UNIVERSITÄT ZU LÜBECK
INSTITUT FÜR INFORMATIONSSYSTEME

Özgür L. Özçep

Database Repairs

Lecture 12
2 July, 2020

Informationssysteme CS4130
(Summer 2020)

References

- The paper which started it all

  Lit: M. Arenas, L. Bertossi, and J. Chomicki. Consistent query answers in inconsistent databases. In Proceedings of the eighteenth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, PODS '99, pages 68–79, New York, NY, USA, 1999. ACM.

- Recent overview given by one of the founders at the gems of PODS session 2019

  Lit: L. Bertossi. Database repairs and consistent query answering: Origins and further developments. In Proceedings of the 38th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS '19, pages 48–58, New York, NY, USA, 2019. Association for Computing Machinery.

- The core ideas in a small textbook

  Lit: L. Bertossi. Database Repairing and Consistent Query Answering. Morgan & Claypool Publishers, 2011.

References (continued)

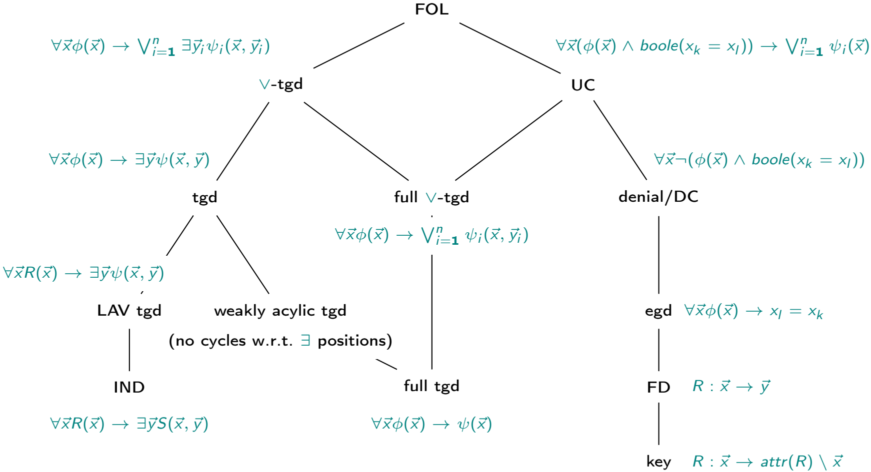
- Slides that are adapted and extended in the following
  Lit: Phokion Kolaitis: Coping with Inconsistent Databases Semantics. Algorithms, and Complexity. Invited talk given at the International Conference on Theory and Applications of Satisfiability Testing (SAT 2016), 2016.
- Another nice overview on recent developments (also used here)
  Lit: J. Wijsen. Foundations of query answering on inconsistent databases. SIGMOD Rec., 48(3):6–16, Dec. 2019.
- Will occasionally draw connections to belief revision
- Own more extensive additions prefixed by "Ö.:"

Basic Notions

Integrity constraints (ICs)

- For many (DB) purposes sufficient
  - tuple-generating dependencies (TGDs)
  - equality generating dependencies (EGDs) over the same schema (Compare lectures on data integration and data exchange)
- Special cases of EGDs
  - Functional dependency (FD) $R : X \to Y$
    If two tuples in R agree on X then they agree on Y .
  - Key Constraint
    $R : X \longrightarrow Y$ and $Y = \mathit{Attributes}(R) \setminus X$.
- There are further ICs ...

Common Classes of ICs (Wijsen 19))



Convention: $\phi(\vec{x})$ contains exactly variables in $\vec{x}$; $\psi(\vec{x}), \psi_i(\vec{x})$ may contain subvector of $\vec{x}$

Inconsistent Databases

- Σ: Set of integrity constraints (ICs)
- Inconsistent DB $\mathfrak{A}$: Does not satisfy Σ, for short: $\mathfrak{A} \not\models \Sigma$
- Context of and reasons for inconsistent DBs
  - Lacking support of (some) ICs
  - Heterogeneous sources with different ICs in data integration
  - Data warehouse/ETL: data to be cleaned up beforehand

