Course Website:

http:// picasso.cs.qc.cuny.edu/cs722.

username: Complexity = cs722

password: NP-completeness722

8/29 CS722.

Computability: Studies what is and is not computable *in principle* by idealized digital computers ("algorithms") *without regard to* the amount of resources used (eg. time, memory space)

Structures and classification of such problems

pioneers: Turing – Turing machines.

Church – lambda calculus.

~~Post~~ Post – formal rewriting system.

1930's

These tree have been shown to be equivalent

Complexity Theory:

Classify computable functions by the amount of resource used and study their structures and relationships. "Complexity classes"

Complexity measure = the kind of resource studied.

time – in this course measured by the # of Turing machine state-transition steps. (time complexity)

memory space – " " " the # of " " tape cells used space "

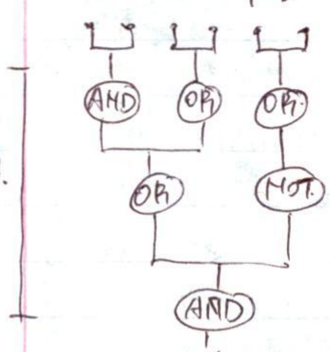
\# of logic gates used — *size complexity* = total # gates used.

(AND, OR, NOT) *parallel-time complexity* = height of circuit (also called depth complexity)

circuit consisting of these gates. circuit complexity

6 bits input

depth. height. =3



size complexity = 6

paralle-time complexity = 3 (depth complexity)

value computed

1960's levin...

communication complexity: pioneers: Cobham, Stearns, Hartmanis, Edmonds, Cook.

CompClasses.html (Outside P)

EXPTIME: $O(a^n)$ 看网页上的 conjecture 很重要!

P : $O(n^k)$ P = NP.

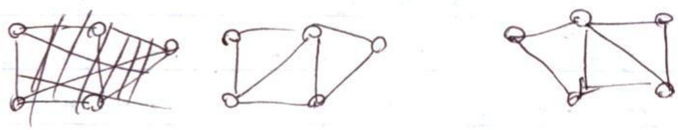
☆. P is the proper subset of EXPTIME $\boxed{\text{Surprising fact!}}$

$P \cap NP = \emptyset$

go through inside/ outside p.

graph isomorphism

decide if two given graphs are isomorphic (同构).



Review of Turing machines

A deterministic Turing machine, DTM, is a $\boxed{7}$-tuple

$(Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$ where

- Q is a finite set of control states
- Σ is a finite set of input symbols without the blank symbol $\sqcup$
- Γ is a finite set of tape symbols with $\sqcup \in \Gamma$ and $\Sigma \subset \Gamma$
- δ is a state-transition function $Q' \times \Gamma \to Q \times \Gamma \times \{L, R\}$

  where $Q' = Q - \{q_{accept}, q_{reject}\}$
- $q_0 \in Q$ is the start state
- $q_{accept} \in Q$ is the accept state
- $q_{reject} \in Q$ is the reject state, with $q_{accept} \neq q_{reject}$.

  $q_0 = q_{accept}$ or $q_0 = q_{reject}$ is possible.

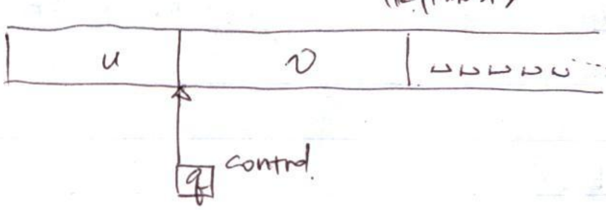
we denote the empty string by ϵ

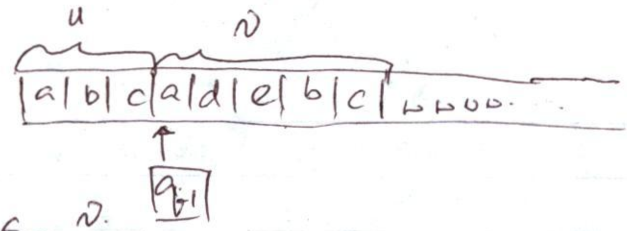
Configurations and the one-step transition relation $\vdash$

