CSCI-UA 472 Artificial Intelligence

Muhammad Wajahat Mirza

mwm356@nyu.edu

Homework 02

September 24, 2020

Problem 1

Translate the following sentences into CNF:

1. $A \Rightarrow \neg(B \land C)$
2. $C \Leftrightarrow \neg(D \lor E)$
3. $D \Rightarrow B$
4. $(D \land E) \Rightarrow \neg B \land C$
5. $D \Leftrightarrow E$

Solution to Problem 1

1. $A \Rightarrow \neg(B \land C)$
   translate ($\Rightarrow$)
   $\neg A \lor (\neg(B \land C))$ Distribute ($\neg$) over B, C
   $\neg A \lor (\neg B \lor \neg C)$
   $\neg A \lor \neg B \lor \neg C$

   $\longrightarrow \mathbf{S} = \{\neg A \lor \neg B \lor \neg C.\}$

2. $C \Leftrightarrow \neg(D \lor E)$
   translate ($\Leftrightarrow$)
   $(C \Rightarrow \neg(D \lor E)) \land (\neg(D \lor E) \Rightarrow C)$
   translate ($\Rightarrow$)
   $(\neg C \lor \neg(D \lor E)) \land (\neg(\neg(D \lor E)) \lor C)$
   $(\neg C \lor (\neg D \land \neg E)) \land (\neg(\neg D \land \neg E) \lor C)$
   $(\neg C \lor \neg D) \land (\neg C \lor \neg E) \land (D \lor E \lor C)$

$\longrightarrow \mathbf{S} = \{$
$\neg C \vee \neg D.$
$\neg C \vee \neg E.$
$D \vee E \vee C.\}$

3. $D \Rightarrow B$
   Translate ($\Rightarrow$)
   $\neg D \vee B$

   $\longrightarrow \mathbf{S} = \{\neg D \vee B.\}$

4. $(D \wedge E) \Rightarrow \neg B \wedge A$
   Translate $\Rightarrow$
   $\neg(D \wedge E) \vee (\neg B \wedge A)$
   $(\neg D \vee \neg E) \vee (\neg B \wedge A)$
   $(\neg D \vee \neg E \vee \neg B) \wedge (\neg D \vee \neg E \vee A)$

   $\longrightarrow \mathbf{S} = \{$
   $\neg D \vee \neg E \vee \neg B.$
   $\neg D \vee \neg E \vee A.\}$

5. $D \Leftrightarrow E$
   Translate ($\Leftrightarrow$)
   $(D \Rightarrow E) \wedge (E \Rightarrow D)$
   Translate ($\Rightarrow$)
   $(\neg D \vee E) \wedge (\neg E \vee D)$

   $\longrightarrow \mathbf{S} = \{$
   $\neg D \vee E.$
   $\neg E \vee D.\}$

Problem 2

Show a trace of the running of the Davis-Putnam algorithm on the clauses that you have derived in problem 1 (the set of all the clauses combined). When you come to a choice point, choose the first unbound symbols in alphabetical order, and try the assignment to TRUE before the assignment to FALSE.

Solution to Problem 2

In this problem, we use the clauses drived in Problem 1. As asked, we used them as a set to run **Davis-Putnam algorithm**.

S0 = {
$C1 : \neg A \vee \neg B \vee \neg C.$
$C2 : \neg C \vee \neg D.$
$C3 : \neg C \vee \neg E.$
$C4 : D \vee E \vee C.$
$C5 : \neg D \vee B.$
$C6 : \neg D \vee \neg E \vee \neg B.$
$C7 : \neg D \vee \neg E \vee A.$
$C8 : \neg D \vee E.$
$C9 : \neg E \vee D.$
}

> B = {}
>
> **No easy cases in S0**
> Try **A = True**
> Delete clause: **C7**
> Delete literal **A** from **C1**
>
> Add {A-T} to {B}

S1 = {
$C1 : \neg B \vee \neg C.$
$C2 : \neg C \vee \neg D.$
$C3 : \neg C \vee \neg E.$
$C4 : D \vee E \vee C.$
$C5 : \neg D \vee B.$
$C6 : \neg D \vee \neg E \vee \neg B.$
$C8 : \neg D \vee E.$
$C9 : \neg E \vee D.$
}

> **No easy cases in S1**
> Try **B = True**
> Delete clause: **C5**
> Delete literal **B** from **C1, C6**
> Add {B-T} to {B}
>
> B = {A-T, B-T}

S2 = {
$C1 : \neg C.$
$C2 : \neg C \vee \neg D.$
$C3 : \neg C \vee \neg E.$
$C4 : D \vee E \vee C.$
$C6 : \neg D \vee \neg E.$
$C8 : \neg D \vee E.$
$C9 : \neg E \vee D.$
}

Singleton clause: C1
Set **C = False**
Delete clauses: **C1, C2, C3**
Delete literal **C** from **C4**
Add {C-F} to {B}

B = {A-T, B-T, C-F}

S2 = {
$C4 : D \vee E$
$C6 : \neg D \vee \neg E.$
$C8 : \neg D \vee E.$
$C9 : \neg E \vee D.$
}

No easy cases in S2
Try **D = True**
Delete Clause: **C4, C9**
Delete literal **D** from **C6, C8**
Add {D-T} to {B}

B = {A-T, B-T, C-F, D-T}

S3 = {
$C6 : \neg E.$
$C8 : E.$
}

C6 is Singleton clause
Set **E = False**
Delete clauses: **C6**
Delete literal **E** from **C8**
C8 is null clause.
Return Fail.

