

# Rules with Contextually Scoped Negation for the Web

Axel Polleres

Universidad Rey Juan Carlos, Madrid  
axel@polleres.net

March 6, 2006

# Overview

- ▶ The Semantic Web
- ▶ Where to add Rules in the “Layer Cake”?
- ▶ A lightweight approach: Logic Programs with Context and Scoped Negation
  - ▶ Contextually Bounded Semantics
  - ▶ Contextually Closed Semantics
  - ▶ Summary/Open Issues
- ▶ Other approaches . . . time allowed.
  - ▶ SWRL – Rules on top of OWL
  - ▶ DLP – Intersection of LP and DL
  - ▶ dl-programs – a query interface between LP and OWL

# Motivation - Semantic Web



Plan 9 from Outer Space (1959)

Directed by [Edward D. Wood Jr.](#)

Wrote credits [Edward D. Wood Jr.](#)

Shop **PLAN 9 FROM OUTER SPACE**

Available on Amazon.com

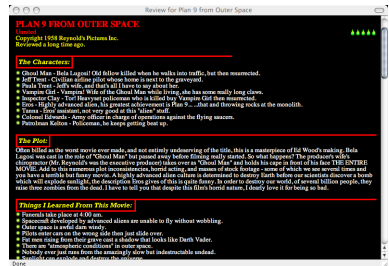
Genre: [Horror](#) / [Sci-Fi](#) ([more](#))

Tags: [Unspookable Horror From Outer Space Paralyze The Living And Resurrect The Dead!](#) ([more](#))

Plot Outline: Aliens resurrect dead humans as zombies and vampires to stop human kind from creating the Solaranic (a sort of sun-driven bomb). ([more](#))

User Comments: A classic of the bad sci-fi genre. ([more](#))

User Rating: 3.4/10 (8,935 votes)



PLAN 9 FROM OUTER SPACE

Copyright 1958 Reynolds Pictures Inc.  
Reviewed a long time ago.

**The Characters:**

- Ghoul Man - Bela Lugosi! Old fellow killed when he walks into traffic, but then resurrected.
- Jeff Tami - Civilian airline pilot whose home is next to the graveyard.
- Paula Tami - Jeff's wife, and that's it! There is no story line.
- Vampira Girl - Vampira! Wife of the Ghoul Man while living, she has some really long claws.
- Inspector Clay - The Heavyest policeman who is killed by Vampira Girl then resurrected.
- Eric - Highly advanced alien, his greatest achievement is Plan 9...that and throwing rocks at the moonlight.
- Tanna - Eric's assistant, not very good at this "alien" stuff.
- Colonel Edwards - Army officer in charge of operations against the flying saucers.
- Patrolman Keltus - Policeman, he keeps getting beat up.

**The Plot:**

Often billed as the worst movie ever made, and not entirely undeserving of the title, this is a masterpiece of Ed Wood's making. Bela Lugosi was cast in the role of "Ghoul Man" but passed away before filming really started. So what happens? The producer's wife's (daughter?) Mr. Reynolds's was the executive producer takes over as "Ghoul Man" and holds his eye in front of his face THE ENTIRE MOVIE. Add to this numerous plot inconsistencies, horrid acting, and masses of stock footage - some of which we see several times and you have a terrible but funny movie. A highly advanced alien culture is determined to destroy Earth before our scientists discover a bomb which will explode tonight, the description Eric gives of this is quite funny. In order to destroy our world, of several billion people, they raise three zombies from the dead. I have to tell you that despite this film's horrid nature, I don't love it for being so bad.

**Things I Learned From This Movie:**

- Parents take place at 4:00 am.
- Spacecraft designed by advanced aliens are unable to fly without wobbling.
- Outer space is awful damn windy.
- Pointy outer ears on the wrong side than just side only.
- The man sitting from their grave cast a shadow that looks like Darth Vader.
- There are "atmospheric conditions" in outer space.
- Nobody ever just runs from the eternally slow but indestructible undead.
- [Click here to archive and destroy the internet.](#)

<http://imdb.com>

<http://badmovies.org>

- ▶ The Semantic Web promises machine readable **metadata** annotations of such sites allowing to combine and query their content, draw additional inferences.

# Motivation - Semantic Web



Plan 9 from Outer Space (1959)

Directed by [Edward D. Wood Jr.](#)

Writing credits [Edward D. Wood Jr.](#)

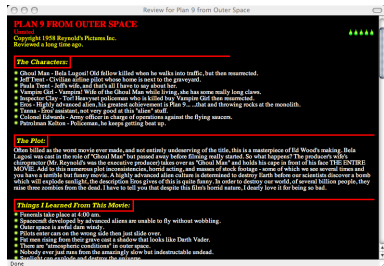
Genre: [Horror](#) / [Sci-Fi](#) ([more](#))

Tags: [Unspookable Horror From Outer Space Paralyze The Living And Resurrect The Dead!](#) ([more](#))

Plot Outline: Aliens resurrect dead humans as zombies and vampires to stop human kind from creating the Solaranic (a sort of sun-driven bomb). ([more](#))

User Comments: A classic of the bad sci-fi genre. ([more](#))

User Rating: 3.4/10 (8,935 votes)



PLAN 9 FROM OUTER SPACE

Copyright 1958 Reynolds Pictures Inc.  
Reviewed a long time ago.

**The Characters:**

- Ghoul Man - Bela Lugosi! Old fellow killed when he walks into traffic, but then resurrected.
- Jeff Tami - Civilian airline pilot whose home is next to the graveyard.
- Paula Trent - Jeff's wife, and that's all I have on my Ghoul list.
- Vampire Girl - Vampire! Wife of the Ghoul Man while living, she has some really long claws.
- Inspector Clay - The Heavyest policeman who is killed by Vampire Girl then resurrected.
- Eric - Highly advanced alien, his greatest achievement is Plan 9 - just not throwing rocks at the moonlit.
- Dana - Eric's assistant, not very good at this "alien" stuff.
- Colonel Edwards - Army officer in charge of operations against the flying saucers.
- Patrolman Keltus - Policeman, he keeps getting beat up.

**The Plot:**

Often billed as the worst movie ever made, and not entirely undeserving of the title, this is a masterpiece of Ed Wood's making. Bela Lugosi was cast in the role of "Ghoul Man" but passed away before filming really started. So what happens? The producer's wife's disapproval (Mc Reynolds's was the executive producer) takes over as "Ghoul Man" and holds his eye in front of his face THE ENTIRE MOVIE. Add to this numerous plot inconsistencies, horrid acting, and masses of stock footage - some of which we see several times and you have a smelly but funny movie. A highly advanced alien culture is determined to destroy Earth before our scientists discover a bomb which will explode tonight, the description Eric gives of this is quite funny. In order to deny our world, of several billion people, they raise three zombies from the dead. I have to tell you that despite this film's horrid nature, I don't love it for being so bad.

