Description Logics and other Decidable Logics for Graphs

Magdalena Ortiz

Umeå University, Sweden

WoLLIC 2023, July 11-14, Halifax, Canada

Graph-shaped Data

Graph-shaped data is **ubiquitous**:

- Web data:

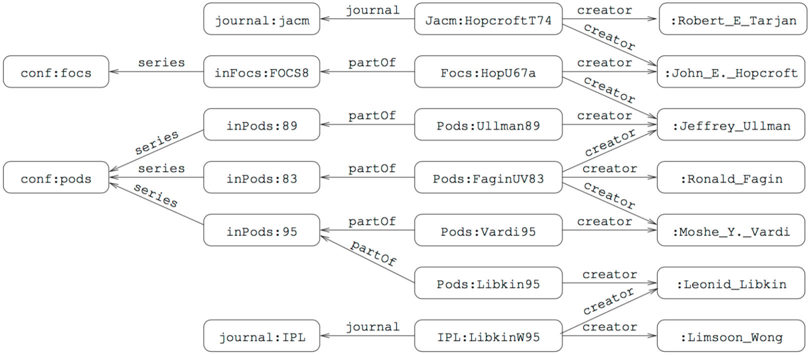
  RDF data linked open data **knowledge graphs**

- Networks of all kinds

  social biological transport interactions

- **Graph databases** eg., Neo4j

Graph data



journal:jacm
journal
Jacm:HopcroftT74
creator
:Robert_E_Tarjan
creator
conf:focs
series
inFocs:FOCS8
partOf
Focs:HopU67a
creator
:John_E._Hopcroft
creator
inPods:89
partOf
Pods:Ullman89
creator
:Jeffrey_Ullman
series
creator
conf:pods
series
inPods:83
partOf
Pods:FaginUV83
creator
:Ronald_Fagin
series
creator
inPods:95
partOf
Pods:Vardi95
creator
:Moshe_Y._Vardi
partOf
Pods:Libkin95
creator
:Leonid_Libkin
creator
journal:IPL
journal
IPL:LibkinW95
creator
:Limsoon_Wong

Logics for graph data

Graph-structured Data

- **edge- and node-labelled graphs**
- sets of ground facts over **unary** and **binary** predicates

Logic languages for **implicit facts** (background knowledge),
constraints,
queries over graphs,
data quality, data evolution, ...

Desiderata

- Expressiveness suitable for applications
- Decidable, and keeping the complexity in check

Logics for graph data

Graph-structured Data

- **edge- and node-labelled graphs**
- sets of ground facts over **unary** and **binary** predicates

Logic languages for **implicit facts** (background knowledge),
constraints,
queries over graphs,
data quality, data evolution, ...

Desiderata

- Expressiveness suitable for applications
- Decidable, and keeping the complexity in check

FOL for Graphs

We want **decidable** fragments of First-Order Logic

- FO^2, the **two variable fragment** of FOL
  - Small model property (exponential size)
  - Satisfiability is NExpTime-complete
- A popular restriction: **guarded quantification**

$$\exists y.p(x,y) \land \varphi(y) \qquad \forall y.p(x,y) \to \varphi(y)$$

$\mathcal{G}^2$, the **modal** fragment of FO^2

- a form of *locality*: tree/forest like structures
- satisfiability is in ExpTime
- More robust under extensions

Description Logics (DLs)

- mostly decidable fragments of FOL
- only unary and binary predicates
- classical FO-semantics
- closely related to FO^2, $\mathcal{G}^2$ and modal logics
- special KR-oriented syntax with no explicit variables

DLs as a toolbox

- **different DLs** with **different expressiveness**
- support **application-specific** choice of language
- focus on **expressiveness vs. complexity** of reasoning

Description Logics (DLs)

- mostly decidable fragments of FOL
- only unary and binary predicates
- classical FO-semantics
- closely related to FO^2, $\mathcal{G}^2$ and modal logics
- special KR-oriented syntax with no explicit variables

