NAME: _____

SPIRE ID: _____

COMPSCI 250
Introduction to Computation
Final Midterm **Solutions** Spring 2024

D. A. M. Barrington and M. J. Golin 13 May 2024

DIRECTIONS:

- Answer the problems on the exam pages.
- There are seven problems on pages 2-11, some with multiple parts, for 125 total ordinary points and 5 extra credit points. The final scale was A = 104, B = 83, C = 62, D = 41, F = 20.
- Pages 12 and 13 contains useful definitions and is given to you separately – do not put answers on it!
- If you need extra space use the back of a page – both sides are scanned.
- If absolutely necessary, you may attach additional pages to be graded, but this is a big nuisance to us.
- No books, notes, calculators, or collaboration.
- Arithmetic expressions, such as "$23 + (4^{17} \cdot (35\%3))$", do not need to be evaluated as single numbers for full credit.
- Scrap paper during the exam may be used, if it starts out blank. Please transfer any answers to the exam pages.

1	/10
2	/10
3	/15
4	/10
5	/20
6	/40+5
7	/20
Total	/125+5

Recently, while their humans were away, Blaze and Rhonda stayed with a dog sitter, along with the sitter's dog Hawkeye. Three times during the visit (Friday, Saturday, and Sunday), the sitter emailed to report what the three dogs were doing. Let D be the set $\{B, H, R\}$ of the three dogs, T be the set $\{Fri, Sat, Sun\}$ of the reporting times, and A be a set $\{ba, di, lo, re\}$ of activities: barking, digging, lounging, and retrieving. Your goal is to determine all the values of the function f from $D \times T$ to A given by the three reports. You may assume that f is a function, so that each dog at each time is engaged in exactly one of the four possible activities in A.

We also define four propositions to be used in some of the statements:

p_1 means "$f(H, Fri) = re$"
p_2 means "$f(H, Sat) = re$"
p_3 means "$f(R, Fri) = re$"
p_4 means "$f(R, Sat) = re$"

Question 1 (Dog Proof: Translations) Translate each statement as according to the directions:

- (a, 2) (to symbols) (Statement I)
  If Hawkeye did not retrieve on Friday, then Hawkeye retrieved on Saturday and Rhonda retrieved on Friday.
  $\neg(f(H, Fri) = re) \rightarrow (f(H, Sat) = re) \wedge (fR, Fri) = re))$.
  Pretty much everyone got this right.

- (b, 2) (to English) (Statement II)
  $[(f(H, Fri) = re) \vee (f(R, Sat) = re)] \leftrightarrow (f(H, Sat) = re) \wedge \neg((f(R, Fri) = re) \rightarrow (f(H, Sat) = re))$
  Either Hawkeye retrieved on Friday or Rhonda retrieved on Saturday if and only if Hawkeye retrieved on Saturday and is not the case that if Rhonda retrieved on Friday, then Hawkeye retrieved on Saturday.
  Many people got the scope of the NOT or $\rightarrow$ operations wrong, or miscomputed the negation of the $\rightarrow$.

- (c, 2) (to symbols) (Statement III)
  No activity was done by all three dogs, and if any activity was ever done by two different dogs, then that activity was retrieving. (Note added later: "activity was done" means "activity was done at any time".)
  $[\neg\exists a : \forall d : \exists t : f(d, t = a)] \wedge [\forall a : \forall d : \forall d' : (\exists t : \exists t' : ((f(d, t) = a) \wedge (f(d', t') = a) \wedge (d = d')) \rightarrow (a = re)]$

This was a difficult problem, and hardly anyone got it completely correctly. The time of the events is not mentioned, but your answer has to refer to them, because you can't say anything about an activity happening unless you say when. So you need existential quantifiers to say "did the activity at any time". Fortunately, many of those who got this translation wrong relied on the English version for Question 3.

- (d, 2) (to English) (Statement IV)
  $\forall a : [(a \neq re) \rightarrow \exists d : \exists t : \exists t' : (f(d,t) = f(d,t') = a) \wedge (t \neq t')]$
  For any activity other than retrieving, there was a dog that did that activity on two different days.
  The vast majority of you got this correctly.

- (e, 2) (to symbols) (Statement V)
  On Sunday, Hawkeye barked and Rhonda lounged.
  $f(H, Sun) = ba \wedge f(R, Sun) = lo$
  Pretty much everyone got this right.

Question 2 (Boolean Dog Proof) (10): Both Questions 2 and 3 use the definitions, predicates, and statements above.

Using Statements I and II *only*, determine the truth values of the four propositions p_1, meaning "$f(H, Fri) = re$", p_2, meaning "$f(H, Sat) = re$", p_3, meaning "$f(R, Fri) = re$", and p_4, meaning "$f(R, Sat) = re$".

You may use a truth table or a deductive sequence proof. Don't forget that if you use a deductive sequence proof, your argument must show both that your solution satisfies all of the statements, and that there is no other solution.

Using our abbreviations, Statement I is "$\neg(p_1 \rightarrow (p_2 \wedge p_3))$ *and Statement II is* $(p_1 \vee p_4) \leftrightarrow (p_2 \wedge \neg(p_3 \rightarrow p_2))$*. The right-hand side of Statement II is equivalent (by Definition of Implication and DeMorgan) to* $p_2 \wedge (p_3 \wedge \neg p_2)$*, which is a contradiction. So the left-hand side of Statement II is false, and we thus know that* p_1 *and* p_4 *are both false by DeMorgan. Then by Modus Ponens on Statement I,* p_2 *and* p_3 *are both true.*

We will omit the truth table.

For the deductive sequence proof, a lot of people still lost a point because they didn't show that their solution satisfied the statements, even though the same issue came up on the first midterm and we mentioned again in the problem statement. The correct solution relied heavily on taking the negation of the implication – again, I think many people wisely used the symbolic version for the argument rather than their English translation.

A complication in the truth table was that Statement I refers to only three of the variable, so you have to take that into consideration when you compare the results with Statement II, which uses all four variables. A lot of solutions were less clear because they did not state their conclusions. If you do a truth table and you find there to be multiple solutions, the likely reason is that you have solved the table for an incorrect version of the problem. You will definitely get more points if you recognize that there is a problem, and say what your table actually implies.

Question 3 (Predicate Dog Proof) (15):]

This question also uses the definitions, predicates, and statements above. Now assume that all of Statements I-V are true. (If you proved any results from Question 2, you may quote them here without further proof.) **Determine all nine values of the function f. That is, which dog performed which activity on each of the three days.** Prove your answer, making clear when you are using each of the quantifier rules.

From Question 2 we know that Hawkeye retrieved on Saturday and that Rhonda retrieved on Friday. From the first clause of Statement III, specifying a to re, Blaze cannot have retrieved, since if she did we would have a contradiction if all three dogs retrieved. Specifying Statement IV to each of the non-retrieving activities, each must have been done twice by a single dog. Using Separation on Statement V, we know that Hawkeye barked on Sunday, and must have barked twice, so he barked on Friday and Sunday. Similarly Rhonda lounged on Sunday, so she lounged twice, so this must have been on Saturday and Sunday. There was a dog who dug at least twice, and this must have been Blaze because the activities of the other two dogs have been determined. Specifying Statement III to each of ba and lo, we know that Blaze could not have done either, and we already determined that Blaze did not retrieve, so Blaze must have dug on all three days.

