Answer Set Programming for the Semantic Web

Thomas Eiter

Institute of Information Systems

Vienna University of Technology (TU Wien)

KBS Knowledge-Based Systems Group

eiter@kr.tuwien.ac.at

23rd Int'l Conference on Logic Programming, Sept 8-13, 2007, Porto, Portugal

Collaborators: Jos de Bruijn (Free U. Bolzano), Giovambattista Ianni (U. Calabria), Thomas Lukasiewicz (U. Rome "La Sapienza"), Axel Polleres (DERI Galway) Roman Schindlauer (TU Wien), Hans Tompits (TU Wien)

(IST REWERSE, FWF Project P17212-N04)

Contents

- Recall Answer Set Programming
- Semantic Web
- ASP for Semantic Web
- Focus: Combining Rules and Ontologies
- Research Issues

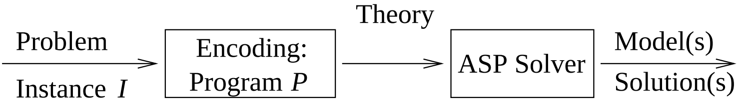
Answer Set Programming

- Term was coined by Vladimir Lifschitz (1999)
- Roots in KR, logic programming, nonmonotonic reasoning
- Proposed by other people (Marek & Truszczyński, Niemelä,...) at about the same time
- Makes fruitful use of the (often criticized) fact that non-monotonic logic programs (LPs) have multiple stable models (aka *answer sets*)
- At an abstract level, relates to SAT solving and CSP.

ASP Paradigm

General idea: stable models are solutions!

Reduce solving a problem instance *I* to computing stable models of an LP



1. *Encode* *I* as a (non-monotonic) logic program *P*, such that solutions of *I* are represented by models of *P*
2. *Compute* some model *M* of *P*, using an ASP solver
3. *Extract* a solution for *I* from *M*.

Variant: Compute multiple models (for multiple / all solutions)

Answer Set Semantics

- A *normal logic program* P is a (finite) set of rules of the form

$$a \leftarrow \; b_1, \ldots, b_m, not \;\; c_1, \ldots, not \;\; c_n$$

  where all a, b_i, c_j are literals of the form p or $-p$, where p is an atom.

  - "$-$" is called strong negation (also written as "$\neg$")
  - a may be missing (constraint)

- HB_P is the set of all literals p and $-p$ with predicates and constants from P.
- In *disjunctive programs*, the rule head may be a disjunction $a_1 \mid \cdots \mid a_k$ of literals

Answer Sets

- Consider a consistent set of ground literals $M \subseteq HB_P$
- Recall the Gelfond-Lifschitz (GL) reduct P^M:

  remove from the grounding of P, $Ground(P)$,

  1. every rule $a \leftarrow b_1, \ldots, b_m, not\ \ c_1, \ldots, not\ \ c_n$, where some c_i is in M, and
  2. all literals $not\ \ c_j$ from the remaining rules.

- Then, M is an answer set of P iff M is the least model of P^M
- Such an M satisfies all rules, and intuitively P justifies each atom in M
- Easily generalized to disjunctive P: "the least" $\rightsquigarrow$ "a minimal"

Example

$$P = \{\ person(joey);\ \ male(X) \mid female(X) \leftarrow person(X)\ \}$$

- $M_1 = \{person(joey), male(joey), bachelor(joey)\}$ is stable
- $M_2 = \{person(joey), male(joey), married(joey)\}$ is not stable

In general, no, one, or multiple stable models exist.

Further stable model:

- $M_3 = \{person(joey), female(joey)\}$

ASP vs. Prolog

- **ASP features “pure” declarative programming**
  Under answer set semantics,
  - the order of program rules does not matter;
  - the order of subgoals in a rule does not matter;
- **Nondeterminism in ASP:** Means to make guesses
- **Only limited support of function symbols in current ASP solvers** (more is emerging)
  Simple Horn LPs with function symbols are undecidable, while ASP strives for decidability

  Some decidable fragments of ASP with function symbols:
  - ω-restricted programs [Syrjänen, LPNMR01]
  - finitary programs [Bonatti, 04],
    finitely recursive programs [Baselice et al., ICLP07]
  - FDNC programs [Simkus & E_, LPAR07]

ASP Applications

Problems in different domains (some with substantial amount of data), see
`http://www.kr.tuwien.ac.at/projects/WASP/report.html`

- information integration
- constraint satisfaction
- planning, routing
- biology
- diagnosis
- security analysis
- configuration
- computer-aided verification
- ...

ASP Showcase: `http://www.kr.tuwien.ac.at/projects/WASP/showcase.html`

ASP Solvers

DLV	`http://www.dbai.tuwien.ac.at/proj/dlv/`
Smodels	`http://www.tcs.hut.fi/Software/smodels/`
GnT	`http://www.tcs.hut.fi/Software/gnt/`
Cmodels	`http://www.cs.utexas.edu/users/tag/cmodels/`
ASSAT	`http://assat.cs.ust.hk/`
NoMore	`http://www.cs.uni-potsdam.de/~linke/nomore/`
Platypus	`http://www.cs.uni-potsdam.de/platypus/`
clasp	`http://www.cs.uni-potsdam.de/clasp/`
XASP	distributed with XSB v2.6 `http://xsb.sourceforge.net`
aspps	`http://www.cs.engr.uky.edu/ai/aspps/`
ccalc	`http://www.cs.utexas.edu/users/tag/cc/`

- Some provide a number of extensions to the language described here.
- Answer Set Solver Implementation: see Niemelä's ICLP'04 tutorial.

ASP Extensions

- Many extensions have been proposed, partly motivated by applications
- Some are syntactic sugar, other strictly add expressiveness
- Incomplete list:
  - cardinality constraints (Smodels)
  - optimization: weight constraints, *minimize* (Smodels); weak constraints (DLV)
  - aggregates (Smodels, DLV)
  - templates (for macros)
  - preferences: e.g., PLP
  - KR frontends (diagnosis, planning,...) in DLV
- Comprehensive survey of extensions:

  `http://www.tcs.hut.fi/Research/Logic/wasp/wp3/`

Semantic Web

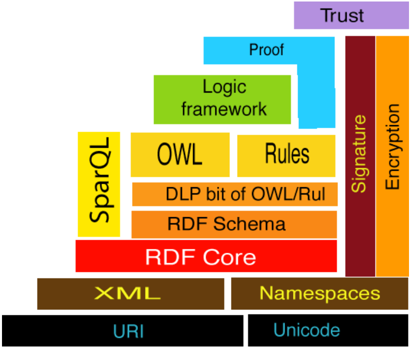
Why a "Semantic" Web?

- Undoubtedly, the World Wide Web is one of the most significant technical innovations of the last decades,
- The WWW will be (and already is) strongly impacting, and changing, our living, culture, and society.
- The WWW has been designed for human users, not for machines
- To process the data and information out on the Web, semantic annotation and description is needed.
- The Semantic Web is the vision of such an enriched, future generation Web
- Logic and logic-based formalisms (should/might) play an important role in this endeavor.

