UNIVERSITÄT ZU LÜBECK
INSTITUT FÜR INFORMATIONSSYSTEME

Özgür L. Özçep

Data Exchange 2

Lecture 6: Universal Solutions, Core, Certain Answers

23 November, 2016

Foundations of Ontologies and Databases for Information Systems

CS5130 (Winter 16/17)

Recap of Lecture 5

Data Exchange

- Specific semantic integration scenario for two data sources with possibly different schemata
- Mapping $\mathcal{M} = (\sigma, \tau, M_{\sigma\tau}, M_{\tau})$
  - σ: source schema
  - τ: target schema
  - $M_{\sigma\tau}$: source target dependencies (mostly: st-tgds)
  - M_{τ}: target dependencies
- Ultimate aim: For given σ instance find appropriate τ instance (solution) to do query answering on it

- SOLEXISTENCE$_{\mathcal{M}}$: Is there a solution for a given $\mathcal{M}$
- Chase construction for finding solutions
- Chase construction gives sufficient and necessary condition if termination is guaranteed
- Termination with weakly acyclic dependencies

End of Recap

Universal Solutions

What are Good Solutions?

- We are seeking **universal** solutions: they represent all other ones

- A solution $\mathfrak{T}$ may contain NULLs
- A DB instance is **complete** iff it does not contain NULLs

- $\mathit{Rep}(\mathfrak{T})$ = all complete DBs instances that represent $\mathfrak{T}$
- Explicate "represent" by homomorphism notion
- Now formally define

$$\mathit{Rep}(\mathfrak{T}) = \{\mathfrak{T}' \mid \text{There is } h : \mathfrak{T} \xrightarrow{hom} \mathfrak{T}' \text{ for complete } \mathfrak{T}'\}$$

Homomorphism

- Intuitively, homomorphisms are structure preserving mappings
- Defined here for DB instances but similarly definable for arbitrary structures

Definition

A **Homomorphism** $h : \mathfrak{T} \xrightarrow{hom} \mathfrak{T}'$ is a map

$$h : Var(\mathfrak{T}) \cup CONST \rightarrow VAR(\mathfrak{T}') \cup CONST$$

s.t.

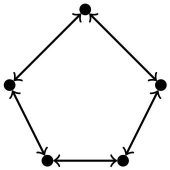
- $h(c) = c$ for all $c \in CONST$ and
- if $R(\vec{t}) \in \mathfrak{T}$, then $R(h(\vec{t})) \in \mathfrak{T}'$

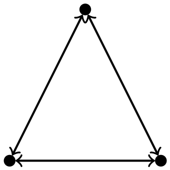
Wake-Up Exercise

Consider two instances that are graphs, namely

- $\mathfrak{G}$ = cycle on 5 nodes with marked nulls $\nu_1, \ldots, \nu_5$
- $\mathfrak{G}'$ = cycle on 3 nodes with marked nulls ν_1', ν_2', ν_3'.

Give examples of a mapping $h : \mathfrak{G} \to \mathfrak{G}'$ that is a homomorphism, resp. not a homomorphism.



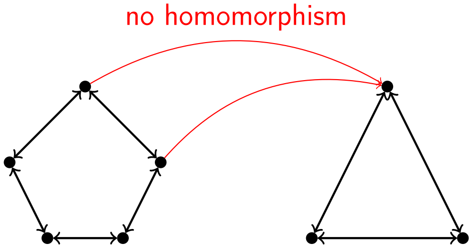


Wake-Up Exercise

Consider two instances that are graphs, namely

- $\mathfrak{G}$ = cycle on 5 nodes with marked nulls $\nu_1, \ldots, \nu_5$
- $\mathfrak{G}'$ = cycle on 3 nodes with marked nulls ν_1', ν_2', ν_3'.

Give examples of a mapping $h : \mathfrak{G} \to \mathfrak{G}'$ that is a homomorphism, resp. not a homomorphism.



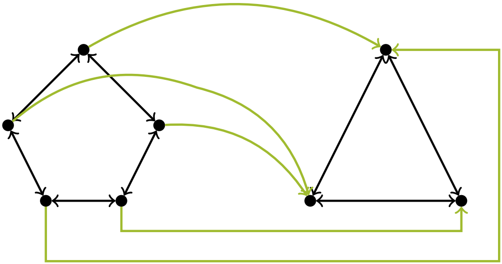
Wake-Up Exercise

Consider two instances that are graphs, namely

- $\mathfrak{G}$ = cycle on 5 nodes with marked nulls $\nu_1, \ldots, \nu_5$
- $\mathfrak{G}'$ = cycle on 3 nodes with marked nulls ν_1', ν_2', ν_3'.

Give examples of a mapping $h : \mathfrak{G} \to \mathfrak{G}'$ that is a homomorphism, resp. not a homomorphism.

homomorphism



Universal Solutions

- There are three **equivalent** characterizations of universal solutions $\mathfrak{T}$; mainly work with third as definition

Definition (Universal Solution)

1. **Solution $\mathfrak{T}$ describing all others**

$$\{\mathfrak{T}' \in \mathit{SOL}_{\mathcal{M}}(\mathfrak{S}) \mid \mathfrak{T}' \text{ complete}\} \subseteq \mathit{Rep}(\mathfrak{T})$$

2. **Solution $\mathfrak{T}$ as general as all others**

$$\mathit{Rep}(\mathfrak{T}') \subseteq \mathit{Rep}(\mathfrak{T}) \quad \text{for every } \mathfrak{T}' \in \mathit{SOL}_{\mathcal{M}}(\mathfrak{S})$$

3. **Solution $\mathfrak{T}$ mapping homomorphically into others**

$$\text{For all } \mathfrak{T}' \in \mathit{SOL}_{\mathcal{M}}(\mathfrak{S}) \text{ there is } h : \mathfrak{T} \xrightarrow{\mathit{hom}} \mathfrak{T}'$$

Example (Universal Solution)

Source DB

Flight (	src,	dest,	airl,	dep	)
	paris	sant.	airFr	2320	

Target DB

Routes(fno, src, dest)

Info(fno, dep, arr, airl)

- **Dependencies** $M_{\sigma\tau}$

$$\begin{aligned}&Flight(src, dest, airl, dep) \longrightarrow \\ &\quad \exists fno\, \exists\, arr(Routes(fno, src, dest) \wedge Info(fno, dep, arr, airl))\end{aligned}$$

