Project: Bounds on the Output Size of CQs

Data and Information Management (2025-2026)

VRIJE UNIVERSITEIT BRUSSEL

Read the entire project description carefully before you start.

General Details

Context

This project will test your understanding of CQ output size bounds. It spans the remainder of the semester, and gives you more difficult problems for which you can use your creativity.

Note that this project defines the "PRAC Teamwork" part mentioned in the course specification, hence participation in this project is **mandatory** to be able to pass the course and will determine 20% of your overall mark for the course. (The precise formula and exceptions are defined in the course specification on VUB's Cali platform.) The project itself will be graded on a 100-point scale, with each requirement having a weight indicated in this document.

Teams

The project is designed to be made by **pairs** of students. You are free to assemble your team as you see fit. You are also free to choose to work **individually**. Nonetheless, working individually will not grant you extra points. Once you have formed a team (or have decided to work individually), precisely **one** of the team members should fill out this form at the latest by **Tuesday the 28th of October 2025 at 12:30 in the afternoon**, Brussels time. In case you have trouble finding a team member, this can be indicated by selecting the appropriate option in the form. We will try to match students who selected this option (although we cannot guarantee this).

Deadline

The deadline for this project is on **Sunday the 4th of January 2026 at 23:59**, Brussels time. You have to upload your solution to Canvas, as outlined in the Submission Instructions. The number of submissions will not be limited so that it will be possible to resubmit your project until the deadline. The latest submission before the deadline will be considered and graded.

I. Project Description [90 points]

We have seen in the course that the *fractional edge covering number* $\rho^*(q)$ plays an important role in providing a bound on the output size of join queries q, a result known as the *AGM bound*. If we refer to CQs ahead, we always mean join queries (full and self-join free). If the numbers assigned to hyperedges are *integers* 0 or 1, namely "excluded" or "included", we call them *(integral) edge covers* instead (*c.f.*, Definition 25.2 in the course's PDM book). The *(integral) edge cover number* $\rho(q)$ is then the number of edges included in a minimal edge cover of q's hypergraph.

We now give you three problems that are related to these concepts. The first asks you to compute (fractional) edge cover numbers, the second asks you for which queries the fractional edge cover number equals the edge cover number, and the final problem asks you on which queries the edge cover number can be computed in polynomial time.

P1. Computing Edge Covers **[20 points]**

For each of the queries q defined below, we ask you to:

(a) give the *fractional edge cover number* $\rho^*(q)$; and

(b) give the *edge cover number* $\rho(q)$.

Important: provide a formal argument for why the numbers you give are in fact minimal.

In what follows, we give only the body of the join queries using datalog notation. Given that all join queries must be full and self-join free, the head is always $\text{Answer}(\bar{x})$ where $\bar{x}$ contains all variables occuring in the body of the query. We do not provide a schema as this can be inferred from each query (and is different for each query).

