

York University
CSE 2001 – Unit 1 First Order Logic
Instructor: Jeff Edmonds

Don't cheat by looking at these answers prematurely.

1. 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.

- 1) $\forall x \exists y x + y = 5$ 2) $\exists y \forall x x + y = 5$
- 3) $\forall x \exists y x \cdot y = 5$ 4) $\forall x \exists y x \cdot y = 0$
- 5) $[\forall x \exists y P(x, y)] \Rightarrow [\exists y \forall x P(x, y)]$
- 6) $[\forall x \exists y P(x, y)] \Leftarrow [\exists y \forall x P(x, y)]$
- 7) $\forall a \exists y \forall x x \cdot (y + a) = 0$
- 8) $\exists a \forall x \exists y [x = 0 \text{ or } x \cdot y = 5]$

- Answer:

- (a) $\forall x \exists y x + y = 5$ is true. Let x be an arbitrary real value and let $y = 5 - x$. Then $x + y = 5$.
- (b) $\exists y \forall x x + y = 5$ is false, because $\forall y \exists x x + y \neq 5$ is true. Let y be an arbitrary real value and let $x = 6 - y$. Then $x + y = 6 \neq 5$.
- (c) $\forall x \exists y x \cdot y = 5$ is false, because $\exists x \forall y x \cdot y \neq 5$ is true. Let $x = 0$ and let y be an arbitrary real value. Then $x \cdot y = 0 \neq 5$. Note that y must be $\frac{5}{0}$, which is impossible.
- (d) $\forall x \exists y x \cdot y = 0$ is true. Let x be an arbitrary real value and let $y = 0$. Then $x \cdot y = 0$. The odd thing about this example is that even though the value of y can depend on the value of x , it does not. $\exists y \forall x x \cdot y = 0$ is a stronger statement that is also true.
- (e) $[\forall x \exists y P(x, y)] \Rightarrow [\exists y \forall x P(x, y)]$ is false. Let $P(x, y) = [x + y = 5]$. Then as seen above the first is true and the second is false.
- (f) $[\forall x \exists y P(x, y)] \Leftarrow [\exists y \forall x P(x, y)]$ is true. Assume the right is true. Let y_0 be 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.
- (g) $\forall a \exists y \forall x x \cdot (y + a) = 0$ is true. Let a be an arbitrary real value. Let $y = -a$. Let x be an arbitrary real value. Then $x \cdot (y + a) = 0$ is true.
- (h) $\exists a \forall x \exists y [x = 0 \text{ or } x \cdot y = 5]$ is true. Let $a = 0$. Let x be an arbitrary real value. If $x = 0$ then $[x = 0 \text{ or } x \cdot y = 5]$ is true because of the left. If $x \neq 0$ then let $y = \frac{5}{x}$ and $[x = 0 \text{ or } x \cdot y = 5]$ is true because of the right.

2. The game *Ping* has two rounds. Player-A goes first. Let m_1^A denote his first move. Player-B goes next. Let m_1^B denote his move. Then player-A goes m_2^A and player-B goes m_2^B . The relation $Twins(m_1^A, m_1^B, m_2^A, m_2^B)$ is true iff player-A wins with these moves.

- (a) Use universal and existential quantifiers to express the fact that player-A has a strategy in which he wins no matter what player-B does. Use $m_1^A, m_1^B, m_2^A, m_2^B$ as variables.
- (b) What steps are required in the Prover/Adversary technique to prove this statement?
- (c) What is the negation of the above statement in standard form?
- (d) What steps are required in the Prover/Adversary technique to prove this negated statement?

- Answer: Regarding the game *Ping*.

- (a) The statement that player-A has a strategy in which he wins no matter what player-B does is $\exists m_1^A \forall m_1^B \exists m_2^A \forall m_2^B Twins(m_1^A, m_1^B, m_2^A, m_2^B)$. His strategy specifies his first move m_1^A . Then for each move m_1^B his opponent makes, he must specify his next move m_2^A . This must lead to a win no matter what his opponents next move is.

