Homework 3

1.5 :

3.(a) From the equivalences

$$\phi \vee \psi \equiv \neg(\neg\phi \wedge \neg\psi) \text{ and}$$
$$\phi \to \psi \equiv \neg(\phi \wedge \neg\psi)$$

it follows that $\{\neg, \wedge\}$ is adequate. $\{\neg, \to\}$ is similar. From the equivalence

$$\neg\phi \equiv \phi \to \bot$$

it follows that $\{\to, \bot\}$ is adequate.

(b) The result follows from this fact:

Lemma 1. *Let ϕ be a wff whose only connectives are $\vee$, $\wedge$, $\to$. If all the atoms in ϕ are assigned T, then the truth value of ϕ is T.*

Proof. We use course-of-values induction on the height of the parse tree of ϕ (as we did for the proof of completeness).

Assume h is the height of ϕ's parse tree, and the lemma is true for all wffs whose parse tree has height less than h.

We have four cases, depending on the label of the root of ϕ's parse tree: it is an atom, or it is one of $\vee$, $\wedge$, or $\to$. If it is an atom, then ϕ itself is an atom, and the lemma holds immediately.

The other three cases are similar, and we will do only one—the case when the root is labeled $\vee$. Then $\phi = \psi \vee \chi$, where ψ and χ are wffs with parse trees of height less than h. By the induction assumption, when all the atoms of ψ and χ are T, the truth value of both ψ and χ is T. By the truth table for $\vee$, ϕ is also T. □

To finish this problem, assume C does not have $\neg$ or $\bot$. Then any formula that uses only connectives in C satisfies the conditions of the lemma. Therefore, it can't be a formula whose truth value is F when all its atoms are T. There are many examples of such formulas, e.g., $\neg p$, so C can't be adequate.

(c) $\{\leftrightarrow, \neg\}$ is not adequate. This follows from

Lemma 2. *Any wff that uses only the connectives $\leftrightarrow$ and $\neg$ is equivalent to a wff of the form*

$$p_1 \oplus \cdots \oplus p_k \oplus a,$$

where the atoms $p_1, \ldots, p_k$ are all distinct, a is either the constant $\bot$ or $\top$, and $\oplus$ is the exclusive or operator. ($\phi \oplus \psi$ is T if and only if exactly one of ϕ and ψ is T. When a is $\bot$, we could omit it, but we use it so we need to consider only one form.)

Proof. We use course-of-values induction on the height of the parse tree of ϕ.

Assume h is the height of ϕ's parse tree, and the lemma is true for all wffs whose parse tree has height less than h.

We have three cases, depending on the label of the root of ϕ's parse tree: it is an atom, or it is $\leftrightarrow$ or $\neg$. If it is an atom, then ϕ itself is an atom, and the lemma holds immediately.

In the case when the root is labeled $\neg$, $\phi = \neg\psi$ where ψ is a wff with parse tree of height less than h. (It's actually $h-1$.) By the induction assumption,

$$\psi \equiv p_1 \oplus \cdots \oplus p_k \oplus a$$

where $p_1, \ldots, p_k$ and a are as in the lemma. Then

$$\begin{aligned} \phi &= \neg\psi \\ &\equiv \neg(p_1 \oplus \cdots \oplus p_k \oplus a) \\ &\equiv \top \oplus p_1 \oplus \cdots \oplus p_k \oplus a \\ &\equiv p_1 \oplus \cdots \oplus p_k \oplus \top \oplus a \\ &\equiv p_1 \oplus \cdots \oplus p_k \oplus \oplus b, \end{aligned}$$

where b is $\top$ if a is $\bot$, and b is $\bot$ if a is $\top$. Thus the lemma holds when the root is labled $\neg$.

If the root is labeled $\leftrightarrow$, then $\phi = \psi \leftrightarrow \chi$, where ψ and χ are wffs with parse trees of height less than h. By the induction assumption,

$$\begin{aligned} \psi &\equiv p_1 \oplus \cdots \oplus p_k \oplus a \text{ and} \\ \chi &\equiv q_1 \oplus \cdots \oplus q_l \oplus b, \end{aligned}$$

where all the $p_1, \ldots, p_k$ are distinct, all the $q_1, \ldots, q_l$ are distinct, and a and b are the constants $\bot$ or $\top$. Then

$$\begin{aligned} \phi &= \psi \leftrightarrow \chi \\ &\equiv \psi \oplus \chi \oplus \top \\ &\equiv p_1 \oplus \cdots \oplus p_k \oplus a \oplus q_1 \oplus \cdots \oplus q_l \oplus b \oplus \top \\ &\equiv r_1 \oplus \cdots \oplus r_m \oplus c, \end{aligned}$$

where $r_1, \ldots, r_m$ is a list of all the atoms that occur exactly once in $p_1, \ldots, p_k$ or $q_1, \ldots, q_l$, and $c = a \oplus b \oplus \top$. We are using the facts that $\oplus$ is commutative, $s \oplus s \equiv \bot$ for any atom s, and $s \oplus \bot = s$. Thus the lemma holds when the root is labeled $\leftrightarrow$. This completes the proof. □

To finish the problem, we need to find some wff that is not equivalent to the form described in the lemma. An example is $p \vee q$. The only possible forms of wffs with atoms p and q that fit the form described in the lemma are $p \oplus q \oplus \bot$ and $p \oplus q \oplus \top$, and using truth tables it is easily seen that they are not equivalent to $p \vee q$.

