Query reformulation under expressive constraints: the role of determinacy and of domain independence

Enrico Franconi,
Paolo Guagliardo, Volha Kerhet, and Nhung Ngo

KRDB Research Centre, Free University of Bozen-Bolzano, Italy

iPRA
18 July 2014

Summary

- Databases as DBoxes
- Query Reformulation
- Query Determinacy
- Exact Query Reformulation
- Safe Range Query Reformulation
- Conditions for Exact Query Reformulation
- An application to description logics
  - Various Codd's theorems for expressive description logics

Adding Databases to FOL

- A *set of contraints* $\mathcal{T}$ is:
  - a set of formulas in (a fragment of) first-order logic $\mathcal{L}$.
- A *DBox* $\mathcal{D}$ is a finite set of:
  - a set of atomic ground formulas in $\mathcal{L}$.
- A *query* $\mathcal{Q}$ is:
  - an open formula with one free variable in $\mathcal{L}$.

- $\mathbb{P}$ is the set of:
  - all predicate symbols in $\mathcal{L}$.
- $\mathbb{P}_{\mathcal{D}}$ is the set of all symbols of $\mathbb{P}$ appearing in $\mathcal{D}$: the *DBox predicates*.

- *Standard Name Assumption* for the individuals

DBox Semantics

- Ground atoms in FOL (also called ABoxes) and DBoxes have different semantics.
- An interpretation $\mathcal{I}$ *embeds a DBox* $\mathcal{D}$ – written $\mathcal{I}_{(\mathcal{D})}$ – if the interpretation of each DBox predicate is exactly defined by $\mathcal{D}$ with standard names.

Example

Given:

- ABox $\mathcal{A} : \{C(a)\}$
- DBox $\mathcal{D} : \{C(a)\}$

Then:

- for any interpretation $\mathcal{I}$ satisfying $\mathcal{A}$, $C^{\mathcal{I}} = \{a^{\mathcal{I}}, \ldots\}$;
- for any interpretation $\mathcal{I}$ embedding $\mathcal{D}$, $C^{\mathcal{I}} = \{a^{\mathcal{I}}\}$.

Certain Answer of a Query

Ans$(\mathcal{Q}, \mathcal{D}, \mathcal{T})$

The *(certain) answer of a query* $\mathcal{Q}$ *to a DBox* $\mathcal{D}$ *under constraints* $\mathcal{T}$ is the set of individuals which make the query true in *all* the models of the constraints embedding the DBox.

Ans$(\mathcal{Q}, \mathcal{A}, \mathcal{T})$

The *(certain) answer of a query* $\mathcal{Q}$ *to an ABox* $\mathcal{A}$ *under constraints* $\mathcal{T}$ is the set of individuals which make the query true in *all* the models of the constraints satisfying the ABox.

Example

$\mathbb{P} : \{C, D, R\}$

$\mathbb{P}_{\mathcal{D}} : \{D, R\}$

$\mathcal{T} : \forall x.C(x) \leftrightarrow (D(x) \wedge \neg \exists y.R(x, y))$

$\mathcal{Q} : C$

$\mathcal{D} = D(a)$ $\qquad$ $\mathcal{A} = D(a)$

$Ans(\mathcal{Q}, \mathcal{D}, \mathcal{T}) \rightsquigarrow \{a\}$ $\qquad$ $Ans(\mathcal{Q}, \mathcal{A}, \mathcal{T}) \rightsquigarrow \{\}$

Query Reformulation

- Given a set of constraints $\mathcal{T}$, a set of DBox predicates $\mathbb{P}_{\mathcal{D}}$, and a query $\mathcal{Q}$, a perfect reformulation $\hat{\mathcal{Q}}$ is an open first-order formula in $\mathbb{P}_{\mathcal{D}}$ such that for any $\mathcal{D}$ it holds

$$\mathit{Ans}(\mathcal{Q}, \mathcal{D}, \mathcal{T}) = \mathit{Ans}(\hat{\mathcal{Q}}, \mathcal{D}, \{\})$$

- Already for very simple languages (e.g., DL-Lite), a perfect reformulation does NOT always exist in presence of a DBox.

- We focus on the queries for which a reformulation exists, namely the queries determined by the DBox predicates.

