Fundamentals

1. *Definitions.* Be able to define carefully:
   - DFA, NFA, and regular expression.
   - PDA and context-free grammar.
   - Ambiguous CFG
   - Chomsky normal form.
   - Turing machine
   - Turing-recognizable set, Decidable set

2. *Theorems.* The following are the cornerstone results of the course. Be able to state each of them correctly.
   - Equivalence theorems between NFAs, DFAs, and regular expressions.
   - Equivalence theorems between PDAs and context-free grammars.
   - The Pumping Lemmas (regular and context-free)
   - The undecidability of the halting problem.

3. *Proofs.* You should be able to provide proofs for the following fundamental results.
   - For each class of languages (regular, context-free, etc) know which of the basic closure properties (closure under union, intersection, complement, etc) holds; be able to prove or provide counterexamples where appropriate.
   - Proof of the Pumping Lemmas for regular sets.
   - Simple proofs of non-regularity and non-context-freeness for various languages, *using* each of the pumping lemmas;
   - Proof of the undecidability of the halting problem.

Sample questions:

1. Be able to translate between NFAs, DFA, and regular expressions.

2. Be able to decide whether equations between regular expressions are valid.

3. Be able to put a context-free grammar into Chomsky normal form.
4. Complete each of the following sentences with one of the following: “regular”, “context-free”, “decidable”, “recognizable”, “co-recognizable”, or “none of the above”. You are to make the strongest claim that is always true. For example, for the question “If A and B are regular then A ∪ B is . . .” completing the sentence with “decidable” would make a true statement, but would earn you no points here, since completing the sentence with “regular” would make a stronger true statement. (Recall that to say that a set X is co-recognizable is simply to say that the complement of X is recognizable.)

- If A is context-free and B is context-free then A ∪ B is . . .
- If A is context-free and B is context-free then A ∩ B is . . .
- If A is context-free and B is context-free then A \ B is . . .
- If A is context-free and B is context-free then AB is . . .
- If A is context-free then Σ* \ A is . . .
- If A is context-free then A* is . . .

NOW, replace “context-free” by “regular”, “decidable”, “recognizable”, or “co-
recognizable”. Many possible questions here. Mix and match. Lots of fun.

5. Categorize each of the following languages as being regular, context-free, or “neither.” Give the most restrictive category that applies.

(a) \{a^n b^{2m} | n ≥ 0, m ≥ 0\}
(b) \{a^n b^m | n ≠ m\}
(c) \{a^n b^m c^{2k} d^l | 2n = 3k or 5m = 7l\}
(d) \{a^n b^m c^{2k} d^l | 2n = 3k and 5m = 7l\}
(e) \{a^n b^m c^{2k} d^l | 2n = 3m and 5k = 7l\}
(f) \{a^n b^m c^{2k} d^l | 2n = 3l and 5k = 7m\}
(g) \{x | #(a) in x > #(b) in x\}
(h) \{x | #(a) in x < 2#(b) in x\}
(i) \{a^n | n is a power of 2\}

6. Categorize each of the following languages as being Turing-decidable, Turing-
recognizable, co-Turing-recognizable, or “none of these”. Give the most restrict-
tive category that applies.

(a) \{<G> | G is a CFG and L(G) = ∅\}
(b) \{<G, H> | G and H are CFGs and L(G) = L(H)\}
(c) \{<G, H> | G and H are CFGs and L(G) ⊆ L(H)\}
(d) \{< M > | M is a Turing machine with fewer than 117 states.\}
(e) \{< M > | M is a TM accepting all inputs\}
(f) \{< M > | M is a TM taking fewer than 117 steps on some input.\}
(g) \{< M > | M is a TM taking fewer than 117 steps on all inputs.\}
(h) \{< M > | M is a TM taking fewer than 117 steps on at least 117 different inputs.\}
(i) \{< M > | M is a TM accepting at least 117 different inputs.\}
(j) \{< M > | M is a TM accepting at most 117 different inputs.\}
(k) \{< M > | M is a TM accepting all inputs of length at most 117.\}
(l) \{< M, N > | M and N are TMs and \(L(M) = L(N)\).\}
(m) \{< M, N > | M and N are TMs and \(L(M) \neq L(N)\).\}
(n) \{< P, Q > | P and Q are C++ programs and P and Q compute the same function.\}
(o) \{< P, Q > | P and Q are C++ programs and P and Q compute different functions.\}
(p) \{< P > | P is a C++ program and P terminates normally on input 0.\}
(q) \{< P > | P is a C++ program and P terminates normally on all inputs.\}