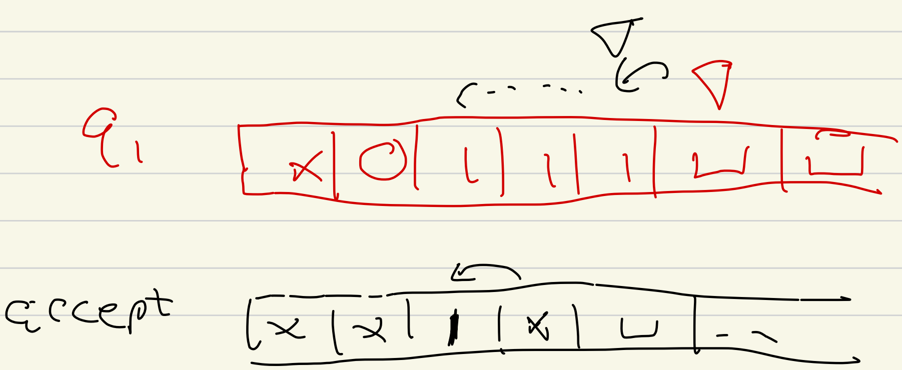
is never encountered



it's OK to leave it out

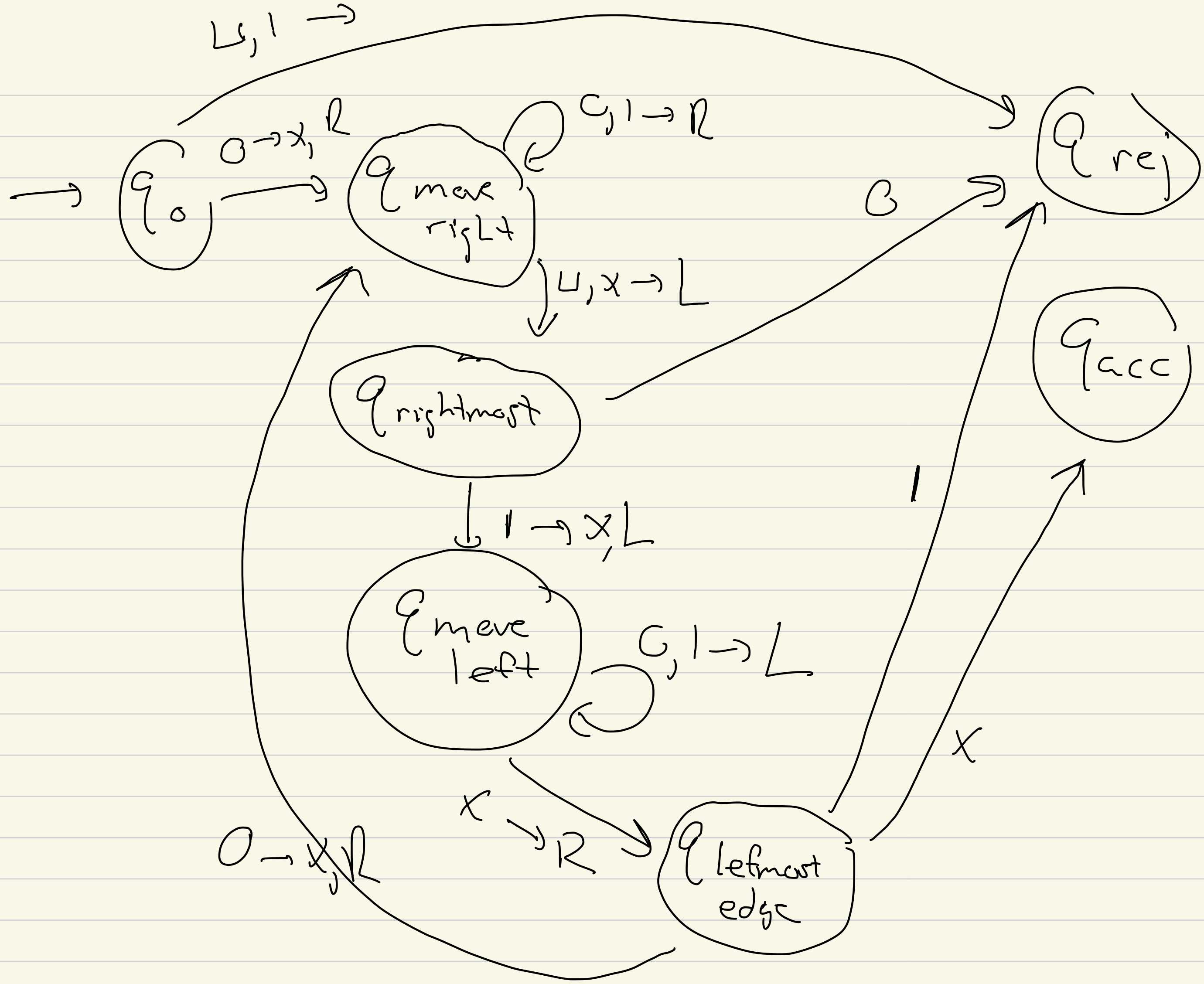
So far: $L = \{01, 0011, \ldots\}$, use textbook shorthand