**Things I Learned From This Movie:**

- Parents take place at 4:00 am.
- Spacecraft developed by advanced aliens are unable to fly without wobbling.
- Outer space is awfully damn windy.
- Points enter cars on the wrong side than just side over.
- Put one string from their garter used a shadow that looks like Darth Vader.
- There are "atmospheric conditions" in outer space.
- Nobody ever just runs from the eternally slow but indestructible undead.
- [Click here to watch and destroy the release.](#)

<http://imdb.com>

<http://badmovies.org>

- ▶ The Semantic Web promises machine readable **metadata** annotations of such sites allowing to combine and query their content, draw additional inferences.
- ▶ E.g., imagine a "Semantic" search engine gathering metadata on movies and ratings, using an agreed vocabulary, I want to ask **queries**, such as: *"Search for science fiction movies which are rated as bad?"*

# Motivation - Semantic Web



Plan 9 from Outer Space (1959)

Directed by [Edward D. Wood Jr.](#)

Writing credits [Edward D. Wood Jr.](#)

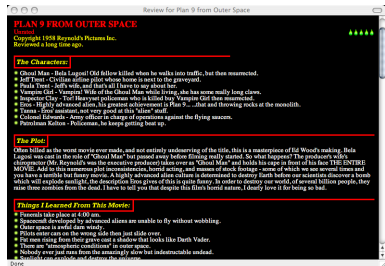
Genre: [Horror](#) / [Sci-Fi](#) ([more](#))

Tags: [Unspookable Horror From Outer Space Paralyze The Living And Resurrect The Dead!](#) ([more](#))

Plot Outline: [Aliens resurrect dead humans as zombies and vampires to stop human kind from creating the Solaranic \(a sort of sun-driven bomb\).](#) ([more](#))

User Comments: [A classic of the bad sci-fi genre.](#) ([more](#))

User Rating: 3.4/10 (8,935 votes)



PLAN 9 FROM OUTER SPACE

Copyright 1958 Reynolds Pictures Inc.  
Reviewed a long time ago.

**The Characters:**

- Ghoul Man - Bela Lugosi! Old fellow killed when he walks into traffic, but then resurrected.
- Jeff Tarr - Civilian airline pilot whose home is next to the graveyard.
- Paula Trent - Jeff's wife, and that's all there is on that line.
- Vampire Girl - Vampires! Wife of the Ghoul Man while living, she has some really long claws.
- Inspector Clay - The Heavyset policeman who is killed by Vampires! Ghoul Man resurrects.
- Eric - Highly advanced alien, his greatest achievement is Plan 9...just not throwing rocks at the moonlit.
- Dana - Eric's assistant, not very good at this "alien" stuff.
- Colonel Edwards - Army officer in charge of operations against the flying saucers.
- Patrolman Keltus - Policeman, he keeps getting beat up.

**The Plot:**

Often billed as the worst movie ever made, and not entirely undeserving of the title, this is a masterpiece of Ed Wood's making. Bela Lugosi was cast in the role of "Ghoul Man" but passed away before filming really started. So what happens? The producer's wife's chaperone (Mc Reynolds) had the executive producer taken over as "Ghoul Man" and holds his eye in front of his face THE ENTIRE MOVIE. Add to this numerous plot inconsistencies, horrid acting, and masses of stock footage - some of which we saw several times and you have a mumble but funny movie. A highly advanced alien culture is determined to destroy Earth before our scientists discover a bomb which will explode tonight. The description Eric gives of this is quite funny. In order to destroy our world, of several billion people, they raise three zombies from the dead. I have to tell you that despite this film's horrid nature, I don't love it for being so bad.

**Things I Learned From This Movie:**

- Parents take place at 4:00 am.
- Spacecraft developed by advanced alien are unable to fly without wobbling.
- Outer space is awfully damn windy.
- Points enter cars on the wrong side than just side over.
- The man sitting from their grave cast a shadow that looks like Darth Vader.
- There are "atmospheric conditions" in outer space.
- Nobody ever just runs from the screamingly slow but indestructible undead.
- [Click here to watch and destroy the universe.](#)

<http://imdb.com>

<http://badmovies.org>

- ▶ The Semantic Web promises machine readable **metadata** annotations of such sites allowing to combine and query their content, draw additional inferences.
- ▶ E.g., imagine a "Semantic" search engine gathering metadata on movies and ratings, using an agreed vocabulary, I want to ask **queries**, such as: *"Search for science fiction movies which are rated as bad?"*
- ▶ I want to express **taxonomies** such as *"Science-fiction movies are movies."*

# Motivation - Semantic Web



Plan 9 from Outer Space (1959)

Directed by [Edward D. Wood Jr.](#)

Writing credits [Edward D. Wood Jr.](#)

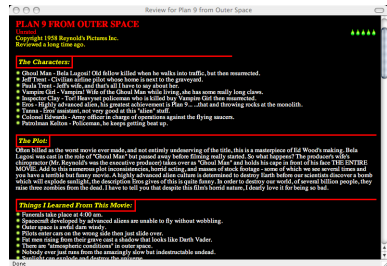
Genre: [Horror](#) / [Sci-Fi](#) ([more](#))

Tagline: Unspookable Horror From Outer Space Paralyze The Living-And Resurrect The Dead! ([more](#))

Plot Outline: Aliens resurrect dead humans as zombies and vampires to stop human kind from creating the Solarium (a sort of sun-driven bomb). ([more](#))

User Comments: A classic of the bad sci-fi genre. ([more](#))

User Rating: 3.4/10 (8,935 votes)



PLAN 9 FROM OUTER SPACE

Copyright 1958 Reynolds Pictures Inc.  
Reviewed a long time ago.

**The Characters:**

- Ghoul Man - Bela Lugosi! Old fellow killed when he walks into traffic, but then resurrected.
- Jeff Tont - Civilian airline pilot whose home is next to the graveyard.
- Paula Trent - Jeff's wife, and that's all there is on that list.
- Vampire Girl - Vampire! Wife of the Ghoul Man while living, she has some really long claws.
- Inspector Clay - Top Heavyweight policeman who is killed by Vampires! Girl then resurrects.
- Eric - Highly advanced alien, his greatest achievement is Plan 9...just not throwing rocks at the moonlight.
- Diana - Eric's assistant, not very good at this "alien" stuff.
- Colonel Edwards - Army officer in charge of operations against the flying saucers.
- Patrolman Keltus - Policeman, he keeps getting beat up.

**The Plot:**

Often billed as the worst movie ever made, and not entirely undeserving of the title, this is a masterpiece of Ed Wood's making. Bela Lugosi was cast in the role of "Ghoul Man" but passed away before filming really started. So what happens? The producer's wife's disapproval (Mc Reynolds had the executive producer taken over as "Ghoul Man" and holds his ego in front of his face) THE ENTIRE MOVIE. Add to this numerous plot inconsistencies, horrid acting, and masses of stock footage - some of which we see several times and you have a mumble but funny movie. A highly advanced alien culture is determined to destroy Earth before our scientists discover a bomb which will explode tonight. The description Eric gives of this is quite funny. In order to destroy our world, of several billion people, they raise three zombies from the dead. I have to tell you that despite this film's horrid nature, I don't love it for being so bad.

**Things I Learned From This Movie:**

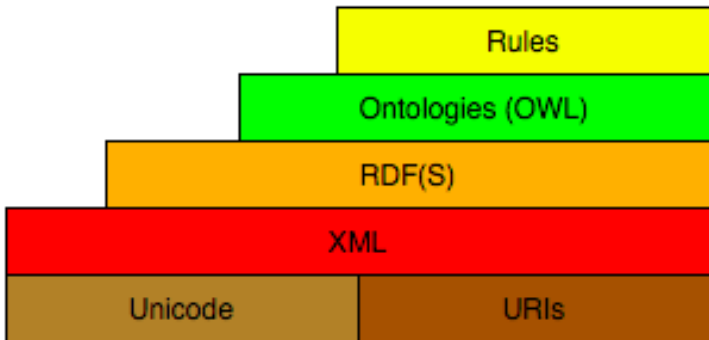
- Parents take place at 4:00 am.
- Specialized developed by advanced alien are enable to fly without wobbling.
- Outer space is awfully damn windy.
- Moon enter came in the wrong side than just side one.
- The man sitting from their grave cast a shadow that looks like Darth Vader.
- There are "atmospheric conditions" in outer space.
- Nobody ever just runs from the monstrously slow but indestructible undead.
- Ghoulies can scratch and destroy the release.