A configuration is denoted by $u\,q\,v$ where

- $uv \in \Gamma^*$ is the current tape string. All tape ~~symbols~~ cells after the rightmost symbol of uv contain $\sqcup$
- q is the current control state
- The tape head is at 1st symbol of v. (leftmost)

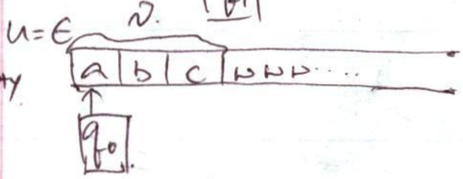
[illegible]



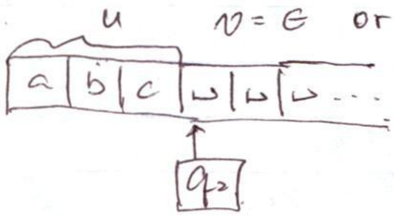


$abcq_1adeb'c.$

u is empty



$\epsilon q_0abc = q_0abc$

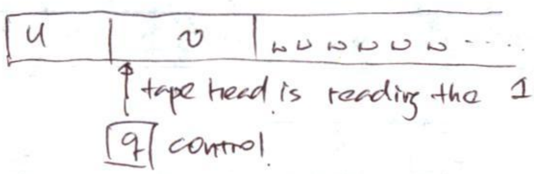


$abcq_2\epsilon = abcq_2$

$\boxed{8/30}$ ~~DTM~~. Configuration.

uqv.

tape



tape head is reading the 1st symbol of v

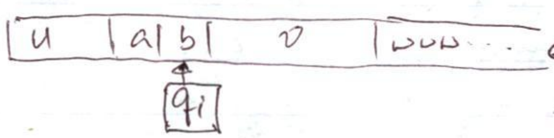
control

A configuration is called an "instantaneous description" in some books. Intuitively, it is a "state" ("snapshot") of the whole DTM.

One-step Transition relation $\vdash$ over configurations, determined by the transition function δ. Suppose $uv \in \Gamma^*$, $a, b \in \Gamma$, $q_i \neq q_{accept}$, $q_i \neq q_{reject}$.

case

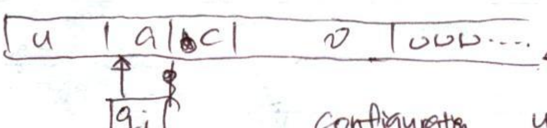
①



$\boxed{uaq_ibv \vdash uq_jacv}$

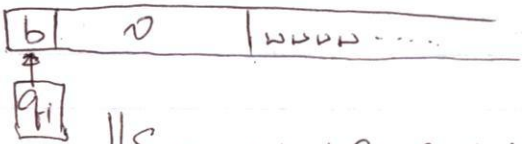
$\Downarrow$ $\delta(q_i, b) = (q_j, c, L)$ move left

所以 pointer 左移



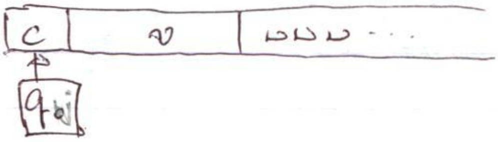
configuration uaq_ibv transits (yields) to uq_jacv in one step