Go back to **S2**

S2 = {
$C4 : D \lor E$
$C6 : \neg D \lor \neg E.$
$C8 : \neg D \lor E.$
$C9 : \neg E \lor D.$
}

No easy cases in S2.
Try **D = False**
Delete Clause: **C6, C8**
Delete literal **D** from **C4, C9**
Add {D-F} to {B}

B = {A-T, B-T, C-F, D-F}

S4 = {
$C4 : E.$
$C9 : \neg E.$
}

C4 is a Singleton clause
Set **E = True**
Delete clauses: **C4**
Delete literal **E** from **C9**
C9 is null clause.
Return Fail.

Go to **S1**

S1 = {
$C1 : \neg B \lor \neg C.$
$C2 : \neg C \lor \neg D.$
$C3 : \neg C \lor \neg E.$
$C4 : D \lor E \lor C.$
$C5 : \neg D \lor B.$

$C6 : \neg D \vee \neg E \vee \neg B.$
$C8 : \neg D \vee E.$
$C9 : \neg E \vee D.$
}

No easy cases in S1
Try **B = False**
Delete clauses: **C1, C6**
Delete literal **B** from **C5**
Add {B-F} to {B}

B = {A-T, B-F}

S5 = {
$C2 : \neg C \vee \neg D.$
$C3 : \neg C \vee \neg E.$
$C4 : D \vee E \vee C.$
$C5 : \neg D.$
$C8 : \neg D \vee E.$
$C9 : \neg E \vee D.$
}

C5 is a singleton clause
Set **D = False**
Delete clauses: **C2, C5, C8**
Delete literal **D** from **C4, C9**
Add {D-F} to {B}

B = {A-T, B-F, D-F}

S5 = {
$C3 : \neg C \vee \neg E.$
$C4 : E \vee C.$
$C9 : \neg E.$
}

C9 is a singleton clause
Set **E = False**
Delete clauses: **C3, C9**
Delete literal **E** from **C4**
Add {E-F} to {B}

B = {A-T, B-F, D-F, E-F}

S5 = {
$C4 : C.$
}

C4 is a singleton clause
Set **C = True**
Delete clauses: **C4**
Add {C-T} to {B}

B = {A-T, B-F, D-F, E-F, C-T}

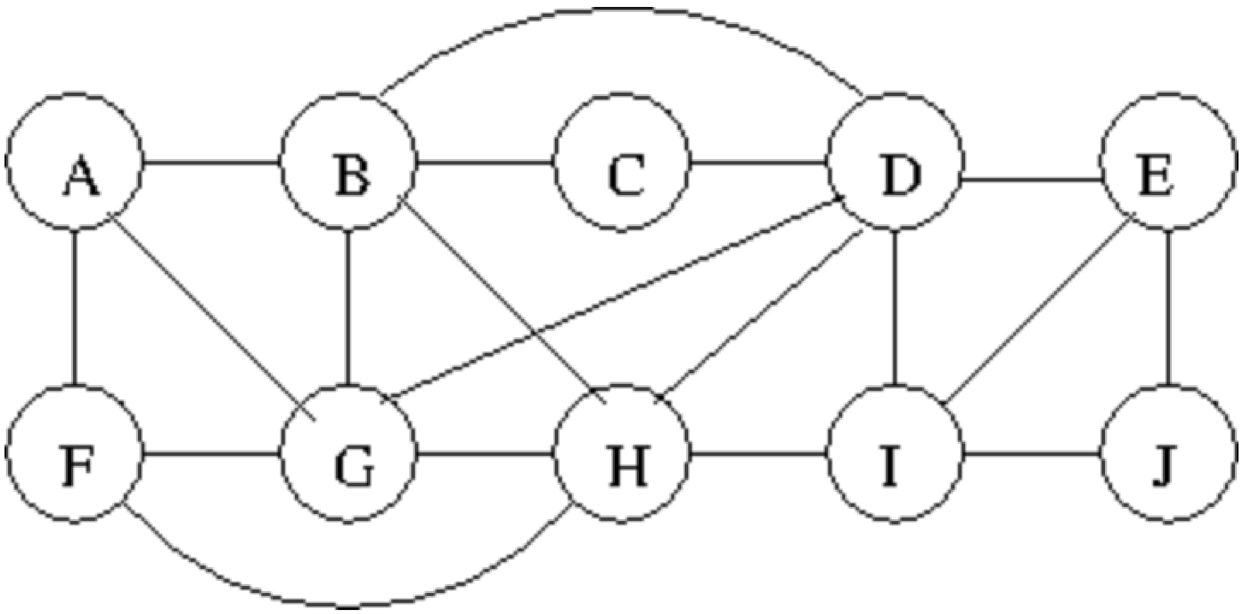
S5 is now an empty set! Success.