- τ **solutions**

$$\begin{aligned}\mathfrak{T} &= \{Routes(\bot_1, paris, sant), Info(\bot_1, 2320, \bot_2, airFr)\}\\ \mathfrak{T}' &= \{Routes(\bot_1, paris, sant), Info(\bot_1, 2320, \bot_1, airFr)\}\\ \mathfrak{T}'' &= \{Routes(123, paris, sant), Info(123, 2320, 930, airFr)\}\end{aligned}$$

- $\mathfrak{T}$ is a universal solution, $\mathfrak{T}'$ and $\mathfrak{T}''$ are not

Example (Non-existence of Universal Solutions)

- $M_{\sigma\tau} = \{\ E(x, y) \rightarrow G(x, y)\ \}$
- $M_{\tau} = \{\ G(x, y) \rightarrow \exists z\ L(y, z), \quad L(x, y) \rightarrow \exists z\ G(y, z)\ \}$
- Source instance $\mathfrak{S} = \{E(a, b)\}$

- $\mathfrak{T} = \{G(a, b), L(b, a)\}$ is a solution
- But there is no universal solution

Proof sketch (by contradiction)

- A universal solution must have an infinite sequence $(\mathfrak{S}, \{G(a, b), L(b, \nu_1), G(\nu_1, \nu_2), L(\nu_2, \nu_3), G(\nu_3, \nu_4) \dots\})$
- As $\mathfrak{T}$ is finite there must be some identification of an ν_i with a or b or with another ν_j
- In any case a contradiction follows (by constructing a solution into which no homomorphic embedding of $\mathfrak{T}$ is possible)

Example (Non-existence of Universal Solutions)

- $M_{\sigma\tau} = \{\ E(x,y) \to G(x,y)\ \}$
- $M_{\tau} = \{\ G(x,y) \to \exists z\ L(y,z), \quad L(x,y) \to \exists z\ G(y,z)\ \}$
- Source instance $\mathfrak{S} = \{E(a,b)\}$

- $\mathfrak{T} = \{G(a,b), L(b,a)\}$ is a solution
- But there is no universal solution

Proof sketch (by contradiction)

- A universal solution must have an infinite sequence $(\mathfrak{S}, \{G(a,b), L(b,\nu_1), G(\nu_1,\nu_2), L(\nu_2,\nu_3), G(\nu_3,\nu_4)\ldots\})$
- As $\mathfrak{T}$ is finite there must be some identification of an ν_i with a or b or with another ν_j
- In any case a contradiction follows (by constructing a solution into which no homomorphic embedding of $\mathfrak{T}$ is possible)

Example (Non-existence of Universal Solutions)

- $M_{\sigma\tau} = \{\ E(x,y) \rightarrow G(x,y)\ \}$
- $M_{\tau} = \{\ G(x,y) \rightarrow \exists z\ L(y,z), \quad L(x,y) \rightarrow \exists z\ G(y,z)\ \}$
- Source instance $\mathfrak{S} = \{E(a,b)\}$

- $\mathfrak{T} = \{G(a,b), L(b,a)\}$ is a solution
- But there is no universal solution

Proof sketch (by contradiction)

- A universal solution must have an infinite sequence $(\mathfrak{S}, \{G(a,b), L(b,\nu_1), G(\nu_1,\nu_2), L(\nu_2,\nu_3), G(\nu_3,\nu_4)\ldots\})$
- As $\mathfrak{T}$ is finite there must be some identification of an ν_i with a or b or with another ν_j
- In any case a contradiction follows (by constructing a solution into which no homomorphic embedding of $\mathfrak{T}$ is possible)

Example (Non-existence of Universal Solutions)

- $M_{\sigma\tau} = \{ E(x,y) \to G(x,y) \}$
- $M_{\tau} = \{ G(x,y) \to \exists z\, L(y,z), \quad L(x,y) \to \exists z\, G(y,z) \}$
- Source instance $\mathfrak{S} = \{E(a,b)\}$

- $\mathfrak{T} = \{G(a,b), L(b,a)\}$ is a solution
- But there is no universal solution

Proof sketch (by contradiction)

- A universal solution must have an infinite sequence $(\mathfrak{S}, \{G(a,b), L(b,\nu_1), G(\nu_1,\nu_2), L(\nu_2,\nu_3), G(\nu_3,\nu_4) \ldots \})$
- Consider case where $\nu_{2i-1} = a$ and define solution $\mathfrak{T}' = \{G(a,b), L(b,c_1), G(c_1,c_2), L(c_2,c_3), \ldots, G(c_j,c_{j-1})$ for $2i < j$ and fresh c_i
- There must be an $h : \mathfrak{T} \xrightarrow{hom} \mathfrak{T}'$.
- But then $h(\nu_l) = c_l$ and hence $h(\nu_{2i-1}) = c_{2i-1}$, but also $h(\nu_{2i-1}) = h(a) = a$. $\lightning$

Undecidability of Universal Solution Existence

UNISOLEXISTENCE$_{\mathcal{M}}$

- Input: A source instance $\mathfrak{S}$
- Output: Is there a universal solution for $\mathfrak{S}$ under $\mathcal{M}$?

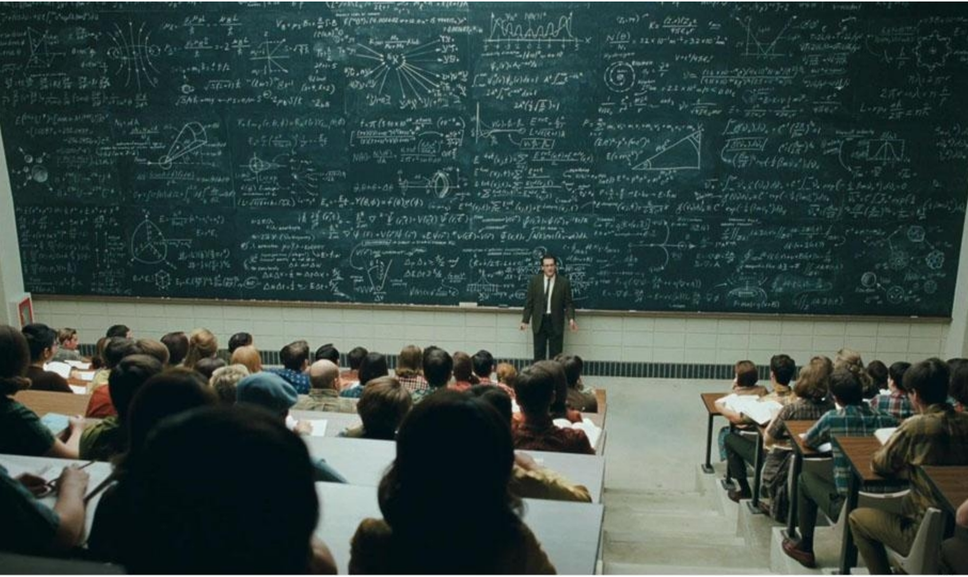
- Allowing arbitrary dependencies leads to undecidability
- Shown by of reduction of halting problem

