EECS 1019/1090 – Predicate Logic - Informal Understanding & Proofs Practice

Instructor: Jeff Edmonds

Not to be handed in.

The oracle game is simply
- Prove $\exists x$: Let x be ... I must construct it.
- Prove $\forall y$: Let y be arbitrary give to me by an adversary (i.e. worst case. Not random. Not benevolent.)
- Assume $\exists x$: Assume some oracle can give an object x to you and she goes on to assures you that the statement about it is true.
- Assume $\forall y$: You can give any object y to the oracle and she will assure it is true for that y.

I have not seen a single proof in the book - even the proofs by contradiction that I don't think are better when proved this way.
Challenge: Find me a proof that you like better.
My recommendation is that you don't fight purple table or the oracle game.

1. Multiple Choice. Which sentence relates best to the given English?
   (a) Some: a) $\forall x$ I-can-have(x); b) **(Ans)** $\exists x$ I-can-have(x); c) other
   (b) Can I have anything I want? a) **(Ans)** $\forall x$ I-can-have(x); b) $\exists x$ I-can-have(x); c) other
   (c) Is there anything that I can have? a) $\forall x$ I-can-have(x); b) **(Ans)** $\exists x$ I-can-have(x); c) other
   (d) Everyone is married: a) **(Ans)** $\forall x\ \exists y\ loves(x,y)$; b) $\exists y\ \forall x\ loves(x,y)$; c) other
   (e) Everyone loves her: a) $\forall x\ \exists y\ loves(x,y)$; b) **(Ans)** $\exists y\ \forall x\ loves(x,y)$; c) other
   (f) Every real has an inverse: a) **(Ans)** $\forall x\ \exists y$; b) $\exists y\ \forall x$; c) other
   (g) Every Hindu has God: a) **(Ans)** $\forall x\ \exists y$; b) $\exists y\ \forall x$; c) other
   (h) Every Christian has God: a) $\forall x\ \exists y$; b) **(Ans)** $\exists y\ \forall x$; c) other
   (i) Only Jeff would: a) **(Ans)** $would(\text{Jeff}) \wedge \forall x \neq \text{Jeff}\ \neg would(x)$; b) $would(\text{Jeff}) \wedge \exists x \neq \text{Jeff}\ \neg would(x)$; c) $would(\text{Jeff}) \vee \forall x \neq \text{Jeff}\ negwould(x)$; d) other
   (j) Jeff is not alone in that: a) $would(\text{Jeff}) \wedge \forall x \neq \text{Jeff}\ \neg would(x)$;
   b) $would(\text{Jeff}) \wedge \exists x \neq \text{Jeff}\ \neg would(x)$; c) **(Ans)** $would(\text{Jeff}) \wedge \exists x \neq \text{Jeff}\ would(x)$; d) other
   (k) Would Jeff or anybody do the dishes: a) $would(\text{Jeff}) \wedge \forall x \neq \text{Jeff}\ \neg would(x)$; b) $would(\text{Jeff}) \vee \forall x \neq \text{Jeff}\ would(x)$; c) $would(\text{Jeff}) \vee \exists x \neq \text{Jeff}\ would(x)$; d) $\exists x\ would(x)$; e) **(Ans)** c & d

2. For each prove whether true or not when each variable is a real value. Be sure to play the correct game as to who is providing what value.

   a) $\forall x\ \exists y\ x+y=5$ b) $\exists y\ \forall x\ x+y=5$
   c) $\forall x\ \exists y\ x \cdot y=5$ d) $\exists y\ \forall x\ x \cdot y=5$
   e) $\forall x\ \exists y\ x \cdot y=0$ f) $\exists y\ \forall x\ x \cdot y=0$
   g) $\forall y,\ \exists x,\ y=2x+1$ h) Same statement as (g) except over integers.
   i) $\exists x,\ \forall y,\ y+x>y$ j) $\exists x,\ \forall y,\ y+x>2y$
   k) $\forall a\ \exists y\ \forall x\ x \cdot (y+a)=0$ l) $\exists a\ \forall x\ \exists y\ [x=a \text{ or } x \cdot y=5]$
   m) $\forall x\ \exists y\ x+2xy+7y \neq 0$. Hint: First try $y=0$ and eliminate cases for x.
   n) $\forall x,\ \exists y,\ y=x^2$ o) $\forall y,\ \exists x,\ y=x^2$
   p) $[\forall x\ \exists y\ P(x,y)] \Rightarrow [\exists y\ \forall x\ P(x,y)]$ q) $[\forall x\ \exists y\ P(x,y)] \Leftarrow [\exists y\ \forall x\ P(x,y)]$

   - Answer:
     (a) $\forall x\ \exists y\ x+y=5$ is true. Let x be an arbitrary real value. Let $y=5-x$. Then $x+y=5$.
     (b) $\exists y\ \forall x\ x+y=5$ is false. Let y be an arbitrary real value. Let $x=6-y$. Then $x+y\neq 5$.
     (c) $\forall x\ \exists y\ x \cdot y=5$ is false. We prove that its negation $\exists x\ \forall y\ x \cdot y \neq 5$ is true. Let $x=0$. Let y be arbitrary. Then $x \cdot y = 0 \cdot y = 0 \neq 5$.
     (d) $\exists y\ \forall x\ x \cdot y=5$ is false. We prove that its negation $\forall y\ \exists x\ x \cdot y \neq 5$ is true. Let y be arbitrary. Let $x=0$. Then $x \cdot y = 0 \cdot y = 0 \neq 5$.

(e) $\forall x\ \exists y\ x \cdot y = 0$ is true. Let x be an arbitrary real value. Let $y = 0$. Then $x \cdot y = x \cdot 0 = 0$.

(f) $\exists y\ \forall x\ x \cdot y = 0$ is true. Let $y = 0$. Let x be an arbitrary real value. Then $x \cdot y = x \cdot 0 = 0$.

(g) $\forall y,\ \exists x,\ y = 2x+1$ is true over the reals. Let y be arbitrary. Let x be $\frac{1}{2}(y-1)$. Then $2x+1 = 2[\frac{1}{2}(y-1)]+1 = y$.

(h) $\forall y,\ \exists x,\ y=2x+1$ is false over the integers. We prove that its negation $\exists y \forall x\,(\,y\ -\ 2x\ \neq\ 1\,)$ is true. Let $y = 0$. Let x be an arbitrary integer. Note that $y\ -\ 2x$ is even but 1 is odd and hence they are not equal.

(i) $\exists x,\ \forall y,\ y + x > y$ is true. Let x be 1. Let y be arbitrary. Then $y + x = y + 1 > y$.

