



ONTOLOGIES FOR KNOWLEDGE GRAPHS?

Markus Krötzsch[†]

reporting on joint work with Stefan Bischoff, Fredo Erxleben, Michael Günther, Maximilian Marx[†], Julian Mendez, Ana Ozaki[†], Axel Polleres, Sebastian Rudolph, Veronika Thost[†], and Denny Vrandečić

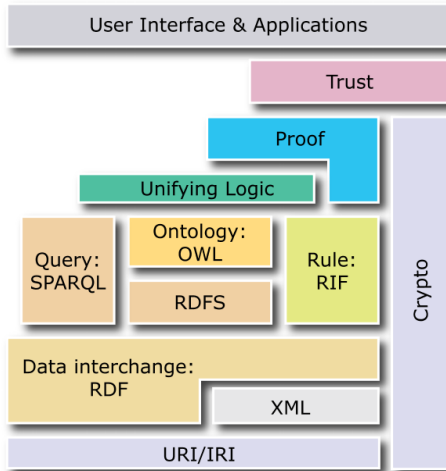
[†] Knowledge-Based Systems

TU Dresden

Full paper: <https://iccl.inf.tu-dresden.de/web/DL2017-keynote>

DL Workshop 2017

The Semantic Web (2007)



2012: The Knowledge Graph

Google Inside Search

Home How Search Works Tips & Tricks **Features** Search Stories Playground Blog Help

The Knowledge Graph

Learn more about one of the key breakthroughs behind the future of search.

See it in action

Discover answers to questions you never thought to ask, and explore collections and lists.

Leonardo da Vinci

Leonardo di ser Piero da Vinci was an Italian Renaissance polymath: painter, sculptor, architect, musician, scientist, mathematician, engineer, inventor, anatomist, geologist, cartographer, botanist, and writer.

Born: April 15, 1452, Anchiano
Died: May 2, 1519, Clos Lucé
Buried: Chateau d'Amboise

Parents: Caterina da Vinci, Piero da Vinci
Structures: [Wikipedia:Leonardo da Vinci Project](#)

“... one of the key breakthroughs behind the future of search”

More Knowledge Graphs



schema.org



Freebase™



yago
select knowledge

BIO2RDF



W3C SPARQL

What is a Knowledge Graph?

More than “a database used in an AI application”?

What is a Knowledge Graph?

More than “a database used in an AI application”?

Charateristics of today's KGs:

Normalised: Data decomposed into small units (“edges”)

Connected: Knowledge represented by relationships between these units

Annotated: Enriched with contextual information to record meta-data and auxiliary details

What is a Knowledge Graph?

More than “a database used in an AI application”?

Charateristics of today's KGs:

Normalised: Data decomposed into small units (“edges”)

Connected: Knowledge represented by relationships between these units

Annotated: Enriched with contextual information to record meta-data and auxiliary details

- Typical for many KG applications
- Often comes with a promise of declarative processing

Summary

Knowledge graphs

- introduce **graph-based** data models
- requiring **declarative** analytics
- that make **non-local** connections

Summary

Knowledge graphs

- introduce graph-based data models
- requiring declarative analytics
- that make non-local connections

} reasoning on graphs

Summary

Knowledge graphs

- introduce graph-based data models
- requiring declarative analytics
- that make non-local connections

} reasoning on graphs

Conclusion

Symbolic KR is the **key technology**
in modern data management
especially in AI applications

Summary

Knowledge graphs

- introduce graph-based data models
- requiring declarative analytics
- that make non-local connections

} reasoning on graphs

Conclusion

Symbolic KR is the key technology
in model management
especially in AI applications

**Not really
happening**



A Free Knowledge Graph



Wikidata

- Wikipedia's knowledge graph
- Free, community-built database
- Large graph
(July 2017: >165M statements on >29M entities)
- Large, active community
(July 2017: >175,000 logged-in human editors)
- Many applications

Freely available, relevant, and active knowledge graph

[Vrandečić & K; Comm. ACM 2014]

I'm in ur phone ...

Who is Grover Cleveland

Tap to Edit >

OK. Check it out:



KNOWLEDGE

Grover Cleveland

22nd and 24th president of the United States



Stephen Grover Cleveland was an American politician and lawyer who was the 22nd and 24th President of the United States. He won the popular vote for three presidential elections – in 1884, 1888, and 1892 – and was one of two Democrats to be elected president during the era of Republican political domination dating from 1861 to 1933. He was also the first and to date only President in American history to serve two non-consecutive terms in office.