Theorem

There exists a relational mapping $\mathcal{M} = (\sigma, \tau, M_{\sigma\tau}, M_\tau)$ s.t. UNISOLEXISTENCE$_{\mathcal{M}}$ is undecidable

- Proof in book of Arenas et al. 5 pages long, so ... we do not show it here

By the way: There are Longer Proofs



By the way: There are Longer Proofs

- Recent example: A computer aided proof for a particular case (C = 3) of the Erdős Discrepancy Problem by Lisitsa/Konev
- File containing the proof about 13 GB

- **Lit:** B. Konev and A. Lisitsa. Computer-aided proof of erdos discrepancy properties. Artif. Intell., 224(C):103–118, July 2015.
- **Lit:** `https://rjlipton.wordpress.com/2014/02/28/practically-pnp/`

Definition (Erdős Discrepancy Problem (EDP))

Let (x_n) be a sequence of 1s and 0s and C be a constant. Can one always find positive integers d, k s.t.:

$$| \sum_{i=1}^{k} x_{id} | > C$$

By the way: There are Longer Proofs

Definition (Erdős Discrepancy Problem (EDP))

Let (x_n) be a sequence of 1s and 0s and C be a constant. Can one always find positive integers d, k s.t.: $|\sum_{i=1}^{k} x_{id}| > C$

Illustration:

"A precipice lies two paces to your left and a pit of vipers two paces to your right. Can you devise a series of steps that will avoid the hazards, even if you are forced to take every second, third or Nth step in your series?"

Lit:

https://www.quantamagazine.org/20151001-tao-erdos-discrepancy-problem/

- **Update:** There is now an elegant short proof for the full case by mathematician Terence Tao
- **Lit:** The Erdős Discrepancy Problem. arXiv:1509.05363, https://arxiv.org/abs/1509.05363

Desiderata

- Due to the undecidabiltiy result one has to constrain dependencies
- Constraints such that the following are fulfilled:

(C1) Existence of solutions entails existence of universal solutions

(C2) UNIVSOLEXISTENCE decidable and even tractable

(C3) If solutions exists, then universal solutions should be constructible in polynomial time

Chase Helps Again

Theorem

*Results of successful chase sequences are universal solutions (and these are sometimes called **canonical** universal solutions).*

Proof Sketch

- Have to show only universality of chase $\mathfrak{T}$
- Use the third definition of universality
- Let $\mathfrak{T}'$ be any solution
- Lemma: Adding facts in chase step preserves homomorphism
  (If $\mathfrak{T}1 \overset{\chi}{\rightsquigarrow} \mathfrak{T}2$ by dependency χ, $\mathfrak{T}3$ fulfills χ and there is $h : \mathfrak{T}1 \xrightarrow{hom} \mathfrak{T}3$, then there is $h' : \mathfrak{T}2 \xrightarrow{hom} \mathfrak{T}3$)
- Argue inductively starting from empty homomorphism

Nice Properties of Universal Solutions

Theorem

Let $\mathcal{M} = (\sigma, \tau, M_{\sigma\tau}, M_\tau)$ be a mapping where M_τ is the union of egds and weakly acyclic tgds. Then:

- *UNISOLEXISTENCE$_{\mathcal{M}}$ can be solved in PTIME (C2).*
- *And if solutions exist, then a universal solution exists (C1),*
- *and a canonical universal solution can be computed in polynomial time (C3).*

Example (Non-uniqueness of Canonical Universal Solutions)

- $M_{\sigma\tau} = \{\ P(x) \rightarrow \exists y \exists w (E(x, y) \wedge E(x, w))\ \}$
- $M_{\tau} = \{\ \underbrace{E(x, y) \rightarrow \exists z\ F(y, z)}_{\chi_1}, \underbrace{E(x, y) \wedge E(x, y') \rightarrow y = y'}_{\chi_2}\ \}$
- Source instance $\mathfrak{S} = \{P(a)\}$
- First step: $\mathfrak{T} = \{E(a, \perp_1), E(a, \perp_2)\}$
- Two different solutions
  - Apply χ_1, then χ_2:

    $$\mathfrak{T}_1 = \{E(a, \perp_1), F(\perp_1, \perp_3)), F(\perp_1, \perp_4)\}$$

  - Apply χ_2, then χ_1:

    $$\mathfrak{T}_2 = \{E(a, \perp_1), F(\perp_1, \perp_2)\}$$

Example (Non-uniqueness of Canonical Universal Solutions)

- $M_{\sigma\tau} = \{ \ P(x) \to \exists y \exists w(E(x,y) \wedge E(x,w)) \ \}$
- $M_\tau = \{ \ \underbrace{E(x,y) \to \exists z \ F(y,z)}_{\chi_1}, \underbrace{E(x,y) \wedge E(x,y') \to y = y'}_{\chi_2} \ \}$
- Source instance $\mathfrak{S} = \{P(a)\}$

- First step: $\mathfrak{T} = \{E(a,\perp_1), E(a,\perp_2)\}$
- Two different solutions
  - Apply χ_1, then χ_2:
    $$\mathfrak{T}_1 = \{E(a,\perp_1), F(\perp_1,\perp_3)), F(\perp_1,\perp_4)\}$$
  - Apply χ_2, then χ_1:
    $$\mathfrak{T}_2 = \{E(a,\perp_1), F(\perp_1,\perp_2)\}$$

Example (Non-uniqueness of Canonical Universal Solutions)

- $M_{\sigma\tau} = \{\ P(x) \to \exists y \exists w (E(x,y) \wedge E(x,w))\ \}$
- $M_{\tau} = \{\ \underbrace{E(x,y) \to \exists z\ F(y,z)}_{\chi_1}, \underbrace{E(x,y) \wedge E(x,y') \to y = y'}_{\chi_2}\ \}$
- Source instance $\mathfrak{S} = \{P(a)\}$