(j) $\exists x,\ \forall y,\ y+x>2y$ is false. We prove the negation $\forall x,\ \exists y,\ y+x\leq 2y$. Let x be arbitrary. Let y be x. Then $y+x = x+x = 2y$.

(k) $\forall a\ \exists y\ \forall x\ x \cdot (y + a) = 0$ is true. Let $a = 0$ be an arbitrary real value. Let $y = -a$. . Let x be an arbitrary real value. Then $x \cdot (y + a) = x \cdot ((-a) + a) = x \cdot 0 = 0$.

(l) $\exists a\ \forall x\ \exists y\ [x = a$ or $x \cdot y = 5]$ is true. Let $a = 0$. Let x be an arbitrary real value. If $x{\neq}0$, let $y = \frac{1}{x}$. Otherwise, let $y = 5$. Either way $[x = a$ or $x \cdot y = 5]$ is true.

(m) $\forall x\ \exists y\ x + 2xy + 7y \neq 0$ is true. Let x be an arbitrary real value. If $x \neq 0$, let $y = 0$. In this case, $x + 2xy + 7y = x + 2x \cdot 0 + 7 \cdot 0 = x \neq 0$. If $x = 0$, let $y = 1$. In this case, $x + 2xy + 7y = 0 + 2 \cdot 0 \cdot 1 + 7 \cdot 1 = 7 \neq 0$.

(n) $\forall x,\ \exists y,\ y = x^2$ is true. Let x be arbitrary. Let y be x^2. Well that was easy.

(o) $\forall y,\ \exists x,\ y = x^2$ is false. We prove the negation $\exists y,\ \forall x,\ y \neq x^2$. Let y be -1. Let x be arbitrary. Whether x is positive or negative, we have that x^2 is positive and hence can't be $y{=}{-1}$.

(p) $[\forall x\ \exists y\ P(x, y)] \Rightarrow [\exists y\ \forall x\ P(x, y)]$ is false. Let $P(x, y) = [x + y = 5]$. Then as seen above the first is true and the second is false.

(q) $[\forall x\ \exists y\ P(x, y)] \Leftarrow [\exists y\ \forall x\ P(x, y)]$ is true. Assume the right is true. Let y_0 the that for which $[\forall x\ P(x, y_0)]$ is true. We prove the left as follows. Let x be an arbitrary real value and let $y = y_0$. Then $P(x, y_0)$ is true.

3. Quantifiers over the reals.

(a) What does the sentence $\forall x\ \exists y\ [x{\times}y{=}1]$ mean?
Do NOT say "Forall x, there is a y, ... — Zero marks.
Instead, what property does it attribute to real values x?
Is the statement true?
If not what is a counter example?
Are there more than one counter examples?

- Answer: It says that every real value x has a multiplicative inverse $\frac{1}{x}$.
  It is not true because it is not true for the single counter example $x = 0$.

(b) What does the sentence $\exists a\ \forall x\ \exists y\ [[x{=}a] \vee [x{\times}y{=}1]]$ say about the real values a?
Is it true? If so, what is a?
Hint: The OR/$\vee$ part may be particularly confusing. Ignore it.
Guess. What key role do you think a will play in the discussion we are having?

- Answer: It is true. It says that every real value x has a multiplicative inverse except for the one counter example $a = 0$.

(c) Prove that the sentence is true under the reals:
$\exists a\ \forall x\ \exists y\ [[x{=}a] \vee [x{\times}y{=}1]]$
Don't panic. Just play the game. Who gives which objects and in what order?
Hint: There will be two cases. Say "If case ...., then ...., else, ...".
Hint: Use the notation $c_\forall$ and $c_\exists$.
Hint: If you don't know which object to give, I personally like the object 5.

- Answer: Prove $\exists a\ \forall x\ \exists y\ [[x{=}a] \vee [x{\times}y{=}1]]$
  Let $a_\exists = 0$ (constructed by the prover)
  Let $x_\forall$ be an arbitrary real number (given to us by the adversary)
  If $x_\forall \neq 0$, then let $y_\exists = \frac{1}{x_\forall}$
  Else let $y_\exists = 5$ (a fine number constructed by the prover)
  Now we need to verify $[[x_\forall{=}a_\exists] \vee [x_\forall{\times}y_\exists{=}1]]$
  If $x_\forall \neq 0$, then $x_\forall \times y_\exists = x_\forall \times \frac{1}{x_\forall} = 1$
  Else $x_\forall = 0 = a_\exists$
  Either way the OR is true.
  Prover can always win. Hence, the statement is true.
  "Every real value x has a multiplicative inverse except for the one counter example $a = 0$."

4. Define $Even(x) \equiv [\exists k,\ x = 2k]$. Prove $\forall x,\ [Even(x) \rightarrow Even(x+2)]$

   - Answer:

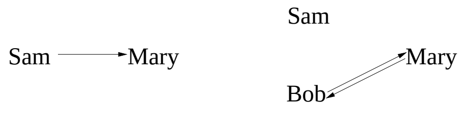
1) Let x be arbitrary.	
2) Deduction Goal: $Even(x) \rightarrow Even(x+2)$	
3) $\quad Even(x)$	Assumption/Premise
4) $\quad \exists k,\ x = 2k$	Definition
5) $\quad x = 2k_\exists$	That assumed to exists in 4
6) $\quad$ Let $k'_\exists = k_\exists + 1$	
7) $\quad (x+2) = 2k_\exists + 2 = 2(k_\exists + 1) = 2k'_\exists$	
8) $\quad Even(x+2)$	Definition
9) $Even(x) \rightarrow Even(x+2)$	Conclude deduction.

5. Let $Loves(b, g)$ denote that boy b loves girl g. If Sam loves Mary and Mary does not love Sam back, then we say that "Sam loves in vain."

   (a) Express the following statements using universal and existential quantifiers. Move any negations to the right.

      i. "Sam has loved in vain."
      ii. "There is a boy who has loved in vain."
      iii. "Every boy has loved in vain."
      iv. "No boy has loved in vain."

   (b) For each of the above statements and each of the two relations below either prove that the statement is true for the relation or that it is false.



   (c) Use $Loves(p_1, p_2, t)$ to add in the time at which it is occurring. Express the statement "Everyone has loved in vain at some point."

      - Answer: $\forall p_1\ \exists p_2\ \exists t\ Loves(p_1, p_2, t)$ and not $Loves(p_2, p_1, t)$

   (d) Sketch how the game would go to prove this statement.

      - Answer: The adversary supplies p_1. I supply p_2 and t. Then I prove $Loves(p_1, p_2, t)$ and not $Loves(p_2, p_1, t)$.

