VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

B.Tech. Winter Semester 2023-24
School Of Computer Science and Engineering
(SCOPE)

# Digital Assignment - I

## Artficial Intelligence

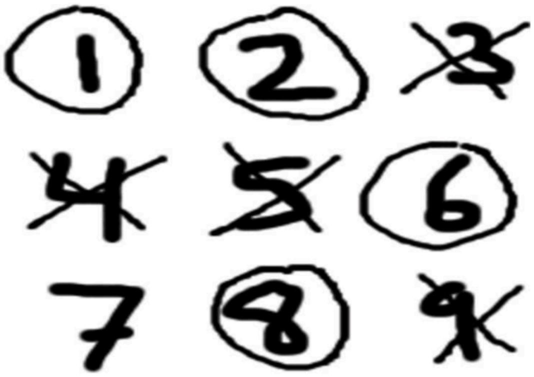
## Apurva Mishra, 22BCE2791

10 October 2024

# 1. Questions

**Problem 1.1.**
The 15 Game: To play the 15 game, write the numbers 1 through 9 in a square. Player one is X and player two is O. Player one starts by drawing an X through any unchosen number. Then player two circles an unchosen number. The game is won when one player has collected three numbers which add up to 15. In the sample game, player two has won because $8 + 6 + 1 = 15$. Note that having 9 and 6 alone would not be a win even though $9 + 6 = 15$, because you need three numbers which sum to 15.

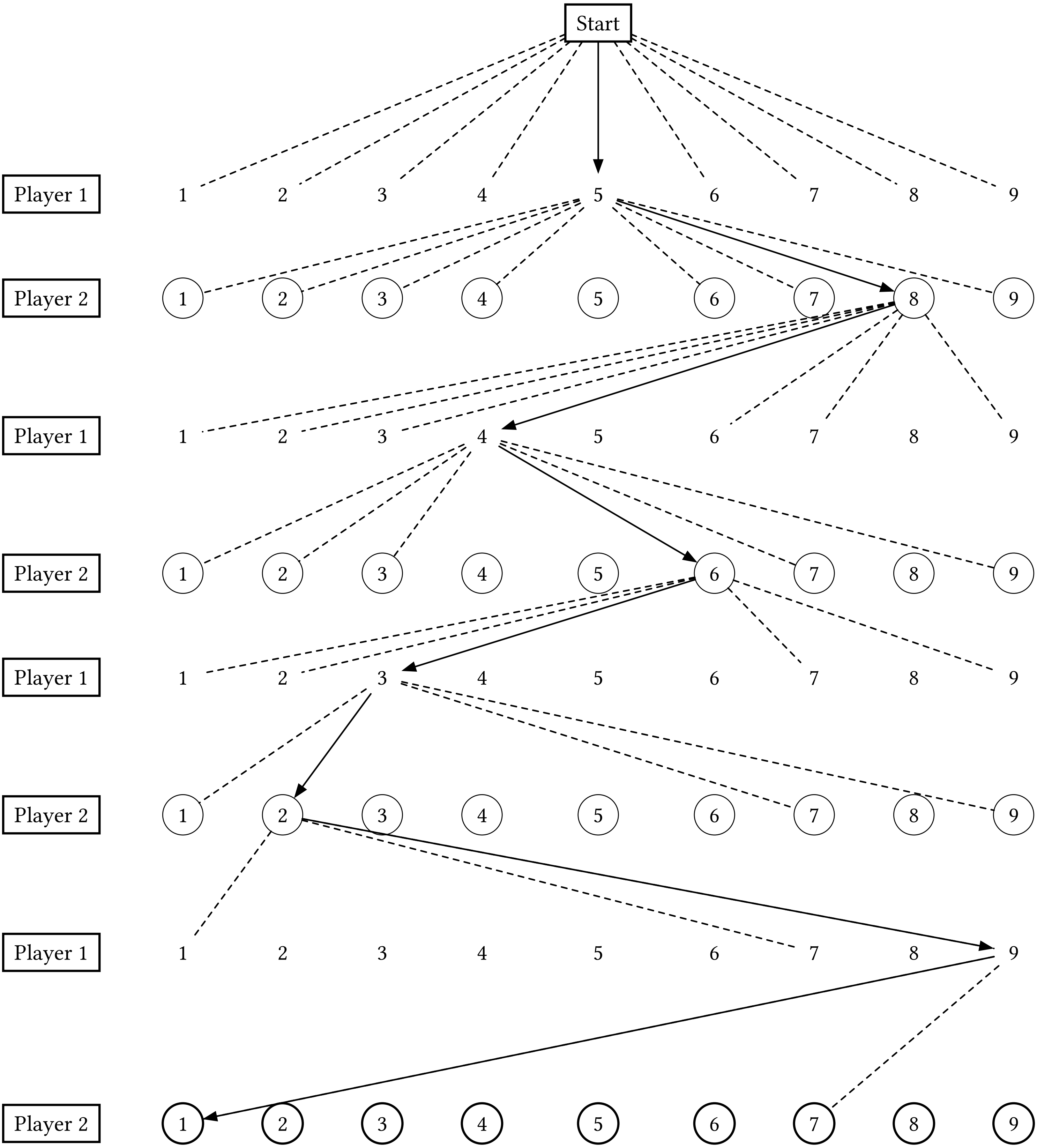
### Q1. Draw the game tree for this game.



