CS230 Recitation/Assignment 02 - Proof Methods

Recitation 01/29, Assignment Due 02/04

1 Recitation

1. (Ungraded activity) Continued icebreaking and meeting each other in recitations.
2. (Warmup - translation and proof by cases) We are going to prove this theorem together:

$$\exists x \in \mathbb{R}\, \exists y \in \mathbb{R}\, [(x, y \notin \mathbb{Q}) \wedge (x^y \in \mathbb{Q})]$$

   (a) First of all, we are tired of working with excessive symbolism. Let's translate the theorem above into words.

   (b) Recall that $\sqrt{2}$ is irrational (Theorem 1.8.1 in MCS, where it is used as an example of proof by contradiction.) Let's prove the theorem above by cases. Unlike the easier examples we have seen in the peer review, the cases here are not obvious. (*Hint: the cases are not directly about x or y.)*

   - Case I.

   - Case II.

   (c) Which one of the cases is true? Does that matter?

3. (Predicate logic - continued from class meeting)

   Here are several useful inference rules in predicate logic, the first of which we have covered in class meeting as the basis of proof by construction.

   - (Existential Generalization) $P(c) \Rightarrow \exists x P(x)$
   - (Universal Instantiation) $\forall x P(x) \Rightarrow P(c)$
   - (Existential Instantiation) $\exists x P(x) \Rightarrow P(y)$
     - (Restriction: Here y must be a *new* variable that can represent any arbitrary element in the domain. By introducing y here, we are now given it a meaning: we assume the property $P(y)$ is true.)
   - (Universal Generalization) $P(y) \Rightarrow \forall x P(x)$
     - (Restriction: Here y must *not* hold *any* specific meanings. No properties of y can be assumed.)

   We have to be careful when using these inference rules. For an example, let the universal domain be $\mathbb{Z}$ and let $P(x)$ represent x is odd. We start with $P(1)$ as an accepted fact.

   (1) $P(1)$ (Assumption)

   (2) $\exists x P(x)$ (Existential Generalization on (1))

   (3) $P(y)$ (Existential Instantiation on (2))

   (4) $\forall x P(x)$ (Universal Generalization on (3))

   We have just "proven" all integers are odd. This is obviously wrong. What is the problem?

4. (De Morgan's laws for predicates - continued from class meeting) We now explicitly state the De Morgan's laws[1] for quantified predicates:

$$\neg(\forall x[P(x)]) \Leftrightarrow \exists x[\neg P(x)];$$
$$\neg(\exists x[P(x)]) \Leftrightarrow \forall x[\neg P(x)].$$

   Conceptually, you can "imagine" a special case where the domain of x is $\{x_1, x_2, \ldots, x_k\}$ (i.e., x is just one of these). Let $P(x)$ simply represent "x is true". Then these two become the general De Morgan's laws (for k propositions) that you saw in last week's recitation:

$$\neg(x_1 \wedge x_2 \wedge \ldots \wedge x_k) \Leftrightarrow \neg x_1 \vee \neg x_2 \vee \ldots \vee \neg x_k;$$
$$\neg(x_1 \vee x_2 \vee \ldots \vee x_k) \Leftrightarrow \neg x_1 \wedge \neg x_2 \wedge \ldots \wedge \neg x_k.$$

   Now, let $f(x)$ be a function defined on $\mathbb{R}$. Recall the following predicate that represents *"the limit of $f(x)$ as x approaches a exists"*: (note the underlined part which we went over in class)

$$\exists L \in \mathbb{R} \quad \forall \varepsilon > 0 \quad \exists \delta > 0 \quad \underline{\forall x \in \mathbb{R}} \quad \big[(0 < |x - a| < \delta) \rightarrow (|f(x) - L| < \varepsilon)\big]. \tag{1}$$

   Now let's write a predicate that means "the limit of $f(x)$ as x approaches a does not exist". Your final predicate *cannot* use the negation symbol $\neg$.

[1] This is not the last time you see De Morgan's laws - there are De Morgan's laws for sets too. Let's worry about those when we get there.

2 PrairieLearn Homework

Complete the CM2 Homework on PrairieLearn individually (There are 4 questions in total there).

3 Gradescope Assignment (Complete outside recitations)

1. Prove or disprove the following theorem:[2] *"For every positive integer a, there is a positive integer b in the interval $(a, 2a]$ (i.e., $a < b \leq 2a$) such that b is a power of 2."*

2. (Using lemmas to prove a theorem)

   (a) Prove the following lemma:
   Let p, q be distinct positive prime numbers. Then $p \nmid q$.

   (b) *Think about* the following lemma: (*no need to submit anything for this part*)
   Let m, n be positive integers and p be a prime number. Then $p \mid m^n \Rightarrow p \mid m$.
   Convince yourself that this lemma holds. You can use it below without formally proving it.

