Semantic Web & Rule Inference

장민수 <minsu@etri.re.kr>
한국전자통신연구원 지식및추론연구팀
2006-10-13 제2회 시맨틱 비즈니스 워크샵

Contents

- What is rule?
- Ontology Reasoning with Rules
  - Background: Knowledge Representation, OWL Syntax & Semantics
  - Translation-based Approach
  - Meta-Reasoning Approach

Contents

- **What is rule?**
- Ontology Reasoning with Rules
    - Background: Knowledge Representation, OWL Syntax & Semantics
    - Translation-based Approach
    - Meta-Reasoning Approach

What is rule?

Propositional Calculus, Predicate Calculus, First-Order Logic,...

Automated Reasoning... complexity, decidability...

Roots of Rules

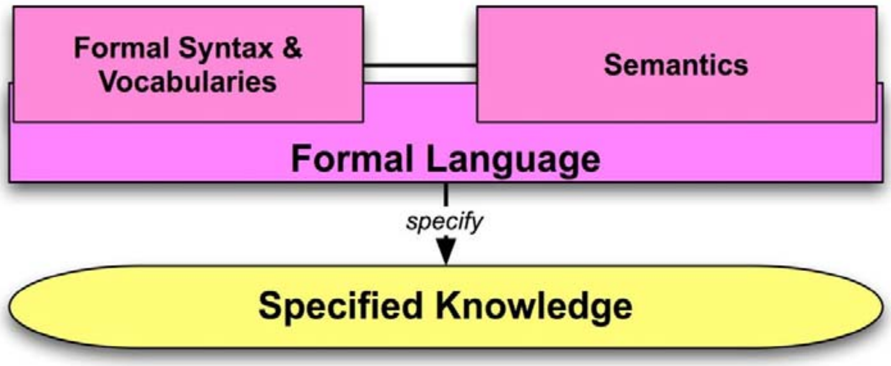
- **Horn Logic**: a small fragment of FOL. Became the base language of logic programming(LP).
- **Logic Programming**: includes extra-logical features e.g. Negation-As-Failure, Procedural Attachment, etc
- **Production Rule System**: roughly based on LP. Spawned the business rule systems market.
- **Deductive Database**: based on Datalog.

***Rule** mentions all the above formalisms with some features added/removed.*

Contents

- What is rule?
- **Ontology Reasoning with Rules**
  - **Background: Knowledge Representation, OWL Syntax & Semantics**
  - Translation-based Approach
  - Meta-Reasoning Approach

Specifying Knowledge (1)



Specifying Knowledge (2)

Formal Syntax:

```
Man(X) :- IsFatherOf(X,Y).
IsFatherOf(a,b).
```

Semantics

```
(X :- Y) ==> if Y holds, then X also holds.
```

What does the Semantics do? (1)

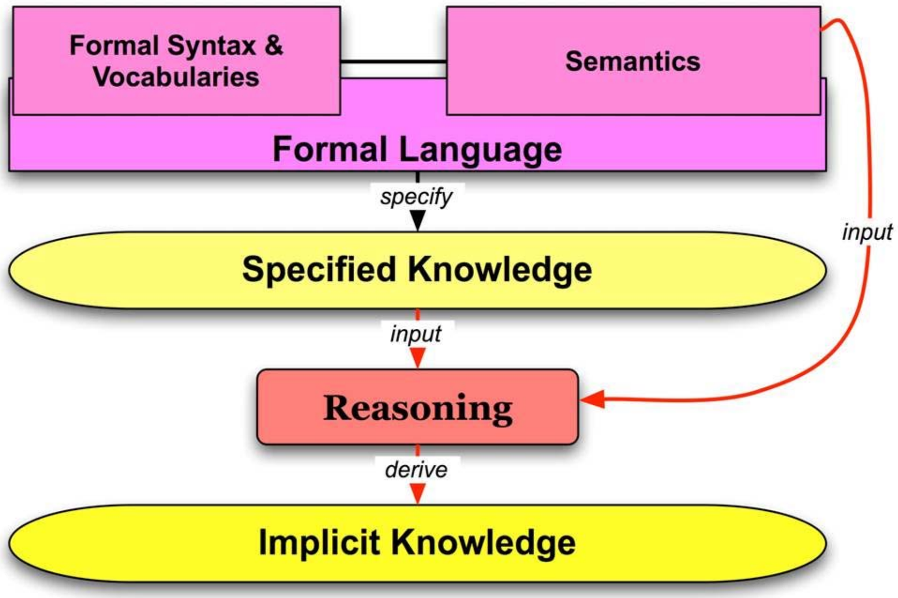
What is reasoning?

