Jian Li

**Problem 1**

Formulate this problem to SAT,

First in this problem, we have to deal with four constraints below,

1. every task can only be done once,
2. an engineer can only do one task per week,
3. if one task is dependent on the other, the other one must be completed in earlier week than this one,
4. All tasks should be finished.

Assume Xetw represents ***engineer e*** finish the ***taks t*** in ***week w,*** ( 1<=e<=E, 1<=t<=T, 1<=w<=W)

A collection D keep track of all the dependencies among these tasks, if two taks t1, t2 has a dependency relationship as t1 depend on t2, we represent this as a pair (t1, t2), D is a collection of all these pairs.

Thus,

For constraint 1,

For every task t in T, we need to consider all the combinations of engineers and weeks, make every two of them cannot be true at the same time. (I use pseudo code here to express my idea)

initial SAT exp

for every tk in T:

for all pair(ei, wm), (ej,wn) not same in E,W:

SAT and= (-Xeitkwm or –Xejtkwn))

For constraint 2,

For every engineer e in E, for every week w in W,

we need to consider all the two size combinations of tasks, make every of them cannot be true at the same time.

take the SAT exp above

for every ek in E:

for every wn in W:

for all pair(ti, tj), in T:

SAT and= (-Xektiwn or –Xektjwn))

For constraint 3,

For every dependency (n,m) in D, we need to make sure week of n and m, wn and wm has the relationship wn < wm.

take the SAT exp above

for every pair (n,m) in D:

for every wn,wm (where wn< wm) in W:

for all pair(ei, ej), in E:

SAT and= (Xeinwn and Xejmwm)

For constraint 4,

we need to consider all the combinations of E,T and W.

for every e in E:

for every t in T:

for every w in W:

SAT and= Xetw

the final SAT is expression is the one that express this problem.

**Problem 2**

The answer is:

(A -B Y1) and (-Y1 –G H) and (-B C Y2) and (-Y2 D Y3) and (-Y3 –F –G) and (A D Y4) and (-Y4 –E Y5) and ( -Y5 –H –I) and (F I Y6) and (-Y6 F I)