<http://imdb.com>

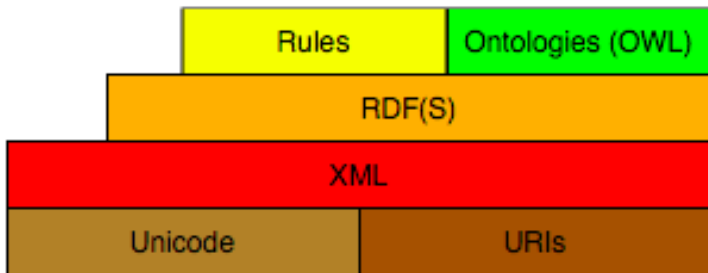
<http://badmovies.org>

- ▶ The Semantic Web promises machine readable **metadata** annotations of such sites allowing to combine and query their content, draw additional inferences.
- ▶ E.g., imagine a "Semantic" search engine gathering metadata on movies and ratings, using an agreed vocabulary, I want to ask **queries**, such as: "Search for science fiction movies which are rated as bad?"
- ▶ I want to express **taxonomies** such as "Science-fiction movies are movies."
- ▶ Besides **facts** in RDF, I want to express more complex **rules** such as for instance: "All movies listed on badmovies.org are rated bad."

## The W3C's Semantic Web “layer cake”



## The W3C's Semantic Web “layer cake”

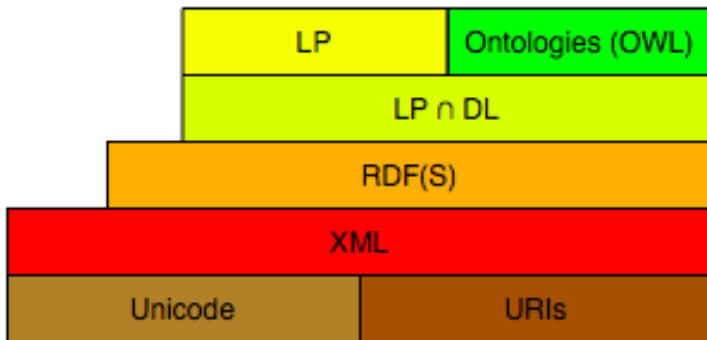


Can LP style rules really be layered ON TOP of OWL?

I. Horrocks , B. Parsia , P. Patel-Schneider , J. Hendler. *Semantic Web Architecture: Stack or Two Towers?* PPSWR, 2005.



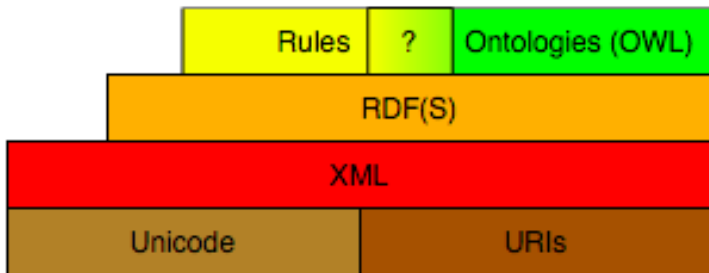
## The W3C's Semantic Web “layer cake”



What is the common interoperability layer?

B. Grosz, I. Horrocks, R. Volz, S. Decker. *Description Logic Programs: Combining Logic Programs with Description Logic*. WWW, 2003.

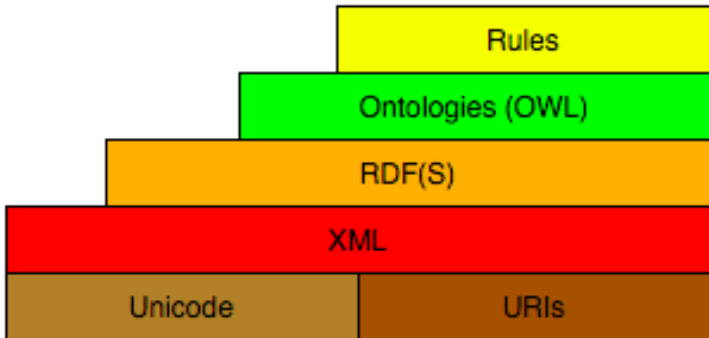
## The W3C's Semantic Web “layer cake”



Can we define a “safe” interface between LP and OWL?

T. Eiter, T. Lukasiewicz, R. Schindlauer, H. Tompits *Combining Answer Set Programming with Description Logics for the Semantic Web*. KR, 2004.

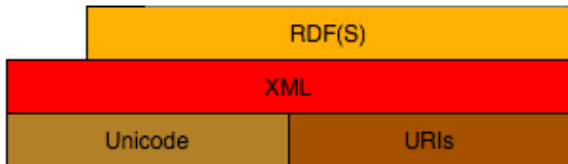
## The W3C's Semantic Web “layer cake”



What is the “right” way to go?

## RDF - A standard for metadata

Let's start at the level where concerns are still (more or less) clear:



- ▶ RDF allows to define *factual* metadata in about resources in form of triples

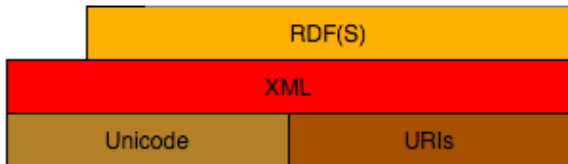
*⟨Subject, Predicate, Object⟩*

e.g. **StarWars** is directed by **Goerge Lucas**.

- ▶ Resources identified by URIs

## RDF - A standard for metadata

Let's start at the level where concerns are still (more or less) clear:



- ▶ RDF allows to define *factual* metadata in about **r**esources in form of triples

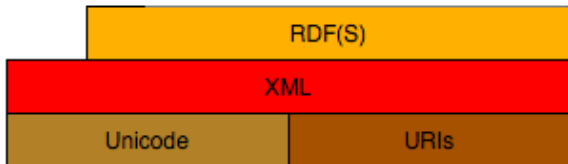
*<Subject, Predicate, Object>*

e.g. **S**tarWars **i**s directed by **G**oerge **L**ucas.

- ▶ Resources identified by URIs
- ▶ RDFS allows to define simple taxonomies on RDF vocabularies using `rdf:type`, `rdf:subClassOf`

## RDF - A standard for metadata

Let's start at the level where concerns are still (more or less) clear:



- ▶ RDF allows to define *factual* metadata in about resources in form of triples

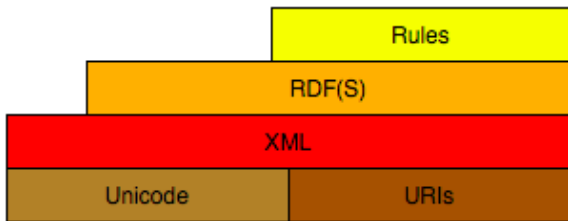
*⟨Subject, Predicate, Object⟩*

e.g. *StarWars* is directed by Goerge Lucas.

- ▶ Resources identified by URIs
- ▶ RDFS allows to define simple taxonomies on RDF vocabularies using `rdf:type`, `rdf:subClassOf`
- ▶ Some subtleties in RDF semantics (blank nodes, XML literals, RDF keywords treated as normal resources, reification, etc.)