Finding the facts, which are not explicitly specified but implicitly true based on the specified knowledge.

How?

The semantics of the formal language guides the reasoning process!

What does the Semantics do? (2)



What does the Semantics do? (3)

Inference Rules guided by the Semantics

```
Y :- X ===>
{(X -> Y) AND X} -> Y  [modus ponens]
```

Apply the rule!

```
Man(X) :- IsFatherOf(X,Y).
          IsFatherOf(a,b).
---------------------------
          Man(a)
```

OWL follows the same rules...

OWL Syntax

```
Class( C partial D )
EquivalentClasses(C D)
...
```

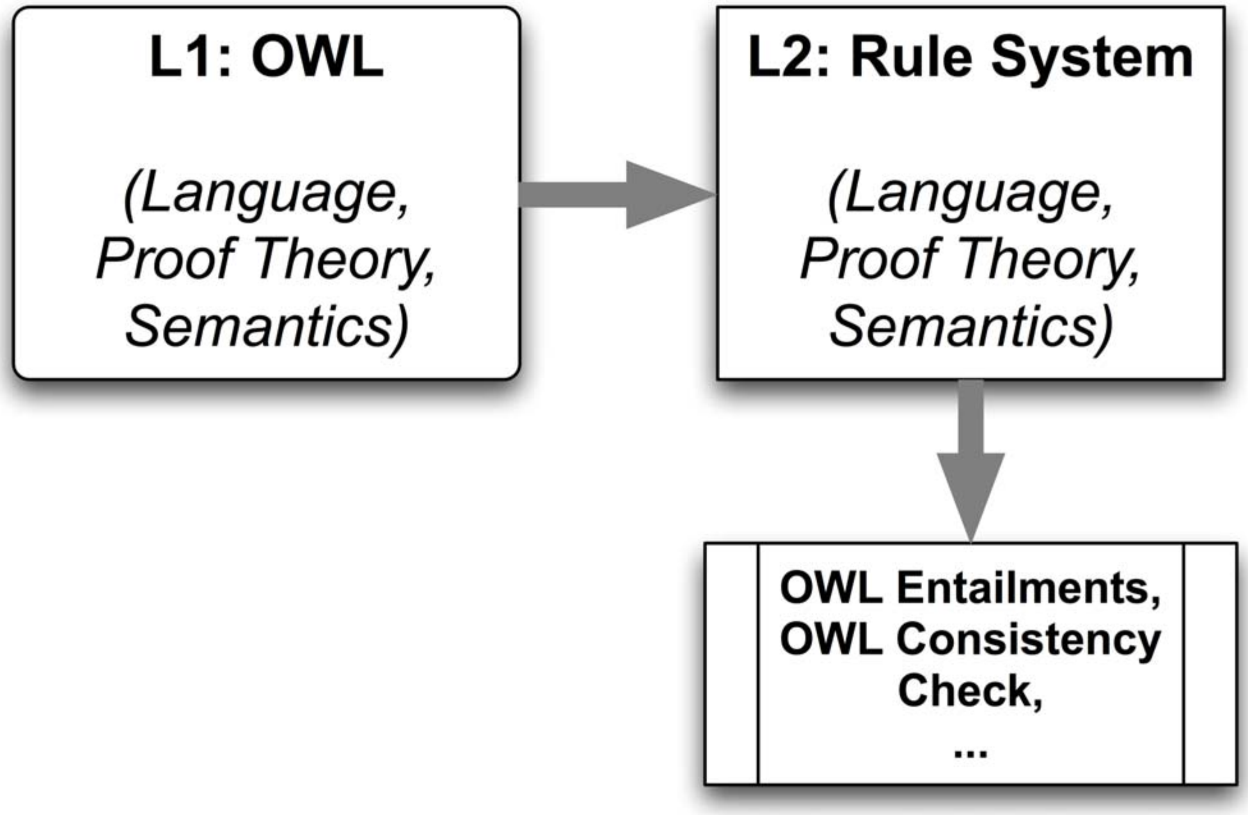
OWL Semantics

```
Class(C partial D) ==> EC(C) ⊆ EC(D)
EquivalentClasses(C D) ==> EC(C) ≡ EC(D)
```

