NAME: ____________________________

SPIRE ID: ____________________________

COMPSCI 250
Introduction to Computation
Final Exam Fall 2024 – Solution Key

D. A. M. Barrington and M. Golin 12 December 2024

DIRECTIONS:

- Answer the problems on the exam pages.
- There are **five** problems on pages **2-13**, some with multiple parts, for 125 total points plus 10 extra credit. Final scale will be determined after the exam.
- Page **14** 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.
- But, if you do write a solution on the back, you must explicitly have a note on the front stating that you used the back 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.
- Your answers must be LEGIBLE, and not cramped. Write only short paragraphs with space between paragraphs.

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

Family Name: ______________________

Question 1 (Dog Proof, 35 total points) This fall, Dave's family (Blaze, Dave, Jessica, and Rhonda) had a number of notable encounters with wildlife. Let M be the set $\{b, d, j, r\}$ of the four Members in the family, and let A be the set $\{B, D, F, O, W\}$ of the Animals (Bear, Deer, Fox, Opossum, and Wolverine) that they may have observed. Let $S(x, y)$ be the predicate meaning "Member x saw Animal y". Finally, H is the subset $\{d, j\}$ of the Members that are Human, and $H(x)$ is the corresponding unary predicate on the set M meaning "Member x is Human".

(A, 10) Translations: Translate each of these statements as indicated (two points each).

- **Statement I** (to English): $(S(b, F) \vee S(d, B)) \rightarrow (\neg S(d, B) \wedge S(d, O))$

  Solution: If Blaze saw a Fox or Dave saw a Bear, then Dave saw an Opossum but not a Bear.

- **Statement II** (to symbols): Dave saw an Opossum, and either Blaze saw a Fox or Dave saw a Bear, but not both.

  Solution: $(d, O) \wedge (s(b, F) \oplus s(d, B))$.

- **Statement III** (to English): $\forall a : (\exists u : S(u, a)) \rightarrow (\exists v : H(v) \wedge S(v, a))$

  Solution: For any Animal, if any Member saw that Animal, then some Human also saw that Animal.
  Fortunately, 73% of you got this one right, which contributed to the high scores on the predicate proof.

- **Statement IV** (to symbols): Blaze and Jessica each saw exactly one animal.

  Solution: $\exists u : \exists v : \forall z : (S(b, z) \leftrightarrow (z = u) \wedge (S(j, z) \leftrightarrow (z = v))$.
  This was the hardest of the translations for you, with only 18% of you getting it entirely right (and only 12% getting a perfect 10/10 on this question). A lot of this was sloppy reading, ignoring the word "exactly" which meant that Blaze and Jessica each saw one Animal, which might or might not be the same Animal, and neither saw more than one.

- **Statement V** (to English): $\forall x : \exists y : \exists z : (y \neq z) \wedge \neg S(x, y) \wedge \neg S(x, z)$

  Solution: For every Member x, there exist two different Animals that x did not see.
  Another correct, and useful, equivalent translation was "no Member saw more than three Animals".

(B, 10) Boolean Proof: Using only Statements I and II, let $p = S(b, F)$, $q = S(d, B)$, and $r = S(d, O)$. Prove that p and r are true, and that q is false. You may use either a truth table or deductive rules. If you use deductive rules, remember that you must prove *both* that this solution agrees with the statements, and that no other other solution agrees with the statements.

Solution: We can represent Statement I as $(p \vee q) \rightarrow (\neg q \wedge r)$, and Statement II as $r \wedge (p \oplus q)$. If q were true, Statement I would imply $\neg q$, a contradiction. So q must be false. Substituting 0 for q in Statement II, and simplifying $p \oplus 0$ to p, we get $r \wedge p$, telling us that p and r are false.

This proves that the *only* way that Statements I and II can both be true is if p and r are true and q is false. We still have to verify that this setting of the variables makes the statements true. But in this case, the conclusion of I is true, so I is true trivially. And II is true because r is true and $p \oplus q$ is true.

Here is a truth table:

$((p$	$\vee$	$q)$	$\rightarrow$	$(\neg$	q	$\wedge$	$r))$	$\wedge$	$(r$	$\wedge$	$(p$	$\oplus$	$q))$
0	0	0	1	1	0	0	0	**0**	0	0	0	0	0
0	0	0	1	1	0	1	1	**0**	1	0	0	0	0
0	1	1	0	0	1	0	0	**0**	0	0	0	1	1
0	1	1	0	0	1	0	1	**0**	1	1	0	1	1
1	1	0	0	1	0	0	0	**0**	0	0	1	1	0
1	1	0	1	1	0	1	1	**1**	1	1	1	1	0
1	1	1	0	0	1	0	0	**0**	0	0	1	0	1
1	1	1	0	0	1	0	1	**0**	1	0	1	0	1

These went generally pretty well, with an average score of 8.2/10. There were 18% with a correct truth table, 19% with an entirely correct deductive proof, and 32% with an otherwise correct deductive proof but no check for the validity of your solution. No matter how many times we tell you, many of you don't understand that a deductive proof ("the premises imply these truth values") only proves that no other *solution apart from yours can be correct. It doesn't say that your solution* does *agree with the premises, so you have to prove that directly by checking each premise.*

(C, 15) Predicate Proof: In this problem you must determine all twenty boolean values of the predicate $S(x, y)$ for every member x and every animal y. You may use Statements I, II, III, IV, and V, as well as:

Statement VI: Rhonda saw a Bear, a Deer, and a Fox.

Do this by filling out this table, with a "0" for "false" and a "1" for "true".

$S(b, B) =$	$S(d, B) =$ **0**	$S(j, B) =$	$S(r, B) =$
$S(b, D) =$	$S(d, D) =$	$S(j, D) =$	$S(r, D) =$
$S(b, F) =$ **1**	$S(d, F) =$	$S(j, F) =$	$S(r, F) =$
$S(b, O) =$	$S(d, O) =$ **1**	$S(j, O) =$	$S(r, O) =$
$S(b, W) =$	$S(d, W) =$	$S(j, W) =$	$S(r, W) =$

You must **justify** each of the twenty answers in this table, except for the three that have been filled for you. (These are the answers from Part (B).) Many of these answers have very short justifications, but some are more involved and may require quoting other answers you have already derived. If your answer involves the use of a quantifier proof rule, and many of them should, make clear which rule you are using and when. Using complete sentences will generally make your answers more readable.

Solution to Predicate Proof

$S(b, B) = 0$	$S(d, B) =$ **0**	$S(j, B) = 1$	$S(r, B) = 1$
$S(b, D) = 0$	$S(d, D) = 1$	$S(j, D) = 0$	$S(r, D) = 1$
$S(b, F) =$ **1**	$S(d, F) = 1$	$S(j, F) = 0$	$S(r, F) = 1$
$S(b, O) = 0$	$S(d, O) =$ **1**	$S(j, O) = 0$	$S(r, O) = 0$
$S(b, W) = 0$	$S(d, W) = 0$	$S(j, W) = 0$	$S(r, W) = 0$

We are given that Blaze saw a Fox, and part of Statement IV says that Blaze saw no other animals. Thus Blaze did not see a Bear, Deer, Opossum or Wolverine.

