ECE 653 Assignment 2

Name: Saride Chandra Mouli

Student ID : 20810347

Email : cmsaride@uwaterloo.ca

1(a):

Given program has 4 execution paths. They are as follows:

Path 1: 1,2,3,4,9,11,12,13,17

Path 2: 1,2,3,4,9,11,15,16,17

Path 3: 1,2,6,7,9,11,15,16,17

Path 4: 1,2,6,7,9,11,12,13,17

1(b):

Path 1 : 1,2,3,4,9,11,12,13,17

Edge	**Symbolic State**	**Path Condition (PC)**
1->2	x-> X_0, y->Y_0	true
2->3	x->X_0, y->Y_0	$X_0+Y_0>10$
3->4	x->X_0+1, y->Y_0	
4->9	x->X_0+1, y->Y_0-2	
9->11	x->X_0+3, y->Y_0-2	
11->12	x->X_0+3, y->Y_0-2	$X_0+Y_0>10 \wedge 2*(X_0+Y_0+1)>27$
12->13	x->3*(X_0+3), y->Y_0-2	
13->17	x->3*(X_0+3), y->2*(Y_0-2)	

Path condition for path 1 is : $X_0+Y_0>10 \wedge 2*(X_0+Y_0+1)>27$

Path 2 : 1,2,3,4,9,11,15,16,17

Edge	**Symbolic State**	**Path Condition (PC)**
1->2	x-> X_0, y->Y_0	true
2->3	x->X_0, y->Y_0	$X_0+Y_0>10$
3->4	x->X_0+1, y->Y_0	
4->9	x->X_0+1, y->Y_0-2	
9->11	x->X_0+3, y->Y_0-2	
11->15	x->X_0+3, y->Y_0-2	$X_0+Y_0>10$ ∧ $2*(X_0+Y_0+1)\leq 27$
15->16	x->4*(X_0+3), y->Y_0-2	
16->17	x->4*(X_0+3), y->3*(Y_0-2)+ 4*(X_0+3)	

Path condition for path 2 is : $X_0+Y_0>10 \wedge 2*(X_0+Y_0+1)\leq 27$

Path 3: 1,2,6,7,9,11,15,16,17

Edge	Symbolic State	Path Condition (PC)
1->2	x-> X_0, y->Y_0	true
2->6	x->X_0, y->Y_0	$X_0+Y_0 \leq 10$
6->7	x->X_0, y->Y_0+7	
7->9	x->X_0-3, y->Y_0+7	
9->11	x->X_0-1, y->Y_0+7	
11->15	x->X_0-1, y->Y_0+7	$X_0+Y_0 \leq 10$ ∧ $2*(X_0+Y_0+6) \leq 27$
15->16	x->4*(X_0-1), y->Y_0+7	
16->17	x->4*(X_0-1), y->3*(Y_0+7)+ 4*(X_0-1)	

Path condition for path 3 is $X_0+Y_0 \leq 10 \ \wedge \ 2*(X_0+Y_0+6) \leq 27$

Path 4: 1,2,6,7,9,11,12,13,17

<table>
<tr><th>Edge</th><th>Symbolic State</th><th>Path Condition (PC)</th></tr>
<tr><td>1->2</td><td>x-> X_0, y->Y_0</td><td>true</td></tr>
<tr><td>2->6</td><td>x->X_0, y->Y_0</td><td rowspan="4">$X_0+Y_0 \leq 10$</td></tr>
<tr><td>6->7</td><td>x->X_0, y->Y_0+7</td></tr>
<tr><td>7->9</td><td>x->X_0-3, y->Y_0+7</td></tr>
<tr><td>9->11</td><td>x->X_0-1, y->Y_0+7</td></tr>
<tr><td>11->12</td><td>x->X_0-1, y->Y_0+7</td><td rowspan="3">$X_0+Y_0 \leq 10$ ∧
$2*(X_0+Y_0+6)>27$</td></tr>
<tr><td>12->13</td><td>x->3*(X_0-1), y->Y_0+7</td></tr>
<tr><td>13->17</td><td>x->3*(X_0-1),
y->2*(Y_0+7)</td></tr>
</table>

Path condition for path 4 is $X_0+Y_0 \leq 10 \land 2*(X_0+Y_0+6)>27$

1(c):

Path 1: It is a feasible path.

