Nondeterministic Turing machines

Enumerators

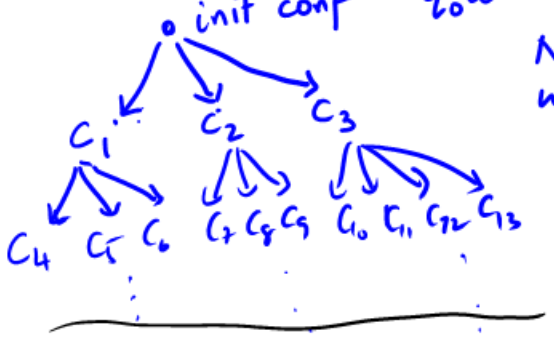
Dove-tailing

M is a NTM

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

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

A (det) TM $\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$



NTM M accepts w if there is some path in this tree which ends in q_{acc}

A NTM M acc w if there exists a sequence of configurations

$C_0 \vdash C_1 \vdash C_2 \cdots \vdash C_n$

where $C_0 = q_0 w$

$C_i \vdash C_{i+1}$ according to δ

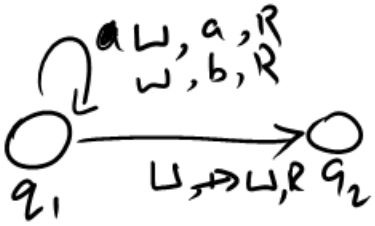
$C_n = u q_{acc} v$ for some $u, v \in \Gamma^*$.

A NTM is a decider if all paths in the runs halt.

$L = \{ \langle A, B \rangle \mid A \text{ and } B \text{ are DFAs and } L(A) \cap L(B) \neq \emptyset \}$

Algorithm (non-det.)

1. generate an arbitrary word $w \in \Sigma^*$ $|w| \leq |Q_A| \cdot |Q_B|$
2. Check if A acc w and B acc w.
3. If it does then halt & acc
   else halt & rej.


a ␣, a, R
␣, b, R
␣, ␣, R
q_1
q_2

$Var = \{x_1, x_2, \ldots\}$

$$\alpha, \beta ::= \top \mid \bot \mid x_i \mid (\alpha \wedge \beta) \mid \neg(\alpha) \mid (\alpha \vee \beta)$$

$((x_1 \wedge x_2) \vee x_3)$

$((x_1 \wedge x_2) \wedge x_3) \vee x_4$.

NTM. Nondet. generate a valuation for the variables.
Evaluate the formula.
If formula evaluates to True, accept
else reject.

NTMs are only as powerful as DTMs

NTM

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

$|Q| \cdot |\Gamma| \cdot 2$

DTM

$q_0 \cdots$ C_1, C_2

$C_3, C_4, C_5, C_6, C_7, C_8, C_9$

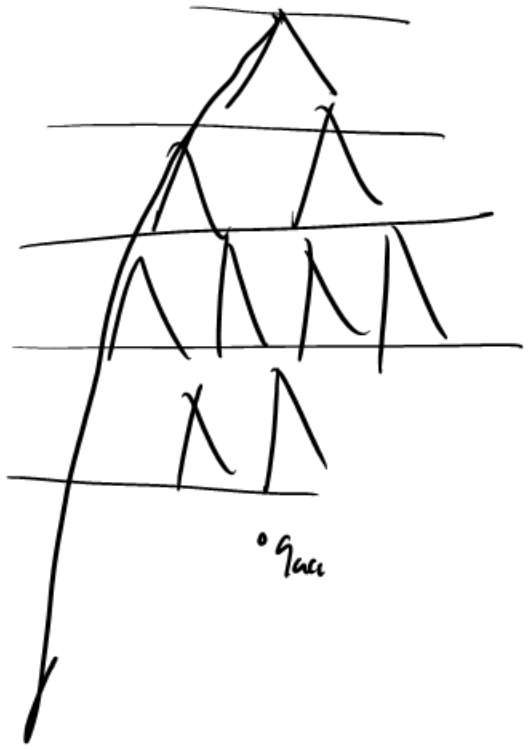
DTM will halt and accept if any config is an accepting config.