Topics in the Semantic Web

- Web services (description, discovery, invocation, composition, choreography, etc.)
- Knowledge management
- Trust, Privacy, Security
- Searching
- Data Annotation
- Ontology Management, Alignment
- Web Mining
- Reasoning on the Semantic Web
- …

Building the Semantic Web (T. Berners-Lee, 04/2005)



- RDF is the data model of the Semantic Web
- RDF Schema semantically extends RDF by simple taxonomies and hierarchies
- OWL is a W3C standard, which builds on Description Logics
- Rule languages: *Rule Interchange Format (RIF)* WG of W3C

RDF/S (Resource Description Framework)

- **RDF data model:** labeled graph of resources (nodes) linked to other resources or literals by predicates.

  - Usually represented in form of triples ⟨*Subject*, *Predicate*, *Object*⟩ e.g.

```
http://www.polleres.net/index.html dc:creator
           http://www.polleres.net/foaf.rdf#me.
http://www.polleres.net/foaf.rdf#me foaf:name "Axel Polleres"
```

```
<rdf:Description rdf:about="http://www.polleres.net/index.html">
  <dc:creator>
    <rdf:Description rdf:about="http://www.polleres.net/foaf.rdf#me">
      <foaf:Name>Axel Polleres</foaf:Name>
    </rdf:Description>
  </dc:creator>
</rdf:Description>
```

  - Resources identified by URIs

- **RDF Schema (RDFS):** simple taxonomies on RDF vocabularies using `rdf:type, rdf:subClassOf,rdfs:subPropertyOf`
- RDF semantics has some subtleties (blank nodes, XML literals, RDF keywords treated as normal resources, reification, etc.)

Ontologies

- Knowledge about concepts, individuals, their properties and relationships
- W3C Standard (04/2004): *Web Ontology Language (OWL)*
- Three increasingly expressive sublanguages

OWL Hierarchy

- OWL Lite: Concept hierarchies,
  simple constraint features. $(\risingdotseq \mathcal{SHIF}(\mathbf{D}))$
- OWL DL : Basically, DAML+OIL. $(\risingdotseq \mathcal{SHOIN}(\mathbf{D}))$
- OWL Full: Allow e.g. to treat classes as individuals.

- OWL Syntax is based on RDF
- Most widely considered: OWL DL
- Currently, OWL 1.1 (an extension of OWL DL) is a submitted to W3C

Description Logics (DLs)

OWL / Description Logics offers more expressivity than RDF/S!

- The vocabulary of basic DLs comprises:
  - Concepts (e.g., *Wine*, *WhiteWine*)
  - Roles (e.g., *hasMaker*, *madeFromGrape*)
  - Individuals (e.g., *SelaksIceWine*, *TaylorPort*)
- Statements relate individuals and their properties using
  - logical connectives ($\sqcap$, $\sqcup$, $\neg$, $\sqsubseteq$, etc), and
  - quantifiers ($\exists$, $\forall$, $\leq k$, $\geq k$, etc)
- A DL knowledge base usually comprises
  - a T-Box (terminology, conceptualization), and
  - an A-Box (assertions, extensional knowledge)
- DLs are tailored for decidable reasoning (key task: satisfiability)

Example: The Wine Ontology

- Available at `http://www.w3.org/TR/owl-guide/wine.rdf`
- Some axioms from the T-Box

$$Wine \sqsubseteq PotableLiquid \sqcap {=}1\,hasMaker \sqcap \forall hasMaker.Winery;$$
$$\exists hasColor^{-}.Wine \sqsubseteq \{"White","Rose","Red"\};$$
$$WhiteWine \equiv Wine \sqcap \forall hasColor.\{"White"\}.$$

  - A wine is a potable liquid, having exactly one maker, who is a member of the class "*Winery*".
  - Wines have colors "*White*", "*Rose*", or "*Red*".
  - A *WhiteWine* is a wine with color "*White*".

- The A-Box contains, e.g., $WhiteWine("StGenevieveTexasWhite")$, $hasMaker("TaylorPort","Taylor")$

Formal OWL / DL Semantics

- The semantics is given by a mapping to First-Order Logic
  Thus, DLs are (in essence) FO Logic in disguise

OWL property axioms as RDF Triples	DL syntax	FOL short representation
$\langle P$ rdfs:domain $C\rangle$	$\top \sqsubseteq \forall P^{-}.C$	$\forall x, y.P(x, y) \supset C(x)$
$\langle P$ rdfs:range $C\rangle$	$\top \sqsubseteq \forall P.C$	$\forall x, y.P(x, y) \supset C(y)$
$\langle P$ owl:inverseOf $P_0\rangle$	$P \equiv P_0^{-}$	$\forall x, y.P(x, y) \equiv P_0(y, x)$
$\langle P$ rdf:type owl:SymmetricProperty $\rangle$	$P \equiv P^{-}$	$\forall x, y.P(x, y) \equiv P(y, x)$
$\langle P$ rdf:type owl:FunctionalProperty $\rangle$	$\top \sqsubseteq \leqslant 1P$	$\forall x, y_1, y_2.P(x, y_1) \wedge P(x, y_2) \supset y_1 = y_2$
$\langle P$ rdf:type owl:InverseFunctionalProperty $\rangle$	$\top \sqsubseteq \leqslant 1P^{-}$	$\forall x_1, x_2, y.P(x_1, y) \wedge P(x_2, y) \supset x_1 = x_2$
$\langle P$ rdf:type owl:TransitiveProperty $\rangle$	$P^{+} \sqsubseteq P$	$\forall x, y, z.P(x, y) \wedge P(y, z) \supset P(x, z)$

OWL complex class descriptions	DL syntax	FOL short representation
owl:Thing	$\top$	$x = x$
owl:Nothing	$\bot$	$\neg x = x$
owl:intersectionOf ($C_1 \ldots C_n$)	$C_1 \sqcap \ldots \sqcap C_n$	$\bigwedge C_i(x)$
owl:unionOf ($C_1 \ldots C_n$)	$C_1 \sqcup \ldots \sqcup C_n$	$\bigvee C_i(x)$
owl:complementOf (C)	$\neg C$	$\neg C(x)$
owl:oneOf ($o_1 \ldots o_n$)	$\{o_1 \ldots o_n\}$	$\bigvee x = o_i$
owl:restriction (P owl:someValuesFrom (C))	$\exists P.C$	$\exists y.P(x, y) \wedge C(y)$
owl:restriction (P owl:allValuesFrom (C))	$\forall P.C$	$\forall y.P(x, y) \supset C(y)$
owl:restriction (P owl:value (o))	$\exists P.\{o\}$	$P(x, o)$
owl:restriction (P owl:minCardinality (n))	$\geqslant n\,P$	$\exists_{i=1}^{n} y_i.\ \bigwedge_{j=1}^{n} P(x, y_j) \wedge \bigwedge_{i \neq j} y_i \neq y_j$
owl:restriction (P owl:maxCardinality (n))	$\leqslant n\,P$	$\forall_{i=1}^{n+1} y_i.(\bigwedge_{j=1}^{n} P(x, y_i) \supset \bigvee_{i \neq j} y_i = y_j)$

