Maximilian Marx

Attributed Logics for Reasoning over Knowledge Graphs

Dresden, 2025-07-08

Knowledge Representation & Reasoning

Goals

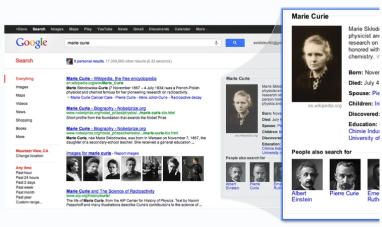
- Encode data about the real world
- Infer new data

Long History:

- 1970s: Frames
- 1980s–1990s: KL-ONE, Cyc, Description Logics
- 2000s: Semantic Web

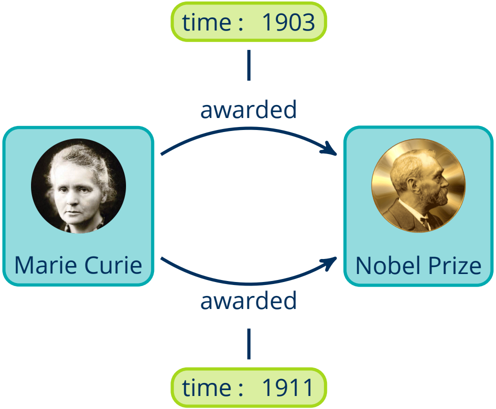
- RDF as a graph format
- SPARQL as a graph query language
- OWL as an ontology language

Knowledge Graphs



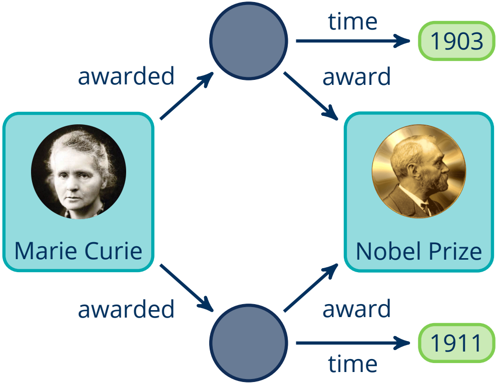
None of these use RDF for their internal representation. Why?

RDF is not enough



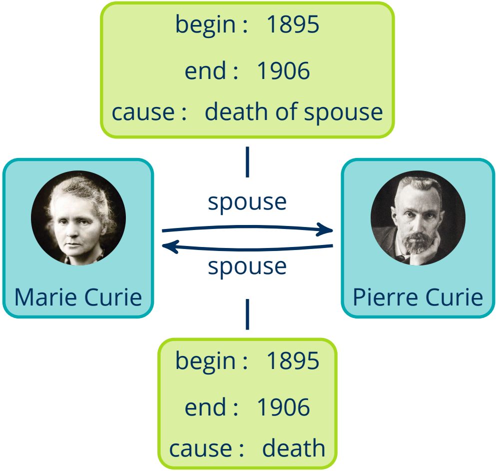
- How to express annotations in RDF?

RDF is not enough



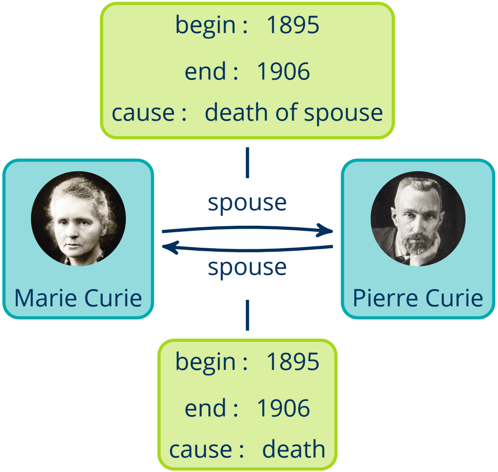
- How to express annotations in RDF?
- Add auxiliary nodes: **Reification**

OWL is not enough



- Spouse is a symmetric relationship

OWL is not enough



- Spouse is a symmetric relationship
- OWL can declare properties as symmetric
- But annotations are not identical

Goal: Reasoning with Annotations

Fact entailment

- Given: annotated Knowledge Graph and background knowledge
- Does fact α follow?

