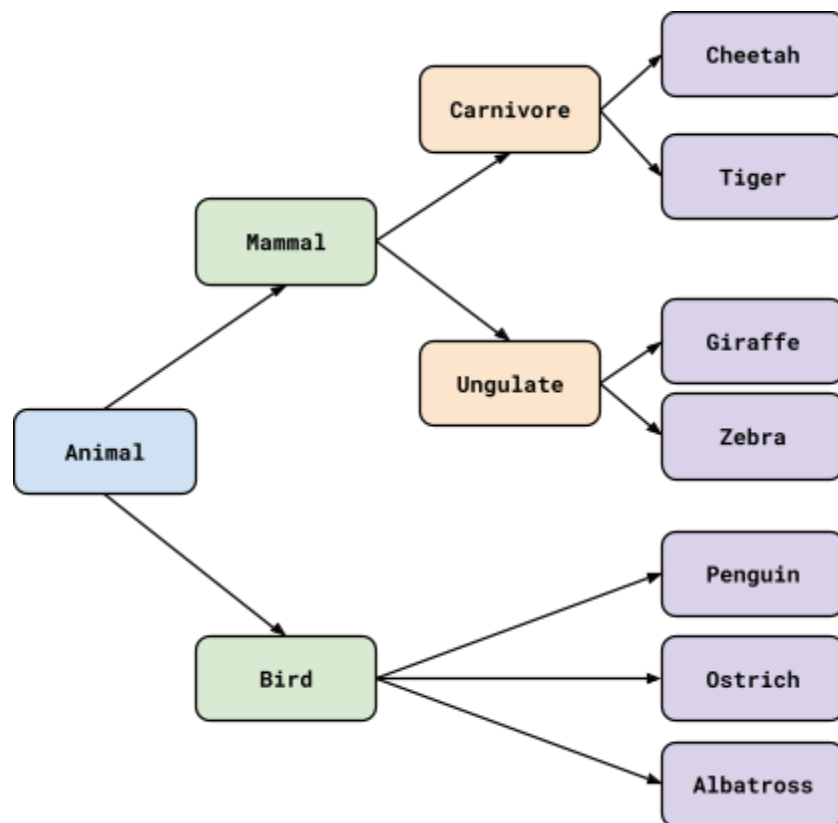


Problem One

Use frames to represent the info in ZooKeeper. Add a few more frames to be able to represent a house cat, a songbird, and cartoon characters. Create an instance of a zebra and a house cat. Also do Tweety and Sylvester. Specify how you deal with situations where there is more than one category applicable to an object; which category does it inherit from? Does it get info from both? What if there is a conflict? Can we handle continuous filler values?

Representing ZooKeeper Information Using Frames

Before explaining the frames for each animal, we can look at the ZooKeeper RuleBase and show the hierarchy between the isa links.



We added animal -although it does not exist in the ZooKeeper rule base- to merge bird and mammal into a single category as well. As we can see, an *animal* is the most general representation of our fact-base. We can start by identifying the specific characteristics of each category and which characteristics they inherit from the top-level definition.

This is what the animal frame will look like:

```
(ANIMAL
  (status Alive)
  (has-skin True)
  (can move)
  (needs-food True)
```

The animal class branches into two specific categories: Mammals and Birds. This is what a Mammal and Bird frame would look like.

| | |
|--|---|
| <pre>(MAMMAL (is-a <i>ANIMAL</i>) (has-hair <i>True</i>) (gives-milk <i>True</i>))</pre> | <pre>(BIRD (is-a <i>ANIMAL</i>) (flies <i>True</i>) (lays-eggs <i>True</i>) (has-feathers <i>True</i>))</pre> |
|--|---|

While the bird class does not branch out to anything (other than specific bird types), the mammal class branches out to Ungulate and Carnivore. This is what a Carnivore and Ungulate frame would look like:

| | |
|---|--|
| <pre>(CARNIVORE (is-a <i>MAMMAL</i>) (eats meat) (has-pointed-teeth <i>True</i>) (has-claws <i>True</i>))</pre> | <pre>(UNGULATE (is-a <i>MAMMAL</i>) (has-hoofs <i>True</i>) (chews-cud))</pre> |
|---|--|

