Formal Methods for High-Assurance Software Engineering

HomeWork Assignment 01

Shahin Roozkhosh
shahin@bu.edu
U01670169

September 2020

Problem 1. The Boolean majority function over three variables $\{x, y, z\}$ returns true if at least two of the three variables are assigned true, and returns false otherwise.

1. Write a DNF (Disjunctive Normal Form) ϕ of the majority function over three variables {x, y, z}.

Solution. $\phi ::= (x \wedge y) \vee (x \wedge z) \vee (y \wedge z)$

2. Write a CNF (Conjunctive Normal Form) ψ of the majority function over three variables {x, y, z}.

Solution. $\psi ::= (x \vee y) \wedge (x \vee z) \wedge (y \vee z)$

3. Write a script to certify with Z3 that ϕ and ψ define the same function, i.e., the wff $\phi \leftrightarrow \psi$ is a tautology. Your script should use the three variables x, y, z and start with the declarations:

(declare-const x Bool)
(declare-const y Bool)
(declare-const z Bool)

Solution. In order to show that some wff is a tautology, it is sufficient to show that the negation of that wff is unsatisfiable. Hence in the z3 script we assert $!(\phi \leftrightarrow \psi)$ and expect to see "unsat" on output.

```
; Boolean majority function over three
; variables {x,y,z}
```

```
(echo "Hello Z3 world!")

(declare-const  x Bool)
(declare-const  y Bool)
(declare-const  z Bool)

(declare-fun phi (Bool Bool Bool) Bool)
(assert (= (phi x y z) (or (and x y)(or (and x z)(and y z)))))

(declare-fun psi (Bool Bool Bool) Bool)
(assert (= (psi x y z) (and (or x y)(and(or x z)(or y z)))))

;(assert (= (phi x y z) (psi x y z)))
(assert (not (= (phi x y z) (psi x y z))))

(check-sat)
;(get-model)
```

4. Write a script to certify with Z3Py that ϕ and ψ define the same function. Your script should use the three variables {x, y, z} and start with the declarations:

$x, y, z = Bools(`x\ y\ z')$

Solution. In order to show that some wff is a tautology, it is sufficient to show that the negation of that wff is unsatisfiable. Hence in the z3 script we assert $!(\phi \leftrightarrow \psi)$ and expect to see "unsat" on output.

```
from z3 import *

x,y,z = Bools('x y z')

phi = Or (And(x,y), And(x,z), And(y,z))
psi = And (Or(x,y), Or(x,z), Or(y,z))

s = Solver()
s.add(phi != psi)

print s.check()
```

Remark 1: You will easily find the answers for **Problem 1** by searching the Web. It is perfectly acceptable if you search for the answers, but make sure the answers you find are correct (they are often wrong on the Web!).

Remark 2: In parts 3 and 4, we ask you to write the scripts only, we do not ask you to execute the scripts. But of course, if you wish, you may decide to also execute them to make sure they are bug-free, before inserting them in your Latex source file. In the Latex source file, you should insert each of your scripts inside a *verbatim* environment, i.e., insert them between $\backslash$*begin*$\{$*verbatim*$\}$ and $\backslash$*end*$\{$*verbatim*$\}$.

Problem 2. [LCS, page 80]: Exercise 1.2.5

1.2.5 (a)

Solution.
Natural Deduction for: $((p \to q) \to q) \to ((q \to p) \to p))$

1	$(p \to q) \to q$	premise
2	$q \to p$	assume
3	$\neg p$	assume
4	p	assume
5	$\bot$	$\neg$e $4, 3$
6	q	$\bot$e 5
7	$p \to q$	$\to$i $4 - 6$
8	q	$\to$e $1, 7$
9	p	$\to$e $2, 8$
10	$\bot$	$\neg$e $9, 3$
11	$\neg\neg p$	$\neg$i $3 - 10$
12	p	$\neg\neg$e 11
13	$(p \to q) \to p$	$\to$i $2 - 12$

1.2.5 (c)

Solution.
Natural Deduction for: $((p \to q) \land (q \to p)) \to ((p \lor q) \to (p \land q))$