Goal: Reasoning with Annotations

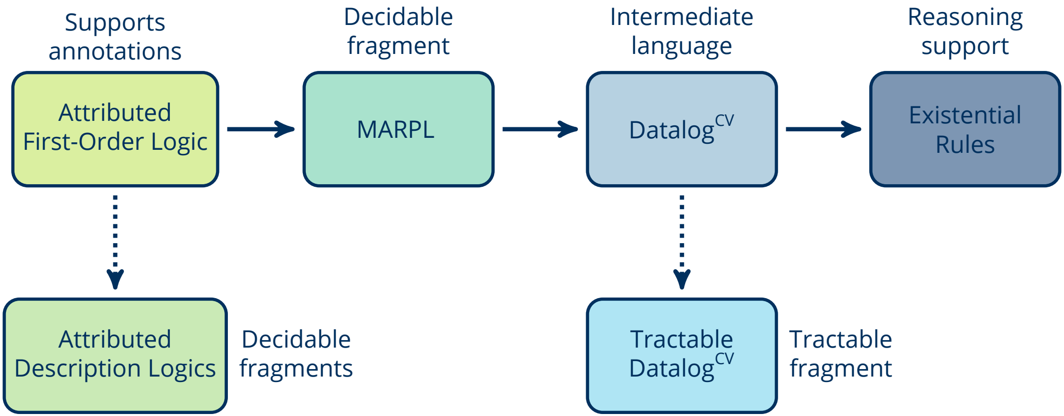
Fact entailment

- Given: annotated Knowledge Graph and background knowledge
- Does fact α follow?

Requirements

- Modelling features for selecting & constructing annotations
- Decidable fact entailment
- Software implementation

Roadmap


Supports annotations
Attributed First-Order Logic
Decidable fragment
MARPL
Intermediate language
Datalog^{CV}
Reasoning support
Existential Rules
Attributed Description Logics
Decidable fragments
Tractable Datalog^{CV}
Tractable fragment

Attributed First-Order Logic

Marx et al., International Joint Conferences on Artificial Intelligence 2017

Extension of First-Order Logic (FOL):

- Extend predicates with annotations: $p(x, y)$ becomes $p(x, y)@S$

$$\text{spouse}(\mathit{Marie}, \mathit{Pierre})@\left\{\begin{array}{c}\text{begin} : 1895,\\ \text{end} : 1906,\\ \text{cause} : \text{death of spouse}\end{array}\right\}$$

- Annotations are finite sets of pairs: $\{a : v\}$
- Variables and quantification over annotations

Example: Married Nobel Laureates

- Find spouses that share a Nobel Prize

$$\forall U, V.\ \forall x, y.\ \big(\mathrm{spouse}(x, y)@U \wedge \mathrm{awarded}(x, \mathit{Nobel})@V \wedge (\mathrm{with} : y) \in V$$
$$\rightarrow \mathrm{nobelSpouse}(x, y)@\varnothing\big)$$

Example: Married Nobel Laureates

- Find spouses that share a Nobel Prize

$$\forall U, V.\ \forall x, y.\ \big(\mathrm{spouse}(x, y)@U \land \mathrm{awarded}(x, \mathit{Nobel})@V \land (\mathrm{with} : y) \in V \\ \rightarrow \mathrm{nobelSpouse}(x, y)@\varnothing\big)$$

- Attributed FOL allows reasoning with annotations!
- The symmetric spouse example is also expressible

Undecidability

Theorem

Attributed First-Order Logic is as expressive as weak Second-Order Logic.

Undecidability

Theorem

Attributed First-Order Logic is as expressive as weak Second-Order Logic.