Exploiting ASP for the Semantic Web

- Different ways to exploit ASP and ASP techniques have been considered
  - **As a host language for Web/Semantics Web formalisms**
    - Mapping / encoding of ontologies and DLs into ASP
    - Encoding of web query languages
      E.g., SPARQL [Polleres, WWW07], [Polleres & Schindlauer, ALPSWS07]
  - **For ad hoc problem solving.** E.g.,
    - Web service composition, e.g., [Rainer, KI-WS05]
    - Web service repair, e.g, [Friedrich et al., in progress]
  - **For combining rules and ontologies**

Special Needs of Semantic Web

- Dealing with open worlds and domains
- Access to (semi-)structured data

  cf. Frame-Syntax of HiLog, F-Logic
- Heterogeneity (integration)
- External sources, distributed computation
- Dynamics

Some Special ASP Extensions

- **Open logic programs** [Van Belleghem et al., 97], [Bonatti, ASP03]

  Keep definition of some predicates open

- **Open Answer Set Programs** [Heymans & Vermeir, ASP03;DEXA03]

  relax domain closure: add countably many anonymous elements

  decidability via restrictions (e.g., conceptual LPs)

- **ASP-EX** [Calimeri & Ianni LPNMR05]

  External sources of computation; impose safety

- **HEX-Programs** [E_ et al. IJCAI05]

  HiLog ASP plus external sources

- **ONTODLV** [Dell'Armi et al., ASP07]

  ASP + non-standard ontologies + OWL interface

- Several extensions to combine rules and ontologies

Expressing RDF/S in ASP

- Common: use a predicate `triple/3`:

```
triple("http://www.polleres.net/foaf.rdf#me",
       "foaf:name", "Axel Polleres").
```

- RDFS semantics can be fully captured by DLV-EX, DLV-HEX [Ianni et al., ASP07]

  E.g., subclass relation:

```
[...]
triple(S,rdf:type,C2) :- triple(S,rdf:type,C1),
                         triple(C1,rdfs:subClassOf,C2).
```

  map triples to predicate notation (using higher-order features) and to frame constructs

```
P(S,O) :- triple(S,P,O).
S : C  :- triple(S,rdf:type,C).
S : C2 :- S : C1, rdfs:subClassOf(C1,C2).
```

- Technicalities for treating blank nodes and infinite axiomatic triples.

Expressing Ontologies in ASP/2

- ASP techniques for encoding OWL / DLs have been considered e.g. by
  - Alsac and Baral [2002]
  - Swift [LPNMR04]
  - Hustadt, Motik, and Sattler [KR04]
  - Heymans et al. [2003+]
- Works only under limitations
- Noticable: Hustadt et al.'s mapping of $\mathcal{SHIQ}$ to disjunctive datalog; basis for the KAON2 DL solver
- **Main problem:** important differences between ASP and OWL/DLs

LP vs. Classical Logic

- ASP and OWL/DLs have related yet different underlying settings
- At the heart, the difference is between LP and Classical logic
- **Main Differences:**
  - Closed vs. Open World Assumption
  - Negation as failure vs. classical negation
  - Strong negation vs. classical negation
  - Unique names, equality

LP / Classical Logic: CWA vs. Open World Assumption (OWA)

- LP aims at building a single model, by closing the world

Reiter's CWA:

If $T \not\models A$, *then conclude* $\neg A$, for ground atom A

- FO logic / description logics keep the world open
- In the Semantic Web, this is often reasonable
- However, taking the agnostic stance of OWA may be not helpful for drawing rational conclusions under incomplete information
- A mix of CWA and OWA may be appropriate [Damasio et al., PPSWR06], [Polleres et al., EWSC06]

LP / Classical Logic: NAF vs. Classical Negation

$$
\begin{array}{ll}
P: & wine(X) \leftarrow whiteWine(X). \\
& nonWhite(X) \leftarrow not \;\; whiteWine(X). \\
& wine(myDrink).
\end{array}
$$

$$
\begin{array}{ll}
\mathcal{T}: & \forall X.\,(WhiteWine(X) \supset Wine(X)) \wedge \\
& \forall X.\,(\neg WhiteWine(X) \supset NonWhite(X)) \wedge \\
& Wine(myDrink).
\end{array}
$$

- Query $nonWhite(myDrink)$?
  - Conclude $nonWhite(myDrink)$ from P.
  - Do not conclude $nonWhite(myDrink)$ from T.

LP / Classical Logic: Strong vs. Classical Negation

$$\begin{array}{ll} P: & wine(X) \leftarrow whiteWine(X). \\ & -wine(myDrink). \\ \mathcal{T}: & \forall X.\,(WhiteWine(X) \supset Wine(X)) \land \\ & \neg Wine(myDrink). \end{array}$$

- Conclude $\neg WhiteWine(myDrink)$ from T;
- Do not conclude $-whiteWine(myDrink)$ from P
- Note: no contraposition in LP!

$$-whiteWine(X) \leftarrow -wine(X).$$

is not equivalent to

$$wine(X) \leftarrow whiteWine(X).$$

LP / Classical Logic: Unique Names, Equality

- In LP, usally we have *Unique Names Assumption (UNA)*: *Syntactically different ground terms are different objects.*
- Thus, usually only Herbrand interpretations are considered in LP
- Ontology languages like OWL don't make UNA, and allow to link objects (owl:sameAs)
- OWL considers also non-Herbrand interpretations
- Further, related problems with existential quantifiers:

$$\mathcal{T}: \quad \forall X \exists Y. (Wine(X) \supset hasColor(X, Y))$$

(in DL Syntax, $Wine \sqsubseteq \exists hasColor$)

Simple skolemization does not work in general

Generic Settings for KB Combination

A hybrid knowledge base $\mathcal{KB} = \langle \mathcal{T}, P \rangle$ has

- a FO theory $\mathcal{T}$ (the *classical component*)
- an LP P (the *rules component*)

- Predicates are either *"classical"* or "*rules*" predicates
- Occurrence of rules-predicates in a FO theory is usually restricted, function symbols are disallowed.

  Main reason: Combinations of Horn logic and very simple DLs are undecidable [Levy & Rousset, AIJ98].

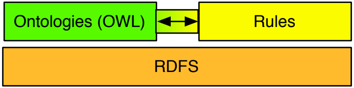
  - Problems with recursion and *unsafety* of rules
  - L&R used *role-safety*: at least one of X, Y in a role atom $R(X, Y)$ in a rule r occurs in rules-predicate in r not occurring in any rule head.

