Completeness Statements about RDF Data Sources and Their Use for Query Answering
+
Research Story

Fariz Darari

Werner Nutt
Giuseppe Pirrò
Simon Razniewski

Freie Universität Bozen
Libera Università di Bolzano
Free University of Bozen · Bolzano

EMCL Workshop 2014, Vienna

Slides about Research

In this color!

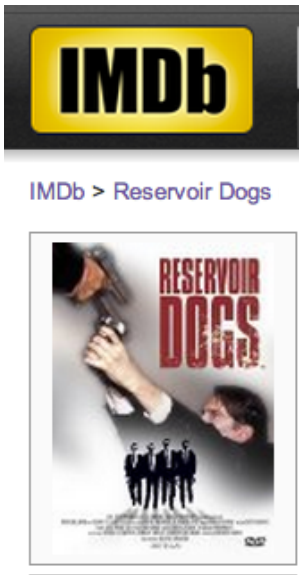
Slides about Research Story (Meta-Research)

In this color!

Story: Motivation

- I want to do a project and a thesis.
- I like Semantic Web.
- What can be a good research topic? Finding a good research topic (and good supervisor) is also part of research!
- Werner: “Hi, we are doing research about completeness reasoning on databases”
- Me: “Why not also on the Semantic Web?”

Motivation



Completeness statement about the IMDB data source

Quentin Tarantino was the character Mr. Brown

Motivation

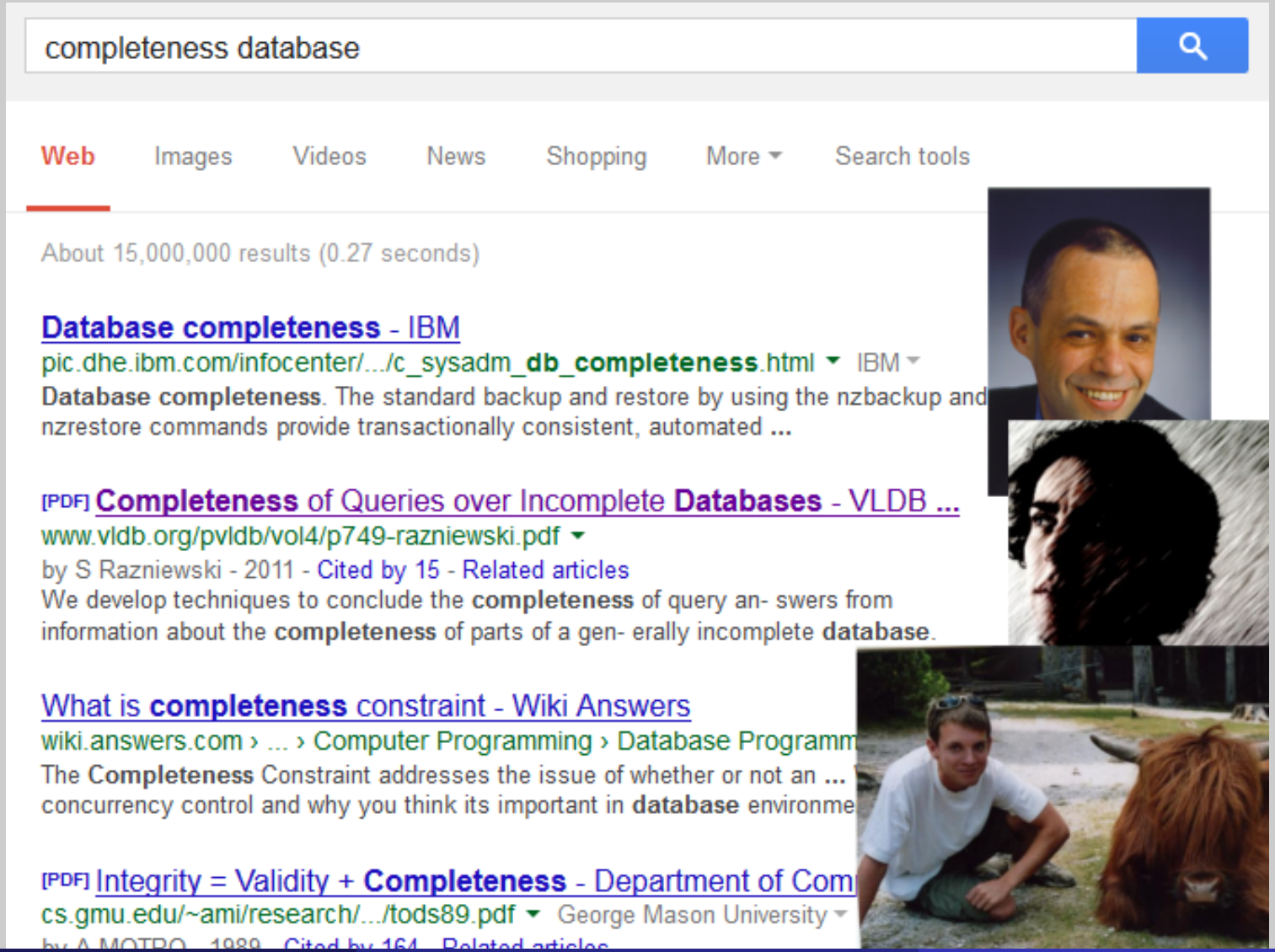

I want to find out
all movies directed by
and starring Tarantino
Syd
W3C SPARQL
LMDb
RDF
DBpedia
RDF
LMDb is complete
for Tarantino movies and
their actors
DBpedia is complete
for Tarantino movies

Story: Literature Study

- Hmm..for realizing my goal, what should I learn?
- Aha, I need to learn Semantic Web, I think the Semantic Web Technologies course offered by FUB would be useful for me
- Aha, I also need to learn existing work on completeness reasoning for databases, I guess this paper[1] is worth to read!

[1] Completeness of Queries over Incomplete Databases by Simon and Werner

Story: Google (and your supervisors) are your Googles, ask them!



