TECHNISCHE UNIVERSITÄT DRESDEN

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

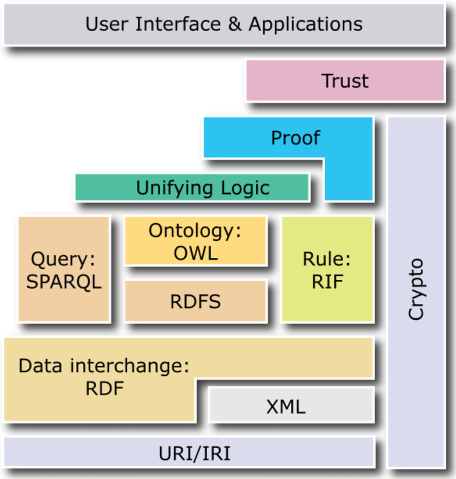
[†] Knowledge-Based Systems

TU Dresden

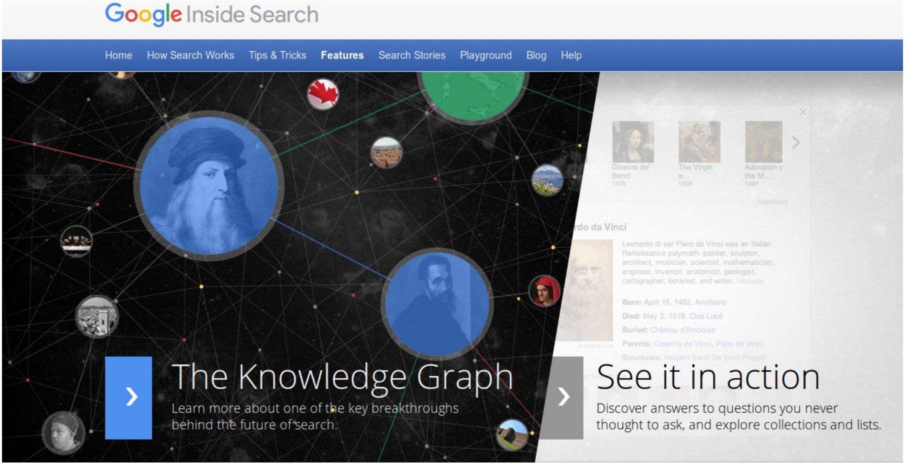
Full paper: `https://iccl.inf.tu-dresden.de/web/DL2017-keynote`

DL Workshop 2017

The Semantic Web (2007)



2012: The Knowledge Graph



"... one of the key breakthroughs behind the future of search"

More Knowledge Graphs

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 th[illegible]y technology
in mode[illegible]agement

especia[illegible]y in AI applications

Not really happening

WIKIDATA

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 . . .