Statement VI says that Rhonda saw a Bear, Deer, and Fox. Specializing Statement V to Rhonda, there must be two Animals that Rhonda did not see. These must be the two remaining unknown values, the Opossum and Wolverine.

Specializing Statement III to the Bear, we know that if any Member saw the Bear, some Human must have seen it. But Rhonda did see the Bear, and Dave did not. The only Human who could have seen the Bear was Jessica, so she saw it. Then, from Statement IV, we know that Jessica did not see any Animal other than the Bear – she did not see the Deer, Fox, Opossum, or Wolverine.

We know that Dave saw the Opossum but not the Bear. Specializing Statement III to the Deer, we know that Rhonda saw it and Jessica did not, so since a Human must have seen it, Dave must have seen it. Similarly, if we specialize Statement III to the Fox, since Rhonda saw it and Jessica did not, Dave must have seen it. Finally, Specializing Statement V to Dave, there must be two different Animals he did not see. One was the Bear, and the other must have been the Wolverine.

No one saw a Wolverine, which stands to reason as Wolverines are not native to Massachusetts. (Though you would not get many points for that justification – we might have visited a zoo or something. Someone claimed that Wolverines are mythical, which is not true, but someone else confused them with Werewolves, which are.)

I was pretty happy with these. The majority (60%) of the solutions were correct, and most of those were clearly written and pretty easy to verify. We can both thank Mordecai who proposed laying out the table for the answers. Note that the grading was not just a matter of counting your correct boolean answers! Your task was to prove *that all of your twenty answers were correct, and you were graded on how well you did that overall.*

Family Name: ____________________

Question 2 (A) (10): (Induction 1)

For $n \geq 1$, set $S(n) = \sum_{i=1}^{n}(4i-1)$.

Note that $S(3) = 3 + 7 + 11 = 21 = 3 \cdot 7 = 3 \cdot (2 \cdot 3 + 1)$ and

$S(4) = 3 + 7 + 11 + 15 = 36 = 4 \cdot 9 = 4 \cdot (2 \cdot 4 + 1)$.

Prove by induction that, for all positive naturals n, $S(n) = n(2n+1)$.

(i) First write your induction hypothesis in the box below. This should be in the form $P(x)$, where you *must* explicitly explain what x is and write an unambiguous statement of $P(x)$.

Solution: $P(n)$ is the statement: $S(n) = n(2n+1)$.
It is defined for all positive naturals n.

(ii) Next, write your base case(s) in the box below.

Solution: The base case is $P(1)$.
This is correct because $S(1) = 3 = 1 \cdot (2 \cdot 1 + 1)$.

(iii) Finally, provide your induction step. This step will be marked on how clear and mathematically precise your proof is. Ambiguous explanations will have points deducted.
Be sure to clearly describe your induction goal. If you run out of space, continue the proof on the back page (with a note stating that you are writing on the back).

Solution:

Assume $P(n)$ is true.

The goal is to prove $P(n+1)$, i.e., that

$$S(n+1) = (n+1)(2(n+1)+1) = (n+1)(2n+3) = 2n^2 + 5n + 3.$$

This will follow from the Induction Hypothesis $P(n)$, as follows:

$$\begin{aligned}
S(n+1) &= \sum_{i=1}^{n+1}(4i-1) \\
&= \left(\sum_{i=1}^{n}(4i-1)\right) + (4(n+1)-1) && \text{definition of } S(n+1) \\
&= n(2n+1) + (4(n+1)-1) && \text{IH} \\
&= (2n^2+n) + (4n+3) && \text{algebra} \\
&= 2n^2 + 5n + 3.
\end{aligned}$$

Marking Note Q2A(1): For full credit, $P(n)$ had to be a predicate that evaluated to a boolean. Setting $P(n) = \sum_{i=1}^{n}(4(i-1)$ was an error.

Marking Note Q2A(2): Problem stated that it was only for positive naturals. So, the base case must be $n = 1$, not $n = 0$.

Marking Note Q2A(3): As explained in the induction marking/study guide sent out before the exam, for full credit it was necessary to *explicitly* show where in the proof the Induction Hypothesis was being used. A point was deducted for just using it without identifying that it was the IH.

Marking Note Q2A(4): As explained in the problem instructions, it was necessary to clearly describe the induction goal. In particular, proofs that used a LHS, RHS format and then showed that LHS=RHS, might have had a point deducted if they did not explicitly identify in the induction step which of the LHS or RHS was the goal, and which was the proof. This depended upon the clarity of the exposition.

Marking Note Q2A(5): As noted, the marking was based on the clarity and precision of the proof.

Proofs that did not show details had points deducted.
For example, many lines of algebra being skipped with the explanation "this simplifies to the goal", might have had points deducted based on how non-obvious the simplification was. NOT on the correctness of the simplification.

Obviously, the IH will imply the IG. So it would always be truthful to write, "and now we simplify to get the goal". The proof has to show clearly how to get from the first to the second.

Family Name: ____________________

Question 2 (B) (10): (Induction Extra Credit)

For $n \geq 0$, set $S(n) = \sum_{i=0}^{n} i2^i$.

Note that $S(1) = 1{\cdot}2^1 = 2 = 2(1{\cdot}2^1{-}2^1{+}1)$ and $S(2) = 1{\cdot}2^1{+}2{\cdot}2^2 = 10 = 2(2{\cdot}2^2{-}2^2{+}1)$.

Prove by induction that, for all naturals n, $S(n) = 2\left(n2^n - 2^n + 1\right)$.

(i) First write your induction hypothesis in the box below. This should be in the form $P(x)$, where you *must* explicitly explain what x is and write an unambiguous statement of $P(x)$.

Solution: $P(n)$ is the statement: $S(n) = 2\left(n2^n - 2^n + 1\right)$.
It is defined for all naturals n.

(ii) Next, write your base case(s) in the box below.

Solution: The base case is $P(0)$.
This is correct because $S(0) = 0 = 2 \cdot (0 \cdot 2^0 - 2^0 + 1)$.

(iii) Finally, provide your induction step. This step will be marked on how clear and mathematically precise your proof is. Ambiguous explanations will have points deducted.
Be sure to clearly describe your induction goal. If you run out of space, continue the proof on the back page (with a note stating that you are writing on the back).

Solution:

Assume $P(n)$ is true.

