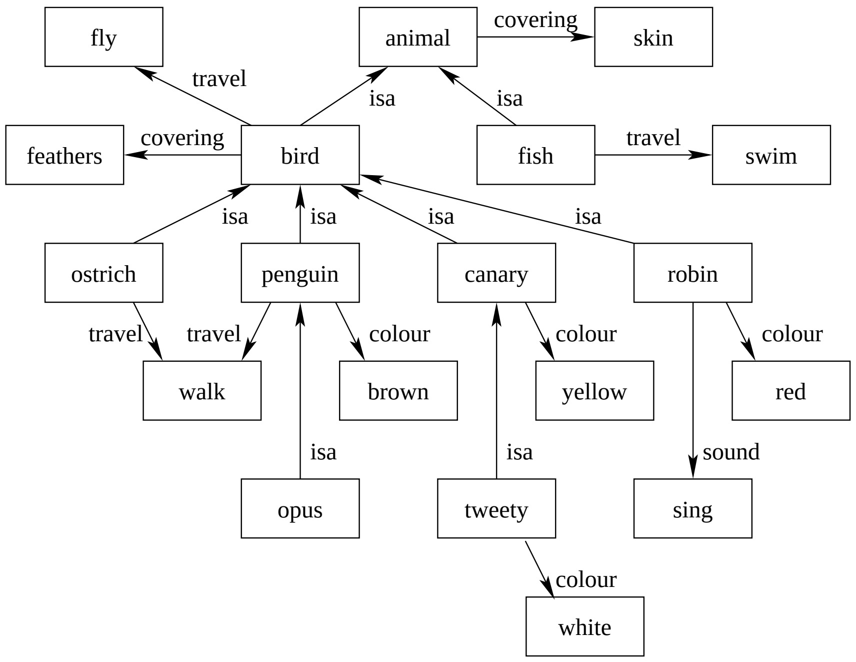
CARLETON UNIVERSITY Department of Systems and Computer Engineering
SYSC 3101 Programming Languages

Assignment 3 **Logic Programming** **Sunday, April 3rd, Midnight.**

Semantic Networks

5 Marks

The term “semantic network” depicts a family of graph-based representations. Collins and Quillian (1969) modeled human information storage and management using a semantic network (cf. figure below). In the semantic net below, nodes represent individuals (objects) such as the canary tweety and classes such as **ostrich**, **robin**, and **bird**. *isa* links represent the class hierarchy relationship. The following canonical forms for the data relationships within the net are adopted. We use *isa(Object, Parent)* to indicate that Object is a member of Parent and a *hasproperty(Object, Property, Value)* predicate to represent property relations. *hasproperty* indicates that Object has Property with Value. Object and Value are nodes in the network, and Property is the name of the link that joins them. For example, *isa(bird, animal)* says that bird is a member of **animal** and *hasporperty(bird, cover, feathers)* says that bird is covered by feathers.



A semantic network describing birds

Write in Prolog, a simple recursive search algorithm (called `whatProperty`) to find whether an object in the semantic net has a particular property. Properties are stored in the net at the most general level at which they are true. Through inheritance, an individual or subclass acquires the properties of its superclasses. Thus, the property fly holds for bird and all its subclasses. Use SWI-Prolog for your program. SWI-Prolog will be used to test your program.

For example, the Prolog query: `whatProperty(bird, PropertyType, PropertyValue)` should produce the following response:

PropertyType = **travel** *i.e. a bird can fly*
PropertyValue = **fly** ;

PropertyType = **cover** *i.e. a bird is covered by feathers*
PropertyValue = **feathers** ;

bird - inherits the properties of - animal (at this point, bird inherits the properties of its superclass animal)

PropertyType = **cover** *because of inheritance, a bird is also an animal*
thus having skin for covering
PropertyValue = **skin** ;

Hint:- You will have to create a facts base consisting of all the *isa* and *hasProperty* relations. Your simple recursive search algorithm will use this facts base to answer queries.

Unon

(5 marks)

Write a prolog function to return the union of two sets. (Note that union is built in, so call your function `myunion`). For example

```
myunion( [a b c], [a d b], X )
```

will return `X=[a b c d]`. You should use the function `member(X,Y)` to determine whether `X` is in the list `Y`. Hint: You will have to use cut `(!)` to stop backtracking if `memeber` is true, otherwise your new list may end up with duplicates.

Remove At position

(15 marks)

Implement the function "`remove_at(list,pos,Item,Remainder)`" which removes the `Item` from the position `pos` from `list` and returns a new list `Remainder` with `Item` removed.

Example:

```
?- remove_at([a,b,c,d,e],3,I,R).
I = c
R = [a,b,d,e]
```

Don't forget that Prolog has a rather weird way of doing math, i.e., $K = K + 1$ won't work. Further, since you need to count, don't forget that you can test variables, i.e., `J > 1`.