②

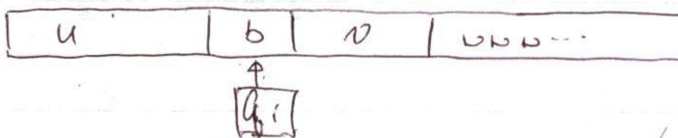


$\Downarrow \delta(q_i, b) = (q_j, c, L)$ ↳ move left

$\boxed{q_i b v \vdash q_j c v}$

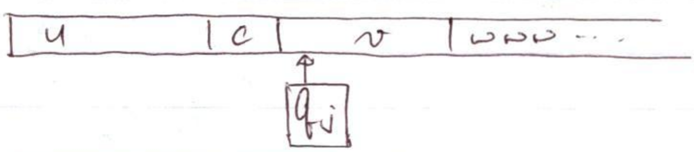


③



↙ moving to Right

$\Downarrow \delta(q_i, b) = (q_j, c, R)$



$\boxed{u q_i b v \vdash u c q_j v}$

✓ An accepting configuration is one with control state q_{accept}.

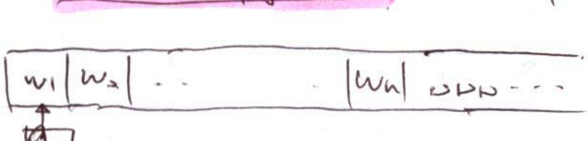
A rejecting configuration - - - - - - - q_{reject}.

A halting - - - - an accepting OR a rejecting configuration

$Q' = Q - \{q_{accept}, q_{reject}\}$ $\delta : Q' \times T \to Q \times P \times \{L, R\}$

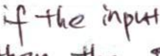
δ is undefined for q_{accept} q_{reject}. The DTM halts whenever an accepting or rejecting configuration reached.

The start configuration on input string $w = w_1 w_2 \cdots w_n \in \Sigma^*$ is $q_0 w$



$|w| = n$ is the input size

if the input string is empty string ϵ then the start configuration is $q_0 \sqcup$



input size is $0 = |\epsilon|$

A sequence of configuration $C_1, C_2, C_3, \ldots, C_k, \ldots$ (possibly infinite) is a transition sequence on an input string $w \in \Sigma^*$ if, decided by the δ function,

- C_1 is the start configuration $q_0 w$; and.
- $C_i \vdash C_{i+1}$, for all $i \geq 1$

If a transition sequence is finite halting in accepting or rejecting configuration $C_1 \vdash C_2 \vdash \dots \vdash C_k$, C_k is an accepting or rejecting configuration, the value of k is the time to accept / reject the input w

A. DTM, M, accepts (rejects) an input string w if there exists a finite transition sequence $C_1, C_2, \dots, C_k$ on w where C_k is an accepting configuration (rejecting, respectively)

The language recognized by a DTM, M, is
$L(M) = \{ w \in \Sigma^* \mid M \text{ accepts } w \}$ M may reject w or run infinitly
$L(M)$ is a recursively enumerable language. (Also called a semi-decidable or partially decidable language)

Define a decider DTM,

A DTM, M is a decider if it $\boxed{\text{halts}}$ on $\boxed{\text{all}}$ input strings $w \in \Sigma^*$. In this case, $L(M)$ is a recursive language (Also called a decidable language)

[A famous example of semi-decidable but not decidable language.]
halting problem — actually true for any programming language
Input: String x encoding $\langle M, w \rangle$ where M is a DTM and w is an input string to M.
That this is not decidable is proved by diagonalized technique
Output: accept if M halts on w
reject if M does not halt on w
Semi-decidable. Just Run M on w and see what happens
let the be M'
on a meta-DTM (interpreter DTM) also called a universal DTM
accept if it halts
otherwise, it runs infinitely
$L(M') = \{ x \text{ encoding } \langle M, w \rangle \mid M' \text{ accepts } x \}$
$\{ \text{ - - - - - - } M \text{ halts on } w \}$ 根據以上規則

Eg. Basic, Java, C++, ...
string x encoding $\langle M, w \rangle$ where M is a Basic program, w is any valid input to M

DTM to decide power2.html

unary ~~number~~ notation

represent a ~~normal~~ natural number n by a string of n symbols (e.g "0")

$32 \to 16 \to 8 \to 4 \to 2 - 1$

$20 \to 10 \to 5$

eg 3.7 in the book.

example 16

0000 0000 0000 0000 ⊔⊔⊔ ...