6. Here is a question the book proved by contradiction. Jeff NOT does like proof by contradiction because it turns things around unnecessarily. Also it is not constructive.
   You are now to prove it using the prover adversary game.

Prove that every set $S = \{a_1, a_2, \ldots, a_n\}$ of numbers contains a number that is at least the average of the numbers, namely
$\forall$ *sets* S, $\exists x \in S, x \geq a_{avg}$.
Hint: Let $a_{avg} = \frac{1}{n}[a_1+a_2+\ldots+a_n]$ denote the average.
Hint: Let a_{max} denote the maximum in the set, i.e., that which is at least as big as each of the a_i.
Hint: Just play the game. There are three clear lines in the proof. What are they?
4 sentences

- Answer:
  1) Let S be an arbitrary set of values.
  2) Let x be a_{max}, its maximum value.
  3) $a_{avg} = \frac{1}{m}[a_1+a_2+\ldots+a_n] \leq \frac{1}{n}[a_{max}+a_{max}+\ldots+a_{max}] = \frac{1}{n} \times n \times a_{max} = a_{max}$
  Since the prover wins, it must be true.

7. Minimum Value (Forall Exists Game): Let $S^+ = \{x \in Reals \mid x > 0\}$ denote the set of positive reals.

   (a) Which is true about the minimum value in S^+?
      i. It is like 0.00001 but has an infinite description.
      ii. It is zero.
      iii. **(Ans)** It does not exist.
      iv. It is 1.

   (b) Which of these say something different?
      i. $\exists x \in S^+,\ \forall y \in S^+, x \leq y$.
      ii. **(Ans)** $\forall x \in S^+, \exists y \in S^+, y < x$.
      iii. x is the minimum in S^+.
      iv. x is in S^+ and other values in S^+ are bigger.
      v. They all say the same thing.

   (c) Let $S^+ = \{x \in Reals \mid x > 0\}$ denote the set of positive reals. Explain what the following means and give a game proof of it: $\forall x \in S^+, \exists y \in S^+, y < x$.
   5 sentences

   - Answer: Each x fails to be the minimum in S^+ because there is a y in S^+ that is smaller. Hence, the set S^+ has no minimum x.
     Proof:
     – Let x be an arbitrary value in S^+.
     – Let $y = \frac{1}{2}x$,
     – Note $y \in S^+$ and $y < x$.
     – Hence x (and all values) is not the minimum value in S^+.

8. Let A denote the sentence $\exists x \forall y P(x,y)$ and let B denote $\exists y \forall x \neg P(x,y)$.
   Our goal is to either construct a P for which for which both are true,
   or to prove that for every evaluation of the relation P, they can't both be true.

   (a) As one does for "proof by contradiction", lets start by assuming that A, i.e., $\exists x \forall y P(x,y)$ is true.
   Let's determine what this says about P.
   Assume that you have an oracle A that assures you that it is true.
   Remember Jeff's oracle game.
   - What are you allowed to give her? What then does she assure?
   - What will she give you? What then does she assure?
   Use the notation $c_{\langle A,\forall\rangle}$ and $c_{\langle A,\exists\rangle}$.

   - Answer: With $\exists x \forall y P(x,y)$, she is assuring $\exists x$. Hence, she gives you such an object, which we will denote $x_{\langle A,\exists\rangle}$ for which she assures $\forall y P(x_{\langle A,\exists\rangle}, y)$. Now she is assuring $\forall y$. Hence, you can give her any object, which we will denote $y_{\langle A,\forall\rangle}$ for which she assures $P(x_{\langle A,\exists\rangle}, y_{\langle A,\forall\rangle})$.

(b) Fill in as much of the first table that you know.
In the first row and column, put in the names of the objects $c_{\langle A,\forall\rangle}/c_{\langle A,\exists\rangle}$ that you know.

- Answer:

A	x_1		$x_{\langle A,\exists\rangle}$	
y_1			T	
			T	
$y_{\langle A,\forall\rangle} = y_{\langle B,\exists\rangle}$			T	
			T	

B	x_1		$x_{\langle B,\forall\rangle} = x_{\langle A,\exists\rangle}$	
y_1				
$y_{\langle B,\exists\rangle}$	F	F	F	F

(c) Now assume that you have an oracle B that assures you that $\exists y \forall x \neg P(x,y)$ is true.
Repeat the previous two questions. Use the second table this time.

- Answer: With $\exists y \forall x \neg P(x,y)$, she is assuring $\exists y$. Hence, she gives you such an object, which we will denote $y_{\langle B,\exists\rangle}$ for which she assures $\forall x \neg P(x, y_{\langle B,\exists\rangle})$. Now she is assuring $\forall x$. Hence, you can give her any object, which we will denote $x_{\langle B,\forall\rangle}$ for which she assures $\neg P(x_{\langle B,\forall\rangle}, y_{\langle B,\exists\rangle})$.

(d) Can both of these statements be true at the same time? Yes or No?
If Yes, your tables should be showing such an example.
If No, reveal to us a contradiction.
Use the oracles again. Give and receive objects from them, until oracle A assures us of some fact and oracle B assures us of a different fact and these contradict each other, i.e., $\beta \wedge \neg\beta$.

- Answer: No. They cannot both be true.
By way of contradiction assume that both are true.
We begin with the first half of each of the previous games.
Oracle A gives you $x_{\langle A,\exists\rangle}$ and Oracle B gives you $y_{\langle B,\exists\rangle}$.
Oracle A is assuring that $\forall y P(x_{\langle A,\exists\rangle}, y)$. Hence, you can give her any object $y_{\langle A,\forall\rangle}$. You give her $y_{\langle B,\exists\rangle}$. She then assures $P(x_{\langle A,\exists\rangle}, y_{\langle B,\exists\rangle})$.
Oracle B is assuring that $\forall x \neg P(x, y_{\langle B,\exists\rangle})$. Hence, you can give her any object $x_{\langle B,\forall\rangle}$. You give her $x_{\langle A,\exists\rangle}$. She then assures $\neg P(x_{\langle A,\exists\rangle}, y_{\langle B,\exists\rangle})$.
These $P(x_{\langle A,\exists\rangle}, y_{\langle B,\exists\rangle})$ and $\neg P(x_{\langle A,\exists\rangle}, y_{\langle B,\exists\rangle})$ contradict each other.
Hence, the sentence $[\exists x \forall y P(x,y)] \wedge [\exists y \forall x \neg P(x,y)]$ can never be true.

(e) Redo your previous proof more formally without mention of oracles.

- Answer: No. They cannot both be true.

