CS103 Practice Midterm Exam

This midterm exam is open-book, open-note, open-computer, but closed-network. This means that if you want to have your laptop with you when you take the exam, that's perfectly fine, but you **must not** use a network connection. You should only use your computer to look at notes you've downloaded in advance. Although you may use laptops, you **must** hand-write all of your solutions on this physical copy of the exam. No electronic submissions will be considered without prior consent of the course staff.

Normally, we would provide space on the exam for you to write your answers, but in the interest of saving paper we've eliminated most whitespace from this practice exam.

You have three hours to complete this midterm. There are 180 total points, and this midterm is worth 15% of your total grade in this course. The first three problems are shorter and simpler than the last three, so be sure to allocate your time appropriately. You may find it useful to read through all the questions to get a sense of what this midterm contains.

Good luck!

Question		**Points**	**Grader**
(1) First-Order Logic	(20)	/20	
(2) Finding Flaws in Proofs	(20)	/20	
(3) Finite Automata	(20)	/20	
(4) Utopian Tournament Graphs	(40)	/40	
(5) The Well-Ordering Principle	(40)	/40	
(6) Euclid's Algorithm	(40)	/40	
	(180)	**/180**	

Problem 1: Translating into Logic **(20 points total)**

In each of the following, you will be given a list of first-order predicates and functions along with an English sentence. In each case, write a statement in first-order logic that expresses the indicated sentence. The statement you write can use any first-order construct (equality, connectives, quantifiers, etc.), but you must only use the predicates and functions provided.

As an example, if you were given the predicate *Integer(x)*, which returns whether x is an integer, and the function *Plus(x, y)*, which returns x + y, you could write the statement "there is some even integer" as

$$\exists n.\ \exists k.\ (Integer(n) \land Integer(k) \land Plus(k, k) = n)$$

since this asserts that some number n is equal to $2k$ for integer k. However, you could not write

$$\exists n.\ (Integer(n) \land Even(n))$$

because there is no *Even* predicate.

(i) Never Gonna Give You Up **(5 Points)**

Given the predicate

Knows(x, y), which says that x and y know each other

and the constant symbols *me*, *you*, *love*, and *rules*, write a statement in first-order logic that says "We're no strangers to love. You know the rules, and so do I." You can assume that if x does not know y, then x and y are strangers.

(ii) Gotta Catch 'em All! **(5 Points)**

Given the predicates

WantsToBeBetterThan(x, y), which says that x wants to be better than y,
WasBetterThan(x, y), which says that x was, in the past, better than y.

and the constant symbol *me*, write a statement in first-order logic that says "I want to be the very best, like no one ever was." (That is, I want to be better than everyone else, and in the past no one was better than everyone else.)

iii) Good Advice **(10 Points)**

Given the predicates

> *Fools(x, y, t)*, which says that *x* fools *y* at time *t*,
> *Person(x)*, which says whether *x* is a person, and
> *Time(t)*, which says whether *t* is a time,

along with the constant *you*, write a statement in first-order logic that says "you can fool some people all the time or all the people some of the time, but not all the people all the time." To clarify, the statement "all the people some of the time" doesn't necessarily mean that there is some instant in time at which you can fool everyone; it just says that for any person, you can fool them at some point in time.

Problem 2: Finding Flaws in Proofs **(20 points)**

Consider the following modification of the *RS(x, y)* function from the second problem set:

$$RS^?(x,y)=\begin{cases} 1 & \text{if } y=0 \\ RS^?(x,\frac{y}{2})^2 & \text{if } y>0 \text{ and } y \text{ is even} \\ \frac{1}{x}RS^?(x,\frac{y+1}{2})^2 & \text{otherwise} \end{cases}$$

This function is *completely wrong* and, in most cases, does not correctly compute x^y. Below is a purported proof that this function does compute the correct value:

Theorem: $RS^?(x, y) = x^y$ for all $x \in \mathbb{R}$, $y \in \mathbb{N}$, where $x \neq 0$.

