CSE 311: Foundations of Computing I

Homework 4 (due Monday, October 31st at 11:00 PM)

Directions: *Write up carefully argued solutions to the following problems. Each solution should be clear enough that it can explain (to someone who does not already understand the answer) why it works. However, you may use results from lecture, the reference sheets, and previous homeworks without proof.*

1. Even on a Jet Plane (12 points)

Let domain of discourse be the integers. Consider the following claim:

$$\forall x\, \forall y\, \forall z\, ((\mathsf{Odd}(x+y) \wedge \mathsf{Odd}(y+z)) \rightarrow \mathsf{Even}(x+z))$$

(a) [1 Point] Translate the claim into English.

(b) [8 Points] Write a **formal proof** that the claim holds.

Some important notes:

- You will need to use the definitions of Odd and Even (via Cozy's `defof` and `undef` rules).
- You will also need to use Cozy's `algebra` rule, which lets you infer equations implied by others:

Algebra

$$\frac{x_1 = y_1 \quad \ldots \quad x_n = y_n}{\therefore\ x = y \text{ (if implied)}}$$

- Remember that Cozy (like Java) expects a "`*`" for multiplication. It will misunderstand if you write `2a + 2 = 2(a+1)`. You have to write that as `2*a + 2 = 2*(a+1)`.
- My solution to this problem in Cozy has subproofs nested *several* levels deep in order to prove all the $\forall$ and $\rightarrow$ parts of the claim!

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

`http://cozy.cs.washington.edu`

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

(c) [3 Points] Translate your proof to English.

This week, your English proofs will be graded not just for effort, but also for correctness. Keep in mind that your proof will be read by a human, so you need to explain the algebra steps in more detail than you would with Cozy. In particular, you do not need to explain when you simply rewrite an expression in an equivalent manner (e.g., 1a = a or ab + ac = a(b+c)). However, you should explain anywhere that you substitute one equation into another, add or multiply two equations, or add or multiply the same value on both sides of an equation to get a new one.

2. Weekend At Cape Mod (16 points)

Let domain of discourse be the integers, and let n and c be *nonzero* integers. Consider the following claim:

$$\forall a \, \forall b \, ((ca \equiv_{cn} cb) \to (a \equiv_n b))$$

This last "$\to$" is actually an "$\leftrightarrow$", but to make the problem easier, we have only asked you to prove "$\to$".

(a) [1 Point] Translate the claim into English.

(b) [12 Points] Write a **formal proof** that the claim holds.

Some important notes:

- Cozy has special notation for the predicate "$=$", but for everything else, it uses predicate notation. In particular, $a \mid b$ is written Divides(a, b), and $a \equiv_m b$ is written Congruent(a, b, m)
- You will need to make use of Cozy's `cite` rule, which cite's a known theorem. In this case, the theorem you want to use is DivideEqn, which says the following about integers:

  $$\forall a \, \forall b \, \forall c \, ((ca = cb) \wedge \neg(c = 0)) \to (a = b))$$

  Cozy's `apply` rule is an even easier way to cite and use a Theorem. See the Cozy documentation for an explanation of how to use `apply`.
- Cozy can only apply DivideEqn to an equation that looks *exactly* like $c(\dots) = c(\dots)$. For example, it cannot be applied to the equation $c = ca + cb$. Instead, you would first rewrite it as $c \cdot 1 = c(a + b)$ using the `algebra` rule and then apply DivideEqn. In particular, note that cab means $(ca)b$ in Cozy because multiplication associates to the left, so you would need to explicitly transform $cab = cde$ to $c(ab) = c(de)$ using `algeaba` before you can divide both sides by c.
- Remember that Cozy expects a "`*`" for multiplication. It will misunderstand `ca = cb`. You have to write that as `c*a = c*b` in Cozy, even though we would write it as $ca = cb$ for a human reader.

Submit and check your **formal proof** here:
`http://cozy.cs.washington.edu`
You can make as many attempts as needed to find a correct answer.

(c) [3 Points] Translate your proof to English.

This week, your English proofs will be graded not just for effort, but also for correctness. Keep in mind that your proof will be read by a human, so you need to explain the algebra steps in more detail than you would with Cozy. (Follow the same rules as in Problem 1.)

3. Where the One Don't Shine (18 points)

Let domain of discourse be the positive integers. Consider the following claim:

$$\forall a \, \forall b \, \forall m \, ((\neg(a = 1) \wedge (m = ab)) \rightarrow \neg(b = m))$$

(a) [1 Point] Translate the claim into English.