- **Good**: We have added expressive power
- **Bad**: Attributed FOL is not even semi-decidable

Undecidability

Theorem

Attributed First-Order Logic is as expressive as weak Second-Order Logic.

- **Good**: We have added expressive power
- **Bad**: Attributed FOL is not even semi-decidable

We need a decidable fragment!

MARPL

Marx et al., International Joint Conferences on Artificial Intelligence 2017

Rule fragment of FOL: **Datalog**

$$\text{spouse}(x, y) \rightarrow \text{spouse}(y, x)$$

Rule fragment of attributed FOL: **MARPL**

$$\text{spouse}(x, y)@U \rightarrow \text{spouse}(y, x)@U$$

- **Specifiers** for matching annotations
- **Function definitions** for constructing annotations

Example: Married Nobel Laureates

- Find spouses that share a Nobel Prize

$$\text{spouse}(x, y)@\lfloor\rfloor \wedge \text{awarded}(x, \mathit{Nobel})@\lfloor \text{with} : y \rfloor \rightarrow \text{nobelSpouse}(x, y)@[]$$

Example: Married Nobel Laureates

- Find spouses that share a Nobel Prize

$$\mathrm{spouse}(x,y)@\lfloor\rfloor \land \mathrm{awarded}(x,\mathit{Nobel})@\lfloor \mathrm{with} : y\rfloor \rightarrow \mathrm{nobelSpouse}(x,y)@[]$$

- MARPL allows reasoning with annotations!

Example: Inverses for Spouse

- Spouse in marriage ended due to death on day d has inverse with new annotation $\text{Inverse}(U, d)$

$$\text{spouse}(x, y)@U \land \lfloor \text{cause} : \text{death} \rfloor(U) \land \text{died}(x, d)@\lfloor\rfloor \\ \rightarrow \text{spouse}(y, x)@\text{Inverse}(U, d)$$

Function definition for $\text{Inverse}(U, d)$

- Cause is death of spouse $\implies \text{insert}(\text{cause} : \text{death of spouse})$
- end is d $\implies \text{insert}(\text{end} : d)$
- Inherit begin, if present $\lfloor \text{begin} : b \rfloor(U) \implies \text{insert}(\text{begin} : b)$

Complexities

Fact entailment

- Given: annotated Knowledge Graph and background knowledge
- Does fact α follow?

How hard is fact entailment?

Complexities

Fact entailment

- Given: annotated Knowledge Graph and background knowledge
- Does fact α follow?

How hard is fact entailment?

- **Data complexity** considers background knowledge fixed
- **Combined complexity** considers background knowledge part of the input
- Typical graphs much larger than background knowledge

Complexities of MARPL

Theorem

Combined complexity of MARPL fact entailment is ExpTime*-complete.*

Complexities of MARPL

Theorem

Combined complexity of MARPL fact entailment is ExpTime*-complete.*

Theorem

Data complexity of MARPL fact entailment is

- ExpTime*-complete in general*
- PTime*-complete if annotation size is bounded*

Complexities of MARPL

Theorem

Combined complexity of MARPL fact entailment is ExpTime*-complete.*

Theorem

Data complexity of MARPL fact entailment is

- ExpTime*-complete in general*
- PTime*-complete if annotation size is bounded*

Wikidata: 520 out of 1.6 Billion statements have annotation size 10 or larger

SQID: A Wikidata Browser

Marie Curie (Q7186)

Maria Salomea Skłodowska | Maria Skłodowska-Curie | Marie Curie-Sklodowska | Maria Skłodowska

Polish and French physicist and chemist (1867–1934)

