

Please submit your work electronically as a PDF file named using the scheme `PublicJQ-mt-MNNN.pdf` (replacing `PublicJQ` with your last name and initials, and `NNNN` with a 4-digit number) by anonymous FTP upload to `ftp.cs.umd.edu`, directory `/incoming/chaw/724`.

1. (20 pts) Describe a method for clustering graph data. Specifically, the input to the clustering algorithm consists of N data points, each of which is a edge- and node-labeled graph. The output consists of clusters of similar graphs. Part of this problem is the task of describing desired properties of such clusters. (For example, how is graph similarity measured?)

You may describe any method you choose, including the method outlined in class, a method you find in the literature (cited properly), or a method you devise on your own. However, you must describe the method clearly enough to enable direct implementation by a programmer who is not familiar with such algorithms. Provide an example input for which this method performs better than the methods studied in class (explaining why). Similarly, provide an example for which the method performs poorly. Comment on the scalability of this method (for disk-resident data).

As should be obvious, this question is open-ended and has a diverse set of correct answers. Your score will mainly depend on whether you describe a *reasonable* method and on how well you analyze it (so don't spend too much time trying to invent the best method in the world unless you really want to).

2. (10 pts) Prove the *Facts* on pages 136 and 138 of [ABS99].
3. (10 pts) Prove or disprove the following claim: If R is a simulation from G_1 to G_2 and R' is a simulation from G_2 to G_1 , then $R \cup R'$ is a bisimulation (as defined on in Section 6.4.3 of [ABS99]).
4. (10 pts) Suppose we modify the definition of schema graphs to use bisimulations instead of simulations. Comment on the relative sizes of a database and its schema graph using the original and the modified definition. Comment on the pros and cons of using bisimulation instead of simulation in this context.
5. (10 pts) In class, we discussed the connections between structuring constraints (informally, typing) expressed by schema graphs and those expressed using Datalog rules. Provide algorithms for translating one to the other. That is, given a schema graph, your algorithm should produce Datalog rules expressing equivalent constraints, and vice versa. Describe your algorithms clearly and give at least an informal justification of their correctness.
6. (10 pts) Exhibit a Datalog program with three different models.

7. (10 pts) Present an algorithm for computing the greatest fixedpoint of a Datalog program. Prove termination.
8. (20 pts) Solve Exercise 12.6 from [AHV95]. Reconcile the fact that we are testing for parity in this query with the fact (or claim, since we have not yet proved it) that the infamous parity query is not expressible in Datalog.

References

- [ABS99] Serge Abiteboul, Peter Buneman, and Dan Suciu. *Data on the Web: From Relations to Semistructured Data and XML*. Morgan Kaufmann, first edition, October 1999.
- [AHV95] Serge Abiteboul, Richard Hull, and Victor Vianu. *Foundations of Databases*. Addison-Wesley, 1995.