Finding Patterns in Semantic Graph Formalisms

Gokarna Sharma

Advisors
Prof. Enrico Franconi
Dr. Peter F. Patel-Schneider

Faculty of Computer Science
Free University of Bozen-Bolzano

October 17, 2008
Thesis Defense
Talk Outline

1. Semantic Graph Formalisms
   - Motivation
   - Semantic Graphs
   - Ontology Graphs

2. Reasoning on Semantic Graphs using DLP
   - Pattern Finding in Semantic Graphs
   - Performance Analysis

3. Reasoning on Semantic Graphs using DLs
   - Test Knowledge Bases and Queries
   - Performance Evaluation

4. Prospects
Example Scenario (1/2)

Bibliographical example

M wrote P.
P is published in J.
R publishes J.
M belongs to R.

R is an organization.
J is a journal.
M is an author.
P is a paper.

How this information is related and can be represented?

One approach:

\[ M \xrightarrow{Wrote} P \xrightarrow{Published\_in} J \xrightarrow{Publishes} R \xrightarrow{Belongs\_to} M \]

Is it sufficient?
Example Scenario (1/2)

Bibliographical example

M wrote P.
P is published in J.
R publishes J.
M belongs to R.

R is an organization.
J is a journal.
M is an author.
P is a paper.

How this information is related and can be represented?

One approach:

Is it sufficient?
Example Scenario (1/2)

Bibliographical example

- M wrote P.
- P is published in J.
- R publishes J.
- M belongs to R.
- R is an organization.
- J is a journal.
- M is an author.
- P is a paper.

- How this information is related and can be represented?
- One approach:

  ![Diagram](image)

  - Is it sufficient?
Bibliographical example

- M wrote P.
- P is published in J.
- R publishes J.
- M belongs to R.
- R is an organization.
- J is a journal.
- M is an author.
- P is a paper.

- How this information is related and can be represented?
- One approach:

  ![Diagram](image)

- Is it sufficient?
Example Scenario (1/2)

Bibliographical example

M wrote P.
P is published in J.
R publishes J.
M belongs to R.

R is an organization.
J is a journal.
M is an author.
P is a paper.

How this information is related and can be represented?

One approach:

Is it sufficient?
Example Scenario (1/2)

Bibliographical example

M wrote P.
P is published in J.
R publishes J.
M belongs to R.

R is an organization.
J is a journal.
M is an author.
P is a paper.

How this information is related and can be represented?

One approach:

How is it sufficient?
Example Scenario (1/2)

Bibliographical example

- M wrote P.
- P is published in J.
- R publishes J.
- M belongs to R.
- R is an organization.
- J is a journal.
- M is an author.
- P is a paper.

- How this information is related and can be represented?
- One approach:

![Diagram]

- Is it sufficient?

Gokarna Sharma
Finding Patterns in Semantic Graph Formalisms
Example Scenario (1/2)

Bibliographical example

M wrote P.
P is published in J.
R publishes J.
M belongs to R.
R is an organization.
J is a journal.
M is an author.
P is a paper.

- How this information is related and can be represented?
- One approach:

![Graph diagram]

- Is it sufficient?
Alternative approach:

This approach gives concept of a semantic graph (SG) regulated by an ontology graph (OG).

How?
This approach gives concept of a semantic graph (SG) regulated by an ontology graph (OG).

How?
Example Scenario (2/2)

- Alternative approach:

```
Author : M

Wrote

Paper : P

Published_in

Organization : R

Belongs_to

Publishes

Journal : J
```

- This approach gives concept of a **semantic graph** (SG) regulated by an **ontology graph** (OG).

- How?
Alternative approach:

This approach gives concept of a semantic graph (SG) regulated by an ontology graph (OG).

How?
Alternative approach:

- This approach gives concept of a semantic graph (SG) regulated by an ontology graph (OG).

- How?
  - Type information is associated with nodes and edges.
  - Map instances of objects in SG to associated object types in OG.
This approach gives concept of a **semantic graph** (SG) regulated by an **ontology graph** (OG).