Human relationships

Own statements	From related entities
sibling	Bronisława Dłuska (Polish physician, first director of the Radium-Institut in Warsaw (1865-1939))
	Józef Skłodowski (Polish physician)
	Helena Skłodowska-Szaley (Polish educator, school inspector and educational activist)
spouse	Pierre Curie (French physicist (1859-1906)) start time : 1895-07-26 end time : 1906-04-19 place of marriage : Sceaux (commune in Hauts-de-Seine, France) end cause : death of subject's spouse (end cause of marriage, significant event)
relative 6+3 statements	Helena Dłuska (Polish sportsperson (1892-1921)) kinship to subject : sororal niece (female child of a sister or half-sister)
	Kazimierz Dłuski (Polish physician and politician) kinship to subject : sister's husband (husband of sister)
	Jacques Curie (French physicist (1855–1941)) kinship to subject : husband's brother (brother of husband)
father	Władysław Skłodowski (Polish scientist and educator)
child	Irène Joliot-Curie (French scientist (1897-1956))
	Ève Curie (writer, journalist and pianist, younger daughter of Marie and Pierre Curie)
mother	Bronisława Skłodowska (mother of Marie Curie)

Links

- **Wikidata page**
- **Wikipedia article**
- **Reasonator**

Identifiers

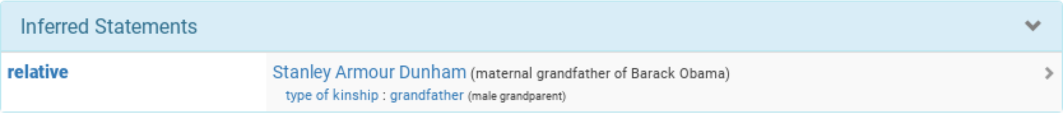
Hrvatska enciklopedija ID	12996
Brockhaus Enzyklopädie	curie-...-marya

MARPL Reasoning in SQID

Marx & Krötzsch, Demo at International Semantic Web Conference 2017

- nonrecursive MARPL reasoning via SPARQL
- male parent of parent is a grandfather

```
                         (?grandfather.gender=male)@?X,
                        (?grandfather.child=?parent)@?Y,
                                (?parent.child=?child)@?Z
-> (?child.relative=?grandfather)@{kinship=grandfather}
```



Nemo

Nemo Graph Rule Engine v0.7.2-dev Nemo on Github Web Interface on Github Docs

Code editor | Examples | Open file | Save file

```
@import spouse :- sparql{
  endpoint = "https://query.wikidata.org/sparql",
  query = """
  PREFIX p: <http://www.wikidata.org/prop/>
  PREFIX ps: <http://www.wikidata.org/prop/statement/>
  SELECT ?stmt ?subj ?obj WHERE {
                ?subj p:P26 ?stmt .
                ?stmt ps:P26 ?obj .
          }""" } .
violations(?stmt) :- spouse(?stmt, ?subj, ?obj), ~spouse(?reverse_stmt, ?obj, ?subj) .
@export violations :- csv{ resource = "" } .
```

- Reasoner for Datalog & Existential Rules
- Datalog + value invention: **Existential Rules**
- Supports SPARQL & RDF

Nemo

- Reasoner for Datalog & Existential Rules
- Datalog + value invention: **Existential Rules**
- Supports SPARQL & RDF

Can we use Nemo for MARPL reasoning?

Nemo & The Chase

The Chase

- Find applicable rule: premise holds, but conclusion does not
- Apply rule: add conclusion
- Repeat until nothing changes

Computes a model that can be used for deciding fact entailment

- Always terminates for Datalog

Nemo & The Chase

The Chase

- Find applicable rule: premise holds, but conclusion does not
- Apply rule: add conclusion
- Repeat until nothing changes

Computes a model that can be used for deciding fact entailment

- Always terminates for Datalog
- May not terminate for Existential Rules: “every person has a parent”
- A variant works for MARPL (always terminates)

Nemo & The Chase

The Chase

- Find applicable rule: premise holds, but conclusion does not
- Apply rule: add conclusion
- Repeat until nothing changes

Computes a model that can be used for deciding fact entailment

- Always terminates for Datalog
- May not terminate for Existential Rules: “every person has a parent”
- A variant works for MARPL (always terminates)