**Problem 3**

**a.**

0 dpll on Exp() =

VXY and -W-Y-Z and V-YZ and W-YZ and -V-WY and -VX-Z and

V-XY and -V-WZ and WX-Z and -VXZ and -V-W-Z and -WY-Z and

V-WZ and WXZ and WY-Z and -XYZ and -WYZ and -W-XZ and

X-Y-Z and W-X-Y and V-W-Y

1 Trying V = True

1 dpll on Exp(V=True) =

-W-Y-Z and W-YZ and -WY and X-Z and

-WZ and WX-Z and XZ and -W-Z and -WY-Z and

WXZ and WY-Z and -XYZ and -WYZ and -W-XZ and

X-Y-Z and W-X-Y

2 Trying W = True

2 dpll on Exp(V=True, W=True) =

-Y-Z and Y and X-Z and

Z and XZ and -Z and Y-Z and -XYZ and YZ and -XZ and X-Y-Z

3 Using unit clause to set Y=True

3 dpll on Exp(V=True, W=True, Y=True) =

-Z and X-Z and Z and XZ and -Z and -XZ and X-Z

4 Using unit clause to set Z=False

4 dpll on Exp(V=True, W=True, Y=True, Z=True) =

-Z and X-Z and Z and XZ and -Z and -XZ and X-Z

not satisfied

end 4

not satisfied

end 3

2 Trying W = False

2 dpll on Exp(V=True, W=False) =

-YZ and X-Z and X-Z and XZ and XZ and Y-Z and -XYZ and X-Y-Z and -X-Y

3 Trying X = True

3 dpll on Exp(V=True, W=False, X= True) =

-YZ and Y-Z and YZ and -Y

4 Using unit clause to set Y=False

4 dpll on Exp(V=True, W=False, X=True, Y=False) =

-Z and Z

5 Using unit clause to set Z=False

5 dpll on Exp(V=True, W=False, X=True, Y=False, Z=False) =

not satisfied

end 5

not satisfied

end 4

3 Trying X = False

3 dpll on Exp(V=True, W=False, X=False) =

-YZ and -Z and -Z and Z and Z and Y-Z and -Y-Z

4 Using unit clause to set Z=False

4 dpll on Exp(V=True, W=False, X=True, Z=False) =

not satisfied

end 4

not satisfied

end 3

not satisfied

end 2

1 Trying V = False

1 dpll on Exp(V=False) =

XY and -W-Y-Z and -YZ and W-YZ and

-XY and WX-Z and -WY-Z and

-WZ and WXZ and WY-Z and -XYZ and -WYZ and -W-XZ and

X-Y-Z and W-X-Y and -W-Y

2 Trying W = True

2 dpll on Exp(V=False, W=True) =

XY and -Y-Z and -YZ and

-XY and Y-Z and Z and -XYZ and YZ and -XZ and

X-Y-Z and -Y

3 Using unit clause to set Z=True

3 dpll on Exp(V=False, W=True, Z=True) =

XY and -Y and -XY and Y and X-Y and -Y

4 Using unit clause to set Y=False

4 dpll on Exp(V=False, W=True, Z=True, Y=False) =

not satisfied

end 4

not satisfied

end 3

2 Trying W = False

2 dpll on Exp(V=False, W=False) =

XY and -YZ and -YZ and

-XY and X-Z and XZ and Y-Z and -XYZ and X-Y-Z and -X-Y

3 Trying X = True

3 dpll on Exp(V=False, W=False, X=True) =

-YZ and -YZ and Y and Y-Z and YZ and -Y

4 Using unit clause to set Y=True

4 dpll on Exp(V=False, W=False, X=True, Y=True) =

not satisfied

end 4

3 Trying X = False

3 dpll on Exp(V=False, W=False, X=False) =

Y and -YZ and -YZ and

-Z and Z and Y-Z and -Y-Z and

4 Using unit clause to set Y=True

4 dpll on Exp(V=False, W=False, X=False, Y=True) =

Z and Z and -Z and Z and -Z

5 Using unit clause to set Z=True

5 dpll on Exp(V=False, W=False, X=False, Y=True,Z=False) =

not satisfied

end 5

not satisfied

end 4

not satisfied

end 3

not satisfied

end 2

not satisfied

end 1

not satisfied

end 0

**b.**

0 dpll on Exp() =

VWX and VXY and V-YZ and -V-XY and VX-Z and V-WY and

-VW-X and V-W-X and -V-X-Z and -XY-Z and -VY-Z and -W-X-Z and

-WY-Z and V-W-Z and -WYZ and V-XZ and VYZ and WYZ and

-V-W-Y and -W-X-Y and -V-WX

1 Trying V = True

1 dpll on Exp(V=True) =

-XY and W-X and -X-Z and -XY-Z and Y-Z and -W-X-Z and

-WY-Z and -WYZ and WYZ and-W-Y and -W-X-Y and -WX

2 Trying W = True

2 dpll on Exp(V=True, W=True) =

-XY and -X-Z and -XY-Z and Y-Z and -X-Z and

Y-Z and YZ and -Y and -X-Y and X

3 Using unit clause to set Y = False

3 dpll on Exp(V=True, W=True, Y=False) =

-X and -X-Z and -X-Z and -Z and -X-Z and

-Z and Z and X

4 Using unit clause to set X = False

4 dpll on Exp(V=True, W=True, Y=False, X=False) =

not satisfied

end 4

not satisfied

end 3

2 Trying W = False

2 dpll on Exp(V=True, W=False) =

-XY and -X and -X-Z and -XY-Z and Y-Z and YZ

3 Using unit clause to set X=False

3 dpll on Exp(V=True, W=False, X=False) =

Y-Z and YZ

4 Trying Y = True

4 dpll on Exp(V=True, W=False, X=False, Y=True) =

Y-Z and YZ

satisfied

end 4

satisfied

end 3

satisfied

end 2

satisfied

end 1

satisfied

end 0

**Problem 4**

a.

This problem can be formulated as a SAT problem:

Assume we have n vertex(v1,v2,…, vn) and m edges(e1,e2,…,em)in our graph, the available colors are c1, c2, and c3.

For each vertex vn, we assume three variable vn\_c1, vn\_c2, vn\_c3 to represent the color that this node is assigned.