0)	Proof by contradiction goal: $\neg(A \wedge B)$	
1)	$A \wedge B$	assumption/premise
2)	$\exists x \forall y P(x,y)$	Separating And (1)
3)	$\exists y \forall x \neg P(x,y)$	
4)	$\forall y P(x_{\langle A,\exists\rangle}, y)$	Names the value of x claimed to exist in (2)
5)	$\forall x \neg P(x, y_{\langle B,\exists\rangle})$	Names the value of y claimed to exist in (3)
6)	$P(x_{\langle A,\exists\rangle}, y_{\langle B,\exists\rangle}))$	(4) true forall y so true for this one
7)	$\neg P(x_{\langle A,\exists\rangle}, y_{\langle B,\exists\rangle})$	(5) true forall x so true for this one
8)	$P(x_{\langle A,\exists\rangle}, y_{\langle B,\exists\rangle})) \wedge \neg P(x_{\langle A,\exists\rangle}, y_{\langle B,\exists\rangle})$	Build And (6) & (7)
9)	Contradiction	Excluded Middle (8)
10)	By way of contradiction, they cannot both be true.	

9. A computational problem is a function P from inputs I to required output $P(I)$. An algorithm is a function A from inputs I to actual output $A(I)$.

(a) Use universal and existential quantifiers to express the following (in standard form).

i. A computes P

- Answer: $\forall I \; A(I) = P(I)$

ii. A does not computes P

- Answer: $\exists I\ A(I) \neq P(I)$

iii. P is computable

- Answer: $\exists A \forall I\ A(I) = P(I)$

iv. P is not computable

- Answer: $\forall A \exists I\ A(I) \neq P(I)$

(b) How does the adversarial game proceed to prove that P is not computable?

- Answer: The adversary gives you an algorithm A that he claims solves the problem. You as the prover give an input I that you think is hard for his algorithm. You prove that his program does not work for your input, i.e. $A(I) \neq P(I)$.

(c) P is computable

- Answer: You as the prover give an algorithm A that you claim solves the problem. The adversary gives you an input I that he thinks is hard for your algorithm. You prove that your program works for this input, i.e. $A(I) = P(I)$.

(d) The computational problem $HALT$ is an uncomputable problem. (It asks if I, when viewed as a program, halts on input 0.) Prove the following true or false $\forall I \exists A\ A(I) = HALT(I)$.

- Answer: It is true. Consider any input I. I is some fixed string. HALT(I) has some fixed output. Let A be the algorithm that does not look at the input but simply outputs the answer $HALT(I)$. This algorithm gives the correct answer on input I.

10. A computational problem is a function P from inputs I to required output $P(I)$. An algorithm is a function A from inputs I to actual output $A(I)$. $Time(A, I)$ gives the running time for algorithm A on input I. Use universal and existential quantifiers (in standard form) to express the following. How does the prover-adversary game proceed to prove this statement.

(a) P is computable in time $3n^2$.

- Answer: $\exists A\ \forall I\ \left[A(I) = P(I)\ \textit{and}\ Time(A, I) \leq 3|I|^2\right]$

(b) P is computable in polynomial time.

- Answer: $\exists c\ \exists A\ \forall I\ [A(I) = P(I)\ \textit{and}\ Time(A, I) \leq |I|^c]$
  The prover provides a c and an algorithm A. The adversary provides an input I. The prover wins if A give the correct answer and runs takes at most time $|I|^c$.

(c) P is not computable in polynomial time.

- Answer: $\forall c\ \forall A\ \exists I\ [A(I) \neq P(I)\ \textit{or}\ Time(A, I) > |I|^c]$
  The adversary provides a c and an algorithm A. You must find an input I for which either A give the wrong answer or runs slower than $|I|^c$.

(d) Let P be the Halting problem. Note it is uncomputable. Argue whether or not the following is true. $\exists I\ \forall A\ A(I) \neq HALTING(I)$

- Answer: It is not true. We argued in class that the complement $\forall I\ \exists A\ A(I) = HALTING(I)$ is true. The reason is that the algorithm that always says "Yes" solves the problem for I if $P(I)$="Yes". Otherwise, the algorithm that always says "No" solves it.

(e) Use universal and existential quantifiers to express the following. The computational class "Exponential Time" is strictly bigger than the computational class "Polynomial Time," i.e. something can be done in "Exponential Time" that cannot be done in "Polynomial Time." (Be careful how you use c and n.)

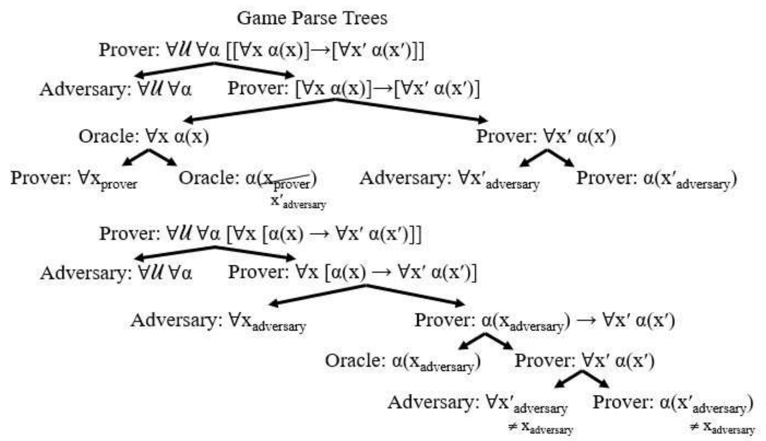
- Answer: $\exists P\ \left[\exists c > 0\ \exists A\ \forall I\ A(I) = P(I)\ \textit{and}\ Time(A, I) \leq 2^{c|I|}\right]$ and $[\forall c\ \forall A\ \exists I\ A(I) \neq P(I)\ \textit{or}\ Time(A, I) > |I|^c]$

11. The formal definition of $f(n) \in n^{\Theta(1)}$ is $\exists c_1, c_2, n_0,\ \forall n \geq n_0,\ n^{c_1} \leq f(n) \leq n^{c_2}$. Formally prove that if $f(n)$ and $g(n) \in n^{\Theta(1)}$, then $f(n) \cdot g(n) \in n^{\Theta(1)}$. What is this class of functions called?