People generally did well on this, at least if they correctly solved Question 2. The statements never refer to Blaze, or to digging, but all the other negative statements force you to the conclusion that Blaze dug all three three times. But what if you got Question 2 wrong?

If you still had both Hawkeye and Rhonda each retrieving on one of Friday and Saturday and not retrieving on the other, the problem becomes very similar to the intended one, and you got full credit if you solved your version. If one of those dogs retrieved on both Friday and Saturday, there is no solution because that dog would be doing its Sunday activity only once, and no other dog could do that activity twice. In that case I still gave your 11/15 **if you figured out that Blaze still needed to dig all three days. If if either Hawkeye or Rhonda failed to retrieve at all, there were then multiple solutions, because that dog could do its Sunday activity all three days, and Blaze could retrieve on one of the three days while digging on the other two.**

Question 4 (10): (Induction 1) Let $S(n)$ be the sum of the first $n+1$ odd squares minus the sum of the first $n+1$ even squares. Mathematically, this is defined by

$$S(n) = \sum_{i=0}^{n}(2i+1)^2 - \sum_{i=0}^{n}(2i)^2.$$

Prove *by induction* that for all naturals n, $S(n) = 2n^2 + 3n + 1$.

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: "$S(n) = 2n^2 + 3n + 1$".*
It is defined over all natural n.

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

Solution: *The base case is $n = 0$. Since*

$$S(n) = \sum_{i=0}^{0}(2i+1)^2 - \sum_{i=0}^{0}(2i)^2 = 1 - 0 = 1 = 2 \cdot 0^2 + 3 \cdot 0 + 1,$$

$P(0)$ is correct.

(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.
You must use the mathematical notation we provided when writing the proof.
If your proof has multiple pieces, place each piece in a separate paragraph with space between the paragraphs. 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: *Our goal is to assume $P(n)$ and prove $P(n+1)$, i.e., that*

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

Assume $P(n)$. Then

$$\begin{aligned}
S(n+1) &= \sum_{i=0}^{n+1}(2i+1)^2 - \sum_{i=0}^{n+1}(2i)^2 \\
&= \left(\sum_{i=0}^{n}(2i+1)^2 - \sum_{i=0}^{n}(2i)^2\right) + (2(n+1)+1)^2 - (2(n+1))^2 \\
&= S(n) + (2(n+1)+1)^2 - (2(n+1))^2 \\
&= S(n) + 2 \cdot 2(n+1) + 1 \\
&= \left(2n^2 + 3n + 1\right) + (4(n+1)+1) && \textit{(from IH } P(n)\textit{)} \\
&= \left(2n^2 + 4n + 2\right) + 3(n+1) + 1 \\
&= 2(n+1)^2 + 3(n+1) + 1. && \textit{(Goal!)}
\end{aligned}$$

Thus $P(n+1)$ is correct.

Marking Note 4G:

In general, an induction step in a clear mathematical proof needs to *explicitly* identify where the induction hypothesis is being used. A proof that does not do this is not a clear proof and would usually have points deducted.

Marking Note 4a: For (i), some students wrote $P(x) = 2x^2 + 3x + 1$.

This was incorrect. $P(x)$ was specifically required by the problem to be an induction hypothesis, i.e. **a logical statetment.** It could not be a **numerical quantity.**

Some students wrote something like "$P(x) = 2x^2 + 3x + 1$ where $P(x)$ is the first $x + 1$ odd squares minus the first $x + 1$ even squares".
This would also be incorrrect. Again, $P(x)$ is not expressing a logical statement.

This type of error propagated and frequently led to point deductions in part (iii) as well. Since $P(x)$ is not well defined, any statement using it led to mathematical ambiguity. For example, in the situation above, since $P(x)$ is not a logical statement, it is not possible to write "assuming $P(x)$".

Marking Note 4b: Some students started by writing

$$S(n+1) = 2n^2 + 3n + 1 + (2(n+1)+1)^2 - (2(n+1))^2$$

or something similar, without showing where that line came from. They had points deducted for missing details.

Marking note 4c: Some solutions did not differentiate between goals, definitions and assumptions. In these cases, points were deducted (depending upon what was clearly explained and/or missing)

As an example, some solutions split into two parts. One saying

$$(A) \quad S(n+1) = 2(n+1)^2 + 3(n+1) + 1.$$

The second that

$$(B) \quad S(n+1) = \sum_{i=0}^{n+1} (2i+1)^2 - \sum_{i=0}^{n+1} (2i)^2,$$

and then showed that (A) equals (B) (using the IH when evaluating (B)).

The problem with this is that
(A) is a **goal**.
(B) states how $S(n+1)$ is defined.
If that distinction wasn't made clear, the proof was ambiguious and points were deducted.

Similarly, if the statement of the IH was used in (B) without some indication that the statement WAS the the IH, points were sometime also deducted (depending upon the clarity of the statement).

Marking note 4d: Some students wrote

$$S(n+1) = 2n^2 + 3n + 1 + (4n + 5)$$

without a clear explanation as to where the $2n^2 + 3n + 1$ and/or the $4n + 5$ came from.

Such a solution had points deducted. Even though it was not "wrong", it was missing detail.s The only ones who could understand what it was doing would be someone who already knew the solution and knew what was missing. That's not a *clear, precise, unambiguous mathematical proof.*

A clear, precise, unambiguous mathematical proof has to be a convincing argument for someone who does not already know the result.

Marking Note Compendium: Short list of the major issues: (MN = "Marking note")

1. (MN 4a) IH was not a **logical** statement or IH was not clear on what parameter induction was doing done upon
2. (MN 4G) IS did not identify WHERE in the proof the IH was being used.
3. (MN 4c) Proof in IS did not clearly differentiate between the Induction Goal, the Induction Hypothesis and things that weer already defined (in this case $S(n)$).
4. (MN 4G, MN 4b, MN 4d) Lack of preciseness and clarity in the proof, i.e., missing explanation.

Question 5 (20): (Induction 2) Define the language $\mathbf{B}$ on $\Sigma = \{a, b\}$ 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: $\mathbf{w} = aa\mathbf{v}b$ where $\mathbf{v} \in \mathbf{B}$ or

R4: $\mathbf{w} = \mathbf{uv}$ where $\mathbf{u} \in \mathbf{B}$, $\mathbf{u} \neq \lambda$, $\mathbf{v} \in \mathbf{B}$, and $\mathbf{v} \neq \lambda$.

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

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

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

Examples: Set $\mathbf{w}_1 = abab$, $\mathbf{w}_2 = abaabb$, $\mathbf{w}_3 = baba$, $\mathbf{w}_4 = aba$, $\mathbf{w}_5 = aaabb$.
Then $\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_5 \in \mathbf{B}$, $\mathbf{w}_3, \mathbf{w}_4 \notin \mathbf{B}$
$N_a(\mathbf{w}_1) = N_a(\mathbf{w}_3) = N_a(\mathbf{w}_4) = 2$. $N_a(\mathbf{w}_2) = N_a(\mathbf{w}_5) = 3$.
$N_b(\mathbf{w}_1) = N_b(\mathbf{w}_3) = N_b(\mathbf{w}_3)2$. $N_b(\mathbf{w}_2) = 3$. $N_b(\mathbf{w}_4) = 1$.