Coping with Inconsistent DBs

- Data cleaning: Make DB consistent
  - By say adding, deleting, updating rows
  - Chase-procedure can be understood as systematic cleaning (compare notion of Null-based repairs)
  - In industrial-strength practice: ad-hoc, based on heuristics, for specific domains only
  - Main approach in industry (e.g., IBM InfoSphere Quality Stage, Microsoft DQS)

- Database repairs: repair (only) virtually & provide DB services
  - In particular: Enable consistent query answering (cqa) over inconsistent DBs
  - Parameters
    - Kinds of allowed repair operations (not discussed in detail)
    - Minimality notion for repairs (see next slides)
  - Compare also: Paraconsistent logics

Ö.: Relation to Belief Revision

- See discussion on database update in previous lectures
- Can consider DB as a model (corresponding to a complete theory)
- Model describable as belief base B
- Idea: Use Σ as trigger info (multiple revision):
- Data cleaning becomes $B * \Sigma$

- Hence there is actually a theoretically well investigated theory (belief revision) that could be used for data cleaning
- Why not used?
  - Mainly due to worst-case complexity
  - The same holds, imho, also for data repair (only prototypes available, not industrial-strength software)

Definition (Database Repair, (Arenas/Bertossi/Chomicki 99))

DB $\mathfrak{B}$ is a repair of inconsistent DB $\mathfrak{A}$ w.r.t. ICs Σ iff

1. $\mathfrak{B} \models \Sigma$
2. $\mathfrak{B}$ is minimally different from $\mathfrak{A}$

- No unique definition for being minimally different
- In DB community the following instances were investigated
  - Classical set-based repair
  - Cardinality-based repairs
  - Attribute-based repairs (in particular: null-based repairs)
  - Preferred repairs

Ö.: The problem of minimality in revision

- The idea of minimal repair does not work for belief sets
- Symmetric difference $X \oplus Y = X \setminus Y \cup Y \setminus X$
- Assume a propositional logic with AGM-constraints

Proposition (Rott 2000)

Assume that

- X *is a consistent belief set with* $\neg\alpha \in X$
- $Y_1 \neq Y_2$ *are belief sets with* $\alpha \in Y_1 \cap Y_2$.

Then $X \oplus Y_1$ *und* $X \oplus Y_2$ *are not comparable w.r.t.* $\subseteq$, *i.e.*
$X \oplus Y_1 \not\subseteq X \oplus Y_2$ *and* $X \oplus Y_2 \not\subseteq X \oplus Y_1$

- Hence all repairs used for revision $X * \alpha$ would be $\oplus$-minimal!

Lit: H. Rott. Two dogmas of belief revision. The Journal of Philosophy, 97(9):503–522, 2000.

Ö.: The problem of minimality in revision (continued)

- DBs are rather belief bases (not deductively closed)
- Hence Rott's observation not applicable

- Under what kinds of closures non-trivial minimality notion ensured?
- Disjunctive closure allows minimality considerations

  Lit: Özgür L. Özcep. Semantische Integration durch Reinterpretation - ein formales Modell, 2009, PhD thesis, http://www.sub.uni-hamburg.de/opus/volltexte/2010/4428/pdf/oezcepDiss2009.pdf (in German)

  Lit: Özgür L. Özcep. Representation Theorems in Computer Science - A Treatment in Logic Engineering. Springer, 2019.

Definition (Classical Subset Repair)

For a set of integrity constraints Σ and an inconsistent database $\mathfrak{A}$ we say that $\mathfrak{B}$ is a classical subset-repair of $\mathfrak{A}$ w.r.t. Σ iff

1. $\mathfrak{B} \subseteq \mathfrak{A}$
2. $\mathfrak{B} \models \Sigma$
3. and there is no $\mathfrak{B}'$ with properties 1. and 2. and $\mathfrak{B} \subsetneq \mathfrak{B}'$

Example (Classical subset repairs)

- ICs and database

$$\begin{aligned}\Sigma &= \{\, \forall x \forall y \forall z (\, (R(x,y) \wedge R(x,z)) \rightarrow y = z) \,\} \\ \mathfrak{A} &= \{R(a_1,b_1), R(a_1,b_2), R(a_2,b_1), R(a_2,b_2)\}\end{aligned}$$

