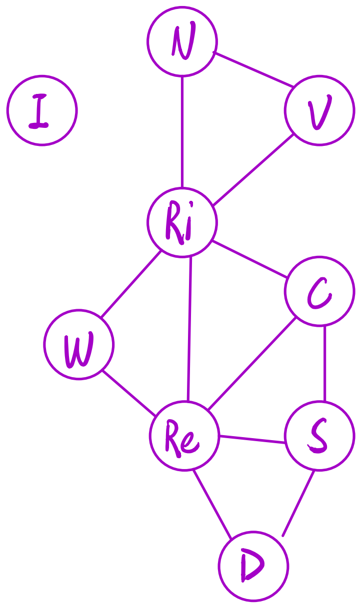
Problem 1: Solution

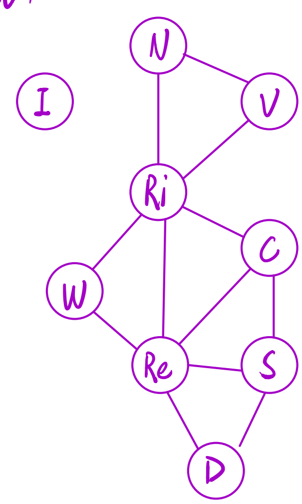
a. Constraint Graph:




① General-purpose CSP algorithm use the graph structure to speed up search: Ⓘ is an independent subproblem, so the algorithm can be paralleled.

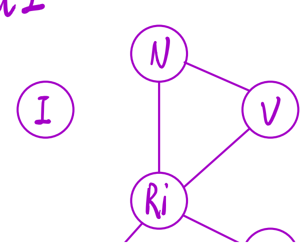
② Taking advantage of constraint propagation, a search procedure never have to consider invalid situation, so we have less assignment to look at.

b. Level 1:

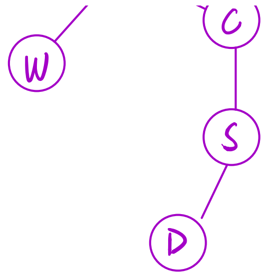



N: { MRV: 3 , dh: 2 }
V: { MRV: 3 , dh: 2 }
Ri { MRV: 3 , dh: 5 }
W: { MRV: 3 , dh: 2 }
C: { MRV: 3 , dh: 3 }
① Re: { MRV: 3 , dh: 5 } chose it!
S: { MRV: 3 , dh: 3 }
D: { MRV: 3 , dh: 2 }
I: { MRV: 3 , dh: 0 }

Level 2:




N: { MRV: 3 , dh: 2 }
V: { MRV: 3 , dh: 2 }
② Ri { MRV: 2 , dh: 4 } Chose it!
W: { MRV: 2 , dh: 1 }




C: { MRV: 2 , dh: 2 }

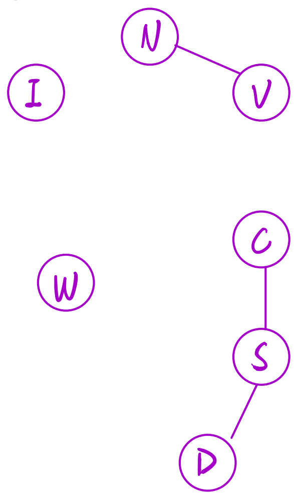
① ~~Re: { MRV: 3 , dh: 5 }~~

S: { MRV: 2 , dh: 2 }

D: { MRV: 2 , dh: 1 }

I: { MRV: 3 , dh: 0 }

Level 3:




N: { MRV: 2 , dh: 1 }

V: { MRV: 2 , dh: 1 }

② ~~Ri { MRV: 2 , dh: 4 }~~

W: { MRV: 1 , dh: 0 }

③ C: { MRV: 1 , dh: 1 } choose it!