1	$((p \to q) \land (q \to p))$	premise
2	p	assume
3	$p \to q$	$\land e_1, 1$
4	q	$\to e, 2, 3$
5	$p \land q$	$\land i, 2, 4$
6	$p \to (p \land q)$	$\to i, 2-5$
7	q	assume
8	$q \to p$	$\land e_2, 1$
9	p	$\to e, 7, 8$
10	$p \land q$	$\land i, 7, 9$
11	$q \to (p \land q)$	$\to i, 7-10$
12	$p \lor q$	assume
13	$p \land q$	$\lor e, 6, 11, 12$
14	$(p \lor q) \to (p \land q)$	$\to i, 12-13$

1.2.5 (d)

Solution.
Natural Deduction for: $(p \to q) \to ((\neg p \to q) \to q)$

1	$(p \to q)$	premise
2	$\neg p \to q$	assume
3	$p \vee \neg q$	LEM
4	p	assume
5	q	$\to\mathcal{E}$ 1, 4
6	$p \to q$	$\to$i 4 – 5
7	$\neg p$	assume
8	q	$\to$e 2, 7
9	$\neg p \to q$	$\to$i 7 – 8
10	q	$\vee$e 3, 6, 9
11	$(\neg p \to q) \to q)$	$\to$e 2 – 10

Problem 3. [LCS, page 81]: Exercise 1.2.8

Solution. Deriving natural deduction rules of negation by banning $\neg$ operand from propositional logic and think of $\phi \to \bot$ as 'being' $\neg\phi$.
Deriving $\neg i$

1	ϕ $\vdots$ $\bot$	Given
2	$\phi \to \bot$	$\to$i, 1

The above **outer** box is just for neatness and I don't mean local assumption.

Deriving $\neg e$

1	ϕ	Given
2	$\phi \to \bot$	Given
3	$\bot$	$\to$e, 1, 2

The above **outer** box is just for neatness and I don't mean local assumption.

Deriving $\neg\neg i$

1	ϕ	Given
2	$\phi \to \bot$	Premise
3	$\bot$	$\to$e, 1, 2
4	$(\phi \to \bot) \to \bot$	$\to$i, $2-3$

Deriving $\neg\neg e$
$\neg\neg e$ cannot be simulated

Problem 4. [LCS, page 86]: Exercise 1.4.9

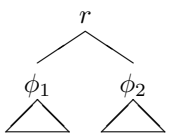
Solution. According to the LCS page 44 Definition 1.32: Given a well-formed formula ϕ, we define its height to be 1 plus the length of the longest path of its parse tree.

Proving by induction on height of ϕ that $rank(\phi)$ is height of ϕ. Let height of ϕ to be $H(\phi)$.

Base Case: Show it is true for $H(\phi) = 1$. Since the definition of the height of a tree is 1 plus the length of the longest path of its parse tree, if for some wff ϕ, $H(\phi) = 1$ is equal to one, then the longest path of it's parse tree is Zero. Hence that wff only consists of an atom p. More formally: $\phi ::= p$. According to the definition, $rank(p) = 1$. Hence it is evident that in this case ($\phi ::= p$), $rank(\phi) = rank(p) = H(p) = 1$.

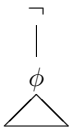
Inductive Step: Let us assume that the statement holds for n, meaning that for all wff's like ϕ with the maximum height of n, $rank(\phi) = H(\phi) = n$. We show that for any ψ with the height equal to $n+1$, $rank(\psi) = n+1$. Let r be the root of the parse tree of ψ where $r \in \{\to, \wedge, \vee, \neg\}$.

Case 1: $r \in \{\to, \wedge, \vee\}$ In this case, without loss of generality, since $H(\psi) = n+1$ we can show the parse tree of ψ as following with the root r and left and right sub-trees with the maximum height of n (and minimum height of 0).