⊔000 0000 0000 0000 ⊔⊔⊔ ...

⊔x0x0x0x 0x0x 0x0x ... # of crossed off 0's = 8.

⊔xx0xxx 0xxx 0xxx ... 4

⊔xx xxxx 0xxx xxxx ... 2

⊔xx xxxx xxxx xxxx ... 1

$\boxed{9/1}$

$20 \to 10 \to 5$. x

$40 \to 20 \to 10 \to 5$ x

$6 \to 3$. x

$8 \to 4 \to 2 \to 1$. ✓

$12 \to 6 \to 3$ x

$36 \to 18 \to 9$ x

$\delta(q_i, a) = (q_j, b, L/R)$ $a \neq b$

$$q_i \xrightarrow[a \to b, R]{a \to b, L \text{ or } R} q_j$$

if $a = b$

$\delta(q_i, a) = (q_j, a, L/R)$

$$q_i \xrightarrow[a \to R]{a \to L \text{ or } R} q_j$$

对照网上eg.

网上有reject的了.

input size = 4

① $\underset{q_1}{\underline{0}}$ 000 ⊔ ⊢

② ⊔ $\underset{q_2}{\underline{0}}$ 00 ⊔ ⊢ k=4

③ ⊔ x $\underset{q_3}{\underline{0}}$ 0 ⊔ ⊢

④ ⊔ x 0 $\underset{q_4}{\underline{0}}$ ⊔ ⊢

⑤ ⊔ x 0 x $\underset{q_3}{\underline{⊔}}$ ⊢

⑥ ⊔ x 0 $\underset{q_5}{\underline{x}}$ ⊔ ⊢

⑦ ⊔ x $\underset{q_5}{\underline{0}}$ x ⊔ ⊢

⑧ ⊔ $\underset{q_5}{\underline{x}}$ 0 x ⊔ ⊢

⑨ $\underset{q_5}{\underline{⊔}}$ x 0 x ⊔ ⊢

⑩ ⊔ $\underset{q_2}{\underline{x}}$ 0 x ⊔ ⊢

⑪ ⊔ x $\underset{q_2}{\underline{0}}$ x ⊔ ⊢

⑫ ⊔ x x $\underset{q_3}{\underline{x}}$ ⊔ ⊢

⑬ ⊔ x x x $\underset{q_3}{\underline{⊔}}$ ⊢

The time to decide 0000 is $\boxed{21}$

last one not counted.

只读⊢

最后一个是transition.

crossed off 0's

k = 2

⑭ $\sqcup\ x\ x\ \underset{q_5}{\underline{x}}\ \sqcup\ \vdash$

⑮ $\sqcup\ x\ \underset{q_5}{\underline{x}}\ x\ \sqcup\ \vdash$

⑯ $\sqcup\ \underset{q_5}{\underline{x}}\ x\ x\ \sqcup\ \vdash$

⑰ $\underset{q_5}{\underline{\sqcup}}\ x\ x\ x\ \sqcup\ \vdash$

⑱ $\sqcup\ \underset{q_2}{\underline{x}}\ x\ x\ \sqcup\ \vdash$

⑲ $\sqcup\ x\ \underset{q_2}{\underline{x}}\ x\ \sqcup\ \vdash$

⑳ $\sqcup\ x\ x\ \underset{q_2}{\underline{x}}\ \sqcup\ \vdash$

㉑ $\sqcup\ x\ x\ x\ \underset{q_2}{\underline{\sqcup}}\ \vdash$

$\sqcup\ x\ x\ x\ \sqcup\ \underset{q_{accept}}{\underline{\sqcup}}$

刚刚那例子是 000000 是reject的.