Generic Settings for KB Combination

- Safety (in several versions) is a key tool to decidability for several combination approaches.
- Different approaches to a semantics for $\mathcal{KB} = \langle \mathcal{T}, P \rangle$
  - Strict semantic separation (loose coupling)
  - Tight integration
  - Full integration
- Surveys and Discussion
  - KNOWLEDGEWEB [Pan et al., 2004]
  - REWERSE [Antoniou et al., 2005]
  - Rosati, E_ et al. [ReasoningWeb2006]

KB Combination: Loose Coupling

- **Strict semantic separation between rules / ontology**



  - View rule base and FO theory as separate components
  - They are connected through a minimal interface for exchanging knowledge (formulas).

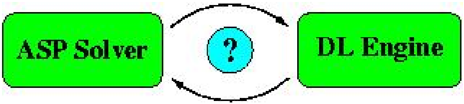
"safe interface"

Examples: TRIPLE [Sintek & Decker, 02]; non-monotonic dl-programs [E_ et al., 04]; defeasible logic+DLs [Antoniou et al., 04]

Loose Coupling: Non-monotonic dl-Programs

- An extension of answer set programs with *queries to DL knowledge bases* (through dl-*atoms*)
- dl-atoms allow to query a DL knowledge base differently

  bidirectional flow of information, with clean technical separation of DL engine and ASP solver



- Use dl-programs as "glue" for combining inferences on a DL base.

DL-atoms

Basic Idea:

- Query the DL base $\mathcal{T}$ using the query interface of the DL engine

  Query Q may be concept/role instance $C(X)/R(X,Y)$, subsumption $C \sqsubseteq D$ etc (recent extension: conjunctive queries)

- Allow to modify the extensional part (ABox) of $\mathcal{T}$, by adding positive ($\uplus$) or negative ($\cup\!\!\!\!-$) assertions
- Q evaluates to true iff the modified $\mathcal{T}$ proves Q.

Examples: wine ontology

- $DL[Wine](\text{"}ChiantiClassico\text{"})$
- $DL[Wine](X)$
- $DL[RedWine \uplus my\_red; Wine](X)$

  add all assertions $RedWine(c)$ to $\mathcal{T}$, such that $my\_red(c)$ holds.

- $DL[RedWine \cup\!\!\!\!- my\_white; hasColor](X, \text{"}Red\text{"})$

  add all assertions $\neg RedWine(c)$ to $\mathcal{T}$, such that $my\_white(c)$ holds.

dl-Programs: Answer Sets

Answer Sets:

- As usual, via grounding of the rules
- Consider a consistent set of ground literals $M \subseteq HB_{P^*}$ ($P^* = P$ with additional constants from $\mathcal{T}$)
- A ground dl-atom $DL[\langle Add\rangle; Q](\mathbf{c})$ is true in M, iff $\mathcal{T} \cup \langle Add\rangle^M \models Q(\mathbf{c})$ for the modification $\langle Add\rangle^M$
- Use a reduct P^M akin to the Gelfond-Lifschitz reduct
- In building P^M, treat dl-atoms like ordinary atoms
- M is an answer set iff M is the least (resp. a minimal) model of P^M

Variants by different treatment of the dl-atoms.

Example: Wine Selection

$is\_redWine(\text{"}Chianti\_21\text{"}).$

$person(\text{"}axel\text{"}).\quad my\_Wine(\text{"}axel\text{"}, \text{"}whiteWine\text{"}).$

$person(\text{"}gibbi\text{"}).\quad my\_Wine(\text{"}gibbi\text{"}, \text{"}redWine\text{"}).$

$wineBottle(X) \leftarrow DL[\langle Add\rangle; \text{"}Wine\text{"}](X).$

$$ok\_Bottle(X,Z) \leftarrow my\_Wine(X, \text{"}RedWine\text{"}), wineBottle(Z), DL[\langle Add\rangle; \text{"}RedWine\text{"}](Z).$$

$$ok\_Bottle(X,Z) \leftarrow my\_Wine(X, \text{"}WhiteWine\text{"}), wineBottle(Z), DL[\langle Add\rangle; \text{"}WhiteWine\text{"}](Z).$$

$$takeBottle(X) \mid -takeBottle(X) \leftarrow wineBottle(X).$$

$$hasBottle(X) \leftarrow takeBottle(X), ok\_Bottle(X,Z).$$

$$\leftarrow person(X), not\ \ hasBottle(X).$$

- Answer sets depend on the instances of *Wine*, *RedWine*, *WhiteWine*

  E.g., { $wineBottle(\text{"}SelaksIceWine\text{"}), wineBottle(\text{"}TaylorPort\text{"}), \ldots,$
  $ok\_Bottle(\text{"}axel\text{"}, \text{"}SelaksIceWine\text{"}), ok\_Bottle(\text{"}gibbi\text{"}, \text{"}TaylorPort\text{"}), \ldots,$
  $takeBottle(\text{"}SelaksIceWine\text{"}), takeBottle(\text{"}TaylorPort\text{"}), \ldots$ }

- Add knowledge $is\_redWine(\text{"}Chianti\_21\text{"})$ to $\mathcal{T}$: $\langle Add\rangle = RedWine \uplus is\_redWine$

  $wineBottle(\text{"}Chianti\_21\text{"})$ is in all answer sets

Example: Mutual Flow

- Existing network, encoded in OWL KB
- New nodes to be added

  Condition: don't connect to nodes with high traffic (traffic depends on number of connections)

- Specify models for possibilities

$$connect(X, Y) \leftarrow newNode(X), DL[Node](Y), not\ overloaded(Y).$$
$$overloaded(X) \leftarrow DL[wired \uplus connect; HighTrafficNode](X).$$

- Mutual effects between the LP and the ontology!

dl-Programs: Applications

- dl-programs facilitate some advanced reasoning tasks
  - **Closed World Reasoning**

    Emulate CWA and *Extended CWA (ECWA)* on top of a DL knowledge base.
  - **Default Reasoning**

    Poole's-style and Reiter's Default Logic over DL bases (for restricted fragments)

    E.g., network example: connect new node X by default to node Y.
  - **Minimal Model Reasoning**

    Single out "minimal" models of a DL base

dl-Programs: Minimal Models

- A DL base with disjunctive information:

$$\mathcal{T} \quad = \quad \{\; \mathit{Artist}(\text{“}\mathit{Jody}\text{”}), \quad \mathit{Artist} \equiv \mathit{Painter} \sqcup \mathit{Singer} \;\}$$

- Single out “minimal” models (in the setting of Extended CWA):