**How?**
- Type information is associated with nodes and edges.
- Map instances of objects in SG to associated object types in OG.
Semantic Graphs (1/2)

- Carry semantic information on nodes and edges.
  - Nodes represent objects of different types (e.g., person, paper, etc.) and links (or edges) represent binary relation between those objects (e.g., friendship, citation, etc.). Examples: Kinship network, WordNet, etc.
    - Contrast to other graphs — heterogenous nodes and edges.
  - Nodes and links related through ontology graph (or schema).
  - Similar to multi-relational networks.
  - Biasness due to subjectiveness in selection of nodes and edges.
Semantic Graphs (1/2)

- Carry semantic information on nodes and edges.
- Nodes represent objects of different types (e.g., person, paper, etc.) and links (or edges) represent binary relation between those objects (e.g., friendship, citation, etc.). Examples: Kinship network, WordNet, etc.
  - Contrast to other graphs — heterogeneous nodes and edges.

- Nodes and links related through ontology graph (or schema).
- Similar to multi-relational networks.
- Biasness due to subjectiveness in selection of nodes and edges.
Semantic Graphs (1/2)

- Carry semantic information on nodes and edges.

- Nodes represent **objects of different types** (e.g., person, paper, etc.) and links (or edges) represent **binary relation between those objects** (e.g., friendship, citation, etc.). Examples: Kinship network, WordNet, etc.
  
  - Contrast to other graphs — **heterogenous** nodes and edges.

- Nodes and links related through ontology graph (or **schema**).
  
  - Similar to multi-relational networks.
  
  - **Biasness** due to subjectiveness in selection of nodes and edges.
Semantic Graphs (1/2)

- Carry semantic information on nodes and edges.
- Nodes represent **objects of different types** (e.g., person, paper, etc.) and links (or edges) represent **binary relation between those objects** (e.g., friendship, citation, etc.). Examples: Kinship network, WordNet, etc.

  - Contrast to other graphs — **heterogenous** nodes and edges.

- Nodes and links related through **ontology graph** (or **schema**).

- **Similar to multi-relational networks.**

- **Biasness** due to subjectiveness in selection of nodes and edges.
Semantic Graphs (1/2)

- Carry semantic information on nodes and edges.
- Nodes represent **objects of different types** (e.g., person, paper, etc.) and links (or edges) represent **binary relation between those objects** (e.g., friendship, citation, etc.). Examples: Kinship network, WordNet, etc.
  - Contrast to other graphs — **heterogenous** nodes and edges.
- Nodes and links related through ontology graph (or **schema**).
- Similar to **multi-relational networks**.
- **Biasness** due to subjectiveness in selection of nodes and edges.
A semantic graph is a quintuple $G = (V, E, L, vt, et)$, where

- $V = \{v_1, \cdots, v_n\}$ is a finite set of vertices,
- $L$ is a finite set of edge labels in the semantic graph,
- $E \subseteq V \times L \times V$, where $v_i, v_j \in V$, $l \in L$ and $i, j = 1, \cdots, n$ is a finite set of edges,
- $vt$ is a mapping from $V$ to $T_V$ that associates a vertex type of the ontology graph with each vertex of semantic graph, and
- $et$ denotes a mapping from $E$ to $T_E$ that associates an edge type of the ontology graph to each edge of semantic graph.
Ontology Graph (1/2)

- Maintains abstract view of semantic graphs and helps to formalize patterns.
  - Semantic graphs contain **ONLY** the instantiations of vertex and edge types associated in their corresponding ontology graph.
  - Controls semantic graphs - permissible relationships only.

- Helpful: finding exact/inexact matches - through ontology hierarchy.
Ontology Graph (1/2)

- Maintains abstract view of semantic graphs and helps to formalize patterns.
- Semantic graphs contain **ONLY** the instantiations of vertex and edge types associated in their corresponding ontology graph.
  - Controls semantic graphs - permissible relationships only.
- Helpful: finding exact/inexact matches - through ontology hierarchy.
Maintains abstract view of semantic graphs and helps to formalize patterns.

Semantic graphs contain **ONLY** the instantiations of vertex and edge types associated in their corresponding ontology graph.

- Controls semantic graphs - permissible relationships only.

Helpful: finding exact/inexact matches - through ontology hierarchy.
Ontology Graph

An ontology graph is a quadruple $T = (T_V, T_E, L, I)$, where

- $T_V = \{t_1, \cdots, t_n\}$ is a finite set of $n$ vertex types,
- $L$ is a finite set of edge labels in the ontology graph,
- $T_E \in \{ (t_i, l, t_j) \subseteq T_V \times L \times T_V, \text{ where } t_i, t_j \in T_V, l \in L \text{ and } i, j = 1, \cdots, n \}$ is a finite set of edge types, and
- $I$ is the partial order binary relation “$\subseteq$” over the finite set of vertex types $T_V$ which is reflexive, antisymmetric, and transitive, i.e., for all $a$, $b$, and $c$ in $T_V$, we have that:
  - $a \subseteq a$ (reflexivity);
  - if $a \subseteq b$ and $b \subseteq a$ then $a = b$ (antisymmetry); and
  - if $a \subseteq b$ and $b \subseteq c$ then $a \subseteq c$ (transitivity).
Where are semantic graphs useful?