Plan: translate MARPL into Existential Rules

Datalog$^{\mathbf{CV}}$: **Datalog with Complex Values**

Marx & Krötzsch, International Conference on Database Theory 2022

Datalog + Sets + Tuples: Datalog$^{\mathbf{CV}}$

- Use sets of pairs for annotations: $p(x, y)@\{a : v\}$ becomes $p(x, y, \{\langle a, v\rangle\})$

$$\text{spouse}\left(\mathit{Marie}, \mathit{Pierre}, \left\{\begin{array}{c}\langle\text{begin}, 1895\rangle\,, \\ \langle\text{end}, 1906\rangle\,, \\ \langle\text{cause}, \text{death of spouse}\rangle\end{array}\right\}\right)$$

- The Chase also works for Datalog$^{\text{CV}}$ (and always terminates)

Datalog$^{\mathbf{CV}}$: **Datalog with Complex Values**

Marx & Krötzsch, International Conference on Database Theory 2022

Datalog + Sets + Tuples: Datalog$^{\mathbf{CV}}$

- Use sets of pairs for annotations: $p(x, y)@\{a : v\}$ becomes $p(x, y, \{\langle a, v\rangle\})$

$$\text{spouse}\left(\mathit{Marie}, \mathit{Pierre}, \left\{\begin{matrix}\langle \text{begin}, 1895\rangle\,, \\ \langle \text{end}, 1906\rangle\,, \\ \langle \text{cause}, \text{death of spouse}\rangle\end{matrix}\right\}\right)$$

- The Chase also works for Datalog$^{\mathrm{CV}}$ (and always terminates)

Datalog$^{\mathrm{CV}}$ + stratified negation: Datalog$^{\mathbf{CV}}_{\mathbf{neg}}$

- Stratification ensures that the Chase works in presence of negation

MARPL to Datalog^{CV}

Theorem

MARPL admits a translation into $\mathrm{Datalog}^{CV}_{neg}$ *that*

- *Preserves consequences*
- *Can be computed in* PTime
- *Preserves complexities of fact entailment*

A function-free fragment of MARPL translates into Datalog^{CV}

Data Complexity of $\text{Datalog}^{\mathbf{CV}}_{\mathbf{neg}}$

$\text{Datalog}^{\text{CV}}_{\text{neg}}$ can express all ELEMENTARY queries

Two parameters are crucial for fine-grained complexity bounds:

- maximal nesting depth of sets: **set height**
- maximal arity of tuples: **tuple width**

Theorem

Data complexity of fact entailment for $\text{Datalog}^{CV}_{neg}$ *with set height k is* kExpTime-*complete.*

Data Complexity of $\text{Datalog}^{\mathbf{CV}}_{\mathbf{neg}}$

$\text{Datalog}^{\text{CV}}_{\text{neg}}$ can express all ELEMENTARY queries

Two parameters are crucial for fine-grained complexity bounds:

- maximal nesting depth of sets: **set height**
- maximal arity of tuples: **tuple width**

Theorem

Data complexity of fact entailment for $\text{Datalog}^{CV}_{neg}$ *with set height k is k*ExpTime-*complete.*

- Optimal for Datalog (set height 0)
- Optimal for MARPL (set height 1)

Combined Complexity of $\text{Datalog}^{\mathbf{CV}}_{\mathbf{neg}}$

Theorem

Combined complexity of fact entailment for $\text{Datalog}^{CV}_{neg}$ *with set height k is*

- $(k + 1)$ExpTime-*complete if tuple width is bounded*
- $(k + 2)$ExpTime-*complete in general*

Combined Complexity of $\text{Datalog}^{\mathbf{CV}}_{\mathbf{neg}}$

Theorem

Combined complexity of fact entailment for $\text{Datalog}^{CV}_{neg}$ *with set height k is*

- $(k+1)$ExpTime-*complete if tuple width is bounded*
- $(k+2)$ExpTime-*complete in general*