- Classical subset repairs

$$\begin{aligned}\mathfrak{B}_1 &= \{R(a_1,b_1), R(a_2,b_1)\} \\ \mathfrak{B}_2 &= \{R(a_1,b_1), R(a_2,b_2)\} \\ \mathfrak{B}_3 &= \{R(a_1,b_2), R(a_2,b_1)\} \\ \mathfrak{B}_4 &= \{R(a_1,b_2), R(a_2,b_2)\}\end{aligned}$$

- Exponentially many repairs in general

Minimal repairs (Wijsen 19)

- Capture notions of minimal repairs with relation $\leq^{\Sigma}_{\mathfrak{A}}$

Definition (Binary repair relation $\leq^{\Sigma}_{\mathfrak{A}}$ (informal))

$\mathfrak{B}_1 \leq^{\Sigma}_{\mathfrak{A}} \mathfrak{B}_2$ iff repair of $\mathfrak{A}$ into $\mathfrak{B}_1$ requires no more effort than repair of $\mathfrak{A}$ into $\mathfrak{B}_2$

- Minimal repairs $Min_{\leq^{\Sigma}_{\mathfrak{A}}}(\mathfrak{A})$ of $\mathfrak{A}$:

$$\{\mathfrak{B} \mid \mathfrak{B} \models \Sigma \text{ and there is no } \mathfrak{B}' \text{ s.t.: } \mathfrak{B}' \models \Sigma \text{ and } \mathfrak{B}' <^{\Sigma}_{\mathfrak{A}} \mathfrak{B}\}$$

- What properties to require of $<^{\Sigma}_{\mathfrak{A}}$?
  - Acyclicity

Minimal repairs w.r.t. some order

- Symmetric-difference order

$$\mathfrak{B}_1 \leq_{\mathfrak{A},\oplus} \mathfrak{B}_2 \text{ iff } \mathfrak{B}_1 \oplus \mathfrak{A} \subseteq \mathfrak{B}_2 \oplus \mathfrak{A}$$

  - Transitive, reflexive, antisymmetric (partial order)
  - Note: $\mathfrak{B}_1, \mathfrak{B}_2$ not necessarily subsets of $\mathfrak{A}$
- Cardinality order

$$\mathfrak{B}_1 \leq_{\mathfrak{A},c} \mathfrak{B}_2 \text{ iff } |\mathfrak{B}_1 \oplus \mathfrak{A}| \leq |\mathfrak{B}_2 \oplus \mathfrak{A}|$$

  - Transitive and reflexiv (pre-order)

Definition (General Subset [Superset] Repair)

$\mathfrak{B}$ is a general subset-repair [superset-repair] of $\mathfrak{A}$ w.r.t. Σ iff it is a $\leq^{\Sigma}_{\mathfrak{A}}$-minimal repair (for some relation $\leq^{\Sigma}_{\mathfrak{A}}$) and $\mathfrak{B} \subseteq \mathfrak{A}$ [$\mathfrak{B} \supseteq \mathfrak{A}$].

Preferred/Prioritized repairs

- Assume Σ consists of FDs
- Inconsistent prioritizing db $(\mathfrak{A}, \succ)$
  - Intuition: $f \succ g$ iff f is fact prioritized over fact g
  - $\succ$ acyclic with: $f \succ g$ entails $\{f, g\} \not\models \Sigma$.

Definition (Prioritization-order and g-repair)

- For $\mathfrak{B}_1, \mathfrak{B}_2 \subseteq \mathfrak{A}$, $\mathfrak{B}_1 \models \Sigma$, and $\mathfrak{B}_2 \models \Sigma$:
  $\mathfrak{B}_1 \leq_{\mathfrak{A},\succ} \mathfrak{B}_2$ iff
  for every $g \in \mathfrak{B}_2 \setminus \mathfrak{B}_1$ there is $f \in \mathfrak{B}_1 \setminus \mathfrak{B}_2$ s.t. $f \succ g$.
- $\mathfrak{B}$ is a globally optimal repair/g-repair iff $\mathfrak{B} \in Min_{\leq_{\mathfrak{A},\succ}}(\mathfrak{A})$.

Lit: S. Staworko, J. Chomicki, and J. Marcinkowski. Prioritized repairing and consistent query answering in relational databases. Annals of Mathematics and Artificial Intelligence, 64(2):209–246, 2012.