Determinacy

Given a set of constraints $\mathcal{T}$ and a DBox $\mathcal{D}$, a query $\mathcal{Q}$ is determined by the DBox predicates $\mathbb{P}_{\mathcal{D}}$ if its answer functionally depends only from the extension of the DBox predicates.

Definition (Determinacy)

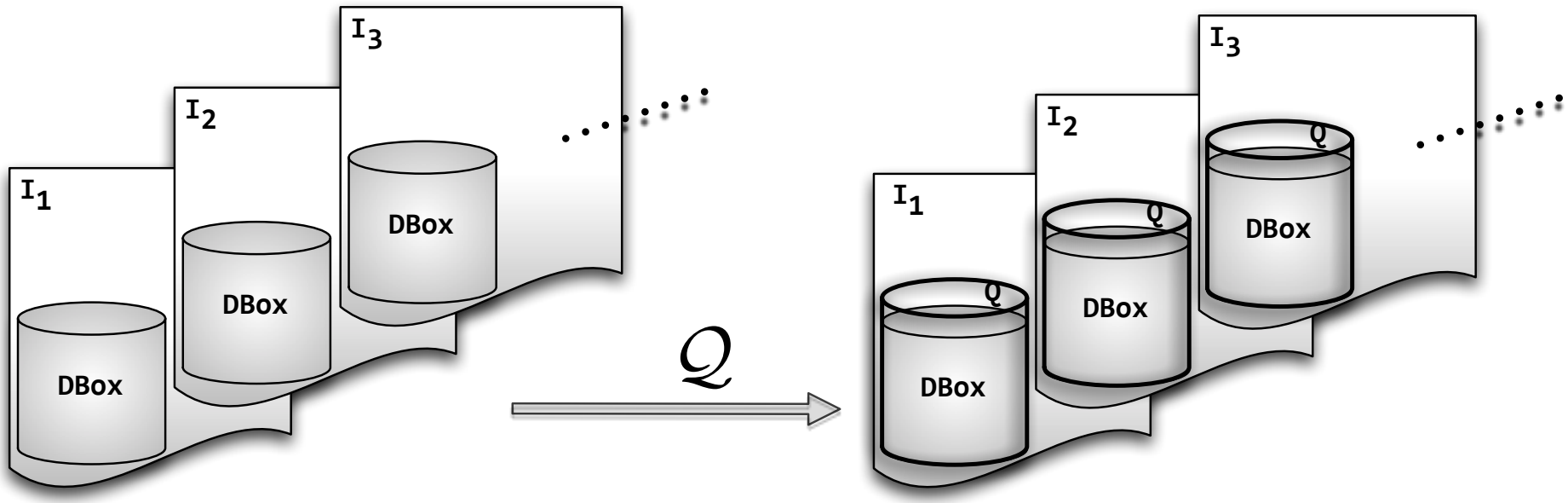
Let $\mathcal{I}^i_{(\mathcal{D})}$ and $\mathcal{I}^j_{(\mathcal{D})}$ be any two models of a set of constraints $\mathcal{T}$ embedding a DBox $\mathcal{D}$.
A query $\mathcal{Q}$ is determined by the DBox predicates $\mathbb{P}_{\mathcal{D}}$ given the set of constraints $\mathcal{T}$ *if* the answer of $\mathcal{Q}$ over $\mathcal{I}^i_{(\mathcal{D})}$ is the same as the answer of $\mathcal{Q}$ over $\mathcal{I}^j_{(\mathcal{D})}$.

Note: the answer of a query over an interpretation is the set of all the individuals which, when substituted to the free variable of the query, make the closed formula true in the interpretation.

Checking determinacy of a query with a first-order set of constraints is reducible to entailment.

Why determinacy?

- It captures *exactly* the notion of "non ambiguous" answers.
- Consider the models of a set of constraints $\mathcal{T}$ with a set of DBox predicates $\mathbb{P}_{\mathcal{D}}$. A query $\mathcal{Q}$ determined by $\mathbb{P}_{\mathcal{D}}$ generates an "extended" DBox augmented with the relation associated to the determined query.
- Arbitrary determined queries can be composed and decomposed without affecting the outcome.



Exact reformulation

Theorem

Given a set of constraints $\mathcal{T}$ and a DBox $\mathcal{D}$, a query $\mathcal{Q}$ is determined by the DBox predicates $\mathbb{P}_{\mathcal{D}}$ if and only if there exists an exact reformulation of $\mathcal{Q}$.