- Intelligence analysis in homeland security and crime analysis.
  - Identify useful information (normal/abnormal patterns and instances) for intelligence analysis in large data sets.

  - Problem is not a lack of information but information overload.

  - Till today, done “manually” - time consuming and labor-intensive.

- Several existing machine learning frameworks:
  - Not so powerful.

  - Lack formal syntax and semantics.

- What semantic graphs do?:
  - Capture meaning about the situation and dynamics of the actors – knowledge discovery.

- **Goal:** a new approach to find useful patterns and instances in semantic graphs from logic based KR&R perspective.
Where are semantic graphs useful?

- Intelligence analysis in homeland security and crime analysis.
  - Identify useful information (normal/abnormal patterns and instances) for intelligence analysis in large data sets.
  - Problem is not a lack of information but information overload.
  - Till today, done “manually” - time consuming and labor-intensive.

- Several existing machine learning frameworks:
  - Not so powerful.
  - Lack formal syntax and semantics.

- What semantic graphs do?:
  - Capture meaning about the situation and dynamics of the actors — knowledge discovery.

- Goal: a new approach to find useful patterns and instances in semantic graphs from logic based KR&R perspective.
Where are semantic graphs useful?

- Intelligence analysis in homeland security and crime analysis.
  - Identify useful information (normal/abnormal patterns and instances) for intelligence analysis in large data sets.
  
  - Problem is not a lack of information but information overload.
  
  - Till today, done “manually” - time consuming and labor-intensive.

- Several existing machine learning frameworks:
  - Not so powerful.
  
  - Lack formal syntax and semantics.

- What semantic graphs do?:
  - Capture meaning about the situation and dynamics of the actors – knowledge discovery.

- **Goal**: a new approach to find useful patterns and instances in semantic graphs from logic based KR&R perspective.
Where are semantic graphs useful?

- Intelligence analysis in homeland security and crime analysis.
  - Identify useful information (normal/abnormal patterns and instances) for intelligence analysis in large data sets.
  - Problem is not a lack of information but information overload.
  - Till today, done “manually” - time consuming and labor-intensive.

- Several existing machine learning frameworks:
  - Not so powerful.
  - Lack formal syntax and semantics.

- What semantic graphs do?:
  - Capture meaning about the situation and dynamics of the actors – knowledge discovery.

- Goal: a new approach to find useful patterns and instances in semantic graphs from logic based KR&R perspective.
Why Disjunctive Logic Programming (DLP)?  
(for semantic graph formalisms)

- Powerful answer set programming tools.
- Disjunction, true negation and constraints.
- Result optimization and querying.
Pattern Analysis Framework

Figure: DLP based pattern finding framework
Pattern Induction

Patterns

- Labeled graph that describes the structure and content of the semantic graph entities to be matched.

- Continuous path along root node to its leaf node.

- Two approaches:
  - **Inverse resolution** — invert the SLD-resolution proof procedure.
  - **Variable relaxation** — Replace the constant with variables for finding paths go through the same nodes.

\[
\text{cites}(P_3, P_1) \land \text{published_in}(P_1, J_1)
\]

\[
\text{cites}(P_4, P_3) \land \text{published_in}(P_3, J_1)
\]

After relaxation:

\[
\text{cites}(X, Y) \land \text{published_in}(Y, Z)
\]
Pseudo-code Algorithm

Input $G$ as a semantic graph $(V, E, L, vt, et)$ with nodes $V = \{v_1, v_2, \ldots, v_{|V|}\}$, edge relations $E = \{e_1, e_2, \ldots, e_{|E|}\}$, respectively the vertex types $T_V = \{t_{v1}, t_{v2}, \ldots, t_{|T_V|}\}$ of ontology graph $O = (T_V, T_E, L, I)$, and each $E_i$ links a source node $u$ to a target node $v$ via link of type $e_i$ (same for $T_E$ too) and $vt$ maps the $V$ of the semantic graph to the $T_V$ of ontology graph $O$.