   - Answer: These functions are polynomials.
     It includes $f(n) = n^2 logn$ because it is bounded between n^2 and n^3.
     Because $f(n)$ and $g(n) \in n^{\Theta(1)}$, we know that
     $\exists c_{\langle f,1\rangle}, c_{\langle f,2\rangle}, n_{\langle f,0\rangle},\ \forall n \geq n_{\langle f,0\rangle},\ n^{c_{\langle f,1\rangle}} \leq f(n) \leq n^{c_{\langle f,2\rangle}}$
     $\exists c_{\langle g,1\rangle}, c_{\langle g,2\rangle}, n_{\langle g,0\rangle},\ \forall n \geq n_{\langle g,0\rangle},\ n^{c_{\langle g,1\rangle}} \leq g(n) \leq n^{c_{\langle g,2\rangle}}$
     Consider some such constants $c_{\langle f,1\rangle}, c_{\langle f,2\rangle}, n_{\langle f,0\rangle}, c_{\langle g,1\rangle}, c_{\langle g,2\rangle}$, and $n_{\langle g,0\rangle}$.
     Our goal is to now prove $f(n) \cdot g(n) \in n^{\Theta(1)}$, i.e.
     $\exists c_1, c_2, n_0,\ \forall n \geq n_0,\ n^{c_1} \leq f(n) \cdot g(n) \leq n^{c_2}$.
     Let $c_1 = c_{\langle f,1\rangle} + c_{\langle g,1\rangle}$, $c_2 = c_{\langle f,2\rangle} + c_{\langle g,2\rangle}$, and $n_0 = \max(n_{\langle f,0\rangle}, n_{\langle g,0\rangle})$.
     Let n any value (given by an adversary) such that $n \geq n_0 = \max(n_{\langle f,0\rangle}, n_{\langle g,0\rangle})$.
     Because $n \geq n_{f,0}$, we have that $n^{c_{f,1}} \leq f(n) \leq n^{c_{f,2}}$
     Because $n \geq n_{g,0}$, we have that $n^{c_{g,1}} \leq g(n) \leq n^{c_{g,2}}$
     Because these are all positive, we can multiply the above two lines together giving
     $n^{c_1} = n^{c_{\langle f,1\rangle}} \cdot n^{c_{\langle g,1\rangle}} \leq f(n) \cdot g(n) \leq n^{c_{\langle f,2\rangle}} \cdot n^{c_{\langle g,2\rangle}} = n^{c_2}$.

12. Building the parse tree for $[\forall x\ \alpha(x)] \rightarrow [\forall x'\ \alpha(x')]$ and for $\forall x\ [\alpha(x) \rightarrow \forall x'\ \alpha(x')]$. Play the proof game for each.

   - Answer:



   - For each of the following either play the logic game to prove the statement is true or find a counter example, i.e. a table $\alpha(x, y)$ in which the left hand side is true and the right hand side is false.
     (a) $\forall x\ \alpha(x) \rightarrow \exists y\ \alpha(y)$
     (b) $\exists y\ \alpha(y) \rightarrow \forall x\ \alpha(x)$
     (c) $\exists y \forall x\ \alpha(x, y) \rightarrow \exists y'\ \alpha(y', y')$
     (d) $\forall x \exists y\ \alpha(x, y) \rightarrow \exists y'\ \alpha(y', y')$
     (e) $\forall y\ \alpha(y, y) \rightarrow \forall x' \exists y'\ \alpha(x', y')$
     – Answer:
       (a) The oracle assures the prover that $\forall x\ \alpha(x)$ is true. In order to prove $\exists y\ \alpha(y)$, the prover must produce some value of y_{prover} for which $\alpha(y_{prover})$ is true. In this case, he can let y_{prover} be anything he wants. His oracle assuring $\forall x\ \alpha(x)$, lets the prover provide any value of x_{prover} that he wants and she will assure him of $\alpha(x_{prover})$ for this value. In this case, the prover provides his y_{prover}. The oracle assures that $\alpha(y_{prover})$ is true for this value. This completes the prover's game.
       (b) This statement is false when $\alpha(0)$ is true and $\alpha(1)$ is false, because this makes $\exists y\ \alpha(y)$ true and $\forall x\ \alpha(x)$ false.

(c) The oracle assures the prover that $\exists y \forall x\ \alpha(x,y)$ is true. In order to prove $\exists y'\ \alpha(y',y')$, the prover needs to find a value y'_{prover} for which α is true on this diagonal. His oracle assuring $\exists y \forall x\ \alpha(x,y)$ gives the prover a value y_{oracle} for which $\forall x\ \alpha(x,y_{oracle})$ is true. Assuring him of $\forall x\ \alpha(x,y_{oracle})$, the oracle allows the prover to give her his favorite x_{prover} and she will assure $\alpha(x_{prover},y_{oracle})$ for this value. He gives her for x_{prover} this same value y_{oracle}, i.e. $x_{prover} = y_{oracle}$. Hence, what she assures him of is $\alpha(y_{oracle},y_{oracle})$ is true. In order to prove $\exists y'\ \alpha(y',y')$, the prover sets y'_{prover} to be this value y_{oracle}, $y'_{prover} = y_{oracle}$ giving him as needed that $\alpha(y'_{prover},y'_{prover})$. This completes the prover's game.

(d) This is not true for the following α.

α	$x=0$	1	2	3
$y=0$	F	T	T	T
1	T	F		
2			F	
3				F

Note that $\forall x \exists y\ \alpha(x,y)$ is true because each column (value of x) has a row (value of y) for which α is true. Note that $\exists y'\ \alpha(y',y')$ is not true because every diagonal is false.

(e) The oracle assures the prover that $\forall y\ \alpha(y,y)$ is true. In order to prove $\forall x' \exists y'\ \alpha(x',y')$, the prover gets from his adversary a value for $x'_{adversary}$ and he must prove $\exists y'\ \alpha(x'_{adversary},y')$ for this value. Because his oracle assures him of $\forall y\ \alpha(y,y)$, he can give this value $y_{prover} = x'_{adversary}$ and she will assure him of $\alpha(x'_{adversary},x'_{adversary})$. In order to prove $\exists y'\ \alpha(x'_{adversary},y')$, he choose y'_{prover} to be $x'_{adversary}$. Because his oracle assures him of $\alpha(x'_{adversary},x'_{adversary})$ and $y'_{prover} = x'_{adversary}$, he knows that $\alpha(x'_{adversary},y'_{prover})$. This completes his game.

– Answer:

(a) The following are fields: *Reals*; *Complex Numbers*; and *Rationals/Fractions.* The set of *Integers* is not because the multiplicative inverse of 2 is $\frac{1}{2}$ which is not an integer. The *Invertible Square Matrices* is not because it is not multiplicatively communicative, i.e. $M \times N$ might be different than $N \times M$ because spinning an object 90 degrees along the z and then 90 degrees along the x results in a different orientating than doing them in the reverse order.

(b) $3\times4 = (1{+}1{+}1)\times(1{+}1{+}1{+}1) = ((1{+}1){+}1)\times(1{+}1{+}1{+}1) = (1{+}1)\times(1{+}1{+}1{+}1) + 1\times(1{+}1{+}1{+}1) = 1\times(1{+}1{+}1{+}1) + 1\times(1{+}1{+}1{+}1) + 1\times(1{+}1{+}1{+}1) = 1{+}1{+}1{+}1{+}1{+}1{+}1{+}1{+}1{+}1{+}1{+}1 = 12$.