15. **(b)** Using the algorithm for testing Horn formulas for satisfiability, the following atoms get marked: r, q, u, p, w Depending on the order of execution, other atoms may get marked. But eventually, all the atoms in the first clause, i.e., p, q, w are marked, but the right side of the $\rightarrow$ is $\perp$, so the formula is unsatisfiable.

2.1:

2. **(a)** $\forall x(F(x) \rightarrow \exists y Q(y, x))$

(b) $B(j, c) \rightarrow \neg\forall x(F(x) \rightarrow L(j, x))$

(c) $\exists x(F(x) \wedge B(c, x) \wedge B(x, j))$

5. **(h)** Let

$$\begin{aligned} E(x) &\text{ mean } x \text{ is an end user device} \\ C(x) &\text{ mean } x \text{ is a credential} \\ D(x, y) &\text{ mean } x \text{ may download } y \\ U(x, y) &\text{ mean } x \text{ may upload } y \\ M(x, y) &\text{ mean } x \text{ may manage } y \end{aligned}$$

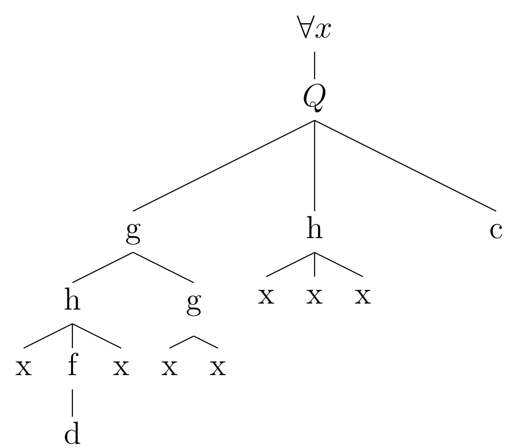
$$\forall x \forall y(E(x) \wedge E(y) \rightarrow \forall z(C(z) \rightarrow \\ (D(x, z) \leftrightarrow D(y, z)) \wedge (U(x, z) \leftrightarrow U(y, z)) \wedge (M(x, z) \leftrightarrow M(y, z))))$$

2.2:

3. **(b)** **i.** Not a formula: P has only two arguments.

ii. Not a formula: same reason, and also the function h has the predicate symbol P occurring in it.

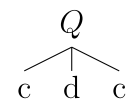
iii.



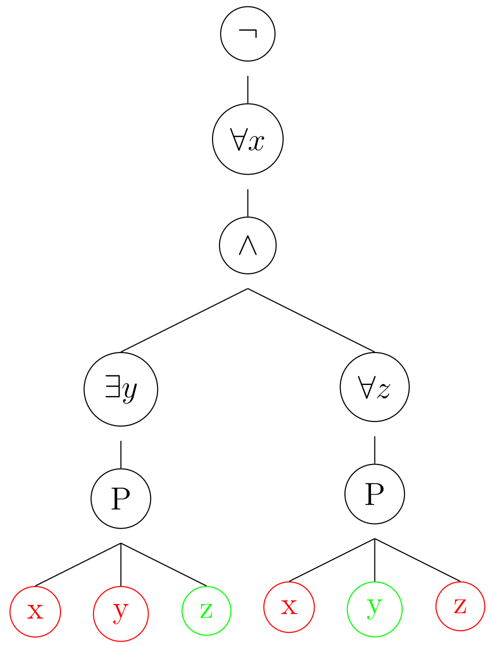
iv. Not a formula: P has only one argument.

v. Not a formula: $g(x, y) \rightarrow P(x, y, x)$ is not a formula because $g(x, y)$ is not a formula because g is not a predicate symbol.

vi.



5. **(a)**



(b) Free occurrences of variables are green; bound occurrences of variables are red.

(c) y and z occur free and bound.

(d)

$$\psi[t/x] = \neg(\forall x((\exists y P(x,y,z)) \wedge (\forall z P(x,y,z)))) \text{ (no change)}$$
$$\psi[t/y] = \neg(\forall x((\exists y P(x,y,z)) \wedge (\forall z P(x,g(f(g(y,y)),y),z))))$$
$$\psi[t/z] = \neg(\forall x((\exists y P(x,y,g(f(g(y,y)),y))) \wedge (\forall z P(x,y,z))))$$

t is free for x and y, but not z, because the y in t get bound in $\psi[t/z]$.