See More on Wikipedia



Date of birth

March 18, 1837

Birthplace

Caldwell >

Date of death

June 24, 1908

Deathplace

Princeton >

Tim Berners-Lee (Q80)

British computer scientist

 [edit](#)

[TimBL](#) | [Sir Tim Berners-Lee](#) | [Timothy John Berners-Lee](#) | [TBL](#) | [Tim Berners Lee](#) | [T. Berners-Lee](#) | [T Berners-Lee](#) | [Tim Berners-Lee](#) | [T.J. Berners-Lee](#)

Tim Berners-Lee (Q80)

British computer scientist

 [edit](#)

[TimBL](#) | [Sir Tim Berners-Lee](#) | [Timothy John Berners-Lee](#) | [TBL](#) | [Tim Berners Lee](#) | [T. Berners-Lee](#) | [T Berners-Lee](#) | [Tim Berners-Lee](#) | [T.J. Berners-Lee](#)



instance of



[human](#)

 [edit](#)

[▶ 1 reference](#)

Tim Berners-Lee (Q80)

British computer scientist

 [edit](#)

[TimBL](#) | [Sir Tim Berners-Lee](#) | [Timothy John Berners-Lee](#) | [TBL](#) | [Tim Berners Lee](#) | [T. Berners-Lee](#) | [T Berners-Lee](#) | [Tim Berners-Lee](#) | [T.J. Berners-Lee](#)

⋮

instance of



human

 [edit](#)

[▶ 1 reference](#)

⋮

employer



CERN

 [edit](#)

start time

1984

end time

1994

position held

Fellow

[▼ 0 references](#)

[+ add reference](#)

Tim Berners-Lee (Q80)

British computer scientist

 edit

[TimBL](#) | [Sir Tim Berners-Lee](#) | [Timothy John Berners-Lee](#) | [TBL](#) | [Tim Berners Lee](#) | [T. Berners-Lee](#) | [T Berners-Lee](#) | [Tim Berners-Lee](#) | [T.J. Berners-Lee](#)

⋮

instance of



human

 edit

▶ 1 reference

⋮

employer



CERN

 edit

start time

1984

end time

1994

position held

Fellow

▼ 0 references

+ add reference

⋮

award received



Queen Elizabeth Prize for Engineering

 edit

point in time

2013

together with

[Robert Kahn](#)

[Vint Cerf](#)

[Louis Pouzin](#)

[Marc Andreessen](#)

▶ 1 reference

Statements in Wikidata

Wikidata's basic information units

- Built from **Wikidata items** (“CERN”, “Vint Cerf”), **Wikidata properties** (“award received”, “end time”), and **data values** (“2013”)
- Based on **directed edges**
 (“Tim Berners-Lee –employer→ CERN”)
- **Annotated** with property-value pairs (“end time: 1994”)
 - same property can have multiple annotation values
 (“together with: Robert Kahn, Vint Cerf, . . . ”)
 - same properties/values used in directed edges and annotations
- Items and properties can be subjects/values in statements
- **Multi-graph**

Elizabeth Taylor (Q34851)

Elizabeth Rosemond Taylor | Liz Taylor | Dame Elizabeth Rosemond Taylor

British-American actress

instance of: Elizabeth Taylor is a(n) human

Human relationships	
Own statements	From related entities
spouse	Larry Fortensky (construction worker and seventh husband of Elizabeth Taylor) > end time : 1996-10-31 start time : 1991-10-06
8 statements v	John Warner (Republican politician and Secretary of the Navy from the United States) > end time : 1982-11-07 start time : 1976-12-04
	Richard Burton (Welsh actor) > start time : 1975-10-10 end time : 1976-07-29
	Richard Burton (Welsh actor) > start time : 1964-03-15 end time : 1974-06-26
	Eddie Fisher (American entertainer and singer) > end time : 1964-03-06 start time : 1959-05-12
	Mike Todd (American theatre and film producer) > end time : 1958-03-22 start time : 1957-02-02
	Michael Wilding (English television and film actor) > end time : 1957-01-30 start time : 1952-02-21
	Conrad Hilton, Jr. (American hotelier) > end time : 1951-01-29 start time : 1950-05-06

edit label



Links

[Wikidata page](#)