(b) [2 Points] Use a truth table or a chain of equivalences to prove that the implication above (inside all three $\forall$ quantifiers) is equivalent to

$$((m = ab) \wedge (b = m)) \rightarrow (a = 1)$$

If you use a truth table, treat each of the equations as an atomic proposition that can be either T or F. If you use equivalences, feel free to do multiple applications of Associativity and Community in one step.

Note: In the new implication, none of the facts are negated. That will make it easier for us to prove.

(c) [12 Points] Write a **formal proof** that the claim from (a) holds. Use your equivalence from part (b).

Some important notes:

- Cozy's `algebra` rule can also be used rewrite an equation in an equivalent form. For example, you can use it to rewrite $a = 1 \cdot b$ as $a = b$ or $b = a$ (or vice versa) since all three forms are equivalent.
- Cozy's `algebra` rule primarily works by adding equations or multiplying them by constants to get new equations. In particular, it cannot do the same things as Cozy's `substitute` rule. If you have an equation of the form $x = \ldots$ and you want to substitute the right-hand side for x in another equation, you have to use the `substitute` rule for that, rather than `algebra`.
- You will need to make use of Cozy's `cite` or `apply` rules once again to use a theorem that allows us to divide on both sides of an equation. In this case, since our domain is the positive integers, we can use a simpler theorem called DividePosEqn, which says the following:

$$\forall a \, \forall b \, \forall c \, ((ca = cb) \rightarrow (a = b))$$

(Here, there is no need to require $c \neq 0$ because 0 is not in the domain of discourse.)

Hint: My proof uses Cozy's `equivalent`, `substitute`, `apply`, and `algebra` rules, with the latter used more than once.

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

`http://cozy.cs.washington.edu`

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

(d) [3 Points] Translate your proof to English.

This week, your English proofs will be graded not just for effort, but also for correctness. Keep in mind that your proof will be read by a human, so you need to explain algebra or any equivalences used in more detail than you would with Cozy (though you can just cite your explanation in part (b) rather than explaining the same equivalence again here). Follow the same rules as in Problem 1.

4. A Good Prime Was Had By All (18 points)

Let domain of discourse be the positive integers. Recall the definition of prime given in lecture:

$$\text{Prime}(p) ::= \neg(p = 1) \wedge \forall x\,((x \mid p) \rightarrow (x = 1 \vee x = p))$$

Now, consider the following claim:

$$\forall p\,(\text{Prime}(p) \rightarrow \forall n\,\forall m\,((p = nm) \rightarrow (n = 1 \vee m = 1)))$$

(a) [1 Point] Translate the claim into English.

(b) [2 Points] Use a chain of equivalences to prove that the outer implication in the claim is equivalent to

$$(\exists n\,\exists m\,((p = nm) \wedge (n \neq 1) \wedge (m \neq 1))) \rightarrow \neg\,\text{Prime}(p)$$

Hint: Start by applying the Contrapositive equivalence.

(c) [12 Points] Write a **formal proof** that the claim holds. Use your equivalence from part (b).

Some important notes:

- Cozy's `equivalent` rule may not be able to see the equivalence from part (b).[1] However, you can certainly explain it to Cozy with two applications of the `equivalent` rule, where the first would should form the contrapositive and the second would do the rest of the steps.
- You can `apply` or `cite` the result you proved in Problem 3, which is now called `Lemma2` in Cozy.

Hint: To show that a number p is *not* prime, by definition, you need to either show that $p = 1$ or that there is some number x with $x \neq 1$, $x \neq p$, and $x \mid p$. You will want to show that the second of those options holds for some number x. (You can then use Intro $\vee$ to build up the definition of $\neg\text{Prime}(p)$.)

Submit and check your **formal proof** here:
`http://cozy.cs.washington.edu`
You can make as many attempts as needed to find a correct answer.

(d) [3 Points] Translate your proof to English.

This week, your English proofs will be graded not just for effort, but also for correctness. Keep in mind that your proof will be read by a human, so you need to explain algebra or any equivalences used in more detail than you would with Cozy (though you can just cite your explanation in part (b) rather than explaining the same equivalence again here). Follow the same rules as in Problem 1.

[1] We will talk about why that is later in the course.

5. A Wink and a Mod (12 points)

(a) [6 Points] Compute the multiplicative inverse of 26 modulo 101 using the Extended Euclidean Algorithm.

Your answer should be a number between 0 and 100. Show your work in tableau form: the divisions performed, the equations for the remainders, and the sequence of substitutions.

(b) [4 Points] Find all integer solutions $x \in \mathbb{Z}$ to the modular equation

$$26x \equiv_{101} 4$$

(The solutions should be all numbers of the form $x = a + 101k$ for some fixed, positive integer a, which you should calculate. For example, it could be all numbers of the form $x = 12 + 101k$.)

