Attributed Description Logics: Reasoning on Knowledge Graphs

Markus Krötzsch[1] **Maximilian Marx**[1] Ana Ozaki[2] Veronika Thost[3]

[1] Knowledge-Based Systems Group, TU Dresden

[2] KRDB Research Centre for Knowledge and Data, Free University of Bolzano

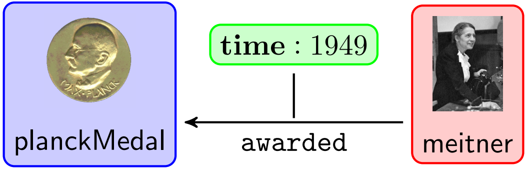
[3] MIT-IBM Watson AI Lab, IBM Research

27th International Joint Conference on Artificial Intelligence

Full paper: `https://iccl.inf.tu-dresden.de/web/Inproceedings3045/en`

Knowledge Graphs

Wikidata: the free and open Knowledge Graph of the Wikimedia Foundation



time : 1949

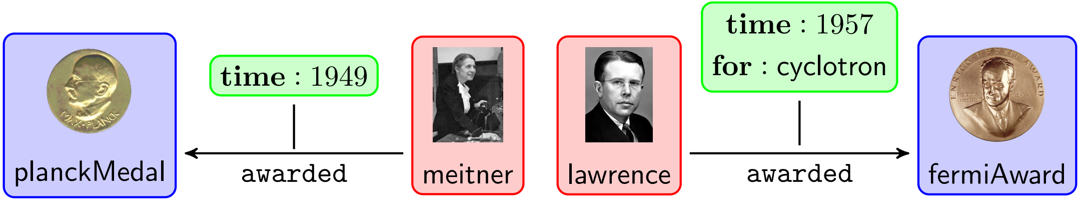
planckMedal

awarded

meitner

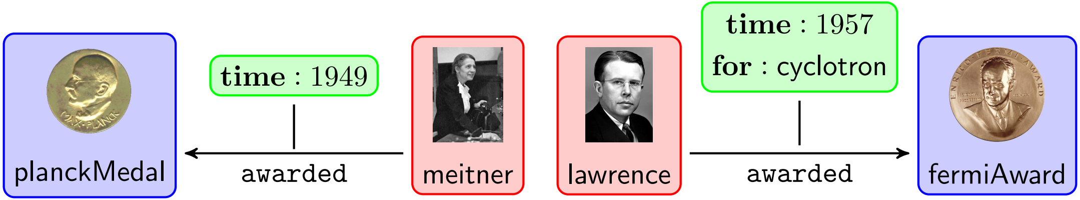
Knowledge Graphs

Wikidata: the free and open Knowledge Graph of the Wikimedia Foundation



Knowledge Graphs

Wikidata: the free and open Knowledge Graph of the Wikimedia Foundation



$$\mathtt{awarded}(\mathsf{meitner}, \mathsf{planckMedal})@[\mathbf{time} : 1949]$$

$$\mathtt{awarded}(\mathsf{lawrence}, \mathsf{fermiAward})@[\mathbf{time} : 1957, \mathbf{for} : \mathsf{cyclotron}]$$

Description Logics

Description Logics:

- decidable fragments of First-Order Logic
- variable-free syntax
- theoretical foundation of the Web Ontology Language (OWL)

Description Logics

Description Logics:

- decidable fragments of First-Order Logic
- variable-free syntax
- theoretical foundation of the Web Ontology Language (OWL)

Example

“Everybody who has received an award should be known for something”

Description Logics

Description Logics:

- decidable fragments of First-Order Logic
- variable-free syntax
- theoretical foundation of the Web Ontology Language (OWL)

Example

“Everybody who has received an award should be known for something”

$$\exists \texttt{awarded}.\, \text{Award} \sqsubseteq \exists \texttt{knownFor}.\, \top$$

Description Logics

Description Logics:

- decidable fragments of First-Order Logic
- variable-free syntax
- theoretical foundation of the Web Ontology Language (OWL)

Example

“Everybody who has received an award should be known for something”

$$\exists \texttt{awarded}.\,\mathsf{Award} \sqsubseteq \exists \texttt{knownFor}.\,\top \qquad\qquad \mathsf{Award}(\mathsf{planckMedal})$$

Description Logics

Description Logics:

- decidable fragments of First-Order Logic
- variable-free syntax
- theoretical foundation of the Web Ontology Language (OWL)

Example

“Everybody who has received an award should be known for something”

$$\exists\texttt{awarded}.\,\mathsf{Award} \sqsubseteq \exists\texttt{knownFor}.\,\top \qquad \mathsf{Award}(\mathsf{planckMedal})$$

But what about the annotations?