Story: Start Doing Real Research – Problem Understanding

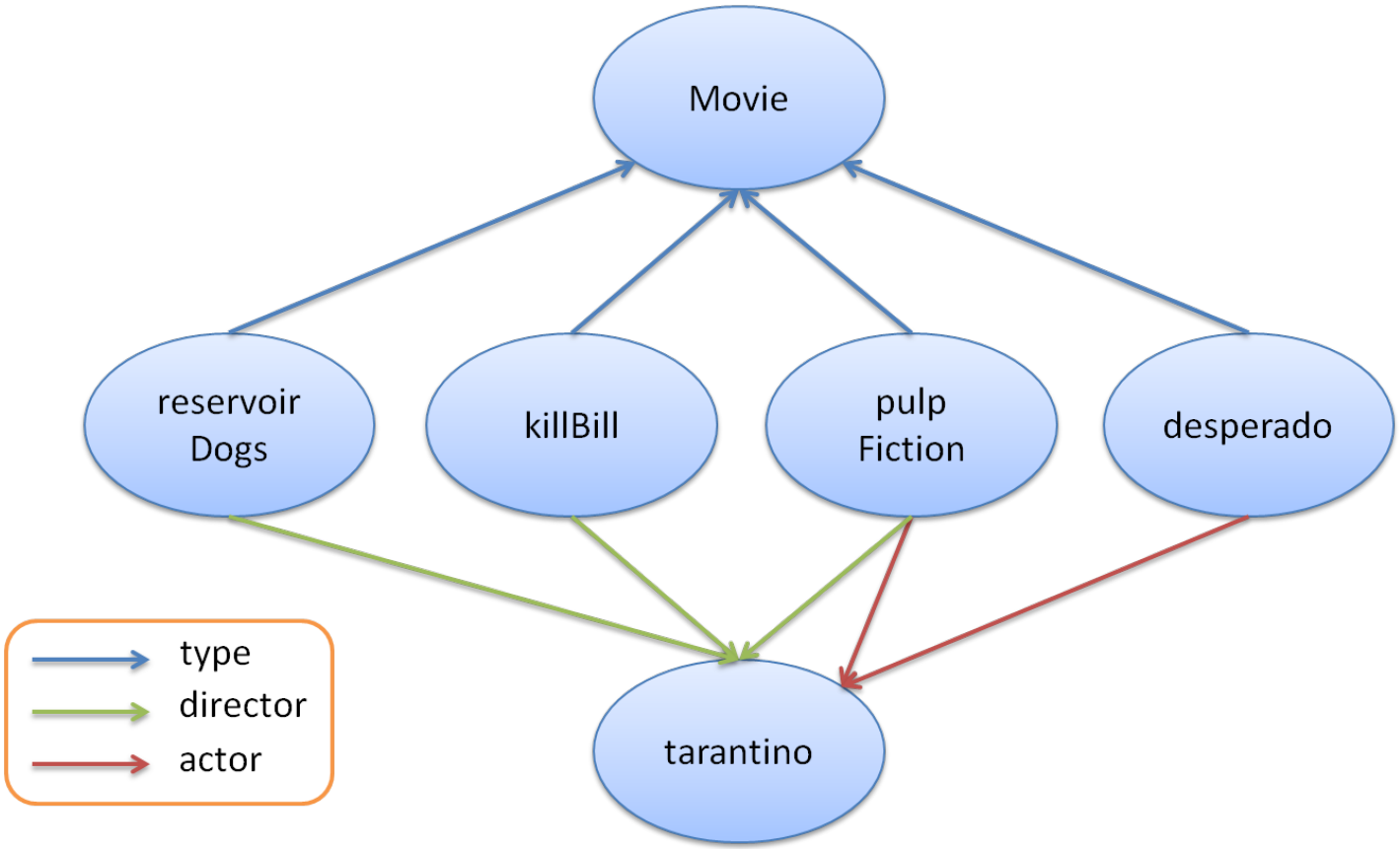
- Completeness reasoning on databases has been investigated
- Hmm..but databases are different than Linked Data, what should be adjusted?
- Well, in Linked Data, instead of databases, we have RDF data sources
- Well, in Linked Data, instead of SQL, we have SPARQL as a query language
- Well, Linked Data also is more open, more heterogeneous and federated
- Well, Linked Data also has ontologies
- Then, I have to discuss these issues with my supervisors!

Incomplete Data Source

Quentin Tarantino is missing..

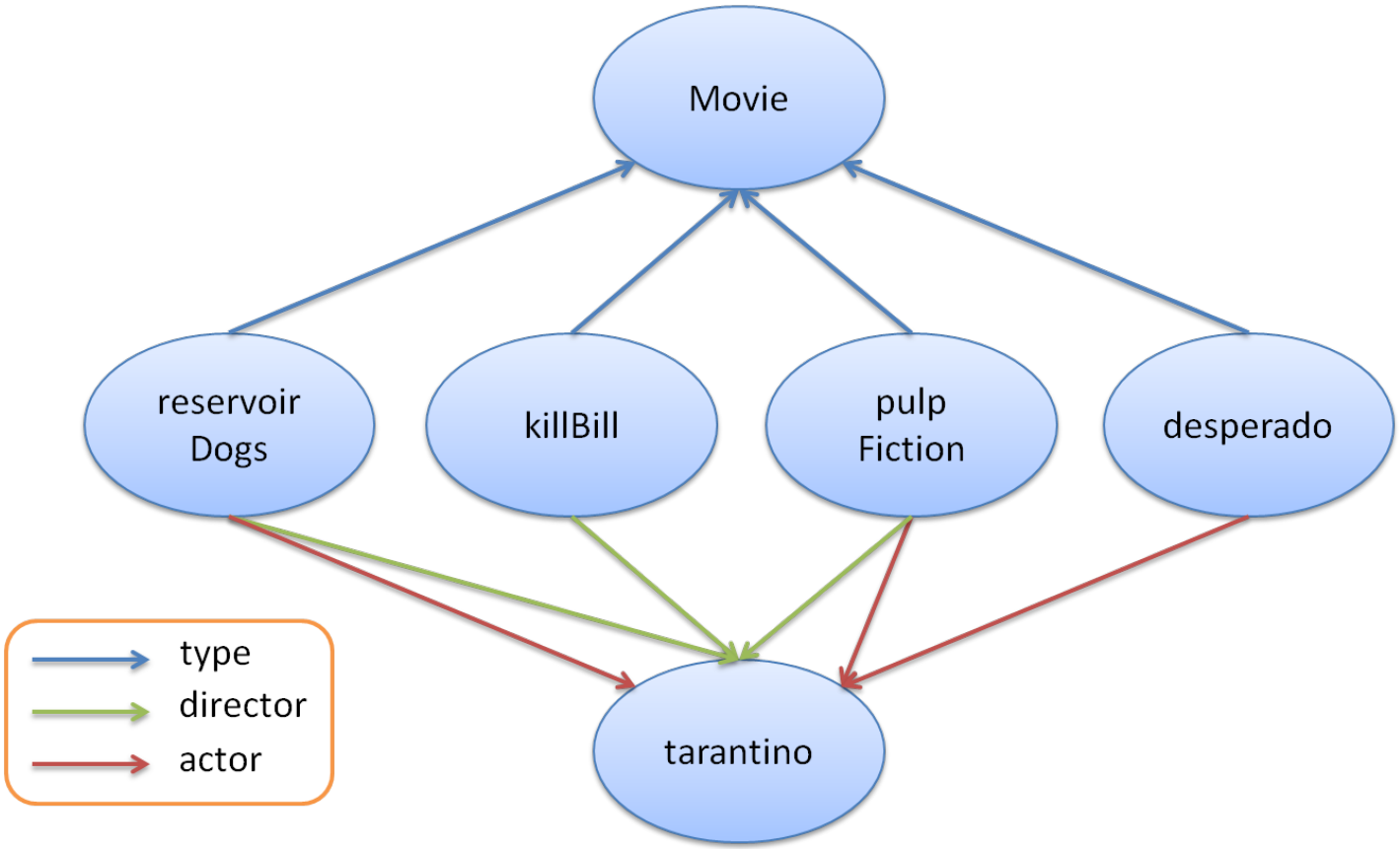
Incomplete Data Source

An incomplete data source of Tarantino movies, $\mathcal{G}_{dbp} = (G^a_{dbp}, G^i_{dbp})$:



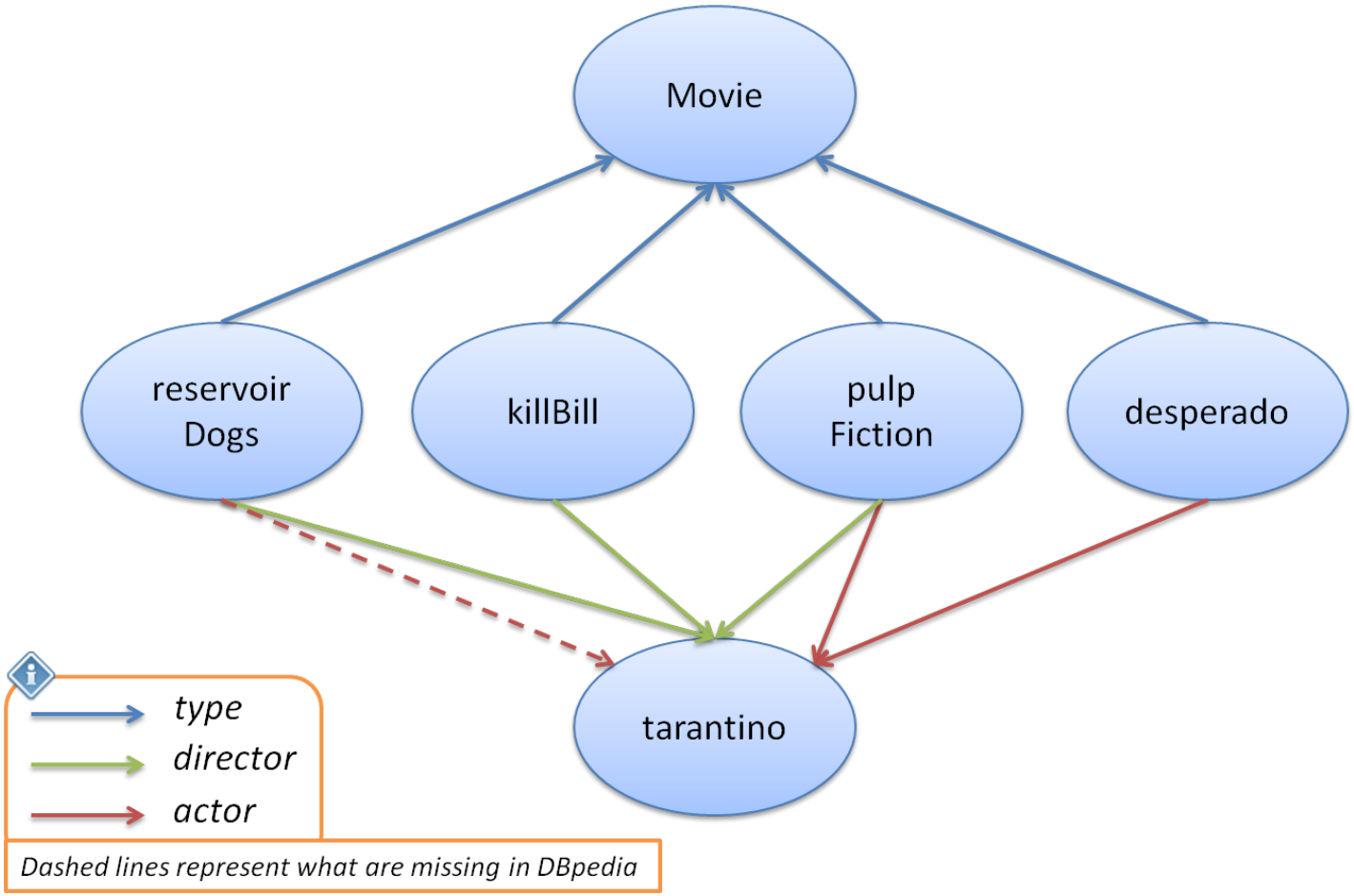
Incomplete Data Source

An incomplete data source of Tarantino movies, $\mathcal{G}_{dbp} = (G^a_{dbp}, G^i_{dbp})$:

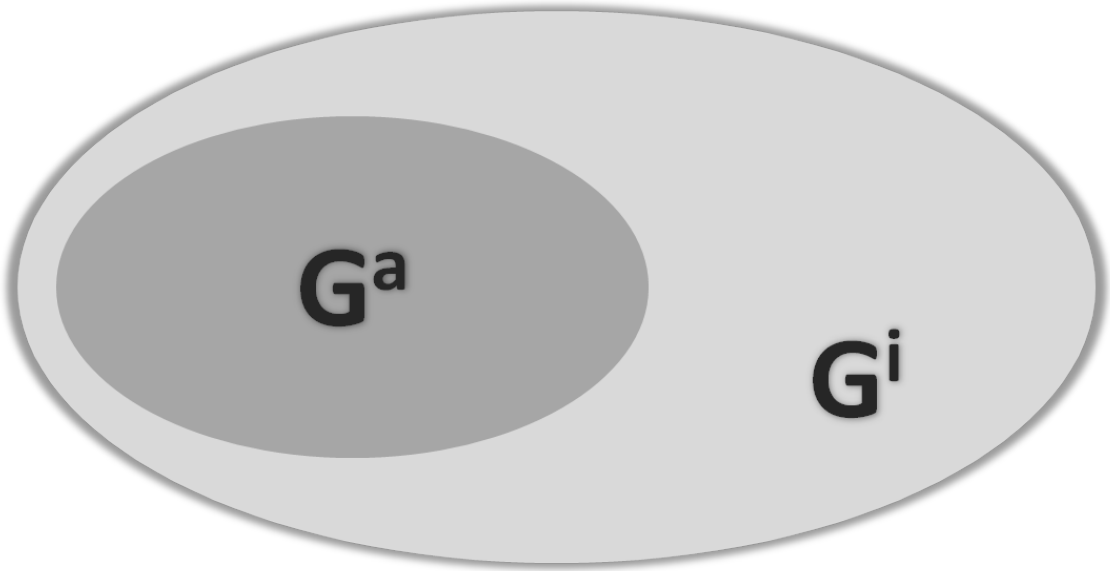


Incomplete Data Source

An incomplete data source of Tarantino movies, $\mathcal{G}_{dbp} = (G^a_{dbp}, G^i_{dbp})$:



Incomplete Data Source



Incomplete Data Source

An **incomplete data source** is a pair of two graphs,

$$\mathcal{G} = (G^a, G^i), \text{ where } G^a \subseteq G^i.$$

We call G^a the **available** graph and G^i the **ideal** graph.

Completeness Statements: Examples

To express that a source is complete
for all the triples about movies directed by Tarantino,
we use the statement

$$C_{dir} = Compl((?m, type, Movie), (?m, director, tarantino) \mid \emptyset),$$

Completeness Statements: Examples

To express that a source is complete
for all the triples about movies directed by Tarantino,
we use the statement

$$C_{dir} = Compl((?m, type, Movie), (?m, director, tarantino) \mid \emptyset),$$

To express that a source is complete
for all triples about actors in movies directed by Tarantino,
we use

$$C_{act} =$$

$$Compl((?m, actor, ?a) \mid (?m, type, Movie), (?m, director, tara))$$

Completeness Statement: Definition

Let P_1 be a non-empty BGP (Basic Graph Pattern) and P_2 a BGP.
A **completeness statement** is defined as

$$Compl(P_1 \mid P_2)$$

where we call P_1 the **pattern** and P_2 the **condition** of the statement.

