Université de Montpellier - Master informatique Janvier 2022

Théorie des bases de données et de connaissances (HAI933I)

Examen

Exercise 4 (First Order Queries) - 2 pts

Consider the following database:

Films

Title	**Director**	**Actor**
The Imitation Game	Tyldum	Cumberbatch
The Imitation Game	Tyldum	Knightley
...	...	...
Internet's Own Boy	Knappenberger	Swartz
Internet's Own Boy	Knappenberger	Lessig
Internet's Own Boy	Knappenberger	Berners-Lee
...	...	...
Dogma	Smith	Damon
Dogma	Smith	Affleck

Venues

Cinema	**Address**	**Phone**
UFA	St. Petersburger Str. 24	4825825
Schauburg	Königsbrücker Str. 55	8032185
...	...	...

Program

Cinema	**Title**	**Time**
Schauburg	The Imitation Game	19:30
Schauburg	Dogma	20:45
UFA	The Imitation Game	22:45

Write the following queries as first order queries (half a point each):

1. Who are the directors that have worked with the actor "Cumberbatch"?

$$\exists y_T.Films(y_T, x_D, Cumberbatch)$$

2. List the directors that have directed a film shown at "UFA".

$$\exists y_T, y_A, z_T.Films(y_T, x_D, y_A) \wedge Program(UFA, y_T, z_T)$$

3. Find out all the actors that do not have "Tyldum" as one of their directors.

$$\exists y_T, y_D.Films(y_T, y_D, x_A) \wedge \forall z_T, z_D.(Films(z_T, z_D, x_A) \rightarrow z_D \not\approx Tyldum)$$

4. Write a Boolean query to determine if two different directors have directed the same movie.

$$\exists y_T, y_D, z_D, y_A, z_A.Films(y_T, y_D, y_A) \wedge Films(y_T, z_D, z_A) \wedge y_D \not\approx z_D$$

Exercise 5 (Join Trees) - 2 pts

Consider the following queries:

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

2. $\exists z_1, \ldots, z_6.V(z_1, z_2, z_6) \wedge O(z_1, z_2, z_3) \wedge M(z_2, z_4, z_3) \wedge L(z_1, z_3, z_5)$

If possible, define a join tree for each of these queries. If it is not possible to do so, explain why.

To determine if we can define a join tree for a query, we can apply the following algorithm:

GYO-reduction algorithm to check acyclicity:
(after Graham [1979] and Yu & Özsoyoğlu [1979])

Input: hypergraph $H = \langle V, E \rangle$ (we don't need relation labels here)
Output: GYO-reduct of H

Apply the following simplification rules as long as possible:

(1) Delete all vertices that occur in at most one hyperedge

(2) Delete all hyperedges that are empty or that are contained in other hyperedges

Definition

A hypergraph is **acyclic** if its GYO-reduct is $\langle \emptyset, \emptyset \rangle$.
A CQ is **acyclic** if its associated hypergraph is.

Applying the algorithm, we can check that:

- Query (1) is not acyclic.
- Query (2) is acyclic. Furthermore, here's a join tree for this query: $\{O(z_1, z_2, z_3) \to V(z_1, z_2, z_6), O(z_1, z_2, z_3) \to L(z_1, z_3, z_5), O(z_1, z_2, z_3) \to M(z_2, z_4, z_3)\}$