

Assignment 3 - Report

Nakkina Vinay (B21AI023)

Part-B

1. Input 1

a. Giving input to uninformed search algorithm with $m=0$

```
4 0
P|(Q&(R>T))
P>R
Q>T
Q>(R=T)
R
Result: 1
Number of Nodes Explored: 1
```

First line inputs the value $n=4$ and $m=0$

The next four lines are the formulas with one formula per line

The fifth line contains the query that needs to be proved

The last line is the output after the evaluation. Since the value of $m=0$ only the output value is printed and the total number of nodes explored are 1

b. Giving input to uninformed search algorithm with $m=1$

```

4 1
P|(Q&(R>T))
P>R
Q>T
Q>(R=T)
R
Clauses <- The set of clauses in the CNF representation of (KB & !Q)
Clauses: ['(P|Q)', '(P|!R|T)', '(!P|R)', '(!Q|T)', '(!Q|!R|T)', '(!Q|!T|R)', '(!R)']
New Clauses <- {}
For each pair of clauses C_i, C_j in Clauses do:
    (P, Q, !R, T) <- RESOLVE((P, Q), (P, !R, T))
    New Clauses <- New Clauses U Resolvents
    (Q, R) <- RESOLVE((P, Q), (!P, R))
    New Clauses <- New Clauses U Resolvents
    (P, T) <- RESOLVE((P, Q), (!Q, T))
    New Clauses <- New Clauses U Resolvents
    (P, !R, T) <- RESOLVE((P, Q), (!Q, !R, T))
    New Clauses <- New Clauses U Resolvents
    (P, !T, R) <- RESOLVE((P, Q), (!Q, !T, R))
    New Clauses <- New Clauses U Resolvents
    (P, Q, !R) <- RESOLVE((P, Q), (!R))
    New Clauses <- New Clauses U Resolvents
    (T) <- RESOLVE((P, !R, T), (!P, R))
    New Clauses <- New Clauses U Resolvents
    (P, !R, T, !Q) <- RESOLVE((P, !R, T), (!Q, T))
    New Clauses <- New Clauses U Resolvents
    (P, !R, T, !Q) <- RESOLVE((P, !R, T), (!Q, !R, T))
    New Clauses <- New Clauses U Resolvents
    (P, !Q) <- RESOLVE((P, !R, T), (!Q, !T, R))
    New Clauses <- New Clauses U Resolvents
    (P, !R, T) <- RESOLVE((P, !R, T), (!R))
    New Clauses <- New Clauses U Resolvents
    (!R, T) <- RESOLVE((P, Q, !R, T), (!P, !Q, T))
    New Clauses <- New Clauses U Resolvents
    () <- RESOLVE((P, Q, !R, T), (!P, R, !Q, !T))
    If Resolvents contains the empty clause: Return True.
Result: 1
Number of Nodes Explored: 1

```

Since the value of $m=1$, the resolution steps are printed. Since the output is long, I just attached some snippets of the output. In the last line the output value is printed which is 1 and the total number of nodes explored are 1.

c. Giving input to greedy search algorithm with $m=0$

```

4 0
P|(Q&(R>T))
P>R
Q>T
Q>(R=T)
R
Result: 1
Total Nodes Explored: 249

```

First line inputs the value n=4 and m=0

The next four lines are the formulas with one formula per line

The fifth line contains the query that needs to be proved

The last line is the output after the evaluation. Since the value of m=0 only the output value is printed and the total number of nodes explored are 249

d. Giving input to greedy search algorithm with m=1

```

4 1
P|(Q&(R>T))
P>R
Q>T
Q>(R=T)
R
Clauses <- The set of clauses in the CNF representation of (KB & !Q)
Clauses: ['(P|Q)', '(P|!R|T)', '(!P|R)', '(!Q|T)', '(!Q|!R|T)', '(!Q|!T|R)', '(!R)']
New Clauses <- {}
For each pair of clauses C_i, C_j in Clauses do:
  (P, Q, !R, T) <- RESOLVE((P, Q), (P, !R, T))
  New Clauses <- New Clauses U Resolvents
  (Q, R) <- RESOLVE((P, Q), (!P, R))
  New Clauses <- New Clauses U Resolvents
  (P, T) <- RESOLVE((P, Q), (!Q, T))
  New Clauses <- New Clauses U Resolvents
  (P, !R, T) <- RESOLVE((P, Q), (!Q, !R, T))
  New Clauses <- New Clauses U Resolvents
  (P, !T, R) <- RESOLVE((P, Q), (!Q, !T, R))
  New Clauses <- New Clauses U Resolvents
  (P, Q, !R) <- RESOLVE((P, Q), (!R))
  New Clauses <- New Clauses U Resolvents
  (T) <- RESOLVE((P, !R, T), (!P, R))
  New Clauses <- New Clauses U Resolvents
  (P, !R, T, !Q) <- RESOLVE((P, !R, T), (!Q, T))
  New Clauses <- New Clauses U Resolvents
  (P, !R, T, !Q) <- RESOLVE((P, !R, T), (!Q, !R, T))
  New Clauses <- New Clauses U Resolvents
  (P, !Q) <- RESOLVE((P, !R, T), (!Q, !T, R))
  New Clauses <- New Clauses U Resolvents
  (P, !R, T) <- RESOLVE((P, !R, T), (!R))

```

```

(T) <- RESOLVE((P, Q, !R, T), (!P, R, !Q, T))
New Clauses <- New Clauses U Resolvents
() <- RESOLVE((P, Q, !R, T), (!P, R, !Q, !T))
If Resolvents contains the empty clause: Return True.
Result: 1
Total Nodes Explored: 249

```

Since the value of $m=1$, the resolution steps are printed. Since the output is long, I just attached some snippets of the output. In the last line the output value is printed which is 1 and the total number of nodes explored are 249.