- (b) The Prover/Adversary technique to prove this statement is a strategy for the prover to win the following game. The prover gives m_1^A , the adversary give m_1^B , the prover gives m_2^A , the adversary give m_2^B , and the prover wins if $AWins(m_1^A, m_1^B, m_2^A, m_2^B)$ is true.
 - (c) The negation of the above statement is $\forall m_1^A \exists m_1^B \forall m_2^A \exists m_2^B \neg AWins(m_1^A, m_1^B, m_2^A, m_2^B)$.
 - (d) The Prover/Adversary technique to prove this negated statement is a strategy for the prover to win the game when he takes the role of player-B.
3. Let $Works(P, A, I)$ to true if algorithm A halts and correctly solves problem P on input instance I . Let $P = Halting$ be the Halting problem which takes a Java program I as input and tells you whether or not it halts on the empty string. Let $P = Sorting$ be the sorting problem which takes a list of numbers I as input and sorts them. For each part, explain the meaning of what you are doing and why you don't do it another way.

Extra:

Let $A_{insertionsort}$ be the sorting algorithm which we learned in class.

Let A_{yes} be the algorithm that on input I ignores the input and simply halts and says "yes".

Let A_∞ be the algorithm that on input I ignores the input and simply runs for ever.

- (a) Recall that a problem is *computable* if and only if there is an algorithm that halts and returns the correct solution on every valid input. Express in first order logic that $Sorting$ is computable.
- (b) Express in first order logic that $Halting$ is not computable.
- (c) Express in first order logic that there are uncomputable problems.
- (d) What does the following mean and either prove or disprove it: $\forall I, \exists A, Works(Halting, A, I)$. (Not simply by saying the same in words "For all I , exists A , $Works(Halting, A, I)$ ")
- (e) What does the following mean and either prove or disprove it $\forall A, \exists P, \forall I, Works(P, A, I)$. Hint: An algorithm A on an input I can either halt and give the correct answer, halt and give the wrong answer, or run for ever.

• Answer:

- (a) $\exists A, \forall I, Works(Sorting, A, I)$. We know that there at least one algorithm, eg. $A = mergesort$, that works for every input instance I .
- (b) $\forall A, \exists I, \neg Works(Halting, A, I)$ We know that opposite statement is true. Every algorithm fails to work for at least one input instance I .
- (c) $\exists P, \forall A, \exists I, \neg Works(P, A, I)$
- (d) It says that every input has some algorithm that happens to output the right answer. It is true. Consider an arbitrary instance I . If on instance I , $Halting$ happens to say yes, then let A be the algorithm that simply halts and says "yes". Otherwise, let A be the algorithm that simply halts and says "no". Either way A "works" for this instance I .
- (e) It says that every algorithm correctly solves some problem. This is not true because some algorithm do not halt on some input instances. We prove the complement $\exists A, \forall P, \exists I, \neg Works(P, A, I)$ as follows. Let A be an algorithm that runs for ever on some instance I' . Let P be an arbitrary problem. Let I be an instance I' on which A does not halt. Note $Works(P, A, I)$ is not true.

4. First Order Logic:

Let P be some computable problem,

k an integer,

A an algorithm (Java Program),

and I an input string.

Let $Lines(A, I) = k$ to be the statement that algorithm A has k actual lines of code (in the print out of the program) when run on input I .

Let $A(I) = L(I)$ to be the statement that A gives the correct answer for P on input I .

For each of the following first order logic statements, is it true and what are the ramifications/consequences of this with respect to solving P ? i.e. why is it true/false.

The types of things that the first order logic will say are “Computable means that a fixed algorithm can get the right answer on each and every input” and “The number of lines of code does not change with the input.”

(a) $\exists k, \exists A, \forall I, \text{Lines}(A, I) = k$ and $A(I) = P(I)$

- Answer: True. The algorithm designer can build an algorithm A with some number k of lines and then no matter what the input I , this A will solve P with this number of lines.

(b) $\forall k, \exists A, \forall I, \text{Lines}(A, I) = k$

- Answer: True. The algorithm designer can build an algorithm with any number k of lines and then this number is fixed as the input grows.

(c) $\forall A, \exists k, \forall I, \text{Lines}(A, I) = k$

- Answer: True. Each algorithm has a fixed number of lines that does not change as the input grows.

(d) $\forall k, \exists A, \forall I, \text{Lines}(A, I) = k$ and $A(I) = P(I)$

- Answer: False. There is some minimum number of lines that the algorithm needs below which it can't solve P .

(e) $\forall I, \exists A, A(I) = P(I)$

- Answer: True even for undecidable problems A . It is not a good statement because there should not be a different algorithm for each input I . One of A_{yes} or A_{no} happens to get the right answer for this I .

(f) $\forall A, \exists I, A(I) \neq P(I)$

- Answer: False. This is stating that P is not computable because no algorithm solves it for every input.