- Optimal for Datalog (set height 0, bounded tuple width)
- Not optimal for MARPL (set height 1, bounded tuple width)

Combined Complexity of $\text{Datalog}^{\textbf{CV}}_{\textbf{neg}}$

Theorem

Combined complexity of fact entailment for $\text{Datalog}^{CV}_{neg}$ *with set height k is*

- $(k+1)$ExpTime-*complete if tuple width is bounded*
- $(k+2)$ExpTime-*complete in general*

- Optimal for Datalog (set height 0, bounded tuple width)
- Not optimal for MARPL (set height 1, bounded tuple width)
- We can still show the ExpTime upper bound since tuples range only over domain elements

From Datalog$^{\mathbf{CV}}$ **to Existential Rules**

Theorem

Datalog^{CV} *admits a translation into Existential Rules that*

- *Preserves consequences*
- *Can be computed in* PTime
- *Preserves complexities of fact entailment*
- *Produces rules for which the Chase always terminates*

From Datalog$^{\mathbf{CV}}$ **to Existential Rules**

Theorem

Datalog^{CV} *admits a translation into Existential Rules that*

- *Preserves consequences*
- *Can be computed in* PTime
- *Preserves complexities of fact entailment*
- *Produces rules for which the Chase always terminates*

- Similar translation from Datalog$^{\mathrm{CV}}_{\mathrm{neg}}$ into Existential Rules with stratified negation
- We can translate MARPL into Existential Rules!

Tractable Datalog^{CV}

Sets derived over any database have at most k elements:
***k*-bounded cardinality**

Theorem

Data complexity of fact entailment for bounded cardinality Datalog^{CV} *is* PTime-*complete.*

Tractable Datalog^{CV}

Sets derived over any database have at most *k* elements:
***k*-bounded cardinality**

Theorem

Data complexity of fact entailment for bounded cardinality Datalog^{CV} *is* PTime-*complete.*

- **Good**: bounded cardinality Datalog^{CV} is a tractable fragment!
- **Bad**: bounded cardinality is undecidable

Tractable Datalog$^{\mathbf{CV}}$

Sets derived over any database have at most k elements:
k**-bounded cardinality**

Theorem

Data complexity of fact entailment for bounded cardinality Datalog^{CV} *is* PTime-*complete.*

- **Good**: bounded cardinality Datalog^{CV} is a tractable fragment!
- **Bad**: bounded cardinality is undecidable
- Sufficient conditions for bounded cardinality:
  - Set acyclicity checks for recursively constructed sets
  - Cardinality contraints estimates cardinality with inequalities

Attributed Description Logics

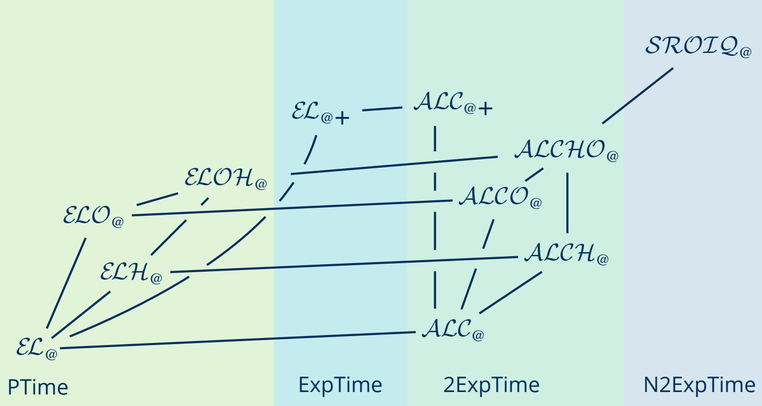
Krötzsch, Marx et al., International Semantic Web Conference 2017; International Joint Conferences on Artificial Intelligence 2018

