

Assignment 2

Due: Tuesday, May 23rd (in class, after class)

Problem 1. Assume you want to formalize in logic a notion of “part-of”, i.e., a binary relation $\text{part_of}(X, Y)$ that is true for a pair (x, y) if and only if x is a part of y . For example, one might say that a cellular *compartment* is part of a *neuron*, or that *engine* is part of *car*.

- Give different “readings” or meanings of x is-part-of y . Hint: think whether x and y are instances (of some classes/concepts), or whether they are classes/concepts themselves. Also think about whether it means that some/all y 's have some/only x 's as their parts.
- Give a set of logic formulas or Datalog rules that capture some basic properties of $\text{part_of}/2$. Hint: e.g., you might want to state that part_of is asymmetric, i.e., if $\text{part_of}(x, y)$ then it cannot be that $\text{part_of}(y, x)$.
- Give a relation that is distinct from $\text{part_of}/2$, yet satisfies the same axioms that you gave in (b). Briefly discuss what this tells us about our axioms.

Problem 2. Formalize the following statements as logic formulas or Datalog rules:

- For every person x , there exist persons y and z , such that x has parents y and z , where y and z are distinct persons.*
Hint: use relations $\text{person}/1$, $\text{hasParent}/2$, and $\text{=}/2$
- For every married person x , there is a person y , different from x , to which x is married.*
Hint: use relations $\text{married}/1$, $\text{married_to}/2$ (and possibly others, as needed).
- A bachelor x is a person who is not married to anyone.*

Problem 3. Consider the slideset # 2 (Querying Relational Data [2-rel-fo.pdf]), and specifically the slides #19 (Datalog Summary) and #20 (Relational Operators in Datalog).

Assume that the rule

$$\text{boss}(E, \text{Mgr}) \leftarrow \text{employee}(E, \text{Sal}, \text{Dept}), \text{dept}(\text{Dept}, \text{Mgr})$$

derives some concrete answer tuple $\text{boss}(e, m)$

- Give an example database instance I that allows to derive this answer tuple.
- Explain the “lineage” of the answer tuple from I , e.g., by giving a proof tree.