Formal Methods for High-Assurance Software Engineering
HomeWork Assignment 02

Shahin Roozkhosh
shahin@bu.edu
U01670169

September 2020

Problem 1. [LCS, page 88]: Exercise 1.5.7

1.5.7 (a) $\phi_1 = (\neg p \vee \neg q) \wedge (p \vee \neg q) \wedge (\neg p \vee q)$

1.5.7 (b) $\phi_2 = (\neg p \vee \neg q \vee r) \wedge (\neg p \vee q \vee \neg r) \wedge (\neg p \vee q \vee r) \wedge (p \vee \neg q \vee r) \wedge (p \vee q \vee r)$

1.5.7 (c) $\phi_3 = (\neg r \vee \neg s \vee \neg q) \wedge (\neg r \vee s \vee \neg q) \wedge (r \vee \neg s \vee \neg q) \wedge (r \vee \neg s \vee q) \wedge (r \vee s \vee \neg q)$

Problem 2.

[Lec. Slide 07]: part 1 We begin by parenthesizing the wff, clearly the original logic is being preserved.

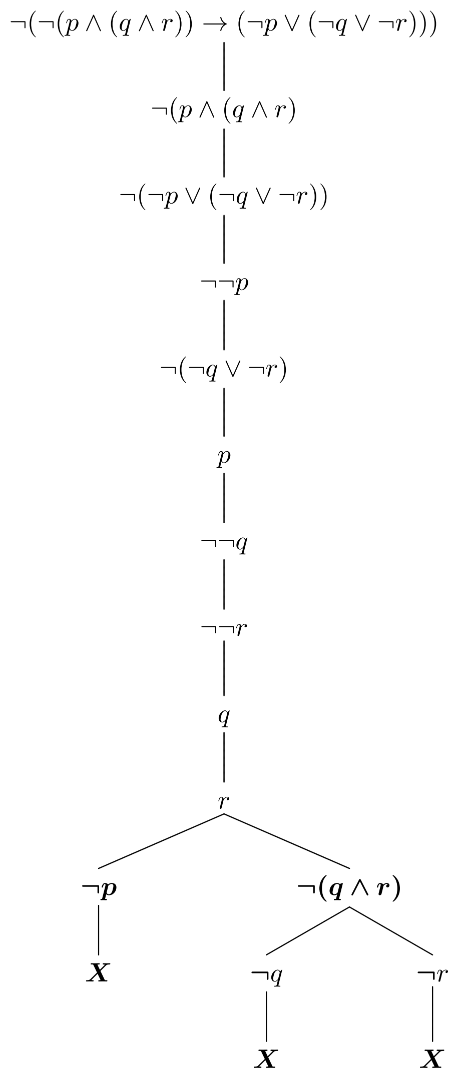
$$\text{Prove: } \neg(p \wedge (q \wedge r)) \rightarrow (\neg p \vee (\neg q \vee \neg r))$$

1	$\neg(p \wedge (q \wedge r))$	premise
2	$\neg(\neg p \vee (\neg q \vee \neg r))$	assume
3	$\neg p$	assume
4	$\neg p \vee (\neg q \vee \neg r))$	$\vee \mathrm{i}_1, 3$
5	$\perp$	$\neg \mathrm{e}, 2, 4$
6	$\neg\neg p$	$\neg \mathrm{i}, 3 - 5$
7	$\neg q \vee \neg r$	assume
8	$\neg p \vee (\neg q \vee \neg r))$	$\vee \mathrm{i}_2, 7$
9	$\perp$	$\neg \mathrm{e}, 2, 8$
10	$\neg(\neg q \vee \neg r)$	$\neg \mathrm{i}, 7 - 9$
11	$\neg q$	assume
12	$\neg q \vee (\neg r))$	$\vee \mathrm{i}_1, 11$
13	$\perp$	$\neg \mathrm{e}, 10, 12$
14	$\neg\neg q$	$\neg \mathrm{i}, 11 - 13$
15	$\neg r$	assume
16	$\neg q \vee (\neg r)$	$\vee \mathrm{i}_2, 15$
17	$\perp$	$\neg \mathrm{e}, 10, 16$
18	$\neg\neg r$	$\neg \mathrm{i}, 15 - 17$
19	q	$\neg\neg \mathrm{i}, 14$
20	r	$\neg\neg \mathrm{i}, 18$
21	$q \wedge r$	$\wedge \mathrm{i}, 19, 20$
22	p	$\neg\neg \mathrm{e}, 6$
23	$p \wedge (q \wedge r)$	$\wedge \mathrm{i}, 21, 22$
24	$\perp$	$\neg \mathrm{e}, 1, 23$
25	$\neg\neg(\neg p \vee (\neg q \vee \neg r))$	$\neg \mathrm{i}, 2 - 24$
26	$\neg p \vee (\neg q \vee \neg r)$	$\neg\neg \mathrm{e}, 25$

