NAME: ____________________________

SPIRE ID: ____________________________

COMPSCI 250
Introduction to Computation
Final Exam Spring 2025

D. A. M. Barrington and M. Golin 14 May 2025

DIRECTIONS:

- Answer the problems on the exam pages.
- There are **five** problems on pages **2-15**, some with multiple parts, for 125 total points plus 10 extra credit. Final scale will be determined after the exam.
- Page **16** 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** add a note on the front stating that you used the back page. Otherwise, we might not see your solution on Gradescope.
- 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+5**
3	**/20**
4	**/40+5**
5	**/20**
Total	**/125+10**

Question 1 (Dog Proof, 35 total points) With the warmer weather, Blaze and Rhonda have spent more time in their backyard, observing various animals and attempting to catch some of them.

Let $D = \{b, r\}$ be the set of dogs $\{\text{Blaze}, \text{Rhonda}\}$.

Let $A = \{\text{Bat}, \text{Cat}, \text{Crow}, \text{Rabbit}, \text{Robin}, \text{Skunk}, \text{Toad}, \text{Vole}\}$ be the set of animals they encountered.

Let F and M be two unary predicates on A such that $F(x)$ means "animal x can fly" and $M(x)$ means "animal x is a mammal". The set of mammals in A is $\{\text{Bat}, \text{Cat}, \text{Rabbit}, \text{Skunk}, \text{Vole}\}$ and the set of flying animals in A is $\{\text{Bat}, \text{Crow}, \text{Robin}\}$.

Finally, $C \subseteq D \times A$ is a binary predicate such that $C(d, a)$ means "dog d caught animal a".

(A, 10) Translations: Translate each statement as indicated.

- **Statement I:** (to symbols) Blaze did not catch a Skunk, and Blaze caught a Rabbit if and only if Blaze caught a Cat.

  Solution (78% correct): $\neg C(b, \text{Skunk}) \wedge (C(b, \text{Rabbit}) \leftrightarrow C(b, \text{Cat}))$

- **Statement II:** (to English) $C(b, \text{Cat}) \rightarrow \neg(C(b, \text{Skunk}) \vee C(b, \text{Rabbit}))$.

  Solution (85% correct): If Blaze caught a Cat, then Blaze did not catch either a Skunk or a Rabbit.

- **Statement III:** (to symbols) No dog caught any flying animal.

  Solution (50% correct): $\forall d : \forall a : F(a) \rightarrow \neg C(d, a)$
  There were a lot of correct ways to say this, but a fair number of incorrect ones as well.

- **Statement IV:** (to English) $\forall x : [C(r, x) \rightarrow \neg M(x)] \wedge [C(b, x) \rightarrow M(x)]$

  Solution (91% correct): Given any animal, if Rhonda caught it then it was not a mammal, and if Blaze caught it then it was a mammal.
  For future reference, it was important to note that this statement does not say that any dogs catch animals, only the ones that they do not.

- **Statement V:** (to symbols) Each dog caught exactly one animal.

  Solution (26% correct): $\forall d : \exists a : \forall x : C(d, a) \leftrightarrow (x = a)$
  We ask something like this every time, and people have trouble getting all the quantifiers right. You need to say both that each dog catches one, and that no dog catches more than one. The solution above is probably the simplest, but there are many others.

Family Name: ____________________

(B, 10) Boolean Proof:

Using Statement I and II *only*, prove that the three boolean statements $p_1 = C(b, \text{Cat})$, $p_2 = C(b, \text{Rabbit})$, and $p_3 = C(b, \text{Skunk})$ are **all false**.

You may use either a truth table or a propositional proof. Remember that you must prove *both* that your solution satisfies Statements I and II, *and* that no other solution satisfies both of them.

Note: These two statements translate to "$\neg p_3 \wedge (p_2 \leftrightarrow p_1)$" and "$p_1 \rightarrow \neg(p_2 \vee p_3)$".

Hint: When we say "Statements I and II only", this includes the fact that you should not use Statement V.

Solution:

If we assume that p_1 is true, then by Modus Ponens on Statement II, p_2 and p_3 are both false. But $p_1 \wedge \neg p_2$ contradicts $p_2 \leftrightarrow p_!$, which follows from Statement I by Right Separation. Since assuming p_1 leads to a contradiction, p_1 must be false, and p_2 must be false because it is equivalent to p_1. Finally, p_3 is false by Left Separation on Statement I, and we have proved that all three variables must be false.

(Actually, not "finally", because we have to verify that Statements I and II are true when all three variables are false. This is true for Statement I because it is the conjunction of a true literal and the equivalence of two false variables. Statement II is satisfied vacuously.)

Here is an abbreviated truth table. It shows that the ONLY values that satisfy both "$\neg p_3 \wedge (p_2 \leftrightarrow p_1)$" and "$p_1 \rightarrow \neg(p_2 \vee p_3)$" to be true are $p_1 = p_2 = p_3 = F$.

p_1	p_2	p_3	$\neg p_3$	$p_2 \leftrightarrow p_1$	$\mathbf{\neg p_3 \wedge (p_2 \leftrightarrow p_1)}$	$\neg(p_2 \vee p_3)$	$\mathbf{p_1 \rightarrow \neg(p_2 \vee p_3)}$
T	*T*	*T*	*F*	*T*	*F*	*F*	*F*
T	*T*	*F*	*T*	*T*	*T*	*F*	*F*
T	*F*	*T*	*F*	*F*	*F*	*F*	*F*
T	*F*	*F*	*T*	*F*	*F*	*T*	*T*
F	*T*	*T*	*F*	*F*	*F*	*F*	*T*
F	*T*	*F*	*T*	*F*	*F*	*F*	*T*
F	*F*	*F*	*F*	*T*	*F*	*F*	*T*
F	*F*	*F*	*T*	*T*	**T**	*T*	**T**

There were 39% of the students who got full credit for a truth table, 15% with a correct deductive proof, and 12% with deductive proofs that were correct except for the check that the solution works.

Family Name: ____________________

(C, 15) Predicate Proof:

Using any or all of Statements I, II, III, IV, and V, **determine exactly which** animals, if any, were caught by each dog. Your argument should make clear, for each dog and each animal, whether that dog caught it.

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

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.

