Uncertain Data Management
Open-World Query Answering

Antoine Amarilli[1], Silviu Maniu[2]

November 28th, 2016

[1]Télécom ParisTech

[2]LRI

Table of contents

Incompleteness

- We have an instance I
- The true state of the world is W
- We may have $I \neq W$

Incompleteness

- We have an instance I
- The true state of the world is W
- We may have $I \neq W$
- I may be correct: $I \subseteq W$
- I may be complete: $W \subseteq I$

Incompleteness

- We have an instance I
- The true state of the world is W
- We may have $I \neq W$
- I may be correct: $I \subseteq W$
- I may be complete: $W \subseteq I$

$\rightarrow$ Today, I is correct but not complete

Incompleteness and query evaluation

- We know: evaluate a query Q on I
- We want: evaluate Q on W

Incompleteness and query evaluation

- We know: evaluate a query Q on I
- We want: evaluate Q on W
- We don't have W

$\rightarrow$ What can we do?!

Constraints to the rescue!

- We know that $I \subseteq W$ (correct)
- We know that W satisfies some logical constraints Θ

Constraints to the rescue!

- We know that $I \subseteq W$ (correct)
- We know that W satisfies some logical constraints Θ
- Q is entailed if $W \models Q$ for all $W \supseteq I$ such that $W \models \Theta$

Constraints to the rescue!

- We know that $I \subseteq W$ (correct)
- We know that W satisfies some logical constraints Θ
- Q is entailed if $W \models Q$ for all $W \supseteq I$ such that $W \models \Theta$

Definition (Open-World Query Answering – OWQA)
Given an instance I, Boolean CQ Q, and constraints Θ, decide whether all $W \supseteq I$ that satisfy Θ also satisfy Q.

Constraints to the rescue!

- We know that $I \subseteq W$ (correct)
- We know that W satisfies some logical constraints Θ
- Q is entailed if $W \models Q$ for all $W \supseteq I$ such that $W \models \Theta$

Definition (Open-World Query Answering – OWQA)
Given an instance I, Boolean CQ Q, and constraints Θ, decide whether all $W \supseteq I$ that satisfy Θ also satisfy Q.

Combined complexity. Input is I, Θ, Q

Data complexity. Input is I

Example

Relation Class in I

date	teacher	resp	name	num
2016-11-28	Antoine	Fabian	Uncert. Data Mgmt	2
2016-12-05	Antoine	Fabian	Uncert. Data Mgmt	3

Book in I

date	room	prof
2016-12-05	E242	John

Example

Relation Class in I

date	teacher	resp	name	num
2016-11-28	Antoine	Fabian	Uncert. Data Mgmt	2
2016-12-05	Antoine	Fabian	Uncert. Data Mgmt	3

Book in I

date	room	prof
2016-12-05	E242	John

$$\Theta : \forall\, \mathrm{date},\ \mathrm{prof},\ r,\ n,\ i,\ \mathrm{Class}(\mathrm{date}, \mathrm{prof}, r, n, i) \Rightarrow \exists \mathrm{room},\ \mathrm{Book}(\mathrm{date}, \mathrm{room}, \mathrm{prof})$$

"Every class has a booking."

Example

Relation Class in I

date	teacher	resp	name	num
2016-11-28	Antoine	Fabian	Uncert. Data Mgmt	2
2016-12-05	Antoine	Fabian	Uncert. Data Mgmt	3

Book in I

date	room	prof
2016-12-05	E242	John

$$\Theta : \forall\, \mathrm{date},\ \mathrm{prof},\ \mathrm{r},\ \mathrm{n},\ \mathrm{i},\ \mathrm{Class}(\mathrm{date}, \mathrm{prof}, \mathrm{r}, \mathrm{n}, \mathrm{i}) \Rightarrow \exists \mathrm{room},\ \mathrm{Book}(\mathrm{date}, \mathrm{room}, \mathrm{prof})$$

"Every class has a booking."

$$Q : \exists t\, r\ \mathrm{Book}(\text{“2016-11-28”}, t, r)$$

"Is there a room booked on Nov 28th?"

Table of contents

Logical satisfiability

OWQA is equivalent to:

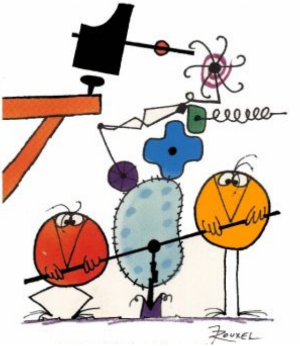
is $I \wedge \Theta \wedge \neg Q$ satisfiable?

Logical satisfiability

OWQA is equivalent to:

is $I \wedge \Theta \wedge \neg Q$ satisfiable?

Is it just logical satisfiability then?!



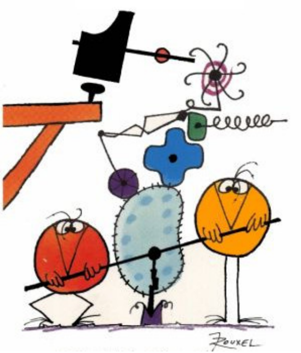
Logical satisfiability

OWQA is equivalent to:

is $I \wedge \Theta \wedge \neg Q$ satisfiable?

Is it just logical satisfiability then?!

- I is where we want to scale



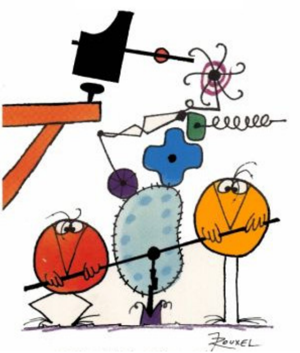
Logical satisfiability

OWQA is equivalent to:

is $I \wedge \Theta \wedge \neg Q$ satisfiable?