As the problem states: $rank(\phi \circ \psi) = 1 + max(rank(\phi), rank(\psi))$. Moreover, since $H(\psi) = n+1$ by looking at the figure above it is intuitive that $H(\psi) = max\{H(\phi_1), H(\phi_2)\} + 1$ and $max\{H(\phi_1), H(\phi_2)\} = n$. According to the assumption of the induction since $H(\phi_1), H(\phi_2) \leq n$ then $rank(\phi_1) = H(\phi_1)$ and $rank(\phi_2) = H(\phi_2)$. This leads to: $rank(\phi \circ \psi) = 1 + max(rank(\phi), rank(\psi)) = 1 + max\{H(\phi_1), H(\phi_2)\} = 1 + n = H(\psi)$.

Case 2: $r = \neg$ In this case, for some ϕ, $\psi = \neg\phi$. Since $H(\psi) = n + 1$ and $\psi = \neg\phi$ then $H(\phi) = n$ (look at the figure below). According to the assumption of the induction since $H(\phi) = n$ then $rank(\phi) = H(\phi) = n$. Also, according to the problem $rank(\neg\phi) = 1 + rank(\phi)$ which is equal to $rank(\neg\phi) = 1 + rank(\phi) = 1 + n$.



According to the case1 and case2, the induction proves. □

Problem 5. [LCS, page 87]: Exercise 1.5.3

Solution. In this solution we will implicitly use the De Morgan's Law, hence we define it as a lemma for any wff P and wff Q:

Lemma:
De Morgan's Law (negation of conjunction): $\neg(P \wedge Q) \equiv \neg P \vee \neg Q$
De Morgan's Law (negation of alternative): $\neg(P \vee Q) \equiv \neg P \wedge \neg Q$

1.5.3 (a)
$\{\neg, \vee\}$ is adequate beause:

- $\phi \rightarrow \psi \equiv \neg\phi \vee \psi$
- $\phi \wedge \psi \equiv \neg(\neg\phi \vee \neg\psi)$

$\{\neg, \wedge\}$ is adequate beause:

- $\phi \to \psi \equiv \neg(\phi \vee \neg\psi)$
- $\phi \vee \psi \equiv \neg(\neg\phi \wedge \neg\psi)$

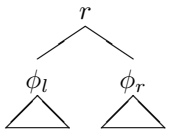
$\{\neq, \to\}$ is adequate beause:

- $\phi \wedge \psi \equiv \neg(\neg\phi \vee \neg\psi) \equiv \neg(\phi \to \neg\psi)$
- $\phi \vee \psi \equiv \neg\phi \to \psi$

$\{\to, \bot\}$ is adequate beause: (let $\bot \equiv F$)

- $\neg\phi \equiv \phi \to \bot \equiv \phi \to \ F$
- $\phi \wedge \psi \equiv \neg(\phi \to \neg\psi) \equiv (\phi \to \neg\psi) \to \bot \equiv (\phi \to (\psi \to \bot)) \to \bot$
- $\phi \vee \psi \equiv \neg\phi \to \psi \equiv (\phi \to \bot) \to \psi$

1.5.3 (b) Suppose C contains neither $\neg$ nor $\bot$. Let ϕ to be a wff in this language with the parse tree with the **shortest** height in the language, which $\phi \equiv F$. Since ϕ is a member of this language, then it's well formed formula only consists of atoms and operands in $\{\wedge, \vee, \to\}$. Consider the parse tree of ϕ; th root of the tree is either $\wedge$ or $\vee$ or $\to$ where none of them is a unary operator. Hence, we can safely assume that the pars tree of ϕ is in the following form:

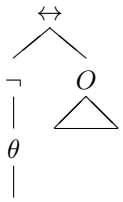


Remember that we assumed that ϕ is the shortest wff in this language with the truth value of "False" and it's intuitive that the height of left and right sub-trees ϕ_l, ϕ_r are strictly less than the height of the parse-tree of ϕ. Hence $\phi_l \equiv \ T$ and $\phi_r \equiv T$ while ϕ itself is equivalent to F. By considering all three cases of r: $\wedge$ or $\vee$ or $\to$, the truth value of ϕ evaluates to either $T \wedge T, T \vee T, T \to T$ respectively and none of them results in a False value. Hence, we reach a contradiction and we can conclude that such language without $\neg$ or $\bot$ could not be adequate. $\square$

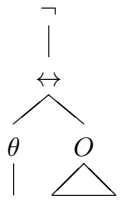
1.5.3 (c) Is $\{\leftrightarrow, \neg\}$ adequate?
The answer is no.