C(b, Bat)	=	0	C(r, Bat)	=	0
C(b, Cat)	=	**0**	C(r, Cat)	=	0
C(b, Crow)	=	0	C(r, Crow)	=	0
C(b, Rabbit)	=	**0**	C(r, Rabbit)	=	0
C(b, Robin)	=	0	C(r, Robin)	=	0
C(b, Skunk)	=	**0**	C(r, Skunk)	=	0
C(b, Toad)	=	0	C(r, Toad)	=	1
C(b, Vole)	=	1	C(r, Vole)	=	0

Solution: From Statement III we can eliminate the possibility of either dog catching a Bat, Crow, or Robin, by specifying each dog to each flying animal.

From Statement IV, specifying x to each of the five mammals, we know that Rhonda did not catch them. This rules out seven of the eight animals for Rhonda to catch (the Bat was eliminated twice), leaving the Toad as the only possibe animal for Rhonda to catch. For Blaze, we can now eliminate the three flying non-mammals Crow, Robin, and Toad by specifying each of those animals to the second clause. Since the Bat was eliminated as non-flying from Statement III, and the Cat, Rabbit, and Skunk were all eliminated in Question 2, the only remaining possible animal for Blaze to catch is the Vole.

We can specialize the dog in Statement V to Rhonda and instantiate an animal that Rhonda caught, and this must have been the Toad. Similarly, we can specialize the dog in Statement V to Blaze an instantiate an animal that Blaze caught, and this must have been the Vole.

This turns out to have been a pretty easy Dog Proof, with 75% of you getting full credit and a mean score of 13.4/15.

Family Name: ____________________

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

For natural n, set $S_n = \sum_{i=1}^{n} \frac{1}{(2k-1)(2k+1)}$.

Prove by induction that, for all positive naturals n, $S_n = \frac{n}{2n+1}$.

As an example, $S_3 = \frac{1}{1\cdot 3} + \frac{1}{3\cdot 5} + \frac{1}{5\cdot 7} = \frac{3}{7} = \frac{3}{2\cdot 3+1}$.

(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:** $P(n) : S_n = \frac{n}{2n+1}$. n is a positive natural.

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

> **Solution:** The base case is $n = 1$.
> $S_1 = \frac{1}{1\cdot 3} = \frac{1}{2\cdot 1+1}$, so $P(1)$ is true.

(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 and, in your induction step, exactly where the induction hypothesis is being used.

Solution

(i) $P(n) : S_n = \frac{n}{2n+1}$. n is a positive natural.

(ii) The base case is $n = 1$. $S_1 = \frac{1}{1\cdot 3} = \frac{1}{2\cdot 1+1}$, so $P(1)$ is true.

(iii) Assume correctness of $P(n)$. Then

$$
\begin{aligned}
S_{n+1} &= \sum_{i=1}^{n+1} \frac{1}{(2k-1)(2k+1)} \\
&= \frac{1}{(2n+1)(2n+3)} + S_n && \text{Defn of } S_n \\
&= \frac{1}{(2n+1)(2n+3)} + \frac{n}{2n+1} && \text{Substituting IH} \\
&= \frac{1 + n(2n+3)}{(2n+1)(2n+3)} && \text{Algebraic Manipulation} \\
&= \frac{2n^2 + 3n + 1}{(2n+1)(2n+3)} && \text{Algebraic Manipulation} \\
&= \frac{(2n+1)(n+1)}{(2n+1)(2n+3)} && \text{Algebraic Manipulation} \\
&= \frac{n+1}{2n+3} = \frac{n+1}{2(n+1)+1}. && \text{Algebraic Manipulation}
\end{aligned}
$$

So $\left(S_n = \frac{n}{2n+1}\right) \rightarrow \left(S_{n+1} = \frac{n}{2(n+1)+1}\right)$, i.e., $P(n) \rightarrow P(n+1)$.

Marking Note (2A1): LHS-RHS error

A very common error is using the LHS=RHS method (Left Hand Side = Right Hand Side method) without clearly labelling the pieces. As an example, someone might have written

We want to prove the goal $S_{n+1} = \frac{n+1}{2(n+1)+1}$.
And then written

$$\begin{aligned}
\frac{n+1}{2(n+1)+1} &= S_{n+1} \\
\frac{n+1}{2(n+1)+1} &= S_n + \frac{1}{(2n+1)(2n+3)} && \text{Defn of } S_n \\
\frac{n+1}{2(n+1)+1} &= \frac{n}{2n+1} + \frac{1}{(2n+1)(2n+3)} && \text{IH} \\
\frac{n+1}{2(n+1)+1} &= \frac{(2n+1)(n+1)}{(2n+1)(2n+3)} && \text{Algebraic Manipulation} \\
\frac{n+1}{2(n+1)+1} &= \frac{n+1}{2n+3} \\
\frac{n+1}{2(n+1)+1} &= \frac{n+1}{2(n+1)+1}.
\end{aligned}$$

This would be marked INCORRECT because it makes no sense. It is repurposing the "=" sign in a way that is not naturally understandable. By default, the "=" sign denotes that the thing on the left equals the thing on the right.

The solver's intent was to start with two things that are a-priori different and show that the thing on the right becomes equal to the thing on the left through various manipulations.

It would would have been acceptable if that intent was explained clearly in the text with the left side being labelled as the goal and the right as the starting assumption. It would also have been acceptable if the formatting was expalined at the start of the proof. But neither was done, so someone reading it would have no idea as to what was intended.

Since this particular error was clearly described and explicitly warned against in the class notes (Lecture 23) it had points deducted for it. The class notes also show how something formatted this way could be made acceptable.

Marking Note (2A2): Bad Derivations.

The correct first step was

$$S_{n+1} = S_n + \frac{1}{(2n+1)(2n+3)} + S_n.$$

Some submissions wrote

$$S_{n+1} = S_n + \frac{1}{(2n)(2n+2)} + S_n.$$

i.e., adding products of evens rather than odds. This could not be fixed later on.

Marking Note (2A3): Missing Algebra. Some students got to

$$S_{n+1} = \frac{n}{2n+1} + \frac{1}{(2n+1)(2n+3)}$$

and then said that "using algebra" this resolves to

$$S_{n+1} = \frac{n+1}{2n+3}.$$

This had some points deducted. A correct solution needed to show enough of the algebra so that a reader could instantly see the correctness of one line being equivalent to the next. That was part of what the "how clear and mathematically precise your proof is" part of the instructions was marking.

