Reductions using computation histories

(adapted from Sipser Definition 5.5) Let M be a Turing machine and w an input string. A computation history for M on w is a finite sequence of configurations, $C_1, C_2, \cdots, C_l$, where C_1 is the start configuration of M on w, and each C_i legally follows from C_{i-1} according to the rules of M. A computation history is *accepting* if C_l is an accepting configuration, and *rejecting* if C_l is a rejecting configuration.
Assume that we encode computation history $H = C_1, C_2, \cdots, C_l$ by
$\langle H \rangle = \#C_1\#C_2^R\#C_3\#C_4^R \cdots C_l^{(R)}\#$, i.e. we reverse every other configuration and intersperse # signs around them. Fix TM M and string w, and let $ACCHIST_{M,w} = \{\langle H \rangle \mid H$ is an accepting configuration history for M on $w\}$. We are going to do parts of the construction for a PDA which recognizes the language $\overline{ACCHIST_{M,w}}$. Describe all PDAs at a high level (you don't need to actually draw state diagrams).

a Describe a PDA which accepts if a given string x is *not* a valid configuration of M. (Don't worry about whether x is a *reachable* configuration; just check whether it has the correct form to be a configuration at all.)

b (For this and all remaining parts of the construction, you may assume that the input $\langle H \rangle$ does have at least two # symbols, and that between each pair of # symbols is some valid configuration of M.) Describe a PDA which accepts input $\langle H \rangle$ if H does *not* start with a valid C_1 (for our fixed M and w).

c Describe a PDA which accepts if input $\langle H \rangle$ does *not* end with a valid accepting configuration C_l.

d Assume that we have some way to finish this construction, i.e. given a TM M and a string w we can create a PDA P which recognizes $\overline{ACCHIST_{M,w}}$. Show that $ALL_{PDA} = \{\langle M \rangle \mid M \text{ is a PDA and } L(M) = \Sigma^*\}$, and the similarly-defined ALL_{CFG}, are undecidable languages.

e Let's go back and fill in more details of the PDA's construction. Assume for this part that $H = C_1, C_2$, and that the final symbols of C_1 and C_2 are the same. Describe a PDA which accepts input $\langle H \rangle$ if C_2 is *not* the valid next configuration for M starting from C_1 (do not assume that C_1 is the starting configuration).

f Extending part e, describe a PDA which accepts input $\langle H \rangle$ if *some* C_i is not the valid next configuration for M starting from C_{i-1}. Then once we have all the pieces of the construction, how do we tie them together into a single PDA which recognizes $\overline{ACCHIST_{M,w}}$?

g If we tried to adapt our construction to recognize $ACCHIST_{M,w}$ instead, what parts could be easily adapted and what parts could not?