1. Recap

DFA/NFA : computational model for devices with limited memory.

PDA : Devices that has unlimited stack memory.

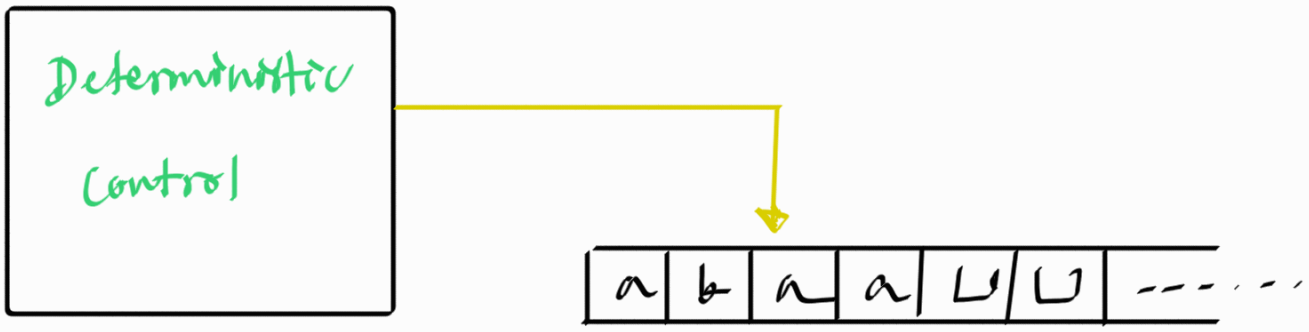
These models have limitations

These are too restrictive.

$A = \{ a^n b^n c^n \mid n \geq 0 \}$

Model for general computers

- has unlimited and unrestricted memory.



Infinite tape

1. Initially tape contains the input, and blank everywhere else.
2. Machine can read/write information on tape.
3. To read information from the tape, machine can move its head both to the left and to the right.
4. Special states for accepting and rejecting take effect immediately.
5. Machine computes until it produces an output.

- Accept / Reject

Ex: $A = \{0^{2^n} \mid n \geq 0\}$

$0 \in A$

$00 \in A$

$0000 \in A$

$00000000 \in A$

general programming language

C, C++, Java, ...

$x = 8$

$x \,\%\, 2 = 0$

$x \leftarrow 3$

test if # of 0's $(x) = 2^n$

Algo(x):

$x \,\%\, 2 == 1$, No

$x \,\%\, 2 == 0$, recurse on $x/2$

Any function that can be computed by an intuitive algorithm can be computed by an Turing machine.

intuitive algorithm

1. A step-by-step finite process
2. Using discrete, finite symbols
3. Running in finite time.

Differences between finite automata and Turing Machines.

1. Turing Machine can both write on tape and read from tape.

2. Read-write head can move both left and right.

3. Tape is infinite.

4. Special states for accepting and rejecting take effect immediately.

Def

A turing Machine is a 7-tuple, $(Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$, where

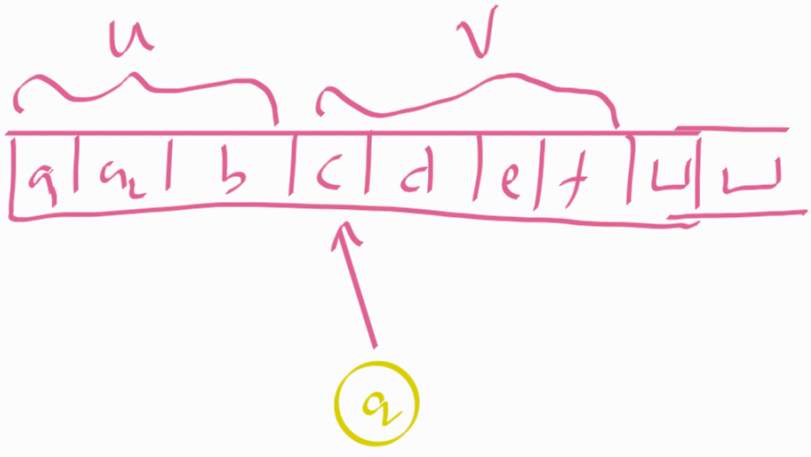
1. Q is the set of states.
2. Σ is the input alphabet
3. Γ is the tape alphabet, where $\sqcup \in \Gamma$ and $\Sigma \subseteq \Gamma$ ← $\sqcup$ blank symbol
4. $\delta : Q \times \Gamma \longrightarrow Q \times \Gamma \times \{L, R\}$
5. $q_0 \in Q$ is the start state
6. $q_{accept} \in Q$ is the accept state
7. $q_{reject} \in Q$ is the reject state