(c) To prove $a\times0 = 0$, lets start in the middle with the distributive law, $a\times(0{+}1) = (a\times0){+}(a\times1)$. The $LHS = a\times(1) = a$ and the $RHS = (a\times0)+(a)$. Adding $-a$ to both sides gives $LHS = a+(-a) = 0$ and the $RHS = (a\times0)+a+(-a) = a\times0$. This gives the result.

(d) If $a = \frac{1}{0}$ is zero's multiplicative inverse, then by definition $0\times a = 1$. Commutativity then gives $a\times0 = 1$. But our last proof shows that $a\times0 = 0$. This can be resolved by having $1 = 0$. But then by the $\times$ identity $\forall a\ a\times0 = a$. But again we just proved that $a\times0 = 0$. This can be resolved by having $a = 0$, for all values a. This makes a fine field with one element 0.
Even if ∞ is added to your set of objects, you better have $\infty\times0 = 0$.

(e) In the integers mod 7, the multiplicative inverse of 3 is 5 because $3\times5 = 15 = 7{+}7{+}1 =_{mod\ 7} 0{+}0{+}1 = 1$.
The problem that arises with $2\times3 = 6 =_{mod\ 6} 0$ is that if 2 has a multiplicative inverse $\frac{1}{2}$, then multiplying both sides by it gives that $3 = 0$.

- For which α and f are the following true and for which is is false?
  (a) $(\forall y\ \alpha(y)) \rightarrow (\forall x\ \alpha(f(x)))$
  (b) $(\forall x\ \alpha(f(x))) \rightarrow (\forall y\ \alpha(y))$
  (c) Answer the previous question knowing that $\forall y' \exists x'\ y' = f(x')$.
  (d) $(\exists x\ \alpha(f(x))) \rightarrow (\exists y\ \alpha(y))$

(e) $(\forall x\ \alpha(f(x))) \rightarrow (\exists y\ \alpha(y))$

- Answer:

(a) This is true for every α and every f. The oracle assures the prover that $\forall y\ \alpha(y)$ is true. The prover proves $\forall x\ \alpha(f(x))$ by getting his adversary to give him a value for $x_{adversary}$ and then proving $\alpha(f(x_{adversary}))$ for this value. Because his oracle assures him $\forall y\ \alpha(y)$, the prover can give the value $f(x_{adversary})$ to her for y_{prover} and she will assure $\alpha(y_{prover})$ for his value, namely she assures him of $\alpha(f(x_{adversary}))$ as needed. This completes his game.

(b) This is not true when $\forall x\ f(x)=0$ and $\alpha(y)$ is true only for $y=0$ because $\forall x\ \alpha(f(x))$ is then true and $\forall y\ \alpha(y)$ is false.
The statement is only true when function f is *onto*, namely every value of y has some value of x for which $y=f(x)$, i.e. $\forall y' \exists x'\ y'=f(x')$.

(c) The first oracle assures the prover that $\forall y' \exists x'\ y'=f(x')$ is true and the second that $\exists x\ \alpha(f(x))$ is true. In order to prove $\forall y\ \alpha(y)$, the prover gets from his adversary a value $y_{adversary}$ and he must prove $\alpha(y_{adversary})$ for this value. Because his first oracle assures him of $\forall y' \exists x'\ y' = f(x')$, he can give her this value $y_{adversary}$ for y'_{prover} and she assures him of $\exists x'\ y_{adversary} = f(x')$. Assuring him of this, she gives him a value x'_{oracle} such that $y_{adversary} = f(x'_{oracle})$. Because his second oracle assures him of $\forall x\ \alpha(f(x))$, he can give her this value x'_{oracle} for x_{prover} and she assures him of $\alpha(f(x'_{oracle}))$. Because $y_{adversary} = f(x'_{oracle})$, he knows $\alpha(y_{adversary})$ as needed. This completes his game.

(d) This is true for every α and every f. The oracle assures the prover that $\exists x\ \alpha(f(x))$ is true. In order to prove $\exists y\ \alpha(y)$, the prover must construct a value for y_{prover} and prove $\alpha(y_{prover})$ for this value. Because his oracle assures him of $\exists x\ \alpha(f(x))$, she gives him a value x_{oracle} for which $\alpha(f(x_{oracle}))$ is true. The prover sets y_{prover} to be $f(x_{oracle})$ which gives him $\alpha(y_{prover})$ as needed. This completes his game.

(e) Exercise **??**.1 proved $\forall x\ \alpha(x) \rightarrow \exists y\ \alpha(y)$. This is almost the same as $(\forall x\ \alpha(f(x))) \rightarrow (\exists x\ \alpha(f(x)))$. The last question proves $(\exists x\ \alpha(f(x))) \rightarrow (\exists y\ \alpha(y))$. Transitivity then gives us what we want.

13. For each either play the logic game to prove the statement is true or find a counter example, i.e. a tables $\alpha(x)$ and $\beta(x)$ in which the left hand side is true and the right hand side is false.

(a) $[\forall x\ (\alpha(x) \rightarrow \beta(x))] \rightarrow [\forall x\ \alpha(x) \rightarrow \forall x\ \beta(x)]$

(b) $[\forall x\ \alpha(x) \rightarrow \forall x\ \beta(x)] \rightarrow [\forall x\ (\alpha(x) \rightarrow \beta(x))]$

(c) $[\exists x(\alpha(x) \vee \beta(x))] \rightarrow [(\exists x \alpha(x)) \vee (\exists x \beta(x))]$

(d) $[(\exists x \alpha(x)) \vee (\exists x \beta(x))] \rightarrow [\exists x(\alpha(x) \vee \beta(x))]$

(e) $[\exists x(\alpha(x) \wedge \beta(x))] \rightarrow [(\exists x \alpha(x)) \wedge (\exists x \beta(x))]$

(f) $[(\exists x \alpha(x)) \wedge (\exists x \beta(x))] \rightarrow [\exists x(\alpha(x) \wedge \beta(x))]$

14. A *Field* has a universe U of values, two operations: $+$ and $\times$, and the following axioms.
$+$ Identity: $\exists 0\ \forall a\ a+0=a$
$\times$ Identity: $\exists 1\ \forall a\ a \times 1=a$
Associative: $a+(b+c) = (a+b)+c$ and $a\times(b\times c) = (a\times b)\times c$
Commutative: $a+b = b+a$ and $a\times b = b\times a$
Distributive: $a\times(b+c) = (a\times b)+(a\times c)$
$+$ Inverse: $\forall a\ \exists b\ a+b = 0$, i.e. $b=-a$
$\times$ Inverse: $\forall a\neq 0\ \exists b\ a\times b = 1$, i.e. $b=\frac{1}{a}$

(a) Which of these are fields: *Reals*; *Complex Numbers*; *Rationals/Fractions*; *Integers*; and *Invertible Square Matrices*?