- First step: $\mathfrak{T} = \{E(a,\perp_1), E(a,\perp_2)\}$
- Two different solutions
  - Apply χ_1, then χ_2:

$$\mathfrak{T}_1 = \{E(a,\perp_1), F(\perp_1,\perp_3)), F(\perp_1,\perp_4)\}$$

  - Apply χ_2, then χ_1:

$$\mathfrak{T}_2 = \{E(a,\perp_1), F(\perp_1,\perp_2)\}$$

Example (Non-uniqueness of Canonical Universal Solutions)

- $M_{\sigma\tau} = \{\ P(x) \to \exists y \exists w (E(x,y) \land E(x,w))\ \}$
- $M_{\tau} = \{\ \underbrace{E(x,y) \to \exists z\ F(y,z)}_{\chi_1}, \underbrace{E(x,y) \land E(x,y') \to y = y'}_{\chi_2}\ \}$
- Source instance $\mathfrak{S} = \{P(a)\}$

- First step: $\mathfrak{T} = \{E(a,\perp_1), E(a,\perp_2)\}$
- Two different solutions
  - Apply χ_1, then χ_2:

$$\mathfrak{T}_1 = \{E(a,\perp_1), F(\perp_1,\perp_3)), F(\perp_1,\perp_4)\}$$

  - Apply χ_2, then χ_1:

$$\mathfrak{T}_2 = \{E(a,\perp_1), F(\perp_1,\perp_2)\}$$

Non-uniqueness

- Non-uniqueness no serious problem as all universal solutions are good
- Nonetheless one can show

Proposition

Let $\mathcal{M} = (\sigma, \tau, M_{\sigma\tau}, M_\tau)$ be a mapping s.t. M_τ consists of egds only. Then every source instance $\mathfrak{S}$ has a unique canonical solution $\mathfrak{T}$ (up to a renaming of NULLS) under $\mathcal{M}$.

The Core

Running Example: Flight Domain

Source DB σ

Geo(city, coun, pop)

city	coun	pop
paris,	france,	2M

Flight (src, dest, airl, dep)

src	dest	airl	dep
paris	amst.	KLM	1410
paris	amst.	KLM	2230

Canonical Solution $\mathfrak{T}$

Routes(fno, src, dest)

fno	src	dest
$\perp_1$,	paris,	amst.
$\perp_3$,	paris,	amst.

Info(fno, dep, arr, airl)

fno	dep	arr	airl
$\perp_1$,	1410,	$\perp_2$	klm
$\perp_3$,	2320,	$\perp_4$	klm

Serves(airl, city, coun, phone)

airl	city	coun	phone
klm,	paris,	france,	$\perp_5$
klm,	paris,	france,	$\perp_6$

Mapping rules $M_{\sigma\tau}$

1. $Flight(src, dest, airl, dep) \longrightarrow$
   $\exists fno\ \exists\ arr(Routes(fno, src, dest) \wedge Info(fno, dep, arr, airl))$
2. $Flight(city, dest, airl, dep) \wedge Geo(city, coun, pop) \longrightarrow$
   $\exists phone(Serves(airl, city, coun, phone))$
3. $Flight(src, city, airl, dep) \wedge Geo(city, coun, pop) \longrightarrow$
   $\exists phone\ (Serves(airl, city, coun, phone))$

Running Example: Flight Domain

Source DB σ

Geo(city, coun, pop)

city	coun	pop
paris,	france,	2M

Flight (src, dest, airl, dep)

src	dest	airl	dep
paris	amst.	KLM	1410
paris	amst.	KLM	2230

Smallest Solution $\mathfrak{T}^*$

Routes(<u>fno</u>, src, dest)

<u>fno</u>	src	dest
$\perp_1$,	paris,	amst.
$\perp_3$,	paris,	amst.

Info(<u>fno</u>, dep, arr, airl)

<u>fno</u>	dep	arr	airl
$\perp_1$,	1410,	$\perp_2$	klm
$\perp_3$,	2320,	$\perp_4$	klm

Serves(airl, city, coun, phone)

airl	city	coun	phone
klm,	paris,	france,	$\perp_5$
~~klm,~~	~~paris,~~	~~france,~~	$\cancel{\perp_6}$

Mapping rules $M_{\sigma\tau}$

1. $Flight(src, dest, airl, dep) \longrightarrow$
   $\exists fno\ \exists\ arr(Routes(fno, src, dest) \wedge Info(fno, dep, arr, airl))$

2. $Flight(city, dest, airl, dep) \wedge Geo(city, coun, pop) \longrightarrow$
   $\exists phone(Serves(airl, city, coun, phone))$

3. $Flight(src, city, airl, dep) \wedge Geo(city, coun, pop) \longrightarrow$
   $\exists phone\ (Serves(airl, city, coun, phone))$

Running Example: Flight Domain

Source DB σ

Geo(city, coun, pop)

city	coun	pop
paris,	france,	2M

Flight (src, dest, airl, dep)

src	dest	airl	dep
paris	amst.	KLM	1410
paris	amst.	KLM	2230

Routes(fno, src, dest)

fno	src	dest
$\perp_1$,	paris,	amst.
$\perp_3$,	paris,	amst.

Info(fno, dep, arr, airl)

fno	dep	arr	airl
$\perp_1$,	1410,	$\perp_2$	klm
$\perp_3$,	2320,	$\perp_4$	klm

Serves(airl, city, coun, phone)

airl	city	coun	phone
klm,	paris,	france,	$\perp_5$
~~klm,~~	~~paris,~~	~~france,~~	$\cancel{\perp_6}$

Wake-Up-Question

Why not delete similarly $Routes(\perp_3, paris, amst)$?

Running Example: Flight Domain

Source DB σ

Geo(city, coun, pop)

city	coun	pop
paris,	france,	2M

Flight (src, dest, airl, dep)

src	dest	airl	dep
paris	amst.	KLM	1410
paris	amst.	KLM	2230

Routes(fno, src, dest)

fno	src	dest
$\perp_1$,	paris,	amst.
$\perp_3$,	paris,	amst.

Info(fno, dep, arr, airl)

fno	dep	arr	airl
$\perp_1$,	1410,	$\perp_2$	klm
$\perp_3$,	2320,	$\perp_4$	klm

Serves(airl, city, coun, phone)

airl	city	coun	phone
klm,	paris,	france,	$\perp_5$
~~klm,~~	~~paris,~~	~~france,~~	$\cancel{\perp_6}$

Wake-Up-Question

Why not delete similarly $\mathit{Routes}(\perp_3, \mathit{paris}, \mathit{amst})$?