$q_{accept} \neq q_{reject}$

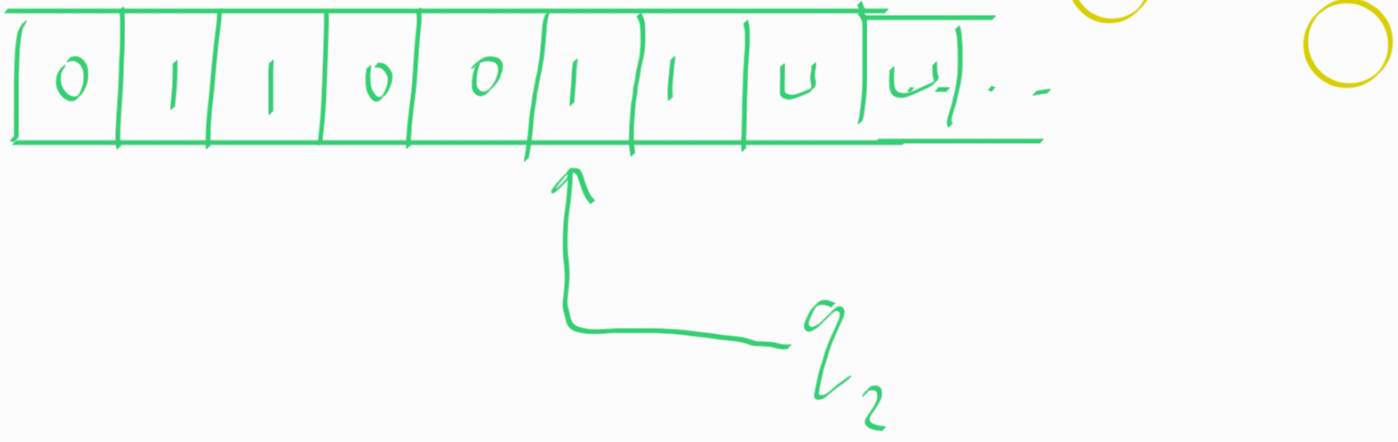
How to represent the state of the machine (configuration)

we have 3 items.