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. Here is a derivation that $aaababbab \in \mathbf{B}$.
1. $v_1 = \lambda \in \mathbf{B}$. (R1)
2. $v_2 = ab = av_1b \in \mathbf{B}$. (R2 and line 1)
3. $v_3 = ab\,ab = v_2v_2 \in \mathbf{B}$. (R4 and line 2)
4. $v_4 = aa\,abab\,b = aav_3b \in \mathbf{B}$. (R3 and line 3)
5. $w = aaababb\,ab = v_4v_2 \in \mathbf{B}$. (R4 and lines 2 and 4)

For each of part (a), (b), (c) write your answer on the page it was given. If you run out of space, continue the solution on the back of that page (with a note stating that you are writing on the back).

Question 5a (4):

(i) Give a derivation that shows that the string *aabaabbaab* is in **B**.

Solution: There were two different possible legal derivations. Notice that they differ only in lines 4 and 5. Also, the orderings of some of the lines in the derivations could be swapped.

Derivation 1:

1. $v_1 = \lambda \in \mathbf{B}$. (R1)
2. $v_2 = aa\,b = aav_1b \in \mathbf{B}$. (R3 and line 1)
3. $v_3 = a\,b = av_1b \in \mathbf{B}$. (R2 and line 1)
4. $v_4 = a\,ab\,b = av_3b \in \mathbf{B}$. (R2 and line 3)
5. $v_5 = aab\,aabb = v_3v_4 \in \mathbf{B}$. (R4 and lines 3 and 4)
6. $v_6 = aabaabb\,aab = v_5v_2 \in \mathbf{B}$. (R4 and lines 5 and 2)

Derivation 2:

1. $v_1 = \lambda \in \mathbf{B}$. (R1)
2. $v_2 = a\,ab = aav_1b \in \mathbf{B}$. (R3 and line 1)
3. $v_3 = a\,b = av_1b \in \mathbf{B}$. (R2 and line 1)
4. $v_4' = ab\,aab = v_3v_2 \in \mathbf{B}$. (R4 and lines 2 and 3)
5. $v_5 = a\,abaab\,b = a\,v_4'\,b \in \mathbf{B}$. (R2 and line 4)
6. $v_6 = aabaabb\,aab = v_5v_2 \in \mathbf{B}$. (R4 and lines 5 and 2)

Marking Note 5ai1: Solutions were expected to be in the derivation format described above, i.e. separate lines, with each line showing one construction.

In addition, each line had to indicate which rules it had used.

Solutions not in that format had points deducted.

(ii) Of the two strings *abbaabab* and *aaabbaab*, one is in **B** and the other is not. For the one that is not, given an argument as to why you know that it is not.

Solution: The first string is not in **B** and the second string is in **B**.

The first string is not in **B** because no string in **B** begins with *abb*. This is because every b must have at least one a that appears before it and "matches" that b.

Saying the above would have been sufficient. For those that want a formal proof, the statement $P(w)$, "if $w \in \mathbf{B}$ of length greater than 2, then w must start with *aaa*, *aab* or *aba*" can be proven by induction on w. Since $P(w)$ implies that 'no string in **B** begins with *abb*", that would suffice.

To prove $P(w)$ note that
I. Every non-empty $w \in \mathbf{B}$ must start with an a. (Simple induction on the rules.)
II. From I, every $w \in \mathbf{B}$ of length greater than 1, must start with *aa* or *ab*.
III. We can now prove by $P(w)$ by induction for all w, $|w| \geq 3$.

To prove III, note that if $|w| = 3$ then $w = aab$ and the statement is correct.
If $|w| > 3$, then consider w's last derivation step. It cannot be R1.
If it was R2, then, from II, w starts with *aaa* or *aba*.
If it was R3, then, from I, w starts with *aaa*.
If it was R4, then $w = uv$ with $u, b \in \mathbf{B}$.
If $|u| \geq 3$, $P(w)$ follows from $P(u)$.
If $|u| = 2$, then from II, $u = aa$ or $u = ab$ and $P(w)$ follows from I.

Since $P(w)$ is true for all $w \in \mathbf{B}$,"no string in **B** begins with *abb*".,

Alternatively, one could derive that $w_2 = aaabbaab \in \mathbf{B}$.
Derivation:
1. $v_1 = \lambda \in \mathbf{B}$. (R1)
2. $v_2 = aa\,b = aav_1b \in \mathbf{B}$. (R3 and line 1)
3. $v_3 = a\,aab\,b \in \mathbf{B}$. (R2 and line 1)
4. $w_2 = aaabb\,aab = v_3v_2 \in \mathbf{B}$. (R3 and line 3)

Question 5b (8): Prove *by induction* that for all $\mathbf{w} \in \mathbf{B}$, $N_b(\mathbf{w}) \leq N_a(\mathbf{w})$.

Marking Note 5bG: It was possible to solve this via induction on words, on the size of words, on the number of construction rules used to generate words, or even on the number of a's or b's in a word. In solutions 1 and 2 following, we illustrate the first 2. No matter what was used, $P(x)$ had to clearly indicate what the induction was being done on.

Solution I: Induction on words in B.

(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: $P(w)$: *If* $w \in \mathbf{B}$, *then* $N_b(w) \leq N_a(w)$.
Stated for $w \in \{a, b\}^*$.

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

Solution: *Base case is* $w = \lambda$.
Then $N_b(\lambda) = 0 = N_a(\lambda)$, *so* $P(\lambda)$ *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.
You must use the mathematical notation we provided when writing the proof.
If your proof has multiple pieces, place each piece in a separate paragraph with space between the paragraphs. Be sure to clearly describe your induction goal.

Solution:

Need to show that $P(w)$ *is true. Assume* $w \in \mathbf{B}$ *where* $w \neq \lambda$.
Then w *must have been built using one of the three construction rules.*
We show that in all three cases, $N_b(w) \leq N_a(w)$.

R2: *If* $\mathbf{w} = a\mathbf{v}b$ *where* $\mathbf{v} \in \mathbf{B}$, *then, by the IH,* $P(v)$ *is true, so* $N_b(v) \leq N_a(v)$.
Since $N_b(w) = N_b(v) + 1$ *and* $N_a(w) = N_a(v) + 1$,

$$N_b(w) = N_b(v) + 1 \leq N_a(v) + 1 = N_a(w).$$

R3: *If* $\mathbf{w} = aa\mathbf{v}b$ *where* $\mathbf{v} \in \mathbf{B}$, *then, by the IH,* $P(v)$ *is true, so* $N_b(v) \leq N_a(v)$.
Since $N_b(w) = N_b(v) + 1$ *and* $N_a(w) = N_a(v) + 2$,

$$N_b(w) = N_b(v) + 1 \leq N_a(v) + 1 < N_a(w).$$

R4: $\mathbf{w} = \mathbf{uv}$ *where* $\mathbf{u} \in \mathbf{B}$ *and* $\mathbf{v} \in \mathbf{B}$ *then, by the IH,* $P(u)$ *and* $P(v)$ *are true so* $N_b(u) \leq N_a(u)$ *and* $N_b(v) \leq N_a(v)$.
Since $N_b(w) = N_b(u) + N_b(v)$ *and* $N_a(w) = N_a(u) + N_a(v)$,

$$N_b(w) = N_b(u) + N_b(v) \leq N_a(u) + N_a(v) = N_a(w).$$

Since we have shown that is true across all possible ways of constructing w, *the fact is proven.*

Comment: It is not true that w *must be built using* **exactly** *one of the 3 rules.* w *might have multiple derivations. But that has no impact on the proof.*

Solution II for 5b: Induction on the size of w.

