CS 334 Fall 2020: Problem Set 7.

Problem 1. (5 points) A 2-stack PDA (2PDA for short) is a pushdown automaton with two stacks. In one step each stack can be popped or pushed independently. For example, the 2PDA may read an input symbol, pop the first stack, ignore the second stack and, based on the values seen, change state, push different symbols on the two stacks. The input tape is read only.
Describe a 2-stack PDA that accepts the language $L = \{a^i b^k c^i d^k \ : i, k \geq 0\}$. You do not need to give a PDA diagram – a high-level, but complete, description of how your PDA works will suffice.

Problem 2. (10 points) In Problem Set 6 you showed that the language $L_{mult} = \{a^i b^{ij} c^j : i, j \geq 0\}$ is not context free. Either show that this language can be recognized by a 2PDA or explain why that is not possible.

Problem 3. (10 points) Prove that the intersection of a CFL and a Regular language is always context free.

Optional Problem 4. (10 points) Prove that the language $A \backslash B = \{w : wx \in A, x \in B\}$, where A is a CFL and B is regular is a CFL.

Optional Problem 5. (10 points) A ***queue automaton*** is like a push-down automaton except that the stack is replaced by a queue. A queue is a tape allowing symbols to be written only on the left-hand end and read only at the right-hand end. Each write operation (we'll call it a push) adds a symbol to the left-hand end of the queue and each read operation (we'll call it a pull) reads and removes a symbol at the right-hand end. As with a PDA, the input is placed on a separate read-only input tape, and the head on the input tape can move only from left to right. The input tape contains a cell with a blank symbol following the input, so that the end of the input can be detected. A queue automaton accepts its input by entering a special accept state at any time. Show that a language can be recognized by a deterministic queue automaton iff the language is Turing-recognizable.