CSE 311 : Autumn 2023 Final Exam, Form A Solutions

Name:	NetID: @uw.edu

Instructions

- You have one-hour and fifty-minutes to complete this exam.
- You are permitted one piece of 8.5x11 inch paper with handwritten notes (notes are allowed on both sides of the paper).
- You may not use a calculator or any other electronic devices during the exam.
- We will be scanning your exams before grading them. Please write legibly, and avoid writing up to the edge of the paper.
- **Problems are printed on both the front and back of each page!**
- You may also use the last page for extra space, but tell us where to find your answer if it's not right below the problem.
- If you want us to grade something you wrote on scratch paper, put your name and netid on the paper and tell us when you turn in your exam that you have an extra sheet.

Advice

- Remember to properly format English proofs (e.g. introduce all your variables).
- All proofs for this exam must be English proofs.
- We give partial credit for the beginning and end of a proof. Even if you don't know how the middle goes, you can write the start of the proof and put the "target" and conclusion at the bottom.
- Remember to take deep breaths.

Question	Max points
Training Wheels	14
First Proof	11
Models of Computation	13
Induction I	20
Short Answer	11
Induction II	20
Wait, That's Illegal	15
Grading Morale	1
Total	**105**

1. Training Wheels [14 points]

Let your domain of discourse be integers. Define the following predicates

- $\texttt{Divides}(x, y)$ returns true if and only if $x|y$.
- $\texttt{Even}(x)$ returns true if and only if x is even.
- $\texttt{Prime}(x)$ returns true if and only if x is prime.
- $\texttt{IsTwo}(x)$ returns true if and only if x is the number 2.
- You may also use standard arithmetic connectors like $=$, $<$, $\geq$, etc.

(a) Translate this sentence into predicate logic notation. "Strictly between a and b" means neither a nor b count. [3 points]

For every integer k, there is at least one prime integer j strictly between k and $3k$.

Solution:

$\forall k \exists j(\texttt{Prime}(j) \land k < j \land j < 3k)$

(b) Take the contrapositive of the following statement (leave your answer as symbols; negations must be applied to single predicates, but you do not need to reformat your answer to show domain restriction). [3 points]

$$\forall y(\texttt{IsTwo}(y) \rightarrow [\texttt{Even}(y) \land \texttt{Prime}(y)])$$

Solution:

$\forall y([\neg\,\texttt{Even}(y) \lor \neg\,\texttt{Prime}(y)] \rightarrow \neg\,\texttt{IsTwo}(y))$

(c) Translate this sentence into predicate logic notation. [3 points]

If x is even and x is prime then x is less than 3 and greater than 1.

Solution:

$\forall x([\texttt{Even}(x) \land \texttt{Prime}(x)] \rightarrow x < 3 \land x > 1)$

(d) Negate the statement in part (c) (not your answer). Give your answer in English. You must explicitly state all quantifiers (even if you might omit them in everyday English). Additionally "not"s must apply only to single predicates [3 points] **Solution:**

There is an integer x such that x is even and prime, but x is greater-than-or-equal-to 3 or x is less-than-or-equal-to 1.

(e) The (original) statement in (b) is [1 point]

◯ True
◯ False **Solution:**

True

(f) The contrapositive of the original statement in (b) has: [1 point]

◯ The same truth value as the original statement in part (b).

◯ The negation of the truth value of the original statement in part (b).
◯ It depends on what y is. **Solution:**

The same truth value

2. First Proof [11 points]

Let $A = \{n \in \mathbb{Z} : n \equiv 0 \pmod{30}\}$
Let $B = \{n \in \mathbb{Z} : 5 \mid n\}$
Let $C = \{n \in \mathbb{Z} : 3 \mid n\}$
In the notation above, ':' is "such that" and '|' is "divides."

(a) Prove that $A \subseteq B \cap C$ [7 points]

Solution:

Suppose a is an arbitrary element in the set $\{n \in \mathbb{Z} : n \equiv 0 \pmod{30}\}$.

We aim to prove that a belongs to the set $\{n \in \mathbb{Z} : 5 \mid n\} \cap \{n \in \mathbb{Z} : 3 \mid n\}$, i.e., a is divisible by both 5 and 3.

By definition of modular equivalence, $a \equiv 0 \pmod{30}$ means $30 \mid a$. Then, by definition of divides, we know $a = 30k$ for some integer k. We can then rewrite this as $a = 5(6k)$ or $a = 3(10k)$. By definition of divides, because $6k$ is an integer, $5 \mid a$, and likewise, because $10k$ is an integer, $3 \mid a$.