① $\underset{q_1}{\underline{0}}\ 0\ 0\ 0\ 0\ 0\ \sqcup\ \vdash$

② $\sqcup\ \underset{q_2}{\underline{0}}\ 0\ 0\ 0\ 0\ \sqcup\ \vdash$

③ $\sqcup\ x\ \underset{q_3}{\underline{0}}\ 0\ 0\ 0\ \sqcup\ \vdash$

④ $\sqcup\ x\ 0\ \underset{q_4}{\underline{0}}\ 0\ 0\ \sqcup\ \vdash$

⑤ $\sqcup\ x\ 0\ x\ \underset{q_3}{\underline{0}}\ 0\ \sqcup\ \vdash$

⑥ $\sqcup\ x\ 0\ x\ 0\ \underset{q_4}{\underline{0}}\ \sqcup\ \vdash$

⑦ $\sqcup\ x\ 0\ x\ 0\ x\ \underset{q_3}{\underline{\sqcup}}\ \vdash$

⑧ $\sqcup\ x\ 0\ x\ 0\ \underset{q_5}{\underline{x}}\ \sqcup\ \vdash$

⑨ $\sqcup\ x\ 0\ x\ \underset{q_5}{\underline{0}}\ x\ \sqcup\ \vdash$

⑩ $\sqcup\ x\ 0\ \underset{q_5}{\underline{x}}\ 0\ x\ \sqcup\ \vdash$

⑪ $\sqcup\ x\ \underset{q_5}{\underline{0}}\ x\ 0\ x\ \sqcup\ \vdash$

⑫ $\sqcup\ \underset{q_5}{\underline{x}}\ 0\ x\ 0\ x\ \sqcup\ \vdash$

⑬ $\underset{q_5}{\underline{\sqcup}}\ x\ 0\ x\ 0\ x\ \sqcup\ \vdash$

⑭ $\sqcup\ \underset{q_2}{\underline{x}}\ 0\ x\ 0\ x\ \sqcup\ \vdash$

⑮ $\sqcup\ x\ \underset{q_2}{\underline{0}}\ x\ 0\ x\ \sqcup\ \vdash$

⑯ $\sqcup\ x\ x\ \underset{q_3}{\underline{x}}\ 0\ x\ \sqcup\ \vdash$

⑰ $\sqcup\ x\ x\ x\ \underset{q_3}{\underline{0}}\ x\ \sqcup\ \vdash$

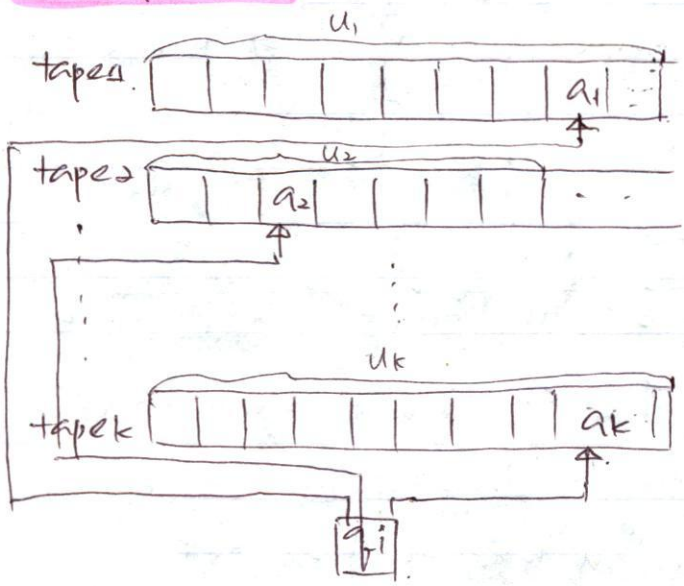
⑱ $\sqcup\ x\ x\ x\ 0\ \underset{q_4}{\underline{x}}\ x\ \sqcup\ \vdash$