Definition

A reformulation $\hat{\mathcal{Q}}$ of a query $\mathcal{Q}$ is exact if $\mathcal{T} \models \forall x.\mathcal{Q}(x) \leftrightarrow \hat{\mathcal{Q}}(x)$.

Finitely Controllable Determinacy

- We focus on logics with finitely controllable determinacy:
  - whenever a query is finitely determined (i.e., determined over finite models) then it is also determined over unrestricted models (the reverse is trivially true).

Example

- $\forall x.Person(x) \rightarrow Man(x) \vee Women(x)$
  $\forall x.Man(x) \rightarrow Person(x)$
  $\forall x.Women(x) \rightarrow Person(x)$
  $\forall x.Women(x) \rightarrow \neg Man(x)$.
- $\mathcal{D} = \{Person(\mathbf{john}), Person(\mathbf{maria}), Man(\mathbf{john})\}$

Person
john
maria

Man
john

- $\mathcal{Q}(x) = Women(x)$
- $\widehat{\mathcal{Q}}(x) = Person(x) \wedge \neg Man(x)$.

$\widehat{\mathcal{Q}}$ is an exact reformulation of $\mathcal{Q}$ under the constraints $\mathcal{T}$ over the set of DBox predicates $\{Person, Man\}$. In fact:

$$\mathcal{T} \models \forall(x).Women(x) \leftrightarrow Person(x) \wedge \neg Man(x).$$

Safe-range queries

- The *safe-range* fragment is a syntactic fragment of FOL
- Intuition: A formula is safe-range if its variables (free and quantified) are bounded by positive predicates or equalities.
  - $\exists x.\, A(x) \wedge \neg B(x)$ - safe-range
  - $A(x) \vee B(x)$ - safe range
  - $\forall x.C(x)$ - not safe-range
- The safe-range fragment of FOL is equally expressive to the domain independent fragment of FOL and to relational algebra – the core of SQL.

$\Longrightarrow$ SQL can be used for the evaluation of safe-range formulas.

Query Answering: from a set of models to one model

Theorem

Let $\mathcal{T}$ be set of constraints, $\mathcal{Q}$ be a query, $\mathcal{D}$ be a consistent DBox for $\mathcal{T}$, and AD the set of all individuals in $\mathcal{D}$ and $\mathcal{Q}$.
If $\widehat{\mathcal{Q}}$ is

- an **exact** reformulation of $\mathcal{Q}$,
- **safe-range**,

then

$$Ans(\mathcal{Q}, \mathcal{D}, \mathcal{T}) = Ans(\widehat{\mathcal{Q}}, \mathcal{D}, \{\}) = \{a \mid \langle AD, \mathcal{D}\rangle \models \widehat{\mathcal{Q}}(a)\}$$

The original query answering problem (based on entailment) is reduced to the problem of checking the validity of the reformulation $\widehat{\mathcal{Q}}$ over the *single* interpretation given by the DBox with the active domain (model checking problem), which can be executed by an SQL engine.

Example

- $\forall x.Person(x) \rightarrow Man(x) \vee Women(x)$
  $\forall x.Man(x) \rightarrow Person(x)$
  $\forall x.Women(x) \rightarrow Person(x)$
  $\forall x.Women(x) \rightarrow \neg Man(x)$.
- $\mathcal{D} = \{Person(\mathbf{john}), Person(\mathbf{maria}), Man(\mathbf{john})\}$

Person
john
maria

Man
john

- $\mathcal{Q}(x) = Women(x)$
- $\widehat{\mathcal{Q}}(x) = Person(x) \wedge \neg Man(x)$.
- $\widehat{\mathcal{Q}}$ is an exact reformulation of $\mathcal{Q}$ under $\mathcal{T}$ over the set of DBox predicates $\{Person, Man\}$ and it is safe-range.

$Ans(\mathcal{Q}, \mathcal{D}, \mathcal{T}) =$
$\{o \mid \langle\{\mathbf{john}, \mathbf{maria}\}, \mathcal{D}\rangle \models Person(o) \wedge \neg Man(o)\} = \{\mathbf{maria}\}$

Problem Statement

Given

- set of constraints $\mathcal{T}$ in $\mathcal{L}$;
- a DBox $\mathcal{D}$ in $\mathcal{L}$;
- a concept query $\mathcal{Q}$ in $\mathcal{L}$.