The goal is to prove $P(n+1)$, i.e., that

$$S(n+1) = 2\left((n+1)2^{n+1} - 2^{n+1} + 1\right) = 2n2^{n+1} + 2.$$

This will follow from the Induction Hypothesis $P(n)$, as follows:

$$\begin{aligned}
S(n+1) &= \sum_{i=0}^{n+1} i2^i \\
&= \left(\sum_{i=0}^{n} i2^i\right) + (n+1)2^{n+1} && \text{definition of } S(n+1) \\
&= 2\left(n2^n - 2^n + 1\right) + (n+1)2^{n+1} && \text{IH} \\
&= n2^{n+1} - 2^{n+1} + 2 + n2^{n+1} + 2^{n+1} && \text{algebra} \\
&= 2n2^{n+1} + 2.
\end{aligned}$$

Marking Note Q2B(1): For full credit, $P(n)$ had to be a predicate that evaluated to a boolean. Setting $P(n) = \sum_{i=0}^{n} i2^i$ was an error.

Marking Note Q2B(2): Problem stated that it was only for positive naturals. So, the base case must be $n = 0$, not $n = 1$.

Marking Note Q2B(3): As explained in the induction marking/study guide sent out before the exam, for full credit it was necessary to *explicitly* show where in the proof the Induction Hypothesis was being used. A point was deducted for just using it without identifying that it was the IH.

Marking Note Q2B(4): As explained in the problem instructions, it was necessary to clearly describe the induction goal. In particular, proofs that used a LHS, RHS format and then showed that LHS=RHS might have had a point deducted if they did not explicitly identify in the induction step which of the LHS or RHS was the goal, and which was the proof. This depended upon the clarity of the exposition.

Marking Note Q2B(5): As noted, the marking was based on the clarity and precision of the proof.

Proofs that did not show details had points deducted.
For example, many lines of algebra being skipped with the explanation "this simplifies to the goal", might have had points deducted based on how non-obvious the simplification was. NOT on the correctness of the simplification.

Obviously, the IH will imply the IG. So it would always be truthful to write, "and now we simplify to get the goal". The proof has to show clearly how to get from the first to the second.

Family Name: ____________________

Question 3 (20 points): (Induction 2) Define the language $\mathbf{B}$ on $\Sigma = \{a, b, c\}$ as follows.

String $\mathbf{w} \in \Sigma^*$ is in $\mathbf{B}$ if

R1: $\mathbf{w} = \lambda$ (the empty string) or

R2: $\mathbf{w} = a\mathbf{v}b$ where $\mathbf{v} \in \mathbf{B}$ or

R3: $\mathbf{w} = a\mathbf{u}b\mathbf{v}c$ where $\mathbf{u} \in \mathbf{B}$ and $\mathbf{v} \in \mathbf{B}$.

No other strings are in $\mathbf{B}$.

Define $N_a(\mathbf{w})$, $N_b(\mathbf{w})$, and $N_c(\mathbf{w})$, to be, respectively, the number of a's, b's and c's in $\mathbf{w}$.

Also define $|\mathbf{w}| = N_a(\mathbf{w}) + N_b(\mathbf{w}) + N_c(\mathbf{w})$ to be the total number of characters in $\mathbf{w}$.

Examples: Let $\mathbf{w}_1 = ab$, $\mathbf{w}_2 = abc$, $\mathbf{w}_3 = aaabbbabc$, $\mathbf{w}_4 = aba$, $\mathbf{w}_5 = abcc$.
Then $\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3 \in \mathbf{B}$, $\mathbf{w}_4, \mathbf{w}_5 \notin \mathbf{B}$

$N_a(\mathbf{w}_1) = N_b(\mathbf{w}_1) = 1, N_c(\mathbf{w}_1) = 0.$ $\quad N_a(\mathbf{w}_2) = N_b(\mathbf{w}_2) = N_c(\mathbf{w}_2) = 1.$

$N_a(\mathbf{w}_3) = N_b(\mathbf{w}_3) = 4$, $N_c(\mathbf{w}_3) = 1$.

A *derivation* that $\mathbf{w} \in \mathbf{B}$ is a listing of the rules that show that $\mathbf{w} \in \mathbf{B}$, one item per line. Each line in the derivation should justify the creation of a specific string in $\mathbf{B}$. The final line in the derivation should be the justification of $\mathbf{w}$.

Here is a derivation that $aaabbabcb \in \mathbf{B}$.
1. $\mathbf{v_1} = \lambda \in \mathbf{B}$. (R1)
2. $\mathbf{v_2} = ab = a\mathbf{v_1}b \in \mathbf{B}$. (R2 and line 1)
3. $\mathbf{v_3} = a\,ab\,b\,ab\,c = a\mathbf{v_2}b\mathbf{v_2}c \in \mathbf{B}$. (R3 and line 2)
4. $\mathbf{w} = a\,aabbabc\,b = a\mathbf{v_3}b \in \mathbf{B}$. (R2 and line 3)

Part (A) of this problem is below. Parts (B) and (C) are on the following pages. When writing their solutions, you must use the the mathematical notation we provided above.

If your proofs of (B) and (C) have multiple pieces, place each piece in a separate paragraph with space between the paragraphs. Be sure to clearly describe your induction goal.

If you need more space, continue your proof on the back of the page, noting explicitly that you are continuing on the back.

(A) (2 pts) Give a derivation that shows that the string $\mathbf{w} = aabcbaabbc$ is in $\mathbf{B}$.
Write your answers on the lines below (you shouldn't need all of the lines.).

Solution:

1. $\mathbf{v_1} = \lambda \in \mathbf{B}$. (R1)
2. $\mathbf{v_2} = ab = a\mathbf{v_1}b \in \mathbf{B}$. (R2 and line 1)
3. $\mathbf{v_3} = abc = a\mathbf{v_1}b\mathbf{v_1}c \in \mathbf{B}$. (R3 and line 1)
4. $\mathbf{v_4} = a\,ab\,b = a\mathbf{v_2}b \in \mathbf{B}$. (R2 and line 2)
5. $\mathbf{w} = a\,abc\,b\,aabb\,c = a\mathbf{v_3}b\mathbf{v_4}c \in \mathbf{B}$. (R3 and lines 3 and 4)

(B) 9 points. Prove by induction that, for every $\mathbf{w} \in B$, $N_a(\mathbf{w}) = N_b(\mathbf{w})$.

a) First write your induction hypothesis in the box below. This should be in the form $P(x)$, where you *must* explicitly explain what x is and write an unambiguous statement of $P(x)$.

(b) Next, write your base case(s) in the box below.

(c) Finally, provide your induction step. This step will be marked on how clear and mathematically precise your proof is. Ambiguous explanations or explanations missing details will have points deducted.

Solution for Induction 3 (B): Proof by structural induction

Prove by induction that, for every $\mathbf{w} \in B$, $N_a(\mathbf{w}) = N_b(\mathbf{w})$.

(i) $P(\mathbf{w})$ *is the statement:* $N_a(\mathbf{w}) = N_b(\mathbf{w})$.
It is defined over $\mathbf{w} \in \mathbf{B}$.

(ii) The base case is $\mathbf{w} = \lambda$ *(the empty string).*
$P(\lambda)$ *is true because* $N_a(\lambda) = 0 = N_b(\lambda)$.

(iii) The proof will be by structural induction.