## RDF - A standard for metadata

Let's start at the level where concerns are still (more or less) clear:



- ▶ RDF allows to define *factual* metadata in about resources in form of triples

*<Subject, Predicate, Object>*

e.g. *StarWars* is directed by Goerge Lucas.

- ▶ Resources identified by URIs
- ▶ RDFS allows to define simple taxonomies on RDF vocabularies using `rdf:type`, `rdf:subClassOf`
- ▶ Some subtleties in RDF semantics (blank nodes, XML literals, RDF keywords treated as normal resources, reification, etc.)

- ▶ The presented approach discuss rules on top of RDF(S) only.

## Metadata on the Web as RDF facts.

<pre>http://moviereviews.com/ triple(ex:m1,ex:rate,ex:bad).</pre>	<pre>http://polleres.net/myreviews triple(ex:m2,ex:rate,ex:bad). triple(ex:m2,rdf:type,movie).</pre>
<pre>http://imdb.com/ triple(ex:m1,rdf:type,ex:sciFiMovie). triple(ex:m1,ex:title,"Plan 9 from Outer Space"). triple(ex:m1,ex:directedBy,"Ed Wood"). triple(ex:m2,rdf:type,ex:sciFiMovie). triple(ex:m2,ex:title,"Matrix Revolutions"). triple(ex:m2,ex:directedBy,"Andy Wachowski"). triple(ex:m2,ex:directedBy,"Larry Wachowski"). triple(ex:m3,rdf:type,ex:sciFiMovie). triple(ex:m3,ex:title,"Bride of the Monster"). triple(ex:m3,ex:directedBy,"Ed Wood"). triple(ex:sciFiMovie,rdf:subClassOf,ex:movie). ...</pre>	

Figure: RDF triples for some movie information sites



## RDFS semantics

RDFS semantics can (to a large extent) be captured by LP style rules:

```
http://www.example.org/rdfs-semantics :  
triple(P,rdf:type,rdf:Property) :- triple(S,P,O).  
triple(S,rdf:type,rdfs:Resource) :- triple(S,P,O).  
triple(O,rdf:type,rdfs:Resource) :- triple(S,P,O).  
triple(S,rdf:type,C) :- triple(S,P,O), triple(P,rdfs:domain,C).  
triple(O,rdf:type,C) :- triple(S,P,O), triple(P,rdfs:range,C).  
triple(C,rdfs:subClassOf,rdfs:Resource) :- triple(C,rdf:type,rdfs:Class).  
triple(C1,rdfs:subClassOf,C3) :- triple(C1,rdfs:subClassOf,C2),  
triple(C2,rdfs:subClassOf,C3).  
triple(S,rdf:type,C2) :- triple(S,rdf:type,C1),  
triple(C1,rdfs:subClassOf,C2).  
triple(C,rdf:type,rdfs:Class) :- triple(S,rdf:type,C).  
triple(C,rdfs:subClassOf,C) :- triple(C,rdf:type,rdfs:Class).  
triple(P1,rdfs:subPropertyOf,P3) :- triple(P1,rdfs:subPropertyOf,P2),  
triple(P2,rdfs:subPropertyOf,P3).  
triple(S,P2,O) :- triple(S,P1,O),  
triple(P1,rdfs:subPropertyOf,P2).  
triple(P,rdfs:subPropertyOf,P) :- triple(P,rdf:type,rdf:Property).
```

plus the respective axiomatic triples in RDF/RDFS, cf. Sections 3.1 and 4.1 of

<http://www.w3.org/TR/rdf-mt/>.

## A lightweight approach to add rules:

Adding normal logic programs on top of RDF(S)

- ▶ We want to add arbitrary LP style rules on top of RDF(S)

## A lightweight approach to add rules:

Adding normal logic programs on top of RDF(S)

- ▶ We want to add arbitrary LP style rules on top of RDF(S)
- ▶ We want to allow negation as failure (normal logic programs)

## A lightweight approach to add rules:

Adding normal logic programs on top of RDF(S)

- ▶ We want to add arbitrary LP style rules on top of RDF(S)
- ▶ We want to allow negation as failure (normal logic programs)
- ▶ We want to base our semantics on the stable model semantics for logic programs

## A lightweight approach to add rules:

Adding normal logic programs on top of RDF(S)

- ▶ We want to add arbitrary LP style rules on top of RDF(S)
- ▶ We want to allow negation as failure (normal logic programs)
- ▶ We want to base our semantics on the stable model semantics for logic programs
- ▶ But: There are some problems when allowing negation as failure on the Web

## The stable model semantics for logic programs (1/2)

### Syntax:

A normal logic programs  $P$  is a set of rules of the form:

$$h : -l_1, \dots, l_n.$$

- ▶  $l_1, \dots, l_n$  are literals, i.e. atoms  $p(t_1, \dots, t_m)$  or negated atoms  $\text{not } p(t_1, \dots, t_m)$ , such that  $t_1, \dots, t_m$  are either constants or variables.

# The stable model semantics for logic programs (1/2)

## Syntax:

A normal logic programs  $P$  is a set of rules of the form:

$$h : -l_1, \dots, l_n.$$

- ▶  $l_1, \dots, l_n$  are literals, i.e. atoms  $p(t_1, \dots, t_m)$  or negated atoms  $\text{not } p(t_1, \dots, t_m)$ , such that  $t_1, \dots, t_m$  are either constants or variables.
- ▶  $h$  is an atom.

## Semantics:

Herbrand models defined as usual:

- ▶  $U_H$  consists of the the set of all constants appearing in  $P$
- ▶  $B_H$  is the set of all atoms constructible from predicate symbols in  $P$  and constants in  $U_H$ .
- ▶ Since there are no function symbols,  $B_H$  is finite.
- ▶ A Herbrand interpretation  $I$  is a subset of  $B_H$ .
- ▶ We denote by  $\text{ground}(P)$  the set of all possible **ground instantiations** of rules in  $P$  where variables are substituted with the constants in  $U_H$ .
- ▶ A Herbrand interpretation  $I$  is called Herbrand model of  $P$  if all rules in  $\text{ground}(P)$  are satisfied wrt.  $I$ .
- ▶ Each positive (not-free) program  $P$  has a unique minimal Herbrand model  $M$ .

## The stable model semantics for logic programs (2/2)

The stable models for programs with negation is defined via the *Gelfond-Lifschitz-reduct*:

Let  $I$  be a Herbrand interpretation of  $P$ . Then the reduct  $P^I$  denotes the set of rules obtained from  $ground(P)$  by

- ▶ removing all rules  $r$  such that  $\text{not } a$  occurs in the body of  $r$  for some  $a \in I$
- ▶ removing all literals  $\text{not } a$  from the remaining rules.



## The stable model semantics for logic programs (2/2)

The stable models for programs with negation is defined via the *Gelfond-Lifschitz-reduct*:

Let  $I$  be a Herbrand interpretation of  $P$ . Then the reduct  $P^I$  denotes the set of rules obtained from  $ground(P)$  by

- ▶ removing all rules  $r$  such that  $\text{not } a$  occurs in the body of  $r$  for some  $a \in I$
- ▶ removing all literals  $\text{not } a$  from the remaining rules.

A Herbrand interpretation  $M$  is called *stable model* of a normal logic program  $P$  iff  $M$  is the minimal Herbrand model of  $P^M$ .

## The stable model semantics for logic programs (2/2)

The stable models for programs with negation is defined via the *Gelfond-Lifschitz-reduct*:

Let  $I$  be a Herbrand interpretation of  $P$ . Then the reduct  $P^I$  denotes the set of rules obtained from  $ground(P)$  by

- ▶ removing all rules  $r$  such that `not`  $a$  occurs in the body of  $r$  for some  $a \in I$
- ▶ removing all literals `not`  $a$  from the remaining rules.

