Solutions to Homework # 0

1. I only prove for (2). Suppose that M_1 is a finite automaton accepting L_1 and M_2 is a PDA accepting L_2. We now construct a PDA M accepting L, then the result follows. On an input word w, M simulates M_1 and M_2 (using M's stack to simulate the stack in M_2) in parallel while reading symbols from w. For each symbol, M, nondeterministically, either feeds it to the input of M_1 or to the input of M_2. M accepts w when both M_1 and M_2 accepts. It is left to you to show that M, which is a PDA, indeed accepts $L_1 \| L_2$.

2. I only show (1). Let M be an NFA accepting L. I change M to M' as follows: whenever M reads $a \in \Sigma$, M' will read $h(a) \in \Sigma'$. M' will preserve all the state transitions of M. Clearly, M' accepts $h(L)$. The result follows.

3. Talked in class.

4. Notice that a B-bounded PDA is equivalent to an NFA. It is also known that the given language is not regular. The result follows.

5. Without loss of generality, we assume that the PDA M never reads the input tape.

Given a state s of M, we say M has a s-loop if M can (starting from the initial state) reach state s (suppose a is the current top of the stack) and later come back to state s during which the top symbol a is not popped out. Key observation: M has an infinite execution sequence iff M has a s-loop for some s. So, the result follows if we can detect, for any given s, whether M has a s-loop. Let s be any fixed state. Let M' be a PDA that works as follows. On an input word w, M' faithfully simulates M. Notice that even though M' has the input, it, for now, does not read it. At some moment nondeterministically chosen when M enters state s, M' decides to do something as follows. M' remembers the current stack top symbol, say a, and push a new symbol # onto the stack. Then, M' continues to simulate M and, using its finite control, treats # as a in M. M' crashes whenever # is popped out. M' accepts the input when some moments later, s is reached. Clearly, M has a s-loop iff $L(M')$ is not empty. But, since M' is a PDA and you know there is an algorithm to decide whether $L(M')$ is empty, the result follows.

6. Explained in class. Will read your answer.

7. In class. Also it is helpful to read Alur and Dill's classic paper in *Theoretical Computer Science* **126**(2): 183–235, 1994.
(WSU (digital) library has free access)