LGruDat: Logical Foundations of Databases
Lecture 11 — Relational algebra and FOL

Tadeusz Litak

February 10, 2014

1 Relational algebra

We could not quite get straight the right definition of named variant of relational algebra, its supply of connectives and the most suitable variant of first-order syntax. The aim of this note is to get it right and to clarify *why* we had some of those problems. It is also intended as wrap-up and closure for everything we did.

The exercises are for you to practice and think over, don't submit them of course.

Please have a look at Table 1. From now on, this is the one and only official syntax of SPJR(U) and Codd's (named) relational algebra. Note in particular:

- The exact syntax of the input relation primitive: no point in allowing $R(i, j, \dots)$ as we have the renaming operator anyway
- I don't take the empty relation as a primitive. Why? See below.
- However, note also that nothing prevents us from taking a projection to empty subset!

Exercise **1**. **(3 pts)** Write down the semantics of these queries: the semantics should be an **ordered pair** consisting of the type/arity M of the query **and** a subset of functions $M \to \mathsf{A}$. Then write the obvious translation into FOL. Then suitably adjust the semantics of FOL so that the meaning of a formula is of the same type as models of C(N)RA. **What is the semantics of sentences now?** Prove that the semantics of query and its translation coincide.

2 SRNF revisited

The correct definition of SRNF (Substituted Range Normal Form) is

Table 1: Operations of SPJR(U) and Codd's (named) relational algebra

$$\frac{R \in \Sigma}{\Sigma \vdash R : \{1, \ldots, \Sigma(R)\}} \qquad \frac{\mathsf{c} \in \Sigma}{\Sigma \vdash \langle n : \mathsf{c} \rangle : \{n\}}$$

$$\frac{\Sigma \vdash \gamma_1 : N_1 \quad \ldots \quad \Sigma \vdash \gamma_m : N_m}{\Sigma \vdash \gamma_1 \bowtie \ldots \bowtie \gamma_m : N_1 \cup \cdots \cup N_m} \qquad \frac{\Sigma \vdash \gamma : N \quad M \subseteq N}{\Sigma \vdash \pi_M \gamma : N}$$

$$\frac{\Sigma \vdash M : N \quad i_1, \ldots i_l, k_1, \ldots, k_l \in N}{\Sigma \vdash \sigma_{i_1 = k_1, \ldots, i_l = k_l} \gamma : N} \qquad \frac{\Sigma \vdash \gamma : M \quad \rho : M \Vdash\!\!\to \mathbb{N}}{\Sigma \vdash \delta_\rho \gamma : \rho M}$$

$$\frac{\Sigma \vdash \gamma_1 : N \quad \ldots \quad \Sigma \vdash \gamma_m : N}{\Sigma \vdash \gamma_1 \cup \ldots \cup \gamma_m : N} \qquad \frac{\Sigma \vdash \gamma : N \quad \Sigma \vdash \beta : N}{\Sigma \vdash \gamma - \beta : N}$$

$$\alpha_1, \ldots, \alpha_n, \beta ::= R(\overline{t}) \mid t_1 = t_2 \mid \alpha_1 \wedge \cdots \wedge \alpha_n \mid \alpha_1 \vee \cdots \vee \alpha_n \mid \alpha - \beta \mid \exists \overline{x}. \alpha$$

Again, a couple of points:

- I allow now **no unrestricted negation**
- I allow arbitrary combinations of variables and constants in atomic formulas
- Again, conjunctions and disjunction are tacitly assumed to be not only polyadic, but also associative: they are always automatically flattened.

Why? Let us recall the definition and clarify a point which perhaps wasn't properly discussed during the lecture:

Definition 1. A first-order formula α has Σ-*domain independence property* (Σ-*d.i.p.*), where Σ is any signature extending the supply of symbols Σ' used in α **with possibly some constants (but no added relation symbols!)** if for any $\mathfrak{A} \subseteq \mathfrak{B}$ adequate for Σ s.t. for any $R \in \Sigma', R^{\mathfrak{A}} = R^{\mathfrak{B}}$, we have that $\alpha(\mathfrak{A}) = \alpha(\mathfrak{B})$.

