Assignment 2

Name: Zhijun Wei
Student Number: 21027550
Email: z29wei@uwaterloo.ca

Question 1:

(a) There are 4 execution paths:

Path 1: 1→2→3→4→9→11→12→13→17

Path 2: 1→2→3→4→9→11→14→15→16→17

Path 3: 1→2→5→6→7→9→11→12→13→17

Path 4: 1→2→5→6→7→9→11→14→15→16→17

(b)

Path 1:

Edge	Symbolic State (PV)	Path Condition (PC)
1→2	$x \mapsto X_0, y \mapsto Y_0$	true
2→3	$x \mapsto X_0, y \mapsto Y_0$	$X_0 + Y_0 > 15$
3→4	$x \mapsto X_0 + 7, y \mapsto Y_0$	$X_0 + Y_0 > 15$
4→9	$x \mapsto X_0 + 7, y \mapsto Y_0 - 12$	$X_0 + Y_0 > 15$
9→11	$x \mapsto X_0 + 9, y \mapsto Y_0 - 12$	$X_0 + Y_0 > 15$
11→12	$x \mapsto X_0 + 9, y \mapsto Y_0 - 12$	$X_0 + Y_0 > 15 \land 2 * (X_0 + Y_0) > 21$
12→13	$x \mapsto 3 * (X_0 + 9), y \mapsto Y_0 - 12$	$X_0 + Y_0 > 15 \land 2 * (X_0 + Y_0) > 21$
13→17	$x \mapsto 3 * (X_0 + 9), y \mapsto 2 * (Y_0 - 12)$	$X_0 + Y_0 > 15 \land 2 * (X_0 + Y_0) > 21$

Path 2:

Edge	Symbolic State (PV)	Path Condition (PC)
1→2	$x \mapsto X_0, y \mapsto Y_0$	true
2→3	$x \mapsto X_0, y \mapsto Y_0$	$X_0 + Y_0 > 15$
3→4	$x \mapsto X_0 + 7, y \mapsto Y_0$	$X_0 + Y_0 > 15$

4→9	$x \mapsto X_0 + 7, y \mapsto Y_0 - 12$	$X_0 + Y_0 > 15$
9→11	$x \mapsto X_0 + 9, y \mapsto Y_0 - 12$	$X_0 + Y_0 > 15$
11→14	$x \mapsto X_0 + 9, y \mapsto Y_0 - 12$	$X_0 + Y_0 > 15 \wedge 2 * (X_0 + Y_0) \leq 21$
14→15	$x \mapsto X_0 + 9, y \mapsto Y_0 - 12$	$X_0 + Y_0 > 15 \wedge 2 * (X_0 + Y_0) \leq 21$
15→16	$x \mapsto 4 * (X_0 + 9), y \mapsto Y_0 - 12$	$X_0 + Y_0 > 15 \wedge 2 * (X_0 + Y_0) \leq 21$
16→17	$x \mapsto 4 * (X_0 + 9), y \mapsto 3 * Y_0 + 4 * X_0$	$X_0 + Y_0 > 15 \wedge 2 * (X_0 + Y_0) \leq 21$

Path 3:

Edge	Symbolic State (PV)	Path Condition (PC)
1→2	$x \mapsto X_0, y \mapsto Y_0$	true
2→5	$x \mapsto X_0, y \mapsto Y_0$	$X_0 + Y_0 \leq 15$
5→6	$x \mapsto X_0, y \mapsto Y_0$	$X_0 + Y_0 \leq 15$
6→7	$x \mapsto X_0, y \mapsto Y_0 + 10$	$X_0 + Y_0 \leq 15$
7→9	$x \mapsto X_0 - 2, y \mapsto Y_0 + 10$	$X_0 + Y_0 \leq 15$
9→11	$x \mapsto X_0, y \mapsto Y_0 + 10$	$X_0 + Y_0 \leq 15$
11→12	$x \mapsto X_0, y \mapsto Y_0 + 10$	$X_0 + Y_0 \leq 15 \wedge 2 * (X_0 + Y_0) > 21$
12→13	$x \mapsto 3 * X_0, y \mapsto Y_0 + 10$	$X_0 + Y_0 \leq 15 \wedge 2 * (X_0 + Y_0) > 21$
13→17	$x \mapsto 3 * X_0, y \mapsto 2 * (Y_0 + 10)$	$X_0 + Y_0 \leq 15 \wedge 2 * (X_0 + Y_0) > 21$