Give an **English proof** that every number of the form you describe is a solution and that every solution of the equation is of that form (i.e., it is a solution iff it is of that form).

(c) [2 Points] Given an **English proof** that your solutions to the modular equation in part (b) are also the solutions to the modular equation

$$26x + 12 \equiv_{101} 16$$

In Predicate Logic, the statement to be proved is $\forall x\,((26x \equiv_{101} 4) \leftrightarrow (26x + 12 \equiv_{101} 16))$. Use the structure of the formal statement to guide your English proof.

You must turn in an English proof, not a formal proof, but remember that your English proof is correct if it leaves the *reader* convinced that they could translate what you wrote into a formal proof. Your English proof be structured so that it is easy to see how to formalize it.

6. The Mod Couple (16 points)

(a) [6 Points] Compute the multiplicative inverse of 28 modulo 99 using the Extended Euclidean Algorithm.

Your answer should be a number between 0 and 98. Show your work in tableau form: the divisions performed, the equations for the remainders, and the sequence of substitutions.

(b) [4 Points] Find all integer solutions $x \in \mathbb{Z}$ to the modular equation

$$28x \equiv_{99} 15$$

Give an **English proof** that every number of the form you describe is a solution and that every solution of the equation is of that form (i.e., it is a solution iff it is of that form).

(c) [2 Points] Give a **English proof** that your solutions to the modular equation in part (b) are also the solutions to the modular equation

$$30x + 5 \equiv_{99} 2x + 20$$

You must turn in an English proof, not a formal proof, but remember that your English proof is correct if it leaves the *reader* convinced that they could translate what you wrote into a formal proof. Your English proof be structured so that it is easy to see how to formalize it.

(d) [4 Points] Give an **English proof** there are no integer solutions to the equation

$$9x \equiv_{99} 16$$

Note: This does not follow just from the fact that 9 doesn't have a multiplicative inverse modulo 99. That argument, if true, would apply to the equation $9x \equiv_{99} 18$, which actually does have solutions (e.g., $x = 2$)! Hence, a different argument is required to show that this equation has no solutions.

Hint: We are asked to prove $\neg\exists x \in \mathbb{Z}\,(9x \equiv_{99} 16)$. By De Morgan's law, this is equivalent to $\forall x \in \mathbb{Z}\,\neg(9x \equiv_{99} 16)$. To show that $\neg(9x \equiv_{99} 16)$, we can argue by contradiction: suppose that $9x \equiv_{99} 16$ holds and show that this lets us infer something we know is false. (The false claim will be that two numbers, which we can see are *definitely* different, are actually equal.)

7. Modding Off (8 points)

(a) [6 Points] Compute $3^{301} \bmod 100$ using the efficient modular exponentiation algorithm.
Show all intermediate results.

(b) [2 Points] How many multiplications does the algorithm use for this computation? (Assume that we do not need to perform a multiplication to calculate $3^1 = 3$ since we know that $x^1 = x$ for any x.)

8. Extra Credit: Walk Like an Encryption (0 points)

We know that we can reduce the *base* of an exponent modulo m : $a^k \equiv_m (a \bmod m)^k$. But the same is not true of the exponent! That is, we cannot write $a^k \equiv_m a^{k \bmod m}$. This is easily seen to be false in general. Consider, for instance, that $2^{10} \bmod 3 = 1$ but $2^{10 \bmod 3} \bmod 3 = 2^1 \bmod 3 = 2$.

The correct law for the exponent is more subtle. We will prove it in steps....

(a) Let $R = \{n \in \mathbb{Z} : 1 \leq n \leq m-1 \wedge \gcd(n, m) = 1\}$. Define the set $aR = \{ax \bmod m : x \in R\}$. Prove that $aR = R$ for every integer $a > 0$ with $\gcd(a, m) = 1$.

(b) Consider the product of all the elements in R modulo m and the elements in aR modulo m. By comparing those two expressions, conclude that, for all $a \in R$, we have $a^{\phi(m)} \equiv_m 1$, where $\phi(m) = |R|$.

(c) Use the last result to show that, for any $b \geq 0$ and $a \in R$, we have $a^b \equiv_m a^{b \bmod \phi(m)}$.

(d) Finally, prove the following two facts about the function ϕ above. First, if p is prime, then $\phi(p) = p - 1$. Second, for any primes a and b with $a \neq b$, we have $\phi(ab) = \phi(a)\phi(b)$. (Or slightly more challenging: show this second claim for *all positive integers* a and b with $\gcd(a, b) = 1$.)

The second fact of part (d) implies that, if p and q are primes, then $\phi(pq) = (p-1)(q-1)$. That along with part (c) prove the final claim from lecture about RSA, completing the proof of correctness of the algorithm.