NAME: ____________________________

SPIRE ID: ____________________________

COMPSCI 250
Introduction to Computation
Final Exam SOLUTIONS Spring 2023

D. A. M. Barrington and G. Parvini 24 May 2023

DIRECTIONS:

- Answer the problems on the exam pages.
- There are six problems on pages 2-10, some with multiple parts, for 120 total points plus 10 extra credit. The probable scale is around A=110, C=70, but will be determined after we grade the exam.
- If you need extra space, use the back of a page.
- No books, notes, calculators, or collaboration.
- In case of a numerical answer, an arithmetic expression like "$2^{17} - 4$" need not be reduced to a single integer.
- If you don't know the answer to any question, you can just write "Pass" or "I don't know", and you will receive 20% of the points of that question. You can use it as many times as you want. Note that it doesn't apply to wrong answers, and you may get less than 20% if you attempt a question and your answer is not correct. (It would be silly to "Pass" on a true/false question, since it's far better for you guess, and to save us grading effort we will not allow you to pass on those.)

Question 1 (35): Dave's dogs Blaze and Rhonda often see animals during their daily walks. They walk through five locations, the Neighborhood, the Farm, the Woods, the Park, and the Village.

The relation X is a subset of $D \times A \times L$, where $D = \{b, r\}$ is the set of dogs Blaze and Rhonda, $A = \{O, R, S\}$ is the set of species they may have seen, Opossum, Rabbit, or Squirrel, and $L = \{f, n, p, v, w\}$ is the set of five locations. The predicate $X(d, a, \ell)$ means "dog g observed an animal of species a in location ℓ". (Sorry that "Rhonda" and "Rabbit" have the same letter — we will use r for the dog and R for the species.)

- (a, 10) **Translations:** Translate these four statements as indicated (point values as indicated):

  Statement I: (to symbols, 3) An Opossum was seen in exactly one location, both dogs saw it, and it was the only location in which Blaze saw both a Rabbit and a Squirrel.

  $\exists y : X(b, O, y) \wedge X(r, O, y) \wedge \forall z : (X(b, R, z) \wedge X(b, S, z)) \leftrightarrow (z = y))$

  This was a difficult one. I gave two of the three points for $\exists y : X(b, O, y) \wedge X(r, O, y) \wedge X(b, r, y) \wedge X(b, s, y)$, or anything that close. I did not make any note on Gradescope for which mistake you made in that two-point range – just check what you wrote against the solution. **Statement II:** (to English, 2)

  $\forall y : X(r, S, y)$

  Rhonda saw a Squirrel in every location.

  Pretty much everyone got this right. **Statement III:** (to symbols, 3) Wherever

  Rhonda saw a Rabbit, Blaze also saw a Rabbit in the same location.

  $\forall y : X(r, R, y) \rightarrow X(b, R, y)$

  I took off one point for a single error like the wrong quantifier, or $\wedge$ or $\leftrightarrow$ instead of the $\rightarrow$. More than one error, or a worse error, lost you two points.

  Statement IV: (to English, 2) $\neg(X(r, S, f) \leftrightarrow X(b, S, f))$

  At the farm, it was not the case that each dog saw a Squirrel if and only if the other did.

  Some people correctly translated this to mean $X(r, S, f) \oplus X(b, S, f)$, and others somehow got this turned around to say that the "if and only if" was true.

- (b, 10) **Boolean Proof:**

  Let $q_1 = X(b, R, n)$, $q_2 = X(b, R, p)$, $q_3 = X(b, R, v)$, and $q_4 = X(b, R, w)$. Determine the truth values of q_1, q_2, q_3, and q_4, given the premises

  $(q_2 \vee q_3) \rightarrow \neg q_1$,
  $\neg q_4 \rightarrow (q_1 \wedge \neg q_3)$, and
  $q_4 \rightarrow (q_1 \wedge q_2)$.

  You may use either a truth table or deductive proof rules.

  The truth table is omitted here.

  Both q_4 and $\neg q_4$ imply q_1, so q_1 must be true. By contrapositive, q_1 implies both $\neg q_2$ and $\neg q_3$. Since this makes the conclusion of the third premise false, q_4 must be false. With the setting where q_1 is true and the other three are false, the first premise is true vacuously, the second trivially, and the third vacuously.