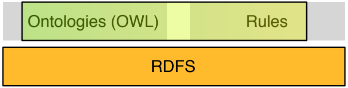
$$\begin{aligned} \overline{p}(X) &\leftarrow \mathit{not}\;\; p(X). \\ \overline{s}(X) &\leftarrow \mathit{not}\;\; s(X). \\ p(X) &\leftarrow \mathit{DL}[\mathit{Painter} \uplus \overline{p}, \mathit{Singer} \uplus \overline{s}; \mathit{Painter}](X). \\ s(X) &\leftarrow \mathit{DL}[\mathit{Painter} \uplus \overline{p}, \mathit{Singer} \uplus \overline{s}; \mathit{Singer}](X). \end{aligned}$$

Answer sets (corresponding to minimal models):

$$M_1 \;=\; \{p(\text{“}\mathit{Jody}\text{”}), \overline{s}(\text{“}\mathit{Jody}\text{”})\}, \qquad M_2 \;=\; \{s(\text{“}\mathit{Jody}\text{”}), \overline{p}(\text{“}\mathit{Jody}\text{”})\}$$

- Extendible to keep concepts “fixed” $\leadsto$ ECWA(ϕ; P; Q; Z)

KB Combination: Tight semantic integration



- Integrate FOL statements and the logic program to a large extent, but keep predicates separate.
- Build an integrated model as the union of two models, one for the FO theory and one for the rules, which share the same domain.

  "safe interaction"

Examples: CARIN [Levy & Rousset, 98]; DLP [Grosof et al., 03]; SWRL [Horrocks et al., 04]; DL-safe rules [Motik etal., 05]; R-hybrid KBs, R^+-hybrid KBs, $\mathcal{DL}$+*log* [Rosati, DL99;05] [Rosati, PPSWR05;KR06]

Tight integration: $\mathcal{DL}$+*log*

[Rosati, KR06] Latest in chain of extensions of the DL $\mathcal{ALC}$ with rules [$\mathcal{AL}$-*Log*; R-, R$^+$-hybrid KBs]

- Fixed countably infinite domain, *standard names* for the elements.
- Models of $\mathcal{KB} = \langle \mathcal{T}, P \rangle$ are of form $\mathcal{I} \cup M$, where $\mathcal{I}$ is a model of the classical predicates, M of the rules-predicates
- No strong negation, weak negation limited to rules-predicates
- Uses *weak (DL-)safety*: each variable X occurs in some positive body atom, which is a rules atom if X occurs in the head.
- Weak safety allows to *access unnamed individuals* in classical atoms (not possible in dl-programs)

$$girl(X) \leftarrow kid(X), father(Y, X), not\ likes(X, cars);$$

- Decidable, if certain union of conjunctive queries containment (CQ/UCQ) in $\mathcal{T}$ is decidable

$\mathcal{DL}$+*log*: Semantics

- $\mathcal{DL}$+*log* has a stable model (answer set) semantics
- Roughly, a 2-step reduction
- **Step 1:** Take some interpretation $\mathcal{I}$ of the classical predicates.
  - Ground *P* and "reduce" it wrt. $\mathcal{I}$, by "evaluating" classical atoms in rules wrt. $\mathcal{I}$.
  - The resulting ground program $P_{\mathcal{I}}$ contains no classical predicates.
- **Step 2:** Build a stable model *M* of $P_{\mathcal{I}}$ as usual

$\mathcal{DL}$+*log*: Example

$$\mathcal{T} = \{\ \mathit{Female} \sqsubseteq \neg \mathit{Male};\quad \mathit{Female} \sqcup \mathit{Male} \sqsubseteq \mathit{Person}$$
$$\mathit{Person} \sqsubseteq \exists \mathit{father}^{-}.\mathit{Male};\quad \mathit{Person}(\mathit{joey})\ \}$$

$$P = \{\ \mathit{boy}(X) \mid \mathit{girl}(X) \leftarrow \mathit{kid}(X);$$
$$\mathit{Male}(X) \leftarrow \mathit{boy}(X);$$
$$\mathit{Female}(X) \leftarrow \mathit{girl}(X);$$
$$\mathit{girl}(X) \leftarrow \mathit{kid}(X), \mathit{father}(Y, X), \mathit{not}\ \mathit{likes}(X, \mathit{cars});$$
$$\mathit{kid}(\mathit{joey})\ \}$$

- Classical predicates *Male, Female, father* occur in rules.
- Take $\mathcal{I}$ where *joey* belongs to *Female*, *Person*:

  $M = \{\mathit{kid}(\mathit{joey}), \mathit{girl}(\mathit{joey})\}$ is a stable model

  $M = \{\mathit{kid}(\mathit{joey}), \mathit{boy}(\mathit{joey})\}$ is not a stable model
- Take $\mathcal{I}$ where *joey* belongs to *Male*, *Person*:

  No stable model (in any such M, $\mathit{likes}(\mathit{joey}, \mathit{cars})$ must be false)

KB Combination: Full integration

- No separation between vocabularies

  Examples:

  - Hybrid MKNF knowledge bases [Motik & Rosati, IJCAI07];
  - G-hybrid knowledge bases [Heymans et al., ALPSWS06]
  - Quantified Equilibrium Logic [Pearce et al., RR07]
  - Autoepistemic Logic [de Bruijn et al., IJCAI07]
  - Open Answer Set Programs [Heymans et al., 03+]

- Related:
  - Terminological Default Logic [Baader and Hollunder, 1995]
  - Description Logics of Minimal Knowledge [Donini et al., 2002]

Full Integration: Quantified Equilibrium Logic (QEL)

- Builds on FOL version of Equilibrium Logic (EL) [Pearce, 96;06]
- Reconstructs Answer Sets in terms of the standard, nonclassical Logic of here & there (resp. 3-valued Gödel Logic).
- Models $\mathcal{M} = \langle (D, \sigma), I_h, I_t \rangle$ of the func.-free FO Logic **QHT**^{s} correspond to intuitionistic Kripke models for two worlds, $h \leq t$, with domain D.
- Axiomatization by first-order intuitionistic logic plus further axioms

equality axioms	$x = x,\ x = y \rightarrow (F(x) \rightarrow F(y))$ (y substitutable for x in formula $F(X)$)
Hosoi's axiom	$\alpha \vee (\neg\beta \vee (\alpha \rightarrow \beta))$
SQHT	$\exists x(F(x) \rightarrow \forall x F(x))$
DE	$x = y \vee x \neq y$

- A model $\mathcal{M}$ of $\mathcal{T}$ is *in equilibrium*, if $I_h = I_t$ and I_h is minimal, i.e., can not be decreased.