$\mathcal{R}$: complex roles
$\mathcal{Q}$: number restrictions

+: One-or-more Annotations

$\mathcal{O}$: Nominals

$\mathcal{H}$: Role Hierarchies



Attributed Description Logics

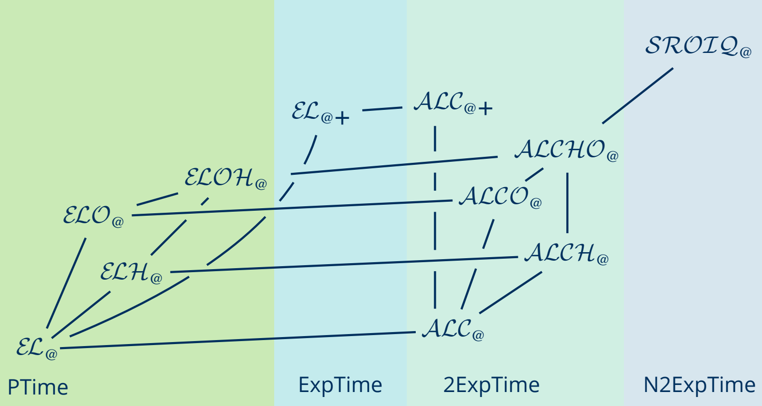
Krötzsch, Marx et al., International Semantic Web Conference 2017;
International Joint Conferences on Artificial Intelligence 2018

$\mathcal{R}$: complex roles
$\mathcal{Q}$: number restrictions

$+$: One-or-more Annotations

$\mathcal{O}$: Nominals

$\mathcal{H}$: Role Hierarchies



Attributed Description Logics

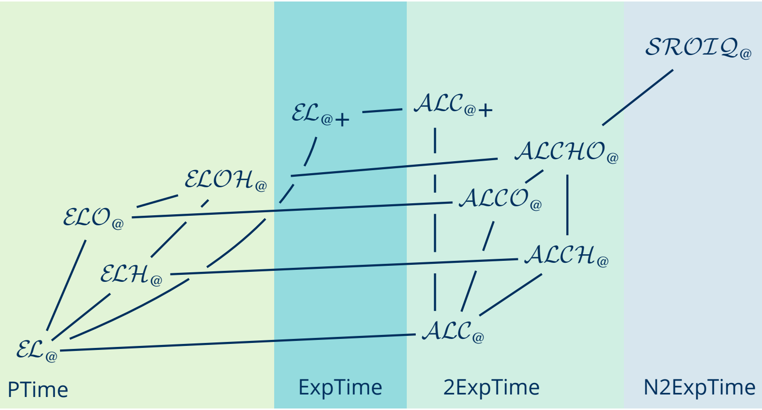
Krötzsch, Marx et al., International Semantic Web Conference 2017; International Joint Conferences on Artificial Intelligence 2018

$\mathcal{R}$: complex roles
$\mathcal{Q}$: number restrictions

+: One-or-more Annotations

$\mathcal{O}$: Nominals

$\mathcal{H}$: Role Hierarchies



Attributed Description Logics

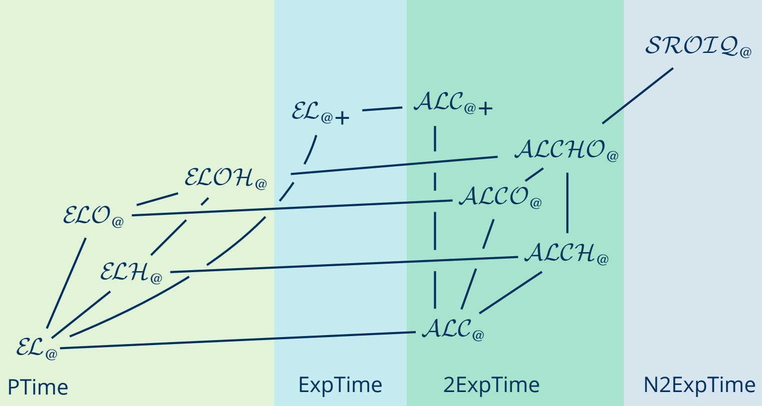
Krötzsch, Marx et al., International Semantic Web Conference 2017; International Joint Conferences on Artificial Intelligence 2018