⑲ $\sqcup\ x\ x\ x\ 0\ x\ \underset{q_4}{\underline{x}}\ \sqcup\ \vdash$

⑳ $\sqcup\ x\ x\ x\ 0\ x\ x\ \underset{q_4}{\underline{\sqcup}}\ \vdash$

$\sqcup\ x\ x\ x\ 0\ x\ x\ \underline{\sqcup}\ q_{reject}$

Multitap DTM



has (k) tapes for some constant k.

has only 1 control

The only difference is δ : $Q' = Q - \{q_{accept}, q_{reject}\}$

$\delta : Q' \times \Gamma^k \to Q \times \Gamma^k \times \{L, R, \boxed{S}\}$

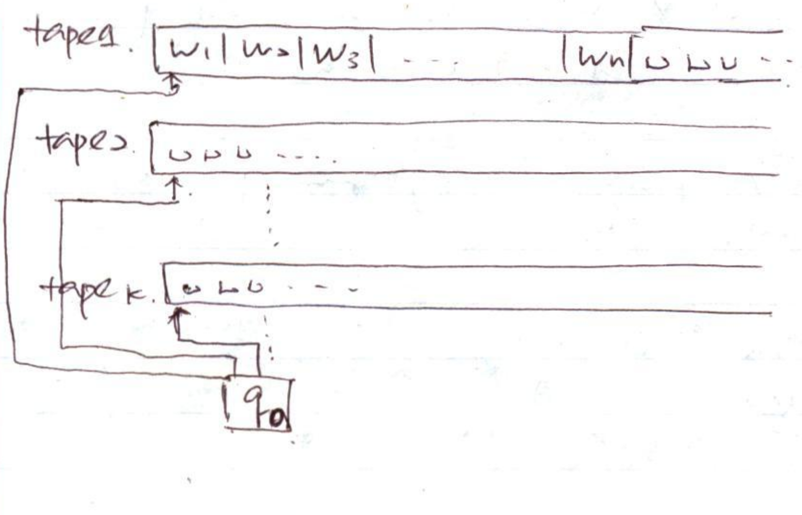
stay put in the same position

$\delta(q_i, a_1, \cdots, a_k) = (q_j, b_1 \cdots b_k, d_1, \cdots d_k)$

$d_i \in \{L, R, S\}$

Abstraction of computers with k independent memories.

Initial configuration on input $w = w_1 \cdots w_n$



Simulation of k-tape DTM by 1-tape DTM

Let M = given k-tape DTM.

M' = simulating 1-tape DTM

Let $T = \{s_1, \dots, s_n\}$ be the tape alphabete of M.

T' of M' extands T as follows:

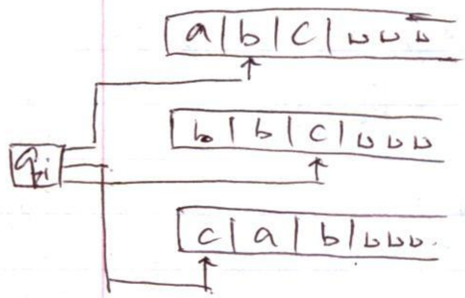
$T' = \{s_1, \dots, s_n, \dot{s}_1, \dots, \dot{s}_n, \#\}$ # is any symbol not in T.

Let u_i be the current tape string on tape i, where the right most symbol of u_i is the rightmost nonblank symbol

M' simulates the u_i by:

The dotted symbol simulate the positions of k tape heads of M.

q_i



simulate by

| a | $\dot{b}$ | c | # | b | b | $\dot{c}$ | # | $\dot{c}$ | a | b | # | ␣ ␣ ␣

$\delta(q_i, b, c, c) = (q_j, a, b, c, R, L, R)$

| a | a | $\dot{c}$ | # | b | $\dot{b}$ | b | # | c | $\dot{a}$ | b | # | ␣␣

Whenever M moves the head onto #,
M' must move the string to the left of it one position to right

n = length of input string

Every scan.