1. For a vertex vn , it must have only one kind of color, we express it in SAT as :

*( vn\_c1 or vn\_c2 or vn\_c3) and*

*not (vn\_c1 and vn\_c2) and*

*not (vn\_c1 and vn\_c3) and*

*not (vn\_c2 and vn\_c3)*

*this is equal to,*

*( vn\_c1 vn\_c2 vn\_c3) and (-vn\_c1 -vn\_c2) and (-vn\_c1 -vn\_c3) and (-vn\_c2 -vn\_c3)*

2. For an edges em, the vertexes(vi, vj)that share this em cannot have the same color, we express the constraint as:

*not ( vi\_c1 and vj\_c1) and*

*not ( vi\_c2 and vj\_c2) and*

*not ( vi\_c3 and vj\_c3)*

*this equals to,*

*(-vi\_c1 –vj\_c1) and (-vi\_c2 –vj\_c2) and (-vi\_c3 –vj\_c3)*

After we get the first constraint of all vertexes and second of all edges together, we formulate this as a SAT problem.

b.

According to the solution above, we formulate this problem as a SAT expression below: ( represent vertex 1 2 3 4 as A B C D for convenience)

*(A\_c1 A\_c2 A\_c3) and (B\_c1 B\_c2 B\_c3) and (C\_c1 C\_c2 C\_c3) and (D\_c1 D\_c2 D\_c3)*

*and (-A\_c1 -A\_c2) and (-A\_c1 -A\_c3) and (-A\_c2 -A\_c3)*

*and (-B\_c1 -B\_c2) and (-B\_c1 -B\_c3) and (-B\_c2 -B\_c3)*

*and (-C\_c1 -C\_c2) and (-C\_c1 -C\_c3) and (-C\_c2 -C\_c3)*

*and (-D\_c1 -D\_c2) and (-D\_c1 -D\_c3) and (-D\_c2 -D\_c3)*

*and (-A\_c1 -B\_c1) and (-A\_c2 -B\_c2) and (-A\_c3 -B\_c3)*

*and (-A\_c1 -C\_c1) and (-A\_c2 -C\_c2) and (-A\_c3 -C\_c3)*

*and (-B\_c1 -C\_c1) and (-B\_c2 -C\_c2) and (-B\_c3 -C\_c3)*

*and (-B\_c1 -D\_c1) and (-B\_c2 -D\_c2) and (-B\_c3 -D\_c3)*

*and (-C\_c1 -D\_c1) and (-C\_c2 -D\_c2) and (-C\_c3 -D\_c3)*

**Solve it with DPLL:**

0 dpll on Exp() =

(A\_c1 A\_c2 A\_c3) and (B\_c1 B\_c2 B\_c3) and (C\_c1 C\_c2 C\_c3) and (D\_c1 D\_c2 D\_c3)

and (-A\_c1 -A\_c2) and (-A\_c1 -A\_c3) and (-A\_c2 -A\_c3)

and (-B\_c1 -B\_c2) and (-B\_c1 -B\_c3) and (-B\_c2 -B\_c3)

and (-C\_c1 -C\_c2) and (-C\_c1 -C\_c3) and (-C\_c2 -C\_c3)

and (-D\_c1 -D\_c2) and (-D\_c1 -D\_c3) and (-D\_c2 -D\_c3)

and (-A\_c1 -B\_c1) and (-A\_c2 -B\_c2) and (-A\_c3 -B\_c3)

and (-A\_c1 -C\_c1) and (-A\_c2 -C\_c2) and (-A\_c3 -C\_c3)

and (-B\_c1 -C\_c1) and (-B\_c2 -C\_c2) and (-B\_c3 -C\_c3)

and (-B\_c1 -D\_c1) and (-B\_c2 -D\_c2) and (-B\_c3 -D\_c3)

and (-C\_c1 -D\_c1) and (-C\_c2 -D\_c2) and (-C\_c3 -D\_c3)