A Herbrand interpretation  $M$  is called *stable model* of a normal logic program  $P$  iff  $M$  is the minimal Herbrand model of  $P^M$ .

There are efficient solvers to compute stable models: `d1v`, `smodels`, `cmmodels`, `assat`, etc.

## Incomplete knowledge on the Web

Problems:

- ▶ **Incompleteness**: The knowledge of a search engine about the Web is notoriously incomplete, i.e. it does not know about all available Websites.

*“Search for **all** movies by Ed Wood”*

Cannot be answered, without e.g. local completeness assumptions. Usually, this is not a problem as long as query results are good enough (sound, at least).

# Incomplete knowledge on the Web

## Problems:

- ▶ **Incompleteness**: The knowledge of a search engine about the Web is notoriously incomplete, i.e. it does not know about all available Websites.

*“Search for **all** movies by Ed Wood”*

Cannot be answered, without e.g. local completeness assumptions. Usually, this is not a problem as long as query results are good enough (sound, at least).

- ▶ More severe problems with **negation** in rules and queries:

*“Search for science fiction movies which are **NOT** rated as bad?”*

problematic, since using normal *negation as failure* over a finite subset of webpages is not only incomplete, but incorrect!

# Incomplete knowledge on the Web

## Problems:

- ▶ **Incompleteness**: The knowledge of a search engine about the Web is notoriously incomplete, i.e. it does not know about all available Websites.

*“Search for **all** movies by Ed Wood”*

Cannot be answered, without e.g. local completeness assumptions. Usually, this is not a problem as long as query results are good enough (sound, at least).

- ▶ More severe problems with **negation** in rules and queries:

*“Search for science fiction movies which are **NOT** rated as bad?”*

problematic, since using normal *negation as failure* over a finite subset of webpages is not only incomplete, but incorrect!

**Solution:** Enforce to make the **scope** for negation as failure always explicit!

## Metadata on the Web as distributed rule sets

<pre>http://moviereviews.com/ rated(m1,bad). rated(X,bad) :-     directedBy(X,"Ed Wood").</pre>	<pre>http://polleres.net/myreviews rated(m2,bad). movie(m2). rated(X,bad) :- movie(X),                 not movie(X)@http://imdb.com.</pre>
	<pre>http://badmovies.org/ movie(m1). ... rated(X,bad) :- movie(X)@http://badmovies.org.</pre>
<pre>http://imdb.com/ sciFiMovie(m1). hasTitle(m1,"Plan 9 from Outer Space"). directedBy(m1,"Ed Wood"). sciFiMovie(m2). hasTitle(m2,"Matrix Revolutions"). directedBy(m2,"Andy Wachowski"). directedBy(m2,"Larry Wachowski"). sciFiMovie(m3). hasTitle(m3,"Bride of the Monster"). directedBy(m3,"Ed Wood"). movie(X) :- sciFiMovie(X). ...</pre>	

Figure: We use a more LP notation than before ... and add rules

## Syntax: Logic Programs with scoped literals

Assumption: A *program* is a set of rules associated with a URI  $u$ , where it is accessible:

$$h : -l_1, \dots, l_n.$$

- ▶ Body Literals:

## Syntax: Logic Programs with scoped literals

Assumption: A *program* is a set of rules associated with a URI  $u$ , where it is accessible:

$$h : -l_1, \dots, l_n.$$

- ▶ Body Literals:
  - ▶ open (unscoped) literals  $a$ .



## Syntax: Logic Programs with scoped literals

Assumption: A *program* is a set of rules associated with a URI  $u$ , where it is accessible:

$$h : -l_1, \dots, l_n.$$

- ▶ Body Literals:
  - ▶ open (unscoped) literals  $a$ .
  - ▶ scoped literals (*not*)  $a@u$ .

## Syntax: Logic Programs with scoped literals

Assumption: A *program* is a set of rules associated with a URI  $u$ , where it is accessible:

$$h : -l_1, \dots, l_n.$$

- ▶ Body Literals:
  - ▶ open (unscoped) literals  $a$ .
  - ▶ scoped literals (*not*)  $a@u$ .
  - ▶ negative literals (negation as failure) **MUST** be scoped!

## Syntax: Logic Programs with scoped literals

Assumption: A *program* is a set of rules associated with a URI  $u$ , where it is accessible:

$$h : -l_1, \dots, l_n.$$

- ▶ Body Literals:
  - ▶ open (unscoped) literals  $a$ .
  - ▶ scoped literals (*not*)  $a@u$ .
  - ▶ negative literals (negation as failure) **MUST** be scoped!
- ▶ Head atoms are always unscoped (to be viewed in the scope/context of the program (uri) where they appear), i.e. if  $h$  is derived from  $p$  we would also expect  $h@p$  to be derivable.

## Syntax: Logic Programs with scoped literals

Assumption: A *program* is a set of rules associated with a URI  $u$ , where it is accessible:

$$h : -l_1, \dots, l_n.$$

- ▶ Body Literals:
  - ▶ open (unscoped) literals  $a$ .
  - ▶ scoped literals (*not*)  $a@u$ .
  - ▶ negative literals (negation as failure) MUST be scoped!
- ▶ Head atoms are always unscoped (to be viewed in the scope/context of the program (uri) where they appear), i.e. if  $h$  is derived from  $p$  we would also expect  $h@p$  to be derivable.

Examples of **open** and **scoped** rules:

```
http://moviereviews.com :
```

```
  rated(X,bad) :- directedBy(X,"Ed Wood").
```

```
http://badmovies.org :
```

```
  movie(m1).
```

```
  ...
```

```
  rated(X,bad) :- movie(X)@http://badmovies.org!
```



## Requirements for a reasonable semantics for such rules

Let  $Cn_{\mathcal{S}}(\mathcal{P})$  denote the set of consequences from a set of programs  $\mathcal{P}$  wrt. semantics  $\mathcal{S}$

## Requirements for a reasonable semantics for such rules

Let  $Cn_{\mathcal{S}}(\mathcal{P})$  denote the set of consequences from a set of programs  $\mathcal{P}$  wrt. semantics  $\mathcal{S}$

**R1 Context-Monotonicity:** When asking query  $q$  over (open and scoped) literals to an agent which is aware of a set of programs  $\mathcal{P}$  (query context), I expect that I don't need to retract any inferences when asking another agent aware of  $\mathcal{R} \supset \mathcal{P}$ , i.e.

$$\mathcal{P} \subseteq \mathcal{R} \Rightarrow Cn_{\mathcal{S}}(\mathcal{P}) \subseteq Cn_{\mathcal{S}}(\mathcal{R})$$

## Requirements for a reasonable semantics for such rules

Let  $Cn_{\mathcal{S}}(\mathcal{P})$  denote the set of consequences from a set of programs  $\mathcal{P}$  wrt. semantics  $\mathcal{S}$

**R1 Context-Monotonicity:** When asking query  $q$  over (open and scoped) literals to an agent which is aware of a set of programs  $\mathcal{P}$  (query context), I expect that I don't need to retract any inferences when asking another agent aware of  $\mathcal{R} \supset \mathcal{P}$ , i.e.

$$\mathcal{P} \subseteq \mathcal{R} \Rightarrow Cn_{\mathcal{S}}(\mathcal{P}) \subseteq Cn_{\mathcal{S}}(\mathcal{R})$$

**R2** The chosen semantics should be **closed under context closure, i.e.**

$$Cn_{\mathcal{S}}(\mathcal{P}) = Cn_{\mathcal{S}}(Cl(\mathcal{P}))$$

where  $Cl(\mathcal{P})$  is the set of all programs in  $\mathcal{P}$  plus the ones “linked” recursively via scoped literals.