We can make up inference rules based on the semantics.

```
EC(C) ⊆ EC(D) ==> D(?x) :- C(?x)
EC(C) ≡ EC(D) ==> D(?x) :- C(?x), C(?x) :- D(?x)
```

OWL Reasoning using Rules: Settings



- **Two Approaches**
  - Translation-based Approach
  - Meta-Reasoning Approach

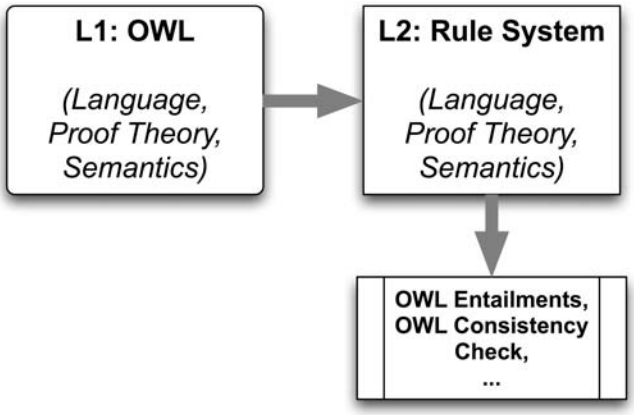
Contents

- What is rule?
- **Ontology Reasoning with Rules**
  - Background: Knowledge Representation, OWL Syntax & Semantics
  - **Translation-based Approach**
  - Meta-Reasoning Approach

Translation-based Approach (1)

Reasoning L1 using L2

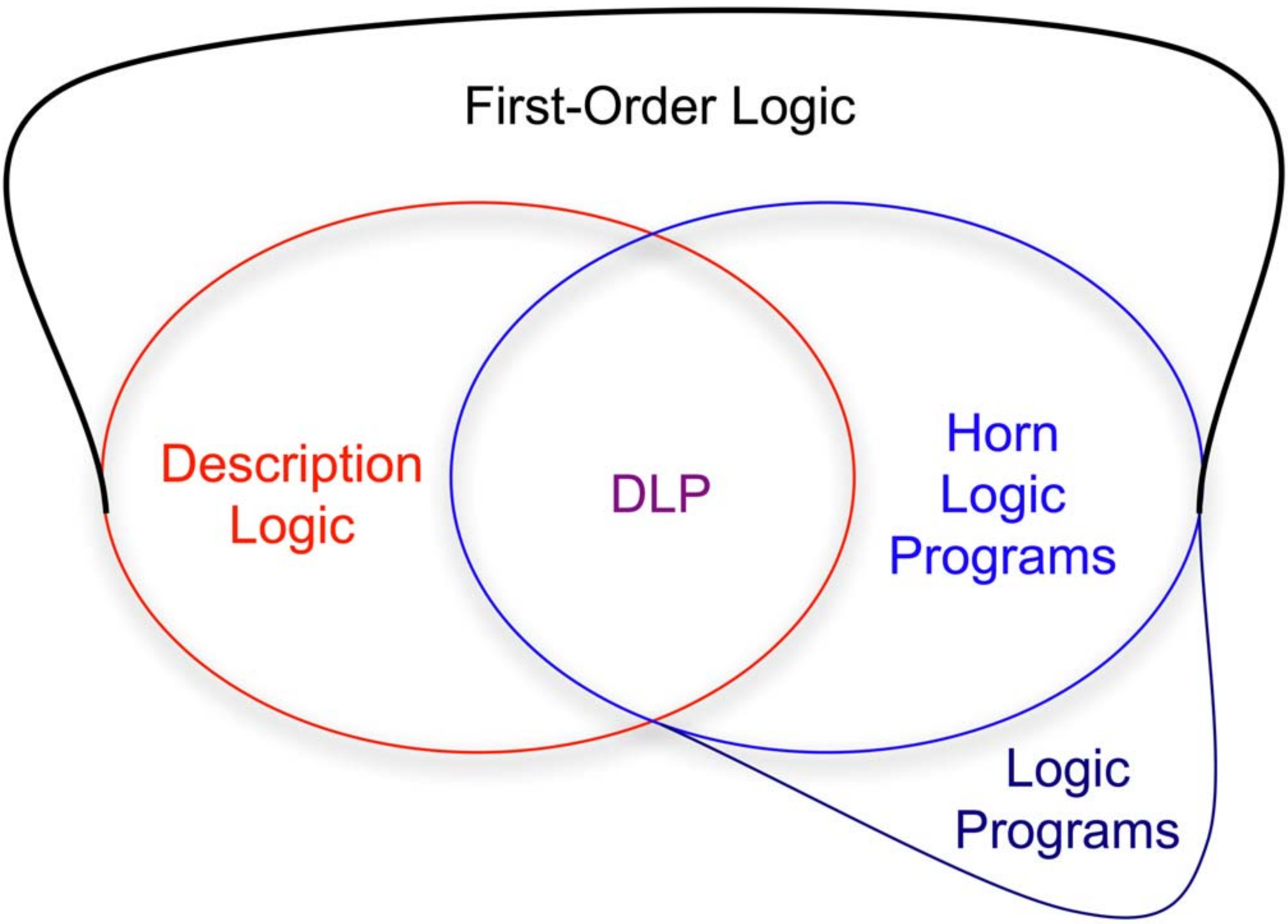
- Find the common expressiveness fragment L3 of L1 and L2
- Establish a mapping from L1 to L3
- Translate L1 declarations into L3 according to the mapping.
- Use L2-compatible system to perform inferencing over L3-fragment of L1.