(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* $N_b(w) \leq N_a(w)$.
Stated for natural n.

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

Solution: *Base case is* $n = 0$.
The only string with $|w| = 0$ *is* λ *which is in* $\mathbf{B}$.
$N_b(\lambda) = 0 = N_a(\lambda)$, *so* $P(\lambda)$ *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.
You must use the mathematical notation we provided when writing the proof.
If your proof has multiple pieces, place each piece in a separate paragraph with space between the paragraphs. Be sure to clearly describe your induction goal.

Solution: *Assume that* $P(n)$ *is true.*

We need to show that $P(n+1)$ *is true.That is,*
We need to show that *for all* w satisfying $|w| \leq n+1$ **and** $w \in \mathbf{B}$, $N_b(w) \leq N_a(w)$.

Let $w \in \mathbf{B}$ *satisfying* $|w| \leq n+1$.

If $|w| < n+1$ *then, by the IH, we know that* $N_b(w) \leq N_a(w)$.
So it only remains to prove the case where $|w| = n+1$.

w *can only have been built by one of the three construction rules.*
We look at each of them in turn.

R2: *If* $\mathbf{w} = a\mathbf{v}b$ *where* $\mathbf{v} \in \mathbf{B}$, *then* $|v| = n-1$. *So, by the IH,* $N_b(v) \leq N_a(v)$.
Then, since $N_b(w) = N_b(v) + 1$ *and* $N_a(w) = N_a(v) + 1$,

$$N_b(w) = N_b(v) + 1 \leq N_a(v) + 1 = N_a(w).$$

R3: *If* $\mathbf{w} = aa\mathbf{v}b$ *where* $\mathbf{v} \in \mathbf{B}$, *then* $|v| = n-2$. *So, by the IH,* $N_b(v) \leq N_a(v)$.
Since $N_b(w) = N_b(v) + 1$ *and* $N_a(w) = N_a(v) + 2$,

$$N_b(w) = N_b(v) + 1 \leq N_a(v) + 1 < N_a(w).$$

R4: $\mathbf{w} = \mathbf{uv}$ *where* $\mathbf{u} \in \mathbf{B}$ *and* $\mathbf{v} \in \mathbf{B}$, $u \neq \lambda$ *and* $v \neq \lambda$ *then,* $|u| \leq n$ *and* $|v| \leq n$.
So, the IH, $N_b(u) \leq N_a(u)$ *and* $N_b(v) \leq N_a(v)$.
Since $N_b(w) = N_b(u) + N_b(v)$ *and* $N_a(w) = N_a(u) + N_a(v)$,

$$N_b(w) = N_b(u) + N_b(v) \leq N_a(u) + N_a(v) = N_a(w).$$

Since we have shown that $N_b(w) \leq N_a(w)$ *is true across all possible ways of constructing* w, *we have proven that if* $|w| = n+1$ *and* $w \in \mathbf{B}$ *then* $N_b(w) \leq N_a(w)$, *completing the proof.*

Comment: It is not true that w *must be built using* **exactly** *one of the 3 rules.* w *might have multiple derivations. But that has no impact on the proof.*

Marking Note5b1: Any induction, either on words in B, the lengths of words, pr some other parameter, MUST work by examining the three cases separately.

It was not possible to do induction by decomposing word w into say, $w = ux$, where $x \in \{a, b\}$. This is because $u \notin \mathbf{B}$ so you would not have an induction hypothesis to rely upon.

Marking Note5b2: When using Solution2, i.e., induction on length, it was necessary to explain WHY, in all cases the IH could be used. This is because, due to the construction rules, $|v| \leq n$ (and in **R4**, $|u| \leq n$).

Marking Note5b3:

The Induction Hypothesis had to have been in the form $P(x)$ where $P(x)$ is a logical statement and not a number. It also had to clearly state how x was used in the definition of $P(x)$.

For example, in solution 1, the induction was done on all words w and $P(w)$ was
$P(w)$: If $w \in \mathbf{B}$, then $N_b(w) \leq N_a(w)$.

In solution 2, the induction was done on all words of length $\leq n$ and $P(n)$ was
$P(n)$: If $w \in \mathbf{B}$, and $|w| \leq n$ then $N_b(w) \leq N_a(w)$.

As an example of an incorrect IH, some students claimed an induction hypothesis $P(w)$ where "w is a string of length n".

This would make no sense because this statement has two parameters, w and n. In order for this to work they would have to write $P(w, n)$ and prove an induction on BOTH w and n which is a lot more complicated to do correctly and not something that we learned how to do in this class.

Marking Note 5b4: A common error, when doing induction on strings, i.e., solution 1, was to say: Assume $P(w)$ and then prove $P(awb)$, $P(aawb)$ and $P(uw)$ for some for some u using rules 2,3 and 4. Some students called this "extending w" or "lengthening w".

This bottom-up approach is incorrect and is logically wrong (at least in this class, this is not the way we derived induction) and justifying the bottom-up approach would be tricky because of the way Rule R4 is structured. **Without a proper IS structure, it's not possible to assume that $P(u)$ is correct.** You can't just write that you know $P(u)$ is correct by induction. The IS structure has to be designed so that you can KNOW that, when processing w, $P(u)$ is correct by induction. The top-down method ensures this. The "bottom-up" one does not.

Any solution, independent of whether it was on strings or naturals, that used the bottom-up approach had points deducted for being incorrect.

To clarify: the IS, as we taught it, is always structured as being proven **top-down** using the following structure.

- The goal of the IS is to prove $P(w)$.
- The IS starts by clearly stating that it wants to prove $P(w)$ for some specific w.
- From the construction rules we know that at least one of the following is true: $w = aub$, $w = aaub$, or $w = uv$, all for for $u \in \mathbf{B}$ and, in the last case, $v \in \mathbf{B}$ as well.
- It then uses the IH, which tells us $P(u)$ (and $P(v)$) to prove $P(w)$ for each of those cases.

Solutions that did not follow this structure had points deducted.

See the actual solutions for examples of how this would be correctly written.

Note that there was a similar error for proofs that claimed to be doing induction on n, the *length* of the string. In those cases, $P(n+1)$ needed to be proven, based on $P(n-1)$. A proof that, say, claimed to prove $P(n+2)$ and $P(n+3)$ from $P(n)$ would be wrong.

Marking Note 5b5: Some students wrote their IH as $P(n)$ where $n \in \mathbf{B}$ and then, in the induction step, tried to prove something about "$n+1$". This was immediaely incorrect because you cannot add "1" to a string. This meant that the entire induction step was wrong because it indicated a major flaw in understanding how induction on strings works.