Let $\mathbf{w} \in \mathbf{B}$ *where* $\mathbf{w} \neq \lambda$. *Then* $\mathbf{w}$ *must be the immediate consequence of either R2 or R3.*

- *If R2. Then* $\mathbf{w} = a\mathbf{v}b$ *where* $\mathbf{v} \in \mathbf{B}$. *By the induction hypothesis,* $N_a(\mathbf{v}) = N_b(\mathbf{v})$. *So,*

$$\begin{aligned} N_a(\mathbf{w}) &= N_a(\mathbf{v}) + 1 && \textit{By construction of } \mathbf{w} \\ &= N_b(\mathbf{v}) + 1 && \textit{By Induction Hypothesis} \\ &= N_b(\mathbf{w}) && \textit{By construction of } \mathbf{w}. \end{aligned}$$

- *If R3. Then* $\mathbf{w} = a\mathbf{u}b\mathbf{v}c$ *where* $\mathbf{u} \in \mathbf{B}$ *and* $\mathbf{v} \in \mathbf{B}$.
  By the induction hypothesis, $N_a(\mathbf{u}) = N_b(\mathbf{u})$ *and* $N_a(\mathbf{v}) = N_b(\mathbf{v})$. *So,*

$$\begin{aligned} N_a(\mathbf{w}) &= N_a(\mathbf{u}) + N_a(\mathbf{v}) + 1 && \textit{By construction of } \mathbf{w} \\ &= N_b(\mathbf{u}) + N_b(\mathbf{v}) + 1 && \textit{By Induction Hypothesis} \\ &= N_b(\mathbf{w}) && \textit{By construction of } \mathbf{w}. \end{aligned}$$

Since we have shown $P(\mathbf{w})$ *for both possible ways of constructing* $\mathbf{w}$, $P(\mathbf{w})$ *is always true.*

Solution for Induction 3 (B): Proof by induction on length.

Prove by induction that, for every $\mathbf{w} \in B$, $N_a(\mathbf{w}) = N_b(\mathbf{w})$.

Unlike the previous proof, this proof will not use structural induction but, instead be induction on the lengths of strings.

(i) $P(n)$ is the statement: For every $\mathbf{w} \in \mathbf{B}$ satisfying $|\mathbf{w}| \le n$, $N_a(\mathbf{w}) = N_b(\mathbf{w})$.
$P(n)$ is defined over all natural n.

(ii) The base case is $n = 0$.
The only string $\mathbf{w} \in \mathbf{B}$ satisfying $|\mathbf{w}| = 0$ is $\mathbf{w} = \lambda$ (the empty string).
$P(0)$ is true because $N_a(\lambda) = 0 = N_b(\lambda)$.

(iii) The proof will be by induction on naturals n. Assume $P(n)$ is true. To show that $P(n+1)$ is true, we must show that $N_a(\mathbf{w}) = N_b(\mathbf{w})$ for all $\mathbf{w} \in \mathbf{B}$ satisfying $|\mathbf{w}| = n+1$.

Let $\mathbf{w} \in \mathbf{B}$ where $\mathbf{w} \neq \lambda$. Then $\mathbf{w}$ must be the immediate consequence of either R2 or R3.

- If R2. Then $\mathbf{w} = a\mathbf{v}b$ where $\mathbf{v} \in \mathbf{B}$.
  Since $|\mathbf{v}| = |\mathbf{w}| - 2 = n - 1 < n$, by the induction hypothesis, $N_a(\mathbf{v}) = N_b(\mathbf{v})$. So,

$$\begin{aligned} N_a(\mathbf{w}) &= N_a(\mathbf{v}) + 1 && \text{By construction of } \mathbf{w} \\ &= N_b(\mathbf{v}) + 1 && \text{By Induction Hypothesis} \\ &= N_b(\mathbf{w}) && \text{By construction of } \mathbf{w}. \end{aligned}$$

- If R3. Then $\mathbf{w} = a\mathbf{u}b\mathbf{v}c$ where $\mathbf{u} \in \mathbf{B}$ and $\mathbf{v} \in \mathbf{B}$.
  Note that by, the construction, $|\mathbf{u}| = |\mathbf{w}| - 3 = n - 2 < n$ and $|\mathbf{v}| = |\mathbf{w}| - 3 = n - 2 < n$.
  Thus, by the induction hypothesis, $N_a(\mathbf{u}) = N_b(\mathbf{u})$ and $N_a(\mathbf{v}) = N_b(\mathbf{v})$. So,

$$\begin{aligned} N_a(\mathbf{w}) &= N_a(\mathbf{u}) + N_a(\mathbf{v}) + 1 && \text{By construction of } \mathbf{w} \\ &= N_b(\mathbf{u}) + N_b(\mathbf{v}) + 1 && \text{By the Induction Hypothesis} \\ &= N_b(\mathbf{w}) && \text{By construction of } \mathbf{w}. \end{aligned}$$

So $N_a(\mathbf{w}) = N_b(\mathbf{w})$, no matter whether $\mathbf{w}$ was constructed using R_2 or R_3.

Since we have shown $N_a(\mathbf{w}) = N_b(\mathbf{w})$ for all $\mathbf{w} \in \mathbf{B}$ satisfying $|\mathbf{w}| = n+1$, $P(n+1)$ is true.

(C) 9 points. Prove by induction that, for every $\mathbf{w} \in B$, $N_c(\mathbf{w}) \leq N_a(\mathbf{w})$.

a) First write your induction hypothesis in the box below. This should be in the form $P(x)$, where you *must* explicitly explain what x is and write an unambiguous statement of $P(x)$.

(b) Next, write your base case(s) in the box below.

(c) Finally, provide your induction step. This step will be marked on how clear and mathematically precise your proof is. Ambiguous explanations or explanations missing details will have points deducted.

Solution for Induction 3 (C): Proof by structural induction.

Prove by induction that, for every $\mathbf{w} \in B$, $N_c(\mathbf{w}) \leq N_a(\mathbf{w})$.

(i) $P(\mathbf{w})$ *is the statement:* $N_c(\mathbf{w}) \leq N_a(\mathbf{w})$. *It is defined over* $\mathbf{w} \in \mathbf{B}$.

(ii) The base case is $\mathbf{w} = \lambda$ *(the empty string).* $P(\lambda)$ *is true because* $N_c(\lambda) = 0 = N_a(\lambda)$.

(iii) The proof will be by structural induction.

