

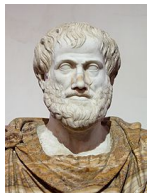
Data and Knowledge Management

MSc topics 2025

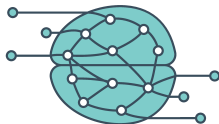
Universitetet i Oslo

24 September 2025

Data and Knowledge Management



Logic,
Reasoning



Data Science
AI/ML



Digital
Twins



Semantic Web
Technologies



Data
Management

DKM MSc topics

DKM:

<https://www.mn.uio.no/ifi/studier/masteroppgaver/dkm/>

Topic areas

- Machine Learning, Data Science
- Ontology-Based Information Systems
- Data Streams and Applications
- Logic and Theoretical Computer Science

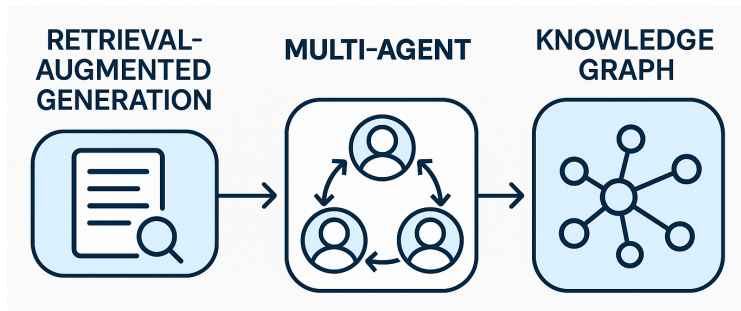
How trainable are GNNs?

(Comparing trainability of Graph Neural Networks for logic-expressible functions)

- Graph Neural Networks (GNNs):
 - a family of modern NN architectures
 - work with structured data (e.g., knowledge graphs)
- (Theoretical) expressivity of GNNs:
 - different GNN architectures express different sets of functions on graphs
 - well-understood by connecting to logics
- (Practical) trainability of GNNs:
 - "can express in theory" does not imply "can be trained"
 - Question:
Which GNNs architectures can be trained for which logic-expressed functions?

Supervisors: Egor V. Kostylev (egork@ifi.uio.no), Roxana Pop (roxanap@ifi.uio.no)

Multiagent RAG Systems for Industrial Applications



- RAG: Retrieval Augmented Generation, LLMs+structured data
- Project: use multi-agent methods to improve RAG
- Context is industrial data in Bosch

SPARQL Query Optimisation with SHACL Constraints

SPARQL

```
SELECT ?x ?y WHERE {  
  ?x a :Employee .  
  OPTIONAL {  
    ?x :hasAddress ?y .  
  }  
}
```

SPARQL Query Optimisation with SHACL Constraints

SPARQL

```
SELECT ?x ?y WHERE {  
  ?x a :Employee .  
  OPTIONAL {  
    ?x :hasAddress ?y .  
  }  
}
```

SHACL

```
[  
  a :NodeShape ;  
  sh:targetClass :Employee ;  
  sh:property [  
    sh:path :hasAddress ;  
    sh:minCount 1  
  ] .  
]
```

SPARQL Query Optimisation with SHACL Constraints

SPARQL

```
SELECT ?x ?y WHERE {  
  ?x a :Employee .  
  OPTIONAL {  
    ?x :hasAddress ?y .  
  }  
}
```

- 'OPTIONAL' is clearly not needed in the query

SHACL

```
[] a :NodeShape ;  
  sh:targetClass :Employee ;  
  sh:property [  
    sh:path :hasAddress ;  
    sh:minCount 1  
  ] .
```

SPARQL Query Optimisation with SHACL Constraints

SPARQL

```
SELECT ?x ?y WHERE {  
  ?x a :Employee .  
  OPTIONAL {  
    ?x :hasAddress ?y .  
  }  
}
```

SHACL

```
[] a :NodeShape ;  
  sh:targetClass :Employee ;  
  sh:property [  
    sh:path :hasAddress ;  
    sh:minCount 1  
  ] .
```

- 'OPTIONAL' is clearly not needed in the query
- But who would be so stupid to write it then?

SPARQL Query Optimisation with SHACL Constraints

SPARQL

```
SELECT ?x ?y WHERE {  
  ?x a :Employee .  
  OPTIONAL {  
    ?x :hasAddress ?y .  
  }  
}
```

SHACL

```
[] a :NodeShape ;  
  sh:targetClass :Employee ;  
  sh:property [  
    sh:path :hasAddress ;  
    sh:minCount 1  
  ] .
```

- 'OPTIONAL' is clearly not needed in the query
- But who would be so stupid to write it then?
 - Queries generated from User Interfaces

SPARQL Query Optimisation with SHACL Constraints

SPARQL

```
SELECT ?x ?y WHERE {  
  ?x a :Employee .  
  OPTIONAL {  
    ?x :hasAddress ?y .  
  }  
}
```