Thus, by definition of intersection, we can conclude that a is in the set $\{n \in \mathbb{Z} : 5 \mid n\} \cap \{n \in \mathbb{Z} : 3 \mid n\}$. Since a was arbitrary, the claim holds by definition of subset.

(b) Does the subset relationship hold in the other direction? That is, is this statement true?:

$$B \cap C \subseteq A$$

The statement is: [2 points]

◯ True
◯ False

Solution:

False. For example $15 \in B \cap C$, but not in A.

(c) Let D, E be sets. If D has eight elements, $D \cup E$ has nine elements, and $D \setminus E$ has four elements, then E must contain four elements. ***Hint**: It may helpful to visualize this by drawing a Venn Diagram*

The statement is: [2 points]

◯ True
◯ False

Solution:

False, $|E| = 5$.

3. Models of Computation [13 points]

Let L_1 be the language containing all strings over $\{0, 1\}$ such that it satisfies **both** of these two requirements:

- 0 cannot be immediately followed by a 1.
- The sum of all the digits in the string is even.

For example:

- 101 is not in L_1 (the 0 is followed by a 1)
- 00 is in L_1.
- ε is in L_1 (the sum is 0, which is even).

(a) Write a CFG that generates L_1. Be sure to tell us which symbol is the start symbol; also include a sentence or two of explanation of how your CFG works. [4 points]

Solution:

This string is actually just an even number of 1s followed by any number of 0s, since once a 0 appears no 1 can appear afterwards. S is the starting symbol.

$$\mathbf{S} \to \varepsilon \mid 11\mathbf{S} \mid \mathbf{S}0$$

(b) A specific computer only accepts passwords that satisfy the following requirements:

- The password can only use the digits 1, 2, 3
- The password must be at least 4 digits long.
- The password when interpreted as a base-10 number is even.

Write a regular expression for all valid passwords. [3 points]

Solution:

This expression has a requirement on the last digit of the number. It must be even in base 10, which means the last digit can only be 2 here. We must also enforce there are at least 4 digits in the number, which is achieved by adding 3 mandatory digits before the last digit.

$$(1 \cup 2 \cup 3)^*(1 \cup 2 \cup 3)(1 \cup 2 \cup 3)(1 \cup 2 \cup 3)2$$

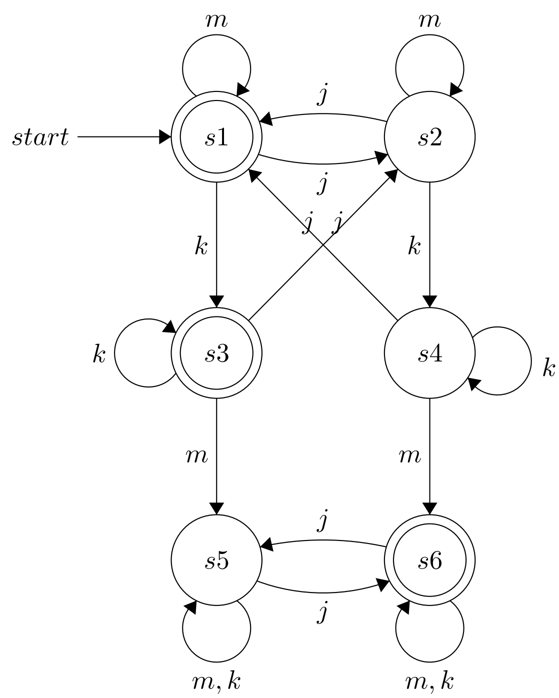
(c) Let L_2 be the language containing all strings over $\{j, k, m\}$ such that **either**

- the number of j's is even, **or**
- contains the subsequence km, i.e. there's a k immediately followed by an m.

but not both.

Draw a DFA that accepts L_2. Include a brief description of how your machine works (You'll probably want about 2-3 sentences total, or a few words for each state about what it does). [6 points]

Solution:



s_1 is the when there's even number of j's, and there's no sequence km, and the last letter is not k.
s_2 is the when there's even odd of j's, and there's no sequence km, and the last letter is not k.
s_3 is the when there's even number of j's, and the last letter is k.
s_4 is the when there's odd number of j's, and the last letter is k.
s_5 is the when there's even number of j's, and there's already the sequence km.
s_6 is the when there's odd of j's, and there's already the sequence km.