Is it just logical satisfiability then?!

- I is where we want to scale
- Θ and Q are usually different languages...

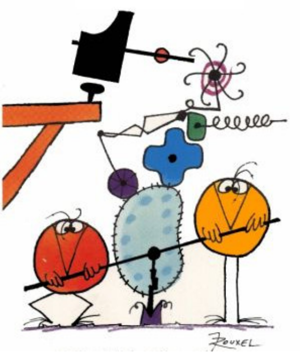


Logical satisfiability

OWQA is equivalent to:

is $I \land \Theta \land \neg Q$ satisfiable?

Is it just logical satisfiability then?!



- I is where we want to scale
- Θ and Q are usually different languages...
- ... if we express both in the same language,
  it will be hard to achieve good complexities!
  (or even decidability...)

Repairing the database

Class

date	teacher	resp	name	num
2016-11-28	Antoine	Fabian	Uncert. Data Mgmt	2
2016-12-05	Antoine	Fabian	Uncert. Data Mgmt	3
2016-12-12	Antoine	Fabian	Uncert. Data Mgmt	4
2017-01-09	Silviu	Fabian	Uncert. Data Mgmt	5

Repairing the database

Class

date	teacher	resp	name	num
2016-11-28	Antoine	Fabian	Uncert. Data Mgmt	2
2016-12-05	Antoine	Fabian	Uncert. Data Mgmt	3
2016-12-12	Antoine	Fabian	Uncert. Data Mgmt	4
2017-01-09	Silviu	Fabian	Uncert. Data Mgmt	5

Constraints:

- The responsible for a class must teach some class
- Every class must have a first session

→ What can we deduce?

Repairing the database

Class

date	teacher	resp	name	num
2016-11-28	Antoine	Fabian	Uncert. Data Mgmt	2
2016-12-05	Antoine	Fabian	Uncert. Data Mgmt	3
2016-12-12	Antoine	Fabian	Uncert. Data Mgmt	4
2017-01-09	Silviu	Fabian	Uncert. Data Mgmt	5
?	Fabian	?	?	?

Constraints:

- The responsible for a class must teach some class
- Every class must have a first session

$\rightarrow$ What can we deduce?

Repairing the database

Class

date	teacher	resp	name	num
2016-11-28	Antoine	Fabian	Uncert. Data Mgmt	2
2016-12-05	Antoine	Fabian	Uncert. Data Mgmt	3
2016-12-12	Antoine	Fabian	Uncert. Data Mgmt	4
2017-01-09	Silviu	Fabian	Uncert. Data Mgmt	5
?	Fabian	?	?	?
?	?	?	?	1

Constraints:

- The responsible for a class must teach some class
- Every class must have a first session

→ What can we deduce?

Repairing the database

Class

date	teacher	resp	name	num
2016-11-28	Antoine	Fabian	Uncert. Data Mgmt	2
2016-12-05	Antoine	Fabian	Uncert. Data Mgmt	3
2016-12-12	Antoine	Fabian	Uncert. Data Mgmt	4
2017-01-09	Silviu	Fabian	Uncert. Data Mgmt	5
?	Fabian	?	?	?
?	?	?	?	1

Constraints:

- The responsible for a class must teach some class
- Every class must have a first session

→ What can we deduce?

→ Q is true iff it is true on all completions

But why deal with broken databases?

- The data may have come from a different source
- The constraints may have been imposed after the fact
- User input may be incorrect
- You want a resilient system…

But why deal with broken databases?

- The data may have come from a different source
- The constraints may have been imposed after the fact
- User input may be incorrect
- You want a resilient system…

Date	créneau	Type	Sigle	Titre de l'activité pédagogique	Groupe	Equipe enseignante	Salle
27/10/2015	13:30 - 16:45	Leçon	INF922	ISC651 Technologies Applicatives	1	Jean DUPONT	C47
03/11/2015	09:00 - 12:00	Leçon	INF922	Integration dapplications (EAI, SOA) Ph Bron	1		C46
03/11/2015	13:30 - 16:45	Leçon	INF922	ISC651 Cloud Computing Ph Bron;	1		C46
	08:30			INF922: ingénierie des			

Reasoning (AI)

- Artificial reasoning: draw consequences from what you know
    - I contains the facts
    - Θ are the reasoning rules
    - Q is what we want to figure out

Reasoning (AI)

- Artificial reasoning: draw consequences from what you know
  - I contains the facts
  - Θ are the reasoning rules
  - Q is what we want to figure out

$\rightarrow$ Can we deduce Q from I using Θ?

$\rightarrow$ Is Q certain to hold?

Data integration

- I contains $I_1, \ldots, I_n$, the course databases of all D&K schools
- We want to create a virtual global database I of classes

Data integration

- I contains $I_1, \ldots, I_n$, the course databases of all D&K schools
- We want to create a virtual global database I of classes
- Fix a relation Class for the global database
- Θ: whenever some I_i contains a class, create it in R

Class1

date	**name**
2016-11-28	UDM
2016-12-05	UDM
2016-12-12	UDM
2017-01-14	UDM

Data integration

- I contains $I_1, \ldots, I_n$, the course databases of all D&K schools
- We want to create a virtual global database I of classes
- Fix a relation Class for the global database
- Θ: whenever some I_i contains a class, create it in R

Class1

date	name
2016-11-28	UDM
2016-12-05	UDM
2016-12-12	UDM
2017-01-14	UDM

Class

date	teacher	resp	name	num

Data integration

- I contains $I_1, \ldots, I_n$, the course databases of all D&K schools
- We want to create a virtual global database I of classes
- Fix a relation Class for the global database
- Θ: whenever some I_i contains a class, create it in R

Class1

date	name
2016-11-28	UDM
2016-12-05	UDM
2016-12-12	UDM
2017-01-14	UDM

Class

date	teacher	resp	name	num
2016-11-28	?	?	UDM	?
2016-12-05	?	?	UDM	?
2016-12-12	?	?	UDM	?
2017-01-14	?	?	UDM	?