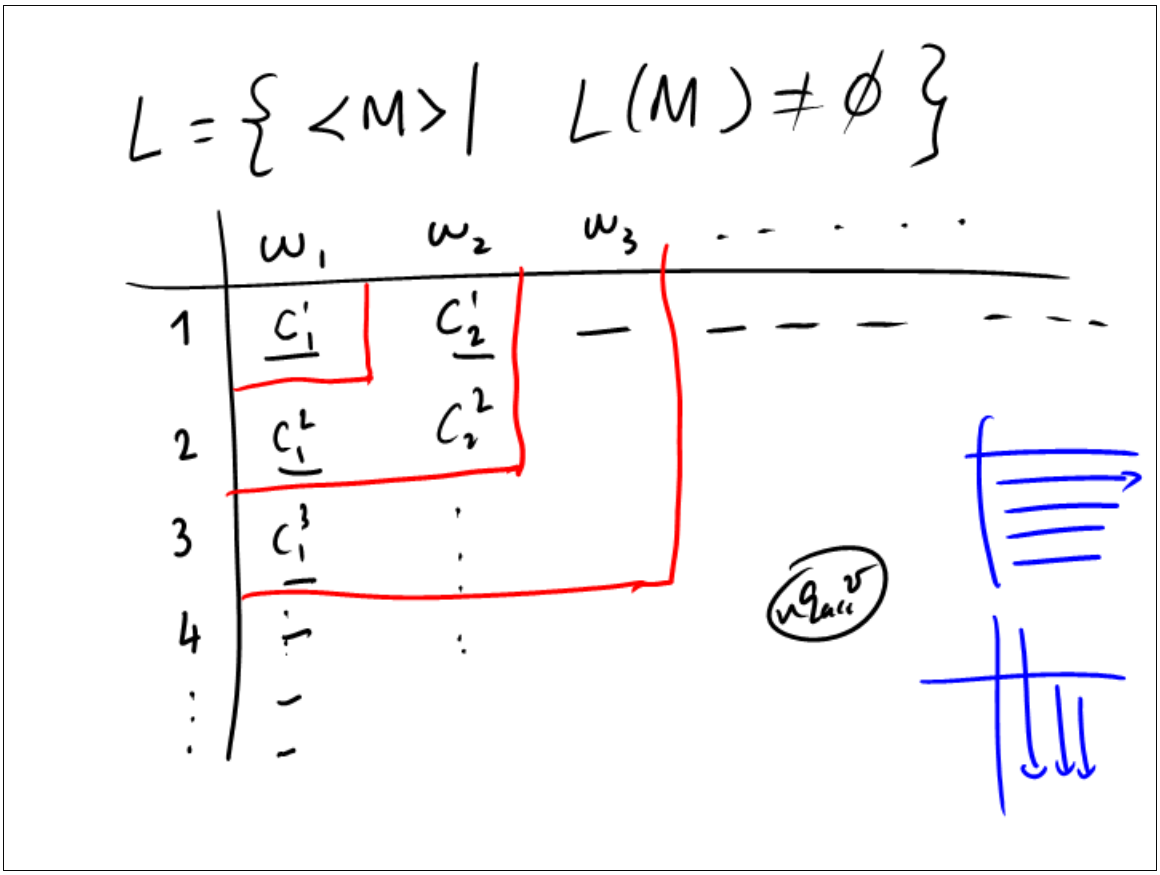
DTM will reject if all config reject.

Thm. The class of all languages recognized by NTMs is precisely the class of TM-recognizable languages.

The class of all languages decided by NTMs is precisely the class of TM-decidable languages.

Dovetailing

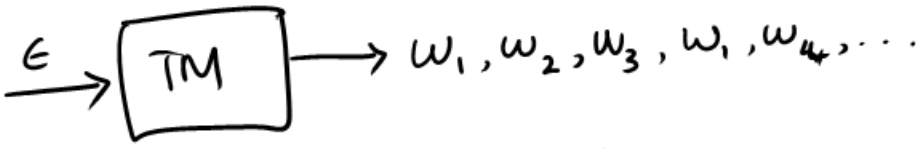



L = { <M> | L(M) ≠ ∅ }
w1
w2
w3
1
2
3
4
c11
c12
c21
c22
c13

Recognizer for $L = \{\langle M \rangle \mid L(M) \neq \emptyset\}$

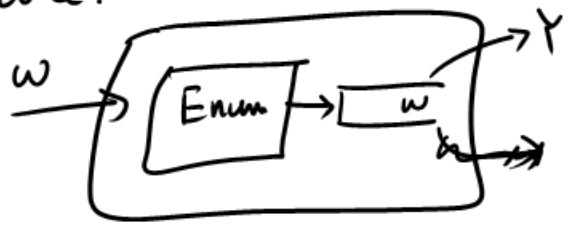
```
1.  i := 1
2.  while (true) {
3.      for (j = 1 .. i) {
            Simulate M on w_j for i-steps.
            and halt & accept if it accepts
        }
4       i++;
5.  }
```

Enumerators.