  Most people did well, with either method though using the rules was more popular. I generally gave at most 4/10 if you got the wrong answer, or if you didn't state your answer. Some of the truth table people had trouble integrating the three table for the three premises, which need to be put together to talk about all four variables.

- (c, 15) **Dog Proof:** Using the four premises from part (a) *and* the three premises from part (b), determine where the Opossum was seen, and prove your answer.

  By Statement I, we may eliminate any location in which Blaze either fails to see a Rabbit or fails to see a Squirrel. From part (b), Blaze did not see a Rabbit in p, v, or w. In location f, Statement IV tells us that exactly one of the two dogs saw a Squirrel there. By Specification on II, Rhonda did see a Squirrel there, so Blaze did not see one there. The only location where Statement I can be fulfilled is n. Statement III takes no part in the proof.

  Most of these were good, though some people got in trouble from bad translations or from bad versions of Q1b. I gave around 9/15 if you had the right answer with bad reasoning. In particular, I gave 10/15 for the claim that the Neighborhood was the only place where Blaze saw a Rabbit, ignoring the possibility of the farm. For wrong answers, I gave 7/15 if you either dealt with the farm correctly or correctly ruled out the three places from Q1b. I took off a point if you reversed Statement III, or three points if you used this bad version in your argument.

Question 2 (15+10): Recall that the Fibonacci function $F(n)$ is defined by the rules $F(0) = 0$, $F(1) = 1$, and for all n with $n > 1$, $F(n) = F(n-1) + F(n-2)$.

- (a, 15) Show that for *all* naturals n, $F(1) + F(3) + \ldots + F(2n+1) = F(2n+2)$, that is:

$$\sum_{i=0}^{n} F(2i+1) = F(2n+2)$$

Base case: $n = 0$, the sum is $F(1)$, and $F(2 \cdot 0 + 2) = F(2)$, and this is true because $F(1) = F(2) = 1$.
IH: $\sum_{i=0}^{n} F(2i+1) = F(2n+2)$
IG: $\sum_{i=0}^{n+1} F(2i+1) = F(2n+4)$
Inductive step: The sum up to $n+1$ is the sum up to n plus one more term, $F(2n+3)$. By the IH, the sum up to n is $F(2n+2)$. So the sum up to $n+1$ is $F(2n+2)+F(2n+3)$, which is $F(2n+4)$ by the Fibonacci definition.

These went well in general, with the biggest errors coming in the base case, of all things. Part of the definition of the Fibonacci sequence mentions $n > 1$, and many of you thus ignored where you were asked to prove the statement *for all naturals*, and left off $n = 0$ (two points off) or also $n = 1$ (three points off). Many purported base cases didn't address the question. The correct base case for $n = 0$ requires that you consider that it says essentially that $F(0) = F(1)$.

- (b, 10XC) In Midterm #2, we learned that in the village of Gigili there is a single child for each age in the set $\{1, 2, 4, 8, \ldots\}$, that is, for all naturals of the form 2^k. We were asked to prove that any positive natural can be expressed as a sum of *distinct* numbers in this set, using induction.

  In the nearby village of Figili, they have an identical situation except the ages of the children in the set are $\{1, 2, 3, 5, 8, \ldots\}$, that is, all the Fibonacci numbers except $F(0) = 0$ and $F(1) = 1$. Prove that for all positive naturals n, by induction, that n can be expressed as a sum of *distinct* numbers in this set, with the added condition that the sum may not include two *consecutive* numbers. (For example, the sum could not include both 3 and 5, or both 5 and 8.)

Strong Induction on all positive naturals:

Base Case: $n = 1$, *true because we can make* 1 *by taking* $F(2) = 1$ *by itself.*

SIH: Every positive number i *with* $1 \leq n$ *can be made by sums of distinct numbers in the set, with no two numbers consecutive.*

IG: $n + 1$ *can be made by such a sum.*

Case 1: $n+1$ *is a Fibonacci number* $F(k)$. *Then we can make* $n+1$ *with* $F(k)$ *by itself, and the consecutive numbers rule is certainly satisfied.*

Case 2: $n+1$ *is not a Fibonacci number. Let* $F(k)$ *be the largest Fibonacci number with* $F(k) \leq n$. *Let* $m = n+1-F(k)$. *Note that* $m < F(k-1), sinceotherwiseif m \geq F(k-1)$, *we would have* $n + 1 \geq F(k + 1)$, *contradicting the choice of* $F(k)$. *By the IH,* m *is the sum of distinct numbers in the set, with no two numbers consecutive, and* $F(k-1)$ *cannot be in this sum. If we include* $F(k)$ *in this sum, the new total is* $n + 1$, *and we have not put two consecutive numbers into the set.*

You don't need to separate the case where $n + 1$ is a Fibonacci number, if you also prove the statement for $n = 0$. You could also do this by ordinary induction, with some effort, by showing that given a correct sum for n, you can add 1 and then replace terms of the form $F(i) + F(i + 1)$ with $F(i + 2)$ as long as you need to until there are no consecutive terms in the sum.

