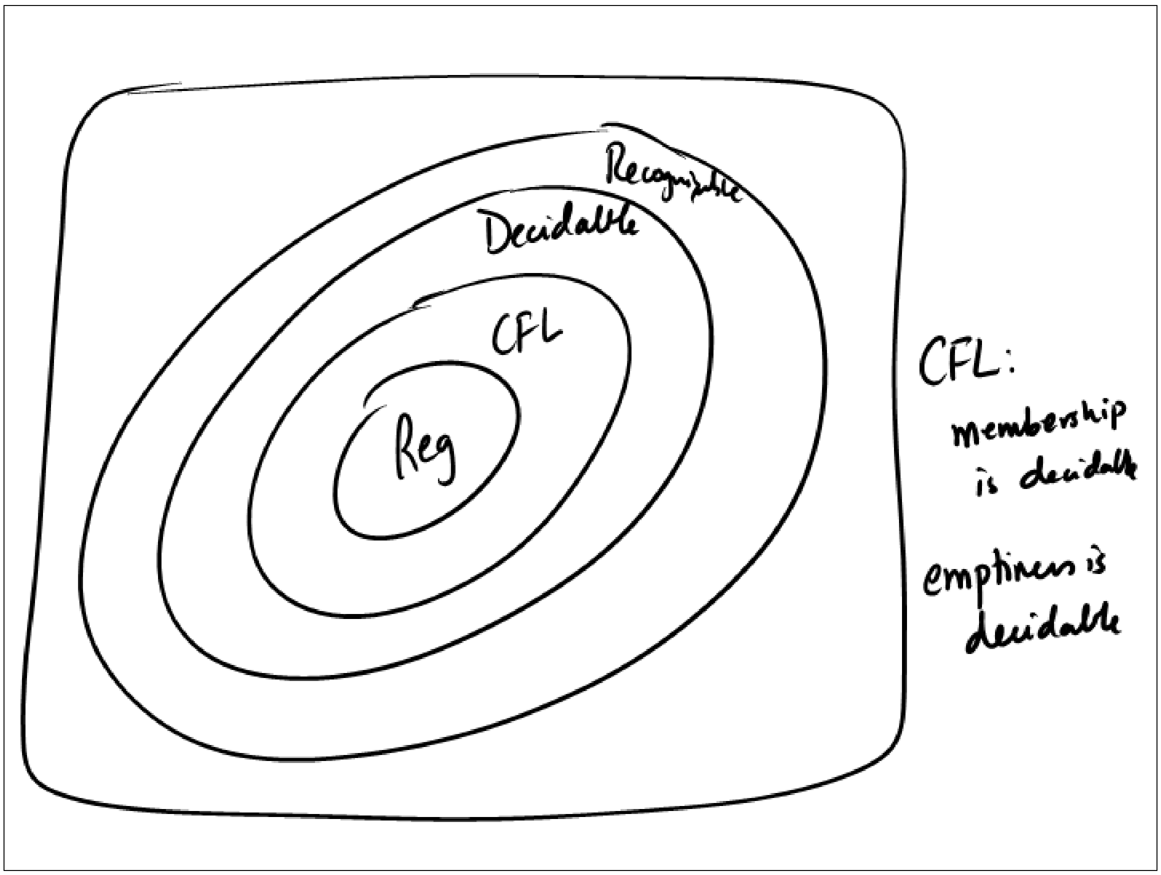
Undecidable problems on Linear bounded TMs & Context-free languages.


Recognizable
Decidable
CFL
Reg
CFL:
membership is decidable
emptiness is decidable

	$\subseteq$	$=$	$=\Sigma^*$
Reg	Dec	Dec	Dec
CFL	Undec	Undec	Undec

Linear-bounded Turing machines

A TM M is a linear-bdd TM if M on any input w takes only $O(|w|)$ space to decide/halt.

no. of cells used by TM.

$$f(n) = O(g(n))$$

if $\exists n_0 . \ \forall n \geq n_0 , \quad f(n) \leq c\, g(n) + d$.
$\exists c, d$

~~for some c, d~~

$L = \{ a^n b^n c^n \mid n \in \mathbb{N} \}$.

L is accepted by a LBTM.