Translation-based Approach (2)

Description Logic Program's Case

The common fragment of DL and Logic Program is identified.



adapted from [Grosof03]

Translation-based Approach (3)

Description Logic Program's Case

Establish a mapping from DL into DLP

$$
\begin{array}{lcl}
\mathcal{T}(C \sqsubseteq D) & \longrightarrow & Th(D,y) \;\leftarrow\; Tb(C,y) \\
Th(A,x) & \longrightarrow & A(x) \\
Th((C \sqcap D),x) & \longrightarrow & Th(C,x) \wedge Th(D,x) \\
Th((\forall R.C),x) & \longrightarrow & Th(C,y) \;\leftarrow\; R(x,y) \\
Tb(A,x) & \longrightarrow & A(x) \\
Tb((C \sqcap D),x) & \longrightarrow & Tb(C,x) \wedge Tb(D,x) \\
Tb((C \sqcup D),x) & \longrightarrow & Tb(C,x) \vee Tb(D,x) \\
Tb((\exists R.C),x) & \longrightarrow & R(x,y) \wedge Tb(C,y)
\end{array}
$$

$$
\begin{array}{lcl}
\mathcal{T}(P \sqsubseteq Q) & \longrightarrow & Q(x,y) \;\leftarrow\; P(x,y) \\
\mathcal{T}(P \equiv Q) & \longrightarrow & \left\{ \begin{array}{lcl} Q(x,y) & \leftarrow & P(x,y) \\ P(x,y) & \leftarrow & Q(x,y) \end{array} \right. \\
\mathcal{T}(P \equiv Q^{-}) & \longrightarrow & \left\{ \begin{array}{lcl} Q(x,y) & \leftarrow & P(y,x) \\ P(y,x) & \leftarrow & Q(x,y) \end{array} \right. \\
\mathcal{T}(P^{+} \sqsubseteq P) & \longrightarrow & P(x,z) \;\leftarrow\; P(x,y) \wedge P(y,z)
\end{array}
$$

excerpted from [Grosof03]

Translation-based Approach (4)

Description Logic Program's Case

Translations(1): An OWL Ontology

```
:human rdfs:subClassOf :mortal.
:person a owl:Class.
:human owl:equivalentClass :person.
:is-author a owl:ObjectProperty;
           rdfs:domain :person;
           rdfs:range :document.
:Socrates a :person;
          :is-author :Politeia.
```

Translation-based Approach (5)

Description Logic Program's Case

Translations(2): The Translated DLP Program

```
person(Socrates).
person(X) :- human(X).
mortal(X) :- human(X).
human(X) :- person(X).
is_author(socrates, politeia).
person(X) :- is_author(X, Y).
document(Y) :- is_author(X, Y).
```

Now, it's possible to perform reasoning with LP reasoners e.g. prolog or datalog engines!! We get the following conclusions!

```
human(Socrates).
mortal(Socrates).
```

Translation-based Approach (6)

Issues!

Expressiveness Power Discrepancy!

In DLP's case...