DLs as a toolbox

- **different DLs** with **different expressiveness**
- support **application-specific** choice of language
- focus on **expressiveness vs. complexity** of reasoning

$\mathcal{ALC}$, the basic DL

- FOL vocabulary with three alphabets:
  - unary predicates $\mathbf{C}$: concept names, for labelling nodes
  - binary predicates $\mathbf{R}$: role names, for labelling edges
  - constants $\mathbf{I}$: individuals, node names

- We write 'unary' formulas called concepts $\quad A \in \mathbf{C}, \quad r \in \mathbf{R}$

$$C \quad ::= \quad A \mid \neg C \mid C \sqcap C \mid C \sqcup C \mid \forall r.C \mid \exists r.C$$

Syntactic variant of multi-dimensional modal logic

We use $ST_x(C)$ to denote the usual FOL translation of C with x free

$\mathcal{ALC}$, the basic DL

- FOL vocabulary with three alphabets:
  - unary predicates $\mathbf{C}$: concept names, for labelling nodes
  - binary predicates $\mathbf{R}$: role names, for labelling edges
  - constants $\mathbf{I}$: individuals, node names
- We write 'unary' formulas called concepts $A \in \mathbf{C}, \quad r \in \mathbf{R}$

$$C \quad ::= \quad A \mid \neg C \mid C \sqcap C \mid C \sqcup C \mid \forall r.C \mid \exists r.C$$

Syntactic variant of multi-dimensional modal logic

We use $ST_x(C)$ to denote the usual FOL translation of C with x free

$\mathcal{ALC}$, the basic DL

- FOL vocabulary with three alphabets:
  - unary predicates $\mathbf{C}$: concept names, for labelling nodes
  - binary predicates $\mathbf{R}$: role names, for labelling edges
  - constants $\mathbf{I}$: individuals, node names

- We write 'unary' formulas called concepts $\quad A \in \mathbf{C}, \quad r \in \mathbf{R}$

$$C \quad ::= \quad A \mid \neg C \mid C \sqcap C \mid C \sqcup C \mid \forall r.C \mid \exists r.C$$

Syntactic variant of multi-dimensional modal logic

We use $ST_x(C)$ to denote the usual FOL translation of C with x free

Knowledge Bases

In DLs, we consider **knowledge bases** with two parts:

- **ABox, Data**: facts, a graph
- **TBox, Ontology**: knowledge, universally quantified implications between unary formulas

$$C \sqsubseteq D \qquad \forall x\; ST_x(C)\, x \rightarrow ST_x(D)\, x$$

Models

FO interpretations $\mathcal{I}$ (multi-dimensional Kripke structures, edge- and node-labeled graphs)

An interpretation is a model if it satisfies the ABox and the TBox

Complexity

Satisfiability of a KB is ExpTime-complete

Knowledge Bases

In DLs, we consider **knowledge bases** with two parts:

- **ABox, Data**: facts, a graph
- **TBox, Ontology**: knowledge, universally quantified implications between unary formulas

$$C \sqsubseteq D \qquad \forall x \; ST_x(C)\, x \to ST_x(D)\, x$$

Models

FO interpretations $\mathcal{I}$ (multi-dimensional Kripke structures, edge- and node-labeled graphs)

An interpretation is a model if it satisfies the ABox and the TBox

Complexity

Satisfiability of a KB is ExpTime-complete

Knowledge Bases

In DLs, we consider **knowledge bases** with two parts:

- **ABox, Data**: facts, a graph
- **TBox, Ontology**: knowledge, universally quantified implications between unary formulas

$$C \sqsubseteq D \qquad \forall x \; ST_x(C)\, x \rightarrow ST_x(D)\, x$$

Models

FO interpretations $\mathcal{I}$ (multi-dimensional Kripke structures, edge- and node-labeled graphs)

