LBA — CSL

$$\alpha A \beta \rightarrow \alpha X \beta$$

$a_1 a_2 a_3 \, q \, a_4 a_5 a_6$ ← configuration

M is an LBA w/ q states
g symbols and input is
length n

distinct config. $= q \cdot n \cdot g^n$

Computational History

- sequence of configurations

Accepting Comp. History

$C_1, C_2, C_3, \ldots C_\ell$

a) C_1 initial config

b) C_ℓ accepting configuration

c) each C_i follows from C_{i-1} according to transitions of M.

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

A_{LBA} is decidable

L = on input $\langle M, w \rangle$

1. run M on w for $q \cdot n \cdot q^n$ steps, or until M halts.

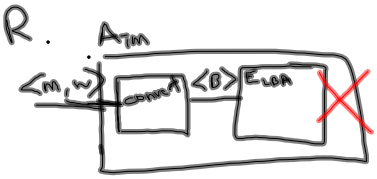
2. If M accepts, accept
   If M rejects, reject
   Else reject (loops forever)

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

is not decidable

Proof (reduction from A_{TM})

Suppose E_{LBA} is decidable w/ decider R.



Construct B such that $L(B) \neq \emptyset$ iff M accepts w.

LBA B recognizes all accepting computational histories of M on input w.

B = on input $\#C_1\#C_2\#C_3\#\ldots\#C_\ell\#$ where c_i is a configuration of M

1. check if c_1 is the initial configuration of M on w.
   e.g. $q_0 w_1 w_2 w_3 \ldots w_n$
2. check if C_ℓ is an accepting config. of M on w.
3. for each pair $c_i \# c_{i+1}$ check that c_{i+1} follows from c_i given M.

$$w_1 w_2 \ldots w_k \boxed{q_u w_{k+1}} \ldots$$

---

if $L(B)$ is empty, then there are no accepting comp. histories for M on w.

if $L(B)$ is not empty, then there is some accepting comp. history.

S = on input $\langle M, w \rangle$
1. construct LBA B as described.
2. Run R on $\langle B \rangle$
3. If R rejects, accept.
   If R accepts, reject.