The 15 Game

**Following is one of the several game trees which are possible for the given snapshot of the game**

Dashed Lines represent possible moves which iteratively decreases as the game continues. Line arrows represent played move



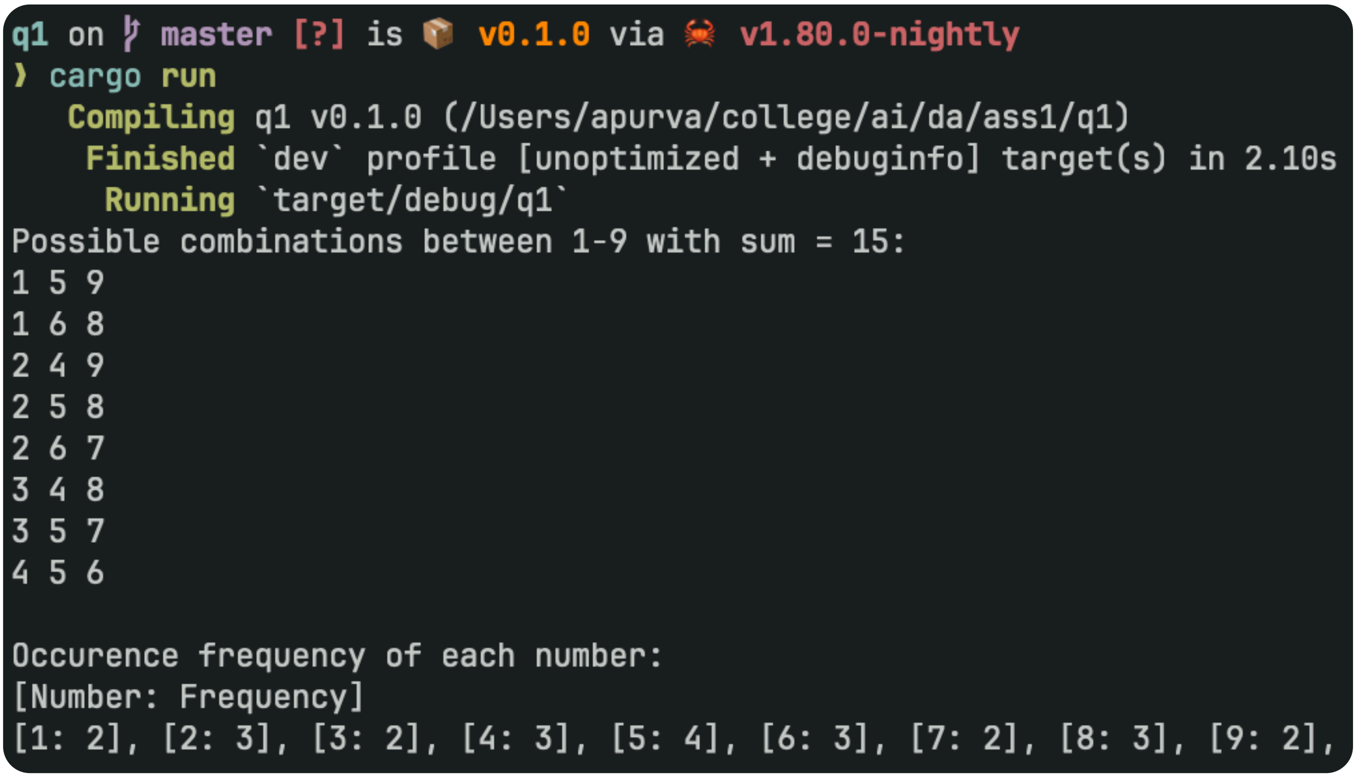
**Player 2 Won!**

## Q2. What is the first best move for player one?

To calculate the best move, we find all the possible combination of numbers which sum to 15