begin
1. $P_t := \text{answer\_query\_pattern}(G, O, Q, F)$
   define DLV query pattern : $Q$
   for $n = 1$ to $|V|$
   while $p_t \in P_t$
   extract pattern answers $p_{ti} := dlv(G, O, \phi_{Qi}(G), F)$
   end

2. $P_t := \text{find\_answer\_sets}(G, O, p_t, F)$
   define DLV program : $p_t$
   specify constraints $c$
   while $p_t \in P_t$
   extract answer sets $p_{ti} := dlv(G, O, \phi_{ci}(G), F)$
   end

3. $P_t := \text{find\_complementary\_answer\_sets}(G, O, p_t, F)$
   define DLV program : $p_t$
   specify constraints $c$
   while $p_t \notin P_t$
   extract complementary answer sets $p_{ti} := dlv(G, O, \phi_{ci}(G), F)$
   end

4. $P_t := \text{find\_abnormal\_instance}(G, O, p_t, F)$
   define DLV program : $p_t$
   specify constraints $c$
   while $p_t \in P_t$
   extract abnormal instance $p_{ti} := dlv(G, O, \phi_{ci}(G), F)$
   end

output $P_t$
end
Experiments on Movie database

- Movies Database available in UCI KDD archive — contains information about movies, actors, awards, directors, casts, locations, etc.
- Performed several experiments with Movies database to find useful patterns and instances.
- Finding instances:
  \[ \text{abnormal}(X) : \neg \text{remake of}(X, Y), \frac{\text{fraction copied}(X, Z)}{Z} > 95, Z <= 100. \]
  \[ \text{normal}(X) : \neg \text{remake of}(X, Y), \frac{\text{fraction copied}(X, Z)}{Z}, \text{not abnormal}(X). \]
- Finding patterns:
  \[ \text{married to}(X, Y), \text{acted in}(X, Z), \text{acted in}(Y, Z), \text{nominated}(Z, "Oscar"). \]
- Outperforms existing systems — results compared with supervised/unsupervised machine learning approaches.
Experiments on Movie database

- Movies Database available in UCI KDD archive — contains information about movies, actors, awards, directors, casts, locations, etc.

- Performed several experiments with Movies database to find useful patterns and instances.

  Finding instances:

  \begin{align*}
  \text{abnormal}(X) & : \neg \text{remake}\_\text{of}(X, Y), \text{fraction}\_\text{copied}(X, Z), Z > 95, Z \leq 100. \\
  \text{normal}(X) & : \neg \text{remake}\_\text{of}(X, Y), \text{fraction}\_\text{copied}(X, Z), \text{not abnormal}(X).
  \end{align*}

  Finding patterns:

  \begin{align*}
  \text{married}\_\text{to}(X, Y), \text{acted}\_\text{in}(X, Z), \text{acted}\_\text{in}(Y, Z), \text{nominated}(Z, "Oscar").
  \end{align*}

- Outperforms existing systems — results compared with supervised/unsupervised machine learning approaches.
Performance Analysis

Experiments on Movie database

- Movies Database available in UCI KDD archive — contains information about movies, actors, awards, directors, casts, locations, etc.

- Performed several experiments with Movies database to find useful patterns and instances.

- Finding instances:
  
  abnormal\( (X) : \neg \text{remake\_of}(X, Y), \text{fraction\_copied}(X, Z), Z > 95, Z \leq 100. \)
  
  normal\( (X) : \neg \text{remake\_of}(X, Y), \text{fraction\_copied}(X, Z), \text{not abnormal}(X). \)

- Finding patterns:
  
  married\_to\( (X, Y), \text{acted\_in}(X, Z), \text{acted\_in}(Y, Z), \text{nominated}(Z, "Oscar" ). \)

- Outperforms existing systems — results compared with supervised/unsupervised machine learning approaches.
Performance Analysis

Experiments on Movie database

- Movies Database available in UCI KDD archive — contains information about movies, actors, awards, directors, casts, locations, etc.

- Performed several experiments with Movies database to find useful patterns and instances.

