Exercises on "Foundational Aspects of Graph Data Management"

Wim Martens

March 23, 2020

Exercise 1

Consider conjunctive queries over graphs as we defined them in the lecture. Define such a query to be *tree-shaped*, if its undirected graph representation is acyclic. More precisely, query

$$Q(\overline{x}) := \big((y_1 \xrightarrow{a_1} z_1) \wedge \cdots \wedge (y_n \xrightarrow{a_n} z_n)\big)$$

is tree-shaped if and only if the undirected graph $G_Q = (V, E)$ with $V = \text{Vars}(Q)$ and $E = \{\{y_i, z_i\} \mid i = 1, \ldots, n\}$ is acyclic.

Give a self-contained algorithm that evaluates Boolean, tree-shaped conjunctive queries on graphs in polynomial time combined complexity. Argue why your algorithm runs in polynomial time.

Exercise 2

In this exercise we dive a bit more deeply into the notion of *treewidth*. We use the definition from Diestel's book on graph theory (Diestel, 2012). Let G be an undirected graph and T be an undirected tree. To simplify notation, we will denote the vertices and edges of G by $V(G)$ and $E(G)$, respectively; and similar for T.

Let $\mathcal{V} = (V_t)_{t \in V(T)}$ be a family of vertex sets of G, that is, $V_t \subseteq V(G)$. The pair $(T, \mathcal{V})$ is called a *tree-decomposition* of G if it satisfies all of the following:

(1) $V(G) = \bigcup_{t \in V(T)} V_t$;

(2) for every edge $e = \{u, v\} \in E(G)$, there exists a node $t \in V(T)$ such that $\{u, v\} \in V_t$;

(3) whenever t_1, t_2, t_3 are nodes in T such that t_2 lies on the undirected path between t_1 and t_3, then $V_{t_1} \cap V_{t_3} \subseteq V_{t_2}$.

The *width* of $(T, \mathcal{V})$ is the number $\max\{|V_t| - 1 : t \in V(T)\}$. (The "minus one" in this definition is there to ensure that a tree has treewidth 1.) The *treewidth* of a graph G is the smallest width k of any tree-decomposition $(T, \mathcal{V})$ of G.

Prove that:

(a) Every undirected tree has treewidth 1.

(b) Every undirected cycle has treewidth 2.

(c) Every $n \times m$ grid has treewidth $\leq \min(n, m)$.

(d) Every k-clique has treewidth $\leq k - 1$.

If you want to argue lower bounds in cases (c)–(d) too, please go ahead. (You can dig into the literature and use theorems.)

Exercise 3

Assume that A and B are non-deterministic finite automata.

(a) Show that you can test in PSPACE whether $L(A) \subseteq L(B)$.

(b) Prove that, in general, testing whether $L(A) \subseteq L(B)$ is PSPACE-hard.

For (b), I suggest that you try reducing from Corridor Tiling:

Corridor Tiling	
Given:	A tiling systems $\mathcal{T} = (T, H, V, t_s, t_f, n)$.
Question:	Does there exist a $k \in \mathbb{N}$ and a mapping $m : [n] \times [k] \to T$ such that all tiling conditions are correct?

Notice that the definition is exactly the same as for exponential corridor tiling, except that we now have rows of width n instead of 2^n.

(c) Can your reduction be adapted to testing $L(r_1) \subseteq L(r_2)$ for regular expressions r_1 and r_2?

(d) What if, in addition, r_2 needs to be a concatenation of expressions of the following form: $(a_1 + \cdots + a_n)$ or $(a_1 + \cdots + a_n)^*$. You can choose $n = 1$.

For more background on tiling problems, see (van Emde Boas, 1997). A copy is freely available on the Internet.

Exercise 4

Assume that you have CRPQs that can only use a or a^* as atoms. Call these *simple transitive CRPQs*, or *stCRPQs*.

(a) Assume that stCRPQs are isomorphic if there exists a bijection between the nodes that preserves the languages on the edges. Prove that, under this notion of isomorphism, there are equivalent minimal stCRPQs that are not isomorphic.

(b) Now let's consider a weaker form of isomorphism. Say that stCRPQs are *weakly isomorphic* if there exists a bijection between the nodes that preserves the edges (but may change the languages). Intuitively, stCRPQs are weakly isomorphic if they have the same shape.

    Are equivalent minimal stCRPQs always weakly isomorphic? Why?

Exercise 5 (*)

We will consider the *data complexity* of evaluating RPQs on *undirected* graphs under simple path semantics. So we consider the following problems:

	SUP(r)
Given:	An undirected graph $G = (V, E)$, two nodes $u, v \in V$.
Question:	Is there a simple undirected path from u to v in G that matches r?

(a) Prove that the complexity of SUP(a^*ba^*) is in PTIME.

(b) Prove that the complexity of SUP(a^*bc^*) is NP-hard.

(c) What is the complexity of SUP($a^*bb^*aa^*b^*$)?

Note: I do not know the answer to (c) myself at the moment.

References

R. Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate texts in mathematics*. Springer, 2012. ISBN 978-3-642-14278-9.

P. van Emde Boas. The convenience of tilings. In *Complexity, Logic and Recursion Theory*, volume 187 of *Lecture Notes in Pure and Applied Mathematics*, pages 331–363. Marcel Dekker Inc., 1997.