An interpretation is a model if it satisfies the ABox and the TBox

Complexity

Satisfiability of a KB is ExpTime-complete

What are we missing? 1: Inverses

Inverse roles $\mathcal{I}$, i.e., converse modalities

$$\mathsf{Valve} \sqcap \exists \mathsf{hasPart}^{-}.\mathsf{Heart} \sqsubseteq \mathsf{HeartValve}$$

- Still in $\mathcal{G}^2$
- No significant effect on the complexity of expressive DLs

What else do we need? 2: Counting $\mathcal{Q}$

Counting $\mathcal{Q}$, i.e., graded modalities

$$\mathsf{HumanHeart} \sqsubseteq \ \geq 4 \ \mathsf{hasPart.Valve}$$

Not possible with bounded variables!

- $\mathcal{C}^2$ and $\mathcal{GC}^2$ extend FO^2 and $\mathcal{G}^2$ with **counting quantifiers**
- Same complexity as non-counting fragments

Functionality $\mathcal{F}$: simplest form of counting

$$func(\mathsf{hasSocSecNumber})$$

Infinity axioms

$\mathcal{ALCIF}$ does not have the finite model property

$$A(c), B(c) \qquad A \sqsubseteq \exists r.A \qquad \exists r^{-}.\top \sqsubseteq \neg B \qquad \top \sqsubseteq \ \leq_1 r^{-}$$

What else do we need? 2: Counting $\mathcal{Q}$

Counting $\mathcal{Q}$, i.e., graded modalities

$$\text{HumanHeart} \sqsubseteq \ \geq 4 \ \text{hasPart.Valve}$$

Not possible with bounded variables!

- $\mathcal{C}^2$ and $\mathcal{GC}^2$ extend FO^2 and $\mathcal{G}^2$ with **counting quantifiers**
- Same complexity as non-counting fragments

Functionality $\mathcal{F}$: simplest form of counting

$$func(\text{hasSocSecNumber})$$

Infinity axioms

$\mathcal{ALCIF}$ does not have the finite model property

$$A(c), B(c) \qquad A \sqsubseteq \exists r.A \qquad \exists r^-.\top \sqsubseteq \neg B \qquad \top \sqsubseteq \ \leq_1 r^-$$

Nominals $\mathcal{O}$

aka *one-of*

Builds a concept from a set of individuals $\{a_1, \ldots, a_n\}$

$$\text{EU\_Country} \sqsubseteq \{\mathrm{Austria}, \mathrm{Belgium}, \ldots, \mathrm{Sweden}\}$$

$$\text{PrimordialGods} \sqsubseteq \{\mathrm{Gaia}, \mathrm{Chaos}, \mathrm{Chronos}, \mathrm{Ananke}\}$$

- Brings us to **Hybrid Logic**
- Introduces equality to the FO translation

$\mathcal{ALCOIF}$ is hard for NExpTime

With $\mathcal{O} + \mathcal{I} + \mathcal{F}/\mathcal{Q}$ we can enforce an exponentially large grid

Nominals $\mathcal{O}$
aka *one-of*

Builds a concept from a set of individuals $\{a_1, \ldots, a_n\}$

$$\mathsf{EU\_Country} \sqsubseteq \{\mathrm{Austria}, \mathrm{Belgium}, \ldots, \mathrm{Sweden}\}$$

$$\mathsf{PrimordialGods} \sqsubseteq \{\mathrm{Gaia}, \mathrm{Chaos}, \mathrm{Chronos}, \mathrm{Ananke}\}$$

- Brings us to **Hybrid Logic**
- Introduces equality to the FO translation

$\mathcal{ALCOIF}$ is hard for NExpTime

With $\mathcal{O} + \mathcal{I} + \mathcal{F}/\mathcal{Q}$ we can enforce an exponentially large grid