Marking Note Compendium: Short list of the major issues: (MN = "Marking note")

1. (MN 5b3) $P(x)$ had to be a logical statement with only one parameter. This x could be a string, length of string, No. of rules used, No. of a's or No. of b's.
2. (MN 5b5) The IS structure had to be consistent with the IH structure. In particular, if the IH was a hypothesis on words, i.e, $P(w)$ where $w \in \mathbf{B}$, then the IS could NOT talk about $w+1$.
3. (MN 5b2) The use of the IH in the IS needed to be explicitly identified and, in some cases, justified.
4. (MN 5b1) IS needed to explicitly show separately, for each of the 3 construction rules, why they maintained the IH.
5. (MN 5b4) IS step needed to be "top-down" and not "bottom-up". More explicitly, an IS that started by saying, we know $P(w)$ and will now show $P(awb)$, $P(aawb)$ and $P(wu)$ for some $u \in \mathbf{B}$, was not valid. Similarly, an IS on word length that started by saying we know $P(n)$ and wil prove $P(n+1)$, $P(n+2)$ and $P(n+3)$ was also not valid.

Question 5c (8): Prove *by induction* that for all $\mathbf{w} \in \mathbf{B}$, $N_a(\mathbf{w}) \leq 2N_b(\mathbf{w})$.

Solution I: Induction on words in B.

(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: $P(w)$: *If* $w \in \mathbf{B}$, *then* $N_a(w) \leq 2N_2(w)$.
Stated for $w \in \{a, b\}^*$.

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

Solution: *Base case is* $w = \lambda$.
Then $N_b(\lambda) = 0 = N_a(\lambda)$, *so* $P(\lambda)$ *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.
You must use the mathematical notation we provided when writing the proof.
If your proof has multiple pieces, place each piece in a separate paragraph with space between the paragraphs. Be sure to clearly describe your induction goal.

Solution: *Need to show that* $P(w)$ *is true. Assume* $w \in \mathbf{B}$ *where* $w \neq \lambda$.
Then w *must have been built using one of the three construction rules.*
We show that in all three cases, $N_a(w) \leq 2N_b(w)$.

R2: *If* $\mathbf{w} = a\mathbf{v}b$ *where* $\mathbf{v} \in \mathbf{B}$ *then, by the IH,* $P(v)$ *is true, so* $N_a(v) \leq 2N_b(v)$.
Since $N_b(w) = N_b(v) + 1$ *and* $N_a(w) = N_a(v) + 1$,

$$N_a(w) = N_a(v) + 1 \leq 2N_b(v) + 1 < 2(N_b(v) + 1) = 2N_b(w)$$

R3: *If* $\mathbf{w} = aa\mathbf{v}b$ *where* $\mathbf{v} \in \mathbf{B}$, *then, by the IH,* $P(v)$ *is true, so* $N_a(v) \leq 2N_b(v)$.
Since $N_b(w) = N_b(v) + 1$ *and* $N_a(w) = N_a(v) + 2$,

$$N_a(w) = N_a(v) + 2 \leq 2N_b(v) + 2 = 2(N_b(v) + 1) = 2N_b(w)$$

R4: $\mathbf{w} = \mathbf{uv}$ *where* $\mathbf{u} \in \mathbf{B}$ *and* $\mathbf{v} \in \mathbf{B}$, *then, by the IH,* $P(u)$ *and* $P(v)$ *are true, so* $N_a(u) \leq 2N_a(u)$ *and* $N_a(v) \leq 2N_b(v)$.
Since $N_a(w) = N_a(u) + N_b(v)$ *and* $N_a(w) = N_a(u) + N_a(v)$,

$$N_a(w) = N_a(u) + N_a(v) \leq 2N_b(u) + 2N_b(v) = 2N_b(w).$$

Since we have shown that is true across all possible ways of constructing w, *the fact is proven.*

Solution II: Induction on the size of w

(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* $N_a(w) \leq 2N_b(w)$.
Stated for natural n.

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

Solution: *Base case is* $n = 0$.
The only string with $|w| = 0$ *is* λ *which is in* $\mathbf{B}$.
$N_b(\lambda) = 0 = N_a(\lambda)$, *so* $P(\lambda)$ *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.
You must use the mathematical notation we provided when writing the proof.
If your proof has multiple pieces, place each piece in a separate paragraph with space between the paragraphs. Be sure to clearly describe your induction goal.

Solution:

Assume that $P(n)$ *is true.*

We need to show that *for all* w satisfying $|w| \leq n+1$ **and** $w \in \mathbf{B}$, $N_a(w) \leq 2N_b(w)$.

Let $w \in \mathbf{B}$.

If $|w| < n+1$ *then, by the IH, we know that* $N_a(w) \leq 2N_b(w)$.
So it only remains to prove the case where $|w| = n+1$

w *can only have been built by one of the three construction rules.*
We look at each of them in turn.

R2: *If* $\mathbf{w} = a\mathbf{v}b$ *where* $\mathbf{v} \in \mathbf{B}$ *then* $|v| = n-1$. *By the induction hypothesis* $N_a(v) \leq 2N_b(v)$.
Since $N_b(w) = N_b(v) + 1$ *and* $N_a(w) = N_a(v) + 1$,

$$N_a(w) = N_a(v) + 1 \leq 2N_b(v) + 1 = 2(N_b(v) + 1) = 2N_b(w).$$

R3: *If* $\mathbf{w} = aa\mathbf{v}b$ *where* $\mathbf{v} \in \mathbf{B}$, *then* $|v| = n-2$. *By the induction hypothesis,* $N_a(v) \leq N2_b(v)$.
Since $N_b(w) = N_b(v) + 1$ *and* $N_a(w) = N_a(v) + 2$,

$$N_a(w) = N_a(v) + 2 \leq 2N_b(v) + 2 = 2(N_b(v) + 1).$$

R4: $\mathbf{w} = \mathbf{uv}$ *where* $\mathbf{u} \in \mathbf{B}$ *and* $\mathbf{v} \in \mathbf{B}$, $u \neq \lambda$ *and* $v \neq \lambda$ *then,* $|u| \leq n$ *and* $|v| \leq n$.
By the induction hypothesis, $N_a(u) \leq 2N_b(u)$ *and* $N_a(v) \leq 2N_b(v)$.
Since $N_b(w) = N_b(u) + N_b(v)$ *and* $N_a(w) = N_a(u) + N_a(v)$,

$$N_a(w) = N_a(u) + N_a(v) \leq 2N_b(u) + 2N_b(v) = 2N_b(w).$$

Since we have shown that $N_b(w) \leq N_a(w)$ *is true across all possible ways of constructing* w, *we have proven that if* $|w| = n+1$ *and* $w \in \mathbf{B}$ *then* $N_a(w) \leq 2N_b(w)$.