Path condition is : $X_0+Y_0>10 \wedge 2*(X_0+Y_0+1)>27$

$X_0=10$, $Y_0=10$ will satisfy the path condition

Path 2: It is a feasible path.

Path condition is : $X_0+Y_0>10 \wedge 2*(X_0+Y_0+1)\leq 27$

$X_0=6$, $Y_0=5$ will satisfy the path condition

Path 3: It is a feasible path.

Path condition is : $X_0+Y_0\leq 10 \wedge 2*(X_0+Y_0+6)\leq 27$

$X_0=1$, $Y_0=1$ will satisfy the path condition

Path 4: It is a feasible path.

Path condition is : $X_0+Y_0\leq 10 \wedge 2*(X_0+Y_0+6)>27$

$X_0=4$, $Y_0=4$ will satisfy the path condition

2(a):

Given constraint at most one (a_1, a_2, a_3, a_4)

Can be encoded as follows :

$$(\neg a1 \vee \neg a2) \wedge (\neg a1 \vee \neg a3) \wedge (\neg a1 \vee \neg a4) \wedge (\neg a2 \vee \neg a3) \wedge (\neg a2 \vee \neg a4) \wedge (\neg a3 \vee \neg a4)$$

2(b):

We are considering the following assumptions and variables for evaluating the given graph reachability decision problem

- v_{int}(starting vertex) is reachable
- Any Vertex v_t is reachable if and only if it's a starting vertex or there exists another vertex v_k so the $P_{vk,vt}$ is true for path existing between v_k,v_t and v_k is reachable.

$$R_t \rightarrow \bigvee_{(v_k,v_t)\in E} (P_{v_k,v_t} \wedge R_k)) for\ each\ vertex\ v_t \in V \setminus \{v_{init}\}$$

- Rechability of any vertex v_t is represented as R_t

Constraint is as follows:

Reachable(G,V,v_{int},v_{end})=

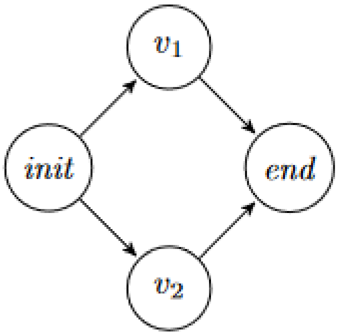
$$R_{init} \wedge (\neg R_t \bigvee_{(v_k,v_t)\in E} (P_{v_k,v_t} \wedge R_k)) \wedge R_{end}$$

Here R_{init} is always true as we considered v_{init} always reachable

Here we are asserting R_{end} to be true

The rest of logic goes in the middle term.

Consider graph 1 from given question which should evaluate to true

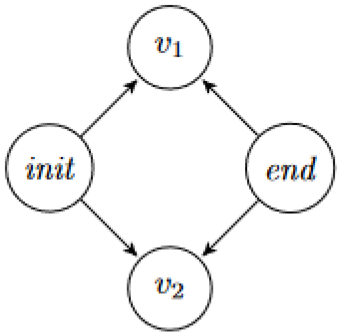


From given constraints R_{init} will be always true.

See edges (v_{int},v_1) and (v_1,v_{end})-> here path exists between (v_{int},v_1) and v_{int} is reachable making v_1 reachable, here path exists between (v_1,v_{end}) and v_1 is reachable making v_{end} reachable , thus proving path exists between v_{int} and v_{end} , making the following term true, Rend is true as path exists.

$$\neg R_t \bigvee_{(v_k, v_t) \in E} (P_{v_k, v_t} \wedge R_k)$$

Consider graph 2 from given question which should evaluate to false



From given constraints R_{init} will be always true.

See edges (v_{int},v_1) and (v_1,v_{end})->)-> here path exists between (v_{int},v_1) and v_{int} is reachable making v_1 reachable, here path doesn't exists between (v_1,v_{end}) ,making v_{end} un-reachable.

See edge (v_2,v_{end}) -> here path doesn't exists between (v_2,v_{end}) ,making v_{end} un-reachable

Thus resulting the evaluation to false, as there exists no path between v_{int} and v_{end} making the following term false,

$$\neg R_t \quad \bigvee_{(v_k, v_t) \in E} \quad (P_{v_k, v_t} \wedge R_k)$$

<u>Reference for this solution:</u>

SAT Modulo Graphs: Acyclicity(Section 3 Examples- Example 2)

Link: https://link.springer.com/chapter/10.1007/978-3-319-11558-0_10

2(c):

For extending encoding of part(a) to n variables , atmost O(n) clauses we use commander encoding.