Satisfaction of Completeness Statements

To the statement

$$C = \mathit{Compl}(P_1 \mid P_2),$$

we associate the `CONSTRUCT` query

$$Q_C = (P_1, P_1 \cup P_2).$$

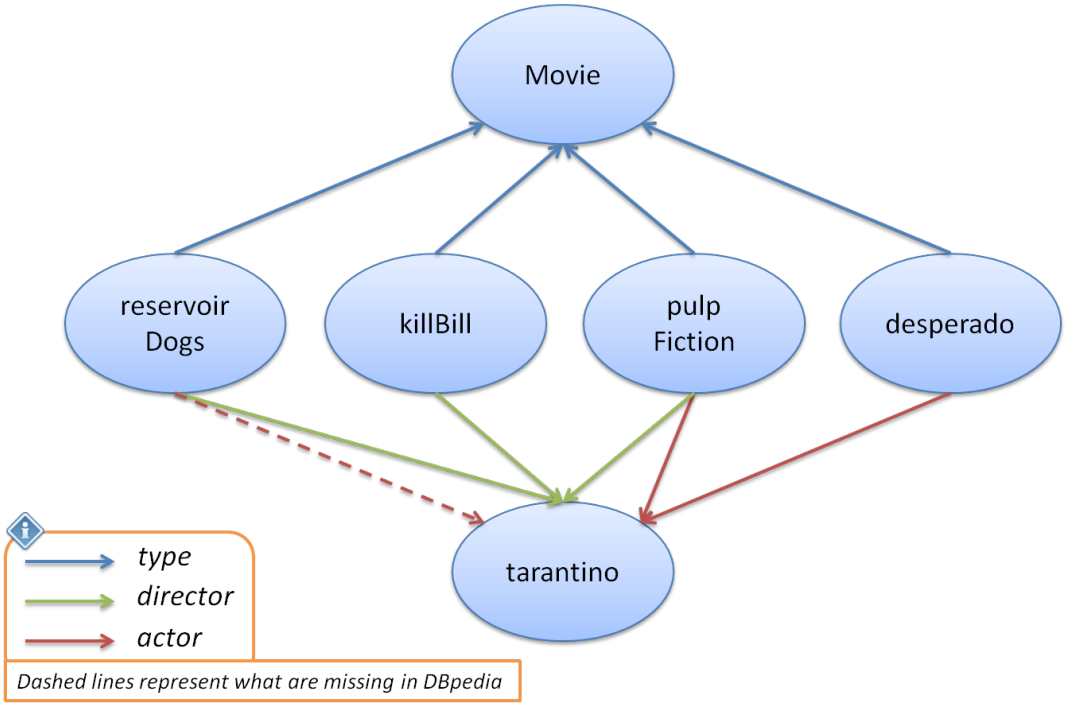
Then, we say:

C is **satisfied** by an IDS $\mathcal{G} = (G^a, G^i)$, written $\mathcal{G} \models C$, if

$$[\![Q_C]\!]_{G^i} \subseteq G^a.$$

Completeness Statements

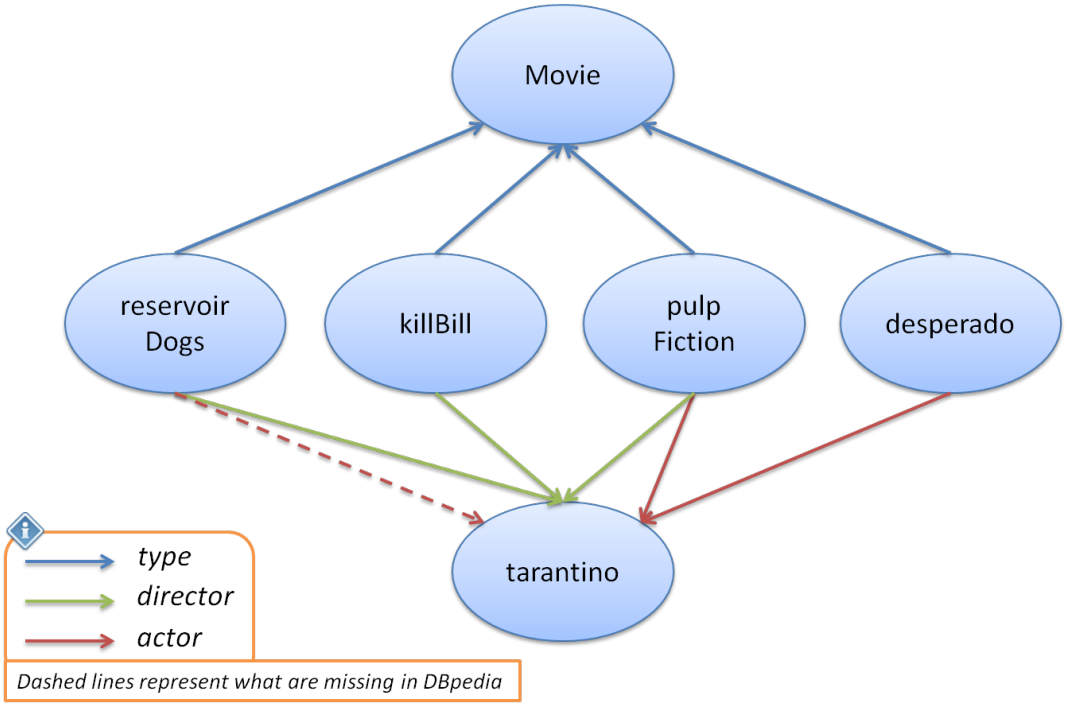
$$C_{dir} = Compl((?m, type, Movie), (?m, director, tarantino) \mid \emptyset)$$



Question: $\mathcal{G}_{dbp} \models C_{dir}$?