[Official website](#)

[Wikipedia article](#)

[Reasonator](#)

Identifiers

SFDb person ID 75200 [↗](#) >

Elonet person ID 224907 [↗](#) >

PORT person ID 7869 [↗](#) >

AllMovie artist ID p70015 [↗](#) >

Fig.: Taylor standing in multiple relations; from <https://tools.wmflabs.org/sqid/#/view?id=Q34851>

Wikidata Statements in Terms of Graphs

Elizabeth Taylor (Q34851)

spouse



Richard Burton

 edit

start time

10 October 1975

end time

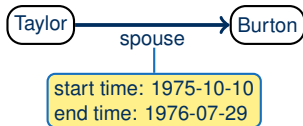
29 July 1976

Wikidata Statements in Terms of Graphs

Elizabeth Taylor (Q34851)

spouse	 Richard Burton	 edit
	start time	10 October 1975
	end time	29 July 1976

“Property Graph”:

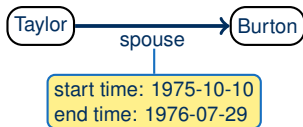


Wikidata Statements in Terms of Graphs

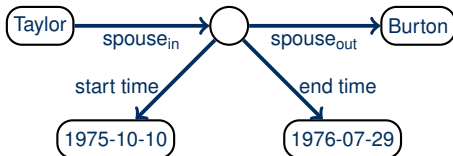
Elizabeth Taylor (Q34851)

spouse	 Richard Burton 
start time	10 October 1975
end time	29 July 1976

“Property Graph”:



“RDF”:



Ontological Modelling in Wikidata

Ontological Modelling in Wikidata

Classification

- 25,298,346 **instance of** statements (for 84.9% of entities)
- 2,056,181 **subclass of** statements (for 4.5% of entities)


Property characteristics/constraints

- **symmetric property** (17 instances)
- **transitive property** (8 instances)
- 12,595 statements specifying other constraints (domain, range, disjointness, ...)

Queries on Wikidata

SPARQL query service: <https://query.wikidata.org>

- officially maintained, live data
- based on RDF mapping [Erxleben et al., ISWC 2014]
- heavily used: 60M–135M queries per month



The screenshot shows the Wikidata Query interface. At the top, there is a header with the Wikidata logo, the text "Wikidata Query", and three buttons: "Examples", "Tools" (with a gear icon), and "Help" (with a question mark icon). Below the header is a text area containing a SPARQL query. To the left of the text area is a vertical toolbar with icons for undo, redo, save, delete, and refresh. At the bottom left of the text area is a blue play button icon.

```
1 #Inventors killed by their own invention
2
3 SELECT ?inventor ?inventorLabel ?gadget ?gadgetLabel WHERE {
4   ?inventor wdt:P157 ?gadget.
5   ?gadget wdt:P61 ?inventor.
6   SERVICE wikibase:label { bd:serviceParam wikibase:language "en". }
7 }
```

Queries on Wikidata

SPARQL query service: <https://query.wikidata.org>

- officially maintained, live data
- based on RDF mapping [Erxleben et al., ISWC 2014]
- heavily used: 60M–135M queries per month

Initial analysis of the non-public logs:

- $\leq 1\%$ queries from human traffic (400–500K per month)
- $\geq 99\%$ service calls from tools and robots
- Irregular distributions and biases – hard to analyse

Queries on Wikidata

SPARQL query service: <https://query.wikidata.org>

- officially maintained, live data
- based on RDF mapping [Erxleben et al., ISWC 2014]
- heavily used: 60M–135M queries per month

Initial analysis of the non-public logs:

- $\leq 1\%$ queries from human traffic (400–500K per month)
- $\geq 99\%$ service calls from tools and robots
- Irregular distributions and biases – hard to analyse

Property paths used for transitivity reasoning

- used in about 50% of human subclass-of queries (20K)
- over 500K queries with subclass-of paths overall

(statistics for May 2017)

OBQA via SPARQL

SPARQL is actually powerful enough for OWL QL reasoning
[Bischoff et al., ISWC 2014]