(i) current state
(ii) tape content
(iii) current head location



u q v

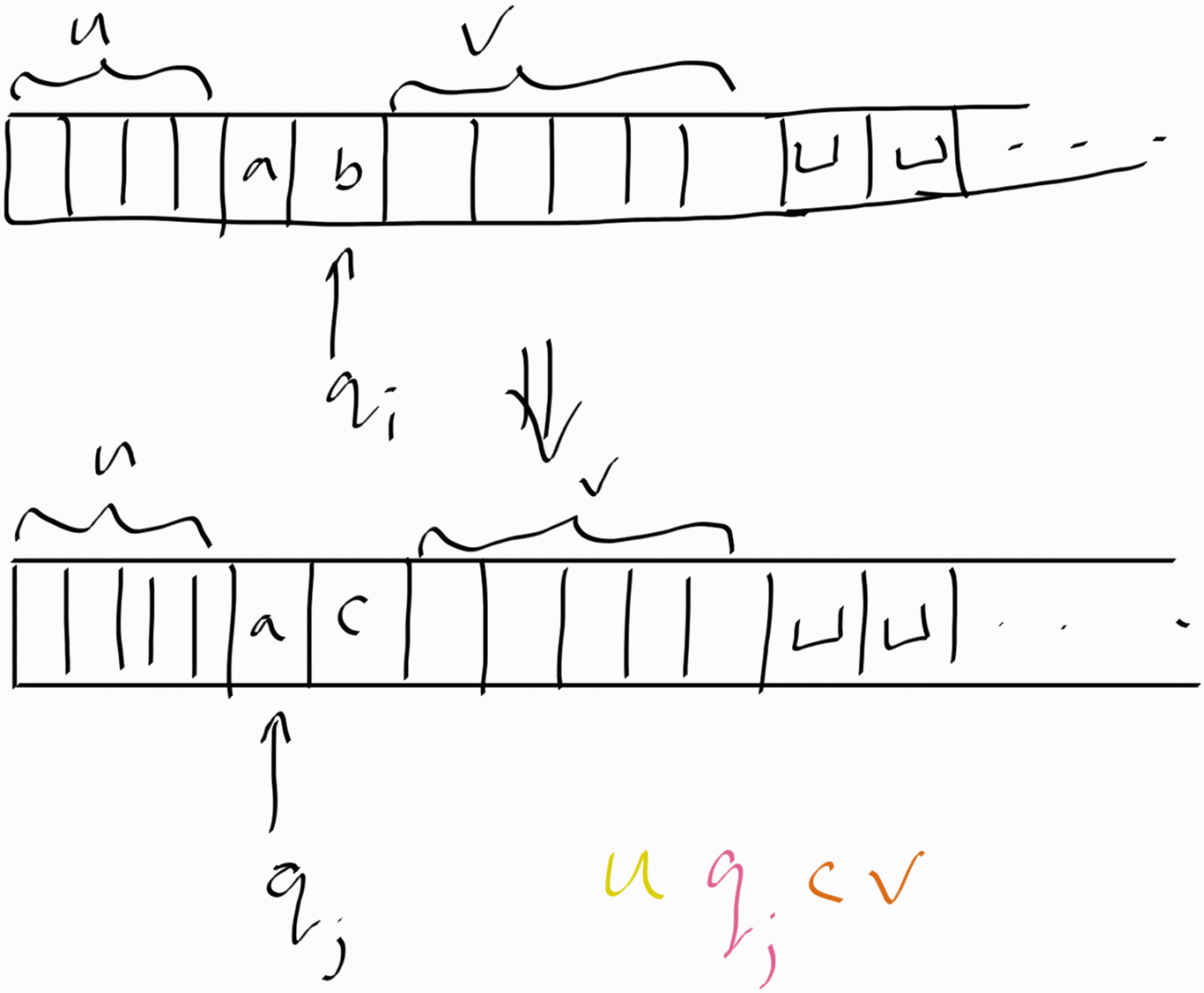


0 1 1 0 0 q_2 1 1

we say configuration C_1 yields configuration C_2 if the turing machine can legally go from C_1 to C_2 in a single step.

If we have $\delta(q_i, b) = (q_j, c, L)$

$u\ a\ q_i\ b\ v$ yields $u\ q_j\ a\ c\ v$



$u\ q_j\ c\ v$

What is the start configuration on string w.

$q_0\ w$

$A = \{0^{2^n} \mid n \geq 0\}$

M = "on input string w"