## Requirements for a reasonable semantics for such rules

Let  $Cn_{\mathcal{S}}(\mathcal{P})$  denote the set of consequences from a set of programs  $\mathcal{P}$  wrt. semantics  $\mathcal{S}$

**R1 Context-Monotonicity:** When asking query  $q$  over (open and scoped) literals to an agent which is aware of a set of programs  $\mathcal{P}$  (query context), I expect that I don't need to retract any inferences when asking another agent aware of  $\mathcal{R} \supset \mathcal{P}$ , i.e.

$$\mathcal{P} \subseteq \mathcal{R} \Rightarrow Cn_{\mathcal{S}}(\mathcal{P}) \subseteq Cn_{\mathcal{S}}(\mathcal{R})$$

**R2** The chosen semantics should be **closed under context closure, i.e.**

$$Cn_{\mathcal{S}}(\mathcal{P}) = Cn_{\mathcal{S}}(Cl(\mathcal{P}))$$

where  $Cl(\mathcal{P})$  is the set of all programs in  $\mathcal{P}$  plus the ones "linked" recursively via scoped literals.

We define two semantics based on the stable model semantics, both fulfilling R1, one of them fulfilling R2.



## Contextually Bounded semantics: $Cn_{CB}$ (1/2)

Intuitively, scoping negative literals alone is not enough, since scoped literals can again depend on open rules, e.g.

```
interestingmovie(X) :- movie(X),  
                       not rated(X,bad)@http://moviereviews.com.
```

depends on whether the agent evaluating this rule knows  
`http://imdb.com` or not.

## Contextually Bounded semantics: $Cn_{CB}$ (1/2)

Intuitively, scoping negative literals alone is not enough, since scoped literals can again depend on open rules, e.g.

```
interestingmovie(X) :- movie(X),  
                        not rated(X,bad)@http://moviereviews.com.
```

depends on whether the agent evaluating this rule knows `http://imdb.com` or not.

1<sup>st</sup> proposal to deal with this: Allow only contextually bounded negation.

*We call a (set of) rules **contextually bounded** if no negative literal recursively depends on unscoped (open) literals.*

## Contextually Bounded semantics: $Cn_{CB}$ (1/2)

Intuitively, scoping negative literals alone is not enough, since scoped literals can again depend on open rules, e.g.

```
interestingmovie(X) :- movie(X),  
                       not rated(X,bad)@http://moviereviews.com.
```

depends on whether the agent evaluating this rule knows `http://imdb.com` or not.

1<sup>st</sup> proposal to deal with this: Allow only contextually bounded negation.

*We call a (set of) rules **contextually bounded** if no negative literal recursively depends on unscoped (open) literals.*

```
interestingmovie(X) :- movie(X),  
                       not rated(X,bad)@http://badmovies.com.
```

## Contextually Bounded semantics: $Cn_{CB}$ (1/2)

Intuitively, scoping negative literals alone is not enough, since scoped literals can again depend on open rules, e.g.

```
interestingmovie(X) :- movie(X),  
                       not rated(X,bad)@http://moviereviews.com.
```

depends on whether the agent evaluating this rule knows `http://imdb.com` or not.

1<sup>st</sup> proposal to deal with this: Allow only contextually bounded negation.

*We call a (set of) rules **contextually bounded** if no negative literal recursively depends on unscoped (open) literals.*

```
interestingmovie(X) :- movie(X),  
                       not rated(X,bad)@http://badmovies.com.
```

Attention: This rule is NOT contextually bounded:

```
rated(X,bad) :- movie(X), not movie(X)@http://imdb.com.
```

```
interestingmovie(X) :- movie(X),  
                      not rated(X,bad)@http://moviereviews.com.
```

```
interestingmovie(X) :- movie(X),  
                      not rated(X,bad)@http://badmovies.com.
```

```
http://moviereviews.com/  
rated(m1,bad).  
rated(X,bad) :-  
    directedBy(X,"Ed Wood").
```

```
http://polleres.net/myreviews  
rated(m2,bad). movie(m2).  
rated(X,bad) :- movie(X),  
                not movie(X)@http://imdb.com.
```

```
http://badmovies.org/  
movie(m1).  
...  
rated(X,bad) :- movie(X)@http://badmovies.org.
```

```
interestingmovie(X) :- movie(X),
                      not rated(X,bad)@http://moviereviews.com.
```

```
interestingmovie(X) :- movie(X),
                      not rated(X,bad)@http://badmovies.com.
```

<pre>http://moviereviews.com/ rated(m1,bad). rated(X,bad) :-     directedBy(X,"Ed Wood").</pre>	<pre>http://polleres.net/myreviews rated(m2,bad). movie(m2). rated(X,bad) :- movie(X),                 not movie(X)@http://imdb.com.</pre>
	<pre>http://badmovies.org/ movie(m1). ... rated(X,bad) :- movie(X)@http://badmovies.org.</pre>
<pre>http://imdb.com/ sciFiMovie(m1). hasTitle(m1,"Plan 9 from Outer Space"). directedBy(m1,"Ed Wood"). sciFiMovie(m2). hasTitle(m2,"Matrix Revolutions"). directedBy(m2,"Andy Wachowski"). directedBy(m2,"Larry Wachowski"). sciFiMovie(m3). hasTitle(m3,"Bride of the Monster"). directedBy(m3,"Ed Wood"). movie(X) :- sciFiMovie(X)@http://imdb.com.</pre>	

## Contextually Bounded semantics: $Cn_{CB}$ (2/2)

Semantics is based on usual stable model semantics, plus “linking” open and closed literals by the following rewriting  $p_{CB}$  for each rule in program  $p$ :

$$h \text{ :- } l_1, \dots, l_n.$$

## Contextually Bounded semantics: $Cn_{CB}$ (2/2)

Semantics is based on usual stable model semantics, plus “linking” open and closed literals by the following rewriting  $p_{CB}$  for each rule in program  $p$ :

$$h :- l_1, \dots, l_n.$$

$\Rightarrow$

$$h@p :- l_1, \dots, l_n.$$

$$h :- h@p.$$



## Contextually Bounded semantics: $Cn_{CB}$ (2/2)

Semantics is based on usual stable model semantics, plus “linking” open and closed literals by the following rewriting  $p_{CB}$  for each rule in program  $p$ :

$$h :- l_1, \dots, l_n.$$

$\Rightarrow$

$$h@p :- l_1, \dots, l_n.$$

$$h :- h@p.$$

Let  $\mathcal{P}_{CB} = \bigcup_{p \in \text{Cl}(\mathcal{P})} p_{CB}$ , then

$$Cn_{CB}(\mathcal{P}) = \bigcap \mathcal{M}(\mathcal{P}_{CB})$$

where  $\mathcal{M}(p)$  is defined as the set of all stable models of program  $p$

## Contextually Bounded semantics: Implications

R1 holds, by contextual boundedness (easy proof in [Polleres, et al. 2006]).

## Contextually Bounded semantics: Implications

- R1 holds, by contextual boundedness (easy proof in [Polleres, et al. 2006]).
- R2 holds trivially ( $\mathcal{P}_{CB}$  is defined via the closure of  $\mathcal{P}$ ).

## Contextually Bounded semantics: Implications

R1 holds, by contextual boundedness (easy proof in [Polleres, et al. 2006]).

