Reg. No: 2022/45/B/ST6/00457; Principal Investigator: prof. dr hab. Jerzy Marcinkowski

Positive First Order Logic and Beyond. Some Puzzles in Database Theory.

a Research Grant Proposal (popular science summary)

Jerzy Marcinkowski

June 2022

Our project is in the area of database theory. This theory, inspired by the practice of databases, tries to: • identify, describe and study, in an abstract, mathematical way, the underlying theoretical mechanisms behind the practical applications; • propose new concepts and mechanisms, which could be applicable in databases practice; • explain the failures of some practical ideas, by proving negative results.

In our proposal we describe, in some detail, three sub-areas of database theory and present fundamental open problems in these sub-areas that we plan to work on. None of them can be honestly explained on one page of popular text. But let us accept the risk of being slightly imprecise, and try to tell something about one of them.

Imagine we have a database storing information about who *likes* whom, and there are five tuples in this database: a *likes* b, b *likes* c, c *likes* a, d *likes* c and a *likes* d.

And imagine there are two database queries:

query α: Give me all the triples $[x, y, z]$ such that x *likes* y and y *likes* z and z *likes* x.

query β: Give me all the triples $[x, y, z]$ such that x *likes* y and x *likes* z.

Now, when you apply query α to our database, then six tuples will be returned[1]: $[a, b, c]$, $[b, c, a]$, $[c, b, a]$, $[a, d, c]$, $[d, c, a]$, $[c, d, a]$.

When you apply query β to our database, then tuples $[a, b, d]$ and $[a, d, b]$ will be returned. But not only, Notice that β does not say that y and z must be different. So there will also be tuples $[b, c, c]$, $[a, b, b]$, $[d, c, c]$, $[a, d, d]$ and $[c, a, a]$. So in total β returns seven tuples, one more than α does.

And, as it turns out, this is not a coincidence: whatever database you take, query β will return at least as many tuples as α. The proof of this fact is maybe not difficult, but certainly not trivial.

In such case, as illustrated by the above example, we say that query β contains query α.

It would be very useful to have an algorithm[2] which, for arbitrary given queries ϕ and ψ, would (preferably in a fraction of a second) tell us whether ϕ contains ψ. Such an algorithm would serve as an important building block in all database engines. But existence of such algorithm is a celebrated open problem:

The Query Containment Problem (Open): Does there exist an algorithm which would read two queries, and tell us whether the first of them contains the second?

A lot of top quality brainpower has been spent in last 30 years on attempts to solve this problem. Some partial results were achieved. For example, it is known that if inequality was allowed (so that we could specify that elements of the returned tuples must be different) then **there is no such algorithm**. Not because we do not know it, but because it simply does not exist[3].

One of the research goals of this project is to try to contribute to this effort.

[1] Notice that the ordering of the elements matters in such tuples.

[2] Or: " a computer program".

[3] We can sometimes *prove* that an algorithm does not exist. This relates to the mathematical phenomenon of *undecidability*.