... query containment: we have 2 sets of conjunctive queries. The containment holds if there is a disjunctive on the left that contains the disjunctive on the right. We can CONSIDER QUERY ONE OF THE TIME
| take, in order to check, ~~~~ queries of the left one of the time with the ones on the right.

CQ are mind twisting, let's study, with them we can manage dBs with incomplete informations.

Let's start with an exercise!

$\boxed{26/03/2019}$

An exercise from an exam, we want to check query containment. Showing evaluation and homomorfism;

$$q_1(\underline{x}) :- R(x,x) \wedge B(x,y) \wedge b(y,x)$$

$$q_2(\underline{x}) :- R(x,y) \wedge B(y,z) \wedge b(z,x)$$

$$q_1(x) \subseteq q_2(x)$$

1. Free variables that ARE FREE. (x) When we do containmement IT'S IMPORTANT TO HAVE THE SAME VARIABLE.

$$q_1(a) \subseteq q_2(a)$$

IMPORTANT! FRESH CONSTANT!

$$q_1(a) :- R(a,a) \wedge B(a,y) \wedge b(y,a)$$

$$q_2(a) :- R(a,y) \wedge B(y,z) \wedge b(z,a)$$

2. Compute the canonical Interpretation of $q_1(a)$. It's important to notice that since we have no more free variables, WE ARE DEALING WITH BOOLEAN QUERIES.

$I_{q_1}(a)$

$\Delta^{I_{q_1}} = \{a, y\}$

$R^{I_{q_1}} = \{(a, a)\}$ $\quad B^{I_{q_1}} = \{(a, y), (y, a)\}$

$a^{I_{q_1}} = a$

$y^{I_{q_1}} = y$

In tables:

R

a	a

B

a	y
y	a

In canonical Interpretation, we put the "strings" let's say, I'm considering the elements as strings no as vars.

. Evaluate $I_{q_1} \models q_2(a)$:

GUESS ASSIGNEMENT α AND CHECK α

We have two variables:

$\Rightarrow$ y and z in q_2:

Evaluate

$\alpha(y) = a$

$\alpha(z) = y$

The query is ~~not~~ true in the canonical dB of $q_1(a)$. SATISFY THE ASSIGNEMENT.

There is a second method for checking query contairnement:

1. Compute I_q
2. Guess homomorphism $h: I_q \rightarrow I$
3. Check h is homomorphism. $\longleftarrow$ YOU HAVE TO CHECK

Let's compute the canonical dB $I_{q_2}(a)$

$\boxed{\text{GUESS \& CHECK}}$

$\Delta^{I_2} = \{a, y, z\}$

$R^{I_2} = \{(a, y)\}$

$b^{I_2} = \{(y, z), (z, a)\}$

R

a	y

B

y	z
z	a

Now I need to GUESS homomorphism. Two ways:

$h(a) = a$

$h(y) = \alpha(y) = a$

$h(z) = \alpha(z)$ [illegible]

by the Theorem of Chandra-Merlin.

IF we don't remember chandra Merlin we should recompute, this IS the second way:

$$h(a) = a \quad , \quad h(y) = a \quad , \quad h(z) = y$$

$\underbrace{h(a)=a}_{\text{By construction.}}$ I have to think...

$(a, y) \in R^{I_2}$

$(*) \Longrightarrow (h(a), \underbrace{h(y)}_{\text{I have to guess.}}) \in R^{I_1}$

I NEED TO GUESS AGAIN SOME z.
According to the match of Tuples I did previously (*)

Now I have to check the guessing!
We check all the tuples and if they are in I_{q_1}, we have satisfied the query containment, since h is an homomorphism.

> NOTE! For the exam is important to reasoning, describing the theory behind. Consider each question as an oral exam.

Now we looking to dataBase with incomplete information, in some dB we have NULL values and it is used for denote!

- something that we don't know
- missing values, maybe because a fields do's not apply.
- value not been assigned yet, if it applies I don't know the value.

THREE DIFF. THINGS
ALL ASSIGNED WITH
NULL

So who works with the dB by submitting the query knows what NULL means in different situation.

If a database has holes it can represent different models, identical except for the nulls information! What we wan

$$\forall I \; I \models Db \Rightarrow I \models Query$$

What about if I get a completly empty dB, you have VALIDI then the query answer becomes undecidable.

However there is one case which can be trotted well in logic, the case in which I don't know a value, so I can use LABEL NULL

IF I USE CONJUNCTIVE QUERIES WITH LABEL NULL EVERYTHING WORKS, SO GREAT! BUT WITH NEGATIONS WE ARE DEAD

Tables with label nulls are called NAIVE-tables. We can take the dB associated to naive tables and make all possible interpretations.

In order to answer a query I will consider CERTAIN ANSWER answer that contains something that I know to be true for sure.

In the query containement we are log space in the first query, since it is not involved in guessing, the guess if for the second query, so it is NP-complete on the second query. The containement can be exploited because I can take a boolean query q and check if the q_D (construct with naive tables) is contained in q. I now know to evaluate a conjunctive query on an incomplete dB.

The canonical dB of q_D is D with nulls replaced with constants (maybe not only nulls). The actual procedure is easy, but what about NON BOOLEAN QUERY? I CAN EVALUA OR NOT THEM? YES, IF THE ASWER IS A SET OF CONSTAN BECAUSE FROM IT YOU CAN CONSTRUCT A BOOLEA QUERY. IF I GET AS ANSWER "null," I CANNOT DO IT, but if I throw away all tuples with null I can evaluate CQ as if I am in a complete dB.

Again, elegant theory, reason why it works is very strange, but it is very easy to do.