Proof: By strong induction. Let P(y) be "for all $x \in \mathbb{R}$, if $x \neq 0$, then $RS^?(x, y) = x^y$." We prove that P(y) is true for all $y \in \mathbb{N}$. As our base case, we prove P(0), that for any nonzero $x \in \mathbb{R}$, $RS^?(x, 0) = x^0$. Since $RS^?(x, 0) = 1 = x^0$ by definition, this is true.

For the inductive step, assume that for some y, for all natural numbers y' such that $0 \le y' \le y$, P(y') is true, so for any nonzero $x \in \mathbb{R}$, $RS^?(x, y') = x^{y'}$. We prove that P($y + 1$) is true, that for all nonzero $x \in \mathbb{R}$, $RS^?(x, y + 1) = x^{y+1}$. We consider two cases:

Case 1: $y + 1$ is even. Then $RS^?(x,y+1)=RS^?(x,\frac{y+1}{2})^2$. By the inductive hypothesis, $RS^?(x,\frac{y+1}{2})=x^{\frac{y+1}{2}}$, so $RS^?(x,y+1)=RS^?(x,\frac{y+1}{2})^2=(x^{\frac{y+1}{2}})^2=x^{y+1}$ as required.

Case 2: $y + 1$ is odd. Then $RS^?(x,y+1)=\frac{1}{x}RS^?(x,\frac{y+2}{2})^2$. By the inductive hypothesis, $RS^?(x,\frac{y+2}{2})=x^{\frac{y+2}{2}}$, so $RS^?(x,y+1)=\frac{1}{x}RS^?(x,\frac{y+2}{2})^2=\frac{1}{x}(x^{\frac{y+2}{2}})^2=\frac{1}{x}x^{y+2}=x^{y+1}$ as required.

Thus in either case $RS^?(x, y + 1) = x^{y+1}$, so P($y + 1$) is true, completing the proof by induction. ■

(i) Does Not Compute **(5 Points)**

The $RS^?$ function does not correctly compute x^y for most choices of x and y. Give an example of a choice of x and y where $x \neq 0$ and $RS^?(x, y)$ does not correctly compute x^y.

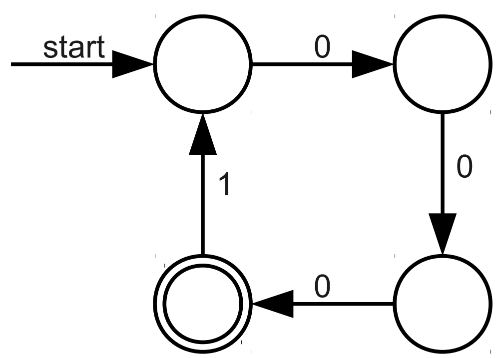
(ii) Your Argument is Invalid **(15 Points)**

The above proof is incorrect. What is wrong with its logic? It is **not enough** to simply state that the proof is incorrect or to give a counterexample; instead, cite the specific part of the proof that is incorrect and explain what logical error is being made.

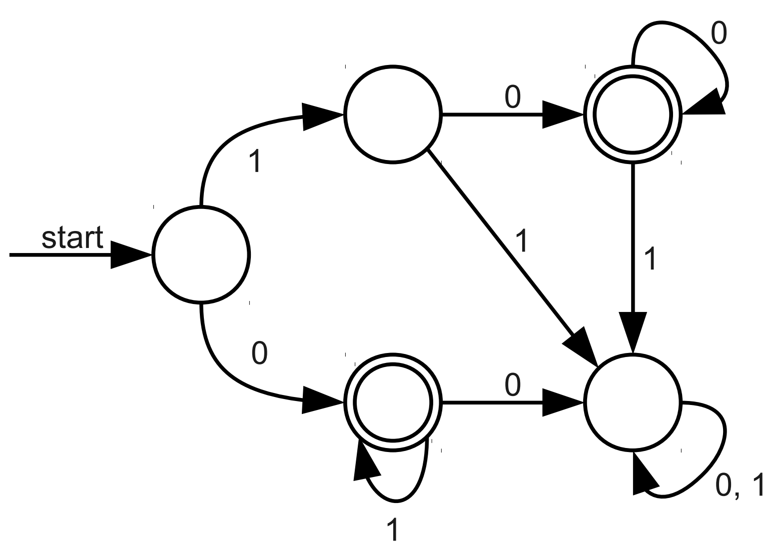
Problem 3: Finite Automata **(20 points total)**

Below are four finite automata, some of which are DFAs and some of which are not. For each automaton, state whether or not it is a DFA. If it is not, explain why that automaton is not a DFA. You do not need to provide an explanation if the automaton is a DFA. You may assume that the language is $\Sigma = \{0, 1\}$.