QEL /2

- QEL provides a useful logical foundation for different combination approaches:
  - The equilibrium models of a logic program P correspond to Heymans et al.'s [ALPSWS2006] generalised open answer sets of P.
  - Models of $\mathcal{KB} = \langle \mathcal{T}, P \rangle$ in several other combination approaches (G-hybrid, R-hybrid, R^+-hybrid KBs) can be elegantly captured
  - For that, take $\mathcal{T} \cup P \cup st(\mathcal{T})$, where

    $$st(\mathcal{T}) = \{\forall x p(x) \vee \neg p(x) \mid p \in \Sigma_{\mathcal{T}}\}.$$

    Note: $st(\mathcal{T}) \models \neg\neg\phi \rightarrow \phi$, for each $\phi \in \mathcal{T}$.
  - QEL has further usages (e.g., *strong equivalence* of hybrid KBs)

Full Integration: Hybrid MKNF

- Builds on a FO-version of Lifschitz's modal logic MKNF [IJCAI91]
- Rules of form

$$\mathbf{K}h_1 \vee \cdots \vee \mathbf{K}h_l \leftarrow \mathbf{K}b_1, \ldots \mathbf{K}b_m, \mathit{not}\ \ b_{m+1}, \ldots, \mathit{not}\ \ b_n$$

  where the h_i, b_j are function-free FO atoms
- $\mathbf{K}\phi \approx \phi$ is known to hold under the values of the *not* -atoms, Kripke-based semantics ("maximal" S5-models wrt. **K**)
- Faithfully extends LP and DL; generalizes CARIN, $\mathcal{AL}$-*Log*
- Allows "closed world glasses" on classical predicates, stating exceptions
- Extensions with both modal and non-modal atoms in rules allow to generalize $\mathcal{DL}$+*log* (equi-satisfiable)
- Decidable under DL-safety: each variable occurs in some positive rule atom in the body.

Hybrid MKNF: Example

$$\mathcal{T} = \{\ \mathit{Female} \sqsubseteq \neg \mathit{Male};\quad \mathit{Female} \sqcup \mathit{Male} \sqsubseteq \mathit{Person}$$
$$\mathit{Person} \sqsubseteq \exists \mathit{father}^{-}.\mathit{Male};\quad \mathit{Person}(\mathit{joey});\quad \mathit{Diabetic} \sqsubseteq \mathit{Person}\ \}$$

$$P = \{\ \mathbf{K}\mathit{boy}(X) \vee \mathbf{K}\mathit{girl}(X) \leftarrow \mathbf{K}\mathit{kid}(X);$$
$$\mathbf{K}\mathit{Male}(X) \leftarrow \mathbf{K}\mathit{boy}(X);$$
$$\mathbf{K}\mathit{Female}(X) \leftarrow \mathbf{K}\mathit{girl}(X);$$
$$\mathbf{K}\mathit{girl}(X) \leftarrow \mathbf{K}\mathit{kid}(X), \mathbf{K}\mathit{father}(Y, X), \mathit{not}\ \mathit{likes}(X, \mathit{cars});$$
$$\mathbf{K}\mathit{kid}(\mathit{joey});$$
$$\mathbf{K}\mathit{offer}(X, \mathit{cake}) \leftarrow \mathbf{K}\mathit{Person}(X), \mathit{not}\ \mathit{Diabetic}(X)\qquad \}$$

- Offer cake to persons, with exception of diabetics
- Assume that, by default, a person is not suffering from diabetes
- Not expressible in $\mathcal{DL}$+*log* in this way

Assessment

	dl-programs	$\mathcal{DL}$+*log*	QEL	hybrid MKNF
Distinguish classical and rule predicates	+	+	–	–
Domain of Discourse for *P*				
Herbrand Universe of *P*	–	+/–	–	+
Combined Signature	+	+/–	–	+
Arbitrary domains	–	–	+	–
Uniqueness of Names				
unique names in HU of *P*	+	+/–[1]	–	+/–[1]
Special equality predicate	–[2]	–[2]	–	≈
No uniqueness	–	+/–	+	+/–
Interaction from FO Theories to Rules				
Single models	–	+	+	+
Multiple models / Entailment	+	–	–	–
Interaction from Rules to FO Theories				
Single models	–	+	+	–
Multiple models / Entailment	+	–	–	+
Decidability	+[3]	+[4]	–	+[3]

1 The setting (standard names, Herbrand interpretations) implies the UNA.

2 Extendible with axioms for a congruence relation.

3 Essentially, satisfiability for the underlying DL must be decidable.

Systems / Implementations of Combinations

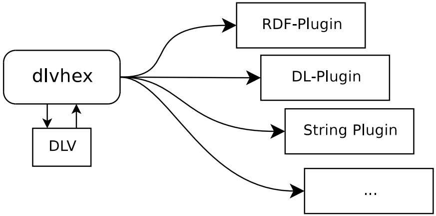
- NLP-DL implements dl-programs (prototype)

  `http://con.fusion.at/nlpdl/`

- DLVHEX implements fragments of HEX-programs (prototype)

  `http://con.fusion.at/dlvhex/`

  Feature: external software *plugins*



  Used for various applications (ontology merging, bio-ontologies, e-government, web querying, policy management, . . .)

- ...

Further Aspects

- **Probabilistic ASP for SW** [Lukasiewicz, ESQUARU05;07]
- **Fuzzy ASP for SW** [Lukasiewicz, RuleML06], [Lukasiewicz & Straccia, RR07]
- **Rules plus RDF/S:**

  Stable model theory for extended RDF/S [Analyti et al., ISWC05]
- **Well-founded semantics**

  [E_ et al., RuleML04], [Drabent & Maluszynski; RR07] [Knorr et al., DL07]

Conclusion

- ASP has potential for Semantic Web, for different uses
- Promising applications and usages (e.g., SPARQL, TRIPLESTORES)
- **Pro's and Con's**
  + Declarativity
  + Expressiveness
  + Rich suite of constructs and extensions
  – Mismatch to standard ontologies
  – Scalability (for some tasks)
  – Less commercialized

Research Issues

- **Semantics for rules plus ontologies?**

  Only small scale KBs; larger examples / use cases are missing
- **Computational and semantic properties**
  - expressiveness
- **Efficient implementations, algorithms**
  - scalable complexity
- **Knowledge combination/integration beyond ontologies**

  E.g., with rule bases under different semantics, cf. [Baral et al., RuleML06]
- **Language extensions of ASP**

  E.g. need for function symbols, value invention (increasing domains)
- **Further applications**

Further Material

More in depth tutorials on “Answer Set Programming for the Semantic Web,” covering different aspects and providing hand-on examples:

- Tutorial at ESWC 2006, available at
  `http://rease.semanticweb.org/`,
  `http://asptut.gibbi.com/`,
  `http://con.fusion.at/asptut`
- Tutorial at ESTC 2007
- Tutorial at “Giornata GULP: Applications of LP and related CL paradigms” March 9th, 2007 (slides are available on the Web)

Background: `https://www.mat.unical.it/ianni/wiki/Answer_Set_Programming_for_the_Semantic_Web`

Bibliography

G. Alsaç and C. Baral.
Reasoning in description logics using declarative logic programming.
Technical report, Department of Computer Science and Engineering, Arizona State University, 2001.

A. Analyti, G. Antoniou, C. V. Damásio, and G. Wagner.
Stable model theory for extended RDF ontologies.
In *Proc. ISWC-2005*, volume 3729 of *LNCS*, pp. 21–36. Springer, 2005.