Takes at most $O(f(n))$ steps

To simulate one step of M, M' performs two scans from left to right.

1st scan: Identify the k dotted tape symbols
move head back

2nd scan: sequentially update tape symbols. If necessary, do
$\delta(q_i, a_1, \dots a_k) = (b_1, \dots b_k \dots)$

Page 254.

Th 7.8. If M's runtime is $O(f(n))$,
M''s runtime is $O((f(n))^2)$.

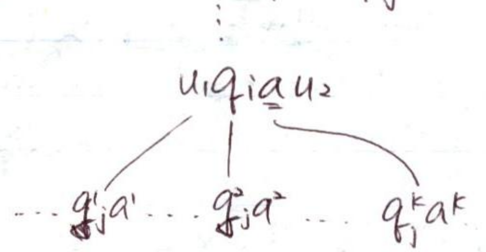
2nd scan: means in each of.
1st scan. uses $O(f(n))$ steps.

Nondeterminstic, 1-tape ~~TMs~~ NTMs.

Incorporates a nondeterminstic choice in state transitions

$\delta(q_i, a) = \{(q_j^1, a^1), (q_j^2, a^2), \ldots, (q_j^k, a^k)\}, \ k \geq 0$

computation tree of configurations.



9/6.

Nondeterminstic, one-tape Turing machines (NTMs)

An NTM is a 6 tuple $(Q, \Sigma, \Gamma, \delta, q_0, q_{accept})$ q_{reject} is not used

$Q, \Sigma, \Gamma, q_0, q_{accept}$ are the same as for DTMs

$\delta: Q' \times \Gamma \to \mathcal{P}(Q \times \Gamma \times \{L, R\})$ where $Q' = Q - \{q_{accept}\}$

The power set operator.

$\mathcal{P}(X) = \{S \mid S \subseteq X\}$

$\delta(q_i, a) = \{(q_{j1}, b_1, d_1), \cdots, (q_{jk}, b_k, d_k)\}, \quad k \geq 0$

$\delta(q_i, a) = \{\}$ if $k = 0$.

Intuitively, one of the k transition is chosen "nondeterministically".

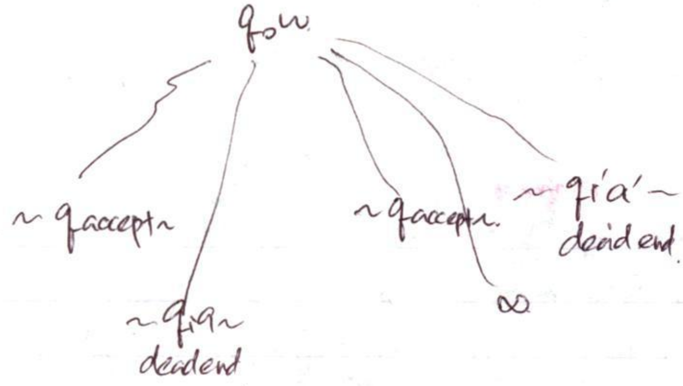
The definitions of configurations and $\vdash$ are same as for DTM

A computation tree on an input string $w \in \Sigma^*$ is a tree of configurations

where
- the root is the start configuration $q_0 w$
- Each configuration C in the tree has children configurations D_i, $1 \leq i \leq k$, iff $C \vdash D_i$ for all i.

