Exercise Sheet 6: Trakhtenbrot's Theorem

Maximilian Marx, Markus Krötzsch

Database Theory, 2023-05-16, Summer Term 2023

Exercise 6.1. Use Trakhtenbrot's Theorem to show that the following problems are undecidable by reducing finite satisfiability to each of them:

1. FO query containment
2. FO query emptiness
3. Domain independence of FO queries

Exercise 6.2. In the lecture, we have seen a logical formula that is finitely satisfiable if and only if the given deterministic Turing machine (DTM) halts after finitely many steps on the given input. Reconsider this formula.

For each of the following statements, decide if it is true or false. Justify your answer in each case by explaining why the statement does (or does not) follow from the formula.

1. If the formula has a model at all, then this model is finite.
2. Every model contains a "start configuration": a right-sequence of elements ("cells") that are not reachable from any other cell via future, and where there is a first element in the chain (a cell with no element to its left).
3. Every model contains exactly one such start configuration.
4. If a cell is reachable from the first cell of the start configuration via future, then it does not have a cell on its left.
5. The future of a cell's neighbour is equal to the neighbour of the cell's future.
6. If the Turing machine halts on the input, then every model of the formula is finite.
7. No cell can ever reach itself via future, i.e., there is no loop in the future relation.

Exercise 6.3. In the lecture, we have seen a logical formula that is finitely satisfiable if and only if the given deterministic Turing machine (DTM) halts after finitely many steps on the given input. Extend this definition so that the resulting formula is satisfiable if and only if:

1. a given non-deterministic TM halts after finitely many steps on a given input.
2. a given DTM halts after at most n steps (for a given number n).
3. a given DTM halts after at most 2^n steps (for a given number n).

Make sure that your encoding is polynomial in n.

Exercise 6.4. Apply the conjunctive query minimisation algorithm to find a core of the following CQs:

1. $\exists x, y, z.\ R(x, y) \wedge R(x, z)$

2. $\exists x, y, z.\ R(x, y) \wedge R(x, z) \wedge R(y, z)$
3. $\exists x, y, z.\ R(x, y) \wedge R(x, z) \wedge R(y, z) \wedge R(x, x)$
4. $\exists v, w.\ S(x, a, y) \wedge S(x, v, y) \wedge S(x, w, y) \wedge S(x, x, x)$

Exercise 6.5. Consider a fixed set of relation names (each with a given arity). Show that there is a Boolean CQ Q_{min} without constant symbols that is most specific in the following sense:

> For every BCQ Q that does not use constants, we find that $Q_{\text{min}} \sqsubseteq Q$.

Is there also a most general BCQ Q_{max} that contains all BCQs without constant names? What is the answer to these questions if the considered BCQs may use constant names? What if we consider FO queries instead?

Exercise 6.6. Explain why the CQ minimisation algorithm is correct:

1. Why is the result guaranteed to be a minimal CQ?
2. Why is the result guaranteed to be unique up to bijective renaming of variables?