As an example of why this is important, many students who made error (2A2) above also skipped some of the algebra step, just saying that it resolved to the correct solution. In their case that was wrong because it could *not* resolve correctly. A clear proof needs to show enough information that such an error could be quickly caught by the reader. Leaving out the algebra at this step makes that impossible.

Family Name: ____________________

Question 2 (B, 5):: (Induction Extra Credit)

Define A_n as follows:

$$A_0 = 0, \quad A_1 = 2, \quad \text{and } \forall n > 1, \quad A_{n+1} = 4A_n - 4A_{n-1}.$$

Note $A_2 = 4A_1 - 4A_0 = 4 \cdot 2 = 8$ and $A_3 = 4A_2 - 4A_1 = 4 \cdot 8 - 4 \cdot 2 = 24$.
So, $A_2 = 2 \cdot 2^2$ and $A_3 = 3 \cdot 2^3$.

Prove by induction that, for all naturals n, $A_n = n2^n$.

(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: $P(n) : A_n = n2^n$. n any natural.

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

Solution: The two bases cases are the given initial conditions $n = 0, 1$.
$A_0 = 0 = 0 \cdot 2^0$ and $A_1 = 2 = 1 \cdot 2^1$.

(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 and, in your induction step, exactly where the induction hypothesis is being used.

Solution:

This uses strong induction. Let $n \geq 1$. Assume $P(i)$ is correct for all $i \leq n$. Then

$$\begin{aligned} A_{n+1} &= 4A_n - 4A_{n-1} && \text{(Definition)} \\ &= 4 \cdot n2^n - 4 \cdot (n-1)2^{n-1} && \text{(two applications of the IH)} \\ &= (2n - (n-1))2^{n+1} \\ &= (n+1)2^{n+1}, \end{aligned}$$

i.e., $P(n+1)$ is true.

Marking Notes:

(2B1) The induction has two base cases, $n = 0, 1$.

Showing that $P(n)$ is true for all n requires first showing that $P(0)$ and $P(1)$ are true. But neither of those can be proven using the defining recurrence, so they must be given as base cases.

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: $\mathbf{w} = abc$ or

R3: $\mathbf{w} = aa\mathbf{v}b$ where $\mathbf{v} \in \mathbf{B}$ or

R4: $\mathbf{w} = a\mathbf{v}bb$ where $\mathbf{v} \in \mathbf{B}$ or

R5: $\mathbf{w} = uv$ where $u \neq \lambda$, $v \neq \lambda$, $\mathbf{u} \in \mathbf{B}$ and $\mathbf{v} \in \mathbf{B}$.

R6: No other strings are in $\mathbf{B}$.

Examples: Let $\mathbf{w}_1 = aabcbb$, $\mathbf{w}_2 = aaabbb$, $\mathbf{w}_3 = abcabb$, $\mathbf{w}_4 = aaaaabb$.
Then $\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3 \in \mathbf{B}$ and $\mathbf{w}_4 \notin \mathbf{B}$.

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 $\mathbf{w}$.

$N_a(\mathbf{w}_1) = 2$, $N_b(\mathbf{w}_1) = 3$, $N_c(\mathbf{w}_1) = 1$. $|\mathbf{w}_1| = 6$.
$N_a(\mathbf{w}_2) = 3$, $N_b(\mathbf{w}_2) = 3$, $N_c(\mathbf{w}_2) = 0$. $|\mathbf{w}_2| = 6$.
$N_a(\mathbf{w}_3) = 2$, $N_b(\mathbf{w}_3) = 3$, $N_c(\mathbf{w}_3) = 1$. $|\mathbf{w}_3| = 6$.
$N_a(\mathbf{w_4}) = 5$, $N_b(\mathbf{w}_4) = 2$, $N_c(\mathbf{w}_4) = 0$. $|\mathbf{w}_4| = 7$.

Parts (A), (B) and (C) are on the following pages.

Parts (B) and (C) require induction proofs. When writing their solutions you must use the the mathematical notation we provided above.

In the proofs, be sure to clearly describe your induction goal and, in your induction step, exactly where the induction hypothesis is being used.

(B) and (C) are being marked on how clear and mathematically precise the proof is. Ambiguous explanations or explanations missing details will have points deducted.

Also if your proofs have multiple pieces, place each piece in a separate paragraph with space between the paragraphs.

(A, 4)

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 $\mathbf{B}$. A justification is a statement of the rules used and previous lines results referred to.

The final line in the derivation should be the justification of $\mathbf{w}$.

Here is a derivation that $\mathbf{w} = aabcabbbb \in \mathbf{B}$.
1. $\mathbf{v_1} = \lambda \in \mathbf{B}$. (R1)
2. $\mathbf{v_2} = a\,bb = a\,\mathbf{v_1}\,bb \in \mathbf{B}$. (R4)
3. $\mathbf{v_3} = abc \in \mathbf{B}$. (R2)
4. $\mathbf{v_4} = abc\,abb = \mathbf{v_2}\mathbf{v_1} \in \mathbf{B}$. (R5 and lines 2 and 3)
5. $\mathbf{w} = a\,abcabb\,bb = a\mathbf{v_4}bb \in \mathbf{B}$. (R4)

Give a derivation that shows that the string $\mathbf{w} = aaabcaabb$ 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. $\mathbf{v_2} = aa\,b = aa\,\mathbf{v_1}\,b \in \mathbf{B}$. (R3)
3. $\mathbf{v_3} = abc \in \mathbf{B}$. (R2)
4. $\mathbf{v_4} = abc\,aab = \mathbf{v}_3\mathbf{v}_2 \in \mathbf{B}$ (Lines 2,3 and R5)
5. $\mathbf{w} = aa\,abcaab\,b = aa\,\mathbf{v_4}b \in \mathbf{B}$. (Line 4 and R3)

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.

A justification is the rule applied along with why it can be applied, i.e., where the substrings being used by the rule came from. Derivations that did not explicitly identify the rules used had points deducted.

Only one rule was permitted per line.

Note. It was impossible to use R4 to create the goal string. Any derivation that used a string created by R4 contained an error.

Family Name: ____________________

(B, 8) Prove by induction that, for every $\mathbf{w} \in B$, $|w|$ is divisible by 3, i.e., $(|w| \bmod 3) = 0$.