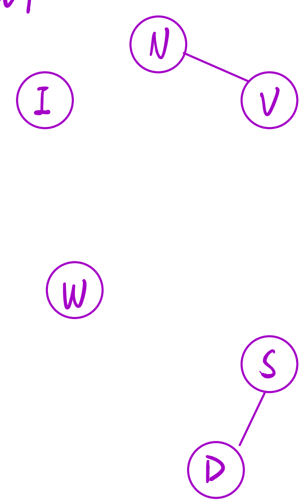
① ~~Re: { MRV: 3 , dh: 5 }~~

S: { MRV: 2 , dh: 2 }

D: { MRV: 2 , dh: 1 }

I: { MRV: 3 , dh: 0 }

Level 4:




N: { MRV: 2 , dh: 1 }

V: { MRV: 2 , dh: 1 }

② ~~Ri { MRV: 2 , dh: 4 }~~

W: { MRV: 1 , dh: 0 }

③ ~~C: { MRV: 1 , dh: 1 }~~

① ~~Re: { MRV: 3 , dh: 5 }~~

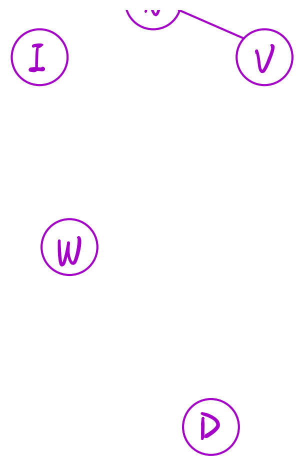
④ S: { MRV: 1 , dh: 1 } choose it!

D: { MRV: 2 , dh: 1 }

I: { MRV: 3 , dh: 0 }

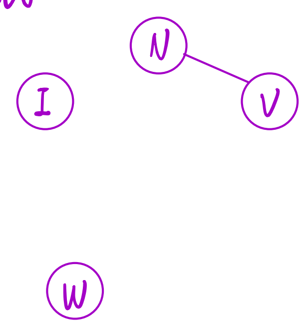
Level 5:

N: { MRV: 2 , dh: 1 }



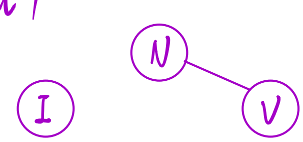

V: { MRV: 2 , dh: 1 }
~~② Ri { MRV: 2 , dh: 4 }~~
W: { MRV: 1 , dh: 0 }
~~③ C: { MRV: 1 , dh: 1 }~~
~~① Re: { MRV: 3 , dh: 5 }~~
~~④ S: { MRV: 1 , dh: 1 }~~
⑤ D: { MRV: 1 , dh: 0 } choose it!
I: { MRV: 3 , dh: 0 }

Level 6:




N: { MRV: 2 , dh: 1 }
V: { MRV: 2 , dh: 1 }
~~② Ri { MRV: 2 , dh: 4 }~~
⑥ W: { MRV: 1 , dh: 0 } choose it!
~~③ C: { MRV: 1 , dh: 1 }~~
~~① Re: { MRV: 3 , dh: 5 }~~
~~④ S: { MRV: 1 , dh: 1 }~~
~~⑤ D: { MRV: 1 , dh: 0 }~~
I: { MRV: 3 , dh: 0 }

Level 7:




⑦ N: { MRV: 2 , dh: 1 } choose it!
V: { MRV: 2 , dh: 1 }
~~② Ri { MRV: 2 , dh: 4 }~~
~~⑥ W: { MRV: 1 , dh: 0 }~~
~~③ C: { MRV: 1 , dh: 1 }~~
~~① Re: { MRV: 3 , dh: 5 }~~
~~④ S: { MRV: 1 , dh: 1 }~~
~~⑤ D: { MRV: 1 , dh: 0 }~~
I: { MRV: 3 , dh: 0 }

I { MRV: 3 , dh: 0 }

Level 8:

I V

⑦ ~~N: { MRV: 2 , dh: 1 }~~

⑧ V: { MRV: 1 , dh: 0 } choose it!

② ~~Ri { MRV: 2 , dh: 4 }~~

⑥ ~~W: { MRV: 1 , dh: 0 }~~

③ ~~C: { MRV: 1 , dh: 1 }~~

① ~~Re: { MRV: 3 , dh: 5 }~~

④ ~~S: { MRV: 1 , dh: 1 }~~

⑤ ~~D: { MRV: 1 , dh: 0 }~~

I: { MRV: 3 , dh: 0 }

Level 9:

I

⑦ ~~N: { MRV: 2 , dh: 1 }~~

⑧ ~~V: { MRV: 2 , dh: 1 }~~

② ~~Ri { MRV: 2 , dh: 4 }~~

⑥ ~~W: { MRV: 1 , dh: 0 }~~

③ ~~C: { MRV: 1 , dh: 1 }~~

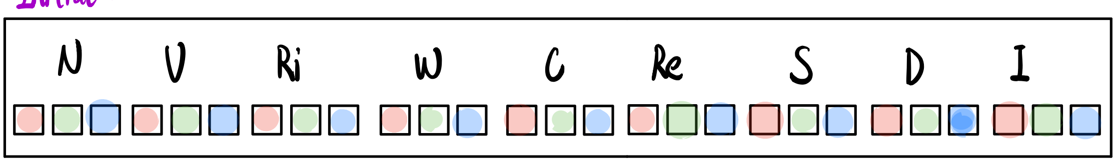
① ~~Re: { MRV: 3 , dh: 5 }~~

④ ~~S: { MRV: 1 , dh: 1 }~~

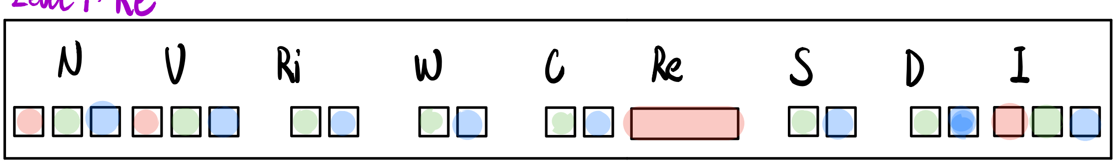
⑤ ~~D: { MRV: 1 , dh: 0 }~~

⑨ I: { MRV: 3 , dh: 0 } choose it!

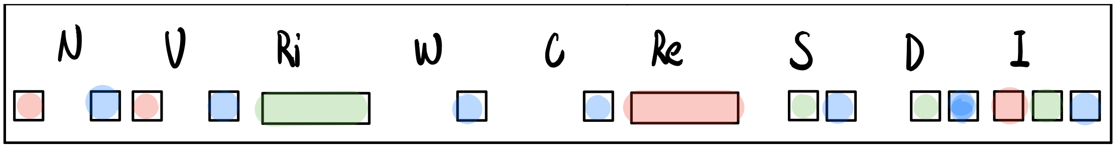
c. Initial:



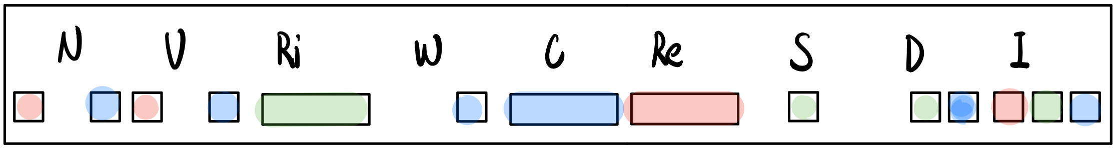

Level 1: Re



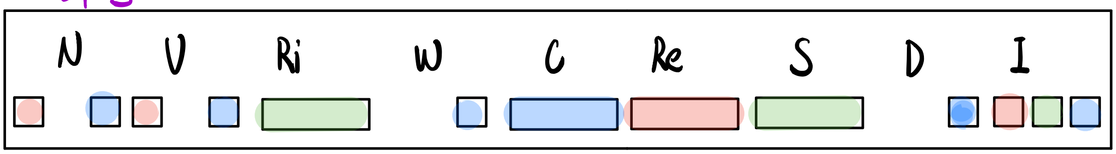

Level 2: Ri



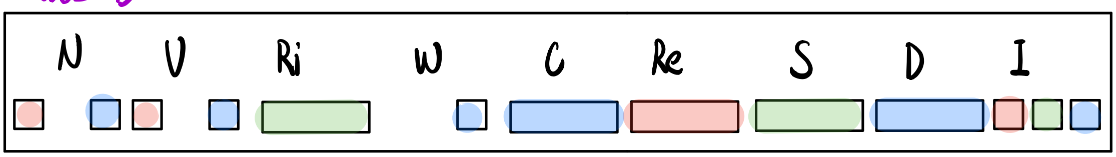

Level 3: C



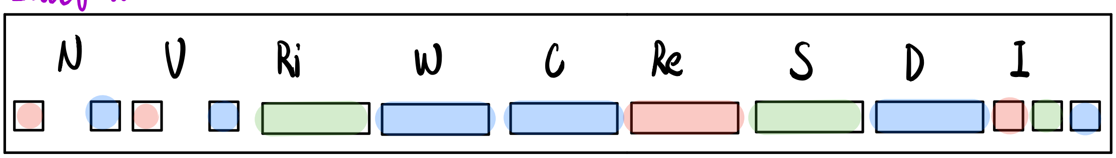

Level 4: S



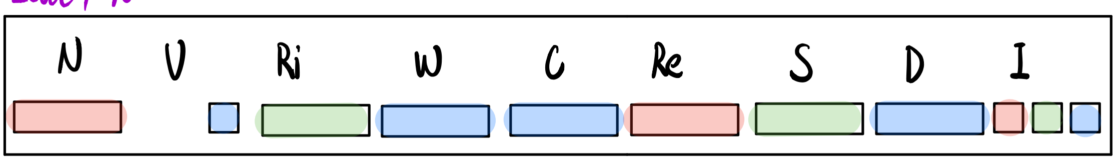

Level 5: D



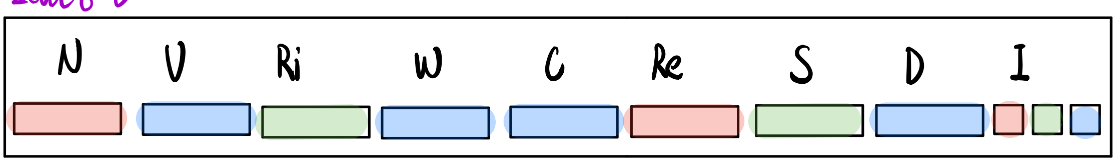

Level 6: W



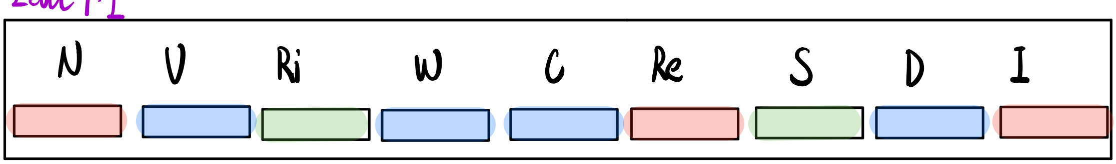

Level 7: N




Level 8: V




Level 9: I




d. Valid Solution

{Re: Red, Ri: Green, C: Blue, S: Green, D: Blue, W: Blue, N: Red, V: Blue, I: Red}.

Problem 2 : Solution

```
Def checkEquivalence(KB1, KB2):
    return TT_ENTAILS(KB1, KB2) == TT_ENTAILS(KB2, KB1)
Def TT_ENTAILS(KB1, KB2):
    symbols ← a list of proposition symbols in KB1 and KB2
    return TT-CHECK-All(KB1, KB2, symbols, [])
Def TT_CHECK_ALL(KB1, KB2, symbols, model)
    if EMPTY?(Symols) THEN
        if PL-TRUE?(KB1, model) then return PL-TRUE?(KB2, model)
        else return true
    else do
        P← FIRST(symbols) ; rest← REST(symbols)
        return TT_CHECK_ALL(KB1, KB2, rest, EXTEND(P, true, model)) and
            TT_CHECK_ALL(KB1, KB2, rest, EXTEND(P, false, model))
```

Problem 3 : Solution

a. according to the definition of entailment : KB entails $\alpha$ iff $\alpha$ is true in all worlds where KB is true.

in line 1, 3, 7. when KB is true, S1 is true.

As a result, $KB \models S1$

b. according to the problem, we obtain:

| NOT(KB) | NOT(S1) |
|---|---|
| F | F |
| T | F |
| F | F |

In line 2,4 , when NOT(KB) is true,

NOT(S1) is false

$\therefore$ NOT(KB) $\not\models$ NOT(S1)

| | |
|---|---|
| T | F |
| T | T |
| T | T |
| F | F |
| T | T |

## Problem 4: Solution.

Let's rewrite the two case to render the knowledge base true:

$$\neg(A \wedge B \wedge \neg C \wedge D) \wedge \neg(A \wedge \neg B \wedge C \wedge \neg D)$$

Conversion to CNF: Move $\neg$ inwards & Apply distributivity law ($\vee$ over $\wedge$) and flaten:

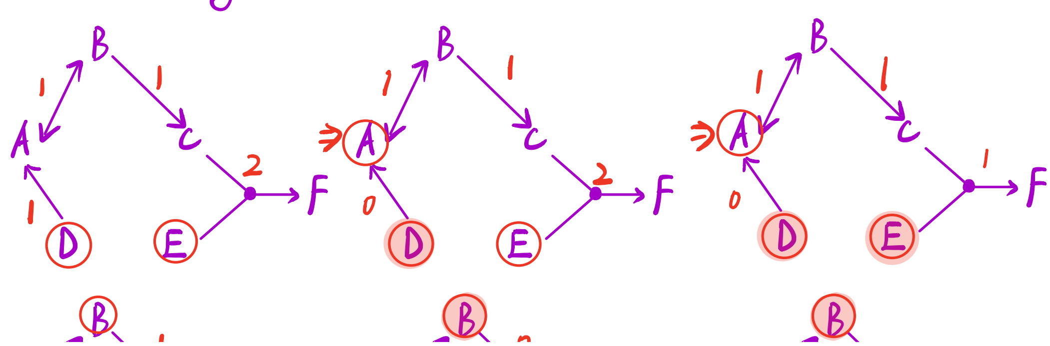
$$(\neg A \vee \neg B \vee C \vee \neg D) \wedge (\neg A \vee B \vee \neg C \vee D)$$

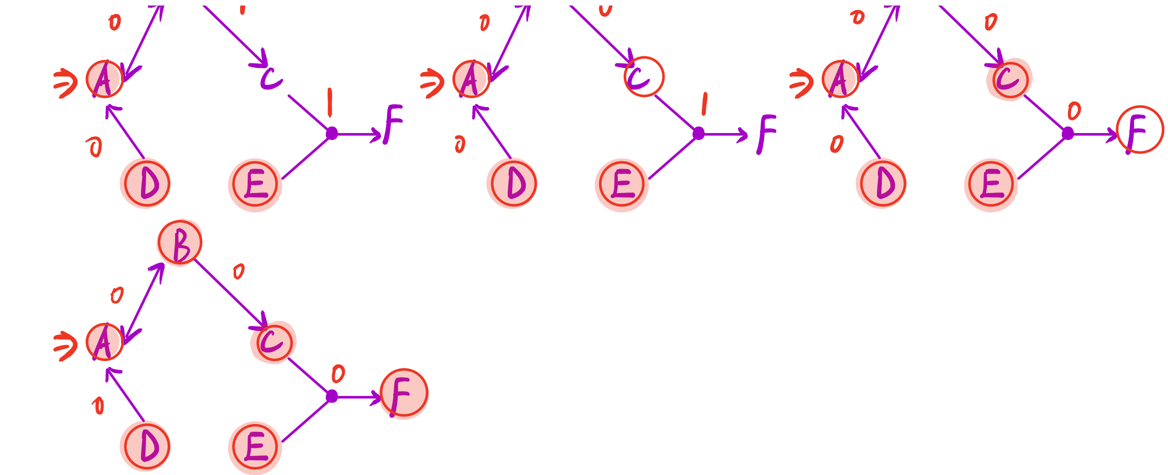
## Problem 5: Solution

Horn Form Checking & Conversion:

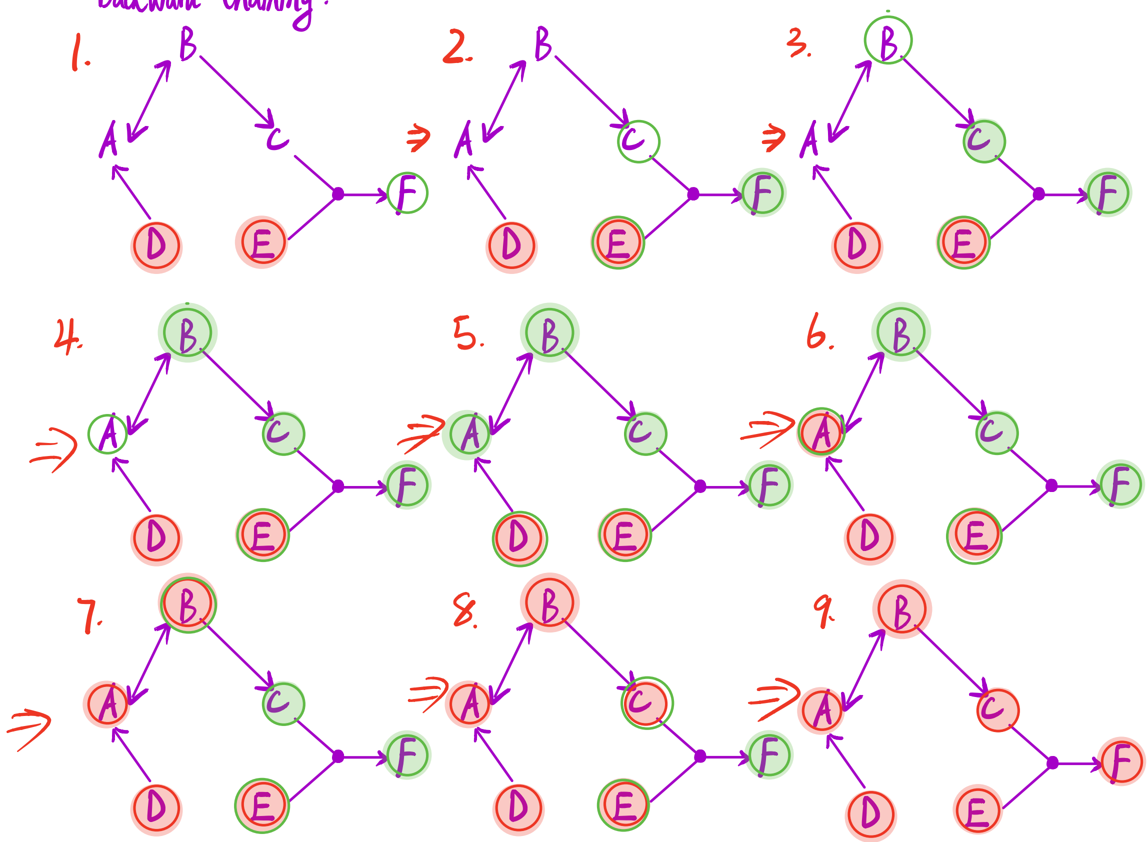
$$\left\{\begin{array}{l} A \Leftrightarrow B \\ B \Rightarrow C \\ D \Rightarrow A \\ C \wedge E \Rightarrow F \\ E \\ D \end{array}\right. \Rightarrow \left\{\begin{array}{l} A \Rightarrow B \,,\; B \Rightarrow A \\ B \Rightarrow C \,,\; D \Rightarrow A \\ C \wedge E \Rightarrow F \\ E \\ D \end{array}\right.$$

Forward chaining:






Backward Chaining:




Resolution:

KB: $(A\Leftrightarrow B)\wedge(B\Rightarrow C)\wedge(D\Rightarrow A)\wedge((C\wedge E)\Rightarrow F)\wedge E\wedge D$ $\quad \alpha = \neg F$

conversion to CNF:

① Eliminate $\Leftrightarrow$

$(A\Rightarrow B)\wedge(B\Rightarrow A)\wedge(B\Rightarrow C)\wedge(D\Rightarrow A)\wedge((C\wedge E)\Rightarrow F)\wedge E\wedge D$