CSL Context-sensitive lang.

$$a B c \longrightarrow D E$$

$A_{LBTM} = \{ \langle M, w \rangle \mid M$ is a TM that accepts w in $|w|$ space$\}$

Run M on w

c_0 config $\leq |w|$

c_1

c_2

$\vdots$

c_n

$c_i = c_j$

TM can be in a finite no. of config.

$$k^n \cdot \underset{\substack{\uparrow \\ \text{tape head}}}{n} \cdot |Q|$$

So A_{LBTM} is decidable. $n = |w|$

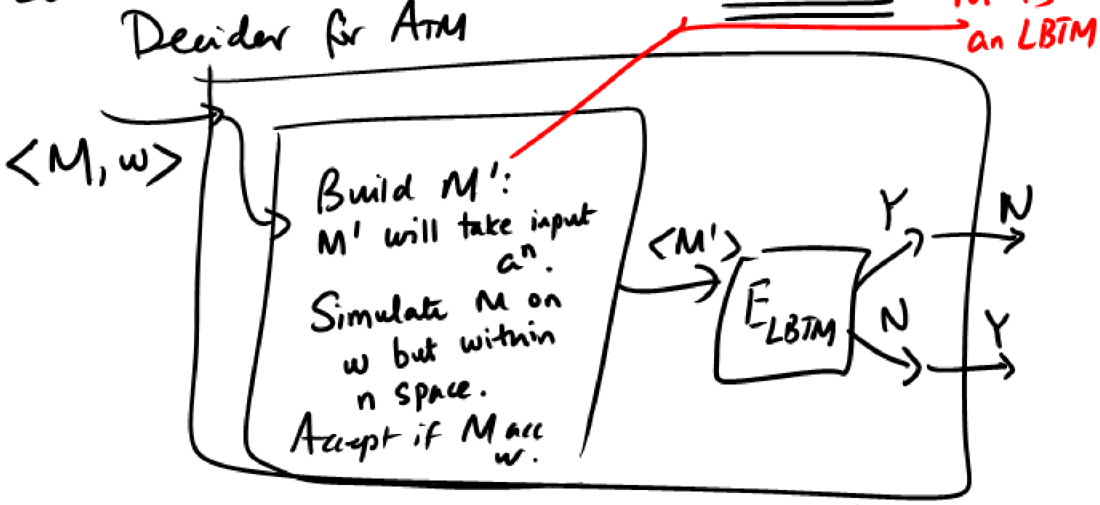
$$E_{LBTM} = \{ \langle M \rangle \mid M \text{ acc. some } w \text{ in } |w| \text{ space} \}$$

is not decidable!

$w = \epsilon \quad a \quad aa \quad b \quad bb \ldots$

$\downarrow$

Let's reduce A_{TM} to E_{LBTM}.



If M accepts w, $L(M') = \{a^i \mid M \text{ acc } w \text{ in } i \text{ space}\}$

hence $L(M') \neq \emptyset$

If M does not accept w, $L(M') = \emptyset$

$\boxed{a\ a\ \ldots\ \ldots\ a} \ \$$

$$ALL_{CFG} = \{ \langle G \rangle \mid L(G) = \Sigma^* \}$$

is undecidable!

Input
$\langle M, w \rangle$

Construct G such that
$L(G) = \Sigma^*$ iff M does not accept w.

G is going to check <u>runs</u> of the TM M on w.

A <u>configuration</u> of a TM is encoded as a finite word in $\underline{\Gamma^* Q \Gamma^*}$.

An encoding $a_1 \ldots a_i \, q \, a_{i+1} \ldots a_n$

<u>means</u> : tape content is $a_1 \ldots a_i a_{i+1} \ldots a_n \#^\infty$

TM is in state q

the tape head points to the $(i+1)^{th}$ cell.

A run of a TM (or a trace)

is a sequence of configurations separated by "\$", that is a valid sequence of moves of the TM

$$C_0 \$ C_1 \$ C_2 \ldots\ldots C_n$$

An accepting run is a run

$C_0 \$ \ldots C_n$ where

C_n is an accepting halting config (i.e. $q \in Q_{acc}$).

A TM M accepts w

iff there is an accepting run

$C_0 \$ C_1 \ldots\ldots C_n$

where $C_0 = q_0 w$

We'll build a CFG G

s.t. $\mathcal{L}(G) = \{ r \mid r$ is <u>not</u> an acc. run of M on $w\}$

So $\mathcal{L}(G) = \Sigma^*$ iff M does not acc. w.

Given M and w

construct CFG G

s.t. $L(G) = \{ r \mid r$ is not an accepting run of M on w$\}$.

First, build G_1 that accepts r if it is not a sequence of configs of M.

i.e. if $r \notin (\Gamma^* Q \Gamma^* \$)^*$.

Also, build G_2 that checks whether

a) C_0 is the initial config encoding w

b) C_n, the last config, is accepting.

If not, G_2 will accept.

So assume $r = C_0 \$ C_1 \$ \ldots\ C_n$
where $C_0 = q_0 w$ & C_n is accepting.

r does not encode an accepting run of M on w

iff

$\exists i.\quad C_i \not\vdash_M C_{i+1}$

Sub-problem

$$c \, \$ \, c'$$

Can a CFG decide if c' is not the next config of c?

No!

