

Here is a database of facts and rules. Write a simple Prolog-style database which contains facts and rules representing this information.

Creatures come in two types: humans and birds.

One type of human is a man.

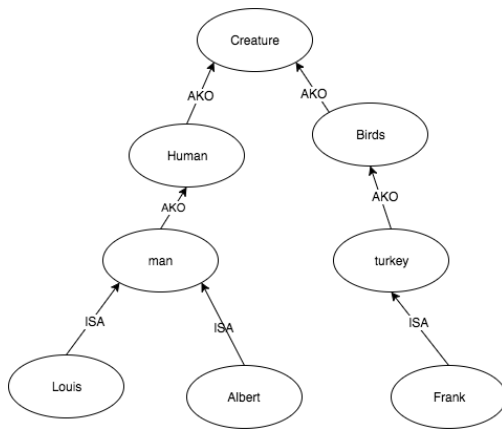
One type of bird is a turkey.

Louis is a man.

Albert is a man.

Frank is a turkey.

1. Draw this taxonomy as a graph, with "creature" at the root, and label the edges with AKO or ISA, whichever is appropriate.



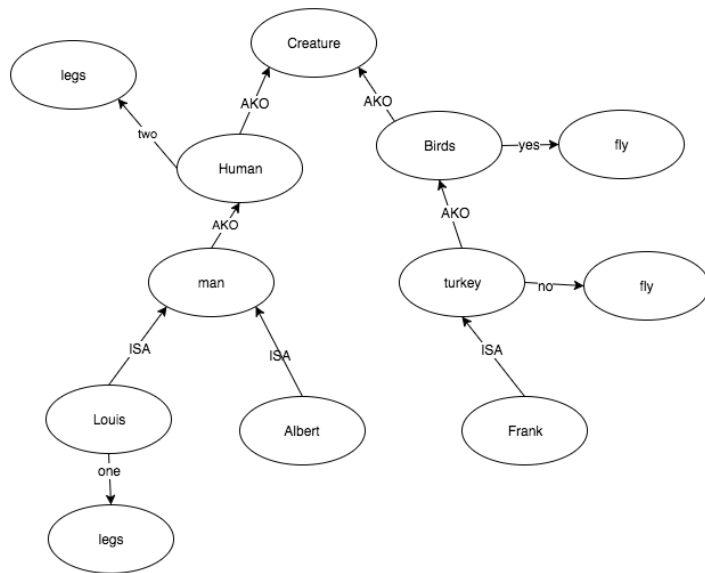
Suppose these facts were represented by seven FOPL facts of the form `edge(<sourceNode>, <linkType>, <destinationNode>)`. Implement these facts as Prolog facts.

Using as a toplevel rule head the syntax `rel(SourceNode, RelationshipType, DestinationNode)` and any other predicates you need, write a set of one or more rules to allow the inference that:

1. Louis is a man, Louis is a human, and Louis is a creature.
2. Albert is a man, Albert is a human, and Albert is a creature.
3. Frank is a turkey, Frank is a bird, and Frank is a creature.

Your rules should follow strict Prolog syntax, and should allow inference over hierarchies of any depth, not just the depth in this example

Now add nodes and edges to the network to represent the knowledge that humans normally have two legs and birds can normally fly, but Louis has one leg and turkeys cannot fly. Using fact syntax such as `property(<node>, legs, two)` and `property(<node>, fly, no)`, indicate which new facts will be necessary, and show in your network sketch from Part (a) where they should be added.



Add rule(s) to allow inference that (i) Frank cannot fly, (ii) Albert has two legs. and (iii) Louis has one leg.