Marking Note: For a complete proof it was necessary to explain WHY, in all cases the IH could be used. This is because, due the the construction rules, $|v| \leq n$ **(and in R3,** $|u| \leq n$**).**

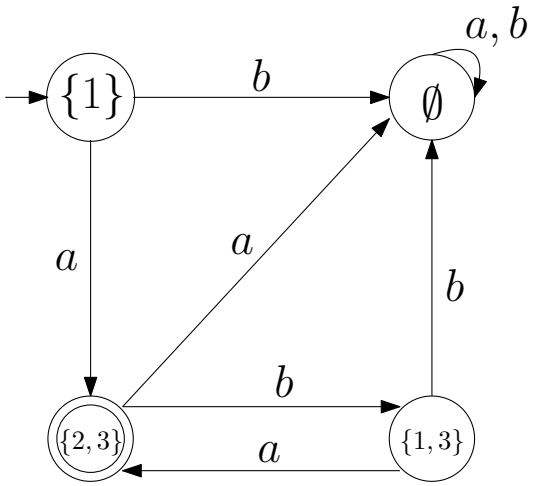
All other marking notes referred to issues raised in Part B.

Question 6 (Kleene Constructions) (40+5): The following ordinary NFA N will be used for parts (a)-(d) of this question. A separate λ-NFA M will be used for part (e).

N has state set $\{1, 2, 3\}$, with 1 the start state and 2 the only final state. The transitions are $(1, a, 2)$, $(1, a, 3)$, $(2, b, 1)$, and $(2, b, 3)$. There is a diagram for N on the supplemental sheet.

- **Question 6a (10):** Using the Subset Construction, build an ordinary DFA D such that $L(D) = L(N)$.

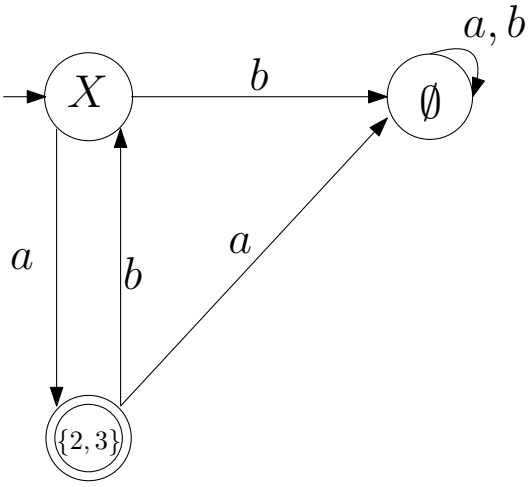
  State $\{1\}$ *has an a-arrow to* $\{2, 3\}$ *and a b-arrow to* $\emptyset$*. State* $\{2, 3\}$ *has an a-arrow to* $\emptyset$ *and a b-arrow to* $\{1, 3\}$*. State* $\{1, 3\}$ *has an a-arrow to* $\{2, 3\}$ *and a b-arrow to* $\emptyset$*. Of course* $\emptyset$ *has both arrows to itself.*



I insisted that if you are asked to produce a DFA, then you should produce a DFA, with a start state, a final state set, and one arrow from each state for each letter in the alphabet. I took off one point for not marking the start state, two points for not marking the final states, and two points for leaving off the arrows on the death state.

- **Question 6b (5XC):** Find a minimal DFA D' such that $L(D') = L(D)$. If D is already minimal, prove that fact, either directly or by carrying out the State Minimization algorithm.

  It is not minimal. We start with $F = \{23\}$ *and* $N = \{1, 13, \emptyset\}$. *On the first round,* 1 *and* 13 *have the same behavior but* $\emptyset$ *does not. On the second round, we have classes* $F = \{23\}$, $D = \{\emptyset\}$, *and* $X = \{1, 13\}$. *Since both 1 and 13 have a-arrows to* F *and b-arrows to* D, *we are done, and we can minimize the DFA by merging states 1 and 13 into state* X.

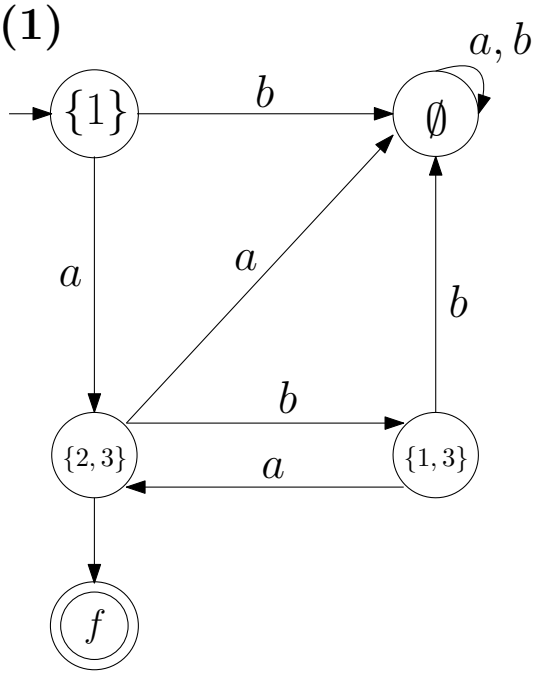


  I gave 5/5 for correctly minimizing a DFA that was *comparable* to the right one, but only 3/5 if you correctly minimized a much easier one, due to a mistake on Q6a. In general I did not give more than 2/5 for wrong answers.

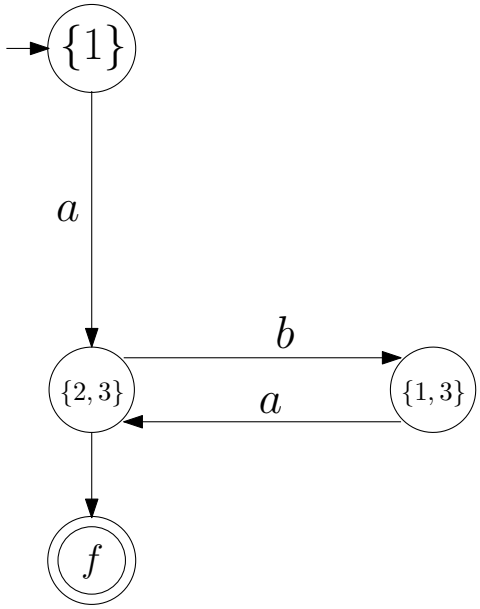
- **Question 6c (10):** Using either D from Question 6a or D' from Question 6b, construct a regular expression R with that $L(R) = L(D)$. If you use the State Elimination algorithm as given in lecture, you do not need to prove your result to be correct. But if you use any other method, you must prove that whatever you do gives the correct result.

  From the original D, *we do not need a new start state, but we must add a new final state* f *with a* λ*-move from* 23 *to* f. *We can first eliminate* $\emptyset$, *which yields no new edges. Then eliminating state 13 gives us a loop on state 23 with label ba. Finally, eliminating state 23 gives us a final regular expression of* $a(ba)^*$.