**Code to find solution combinations**

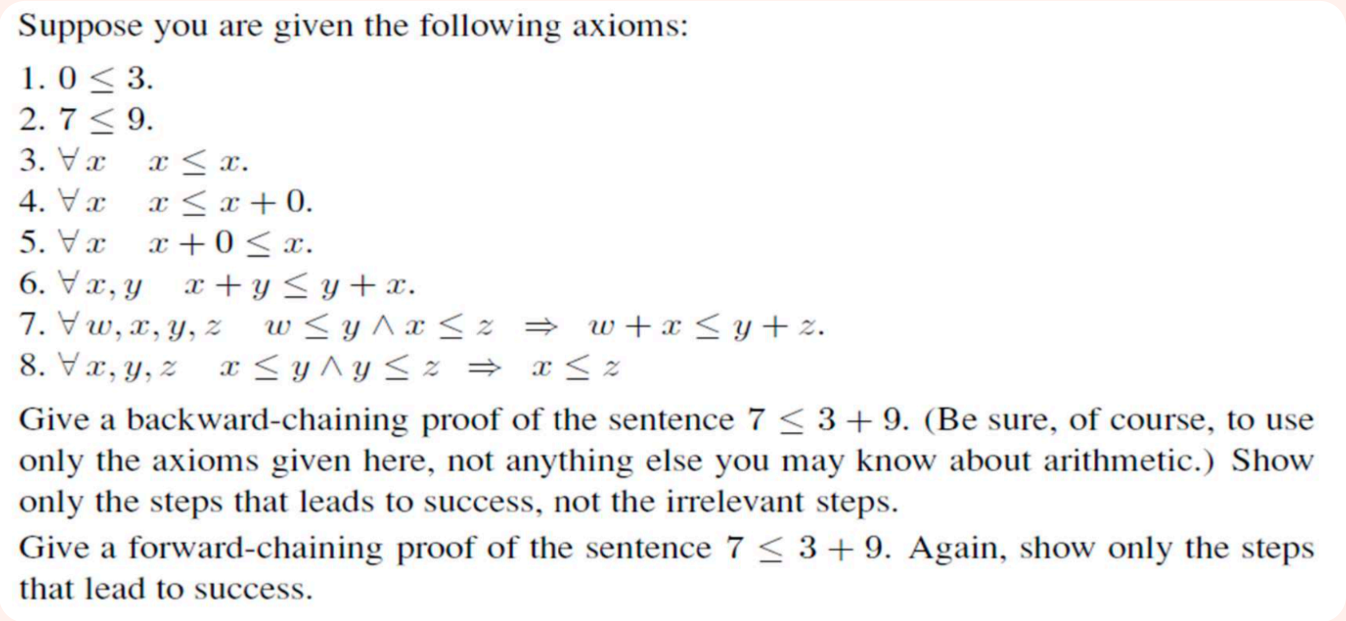
```rust
//Programming Language: Rust
use std::vec::Vec;

fn dup(arr: Vec<u16>) -> bool {
    for i in 0..arr.len() {
        for j in 0..arr.len() {
            if i != j && arr[i] == arr[j] {
                return true;
            }
        }
    }

    false
}

fn rec(i: u16, mut arr: Vec<u16>, cap: &mut Vec<u16>) {
    let sum: u16 = arr.iter().sum();
    if sum == 15 && arr.len() == 3 && !dup(arr.clone()) {
        for i in arr {
            cap[(i - 1) as usize] += 1;
            print!("{i} ");
        }
        println!();

        return;
    }

    if arr.len() == 3 {
        return;
    }

    for j in i..=9 {
        arr.push(j);
        rec(j, arr.clone(), cap);
        arr.pop();
    }
}

fn main() {
    let mut arr = Vec::with_capacity(3);
    let mut cap: Vec<u16> = Vec::with_capacity(9);

    for _ in 0..9 {
        cap.push(0);
    }

    println!("Possible combinations between 1-9 with sum = 15: ");
    for j in 1..=9 {
        arr.push(j);
        rec(j, arr.clone(), &mut cap);
        arr.pop();
    }

    println!();
    println!("Occurence frequency of each number: ");
    println!("[Number: Frequency]");

    for (i, v) in cap.iter().enumerate() {
        print!("[{}: {v}], ", i + 1);
    }
}
```



```
q1 on  master [?] is 📦 v0.1.0 via 🦀 v1.80.0-nightly
❯ cargo run
   Compiling q1 v0.1.0 (/Users/apurva/college/ai/da/ass1/q1)
    Finished `dev` profile [unoptimized + debuginfo] target(s) in 2.10s
     Running `target/debug/q1`
Possible combinations between 1-9 with sum = 15:
1 5 9
1 6 8
2 4 9
2 5 8
2 6 7
3 4 8
3 5 7
4 5 6

Occurence frequency of each number:
[Number: Frequency]
[1: 2], [2: 3], [3: 2], [4: 3], [5: 4], [6: 3], [7: 2], [8: 3], [9: 2],
```