(b) Let "3" be a short form notation for $1+1+1$. Prove from the above axioms that $3\times 4=12$.

(c) Does it follow from these axioms that $a\times 0 = 0$? Warning: The proof is hard. Be sure not to use any facts about the reals that is not listed above. There is no rule that mentions both $\times$ and 0. All rules are paired giving a symmetry between $\langle +, 0\rangle$ and $\langle \times, 1\rangle$ except the distributive law which tie the two together. The same symmetry ties $a\times 0 = 0$ and $a+1 = 1$. Clearly the later is not true. Hence, any proof of $a\times 0 = 0$ must use the distributive law.

(d) What goes wrong with these axioms if zero also has a multiplicative inverse?

(e) The integers mod prime p form a field with only the objects $\{0, 1, \ldots, p-1\}$. From the additional axiom that $7 =_{mod\ 7} 0$, find the multiplicative inverse of 3.

The integers mod 6 do not form a field. $2\times 3 = 6 =_{mod\ 6} 0$ is called a *zero divisor*. What problems arise from this?

15. Define:

A: "Computational problem P is computable by a Java Program," namely

$\exists\ Java\ M\ \forall\ ASCII\ I\ \exists\ time\ t\ P(I) = M(I)$ and $Time(M, I) = t$

B: "The computational problem P treats inputs the same whether in binary or ASCII," namely

$\forall\ binary\ I'\ P(I') = P(I)$ where $B(binary) = ASCII$ and $I = B(I')$

C: "Java programs can be simulated by TM" and
"The TM takes the square of whatever time Java program takes," namely

$\forall\ Java\ M\ \exists\ TM\ M'\ \forall\ binaryI'\ M'(I') = M(I)$
where $M' = Compile_{JAVA\Rightarrow TM}(M)$
and $\forall t\ [Time(M, B(I')) = t \rightarrow Time(M', I') \leq t^2]$

Z: "Computational problem P is computable by a TM," namely

$\exists\ TM\ M'\ \forall\ binary\ I'\ \exists\ time\ t'\ P(I') = M'(I')$ and $Time(M', I') = t'$

Prove $\langle A, B, C\rangle \rightarrow Z$

(a) Using the prover/adversary/oracle game. Fancy parsing is not required. Prover Z and the three oracles A, B, and C each play their game by reading their statement left to right. These four games merge together. Focus on who gives whom what when.

(b) Using our formal proof system.

16. Recall the Fibonacci sequence $\{f_n\}$ that is defined by $f_0 = 0$, $f_1 = 1$, and $f_n = f_{n-1} + f_{n-2}$ for $n \geq 2$. Prove that

$$f_1 + f_3 + f_5 + \cdots + f_{2n-1} = f_{2n} \quad \text{for every positive integer } n.$$

- Answer: Let $S(n)$ denote the truth of the statement for n.
  Base case $S(0)$: The last term of the LHS is f_{-1}, which tells me that there are no terms so the sum is zero. The RHS is $f(0) = 0$. Hence, $S(0)$ is true.
  If that made you nervous, $S(1)$ has LHS $= f_1 = 1$ and RHS $= f(2) = f(0) + f(1) = 1$. Hence, $S(1)$ is true.
  Let $n \geq 1$ be arbitrary. By way of induction assume $S(n-1)$.
  We prove $S(n)$ as follows.
  LHS $= f_1 + f_3 + f_5 + \cdots + f_{2n-3} + f_{2n-1}$
  $= [f_1 + f_3 + f_5 + \cdots + f_{2(n-1)-1}] + f_{2n-1}$
  (By assumption $S(n-1)$
  $= f_{2(n-1)} + f_{2n-1} = f_{2n-2} + f_{2n-1} = f_{2n}$.
  Hence, $S(n)$ and hence by induction $\forall n S(n)$.

17. Let W be the set of all natural numbers n that can be written in the form $n = 3a + 5b$ for some natural numbers a and b. (Recall that the set of natural numbers is $\mathbb{N} = \{0, 1, 2, 3, \ldots\}$.) For example, $28 \in W$ because $28 = 3 \times 6 + 5 \times 2$.

For each $n \in \mathbb{N}$, let $P(n)$ be the statement that $\{n, n+1, n+2\} \subseteq W$. Prove that

$$\forall k \in \mathbb{N}\ (P(k) \to P(k+1)).$$

(Note: Do not try to prove a basis step.)

- Answer: Let n be arbitrary.
  By way of deduction/induction assume that $P(n)$ is true.
  By definition, this assures us that $\{n, n+1, n+2\} \subseteq W$.
  Hence, $n = 3a + 5b$ for some a and b.
  Hence, $(n+3) = 3(a+1) + 5b$ for the same a and b.
  Hence, $(n+3) \in W$.
  We had $(n+2) \in W$ and $(n+2) \in W$ from $P(k)$. Hence $\{n+1, n+2, n+3\} \subseteq W$.
  Hence, $S(n+1)$.
  This proves $\forall k \in \mathbb{N}\ (P(k) \to P(k+1))$.

18. In EECS 1019 you learned structural induction. The professor threw up a proof. She did not say it followed a prover/adversary/oracle game, but it did. And as such, you likely did understand it. For Math 1090, I have a whole section of slides on induction, $[S(0)\ \&\ \forall i\ (S(i{-}1) \to S(i))] \ \to\ \forall i\ S(i)$:

Iterative Algorithms: Your oracle assures you of $S(i{-}1)$ that after $i{-}1$ iterations the loop invariant is true about the structure you are building. You must prove $S(i)$ that it is true after i iterations.

Recursive: Your oracle assures you of $S(i{-}1)$ that your recursive friends can solve any input instance as long as it has size smaller than i. Your must prove $S(i)$ by solving yours of size i.

Graphs: Your oracle assures you of $S(i{-}1)$ that she can 6-color planer graphs with $i{-}1$ nodes. Your must prove $S(i)$ by coloring one with i nodes.

Test 2 and the exam will not require you to know much about "graph theory" but you should be able to play the game required for these.

Here is a practice test 2 problem: Prove by induction that every list of integers is sortable.