- OWL's classical negation cannot be expressed in DLP.
- Cardinality restrictions cannot be expressed in DLP.
- ...

Expressiveness Loss is Inevitable!

Translation-based Approach (7)

Issues! - Semantics Discrepancy!

In DLP's case...

OWL Semantics is based on Model Theory of *classical logic*.

LP semantics is defined in terms of minimal Herbrand models, i.e. sets of ground facts.

Given:

```
portableLiquid(X) :- wine(X).
wine(X) :- whiteWine(X).
whiteWine("Welschriesling").
```

LP interpretation always produces *Gound Entailments*:

```
wine("Welschriesling").
portableLiquid("Welschriesling").
```

Classical logic aka FOL produces *Non-Ground Entailments*.

```
(∀X)whiteWine(X) ⊃ PortableLiquid(X).
```

Again, LP semantic cannot produce *Non-Ground Entailments*.

adapted from [Eiter06]

Translation-based Approach (8)

Conclusion

There's no rule language that has the expressiveness power of OWL DL.

(What about FOL reasoners? Well...)

Expressiveness Loss & Semantic Discrepancy are Inevitable!!

Therefore;

- By well-designing the mapping, we can do **sound** reasoning.
- But, we **cannot** do complete reasoning!! (*due to the expressiveness loss & semantic discrepancy*)

- Tools: KAON2(based on disjunctive datalog), Hoolet(based on FOL), OWLIM(based on DLP)

Contents

- What is rule?

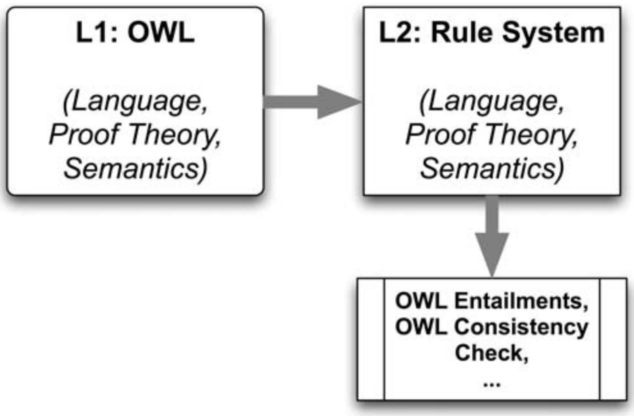
- Ontology Reasoning with Rules
  - Background: Knowledge Representation, OWL Syntax & Semantics
  - Translation-based Approach
  - **Meta-Reasoning Approach**

Meta-Reasoning Approach (1)

Reasoning L1 using L2

- Establish a mapping from L1 to the facts of L3.
- Build up L3 rules for covering L1 inference over factual representation of L1.
- Translate L1 declarations into L3 facts according to the mapping.
- Use L2-compatible reasoner to perform L1 inferencing.

A way of implementing L1 semantics on top of L2; L1 inference rules are implemented using L2.



Meta-Reasoning Approach (2)

Bossam's case

OWL sentences are mapped to Bossam facts.

- OWL sentences are mapped to RDF triple.
- RDF triples are classified into two kinds: Type Declaration & Others
  - Type Declarations:

    ```
    b:Person rdf:type owl:Class.
    a:Cheolsu rdf:type b:Person.
    ```

  - Others:

    ```
    b:Person rdfs:subClassOf a:Animal.
    ```

- Type declarations are translated into 1-ary Facts:

```
b:Person rdf:type owl:Class. ==> owl:Class(b:Person)
```

- Others are translated into 2-ary Facts:

```
b:Person rdfs:subClassOf a:Animal.
==> rdfs:subClassOf(b:Person,a:Animal)
```

Meta-Reasoning Approach (3)

Bossam's case

OWL semantics are specified using Bossam rules

- ex1) rdfs:subClass
  - Transitivity

    ```
    if rdfs:subClassOf(?x,?y) and rdfs:subClassOf(?y,?z)
    then rdfs:subClassOf(?x,?z);
    ```

  - Type Propagation

    ```
    if rdfs:subClassOf(?x,?y) and ?x(?i) then ?y(?i);
    ```

Meta-Reasoning Approach (4)

Bossam's case

OWL semantics are specified using Bossam rules.

- ex2) owl:allValuesFrom
  - Type Reasoning, *Obvious!!*

```
if owl:onProperty(?r,?p) and
   owl:allValuesFrom(?r,?a) and
   ?p(?x,?y) and
   ?r(?x)
then ?y(?a);
```

  - *What about this?* Identifying similar restrictions...