q_1	$R_1(x_1, x_2), R_2(x_2, x_3), R_3(x_3, x_4), R_4(x_4, x_5), R_5(x_5, x_6), R_6(x_6, x_7), R_7(x_7, x_8), R_8(x_8, x_9), R_9(x_9, x_{10})$
q_2	$R_1(x_2, x_3, x_4, x_5), R_2(x_1, x_3, x_4, x_5), R_3(x_1, x_2, x_4, x_5), R_4(x_1, x_2, x_3, x_5), R_5(x_1, x_2, x_3, x_4)$
q_3	$R_1(x_1, x_2, x_3, x_4, x_5, x_6, x_7), R_2(x_1, x_2), R_3(x_1, x_3), R_4(x_1, x_4), R_5(x_1, x_5), R_6(x_1, x_6), R_7(x_1, x_7)$
q_4	$R_1(w_1, w_2, w_3, w_4), R_2(x_1, x_2, x_3, x_4), R_3(y_1, y_2, y_3, y_4), R_4(z_1, z_2, z_3, z_4),$ $S_1(w_1, x_1, y_1, z_1), S_2(w_2, x_2, y_2, z_2), S_3(w_3, x_3, y_3, z_3), S_4(w_4, x_4, y_4, z_4)$
q_5	$R_1(x, u_1, u_2, u_3, u_4), R_2(x, v_1, v_2, v_3, v_4), R_3(x, w_1, w_2, w_3, w_4),$ $R_4(x, y_1, y_2, y_3, y_4), R_5(x, z_1, z_2, z_3, z_4)$
q_6	$R_1(x_1, x_2, x_3), R_2(x_3, x_4, x_5), R_3(x_5, x_6, x_7), R_4(x_7, x_8, x_1), S_1(y_1, y_2, y_3), S_2(y_3, y_4, y_5),$ $S_3(y_5, y_6, y_7), S_4(y_7, y_8, y_1), T_1(x_1, z_1, y_1), T_2(x_3, z_3, y_3), T_3(x_5, z_5, y_5), T_4(x_7, z_7, y_7)$
q_7	$R_1(x_1, x_2, x_3), R_2(x_3, x_4, x_5), R_3(x_5, x_6, x_7), R_4(x_7, x_8, x_1), S_1(y_1, y_2, y_3), S_2(y_3, y_4, y_5),$ $S_3(y_5, y_6, y_7), S_4(y_7, y_8, y_1), T_1(x_1, y_1, x_3, y_3, x_5, y_5, x_7, y_7)$
q_8	$R_1(x_1, x_2), R_2(x_2, x_3), R_3(x_3, x_4), R_4(x_4, x_5), R_5(x_5, x_1),$ $S_1(y_1, y_3), S_2(y_1, y_4), S_3(y_2, y_4), S_4(y_2, y_5), S_5(y_3, y_5)$ $T_1(x_1, y_1), T_2(x_2, y_2), T_3(x_3, y_3), T_4(x_4, y_4), T_5(x_5, y_5)$
q_9	$R_1(u_1, y_1, y_2, v_1), R_2(v_1, y_3, y_4, w_1), R_3(w_1, y_5, y_6, x_1),$ $R_4(u_2, z_1, z_2, v_2), R_5(v_2, z_3, z_4, w_2), R_6(w_2, z_5, z_6, x_2),$ $S_1(u_1, u_2), S_2(v_1, v_2), S_3(w_1, w_2), S_4(x_1, x_2),$ $T_1(u_1, v_2), T_2(u_2, v_1), T_3(v_1, w_2), T_4(v_2, w_1), T_5(w_1, x_2), T_6(w_2, x_1)$
q_{10}	$R_1(u_1, v_1, w_1), S_1(w_1, x_1, y_1), T_1(y_1, z_1, u_1), R_2(u_2, v_2, w_2), S_2(w_2, x_2, y_2), T_2(y_2, z_2, u_2),$ $R_3(u_3, v_3, w_3), S_3(w_3, x_3, y_3), T_3(y_3, z_3, u_3), R_4(u_1, v_1, w_1, x_1, y_1, z_1, u_2, v_2, w_2, x_2, y_2, z_2),$ $S_4(u_2, v_2, w_2, x_2, y_2, z_2, u_3, v_3, w_3, x_3, y_3, z_3), T_4(u_3, v_3, w_3, x_3, y_3, z_3, u_1, v_1, w_1, x_1, y_1, z_1)$

P2. When does $\rho(q)$ equal $\rho^*(q)$? **[35 points]**

In general, for any join query q, it holds that $\rho^*(q) \leqslant \rho(q)$. We ask you to define *a structural property on join queries* such that, for all join queries q for which this property holds, it is the case that $\rho^*(q) = \rho(q)$. The larger the set of queries defined by your property, the higher your grade. For instance, the full CQs are contained within the set of all CQs, in this case the latter is larger. Particularly unique solutions will also be rewarded postively.

Examples of what we mean by structural property could be, for instance, requiring the CQ to be full and self-join free (which you must already assume is the case), or restrictions on the arities of the considered relations. We remark that the aforementioned properties are just stated to illustrate what is meant with a structural property and they may not be actual helpful properties. Note that a

structural property is not a semantic property, such as assuming the query is monotone.

Concretely, we expect that you:

(a) Explain your property intuitively, and give two examples of queries that adhere to it.

(b) Formally argue that for all join queries q on which this property is satisfied, it is the case that $\rho^*(q) = \rho(q)$. This involves a formal definition of your property, and a proof.

