Problem 1

Follow alpha-beta strategy for the min-max tree below. Show which nodes the strategy will not visit (place cats).

Problem 2.

Follow RS method to check if \((\neg a \rightarrow b) \rightarrow ((\neg b \rightarrow c) \rightarrow (\neg c \rightarrow \neg a))\) is a tautology.

**Solution** (Not Tautology).

\[
(\neg a \rightarrow b) \rightarrow ((\neg b \rightarrow c) \rightarrow (\neg c \rightarrow \neg a))
\]

\[
\neg (\neg a \rightarrow b), ((\neg b \rightarrow c) \rightarrow (\neg c \rightarrow \neg a))
\]

\[
\neg a, ((\neg b \rightarrow c) \rightarrow (\neg c \rightarrow \neg a)) \lor \neg b, ((\neg b \rightarrow c) \rightarrow (\neg c \rightarrow \neg a))
\]

\[
\neg a, \neg a, ((\neg b \rightarrow c) \rightarrow (\neg c \rightarrow \neg a)) \lor \neg b, \neg b, ((\neg b \rightarrow c) \rightarrow (\neg c \rightarrow \neg a))
\]

\[
\neg a, \neg b, (\neg c \rightarrow \neg a); \neg a, \neg c, (\neg c \rightarrow \neg a); \neg b, \neg b, (\neg c \rightarrow \neg a); \neg b, \neg c, (\neg c \rightarrow \neg a)
\]
**Problem 3.**

In what order the nodes of the graph GR will be visited by the shortest path algorithm using heuristic distance measure \( f(n) \) from the start node A to the goal node K passing through a node \( n \) in GR where \( f(n) \) is defined as \( f(n) = g(n) + h(n) \).

We assume that:

1. \( g(n) \) is the shortest distance from the start node A to the node \( n \)
2. \( h(n) \) is a heuristic estimation of the distance from node \( n \) to the final node K.

Check if this algorithm is admissible and monotonic (is it A*-algorithm?).

**Graph GR**

![Graph GR](image)

**Solution.**

Admissible: YES

\[
\begin{align*}
H(A) & \leq d(A,K) = 13, \\
H(B) & \leq d(B,K) = 10, \\
H(G) & \leq d(G,K) = 5, \\
H(L) & \leq d(L,K) = 3, \\
H(C) & \leq d(C,K) = 12, \\
H(D) & \leq d(D,K) = 8, \\
H(M) & \leq d(M,K) = 5, \\
H(E) & \leq d(E,K) = 11, \\
H(F) & \leq d(F,K) = 9
\end{align*}
\]

Monotonic: YES

\[
\begin{align*}
f(B) = 11, \\
f(G) = 13, \\
f(C) = 10, \\
f(D) = 12, \\
f(M) = 12, \\
f(K) = 13, \\
f(E) = 10, \\
f(F) = 11
\end{align*}
\]

Path: A -> C -> E -> F -> B -> D -> M -> K
Problem 4.
Consider the following axioms:
Every child sees some witch.
No witch has both a black cat and a pointed hat.
Every witch is good or bad.
Every child who sees any good witch gets candy.
Every witch that is bad has a black cat.
Prove that: If every witch seen by any child has a pointed hat, then every child gets candy.

Hint: Use the following constants, functors, & predicates.
Constants: BC – black cat, PH – pointed hat
Functors: W(x) – x is a witch, C(x) – x is a child
Predicates: SE(x,y) – x sees y, GETS(x) – x gets candy, GOOD(x) – x is good, BAD(x) – x is bad
HAS(x,y) – x has y

Add 2 statements to your KB: GOOD(x) -> ¬BAD(x), BAD(x) -> ¬GOOD(x).

Solution.
(1) (\forall x)(\exists y)SE(C(x),W(y)).  (2) ¬(\exists x)[HAS(W(x),BC) \land HAS(W(x), PH)],
(2) GOOD(W(x)) \lor BAD(W(x)), (4) [SE(C(x),W(y)) \land GOOD(W(y))] \rightarrow GETS(C(x))
(5) BAD(W(x)) \rightarrow HAS(W(x),BC),
Prove: (\exists x)(\forall y)[SE(C(x),W(y)) \rightarrow HAS(W(y),PH)] \rightarrow (\forall z)GETS(C(z))

KB Construction:
(\forall x)(\exists y)SE(C(x),W(y)) = (\forall x)SE(C(x),W(f(x))) = SE(C(x),W(f(x)))
¬(\exists x)[HAS(W(x),BC) \land HAS(W(x), PH)] = ¬HAS(W(x),BC) \lor ¬HAS(W(x), PH)
GOOD(W(x)) \lor BAD(W(x)) \rightarrow GOOD(W(x)) \lor BAD(W(x))
[SE(C(x),W(y)) \land GOOD(W(y))] \rightarrow GETS(C(x)) = ¬[SE(C(x),W(y)) \land GOOD(W(y))] \lor GETS(C(x)) = ¬SE(C(x),W(y)) \lor ¬GOOD(W(y)) \lor GETS(C(x))
BAD(W(x)) \rightarrow HAS(W(x),BC) = ¬BAD(W(x)) \lor HAS(W(x),BC)

Goal: (\exists x)(\forall y)[SE(C(x),W(y)) \rightarrow HAS(W(y),PH)] \rightarrow (\forall z)GETS(C(z)) =
¬(\exists x)(\forall y)[¬SE(C(x),W(y)) \lor HAS(W(y),PH)] \lor (\forall z)GETS(C(z))
Negation of Goal: (\exists x)(\forall y)[¬SE(C(x),W(y)) \lor HAS(W(y),PH)] \land (\exists z)¬GETS(C(z)) =
[¬SE(C(A),W(y)) \lor HAS(W(y),PH)] \land ¬GETS(C(B))

FINAL KB with Negation of the goal:
(1) SE(C(x),W(f(x)))
(2) ¬HAS(W(x),BC) \lor ¬HAS(W(x), PH)
(3) GOOD(W(x)) \lor BAD(W(x))
(4) ¬GOOD(W(x)) \lor ¬BAD(W(x))
(5) ¬SE(C(x),W(y)) \lor ¬GOOD(W(y)) \lor GETS(C(x))
(6) ¬BAD(W(x)) \lor HAS(W(x),BC)
(7) ¬SE(C(A),W(y)) \lor HAS(W(y),PH)
(8) ¬GETS(C(B))
(9) = (5)+(8)+(1): ¬GOOD(W(f(x)))
(10) = (6)+(3): HAS(W(x),BC) \lor GOOD(W(x))
(11) = (9)+(10): HAS(W(f(x)),BC)
(12) = (1)+(7) = HAS(W(f(A)),PH)
(2)+(11)+(12) = NIL