Ontology-based data access

- In general: use a common schema for reasoning
- I contains heterogeneous data sources
- Θ describes mappings from sources to common schema and reasoning rules and constraints on the common schema
- Q is the query posed the common schema

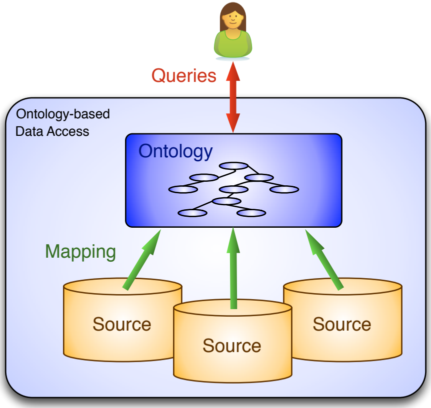


Table of contents

First-order logic

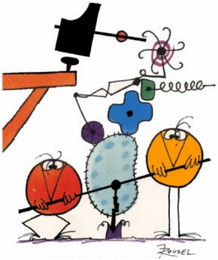
All constraints for Θ are in first-order logic (FO):

- contains atoms $R(x, y, z)$
- closed under Boolean AND, OR, NOT
- existential quantification $\exists x\ \phi(x)$
- universal quantification $\forall x\ \phi(x)$

First-order logic

All constraints for Θ are in first-order logic (FO):

- contains atoms $R(x, y, z)$
- closed under Boolean AND, OR, NOT
- existential quantification $\exists x\ \phi(x)$
- universal quantification $\forall x\ \phi(x)$



$\rightarrow$ Why not just use FO for constraints then?!

FO is undecidable!

Given an input FO formula Θ, is it satisfiable?
(i.e., OWQA with $I = \emptyset$ and Q : False).

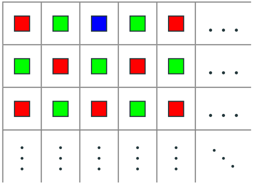
FO is undecidable!

Given an input FO formula Θ, is it satisfiable?
(i.e., OWQA with $I = \emptyset$ and Q : False).

$\rightarrow$ This problem is undecidable!

FO is undecidable!

Given an input FO formula Θ, is it satisfiable?
(i.e., OWQA with $I = \emptyset$ and Q : False).

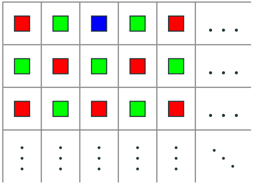


$\rightarrow$ This problem is undecidable!

- Proof: Encode a tiling system, or
  encode transition rules for a Turing machine

FO is undecidable!

Given an input FO formula Θ, is it satisfiable?
(i.e., OWQA with $I = \emptyset$ and Q : False).



- $\rightarrow$ This problem is undecidable!
    - Proof: Encode a tiling system, or
      encode transition rules for a Turing machine
- $\rightarrow$ We consider weaker languages

Tuple-generating dependencies

Tuple-generating dependencies (TGDs), classical database rules:

$$\forall \mathbf{x}\ Q'(\mathbf{x}) \Rightarrow \exists \mathbf{y}\ Q''(\mathbf{x}, \mathbf{y})$$

where Q' and Q'' are CQs.

Tuple-generating dependencies

Tuple-generating dependencies (TGDs), classical database rules:

$$\forall \mathbf{x}\; Q'(\mathbf{x}) \Rightarrow \exists \mathbf{y}\; Q''(\mathbf{x}, \mathbf{y})$$

where Q' and Q'' are CQs.

$$\begin{aligned} \forall\, \text{date},\ \text{prof},\ \text{r},\ \text{n},\ \text{i},\ \ & \text{Class}(\text{date}, \text{prof}, \text{r}, \text{n}, \text{i}) \Rightarrow \\ & \exists \text{room},\ \ \text{Book}(\text{date}, \text{room}, \text{prof}) \end{aligned}$$

Intuition: facts cause more facts to be created

Tuple-generating dependencies

Tuple-generating dependencies (TGDs), classical database rules:

$$\forall \mathbf{x}\ Q'(\mathbf{x}) \Rightarrow \exists \mathbf{y}\ Q''(\mathbf{x}, \mathbf{y})$$

where Q' and Q'' are CQs.

$$\begin{aligned} \forall\, \text{date},\ \text{prof},\ \text{r},\ \text{n},\ \text{i},\ \ & \text{Class}(\text{date}, \text{prof}, \text{r}, \text{n}, \text{i}) \Rightarrow \\ & \exists \text{room},\ \ \text{Book}(\text{date}, \text{room}, \text{prof}) \end{aligned}$$

Intuition: facts cause more facts to be created

Useful for:

- Integrity constraints: see above
- Schema mappings: copy facts from I_1 to I
- Reasoning: $\forall x$ Human(x) $\Rightarrow$ Mortal(x)

OWQA for TGDs is undecidable!

- Satisfiability of TGDs Θ is trivial...

OWQA for TGDs is undecidable!

- Satisfiability of TGDs Θ is trivial...
  - $\rightarrow$ take $W = \emptyset$

OWQA for TGDs is undecidable!