   (c) Using the lemmas above, prove that $\log_3 11$ is irrational.
   Hint: proof by contradiction is useful for at least one of the above parts.

3. (Proving inference rules in predicate logic) Recall that in Assignment 1, we saw the following 16 syllogisms, of which some are valid and some are invalid (you may want to look at your Assignment 1 to see which ones were deemed valid).

$$(1)\ \begin{array}{l}\forall x P(x)\\ \underline{\forall x[P(x) \rightarrow Q(x)]}\\ \therefore \forall x Q(x)\end{array} \quad (2)\ \begin{array}{l}\forall x P(x)\\ \underline{\forall x[P(x) \rightarrow Q(x)]}\\ \therefore \exists x Q(x)\end{array} \quad (3)\ \begin{array}{l}\forall x P(x)\\ \underline{\exists x[P(x) \rightarrow Q(x)]}\\ \therefore \forall x Q(x)\end{array} \quad (4)\ \begin{array}{l}\forall x P(x)\\ \underline{\exists x[P(x) \rightarrow Q(x)]}\\ \therefore \exists x Q(x)\end{array}$$

$$(5)\ \begin{array}{l}\exists x P(x)\\ \underline{\forall x[P(x) \rightarrow Q(x)]}\\ \therefore \forall x Q(x)\end{array} \quad (6)\ \begin{array}{l}\exists x P(x)\\ \underline{\forall x[P(x) \rightarrow Q(x)]}\\ \therefore \exists x Q(x)\end{array} \quad (7)\ \begin{array}{l}\exists x P(x)\\ \underline{\exists x[P(x) \rightarrow Q(x)]}\\ \therefore \forall x Q(x)\end{array} \quad (8)\ \begin{array}{l}\exists x P(x)\\ \underline{\exists x[P(x) \rightarrow Q(x)]}\\ \therefore \exists x Q(x)\end{array}$$

$$(9)\ \begin{array}{l}\forall x \neg Q(x)\\ \underline{\forall x[P(x) \rightarrow Q(x)]}\\ \therefore \forall x \neg P(x)\end{array} \quad (10)\ \begin{array}{l}\forall x \neg Q(x)\\ \underline{\forall x[P(x) \rightarrow Q(x)]}\\ \therefore \exists x \neg P(x)\end{array} \quad (11)\ \begin{array}{l}\forall x \neg Q(x)\\ \underline{\exists x[P(x) \rightarrow Q(x)]}\\ \therefore \forall x \neg P(x)\end{array} \quad (12)\ \begin{array}{l}\forall x \neg Q(x)\\ \underline{\exists x[P(x) \rightarrow Q(x)]}\\ \therefore \exists x \neg P(x)\end{array}$$

$$(13)\ \begin{array}{l}\exists x \neg Q(x)\\ \underline{\forall x[P(x) \rightarrow Q(x)]}\\ \therefore \forall x \neg P(x)\end{array} \quad (14)\ \begin{array}{l}\exists x \neg Q(x)\\ \underline{\forall x[P(x) \rightarrow Q(x)]}\\ \therefore \exists x \neg P(x)\end{array} \quad (15)\ \begin{array}{l}\exists x \neg Q(x)\\ \underline{\exists x[P(x) \rightarrow Q(x)]}\\ \therefore \forall x \neg P(x)\end{array} \quad (16)\ \begin{array}{l}\exists x \neg Q(x)\\ \underline{\exists x[P(x) \rightarrow Q(x)]}\\ \therefore \exists x \neg P(x)\end{array}$$

Now, you are required to pick some valid syllogism(s) above, and prove their validity. How do we prove the validity of a syllogism? Recall that we discussed in class that for a syllogism with two premises P_1 and P_2 and one conclusion C, its validity is shown by proving $(P_1 \land P_2) \Rightarrow C$. The task here is slightly more complicated because the premises are now *quantified predicates* rather than simple propositions. In other words, we need to operate in predicate logic. Your proofs should look similar to the bogus proof in recitation (that "proves" all integers are odd) in format (but they need to be correct!). Remember to leverage the four inference rules introduced in recitation and mind the restrictions. Remember to specify the number of each valid syllogism. ***Do one syllogism for each group member; that is, you need to pick two if completing the assignment in a pair, and otherwise just one if completing the assignment by yourself.***

- Valid syllogism 1 (Number:)
- Valid syllogism 2 (Number:)

[2] There is an amazing theorem of similar form that says there is a positive integer b in the interval $(a, 2a]$ such that b is a *prime* number. The theorem has many applications in CS (e.g., hashing) but proving that will be way beyond the scope of the course.