THE UNIVERSITY OF MELBOURNE
SCHOOL OF COMPUTING AND INFORMATION SYSTEMS
COMP30026 MODELS OF COMPUTATION

Sample Answers to Tutorial Exercises, Week 4

30. Let the propositional variable A stand for "A is a knight" and similarly for B and C. Note that if A makes statement S then we know that $A \Leftrightarrow S$ holds. Or we can consider the two possible cases for A separately: Either A is a knight, and A's statement can be taken face value; or A is a knave, in which case the negation of the statement holds, that is,

$$\Big(A \wedge (A \Rightarrow (\neg B \wedge \neg C))\Big) \vee \Big(\neg A \wedge \neg(A \Rightarrow (\neg B \wedge \neg C))\Big)$$

We can rewrite the implications:

$$\Big(A \wedge (\neg A \vee (\neg B \wedge \neg C))\Big) \vee \Big(\neg A \wedge \neg(\neg A \vee (\neg B \wedge \neg C))\Big)$$

Then, pushing negation in and using the distributive laws, we get

$$\Big(A \wedge (\neg B \wedge \neg C)\Big) \vee \Big(\neg A \wedge A \wedge (B \vee C)\Big)$$

The second disjunct is false, so the formula is equivalent to $A \wedge \neg B \wedge \neg C$. So A must be a knight, and B and C are knaves.

31. Let us call the initial contents of R_1 and R_2 x and y, respectively. We want to see what happens to the individual bits in x and y, but since the $\oplus$ works bitwise, we can just consider x and y in their entirety.

    - After the first assignment, R_1 holds $x \oplus y$, and R_2 holds y.
    - So after the second assignment, R_1 holds $x \oplus y$, and R_2 holds $x \oplus y \oplus y$, that is, x.
    - So after the third assignment, R_2 holds x, and R_1 holds $x \oplus y \oplus x$, that is, y.

32. The formulas F_2, F_3, F_5, and F_6 are logically equivalent. Grouping the formulas into sets of equivalent formulas, we get

$$\{\{F_1\}, \{F_4\}, \{F_2, F_3, F_5, F_6\}\}$$

This can easily be verified by completing the truth tables.

33. These are the clauses generated:

    (a) For each node i generate the clause $B_i \vee G_i \vee R_i$. That's $n+1$ clauses of size 3 each.

    (b) For each node i generate three clauses: $(\neg B_i \vee \neg G_i) \wedge (\neg B_i \vee \neg R_i) \wedge (\neg G_i \vee \neg R_i)$. That comes to $3n+3$ clauses of size 2 each.

    (c) For each pair (i, j) of nodes with $i < j$ we want to express $E_{ij} \Rightarrow (\neg(B_i \wedge B_j) \wedge \neg(G_i \wedge G_j) \wedge \neg(R_i \wedge R_j)$. This means for each pair (i, j) we generate three clauses: $(\neg E_{ij} \vee \neg B_i \vee \neg B_j) \wedge (\neg E_{ij} \vee \neg G_i \vee \neg G_j) \wedge (\neg E_{ij} \vee \neg R_i \vee \neg R_j)$. There are $n(n+1)/2$ pairs, so we generate $3n(n+1)/2$ clauses, each of size 3.

Altogether we generate $3n + 3 + 6n + 6 + 9n(n+1)/2$ literals, that is, $9(n+1)(n/2+1)$.

34. (a) $\neg P$ becomes $\boxed{P \oplus \mathbf{t}}$. With XNF it is perhaps more natural to use 0 for $\mathbf{f}$ and 1 for $\mathbf{t}$. We really are dealing with arithmetic modulo 2, $\oplus$ playing the role of addition, and $\wedge$ playing the role of multiplication.

(b) $P \wedge Q$ is unchanged, or we can write simply $\boxed{PQ}$.

(c) $P \wedge \neg Q$ can be written $P(Q \oplus \mathbf{t})$, so by "multiplying out" we get $\boxed{PQ \oplus P}$.

(d) $P \Leftrightarrow Q$ becomes $\boxed{P \oplus Q \oplus \mathbf{t}}$ (since biimplication is the negation of exclusive or).

(e) $P \vee Q$ becomes $\boxed{P \oplus Q \oplus PQ}$, as truth tables will confirm. But how could we *discover* that solution? Well, we now know how to deal with negation and disjunction, and so we can make use of the fact that $P \vee Q \equiv \neg(\neg P \wedge \neg Q)$. This way we arrive at $\mathbf{t} \oplus ((\mathbf{t} \oplus P)(\mathbf{t} \oplus Q))$. Now all we need to do is to simplify that formula (come on, do it).

(f) Using the insight from part (e), we transform $P \vee (Q \wedge R)$ to $\boxed{P \oplus QR \oplus PQR}$.

(g) Negation is just '$\oplus\, \mathbf{t}$', so we can write $\neg(P \oplus Q)$ as $\boxed{P \oplus Q \oplus \mathbf{t}}$.

(h) For $(P \oplus Q) \wedge R$, we just "multiply out", to get $\boxed{PR \oplus QR}$.

(i) Given $(PQ \oplus PQR \oplus R) \wedge (P \oplus Q)$ we again multiply out. There will be six products, but some will cancel out:

$$
\begin{aligned}
& PQP \oplus PQRP \oplus PR \oplus PQQ \oplus PQRQ \oplus QR \\
= \; & PQ \oplus PQR \oplus PR \oplus PQ \oplus PQR \oplus QR \\
= \; & \boxed{PR \oplus QR}
\end{aligned}
$$

(j) Given $Q \wedge (P \oplus PQ \oplus \mathbf{t})$ we multiply out, to get $PQ \oplus PQ \oplus Q$, that is, $\boxed{Q}$.

(k) From part (e) we know that $A \vee B \equiv A \oplus B \oplus AB$. Applying that to $Q \vee (P \oplus PQ)$ we obtain the formula $Q \oplus P \oplus PQ \oplus (Q(P \oplus PQ))$. Multiplying out, we get

$$Q \oplus P \oplus PQ \oplus PQ \oplus PQ = \boxed{P \oplus Q \oplus PQ}$$

(So the formula in (k) must be equivalent to $P \vee Q$.)