NAME: ______________________________

SPIRE ID:______________________________

COMPSCI 250
Introduction to Computation
Final Exam Fall 2023

D. A. M. Barrington and G. Parvini 12 December 2023

DIRECTIONS:

- Answer the problems on the exam pages.
- There are four problems on pages 2-9, some with multiple parts, for 125 total points plus 5 extra credit. Final scale will be determined after the exam.
- Page 10 contains useful definitions and is given to you separately – do not put answers on it!
- If you need extra space use the back of a page –both sides are scanned.
- If absolutely necessary, you may attach additional pages to be graded, but this is a big nuisance to us.
- No books, notes, calculators, or collaboration.
- Scrap paper during the exam may be used, if it starts out blank. Please transfer any answers to the exam pages.
- You may say "pass" for any question or lettered part of a question, except for the true/false and the extra credit questions, and receive 20% of the points.

1	/35
2	/10
3	/20
4	/40+5
5	/20
Total	/125+5

Question 1 (35): (Dog Proof) On Thanksgiving at Dave's house, his dogs Blaze and Rhonda were joined by his daughter's dog Gwen. The family observed the sounds made by the three dogs – barking, growling, and whining.

Given the set of dogs $D = \{B, G, R\}$, the set of sounds $S = \{ba, gr, wh\}$, and the set of numbers $N = \{0, 1, 2\}$, we define a function f from $D \times S$ to N such that if x is in D and y is in S, $f(x, y) = 0$ if dog x *never* makes sound y, $f(x, y) = 1$ if x *rarely* makes sound y, and $f(x, y) = 2$ if x *often* makes sound y.

We also define three propositions p as "$f(B, ba) = 2$", q as "$f(G, ba) = 2$", and r as "$f(R, ba) = 2$".

Question 1a (10): Translate the following five statements as indicated:

Statement I (to symbols): Either Blaze or Gwen, but not both of them, barks often.

Statement II (to English): $((f(R, ba) = 2) \vee (f(G, ba) = 2)) \wedge (f(B, ba) = 2)$

Statement III (to symbols): There is no dog that never barks.

Statement IV (to English): $\forall n : (\exists d : f(d, gr) = n) \wedge (\exists s : f(R, s) = n) \wedge (\exists t : f(G, t) = n)$

Statement V (to symbols): Given any two sounds, Blaze has the same frequency for each.

Question 1b (10): Using only Statement I and Statement II, determine the truth value of the three propositions p, q, and r. You may use either a truth table or deductive rules. If you do the latter, remember that for full credit you must verify that your solution satisfies both the rules.

Question 1c (15): Using all of Statements I-V, determine the nine values of the function f. That is, for each pair (x, y) in the domain $D \times S$, find $f(x, y)$.

Question 2 (10): (Induction 1) Prove, by induction, *either* one of the following two statements:

"There exists a natural m such that for any natural n, $(n \geq m) \rightarrow (n! < 3^n)$."

"There exists a natural m such that for any natural n, $(n \geq m) \rightarrow (3^n < n!)$."

(We could offer you extra credit for proving both, but we won't.)

(**Hint:** This can be done with ordinary induction, for an appropriate base case.)

Question 3 (20): (Induction 2) Recall that a **rooted binary tree (RBT)** consists of either

- a single node, which is its root, or
- a new node, its root, which is connected to the roots of two other RBT's.

An **edge 3-coloring** is an assignment of colors (red, blue, or green) to the *edges* of an RBT. A coloring is **valid** if there is no node that is adjacent to two edges of the same color.

Question 3a (5):

Give a valid edge 3-coloring of the following RBT:

```
    A
   / \
  /   \
 /     \
B       C
       / \
      /   \
     /     \
    D       E
   / \     / \
  /   \   /   \
 F     G H     I
```

Question 3b (15):

Prove that given any RBT T, we can make a valid edge 3-coloring for T.

(**Hint:** Prove this by induction on the definition of RBT's, but let your $P(T)$ be "for any color c, we can make a valid edge 3-coloring of T such that the root has no adjacent edge of color c".)