A languages L is enumerable
iff L is TM-recognizable.

$\Rightarrow$ L is enumerable.
Recognizer for L



(⇐) TM recognizer for L

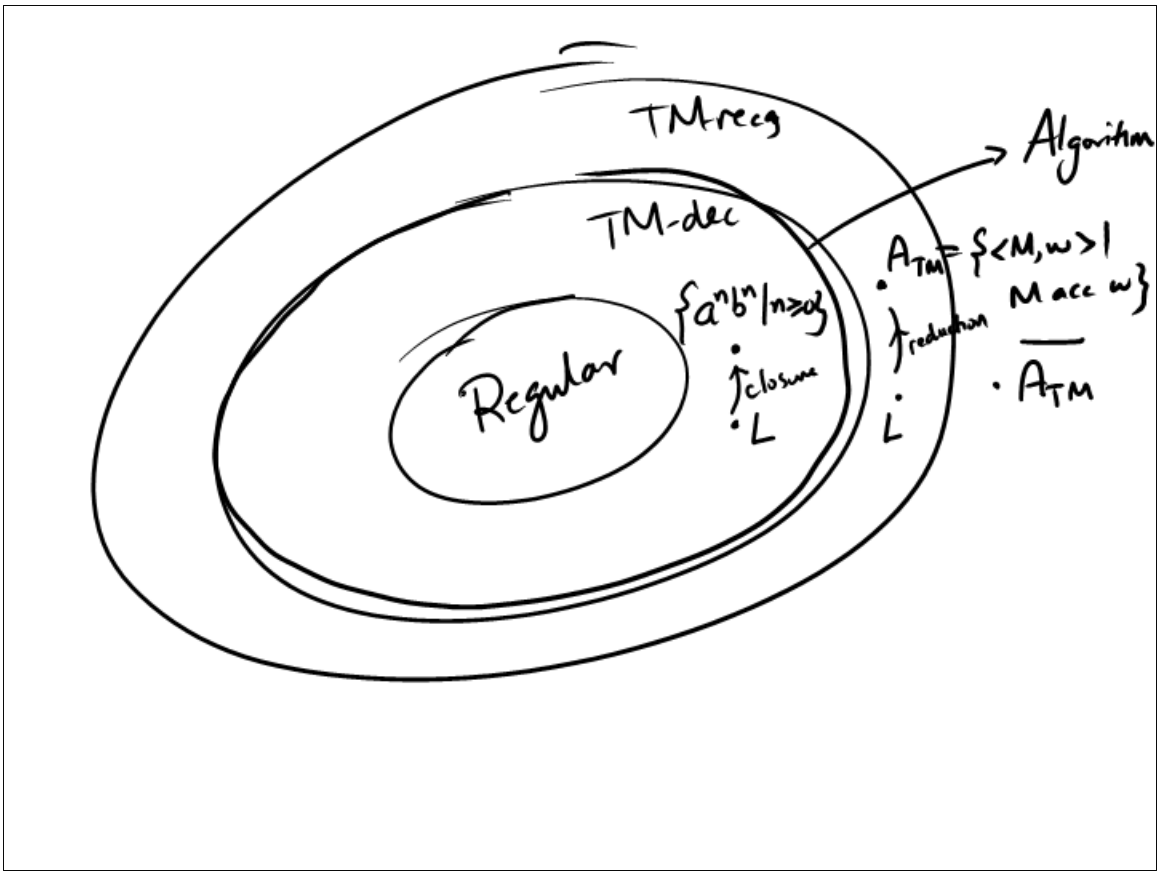
for (i := 1 .. ∞) {

Generate $w_1, \ldots w_i$

Simulate the recognizer on each $w_j \in \{w_1 \ldots w_j\}$ for i-steps

If w_j is accepted, output w_j.

}


TM recg
TM-dec
Regular
$\{a^n b^n / n \geq 0\}$
closure
L
Algorithm
$A_{TM} = \{\langle M, w \rangle |$ M acc w$\}$
reduction
L
$\overline{A_{TM}}$