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 ▸ 1 reference	edit

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

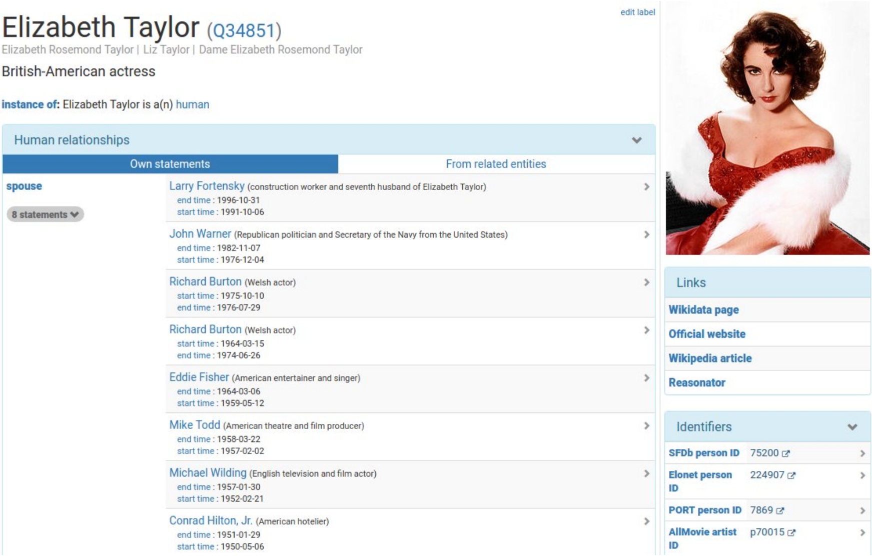


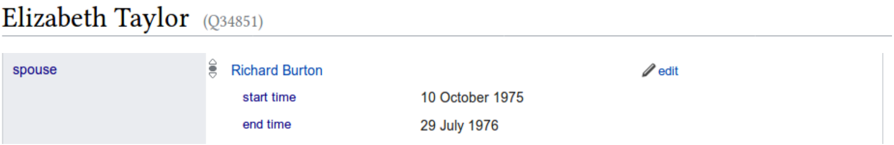
Fig.: Taylor standing in multiple relations; from `https://tools.wmflabs.org/sqid/#/view?id=Q34851`

Wikidata Statements in Terms of Graphs

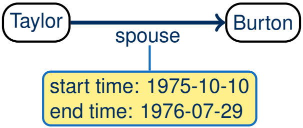
Elizabeth Taylor (Q34851)

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

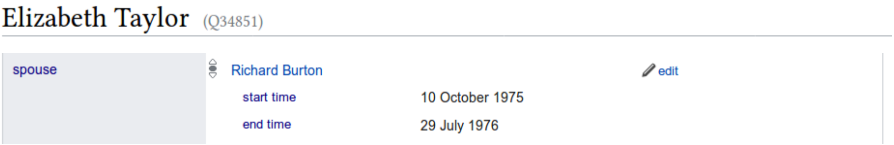
Wikidata Statements in Terms of Graphs



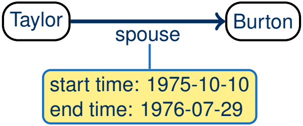
“Property Graph”:



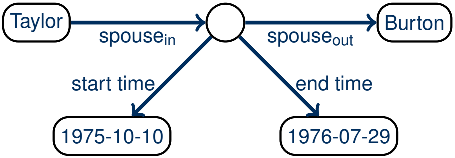
Wikidata Statements in Terms of Graphs



"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)

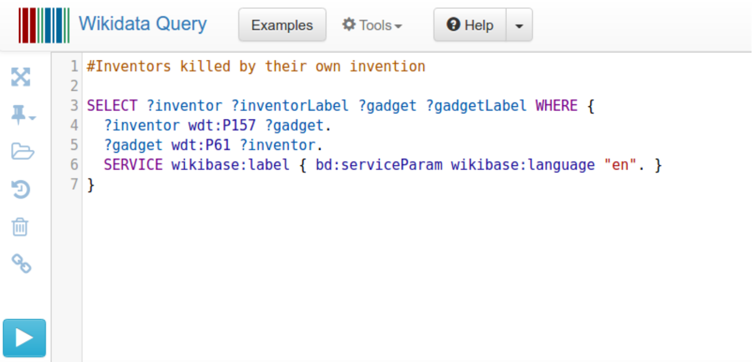
Property characteristics/constraints

- 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



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:

- ≤1% queries from human traffic (400–500K per month)
- ≥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:

- ≤1% queries from human traffic (400–500K per month)
- ≥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 | SpoEqp)* / (^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 . twoMembers[?V, ?D1, ?D2]}
  }} UNION
  {?C (owl:onProperty / (Inv | SpoEqp)*) ?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 . twoMembers[?V, ?Q1, ?Q2]}
    }
  }}
}
```

Fig.: A query that checks if x is equivalent to $\bot$ (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]

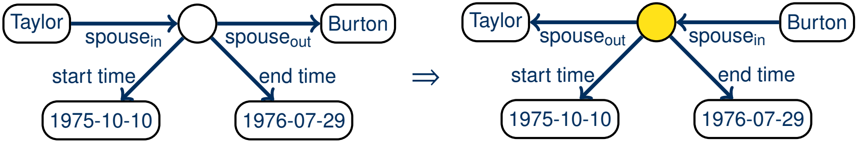
Query-Based Reasoning:

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

Ontologies for Wikidata?

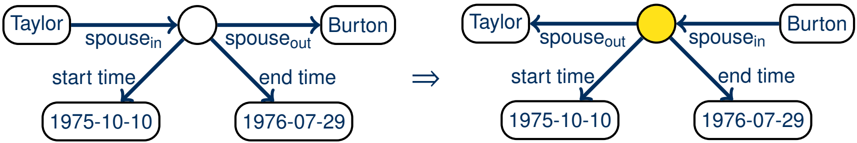
A Simple Example

Wikidata declares the spouse property to be symmetric:



A Simple Example

Wikidata declares the spouse property to be symmetric:



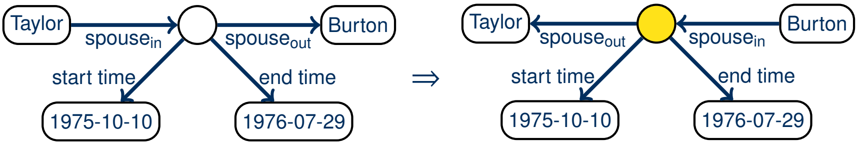
ABox:

$$\mathsf{spouse}_{\mathsf{in}}(\mathsf{taylor}, s) \quad \mathsf{spouse}_{\mathsf{out}}(s, \mathsf{burton})$$

$$\mathsf{start}(s, \mathtt{1975\text{-}10\text{-}10}) \quad \mathsf{end}(s, \mathtt{1976\text{-}07\text{-}29})$$

A Simple Example

Wikidata declares the spouse property to be symmetric:



ABox:

$$\text{spouse}_{\text{in}}(\text{taylor}, s) \quad \text{spouse}_{\text{out}}(s, \text{burton})$$

$$\text{start}(s, \mathtt{1975\text{-}10\text{-}10}) \quad \text{end}(s, \mathtt{1976\text{-}07\text{-}29})$$

An axiom of symmetry:

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

Existential rules

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

This axiom is an existential rule

Existential rules

$$\mathsf{spouse_{in}}(x, v) \land \mathsf{spouse_{out}}(v, y) \land \mathsf{start}(v, z_1) \land \mathsf{end}(v, z_2)$$
$$\rightarrow \exists w.\mathsf{spouse_{in}}(y, w) \land \mathsf{spouse_{out}}(x, y) \land \mathsf{start}(w, z_1) \land \mathsf{end}(w, z_2)$$

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

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

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

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

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)

Existential rules

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

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

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

Multi-Attributed Predicate Logic (MAPL)

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

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

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

$\leadsto$ supported in MARPL rule bodies

MARPL: Additional Features

We really need more expressive features

Conditions on annotation sets Z

- $[\mathsf{start} : 1975, \mathsf{end} : *](Z)$:
  "Z has given start and some end, but nothing more"
- $\lfloor \mathsf{start} : 1975 \rfloor(Z)$: "$Z$ has given start, and possibly more"

$\leadsto$ supported in MARPL rule bodies

Inferring new annotation sets

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

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

$\leadsto$ supported in MARPL rule heads

MARPL Complexity

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

MARPL Complexity

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

Problem?

MARPL Complexity

Theorem: Conjunctive query answering ove 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?

$\rightsquigarrow$ 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?

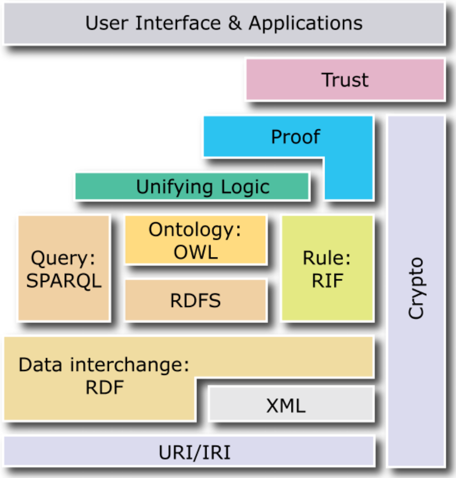
So all we need to marry KG and KR are attributed logics?

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.
- [Erxleben et al., ISWC 2014] Fredo Erxleben, 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.