For #1, the algorithm discussed in class for constructing a machine M to accept $L_1 \cap L_2$ given machines M_1 and M_2 for L_1 and L_2 respectively is for DFAs. The two NFAs given in #1 must first be converted to DFAs.

For #4d, $n + m = k + l$ means that only the total number of as and bs must match the total number of cs and ds — *abbbcddd* and *abbbccdd* are both legal strings. Also, your grammar must be able to generate any legal string in this language — strings where $n = l$ and $m = k$ are in the language, but there are also many other strings not of that form that are in the language, so a grammar generating only $\{a^n b^m c^k d^l \mid n = l \text{ and } m = k\}$ isn't correct.