Answer: There are additional facts distinguishing $\perp_1$ and $\perp_3$

Better than Universal? The Core!

- Universal solutions may still contain redundant information
- Seeking for smallest universal solutions: cores

- $\mathfrak{T}'$ is **subinstance** of $\mathfrak{T}$, for short $\mathfrak{T}' \subseteq \mathfrak{T}$, iff $R^{\mathfrak{T}'} \subseteq R^{\mathfrak{T}}$ for all relation symbols R

Definition

A subsinstance $\mathfrak{T}' \subseteq \mathfrak{T}$ is a **core** of $\mathfrak{T}$ iff there is $h : \mathfrak{T} \xrightarrow{hom} \mathfrak{T}'$ but there is not a homomorphism from $\mathfrak{T}$ to a proper subinstance of $\mathfrak{T}'$.

- Intuitively: An instance can be retracted (structure preservingly) to its core but not further

Properties of Cores

Definition

A subinstance $\mathfrak{T}' \subseteq \mathfrak{T}$ is a **core** of $\mathfrak{T}$ iff there is $h : \mathfrak{T} \xrightarrow{hom} \mathfrak{T}'$ but there is not a homomorphism from $\mathfrak{T}$ to a proper subinstance of $\mathfrak{T}'$.

Proposition

1. *Every instance has a core.*
2. *All cores of the same instance are isomorphic (same up to renaming of NULLs)* *($\Longrightarrow$ Talk of the core justified)*
3. *Two instances are homomorphically equivalent iff their cores are isomorphic*
4. *If $\mathfrak{T}'$ is core of $\mathfrak{T}$, then there is $h : \mathfrak{T} \xrightarrow{hom} \mathfrak{T}'$ s.t. $h(\nu) = \nu$ for all $\nu \in DOM(\mathfrak{T}')$*

Main Theorem for Cores

Theorem

1. *If* $\mathfrak{T} \in SOL_{\mathcal{M}}(\mathfrak{S})$*, then also* $core(\mathfrak{T}) \in SOL_{\mathcal{M}}(\mathfrak{S})$
2. *If* $\mathfrak{T} \in UNIVSOL_{\mathcal{M}}(\mathfrak{S})$ *then also* $core(\mathfrak{T}) \in UNIVSOL_{\mathcal{M}}(\mathfrak{S})$
3. *If* $UNIVSOL_{\mathcal{M}}(\mathfrak{S}) \neq \emptyset$*, then all* $\mathfrak{T} \in UNIVSOL_{\mathcal{M}}(\mathfrak{S})$ *have same core (up to renaming of NULLs), and the core of any universal solution is the smallest universal solution*

Computing the Core

- Easy Case: No tgds in M_τ
- Simple algorithm $COMPUTECORE(\mathcal{M})$
  - Assume $\mathfrak{S}$ has successful sequence with result $\mathfrak{T}$.
  - If $\mathfrak{T} = fail$, then also the output fail
  - Otherwise: remove facts as long as $M_{\sigma\tau}$ fulfilled.

Theorem

If chase not fails, then $COMPUTECORE(\mathcal{M})$ outputs core of universal solutions in polynomial time.

- Algorithm works as egds satisfactions preserved for subinstances
- More sophisticated methods needed in presence of tgds in M_τ

The Core

- Core has nice properties: Uniqueness
- But may be more costly to compute than universal canonical solution
- In the end: We want to use solution for QA—and for this canonical universal solutions suffice

Query Answering

Certain Answers

- Given mapping $\mathcal{M} = (\sigma, \tau, M_{\sigma\tau}, M_\tau)$
- Semantics of query answering specified as certain answer semantics

Definition

The **certain answers** of query Q over τ for given instance $\mathfrak{S}$ is defined as

$$cert_{\mathcal{M}}(Q, \mathfrak{S}) = \bigcap \{ \, Q(\mathfrak{T}) \mid \mathfrak{T} \in SOL_{\mathcal{M}}(\mathfrak{S}) \, \}$$

Certain Answers

- Given mapping $\mathcal{M} = (\sigma, \tau, M_{\sigma\tau}, M_\tau)$
- Semantics of query answering specified as certain answer semantics

Definition

The **certain answers** of query Q over τ for given instance $\mathfrak{S}$ is defined as

$$cert_{\mathcal{M}}(Q, \mathfrak{S}) = \bigcap \{ \, Q(\mathfrak{T}) \mid \mathfrak{T} \in SOL_{\mathcal{M}}(\mathfrak{S}) \, \}$$

- Definition does not tell how to actually compute the certain answers
- In many cases it is not necessary to compute all solutions to get certain answers

Certain Answers

- Given mapping $\mathcal{M} = (\sigma, \tau, M_{\sigma\tau}, M_\tau)$

- Semantics of query answering specified as certain answer semantics

Definition

The **certain answers** of query Q over τ for given instance $\mathfrak{S}$ is defined as

$$cert_{\mathcal{M}}(Q, \mathfrak{S}) = \bigcap \{\ Q(\mathfrak{T}) \mid \mathfrak{T} \in SOL_{\mathcal{M}}(\mathfrak{S})\ \}$$

Wake-up Question

Could it be the case that the certain answer set contains NULLS?

Certain Answers

- Given mapping $\mathcal{M} = (\sigma, \tau, M_{\sigma\tau}, M_{\tau})$
- Semantics of query answering specified as certain answer semantics

Definition

The **certain answers** of query Q over τ for given instance $\mathfrak{S}$ is defined as

$$cert_{\mathcal{M}}(Q, \mathfrak{S}) = \bigcap \{ \, Q(\mathfrak{T}) \mid \mathfrak{T} \in SOL_{\mathcal{M}}(\mathfrak{S}) \, \}$$

Wake-up Question

Could it be the case that the certain answer set contains NULLS?

Answer: No, because one can construct for any solution another with different NULLs, but in the certain answer set you have only tuples in all solutions.

