This is not a complete review, but only deals with those topics that, based on my experience, I have experienced that students have trouble with. I will simply list a number of things you should know.

1. All sliding block problems are $\mathcal{P}$–SPACE. Some of them, including *Rush Hour*, are $\mathcal{P}$–SPACE-complete.

2. We will be spending a lot of effort on $\mathcal{NP}$-completeness. For the test on September 18, I expect you to know that SAT and IND (the independent set problem) are $\mathcal{NP}$-complete. That list will increase.

3. Equivalence of regular expressions is far harder than you might guess. In fact, it is $\mathcal{P}$–SPACE-complete.

4. Equivalence of DFAs is $\mathcal{P}$–TIME. Just replace each one by a minimal DFA and see whether they match.

5. Let $L_1 = \left\{a^i b^i c^j\right\}$ and $L_2 = \left\{a^i b^j c^j\right\}$. Both are CF. L_1 is generated by the CF grammar:
   1. $S \to AB$
   2. $A \to aAb$
   3. $A \to \lambda$
   4. $B \to cB$
   5. $B \to \lambda$

   However, their intersection is $\{a^n b^n c^n\}$, which is not context-free.

6. No algebraic language which has matching parentheses is regular. This can be proved using the pumping lemma, which we haven't covered yet.

7. The philosopher Eratosthenes, who measured the size of the Earth around 200 BC, also invented an algorithm for finding primes, called the Sieve of Eratosthenes. Christopher Columbus had trouble getting funding for his journey because most scientists in his time believed that Eratosthenes' measurement of the size of the Earth was correct, and that Columbus couldn't possibly get to China because it was too far. They were right, but Columbus ran into America before he ran out of supplies.

   The ancient Greeks did not have an efficient way for determining whether a given integer was prime. Finally, after waiting 2200 years, a polynomial time algorithm was found in 2002.

8. No polynomial time algorithm for factoring integers is known. RSA encryption is only secure if factoring is hard. If anyone ever finds a $\mathcal{P}$–sc time algorithm for factoring, RSA encryption will become insecure. Factoring is known to be both $\mathcal{NP}$ and co-$\mathcal{NP}$, but it is not known to be $\mathcal{NP}$-complete.

9. Context-free grammar equivalence is undecidable. As a consequence, the task of grading problem 6/7 in homework1 is, in general, impossible! If you gave an incorrect matching, Miss Wallace would eventually be able to prove it's incorrect. But there is no algorithm which can always find a proof that a given PDA accepts the language generated by a given CFL, if it's true.