- Satisfiability of TGDs Θ is trivial...
  - $\rightarrow$ take $W = \emptyset$
- Satisfiability of TGDs Θ and instance I is easy...

OWQA for TGDs is undecidable!

- Satisfiability of TGDs Θ is trivial...
  - $\rightarrow$ take $W = \emptyset$
- Satisfiability of TGDs Θ and instance I is easy...
  - $\rightarrow$ always possible – infinite repair of violations (the chase – see later)

OWQA for TGDs is undecidable!

- Satisfiability of TGDs Θ is trivial...
  - $\rightarrow$ take $W = \emptyset$
- Satisfiability of TGDs Θ and instance I is easy...
  - $\rightarrow$ always possible – infinite repair of violations (the chase – see later)
- OWQA for I, Θ and Q is undecidable!
  from [Chandra et al., 1981, Beeri and Vardi, 1981]

OWQA for TGDs is undecidable!

- Satisfiability of TGDs Θ is trivial...
  - $\rightarrow$ take $W = \emptyset$
- Satisfiability of TGDs Θ and instance I is easy...
  - $\rightarrow$ always possible – infinite repair of violations (the chase – see later)
- OWQA for I, Θ and Q is undecidable!
  from [Chandra et al., 1981, Beeri and Vardi, 1981]

$\rightarrow$ We need less expressive languages

Inclusion dependencies

Inclusion dependencies (IDs), classical database rules:

$$\forall \mathbf{x}\ A'(\mathbf{x}) \Rightarrow \exists \mathbf{y}\ A''(\mathbf{x}, \mathbf{y})$$

where A and A' are atoms rather than CQs.

Inclusion dependencies

Inclusion dependencies (IDs), classical database rules:

$$\forall \mathbf{x}\; A'(\mathbf{x}) \Rightarrow \exists \mathbf{y}\; A''(\mathbf{x}, \mathbf{y})$$

where A and A' are atoms rather than CQs.

The TGD example was in fact also an ID:

$$\begin{aligned} \forall\, \text{date},\ \text{prof},\ \text{r},\ \text{n},\ \text{i},\ & \text{Class}(\text{date}, \text{prof}, \text{r}, \text{n}, \text{i}) \Rightarrow \\ & \exists \text{room},\ \text{Book}(\text{date}, \text{room}, \text{prof}) \end{aligned}$$

OWQA for IDs is decidable

- Satisfiability for IDs is still trivial
- OWQA: determine whether Q is implied by I and IDs Θ?

OWQA for IDs is decidable

- Satisfiability for IDs is still trivial
- OWQA: determine whether Q is implied by I and IDs Θ?
  - Decidable!
  - PSPACE-complete combined complexity [Johnson and Klug, 1984]
  - Data complexity is PTIME, even AC^0

OWQA for IDs is decidable

- Satisfiability for IDs is still trivial
- OWQA: determine whether Q is implied by I and IDs Θ?
  - Decidable!
  - PSPACE-complete combined complexity [Johnson and Klug, 1984]
  - Data complexity is PTIME, even AC^0

  $\rightarrow$ Intuition: we can rewrite the query Q

OWQA for IDs is decidable

- Satisfiability for IDs is still trivial
- OWQA: determine whether Q is implied by I and IDs Θ?
  - Decidable!
  - PSPACE-complete combined complexity [Johnson and Klug, 1984]
  - Data complexity is PTIME, even AC^0
  - $\rightarrow$ Intuition: we can rewrite the query Q

$\rightarrow$ We will study other decidable classes of TGDs

Table of contents

Chase example

Class

date	teacher	resp	name	num
2016-12-05	Antoine	Fabian	Uncert. Data Mgmt	3
2016-12-12	Antoine	Fabian	Uncert. Data Mgmt	4

Chase example

Class

date	teacher	resp	name	num
2016-12-05	Antoine	Fabian	Uncert. Data Mgmt	3
2016-12-12	Antoine	Fabian	Uncert. Data Mgmt	4

$$\forall\, \text{date},\ \text{prof},\ \text{r},\ \text{n},\ \text{i},\ \text{Class}(\text{date}, \text{prof}, \text{r}, \text{n}, \text{i}) \Rightarrow$$
$$\exists \text{room},\ \text{Book}(\text{date}, \text{room}, \text{prof})$$

Chase example

Class

date	teacher	resp	name	num
2016-12-05	Antoine	Fabian	Uncert. Data Mgmt	3
2016-12-12	Antoine	Fabian	Uncert. Data Mgmt	4

$$\forall\, \text{date},\ \text{prof},\ \text{r},\ \text{n},\ \text{i},\ \text{Class}(\text{date}, \text{prof}, \text{r}, \text{n}, \text{i}) \Rightarrow \exists \text{room},\ \text{Book}(\text{date}, \text{room}, \text{prof})$$

Book

date	room	prof

Chase example

Class

date	teacher	resp	name	num
2016-12-05	Antoine	Fabian	Uncert. Data Mgmt	3
2016-12-12	Antoine	Fabian	Uncert. Data Mgmt	4

$$\forall \text{date}, \text{prof}, r, n, i, \ \text{Class}(\text{date}, \text{prof}, r, n, i) \Rightarrow \exists \text{room}, \ \text{Book}(\text{date}, \text{room}, \text{prof})$$

Book

date	room	prof
2016-12-05	*?1*	Antoine
2016-12-12	*?2*	Antoine

Chase

- OWQA: test if instance I and TGDs Θ entail query Q
- The chase: a most generic repair of I by Θ