- Finding instances:
  
  $$abnormal(X) : \neg \text{remake\_of}(X, Y), \text{fraction\_copied}(X, Z), Z > 95, Z \leq 100.$$  
  $$normal(X) : \neg \text{remake\_of}(X, Y), \text{fraction\_copied}(X, Z), \text{not abnormal}(X).$$

- Finding patterns:
  
  $$\text{married\_to}(X, Y), \text{acted\_in}(X, Z), \text{acted\_in}(Y, Z), \text{nominated}(Z, "Oscar").$$

- Outperforms existing systems — results compared with supervised/unsupervised machine learning approaches.
Performance Analysis

Experiments on Movie database

- Movies Database available in UCI KDD archive — contains information about movies, actors, awards, directors, casts, locations, etc.

- Performed several experiments with Movies database to find useful patterns and instances.

- Finding instances:

  abnormal\((X)\) : \(-\text{remake of}(X, Y), \text{fraction copied}(X, Z), Z > 95, Z \leq 100.\)

  normal\((X)\) : \(-\text{remake of}(X, Y), \text{fraction copied}(X, Z), \text{not abnormal}(X).\)

- Finding patterns:

  married_to\((X, Y), \text{acted in}(X, Z), \text{acted in}(Y, Z), \text{nominated}(Z, "Oscar")\).

- Outperforms existing systems — results compared with supervised/unsupervised machine learning approaches.
Why Description Logics?
(for semantic graph formalisms)

- Associated ontology hierarchy on semantic graph schema.
- Richer syntax and semantics of ontology tools.
- Semantic graph: very large ABox of assertions and small non-trivial TBox of terminologies.
- Several reasoning algorithms and optimization techniques.

Our work: How the existing DL reasoners - KAON2, RACER, and Pellet perform on large semantic graph ABoxes?
Why Description Logics? (for semantic graph formalisms)

- Associated ontology hierarchy on semantic graph schema.
- Richer syntax and semantics of ontology tools.
- Semantic graph: very large ABox of assertions and small non-trivial TBox of terminologies.
- Several reasoning algorithms and optimization techniques.

**Our work:** How the existing DL reasoners - KAON2, RACER, and Pellet perform on large semantic graph ABoxes?
Test Knowledge Bases

Movies Database (MOVIE)
- From UCI KDD archive
- Test queries:
  1. $M_1(x) \equiv person(x)$
  2. $M_2(x, y) \equiv person(x), award(y), won(x, y)$
  3. $M_3(x, y, z) \equiv movie(x), synonym_of(x, y), country(z), filmed_in(x, z)$

Univ-Bench Ontology (Univ)
- From Lehigh University LUBM benchmark database, describes universities, departments and the related activities.
- Test queries:
  1. $U_1(x) \equiv UndergraduateStudent(x)$
  2. $U_2(x, y) \equiv Chair(x), Department(y), worksFor(x, y),
     subOrganizationOf(y, “http://www.University2.edu”)$
  3. $U_3(x, y, z) \equiv Student(x), Course(y), Faculty(z), advisor(x, z),
     takesCourse(x, y), teacherOf(z, y)$
Test Knowledge Bases

Movies Database (MOVIE)
- From UCI KDD archive
- Test queries:
  1. \( M_1(x) \equiv person(x) \)
  2. \( M_2(x, y) \equiv person(x), award(y), won(x, y) \)
  3. \( M_3(x, y, z) \equiv movie(x), synonym_of(x, y), country(z), filmed_in(x, z) \)

Univ-Bench Ontology (Univ)
- From Lehigh University LUBM benchmark database, describes universities, departments and the related activities.
- Test queries:
  1. \( U_1(x) \equiv UndergraduateStudent(x) \)
  2. \( U_2(x, y) \equiv Chair(x), Department(y), worksFor(x, y), subOrganizationOf(y, "http://www.University2.edu") \)
  3. \( U_3(x, y, z) \equiv Student(x), Course(y), Faculty(z), advisor(x, z), takesCourse(x, y), teacherOf(z, y) \)
Test Data