Solved using structural induction

(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: $P(w) : |w|$ is divisible by 3. $w \in \mathbf{B}$.
Alternatively $P_1(w) : (w \in \mathbf{B}) \rightarrow |w|$ is divisible by 3

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

Solution: The two initial conditions for $P(w)$ are $w = \lambda$ and $w = abc$. Since $|\lambda| = 0$ and $|abc| = 3$ and they are both divisible by three, $P(w)$ is true for the the initial conditions.

(iii) 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:

The proof will be by structural induction.

Let $w \in \mathbf{B}$. w must have been built by R3, R4 or R5. We need to examine each of those cases separately.

(R3) If w was created using R3, then $w = aa\mathbf{v}b$ where $\mathbf{v} \in \mathbf{B}$.
By the IH, $|\mathbf{v}|$ is divisible by 3.
Since $|\mathbf{w}| = 3 + |\mathbf{v}|$, this implies $|\mathbf{w}|$ is also divisible by 3.

(R4) If w was created using R4, then $w = a\mathbf{v}bb$ where $\mathbf{v} \in \mathbf{B}$.
By the IH, $|\mathbf{v}|$ is divisible by 3.
Since $|\mathbf{w}| = 3 + |\mathbf{v}|$, this implies $|\mathbf{w}|$ is also divisible by 3.

(R5) If w was created using R5, then $w = \mathbf{uv}$ where $\mathbf{u}, \mathbf{v} \in \mathbf{B}$.
By the IH, $|\mathbf{u}|$ is divisible by 3 and $|\mathbf{v}|$ is divisible by 3.
Since $|\mathbf{w}| = \mathbf{u}| + |\mathbf{v}|$, this implies $|\mathbf{w}|$ is also divisible by 3.

Since this is true for each possible way of constructing $\mathbf{w}$, $|\mathbf{w}|$ is divisible by 3.

(B, 8) Prove by induction that, for every $\mathbf{w} \in B$, $|w|$ is divisible by 3, i.e., $(|w| \bmod 3) = 0$.

Solved using induction on string length

(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:** $P(n)$: If $w \in \mathbf{B}$ and $|w| \leq n$ then $|w|$ is divisible by 3.

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

> **Solution:** This is a bit more complicated. The base case can either be $n = 0$ or $n = 0, 1, 2, 3$. If the base case was only $n = 0$ then, in the IS, the case $n = 3$ needed to be dealt with separately.
> We show the case $n = 0, 1, 2, 3$.
> Since 0 is divisible by 3, $P(0)$ is true.
> No strings of length 1 and 2 could be built with the rules, so $P(1)$ and $P(2)$ are true.
> Since any string with length 3 is divisible by 3, $P(3)$ is true as well.

(iii) 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: Assume $P(n)$ is true and $n \geq 3$.

To prove $P(n+1)$ we must prove that if $w \in \mathbf{B}$ and $|w| = n+1$, then $n+1$ is divisible by 3.

Let $w \in \mathbf{B}$ with $|w| = n+1$ and $n \geq 3$

Since $n \geq 3$, $w \neq abc$, so w was not built by R2. w must therefore have been built by R3, R4 or R5. We need to examine each of those cases separately.

(R3) If $\mathbf{w}$ was created using R3, then $w = aa\mathbf{v}b$ where $\mathbf{v} \in \mathbf{B}$.
Then $|v| = n+1-3 = n-2 < n$ so, by the IH, $|\mathbf{v}|$ is divisible by 3.
Since $|\mathbf{w}| = 3 + |\mathbf{v}|$, this implies $n+1 = |\mathbf{w}|$ is also divisible by 3.

(R4) If $\mathbf{w}$ was created using R4, then $w = a\mathbf{v}bb$ where $\mathbf{v} \in \mathbf{B}$.
Then $|v| = n+1-3 = n-2 < n$ so, by the IH, $|\mathbf{v}|$ is divisible by 3.
Since $|\mathbf{w}| = 3 + |\mathbf{v}|$, this implies $|\mathbf{w}|$ is also divisible by 3.

(R5) If $\mathbf{w}$ was created using R5, then $w = \mathbf{uv}$ where $\mathbf{u}, \mathbf{v} \in \mathbf{B}$ and $u, v \neq \lambda$ so $|u|, |v| \geq 1$
Then, by the IH, $|\mathbf{u}|$ is divisible by 3 and $|\mathbf{v}|$ is divisible by 3.
Since $|\mathbf{w}| = |\mathbf{u}| + |\mathbf{v}|$, this implies $|\mathbf{w}|$ is also divisible by 3.

Since this is true for each possible way of constructing $\mathbf{w}$, $|\mathbf{w}|$ is divisible by 3.

Since this is true for *every* $\mathbf{w} \in \mathbf{B}$ with $|\mathbf{w}| = n+1$, $P(n+1)$ is true,

Family Name: ____________________

(C, 8) Prove by induction that, for every $\mathbf{w} \in B$, $N_a(\mathbf{w}) \leq 2N_b(\mathbf{w})$.

Solved using Structural Induction

(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: $P(w) : N_a(\mathbf{w}) \leq 2N_b(\mathbf{w})$. $w \in \mathbf{B}$.
Alternatively $P_1(w) : (w \in \mathbf{B}) \rightarrow (N_a(\mathbf{w}) \leq 2N_b(\mathbf{w}))$.

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

Solution: The two initial conditions for $P(w)$ are $w = \lambda$ and $w = abc$.
Since $N_a(\lambda) = 0 = N_b(\lambda)$ $N_a(\lambda) = 2N_b(\lambda)$ so $P(\lambda)$ is true,
Since $N_a(abc) = 1 = N_b(abc)$, $N_a(abc) < 2N_b(abc)$, so $P(abc)$ is true.

(iii) 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: The proof will be by structural induction.

Let $w \in \mathbf{B}$. w must have been built by R3, R4 or R5. We need to examine each of those cases separately.