Chase

- OWQA: test if instance I and TGDs Θ entail query Q
- The chase: a most generic repair of I by Θ
- Iterative process: start with I
- At each stage, find violations of each TGD in Θ

Chase

- OWQA: test if instance I and TGDs Θ entail query Q
- The chase: a most generic repair of I by Θ
- Iterative process: start with I
- At each stage, find violations of each TGD in Θ
  - TGD $\forall \mathbf{x}\ Q'(\mathbf{x}) \Rightarrow \exists \mathbf{y}\ Q''(\mathbf{x}, \mathbf{y})$
  - find $\mathbf{a}$ such that $Q'(\mathbf{a})$ but not $Q''(\mathbf{a}, \mathbf{b})$ for any $\mathbf{b}$

Chase

- OWQA: test if instance I and TGDs Θ entail query Q
- The chase: a most generic repair of I by Θ
- Iterative process: start with I
- At each stage, find violations of each TGD in Θ
    - TGD $\forall \mathbf{x}\ Q'(\mathbf{x}) \Rightarrow \exists \mathbf{y}\ Q''(\mathbf{x}, \mathbf{y})$
    - find $\mathbf{a}$ such that $Q'(\mathbf{a})$ but not $Q''(\mathbf{a}, \mathbf{b})$ for any $\mathbf{b}$
- Create new elements $\mathbf{b}$
- Create new facts to make $Q'(\mathbf{a}, \mathbf{b})$ true

Chase

- OWQA: test if instance I and TGDs Θ entail query Q
- The chase: a most generic repair of I by Θ
- Iterative process: start with I
- At each stage, find violations of each TGD in Θ
  - TGD $\forall \mathbf{x}\ Q'(\mathbf{x}) \Rightarrow \exists \mathbf{y}\ Q''(\mathbf{x}, \mathbf{y})$
  - find $\mathbf{a}$ such that $Q'(\mathbf{a})$ but not $Q''(\mathbf{a}, \mathbf{b})$ for any $\mathbf{b}$
- Create new elements $\mathbf{b}$
- Create new facts to make $Q'(\mathbf{a}, \mathbf{b})$ true

$\rightarrow$ Take the infinite result of this process

Infinite chase example

$$\forall t\, u \; \text{Mentor}(t, u) \Rightarrow \exists s \; \text{Mentor}(s, t)$$

Mentor

master	padawan

Infinite chase example

$$\forall t\, u\ \text{Mentor}(t, u) \Rightarrow \exists s\ \text{Mentor}(s, t)$$

Mentor

master	**padawan**
Antoine	Arthur Dent
Silviu	Arthur Dent

Infinite chase example

$$\forall t\, u\ \text{Mentor}(t, u) \Rightarrow \exists s\ \text{Mentor}(s, t)$$

Mentor

master	**padawan**
Antoine	Arthur Dent
Silviu	Arthur Dent
?1	Antoine
?2	Silviu

Infinite chase example

$$\forall t\, u\ \mathrm{Mentor}(t, u) \Rightarrow \exists s\ \mathrm{Mentor}(s, t)$$

Mentor

master	**padawan**
Antoine	Arthur Dent
Silviu	Arthur Dent
$?_1$	Antoine
$?_2$	Silviu
$?_3$	$?_1$
$?_4$	$?_2$

Infinite chase example

$$\forall t\, u\ \text{Mentor}(t, u) \Rightarrow \exists s\ \text{Mentor}(s, t)$$

Mentor

master	**padawan**
Antoine	Arthur Dent
Silviu	Arthur Dent
$?1$	Antoine
$?2$	Silviu
$?3$	$?1$
$?4$	$?2$
$?5$	$?3$
$?6$	$?4$

Infinite chase example

$$\forall t\, u\ \mathrm{Mentor}(t, u) \Rightarrow \exists s\ \mathrm{Mentor}(s, t)$$

Mentor

master	**padawan**
Antoine	Arthur Dent
Silviu	Arthur Dent
$?1$	Antoine
$?2$	Silviu
$?3$	$?1$
$?4$	$?2$
$?5$	$?3$
$?6$	$?4$
$\vdots$	$\vdots$

Chase universality

- The chase is the most generic completion
  - Can be shown using homomorphisms

Chase universality

- The chase is the most generic completion
  - Can be shown using homomorphisms

$\rightarrow$ A query is true in the chase iff it is entailed

Chase universality

- The chase is the most generic completion
  - Can be shown using homomorphisms

→ A query is true in the chase iff it is entailed

Theorem

For any instance I, TGDs Θ, Boolean CQ Q, the following are equivalent:

- *I and Θ entail Q*
- *the chase of I by Θ satisfies Q*

Chase universality

- The chase is the most generic completion
    - Can be shown using homomorphisms

$\rightarrow$ A query is true in the chase iff it is entailed

Theorem

For any instance I, TGDs Θ, Boolean CQ Q, the following are equivalent:

- *I and Θ entail Q*
- *the chase of I by Θ satisfies Q*

$\rightarrow$ How to reason about this infinite chase?

Chase termination

- Sometimes, the chase of I by Θ is finite
- We can then decide whether a query Q is entailed:

Chase termination

- Sometimes, the chase of I by Θ is finite
- We can then decide whether a query Q is entailed:
  - Construct the chase
  - Evaluate Q on the chase

Chase termination

- Sometimes, the chase of I by Θ is finite
- We can then decide whether a query Q is entailed:
  - Construct the chase
  - Evaluate Q on the chase

$\rightarrow$ When is the chase finite?