Let $\mathbf{w} \in \mathbf{B}$ *where* $\mathbf{w} \neq \lambda$. *Then* $\mathbf{w}$ *must be the immediate consequence of either R2 or R3.*

- *If R2. Then* $\mathbf{w} = a\mathbf{v}b$ *where* $\mathbf{v} \in \mathbf{B}$. *By the induction hypothesis,* $N_c(\mathbf{v}) \leq N_a(\mathbf{v})$. *So,*

$$\begin{aligned} N_c(\mathbf{w}) &= N_c(\mathbf{v}) && \textit{By construction of } \mathbf{w} \\ &\leq N_a(\mathbf{v}) && \textit{By Induction Hypothesis} \\ &\leq N_a(\mathbf{v}) + 1 \\ &= N_a(\mathbf{w}) && \textit{By construction of } \mathbf{w}. \end{aligned}$$

- *If R3. Then* $\mathbf{w} = a\mathbf{u}b\mathbf{v}c$ *where* $\mathbf{u} \in \mathbf{B}$ *and* $\mathbf{v} \in \mathbf{B}$.
  By the induction hypothesis, $N_c(\mathbf{u}) \leq N_a(\mathbf{u})$ *and* $N_c(\mathbf{v}) = N_a(\mathbf{v})$. *So,*

$$\begin{aligned} N_c(\mathbf{w}) &= N_c(\mathbf{u}) + N_c(\mathbf{v}) + 1 && \textit{By construction of } \mathbf{w} \\ &\leq N_a(\mathbf{u}) + N_a(\mathbf{v}) + 1 && \textit{By Induction Hypothesis} \\ &= N_a(\mathbf{w}) && \textit{By construction of } \mathbf{w}. \end{aligned}$$

Since we have shown $P(\mathbf{w})$ *for both possible ways of constructing* $\mathbf{w}$, $P(\mathbf{w})$ *is always true.*

Solution for Induction 3 (C): Proof by induction on length.

Prove by induction that, for every $\mathbf{w} \in B$, $N_c(\mathbf{w}) \leq N_a(\mathbf{w})$.

(i) $P(n)$ is the statement: For every $\mathbf{w} \in \mathbf{B}$ satisfying $|\mathbf{w}| \leq n$, $N_c(\mathbf{w}) = N_a(\mathbf{w})$.
$P(n)$ is defined over all natural n.

(ii)
The base case is $n = 0$.
The only string $\mathbf{w} \in \mathbf{B}$ satisfying $|\mathbf{w}| = 0$ is $\mathbf{w} = \lambda$ (the empty string).
$P(0)$ is true because $N_c(\lambda) = 0 = N_a(\lambda)$.

(iii) The proof will be by induction on naturals n. Assume $P(n)$ is true. To show that $P(n+1)$ is true, we must show that $N_c(\mathbf{w}) = N_a(\mathbf{w})$ for all $\mathbf{w} \in \mathbf{B}$ satisfying $|\mathbf{w}| = n + 1$.

Let $\mathbf{w} \in \mathbf{B}$ where $\mathbf{w} \neq \lambda$. Then $\mathbf{w}$ must be the immediate consequence of either R2 or R3.

- If R2. Then $\mathbf{w} = a\mathbf{v}b$ where $\mathbf{v} \in \mathbf{B}$.
  Since $|\mathbf{v}| = |\mathbf{w}| - 2 = n - 1 < n$, by the induction hypothesis, $N_c(\mathbf{v}) \leq N_a(\mathbf{v})$. So,

$$\begin{aligned} N_c(\mathbf{w}) &= N_c(\mathbf{v}) && \text{By construction of } \mathbf{w} \\ &\leq N_a(\mathbf{v}) && \text{By Induction Hypothesis} \\ &\leq N_a(\mathbf{v}) + 1 \\ &= N_a(\mathbf{w}) && \text{By construction of } \mathbf{w}. \end{aligned}$$

- If R3. Then $\mathbf{w} = a\mathbf{u}b\mathbf{v}c$ where $\mathbf{u} \in \mathbf{B}$ and $\mathbf{v} \in \mathbf{B}$.
  Note that by, the construction, $|\mathbf{u}| = |\mathbf{w}| - 3 = n - 2 < n$ and $|\mathbf{v}| = |\mathbf{w}| - 3 = n - 2 < n$.
  Thus, by the induction hypothesis, $N_c(\mathbf{u}) \leq N_a(\mathbf{u})$ and $N_c(\mathbf{v}) \leq N_a(\mathbf{v})$. So,

$$\begin{aligned} N_c(\mathbf{w}) &= N_c(\mathbf{u}) + N_c(\mathbf{v}) + 1 && \text{By construction of } \mathbf{w} \\ &\leq N_a(\mathbf{u}) + N_a(\mathbf{v}) + 1 && \text{By Induction Hypothesis} \\ &= N_a(\mathbf{w}) && \text{By construction of } \mathbf{w}. \end{aligned}$$

So $N_c(\mathbf{w}) \leq N_a(\mathbf{w})$, no matter whether $\mathbf{w}$ was constructed using R_2 or R_3.

Since we have shown $N_c(\mathbf{w}) \leq N_a(\mathbf{w})$ for all $\mathbf{w} \in \mathbf{B}$ satisfying $|\mathbf{w}| = n + 1$, $P(n + 1)$ is true.

Marking Notes for Question 3

Marking Note 3A1: Derivations needed to follow the format shown in the example. One new string in $\mathbf{B}$ created per line, with an associated justification.

For both problem 3B and 3C, two different induction proofs were given. The first used structural induction. The second used induction on string lengths. Each of those required a different format.

There are no problem dependent marking notes for 3C. It reused the marking notes for 3B .

Marking Note 3B1: Bottom-up issue issue.
The Induction step in structural induction is supposed to be of the form:

Let $w \neq \lambda$, $w \in \mathbf{B}$.
Then either $w = avb$ or $w = aubvc$ where $u, v \in \mathbf{B}$.

Note that in this formulation, very explicitly, the v or u, v are *defined by the* w. This is top-down.

One error is to start with some $x \in \mathbf{B}$, assume by the IH that $P(x)$ and then build w from x.

This is the "bottom-up" error explicitly warned against as being incorrect in the class notes and the distributed marking guide.

An associated error that sometimes was connected with this one was to explicitly require $w = axbxc$ in R3, i.e., to always have $u = v$ in R3. This restriction is not valid.

Marking Note 3B2: Improper IS structure.
As noted above, the Induction step in structural induction is supposed to be of the form:

Let $w \neq \lambda$, $w \in \mathbf{B}$.
Then either $w = avb$ or $w = aubvc$ where $u, v \in \mathbf{B}$.

Note that in this formulation, very explicitly, the v or u, v are *defined by the* w.

This preliminary structure is needed to get the IS started. It can't just start by saying $P(avb)$ or $P(aubvc)$. That has no meaning without the proper introduction introducing what is going to be proven, i.e., $P(w)$.

Marking Note 3B3: $P(x+1)$ error.
This is when structural induction is being used and $x \in \mathbf{B}$ but the IS claims it is proving $P(x+1)$.

This makes no sense since a "string + 1" is not defined.
The class notes and distributed marking guide explicitly warned against this error.

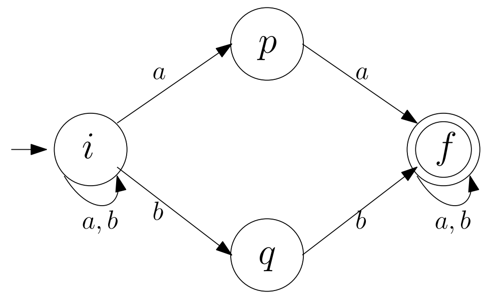
In any case, this is Not how structural induction works.