... but the queries then are getting lengthy ...

```
x (sCO | eqC | ^eqC | INTLISTMEMBER | owl:someValuesFrom |
  (owl:onProperty / (INV | SpEQP)* / (^owl:onProperty | rdfs:domain | rdfs:range))* ?C . {
  {?C SUBCLASSOF owl:Nothing} UNION
  {?C SUBCLASSOF ?D1 {{?C SUBCLASSOF ?D2} UNION UNIVCLASS[?D2]} {
    {?D1 DISJOINTCLASSES ?D2} UNION
    {?V rdf:type owl:AllDisjointClasses . TWO MEMBERS[?V, ?D1, ?D2]}
  }} UNION
  {?C (owl:onProperty / (INV | SpEQP)* ) ?P . {
    {?P SUBPROPERTYOF owl:bottomObjectProperty} UNION
    {?P SUBPROPERTYOF ?Q1 {{?P SUBPROPERTYOF ?Q2} UNION UNIVPROPERTY[?Q2]} {
      {?Q1 (owl:propertyDisjointWith | ^owl:propertyDisjointWith) ?Q2} UNION
      {?V rdf:type owl:AllDisjointProperties . TWO MEMBERS[?V, ?Q1, ?Q2]}
    }}
  }
}}
```

Fig.: A query that checks if x is equivalent to \perp (abbreviated)

Beyond OWL QL

SPARQL cannot support arbitrary OWL reasoning:

- computing power limited by data complexity
- SPARQL can only perform reasoning in NL

Beyond OWL QL

SPARQL cannot support arbitrary OWL reasoning:

- computing power limited by data complexity
- SPARQL can only perform reasoning in NL

Queries with higher data complexities?

- Datalog: PTime-complete data complexity
- Datalog can be used for “query-based” EL reasoning
[K, IJCAI 2011]

Beyond OWL QL

SPARQL cannot support arbitrary OWL reasoning:

- computing power limited by data complexity
- SPARQL can only perform reasoning in NL

Queries with higher data complexities?

- Datalog: PTime-complete data complexity
- Datalog can be used for “query-based” EL reasoning [K, IJCAI 2011]

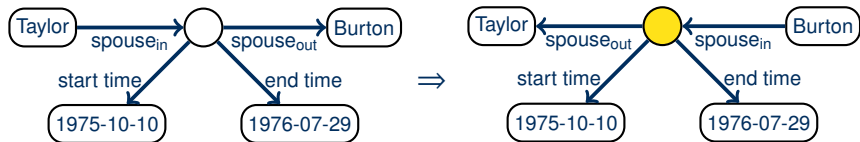
Query-Based Reasoning:

- ontological information as part of data
- logic for meta-reasoning on top
- same data can be viewed under different semantics

Ontologies for Wikidata?

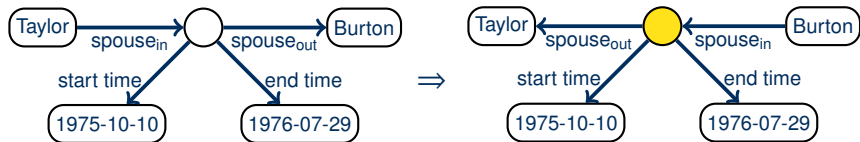
A Simple Example

Wikidata declares the `spouse` property to be symmetric:



A Simple Example

Wikidata declares the `spouse` property to be symmetric:

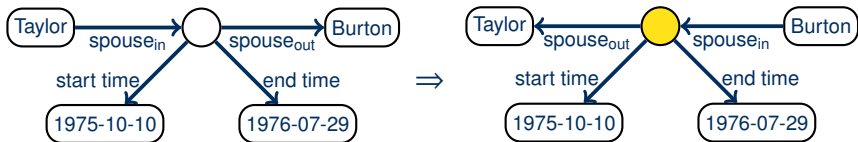


ABox:

$\text{spouse}_{\text{in}}(\text{taylor}, s)$ $\text{spouse}_{\text{out}}(s, \text{burton})$
 $\text{start}(s, 1975-10-10)$ $\text{end}(s, 1976-07-29)$

A Simple Example

Wikidata declares the `spouse` property to be symmetric:



ABox:

$\text{spouse}_{in}(\text{taylor}, s)$ $\text{spouse}_{out}(s, \text{burton})$
 $\text{start}(s, 1975-10-10)$ $\text{end}(s, 1976-07-29)$

An axiom of symmetry:

$\forall x, y, z_1, z_2, v. \text{spouse}_{in}(x, v) \wedge \text{spouse}_{out}(v, y) \wedge \text{start}(v, z_1) \wedge \text{end}(v, z_2)$
 $\rightarrow \exists w. \text{spouse}_{in}(y, w) \wedge \text{spouse}_{out}(x, w) \wedge \text{start}(w, z_1) \wedge \text{end}(w, z_2)$

Existential rules

$$\text{spouse}_{\text{in}}(x, v) \wedge \text{spouse}_{\text{out}}(v, y) \wedge \text{start}(v, z_1) \wedge \text{end}(v, z_2)$$
$$\rightarrow \exists w. \text{spouse}_{\text{in}}(y, w) \wedge \text{spouse}_{\text{out}}(x, y) \wedge \text{start}(w, z_1) \wedge \text{end}(w, z_2)$$

This axiom is an **existential rule**

Existential rules

$$\text{spouse}_{\text{in}}(x, v) \wedge \text{spouse}_{\text{out}}(v, y) \wedge \text{start}(v, z_1) \wedge \text{end}(v, z_2)$$
$$\rightarrow \exists w. \text{spouse}_{\text{in}}(y, w) \wedge \text{spouse}_{\text{out}}(x, y) \wedge \text{start}(w, z_1) \wedge \text{end}(w, z_2)$$

This axiom is an **existential rule**

- it is **not expressible in Datalog**

Existential rules

$$\text{spouse}_{\text{in}}(x, v) \wedge \text{spouse}_{\text{out}}(v, y) \wedge \text{start}(v, z_1) \wedge \text{end}(v, z_2)$$
$$\rightarrow \exists w. \text{spouse}_{\text{in}}(y, w) \wedge \text{spouse}_{\text{out}}(x, y) \wedge \text{start}(w, z_1) \wedge \text{end}(w, z_2)$$

This axiom is an **existential rule**

- it is **not expressible in Datalog**
- it is **not expressible in DL**

Existential rules

$$\text{spouse}_{\text{in}}(x, v) \wedge \text{spouse}_{\text{out}}(v, y) \wedge \text{start}(v, z_1) \wedge \text{end}(v, z_2) \\ \rightarrow \exists w. \text{spouse}_{\text{in}}(y, w) \wedge \text{spouse}_{\text{out}}(x, y) \wedge \text{start}(w, z_1) \wedge \text{end}(w, z_2)$$

This axiom is an **existential rule**

- it is **not expressible in Datalog**
- it is **not expressible in DL**
- it is **not linear**

Existential rules

$$\text{spouse}_{\text{in}}(x, v) \wedge \text{spouse}_{\text{out}}(v, y) \wedge \text{start}(v, z_1) \wedge \text{end}(v, z_2) \\ \rightarrow \exists w. \text{spouse}_{\text{in}}(y, w) \wedge \text{spouse}_{\text{out}}(x, y) \wedge \text{start}(w, z_1) \wedge \text{end}(w, z_2)$$

This axiom is an **existential rule**

- it is **not expressible in Datalog**
- it is **not expressible in DL**
- it is **not linear**
- it is **not (frontier) guarded**

Existential rules

$$\begin{aligned} & \text{spouse}_{\text{in}}(x, v) \wedge \text{spouse}_{\text{out}}(v, y) \wedge \text{start}(v, z_1) \wedge \text{end}(v, z_2) \\ & \rightarrow \exists w. \text{spouse}_{\text{in}}(y, w) \wedge \text{spouse}_{\text{out}}(x, y) \wedge \text{start}(w, z_1) \wedge \text{end}(w, z_2) \end{aligned}$$

This axiom is an **existential rule**

- it is **not expressible in Datalog**
- it is **not expressible in DL**
- it is **not linear**
- it is **not (frontier) guarded**
- it is **not acyclic** (w.r.t. predicate dependencies)

Existential rules

$$\text{spouse}_{\text{in}}(x, v) \wedge \text{spouse}_{\text{out}}(v, y) \wedge \text{start}(v, z_1) \wedge \text{end}(v, z_2) \\ \rightarrow \exists w. \text{spouse}_{\text{in}}(y, w) \wedge \text{spouse}_{\text{out}}(x, y) \wedge \text{start}(w, z_1) \wedge \text{end}(w, z_2)$$