Example

- DB instance over $R(X, Y)$: $\{R(a, b), R(c, b), R(c, d)\}$
  Constraints Σ: $\{R : X \rightarrow Y, R : Y \rightarrow X\}$

- $\oplus$-repairs:
  - $\{R(a, b), R(c, d)\}$
  - $\{R(c, b)\}$

- C-repair
  - $\{R(a, b), R(c, d)\}$

- G-repair for $\succ$ with $R(c, b) \succ R(a, b)$ and $R(c, b) \succ R(c, d)$
  - $\{R(c, b)\}$

Null-based repairs: tuple level

- Input DB $\mathfrak{A}$

  $\{S(c_1, r_1, i_1), S(c_2, r_2, i_2), S(c_3, r_3, i_3),$ Supply(Company,Receiver,Item)

  $A(i_1, 50), A(i_2, 30)\}$ Article(Item, Cost)

- Constraints Σ: $\{\forall x \forall y \forall z [S(x, y, z) \rightarrow \exists v A(z, v)]\}$ (an IND)

- Repairs
  - $\mathfrak{B}_1$: delete $S(c_3, r_3, i_3)$ from $\mathfrak{A}$
  - $\mathfrak{B}_2$: Insert $(i_3, NULL)$ into relation A.

Lit: L. E. Bertossi and L. Bravo. Consistency and trust in peer data exchange systems. Theory Pract. Log. Program., 17(2):148–204, 2017.

Null-based repairs: attribute level

- Input DB $\mathfrak{A}$
  - $\{R(a_4, a_3), R(a_2, a_1), R(a_3, a_3),$
    $S(a_4), S(a_2), S(a_3)\}$

- Constraints Σ: $\{\neg\exists x \exists y (S(x) \wedge R(x, y) \wedge S(y)\}$ (a DC)

- Repairs
  - $\mathfrak{B}_1 : \{\ R(a_4, a_3), R(a_2, a_1), R(a_3, a_3),$
    $S(a_4), S(a_2), S(\mathit{NULL})\}$
  - $\mathfrak{B}_2 : \{\ R(a_4, \mathit{NULL}), R(a_2, a_1), R(a_3, \mathit{NULL}),$
    $S(a_4), S(a_2), S(a_3)\}$

- Note: Null prevents a join

Lit: Bertossi and L. Li. Achieving data privacy through secrecy views and null-based virtual updates. IEEE Transactions on Knowledge and Data Engineering, 25(5):987–1000, May 2013.

Consistent Query Answering (CQA)

Definition (Arenas, Bertossi, Chomicki 99)

The consistent answers of Q on $\mathfrak{A}$ w.r.t. Σ is the set

$$cqa_r(Q, \mathfrak{A}, \Sigma) = \bigcap \{ Q(\mathfrak{B}) \mid \mathfrak{B} \text{ is an } r\text{-repair of } \mathfrak{A} \text{ w.r.t. } \Sigma \}$$

- Follows the usual pattern for dealing with incomplete information
- Compare this with certain answers in data exchange and OBDA
- When clear from context repair type r not mentioned.

Example (Consistent query answering)

- ICs, database, and classical subset-repairs

$$\Sigma = \{ \forall x \forall y \forall z((R(x,y) \wedge R(x,z)) \rightarrow y = z) \}$$

$$\mathfrak{A} = \{R(a_1,b_1), R(a_1,b_2), R(a_2,b_1), R(a_2,b_2)\}$$

$$\begin{aligned}
\mathfrak{B}_1 &= \{R(a_1,b_1), R(a_2,b_1)\}\\
\mathfrak{B}_2 &= \{R(a_1,b_1), R(a_2,b_2)\}\\
\mathfrak{B}_3 &= \{R(a_1,b_2), R(a_2,b_1)\}\\
\mathfrak{B}_4 &= \{R(a_1,b_2), R(a_2,b_2)\}
\end{aligned}$$

- Queries and answers

$$\begin{aligned}
Q_1(x) &= \exists y R(x,y)\\
cqa(Q_1, \mathfrak{A}, \Sigma) &= \{a_1, a_2\}
\end{aligned}$$

$$\begin{aligned}
Q_2(x) &= \exists z R(z,x)\\
cqa(Q_1, \mathfrak{A}, \Sigma) &= \emptyset
\end{aligned}$$

Complexity and Dichotomies

Two main decision problems

Definition (Decision problem $CERTAINTY_r(Q, \Sigma)$)

Boolean query Q and Σ fixed.