G. Antoniou.
Nonmonotonic rule systems on top of ontology layers.
In *Proc. ISWC-2002*, volume 2342 of *LNCS*, pp. 394–398. Springer, 2002.

G. Antoniou, C. V. Damásio, B. Grosof, I. Horrocks, M. Kifer, J. Maluszynski, and P. F. Patel-Schneider.
Combining rules and ontologies: A survey.
Technical Report IST506779/Linköping/I3-D3/D/PU/a1, Linköping University, 2005.

F. Baader and B. Hollunder.
Embedding defaults into terminological knowledge representation formalisms.
J. Autom. Reasoning, 14(1):149–180, 1995.

S. Baselice, P. A. Bonatti, and G. Criscuolo.
On finitely recursive programs.
In *Proc. ICLP 2007*, volume 4670 of *Lecture Notes in Computer Science*, pp. 89–103. Springer, 2007.

P. A. Bonatti.
Finitary open logic program.
In *Proc. of the 2nd Intl. Workshop on Answer Set Programming (ASP)*, vol. 78 of *CEUR Workshop Proc.*

P. A. Bonatti.

Reasoning with infinite stable models.
Artificial Intelligence, 156(1):75–111, 2004.

F. Calimeri and G. Ianni.

External sources of computation for answer set solvers.
In *LPNMR*, volume 3662 of *Lecture Notes in Computer Science*, pp. 105–118. Springer, 2005.

C. V. Damásio, A. Analyti, G. Antoniou, and G. Wagner.

Supporting open and closed world reasoning on the web.
In J. J. Alferes, et al., editors, *PPSWR*, volume 4187 of *LNCS*, pp. 149–163. Springer, 2006.

J. de Bruijn, T. Eiter, A. Polleres, and H. Tompits.

On representational issues about combinations of classical theories with nonmonotonic rules.
In *Proc. KSEM-2006*, volume 4092 of *LNCS/LNAI*, pp. 1–22. Springer, 2006.

J. de Bruijn, T. Eiter, A. Polleres, and H. Tompits.

Embedding non-ground logic programs into autoepistemic logic for knowledge base combination.
In M. Veloso, editor, *Proc. 20th International Joint Con. on Artificial Intelligence* (*IJCAI 2007*), pp. 304–309. AAAI Press/IJCAI, 2007.

J. de Bruijn, D. Pearce, A. Polleres, and A. Valverde.

A logic for hybrid rules.
In *Proc. Second International Conf. on Rules and Rule Markup Languages for the Semantic Web* (*RuleML 2006*). IEEE, 2006.

T. Dell'Armi, L. Gallucci, N. Leone, F. Ricca, and R. Schindlauer.

ONTODLV: an ASP-based system for enterprise ontologies.
In *Proceedings ASP-2007*, pp. 99–113, 2007.

J. de Bruijn, D. Pearce, A. Polleres, and A. Valverde.

Quantified equilibrium logic and hybrid rules.
In M. Marchiori et al., editors, Proc. *RR 2007*, volume 4524 of *LNCS*, pp. 58–72. Springer, 2007.

F. M. Donini, D. Nardi, and R. Rosati.
Description logics of minimal knowledge and negation as failure.
ACM Trans. Comput. Log., 3(2):177–225, 2002.

W. Drabent and J. Maluszynski.
Well-founded semantics for hybrid rules.
In M. Marchiori et al., editors, Proc. *RR 2007*, volume 4524 of *LNCS*, pp. 1–15.

T. Eiter, G. Ianni, A. Polleres, R. Schindlauer, and H. Tompits.
Reasoning with rules and ontologies.
In P. Barahona, et al., editors, *Reasoning Web, Second International Summer School 2006*, number 4126 in LNCS, pp. 93–127. Springer, 2006.

T. Eiter, G. Ianni, R. Schindlauer, and H. Tompits.
A uniform integration of higher-order reasoning and external evaluations in answer-set programming.
In *Proceedings IJCAI-2005*, pp. 90–96. Professional Book Center, 2005.

T. Eiter, T. Lukasiewicz, R. Schindlauer, and H. Tompits.
Combining Answer Set Programming with Description Logics for the Semantic Web.
In D. Dubois, et al., editors, *Proc. Ninth International Conf. on Principles of Knowledge Representation and Reasoning (KR 2004)*, pp. 141–151. Morgan Kaufmann, 2004. Extended paper Tech. Rep. INFSYS RR-1843-07-04, Inst. Inf.Sys., TU Vienna, January 2007.

T. Eiter, T. Lukasiewicz, R. Schindlauer, and H. Tompits.
Well-founded semantics for description logic programs in the Semantic Web.
In *Proceedings RuleML-2004*, volume 3323 of *LNCS*, pp. 81–97. Springer, 2004.

M. Gelfond and V. Lifschitz.
Classical negation in logic programs and disjunctive databases.
New Generation Computing, 9:365–385, 1991.

B. N. Grosof, I. Horrocks, R. Volz, and S. Decker.
Description logic programs: Combining logic programs with description logics.
In *Proc. WWW-2003*, pp. 48–57. ACM Press, 2003.

S. Heymans, L. Predoiu, C. Feier, J. de Bruijn, and D. van Nieuwenborgh.
G-hybrid knowledge bases.
In *Proc. Workshop on Applications of Logic Programming in the Semantic Web and Semantic Web Services (ALPSWS 2006), at FLOC/ICLP 2006*, pp. 39–54, 2006. Online `http://CEUR-WS.org/Vol-196/`.

S. Heymans, D. V. Nieuwenborgh, and D. Vermeir.
Nonmonotonic ontological and rule-based reasoning with extended conceptual logic programs.
In *Proc. ESWC-2005*, volume 3532 of *LNCS*, pp. 392–407. Springer, 2005.

S. Heymans, D. Van Nieuwenborgh, and D. Vermeir.
Conceptual logic programs.
Annals of Mathematics and Artificial Intelligence, 47(1-2):103–137, 2006.

S. Heymans, D. Van Nieuwenborgh, and D. Vermeir.
Open answer set programming for the semantic web.
Journal of Applied Logic, 5(1):144–169, 2007.

S. Heymans and D. Vermeir.
Integrating ontology languages and answer set programming.
In *DEXA Workshops*, pp. 584–588. IEEE Computer Society, 2003.

S. Heymans and D. Vermeir.
Integrating semantic web reasoning and answer set programming.
In *Proc. ASP-2003*, pp. 194–208, 2003.

I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean.
SWRL: A Semantic Web rule language combining OWL and RuleML, May 2004.
W3C Member Submission. Available at `http://www.w3.org/Submission/SWRL/`.

U. Hustadt, B. Motik, and U. Sattler.
Reducing SHIQ-description logic to disjunctive datalog programs.
In *Proceedings KR-2004*, pp. 152–162. AAAI Press, 2004.