~~NTM to dead~~

A computation tree has 3 kinds of branches:
- An accepting branch: a finite branch halting in an accepting configuration
- A dead-end branch: - - - - - - - because of non existence of transitions ($\delta(q_i, a) = \{\}$)
- An infinite branch



(1) If the computation tree on w has at $\boxed{\text{least one}}$ accepting branch, w is $\boxed{\text{accepted}}$. does not matter how many branches it has

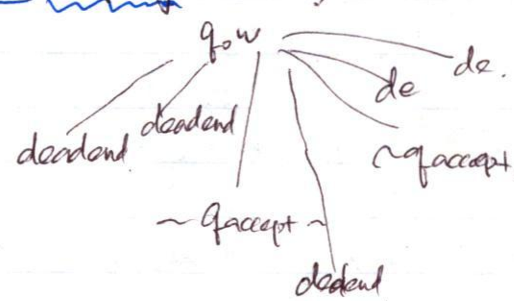
(2) If $\boxed{\text{all}}$ branches of the computation tree on w are dead-end branches, w is $\boxed{\text{rejected}}$

(3) Otherwise, the tree has $\boxed{\text{no}}$ accepting branches and has $\boxed{\text{at least one}}$ infinite branch. In this case, w is $\boxed{\text{neither accepted nor rejected}}$

Define NTM decider

A decider NTM is one whose computation trees have $\boxed{\text{no}}$ infinite branches for $\boxed{\text{all}}$ input strings.

So, for any decider NTM, any computation tree is $\boxed{\text{finite}}$ and $\boxed{\text{has at least one}}$ accepting branch, and w is $\boxed{\text{accepted}}$ OR the computation tree is $\boxed{\text{finite}}$ and $\boxed{\text{all branches}}$ are dead-end. w is $\boxed{\text{rejected}}$

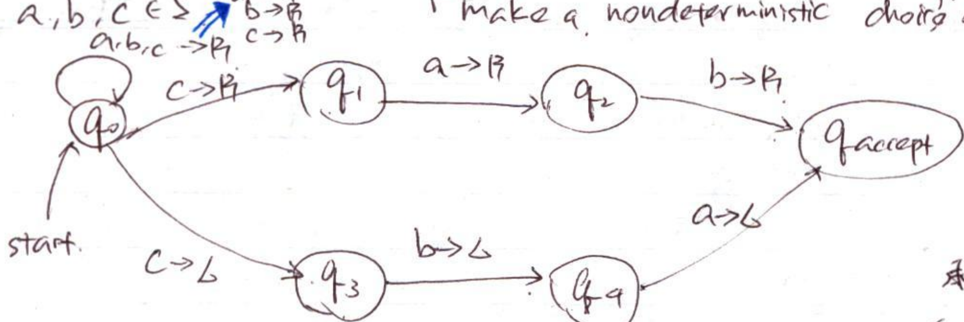


An NTM to decide $\{w\underline{c}abx \mid w,x \in \Sigma^*\} \cup \{wab\underline{c}x \mid w,x \in \Sigma^*\}$,

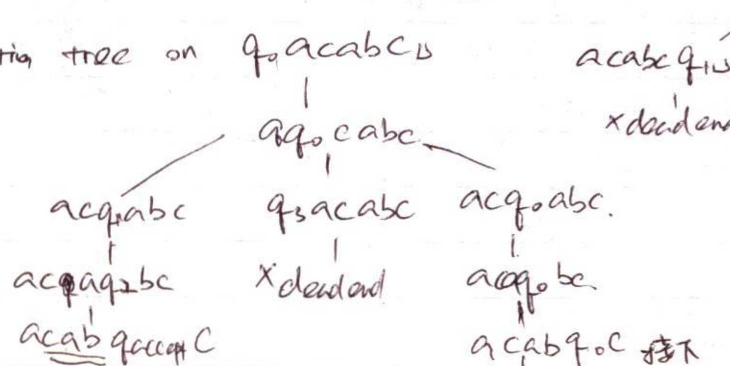
$a, b, c \in \Sigma$ ↑ make a nondeterministic choice on $\underline{c}$

$T = \{a, b, c, \sqcup\}$

w, x are any strings of a, b, c



Computation tree on $q_0 acabc\sqcup$



The input acabc is accepted