- Input: database $\mathfrak{A}$
- Output: $cqa_r(Q, \Sigma, \mathfrak{A})$

Complexity ranges from polynomial time computability to undecidability

Definition (Decision problem $REPAIR_r(\Sigma)$)

Σ fixed.

- Input: databases $\mathfrak{A}$, $\mathfrak{B}$
- Output: Is $\mathfrak{B}$ an r-repair of $\mathfrak{A}$ w.r.t. Σ?

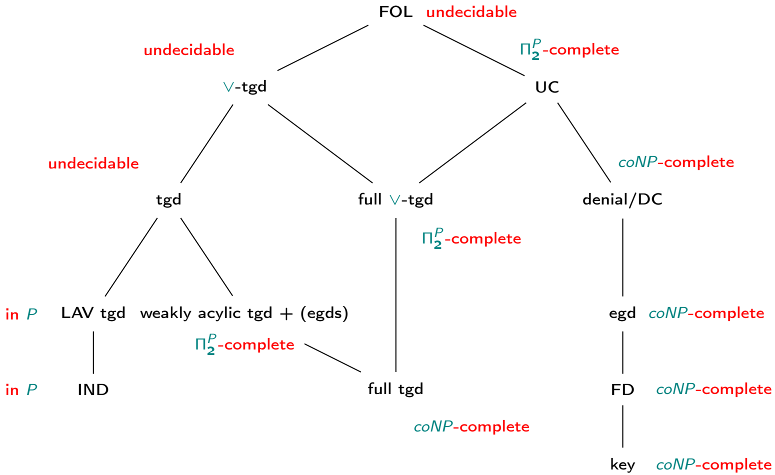
Complexity ranges from polynomial time computability to coNP-completeness

Definition

- Consistent query answering for class of queries $\mathcal{Q}$ and class of integrity constraints $\mathcal{IC}$ is in complexity class C iff $CERTAINTY(Q, \Sigma)$ is in C for all $Q \in \mathcal{Q}$ and all $\Sigma \subseteq \mathcal{IC}$.

- Consistent query answering for $\mathcal{Q}$ and $\mathcal{IC}$ is C-complete iff it is in C and $CERTAINTY(Q, \Sigma)$ is C-complete for some $Q \in \mathcal{Q}$ and some $\Sigma \subseteq \mathcal{IC}$.

Complexities for $CERTAINTY_{\oplus}(Q, \Sigma)$ for $Q \in CQs$

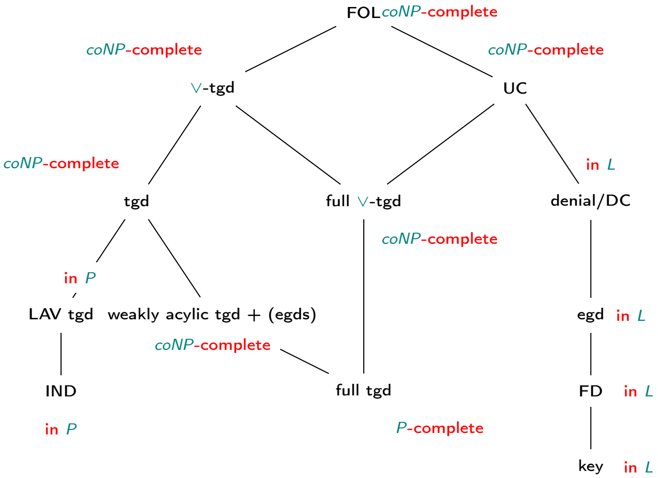


Lit: S. Arming, R. Pichler, and E. Sallinger. Complexity of repair checking and consistent query answering. ICDT, pages 21:1–21:18, 2016.

What do the complexity results tell us?