This axiom is an **existential rule**

- it is **not expressible in Datalog**
- it is **not expressible in DL**
- it is **not linear**
- it is **not (frontier) guarded**
- it is **not acyclic** (w.r.t. predicate dependencies)

but it might be **weakly acyclic/frontier guarded**
(depends on other axioms)

Breaking the Rules

Observation: Normalisation may destroy syntactic properties
[K & Thost; ISWC 2016]

- **Acyclicity** properties are mostly often preserved
- **Linearity** and **guardedness** are lost (syntactically)
- Can sometimes recover by **denormalising** rules

Breaking the Rules

Observation: Normalisation may destroy syntactic properties
[K & Thost; ISWC 2016]

- **Acyclicity** properties are mostly often preserved
- **Linearity** and **guardedness** are lost (syntactically)
- Can sometimes recover by **denormalising** rules

Existential rules are a first step, but:

- Normalised rules are **hard to read and write**
- **Not expressive** enough, e.g., cannot copy arbitrary annotation sets
- **Loss of structure** by flattening annotations, e.g., cannot have closed-world negation on annotation sets

Annotated Logics

MARS

Idea: Change from relational structures to “relational structures with annotated tuples” [Marx, K, Thost, IJCAI 2017]

MARS

Idea: Change from relational structures to “relational structures with annotated tuples” [Marx, K, Thost, IJCAI 2017]

Multi-Attributed Relational Structures (MARS):

- standard interpretation domain Δ^I
- finite annotation sets $S \in \mathcal{P}_{\text{fin}}(\Delta^I \times \Delta^I)$
- n -ary relations r interpreted as
 $r^I \subseteq (\Delta^I)^n \times \mathcal{P}_{\text{fin}}(\Delta^I \times \Delta^I)$

MARS

Idea: Change from relational structures to “relational structures with annotated tuples” [Marx, K, Thost, IJCAI 2017]

Multi-Attributed Relational Structures (MARS):

- standard interpretation domain Δ^I
- finite annotation sets $S \in \mathcal{P}_{\text{fin}}(\Delta^I \times \Delta^I)$
- n -ary relations r interpreted as
 $r^I \subseteq (\Delta^I)^n \times \mathcal{P}_{\text{fin}}(\Delta^I \times \Delta^I)$

Multi-Attributed Predicate Logic (MAPL)

- Ground fact:
spouse(taylor, burton)@{start : 1975, end : 1976}
- Object and set variables:
 $\forall x, y, Z. \text{spouse}(x, y)@Z \rightarrow \text{spouse}(y, x)@Z$

Expressivity of MAPL

Theorem: MAPL is equivalent to Weak Second-Order Logic, hence reasoning is not semi-decidable.

Expressivity of MAPL

Theorem: MAPL is equivalent to Weak Second-Order Logic, hence reasoning is not semi-decidable.

Research goal: Identify practical fragments

Expressivity of MAPL

Theorem: MAPL is equivalent to Weak Second-Order Logic, hence reasoning is not semi-decidable.

Research goal: Identify practical fragments

A decidable fragment:

MAPL Rules (MARPL)

- Horn rules, with all variables universally quantified
- all set variables bound in body atoms

Example: $\forall x, y, Z. \text{spouse}(x, y)@Z \rightarrow \text{spouse}(y, x)@Z$

MARPL: Additional Features

We really need more expressive features

MARPL: Additional Features

We really need more expressive features

Conditions on annotation sets Z

- $[\text{start} : 1975, \text{end} : *](Z)$:
“ Z has given start and some end, but nothing more”
- $[\text{start} : 1975](Z)$: “ Z has given start, and possibly more”

↪ supported in MARPL rule bodies

MARPL: Additional Features

We really need more expressive features

Conditions on annotation sets Z

- $[\text{start} : 1975, \text{end} : *](Z)$:
“ Z has given start and some end, but nothing more”
- $[\text{start} : 1975](Z)$: “ Z has given start, and possibly more”

~> supported in MARPL rule bodies

Inferring new annotation sets

- Support declarative definition of deterministic functions that derive new sets
- Example:

$$\text{employer}(x, \text{cern})@Z \wedge [\text{pos} : \text{fellow}](Z)$$
$$\rightarrow \text{cernFellow}(x)@[\text{start} : Z.\text{start}, \text{end} : Z.\text{end}]$$

~> supported in MARPL rule heads

MARPL Complexity

Theorem: Conjunctive query answering over MARPL ontologies is ExpTime-complete, both for combined complexity and for data complexity.