SHACL

```
[] a :NodeShape ;  
  sh:targetClass :Employee ;  
  sh:property [  
    sh:path :hasAddress ;  
    sh:minCount 1  
  ] .
```

- 'OPTIONAL' is clearly not needed in the query
- But who would be so stupid to write it then?
 - Queries generated from User Interfaces
 - Queries that work over a broad selection of data sets

SPARQL Query Optimisation with SHACL Constraints

SPARQL

```
SELECT ?x ?y WHERE {  
  ?x a :Employee .  
  OPTIONAL {  
    ?x :hasAddress ?y .  
  }  
}
```

SHACL

```
[] a :NodeShape ;  
  sh:targetClass :Employee ;  
  sh:property [  
    sh:path :hasAddress ;  
    sh:minCount 1  
  ] .
```

- 'OPTIONAL' is clearly not needed in the query
- But who would be so stupid to write it then?
 - Queries generated from User Interfaces
 - Queries that work over a broad selection of data sets
- Theory is ready — we need an implementation!

SPARQL Query Optimisation with SHACL Constraints

SPARQL

```
SELECT ?x ?y WHERE {  
  ?x a :Employee .  
  OPTIONAL {  
    ?x :hasAddress ?y .  
  }  
}
```

SHACL

```
[] a :NodeShape ;  
  sh:targetClass :Employee ;  
  sh:property [  
    sh:path :hasAddress ;  
    sh:minCount 1  
  ] .
```

- 'OPTIONAL' is clearly not needed in the query
- But who would be so stupid to write it then?
 - Queries generated from User Interfaces
 - Queries that work over a broad selection of data sets
- Theory is ready — we need an implementation!

Reasoning about Descriptions

- An identifier: “King Harald V”

Reasoning about Descriptions

- An identifier: “King Harald V”
- A description: “The king of Norway”

Reasoning about Descriptions

- An identifier: “King Harald V”
- A description: “The king of Norway”
- In Description Logic: $\epsilon(\exists \text{kingOf}.\{\text{Norway}\})$

Reasoning about Descriptions

- An identifier: “King Harald V”
- A description: “The king of Norway”
- In Description Logic: $\epsilon(\exists \text{kingOf}.\{\text{Norway}\})$
- Recent MSc thesis on this topic

Reasoning about Descriptions

- An identifier: “King Harald V”
- A description: “The king of Norway”
- In Description Logic: $\epsilon(\exists \text{kingOf}.\{\text{Norway}\})$
- Recent MSc thesis on this topic
- Remains to be done:

Reasoning about Descriptions

- An identifier: “King Harald V”
- A description: “The king of Norway”
- In Description Logic: $\epsilon(\exists \text{kingOf}.\{\text{Norway}\})$
- Recent MSc thesis on this topic
- Remains to be done:
 - Implementation, extending some existing DL theorem prover X

Reasoning about Descriptions

- An identifier: “King Harald V”
- A description: “The king of Norway”
- In Description Logic: $\epsilon(\exists \text{kingOf}.\{\text{Norway}\})$
- Recent MSc thesis on this topic
- Remains to be done:
 - Implementation, extending some existing DL theorem prover X
 - Depending on X some theory to fill the gap

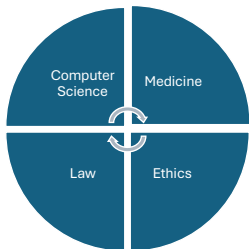
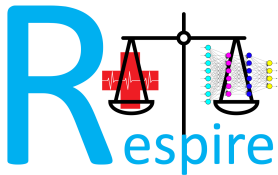
Reasoning about Descriptions

- An identifier: “King Harald V”
- A description: “The king of Norway”
- In Description Logic: $\epsilon(\exists \text{kingOf}.\{\text{Norway}\})$
- Recent MSc thesis on this topic
- Remains to be done:
 - Implementation, extending some existing DL theorem prover X
 - Depending on X some theory to fill the gap
 - ~~Some theory (not implementation-related)~~

Reasoning about Descriptions

- An identifier: “King Harald V”
- A description: “The king of Norway”
- In Description Logic: $\epsilon(\exists \text{kingOf}.\{\text{Norway}\})$
- Recent MSc thesis on this topic
- Remains to be done:
 - Implementation, extending some existing DL theorem prover X
 - Depending on X some theory to fill the gap
 - ~~Some theory (not implementation-related)~~

Contact: Martin Giese (martingi@uio.no)



- xAI for sleep related breathing disorders
 - Infants
 - Severe cases requiring mechanical ventilation
- Potential topics:
 - Empirical evaluation of explanations
 - Testing xAI approaches
 - Knowledge graphs for explanations
- We design the thesis topic together with you!
- Contact: Thomas Plagemann,
plageman@ifi.uio.no