Lemma: $\neg(x \leftrightarrow y) \equiv (\neg x \leftrightarrow y) \equiv (x \leftrightarrow \neg y)$

Let's assume that it is adequate. Then there should be a representation of $\wedge$ in this language. Consider the wff $\phi \equiv (p \wedge\ q)$ in our normal language $\{\wedge, \vee, \rightarrow, \neg\}$ where p and q are atoms. And let ψ to be the representation of ϕ in this language. Hereby, we define a **step**. According to the Lemma above, we can safely replace any occurrence of form $\neg x \leftrightarrow y$ (or $x \leftrightarrow \neg y$ accordingly) with $\neg(x \leftrightarrow y)$. In other words, for any wff in the language and the parse-tree attributed to it, we can replace any occurrence of



with



Where $O \in \{\neg, \leftrightarrow\ \varnothing\}$ and θ is a wff (or $\varnothing$) in the language. Hence it is intuitive (and could be proven by induction if it is not clear or you are picky) that $\psi \equiv \phi \wedge\ q$ could transform to some form of:

$$\neg \cdots \neg\theta \equiv \neg \cdots \neg(p \leftrightarrow q \leftrightarrow \cdots)$$

Where the number of the negations on the left is $0 \leq$ and θ consists of atoms and $\leftrightarrow$ only. By looking at the truth table of θ. Remember this is a representation of ψ and we picked $\psi \equiv \phi p \wedge\ q$. We expect letting $p \equiv T$ and $q \equiv T$ to lead to T while $p \equiv F$ and $q \equiv F$. This is clearly according to the , $\psi \equiv \theta \equiv \neg \cdots \neg\theta \equiv \neg \cdots \neg(p \leftrightarrow q \leftrightarrow \cdots)$ returns a **same** value for both assignment of all atoms to T and assignment of all atoms to F. Hence we reach a contradiction and we just showed that $\wedge\ q$ could not be simulated in this language. $\square$

Problem 6. [LCS, page 87]: Exercise 1.5.4

Solution. Use soundness or completeness to show that a sequent $\phi_1, \phi_2, ...\phi_n \vdash \psi$ has a proof iff $\phi_1 \to \phi_2 \to \cdots \phi_n \to \psi$ is a tautology.

Definition of tautology: A formula of propositional logic ϕ is called a tautology iff it evaluates to T under all it's valuations, i.e. iff $\models \phi$

Definition of soundness: Let $\phi_1, \phi_2, ...\phi_n$ and ψ be propositional logic formulas. If $\phi_1, \phi_2, ...\phi_n \vdash \psi$ is valid, then $\phi_1, \phi_2, ...\phi_n \models \psi$ holds. (Textbook Theorem 1.35)

Definition of Completeness: Let $\phi_1, \phi_2, ...\phi_n$ and ψ be propositional logic formulas. Whenever $\phi_1, \phi_2, ...\phi_n \models \psi$ holds, then there exists a natural deduction proof for the sequent $\phi_1, \phi_2, ...\phi_n \vdash \psi$.

Sounndness and Completeness: Let $\phi_1, \phi_2, ...\phi_n$ and ψ be propositional logic formulas. then $\phi_1, \phi_2, ...\phi_n \models \psi$ holds iff the sequent $\phi_1, \phi_2, ...\phi_n \vdash \psi$ is valid. (Textbook Corollary 1.39)
proving if $\phi_1, \phi_2, ...\phi_n \vdash \psi$ has a proof then $\phi_1 \to \phi_2 \to \cdots \phi_n, \to \psi$ is a tautology:

$\phi_1, \phi_2, ...\phi_n \vdash \psi$ has a proof, then $\phi_1, \phi_2, ...\phi_n \vdash \psi$ holds. Then based on Soundness and Completeness Theorem, $\phi_1, \phi_2, ...\phi_n \models \psi$. This means that for "every" assignment of true values to all $\phi_1, \phi_2, ...\phi_n$, then ψ evaluates to a true value as well.