We need to solve the following PROBLEM:

- find a first-order logic **safe-range exact** reformulation of $\mathcal{Q}$ expressed in terms of DBox predicates.

Semantic Characterisation theorem

Given a set of database predicates $\mathbb{P}_{\mathcal{D}}$,
a domain independent set of constraints $\mathcal{T}$,
and a query $\mathcal{Q}$,

a domain independent exact reformulation $\widehat{\mathcal{Q}}$ of $\mathcal{Q}$ over $\mathbb{P}_{\mathcal{D}}$ under $\mathcal{T}$ exists

if and only if

$\mathcal{Q}$ is implicitly definable from $\mathbb{P}_{\mathcal{D}}$ under $\mathcal{T}$
and it is domain independent with respect to $\mathcal{T}$.

Constructive theorem

If:

1. $\mathcal{T} \cup \widetilde{\mathcal{T}} \models \forall \mathbb{X}.\, \mathcal{Q}_{[\mathbb{X}]} \leftrightarrow \widetilde{\mathcal{Q}}_{[\mathbb{X}]}$
   (that is, $\mathcal{Q}_{[\mathbb{X}]}$ is implicitly definable),
2. $\mathcal{Q}$ is safe-range
   (that is, $\mathcal{Q}$ is domain independent),
3. $\mathcal{T}$ is safe-range
   (that is, $\mathcal{T}$ is domain independent),

then there exists an exact reformulation $\widehat{\mathcal{Q}}$ of $\mathcal{Q}$ as a safe-range query in $\mathcal{FOL}(\mathbb{C}, \mathbb{P})$ over $\mathbb{P}_{\mathcal{D}}$ under $\mathcal{T}$, that can be obtained constructively via INTERPOLATION.

Example

Given: $\mathbb{P} = \{R, V_1, V_2, V_3, A\}$, $\mathbb{P}_{\mathcal{D}} = \{V_1, V_2, V_3\}$
where *Adom* is the active domain of $\mathcal{D}$,

$$\begin{aligned}\mathcal{T} = \{\ & \forall x, y.\ V_1(x, y) \leftrightarrow \exists z, v.\ R(z, x) \wedge R(z, v) \wedge R(v, y), \\ & \forall x, y.\ V_2(x, y) \leftrightarrow \exists z.\ R(x, z) \wedge R(z, y), \\ & \forall x, y.\ V_3(x, y) \leftrightarrow \exists z, v.\ R(x, z) \wedge R(z, v) \wedge R(v, y)\ \}\end{aligned}$$

$$\mathcal{Q}(x, y) = \exists z, v, u.\ R(z, x) \wedge R(z, v) \wedge R(v, u) \wedge R(u, y).$$

The conditions of the theorem are satisfied: $\mathcal{Q}(x, y)$ is implicitly definable from $\mathbb{P}_{\mathcal{D}}$ under $\mathcal{T}$; $\mathcal{Q}(x, y)$ is safe-range; $\mathcal{T}$ is safe-range.

Therefore, with the tableau method one finds the Craig's interpolant to compute $\widehat{\mathcal{Q}}(x, y)$ from a validity proof of the implication $(\mathcal{T} \wedge \mathcal{Q}) \rightarrow (\widetilde{\mathcal{T}} \rightarrow \widetilde{\mathcal{Q}})$ and obtain:

$$\widehat{\mathcal{Q}}(x, y) = \exists z.\ V_1(x, z) \wedge \forall v.\ (V_2(v, z) \rightarrow V_3(v, y))$$

an exact safe-range reformulation of $\mathcal{Q}(x, y)$ from $\mathbb{P}_{\mathcal{D}}$ under $\mathcal{T}$. □

Various Codd's theorems for description logics

- The domain independent fragments of $\mathcal{ALCHOI}$ and $\mathcal{SHOQ}$ and their guarded negation syntactic fragments are equally expressive, and they enjoy finitely controllable determinacy.
- The domain independent fragment of $\mathcal{SHOIQ}$ and its safe range syntactic fragment are equally expressive.