Question 3 (25): The next three questions all concern finite-state machines. Unlike in most previous other exams, each question is about a different language.

- (a, 10) Starting from the given λ-NFA, find the corresponding NFA using the Killing λ-Moves algorithm. Draw the λ-NFA and NFA and show all the steps. Your NFA should have three a-moves and five b-moves.

  $S = \{\iota, S_1, S_2, f\}$, start state ι, $F = \{f\}$, $\Sigma = \{a, b\}$, and transition relation $\Delta = \{< \iota, \lambda, S_1 >, < \iota, a, S_2 >, < S_1, b, S_2 >, < S_1, a, f >, < S_2, b, S_1 >, < S_2, \lambda, f >\}$.

  No new λ-moves are needed, as the current moves are transitively closed already. There is also no need to change the final state set, since there is no λ-path from ι to f.
  $< \iota, a, S_2 >$ *creates itself and* $< \iota, a, f >$. $< S_1, a, f >$ *creates itself and* $< \iota, a, f >$. $< S_1, b, S_2 >$ *creates itself and three more moves:* $< \iota, b, S_2 >$, $< S_1, b, f >$, *and* $< \iota, b, f >$. $< S_2, b, S_1 >$ *creates only itself. We created nine moves, but there are only eight because one was created twice. As claimed, we have three a-moves and five b-moves.*

- (b, 10) Starting from the NFA you found in part (a), apply the Subset Construction algorithm to find the corresponding DFA.

  State $\{\iota\}$ is the start state of the DFA and is non-final. Both its arrows go to the final state $\{S_2, f\}$. On input a, $\{S_2, f\}$ goes to the non-final death state $\emptyset$ (and of course $\emptyset$ has both arrows to itself). On input b, $\{S_2, f\}$ goes to the non-final state $\{S_1\}$. On input a, $\{S_1\}$ goes to the final state $\{f\}$, which in turn has both arrows to $\emptyset$. On input b, $\{S_1\}$ goes to $\{S_2, f\}$, a state we have seen. We have finished the construction with five states.

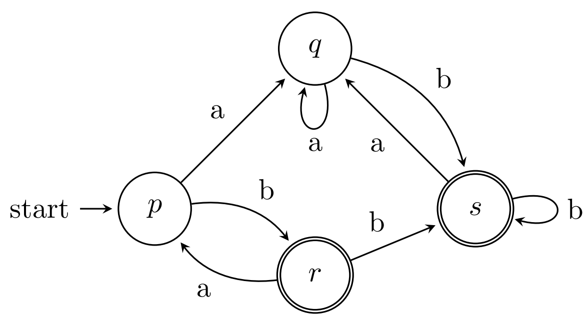
- (c, 5) Determine whether the DFA that you found in part (b) is minimal. Justify your answer.

  It is minimal. The two final states are separated by bb. Of the three non-final states, only $\{\iota\}$ is accepted after a, and only $\{S_2, f\}$ is accepted after bb. If we run the minimization algorithm, the first phase separates the two final states and separates $\{\iota\}$ from the other two non-final states. Then the next phase separates the other two states.

Question 4 (10): Find a λ-NFA for the given regular expression $[((a+b)^*ab)+(ba+a)^*]^*$. Follow the rules of construction from the lecture. (This means including all the λ-moves, even if the λ-NFA could be simplified by removing them.)