(R3) If w was created using R3, then $w = aa\mathbf{v}b$ where $\mathbf{v} \in \mathbf{B}$.
By the IH, $N_a(\mathbf{v}) \leq 2N_b(\mathbf{v})$. Then

$$\begin{aligned} N_a(\mathbf{w}) &= N_a(\mathbf{v}) + 2 \\ &\leq 2N_b(\mathbf{v}) + 2 \qquad (IH) \\ &= 2(N_b(\mathbf{v}) + 1) \\ &= 2N_b(\mathbf{w}). \end{aligned}$$

(R4) If w was created using R4, then $w = a\mathbf{v}bb$ where $\mathbf{v} \in \mathbf{B}$.
By the IH, $N_a(\mathbf{v}) \leq 2N_b(\mathbf{v})$. Then

$$\begin{aligned} N_a(\mathbf{w}) &= N_a(\mathbf{v}) + 1 \\ &\leq 2N_b(\mathbf{v}) + 1 \qquad (IH) \\ &< 2N_b(\mathbf{v}) + 4 \\ &= 2(N_b(\mathbf{v}) + 2) \\ &= 2N_b(\mathbf{w}). \end{aligned}$$

(R5) If w was created using R5, then $w = \mathbf{uv}$ where $\mathbf{u}, \mathbf{v} \in \mathbf{B}$.
By the IH, By the IH, $N_a(\mathbf{u}) \leq 2N_b(\mathbf{u})$ and $N_a(\mathbf{v}) \leq 2N_b(\mathbf{v})$. Then

$$\begin{aligned} N_a(\mathbf{w}) &= N_a(\mathbf{u}) + N_a(\mathbf{v}) \\ &\leq 2N_b(\mathbf{u}) + 2N_b(\mathbf{v}) \qquad (IH) \\ &= 2(N_b(\mathbf{u}) + N_b(\mathbf{v})) \\ &= 2N_b(\mathbf{w}). \end{aligned}$$

Since this is true for each possible way of constructing $\mathbf{w}$, $P(w)$ is always true.

Family Name: ____________________

(C, 8) Prove by induction that, for every $\mathbf{w} \in B$, $N_a(\mathbf{w}) \leq 2N_b(\mathbf{w})$.

Solved using Induction on length.

(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: $P(w) : N_a(\mathbf{w}) \leq 2N_b(\mathbf{w})$ for all $w \in \mathbf{B}$ with $|w| \leq n$.

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

Solution: The base case can either be $n = 0$ or $n = 0, 1, 2, 3$. We show the $n = 0$ version.
The base case is $n = 0$. The only string w with $|w| = 0$ is λ.
Since $N_a(\lambda) = 0 = N_b(\lambda)$, $P(0)$ is true.

(iii) 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:

Assume $P(n)$ is true and $n \geq 0$.

To prove $P(n+1)$ we must prove that if $w \in \mathbf{B}$ and $|w| = n+1$, $N_a(\mathbf{w}) le 2N_b(\mathbf{w})$.

Let $w \in \mathbf{B}$ with $|w| = n+1$. w must have been built by R2, R3, R4 or R5. We need to examine each of those cases separately.

(R2) In this case $\mathbf{w} = abc$ and $N_a(\mathbf{w}) = 1 < 2 \cdot 1 = 2N_b(\mathbf{w})$.

(R3) If w was created using R3, then $w = aa\mathbf{v}b$ where $\mathbf{v} \in \mathbf{B}$.
Since $|v| = |w| - 3 = n - 2 < n$, from the IH, $N_a(\mathbf{v}) \leq 2N_b(\mathbf{v})$. Then

$$\begin{aligned} N_a(\mathbf{w}) &= N_a(\mathbf{v}) + 2 \\ &\leq 2N_b(\mathbf{v}) + 2 \qquad (IH) \\ &= 2(N_b(\mathbf{v}) + 1) \\ &= 2N_b(\mathbf{w}). \end{aligned}$$

(R4) If w was created using R4, then $w = a\mathbf{v}bb$ where $\mathbf{v} \in \mathbf{B}$.
Since $|v| = |w| - 3 = n - 2 < n$, from the IH, $N_a(\mathbf{v}) \leq 2N_b(\mathbf{v})$. Then

$$\begin{aligned} N_a(\mathbf{w}) &= N_a(\mathbf{v}) + 1 \\ &\leq 2N_b(\mathbf{v}) + 1 \qquad (IH) \\ &< 2N_b(\mathbf{v}) + 4 \\ &= 2(N_b(\mathbf{v}) + 2) \\ &= 2N_b(\mathbf{w}). \end{aligned}$$

(R5) If w was created using R5, then $w = \mathbf{uv}$ where $\mathbf{u}, \mathbf{v} \in \mathbf{B}$ with $u, v \neq \lambda$ so $|u|, |v| > 0$.
Therefore $|u|, |v| \leq n+1-1 = n$, So, from the IH, $N_a(\mathbf{u}) \leq 2N_b(\mathbf{u})$ and $N_a(\mathbf{v}) \leq 2N_b(\mathbf{v})$.
Then

$$\begin{aligned} N_a(\mathbf{w}) &= N_a(\mathbf{u}) + N_a(\mathbf{v}) \\ &\leq 2N_b(\mathbf{u}) + 2N_b(\mathbf{v}) \qquad (IH) \\ &= 2(N_b(\mathbf{u}) + N_b(\mathbf{v})) \\ &= 2N_b(\mathbf{w}). \end{aligned}$$

Since this is true for each possible way of constructing $\mathbf{w}$, $P(w)$ is always true.
Since this is true for *every* $\mathbf{w} \in \mathbf{B}$ with $|\mathbf{w}| = n + 1$, $P(n + 1)$ is true,

Marking Notes for Question 3B and 3C

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.
Note. This Marking note is for both problem 3B and problem 3C.

The Induction step in structural induction is supposed to (assuming R2 is in the base case) be of the form:

> Let $w \neq \lambda$, $w \in \mathbf{B}$.
> Then either $w = aavb$ or $w = avbb$ or $w = uv$ 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 the proof with

Let $x \in \mathbf{B}$*, assume by the IH that* $P(x)$*, and now 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. Read those for an explanation as to why this is wrong.

A related error is to start the proof with
Assume $P(u)$ *and* $P(v)$.

That's illegal, again, because the v or u, v should be *defined by the* w. Solutions that started out like this had small deductions.

An associated error that sometimes was connected with this one was to explicitly require $w = vv$ in R5, i.e., to always have $u = v$ in R5. This restriction is not valid.

Marking Note 3B2: Improper IS structure.
Note. This Marking note is for both problem 3B and problem 3C.

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 $w = aavb$ or $w = avbb$ or $w = uv$ 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 $P(aavb)$ or $P(avbb)$ or $P(uv)$. That has no meaning without the proper introduction introducing what is going to be proven, i.e., $P(w)$.