Path 4:

Edge	Symbolic State (PV)	Path Condition (PC)
1→2	$x \mapsto X_0, y \mapsto Y_0$	true
2→5	$x \mapsto X_0, y \mapsto Y_0$	$X_0 + Y_0 \leq 15$
5→6	$x \mapsto X_0, y \mapsto Y_0$	$X_0 + Y_0 \leq 15$

6→7	$x \mapsto X_0$, $y \mapsto Y_0 + 10$	$X_0 + Y_0 \le 15$
7→9	$x \mapsto X_0 - 2$, $y \mapsto Y_0 + 10$	$X_0 + Y_0 \le 15$
9→11	$x \mapsto X_0$, $y \mapsto Y_0 + 10$	$X_0 + Y_0 \le 15$
11→14	$x \mapsto X_0$, $y \mapsto Y_0 + 10$	$X_0 + Y_0 \le 15 \land 2 * (X_0 + Y_0) \le 21$
14→15	$x \mapsto X_0$, $y \mapsto Y_0 + 10$	$X_0 + Y_0 \le 15 \land 2 * (X_0 + Y_0) \le 21$
15→16	$x \mapsto 4 * X_0$, $y \mapsto Y_0 + 10$	$X_0 + Y_0 \le 15 \land 2 * (X_0 + Y_0) \le 21$
16→17	$x \mapsto 4 * X_0$, $y \mapsto 3 * Y_0 + 4 * X_0 + 30$	$X_0 + Y_0 \le 15 \land 2 * (X_0 + Y_0) \le 21$

(c)
Path1: Feasible, $X_0 = 10$, $Y_0 = 10$
Path2: Infeasible, because it must satisfy $X_0 + Y_0 > 15 \land 2 * (X_0 + Y_0) \le 21$, which means $X_0 + Y_0$ should greater than 15 and less than or equal to 11.5, which is not possible.
Path3: Feasible, $X_0 = 6$, $Y_0 = 6$
Path4: Feasible, $X_0 = 1$, $Y_0 = 1$

Question 2:
(a) at-most-one is satisfied if at most one of the boolean variables is true, which means we should deny the possibility of any two variables being true simultaneously. So for varibles a_1, a_2, a_3 and a_4, they should satisfy the CNF below:
$(\neg a_1 \lor \neg a_2) \land (\neg a_1 \lor \neg a_3) \land (\neg a_1 \lor \neg a_4) \land (\neg a_2 \lor \neg a_3) \land (\neg a_2 \lor \neg a_4) \land (\neg a_3 \lor \neg a_4)$

(b) Valid. Proof:

$(\forall x \cdot \exists y \cdot P(x) \lor Q(y))$

$\Leftrightarrow \forall x \cdot \exists y \cdot (P(x) \lor Q(y))$

$\Leftrightarrow \forall x \cdot (\exists y \cdot P(x)) \lor (\exists y \cdot Q(y))$

$\Leftrightarrow \forall x \cdot (P(x)) \lor (\exists y \cdot Q(y))$ Because $\exists y \cdot P(x) \Leftrightarrow P(x)$

$\Leftrightarrow (\forall x \cdot P(x)) \lor (\forall x \cdot \exists y \cdot Q(y))$

$\Leftrightarrow (\forall x \cdot P(x)) \lor (\exists y \cdot Q(y))$ Because $\forall x \cdot \exists y \cdot Q(y) \Leftrightarrow \exists y \cdot Q(y)$

(c) Invalid. The model in which the sentence is false:
$S = \{a, b\}$

$P^M = \{(a, a)\}$
$Q^M = \{(b, b)\}$
For the left side, if we pick x = a, there exists a y = a such that $P(x,y) \vee Q(x,y) = 1 \vee 0 = 1$
if we pick x = b, there exists a y = b such that $P(x,y) \vee Q(x,y) = 0 \vee 1 = 1$

Therefore, $M \models \forall x \cdot \ \exists y \cdot \ P(x,y) \vee Q(x,y)$

However, for the right side, if we pick x = a, $\exists y \cdot \ Q(x,y)$ can not be true, and if we pick x = b,

$\exists y \cdot \ P(x,y)$ can not be true.

