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) $\exists x$ I-can-have(x); c) other
   (b) Can I have anything I want? a) $\forall x$ 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) $\exists x$ I-can-have(x); c) other
   (d) Everyone is married: a) $\forall x\ \exists y\ loves(x, y)$; b) $\exists y\ \forall x\ loves(x, y)$; c) other
   (e) Everyone loves her: a) $\forall x\ \exists y\ loves(x, y)$; b) $\exists y\ \forall x\ loves(x, y)$; c) other
   (f) Every real has an inverse: a) $\forall x\ \exists y$; b) $\exists y\ \forall x$; c) other
   (g) Every Hindu has God: a) $\forall x\ \exists y$; b) $\exists y\ \forall x$; c) other
   (h) Every Christian has God: a) $\forall x\ \exists y$; b) $\exists y\ \forall x$; c) other
   (i) Only Jeff would: 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) $would(\text{Jeff}) \vee \forall x \neq \text{Jeff}\ negwould(x)$; 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) $would(\text{Jeff}) \wedge \exists x \neq \text{Jeff}\ would(x)$; 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) 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) $[\forall x\ \exists y\ P(x, y)] \Leftarrow [\exists y\ \forall x\ P(x, y)]$

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?

(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?

(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.

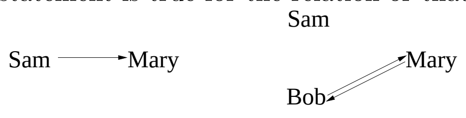
4. Define $Even(x) \equiv [\exists k,\ x = 2k]$. Prove $\forall x,\ [Even(x) \rightarrow Even(x+2)]$

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."

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

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

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. 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. $\forall x \in S^+, \exists y \in S^+, y < x$.
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

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 $c_{\langle A,\exists\rangle}$.

(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.

A	x_1			
y_1				

B	x_1			
y_1				

(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.

(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., $\beta \wedge \neg\beta$.

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

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
ii. A does not computes P
iii. P is computable
iv. P is not computable

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

(c) P is computable

(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 $\forall I \exists A\ A(I) = HALT(I)$.

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 $3n^2$.

    (b) P is computable in polynomial time.

    (c) P is not computable in polynomial time.

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

    (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.)

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?

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

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) $[\forall x\ (\alpha(x) \to \beta(x))] \to [\forall x\ \alpha(x) \to \forall x\ \beta(x)]$

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

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

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

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

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

14. A *Field* has a universe U of values, two operations: $+$ and $\times$, and the following axioms.

    + Identity: $\exists 0\ \forall a\ a{+}0{=}a$
    × 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. $b{=}{-}a$
    × 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

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

Continuation of answer

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

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

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

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, $[S(0)\ \&\ \forall i\ (S(i-1) \rightarrow S(i))] \rightarrow \forall i\ S(i)$:

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 $\langle u, v\rangle$ $u<v \;\rightarrow\; R(u)\leq R(v)$
Write the statement "Every list of numbers is sortable" in logic.

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 $\forall i\ S(i-1) \rightarrow S(i)$.

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.)

(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 $S(i-1)$.

(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.

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.

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