CSE 311: Foundations of Computing I Spring 2023

Problem Set 3

Due: Wednesday, April 19, by 11:59pm

Instructions

Solutions submission. You must submit your solution via Gradescope. In particular:

- Submit a *single* PDF file in GRadescope containing the written solution to all the regular tasks in the homework including the written portion involving the English for question 2, which has an online portion also.
- Submit solutions on cozy.cs.washington.edu for question 1 and the online portion of question 2.
- The extra credit is submitted separately in Gradescope

Task 1 – Proof Positive [12 pts]

Prove the following inferences. (These inferences are the basis for two versions of a strategy for finding proofs called *Proof by Cases* that is common in reasoning and we will use later.)

a) Given $A \to C$ and $\neg A \to C$, prove that C holds.

b) Given $A \vee B$, $A \to C$, and $B \to C$, prove that C holds.

Submit and check your **formal proofs** here:

http://cozy.cs.washington.edu.
The server will open as of 12:00 noon on April 13.

You can make as many attempts as needed to find a correct answer.

Task 2 – Burden of Proof [15 pts]

For each of the following, write a **formal proof** that the claim holds. Then, translate your proof to English.

You are free to use Cozy's English translation of your proof as a *starting point*, but you should rephrase it so that it sounds clear and natural to you. For this problem, your English proof will only be graded for effort, not points, but you will get feedback from the grader about whether your English proof sounds clear to them as well. In future homeworks (and on Task 6 for this homework), we will grade your English proofs not just for effort, but also for clarity and correctness, so be sure to listen carefully to the feedback you receive on your English proofs this week.

a) Given $P \wedge Q$, $P \to (R \wedge S)$, and $(Q \wedge R) \to T$, it follows that T holds.

b) Given Q, $(Q \to T) \wedge (T \to Q)$, and $P \to (R \to S)$, it follows that $T \wedge ((P \wedge R) \to S)$.

Hint: The "Direct Proof" rule will be useful here.

c) Given $P \to (Q \wedge R)$, $S \vee \neg Q$, and $(R \wedge S) \to T$, it follows that $P \to T$.

Hint: The rule "Elim $\vee$" will be useful here.

Submit and check your **formal proofs** here:

http://cozy.cs.washington.edu
The server will open as of 12:00 noon on April 13.

You can make as many attempts as needed to find a correct answer.

You should submit your **English proofs** on Gradescope. Note that the grader will only be looking at the English portion to give you (ungraded) feedback. They will not see your formal proof, so the English proof must make sense on its own.

Task 3 – Proof, Goof, or Spoof? [18 pts]

For each of the claims below, (1) translate the English proof into a formal proof and (2) say which of the following categories describes the formal proof:

Proof The proof is correct.

Goof The claim is true but the proof is wrong.

Spoof The claim is false.

Finally, (3) if it is a goof, point out the errors in the proof and explain how to correct them, and if it is a spoof, point out the *first* error in the proof and then show that the claim is false by giving a counterexample. (If it is a correct proof, then skip part (3).)

Be careful! We want you to translate the English proof to a formal proof as closely as possible, including translating the mistake(s), if any! Also, an incorrect proof does not necessarily mean the claim is false, i.e., a goof is not a spoof!

Note that English proofs often skip steps that would be required in formal proofs. (That is fine as long as it is easy for the reader to see what needs to be filled in.) Skipped steps do not mean that the proof is incorrect. The proof is incorrect when it asserts a fact that is not necessarily true or does not follow by the reason given.

Hint: To give a counterexample to a claim in propositional logic, describe what truth values each atomic variable should have so that all the givens are true but the result is false.

a) **Claim**: Given $\neg q \vee s \vee \neg r$ and $p \vee \neg s$, it follows that $(r \wedge q) \to p$ holds.

Proof or Goof or Spoof: Assume r and q. The first given is equivalent to $\neg(r \wedge q) \vee s$. Since we know that r and q are true, this tells us that s must be true. The second given then tells us that p must be true.

b) **Claim**: Given $p \to (\neg q \to r)$, $s \vee \neg q$, $q \to \neg s$, and $\neg p \to q$, it follows that r holds.

Proof or Goof or Spoof: The second and third givens combined are equivalent to $\neg q$. The contrapositive of the last given is $p \to \neg q$, which combined with $\neg q$ gives us p. From these two, the first given results in r.

c) **Claim**: Given $s \to (p \wedge q)$, $\neg s \to r$, and $(r \wedge p) \to q$, it follows that q holds.

Proof or Goof or Spoof: First, note that $s \to q$ holds, since, assuming s, from given $s \to (p \wedge q)$ we can get p and also q. Second, note that $\neg s \to q$ holds, since, assuming $\neg s$, from given $\neg s \to r$ we can get r, and we already saw that p holds, so together we have $r \wedge p$, from which q follows by the last given. Finally, either s or $\neg s$ is true, and q follows either from our first claim or second claim above, respectively.