4. Induction I [20 points]

Consider the following recursive function:

$$f(n) = \begin{cases} 4 & n = 2 \\ 12 & n = 3 \\ \frac{f(n-1)\cdot n}{2(n-2)} + \frac{f(n-2)\cdot n(n-1)}{2(n-3)(n-2)} & n \geq 4 \end{cases}$$

Prove $f(n) = 2n(n-1)$ for all integers $n \geq 2$.

You **must** use induction; be sure to define a predicate $P()$ and use good style.

Hint: Don't distribute (i.e., FOIL) terms before you have to in your inductive step. Look for things that might cancel.

Solution:

Let $P(n)$:="$f(n) = 2n(n-1)$". We prove $P(n)$ holds for all integers $n \geq 2$ using induction.

Base Cases: $n = 2$
$f(2) = 4 = 2(2)(1) = 2(2)(2-1)$ which is $P(2)$.

$n = 3$
$f(3) = 12 = 2(2)(3) = 2(3)(3-1)$ which is $P(3)$.

Inductive Hypothesis: Suppose $P(k)$ holds for $P(2) \wedge \cdots \wedge P(k)$ for an arbitrary integer k ($k \geq 3$).

Inductive Step:

$$\begin{aligned} f(k+1) &= \frac{f(k)\cdot(k+1)}{2(k-1)} + \frac{f(k-1)\cdot(k+1)k}{2(k-2)(k-1)} && \text{by def of f since } k+1 \geq 4 \\ &= \frac{2(k)(k-1)(k+1)}{2(k-1)} + \frac{2(k-1)(k-2)\cdot(k+1)k}{2(k-2)(k-1)} && \text{IH} \\ &= \frac{2k(k+1)}{2} + \frac{2k(k+1)}{2} \\ &= 2k(k+1) \end{aligned}$$

Conclusion: We have shown $P(n)$ holds for all integers $n \geq 2$ by the principle of induction.

5. Short Answer [11 points]

(a) Write the first sentence of a proof by contradiction trying to prove "If a graph is bipartite, then it has no odd cycles." Make any quantifiers in your sentence explicit and be sure you're following good style for the first sentence of a proof by contradiction. (You do not need to know what 'bipartite' means to do this problem.) [3 points] **Solution:**

> Suppose, for the sake of contradiction, that there exists some bipartite graph G that has at least one odd cycle.

(b) Let L_1 be the language accepted by a DFA, and let L_2 be the language accepted by an NFA. What can we say about $L_1 \cap L_2$? [2 points]

◯ There is a DFA that accepts exactly $L_1 \cap L_2$.
◯ There might or might not be a DFA that accepts exactly $L_1 \cap L_2$.
◯ There cannot be a DFA that accepts exactly $L_1 \cap L_2$.

Solution:

> There is a DFA that accepts exactly $L_1 \cap L_2$.

(c) Which of the following describes what it means when we write $\forall x \exists y P(x, y)$[2 points]

◯ All the values (x) in our domain have the same value y that makes $P(x, y)$ true. (y cannot depend on x)
◯ All the values (x) in our domain have a value y that makes $P(x, y)$ true. (y can depend on x).
◯ There is a value (x) in our domain so that for all values y, $P(x, y)$ is true. **Solution:**

> Second option (All x have a y which may change with x).

6. Fruits and Veggies (Induction II) [20 points]

Consider binary trees that hold fruits and veggies which we'll call `produceTrees`.

Basis Step: `null` is a `produceTree`
Recursive Step 1: If L, R are `produceTrees` then $(L, \texttt{fruit}, R)$ is also a `produceTree`
Recursive Step 2: If L is a `produceTree` then $(L, \texttt{veggie}, \texttt{null})$ is also a `produceTree`.

Intuitively, Recursive Step 1 creates a `fruit` node with two children and Recursive Step 2 creates a `veggie` node with only one child.

Using **structural** induction, show that for all `produceTrees`, if they have f `fruits` and v `veggies` then they have $f + v + 1$ copies of `null`.

Solution:

Let $P(T)$ be "if `produceTree` T has f `fruits` and v `veggies` then it has $f + v + 1$ copies of `null`." We prove $P(T)$ holds for all `produceTrees` t by structural induction.

Base Case: The only basis step has the tree "`null`". `null` contains no data (so no fruit or veggies), has 1 copy of `null`, and $0 + 0 + 1 = 1$, so $P(\texttt{null})$ holds.