R2 holds trivially ( $\mathcal{P}_{CB}$  is defined via the closure of  $\mathcal{P}$ ).

Problem:

## Contextually Bounded semantics: Implications

- R1 holds, by contextual boundedness (easy proof in [Polleres, et al. 2006]).
- R2 holds trivially ( $\mathcal{P}_{CB}$  is defined via the closure of  $\mathcal{P}$ ).

Problem:

- ▶ Contextual boundedness is a prerequisite:

$p$ :

$a :- \text{not } b@p.$

$b :- c.$

## Contextually Bounded semantics: Implications

R1 holds, by contextual boundedness (easy proof in [Polleres, et al. 2006]).

R2 holds trivially ( $\mathcal{P}_{CB}$  is defined via the closure of  $\mathcal{P}$ ).

Problem:

- ▶ Contextual boundedness is a prerequisite:

$p:$	$r:$
$a :- \text{not } b@p.$	$c.$
$b :- c.$	

## Contextually Bounded semantics: Implications

R1 holds, by contextual boundedness (easy proof in [Polleres, et al. 2006]).

R2 holds trivially ( $\mathcal{P}_{CB}$  is defined via the closure of  $\mathcal{P}$ ).

Problem:

- ▶ Contextual boundedness is a prerequisite:

$p:$	$r:$
$a :- \text{not } b@p.$	$c.$
$b :- c.$	

- ▶ Contextual boundedness is hardly maintainable in an open context, especially when contexts change (adding open rule):

$p:$	$r:$
$a.$	$b :- \text{not } a@p.$

## Contextually Bounded semantics: Implications

R1 holds, by contextual boundedness (easy proof in [Polleres, et al. 2006]).

R2 holds trivially ( $\mathcal{P}_{CB}$  is defined via the closure of  $\mathcal{P}$ ).

Problem:

- ▶ Contextual boundedness is a prerequisite:

$p:$	$r:$
$a :- \text{not } b@p.$	$c.$
$b :- c.$	

- ▶ Contextual boundedness is hardly maintainable in an open context, especially when contexts change (adding open rule):

$p:$	$r:$
$a :- c.$	$b :- \text{not } a@p.$



## Contextually Closed semantics $Cn_{CC}$ (1/2)

Alternative approach: Intuitively “close off”, all open rules if referenced via a scoped literal.

We define an alternative rewriting  $p_{CC}$  for each rule in program  $p$ :

$$h \text{ :- } l_1, \dots, l_n.$$

## Contextually Closed semantics $Cn_{CC}$ (1/2)

Alternative approach: Intuitively “close off”, all open rules if referenced via a scoped literal.

We define an alternative rewriting  $p_{CC}$  for each rule in program  $p$ :

$$h \text{ :- } l_1, \dots, l_n.$$

$\Rightarrow$

$$h@p \text{ :- } l'_1, \dots, l'_n.$$

where  $l'_i = l_i$  for scoped literals and  $l'_i = l_i@p$  otherwise.

## Contextually Closed semantics $Cn_{CC}$ (1/2)

Alternative approach: Intuitively “close off”, all open rules if referenced via a scoped literal.

We define an alternative rewriting  $p_{CC}$  for each rule in program  $p$ :

$$h \text{ :- } l_1, \dots, l_n.$$

$\Rightarrow$

$$h@p \text{ :- } l'_1, \dots, l'_n.$$

where  $l'_i = l_i$  for scoped literals and  $l'_i = l_i@p$  otherwise.

Let  $\mathcal{P}_{CC} = \bigcup_{p \in \mathcal{P}} \bigcup_{p \in Cl(\mathcal{P})} p_{CC}$ , then

$$Cn_{CC}(\mathcal{P}) = \bigcap \mathcal{M}(\mathcal{P}_{CC})$$

where  $\mathcal{M}(p)$  is defined as the set of all stable models of program  $p$

## Contextually Closed semantics $Cn_{CC}$ (2/2)

Intuitively, contextually closed semantics is more cautious or “local”:

## Contextually Closed semantics $Cn_{CC}$ (2/2)

Intuitively, contextually closed semantics is more cautious or “local”:

- ▶  $Cn_{CC}(\mathcal{P}) \subseteq Cn_{CB}(\mathcal{P})$  (proof in [Polleres, et al. 2006]).

## Contextually Closed semantics $Cn_{CC}$ (2/2)

Intuitively, contextually closed semantics is more cautious or “local”:

- ▶  $Cn_{CC}(\mathcal{P}) \subseteq Cn_{CB}(\mathcal{P})$  (proof in [Polleres, et al. 2006]).
- ▶ It does not “traverse” closure concerning open literals, i.e. **R2** does not hold:

$p$ :

$a :- b@r.$

$r$ :

$c.$

Here,  $c \in Cn_{CB}(p)$ , but  $c \notin Cn_{CC}(p)$

## Contextually Closed semantics $Cn_{CC}$ (2/2)

Intuitively, contextually closed semantics is more cautious or “local”:

- ▶  $Cn_{CC}(\mathcal{P}) \subseteq Cn_{CB}(\mathcal{P})$  (proof in [Polleres, et al. 2006]).
- ▶ It does not “traverse” closure concerning open literals, i.e. **R2** does not hold:

$p$ :

$a :- b@r.$

$r$ :

$b :- c.$

$c.$

Here,  $a \in Cn_{CB}(p)$ , but  $a \notin Cn_{CC}(p)$  which one might consider more intuitive, i.e. cross-effects of open literals only within the query context.

## Summary of this approach

- ▶ Can be used on top of RDF (modulo blank nodes)



## Summary of this approach

- ▶ Can be used on top of RDF (modulo blank nodes)
- ▶ Clear definition of “scoped” negation

## Summary of this approach

- ▶ Can be used on top of RDF (modulo blank nodes)
- ▶ Clear definition of “scoped” negation
- ▶ First attempt to
  - ▶ issue context-monotonicity
  - ▶ make distinction between open and closed rules

## Summary of this approach

- ▶ Can be used on top of RDF (modulo blank nodes)
- ▶ Clear definition of “scoped” negation
- ▶ First attempt to
  - ▶ issue context-monotonicity
  - ▶ make distinction between open and closed rules
- ▶ The two solutions proposed are simple/cautious on purpose, trying to start discussion about the “right” semantics of scoped negation for the Semantic Web.

## Related works

- ▶ FLORA-2 (Kifer): an engine for F-Logic programs, allows modules, i.e. contexts, open literals/rules supported by allowing variables in place of modules, e.g.

$$a : -b@X.$$

No requirement for context-monotonicity though, well-founded semantics

- ▶ TRIPLE (Decker, et al.) allows parametrized contexts, union, intersection, set difference of contexts, also parameters allowed. Negation unsupported in current implementation, AFAIK.
- ▶ C-OWL extension of OWL by contexts and bridge rules, *local model semantics*, i.e. local inconsistencies do not spread over to the whole.

Sideremark: The approach is orthogonal to LCWA (Local closed world assumption) approaches allowing local completeness statements.