There were many solutions that started with "*Assume* $P(w)$ *is true*" but then went on with a correct proof. These had a small amount deducted because this formulation is incorrect. $P(w)$ was not being *assumed* to be true. The entire purpose of the IS was to *prove* that $P(w)$ is true for a specific w.

Note that this last error is of types both 3B1 and 3B2. This is because many solutions started off with this incorrect formulation and but some continued doing bottom-up while others, without realizing out, switched to top-down.

Marking Note 3B3: $P(x+1)$ error.
Note. This Marking note is for both problem 3B and problem 3C.

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.

Marking Note 3B4: Base case error when doing induction by length.
Note. This Marking note is for both problem 3B and problem 3C.

There is a subtlety in the base case when when doing induction by length. This is that $\mathbf{B}$ contains 3 strings of length 3. One of them *abc*, is given by rule 2 and cannot be built by using the other rules. The other two are *aab* and *abb*, which are built using rules R3 and R4.

If the base cases given in the proof are $n = 0, 1, 2, 3$ then the case $n = 3$ has to explicitly discuss *aab* and *abb*.

If the base case given in the proof is only $n = 1$, then the IS must explicitly deal with the case $n + 1 = 3$ being different.

Marking Note 3B5 Clarity.
Note. This Marking note is for both problem 3B and problem 3C.

The rubric used stressed that the IS would be marked on *how clear and mathematically precise your proof is, with ambiguous explanations or missing details leading to points being deducted.* In particular, note that the intuitive ideas behind the solutions of both 3B and 3C were not complicated. What was being marked was how well these simple ideas could be transformed into a clear and mathematically precise proof.

Among other things, this means that a proof could have correct ideas but still have points deducted because it wasn't clear or mathematically precise. Not being clear included missing details that explained why a line followed from a previous line (and sometimes which previous line(s) it followed from). *Mathematically precise* means that ambiguous prose explanations without precise mathematical descriptions, i.e., formulas, had points deducted as well.

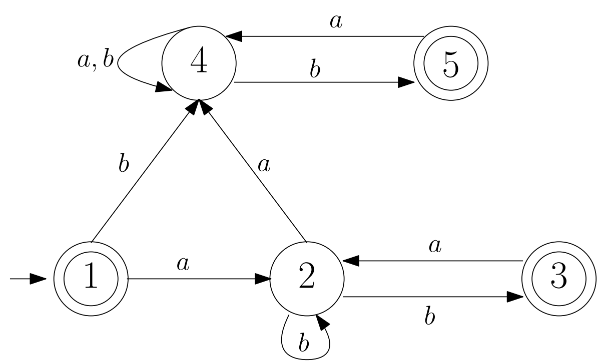
Family Name: ____________________

Question 4 (40+5 points total):

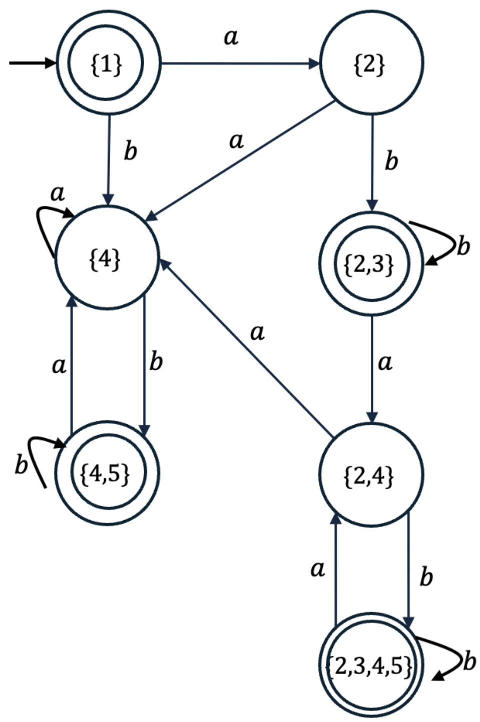
This question involves several of the constructions from Kleene's Theorem.

We begin with an ordinary NFA N, with alphabet $\Sigma = \{a, b\}$, state set $\{1, 2, 3, 4, 5\}$, start state 1, final state set $\{1, 3, 5\}$, and transition relation

$$\{(1, a, 2), (1, b, 4), (2, b, 2), (2, b, 3), (2, a, 4), (3, a, 2), (4, a, 4), (4, b, 4), (4, b, 5), (5, a, 4)\}.$$

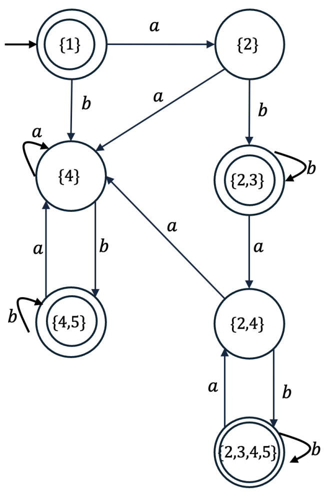


(a, 10) Subset Construction: *Using the Subset Construction*, find a DFA D that is equivalent to the NFA N. It's sufficient to show the new DFA diagram, without further explanation, if it comes from the given construction.



40% of you got full credit, with a mean of 7.65/10.

(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 $L(D) = L(D')$.



First Partition

$F = \left\{ \begin{array}{c} \{1\}, \{2,3\}, \{4,5\}, \\ \{2,3,4,5\} \end{array} \right\}$
$N = \{\{2\}, \{2,4\}, \{4\}\}$