Algorithmic Problems for Certain Answers

Problem: $\mathit{CERTAIN}_{\mathcal{M}}(Q, \mathfrak{S})$

Input: Source instance $\mathfrak{S}$ and tuple of elements $\vec{t} \in \mathit{DOM}(\mathfrak{S})$
Output: Answer whether $\vec{t} \in \mathit{certain}_{\mathcal{M}}(Q, \mathfrak{S})$

- Again, to guarantee tractability or even decidability one has to restrict the involved components
  - Constrain query language (e.g., from FOL to CQs)
  - Constrain dependencies (e.g., to weakly acyclic TGDs)

Proposition

There is an FOL query Q and a $\mathcal{M} = (\sigma, \tau, M_{\sigma\tau})$ s.t. $\mathit{CERTAIN}_{\mathcal{M}}(Q)$ is undecidable.

Answering Conjunctive Queries (CQs)

- Conjunctive queries (CQs)

$$Q(\vec{x}) = \exists \vec{y} \left(\alpha_1(\vec{x_1}, \vec{y_1}) \wedge \cdots \wedge \alpha_n(\vec{x_n}, \vec{y_n})\right)$$

- Unions of conjunctive queries (UCQs)

$$Q(\vec{x}) = CQ_1(\vec{x}) \vee \cdots \vee CQ_n(\vec{x})$$

- Crucial Property: (U)CQs are preserved under homomorphisms

Proposition

Let $h : \mathfrak{S} \xrightarrow{hom} \mathfrak{S}'$ and Q be a UCQ. Then: For all tuples $\vec{a}$ from the domain of $\mathfrak{S}$: If $\vec{a} \in Q(\mathfrak{S})$, then $h(\vec{a}) \in Q(\mathfrak{S}')$

If $\mathfrak{S}$ is complete, then the condition boils down to $Q(\mathfrak{S}) \subseteq Q(\mathfrak{S}')$

Follows easily from homomorphism definition (see Exercise)

As a corollary one immediately gets also preservation for certain query answering.

Proposition

Let $h : \mathfrak{S} \xrightarrow{hom} \mathfrak{S}'$ and Q be a UCQ. Then:

$$cert(Q, \mathfrak{S}) \subseteq cert(Q, \mathfrak{S}')$$

- Here we use a notion of certain answering for general DBs (independently from a DE scenario)

Definition

$$cert(Q, \mathfrak{S}) = \bigcap \{Q(\mathfrak{S}') \in Rep(\mathfrak{S})\}$$

Certain Answering UCQs

Theorem

Let $\mathcal{M} = (\sigma, \tau, M_{\sigma\tau}, M_\tau)$ be a mapping where M_τ is a union of egds and weakly acyclic tgds and let Q be a UCQ.

Then $CERTAIN_{\mathcal{M}}(Q, \mathfrak{S})$ can be solved in PTIME.

Proof Sketch

- Consider naive evaluation strategy Q_{naive}
  - Let $\mathfrak{T}$ arbitrarily chosen universal solution
  - Treat marked NULLS in $\mathfrak{T}$ as constants (i.e. $\bot = \bot$ is true but not $\bot = c$ or $\bot = \bot'$)
  - Calculate $Q(\mathfrak{T})$ under this perspective
  - and then eliminate all tuples from $Q(\mathfrak{T})$ containing a NULL
- Now one can show $certain_{\mathcal{M}}(Q, \mathfrak{S}) = Q_{naive}(\mathfrak{T})$.

Showing $certain_{\mathcal{M}}(Q, \mathfrak{S}) = Q_{naive}(\mathfrak{T})$

- We know that a universal solution $\mathfrak{T}$ can be constructed in polynomial time.
- For every $\mathfrak{T}' \in SOL_{\mathcal{M}}$ there is $\mathfrak{T} \xrightarrow{hom} \mathfrak{T}'$
- NULL-free tuples in $Q(\mathfrak{T}) \subseteq$
  $\bigcap_{\mathfrak{T}' \in SOL_{\mathcal{M}}}$ NULL-free tuples in $Q(\mathfrak{T}')$
- Answering FOL queries (and so of UCQs) computable in PTIME data complexity

QA for Other Classes of Queries

- Proof above used a simple strategy for certain answering by naive evaluation

Naive Evaluation Strategy

$$cert(\mathfrak{S}, Q) = Q_{naive}(\mathfrak{T})$$

where $\mathfrak{T}$ is a (universal) solution

- This strategy works also for Datalog programs as constraints for the target schema τ
  - Reason: Datalog programs are preserved under homomorphisms
  - Even if one adds inequalities, naive evaluation works
  - Hence certain answering is here in PTime

Rewritability

- Naive evaluation is a form of **rewriting**
- Fundamental method that re-appears in different areas of CS
- Rewrite a query w.r.t. a given KB into a new query that "contains" the knowledge of KB

- Challenges
  - Preserve the semantics in the rewriting process: ensure correctness (easy) and completeness (difficult)
  - The language of the output query is constraint to a "simple language" (so rewritability not always guaranteed)

Rewritability for DE

Definition (FOL Rewritability)

Let $\mathcal{M} = (\sigma, \tau, M_{\sigma\tau}, M_\tau)$ be a mapping and Q be a query over τ.

Then Q is said to be **FOL-rewritable** over the canonical universal solution under $\mathcal{M}$ if there is a FOL query Q_{rew} over τ^C such that

$$certain_{\mathcal{M}}(Q, \mathfrak{S}) = Q_{rew}(\mathfrak{T})$$

- Here $\tau^C = \tau \cup \{C\}$ where unary predicate C depicts all constants (not NULLs) in targets
- Works like a type predicate

Rewritability for DE

Definition (FOL Rewritability)

Let $\mathcal{M} = (\sigma, \tau, M_{\sigma\tau}, M_\tau)$ be a mapping and Q be a query over τ.

Then Q is said to be **FOL-rewritable** over the canonical universal solution under $\mathcal{M}$ if there is a FOL query Q_{rew} over τ^C such that

$$certain_{\mathcal{M}}(Q, \mathfrak{S}) = Q_{rew}(\mathfrak{T})$$

There is **one** rewriting for any given pair of source $\mathfrak{S}$ and universal solution $\mathfrak{T}$

- The known component is the mapping $\mathcal{M}$
- The unknown components are all pairs $(\mathfrak{S}, \mathfrak{T})$

Rewritability for DE

Definition (FOL Rewritability)

Let $\mathcal{M} = (\sigma, \tau, M_{\sigma\tau}, M_{\tau})$ be a mapping and Q be a query over τ.