- Even for very common queries and ICs untractable problems (coNP-complete problems)
- But: By definition this means only that there are some intractable (coNP-problems); does not say anything about tractable-untractable boundary
- Tackle this with dichtomy/trichotomy theorems

Complexities for $REPAIR_{\oplus}(\Sigma)$ for $Q \in CQs$



Lit: S. Arming, R. Pichler, and E. Sallinger. Complexity of repair checking and consistent query answering. ICDT, pages 21:1–21:18, 2016.

Complexity of CQA: An Illustration

- R, S: binary relations with first argument as keys:
  $\Sigma = \{R(u, v) \wedge R(u, w) \rightarrow v = w, S(u, v) \wedge S(u, w) \rightarrow v = w\}$

- $Q_1 = \exists x, y, z(R(x, y) \wedge S(y, z))$
  - $CERTAINTY(Q_1, \Sigma) \in P$
  - Even FOL-rewritable
    $\exists x, y, z(R(x, y) \wedge S(y, z) \wedge \forall y'[R(x, y') \rightarrow \exists z' S(y', z')])$

- $Q_2 = \exists x, y(R(x, y) \wedge S(y, x))$
  - $CERTAINTY(Q_2, \Sigma) \in P$
  - but not FOL-rewritable

- $Q_3 = \exists x, y, z(R(x, y) \wedge S(z, y))$
  - $CERTAINTY(Q_2, \Sigma)$ $coNP$-complete

- Note: All queries are CQs but of different types
  $\Longrightarrow$ Classification with di-/trichotomies

Descriptive Complexity and Rewriting

- Instead of computational complexities can also use descriptive complexity

- Remember notion of logic $\mathcal{L}$ capturing a complexity class and notion of rewriting.
- $CERTAINTY(Q, \Sigma)$ is expressible in $\mathcal{L}$, alias is $\mathcal{L}$-rewritable, iff there is $Q_{rew} \in \mathcal{L}$ s.t.

$$cqa(Q, \mathfrak{A}) = true \text{ iff } \mathfrak{A} \models Q_{rew}$$

- Most attractive: $\mathcal{L} = FOL$

Famous open question

Dichotomy-Conjecture

For every set Σ of primary keys, for every query Q that is a disjunction of Boolean CQs, $CERTAINTY(Q, \Sigma)$ is either in P or $coNP$-complete.

Reminder: Dichotomies

Theorem (Ladner 1975)

If $P \neq NP$ then there is a decision problem Q s.t.

- *Q is in NP but not in P*
- *Q is not NP-complete*

(Similar results for *coNP* obtainable.)

The fine structure of *coNP*

coNP-complete
~~not *coNP*-complete, not in *P*~~
P

$CERTAINTY(Q, \Sigma)$

Dichotomy conjecture

Progress towards the Conjecture or: 3 is more than 2

Theorem (Trichotomy (Koutris/Wijsen 19))

For every set Σ of primary keys and self-join-free Boolean CQs Q, $CERTAINTY(Q, \Sigma)$ is either in FOL or L-complete or $coNP$-complete.

The proof moreover reveals:

1. Membership in L shown by rewriting into symmetric stratified Datalog with aggregation.
2. Membership in tractable classes ($FOL \cup L$) iff joins are foreign-key-primary-key joins

   $\Longrightarrow$ most SPJ queries are tractable

Lit: P. Koutris and J. Wijsen. Consistent query answering for primary keys in logspace. In: ICDT 2019, pages 23:1–23:19, 2019

Connections to CSPs

- Unexpected connection between Consistent Query Answering and Constraint Satisfaction Problems (CSPs).
- Shows that the dichotomy conjecture for *CERTAINTY* likely not trivial (as proof of CSP dichotomy highly non-trivial)

Theorem (Fontaine 2015 (informal))

$CERTAINTY(Q, \Sigma)$ dichotomy (Conjecture) $\models$ CSP dichotomy under specific conditions

Lit: G. Fontaine. Why is it hard to obtain a dichotomy for consistent query answering? ACM Trans. Comput. Log., 16(1):7:1–7:24, 2015.