This issue is sometimes accompanied by the bottom-up issue of Marking Note 3B1.

Question 4 (40 points total):

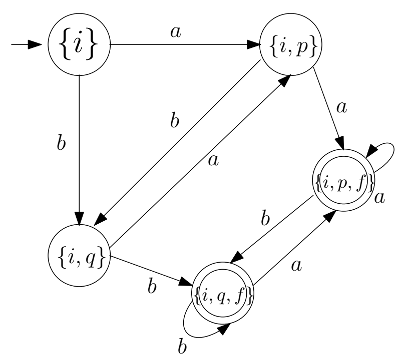
This question involves several of the constructions from Kleene's Theorem. We begin with an ordinary NFA N, with alphabet $\Sigma = \{a, b\}$, state set $\{i, p, q, f\}$, start state i, final state set $\{f\}$, and transition relation

$$\{(i,a,i),(i,b,i),(i,a,p),(i,b,q),(p,a,f),(q,b,f),(f,a,f),(f,b,f)\}.$$



(a, 10) Subset Construction: *Using the Subset Construction*, find a DFA D that is equivalent to the NFA N. It's sufficient to show the new DFA diagram, without further explanation, if it comes from the given construction.

Solution:

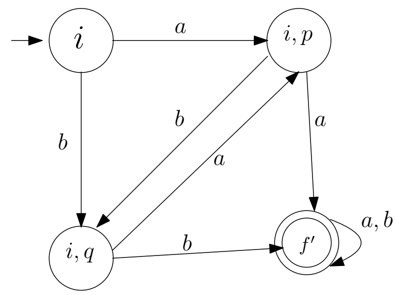


We begin with the non-final start state $\{i\}$. We have an a edge to $\{i, p\}$ and a b-edge to $\{i, q\}$. Non-final state $\{i, p\}$ has an a-edge to $\{i, p, f\}$ and a b-edge to $\{i, q\}$. Non-final state $\{i, q\}$ has an a-edge to $\{i, p\}$ and a b-edge to $\{i, q, f\}$. Final state $\{i, p, f\}$ has an a-edge to itself and a b edge to $\{i, q, f\}$. Final state $\{i, q, f\}$ has an a-edge to $\{i, p, f\}$ and a b-edge to itself. The construction ends with five total states, three non-final and two final.

These were disappointing, with only 41% of you getting full credit and a mean score of 7.07/10. There were three common mistakes: (1) not realizing that there are two final states, and merging them, (2) somehow coming up with a result that is not a DFA, with exactly one arrow for each letter from each state, and (3) getting a path to a death state, which is impossible because any string could remain at the start state. Lots of people made all three mistakes! I took off three points for each, leaving you with two if you did all three. Fortunately, many of the incorrect DFA's could still me minimized in the next part.

(b, 10) Minimization: Find a DFA D' that is minimal and has the same language as your DFA D from part (a). If you do not use the Minimization Construction, prove that your new DFA is minimal and that $L(D) = L(D')$.

Solution



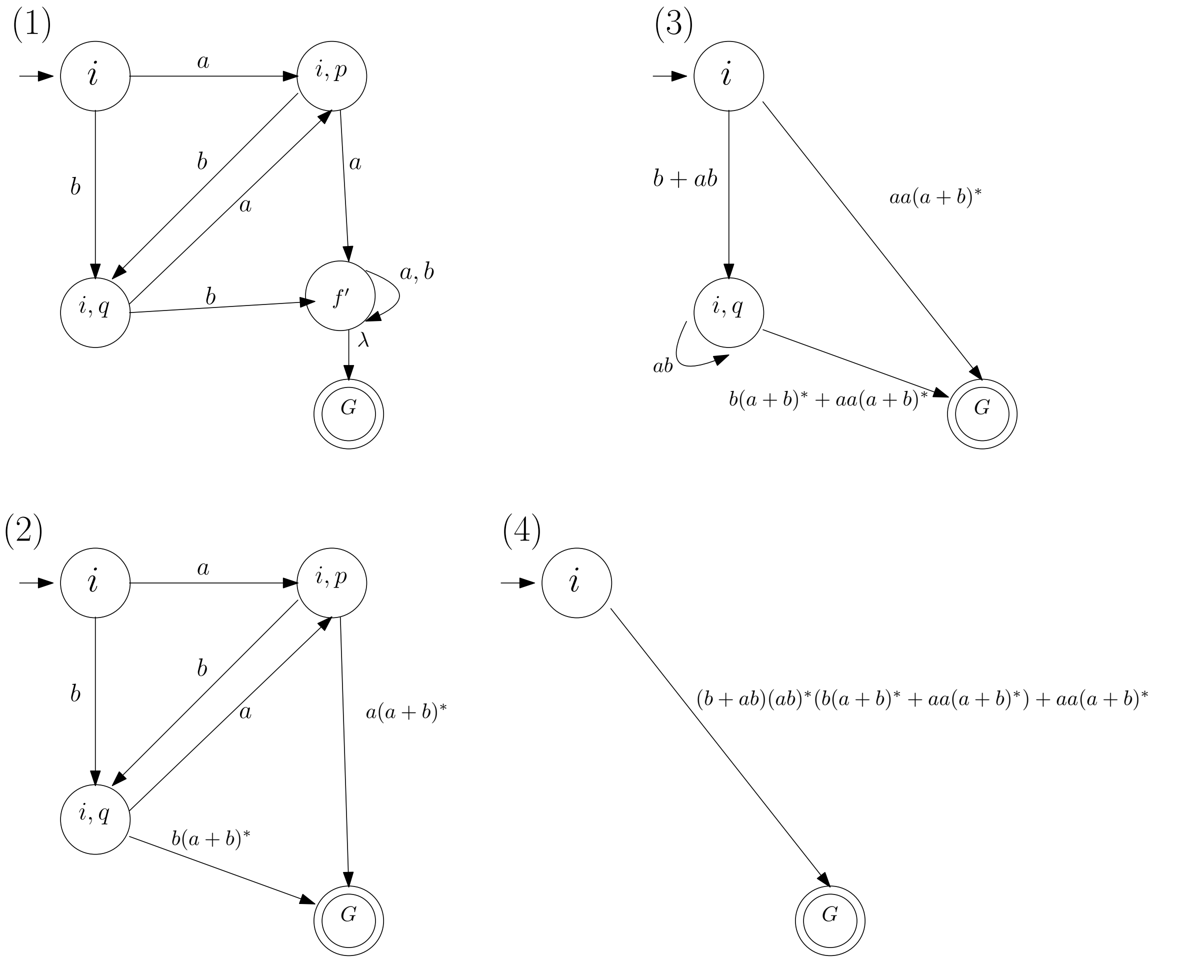
We first partition the states into the non-final states $N = \{i, ip, iq\}$ and the final states $F = \{ipf, iqf\}$. For class N, the behavior of state i is NN, that of state ip is FN, and that of state iq is NF, so this class must be broken up into three classes with one state each. Then we look at class N, and we see that both states have behavior FF. The algorithm ends with four classes, and we form the minimal DFA by merging the two final states into one, giving us four states. The non-final start state is i, with a-edge to ip and b-edge to iq. Non-final state ip has a-edge to F and b-edge to iq. Non-final state iq has a-edge to ip and b-edge to F. Final state F has both edges to itself.