Full dependencies

- If no TGD has an $\exists$, then the chase is finite

$\rightarrow$ No new elements are created

Full dependencies

- If no TGD has an $\exists$, then the chase is **finite**

$\rightarrow$ No **new elements** are created

- Good: $\forall d\, r\, p\ \mathrm{Book}(d, r, p) \Rightarrow \mathrm{Room}(r)$
- Bad: $\forall \mathbf{x}\ \mathrm{Mentor}(\mathbf{x}) \Rightarrow \exists \mathbf{y}\ \mathrm{Mentor}(\mathbf{x}, \mathbf{y})$

Acyclicity

- Simple sufficient condition for finite chase:
  If a relation name occurs at the left of a TGD
  then it does not occur at the right

Acyclicity

- Simple sufficient condition for finite chase:
  If a relation name occurs at the left of a TGD
  then it does not occur at the right
- Good: $\forall \mathbf{x}$ Class($\mathbf{x}$) $\Rightarrow \exists \mathbf{y}$ Book($\mathbf{x}$)
- Bad: $\forall \mathbf{x}$ Mentor($\mathbf{x}$) $\Rightarrow \exists \mathbf{y}$ Mentor($\mathbf{x}, \mathbf{y}$)

Acyclicity

- Simple sufficient condition for finite chase:
  If a relation name occurs at the left of a TGD
  then it does not occur at the right
- Good: $\forall \mathbf{x}$ Class($\mathbf{x}$) $\Rightarrow \exists \mathbf{y}$ Book($\mathbf{x}$)
- Bad: $\forall \mathbf{x}$ Mentor($\mathbf{x}$) $\Rightarrow \exists \mathbf{y}$ Mentor($\mathbf{x}, \mathbf{y}$)
- More general acyclicity conditions

Infinite chase

- What can we do if the chase is infinite?
- Bounded derivation depth: we can truncate the chase:
  - we fix Θ and look at the size of Q
  - bound the maximal depth in the chase
    where Q can be made true

Infinite chase

- What can we do if the chase is infinite?
- Bounded derivation depth: we can truncate the chase:
    - we fix Θ and look at the size of Q
    - bound the maximal depth in the chase where Q can be made true
- Bounded treewidth: the chase is like a tree:
    - we can reason about infinite and regular trees
    - use tree automata, following Courcelle's theorem
    - some rules preserve this, e.g., the guarded fragments

Table of contents

Query rewriting

- The chase: reason about consequences of I under Θ
- Other option: reason about how to prove Q

Query rewriting

- The chase: reason about consequences of I under Θ
- Other option: reason about how to prove Q

$$\Theta : \forall \text{date}, \text{prof}, r, n, i, \; \text{Class}(\text{date}, \text{prof}, r, n, i) \Rightarrow \exists \text{room}, \; \text{Book}(\text{date}, \text{room}, \text{prof})$$

Query rewriting

- The chase: reason about consequences of I under Θ
- Other option: reason about how to prove Q

$$\Theta : \forall \text{date}, \text{prof}, r, n, i, \ \text{Class}(\text{date}, \text{prof}, r, n, i) \Rightarrow \exists \text{room}, \ \text{Book}(\text{date}, \text{room}, \text{prof})$$

$$Q : \exists t\, r\ \text{Book}(\text{“2016-11-28”}, t, r)$$

Query rewriting

- The chase: reason about consequences of I under Θ
- Other option: reason about how to prove Q

$$\Theta : \forall\, \text{date},\ \text{prof},\ \text{r},\ \text{n},\ \text{i},\ \text{Class}(\text{date}, \text{prof}, \text{r}, \text{n}, \text{i}) \Rightarrow \exists \text{room},\ \text{Book}(\text{date}, \text{room}, \text{prof})$$

$$Q : \exists \text{t}\ \text{r}\ \text{Book}(\text{“2016-11-28”}, \text{t}, \text{r})$$

$$Q_2 : \exists \text{prof},\ \text{r},\ \text{n},\ \text{i},\ \text{Class}(\text{“2016-11-28”}, \text{prof}, \text{r}, \text{n}, \text{i})$$

Query rewriting and inclusion dependencies

- To show that OWQA for inclusion dependencies is decidable...

$$\forall \mathbf{x}\; A'(\mathbf{x}) \Rightarrow \exists \mathbf{y}\; A''(\mathbf{x}, \mathbf{y})$$

Query rewriting and inclusion dependencies

- To show that OWQA for inclusion dependencies is decidable...

$$\forall \mathbf{x}\, A'(\mathbf{x}) \Rightarrow \exists \mathbf{y}\, A''(\mathbf{x}, \mathbf{y})$$

- Rewrite all atoms in the query in all possible ways
  - $\rightarrow$ Each atom rewritten by only one atom
  - $\rightarrow$ The query size does not increase

Query rewriting and inclusion dependencies

- To show that OWQA for inclusion dependencies is decidable…

$$\forall \mathbf{x}\, A'(\mathbf{x}) \Rightarrow \exists \mathbf{y}\, A''(\mathbf{x}, \mathbf{y})$$

- Rewrite all atoms in the query in all possible ways
  - $\rightarrow$ Each atom rewritten by only one atom
  - $\rightarrow$ The query size does not increase

- Replace Q by a union of conjunctive queries
  - $\rightarrow$ OWQA for IDs is decidable
  - $\rightarrow$ OWQA for IDs has tractable data complexity

Description logics

- TGDs cannot express everything

$$\forall \mathbf{x}\ Q'(\mathbf{x}) \Rightarrow \exists \mathbf{y}\ Q''(\mathbf{x}, \mathbf{y})$$

Description logics

- TGDs cannot express everything

$$\forall \mathbf{x}\ Q'(\mathbf{x}) \Rightarrow \exists \mathbf{y}\ Q''(\mathbf{x}, \mathbf{y})$$

