1. (10 points) In this problem you will give the details of the reduction of the acceptance problem for Turing machines to the non-emptiness problem for two-headed finite automata, thereby showing its undecidability. Let $M = (Q, \Sigma, \Gamma, \delta, q_0, q_a, q_r)$ be a one-tape Turing machine and let $w \in \Sigma^*$ be an input. The goal is to construct a two-headed finite automaton $M' = (Q', \Sigma', \delta', F)$ such that $M$ accepts $w$ if and only if $M'$ accepts some input. As described in class $M'$ will accept only strings that represent accepting computation histories of $M$ on input $w$. Such a string has the form $C_1 \# C_2 \# \cdots \# C_m \$ where (i) $C_1 = q_0 w \# k$ for some $k$, (ii) $C_{i+1}$ follows from $C_i$ in one step of $M$ for $1 \leq i < m$, (iii) $C_m$ contains $q_a$, and (iv) $|C_1| = |C_i|$ for $1 \leq i \leq m$. Recall that $M'$ runs in three phases. In the first phase, the first heads scans the input to the first $\#$ checking condition (i) above. In the second phase, the two heads scan the input in parallel checking condition (ii) above. At the end of the second phase, if all goes well, the first head is on the $\$ and the second head is on the last $\#$. In the third phase, the second head scans the string between the last $\#$ and the $\$ checking for condition (iii) above. If all goes well, condition (iv) holds.

Recall that in the second phase, after a brief startup, $M'$ uses states of the form $(a_1, a_2, a_3, b_1, b_2, b_3)$ which are six-tuples of symbols from $Q \cup \Gamma \cup \{\#, \}$.

(a) Precisely define all of the states $Q'$ of $M'$. Include the states for all three phases.

(b) Precisely define all the input symbols $\Sigma'$ of $M'$.

(c) Precisely define the transition function $\delta'$ of $M'$. The function $\delta'$ maps $Q' \times \Sigma' \times \Sigma'$ to $Q' \times \{R, S\} \times \{R, S\}$. $R$ means move the head right while $S$ means keep it stationary.

(d) Precisely define the set of final (accepting) states $F$.

2. (10 points) In this problem you will show that the non-empty intersection problem for context-free grammars is undecidable. The non-empty intersection problem is given to context-free grammars, do they generate a common terminal string. That is, given $G_1$ and $G_2$, is $L(G_1) \cap L(G_2) \neq \emptyset$. This can
be done by showing that PCP is reducible to this problem. Use the solution to problem 5.21 on page 212 of Sipser as a hint.