Given the above work, it can be seen that the solution to Problem 2 using **Davis-Putnam algorithm** on clauses drived from problem 1 is:
{A-True, B-False, C-True, D-False, E-False}.

Check the solution again S0 for verification:
S0 = {
$C1 : \neg A \vee \neg B \vee \neg C.$
$C2 : \neg C \vee \neg D.$
$C3 : \neg C \vee \neg E.$
$C4 : D \vee E \vee C.$
$C5 : \neg D \vee B.$
$C6 : \neg D \vee \neg E \vee \neg B.$
$C7 : \neg D \vee \neg E \vee A.$
$C8 : \neg D \vee E.$
$C9 : \neg E \vee D.$
}

Problem 3

The following problem is known as the K-CLIQUE problem. You are given an undirected graph G with V vertices, and a number K V. The goal is to find a set of K vertices all of which are directly connected to each other in G. For instance, if the graph is the one shown below and K=4, then a solution is B,D,G,H because B-D, B-G, B-H, D-G, D-H and G-H are all edges in the graph.



A K-CLIQUE problem can be translated into a SAT problem as follows:

The atoms have the form VX where V is a vertex and X is an index between 1 and K. Thus, in this problem, the atoms would be A1, A2, A3, A4, B1, B2, B3, B4 .... J1, J2, J3, J4.

For some list of the vertices that corresponds to a solution, the vertex VX is true if V is at position X of the list. For instance the list [D,H,G,B] is a solution to the 4-CLIQUE problem in our example; that would be represented by having atoms D1, H2, G3, B4 be TRUE and the remaining 36 atoms be false.

You then have the following kinds of constraints:

1. For each index X, there is at least one vertex at position X
2. No vertex is at more than one position in the list
3. No position in the list is occupied by two vertices
4. If two vertices are not connected in the graph, then they are not both in the list, at any positions

Explain how each of these kinds of constraints can be expressed as a set of sentences in propositional logic. Note that you may need a lot of sentences in some cases, or some fairly long sentences, but you should not have exponentially many sentences or sentences that are exponentially long. Illustrate each of the kinds of sentences with two specific sentences from the above example.

Solution to Problem 3

1. For each index X, there is at least one vertex at position X.
    (a) In this case, for every index, i, a clause need to have assignment of every vertex to it.
    (b) Since the question constraint says "at least one vertex at position X," each index will have at least one vertex assigned to it.
    (c) Pseudocode:
    for **i** 1 to **k**:
    for each **vertex**:
    Clause[i] = vertex + i.
    i. Example:

    $$A1 \vee B1 \vee C1 \vee D1 \vee E1 \vee F1 \vee G1 \vee H1 \vee I1 \vee J1$$
    ...
    $$A4 \vee B4 \vee C4 \vee D4 \vee E4 \vee F4 \vee G4 \vee H4 \vee I4 \vee J4$$

2. No vertex is at more than one position in the list.
    (a) If a vertex is occupied with an index, other indexes cannot occupy the vertex.
    (b) Hence, for vertex **H**, and indices, $\mathbf{i}, \mathbf{j}$, the CNF form would be $\neg(H_i \wedge H_j)$.
    (c) By distributing $(\neg)$, it becomes: $(\neg H_i \vee \neg H_j)$.
        i. Example:

        $$\neg B2 \vee \neg B3$$
        $$\neg C1 \vee \neg C3$$

3. No position in the list is occupied by two vertices.
    (a) If an index is occupied, no other vertex can take that index.
    (b) For for vertices, v, u, and index, i, we have following form: $\neg(u_i \wedge v_i)$
    (c) The CNF form $\neg(u_i \wedge v_i)$ can be distributed over $(\neg)$ and produces: $\neg u_i \vee \neg v_i$
        i. Example:

        $$\neg C3 \vee \neg A3$$
        $$\neg B3 \vee \neg J3$$

4. If two vertices are not connected in the graph, then they are not both in the list, at any positions.

(a) If there is no edge between two vertices in a graph, they fail to become part of K-Clique solution.

(b) Thus, non-connected vertices do not qualify to be part of the solution.

(c) In CNF form, for indices $\mathbf{i}, \mathbf{j}$ and vertices, $\mathbf{u}, \mathbf{v}$, we have: $\neg(u_i \wedge v_j)$.

(d) By distributing $(\neg)$, we have: $\neg u_i \vee \neg v_j$

i. Examples:

A. B and I are not connected by an edge, hence:

$$\neg B1 \vee \neg I2$$
$$\neg B1 \vee \neg I3$$
$$\dots$$
$$\neg B4 \vee \neg I3$$

B. E and F are not connected by an edge, hence:

$$\neg E1 \vee \neg F2$$
$$\neg E1 \vee \neg F3$$
$$\dots$$
$$\neg E4 \vee \neg F3$$

End of Assignment. Thank you!