Task 4 – Something to Prove [20 pts]

For each of the following, write a **formal proof** that the claim holds. Then, translate your proof to English.

Your proof should sound clear and natural to you. For this problem, your English proof will only be graded for effort, not points, but you will get feedback from the grader about whether your English proof sounds clear to them as well. On future homeworks (and on Task 6 on this homework), we will grade your English proofs not just for effort, but also for clarity and correctness, so be sure to listen carefully to the feedback you receive on your English proofs this week.

Let $P(x, y)$, $Q(y)$, and $R(z)$ be predicates defined in some fixed domain of discourse. Let c and d be some well-known constants in that domain.

a) Given $\forall x\, P(x, c)$ and $\forall y\, (Q(y) \to P(c, y))$, it follows that $Q(d) \to \exists z\, (P(d, z) \wedge P(z, d))$.

b) Given $\exists x\, \forall y\, (Q(x) \wedge P(x, y))$, it follows that $\forall y\, \exists x\, (Q(x) \wedge P(x, y))$.

Note: We showed in class that the converse of this statement does not hold!

c) Given $\forall x\, (Q(x) \wedge \exists y\, P(x, y))$, it follows that $\forall x\, \exists y\, (Q(x) \wedge P(x, y))$.

Note: This fact was noted (but not proven) in lecture. You are asked to prove it here.

d) Given $\forall x\, (Q(x) \to R(x))$, it follows that $(\forall x\, Q(x)) \to (\forall x\, R(x))$.

Note: You saw in Homework 2 Task 7 that these two are different. Here, you are proving that they are related: specifically, from the first of these, we can infer the second!

Task 5 – Proof, Goof, or Spoof? Predicate logic edition [18 pts]

For each of the claims below, (1) translate the English proof into a formal proof and (2) say which of the following categories describes the formal proof:

Proof The proof is correct.

Goof The claim is true but the proof is wrong.

Spoof The claim is false.

Finally, (3) if it is a goof, point out the errors in the proof and explain how to correct them, and if it is a spoof, point out the *first* error in the proof and then show that the claim is false by giving a counterexample. (If it is a correct proof, then skip part (3).)

Be careful! We want you to translate the English proof to a formal proof as closely as possible, including translating the mistake(s), if any! Also, an incorrect proof does not necessarily mean the claim is false, i.e., a spoof is not a goof!

Note that English proofs often skip steps that would be required in formal proofs. (That is fine as long as it is easy for the reader to see what needs to be filled in.) Skipped steps do not mean that the proof is incorrect. The proof is incorrect when it asserts a fact that is not necessarily true or does not follow by the reason given.

Hint: To give a counterexample to a claim in predicate logic, describe a domain of discourse and definitions for all predicates such that all the givens are true but the result is false.

a) **Claim**: Given $\forall x\,(P(x) \to \exists y\, R(x,y))$ and $\forall y\,(Q(y) \to \forall x\, R(x,y))$, it must follow that $\forall x\,\big((P(x) \wedge Q(x)) \to \exists z\,(R(x,z) \wedge R(z,x))\big)$.

Proof or Goof or Spoof: Let a be an arbitrary P that is also a Q. By the first given, since a is a P, there exists b such that $R(a,b)$. By the second given, since a is a Q, we also have $R(b,a)$. Since a was arbitrary, the claimed result follows.

b) For this part, assume the domain of discourse is the integers.

Claim: Given $\forall x\,\exists y\, R(x,y)$ and $\forall x\,\forall y\,(R(x,y) \to R(y,x))$, it follows that $\exists x\, R(x,x)$.

Proof or Goof or Spoof: By the first given plugging in 7 for x, we get that there exists b such that $R(7,b)$. By the second given, since $R(7,b)$, we can plug in 7 for both x and y, and we get $R(7,7)$. The result follows.

c) **Claim**: Given $\forall x\,\forall y\,(R(x,y) \to R(y,x))$ and $\forall x\,\forall y\,\forall z\,((R(x,y) \wedge R(y,z)) \to R(x,z))$, it follows that $\forall x\,\forall y\,(R(x,y) \to R(x,x))$.

Proof or Goof or Spoof: Let a and b be arbitrary and assume $R(a,b)$. By the first given, we have $R(b,a)$. By the second given, plugging in b for x, a for y, and b for z, we get $R(b,b)$. Since a and b were arbitrary, the result follows.

Task 6 – Hip to be square [17 pts]

We say that an integer n is a square iff there exists an integer k such that $n = k^2$.

a) Give a formal proof that, if integers n and m are squares, then nm is a square using a predicate $Square(n) := \exists k\, (n = k^2)$ over the domain of integers. In addition to the inference rules discussed in class, you can also rewrite an algebraic expression to equivalent ones using the rule "Algebra".

b) Write your proof from part (a) as an English proof. (Note that this statement and its proof follows the format of some of the formal proofs that we have written in lecture where we have given direct translations to English proofs, so this is the one English proof we will be grading for points this week.)