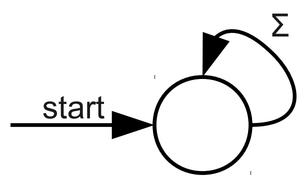
(i) The Preantepenultimate Automaton **(5 Points)**



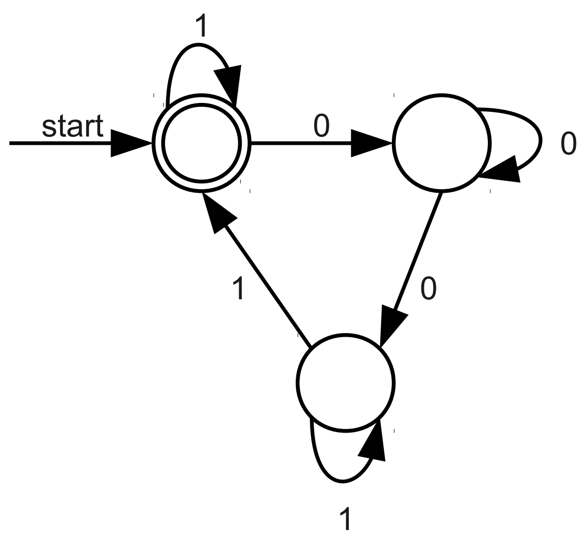
(ii) The Antepenultimate Automaton **(5 Points)**



(iii) The Penultimate Automaton **(5 Points)**



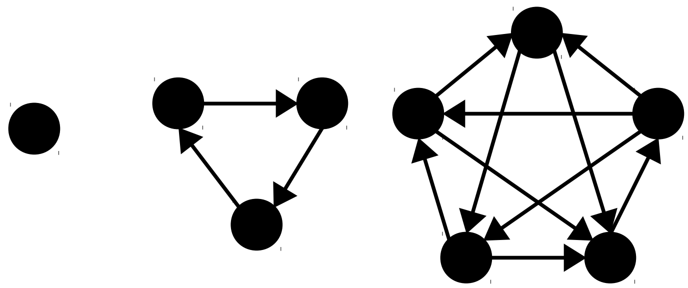
(iv) The Ultimate Automaton **(5 Points)**



Problem 4: Utopian Tournament Graphs **(40 points total)**

Recall from the second problem set that a *tournament graph* is a graph representing the outcome of a tournament with n > 0 players, in which each player plays each other player exactly once. Each game has a winner and a loser, and there are no draws. A tournament graph is a graph of the outcome of the tournament, where each node corresponds to a player and each edge (u, v) means that player u won her game against player v. In the second problem set, you proved that in any tournament graph, there is at least one tournament winner (a player who, for each other player, either won her game against that player, or won a game against someone who in turn beat that player).

It is possible to construct tournament graphs with more than one tournament winner, and in fact it's possible to construct tournament graphs where *everyone* is a winner. For example, here are tournament graphs with 1, 3, and 5 nodes where each player wins:



Prove that for any odd natural number *n*, there is at least one tournament graph for *n* players such that every player is a tournament winner.

Problem 5: The Well-Ordering Principle **(40 Points)**

On the third problem set, you explored the *well-ordering principle*, which states that any nonempty set of natural numbers, whether finite or infinite, contains some smallest natural number. Here, you will explore some other applications of the well-ordering principle.