Then Q is said to be **FOL-rewritable** over the canonical universal solution under $\mathcal{M}$ if there is a FOL query Q_{rew} over τ^{C} such that

$$certain_{\mathcal{M}}(Q, \mathfrak{S}) = Q_{rew}(\mathfrak{T})$$

If, in the definition, one talks about cores $\mathfrak{T}$ instead of universal solutions then Q is said to be **FOL rewritable over cores**

Theorem

FOL rewrit. over core $\quad\models\quad$ *FOL rewrit. over universal solution, but not vice versa.*

Rewritability for DE

Definition (FOL Rewritability)

Let $\mathcal{M} = (\sigma, \tau, M_{\sigma\tau}, M_{\tau})$ be a mapping and Q be a query over τ.

Then Q is said to be **FOL-rewritable** over the canonical universal solution under $\mathcal{M}$ if there is a FOL query Q_{rew} over τ^C such that

$$certain_{\mathcal{M}}(Q, \mathfrak{S}) = Q_{rew}(\mathfrak{T})$$

Example

- $Q(\vec{x})$: a conjunctive query
- Q_{rew}: $Q(\vec{x}) \wedge C(x_1) \wedge \cdots \wedge C(x_n)$
  This is actually the syntactic form of Q_{naive}
- The rewriting is even independent of $\mathcal{M}$
- So: (U)CQs are rewritable for any mapping

Adding Negations to Query Language

- Negations in query languages lead to lose of naive rewriting technique
- Even if one allows only negation in inequalities

Definition (Conjunctive Queries with inequalities $\mathrm{CQ}^{\neq}$)

A conjunctive query with inequalities is a query of the form

$$Q(\vec{x}) = \exists \vec{y} \left(\alpha_1(\vec{x_1}, \vec{y_1}) \wedge \cdots \wedge \alpha_n(\vec{x_n}, \vec{y_n}) \right)$$

where α_i is either an atomic relational formula or an inequality $z_i \neq z_j$.

Example (No Naive Evaluation Possible)

Source DB

Flight (	src,	dest,	airl,	dep	)
	paris	sant.	airFr	2320	
	paris	sant.	lan	2200	

Target DB

Routes(<u>fno</u>, src, dest)

Info(<u>fno</u>, dep, arr, airl)

- **Dependencies** $M_{\sigma\tau}$

$$Flight(src, dest, airl, dep) \longrightarrow$$
$$\exists fno\ \exists\ arr(Routes(fno, src, dest) \wedge Info(fno, dep, arr, airl))$$

- Any universal solution $\mathfrak{T}'$ contains solution τ **solutions**

$$\mathfrak{T} \quad = \quad \{\ Routes(\bot_1, paris, sant),\ Info(\bot_1, 2320, \bot_2, airFr),$$
$$Routes(\bot_3, paris, sant),\ Info(\bot_3, 2320, \bot_4, lan)\ \}$$

- Query $Q(x, z) = \exists y \exists y'(Routes(y, x, z) \wedge Routes(y', x, z) \wedge y \neq y')$
- $Q_{naive}(\mathfrak{T}') = \{(paris, sant)\}$ (for any universal solution $\mathfrak{T}'$)
- But: $cert(Q(x, z), \mathfrak{S})_{\mathcal{M}} = \emptyset$ because there is a solution

$$\mathfrak{T}'' \quad = \quad \{\ Routes(\bot_1, paris, sant),\ Info(\bot_1, 2320, \bot_2, airFr),$$
$$Info(\bot_1, 2320, \bot_2, lan)\ \}$$

Example (No Naive Evaluation Possible)

Source DB

Flight (	src,	dest,	airl,	dep	)
	paris	sant.	airFr	2320	
	paris	sant.	lan	2200	

Target DB

Routes(fno, src, dest)

Info(fno, dep, arr, airl)

- **Dependencies** $M_{\sigma\tau}$

$$\begin{array}{l} \mathit{Flight}(\mathit{src}, \mathit{dest}, \mathit{airl}, \mathit{dep}) \longrightarrow \\ \quad \exists \mathit{fno}\ \exists\ \mathit{arr}(\mathit{Routes}(\mathit{fno}, \mathit{src}, \mathit{dest}) \wedge \mathit{Info}(\mathit{fno}, \mathit{dep}, \mathit{arr}, \mathit{airl})) \end{array}$$

- Any universal solution $\mathfrak{T}'$ contains solution τ **solutions**

$$\begin{array}{rcl} \mathfrak{T} & = & \{\ \mathit{Routes}(\bot_1, \mathit{paris}, \mathit{sant}),\ \mathit{Info}(\bot_1, 2320, \bot_2, \mathit{airFr}), \\ & & \quad \mathit{Routes}(\bot_3, \mathit{paris}, \mathit{sant}),\ \mathit{Info}(\bot_3, 2320, \bot_4, \mathit{lan})\ \} \end{array}$$

- Query $Q(x, z) = \exists y \exists y' (\mathit{Routes}(y, x, z) \wedge \mathit{Routes}(y', x, z) \wedge y \neq y')$
- $Q_{\mathit{naive}}(\mathfrak{T}') = \{(\mathit{paris}, \mathit{sant})\}$ (for any universal solution $\mathfrak{T}'$)
- But: $\mathit{cert}(Q(x, z), \mathfrak{S})_{\mathcal{M}} = \emptyset$ because there is a solution

$$\begin{array}{rcl} \mathfrak{T}'' & = & \{\ \mathit{Routes}(\bot_1, \mathit{paris}, \mathit{sant}),\ \mathit{Info}(\bot_1, 2320, \bot_2, \mathit{airFr}), \\ & & \quad \mathit{Info}(\bot_1, 2320, \bot_2, \mathit{lan})\ \} \end{array}$$

Example (No Naive Evaluation Possible)

Source DB

Flight (	src,	dest,	airl,	dep	)
	paris	sant.	airFr	2320	
	paris	sant.	lan	2200	

Target DB

Routes(fno, src, dest)

Info(fno, dep, arr, airl)

- **Dependencies** $M_{\sigma\tau}$

$$\begin{aligned}&Flight(src, dest, airl, dep) \longrightarrow \\ &\quad \exists fno\ \exists\ arr(Routes(fno, src, dest) \wedge Info(fno, dep, arr, airl))\end{aligned}$$

- Any universal solution $\mathfrak{T}'$ contains solution τ **solutions**

$$\begin{aligned}\mathfrak{T} \quad = \quad \{\ & Routes(\bot_1, paris, sant),\ Info(\bot_1, 2320, \bot_2, airFr), \\ & Routes(\bot_3, paris, sant),\ Info(\bot_3, 2320, \bot_4, lan)\ \}\end{aligned}$$