We can see that for Input 1 the total number of nodes explored are more for the greedy algorithm with heuristic than the uninformed search. Execution time is also more for greedy algorithm

2. Input 2

a. Giving input to uninformed search algorithm with $m=0$

```

3 0
M>I
!M>A
(I|A)>H
!H
Result: 0
Number of Nodes Explored: 35

```

b. Giving input to greedy search algorithm with $m=0$

```

3 0
M>I
!M>A
(I|A)>H
!H
Result: 0
Total Nodes Explored: 962

```

The number of nodes explored may vary but the result will be the same for both $m=0$ and $m=1$, the resolution steps will be printed when $m=1$.

We can see that for Input 2 the total number of nodes explored are more for the greedy algorithm with heuristic than the uninformed search. Execution time is also more for greedy algorithm

3. Input 3

a. Giving input to uninformed search algorithm with m=0

```
7 0
(S&W)>E
(W&P)>H
R>!H
R>G
W
R
S
E
Result: 1
Number of Nodes Explored: 1
```

b. Giving input to greedy search algorithm with m=0

```
7 0
(S&W)>E
(W&P)>H
R>!H
R>G
W
R
S
E
Result: 1
Total Nodes Explored: 342
```

We can see that for Input 3 also the total number of nodes explored are more for the greedy algorithm with heuristic than the uninformed search. Execution time is also more for greedy algorithm

4. Input 4

a. Giving input to uninformed search algorithm with $m=0$

```
6 0
A>B
!B
!A>(C|D)
C>E
F>!E
F
D
Result: 1
Number of Nodes Explored: 1
```

b. Giving input to greedy search algorithm with $m=0$

```
6 0
A>B
!B
!A>(C|D)
C>E
F>!E
F
D
Result: 1
Total Nodes Explored: 4026
```

We can see that for Input 4 also the total number of nodes explored are more for the greedy algorithm with heuristic than the uninformed search. Execution time is also more for greedy algorithm

5. Input 5

a. Giving input to uninformed search algorithm with $m=1$

```

3 1
(P>Q)>Q
(P>P)>R
(R>S)>!(S>Q)
R
Clauses <- The set of clauses in the CNF representation of (KB & !Q)
Clauses: ['(P|Q)', '(!Q|Q)', '(P|R)', '(!P|R)', '(R|S)', '(!S|S)', '(R|!Q)', '(!S|!Q)', '(!R)']
New Clauses <- {}
For each pair of clauses C_i, C_j in Clauses do:
    (P, Q) <- RESOLVE((P, Q), (Q))
    New Clauses <- New Clauses U Resolvents
    (P, Q, R) <- RESOLVE((P, Q), (P, R))
    New Clauses <- New Clauses U Resolvents
    (Q, R) <- RESOLVE((P, Q), (!P, R))
    New Clauses <- New Clauses U Resolvents
    (P, Q, R, S) <- RESOLVE((P, Q), (R, S))
    New Clauses <- New Clauses U Resolvents
    (P, Q, S) <- RESOLVE((P, Q), (S))
    New Clauses <- New Clauses U Resolvents
    (P, R) <- RESOLVE((P, Q), (R, !Q))
    New Clauses <- New Clauses U Resolvents
    (P, !S) <- RESOLVE((P, Q), (!S, !Q))
    (R, S) <- RESOLVE((Q), (R, S, !Q))
    New Clauses <- New Clauses U Resolvents
    () <- RESOLVE((Q), (!Q))
    If Resolvents contains the empty clause: Return True.
Result: 1
Number of Nodes Explored: 1

```

b. Giving input to greedy search algorithm with m=1

```

3 1
(P>Q)>Q
(P>P)>R
(R>S)>!(S>Q)
R
Clauses <- The set of clauses in the CNF representation of (KB & !Q)
Clauses: ['(P|Q)', '(!Q|Q)', '(P|R)', '(!P|R)', '(R|S)', '(!S|S)', '(R|!Q)', '(!S|!Q)', '(!R)']
New Clauses <- {}
For each pair of clauses C_i, C_j in Clauses do:
    (P, Q) <- RESOLVE((P, Q), (Q))
    New Clauses <- New Clauses U Resolvents
    (P, Q, R) <- RESOLVE((P, Q), (P, R))
    New Clauses <- New Clauses U Resolvents
    (Q, R) <- RESOLVE((P, Q), (!P, R))
    New Clauses <- New Clauses U Resolvents
    (P, Q, R, S) <- RESOLVE((P, Q), (R, S))
    New Clauses <- New Clauses U Resolvents
    (P, Q, S) <- RESOLVE((P, Q), (S))
    New Clauses <- New Clauses U Resolvents
    (P, R) <- RESOLVE((P, Q), (R, !Q))
    New Clauses <- New Clauses U Resolvents
    (P, !S) <- RESOLVE((P, Q), (!S, !Q))

    New Clauses <- New Clauses U Resolvents
    () <- RESOLVE((Q), (!Q))
    If Resolvents contains the empty clause: Return True.
Result: 1
Total Nodes Explored: 81

```

After running the two implementations on inputs of different sizes and complexities, we can analyze that the total number of nodes explored and execution time are more for the resolution-refutation algorithm as a greedy search algorithm than the uninformed search problem.