

Why?

Let's say you want to show that the language ALL_{CFG} is undecidable.

$\text{ALL}_{\text{CFG}} = \{ \langle G \rangle \mid G \text{ is a CFG and } L(G) = \Sigma^* \}$

$A_{\text{TM}} \leq \text{ALL}_{\text{CFG}}$



We are going to look at theories that would allow us to do reductions such as $A_{\text{TM}} \leq \text{ALL}_{\text{CFG}}$.

M

| a | b | x | x | a | b | u | u | u | u | - - -



?

abxxqab

Definition

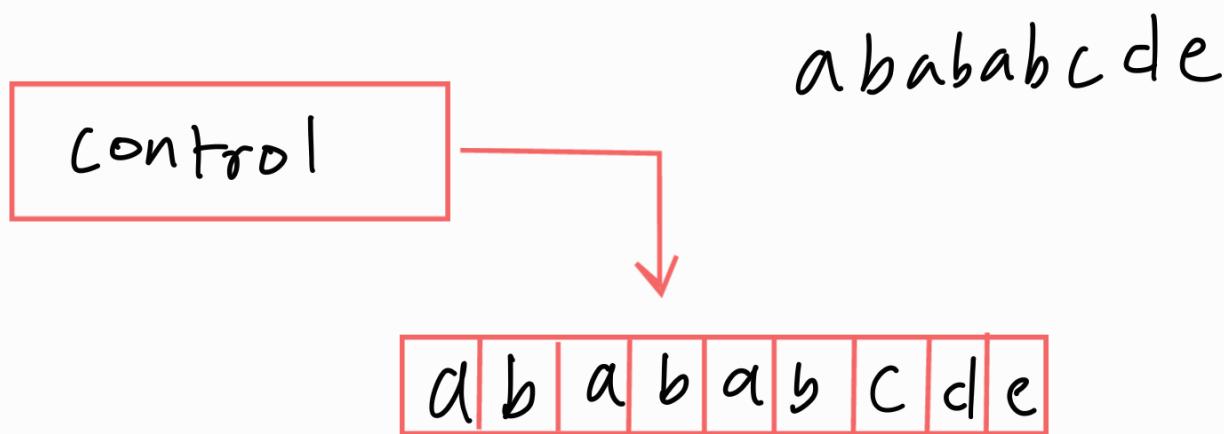
Let M be a Turing Machine and w an input string. An accepting computation history for M on w is a sequence of configurations c_1, c_2, \dots, c_e where c_1 is the start configuration of M on w , c_e is the accepting configuration of M on w , and each c_i legally follows from c_{i-1} according to rules of M .

A rejecting configuration is defined similarly, except c_e is the rejecting configuration.

- * Computation histories are finite sequences.
- * If M does not halt on w , then there are no accepting or rejecting computation history

Linear Bounded Automaton (LBA)

This is a restricted type of TM wherein the tapehead isn't permitted to move off the portion of the tape containing the input. If the machine tries to move its head off either end of the input, the head stays where it is.

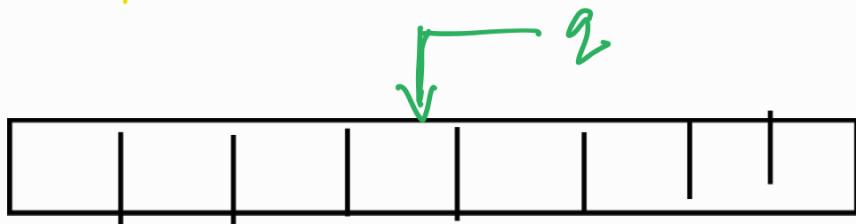


Now we define the following language

$$A_{LBA} = \{ \langle M, w \rangle \mid M \text{ is a LBA and } M \text{ accepts } w \}$$

Theorem 5.9 A_{LBA} is decidable.

Lemma 5.8 : Let M be a LBA with q states and g symbols in the tape alphabet. There are exactly $q^n g^n$ distinct configurations of M on a tape of length n .



$$g \times g \times g \times g \times \dots \times = g^n$$

$$q^n g^n$$

g - Size of the tape alphabet

n - Size of the input

q - # of states in the LBA.

A_{LBA} is decidable proof:

I am going to create a decider for A_{LBA} where the input is $\langle M, w \rangle$ M is a LBA, w is a string.

I am going to create TM L that decides A_{LBA} .

$L =$ on input $\langle M, w \rangle$, where M is a LBA and w is a string

1. Simulate M on w for qn^q^n steps or until it halts.
2. If it halts on an accept state, accept
3. If it halts on a reject state, reject
4. If M does not halt during

q^{ng^n} steps , reject.