At Most One commander encoding works by repetitively partitioning 'n' variables into groups of fixed size and then encode at-most-one to each group with total clauses nearly 3n. Partitioning variables into size 's' for each group. Consider groups as G1,G2,….,Gg, and corresponding commander variables for each group is c1,c2,…cg

- At most one in group is true, encoded as here x_j,x_k are variables of group we are checking

$$\bigwedge_{x_j \in G_i} \bigwedge_{x_k \in G_i, k<j} (\neg x_j \vee \neg x_k)$$

- If Commander variable is false, then none of variables of group can be true, encoded as

$$\bigwedge_{x_j \in G_i} (c_i \vee \neg x_j)$$

- Exactly one commander variables should be true , we encode as following

$$(c_1 \vee c_2 \vee \ldots \vee c_m) \wedge \bigwedge_{i<m} \bigwedge_{j<i} (\neg c_i \vee \neg c_j)$$

Reference: Efficient CNF Encoding for Selecting 1 from N Objects(Section:2, Page:2,3)

Link: https://www.cs.cmu.edu/~wklieber/papers/2007_efficient-cnf-encoding-for-selecting-1.pdf

3(a):

Quantifier free constraints in first order logic for magic square given for any positive interger n , nxn is the square

- Elements in square are in range of 1<=X<=n^2

$$\bigwedge_{i,j \in (o,n-1)} x_{i,j} \geq 1 \wedge x_{i,j} \leq n^2$$

- Elements of square are distinct from each other

$$\bigwedge_{0 \leq a < i \leq n-1,\ 0 \leq b < j \leq n-1} x_{i,j} \neq x_{a,b}$$

- Sum of all elements of each row is same

$$\bigwedge_{0 \leq a < i \leq n-1} \sum_{j=0}^{n-1} x_{i,j} = \sum_{j=0}^{n-1} x_{a,j}$$

- Sum of all elements of each column is same

$$\bigwedge_{0 \leq a < j \leq n-1} \sum_{i=0}^{n-1} x_{i,j} = \sum_{i=0}^{n-1} x_{i,a}$$

- Sum of all elements on each diagonal are equal

$$\sum_{i=0}^{n-1} x_{i,i} = \sum_{i=0}^{n-1} x_{i,n-i-1}$$

- Sum of row, column, diagonal are equal(Since we are already checking that sum of all rows are equal, all columns are equal, all diagonals are equal, to avoid redundancy just checking sum of one row, one column, one diagonal are equal or not)

$$\sum_{j=0}^{n-1} x_{1,j} = \sum_{i=0}^{n-1} x_{i,1} = \sum_{i=0}^{n-1} x_{i,i}$$

All the given constraints should be true for existence of a magic square with requirements as given in the question.

5(a):

$$(\forall x \cdot \exists y \cdot P(x) \vee Q(y)) \iff (\forall x \cdot P(x)) \vee (\exists y \cdot Q(y))$$

The following FOL is valid . below is the proof for it

From the given FOL by hypothesis- (1) $\forall_x.\exists_y.$ P(x) V Q(y)

By Existential Instantiation of 1 - (2) $\forall_x$.P(x) V Q(a)

By Universal Instantiation of 2 - (3) P(b) V Q(a)

By Simplification of 3 - (4) Q(a)

By Existential Generalisation of 4- (5) $\exists_y$.Q(y)

By Simplification of 3 - (6) P(b)

By Universal Generalisation of 6- (7) $\forall_x$.P(x)

By Disjunction of 5,7- (8) $\forall_x$.P(x) V $\exists_y$.Q(y)

Since it's a Bi-Implication we need to prove for vice versa also

From the given FOL by hypothesis- (1) $\forall_x$.P(x) V $\exists_y$.Q(y)

By Existential Instantiation of 1 - (2) $\forall_x$.P(x) V Q(a)

By Universal Instantiation of 2 - (3) P(b) V Q(a)

By Existential Generalisation on 'a' of 3- (4) $\exists_y$.(P(b) V Q(y))