Constrain Satisfaction Problems

- Traditionally (as used in AI research) considered as subclass of search problems with states and a goal test

Definition

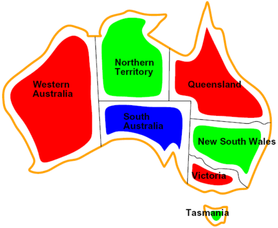
- $D =$ domain
- $\Gamma =$ Constraint language = set of relations $\{R_i\}_{i \in I}$ over D

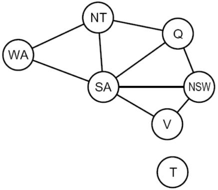
The constraint satisfaction problem $CSP(\Gamma)$ associated with Γ is the problem defined by instances of the form (V, D, C) where

- $V =$ set of variables
- $C =$ set of constraints $(\vec{v}, R_i)$ (notated also as $R_i(\vec{v})$) with
  - $R_i \in \Gamma$ an n-ary relation
  - $\vec{v} = (v_1, \ldots, v_n)$ with $v_i \in V$ is the scope (state variables)

A solution to the problem (the goal) is a mapping $\phi : V \longrightarrow D$ fulfilling all constraints, i.e., $\phi(\vec{v}) \in R_i$ for all $(\vec{v}, R_i) \in C$.

Example: Map Colouring





- $D = \{red, green, blue\}$
- $\Gamma = \{R_1\} = \{\ \{(x, y) \in D \times D \mid x \neq y\}\ \}$
- $V = \{WA, NT, Q, NSW, VI, SA, T\}$
- $C = \{\ WA \neq NT, WA \neq SA, NA \neq SA, NA \neq Q, SA \neq Q,$
  $SA \neq NSW, SA \neq VI, Q \neq NSW, NSW \neq VI\}$

- A solution

$$\phi : \quad WA \mapsto red, NT \mapsto green, SA \mapsto blue, Q \mapsto red,$$
$$NSW \mapsto green, VI \mapsto red, SA \mapsto blue, T \mapsto green$$

CSP = Finding a homomorphism

Definition

Given a $CSP(\Gamma)$ instance (V, D, C) the associated homomorphism instance $h : \mathfrak{S} \xrightarrow{hom} \mathfrak{T}$ is defined by

- source structure $\mathfrak{S}$ = (variables, scopes)
  $\mathfrak{S} = (V, R_i^{\mathfrak{S}} = \{\vec{v} \mid (\vec{c}, R) \in C\})_{i \in I}$
- target structure $\mathfrak{T}$ = (values, constraint relations)
  $\mathfrak{T} = \{D, R_i^{\mathfrak{T}} = R_i\}_{i \in I}$
- homomorphism $h =$ solution ϕ

Each homomorphism problem $?\exists h : \mathfrak{A} \xrightarrow{hom} \mathfrak{B}$ gives rise to a CSP-instance: Generate constraint $(\vec{v}, R^{\mathfrak{B}})$ for each $\vec{v} \in R^{\mathfrak{A}}$.

Lit: T. Feder and M. Y. Vardi. The computational structure of monotone monadic SNP and constraint satisfaction: A study through datalog and group theory. SIAM J. Comput., 28:57–104, 199

Lit: P. G. Kolaitis and M. Y. Vardi. A Logical Approach to Constraint Satisfaction, pages 125–155. Springer, 2008

Dichotomies for CSP

- Complexity of finding solutions depends on Γ
- Dichotomy theorem for subclass of conservative CSPs (c-CSP) which are CSPs with additionally:
  - For each variable $v \in V$ one has a unary relation $R_v \subseteq D$
  - Solution ϕ must fullfil $\phi(v) \in R_v$.

Theorem (Dichotomy for conservative CSPs, (Bulatov 11))

For each Γ the problem $c - CSP(\Gamma)$ is either in P or NP-complete.

- Proof relies on algebraic machinery based on polymorphisms (Barto et al., 17)

Lit: A. Bulatov. Complexity of conservative constraint satisfaction problems. ACM Trans. Comput. Log., 12(4):24:1–24:66, 2011.

Lit: L. Barto, A. Krokhin, and R. Willard. Polymorphisms, and How to Use Them. In A. Krokhin and S. Zivny, editors, The Constraint Satisfaction Problem: Complexity and Approximability, volume 7 of Dagstuhl Follow-Ups, pages 1–44. Schloss Dagstuhl,Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 2017.

