Summary of Lecture 2

- DRC-queries.
- Relational structures and their isomorphisms.
- DB-structures and DB-isomorphisms.

Typed relational structure

Given a database schema

$$R_i(A_{i,1}, \dots A_{i,m_i}), i \leq n$$

we define a relational structure $\mathcal{A}$ with

- domain D,
- unary relations $\mathcal{A}(A_{i,j} = D_{i,j} \subseteq D$, interpreting the attributes $A_{i,j}$ which occur in the database scheme, and
- relations $\mathcal{A}(R_i) = r_i \subseteq D_{i,1} \times \dots D_{i,m_i}$ for each relation scheme R_i.
- The actual domain of $\mathcal{A}$ is $AC(\mathcal{A})$ is the set

$$AC(\mathcal{A}) = \bigcup_{i,j} \{a \in D : \exists a_1, \dots, a_{j-1}, a_{j+1}, \dots, a_{m_i} R_i(a_1, \dots, a_{j-1}, a, a_{j+1}, a_{m_i})\}$$

The family tree example

We practiced DRC using a family tree example:

- $PERSON(ID, gender, name, firstname, birthdate, birthplace, nationality)$
- $Parent(ID_{child}, ID_{parent}, gender)$
- $Father(ID_{child}, ID_{father})$
- $Mother(ID_{child}, ID_{mother})$
- Write queries for Sibling, grandparent, cousin, uncle, second cousin, etc

The natural number as a databases

We practiced DRC using the natural numbers

$$\mathcal{N} =< \mathbb{N}, Add(x,y,z), Mult(x,y,z), LEQ(x,y), 0, 1 >$$

LEQ stands for less than or equal.

- Write a query for "x divides y" and "x properly divides y"
- Write a query for $SQ(x,y)$ which says $y = x^2$.
- Write a query for $Prime(x)$ which says that x is a prime number.
- Express $Mult(x,y,z)$ using Add and $SQ(x,y)$ only.
- One can show that $Mult(x,y,z)$ cannot be defined using Add and $LEQ(x,y)$ alone.

Background on First Order Logic FOL

I use the notation and definitions from my slides at

https://www.cs.technion.ac.il/ janos/COURSES/LOGIC-2014/

Mostly from Lecture 11

Isomorphisms of relational structures

Let τ be a vocabulary and $\mathcal{A}_1, \mathcal{A}_2$ be two τ-structures with domains A_1, A_2 respectively.

A function $f : A_1 \rightarrow A_2$ is an τ-isomorphism if

- f is one-one and onto,
- for each constant symbol $c \in \tau$ we have $f(\mathcal{A}_1(c)) = \mathcal{A}_2(c)$.
- for each n-ary relation symbol $R \in \tau$ and $\bar(a) \in A_1^n$ we have
$$\bar{a} \in \mathcal{A}_1(R) \text{ iff } f(\bar{a}) \in \mathcal{A}_2(R)$$
- for each n-ary function symbol $F \in \tau$ and $\bar(a) \in A_1^n$ we have
$$\mathcal{A}_2(F)f(\bar{a}) = f(\mathcal{A}_1(F)(\bar{a}))$$

Make examples yourself!

DRC-queries are invariant under isomorphisms

Let f be a τ-isomorphim between $\mathcal{A}_1$ and $\mathcal{A}_2$.

Let $z : Var \to A_1$ be an assignment of the variables to elements of $\mathcal{A}_1$, and let z_f be an assignment of the variables to elements of $\mathcal{A}_2$ defined by $z_f(v) = f(z(v))$.

- For every formula $\phi \in$ FOL and its meaning function M we have

$$M(\mathcal{A}_1, \phi, z) = M(\mathcal{A}_2, \phi, z_f)$$

- Hence,

$$\{\bar{x} \in A_1^n : \phi(\bar{a})\} = \{f(\bar{x}) \in A_2^n : \phi(f(\bar{a}))\}$$

In other words, evaluating DRC-queries in $\mathcal{A}_1$ first and then applying f gives the same result as applying f first and then evaluating it in $\mathcal{A}_2$.

Proof of the isomorphism invariance

We proceed by structural induction on ϕ.

- If ϕ is an atomic formula this follows from the definition of τ-isomorphism.
- If ϕ is of the form $(\phi_1 \wedge \phi_2$, $(\phi_1 \vee \phi_2$, or $\neg\phi_1$ and it is true for ϕ_1 and ϕ_2, then this follows from the meaning function of $\wedge, \vee$ and $\neg$.
- If ϕ is of the form $(\exists v \phi_1$ and it is true for ϕ_1, then this follows from the definition of the meaning function for $\exists$.

HOMEWORK

Domain invariance (aka domain independence)

Here are the relevant slides for the definitions and properties.

https://www.cs.technion.ac.il/ janos/COURSES/236356-19/safety-12-13.pdf

HOMWORK:
Prove by structural induction that RA queries are domain invariant.

Codd's Theorem

Let ϕ a formula of FOL (or DRC.

The following are equivalent.

- ϕ is logically equivalent to an expression $T \in RA$.
- ϕ is domain independent.
- ϕ is logically equivalent to ϕ^{AD}, which is ϕ relativized to its actual domain.

For details see

`https://www.cs.technion.ac.il/ janos/COURSES/236356-19/safety-12-13.pdf`