

Assignment 1: Prop it Out
15-414/15-424 Bug Catching: Automated Program Verification

Due: **11:59pm**, Thursday 9/7/17

Total Points: 50

1. **Table practice (3 points)** Give the truth table for the following formula and describe what it tells you:

$$(\neg p \rightarrow p) \rightarrow p$$

2. **Table counting (3 points)** How many rows and columns does the truth-table for the following formula have? Why?

$$\left(\left(\left((p \rightarrow q) \rightarrow (\neg r \rightarrow \neg s) \right) \rightarrow r \right) \rightarrow t \right) \rightarrow \left((t \rightarrow p) \rightarrow (s \rightarrow p) \right)$$

3. **Proof practice (5 points)** Conduct a proof in sequent calculus of the following formula. Be sure to say which proof rule you apply at each step.

$$(a \rightarrow b) \wedge (b \rightarrow c) \wedge (\neg d \rightarrow \neg c) \rightarrow (a \rightarrow b \wedge d)$$

4. **Soundness of \neg L (5 points)** Prove that \neg -L rule is sound. That is, using the semantics of \neg , prove that validity of all its premises implies validity of its conclusion.

$$(\neg\text{L}) \frac{\Gamma \vdash F, \Delta}{\Gamma, \neg F \vdash \Delta}$$

5. **Biimplication (6 points)** The syntax of propositional logic provided the biimplication/bisubjunction operator $A \leftrightarrow B$ which is true iff both A and B have the same truth-value, so both are true or both are false. But this operator is missing sequent calculus proof rules.

Your task is to design proof rules for when the equivalence operator is used in the succedent (right rule \leftrightarrow R) and when it is used in the antecedent (left rule \leftrightarrow L):

$$(\leftrightarrow\text{R}) \frac{\dots}{\Gamma \vdash F \leftrightarrow G, \Delta} \quad (\leftrightarrow\text{L}) \frac{\dots}{\Gamma, F \leftrightarrow G \vdash \Delta}$$

6. **Use it! (5 points)** Use your proof rules \leftrightarrow R and \leftrightarrow L to conduct a sequent calculus proof for the formula:

$$A \wedge B \leftrightarrow B \wedge A$$

7. **Use it again! (5 points)** Use your proof rules \leftrightarrow R and \leftrightarrow L to conduct a sequent calculus proof for the formula:

$$(A \leftrightarrow B) \wedge A \rightarrow B$$

8. **Soundness of \leftrightarrow (8 points)** Proof rules cannot be used unless they are accompanied by a soundness proof. Quickly before anybody notices your answer to the previous two tasks, use the semantics of the biimplication operator to prove soundness of your proof rules \leftrightarrow R and \leftrightarrow L. That is, for each of the rules, prove that validity of all its premises implies validity of its conclusion.

9. **Revisiting soundness of propositional logic (3 points)** What do you need to change in the proof of the soundness theorem for propositional logic (Theorem 7 in the lecture notes) now that you have added proof rules \leftrightarrow L and \leftrightarrow R?

10. **Induction on lists (7 points)** Recall that the list datatype encodes the concept of a sequence of zero or more values of the same type. We can define lists recursively by first specifying a constructor `nil` for the case of the empty list, and then giving a constructor `::` for lists with at least one element:

```
datatype 'a list = nil
  | :: of 'a * 'a list
```

The following SML function then defines the length of a list recursively by matching each constructor:

```
fun length nil = 0
  | length (h::t) = 1 + length t
```

Prove by induction on the structure of the list that it returns the correct number. That is, show the result for the base case of empty lists `nil`. Then assuming that the function is already implemented correctly for smaller lists `l`, show that it is also implemented correctly for the list `h :: t` consisting of head `h` followed by list `t`.