CS 511 Formal Methods for High-Assurance Software Engineering
Homework Assignment 01

Problem 1.

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

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

3. ```
   (declare-const x Bool)
   (declare-const y Bool)
   (declare-const z Bool)
   (declare-fun phi (Bool Bool Bool) Bool)

   (declare-fun psi (Bool Bool Bool) Bool)

   (assert (=  (phi x y z)
               (or (and x y) (or (and x z) (and y z)))))

   (assert (=  (psi x y z)
               (and (or x y) (and (or x z) (or y z)))))

   ;; check (not (= (phi x y z) (psi x y z))) is unsatisfiable.
   (assert (not (= (phi x y z) (psi x y z))))
   (check-sat)
   ```

4. ```
   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(Not (phi == psi))
   # check Not (phi == psi) is unsatisfiable
   s.check()
   ```

Problem 2.

a (not the only correct one)

1	$(p \to q) \to q$	assume
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	$\bot$	$\bot$i $3, 9$
11	$\neg\neg p$	$\neg$i $3, 10$
12	p	$\neg\neg$e 11
13	$(q \to p) \to p$	$\to$i $2 - 12$
14	$((p \to q) \to q) \to ((q \to p) \to p)$	$\to$i $1 - 13$

c (not the only correct one)

1	$(p \to q) \land (q \to p)$	assume
2	$(p \lor q)$	assume
3	p	assume
4	$p \to q$	$\land$e$_1$ 1
5	q	$\to$e $1, 4$
6	$p \land q$	$\land$i $3, 5$
7	q	assume
8	$(q \to p)$	$\land$e$_2$ 1
9	p	$\to$e $1, 8$
10	$p \land q$	$\land$i $7, 9$
11	$q \land p$	$\lor$e $2, 3 - 6, 7 - 10$
12	$(p \lor q) \to (q \land p)$	$\to$i $2 - 11$
13	$((p \to q) \land (q \to p)) \to ((p \lor q) \to (q \land p))$	$\to$i $1 - 12$

d (not the only correct one)

1	$(p \to q)$	assume
2	$(\neg p \to q)$	assume
3	$(p \vee \neg p)$	LEM
4	p	assume
5	q	$\to$e $1, 4$
6	$\neg p$	assume
7	q	$\to$e $1, 6$
8	q	$\vee$e $3, 4-5, 6-7$
9	$(\neg p \to q) \to q$	$\to$i $2-8$
10	$(p \to q) \to (\neg p \to q) \to q$	$\to$i $1-9$

Problem 3. $\neg\neg$i, $\neg$i and $\neg$e are able to be rewrite by letting $\neg\phi$ be $\neg \to \bot$.

$$\frac{\phi \qquad \phi \to \bot}{\bot}\ \textbf{simulate of } \neg\textbf{e by rule} \to\textbf{e} \qquad\qquad \frac{\phi \ldots \bot}{\phi \to \bot}\ \textbf{simulate of } \neg\textbf{i by rule} \to\textbf{i}$$

simulate of ¬¬i

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

simulate of ¬¬e

1	$(\phi \to \bot) \to \bot$	premise
2	$\phi \vee (\phi \to \bot)$	LEM
3	ϕ	assume
4	$(\phi \to \bot)$	assume
5	$\bot$	$\to$e$1, 4$
6	ϕ	$\bot$e5
7	ϕ	$\vee$e$2, 3, 4-6$

Problem 4.

Proof. induction on height of ϕ, we have following cases:

case 1:
By height of ϕ is 1, we know ϕ contains only 1 atom p, i.e., $\phi = p$.
By the definition of rank, we have: $\texttt{rank}(\phi) = \texttt{rank}(p) = 1$. This case is proved.

case $1 + n$ ($n > 0$):

By the height of ϕ is $1 + n$, we know ϕ has two possible form as following:

subcase $\neg\psi$:

By definition of rank, we have: $\texttt{rank}(\phi) = 1 + \texttt{rank}(\psi)$.
By induction hypothesis, we know height of ψ is equal to $\texttt{rank}(\psi)$, i.e., $n = \texttt{rank}(\psi)$.
Then, it can be derived that $1 + n = 1 + \texttt{rank}(\psi) = \texttt{rank}(\phi)$. This case is proved.

subcase $\psi \circ \psi'$:

By definition of rank, we have: $\texttt{rank}(\phi) = 1 + \max(\texttt{rank}(\psi), \texttt{rank}(\psi'))$.
By induction hypothesis, we know height of ψ is equal to $\texttt{rank}(\psi)$, and height of ψ' is equal to $\texttt{rank}(\psi')$, i.e., $n = \max(\texttt{rank}(\psi), \texttt{rank}(\psi'))$.
Then, it can be derived that $1 + n = 1 + \max(\texttt{rank}(\psi), \texttt{rank}(\psi')) = \texttt{rank}(\phi)$. This case is proved.

$\square$

Problem 5.

(a) $\{\neg, \wedge\}$ is adequate set of connectives, since $\rightarrow$ and $\vee$ can be replaced by using the equivalence: $\phi \vee \psi \equiv \neg(\neg\phi \wedge \neg\psi)$, $\phi \rightarrow \psi \equiv \neg(\phi \wedge \neg\psi)$.
$\{\neg, \rightarrow\}$ is adequate set of connectives, since $\wedge$ and $\vee$ can be replaced by using the equivalence: $\phi \vee \psi \equiv \neg\phi \rightarrow \psi$, $\phi \wedge \psi \equiv \neg(\phi \rightarrow \neg\psi)$.
$\{\rightarrow, \perp\}$ is adequate set of connectives, since $\neg$ can be replaced by using the equivalence: $\neg\phi \equiv \phi \rightarrow \perp$, and we know $\{\rightarrow, \neg\}$ is already an adequate set.

(b) able to argue that negative value will never be created.

(c) able to argue that $\vee$ and $\wedge$ cannot be simulate or argue that there will always be even number of T or F by using only $\leftrightarrow$ and $\neg$.

Problem 6.

Proof. This is proved by two directions:

- $\Rightarrow$:
  By rule $\rightarrow$i and $\phi_1, \phi_2, \cdots, \phi_n \vdash \psi$, we know:
  $\vdash \phi_1 \rightarrow \phi_2 \rightarrow \cdots \rightarrow \phi_n \rightarrow \psi$.
  By the completeness theorem: "for any ψ, if $\vdash \psi$, then ψ is a tautology" we know:
  $\phi_1 \rightarrow \phi_2 \rightarrow \cdots \rightarrow \phi_n \rightarrow \psi$ is a tautology. This case is proved.

- $\Leftarrow$:
  By the soundness theorem, since $\phi_1 \rightarrow \phi_2 \rightarrow \cdots \rightarrow \phi_n \rightarrow \psi$ is a tautology, we know:
  $\vdash \phi_1 \rightarrow \phi_2 \rightarrow \cdots \rightarrow \phi_n \rightarrow \psi$ $(\star)$.
  By rule $\rightarrow$e and $(\star)$, we know:
  $\phi_1, \phi_2, \cdots, \phi_n \vdash \psi$. This case is proved.

$\square$