These were also disappointing, with only 36% correct and a mean score of 6.55/10. A very common mistake was to make the entries of the minimization table states *rather than* classes of states *like N or F. Another common problem was re-merging states that had once been separated, or never making clear that they should have been separated.*

Some people got full credit for minimizing a DFA that was incorrect from part (a), and you got 6/10 if you correctly minimized a DFA that was easier than the right one. But if you were trying to minimize an NFA, you probably got 4/10 or less.

(c, 10) State Elimination: Find and justify a regular expression that is equivalent to your DFA D and your minimized D'. If you use State Elimination on either DFA, no further correctness proof is required. If you use another method, prove that your regular expression is equivalent.

Solution to Question 4c



We take the minimized four-state DFA described above. Because state i has no edge into it, we don't need a new start state. We have only one final state, but it has edges out of it (to itself) so we need to add a new final state G with a λ-edge from F to G.

If we kill F first, we create two new edges $(\{i,p\}, a(a+b)^*, G)$ and $(\{i,q\}, b(a+b)^*, G)$.

If we then kill $\{i,p\}$, we create four new edges. We get $(i, ab, \{i,q\})$, which merges with the existing edge to make $(i, b+ab, \{i,q\})$. We also get $(i, aa(a+b)^*, G)$ and $(\{i,q\}, ab, \{i,q\})$, which are new edges, and $(\{i,q\}, aa(a+b)^*, G)$ which merges with an existing edge to make $(\{i,q\}, b(a+b)^* + aa(a+b)^*, G)$.

Now killing the remaining intermediate state $\{i,q\}$ gives us a final regular expression of

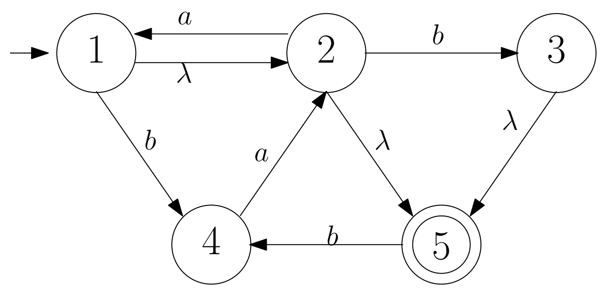
$$aa(a+b)^* + (b+ab)(ab)^*(b(a+b)^* + aa(a+b)^*).$$

There are other possible valid regular expressions depending on the order in which you choose to eliminate states.

These were pretty bad (only 13% fully correct, and a mean score of 4.95/10) and it wasn't in general due to mistakes on earlier parts. The biggest problem, for people who did have a good idea of the construction, was just sloppiness. Huge numbers of people turned an "a, b" arrow into an ab instead of to an $a + b$, or vice versa. This construction is fairly complicated to get your head around, so you really have to be systematic. How many new arrows result from each state elimination, where do they do, and do any of them merge with existing arrows? With a lot of the 4/10 answers, I couldn't point out all the mistakes and just stopped once I was convinced that there were several of them.

A lot of people did State Elimination on the original NFA, rather than on the DFA as you were told to, or just wrote down that regular expression with no justification. In general I gave 6/10 for a correct State Elimination on DFA that was much easier than the one you should have had.

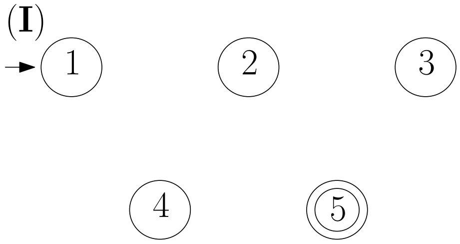
For the last part of this problem, we will work with a *different* language and machine. Here is a new λ-NFA K, which has alphabet $\Sigma = \{a, b\}$, state set $\{1, 2, 3, 4, 5\}$, start state 1, final state set $\{5\}$, and transition relation $(1, \lambda, 2), (1, b, 4), (2, a, 1), (2, b, 3), (2, \lambda, 5), (3, \lambda, 5), (4, a, 2), (5, b, 4)$.



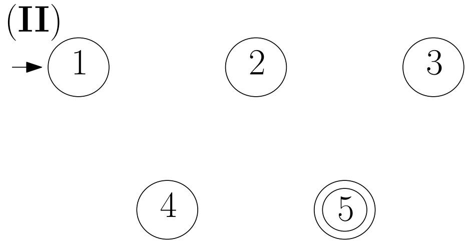
(d, 10) Killing λ-moves: *Using the construction* from the lectures and the textbook, find an ordinary NFA K' that is equivalent to the λ-NFA K above.

- In Graph (I), draw all the edges that are contained in the transitive closure of the λ-edges. Label them with λ.
- In Graph (II), draw all the a-letter moves for K'. Label them as a.
- In Graph (III), draw all the b-letter moves for K'. Label them as b.
- In Graph (IV) draw all the edges in K' properly labeled. Also properly denote the final states of K'.

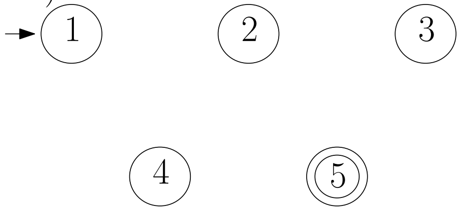
(I)



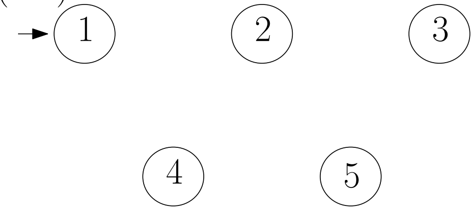
(II)



(III)



(IV)



Solution to Question 4d (Killing λ-Moves):