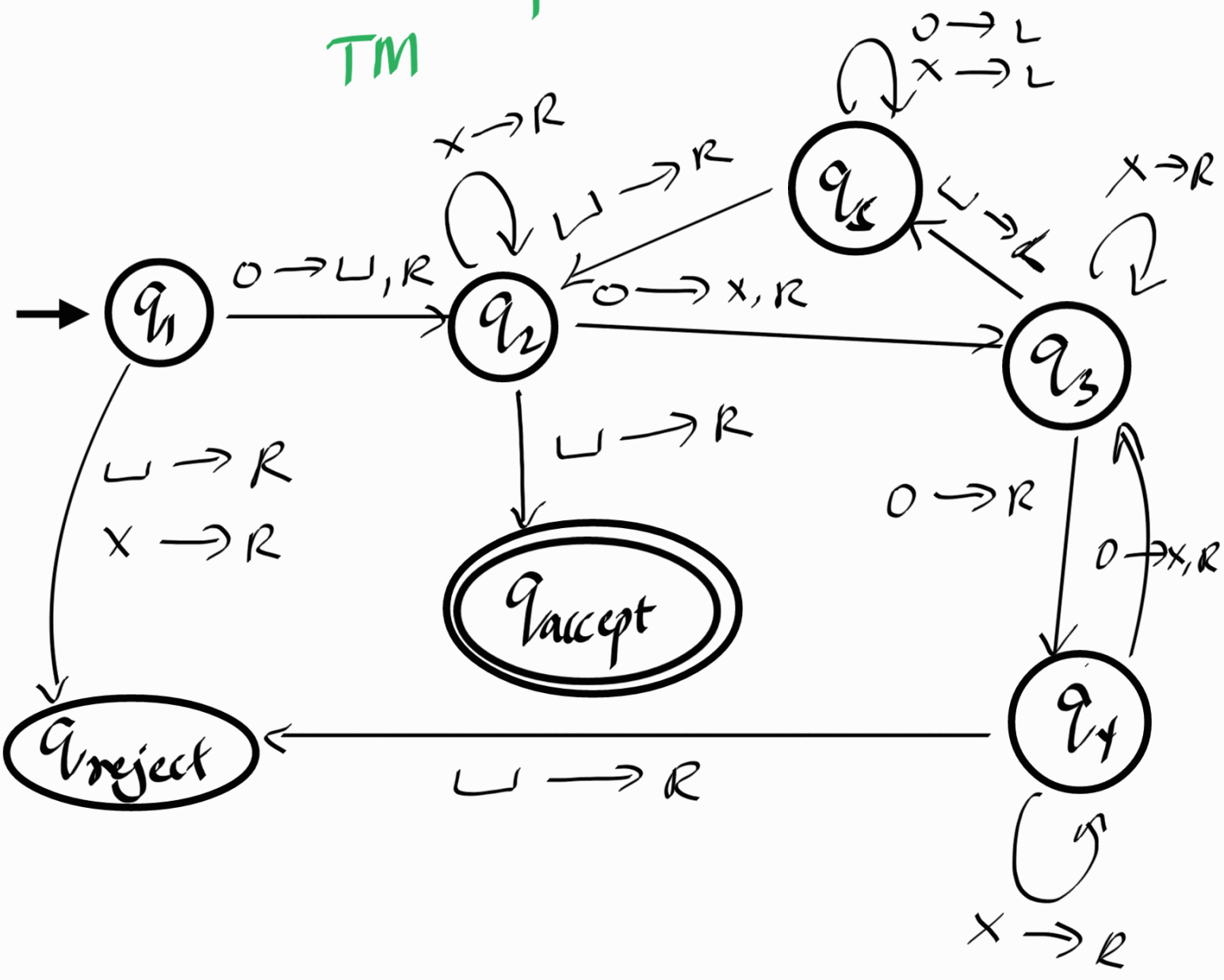
1. Sweep from left to right across the tape, crossing off every other zero.
2. If in stage 1, the tape contains a single 0, accept
3. If in stage 1, the tape contains more than single 0 and the # of 0's was odd, reject
4. Return the head to the left hand end of the tape
5. Go to stage 1.

(Implementation level description of a TM)

Note that whenever we mark a 0, we have to see 2 zeros.
∴ when we mark a 0, we know that we have seen even # of zero's.

* Each iteration halves the number of 0's.
* If the remaining # of 0's is odd and greater than 1, we reject.
* we do this until we have only one 0 left.