Attributed Description Logics

Attributed Description Logics:

- Add annotations to concept names and role names
- Allow variables as placeholders for annotations

Attributed Description Logics

Attributed Description Logics:

- Add annotations to concept names and role names
- Allow variables as placeholders for annotations

Example

"Everybody who has received an award should be known for something, keeping the same annotation"

Attributed Description Logics

Attributed Description Logics:

- Add annotations to concept names and role names
- Allow variables as placeholders for annotations

Example

“Everybody who has received an award should be known for something, keeping the same annotation”

$$\exists\,\texttt{awarded@}S.\,\mathsf{Award} \sqsubseteq \exists\,\texttt{knownFor@}S.\,\top$$

Attributed Description Logics

Attributed Description Logics:

- Add annotations to concept names and role names
- Allow variables as placeholders for annotations

Example

“Everybody who has received an award should be known for something, with a new annotation that has the same $\mathbf{for}$ value”

Attributed Description Logics

Attributed Description Logics:

- Add annotations to concept names and role names
- Allow variables as placeholders for annotations

Example

“Everybody who has received an award should be known for something, with a new annotation that has the same **for** value”

$$\exists\,\texttt{awarded}@S.\,\mathsf{Award} \sqsubseteq \exists\,\texttt{knownFor}@[\mathbf{for} : S.\mathbf{for}]$$

Attributed Description Logics

Attributed Description Logics:

- Add annotations to concept names and role names
- Allow variables as placeholders for annotations

Example

“Everybody who has received an award in 1949 should be known for something, with a new annotation that has the same values for **for**”

Attributed Description Logics

Attributed Description Logics:

- Add annotations to concept names and role names
- Allow variables as placeholders for annotations

Example

“Everybody who has received an award in 1949 should be known for something, with a new annotation that has the same values for **for**”

$$S : \lfloor \mathbf{time} : 1949 \rfloor \left(\exists\, \mathtt{awarded@}S.\,\mathsf{Award} \sqsubseteq \exists\, \mathtt{knownFor@}[\mathbf{for} : S.\mathbf{for}] \right)$$

Attributed Description Logics

Attributed Description Logics:

- Add annotations to concept names and role names
- Allow variables as placeholders for annotations

Example

“Everybody who has received an award in 1949 should be known for something, with a new annotation that has the same values for **for**”

$$S\mathbin{:}[\mathbf{time} : 1949]\Big(\exists\,\mathtt{awarded@}S.\,\mathsf{Award} \sqsubseteq \exists\,\mathtt{knownFor@}[\mathbf{for} : S.\mathbf{for}]\Big)$$

Reasoning:

- Ground all axioms
- Treat all annotated concepts and annotated roles as new names

Attributed Description Logics

Attributed Description Logics:

- Add annotations to concept names and role names
- Allow variables as placeholders for annotations

Example

"Everybody who has received an award in 1949 should be known for something, with a new annotation that has the same values for **for**"

$$S : \lfloor \mathbf{time} : 1949 \rfloor \left(\exists\, \mathtt{awarded@}S.\, \mathsf{Award} \sqsubseteq \exists\, \mathtt{knownFor@}[\mathbf{for} : S.\mathbf{for}] \right)$$

Reasoning:

- Ground all axioms
- Treat all annotated concepts and annotated roles as new names
- Add auxiliary axioms that handle inclusions of annotations

Complexities

Theorem

Combined complexity of satisfiability for attributed Knowledge Bases

non-attributed		*attributed*		
	DL	*ground*	*restricted*	*unrestricted*
P-*complete*	$\mathcal{ELH}_@$	P-*complete*	P-*complete*[1]	Exp-*complete*
Exp-*complete*	$\mathcal{ALCH}_@$	Exp-*complete*	Exp-*complete*	2Exp-*complete*
N2Exp-*complete*	$\mathcal{SROIQ}_@$	N2Exp-*complete*	N2Exp-*complete*	N2Exp-*complete*

Complexities

Theorem

Combined complexity of satisfiability for attributed Knowledge Bases

non-attributed		*attributed*		
	DL	*ground*	*restricted*	*unrestricted*
P-*complete*	$\mathcal{ELH}_@$	P-*complete*	P-*complete*[1]	Exp-*complete*
Exp-*complete*	$\mathcal{ALCH}_@$	Exp-*complete*	Exp-*complete*	2Exp-*complete*
N2Exp-*complete*	$\mathcal{SROIQ}_@$	N2Exp-*complete*	N2Exp-*complete*	N2Exp-*complete*

- Grounding is exponential in the size of the Knowledge Base

Complexities

Theorem

Combined complexity of satisfiability for attributed Knowledge Bases

non-attributed		*attributed*		
	DL	*ground*	*restricted*	*unrestricted*
P-*complete*	$\mathcal{ELH}_@$	P-*complete*	P-*complete*[1]	Exp-*complete*
Exp-*complete*	$\mathcal{ALCH}_@$	Exp-*complete*	Exp-*complete*	2Exp-*complete*
N2Exp-*complete*	$\mathcal{SROIQ}_@$	N2Exp-*complete*	N2Exp-*complete*	N2Exp-*complete*

- Grounding is exponential in the size of the Knowledge Base
- Three syntactic restrictions ensure polynomial groundings
- [1]: Except when violating one (specific) condition, where we show PSpace-hardness

Conclusion & Outlook

Conclusion:

- First-class support for annotations in Knowledge Bases
- Reasoning via translation into standard DLs
- $\leadsto$ Existing tools can be used for attributed reasoning

Outlook:

- Further Extensions push DLs even closer to rule languages
- $\leadsto$ Focus on adapting rules for Knowledge Graphs?
- MARPL [Krötzsch, M., Thost IJCAI'17] is "Attributed Datalog"
- $\leadsto$ "Attributed Existential Rules"?