$\mathcal{R}$: complex roles
$\mathcal{Q}$: number restrictions

+: One-or-more Annotations

$\mathcal{O}$: Nominals

$\mathcal{H}$: Role Hierarchies



Attributed Description Logics

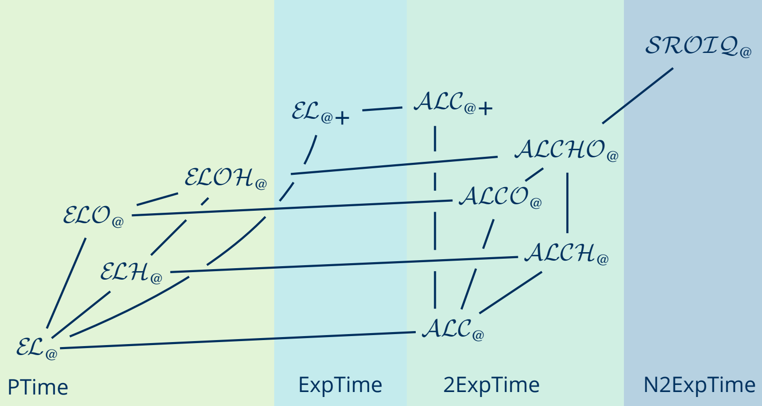
Krötzsch, Marx et al., International Semantic Web Conference 2017; International Joint Conferences on Artificial Intelligence 2018

$\mathcal{R}$: complex roles
$\mathcal{Q}$: number restrictions

+: One-or-more Annotations

$\mathcal{O}$: Nominals

$\mathcal{H}$: Role Hierarchies



Outlook

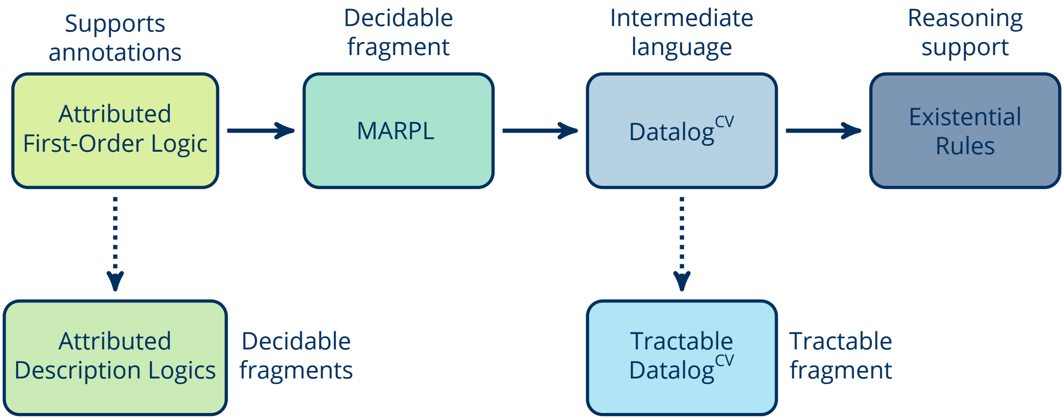
Negation & Aggregation

- When can we avoid negation when translating MARPL to Datalog^{CV}?
- Which kinds of aggregation can we add to MARPL while keeping the translation?

Extending Nemo

- Support Complex Values in Nemo
- Generalise Function Definitions to modules: isolated sets of rules that construct single values

Reflection


Supports annotations
Decidable fragment
Intermediate language
Reasoning support
Attributed First-Order Logic
MARPL
Datalog^CV
Existential Rules
Attributed Description Logics
Decidable fragments
Tractable Datalog^CV
Tractable fragment