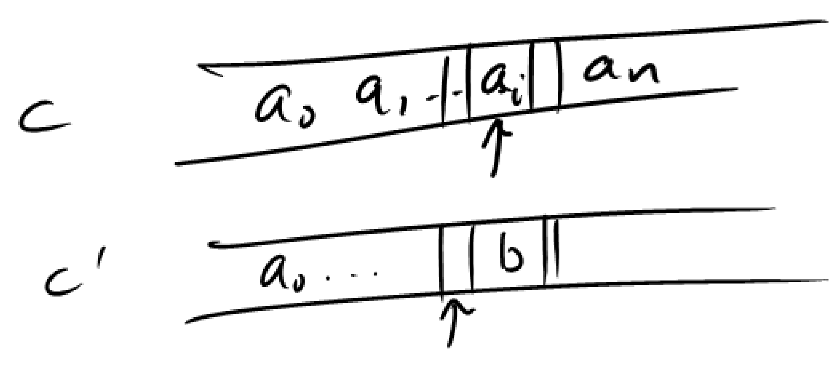
We'll encoder runs of a TM as

$$C_0 \, \$ \, C_1^r \, \$ \, C_2 \, \$ \, C_3^r \, \ldots \, C_n^{(r)}$$

Sub problem

$$C \, \$ \, C'^{r}$$

Can a CFG check if C' is **not** the next config of C.



$C: \quad \underline{a_1 \ldots a_{i-1}} \, \underline{a_i \; q \; a_{i+1}} \, \underline{a_{i+2} \ldots a_n}$

$C': \quad \underline{a_1 \ldots a_{i-1}} \, \underline{q' \, a_i \, b} \;\; \underline{a_{i+2} \ldots a_n}$

$C': \quad a_1 \ldots a_{i-1} \, \underline{a_i \; b \, q'} \; a_{i+2} \ldots a_n$

$a_1 \ldots a_i \, \underbrace{d_1 d_2 d_3} \, a_{i+1} \ldots a_n$

$\$ \; a_n \ldots a_{i+1} \, \underline{e_1 e_2 e_3} \, a_i \ldots a_1$

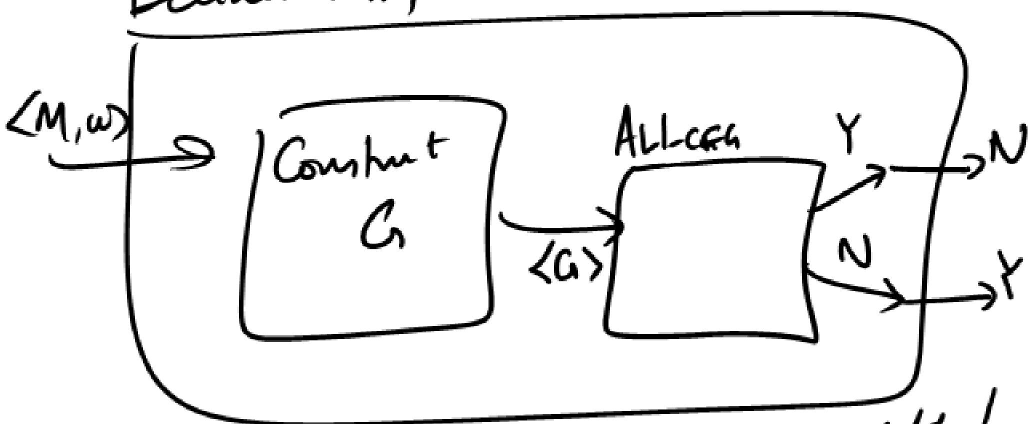
$G: \quad S \rightarrow a S a \mid T V$

$T \rightarrow$ all pairs $d_1 d_2 d_3, e_1 e_2 e_3$

$V \rightarrow a V a \mid \$$

$G = \{ r \mid r$ does <u>not</u> encode a valid accepting run of M on $w\}$.

<u>Decider A_{TM}</u>



So <u>ALL_{CFG} is undecidable!</u>

$$L_{IntCFG} = \{ \langle G_1, G_2 \rangle \mid L(G_1) \cap L(G_2) \neq \emptyset \}$$

is undecidable.

$$L_{EQCFG} = \{ \langle G_1, G_2 \rangle \mid L(G_1) = L(G_2) \}.$$

Because even setting $L(G_2) = \Sigma^*$ makes problem undecidable.

$$L = \{ \langle G, D \rangle \mid L(D) \subseteq L(G) \}$$

D is a DFA

is undecidable.

$$\underbrace{c_1 \,\$\, c_2^r}\ \$\ \underbrace{c_3 \ldots}$$

$G_1 : \quad c_1 \vdash c_2$

$\qquad\quad c_3 \vdash c_4$

$\qquad\quad \vdots$

$G_2 : \quad c_2 \vdash c_3$

$\qquad\quad c_4 \vdash c_4$

$\qquad\quad \vdots$