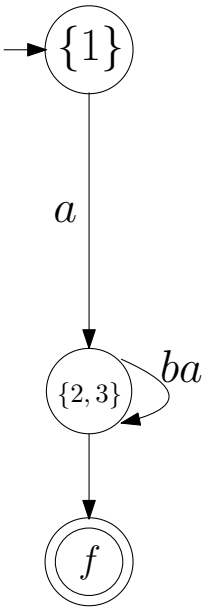
(1)



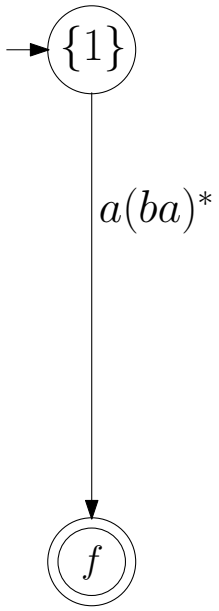
(2)



(3)



(4)



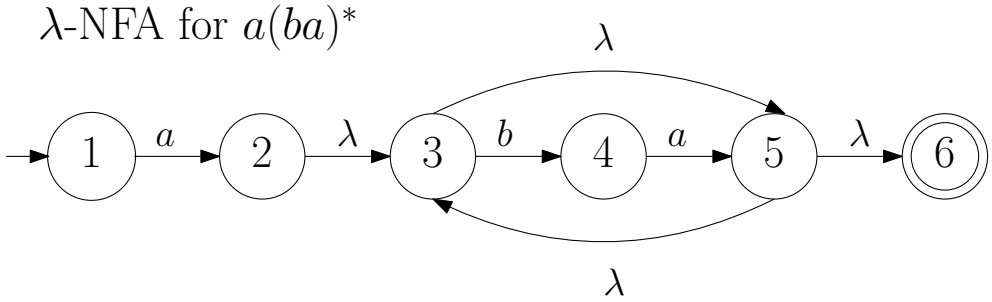
If you chose to eliminate 23 before 13, you get transitions $(1, ab, 13)$, $(1, a, f)$, $(13, ab, 13)$, *and* $13, a, f$, *for a resulting regular expression of* $a + ab(ab)^*a$. *This is equivalent, but will make your job harder in part 6d.*

If instead we use the three-state D', *we need both a new start state* i *and new final state* f, *with* λ*-moves from* i *to* X *and from* 23 *to* f. *It doesn't much matter which of the two intermediate states we eliminate first – if we take* X *first, we get new edges from* i *to* 23 *with label* a *and a loop on* 23 *with label* ba. *Killing* 23 *then leaves a final expression of* $a(ba)^*$. *If you take* 23 *first, we get a new edge from* X *to* f *with label* a *and a loop on* X *with label* ab. *Killing* X *then leaves* $(ab)^*a$ *which is equivalent to* $a(ba)^*$.

In general I gave 4/10 for coherent incorrect answers. There were some people using a variant method to create the expression, but in general they didn't get the right answer so I didn't need to penalize them for not proving correctness. A lot of correct answers got only 6/10 because they were not explained.

- **Question 6d (10):** Construct a new λ-NFA Z from your regular expression R, such that $L(Z) = L(R)$. Since you are *not* going to be working further with Z, we want to come from the algorithm we described in lecture, without any simplifications.

  We get states $\{1, 2, 3, 4, 5, 6\}$, *with start state 1, only final state 6, and transitions* $(1, a, 2)$, $(2, \lambda, 3)$, $(3, b, 4)$, $(3, \lambda, 5)$, $(4, a, 5)$, $(5, \lambda, 3)$, *and* $(5, \lambda, 6)$.



Along with the correct answers by the construction, there were two large classes of solutions – one getting an equivalent λ-NFA by some method other than the construction (getting 7/10) and the other getting an equivalent λ-NFA by the construction, but not getting the λ-moves in exactly the right place. (In many contexts, I wouldn't mind your dropping the move $(2, \lambda, 3)$ **in my example, but we did say "without any simplifications".) In general the people who attempted this problem did pretty well.**

- **Question 6e (10):** Let M be the λ-NFA with state set $\{p, q, r, s\}$, with p as the start state and q as the only final state, and transitions (p, a, q), (p, λ, r), (q, a, s), (r, λ, q), (s, λ, q), and (s, b, r). There is a figure for M on the supplemental sheet. Build an ordinary NFA M', such that $L(M') = L(M)$. If you use the construction from lecture, you do not need to prove correctness of the algorithm. But if you use any other method, you must prove that whatever you do gives the correct result.

  Solution: *(See lower right figure in diagram.) M' has*

  - *4 states $\{p, q, r, s\}$, same as M.*
  - *Final states $\{p, q\}$.*
  - *8 a-moves: (p, a, q), (p, a, s), (q, a, q), (q, a, s), (r, a, q), (r, a, s), (s, a, q), and (s, a, s).*
  - *2 b-moves: (s, b, r) and (s, b, q).*

  To create M', follow the steps taught in class. The appended diagram illustrates this.

  (1) Create the transitive closure of the λ-edges. This only adds the one new λ-edge (p, λ, q).

  (2) Since λ is accepted by M **we must make** *p, the initial state of M,* **a** *new additional* **final state of** *M'.*

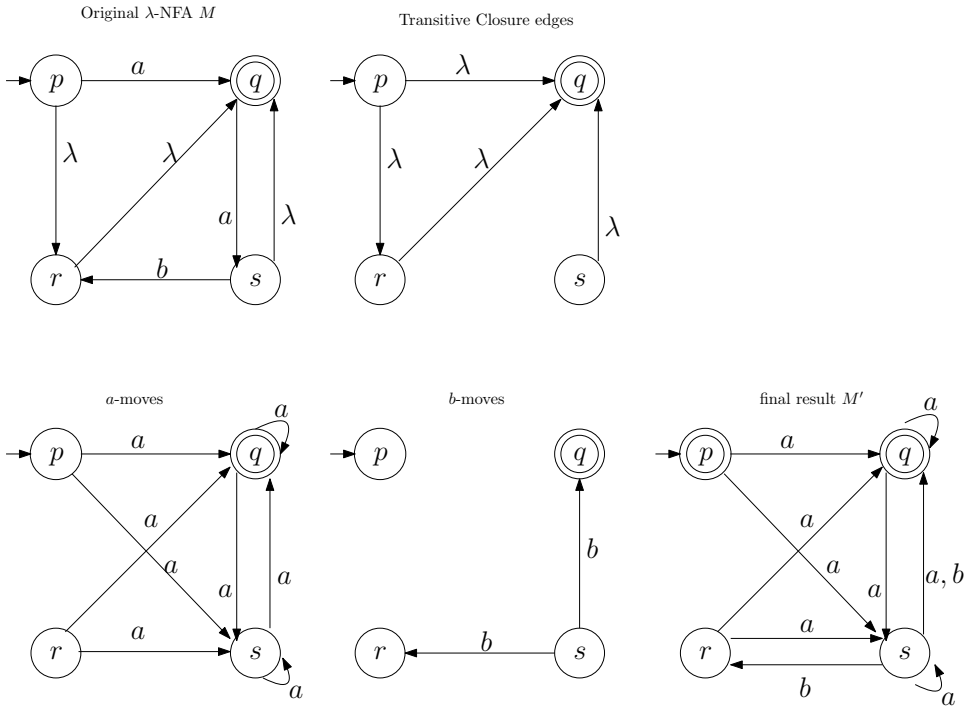
  Now process the letter moves.

  (3) Original a-move (p, a, q) does not add any new a-moves.

  (4) Original a-move (q, a, s) adds **6 new a-moves:**
  (p, a, s), (q, a, q), (r, a, q), (r, a, s), (s, a, q), and (s, a, s)
  It also creates (p, a, q) and (q, a, s) which already existed.