Therefore, $M \not\models (\forall x \cdot \ \exists y \cdot \ P(x,y)) \vee (\forall x \cdot \ \exists y \cdot \ Q(x,y))$

Hence the sentence is invalid.

(d.a) Satisfy. In this model, we need to find x, y, z such that
$x < y$ (P(x, y))
$z < y$ (P(z, y))
$x < z$ (P(x, z))
$z \geq x$ (¬P(x, z))
Consider x = 1, y = 3, z = 2. Here $1 < 3$, $2 < 3$, $1 < 2$, and $2 \geq 1$.
Therefore, M1 satisfies this formula.

(d.b) Violate. In this model, we need to find x, y, z such that
$x + 1 = y$ (P(x, y))
$z + 1 = y$ (P(z, y))
$x + 1 = z$ (P(x, z))
$z + 1 \neq x$ (¬P(x, z))
If $x + 1 = z$, and $z + 1 = y$, that means $x + 2 = y$, which contradicts to $x + 1 = y$.
Therefore, M2 violates this formula.

(d.c) Satisfy. In this model, we need to find x, y, z such that

$x \subseteq y$ (P(x, y))

$z \subseteq y$ (P(z, y))

$x \subseteq z$ (P(x, z))

$z \nsubseteq x$ (¬P(x, z))

Consider $x = \varnothing$, $y = \{1, 2\}$, $z = \{1\}$. Here $\varnothing \subseteq \{1, 2\}$, $\{1\} \subseteq \{1, 2\}$, $\varnothing \subseteq \{1\}$, $\{1\} \nsubseteq \varnothing$.

Therefore, M3 satisfies this formula.

(e) We use the sequential counter encoding to achieve a linear number of clauses:

1. Introduce auxiliary variables to represent intermediate states of the count. For example, for n variables $x_1, x_2, ..., x_n$, we introduce n−1 auxiliary variables $y_1, y_2, ..., y_{n-1}$.
2. For the first variable x_1, we add a clause to indicate that if x_1 is true, then y_1 must be true.
3. For each subsequent variable x_i (where $2 \leq i \leq n$), we add clauses to encode the following:
If x_i is true, then y_{i-1} must be true (indicating a previous true variable).
If x_i is true, then y_i must be true (carrying the count forward).
If y_{i-1} is true, then x_i must be false (enforcing the at-most-one constraint).
4. Finally, add a clause to ensure that y_{n-1} is false, which effectively ensures that no more than one of the x_i can be true.
This method introduces only O(n) clauses and auxiliary variables.

Question 3:
(a)
1. Unique values constraint:
For each pair of cells (a, b) and (c,d) in the magic square grid, we assert that the values are different:

$$\bigwedge_{1\leq a,c\leq n} \bigwedge_{1\leq b,d\leq n} (a \neq c \lor b \neq d) \rightarrow (f_{a,b} \neq f_{c,d})$$

where $f_{a,b}$ denotes the value in the cell at row a and column b.
2. Sum Constraint for Rows and Columns:
The sum of the values in each row and column must be equal to the constant $n*(n^2+1)/2$

$$\bigwedge_{1\leq i \leq n} (\text{sum}(\text{row}_i) = n*(n^2+1)/2 \land (\text{sum}(\text{col}_i) = n*(n^2+1)/2$$

where row_i and col_i are sequences of cells in the i-th row and column, respectively.
3. Sum Constraint for Diagonals:
Similarly, the sum of the values in each diagonal must also be equal to n*(n2+1)/2:

$$\text{sum}(\text{diag}_1) = n*(n^2+1)/2 \land \text{sum}(\text{diag}_2) = n*(n^2+1)/2$$

where diag_1 and diag_2 are the main diagonal and the secondary diagonal, respectively.
4. Range Constraint: Each cell must contain a number between 1 and n^2:

$$\bigwedge_{1\leq a\leq n} \bigwedge_{1\leq b\leq n} (1 \leq f_{a,b} \leq n^2)$$

Question 4:
(e) A program on which the symbolic execution engine diverges:

```
havoc a,b,c,d;
while a>0 do a:=a-1;
while b>0 do b:=b-1;
while c>0 do c:=c-1;
if d<10 then a:=1
```