Name: ____________________________

Logic and Mechanized Reasoning

First Midterm Exam

September 28, 2021

Write your answers in the space provided, using the back of the page if necessary. You may use additional scratch paper. Justify your answers, and provide clear, readable explanations.

Problem	**Points**	**Score**
1	20	
2	10	
3	20	
4	8	
5	18	
6	12	
Total	**88**	

Good luck!

Problem 1. (20 points) *Propositions*: Answer each question with true or false.

		$\top$	$\bot$
(a)	Every propositional formula with exactly three satisfying assignments is falsifiable.	$\top$	
(b)	For all propositional formulas A, B, C it holds that $$\{A, \neg A\} \models (A \to (\neg B \leftrightarrow C) \wedge (\neg C \vee B)) \to \neg A$$	$\top$	
(c)	Let I be an inductively defined set and let f be defined by structural recursion on I. Then f is a total function.	$\top$	
(d)	If A and B are any formulas in negation normal form, the formula $(A \vee \neg B) \wedge \neg A$ is also in negation normal form.		$\bot$
(e)	For some formulas in negation normal form, there exists a logically equivalent formula in negation normal form that is exponentially smaller.	$\top$	
(f)	If p is the only variable occurring in a propositional formula, then the formula is in negation normal form.		$\bot$
(g)	For an element a of type α it holds that $[a] :: [] = [[a]]$.	$\top$	
(h)	The expression $[[1, 2, 3], 4, 5]$ has type *List* (*List* $\mathbb{N}$).		$\bot$
(i)	A recursive definition of F with a recursive case $F_{N+2} = F_{N+1} - F_N$ can have only one base case.		$\bot$
(j)	Structural recursion formulates a definition by recursion on the structure of an inductively defined type.	$\top$	

Problem 2. (10 points) Prove the following:

$$1 + 2 + 4 + \ldots + 2^n = 2^{n+1} - 1.$$

Solution

Use induction on n. When $n = 0$, we have $1 = 1$. Assuming the claim holds for n, we have

$$\begin{aligned}1 + 2 + 4 + \ldots + 2^n + 2^{n+1} &= (2^{n+1} - 1) + 2^{n+1} \\ &= 2 \cdot 2^{n+1} - 1 \\ &= 2^{n+2} - 1.\end{aligned}$$

Problem 3. The following Lean code defines a type of labeled binary trees and two functions. Here `allLe t n` returns `true` if every node of tree `t` is less than or equal to `n` and `false` otherwise, and `allGe t n` returns `true` if `n` is greater than or equal to every node of tree `t` and `false` otherwise.

```
inductive LBinTree (α : Type)
| empty : LBinTree α
| node  : α → LBinTree α → LBinTree α → LBinTree α

open LBinTree

def allLe : LBinTree Nat → Nat → Bool
| empty, _       => true
| node m s t, n => m ≤ n && allLe s n && allLe t n

def allGe : LBinTree Nat → Nat → Bool
| empty, _       => true
| node m s t, n => n ≤ m && allGe s n && allGe t n
```

Part a) (10 points) Define a function `isOrdered : LBinTree Nat → Bool` that determines whether the inorder traversal yields a list of natural numbers that is nondecreasing (each element is less than or equal to the one that follows it).

Solution

See `midterm1.lean`.

Part b) (10 points) Define a function `insert : Nat → LBinTree Nat → LBinTree Nat` such that if `n` is a natural number and `t : LBinTree Nat` is already in order, then `insert n t` inserts `n` in `t` in the right place. You can assume that when this function is called, `isOrdered t` is `true`.

Solution

See `midterm1.lean`.

Problem 4. (8 points) Prove the following statement (directly, from the semantic definitions) or find a counterexample: For any set of propositional formulas Γ and for any propositional formulas A and B, if $\Gamma \models A$ and $\Gamma \models B$ then $\Gamma \models A \wedge B$.

Solution

Suppose $\Gamma \models A$ and $\Gamma \models B$. Let τ be any truth assignment such that for every formula C in Γ, $[\![C]\!]_\tau = \top$. Since $\Gamma \models A$, we have $[\![A]\!]_\tau = \top$, and since $\Gamma \models B$, we have $[\![B]\!]_\tau = \top$. By the definition of truth for propositional formulas, we have $[\![A \wedge B]\!]_\tau = \top$. So $\Gamma \models A \wedge B$.

Problem 5. Professor Lean has a dress code problem. He is aware that wearing neither a tie nor a shirt is impolite. It is clear that one should not wear a tie without a shirt. Personally, he considers wearing a tie and a shirt as overkill. How can we solve this problem?

Part a) (6 pt) Express the dress code problem as a propositional formula D using two propositional variables: s for wearing a shirt and t for wearing a tie.

Solution

$$\neg(\neg t \wedge \neg s) \wedge \neg(t \wedge \neg s) \wedge \neg(t \wedge s)$$

or

$$(t \vee s) \wedge (t \rightarrow s) \wedge \neg(t \wedge s).$$

(Other solutions are possible.)

Part b) (6 pt) Convert D into an equivalent formula in conjunctive normal form.

Solution

$$(t \vee s) \wedge (\neg t \vee s) \wedge (\neg t \vee \neg s)$$

Part c) (6 pt) Is the formula D satisfiable? Either answer *unsatisfiable* or provide a satisfying assignment.

Solution

Define τ by $\tau(s) = \top$ and $\tau(t) = \bot$. Then D is satisfied by τ.

Problem 6. (12 points) Put the following sentence in disjunctive normal form:

$$(\neg(p \wedge q) \to r) \wedge (r \to q).$$

Solution

The expression is equivalent to

$$((p \wedge q) \vee r) \wedge (\neg r \vee q)$$

which is equivalent to

$$(p \wedge q \wedge \neg r) \vee (r \wedge \neg r) \vee (p \wedge q \wedge q) \vee (r \wedge q)$$

and hence

$$(p \wedge q \wedge \neg r) \vee (p \wedge q) \vee (r \wedge q).$$

This last line simplifies to

$$(p \wedge q) \vee (r \wedge q).$$

Using a truth table gives

$$(p \wedge q \wedge r) \vee (p \wedge q \wedge \neg r) \vee (\neg p \wedge q \wedge r),$$

which also simplifies to the same thing.