TECHNISCHE UNIVERSITÄT DRESDEN

Faculty of Computer Science Institute of Theoretical Computer Science, Chair of Automata Theory

Database Theory

Summer Semester 2016

Exercise Sheet 4 – Conjunctive Queries, CSP, and Hypergraphs

2nd May 2016

Dr. rer. pol. Markus Krötzsch & Dipl.-Math. Francesco Kriegel

Exercise 4.1 Decide if the following conjunctive queries are tree queries by applying (one version of) the GYO algorithm.

(a) $\exists x, y, z, v.\ r(x, y) \wedge r(y, z) \wedge r(z, v) \wedge s(x, y, z) \wedge s(y, z, v)$

(b) $\exists x, y, z, u, v, w.\ r(x, y) \wedge s(x, z, v) \wedge r(u, z) \wedge t(x, v, u, w)$

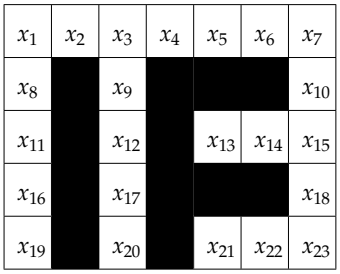
Exercise 4.2 It was outlined in the lecture how to eliminate constants from CQs to transform them to graphs. Apply this transformation to the following query:

$$\exists x, y, z.\ \mathsf{mother}(x, y) \wedge \mathsf{father}(x, z) \wedge \mathsf{bornIn}(y, \texttt{"Dresden"}) \wedge \mathsf{bornIn}(z, \texttt{"Dresden"}).$$

Is the transformed query a tree query?

Imagine we would keep constants in the hypergraph, so that each hypergraph would contain two kinds of vertices (variables and constants). How could we modify the GYO algorithm to handle such hypergraphs directly?

Exercise 4.3 Solve the following combinatorial crossword puzzle using Yannakakis' algorithm (in spirit). Specify the join tree that you are using.



1 hor.:

B	R	I	S	T	O	L
C	A	R	A	M	E	L
P	H	A	R	A	O	H
S	P	I	N	A	C	H
T	S	U	N	A	M	I

1 vert.:

C	L	E	A	R
H	U	M	A	N
P	E	A	C	E
S	H	A	R	K
T	I	G	E	R

3 vert.:

H	A	P	P	Y
I	N	F	E	R
L	A	B	O	R
L	A	T	E	R
U	N	T	I	L

7 vert.:

H	E	A	R	T
H	O	N	E	Y
I	R	O	N	Y
L	O	G	I	C
M	A	G	I	C

13 hor.:

A	N	D
C	A	T
D	I	M
L	A	G
W	I	N

21 hor.:

A	R	C
F	E	E
L	O	W
T	W	O
W	A	Y

Exercise 4.4 It is easy to see that the following is true:

> For every hypergraph $H = \langle V, E \rangle$, the hypergraph $H' = \langle V, E \cup \{e_V\} \rangle$ is acyclic where e_V is a new hyperedge that contains every vertex of V.

In particular, every BCQ can be transformed into a tree query by adding suitable atoms. Does this imply that every BCQ can be answered in polynomial time? Explain your answer.

Exercise 4.5 *Sudoku* is a one-player puzzle game where one has to fill a grid with numbers. An example of a 4×4 Sudoku is as follows:

			3
			4
2			
3			

The grid has to be filled with numbers from $\{1, 2, 3, 4\}$ such that every number occurs exactly once in each row, each column, and each 2×2 subgrid bordered in bold.

For an arbitrary 4×4 Sudoku, specify a BCQ Q and a database instance $\mathcal{J}$ such that $\mathcal{J} \models Q$ if, and only if, the given Sudoku has a solution.

Exercise 4.6 It was shown in the lecture that the 3-colourability problem for graphs can be reduced to the homomorphism problem. Therefore, it can also be expressed as a BCQ answering problem. In which cases is the resulting BCQ a tree query? What is the complexity of solving the 3-colourability problem for these cases?

Exercise 4.7 A propositional formula is in 3CNF if it has the following form:

$$(L_{11} \vee L_{12} \vee L_{13}) \wedge (L_{21} \vee L_{22} \vee L_{23}) \wedge \ldots \wedge (L_{n1} \vee L_{n2} \vee L_{n3})$$

where each L_{ij} is a literal, that is, a propositional variable or the negation of a propositional variable. The 3SAT problem is the problem of deciding if a given 3CNF formula is satisfiable. It is known to be NP-complete.

Reduce 3SAT to the homomorphism problem. To do this, define a suitable hypergraph $\mathcal{I}_\varphi$ for every 3CNF φ and give a template $\mathcal{J}$ such that there is a homomorphism from $\mathcal{I}_\varphi$ to $\mathcal{J}$ if, and only if, φ is satisfiable. Note that this yields an alternative proof of the NP-hardness of the homomorphism problem.

Hint. The template $\mathcal{J}$ can be the same for all instances.