Nominals $\mathcal{O}$

aka *one-of*

Builds a concept from a set of individuals $\{a_1, \ldots, a_n\}$

$$\textsf{EU\_Country} \sqsubseteq \{\mathrm{Austria}, \mathrm{Belgium}, \ldots, \mathrm{Sweden}\}$$

$$\textsf{PrimordialGods} \sqsubseteq \{\mathrm{Gaia}, \mathrm{Chaos}, \mathrm{Chronos}, \mathrm{Ananke}\}$$

- Brings us to **Hybrid Logic**
- Introduces equality to the FO translation

$\mathcal{ALCOIF}$ is hard for NExpTime

With $\mathcal{O} + \mathcal{I} + \mathcal{F}/\mathcal{Q}$ we can enforce an exponentially large grid

Nominals $\mathcal{O}$
aka *one-of*

Builds a concept from a set of individuals $\{a_1, \ldots, a_n\}$

$$\mathsf{EU\_Country} \sqsubseteq \{\mathrm{Austria}, \mathrm{Belgium}, \ldots, \mathrm{Sweden}\}$$

$$\mathsf{PrimordialGods} \sqsubseteq \{\mathrm{Gaia}, \mathrm{Chaos}, \mathrm{Chronos}, \mathrm{Ananke}\}$$

- Brings us to **Hybrid Logic**
- Introduces equality to the FO translation

$\mathcal{ALCOIF}$ is hard for NExpTime

With $\mathcal{O} + \mathcal{I} + \mathcal{F}/\mathcal{Q}$ we can enforce an exponentially large grid

What else do we need?

Transitive relations: partOf nextState ancestors

Two main solutions†:

1. transitive roles ($\mathcal{S}$), usually with role inclusions ($\mathcal{SH}$)

   $trans$(partOf) hasParent$\sqsubseteq$hasAncestor, $trans$(hasAncestor)

   FOL expressible (with three variables)

2. allow Kleene star $*$ (or even *regular expressions* reg)

$$\forall(\text{partOf})^*.\text{Heart} \sqsubseteq \text{CardiacStructure}$$

No type of transitivity in counting concepts!

Wait, this also looks familiar...

We are now in the world of **hybrid PDL** (and outside FOL!)

We can also go further: **hybrid μ calculus**, $\mu\mathcal{ALCIO}$, ...

† The first option is easier to implement efficiently

What else do we need?

Transitive relations: partOf nextState ancestors

Two main solutions[†]:

1. transitive roles ($\mathcal{S}$), usually with role inclusions ($\mathcal{SH}$)

   $trans$(partOf) hasParent$\sqsubseteq$hasAncestor, $trans$(hasAncestor)

   FOL expressible (with three variables)

2. allow Kleene star $*$ (or even *regular expressions* reg)

$$\forall(\text{partOf})^{*}.\text{Heart} \sqsubseteq \text{CardiacStructure}$$

No type of transitivity in counting concepts!

Wait, this also looks familiar...

We are now in the world of **hybrid PDL** (and outside FOL!)

We can also go further: **hybrid μ calculus**, $\mu\mathcal{ALCIO}$, ...

† The first option is easier to implement efficiently

What else do we need?

Transitive relations: partOf nextState ancestors

Two main solutions†:

1. transitive roles ($\mathcal{S}$), usually with role inclusions ($\mathcal{SH}$)

   $trans(\text{partOf})$ hasParent$\sqsubseteq$hasAncestor, $trans(\text{hasAncestor})$

   FOL expressible (with three variables)

2. allow Kleene star $*$ (or even *regular expressions* reg)

$$\forall(\text{partOf})^*.\text{Heart} \sqsubseteq \text{CardiacStructure}$$

No type of transitivity in counting concepts!

Wait, this also looks familiar...

We are now in the world of **hybrid PDL** (and outside FOL!)

We can also go further: **hybrid μ calculus**, $\mu\mathcal{ALCIO}$, ...

† The first option is easier to implement efficiently

What else do we need?

