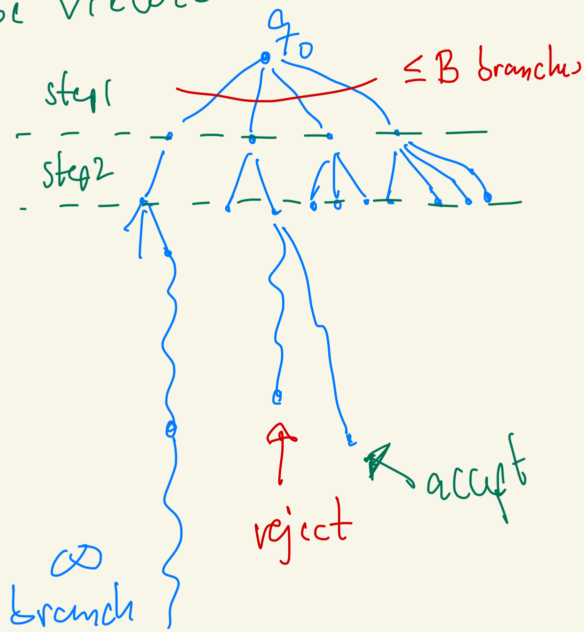
Variants of Turing machines part 2

Non-deterministic TM (NDTM)

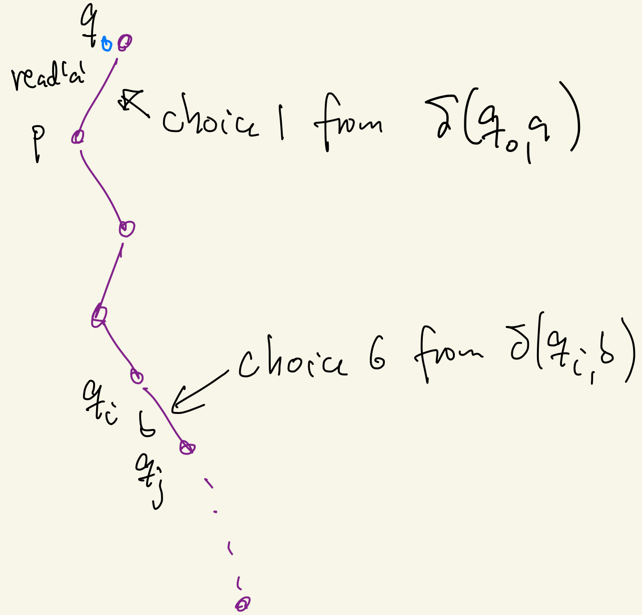
$$\delta : Q \times \Gamma \to \mathcal{P}(Q \times \Gamma \times \{L, R, S\})$$

Important: each $\delta(q, a)$ contains a finite number B of possible transitions where $B \le |Q| \cdot |\Gamma| \cdot 3$

The computation of a NDTM M on a string w can be viewed as a tree



each path p from the root corresponds to a specific choice for each of the $|p|$ states



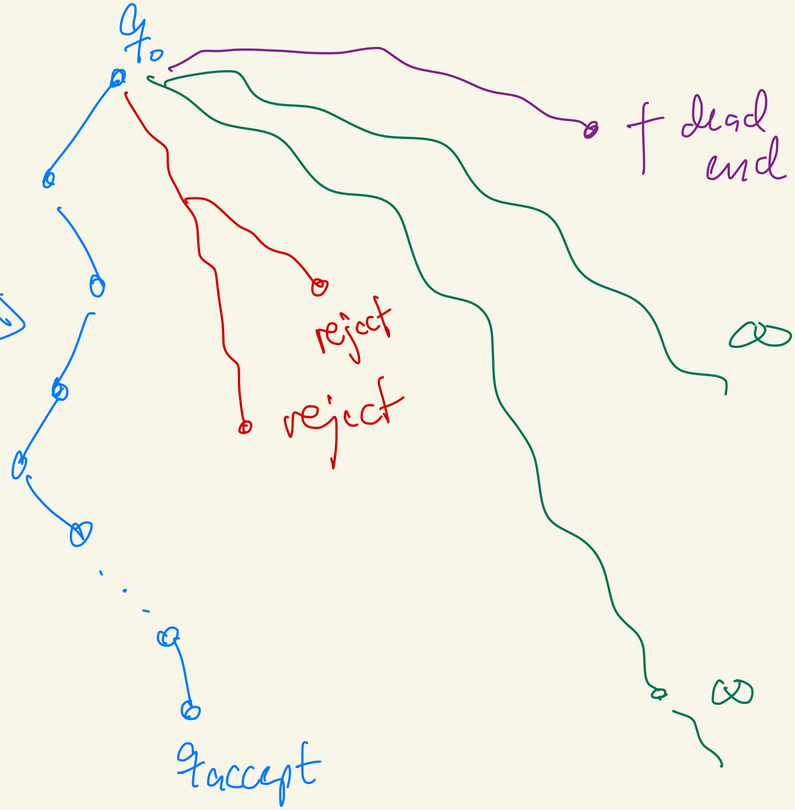
Definition The NDTM M recognizes L if $L = \{ w \mid \exists$ some computation for M on w leading to M's accept state $\}$

Definition The NDTM M decides L if $\forall w \in \Sigma^*$:

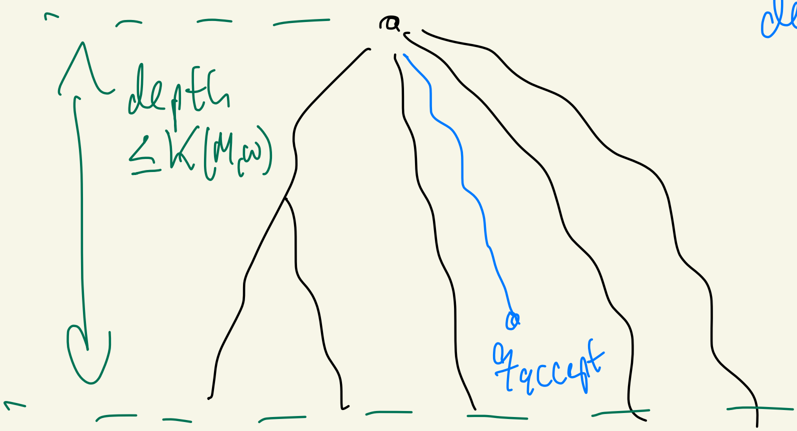
1) $\exists\, K = K(M, w)$ s.t. M never takes more than K steps on w (all paths from root have length $\leq K$)

2) $w \in L \iff M$ has at least one accepting calculation on w

M accepts w because of blue path



decider



Theorem 3.16

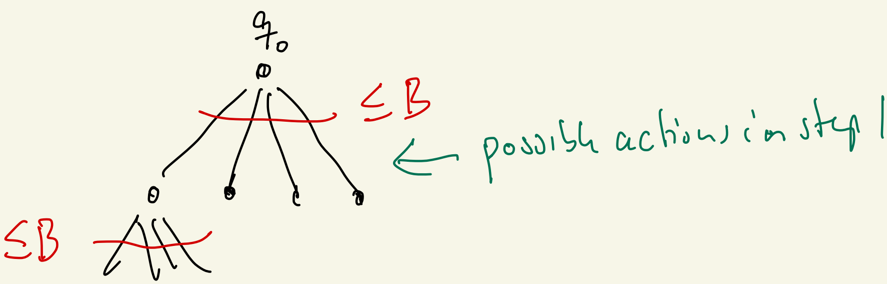
For every NDTM M there exists a deterministic TM D such that

a) $L(D) = L(M)$ and b) D decides $L \Leftrightarrow M$ decides L

proof idea: Simulate M's possible calculations on w in a Breath-First manner

(DFS strategy does not work, due to possible ∞-paths in computation tree)

Recall that $\delta_M(q,a)$ has at most B values (of the form (p,b,γ))



Let $B = \max\{|\delta(q,a)| \mid q \in Q, a \in \Gamma\}$

So B is the maximum # of different transitions M has from any state on any symbol

Assumption: Every $\delta(q,a)$ has exactly B transitions or none at all

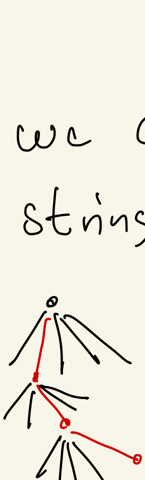
Can achieve this by adding copies of one of the transitions

$B = 4$ $\qquad \delta(q,a) = \{(q',b,R),(p,c,L)\}$

$$\downarrow$$

$$\delta(q,a) = \{(q',b,R),(p,c,L),(p,c,L),(p,c,L)\}$$

Now we can describe M's calculations via strings of numbers in base B+1 (no zero's)



235

For a given string $g_1 g_2 \cdots g_k$ $\quad g_i \in \{1, 2, \ldots, B\}$

We can simulate the corresponding computation of M by taking the g_ith choice in step i

Conclusion: If we can implement the simulation above via a DTM D, then D satisfies a) and b) in Theorem 3.16

In particular $L(D) = L(M)$

How to generate numbers like 237 and in which order?

use lexicographic ordering on strings of the same length $\quad 237 < 239$

and $\quad 237 < 1117$ as left part is shorter

Simple way to keep track of numbers and generate next

#00#000#00000# $\qquad$ B=6

2 3 5

next is

#00#000#000000# $\quad$ 236

The Turing machine D:

3 tapes

$\# w_1 w_2 \cdots w_n$

$\#$ M's calculation on w

$\#$ numbers base $B+1$

eg

$\# w_1 w_2 \cdots w_n$

$\#$ simulate M for 3 steps according to

$\#$ 235

<u>NB:</u> If the current configuration that M would be in has no transition on for the state M is in an the current symbol it reads, then stop simulating according to the current string on tape 3

Also: if the simulation leads to M accepting then D will accept

Formally

1. $\triangleright \underline{w_1} w_2 \cdots w_n$

   $\triangleright \underline{\sqcup}$

   $\triangleright \# 0 \#$

2. Copy tape 1 to tape 2

3. Simulate M on tape 2 using the numbers on tape 3 to select next transition until either
   - dead end = no transition on (q, a)
   - M accepts => D accepts (and stops)
   - M rejects
   - no more digits on tape 3

4. replace $d_1 d_2 \ldots d_r$ on tape 3 by next in lexicographic order

5. Go to 2

How long does this simulation take?

Assume M runs for r steps, then D simulates at most

$$B + B^2 + \ldots + B^r \leq B^{r+1} \text{ steps of } M$$

Exponential time!

Open question: can we make a det TM D satisfying a) and b) in Thm 3.16 such that the running time of D is

$O(r^c)$ for some constant c when

M runs in time $O(r)$

P = NP question later in the course

Corollary of Thm 3.16

L is Turing recognizable/decidable

$\Updownarrow$

L is recognized/decided by some NDTM