(a) Write everything in logic:
For this problem, the input is a list $L = L(1), L(2), L(3), ...L(n)$, where each $L(i)$ is an integer.
A solution is a reordering of R which is the list $R = R(1), R(2), R(3), ...R(n)$, where the $R(i)$'s are the SAME integers.
A query about your solution is a pair of indexes $\langle u, v\rangle$. List R is a valid solution, ie sorted, if every pair of numbers is in the right order. ie. $\forall$ indexes $\langle u, v\rangle$ $u{<}v \ \to\ R(u){\leq}R(v)$
Write the statement "Every list of numbers is sortable" in logic.

- Answer: $\forall$ list L, $\exists$ reordering list R, $\forall$ indexes $\langle u, v\rangle$ $u{<}v \ \to\ R(u){\leq}R(v)$.

You can prove this two ways:

Prove Insertion Sort Works: Define the Loop Invariant / Induction Hypothesis as
$S(i)$ = "After i iterations, the algorithm has constructed a sorted reordering R of the first i numbers." Maintaining the loop invariant proves $\forall i\ S(i{-}1) \to S(i)$.

Prove Merge Sort Works: The Friends / Strong Induction Argument is as follows: Assume $S(i{-}1)$ = "My recursive algorithm works for every input smaller than i."
Prove $S(i)$ = "My recursive algorithm works for every input of size i.

(b) Write the statement $S(i)$ in logic by adding to your statement from question (a) the requirement that your list L has size i (or smaller). Then list the steps of the game in which the prover proves $S(i)$. (You REALLY REALLY need to learn the steps of the game.)

- Answer: Goal is to prove $S(i) \equiv \forall$ lists L of size i, $\exists$ reordering list R, $\forall$ indexes $\langle u, v\rangle$ $u < v \;\rightarrow\; R(u) \leq R(v)$.
  The steps of the game are as follows.
  Let L be an arbitrary list of size i given to me by the adversary.
  The prover constructs a reordering list R.
  Let $\langle u, v\rangle$ be arbitrary indexes given to me by the adversary for which $u < v$.
  The prover proves $R(u) \leq R(v)$.
  This completes the proof.

(c) Write the statement $S(i{-}1)$ in logic by modifying your statement from question (a). First change each object L, R, u, and v to L', R', u', and v' to differentiate between the objects handled by the oracle from those handled by the prover. Second, add the requirement that your list L' has size $i-1$ or smaller. Then list the steps of the game in which the oracle assures us of $S(i{-}1)$.

- Answer: The oracle assures us of $S(i{-}1) \equiv \forall$ lists L' of size smaller than i, $\exists$ reordering list R', $\forall$ indexes $\langle u', v'\rangle$ $u' < v' \;\rightarrow\; R'(u') \leq R'(v')$.
  The steps of the game are as follows.
  The prover gives the oracle a list L' of size smaller than i.
  The oracle constructs a reordering list R'
  The prover gives the oracle indexes $\langle u', v'\rangle$ for which $u' < v'$.
  The oracle assures $R'(u') \leq R'(v')$.
  This completes the proof.

(d) Merge the steps in the two lists either by numbering or rewriting them, into an order so that one gets an object before one gives a mortified version of it. Do NOT add any statement about how to modify objects. We will do those in the next question.
Hint: To do this question you do NOT need to know whether we are doing Insertion or Merge Sort. In fact, the answer would be the same for the planar graph colouring problem and most/many structural induction problems.

- Answer: We merge the steps of the game as follows.
  Let L be an arbitrary list of size i given to me by the adversary.
  The prover gives the oracle a list L' of size smaller than i.
  The oracle constructs a reordering list R'
  The prover constructs a reordering list R
  Let $\langle u, v\rangle$ be arbitrary indexes given to me by the adversary for which $u < v$.
  The prover gives the oracle indexes $\langle u', v'\rangle$ for which $u' < v'$.
  The oracle assures $R'(u') \leq R'(v')$.
  The prover proves $R(u) \leq R(v)$.
  This completes the proof.

We now need to add the statements about how to modify objects. If you look at your previous answer, you will see that going from $S(i{-}1)$ to $S(i)$ requires modifying a full input/list L of size i to become a subinstance input/list L' of size $i-1$ or smaller, ie, made smaller. Going the other way, we must modify a subsolution/reordering R' of size $i-1$ or smaller into a full solution/reordering R of size i, ie, made bigger. Then the query about your solution, which here are indexes $\langle u, v\rangle$, likely have the following two cases.

Case 1: The answer to the query is completely about the subsolution R' and hence is handled by the oracle.

Case 2: The answer is about the part of the solution that the prover created when expanding the subsolution from R' to the full solution R and hence is handled by the prover.

As said, what we have done so far applies generally to Insertion Sort, Merge Sort, Planar Graph Colouring, and most/many structural induction problems. We are now going to focus on Merge Sort.

(e) Give the informal game to prove $\forall i\; S(i{-}1) \rightarrow S(i)$ by adding to your previous answer the required preamble and the statements about how to modify objects.

Hint: The following are Jeff's classic modifying hints specific to Merge Sort: We have two oracles/friends. It is fun to make structures smaller by splitting them into two halves, to merge two solutions into one, and to consider the following two cases:

Case 1: Our indexes are handled completely by the same oracle.

Case 2: They are handled by the different oracle and hence merged.

Hint: We are not learning about the details of Merge Sort and hence only a quick sentence needs to be given for each step.

- Answer: The steps to prove $\forall i\ S(i-1) \rightarrow S(i)$ are as follows:
  Let i be an arbitrary integer given to me by my adversary.
  Assume and let our oracle assure us of $S(i-1)$.
  We must prove $S(i)$.
  Let L be an arbitrary full list of size i given to me by the adversary.
  Let L'_1 be the first half of the list L and L'_2 be the second half.
  The prover gives the first oracle sublist L'_1 and the second oracle L'_2. Note that both are of size smaller than i.
  The oracles construct reordering sublists R'_1 and R'_2.
  The prover constructs a reordering full list R by merging R'_1 and R'_2 together.
  Let $\langle u, v\rangle$ be arbitrary indexes given to me by the adversary for which $u<v$.
  Look at the following two cases:
    - Case 1: $L(u)$ and $L(v)$ are handled completely by the same oracle.
      The prover gives the oracle the same indexes $u'=u \& v'=v$.
      The oracle assures $R'(u') \leq R'(v')$.
      The prover says that the constructed reordering R does not change the oracle's reordering and hence $R(u) \leq R(v)$.
    - Case 2: They are handled by the different oracle and hence merged.
      The prover says that the merging is done in a way that assures that $L(u)$ and $L(v)$ were reordered in the correct order and hence $R(u) \leq R(v)$.

  This completes the proof.

(f) Can you solve other structural induction problems? The EECS 1019 book is full of them.