- Disjunction: if A then B or C
- Negation: you cannot have both A and B

Description logics

- TGDs cannot express everything

$$\forall \mathbf{x}\ Q'(\mathbf{x}) \Rightarrow \exists \mathbf{y}\ Q''(\mathbf{x}, \mathbf{y})$$

- Disjunction: if A then B or C
- Negation: you cannot have both A and B

→ Description logics: expressive rules

  - signature must have arity at most 2

Description logics (2)

- Description logics have a specific syntax

$$\text{Teacher} \sqsubseteq \text{Prof} \sqcup (\exists \text{Advisor}^{-}.\text{Prof})$$

Description logics (2)

- Description logics have a specific **syntax**

$$\mathrm{Teacher} \sqsubseteq \mathrm{Prof} \sqcup (\exists \mathrm{Advisor}^{-}.\mathrm{Prof})$$

- Description logics exist in many **variants**
  - **Idea:** precise **complexity** of OWQA depending on variant

Description logics (2)

- Description logics have a specific **syntax**

$$\text{Teacher} \sqsubseteq \text{Prof} \sqcup (\exists \text{Advisor}^{-}.\text{Prof})$$

- Description logics exist in many **variants**
  - **Idea:** precise **complexity** of OWQA depending on variant

Languages	UNA	Complexity		
		Combined complexity	Data complexity	
		Satisfiability	Instance checking	Query answering
$\mathit{DL\text{-}Lite}_{core}^{[\ \mid\mathcal{H}]}$	yes/no	NLogSpace $\geq$ [A]	in AC^0	in AC^0
$\mathit{DL\text{-}Lite}_{horn}^{[\ \mid\mathcal{H}]}$	yes/no	P $\leq$ [Th.8.2] $\geq$ [A]	in AC^0	in AC^0 $\leq$ [C]
$\mathit{DL\text{-}Lite}_{krom}^{[\ \mid\mathcal{H}]}$	yes/no	NLogSpace $\leq$ [Th.8.2]	in AC^0	coNP $\geq$ [B]
$\mathit{DL\text{-}Lite}_{bool}^{[\ \mid\mathcal{H}]}$	yes/no	NP $\leq$ [Th.8.2] $\geq$ [A]	in AC^0 $\leq$ [Th.8.3]	coNP
$\mathit{DL\text{-}Lite}_{core}^{[\mathcal{F}\mid\mathcal{N}\mid(\mathcal{HF})\mid(\mathcal{HN})]}$	yes	NLogSpace	in AC^0	in AC^0
$\mathit{DL\text{-}Lite}_{horn}^{[\mathcal{F}\mid\mathcal{N}\mid(\mathcal{HF})\mid(\mathcal{HN})]}$	yes	P $\leq$ [Th.5.8, 5.13]	in AC^0	in AC^0 $\leq$ [Th.7.1]
$\mathit{DL\text{-}Lite}_{krom}^{[\mathcal{F}\mid\mathcal{N}\mid(\mathcal{HF})\mid(\mathcal{HN})]}$	yes	NLogSpace $\leq$ [Th.5.7,5.13]	in AC^0	coNP
$\mathit{DL\text{-}Lite}_{bool}^{[\mathcal{F}\mid\mathcal{N}\mid(\mathcal{HF})\mid(\mathcal{HN})]}$	yes	NP $\leq$ [Th.5.6, 5.13]	in AC^0 $\leq$ [Cor.6.2]	coNP
$\mathit{DL\text{-}Lite}_{core/horn}^{[\mathcal{F}\mid(\mathcal{HF})]}$	no	P $\leq$ [Cor.8.8] $\geq$ [Th.8.7]	P $\geq$ [Th.8.7]	P
$\mathit{DL\text{-}Lite}_{krom}^{[\mathcal{F}\mid(\mathcal{HF})]}$	no	P $\leq$ [Cor.8.8]	P	coNP
$\mathit{DL\text{-}Lite}_{bool}^{[\mathcal{F}\mid(\mathcal{HF})]}$	no	NP	P $\leq$ [Cor.8.8]	coNP
$\mathit{DL\text{-}Lite}_{core/horn}^{[\mathcal{N}\mid(\mathcal{HN})]}$	no	NP $\geq$ [Th.8.4]	coNP $\geq$ [Th.8.4]	coNP
$\mathit{DL\text{-}Lite}_{krom/bool}^{[\mathcal{N}\mid(\mathcal{HN})]}$	no	NP $\leq$ [Th.8.5]	coNP	coNP
$\mathit{DL\text{-}Lite}_{core/horn}^{\mathcal{HF}}$	yes/no	ExpTime $\geq$ [Th.5.10]	P $\geq$ [Th.6.7]	P $\leq$ [D]
$\mathit{DL\text{-}Lite}_{krom/bool}^{\mathcal{HF}}$	yes/no	ExpTime	coNP $\geq$ [Th.6.5]	coNP
$\mathit{DL\text{-}Lite}_{core/horn}^{\mathcal{HN}}$	yes/no	ExpTime	coNP $\geq$ [Th.6.6]	coNP
$\mathit{DL\text{-}Lite}_{krom/bool}^{\mathcal{HN}}$	yes/no	ExpTime $\leq$ [F]	coNP	coNP $\leq$ [E]

Equality-generating dependencies

- Another important constraint for Θ: functional dependencies
  - There can't be two bookings for one room at the same time
  - There can't be two rooms for one session

Equality-generating dependencies

- Another important constraint for Θ: functional dependencies
    - There can't be two bookings for one room at the same time
    - There can't be two rooms for one session
- Functional dependencies can be added to Θ for OWQA
    - Decidable for description logics
    - Undecidable with inclusion dependencies

Finite models

Definition (Open-World Query Answering – OWQA)
Given an instance I, Boolean CQ Q, and constraints Θ, decide whether all $W \supseteq I$ that satisfy Θ satisfy Q.

