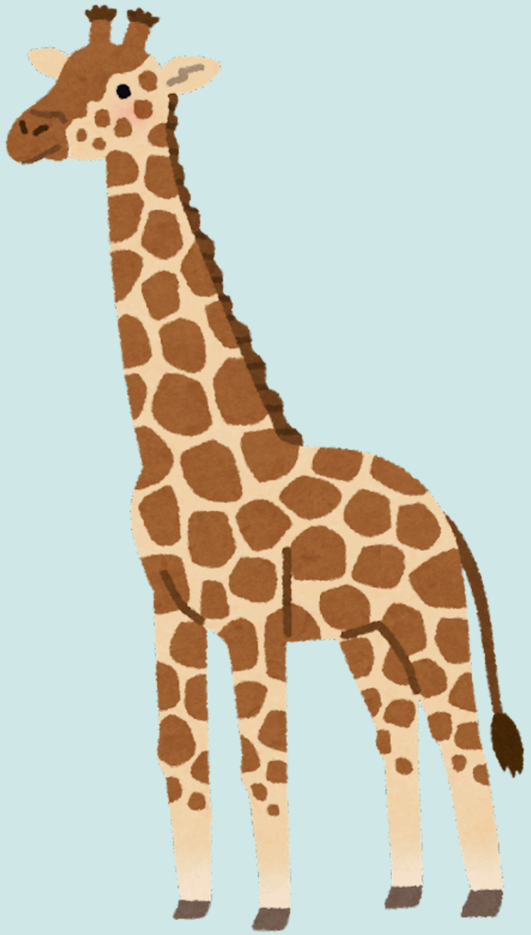
ANIMAL CLASSIFICATION AND CHARACTERISTICS INFERENCE USING PROLOG

COURSE: ARTIFICIAL INTELLIGENCE & EXPERT SYSTEM LAB (CSE 404)
PREPARED BY: ANIKA NAWER NABILA
ID: 22101152
DATE: 12 AUGUST 2025



Problem Statement

Problem:

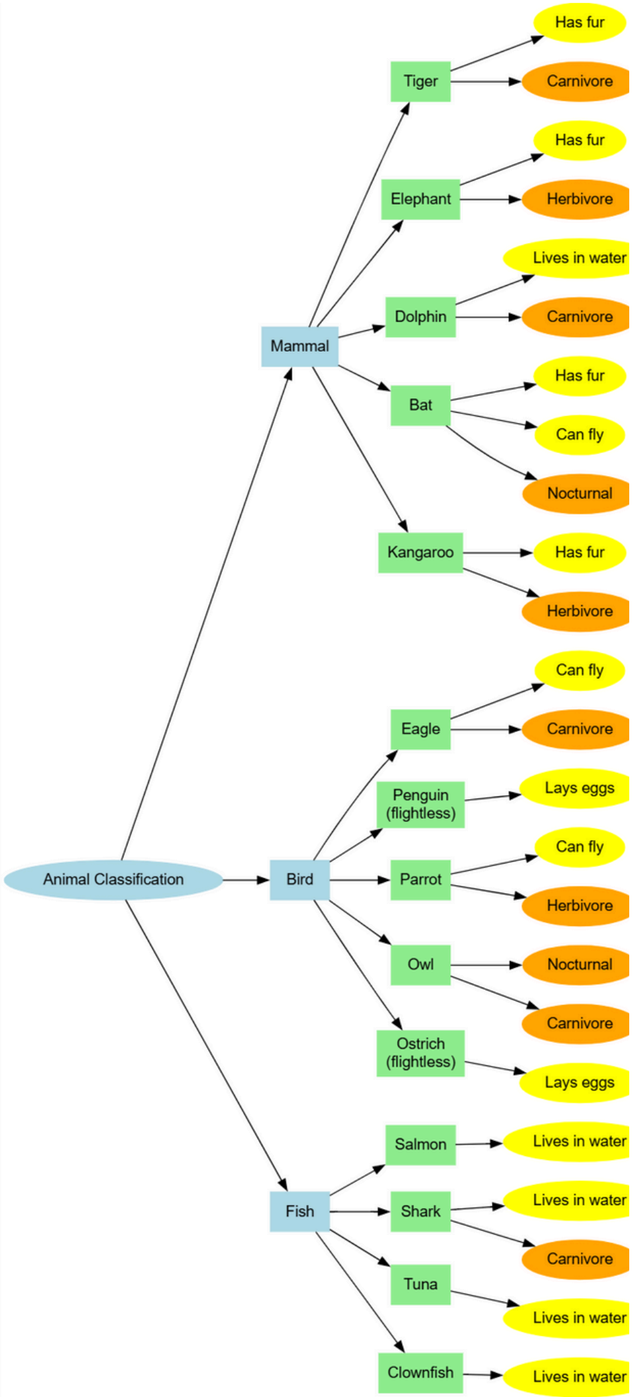
How to classify animals and determine their characteristics automatically based on known facts such as their class (mammal, bird, fish) and traits (can fly, has fur, lives in water, lays eggs)?