Inductive Hypothesis: Suppose $P(L)$ and $P(R)$ for some arbitrary `produceTrees` L and R.

Inductive Step:

Let f_L, v_L, and n_L be the number of `fruits`, `veggies`, and copies of `null` that L has, respectively. Likewise, let f_R, v_R, and n_R be the corresponding values for R, and f_T, v_T, and n_T for T where $T = (L, \texttt{veggie}, \texttt{null})$.

Case 1 - We show $P(T)$.
By the IH, $n_L = f_L + v_L + 1$. T has $v_T = v_L + 1$ `veggies` since its root node adds one, $f_T = f_L$ `fruits`, and $n_T = n_L + 1$ `nulls` as its right child adds one. Adding 1 to each side of the equation from the IH, we have $n_L + 1 = (v_L + 1) + f_l + 1$ which is equivalent to $n_T = f_T + v_T + 1$, so $P(T)$ holds.

Case 2 - We show $P(T)$ where $T = (L, \texttt{fruit}, R)$.
By the IH, $n_L = f_L + v_L + 1$ and $n_R = f_R + v_R + 1$. T has $v_T = v_L + v_R$ `veggies`, $f_T = f_L + f_R + 1$ `fruits` since $T's$ root node adds one, and $n_T = n_L + n_R$ `nulls`. Adding the two equations from the IH, we have $n_L + n_R = (f_L + f_R + 1) + (v_L + v_R) + 1$ which is equivalent to $n_T = f_T + v_T + 1$, so $P(T)$ holds.

We conclude $P(T)$ for all `produceTrees` T by structural induction.

7. Wait, that's illegal [15 points]

Choose to do exactly one of these two problems.

(a) Let $\Sigma = \{0, 1, \#\}$. I.e., our alphabet contains 0 and 1 and a special character #. Let L be the following language:

$$L = \{x : x = 0^k\#1^{2k}\}$$

That is, each string in L has a single # separating two substrings. The substring before the # is all 0's, the substring after the # is twice as many 1's.

- $00\#1111 \in L$
- $00\#111 \notin L$
- $0\#11 \in L$
- $\# \in L$
- $00\#11\#11 \notin L$

Prove that L is not regular. **Solution:**

Let $L = \{x : x = 0^k\#1^{2k}\}$. Suppose for the sake of contradiction that some DFA D accepts L.

Consider $S = \{0^n : n \geq 0\}$. Since S contains infinitely many strings and D has a finite number of states, two strings in S must end up in the same state.

Say these strings are $a = 0^i$ and $b = 0^j$ for some $i, j \geq 0$ such that $i \neq j$.

Append the string $c = \#1^{2i}$ to both of these strings. The two resulting strings are:

$ac = 0^i\#1^{2i}$ Note that $xz \in L$ since there are twice as many 1's.

$bc = 0^j\#1^{2i}$ Note that $yz \notin L$, since $i \neq j$, there are not twice as many 1's.

Since a and b arrive at the same state of D, ac and bc end up in the same state as well, but $ac \in L$ and $bc \notin L$, that state must be both an accept and reject state, which is a contradiction. Thus there is no DFA that recognizes L, so L is not regular.

(b) Let $\Sigma = \{3, 4, 5\}$. Define X to be the set of all functions with domain Σ^* and co-domain $\{0, 1, 2\}$.
For example, the function "given a string, return its length % 2" is an element of X

Prove, by recreating a diagonalization argument like that shown in class, that $\mathcal{P}(X)$ is uncountable. **Solution:**

Suppose for the sake of contradiction that $\mathcal{P}(X)$ is countable. Then there exists a bijection, f, between the natural numbers and $\mathcal{P}(X)$. Consider making a table as follows: the i^{th} row contains a representation of $f(i)$, the i^{th} set in the list created by the bijection.

To represent the function, pick any countably infinite subset of functions in X, and have a column for each function. In the entry at row i column j, record a 1 if function j is an element of set i, and 0 otherwise.

Now we build an element of $\mathcal{P}(X)$ which is not in the table. Beginning with the empty-set, build a set as follows: go to entry i, i in the table. If i, i is 0, then include function i in the set (otherwise do not).

This infinite process defines a subset of X (i.e., an element of $\mathcal{P}(X)$), but this set is not in the table, as for every row i, it disagrees with the set on whether the i^{th} function on the list is in the set. Thus we have a contradiction, and the set is not countable.