Finite models

Definition (Open-World Query Answering – OWQA)

Given an instance I, Boolean CQ Q, and constraints Θ, decide whether all $W \supseteq I$ that satisfy Θ satisfy Q.

- The world W is actually finite
- Shouldn't we reflect this?

Finite models

Definition (Finite Open-World Query Answering – FOWQA)
Given an instance I, Boolean CQ Q, and constraints Θ, decide whether all finite $W \supseteq I$ that satisfy Θ satisfy Q.

- The world W is actually finite
- Shouldn't we reflect this?

- If the chase is infinite, it no longer works
- Imposing finiteness may make a difference

Finite models

Definition (Finite Open-World Query Answering – FOWQA)
Given an instance I, Boolean CQ Q, and constraints Θ, decide whether all finite $W \supseteq I$ that satisfy Θ satisfy Q.

- The world W is actually finite
- Shouldn't we reflect this?

- If the chase is infinite, it no longer works
- Imposing finiteness may make a difference

$\rightarrow$ Very hard to reason about FOWQA!

Partial completeness

- We have assumed that I was incomplete
- Sometimes, we know which relations are complete
  - e.g., the list of rooms may be complete
  - the list of classes may be incomplete

Partial completeness

- We have assumed that I was incomplete
- Sometimes, we know which relations are complete
  - e.g., the list of rooms may be complete
  - the list of classes may be incomplete

$\rightarrow$ Partially complete databases [Razniewski et al., 2015]

Partial completeness

- We have assumed that *I* was incomplete
- Sometimes, we know which relations are complete
  - e.g., the list of rooms may be complete
  - the list of classes may be incomplete

→ Partially complete databases [Razniewski et al., 2015]

- We may know other things:
  - If I know the lecturer of a class, then I know all lecturers
  - If I know one session of a class, I know all sessions

Partial completeness

- We have assumed that I was incomplete
- Sometimes, we know which relations are complete
  - e.g., the list of rooms may be complete
  - the list of classes may be incomplete

$\rightarrow$ Partially complete databases [Razniewski et al., 2015]

- We may know other things:
  - If I know the lecturer of a class, then I know all lecturers
  - If I know one session of a class, I know all sessions

$\rightarrow$ Partial completeness assumption [Galárraga et al., 2013]

Slide credits

- Slide 34: `http://www.slideshare.net/MartnRezk/slides-swat4-ls`, slide 17, licence CC-BY-SA 3.0[1]
- Slides 16 and 36: Jaques Rouxel, *Les Shadoks* (*reproduit en vertu du droit de citation*)
- Slide 34: [Artale et al., 2009], p 18

[1] `http://creativecommons.org/licenses/by-sa/3.0/`

References I

Abiteboul, S., Hull, R., and Vianu, V. (1995).
Foundations of Databases.
Addison-Wesley.
http://webdam.inria.fr/Alice/pdfs/all.pdf.

Artale, A., Calvanese, D., Kontchakov, R., and Zakharyaschev, M. (2009).
The DL-Lite family and relations.
Journal of artificial intelligence research.
https://www.jair.org/media/2820/live-2820-4662-jair.pdf.

References II

Beeri, C. and Vardi, M. Y. (1981).
The implication problem for data dependencies.
In *Automata, Languages and Programming*.
http://www.researchgate.net/publication/226509257_The_implication_problem_for_data_dependencies.

Calì, A., Gottlob, G., and Pieris, A. (2012).
Towards more expressive ontology languages: The query answering problem.
Artif. Intel., 193.
http://www.sciencedirect.com/science/article/pii/S0004370212001026.

References III

Calì, A., Lembo, D., and Rosati, R. (2003).
On the decidability and complexity of query answering over inconsistent and incomplete databases.
In *Proc. PODS*.
`http://www.dis.uniroma1.it/~rosati/publications/Cali-Lembo-Rosati-PODS-03.pdf`.

Chandra, A. K., Lewis, H. R., and Makowsky, J. A. (1981).
Embedded implicational dependencies and their inference problem.
In *Proc. STOC*. ACM.
Unavailable online.

References IV

Galárraga, L. A., Teflioudi, C., Hose, K., and Suchanek, F. (2013).
AMIE: association rule mining under incomplete evidence in ontological knowledge bases.
In *Proc. WWW*.
`http://luisgalarraga.de/docs/amie.pdf`.

Johnson, D. S. and Klug, A. C. (1984).
Testing containment of conjunctive queries under functional and inclusion dependencies.
JCSS.
`https://cs.uwaterloo.ca/~david/kbdb/johnstonandklug.pdf`.

References V

Onet, A. (2013).
The chase procedure and its applications in data exchange.
In *Data Exchange, Integration, and Streams*, pages 1–37. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik.
`http://drops.dagstuhl.de/opus/volltexte/2013/4288/pdf/ch01-onet.pdf`.

Razniewski, S., Korn, F., Nutt, W., and Srivastava, D. (2015).
Identifying the extent of completeness of query answers over partially complete databases.
In *Proc. SIGMOD*.
`https://srazniewski.files.wordpress.com/2015/05/2015_sigmod.pdf`.

References VI

Rosati, R. (2006).
On the decidability and finite controllability of query processing in databases with incomplete information.
In *PODS*.
http://www.sciencedirect.com/science/article/pii/S002200001000053X.