By Universal Generalisation on 'b'of 4- (5) $\forall_x.\exists_y$.(P(x) V Q(y))

5(b):

$$(\forall x \cdot \exists y \cdot P(x,y) \vee Q(x,y)) \implies (\forall x \cdot \exists y \cdot P(x,y)) \vee (\forall x \cdot \exists y \cdot Q(x,y))$$

The following FOL is valid, below is the proof for it

From the given FOL by hypothesis- (1) $\forall_x.\exists_y.$ P(x,y) V Q(x,y)

By Existential Instantiation of 1 - (2) $\forall_x$.P(x,a) V Q(x,a)

By Universal Instantiation of 2 - (3) P(b,a) V Q(b,a)

By Simplification of 3 - (4) P(b,a)

By Existential Generalisation on 'a' of 4- (5) $\exists_y$.P(b,y)

By Universal Generalisation on 'b'of 5- (6) $\forall_x.\exists_y$.P(x,y)

By Simplification of 3 - (7) Q(b,a)

By Existential Generalisation on 'a' of 7- (8) $\exists_y$.Q(b,y)

By Universal Generalisation on 'b'of 8- (9) $\forall_x.\exists_y$.Q(x,y)

By Disjunction of 6,9 (10) ($\forall_x.\exists_y$.P(x,y)) V ($\forall_x.\exists_y$.Q(x,y))

5(c):

$$\exists x \exists y \exists z\,(P(x,y) \land P(z,y) \land P(x,z) \land \neg P(z,x))$$

(a): Given model Satisfies the formula

$$P_1 = \{(x,y) \mid x,y \in \mathbb{N} \land x < y\}.$$

Here P(x,y) is true for x<y

Consider following sample values of x=1,y=3,z=2

P(x,y)=>P(1,3)=> True

P(z,y)=>P(2,3) =>True

P(x,z)=>P(1,2)=>True

¬P(z,x)=>¬P(2,1)=>¬(False)=>True

(b): Given model violates the formula.

$$P_2 = \{(x, x+1) \mid x \in \mathbb{N}\}$$

Here for P(x,y) is true for y=x+1

For any value of x,y,z P(x,y) will be false (c): Given model satisfies the formula.

$$P_3 = \{(A, B) \mid A, B \subseteq \mathbb{N} \wedge A \subseteq B\}$$

Here P(x,y) is true for x,y ⊆N and x ⊆y

Consider following sample values of x={1},y={1,3,2},z={1,2}

P(x,y)=>True , since x ⊆y

P(z,y)=>True, since z ⊆y

P(x,z)=>True, since x ⊆z

¬P(z,x)=>True, since z is not subset of x

5(d):

The following are constraints in First order logic for given 'I' for it to be a pivot with mentioned conditions in question. For any array A,

- $\forall_A$ isArray(A)
- $$\bigwedge_{x=0}^{i-1} \bigwedge_{y=i+1}^{len(A)-1} (read(A,x) < read(A,y))$$

With these two constraints we can determine whether i is a pivot location with mentioned conditions or not.

5(e):

The following are constraints in FOL to determine whether an Array A is a permutation of Array B

- $\forall_A$ isArray(A) $\wedge$ $\forall_B$ isArray(B)
- $\forall_A$ len(A)= $\forall_B$ len(B)
- $$\bigwedge_{i=0}^{len(A)-1} \bigvee_{j=0}^{len(B)-1} (read(A,i) = read(B,j))$$
- $$\bigwedge_{i=0}^{len(B)-1} \bigvee_{j=0}^{len(A)-1} (read(B,i) = read(A,j))$$

When all these constraints are satisfied then we can say that Array A is a permutation of Array B

5(f):

The following are FOL constraints for a STACK . Here x is a stack and y is variable used for stack operations.

- $\forall_{x,y}.top(push(x, y)) = y$
- $\forall_{x,y}.pop(push(x, y)) = x$
- $\forall_{x,y}.(push(x, y) \neq nil)$
- $\forall_{y}.empty(pop(push(nil, y))))$
- $\forall_{x}.(x = nil \rightarrow len(x) = 0)$
- $\forall_{x}.(x \neq nil \rightarrow len(x) > 0)$
- $\forall_{x,y}.(len(push(x, y)) = len(x) + 1$