Remember that we allow empty submodels now—if Σ has no constants! Otherwise, a submodel must agree with the original model on the interpretation of constants.

Why this strange assumption on constants and the asymmetry with relation symbols in the definition of d.i.p? This is to stay as faithful as possible to typical database assumptions regarding the Herbrand character of query semantics and the notion of active domain—see AHV (Alice) book.

Exercise **2.a** **(3 pts)** Show that no sentence **in SRNF** in an empty signature has $\emptyset$-d.i.p. **Correction: The previous version of the exercise stated**

wrongly that also sentences of FOL in ordinary syntax can never have d.i.p. in empty syntax, as soon as we allow empty models. As Thorsten pointed out, this is obviously false. See exercise c.

Exercise **2.b** **(1 pts)** Show that $\mathsf{c} = \mathsf{c}$ and $\exists x_1.\, x_1 = \mathsf{c}$ are always domain independent

Exercise **2.c** **(1 pts)** Show that $\forall x.\,(x = x)$ has $\emptyset$-d.i.p.

In summary, one has domain-independent valid sentences for signatures with constants, simply because the presence of constants rules out empty submodels.

Exercise **2.c** **(2 pts)** Show that in such a case, one can also form **unrestricted** negation of d.i. **sentences**. Does it also apply to arbitrary d.i. **formulas**?

Why the database people are not concerned with these problems regarding absence of constants? Simply because of the Herbrand character of their intended models. For them, the situation where the signature contains no constants at all, is too pathological to even merit an open discussion. Cf. their *domain closure axiom* [AHV, Ch. 2]. However, we needed to discuss it here, as having always a constant in the signature is not our standing assumption. It is even in stark contrast to temporary assumptions made in some other parts of the lecture—do you remember which ones?

Finally, let me point out that there are other specific assumptions database people tend to make, such as *the unique name assumption* [AHV, Ch. 2]: every constant names a **distinct** element. This implies there is no need for them to consider equalities between constants: they are either universally true in the unique case of $\mathsf{c} = \mathsf{c}$ or universally false in all other cases.

3 Translating back

Let us say a SRNF formula is in a *DRNF (disjunctive range normal form)* [1] if it is a disjunction of *basic disjuncts* which in turn are of the form $\exists \overline{x}. \bigwedge \alpha_i$ or $\exists \overline{x}. (\bigwedge \alpha_i - \beta)$ where α's are atomic formulas and β is again a basic disjunct.

Exercise **3.a** **(4 pts)** Show that every SRNF formula can be equivalently rewritten into SRNF. Be a bit careful about bound variables!

A variable is *occurring safely* in a basic disjunct if it occurs in one of its atoms α_i which is **not** of the form $x_i = x_j$. A basic disjunct γ is *safe range* if two conditions are satisfied:

- all free variables of γ are occurring safely in it and
- in all of its basic disjunct subformulas (including possibly basic disjunct β under relative complement, all basic disjunct subformulas of β in turn etc. at arbitrary depth) quantifiers bind only safely occurring variables

[1] A word of warning : several such notions and abbreviations we defined for so far are rather free adaptations from the literature and I don't guarantee you'll see them in exactly such a form and under same names elsewhere

Now denote as SR (safe range) the set of all those DRNF's whose

- each free variable is occurring safely in **all** its basic disjuncts (remember $R(x_1, x_2) \vee S(x_1, x_2, x_3)$ counterexample to domain independence?)
- all basic disjuncts are safe range

Is SR finally the exactly right class of formulas to translate back to relational algebra? In principle yes, but it is still too wide to give an elegant **inductive** definition of the translation. So far, only the second clause in the definition of safe range basic disjuncts looked arbitrarily deep into subformulas. To understand the problem, think of the situation where a variable occurring freely in the "subtracted away" basic subdisjunct β gets to occur safely in the whole formula only because of one of α_i's to the left of the difference operator. We want to bring down all such dangerous, potentially untranslatable situations to the atomic level of building blocks allowing the form $\bigwedge \alpha_i - (x_k = x_l)$ as these ones can be handled (how?).