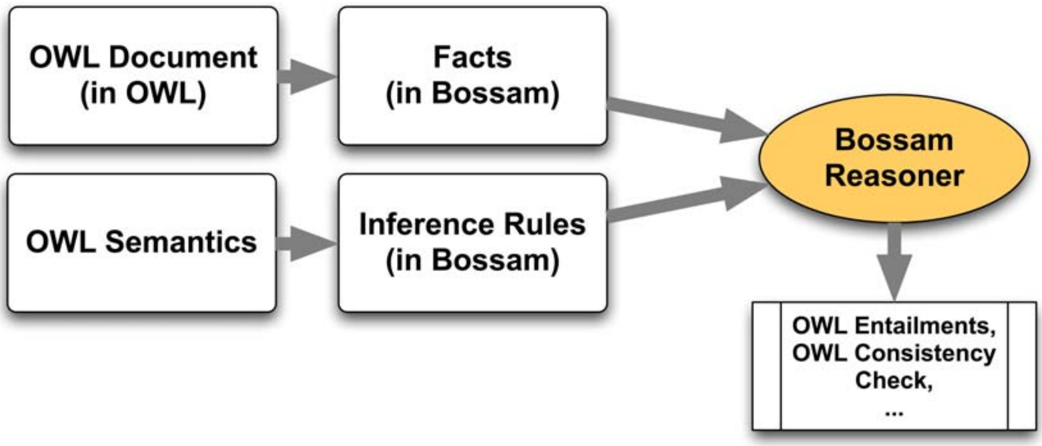
```
if owl:onProperty(?r,?p) and
   owl:onProperty(?s,?p) and
   owl:allValuesFrom(?r,?a) and
   owl:allValuesFrom(?s,?a)
then rdfs:subClassOf(?r,?s);
```

It's almost impossible to enumerate all the cases to cover with rules!!

Meta-Reasoning Approach (5)

Bossam's case

KB = OWL Inference Rules + Factual Representations of OWL Documents



Meta-Reasoning Approach (6)

Issues! - Expressiveness Discrepancy!

It's very difficult or impossible to specify Model Theoretic entailments of OWL Lite/DL!!

```
Person ≡ ∃ parent.Person.
Person(Fred).
```

entails:

```
<some-A,some-B> : parent.
<Fred,some-A> : parent.
```

A kind of non-ground implication. We need to insert an existential variable into the head of a rule, which makes the rule unsafe.

Impossible to implement in most rule systems!

Meta-Reasoning Approach (7)

Issues! - Expressiveness Discrepancy!

Even simple tautologies are not easy to implement. De Morgan's Law!

```
Class(A).
Class(B).
```

entails:

```
owl:intersectionOf(owl:complementOf(A),owl:complementOf(B))
≡
owl:complementOf(owl:unionOf(A,B))
```

Too many conclusions!! What about N classes?

Impossible to implement in most rule systems!

Meta-Reasoning Approach (8)

Issues! - More problems...

- OWA vs CWA
- Classical Negation vs Negation-As-Failure
- FOL-based Interpretation vs LP-based Interpretation
- Bad performance for large ontologies
- Incomplete inference rules
- ...

Therefore;

- By well-designing the inference rules, we can do **sound** reasoning.
- But, we **cannot** do complete reasoning!!

Meta-Reasoning Approach (9)

Then, what do we buy?

- Successful reasoners exist! Euler, CWM, OWLJessKB, Jena, and Bossam(...?),...
- We can utilize full power of rule systems: rules, naf, procedural attachments, builtins etc
- Easy to integrate OWL, Rules, and External Systems.

They are practical!

References

- [Grosof03] BN Grosof, I Horrocks, R Volz, "Description logic programs: combining logic programs with description logic," Proceedings of the twelfth international conference on World Wide Web, 2003.
- [Eiter06] T. Eiter, G. Ianni, A. Polleres, "Reasoning with Rules and Ontologies," 2006.
- EulerSharp Proof Mechanism (http://eulersharp.sourceforge.net/)
- CWN (http://www.w3.org/2000/10/swap/doc/cwm.html)
- Bossam Rule/OWL Reasoner (http://mknows.etri.re.kr/bossam)

감사합니다!!