<table>
<thead>
<tr>
<th>KB</th>
<th>$C \sqsubseteq D$</th>
<th>$C \equiv D$</th>
<th>domain</th>
<th>range</th>
<th>$R \sqsubseteq S$</th>
<th>$C(a)$</th>
<th>$R(a,b)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOVIE(100)</td>
<td>2</td>
<td>0</td>
<td>7</td>
<td>5</td>
<td>10</td>
<td>211</td>
<td>310</td>
</tr>
<tr>
<td>MOVIE(500)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>719</td>
<td>951</td>
</tr>
<tr>
<td>MOVIE(1000)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1500</td>
<td>1731</td>
</tr>
<tr>
<td>MOVIE(5000)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>8013</td>
<td>9191</td>
</tr>
<tr>
<td>MOVIE(11435)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>15017</td>
<td>18022</td>
</tr>
<tr>
<td>Univ(1,0)</td>
<td>36</td>
<td>6</td>
<td>25</td>
<td>18</td>
<td>9</td>
<td>18128</td>
<td>49336</td>
</tr>
<tr>
<td>Univ(2,0)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>30508</td>
<td>113463</td>
</tr>
<tr>
<td>Univ(3,0)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>44897</td>
<td>166682</td>
</tr>
<tr>
<td>Univ(4,0)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>53200</td>
<td>236514</td>
</tr>
<tr>
<td>Univ(5,0)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>65738</td>
<td>393227</td>
</tr>
</tbody>
</table>

Table: Test data statistics
Test Results (1/4)

(a) Movie query $M_1(x)$

(b) Univ query $U_1(x)$

(c) Movie query $M_2(x, y)$

(d) Univ query $U_2(x, y)$

(e) Movie query $M_3(x, y, z)$

(f) Univ query $U_3(x, y, z)$
Test Results (2/4)

Figure: Performance of DL reasoners over Movie and Univ queries

(g) KAON2
(h) RACER
(i) Pellet
Time required to answer a query for KAON2 grows moderately with data set size.

Query evaluation in RACER and Pellet requires much time, and sometimes, beyond computational criteria.

Problem:
- Excessive ABox consistency checking before answering first query.

Although impressive, KAON2:
- No minimum model semantics support.
- DLP transformation is non-disjunctive.
Test Results (3/4)

- Time required to answer a query for KAON2 grows moderately with data set size.

- Query evaluation in RACER and Pellet requires much time, and sometimes, beyond computational criteria.

  **Problem:**
  - Excessive ABox consistency checking before answering first query.

- Although impressive, KAON2:
  - No minimum model semantics support.
  - DLP transformation is non-disjunctive.
Test Results (3/4)

- Time required to answer a query for KAON2 grows moderately with dataset size.

- Query evaluation in RACER and Pellet requires much time, and sometimes, beyond computational criteria.

  **Problem:**
  - Excessive ABox consistency checking before answering first query.

- Although impressive, KAON2:
  - No minimum model semantics support.
  - DLP transformation is non-disjunctive.
How is DL reasoners for non-trivial TBox and small ABox?

- For example, *Wine* Ontology.
- Average-case performance of RACER and Pellet, to compute subsumption hierarchies.
- KAON2 lags far behind *tableau reasoners.*
How is DL reasoners for non-trivial TBox and small ABox?

- For example, Wine Ontology.
- Average-case performance of RACER and Pellet, to compute subsumption hierarchies.
- KAON2 lags far behind tableau reasoners.
How is DL reasoners for non-trivial TBox and small ABox?

- For example, Wine Ontology.
- Average-case performance of RACER and Pellet, to compute subsumption hierarchies.
- KAON2 lags far behind tableau reasoners.
Future Work

Further Research Directions

- Graphical user interface for pattern finding framework.
- Mechanism to deal with temporal information.
- Multiple ontology schemas — issues on ontology integration and aligning.
- DL reasoner that can work on very large semantic graphs with non-trivial TBoxes.
Future Work

Further Research Directions

- Graphical user interface for pattern finding framework.
- Mechanism to deal with temporal information.
- Multiple ontology schemas — issues on ontology integration and aligning.
- DL reasoner that can work on very large semantic graphs with non-trivial TBoxes.
Future Work

Further Research Directions

- Graphical user interface for pattern finding framework.
- Mechanism to deal with temporal information.
- Multiple ontology schemas — issues on ontology integration and aligning.
- DL reasoner that can work on very large semantic graphs with non-trivial TBoxes.
Future Work

Further Research Directions

- Graphical user interface for pattern finding framework.
- Mechanism to deal with temporal information.
- Multiple ontology schemas — issues on ontology integration and aligning.
- DL reasoner that can work on very large semantic graphs with non-trivial TBoxes.
In this talk, we talked about:

- Semantic graph formalisms - introductions and definitions
- Semantic graph analysis
  - From Disjunctive Logic Programming perspective
  - From Description Logics perspective
- Prospects
Acknowledgements

Thanks to:

- Thesis committee
- My thesis advisors
- Erasmus Mundus
- FUB & TUW
- Alcatel-Lucent Bell Labs