P3. When is Computing $\rho(q)$ Tractable? [35 points]

The edge cover problem is known to be NP-complete, and is therefore considered to be *intractable*, we briefly provide its definition:

Edge Cover Problem
Inputs: a join query q and a non-negative integer k. **Output**: *true* if $\rho(q) \leqslant k$ and *false* otherwise.

You are tasked with defining a structural property on join queries that makes the edge cover problem decidable in polynomial time, hence making it *tractable* for such queries. Similar to P2, the larger the set of queries defined by your property, the higher the grade. Particularly unique solutions will also be rewarded postively.

We expect that you:

(a) Explain the property intuitively, and give two examples of queries that adhere to it.

(b) Formally argue why for every join query q on which this property is satisfied, and for every non-negative integer k, that the edge cover problem for q and k is decidable in polynomial time. This involves a formal definition of your property, an algorithm, and a proof of its correctness and its time complexity.

(c) Give an example of an execution of your algorithm on one of the queries you gave in (a).

II. Submission Instructions [10 points]

Your submission should be done via Canvas before the deadline which is on **Sunday the 4th of January 2026** at 23:59, Brussels time. Your submission is a single deliverable, namely the report. It must be submitted in its respective Canvas location. **Only one** student of each team should make the submission in Canvas. The naming convention for each deliverable will be specified below. For your convenience, we include a checklist at the end of this document that you can use to track what you have done and what is missing.

Report [10 points]

You are expected to write a **well-structured** PDF report in English. The report should be created using LaTeX. Precisely, we ask you to use the ACM LaTeX-template, and use the `acmart` document class with the options `nonacm`, `sigconf`, and `screen`. The report is limited to a maximum of 8 pages. In the report, make sure to mention your own name (and that of your colleague, if there is one), in addition to your student id(s) and email(s). Note that the PDF should be named either `report-<student-id1>-<student-id2>.pdf` or `report-<student-id>.pdf` if you work alone.

III. Evaluation Criteria, Plagiarism, and Use of Generative AI Tools

The quality of your report will be evaluated in terms of language, structure, formal language, and most importantly the correctness of your reasoning (including proofs and algorithms).

What is (not) allowed?

This project is to be made strictly by the team members and on their own. This means that you must create your project on an independent basis and you must be able to explain your work, restate your proofs under supervision, and defend your solution. Copying work (text, proofs, etc.) from or sharing it with third parties (*e.g.*, fellow students, websites, public version controlled repositories, etc.) is not allowed. Electronic tools will be used to compare all submissions with online resources and with each other, even across academic years.

What is plagiarism and what happens if plagiarism is suspected?

Any action by a student that deviates from the instructions given and does not comply with the examination regulations is considered an irregularity. Plagiarism is also an irregularity. Plagiarism means the use of other people's work, adapted or otherwise, without careful acknowledgement of sources. (cf. OER, Article 118§2). Plagiarism may relate to various forms of works including text, code, proofs, etc.

Any suspicion of plagiarism will be reported to the dean of faculty without delay. Both user and provider of such code will be reported and will be dealt with according to the plagiarism rules of the examination regulations (cf. OER, Article 118). The dean may decide on (a combination of) the disciplinary sanctions, ranging from 0/20 on the project of the given program unit to a prohibition from (re-)enrolling for one or multiple academic years (cf. OER, Article 118§5). Contact us if you are in doubt as to whether or not something would be considered plagiarism.

Use of Generative AI Tools

Students are **not permitted to use generative AI tools** for any part of their project. Our primary objective is to determine whether *you, the student*, understands the material; and thus, your solutions should be original work originating from you.

Checklist

P1	☐ (a) Gave the fractional edge cover number for each query. ☐ (b) Gave the integral edge cover number for each query.
P2	☐ (a) Explained property intuitively and gave 2 examples. ☐ (b) Gave formal argument, definition, proof, ...
P3	☐ (a) Explained property intuitively and gave 2 examples. ☐ (b) Gave formal argument, definition, algorithm, proof, ... ☐ (c) Gave an example execution of the algorithm on 1 query.
Report	☐ Used the correct LaTeX-template (with correct options). ☐ Report contains $\leqslant$ 8 pages. ☐ Includes names, student IDs, and emails. ☐ Used the correct file naming convention.