Completeness Statements

$$C_{dir} = Compl((?m, type, Movie), (?m, director, tarantino) \mid \emptyset)$$

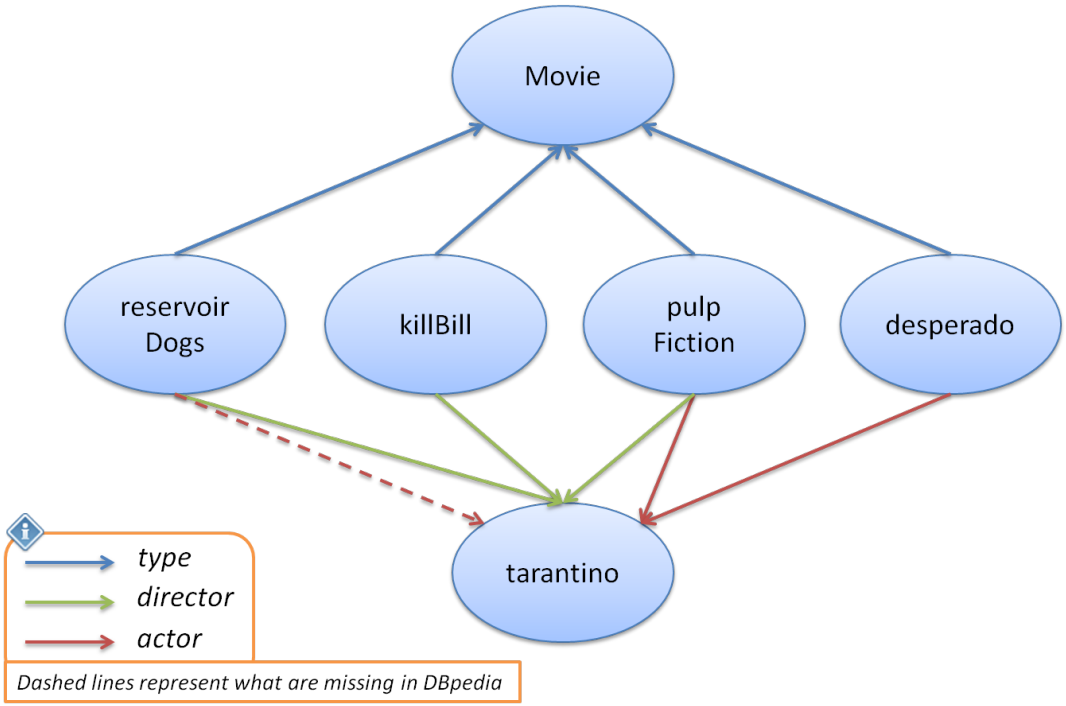


Question: $\mathcal{G}_{dbp} \models C_{dir}$?

Yes, because $[\![Q_{C_{dir}}]\!]_{G^i_{dbp}} \subseteq G^a_{dbp}$.

Completeness Statements

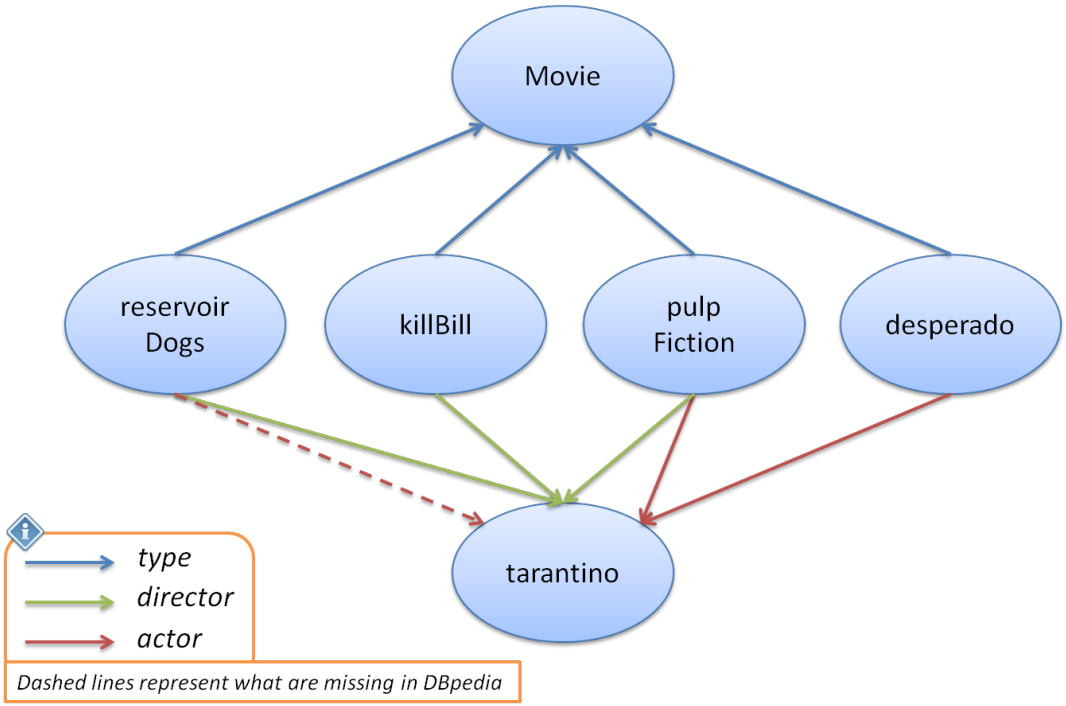
$C_{act} = Compl((?m, actor, ?a) \mid (?m, type, Movie), (?m, director, tara$



Question: $\mathcal{G}_{dbp} \models C_{act}$?

Completeness Statements

$C_{act} = Compl((?m, actor, ?a) \mid (?m, type, Movie), (?m, director, tara$



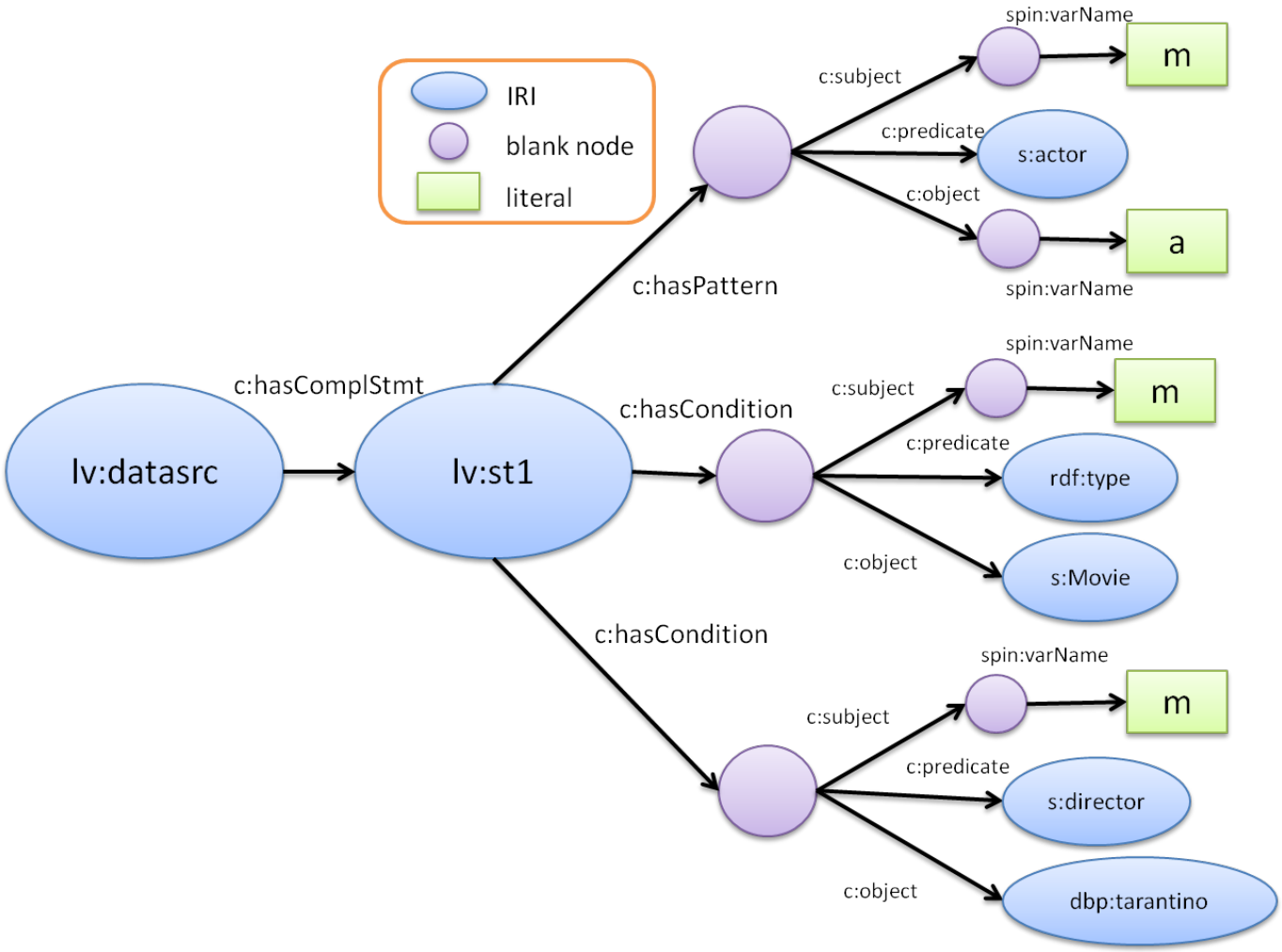
Question: $\mathcal{G}_{dbp} \models C_{act}$?