I got eleven states: $(1, \lambda, 2)$, $(2, \lambda, 3)$, $(2, \lambda, 7)$,$(2, \lambda, 10)$, $(3, \lambda, 4)$, $(3, a, 4)$, $(3, b, 4)$, $(4, \lambda, 3$, $(4, \lambda, 5)$, $(5, a, 6)$, $(6, b, 10)$, $(7, \lambda, 9)$, $(7, b, 8)$, $(7, a, 9)$, $(8, a, 9)$, $(9, \lambda, 7)$, $(9, \lambda, 10)$, $(10, \lambda, 2)$, $(10, \lambda, 11)$. *State* 1 *is the start state, and state* 11 *is the only final state.*

Question 5 (15): Apply the State Elimination Algorithm to find a regular expression for the DFA given in the diagram below.



We first add a new state i *with an edge* $< i, \lambda, p >$, *and a new final state* f *with edges* $< r, \lambda, f >$ *and* $< s, \lambda, f >$.

Eliminating p *gives us* $< i, a, q >$, $< i, b, r >$, *and* $< r, ab, r >$.

Eliminating s *gives us* $< r, b, f >$ *(which is merged to make* $< r, \lambda{+}b, f >$*),* $< r, ba, q >$ *(which is mergeed to make* $< r, aa + ba, q >$*),* $< q, ba, q >$ *(which is merged to make* $< q, a + ba, q >$*) and* $< q, b, f >$.

Eliminating q *gives us* $< i, a(a + ba)^*b, f >$ *and* $< r, (aa + ba)(a + ba)^*b, f >$ *(which merges to make* $< r, \lambda + b + (aa + ba)(a + ba)^*b, f >$*).*

Eliminating r *gives us the final regular expression of*

$$a(a + ba)^*b + b(ab)^*(\lambda + b + (aa + ba)(a + ba)^*b$$

.

Question 6 (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. Some of them refer to the scenarios of the other problems, and/or the entities defined on the supplemental sheet.

- (a) The regular expressions $(a+b)^*$ and $a(a^*+b^*)+b(a^*+b^*)$ do not denote the same language.

  TRUE. The second one does not cover λ.

- (b) It is not the case that there are some languages for which we can design an NFA but we cannot design a DFA.

  TRUE. The Subset Construction lets us convert any NFA to an equivalent DFA.

- (c) Let G be a directed graph, let s be a node in G, and set every edge weight equal to 1. Then a Uniform Cost Search from s will act the same as applying BFS from s.

  TRUE. The priority queue from the UCS acts just like the ordinary queue in the BFS.

- (d) In Question 1(c), the premises used there do not give you enough information to determine all the truth values of the predicate X.

  TRUE. Everything about the Opossum is known, but for example we don't know whether Blaze saw a Squirrel in those places where she did not see a Rabbit.

- (e) Consider a one-tape Turing machine M with tape alphabet $\{a, b, \Box\}$, transition function δ, and states including p and q. If the current configuration of M is $\Box abqa\Box b$, and $\delta(q, a) = \{p, \Box, L\}$, then the next configuration of M is $\Box apb\Box\Box a$.

  FALSE. The machine is supposed to overwrite the "a", move left, and enter state p. It does so, but the "b" on the right has changed to an "a" for no good reason.

- (f) If L and L' are both Turing-recognizable languages, then it could be that $L \cap L'$ is not also Turing recognizable.

  FALSE. If we run recognizers for both L and L' on our input string w, either in series or in parallel, we wait until or unless both accept. In that case we accept w. If one rejects, we can reject, but otherwise we run forever, which is ok because the input is not in $L \cap L'$.

- (g) If X and Y are finite sets of the same size, and f is a function from X to Y, then f is one-to-one if and only if it is onto.

  TRUE. Both happen if and only if f is a bijection.

- (h) If m and n are any two positive naturals, then there exists a natural t such that any natural p with $p \geq t$ can be written as $mx + ny$, where x and y are naturals.

  FALSE. This would be true if m and n were relatively prime, but we don't have that assumption here.

- (i) Consider any Hasse diagram as an undirected graph. Then it may have a cycle.

  TRUE. There are no directed cycles in its role as a directed graph, but the division relation on 1, 2, 3, 6, for example, is a four-node cycle as an undirected graph.

- (j) Let G be a weighted undirected graph, with positive edge weights, and let s and t be nodes of G. Let x be the shortest path from s to t found using uniform-cost search. Let y be the result of an A^* search with s as start node and t as goal node, using a consistent and admissible heuristic function. Then y cannot be strictly smaller than x, no matter what the heuristic.

  TRUE. Both the result of a UCS search and that of an A^* search give the true shortest from s to t, so they are always the same.