// M exhaust all configs

// There is a loop

Theorem 5.10 E_{LBA} is undecidable.

$$A_{TM} \leq E_{LBA}$$

$$E_{LBA} = \{ \langle M \rangle \mid M \text{ is a LBA and } L(M) = \emptyset \}$$

Idea: I am going to create a LBA B such that the language of B recognizes all accepting computation histories for M on w . If M accepts w , $L(B)$ contains exactly one string that is the computation history for M on w . If M does not accept w , $L(B) = \emptyset$.

We construct LBA B as follows:
(M and w are encoded inside this)

$B = \text{on input } \langle x \rangle$

1. If x is an accepting computation history for M on w , then accept x .

* B looks at following 3 conditions,

1. C_1 is the starting config for M on w .
2. Each C_{i+1} legally follows C_i
3. C_ℓ is the accepting config for M on w .

$\# C_1 \# C_2 \# C_3 \# \dots \# C_\ell \#$

Assume R is the decider for E_{LBA} .

We construct TM S for A_{TM}

S = on $\langle M, w \rangle$, where M is a TM
and w is a string

1. Construct LBA B with $\langle M, w \rangle$ encoded inside it
2. Run R on $\langle B \rangle$
3. If R accepts ; reject
4. If R rejects ; accept

$\therefore E_{LBA}$ is undecidable

$$\text{ALL}_{\text{CFG}} = \{ \langle G \rangle \mid G \text{ is a CFG and } L(G) = \Sigma^* \}$$

Proof is similar to theorem 5.10 but with a slight twist.

$$A_{\text{TM}} \leq \text{ALL}_{\text{CFG}}$$

Proof Idea: I am going to create a CFG G for a given $\langle M, w \rangle$ where $L(G) = \Sigma^*$ if and only if M does not accept w .

If M accepts w , we make sure that the computation history for M on w is not generated by the G .

To make my life easier we are going to create a PDA D that behaves as above.

- Let the computation history for M on w be

$$\# c_1 \# c_2 \# c_3 \# \dots \# c_l \#$$

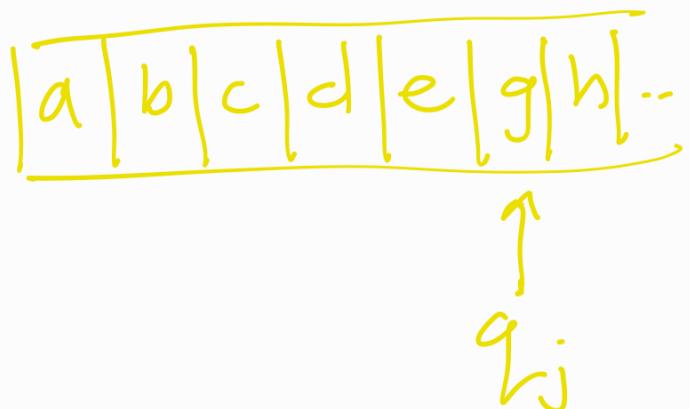
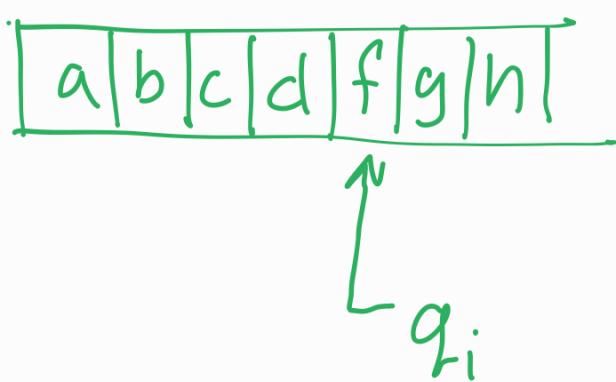
- D accepts all strings

1. do not start with c_1^* branch
2. do not end with c_l^* branch
3. Some c_i does not yield c_{i+1}

- Scan the input until it non-deterministically decides that it has reached c_i . It pushes c_i on the stack until we reach $\#$. Then D pops the stack to compare c_{i+1} .

They should match all except the head position.

$a b c d q_i f g h \Rightarrow a b c d e q_j g h$



$$f(q_i, f) = (q_j, e, R)$$

Proof: Assume R is the decider for ALL_{CFG} . Construct TM S that decides A_{TM} .

S = "on input $\langle M, w \rangle$ where M is a TM and w is a string

1. construct PDA D with $\langle M, w \rangle$ encoded inside.
2. convert D into CFG G .

3. Run R on $\langle G \rangle$

4. If R accepts, reject

If R rejects, accept

S is a decider for A_{TM} .

But A_{TM} is undecidable.

This is a contradiction.

\therefore ALL_{CFG} is undecidable.