Task 7 – Extra Credit: A Step In the Right Direction

In this problem, we will extend the machinery we used in HW1's extra credit problem in two ways. First, we will add some new instructions. Second, and more importantly, we will add *type information* to each instruction.

Rather than having a machine with single bit registers, we will imagine that each register can store more complex values such as

Primitives These include values of types int, float, boolean, char, and String.

Pairs of values The type of a pair is denoted by writing "×" between the types of the two parts. For example, the pair $(1, \text{true})$ has type "int × boolean" since the first part is an int and the second part is a boolean.

Functions The type of a function is denoted by writing a "→" between the input and output types. For example, a function that takes an int as argument and returns a String is written "int → String".

We add type information, describing what is stored in each each register, in an additional column next to the instructions. For example, if R_1 contains a value of type int and R_2 contains a value of type int → (String × int), i.e., a function that takes an int as input and returns a pair containing a String and an int, then we could write the instruction

$$R_3 := \texttt{CALL}(R_1, R_2) \qquad\qquad \text{String} \times \text{int}$$

which calls the function stored in R_2, passing in the value from R_1 as input, and stores the result in R_3, and write a type of "String × int" in the right column since that is the type that is now stored in R_3.

In addition to CALL, we add new instructions for working with pairs. If R_1 stores a pair of type String × int, then $\texttt{LEFT}(R_1)$ returns the String part and $\texttt{RIGHT}(R_1)$ returns the int part. If R_2 contains a char and R_3 contains a boolean, then $\texttt{PAIR}(R_2, R_3)$ returns a pair of containing a char and a boolean, i.e., a value of type char × boolean.

(a) Complete the following set of instructions so that they compute a value of type char in the last register assigned (R_N for some N):

R_1	int × float
R_2	int → (String × boolean)
R_3	(float × String) → char
$R_4 := \ldots$	$\ldots$

The first three lines show the types **already stored** in registers R_1, R_2, and R_3 at the start, before your instructions are executed. You are free to use the values in those registers in later instructions.

Since we have unlimited space, store into a *new register* on each line. **Do not reassign** any registers.

(b) Compare the types listed next to these instructions to the propositions listed on the lines of your proof in Task 2 (a). Give a collection of text substitutions, such as replacing all instances of "P" by "int" (these can include substitutions for atomic propositions and for operators), that will make the sequence of propositions in Task 2(a) *exactly match* the sequence of types in part (a) above. (You may need to change your solution to part (a) slightly to make this work!)

(c) Now, let's add another way to form new types. If A and B are types, then $A + B$ will be the type representing values that can be of either type A or type B. For example, String + int would be a type of values that can be strings or integers.

To work with this new type, we need some new instructions. First, if R_1 has type A, then the instruction $\texttt{CASE}(R_1)$ returns the same value but now having type $A + B$. (Note that we can pick any type B that we want here.) Second, if R_2 stores a value of type $A + B$, R_3 stores a function of type $A \to C$ (a function taking an A as input and returning a value of type C), and R_4 stores a function of type $B \to C$, then the instruction $\texttt{SWITCH}(R_2, R_3, R_4)$ returns a value of type C: it looks at the value in R_2, and, if it is of type A, it calls the function in R_3 and returns the result, whereas, if it is of type B, it calls the function in R_4 and returns the result. In either case, the result is something of type C.

Complete the following set of instructions so that they compute a value of type float + boolean in the last register assigned:

R_1	int × (float + String)
R_2	float → (int → boolean)
R_3	String → (int → boolean)
$R_4 := \ldots$	$\ldots$

The first three lines again show the types of values already stored in registers R_1, R_2, and R_3. As before, do not reassign any registers. Use a new register for each instruction's result.

(d) Suppose that you had the Proof by Cases rule that you proved in Task 1 (b). Give a proof using this rule (but not the Elim ∃ rule) that $Q \vee S$ must follow given $P \wedge (Q \vee R)$, $Q \to (P \to S)$, and $R \to (P \to S)$.

(e) Compare the types listed next to the instructions in part (c) to the propositions listed on the lines of your proof in (d). Give a collection of text substitutions, such as replacing all instances of "r" by "int" (these can include substitutions for atomic propositions and for operators), that will make the sequence of propositions in part (c) *exactly match* the sequence of types in part (d) above. (You may need to change your solution to part (c) slightly to make this work!)

(f) Now that we see how to match up the propositions in our earlier proofs with types in the code above, let's look at the other two columns. Describe how to translate each of the rules of inference used in the proofs from your propositional proofs so that they turn into the instructions in (a) and (c).

(g) One of the important rules **not** used in Task 2 (a) or in part (d) of this problem was Direct Proof. What new concept would we need to introduce to our assembly language so that the similarities noted above apply could also to proofs that use Direct Proof?