IIT CS536: Science of Programming

Homework 1: Course Basics, Logic

Prof. Stefan Muller

Out: Wednesday, Jan. 12
Due: Monday, Jan. 24, 11:59pm CST

This assignment contains 7 written task(s) and 3 task(s) for which you'll submit logic proofs from the online tool, for a total of 65 points.

SOLUTIONS

Logistics

Submission Instructions

Please read and follow these instructions carefully.

- Submit your homework on Blackboard under the correct assignment by the deadline (or the extended deadline if taking late days).
- You may submit multiple times, but we will only look at your last submission. Make sure your last submission contains all necessary files.
- Email the instructor and TA ASAP if
  - You submit before the deadline but then decide to take (more) late days.
  - You accidentally resubmit after the deadline, but did not intend to take late days.

  Otherwise, you do not need to let us know if you're using late days; we'll count them based on the date of your last submission.
- Submit your written answers in a single PDF or Word document. Typed answers are preferred (You can use any program as long as you can export a .pdf, .doc or .docx; LaTeX is especially good for typesetting logic and math, and well worth the time to learn it), but *legible* handwritten and scanned answers are acceptable as well.
- Your Blackboard submission should contain exactly the file with your written answers and all of the .log files with your proofs (with the names given in each problem); *do not rename these files.* Do not compress or put any files in folders.

Collaboration and Academic Honesty

Read the policy on the website and be sure you understand it.

1 Course Basics

Task 1.1 (Written, 10 points).

For each statement, does it describe verification, testing, both, or neither?

a) When you do this, the program is guaranteed to work as intended.

b) The code's been in use for 10 years and nobody's seen a bug.

c) I have a proof that the code will meet its specification.

d) I expect the code to produce an integer, so I wrote it in a statically typed language, annotated the code with the type "int" and it compiled.

e) Doing this on a piece of code can give you more confidence that it will work correctly.

Note that the answer "both" for (a) would be intended to be read "When you do either verification or testing, the program is guaranteed to work as intended" and the answer "neither" would be intended to be read "Neither verification nor testing is sufficient to guarantee the program works as intended.", and similar readings for (e).

a) Neither

b) Testing

c) Verification

d) Verification

e) Both

Task 1.2 (Written, 10 points).

For each pair of propositions, say whether they are

(I) syntactically equal ($\equiv$) (and therefore also semantically equal)

(II) semantically equal ($=$) but not syntactically equal, or

(III) neither

a)	$p \land q \lor r$	$r \lor q \land p$
b)	$p \land q \lor r$	$p \land r \lor q$
c)	$\neg p \lor q \to \neg(p \land \neg q)$	$((\neg p) \lor q) \to \neg(p \land (\neg q))$
d)	$p \to q$	$\neg p \to \neg q$
e)	$p \land \neg$ p	F

a) $=$

b) Neither

c) $\equiv$

d) Neither

e) $=$

2 Propositional Logic

Task 2.1 (Written, 9 points).

Say whether each of the following is a tautology, contradiction or contingency. If it's a contingency, give a state where the proposition is true and a state where it's false.

a) $(P \to (Q \to R)) \leftrightarrow ((P \to Q) \to R)$

b) $(P \to Q) \leftrightarrow (\neg Q \to \neg P)$

c) $\neg P \wedge Q \leftrightarrow \neg Q \wedge P$

a) Contingency. True: $\{P = T, Q = T, R = T\}$. False: $\{P = F, Q = T, R = F\}$

b) Tautology.

c) Contingency. False: $\{P = F, Q = T\}$. True: $\{P = F, Q = F\}$

Task 2.2 (Programming, 5 points).

Prove that $((P \to Q) \wedge \neg P) \vee P$ is a tautology by proving $T \Rightarrow ((P \to Q) \wedge \neg P) \vee P$.

Important: Submit your proof (and only your proof) in the file `tauto.log`. It should be in a format checkable by the proof checker at `http://www.cs.iit.edu/~smuller/cs536-s22/logic/proofs.html`. That is, *you should be able to copy and paste the entire contents of your file into the text box of the proof checker, click "Check" and have it respond that the proof is correct.* To help you, you can select "HW1 Task 2" from the dropdown box on the proof checker page to load the editor with the statement above, and work on your proof there. When you're done, click "Download" to download the finished proof.

Task 2.3 (Written, 3 points).

In the previous task, why didn't we prove that the statement was a tautology by proving $((P \to Q) \wedge \neg P) \vee P \Rightarrow T$?

Everything logically implies T.

Task 2.4 (Programming, 5 points).

Prove the following implication. **Submit your proof in the file `uncurry.log`, following the same instructions as in Task 2.**

$$P \to (Q \to R) \Rightarrow (P \wedge Q) \to R$$

(Functional programmers may recognize this as *uncurrying*.)

3 Predicate Logic

Task 3.1 (Written, 4 points).

Let $A \subset \mathbb{Z}$ (that is, A is a subset of the integers). Write the proposition that formally expresses that A is finite (that is, it has a minimum and a maximum element). You can quantify universally and existentially over A in the same way that you do over $\mathbb{Z}$ ($\forall x \in A.P(x)$, $\exists y \in A.Q(y)$, ...). You can use standard mathematical symbols ($<$, $>$, $\leq$, $\geq$, $=$, etc.).

$\exists m \in A.\exists n \in A.\forall x \in A.m \leq x \wedge x \leq n$

Task 3.2 (Written, 9 points).

Are the following statements true (provable) or false (have a counterexample)? Explain briefly (1-2 sentences).

a) $\forall x \in \mathbb{Z}.\exists y \in \mathbb{Z}.x = y * 2$

b) $\exists x \in \mathbb{Z}.\neg(\exists a \in \mathbb{Z}.\exists b \in \mathbb{Z}.a > 1 \wedge b > 1 \wedge a * b = x)$

c) $\forall x \in \mathbb{Z}.\exists y \in \mathbb{Z}.y = x * 2$.

a) False. 1 is a counterexample.

b) True. This says there exists a prime number.

c) True. This says every number can be multiplied by 2.

Task 3.3 (Programming, 10 points).

Prove the following implication. **Submit your proof in the file `pred.log`, following the same instructions as in Task 2.**

$$(\forall x.P(x)) \rightarrow (\forall y.Q(y)) \Rightarrow (\exists y.\neg Q(y)) \rightarrow (\exists x.\neg P(x))$$

4 One more wrap-up question

Task 4.1 (Written, 0 points).

How long (approximately) did you spend on this homework, in total hours of actual working time? Your honest feedback will help us with future homeworks.