② Eliminate $\Rightarrow$

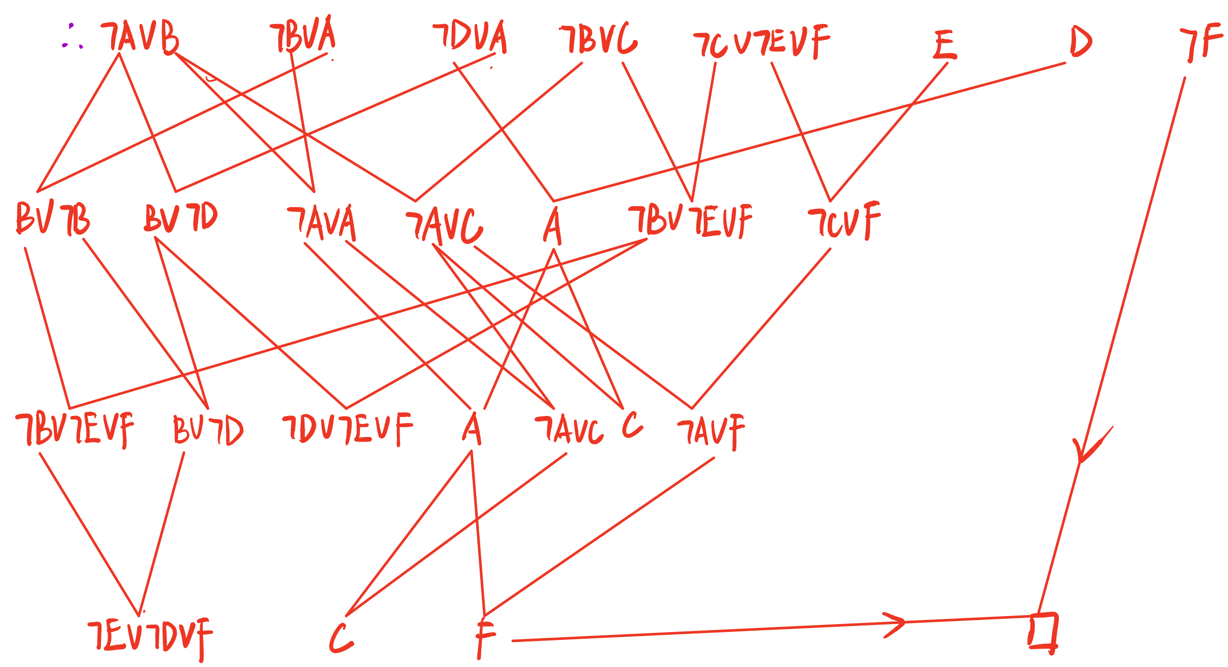
$(\neg A \vee B) \wedge (\neg B \vee A) \wedge (\neg B \vee C) \wedge (\neg D \vee A) \wedge (\neg (C \wedge E) \vee F) \wedge E \wedge D$

③ Move $\neg$ inwards:

$(\neg A \vee B) \wedge (\neg B \vee A) \vee (\neg B \vee C) \wedge (\neg D \vee A) \wedge ((\neg C \vee \neg E) \vee F) \wedge E \wedge D$

④ Flatten:

$(\neg A \vee B) \wedge (\neg B \vee A) \vee (\neg B \vee C) \wedge (\neg D \vee A) \wedge (\neg C \vee \neg E \vee F) \wedge E \wedge D$




## Problem 6:

a. constants:

May 1.2017 : A date, May 2.2017 : A date May 3.2017 ; A date

Mary : A person , John : A person 10000\$ check : A noun lawn : A noun.

predicates :

Rain(x) : True when on x is raining

Give(x, y, w, z) : True when x give w to y on z

Mow(x, y, z) : True when x mow y on z

KB:

(Rain(May1_2017) $\Rightarrow$ Give(John, Mary, 10000\$check, May2.2017)) and
(Give(John, Mary, 10000\$check, May2.2017) $\Rightarrow$ Mow(Mary, lawn, May3_2017))

b. ① not (Rain(May1_2017))

② Give(John, Mary, 10000\$check, May2.2017)

③ Mow(Mary, lawn, May3_2017))

c. Symbols:

R: Rains on May 1, 2017

C: John give Mary \$10000 check on May 2. 2017.

M: Mary mows the lawn on May 3, 2017.

convert a & b we obtain:

a: $(R \Rightarrow C) \wedge (C \Rightarrow M)$

b: $\neg R \wedge C \wedge M$

d: $(R \Rightarrow C) \wedge (C \Rightarrow M)$

$= (\neg R \vee C) \wedge (C \Rightarrow M)$

when $\neg R$, $C$, $M$ is true, the contract can be $(1 \vee C) \wedge (1 \Rightarrow 1)$ which is also true.

$\therefore$ Event $\neg R \vee C \wedge M \models$ contract. $(\neg R \vee C) \wedge (\neg C \vee M)$

It means that the contract isn't violated.

Problem 7: Solution

| | | |
|---|---|---|
| Taller(John, y) | Taller(x, Son(x)) | {x/John, y/Son(x)} |
| Taller(y, Barry) | Taller(Barry, x) | fail Barry taller than himself not make sense |
| Taller(x, Jane) | Taller(Bob. Jane) | {x/Bob} |
| Taller(Son(x). Jane) | Taller(Bob, Jane) | fail Son(x) is a function not a variable |
| Taller(Barry, John) | Taller(x, y) | {x/Barry, y/John} |