The University of Melbourne
School of Computing and Information Systems
COMP30026 Models of Computation

Sample Answers to Tutorial Exercises, Week 3

6.
```
contents :: BinTree a -> [a]
contents Void
  = []
contents (Node x left right)
  = contents left ++ x : contents right
```

7.
```
insertbst :: (Ord a) => a -> BinTree a -> BinTree a
insertbst z Void
  = Node z Void Void
insertbst z (Node x left right)
  | z < x     = Node x (insertbst z left) right
  | otherwise = Node x left (insertbst z right)

buildbst :: (Ord a) => [a] -> BinTree a
buildbst []
  = Void
buildbst (x : xs)
  = insertbst x (buildbst xs)
```

8. The function is well-typed: `f :: Num a => [a] -> a` and all the equations make sense. The function takes a list of numbers and returns a number, where "number" means element of some specific numeric type (Int, Integer, Float, Double). The first equation deals with an input list that is empty, the second deals with any singleton list. The third deals with the remaining case, that is, a list with more than one element. In the second equation, `x` gets bound to a number, of the third, `y` gets bound to a list (instead of `y` it would be customary to use the name `xs`, to suggest that we have a list (a plurality) of elements.

10. (a) Not equivalent:

P	Q	$\neg P$	$\Rightarrow$	Q	P	$\Rightarrow$	$\neg Q$
t	t	f	t	t	t	f	f
t	f	f	t	f	t	t	t
f	t	t	t	t	f	t	f
f	f	t	f	f	f	t	t
			X			X	

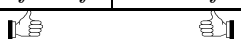
We see that the columns for the two implications are different.

(b) Not equivalent:

P	Q	$\neg P$	$\Rightarrow$	Q	Q	$\Rightarrow$	$\neg P$
t	t	f	t	t	t	f	f
t	f	f	t	f	f	t	f
f	t	t	t	t	t	t	t
f	f	t	f	f	f	t	t
			X			X	

(c) Equivalent:

P	Q	$\neg P$	$\Rightarrow$	Q	$\neg Q$	$\Rightarrow$	P
t	t	f	t	t	f	t	t
t	f	f	t	f	t	t	t
f	t	t	t	t	f	t	f
f	f	t	f	f	t	f	f



(d) and (e): (d) equivalent; but the pair in (e) are not equivalent:

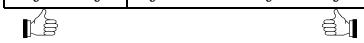
P	Q	R	P	$\Rightarrow$	$(Q$	$\Rightarrow$	$R)$	Q	$\Rightarrow$	$(P$	$\Rightarrow$	$R)$	$(P$	$\Rightarrow$	$Q)$	$\Rightarrow$	R
t	t	t	t	t	t	t	t	t	t	t	t	t	t	t	t	t	t
t	t	f	t	f	t	f	f	t	f	t	f	f	t	t	t	f	f
t	f	t	t	t	f	t	t	f	t	t	t	t	t	f	f	t	t
t	f	f	t	t	f	t	f	f	t	t	f	f	t	f	f	t	f
f	t	t	f	t	t	t	t	t	t	f	t	t	f	t	t	t	t
f	t	f	f	t	t	f	f	t	t	f	t	f	f	t	t	f	f
f	f	t	f	t	f	t	t	f	t	f	t	t	f	t	f	t	t
t	f	f	f	t	f	t	f	f	t	f	t	f	f	t	f	f	f

X X

Note that the formulas in (d) are both equivalent to $(P \wedge Q) \Rightarrow R$, which explains why the order of P and Q does not matter here.

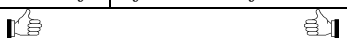
(f) Equivalent:

P	Q	$(P$	$\Rightarrow$	$Q)$	$\Rightarrow$	P
t	t	t	t	t	t	t
t	f	t	f	f	t	t
f	t	f	t	t	f	f
f	f	f	t	f	f	f



(g) Equivalent:

P	Q	R	$(P$	$\vee$	$Q)$	$\Rightarrow$	R	$(P$	$\Rightarrow$	$R)$	$\wedge$	$(Q$	$\Rightarrow$	$R)$
t	t	t	t	t	t	t	t	t	t	t	t	t	t	t
t	t	f	t	t	t	f	f	t	f	f	f	t	f	f
t	f	t	t	t	f	t	t	t	t	t	t	f	t	t
t	f	f	t	t	f	f	f	t	f	f	f	f	t	f
f	t	t	f	t	t	t	t	f	t	t	t	t	t	t
f	t	f	f	t	t	f	f	f	t	f	f	t	f	f
f	f	t	f	f	f	t	t	f	t	t	t	f	t	t
t	f	f	f	f	f	t	f	f	t	f	t	f	t	f



11. The formula is equivalent to $P \Rightarrow Q$. This is easily checked with a truth table, but how can we simplify $P \Leftrightarrow (P \wedge Q)$ when we don't know what it is supposed to be equivalent to? Well, we can just try. Let us expand the biimplication and obtain $(P \Rightarrow (P \wedge Q)) \wedge ((P \wedge Q) \Rightarrow P)$. Intuitively, the conjunct on the right is just true, and we can check that with a truth table. So we have found that the original formula is equivalent to $P \Rightarrow (P \wedge Q)$, which isn't any shorter, but still. We can rewrite the result as $(P \Rightarrow P) \wedge (P \Rightarrow Q)$, and now it becomes clear that all we need is $P \Rightarrow Q$.

12.

P	Q	$(P$	$\oplus$	$Q)$	$\oplus$	Q
t	t	t	f	t	t	t
t	f	t	t	f	t	f
f	t	f	t	t	f	t
f	f	f	f	f	f	f

We see that the columns for P and for $(P \oplus Q) \oplus Q$ are identical.

13.

P	Q	P	$\Leftrightarrow$	Q	$\neg$	P	$\Leftrightarrow$	$\neg$	Q
t	t	t	t	t	f	t	t	f	t
t	f	t	f	f	f	t	f	t	f
f	t	f	f	t	t	f	f	f	t
f	f	f	t	f	t	f	t	t	f

Clearly the columns for $P \Leftrightarrow Q$ and for $\neg P \Leftrightarrow \neg Q$ are identical.

14. Even with three variables the truth table is manageable, so let us construct it.

			(1A)					(1B)					(2)		
P	Q	R	P	$\Leftrightarrow$	$(Q$	$\Leftrightarrow$	$R)$	$(P$	$\Leftrightarrow$	$Q)$	$\Leftrightarrow$	R	$P \wedge Q \wedge R$	$\vee$	$\neg P \wedge \neg Q \wedge \neg R$
t	t	t	t	t	t	t	t	t	t	t	t	t	t	t	f
t	t	f	t	f	t	f	f	t	t	t	f	f	f	f	f
t	f	t	t	f	f	f	t	t	f	f	f	t	f	f	f
t	f	f	t	t	f	t	f	t	f	f	t	f	f	f	f
f	t	t	f	f	t	t	t	f	f	t	f	t	f	f	f
f	t	f	f	t	t	f	f	f	f	t	t	f	f	f	f
f	f	t	f	t	f	f	t	f	t	f	t	t	f	f	f
f	f	f	f	f	f	t	f	f	t	f	f	f	f	t	t

B is a logical consequence of A (we write $A \models B$) iff B is true for any assignment of variables which makes A true. So we see that (1) $\not\models$ (2), because there are cases like tff that make (1) true but (2) false. Similarly, (2) $\not\models$ (1), because the case fff makes (2) true but (1) false.