  - Non-guarded negation should not appear in a cleanly designed ontology, and, if present, it should be fixed.
  - The use of absolute negative information:
    "a non-`male` is a `female`"
    $\neg$ `male` $\sqsubseteq$ `female`;
    is not meaningful in conceptual modelling, since the subsumer includes all sorts of objects in the universe.

    Allow only guarded negative information in the subsumee:
    "a non-`male` `person` is a `female`"
    `person` $\sqcap \neg$ `male` $\sqsubseteq$ `female`.

Conclusion and Future Work

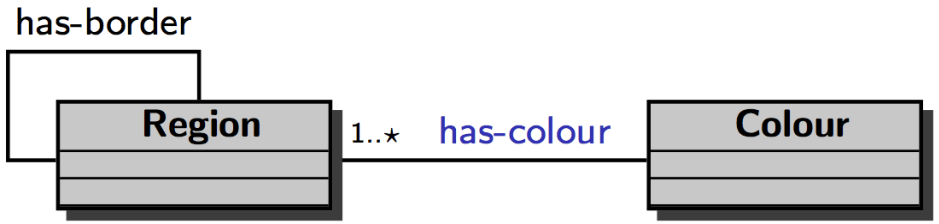
Conclusion:

- We introduced a framework to compute an exact reformulation of a concept query under a description logic ontology over some set of concept and role names (DBox predicates).
- We found the conditions which guarantee that a safe-range exact reformulation exists.
- We proved that such safe-range exact reformulation being evaluated as a relational algebra query over the DBox give the same answer as the original query under the ontology.
- An application of the framework to description logics was studied.

Future Work:

- Study optimisations of reformulations.
  - How to choose the *best* reformulation in terms of query evaluation?

Complexity of Query answering

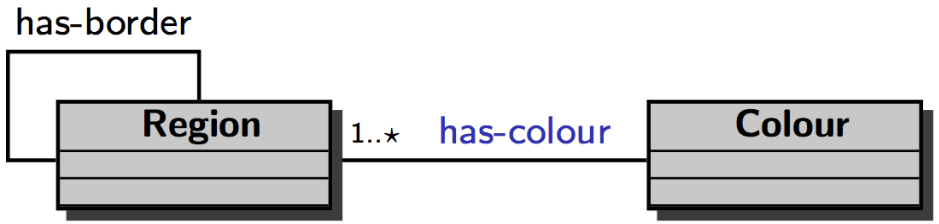


$\mathtt{Region} = \{\mathtt{Italy}, \mathtt{France}, \ldots\};$

$\mathtt{has\text{-}border} = \{\langle \mathtt{Italy}, \mathtt{France} \rangle, \ldots\};$

$\mathtt{Colour} = \{\ \mathtt{Red},\ \mathtt{Green},\ \mathtt{Blue}\ \}.$

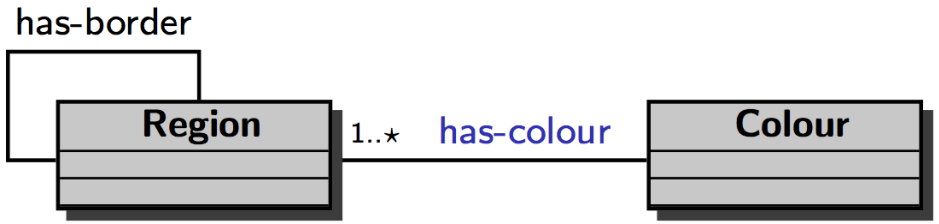
Complexity of Query answering




```
Region = {Italy,France,...};
has-border = {⟨Italy,France⟩,...};
Colour = { Red, Green, Blue }.
Q :- has-col(R1,C),has-col(R2,C),has-border(R1,R2).
```

Is there at least one map in which there are two adjacent regions with the same colour?

Complexity of Query answering

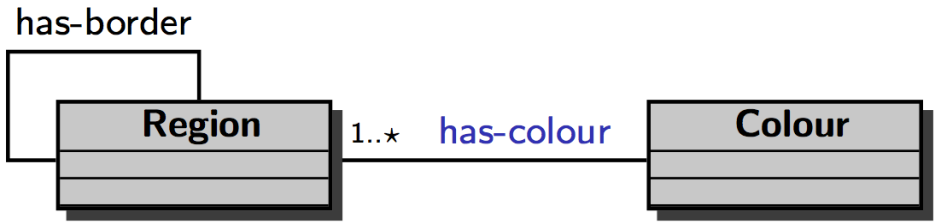



```
Region = {Italy,France,...};
has-border = {⟨Italy,France⟩,...};
Colour = { Red, Green, Blue }.
Q :- has-col(R1,C),has-col(R2,C),has-border(R1,R2).
```

Is there at least one map in which there are two adjacent regions with the same colour?