No, because among the results of $[\![Q_{C_{act}}]\!]_{G^i_{dbp}}$,

there is $(reservoirDogs, actor, tarantino)$ not in G^a_{dbp}.

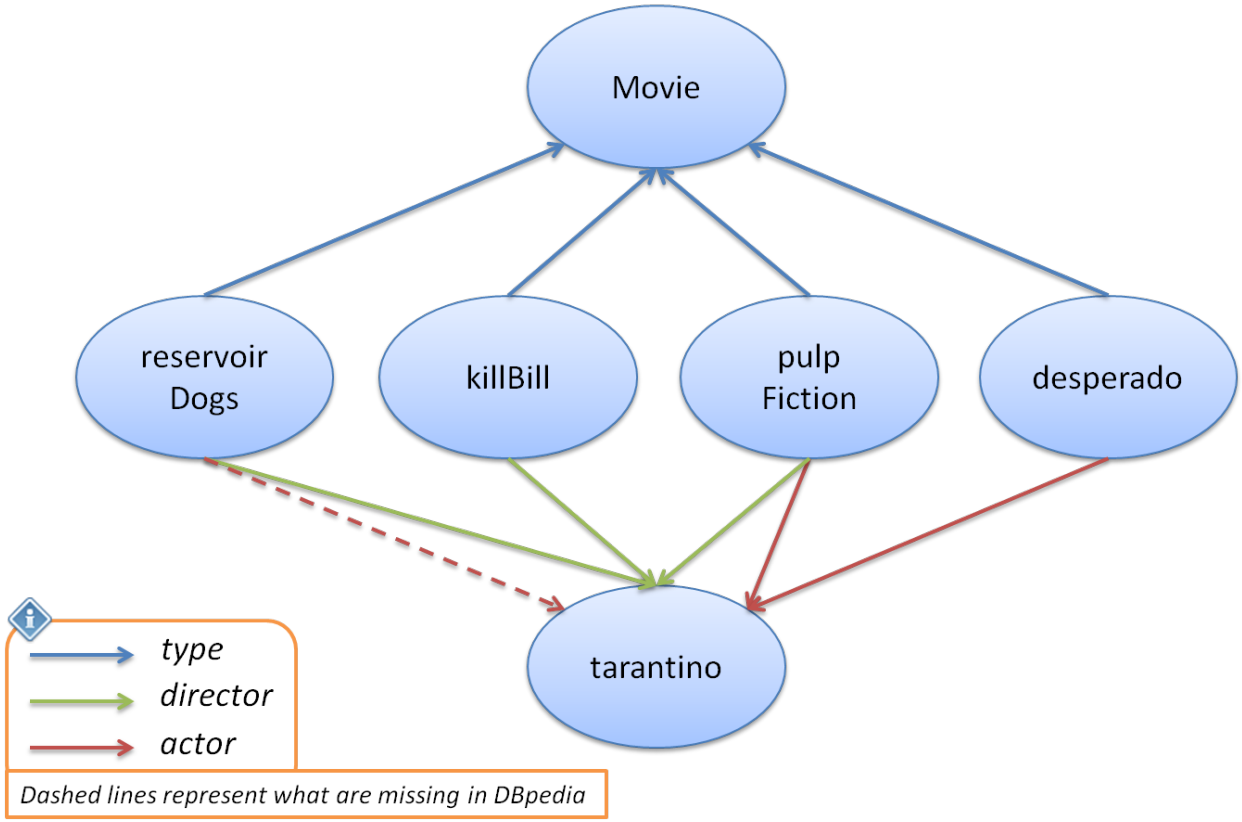
Completeness Statements in RDF

$C_{act} = Compl((?m, actor, ?a) \mid (?m, type, Movie), (?m, director, tara)$



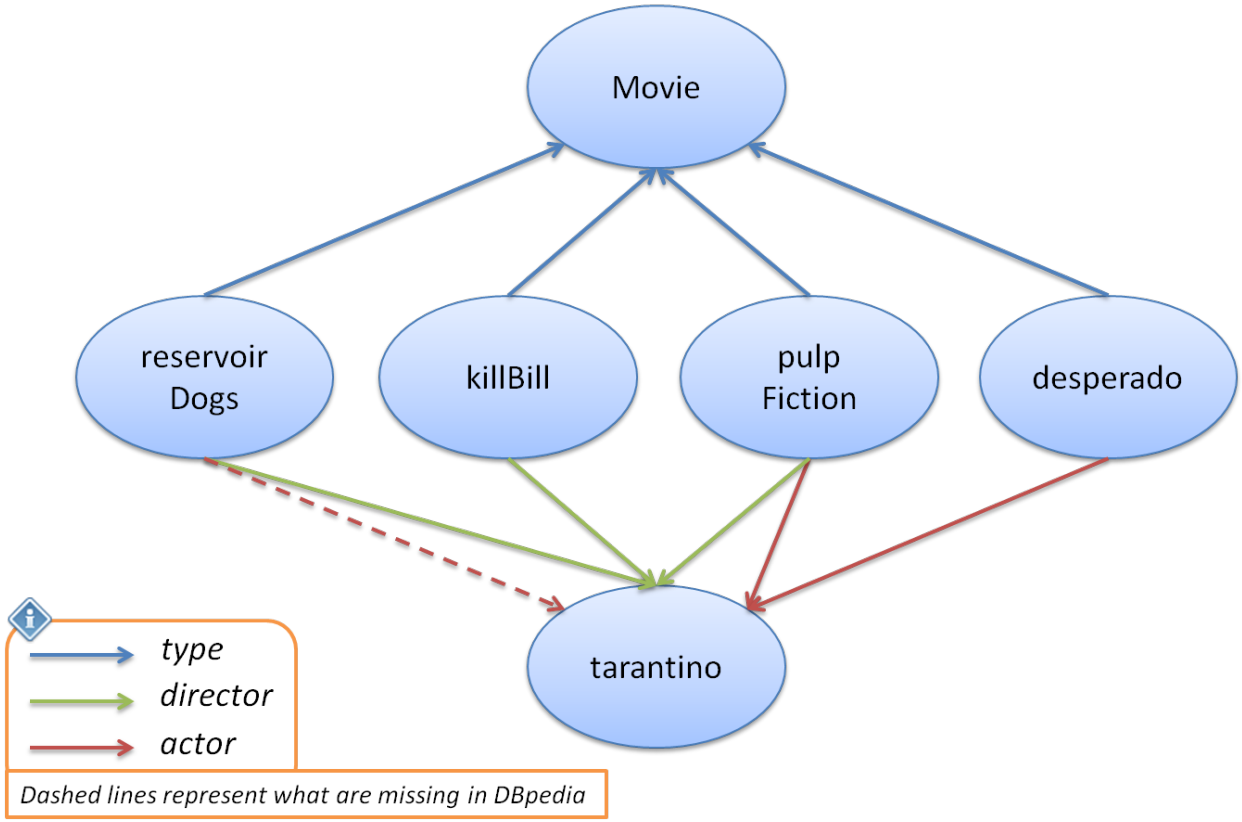
Query Completeness: Example

$$Q_{dir} = (\{?m\}, \{(?m, type, Movie), (?m, director, tarantino)\})$$



Query Completeness: Example

$$Q_{dir} = (\{?m\}, \{(?m, type, Movie), (?m, director, tarantino)\})$$



Then, $[\![Q_{dir}]\!]_{G^i_{dbp}} = [\![Q_{dir}]\!]_{G^a_{dbp}}$, and therefore $\mathcal{G}_{dbp} \models Compl(Q_{dir})$.