G. Ianni, A. Martello, C. Panetta, and G. Terracina.
Faithful and effective querying of RDF ontologies using DLV^{db}.
In *Proceedings ASP-2007*, pp. 115–131, 2007.

M. Knorr, J. J. Alferes, and P. Hitzler.
A well-founded semantics for hybrid MKNF knowledge bases.
In *Proc. 2007 International Workshop on Description Logics (DL2007), Bressanone, 2007*, CEUR Workshop Proc., 2007.

A. Y. Levy and M.-C. Rousset.
Combining Horn rules and description logics in CARIN.
Artificial Intelligence, 104(1–2):165–209, 1998.

V. Lifschitz.
Answer Set Programming and Plan Generation.
Artificial Intelligence, 138:39–54, 2002.
Seminal paper at ICLP'99 (invited talk), coining the term "Answer Set Programming".

T. Lukasiewicz.
Fuzzy description logic programs under the answer set semantics for the Semantic Web.
In *Proc. RuleML-2006*, pp. 89–96. IEEE Computer Society, 2006.
Full paper in *Fundamenta Informaticae*, to appear.

T. Lukasiewicz.
Probabilistic description logic programs.
Int. J. Approx. Reasoning, 45(2):288–307, 2007.

T. Lukasiewicz and U. Straccia.

Tightly integrated fuzzy description logic programs under the answer set semantics for the semantic web.
In *RR*, pp. 289–298, 2007.

V. W. Marek and M. Truszczyński.

Stable Models and an Alternative Logic Programming Paradigm.
In K. Apt, V. W. Marek, M. Truszczyński, and D. S. Warren, editors, *The Logic Programming Paradigm – A 25-Year Perspective*, pp. 375–398. Springer Verlag, 1999.

B. Motik, I. Horrocks, R. Rosati, and U. Sattler.

Can OWL and logic programming live together happily ever after?
In I. F. Cruz, et al., editors, *Proc. 5th International Semantic Web Conference (ISWC 2006)*, volume 4273 of *Lecture Notes in Computer Science*, pp. 501–514. Springer, 2006.

B. Motik and R. Rosati.

A faithful integration of description logics with logic programming.
In M. Veloso, editor, *Proc. 20th International Joint Conference on Artificial Intelligence* (*IJCAI 2007*), pp. 477–482. AAAI Press/IJCAI, 2007. Extended Paper: Closing Semantic Web Ontologies. Tech. Report, Univ. Manchester, UK, 2006.

B. Motik, U. Sattler, and R. Studer.

Query answering for OWL-DL with rules.
J. Web Sem., 3(1):41–60, 2005.

I. Niemelä.

Logic Programming with Stable Model Semantics as Constraint Programming Paradigm.
Annals of Mathematics and Artificial Intelligence, 25(3–4):241–273, 1999.

Thomas Eiter, G. Ianni, R. Schindlauer, and H. Tompits.

dlvhex: A prover for semantic-web reasoning under the answer-set semantics.
In *IEEE/WIC/ACM Int'l Conf. on Web Intelligence (WI 2006)*, pp. 1073–74. IEEE Computer Society, 2006. More extend version at ALPSWS 2006.

J. Z. Pan, E. Franconi, S. Tessaris, G. Stamou, V. Tzouvaras, L. Serafini, I. Horrocks, and B. Glimm.
Specification of coordination of rule and ontology languages.
Project Deliverable D2.5.1, KnowledgeWeb NoE, June 2004.

A. Polleres. From SPARQL to rules (and back).
In *Proceedings of the 16th International Conference on World Wide Web (WWW)*, pp. 787–796. ACM, 2007.

A. Polleres, C. Feier, and A. Harth.
Rules with contextually scoped negation.
In *Proceedings ESWC-2006*, volume 4011 of *LNCS*, pp. 332–347. Springer, 2006.

A. Polleres and R. Schindlauer.
dlvhex-sparql: A SPARQL-compliant query engine based on dlvhex.
In *Proc. 2nd Workshop on Applications of Logic Programming in the Semantic Web and Semantic Web Services (ALPSWS 2007), at ICLP 2007*, pp. 3–11, 2007.

E. Pontelli, T. C. Son, and C. Baral.
A framework for composition and inter-operation of rules in the semantic web.
In *RuleML*, pp. 39–50. IEEE Computer Society, 2006.

A. Rainer.
Web Service Composition under Answer Set Programming.
KI-Workshop “Planen, Scheduling und Konfigurieren, Entwerfen” (PuK), at KI 2005. Winner of the first prize in the [WWW] EEE-Web 2005 Service Composition Contest.

R. Reiter. On closed world data bases.
In H. Gallaire and J. Minker, editors, *Logic and Data Bases*, pp. 55–76. Plenum Press, New York, 1978.

R. Rosati.
Towards expressive KR systems integrating datalog and description logics: Preliminary report.
In *Proc. DL-1999*, pp. 160–164, 1999.

R. Rosati.
On the decidability and complexity of integrating ontologies and rules.
J. Web Sem., 3(1):61–73, 2005.

R. Rosati. Semantic and computational advantages of the safe integration of ontologies and rules.
In *Proc. PPSWR-2005*, volume 3703 of *LNCS*, pp. 50–64. Springer, 2005.

R. Rosati.
Integrating ontologies and rules: Semantic and computational issues.
In *Reasoning Web, Second Int'l Summer School 2006*, volume 4126 of *LNCS*, pp. 128–151. Springer, 2006.

R. Rosati.
$\mathcal{DL}$+*log*: Tight integration of description logics and disjunctive datalog.
In *Proc. 10th Int'l Conf. on Principles of Knowledge Representation and Reasoning* (*KR 2006*), pp. 68–78. AAAI Press, 2006.

M. Šimkus and T. Eiter.
FDNC: Decidable non-monotonic disjunctive logic programs with function symbols.
In *Proceedings 14th Int'l Conf. on Logic for Programming, Artificial Intelligence and Reasoning (LPAR 2007)*, LNCS, pp. 514–530. Springer, 2007.

M. Sintek and S. Decker.
TRIPLE - A query, inference, and transformation language for the Semantic Web.
In *Proc. ISWC-2002*, volume 2342 of *LNCS*, pp. 364–378. Springer, 2002.

T. Swift. Deduction in ontologies via ASP.
In *Proceedings LPNMR-2004*, volume 2923 of *LNCS/LNAI*, pp. 275–288. Springer, 2004.

T. Syrjänen. Omega-restricted logic programs.
In *Proc. LPNMR 2001*, number 2173 in LNAI, pp. 267–279. Springer Verlag, 2001.

K. Van Belleghem, M. Denecker, and D. De Schreye.
A strong correspondence between description logics and open logic programming.
In *ICLP*, pp. 346–360, 1997.

K. Wang, D. Billington, J. Blee, and G. Antoniou.
Combining description logic and defeasible logic for the Semantic Web.
In *Proceedings RuleML-2004*, number 3323 in LNCS, pp. 170–181. Springer, 2004.