Transitive relations: partOf nextState ancestors

Two main solutions†:

1. transitive roles ($\mathcal{S}$), usually with role inclusions ($\mathcal{SH}$)

   $trans$(partOf) hasParent $\sqsubseteq$ hasAncestor, $trans$(hasAncestor)

   FOL expressible (with three variables)

2. allow Kleene star $*$ (or even *regular expressions* reg)

   $$\forall(\text{partOf})^*.\text{Heart} \sqsubseteq \text{CardiacStructure}$$

No type of transitivity in counting concepts!

Wait, this also looks familiar...

We are now in the world of **hybrid PDL** (and outside FOL!)

We can also go further: **hybrid μ calculus**, $\mu\mathcal{ALCIO}$, ...

† The first option is easier to implement efficiently

What else do we need?

Transitive relations: partOf nextState ancestors

Two main solutions†:

1. transitive roles ($\mathcal{S}$), usually with role inclusions ($\mathcal{SH}$)

   $trans$(partOf) hasParent$\sqsubseteq$hasAncestor, $trans$(hasAncestor)

   FOL expressible (with three variables)

2. allow Kleene star $*$ (or even *regular expressions* reg)

$$\forall(\text{partOf})^*.\text{Heart} \sqsubseteq \text{CardiacStructure}$$

No type of transitivity in counting concepts!

Wait, this also looks familiar...

We are now in the world of **hybrid PDL** (and outside FOL!)

We can also go further: **hybrid μ calculus**, $\mu\mathcal{ALCIO}$, ...

† The first option is easier to implement efficiently

Complexity and Decidability

- With any 2 of $\mathcal{O} + \mathcal{I} + \mathcal{F}/\mathcal{Q}$, remains decidable in **ExpTime**

  $\mathcal{SHIQ}$ $\mathcal{ALCHIQ}^*$ $\mathcal{ALCHIQ}_{reg}$ $\mu\mathcal{ALCHIQ}$
  $\mathcal{SHOQ}$ $\mathcal{ALCHOQ}^*$ $\mathcal{ALCHOQ}_{reg}$ $\mu\mathcal{ALCHOQ}$
  $\mathcal{SHOI}$ $\mathcal{ALCHOI}^*$ $\mathcal{ALCHOI}_{reg}$ $\mu\mathcal{ALCHOI}$

- Moreover, we have good reasoners (for $\mathcal{S}, \mathcal{SH}, \mathcal{SR}$)

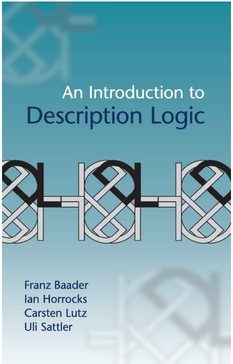
- But with all three $\mathcal{O} + \mathcal{I} + \mathcal{F}/\mathcal{Q}$:
  - $\mathcal{SHOIQ}$ is NExpTime complete
  - $\mu\mathcal{ALCHOIF}$ is **undecidable!**
  - $\mathcal{ALCHOIF}^*$ is a **long standing open problem!**

Some things I didn't mention

- Lightweight DLs are a huge area
- Query languages for graphs
  - DLs / guarded formalisms very poor as query languages
  - Orthogonal formalisms for the data and the query
  - Exciting and evolving area!
- Property graphs
  - Adding data values to the logics
  - Concrete domains
- Here, only **classical semantics**
  - Minimal model reasoning
  - Non-monotonic DLs
  - paraconsistent, probabilistic, . . .

Want to know more?

There is a quite accessible textbook, as well as many papers and tutorial notes.



If you want to read also about query languages for graphs:

- M. Bienvenu, M. Ortiz: Ontology-Mediated Query Answering with Data-Tractable Description Logics. Reasoning Web 2015: 218-307
- N. Francis, A. Gheerbrant, P. Guagliardo, L. Libkin, V. Marsault, W. Martens, F. Murlak, L. Peterfreund, A. Rogova, D. Vrgoc: A Researcher's Digest of GQL (Invited Talk). ICDT 2023: 1:1-1:22