Query Completeness: Definition

Definition

Let Q be a `SELECT` query. We write

$$Compl(Q)$$

to say that Q **is complete**.
An incomplete data source $\mathcal{G} = (G^a, G^i)$ **satisfies** $Compl(Q)$, written

$$\mathcal{G} \models Compl(Q) \qquad \text{if and only if} \qquad [\![Q]\!]_{G^i} = [\![Q]\!]_{G^a}.$$

Story: Defining Theorems to Prove

- Theorems are formal representations of goals you want to achieve (remember the first slides about motivation)
- In fact..all the definitions and framework introduced before are actually to well-define these theorems
- Hmm..I am sure these theorems should hold!
- Then, I have to discuss these issues with my supervisors!

Completeness Entailment

Let $\mathcal{C}$ be a set of completeness statements and
Q a `SELECT` query.
We say that $\mathcal{C}$ **entails the completeness of** Q, written

$$\mathcal{C} \models \mathit{Compl}(Q),$$

if any incomplete data source satisfying $\mathcal{C}$
also satisfies $\mathit{Compl}(Q)$.

Completeness Reasoning

Transfer Operator

For any set $\mathcal{C}$ of completeness statements and a graph G, we define the **transfer operator** $T_{\mathcal{C}}$:

$$T_{\mathcal{C}}(G) = \bigcup_{c \in \mathcal{C}} [\![Q_c]\!]_G$$

Completeness Reasoning

Transfer Operator

For any set $\mathcal{C}$ of completeness statements and a graph G, we define the **transfer operator** $T_{\mathcal{C}}$:

$$T_{\mathcal{C}}(G) = \bigcup_{c \in \mathcal{C}} [\![Q_c]\!]_G$$

Prototypical Graph

Let $Q = (W, P)$ be a `SELECT` query.
The **prototypical graph** $\tilde{P}$ is the graph resulting from the mapping of variables in P to fresh, unique IRIs.

Completeness of Basic Queries

Theorem

Let $\mathcal{C}$ be a set of completeness statements and $Q = (W, P)$ a basic query. Then,

$$\mathcal{C} \models \mathit{Compl}(Q) \qquad \text{if and only if} \qquad \tilde{P} = T_{\mathcal{C}}(\tilde{P}).$$

Story: Start Doing The Hardcore Part - Proving Theorems

- Argh, I got stuck
- This proving theorems stuff always haunts me!
- But:

Because you are Excellent Marvelous Champion Legendary (aka EMCL)

Story: Start Doing The Hardcore Part - Proving Theorems (2)

- Get the proof idea first
- Generate some concrete examples of the problems with their solutions from the theorem you want to prove
- Discuss with your supervisors
- Now, start to prove in detail
- There is a problem on this, and that, then refine your proof idea, go back to the first point
- Start doing the above points until (hopefully :) you are able to prove it

Completeness Reasoning: Example

Consider the set of completeness statements

$$\mathcal{C}_{dir,act} = \{ C_{dir}, C_{act} \}$$

and the query

$$Q_{dir+act} = (\{ ?m \}, P_{dir+act})$$

where

$$P_{dir+act} =$$
$$\{ (?m, type, Movie), (?m, director, tarantino), (?m, actor, tarantino) \}.$$

Completeness Reasoning: Example

Consider the set of completeness statements

$$\mathcal{C}_{dir,act} = \{ C_{dir}, C_{act} \}$$

and the query

$$Q_{dir+act} = (\{ ?m \}, P_{dir+act})$$

Then,

$$\tilde{P}_{dir+act} =$$
$$\{ (c_{?m}, type, Movie), (c_{?m}, director, tarantino), (c_{?m}, actor, tarantino) \}$$

Completeness Reasoning: Example

$\mathcal{C}_{dir,act} = \{ C_{dir}, C_{act} \}$

$Q_{dir+act} = (\{ ?m \}, P_{dir+act})$

Therefore,

$$\mathcal{T}_{\mathcal{C}_{dir,act}}(\tilde{P}_{dir+act})$$

Completeness Reasoning: Example

$$\mathcal{C}_{dir,act} = \{ C_{dir}, C_{act} \}$$

$$Q_{dir+act} = (\{ ?m \}, P_{dir+act})$$

Therefore,

$$\mathcal{T}_{\mathcal{C}_{dir,act}}(\tilde{P}_{dir+act}) = [\![Q_{C_{dir}}]\!]_{\tilde{P}_{dir+act}} \cup [\![Q_{C_{act}}]\!]_{\tilde{P}_{dir+act}}$$

Completeness Reasoning: Example

$$\mathcal{C}_{dir,act} = \{ C_{dir}, C_{act} \}$$

$$Q_{dir+act} = (\{ ?m \}, P_{dir+act})$$

Therefore,

$$\mathcal{T}_{\mathcal{C}_{dir,act}}(\tilde{P}_{dir+act}) = [\![Q_{C_{dir}}]\!]_{\tilde{P}_{dir+act}} \cup [\![Q_{C_{act}}]\!]_{\tilde{P}_{dir+act}}$$

$$= \{ (c_{?m}, type, Movie), (c_{?m}, director, tara),$$

Completeness Reasoning: Example

$\mathcal{C}_{dir,act} = \{ C_{dir}, C_{act} \}$

$Q_{dir+act} = (\{ ?m \}, P_{dir+act})$

Therefore,

$$\mathcal{T}_{\mathcal{C}_{dir,act}}(\tilde{P}_{dir+act})$$
$$= [\![Q_{C_{dir}}]\!]_{\tilde{P}_{dir+act}} \cup [\![Q_{C_{act}}]\!]_{\tilde{P}_{dir+act}}$$
$$= \{ (c_{?m}, type, Movie), (c_{?m}, director, tara), (c_{?m}, actor, tara) \}$$