Exercise **3.b** **(4 pts)** Define a notion of *RANF (relational algebra normal form)* for SR, show that every SR can be rewritten into RANF and define the translation from RANF back to relational algebra.

4 Decidability status

Earlier in the lecture we heard (without proof) of the Trakhtenbrot Theorem, according to which finite validity of FO-formulas is **not** recursively enumerable despite being co-r.e. (or, dually, finite satisfiability is not co-r.e. despite being r.e.). You may ask yourself a question now: does the situation look any better for d.i. formulas, in particular those SR in RANF defined above? After all, if we restrict the class of formulas/sentences in question, we should have better chances for a decidability result . . .

However, decidability still fails. For details and bibliographical references see Chapter 6.3 of the AHV book. This is shown via the use of Post Correspondence Problem.

Exercise **4.** **(3 pts)** Derive from undecidabilty of satisfiability that domain independence is an undecidable property too.

Correction: The proof I tried to quickly sketch at the end of the lecture was taken from the AHV book—claim b of Corollary 6.3.2—and it was horribly messed up. I just cannot believe they printed such nonsense. Apologies for trying to pass this onto you without thinking the matter over. A correct proof can be found in a note by Moshe Vardi in Information Processing Letters 1981 which I distributed by mail.

5 Looking back . . .

Let us briefly recapitulate what we have done in the whole semester . . . (on the blackboard? orally?)

6 . . . and forward

After having to recall or actually learn for the first time the absolute basics of classical, *unrestricted* model theory, we have covered initial chapters of finite model theory books and most of the material in the first two parts of the AHV book on foundations of databases. Clearly, there is much more and depending which direction the course would take if we had more time, here is a non-exhaustive list of things we could possibly do:

- If we went further in the finite model theory direction: 0-1 laws, the Fagin theorem, extensions involving recursion: fixpoints and MSO . . . And, of course, computational complexity issues. Our standard fmt books by Ebbinghaus&Flum or Libkin are good references for all these things. 0-1 laws can be also seen as a fairly surprising tool for disproving first-order definability.
- If we went further in the database direction
  - discussing more of real-life languages, in particular SQL. Discussing the relationship of the basic form of SQL queries with Codd's operations is actually quite easy and we will do it on blackboard, but—as we have mentioned—full SQL does not quite fit into FO straightjacket due to the presence of, e.g., counting operators. Absolute basics can be found in Chapter 7 of the AHV book, but see also the above fmt references for everything that involves logics with counting
  - What I particularly regret now is not being able to do is discussing logical view on database dependencies and constraints, i.e., Part C of the AHV book. This is really where the logical perspective comes into its own
  - Just like in "pure" fmt case, extensions with recursion (Part D of the Alice book)—in the DB setting more in the disguise of Datalog—and complexity issues (Part E) should attract a lot of attention in a more ambitious course
  - Finally, non-relational databases, especially semi-structured ones and their query/navigation languages. This is where we could reap more dividends of the introduction we did on modal logic. Maybe we can still spend some time on this on the blackboard. Very roughly, just like the core of relational query languages is first-order in character, variants of modal logics and modal syntax can be seen as the core of query/navigation languages for both graph-structured databases

(DL, RDF, ontologies) and semi-structured ones, which are simply finite trees.

Remember, however, that there was also another reason why we introduced modal logic in the first place—to see an example of preservation/invariance result which survives in the finite **with a digestible proof**. We did not have much chances of proving the other one: the Rossman Theorem (2005) for unions of conjunctive queries. In fact, I believe that the argument at its core is also essentially modal, but we are getting out of well-investigated waters here.