Suppose that we have two ordered sets $(A, <_A)$ and $(B, <_B)$, where $<_A$ and $<_B$ are strict orders. A **homomorphism from A to B** is a function $f : A \to B$ with the property that

$$\forall a_1 \in A.\ \forall a_2 \in A.\ (a_1 <_A a_2 \to f(a_1) <_B f(a_2))$$

That is, if one element of A is less than some other element of A, then after applying *f* to both of those elements the image of the first element is still less than the image of the second element. If *f* is a homomorphism from A to B, we say that A is **homomorphic** to B.

Let $\mathbb{Z}^-$ be the set of negative integers. That is, $\mathbb{Z}^- = \{-1, -2, -3, \ldots\}$.

(i) Integer Homomorphisms **(10 Points)**

Consider the set $\mathbb{Z}^-$ ordered by the greater-than relation >. Show that $(\mathbb{Z}^-, >)$ is homomorphic to $(\mathbb{N}, <)$ by giving an example of a function $f : \mathbb{Z}^- \to \mathbb{N}$ such that if z_0 and z_1 are negative integers with $z_0 > z_1$, then $f(z_0) < f(z_1)$. Then prove that your function *f* is a homomorphism.

Suppose that (S, $<_S$) is a strictly, totally ordered set; that is, $<_S$ is a strict total order over S. Given a nonempty subset $T \subseteq S$, we say that some element $t \in T$ is a **least element** of T if for all other $t' \in T$, it is true that $t <_S t'$.

(ii) Homomorphisms and Well-Orderings (10 Points)

Suppose that some strictly, totally ordered set (S, $<_S$) is homomorphic to ($\mathbb{N}$, <). Prove that any nonempty subset of S contains a leastz element.

(iii) Well-Ordering and Induction (20 Points)

Suppose that you have some property P(n) where:

- P(0)
- $\forall n \in \mathbb{N}. (P(n) \rightarrow P(n + 1))$

Prove, using the well-ordering principle, that P(n) is true for all natural numbers n. *(Hint: Suppose that P(n) is not true for all natural numbers n, and consider the set of natural numbers for which it is false.)*

Problem 6: Euclid's Algorithm (40 points total)

For any pair of integers x and y, a number d is a *common divisor* of x and y if d divides x and d divides y. That is, there are integers m and n such that $x = md$ and $y = nd$. If either $x \neq 0$ or $y \neq 0$, then the *greatest common divisor* of x and y is the largest number d that is a common divisor of x and y.

One of the oldest known algorithms is *Euclid's algorithm*, which is used to find the greatest common divisor of two integers. Euclid's algorithm is sometimes employed in RSA cryptography, which needs to search for numbers whose greater common divisor is 1. In this problem, you will explore Euclid's algorithm and will formally prove its correctness.

(i) Same Difference (5 Points)

Prove that if d is a common divisor of x and y, then d is a divisor of $ax + by$ for any integers a and b.

(ii) The Division Algorithm **(5 Points)**

Recall from lecture that the *division algorithm* says that for any integers x and y, with $y \neq 0$, that x can be written as $x = qy + r$ for integers q and r such that $0 \leq r < y$.

Prove that d is a common divisor of x and y iff it is a common divisor of y and r.

The result you've just proven shows that the set of divisors of x and y is the same as the set of divisors of y and r (if $y \neq 0$). This means that the greatest common divisor of x and y must be the same as the greatest common divisor of y and r. Euclid's algorithm, which dates back almost 2300 years, is based on this principle. The algorithm is defined as follows:

$$gcd(x, y) = \begin{cases} x & \text{if } y = 0 \\ gcd(y, r) & \text{otherwise}, \end{cases}$$ where r is found by the division algorithm

In the remainder of this problem, you will prove that Euclid's algorithm is correct for all natural numbers x and y (except the special case where $x = 0$ and $y = 0$, where the greatest common divisor is not defined).

(iii) Proving the Base Case **(10 Points)**

Prove that the greatest common divisor of x and 0 is x for any nonzero natural number x.

(iv) Verifying the Algorithm **(20 Points)**

Prove that *gcd(x, y)* returns the greatest common divisor of x and y, assuming that x and y are natural numbers and either $x \neq 0$ or $y \neq 0$. As a hint, try using strong induction on y.