- Query $Q(x, z) = \exists y \exists y'(Routes(y, x, z) \wedge Routes(y', x, z) \wedge y \neq y')$
- $Q_{naive}(\mathfrak{T}') = \{(paris, sant)\}$ (for any universal solution $\mathfrak{T}'$)
- But: $cert(Q(x, z), \mathfrak{S})_{\mathcal{M}} = \emptyset$ because there is a solution

$$\begin{aligned}\mathfrak{T}'' \quad = \quad \{\ & Routes(\bot_1, paris, sant),\ Info(\bot_1, 2320, \bot_2, airFr), \\ & Info(\bot_1, 2320, \bot_2, lan)\ \}\end{aligned}$$

Example (No Naive Evaluation Possible)

Source DB

Flight (	src,	dest,	airl,	dep	)
	paris	sant.	airFr	2320	
	paris	sant.	lan	2200	

Target DB

Routes(<u>fno</u>, src, dest)

Info(<u>fno</u>, dep, arr, airl)

- **Dependencies** $M_{\sigma\tau}$

$$\begin{array}{l} \mathit{Flight}(\mathit{src}, \mathit{dest}, \mathit{airl}, \mathit{dep}) \longrightarrow \\ \quad \exists \mathit{fno}\ \exists\ \mathit{arr}(\mathit{Routes}(\mathit{fno}, \mathit{src}, \mathit{dest}) \wedge \mathit{Info}(\mathit{fno}, \mathit{dep}, \mathit{arr}, \mathit{airl})) \end{array}$$

- Any universal solution $\mathfrak{T}'$ contains solution τ **solutions**

$$\begin{aligned} \mathfrak{T} \quad = \quad \{ & \mathit{Routes}(\bot_1, \mathit{paris}, \mathit{sant}),\ \mathit{Info}(\bot_1, 2320, \bot_2, \mathit{airFr}), \\ & \mathit{Routes}(\bot_3, \mathit{paris}, \mathit{sant}),\ \mathit{Info}(\bot_3, 2320, \bot_4, \mathit{lan}) \ \} \end{aligned}$$

- Query $Q(x, z) = \exists y \exists y'(\mathit{Routes}(y, x, z) \wedge \mathit{Routes}(y', x, z) \wedge y \neq y')$
- $Q_{naive}(\mathfrak{T}') = \{(\mathit{paris}, \mathit{sant})\}$ (for any universal solution $\mathfrak{T}'$)
- But: $\mathit{cert}(Q(x, z), \mathfrak{S})_{\mathcal{M}} = \emptyset$ because there is a solution

$$\begin{aligned} \mathfrak{T}'' \quad = \quad \{ & \mathit{Routes}(\bot_1, \mathit{paris}, \mathit{sant}),\ \mathit{Info}(\bot_1, 2320, \bot_2, \mathit{airFr}), \\ & \mathit{Info}(\bot_1, 2320, \bot_2, \mathit{lan}) \ \} \end{aligned}$$

Example (No Naive Evaluation Possible)

Source DB

Flight (	src,	dest,	airl,	dep	)
	paris	sant.	airFr	2320	
	paris	sant.	lan	2200	

Target DB

Routes(fno, src, dest)

Info(fno, dep, arr, airl)

- **Dependencies** $M_{\sigma\tau}$

$$Flight(src, dest, airl, dep) \longrightarrow$$
$$\exists fno\ \exists\ arr(Routes(fno, src, dest) \wedge Info(fno, dep, arr, airl))$$

- Any universal solution $\mathfrak{T}'$ contains solution τ **solutions**

$$\begin{aligned} \mathfrak{T} \quad = \quad & \{\ Routes(\bot_1, paris, sant),\ Info(\bot_1, 2320, \bot_2, airFr), \\ & Routes(\bot_3, paris, sant),\ Info(\bot_3, 2320, \bot_4, lan)\ \} \end{aligned}$$

- Query $Q(x, z) = \exists y \exists y'(Routes(y, x, z) \wedge Routes(y', x, z) \wedge y \neq y')$
- $Q_{naive}(\mathfrak{T}') = \{(paris, sant)\}$ (for any universal solution $\mathfrak{T}'$)
- But: $cert(Q(x, z), \mathfrak{S})_{\mathcal{M}} = \emptyset$ because there is a solution

$$\begin{aligned} \mathfrak{T}'' \quad = \quad & \{\ Routes(\bot_1, paris, sant),\ Info(\bot_1, 2320, \bot_2, airFr), \\ & Info(\bot_1, 2320, \bot_2, lan)\ \} \end{aligned}$$

CQ$^{\neq}$ is in coNP

- In case of CQ$^{\neq}$ one cannot even find a tractable means to answer them w.r.t. certain answer semantics

Theorem

Let $\mathcal{M} = (\sigma, \tau, M_{\sigma\tau}, M_{\tau})$ be a mapping where M_{τ} is the union of egds and weakly acyclic tgds, and let Q be a UCQ$^{\neq}$ query. Then:

$CERTAIN_{\mathcal{M}}(Q)$ is in coNP

Non-rewritability

- Generally it is not possible to decide whether rewritability holds

Theorem

For mappings without target constraints one can not decide whether a given FOL query is rewritable over the canonical solutions (over the core).

- Showing Non-FOL-rewritability can be done with locality tools
- Actually: One uses Hanf-locality of FOL
- Adaptation to DE setting

Not Covered

- Different semantics for query answering
  - Combinations of open-world (certain answers) and closed-word semantics
- Whole sub-field of mapping management
  - How to compose mappings
  - How to maintain mappings (e.g., w.r.t. consistency)
  - How to invert mappings: Get back source DB from target DB
- DE for non-relational DBs
  - e.g., DE for semi-structured data (XML)
  - different techniques needed

Exercise 5

Exercise 5.1 (4 Points)

Prove the folklore proposition that conjunctive queries are preserved under homomorphisms, i.e., show that if there is a homomorphism h from a DB instance $\mathfrak{I}$ to a DB instance $\mathfrak{I}'$, then for any CQ $\phi(\vec{x})$:

$$\{h(\vec{d}) \mid \vec{d} \in \mathit{ans}(\phi(\vec{x}), \mathfrak{I})\} \subseteq \mathit{ans}(\phi(\vec{x}), \mathfrak{I}')$$

Exercise 5.2 (6 Points)

1. Prove that every finite graph has a core (2 points)
2. Prove that two cores of the same graph are isomorphic. (4 points)