From here we can observe that 5 has highest frequency in probable solution combinations. Therefore 5 should be the best first move.

**Problem 1.2.**
Solve the following problem



Suppose you are given the following axioms:

1. $0 \leq 3$.
2. $7 \leq 9$.
3. $\forall x \quad x \leq x$.
4. $\forall x \quad x \leq x + 0$.
5. $\forall x \quad x + 0 \leq x$.
6. $\forall x, y \quad x + y \leq y + x$.
7. $\forall w, x, y, z \quad w \leq y \wedge x \leq z \;\Rightarrow\; w + x \leq y + z$.
8. $\forall x, y, z \quad x \leq y \wedge y \leq z \;\Rightarrow\; x \leq z$

Give a backward-chaining proof of the sentence $7 \leq 3 + 9$. (Be sure, of course, to use only the axioms given here, not anything else you may know about arithmetic.) Show only the steps that leads to success, not the irrelevant steps.

Give a forward-chaining proof of the sentence $7 \leq 3 + 9$. Again, show only the steps that lead to success.

## Backward Chaining

### Step 1: Conlusion

$7 \leqslant 3 + 9$

### Step 2: Rules

$$\begin{tikzcd}
7 \leqslant 3 + 9 \arrow[d] \\
\forall w, x, y, z \ (w \leqslant y \wedge x \leqslant z \Rightarrow w + x \leqslant y + z)
\end{tikzcd}$$

### Step 3: Known Facts

$$\begin{tikzcd}
& 7 \leqslant 3 + 9 \arrow[d] & \\
& \forall w, x, y, z \ (w \leqslant y \land x \leqslant z \Rightarrow w + x \leqslant y + z) \arrow[dl] \arrow[dr] & \\
7 \leqslant 9 & & 0 \leqslant 3
\end{tikzcd}$$

Hence Proved!

## Forward Chaining

### Step 1: Known Facts

$7 \leqslant 9 \qquad 0 \leqslant 3$

### Step 2: Rules that satisfy the facts

$$\begin{tikzcd}
7 \leqslant 9 \arrow[dr] & & 0 \leqslant 3 \arrow[dl] \\
& \forall w, x, y, z \ (w \leqslant y \land x \leqslant z \Rightarrow w + x \leqslant y + z) &
\end{tikzcd}$$