1 Trying A\_c1 = True

1 dpll on Exp(A\_c1=True) =

(B\_c1 B\_c2 B\_c3) and (C\_c1 C\_c2 C\_c3) and (D\_c1 D\_c2 D\_c3)

and (-A\_c2) and (-A\_c3) and (-A\_c2 -A\_c3)

and (-B\_c1 -B\_c2) and (-B\_c1 -B\_c3) and (-B\_c2 -B\_c3)

and (-C\_c1 -C\_c2) and (-C\_c1 -C\_c3) and (-C\_c2 -C\_c3)

and (-D\_c1 -D\_c2) and (-D\_c1 -D\_c3) and (-D\_c2 -D\_c3)

and (-B\_c1) and (-A\_c2 -B\_c2) and (-A\_c3 -B\_c3)

and (-C\_c1) and (-A\_c2 -C\_c2) and (-A\_c3 -C\_c3)

and (-B\_c1 -C\_c1) and (-B\_c2 -C\_c2) and (-B\_c3 -C\_c3)

and (-B\_c1 -D\_c1) and (-B\_c2 -D\_c2) and (-B\_c3 -D\_c3)

and (-C\_c1 -D\_c1) and (-C\_c2 -D\_c2) and (-C\_c3 -D\_c3)

2 Using unit clause to set A\_c2=False

2 dpll on Exp(A\_c1=True,A\_c2=False) =

(B\_c1 B\_c2 B\_c3) and (C\_c1 C\_c2 C\_c3) and (D\_c1 D\_c2 D\_c3)

and (-A\_c3)

and (-B\_c1 -B\_c2) and (-B\_c1 -B\_c3) and (-B\_c2 -B\_c3)

and (-C\_c1 -C\_c2) and (-C\_c1 -C\_c3) and (-C\_c2 -C\_c3)

and (-D\_c1 -D\_c2) and (-D\_c1 -D\_c3) and (-D\_c2 -D\_c3)

and (-B\_c1) and (-A\_c3 -B\_c3)

and (-C\_c1) and (-A\_c3 -C\_c3)

and (-B\_c1 -C\_c1) and (-B\_c2 -C\_c2) and (-B\_c3 -C\_c3)

and (-B\_c1 -D\_c1) and (-B\_c2 -D\_c2) and (-B\_c3 -D\_c3)

and (-C\_c1 -D\_c1) and (-C\_c2 -D\_c2) and (-C\_c3 -D\_c3)

3 Using unit clause to set A\_c3=False

3 dpll on Exp(A\_c1=True,A\_c2=False,A\_c3=False) =

(B\_c1 B\_c2 B\_c3) and (C\_c1 C\_c2 C\_c3) and (D\_c1 D\_c2 D\_c3)

and (-B\_c1 -B\_c2) and (-B\_c1 -B\_c3) and (-B\_c2 -B\_c3)

and (-C\_c1 -C\_c2) and (-C\_c1 -C\_c3) and (-C\_c2 -C\_c3)

and (-D\_c1 -D\_c2) and (-D\_c1 -D\_c3) and (-D\_c2 -D\_c3)

and (-B\_c1)

and (-C\_c1)

and (-B\_c1 -C\_c1) and (-B\_c2 -C\_c2) and (-B\_c3 -C\_c3)

and (-B\_c1 -D\_c1) and (-B\_c2 -D\_c2) and (-B\_c3 -D\_c3)

and (-C\_c1 -D\_c1) and (-C\_c2 -D\_c2) and (-C\_c3 -D\_c3)

4 Using unit clause to set B\_c1=False

4 dpll on Exp(A\_c1=True,A\_c2=False,A\_c3=False,B\_c1=False) =

(B\_c2 B\_c3) and (C\_c1 C\_c2 C\_c3) and (D\_c1 D\_c2 D\_c3)

and (-B\_c2 -B\_c3)

and (-C\_c1 -C\_c2) and (-C\_c1 -C\_c3) and (-C\_c2 -C\_c3)

and (-D\_c1 -D\_c2) and (-D\_c1 -D\_c3) and (-D\_c2 -D\_c3)

and (-C\_c1)

and (-B\_c2 -C\_c2) and (-B\_c3 -C\_c3)

and (-B\_c2 -D\_c2) and (-B\_c3 -D\_c3)

and (-C\_c1 -D\_c1) and (-C\_c2 -D\_c2) and (-C\_c3 -D\_c3)