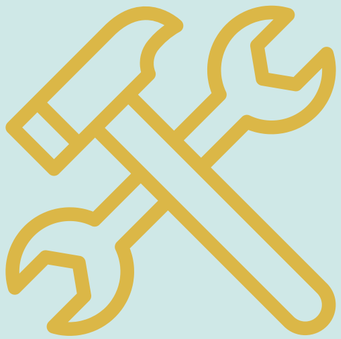
Solution:

- Use Prolog to represent animals as facts categorized by class and traits.
- Define rules to infer additional knowledge, such as whether an animal can fly or if it’s a mammal with fur.
- Enable queries that allow easy identification of animal classes and characteristics, even handling exceptions like penguins that cannot fly despite being birds.
- Build a knowledge-based system that uses logic programming to make classification and characteristic inference efficient and automated.

Diagram


Morris Charts
Sparkline Charts
Easy Pie Charts
25%
50%
75%
100%


Animal Classification
Mammal
Tiger
Has fur
Carnivore
Elephant
Has fur
Herbivore
Dolphin
Lives in water
Carnivore
Bat
Has fur
Can fly
Nocturnal
Kangaroo
Has fur
Herbivore
Bird
Eagle
Can fly
Carnivore
Penguin (flightless)
Lays eggs
Parrot
Can fly
Herbivore
Owl
Nocturnal
Carnivore
Ostrich (flightless)
Lays eggs
Fish
Salmon
Lives in water
Shark
Lives in water
Carnivore
Tuna
Lives in water
Clownfish
Lives in water



Technical Report

For the detailed source code and implementation of the Animal Classification Expert System using Prolog, you can find the complete project repository Here. This repository contains all the facts, rules, and queries used to build the knowledge-based system, demonstrating logical inference for animal classification.

Source Code

```prolog
% ====== Facts: Animal Classifications ======
animal(mammal, tiger).
animal(mammal, elephant).
animal(mammal, dolphin).
animal(mammal, bat).
animal(mammal, kangaroo).

animal(bird, eagle).
animal(bird, penguin).
animal(bird, parrot).
animal(bird, owl).
animal(bird, ostrich).

animal(fish, salmon).
animal(fish, shark).
animal(fish, tuna).
animal(fish, clownfish).

% ====== Facts: Animal Characteristics ======
has_fur(tiger).
has_fur(elephant).
has_fur(bat).
has_fur(kangaroo).
```

```prolog
% ====== Rules ======

% Rule 1: Check if an animal is a mammal
is_mammal(X) :-
    animal(mammal, X).

% Rule 2: Check if an animal can actually fly (exclude flightless birds)
can_really_fly(X) :-
    can_fly(X),
    X \= penguin,
    X \= ostrich.

% Rule 3: Animals that live in water but are not fish
water_animal_not_fish(X) :-
    lives_in_water(X),
    \+ animal(fish, X).

% Rule 4: Animals with fur that are mammals
has_fur_animal(X) :-
    has_fur(X),
    is_mammal(X).

% Rule 5: Same class animals
same_class(X, Y) :-
    animal(Class, X),
    animal(Class, Y),
    X \= Y.

```

Source code for the animal classification expert system in Prolog is available Here.

Query

```
1. Check if a dolphin is a mammal
?- is_mammal(dolphin).
true.

2. List all animals that can really fly
?- can_really_fly(X).
X = eagle ;
X = parrot ;
X = bat ;
false.

3. Find animals that live in water but are not fish
?- water_animal_not_fish(X).
X = dolphin ;
false.

4. List mammals with fur
?- has_fur_animal(X).
X = tiger ;
X = elephant ;
X = bat ;
X = kangaroo ;
false.
```

```
5. Find animals in the same class as a parrot
?- same_class(parrot, X).
X = eagle ;
X = penguin ;
X = owl ;
X = ostrich ;
false.

6. Find nocturnal flyers
?- nocturnal_flyers(X).
X = owl ;
X = bat ;
false.

7. Find carnivores that live in water
?- water_carnivores(X).
X = shark ;
X = dolphin ;
false.
```

Conclusion

Conclusion:

This project demonstrates how Prolog can effectively represent and reason about a set of facts and rules in a knowledge-based system. It enables logical inference, making it easy to classify animals and query their characteristics.

Challenges:

- Representing exceptions (e.g., penguins) requires careful rule design.
- Large knowledge bases may require structured organization for efficient querying.
- Beginners may find Prolog’s declarative logic style challenging initially.

Thank you very much!