- YES: in any legal database (i.e., an assignment of colours to regions) there are at least two adjacent regions with the same colour.

Complexity of Query answering

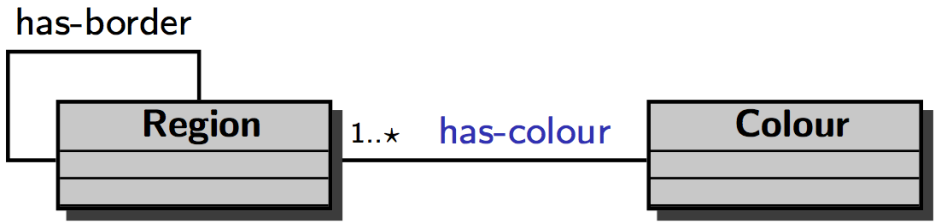



```
Region = {Italy,France,...};
has-border = {⟨Italy,France⟩,...};
Colour = { Red, Green, Blue }.
Q :- has-col(R1,C),has-col(R2,C),has-border(R1,R2).
```

Is there at least one map in which there are two adjacent regions with the same colour?

- YES: in any legal database (i.e., an assignment of colours to regions) there are at least two adjacent regions with the same colour.
- NO: there is at least a legal database (i.e., an assignment of colours to regions) in which no two adjacent regions have the same colour.

Complexity of Query answering

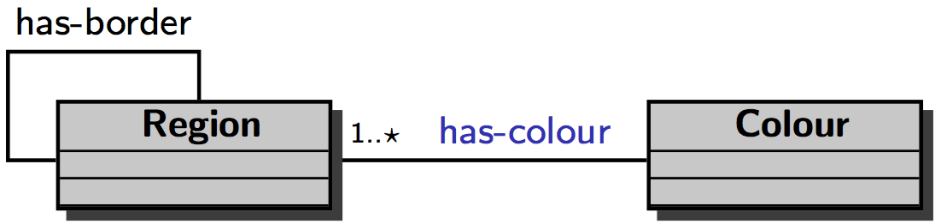



```
Region = {Italy,France,...};
has-border = {⟨Italy,France⟩,...};
Colour = { Red, Green, Blue }.
Q :- has-col(R1,C),has-col(R2,C),has-border(R1,R2).
```

Is there at least one map in which there are two adjacent regions with the same colour?

- YES: in any legal database (i.e., an assignment of colours to regions) there are at least two adjacent regions with the same colour.
- NO: there is at least a legal database (i.e., an assignment of colours to regions) in which no two adjacent regions have the same colour.
- With ABox semantics the answer is always NO, since there is at least a legal database (i.e., an assignment of colours to regions) with enough distinct colours so that no two adjacent regions have the same colour.

Complexity of Query answering




```
Region = {Italy,France,...};
has-border = {⟨Italy,France⟩,...};
Colour = { Red, Green, Blue }.
Q :- has-col(R1,C),has-col(R2,C),has-border(R1,R2).
```

Is there at least one map in which there are two adjacent regions with the same colour?

- YES: in any legal database (i.e., an assignment of colours to regions) there are at least two adjacent regions with the same colour.
- NO: there is at least a legal database (i.e., an assignment of colours to regions) in which no two adjacent regions have the same colour.
- With ABox semantics the answer is always NO, since there is at least a legal database (i.e., an assignment of colours to regions) with enough distinct colours so that no two adjacent regions have the same colour.

Query answering with DBoxes is co-np-hard in data complexity (3-col), and it is strictly harder than with ABoxes!

Query reformulation under expressive constraints: the role of determinacy and of domain independence

Enrico Franconi,
Paolo Guagliardo, Volha Kerhet, and Nhung Ngo

KRDB Research Centre, Free University of Bozen-Bolzano, Italy

iPRA
18 July 2014