5 Using unit clause to set C\_c1=False

5 dpll on Exp(A\_c1=True,A\_c2=False,A\_c3=False,B\_c1=False,C\_c1=False) =

(B\_c2 B\_c3) and (C\_c2 C\_c3) and (D\_c1 D\_c2 D\_c3)

and (-B\_c2 -B\_c3)

and (-C\_c2 -C\_c3)

and (-D\_c1 -D\_c2) and (-D\_c1 -D\_c3) and (-D\_c2 -D\_c3)

and (-B\_c2 -C\_c2) and (-B\_c3 -C\_c3)

and (-B\_c2 -D\_c2) and (-B\_c3 -D\_c3)

and (-C\_c2 -D\_c2) and (-C\_c3 -D\_c3)

6 Trying B\_c2 = True

6 dpll Exp(A\_c1=True,A\_c2=False,A\_c3=False,B\_c1=False,C\_c1=False,B\_c2=True) =

(C\_c2 C\_c3) and (D\_c1 D\_c2 D\_c3)

and (-B\_c3)

and (-C\_c2 -C\_c3)

and (-D\_c1 -D\_c2) and (-D\_c1 -D\_c3) and (-D\_c2 -D\_c3)

and (-C\_c2) and (-B\_c3 -C\_c3)

and (-D\_c2) and (-B\_c3 -D\_c3)

and (-C\_c2 -D\_c2) and (-C\_c3 -D\_c3)

7 Using unit clause to set B\_c3=False

7 dpll Exp(A\_c1=True,A\_c2=False,A\_c3=False,B\_c1=False,C\_c1=False,B\_c2=True,B\_c3=False) =

(C\_c2 C\_c3) and (D\_c1 D\_c2 D\_c3)

and (-C\_c2 -C\_c3)

and (-D\_c1 -D\_c2) and (-D\_c1 -D\_c3) and (-D\_c2 -D\_c3)

and (-C\_c2)

and (-D\_c2)

and (-C\_c2 -D\_c2) and (-C\_c3 -D\_c3)

8 Using unit clause to set C\_c2 = False

8 dpll Exp(A\_c1=True,A\_c2=False,A\_c3=False,B\_c1=False,C\_c1=False,B\_c2=True,B\_c3=False, C\_c2=False) =

(C\_c3) and (D\_c1 D\_c2 D\_c3)

and (-D\_c1 -D\_c2) and (-D\_c1 -D\_c3) and (-D\_c2 -D\_c3)

and (-D\_c2)

and (-C\_c3 -D\_c3)

9 Using unit clause to set C\_c3= True

9 dpll Exp(A\_c1=True,A\_c2=False,A\_c3=False,B\_c1=False,C\_c1=False,B\_c2=True,B\_c3=False, C\_c2=False,C\_c3=True) =

(D\_c1 D\_c2 D\_c3) and (-D\_c1 -D\_c2) and (-D\_c1 -D\_c3) and (-D\_c2 -D\_c3) and (-D\_c2) and (-D\_c3)

10 Using unit clause to set D\_c2= False

10 dpll Exp(A\_c1=True,A\_c2=False,A\_c3=False,B\_c1=False,C\_c1=False,B\_c2=True,B\_c3=False, C\_c2=False,C\_c3=True,D\_c2=False) =

(D\_c1 D\_c3) and (-D\_c1 -D\_c3) and (-D\_c3)

11 Using unit clause to set D\_c3=False

11 dpll Exp(A\_c1=True,A\_c2=False,A\_c3=False,B\_c1=False,C\_c1=False,B\_c2=True,B\_c3=False, C\_c2=False,C\_c3=True,D\_c2=False,D\_c3=False) =

(D\_c1)

12 Using unit clause to set D\_c1=True

12 dpll Exp(A\_c1=True,A\_c2=False,A\_c3=False,B\_c1=False,C\_c1=False,B\_c2=True,B\_c3=False, C\_c2=False,C\_c3=True,D\_c2=False,D\_c3=False,D\_c1=True) =

satisfied

end 12

satisfied

end 11

satisfied

end 10

satisfied

end 9

satisfied

end 8

satisfied

end 7

satisfied

end 6

satisfied

end 5

satisfied

end 4

satisfied

end 3

satisfied

end 2

satisfied

end 1

satisfied

end 0