	a	b
{1}	N	N
{2,3}	N	F
{4,5}	N	F
{2,3,4,5}	N	F

	a	b
{2}	N	F
{2,4}	N	F
{4}	N	F

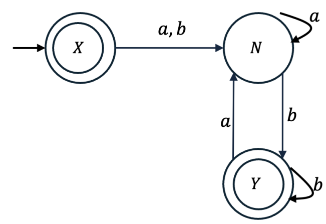
2nd Partition

$X = \{\{1\}\}$
$Y = \{2,3\}, \{4,5\}, \{2,3,4,5\}\}$
$N = \{\{2\}, \{2,4\}, \{4\}\}$

	a	b
{2,3}	N	Y
{4,5}	N	Y
{2,3,4,5}	N	Y

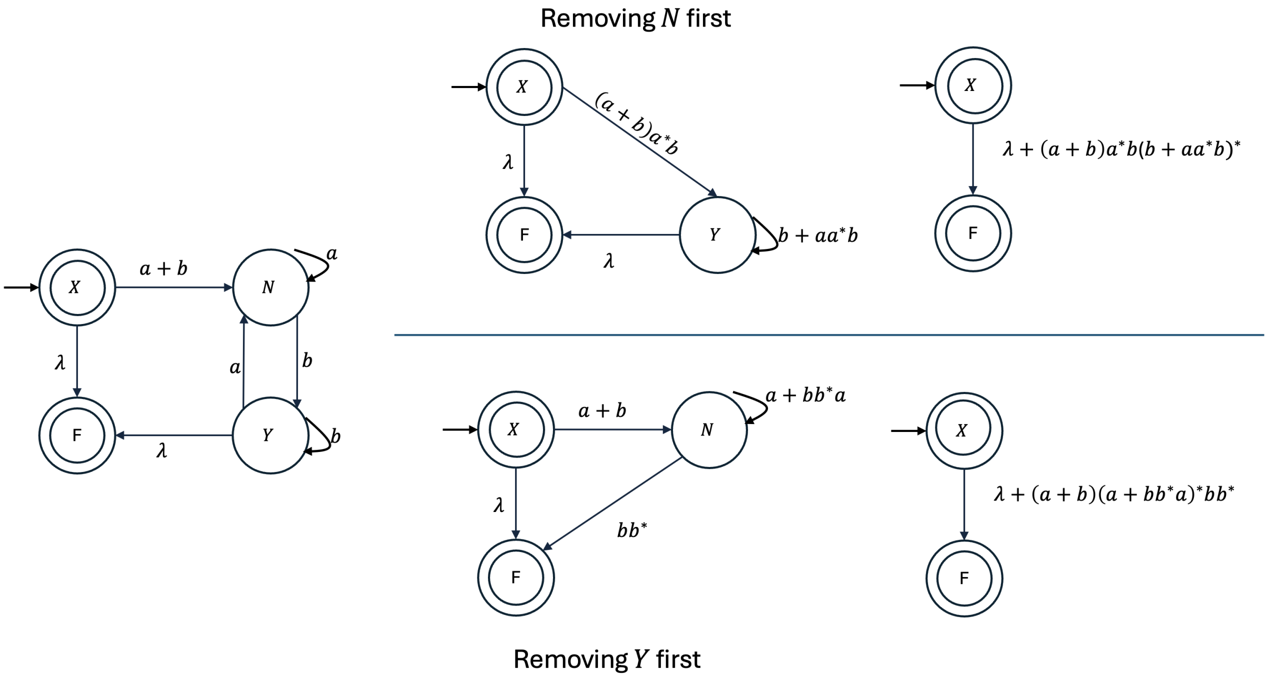
	a	b
{2}	N	Y
{2,4}	N	Y
{4}	N	Y

Final Result



17% of you got full credit, with a mean of 5.32/10.

(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.



13% full credit, mean score 4.54/10, only 8% with no answer. These were generally difficult to grade, particularly with the results of mistakes in Q4a and Q4b. We will need to insist that everyone be clear about the exact DFA that they are starting from. The correct solution to those problems, and many but not all of the wrong ones, had more than one final state. For my survival, I stopped reading further if you didn't put a λ-move from each former final state to the new start state. People who didn't do that generally didn't get more than 4/10, though I probably gave 4/10 to some of those who deserved less.

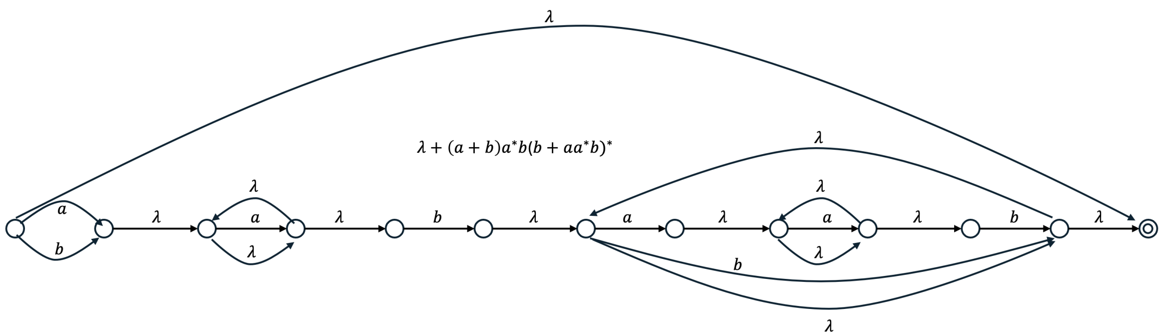
(d, 5XC) Making a λ-NFA: *Using the construction* from lecture and the textbook, compute a λ-NFA from the regular expression in part (c). A different equivalent λ-NFA will get only partial credit.

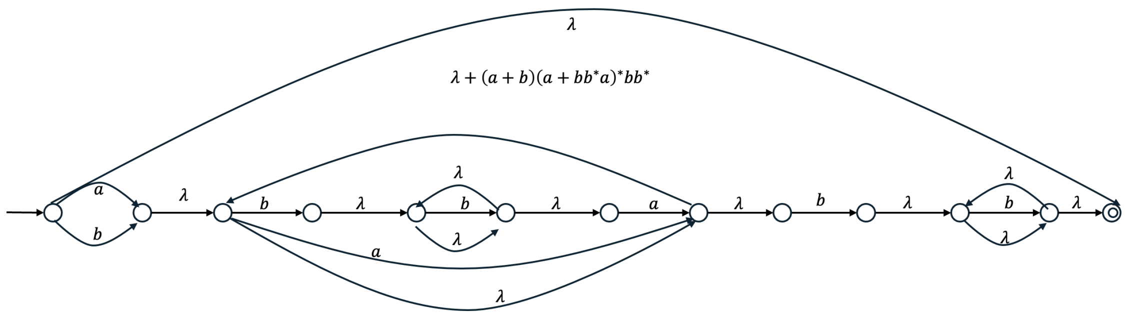
There are two possible solutions for this problem, one each corresponding to the two solutions to the previous one.

Note that both solutions contain "redundant" λ-moves that could be combined with neighboring edges, without changing the language accepted.

Those would not received full credit, though, because they would not have been created by the construction taught in class.