Lastly, the specific animal categories would be like the following:

| | |
|--|--|
| <pre>(CHEETAH (is-a <i>CARNIVORE</i>) (color <i>tawny</i>) (has-dark-spots <i>True</i>))</pre> | <pre>(TIGER (is-a <i>CARNIVORE</i>) (color <i>Tawny</i>) (has-black-stripes <i>True</i>))</pre> |
| <pre>(ZEBRA (is-a <i>UNGULATE</i>) (color <i>White</i>) (has-black-stripes <i>True</i>))</pre> | <pre>(GIRAFFE (is-a <i>UNGULATE</i>) (color <i>White</i>) (has-black-stripes <i>True</i>))</pre> |
| <pre>(OSTRICH (is-a <i>BIRD</i>) (color <i>Black and White</i>) (not flies) (has-long-legs <i>True</i>))</pre> | <pre>(PENGUIN (is-a <i>BIRD</i>) (color <i>Black and White</i>) (not flies) (swims))</pre> |
| <pre>(ALBATROSS (is-a <i>BIRD</i>) (not flies))</pre> | |

Adding Frames for House Cat, Songbird, and Cartoon Characters**(HOUSE-CAT**

(is-a *CARNIVORE*)
 (has-whiskers *True*)
 (domesticated *True*)
 (meows *True*))

(SONGBIRD

(is-a *BIRD*)
 (sings *True*)
 (size *Small*)
 (flies))

(CARTOON-CHARACTER

(is-a *ANIMAL*)
 (exists-in TV))

Creating Instances for House Cat, Tweety, and Sylvester**(zebra_instance_1**

(instance-of *ZEBRA*)
 (name Marty)
 (age 7)
 (location "Central Park"))

(house_cat_1

(instance-of *HOUSE-CAT*)
 (name Eepy)
 (age 3)
 (sleeping *True*))

(tweety_instance_1

(instance-of *SONGBIRD*)
 (instance-of *CARTOON-CHARACTER*)
 (name Tweety)
 (color Yellow)
 (not flies))

(sylvester_instance_1

(instance-of *HOUSE-CAT*)
 (instance-of *CARTOON-CHARACTER*)
 (name Sylvester)
 (color (and Black White))
 (talks *True*))

Situations Where There is More Than One Category

When an object falls into more than one category—for example, Sylvester is a house cat and also a cartoon character—the object inherits the features of both categories. In our case, Sylvester has the characteristics of a house cat (has whiskers, is domesticated, etc.) and also the characteristics of a cartoon character like those on TV.

What if there is a conflict?

If there is a conflict between the features of two inherited categories, the lower level category (for example, Carnivore in the case of Carnivore vs. Animal) will get the priority as it is more specific and closer to that object.

Can we Handle Continuous Filler Values?

This type of initialization can handle continuous filler values as we can just add anything we want into the slots of the frame. For example, zebra is 7 years old. In that field, we could have added 7.5 as well.

Problem Two

Frames can also be used to represent verbs. In this case the verb is the name of the frame and the slots are what is needed to go with the verb to express the thought. Represent the following sentences in frames:

Suzie told Robbie to put the wedge on the red block.

```
(communicate
  (method speech)
  (tense past)
  (sender Suzie)
  (receiver Robbie)
  (message
    (action
      (type move)
      (tense future)
      (effector Robbie)
      (target wedge)
      (destination red-block
        (position on))))))
```

Robbie was unhappy because of the poor grade he received on a test.

```
(status
  (tense past)
  (target robbie)
  (emotion unhappy)
  (cause
    (tense past)
    (object test)
    (result
      (score poor-grade))))
```

Joe moved the car to the garage.

```
(action
  (type move)
  (tense past)
  (effector Joe)
  (target car)
  (destination garage
    (position inside)))
```

Suzie hit Robbie, which made him sad.

```
(status
  (tense past)
  (target robbie)
  (emotion sad)
  (cause
    (tense past)
    (object Suzie)
    (result
      (action
        (type hit)
        (tense past)
        (effector Suzie)
        (target Robbie)
        (object-used hand))))))
```