

Assignment 1

Due: Tuesday, April 25th (in class, after class)

Problem 1

Consider the rock classification hierarchies (Slide 30, Handout 1):¹ A set of rock-names is hierarchically classified according to different properties (*fabric*, *chemical-composition*, *texture*).

- For each of the different hierarchies, explain what the parent-child relation between two nodes means. What does a node stand for?
- Give another example of an (ideally scientific) application domain, in which a set of terms (or names) gives rise to two (or more) different hierarchies. Choose an example, such that a significant amount of terms occur in more than one hierarchy.
- Consider a set of terms $T = \{t_1, \dots, t_n\}$ and two hierarchies $\text{isa}_A/2, \text{isa}_B/2$, i.e., terms T can be classified hierarchically w.r.t. two different organizational aspects A and B. Now consider a database instance I over the schema $S = \{\text{isa}_A/2, \text{isa}_B/2\}$, i.e., a finite set of facts of the form $\text{isa}_A(t_{child}, t_{parent})$ and $\text{isa}_B(t_{child}, t_{parent})$. Discuss how you might “mine new knowledge” about T and/or A and B, using I . Use examples from (a) or (b) to illustrate.

Problem 2

Consider the following definitions of concepts:

- A **HappyFather** is a **Man** such that all of his children are **Rich** or **Happy**.
 - A **HappyMother** is a **Woman** with at least one child that is **Happy**.
- Define **HappyFather** and **HappyMother** using description logic (DL) syntax. Use a binary relation $\text{hasChild}/2$ to model the parent-child relationship.
 - Same as in (a) but use first-order (FO) logic syntax. Hint: Use the translation rules from DL to FO from class.

Now consider the database schema $S = \{$ **Person**($SSN, Name, Gender$),
hasChild(SSN_Parent, SSN_Child),
Rich(SSN), **Happy**(SSN) $\}$

- Define **Man**, **Woman**, **HappyFather**, and **HappyMother** as Datalog views over S .
Hint: to express a universal statement (such as “parents having only girls”) you might want to use double negation (e.g. “not among the parents having no girls”).
- Same as in (c), but use SQL views.
- Give a database instance I of S having a happy father and mother, and an unhappy father and unhappy mother. Try to make I as small as possible.

¹The original article will also be available via the class page.