Thm. 5.10 E_{LBA} is undecidable.

IDEA: We construct LBA B:

$A_{TM} \smile E_{LBA}$
$\langle M, w\rangle \quad \langle B\rangle$

B: on input x

- If x is an accepting computation history for TM M on w, then B accepts x.

M accepts $w \Rightarrow x$ is an accepting computation history for M on w

$\Rightarrow B$ accepts $x \Rightarrow L(B) \neq \phi$.

M doesn't accept $w \Rightarrow B$ accepts nothing $\Rightarrow L(B) = \phi$.

You still need to check 3 conditions:

1. C_1 is the start configuration,
2. Each C_{i+1} legally follows from C_i,
3. C_ℓ is an accepting configuration.

proof: Assume that R is a decider for E_{LBA}.
Construct TM S for A_{TM}.

S: on $\langle M, w\rangle$

1. Construct LBA B (from M and w).
2. Run R on input $\langle B\rangle$.
3. If R accepts, reject,
   If R rejects, accept.

$\therefore S$ is a decider for A_{TM}, a contradiction.

$\therefore E_{LBA}$ is undecidable. $\square$

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

Thm 5.13 ALL_{CFG} is undecidable.

IDEA: A_{TM} ? ALL_{CFG}

$\langle M, w\rangle \quad \langle G\rangle$

As CFG is equivalent to PDA, we construct a PDA D:

- Let the computation history for M on w be $\#C_1\#C_2\#\cdots\#C_\ell\#$.
- D accepts all strings that
  1. do not start with C_1,
  2. do not end with C_ℓ,
  3. Some C_i don't yield C_{i+1}.

M accepts w $\Rightarrow$ there is a valid comp. history (which D can't accept)
$\Rightarrow L(D) \neq \Sigma^*$

M doesn't accept w $\Rightarrow$ there is no valid comp. history
$\Rightarrow L(D) = \Sigma^*$

As PDA D uses a stack, we should reorder the history to facilitate the checking $C_i \to C_{i+1}$:

$\#C_1\#C_2^R\#C_3\#C_4^R\#\cdots\cdots$

$\boxed{\text{Proof}}$: Let R be a decider for ALL_{CFG}, construct TM S for ATM:

S: on $\langle M, w\rangle$
① Construct PDA D.
② Run R on $\langle D\rangle$.
③ If R accepts, reject; if R rejects, accept.

$\therefore$ S is a decider for A_{TM}, a contradiction.

$\therefore$ ALL_{CFG} is undecidable. $\square$