MARPL Complexity

Theorem: Conjunctive query answering over MARPL ontologies is ExpTime-complete, both for combined complexity and for data complexity.

Problem?

MARPL Complexity

Theorem: Conjunctive query answering over MARPL ontologies is ExpTime-complete, both for combined complexity and for data complexity.

Problem?

- **Not really:** hardness requires annotation sets of unbounded size (not a practical concern)
- **Actually, it's a feature:** high data complexity enables powerful meta-reasoning in query-based approaches

Attributed Description Logics

MARPL is a simple rule language
(“Datalog for annotated logic”)

How about DLs?

Attributed Description Logics

MARPL is a simple rule language
(“Datalog for annotated logic”)

How about DLs?

Attributed Description Logics

see DL talk later today [K et al., ISWC 2017]

Attributed Description Logics

MARPL is a simple rule language
(“Datalog for annotated logic”)

How about DLs?

Attributed Description Logics

see DL talk later today [K et al., ISWC 2017]

How about attributed existential rules?

~> future work

The Future of KR

Problem solved?

So all we need to marry KG and KR are attributed logics?

Problem solved?

So all we need to marry KG and KR are attributed logics?

Surely not – many other areas need more work!

Problem solved?

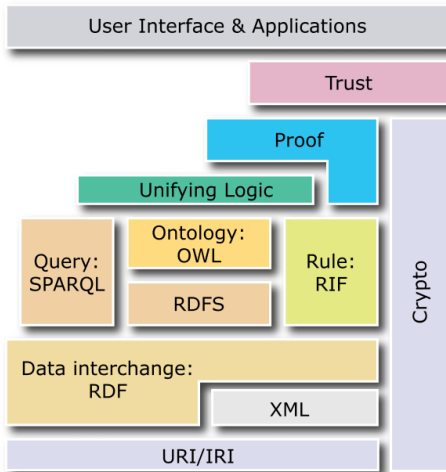
So all we need to marry KG and KR are attributed logics?

Surely not – many other areas need more work!

We also need to change some of our premises:

Traditional KR View	vs.	Knowledge Graphs View
schema first		data first
unique purpose		multi-purpose
fixed application		emerging applications
closed expert team		open community/many teams

Still Looking for the “Unifying Logic”?



Conclusions

Summary

- Knowledge Graphs are enriched graphs
- Wikidata: large ABox / “ontology” / path queries
- Query-based reasoning: plug’n’play semantics for data
- Existential rules & DLs: struggling with annotations
- Attributed logics: MAPL & MARPL (& attributed DLs)

What next?

View KR as a declarative computing paradigm & start facing the competition in this space!

Revisit “Computing in Logic” (but don’t go back to Prolog!)

References

- [Bischoff et al., ISWC 2014] Stefan Bischoff, Markus Krötzsch, Axel Polleres, Sebastian Rudolph: **Schema-Agnostic Query Rewriting for SPARQL 1.1**. ISWC 2014.
- [Erleben et al., ISWC 2014] Fredo Erleben, Michael Günther, Markus Krötzsch, Julian Mendez, Denny Vrandečić: **Introducing Wikidata to the Linked Data Web**. ISWC 2014.
- [K, IJCAI 2011] Markus Krötzsch: **Efficient Rule-Based Inferencing for OWL EL**. IJCAI 2011.
- [K et al., ISWC 2017] Markus Krötzsch, Maximilian Marx, Ana Ozaki, Veronika Thost: **Attributed Description Logics: Ontologies for Knowledge Graphs**. ISWC 2017 (to appear).
- [K & Thost; ISWC 2016] Markus Krötzsch, Veronika Thost: **Ontologies for Knowledge Graphs: Breaking the Rules**. ISWC 2016.
- [Marx, K, Thost, IJCAI 2017] Maximilian Marx, Markus Krötzsch, Veronika Thost: **Logic on MARS: Ontologies for Generalised Property Graphs**. IJCAI 2017 (to appear).
- [Vrandečić & K; Comm. ACM 2014] Denny Vrandečić, Markus Krötzsch: **Wikidata: A Free Collaborative Knowledgebase**. Comm. ACM, 57:10, 2014.