Completeness Reasoning: Example

$\mathcal{C}_{dir,act} = \{ C_{dir}, C_{act} \}$

$Q_{dir+act} = (\{ ?m \}, P_{dir+act})$

Therefore,

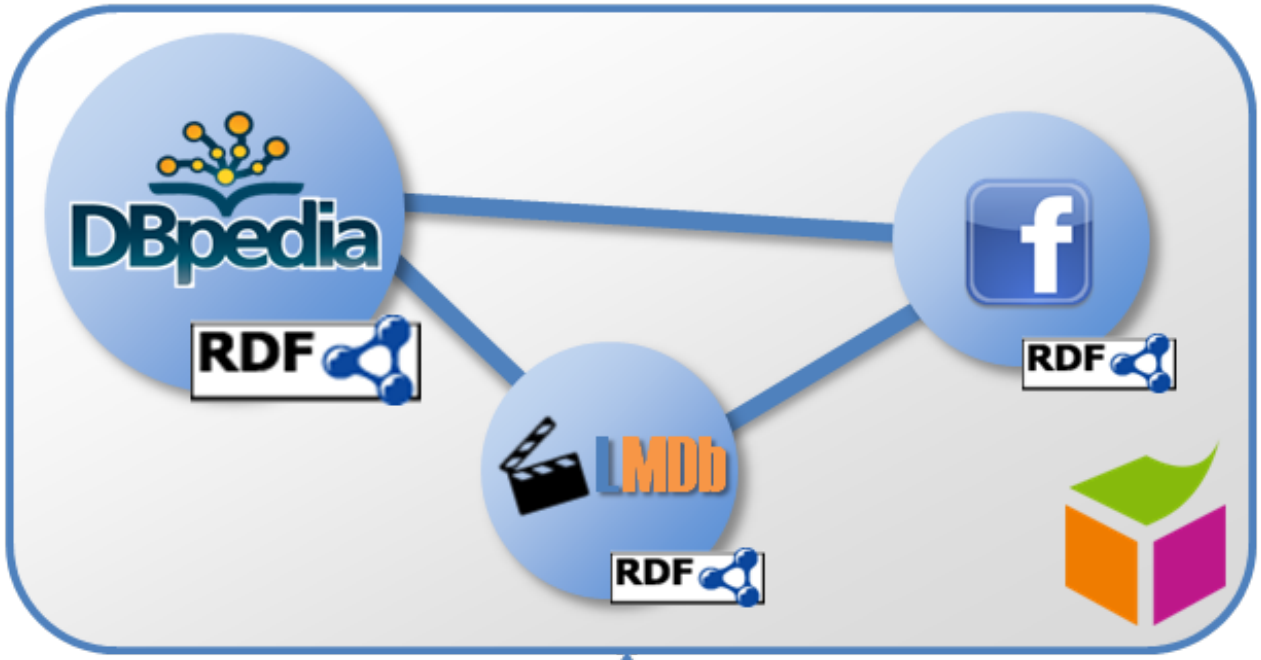
$$\begin{gathered} \mathcal{T}_{\mathcal{C}_{dir,act}}(\tilde{P}_{dir+act}) \\ = [\![Q_{C_{dir}}]\!]_{\tilde{P}_{dir+act}} \cup [\![Q_{C_{act}}]\!]_{\tilde{P}_{dir+act}} \\ = \{ (c_{?m}, type, Movie), (c_{?m}, director, tara), (c_{?m}, actor, tara) \} \\ = \tilde{P}_{dir+act}. \end{gathered}$$

Thus, $\mathcal{C}_{dir,act} \models Compl(Q_{dir+act})$

The framework can also be applied to:

- `DISTINCT` Queries: with set semantics
- `OPT` Queries: eg. get all people and in case they are not single, get also their spouse
- Queries with RDFS Data Sources: incorporating ontologies

Federated Framework



Can I get a complete answer?

```
SELECT  ?m ?l
WHERE { ?m rdf:type s:Movie .
        ?m s:director dbp:tarantino .
        ?m fbo:likes ?l }
```

CoRner: Completeness Reasoner

- Can check the completeness of a subset of SPARQL queries depending on the completeness statements of a single data source.
- Developed in Java using the Apache Jena library.
- Takes three inputs:
  - Completeness statements in RDF format
  - A SPARQL query
  - (optional) an RDFS ontology
- Available at `http://rdfcorner.wordpress.com/`.

Story: Cooling Down, or is it?

- I have got all the results
- But, I haven't started writing the thesis yet
- Nooo, I will miss the deadline :(

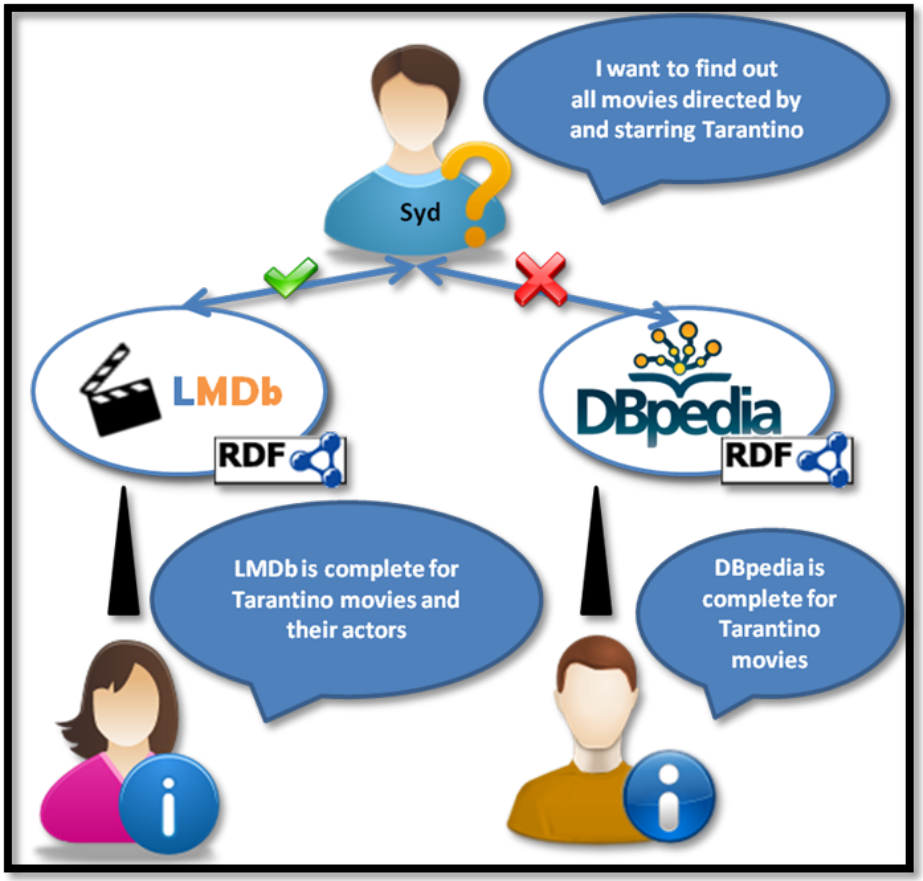
Story: Cooling Down - Alternative

- I have got all the results
- And, I have started writing the thesis since the very beginning (right after I have studied the literature)
- I think I can manage to meet the deadline :)
- Aha, I think our research results can be also useful if shown to the Semantic Web community, why not to try submit it at ISWC 2013? – Another research story :)

Conclusions and Future Work

- We developed a theoretical framework for the expression of completeness statements about data sources, from which we can ensure query completeness.
- We provide a compact RDF representation of completeness statements, which can be embedded in `VoID` descriptions, and implemented the framework in `CoRner`.
- We are interested in studying completeness reasoning with more expressive queries and OWL 2.

Terima kasih, grazie, danke!!



Link to References