might see

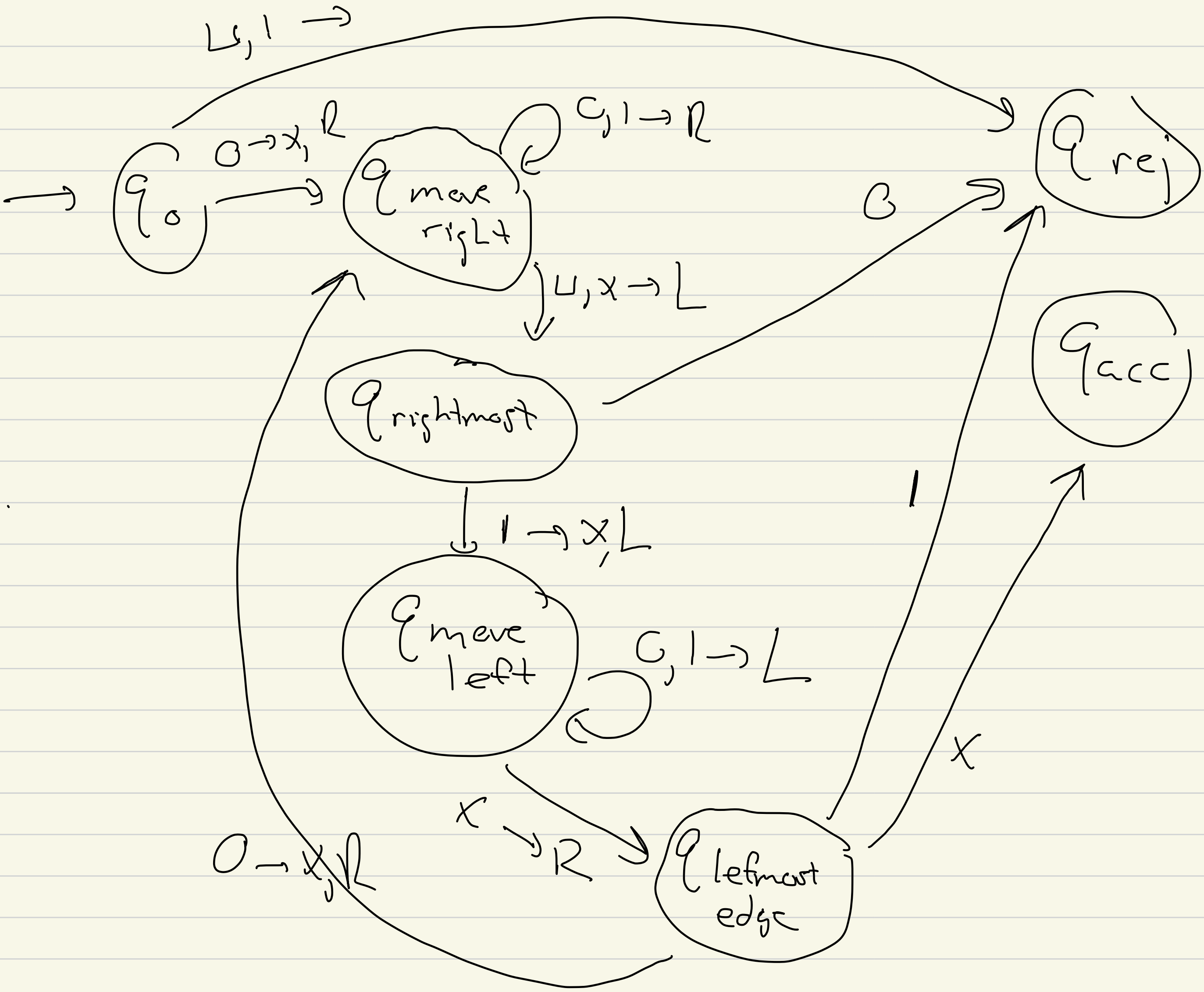
xx0011xx ⊔



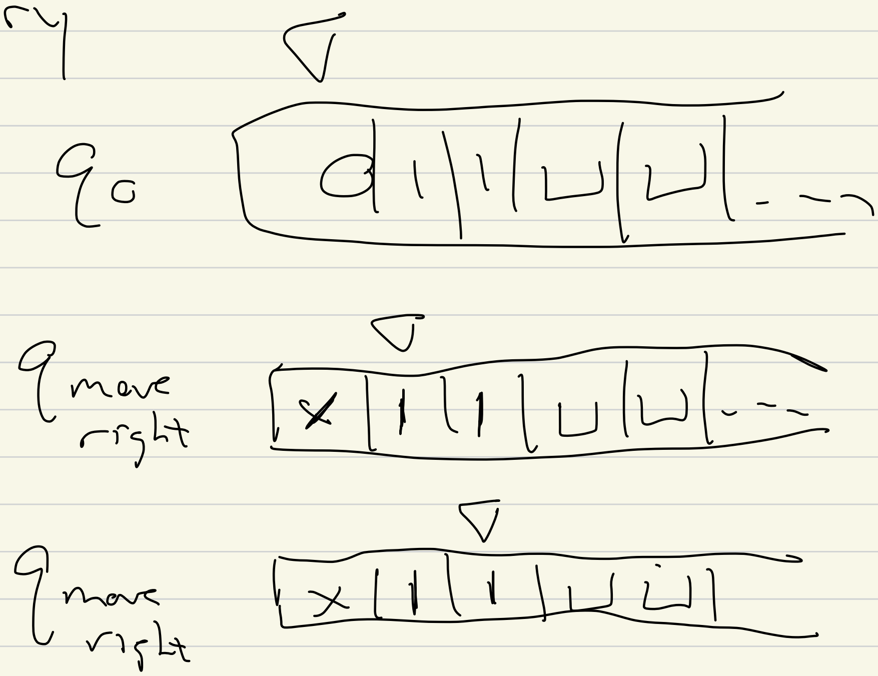
0 1 ⊔ ⊔ → x 1 ⊔ ⊔ → x 1 ⊔ ⊔ → x x ⊔ ⊔

0 0 1 1 ⊔ → x 0 1 1 ⊔ ⊔ → x 0 1 1 ⊔ ⊔

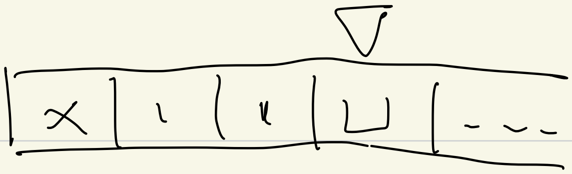
x 0 1 x


⊔, 1 →
0 → x, R
0, 1 → R
q0
q move right
q rej
q acc
⊔, x → L
0
q rightmost
1 → x, L
q move left
0, 1 → L
1
x
x → R
0 → x, R
q leftmost edge

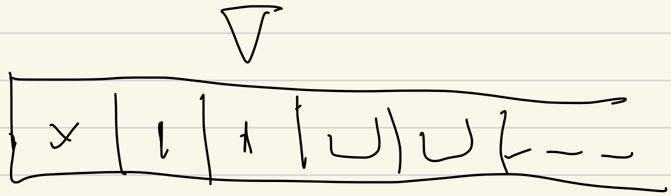
try


q0
q move right
q move right

$q_{\text{move right}}$



$q_{\text{rightmost}}$



---

next time! look a bit more

then introduce 2-tape machines

Individual, Q2! 51 states, not 50 states