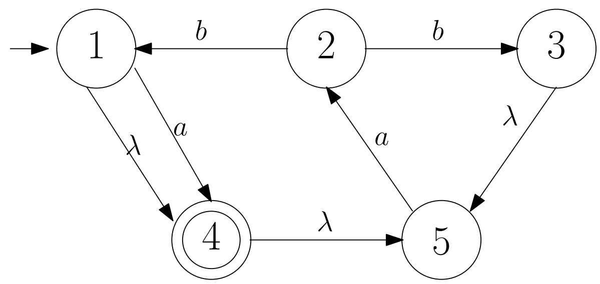
20% with full credit, mean score 2.07/5, 38% no answer.





Family Name: ____________________

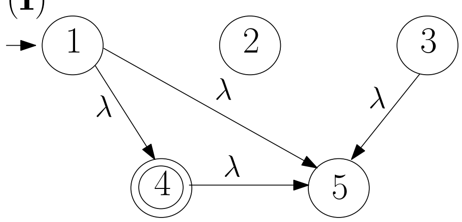
For the last part of this problem, we will work with a *different* language and machine. Here is a new λ-NFA K, which has alphabet $\Sigma = \{a, b\}$, state set $\{1, 2, 3, 4, 5\}$, start state 1, final state set $\{4\}$, and transition relation $(1, \lambda, 4), (1, a, 4), (2, b, 1), (2, b, 3), (3,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



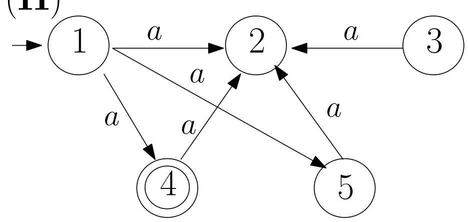
(e, 10) Killing λ-moves: *Using the construction* from the lectures and the textbook, find an ordinary NFA K' that is equivalent to the λ-NFA K above.

- 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'.

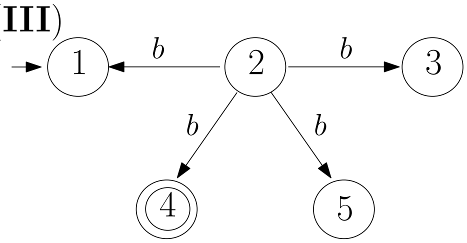
(I)



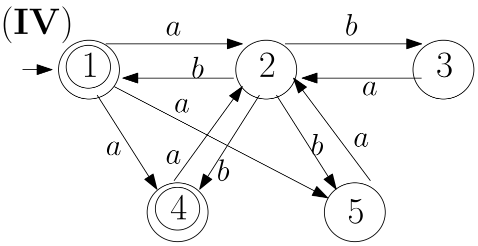
(II)



(III)



(IV)



21% of you got full credit, with a mean of 7.55/10.

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	T	F	T	T	F	F	T	T	F

- (a) Statement V from the dog proof is equivalent to the statement that the relation C is a one-to-one function from D to A.

  Solution: FALSE (52% correct). It is equivalent to the statement that C is a function. C being one-to-one would include the statement that the two dogs did not catch the same animal, but Statement V says nothing about that. Perhaps future generations of 250 students will eventually notice how often the definitions of "function" and "one-to-one function" show up in the exams.

- (b) Let A and B be any two different non-empty finite sets. Then either $|A \cup B| > |A|$ or $|A \cup B| > |B|$, or both.

  Solution: TRUE (70% correct). For the sets to be different, there must be an element in one that is not in the other. If $x \in A$ and $x \notin B$, for example, $|A \cup B| > |B|$ because $A \cup B$ contains all the elements of B plus at $|A \cup B| > |A|$.

- (c) The regular expressions $(aba)^*ba$ and $ab(aba)^*$, over the alphabet $\{a, b\}$, denote the same language.

  Solution: FALSE (79% correct). ba is a member of the first, but not of the second.

- (d) Let N be an NFA with at least one final state. Then it is possible that $L(N) = \emptyset$.

  Solution: TRUE (60% correct). There might be one final state that is not reachable from the start state.

- (e) It is possible to have a DFA D and an NFA N such that $L(D) = L(N)$, but N has strictly fewer states than does D.

  Solution: TRUE (60% correct). D might be a non-minimal DFA and N its minimal DFA, since any NFA is also a DFA.

- (f) Let G be any connected undirected graph, with at least three nodes. Then there must exist two nodes a and b such that a DFS and BFS, with start node a and goal node b, will find different paths from a to b.

  Solution: FALSE (52% correct). If the graph is a tree, then for any nodes a and b, there is exactly one simple path from a to b, and both the BFS and the DFS will find it.

- (g) Let L_1 and L_2 be two languages over the same alphabet $\Sigma = \{a, b\}$. Then if $L_1 \subseteq L_2$ and L_2 is Turing decidable, then L_1 must also be Turing decidable.

Solution: FALSE (45% correct). We know that there exists an undecidable language. We could L_1 be that language, and L_2 be Σ^*. Hardly any set of "easier" language is going to be closed under subset, except perhaps the set of finite languages.

- (h) Let $p > 1$ be a power of two and let $q > 1$ be an odd number. Then for any naturals a and b, if $a \equiv b \pmod{p}$ and $a \equiv b \pmod{q}$, then $a \equiv b \pmod{pq}$.

  Solution: TRUE (69% correct). This follows from the Chinese Remainder Theorem, since p and q must be relatively prime by the Fundamental Theorem of Arithmetic.

- (i) If the statements $p \to q$, $\neg q \to \neg r$, and $\neg(\neg p \wedge \neg r)$ all all true, then q must also be true.

  Solution: TRUE (70% correct). The second statement is equivalent to $r \to q$, and the third statement is equivalent to $p \vee r$. So at least one of p and r are true, and whichever one is true then implies q.

- (j) Let M be a Turing machine whose transition function contains $\delta(i, \square) = (p, a, R)$ and $\delta(p, a) = (p, a, L)$. (Here i is M's start state, $\square$ is its blank symbol, and a is a letter in its input alphabet Σ.) Then it is possible that M halts on every input in Σ^*.

  Solution: FALSE (48% correct). On any input in $a\Sigma^*$, the machine will see the initial blank in state i, write an a over the first letter, and move right to the first original a in state p. Then it will move left twice and hang.