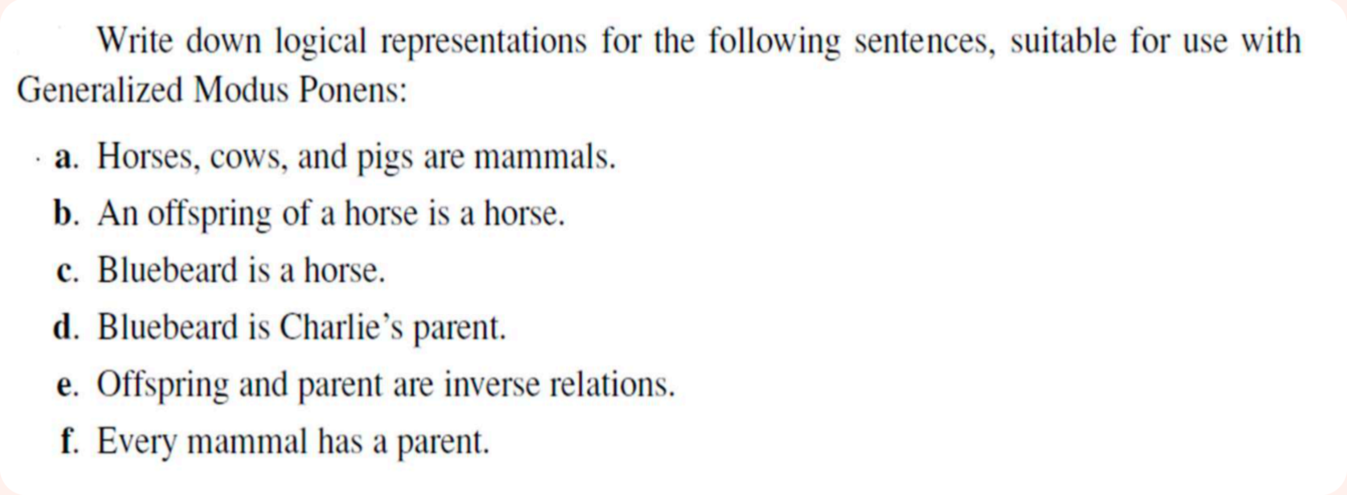
### Step 3: Conlusion

$$\begin{tikzcd}
7 \leqslant 9 \arrow[dr] & & 0 \leqslant 3 \arrow[dl] \\
& \forall w, x, y, z \ (w \leqslant y \land x \leqslant z \Rightarrow w + x \leqslant y + z) \arrow[d] & \\
& 7 \leqslant 3 + 9 &
\end{tikzcd}$$

Hence Proved!

**Problem 1.3.**

Generative AI (GenAI) is a subfield of Artificial Intelligence (AI). It focuses on algorithms that can generate new content, such as text, images, audio, and video, that resembles human- created content. Two such popular tools are: a) Microsoft's Copilot and b) OpenAI's ChatGPT. Use Copilot and ChatGPT to solve the following question. Check the correctness of the answers generated by these tools and do a comparative study, which one can make better results.



Write down logical representations for the following sentences, suitable for use with Generalized Modus Ponens:

**a.** Horses, cows, and pigs are mammals.
**b.** An offspring of a horse is a horse.
**c.** Bluebeard is a horse.
**d.** Bluebeard is Charlie's parent.
**e.** Offspring and parent are inverse relations.
**f.** Every mammal has a parent.

**Generated text by two language model for the given problem**

ChatGPT

To represent the given sentences in logical form, suitable for use with Generalized Modus Ponens (GMP), we'll break down each sentence into predicate logic step-by-step.

## 1.1. Symbols:

- Let the predicate ( Mammal(x) ) denote "x is a mammal."
- Let ( Horse(x) ) denote "x is a horse."
- Let ( Cow(x) ) denote "x is a cow."
- Let ( Pig(x) ) denote "x is a pig."
- Let ( Offspring(x, y) ) denote "x is an offspring of y."
- Let ( Parent(x, y) ) denote "x is a parent of y."
- Let ( Bluebeard ) and ( Charlie ) be individual constants.

## 1.2. Logical Representations:

(a) Horses, cows, and pigs are mammals.

This sentence can be broken down into three individual statements:

- ( forall x , (Horse(x) rightarrow Mammal(x)) )
- ( forall x , (Cow(x) rightarrow Mammal(x)) )
- ( forall x , (Pig(x) rightarrow Mammal(x)) )

(b) An offspring of a horse is a horse.

This can be written as:

- ( forall x , forall y , (Offspring(x, y) land Horse(y) rightarrow Horse(x)) )

(c) Bluebeard is a horse.

Since “Bluebeard” is an individual constant, we simply write:

- ( Horse(Bluebeard) )

(d) Bluebeard is Charlie’s parent.

This is written as:

- ( Parent(Bluebeard, Charlie) )

(e) Offspring and parent are inverse relations.

This indicates a bi-conditional relationship between offspring and parent:

- ( forall x , forall y , (Offspring(x, y) leftrightarrow Parent(y, x)) )

(f) Every mammal has a parent.