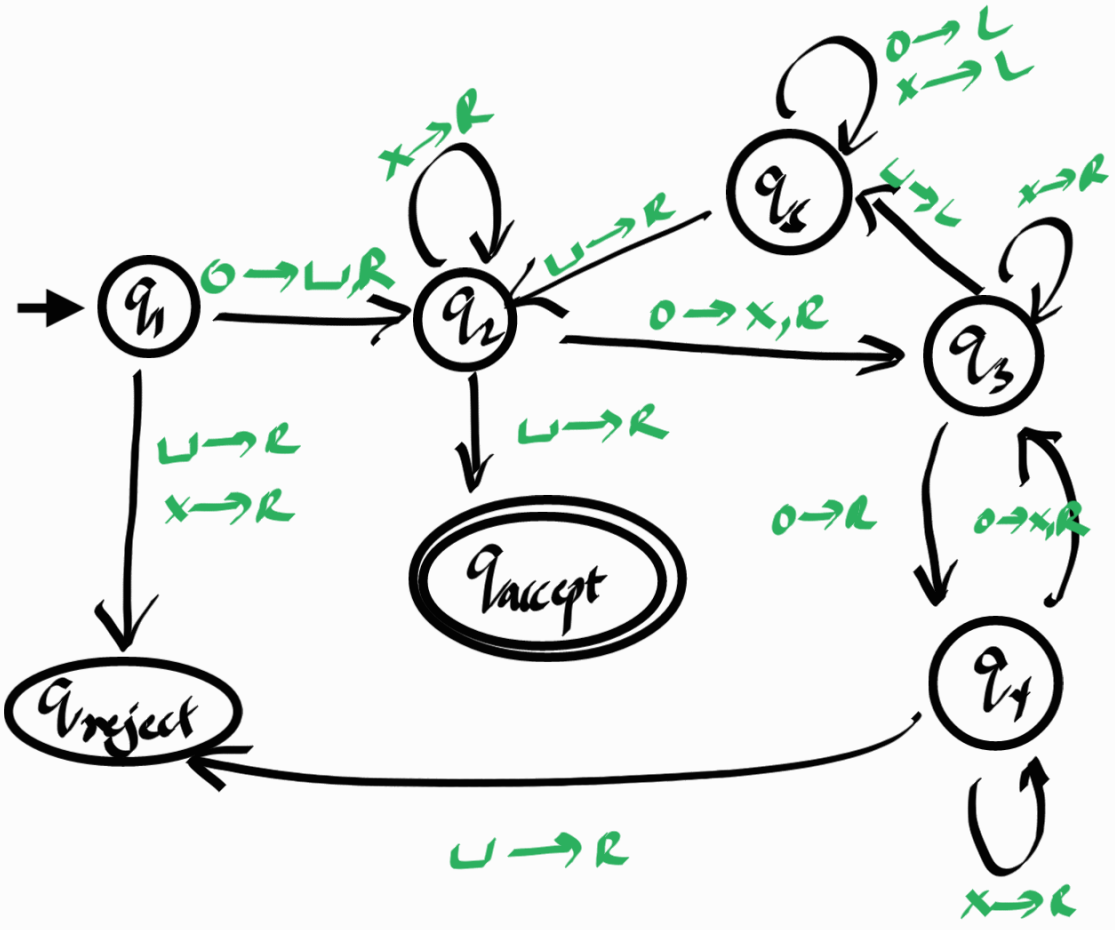
formal description of this TM



$a \longrightarrow \tau, R$ means

read a from head writ τ to stack and move head to write.

$a \longrightarrow L$ means read o from head don't change tape and move left.


q1
q2
q3
q4
qaccept
qreject
q4
0 → L
x → L
x → R
0 → ␣R
␣ → R
␣ → R
␣ → R
0 → x,R
␣ → R
x → R
0 → R
0 → x,R
␣ → R
x → R


␣ 0 0 ␣
q1
␣ 0 ␣
q2
␣ x ␣
q3


␣ x ␣
q5
␣ x ␣
q5
␣ x ␣
q2


␣ x ␣
q2
␣ x ␣ ␣
qaccept

The collection of strings that turing machine M accepts is the language of M, or the language recognized by M, denoted by L(M).

Unlike, DFA, NFA and PDAs, given an input there are 3 possibilities for a turing machine.

(i) Accept

(ii) Reject

(iii) Loop: machine simply does not halt.

given an input any of these 3 things can happen.

Def

We call a language L is Turing-Recognizable if some turing machine recognizes it.

either accept or reject

Turing machines that halts on all inputs are called deciders.

Def

We call a Language L is Turing-Decidable or simply decidable if some turing machine decides it.

Given Language L and turing machine M.

If M decides language L, then

if $x \in L$ M accepts x

if $x \notin L$ M rejects x

If M recognizes language L, then

If $x \in L$ M accepts x

If $x \notin L$ M either rejects x
or
loops indefinitely.

Ex: check if $p(x) = c_0x^n + c_1x^{n-1} + \cdots + c_{n-1}x + c_n$

has an integral root x_0, i.e.,

checking if there is an integer x_0 s.t $p(x_0) = 0$

Algorithm (TM)

check if x_0 is an integral root for $x_0 = 0, \pm 1, \pm 2, \pm 3, \pm 4, \cdots$,

Q1: Does this algorithm provide a recognizer?

Yes, If $x_0 = 2001$, algorithm will find it.

Q2: Does this algorithm provide a decider?

No, because if the $p(x)$ does not have an integral root this will loop forever.

This algorithm/TM is a recognizer.