## Issues/Future work

- ▶ Investigate a Local Model Semantics

## Issues/Future work

- ▶ Investigate a Local Model Semantics
- ▶ Exact relation with SPARQL, RIF

## Issues/Future work

- ▶ Investigate a Local Model Semantics
- ▶ Exact relation with SPARQL, RIF
- ▶ Complexity, Prototype implementation (DLV, YARS)

## Issues/Future work

- ▶ Investigate a Local Model Semantics
- ▶ Exact relation with SPARQL, RIF
- ▶ Complexity, Prototype implementation (DLV, YARS)
- ▶ Investigate different semantics (well-founded vs. stable)



## Issues/Future work

- ▶ Investigate a Local Model Semantics
- ▶ Exact relation with SPARQL, RIF
- ▶ Complexity, Prototype implementation (DLV, YARS)
- ▶ Investigate different semantics (well-founded vs. stable)
- ▶ Classical Negation, integration with the Ontology Layer (OWL)

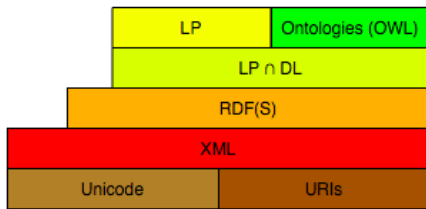
## Time allowed... How to integrate OWL with Rules?

OWL (Web Ontology Language) adds more expressivity on top of RDF, allows to define taxonomies based on intersection, complement, cardinality restrictions, etc.

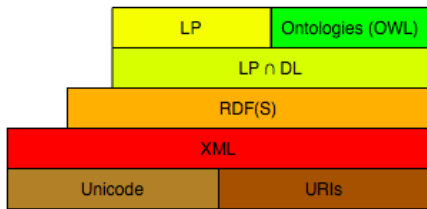
Axiom	DL Syntax	Constructor	DL Syntax
subClassOf	$C_1 \sqsubseteq C_2$	intersectionOf	$C_1 \sqcap \dots \sqcap C_n$
equivalentClass	$C_1 \equiv C_2$	unionOf	$C_1 \sqcup \dots \sqcup C_n$
disjointWith	$C_1 \sqsubseteq \neg C_2$	complementOf	$\neg C$
sameIndividualAs	$\{x_1\} \equiv \{x_2\}$	oneOf	$\{x_1\} \sqcup \dots \sqcup \{x_n\}$
differentFrom	$\{x_1\} \sqsubseteq \neg \{x_2\}$	allValuesFrom	$\forall P.C$
subPropertyOf	$P_1 \sqsubseteq P_2$	someValuesFrom	$\exists P.C$
equivalentProperty	$P_1 \equiv P_2$	maxCardinality	$\leq nP$
inverseOf	$P_1 \equiv P_2^-$	minCardinality	$\geq nP$
transitiveProperty	$P^+ \sqsubseteq P$		
functionalProperty	$\top \sqsubseteq \leq 1P$		
inverseFunctionalProperty	$\top \sqsubseteq \leq 1P^-$		

Expressivity in principle based on the description logic  $\mathcal{SHOIN}(D)$ .  
(OWL DL, this is not not completely true for OWL Full)

# Interoperability on the common (Horn) intersection only



## Interoperability on the common (Horn) intersection only



### DLP:

The Horn fragment of SHOIN(D) can be understood as a rule set. So, you can understand a small part of OWL as rules.

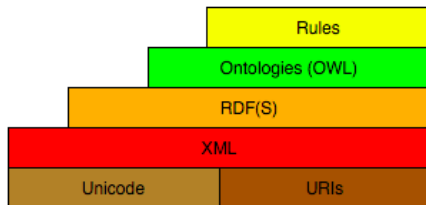
e.g.  $\text{father}(X) \leftarrow \text{parent}(X,Y), \text{person}(Y), \text{male}(X).$

$\Leftrightarrow \text{Father} \sqsubseteq \exists \text{Parent}^{-1}. \text{Human} \sqcap \text{Male}$

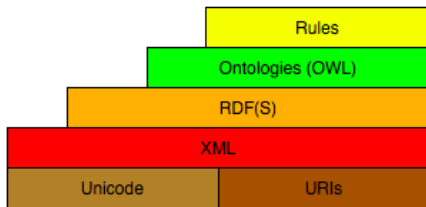
BUT: cannot cover much either on the rules part, nor on the DL part.

Only a basis for extensions in either direction.

## Rules on top of OWL – SWRL



## Rules on top of OWL – SWRL



### SWRL:

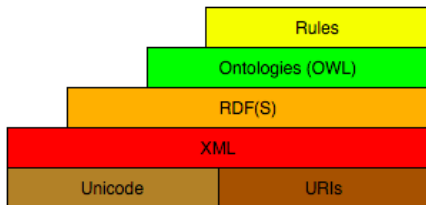
Add Horn rules to OWL syntax, allows you to express e.g.  
`uncle(X,Y) ← male(X), sibling(X,Z),parent(Z,Y).`

But, also:

`∃X parent(X,Y) ← male(Z).`

(from `∃Parent.Human ⊑ male`)

## Rules on top of OWL – SWRL



### SWRL:

Add Horn rules to OWL syntax, allows you to express e.g.  
`uncle(X,Y) ← male(X), sibling(X,Z),parent(Z,Y).`

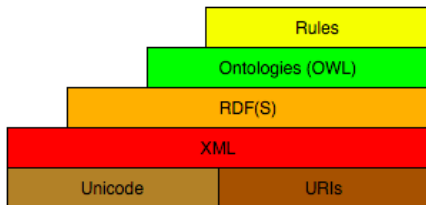
But, also:

`∃X parent(X,Y) ← male(Z).`

(from `∃Parent.Human ⊑ male`)

- ▶ On the one hand naive combination of Horn + DL destroys decidability of either.

## Rules on top of OWL – SWRL



### SWRL:

Add Horn rules to OWL syntax, allows you to express e.g.  
`uncle(X,Y) ← male(X), sibling(X,Z),parent(Z,Y).`

But, also:

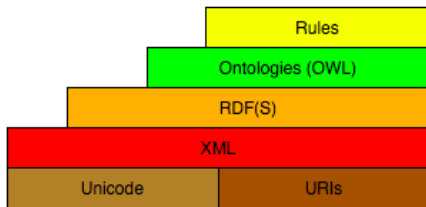
`∃X parent(X,Y) ← male(Z).`

(from `∃Parent.Human ⊑ male`)

- ▶ On the one hand naive combination of Horn + DL destroys decidability of either.
- ▶ On the other hand SWRL does not even allow arbitrary HORN but only binary/unary predicates.



## Rules on top of OWL – SWRL



### SWRL:

Add Horn rules to OWL syntax, allows you to express e.g.  
`uncle(X,Y) ← male(X), sibling(X,Z),parent(Z,Y).`

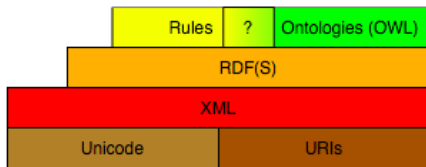
But, also:

`∃X parent(X,Y) ← male(Z).`

(from `∃Parent.Human ⊑ male`)

- ▶ On the one hand naive combination of Horn + DL destroys decidability of either.
- ▶ On the other hand SWRL does not even allow arbitrary HORN but only binary/unary predicates.
- ▶ Issues like open vs. closed rules, negation as failure untouched.

## Interface between LP and DL – dl-programs



T. Eiter, T. Lukasiewicz, R. Schindlauer, H. Tompits *Combining Answer Set Programming with Description Logics for the Semantic Web*. KR, 2004.

Define an extension of LP under the stable model semantics by so-called dl-atoms in the body, which allow to query a DL Knowledge base, but also interchange facts in the other direction. Authors define minimal Herbrand models and stable models for dl-programs.

- pro Decidability remains.
- con DL KB and LP program talk about different things, exchange only via “import/export”.

Generalization of this technique available, HEX-programs.  
Extension to scoped literals? Not straightforward.

Thank you for your attention!