[Lec. Slide 07]: part 2 We begin by parenthesizing the wff, clearly the original logic is being preserved.

$$\text{Prove: } \neg(p \wedge (q \wedge r)) \rightarrow (\neg p \vee (\neg q \vee \neg r))$$

We show that the negation of the statement is unsatisfiable, hence the statement itself is tautology.



Problem 3.

Solution. Write the wff ψ_n and justify how it accomplishes the task.

By remembering the chess and Queen moves, We could easily write the requirements of N-Queen problem in English while writing it in propositional logic is was not easy.. (1) two queens cannot occupy the same row or column, we denote this requirement by ψ_n^{row}. (2) queens cannot be in adjacent rows and columns, We denote this by ψ_n^{col}. (3) a position (i,j) is preferred over position (n,m) if diag(i,j)¡diag(n,m) where diag(i,j) is defined to be the length of the longest diagonal passing through position (i,j). We formalize it by ψ_n^{diag1}. (4) Also we know that we have two types of diagonal, step (3) needs to be for the anti-diagonals as well. Denoted by ψ_n^{diag2}.

Then we can write ψ_n as:

$$\psi_n = \psi_n^{row} \wedge \psi_n^{col} \wedge \psi_n^{diag1} \wedge \psi_n^{diag2}$$

(a) ψ_n^{row} In each row there is at most one queen:

$$\psi_n n^{row1} = \bigwedge_{i=1}^{n} \bigwedge_{j=1}^{n-1} \bigwedge_{k=j+1}^{n} (x_{ij} \to \neg x_{ik})$$

Also there is a subtle point to check in each row there is at least one queen:

$$\psi_n^{row2} = \bigwedge_{i=1}^{n} \bigvee_{j=1}^{n} x_{ij}$$

$\psi_n^{row} = \psi_n^{row1} \wedge \psi_n^{row2}$

(b) ψ_n^{col}

$$\psi_n^{col} = \bigwedge_{j=1}^{n} \bigwedge_{i=1}^{n-1} \bigwedge_{k=i+1}^{n} (x_{ij} \to \neg x_{kj})$$

(c) ψ_n^{diag1}

$$\psi_n^{diag1} = \bigwedge_{i=2}^{n} \bigwedge_{j=1}^{n-1} \bigwedge_{k=1}^{\min\{i-1,n-j\}} (x_{ij} \to \neg x_{i-k,j+k})$$

(d) ψ_n^{diag2}

$$\psi_n^{diag2} = \bigwedge_{i=1}^{n-1} \bigwedge_{j=1}^{n-1} \bigwedge_{k=1}^{\min\{n-i,n-j\}} (x_{ij} \to \neg x_{i+k,j+k})$$

Problem 4.

Problem 5. This is my link to z3 code: `https://github.com/ro0zkhosh/CS511/blob/master/HW2/parity.smt2`

Problem 6. This is my link to z3py code: `https://github.com/ro0zkhosh/CS511/blob/master/HW2/parity.py`