This is represented as:

- ( forall x , (Mammal(x) rightarrow exists y , Parent(y, x)) )

—

## 1.3. Final Summary:

1. ( forall x , (Horse(x) rightarrow Mammal(x)) )
2. ( forall x , (Cow(x) rightarrow Mammal(x)) )
3. ( forall x , (Pig(x) rightarrow Mammal(x)) )
4. ( forall x , forall y , (Offspring(x, y) land Horse(y) rightarrow Horse(x)) )
5. ( Horse(Bluebeard) )
6. ( Parent(Bluebeard, Charlie) )
7. ( forall x , forall y , (Offspring(x, y) leftrightarrow Parent(y, x)) )
8. ( forall x , (Mammal(x) rightarrow exists y , Parent(y, x)) )

This set of logical formulas can be used with Generalized Modus Ponens to infer new facts from the premises.

Copilot

## 1.Analyzing the Sentences and Representing Them Logically

### Understanding the Problem

We need to express the given sentences in a logical form suitable for Generalized Modus Ponens. This involves using predicates, constants, and quantifiers.

### Defining Predicates and Constants

$M(x)$: x is a mammal $H(x)$: x is a horse $C(x)$: x is a cow $P(x)$: x is a pig $O(x, y)$: x is an offspring of y B: Bluebeard Ch: Charlie

### Representing the Sentences

1. a. Horses, cows, and pigs are mammals. $\forall x(H(x) \rightarrow M(x))$ $\forall x(C(x) \rightarrow M(x))$ $\forall x(P(x) \rightarrow M(x))$
2. b. An offspring of a horse is a horse. $\forall x \forall y(H(y) \wedge O(x, y) \rightarrow H(x))$
3. c. Bluebeard is a horse. $H(B)$
4. d. Bluebeard is Charlie's parent. $O(Ch, B)$
5. e. Offspring and parent are inverse relations. $\forall x \forall y(O(x, y) \leftrightarrow O(y, x))$
6. f. Every mammal has a parent. $\forall x(M(x) \rightarrow \exists y O(x, y))$

### Explanation

Quantifiers: ∀ (for all) and ∃ (there exists) are used to express generality and existence. Predicates: M, H, C, P, and O represent properties or relations. Constants: B and Ch represent specific individuals. Implication (→): This logical connective indicates that one statement implies another. Biconditional (↔): This logical connective indicates that two statements are equivalent.

These logical representations can now be used with Generalized Modus Ponens to derive conclusions from the given premises.

### Comparative Study

The answer generated by ChatGPT provide a suitable logical representation for the given problem.

However in case of Copilot, while most of the representations are correct and often match the answer generated by ChatGPT, its answer (Copilot) has few inconsistencies:

- Bluebeard is Charlie’s parent: $O(Ch, B)$
  This differs from ChatGPT’s answer. It uses the offspring relation instead of the parent relation. While logically correct this is unlikely to be seen in human generated answers.
- Offspring and parent are inverse relations: $\forall x \forall y (O(x, y) \leftrightarrow O(y, x))$
  This is incorrect. It states that the offspring relation is symmetric, which is not true.

Therefore only ChatGPT generated correct and consistent answer.

### My approach after reviewing the generated answer would be:

1. Define predicates:
   - Mammal(x): x is a mammal
   - Horse(x): x is a horse
   - Cow(x): x is a cow
   - Pig(x): x is a pig
   - Offspring(x,y): x is an offspring of y
   - Parent(x,y): x is a parent of y
   - Bluebeard, Charlie: constants
2. Logical representations:

   a. Horses, cows, and pigs are mammals:
      $\forall x(\text{Horse}(x) \rightarrow \text{Mammal}(x))$
      $\forall x(\text{Cow}(x) \rightarrow \text{Mammal}(x))$
      $\forall x(\text{Pig}(x) \rightarrow \text{Mammal}(x))$
   b. An offspring of a horse is a horse:
      $\forall x \forall y(\text{Offspring}(x,y) \wedge \text{Horse}(y) \rightarrow \text{Horse}(x))$
   c. Bluebeard is a horse:
      $\text{Horse}(\text{Bluebeard})$
   d. Bluebeard is Charlie’s parent:
      $\text{Parent}(\text{Bluebeard}, \text{Charlie})$
   e. Offspring and parent are inverse relations:
      $\forall x \forall y(\text{Offspring}(x,y) \leftrightarrow \text{Parent}(y,x))$
   f. Every mammal has a parent:
      $\forall x \exists y\ (\text{Mammal}(x) \rightarrow \text{Parent}(y,x))$