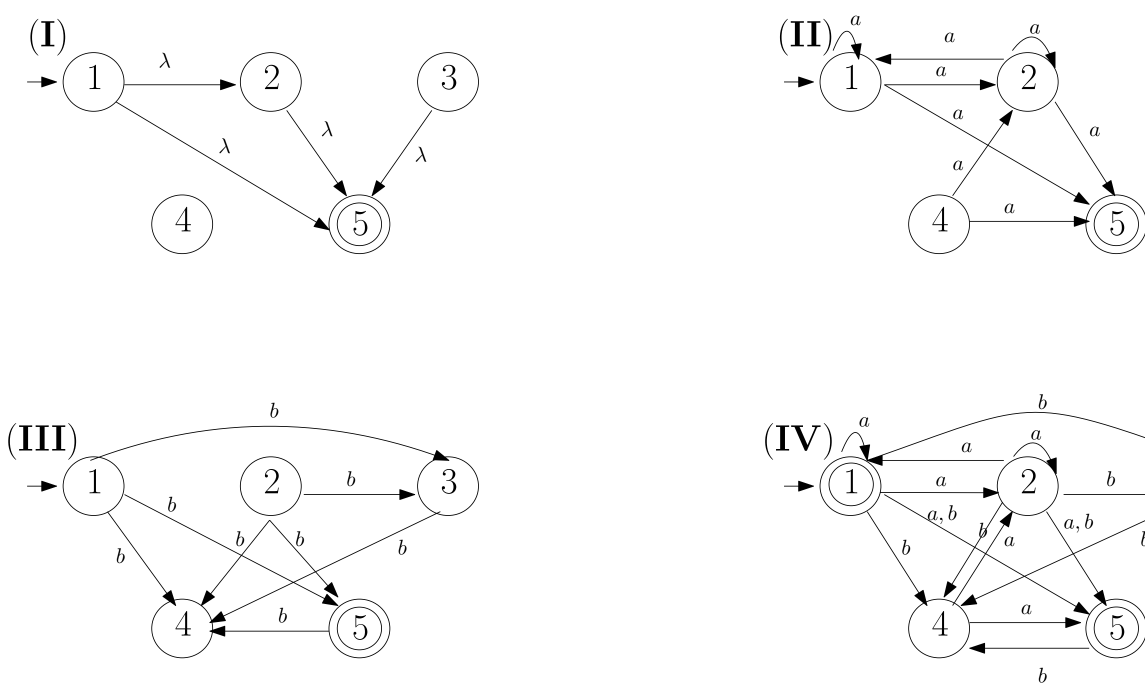
In Graph (I), you should have the existing arrows for $(1, \lambda, 2)$, $(2, \lambda, 5)$, and $(3, \lambda, 5)$, and a new fourth arrow for $(1, \lambda, 5)$ from transitivity.

In Graph (II), the letter move $(2, a, 1)$ causes itself and five other moves, $(1, a, 1)$, $(1, a, 2)$, $(1, a, 5)$, $(2, a, 2)$, and $(2, a, 5)$. The letter move $(4, a, 2)$ causes itself and $(4, a, 5)$. So there are eight a-moves in the new graph.

In Graph (III), the letter move $(1, b, 4)$ causes only itself. The letter move $(2, b, 3)$ causes itself and three other moves: $(1, b, 3)$, $(1, b, 5)$, and $(2, b, 5)$. Finally, the letter move $(5, b, 4)$ causes itself and three other moves, $(1, b, 4)$, $(2, b, 4)$, and $(3, b, 4)$. So there are eight total b-moves, noting that $(1, b, 4)$ was generated twice.

In Graph (IV), then, the sixteen total edges are $(1, a, 1)$, $(1, a, 2)$, $(1, b, 3)$, $(1, b, 4)$, $(1, a, 5)$, $(1, b, 5)$, $(2, a, 1)$, $(2, a, 2)$, $(2, b, 3)$, $(2, b, 4)$, $(2, a, 5)$, $(2, b, 5)$, $(3, b, 4)$, $(4, a, 2)$, $(4, a, 5)$, $(5, b, 4)$.

Also in Graph (IV), the final states should be 1 and 5. We change 1 from non-final to final because there is a λ-path from 1 (the start state) to the final state 5. We should *not* change 2 to a final state. This would not change the language of the machine, but in our proof of the validity of this construction, we assumed that at most one state changed its final status.



The mean score was 6.4/10, and only 7% of you got full credit on this one. Half of you failed to mark 1 as a final state, though that includes a large fraction that either gave up on the question or didn't understand what was being asked on the graphs. As with Q4(c), a lot of you would have benefited from a more systematic approach.

Family Name: ____________________

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.

After reading the questions, write the correct answer, either T (for true) or F (for false), in the corresponding column. Be sure that your "T" and "F" characters are consistent and distinct.

(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	(j)
F	**F**	**T**	**F**	**T**	**F**	**T**	**T**	**T**	**F**

- (a) A function $f : A \to B$ is one-to-one if and only if for every element x of A, there is exactly one element of B mapped from x.

  FALSE. All functions have this property, whether they are one-to-one or not.

- (b) Every positive integer is divisible by some prime number.

  FALSE. This is not true for 1, though it is true for all the others.

- (c) Let G be a strongly connected directed graph with finitely many vertices. Let s be one of the nodes of G. Then both a depth-first search and a breadth-first search will each construct a directed tree rooted at s, in which every node of G will appear in each tree.

  TRUE. Each search will do this for every node that is reachable from s, and since G is strongly connected, there is a path from every node to every other node.

- (d) Let u and v be any two strings over the same alphabet. Then u is a substring of v if and only if u is both a prefix of v and a suffix of v.

  FALSE. This is not true – for example, b is a substring of abc but b is neither a prefix nor a suffix of abc. Two similar statement that are true is that u is a substring of v if and only if it is a suffix of a prefix of v, or if and only if it is a prefix of a suffix of v.

- (e) Let $P(R)$ be a predicate where the variable R ranges over all regular expressions that have alphabet $\Sigma = \{a, b\}$. If we prove $P(\emptyset)$, $P(a)$, $P(b)$, and $\forall R : \forall S : (P(R) \wedge P(S)) \to (P(R{+}S) \wedge P(RS) \wedge P(R^*))$, we may conclude that $P(R)$ is true for all regular expressions over this alphabet.

  TRUE. This is a correct formulation of induction over all regular expressions.

- (f) Let M be a Turing machine with start state i and blank symbol $\Box$. Then if $\delta(i, \Box) = (i, \Box, L)$, the language $L(M)$ of M is $\{\lambda\}$.

  FALSE. On its first move, no matter what the input, the machine will move left and hang. So the language of M is $\emptyset$, not $\{\lambda\}$.

- (g) The two regular expressions $a(a + b)^*b + a(a + b)^*a$ and $aa(a + b)^* + ab(a + b)^*$ represent the same regular language.

  TRUE. Each is the set of all strings of a's and b's with at least two letters and the last letter equal to a.

- (h) Suppose we conduct both a UCS search and an A^* search on the same weighted directed graph. Then it is possible that the two searches find different paths from the start node to the goal node.

  TRUE. They must get the same correct best-path cost, but it is possible for two paths to have the same cost. The heuristic or the tie-breaking procedures could make different choices in the two searches.

- (i) It is possible for an NFA diagram to also represent a DFA.

  TRUE. This happens if the NFA has exactly one arrow from each state for each letter. For any state p and any letter a, we can interpret $\delta(p, a)$ as the unique state q that satisfies $\Delta(p, a, q)$.

- (j) If we apply the Subset Construction to any NFA, the resulting DFA will always have strictly more states than the original NFA had.

  FALSE. If the NFA is already a DFA, the resulting DFA will be identical, and thus it will have the same number of states.