Homework 1
Propositional Sequent Calculus

15-316: Software Foundations of Security & Privacy

Due Tuesday, January 27, 2026
60 points

Your solution should be handed in as file `hw1.pdf` to Gradescope. If at all possible, write your solution in LaTeX. The handout `hw1-proof.zip` includes the LaTeX sources for Lecture 2 (`02-prop.tex`) and the necessary style files which provide some examples for rules, derivations, and proofs.

1 Rule Design (40 points)

In this problem we ask you to give right and left sequent calculus rules for some logical constants and a new connective. Whatever rules you give should preserve all the good properties of the sequent calculus for propositional logic, that is:

(a) They should be *sound*.

(b) They should be *invertible*.

(c) All their premises should be smaller than the conclusion, when counting the number of logical connectives and constants $\top$ and $\bot$ in a sequent.

The last two items combine to entail completeness and decidability. **You only need to prove these properties when explicitly asked but your rules should nonetheless satisfy them. When you do write out proofs, please follow the template for $\to L$ (soundness, page L3, Section 9) and $\vee R$ (invertibility, page L3, Section 10) in the lecture notes to make them easy for us to grade.**

Task 1 (5 points) *Give right and left rules for the logical constant $\top$ (truth).*

Task 2 (5 points) *Give right and left rules for the logical constant $\bot$ (falsehood).*

We define a new connective $F \barwedge G$ by the following truth table.

F	G	$F \barwedge G$
$\top$	$\top$	$\bot$
$\top$	$\bot$	$\top$
$\bot$	$\top$	$\top$
$\bot$	$\bot$	$\top$

In the remainder of this problem we ask you to give right and left rules for this new connective. These rules should preserve all the good properties listed at the beginning of the problem.

Task 3 (5 points) *Give the right rule or rules for* $\overline{\wedge}$.

Task 4 (5 points) *Prove your right rule(s) are* sound.

Task 5 (5 points) *Prove your right rule(s) are* invertible.

Task 6 (5 points) *Give the left rule or rules for* $\overline{\wedge}$.

Task 7 (5 points) *Prove your left rule(s) are* sound.

Task 8 (5 points) *Prove your left rule(s) are* invertible.

2 Arbitrary Assignment (20 points)

Sometimes we can make assumptions about a variable whose specific value is unknown when we are reasoning about a program. For example, we might write a program that accepts alphanumeric user input, so we know which subset of characters the input could contain. In this problem, we will explore how to do this in dynamic logic.

Task 9 (5) *Extend the language discussed in lecture by defining the semantics of a constrained assignment command,* $x := Q(x)$.

Informally, this command should assign an arbitrary value to x that satisfies the formula $Q(x)$. We use the notation $Q(x)$ means that Q is a formula with a free variable x. For example, after running $x := x > y$, the variable x could be assigned any integer greater than y in the current state. If $Q(x)$ is not satisfiable, e.g. if $Q(x)$ is equivalent to $x < 0 \wedge x > 0$, then the command should not enter any final state (i.e., should not terminate).

Task 10 (5) *Design an axiom that allows you to reason about box modalities around this form of assignment:*

$$[x := Q(x)]p(x) \leftrightarrow \dots$$

The right side of this equivalence should not contain a box or diamond modality.

Task 11 (5) *Prove that your axiom from Task 2 is valid using your semantics from Task 1.*

Task 12 (5) *Based on your axiom, propose left and right proof rules for the arbitrary assignment statement.*