Assume for the sake of contradiction that $\phi_1 \to \phi_2 \to \cdots \phi_n \to \psi$ is **not** a tautology, then there exists an assignment of truth-values to $\phi_1, \phi_2, ...\phi_n$ and ψ such that $\phi_1 \to \phi_2 \to \cdots \phi_n \to \psi$ evaluate to a False value. let's name the ψ to θ_{n+1} (Nothing fancy just a rename). Then we are assuming (by contradiction) that $\phi_1 \to \phi_2 \to \cdots \phi_n \to \phi_{n+1}$ evaluates to False.

If for every $1 \leq j \leq n$, $\phi_j \equiv T$ then based on what we said two paragraph above, $\psi \equiv T$ as well because $\phi_1, \phi_2, ...\phi_n \models \psi$. Then for $\phi_1 \to \phi_2 \to \cdots \phi_n \to \phi_{n+1}$ to be False, there should exists an $1 \leq i \leq n$ such that $\phi_i \equiv F$. let $1 \leq i \leq n$ to be the **smallest** value such that $\phi_i \equiv F$ (note that $i \neq n+1$). Then It's safe to rewrite:

$$\phi_1 \to \phi_2 \to \cdots \phi_n, \to \psi \equiv \phi_1 \to \phi_2 \to \cdots \phi_n \to \phi_{n+1} \equiv \phi_1 \to \phi_2 \to \cdots \to \phi_i \to \cdots \to \phi_n \to \phi_{n+1} \equiv \phi_1 \to \phi_2 \to \cdots \to F \to \cdots \phi_n \to \phi_{n+1} \equiv (\phi_1 \to \phi_2 \to \cdots \to F) \to \cdots \phi_n \to \phi_{n+1}$$

However, since we picked i to be the **smallest** value such that $\phi_i \equiv F$ then for every $j \leq i, \phi_j \equiv T$, hence $(\phi_1 \to \phi_2 \to \cdots \to F) \equiv T \to T \to \cdots \to T \to$

$F \equiv F$ (even for the case that $i = 1$) Hence
:

$\phi_1 \to \phi_2 \to \cdots \phi_n, \to \psi \equiv \phi_1 \to \phi_2 \to \cdots \phi_n \to \phi_{n+1} \equiv \phi_1 \to \phi_2 \to \cdots \to \phi_i \to \cdots \to \phi_n \to \phi_{n+1} \equiv \phi_1 \to \phi_2 \to \cdots \to F \to \cdots \phi_n \to \phi_{n+1} \equiv (\phi_1 \to \phi_2 \to \cdots \to F) \to \cdots \phi_n \to \phi_{n+1} \equiv F \to \cdots \phi_{n+1} \equiv T$

Which leads to a contradiction since we assumed $\phi_1 \to \phi_2 \to \cdots \phi_n \to \phi_{n+1} \equiv F$.

We just observed that if $\exists i$ such that $\phi_i \equiv F$ and $i \neq n+1$ leads to a contradiction. Then i should be equal to $n + 1$ meaning that $\phi_{n+1} \equiv \psi \equiv F$ and since we picked i to be the **smallest** value, for every $j \leq n, \phi_j \equiv T$, this case is also a contradiction. Based on Soundness and Completeness Theorem, $\phi_1, \phi_2, ...\phi_n \models \psi$. This means that for "every" assignment of truth-values to $\phi_1, \phi_2, ...\phi_n$, then ψ evaluates to a truth value as well ($\psi \equiv T$).

With all these being said, the $\phi_1 \to \phi_2 \to \cdots \phi_n \to \phi_{n+1} \equiv F$ leads to a contradiction and the statement proves.

proving if $\phi_1 \to \phi_2 \to \cdots \phi_n \to \psi$ is a tautology then $\phi_1, \phi_2, ...\phi_n \vdash \psi$ has a proof:
Steps here would be same as the previous statement proof which we omit to avoid redundancy

---