Connections to CSPs

Theorem (Fontaine 2015 formal)

- $\mathit{CERTAINTY}(Q, \Sigma)$ *dichotomy (Conjecture)* $\models$ *CSP dichotomy where*
  - Σ *is a finite set of Horn constraints (full tgd with atomic head)*
  - Q *is a union of Boolean CQs*
- $\mathit{CERTAINTY}(Q, \Sigma)$ *dichotomy (Conjecture)* $\models$ *c-CSP dichotomy*

Lit: G. Fontaine. Why is it hard to obtain a dichotomy for consistent query answering? ACM Trans. Comput. Log., 16(1):7:1–7:24, 2015.

Alternative route for proving the conjecture

- Breakthrough result in 2017 for wider class of CSP

Theorem (Dichotomy for CSPs, (Bulatov 17))

For each Γ the problem $CSP(\Gamma)$ is either in P or NP-complete.

- Proof strategy for conjecture: Show that it is entailed by Bulatov's dichotomy for CSPs.

Lit: A. A. Bulatov. A dichotomy theorem for nonuniform CSPs. In C. Umans, editor, 58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017, pages 319–330. IEEE Computer Society, 2017.

Systems

Theory and Practice

- Theory of database repairs a theoretical foundation for coping with inconsistent DBs
- Extensively studied in last 20 years
- Only marginally used in data cleaning (few examples given by (Bertossi 19)
- Industrial-strength cqa-systems have yet to be developed

Lit: L. Bertossi. Database repairs and consistent query answering: Origins and further developments. In Proceedings of the 38th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS '19, pages 48–58, New York, NY, USA, 2019.

Prototypes (optional slide)

- Hippo

  Lit: J. Chomicki, J. Marcinkowski, and S. Staworko. Hippo: A system for computing consistent answers to a class of sql queries. In EDBT 2004, pages 841–844, Springer, 2004.

- ConQuer

  Lit: A. Fuxman and R. J. Miller. First-order query rewriting for inconsistent databases. J. Comput. Syst. Sci., 73(4):610–635, 2007.

- ConsEx

  Lit: M. C. Marileo and L. E. Bertossi. The consistency extractor system: Answer set programs for consistent query answering in databases. Data Knowl. Eng., 69(6):545–572, 2010.

- EQUIP

  Lit: P. G. Kolaitis, E. Pema, and W. Tan. Efficient querying of inconsistent databases with binary integer programming. PVLDB, 6(6):397–408, 2013.

Feature Overview (Optional Slide)

System	Constraints	Queries	Method
Hippo	UC	Projection-free with $\cup$ and $\setminus$	Direct Alg.
ConQuer	Keys	CQs	FO-rewriting
ConsEx	UC + IND with acyclicity	Datalog$^{\neg}$	ASP
EQUIP	Keys IND	CQs	Reduction to ILP

Kolaitis' vision for a comprehensive system (Optional Slide)

Module-based system depending on complexity of $cqa(Q, \Sigma)$

- Preprocessing: Determine evaluation strategy based on complexity classification for $cqa(Q, \Sigma)$
- Processing
  - Module A: FOL-rewriting + DB engine if $cqa(Q, \Sigma)$ FOL rewritable
  - Module B: Direct Algorithm or reduction to LP if $cqa(Q, \Sigma) \in P \setminus FOL$
  - Module C: Reduction to ILP (oe SAT or QBF) if $cqa(Q, \Sigma) \in coNP$.

Lifting to ontologies

- Whole idea can of course be lifted also to ontologies
- For a recent contribution for DL-Lite ontologies see (Bienvenu et al 2019)

Lit: M. Bienvenu, C. Bourgaux, and F. Goasdoué. Computing and explaining query answers over inconsistent dl-lite knowledge bases. J. Artif. Int. Res., 64(1):563–644, Jan. 2019.

Synopsis and Outlook (Kolaitis + Ö.)

- Database repair meeting point for database, logic, and complexity
- Further dichotomies; main conjecture open
- Much to be done for industrial-strength systems for different types of repairs and classes of constraints
  - Promising approach: Combine database engines with SAT solvers and QBF solvers
  - Ö: This fits to the general trend of "SQL-incorporates it all'
    - SQL now supports arrays
    - SQL is going to give support for streams ...
- Ö: Systematic study of connections to belief revision
- Ö: Are there dichotomies also for belief revision?