Question 4 (40+5): (Kleene Constructions) Let N be the following λ-NFA. The alphabet is $\Sigma = \{a, b\}$. The state set is $\{i, p, q, r\}$, the start state is i, and the final state set is $\{q, r\}$. The transitions are (i, a, p), (i, a, r), (p, b, p), (p, b, q), (p, b, r), (q, a, r), (q, λ, r), and (r, λ, p).

(a, 10) Apply the Killing λ-Moves Construction to find an ordinary NFA N' such that $L(N') = L(N)$. If you deviate from the construction given in lecture and the textbook, explain why your construction is still correct. The correct NFA has four a-moves and nine b-moves.

(b, 10) Using the Subset Construction on N', create a DFA such that $L(D) = L(N')$.

(c, 5XC) Is your constructed DFA D minimal for its language? If so, argue that it is minimal, either using the Minimization Construction or by direct use of distinguishability. If not, find a DFA D' such that $L(D') = L(D)$ and D' is minimal, justifying your answer if you did not use the Minimization Algorithm.

(d, 10) Using the DFA D (or an equivalent D' if you found one), find a regular expression R such that $L(R) = L(D)$.

(e, 10) Compute a λ-NFA N'' such that $L(N'') = L(R)$. Use a construction starting from R rather than just using N. If you use a construction other than the one given in the lectures and the textbook, explain it and argue that it is correct.

Question 5 (20): The following are ten true/false questions, with no explanation needed or wanted, no partial credit for wrong answers, and no penalty for guessing.

- (a) Let n be a positive natural, and let R be the equivalence relation of mod-n congruence on the *positive* naturals. Then R divides the positive naturals into exactly $n-1$ equivalence classes.

- (b) Let N be an NFA, and let N' be a new NFA that is identical to N except that every final state of N is non-final in N', and vice versa. Then $L(N)$ must be the complement of $L(N')$.

- (c) Let Σ be any finite alphabet. Then the language $(\Sigma^0)^*$ is non-empty if and only if Σ itself is non-empty.

- (d) Let x be any natural. Then if x is divisible by either 3 or 7, then it is also divisible by 21.

- (e) Let G be directed graph with positive edge labels, and let h be an admissible, consistent heuristic for some goal node g. Suppose we conduct two searches with s as start node and g as the goal node, one a uniform-cost search and the other an A^* search using h. Then both searches will return the same distance from s to g.

- (f) Let M be an ordinary Turing machine with tape alphabet $\{a, b, \Box\}$, where $\Box$ is the blank symbol. Let p, q be states of M with $\delta(p, a) = (q, a, R)$ and $\delta(q, a) = (p, a, L)$. Then if we start M in configuration $\Box bpaab\Box$, M will never halt.

- (g) Let R and S be any two regular expressions. Then the expressions $(R+S^*)(R^*+S)$ and $(R^*+S)(R+S^*)$ may fail to have the same language.

- (h) Let w be a binary string. Then the set of substrings of w is equal to the set of prefixes of suffixes of w.

- (i) Let M be a five-tape Turing machine. Then there exists an ordinary single-tape Turing machine M' such that $L(M') = L(M)$.

- (j) Let A and B be two finite sets, and let f be a surjective (onto) function from A to B that is not a bijection. Then there exists an injective (one-to-one) function g from B to A that is not a bijection.

COMPSCI 250 Final Exam Supplementary Handout:12 December 2023

The statements of Question 1a are:

Statement I (to symbols): Either Blaze or Gwen, but not both of them, barks often.

Statement II (to English): $((f(R, ba) = 2) \vee (f(G, ba) = 2)) \wedge (f(B, ba) = 2)$

Statement III (to symbols): There is no dog that never barks.

Statement IV (to English): $\forall n : \exists d : f(d, gr) = n) \wedge (\exists s : f(R, s) = n) \wedge (\exists t : f(G, t) = n)$

Statement V (to symbols): Given any two sounds, Blaze has the same frequency for each.

The λ-NFA N defined in Question 4a is as follows:

The alphabet is $\Sigma = \{a, b\}$. The state set is $\{i, p, q, r\}$, the start state is i, and the final state set is $\{q, r\}$. The transitions are (i, a, p), (i, a, r), (p, b, p), (p, b, q), (p, b, r), (q, a, r), (q, λ, r), and (r, λ, p).