  (5) Original b-move (s, b, r) **adds new b-move** *(s, b, q).*

  Removing all of the λ-edges, adding the six new a-moves and 1 new b-move and making p a final state results in M'. This is the rightmost item on the 2nd line of the figure.



Question 7 (20): The following are ten true/false questions, with no explanation needed or wanted, no partial credit for wrong answers, and no penalty for guessing.

- (a) Recall that the **in-degree** of a node in a directed graph is the number of arcs into it. Then the total number of arcs in a directed graph is equal to the sum of the in-degrees of the nodes.
  TRUE (74% correct). Each arc is counted exactly once in the in-degree of some node.

- (b) Let G be a directed graph with n vertices where $n > 1$. Then if G has a path with n edges, it cannot contain a directed cycle.
  FALSE (76% correct) – in fact such a graph must contain a directed cycle.

- (c) The language **B** (from Question 5) is regular.
  FALSE (36% correct) – We can prove it non-regular by showing that if i and j are natural numbers with $i > j$, $a^{2j}b^j \in$ B but $a^{2i}b^j \notin$ B (because it violates the condition of Question 5c). Thus the infinite sets of strings $\{a^{2j} : j \geq 1\}$ are pairwise L-distinguishable. This score was disappointing. It would be challenging for you to have to prove that B is not regular, but you should have a better idea of what a regular language is, so that you could see that B isn't – it's pretty similar to the Paren language in may ways.

- (d) If L is a language, and two strings u and v are L-distinguishable, then it is possible that there exists a string w such that uw and vw are both in L.
  TRUE (60% correct) – Example. $L = \Sigma^* a \Sigma^*$, $u = \lambda$, $v = a$, $w = a$. Note that $uw \in L$ and $vw \in L$ but u and n are L-distinguishable because $ub \notin L$ while $vb \in L$. More generally, *only one* w has to force uw and vw to have different answers for u and v to be L-distinguishable.

- (e) If N is an NFA, and D is the DFA made from it by the Subset Construction, then D must have a death state (a non-final state with all arrows to itself).
  TRUE and FALSE (100% correct) – The intended answer to this problem was FALSE. The reason was that if N is any NFA with an existing transition arrow for every state and letter, then the basic subset construction algorithm shown in class would never create a death state.

  It was later was pointed out to us that if someone alternatively created the DFA by first constructing ALL of the possible subsets, then $\emptyset$ would be constructed as a death state, even though it would never be reached from the start state. Because of this ambiguity, we decided to mark both TRUE and FALSE as correct for this question.

- (f) If a language is a Turing Decidable language, then we cannot conclude that it can be represented by a regular expression.
  TRUE (49% correct) – We have proved that many such languages, like the Paren language, have no DFA's and thus have no regular expressions by Kleene's Theorem.

- (g) If R is any regular expression, the regular languages $(RR^*)^*$, R^*R^*, and R^* are all equivalent.
  TRUE (75% correct) – Each of the languages is the same. Any string in any of them must be the concatenation of zero or more strings from $L(R)$, and any such concatenation is in each of those languages.

- (h) If two positive naturals n and m are relatively prime, one must be odd and the other even.
  FALSE (87% correct) – as an example, 3 and 5 are relatively prime. It was gratifying that most of you saw through this one.

- (i) If a DFA has no final states, then its language could possibly be infinite.
  FALSE (47% correct). The language is empty, and an empty set is finite.

- (j) Recall that the start configuration for a Turing machine M for input aba is $i\Box aba$, where i is the start state and $\Box$ is the blank character. Then if $\delta(i, \Box) = (p, a, R)$, $\delta(p, a) = (p, b, L)$, and $\delta(p, b) = (h, a, R)$, then $aba \in L(M)$.
  FALSE (40% correct) – The next configuration is $apaba$, the following one is $pabba$, and the machine then hangs because it moves left off of the tape. Since it does not halt, the input is not in $L(M)$. This depended on the exact definition of the behavior of a TM in the textbook. Some other texts, such as Sipser's used in COMPSCI 501, have the machine stay in place when it is told to move right, and in this case this machine *would* halt on this input. But we did have examples in lecture where the hanging behavior was noted.

Supplemental Sheet for COMPSCI 250 Final Exam, 13 May 2024

Recently, while their humans were away, Blaze and Rhonda stayed with a dog sitter, along with the sitter's dog Hawkeye. Three times during the visit (Friday, Saturday, and Sunday), the sitter emailed to report what the three dogs were doing. Let D be the set $\{B, H, R\}$ of the three dogs, T be the set $\{Fri, Sat, Sun\}$ of the reporting times, and A be a set $\{ba, di, lo, re\}$ of activities: barking, digging, lounging, and retrieving. Your goal is to determine all the values of the function f from $D \times T$ to A given by the three reports. You may assume that f is a function, so that each dog at each time is engaged in exactly one of the four possible activities in A.

We also define four propositions to be used in some of the statements:

p_1 means "$f(H, Fri) = re$"
p_2 means "$f(H, Sat) = re$"
p_3 means "$f(R, Fri) = re$"
p_4 means "$f(R, Sat) = re$"

Translate each statement as according to the directions:

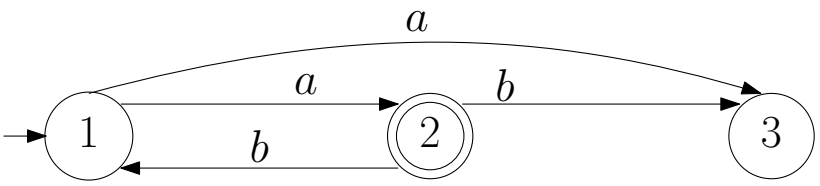
- (Statement I) If Hawkeye did not retrieve on Friday, then Hawkeye retrieved on Saturday and Rhonda retrieved on Friday.
- (Statement II) $[(f(H, Fri) = re) \vee (f(R, Sat) = re)] \leftrightarrow (f(H, Sat) = re) \wedge \neg((f(R, Fri) = re) \rightarrow (f(H, Sat) = re))$
- (Statement III) No activity was done by all three dogs, and if any activity was ever done by two different dogs, then that activity was retrieving.
- (Statement IV) $\forall a : [(a \neq re) \rightarrow \exists d : \exists t : \exists t' : (f(d, t) = f(d, t') = a) \wedge (t \neq t')]$
- (Statement V) On Sunday, Hawkeye barked and Rhonda lounged.

(more on the other side of the paper)

Supplemental Sheet for COMPSCI 250 Final Exam, 13 May 2024

Here is a diagram for the ordinary NFA N used in Question 6 parts (a)-(d).

N has state set $\{1, 2, 3\}$, with 1 the start state and 2 the only final state. The transitions are $(1, a, 2)$, $(1, a, 3)$, $(2, b, 1)$, and $(2, b, 3)$.



Here is a diagram for the λ-NFA M used in Question 6 part (e).

Let M be the λ-NFA with state set $\{p, q, r, s\}$, with p as the start state and q as the only final state, and transitions (p, a, q), (p, λ, r), (q, a, s), (r, λ, q), (s, λ, q), and (s, b, r).