The limits and possibilities of combining Description Logics and Datalog

Riccardo Rosati

Dipartimento di Informatica e Sistemistica

Università di Roma “La Sapienza”

rosati@dis.uniroma1.it

RuleML 2006, Athens, GA, November 2006

Outline

- ontologies and Description Logics (DLs)
- rules, Datalog, and Disjunctive Datalog
- combining DLs and rules: semantic and computational issues
- the decidability issue
- loose integration
- new results: computational analysis for nonrecursive Datalog rules
- open problems

Ontologies, Description Logics, and rules

- **ontology**: central notion for the Semantic Web
- **Description Logics** (DLs) are currently playing a prominent role as ontology formalisms
- recent interest in combining DLs and **rules**
- from the KR viewpoint, DLs and rules are complementary
- rules add expressive power to DLs (and vice versa)
- but: many problems to face when adding rules to DLs

Description Logics

- **Description Logics** (DLs) are logics that represent the domain of interest in terms of **concepts**, denoting sets of objects, and **roles**, denoting binary relations between (instances of) concepts
- Complex concept and role expressions are constructed starting from a set of atomic concepts and roles by applying suitable constructs
- **DLs = fragments of function-free first-order logic**
- as an example of a DL, in the following we formally introduce $\mathcal{ALCI}$

Description Logics: syntax

In $\mathcal{ALCI}$, concepts and roles are formed according to the following syntax:

$$C \quad ::= \quad \top \mid A \mid C_1 \sqcap C_2 \mid C_1 \sqcup C_2 \mid \neg C \mid \exists R.C \mid \forall R.C$$

$$R \quad ::= \quad P \mid P^-$$

where A denotes an atomic concept, P denotes an atomic role

DL **knowledge base** (KB) $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, where:

- **TBox** $\mathcal{T}$ (intensional knowledge) = set of **inclusion assertions**

$$C_1 \sqsubseteq C_2$$

- **ABox** $\mathcal{A}$ (extensional knowledge) = set of **membership assertions**

$$C(a), \qquad R(a, b)$$

Description Logics: semantics

a **DL-interpretation** $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ is a standard FOL interpretation of concepts, roles, and constants, such that

$$
\begin{array}{rcl}
\top^{\mathcal{I}} & = & \Delta^{\mathcal{I}} \\
A^{\mathcal{I}} & \subseteq & \Delta^{\mathcal{I}} \\
P^{\mathcal{I}} & \subseteq & \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \\
\neg C^{\mathcal{I}} & = & \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}} \\
C_1 \sqcap C_2^{\mathcal{I}} & = & C_1^{\mathcal{I}} \cap C_2^{\mathcal{I}} \\
\exists P.C^{\mathcal{I}} & = & \{d \in \Delta^{\mathcal{I}} \mid \exists d'.(d, d') \in P^{\mathcal{I}} \text{ and } d' \in C^{\mathcal{I}}\} \\
\exists P^{-}.C^{\mathcal{I}} & = & \{d \in \Delta^{\mathcal{I}} \mid \exists d'.(d', d) \in P^{\mathcal{I}} \text{ and } d' \in C^{\mathcal{I}}\}
\end{array}
$$

- $\mathcal{I}$ is a **model of** $C_1 \sqsubseteq C_2$ if $C_1^{\mathcal{I}} \subseteq C_2^{\mathcal{I}}$
- $\mathcal{I}$ is a **model of** $C(a)$ ($P(a,b)$) if $a^{\mathcal{I}} \in C^{\mathcal{I}}$ ($(a^{\mathcal{I}}, b^{\mathcal{I}}) \in P^{\mathcal{I}}$)
- $\mathcal{I}$ is a **model of** $(\mathcal{T}, \mathcal{A})$ if $\mathcal{I}$ is a model of all assertions in $\mathcal{T}$ and $\mathcal{A}$

FOL reading of DLs

- $\mathcal{ALCI}$ (and, in practice, almost every DL) is basically a fragment of function-free FOL with a variable-free syntax
- a DL KB $\mathcal{K}$ is semantically equivalent to a FOL theory *FO*$(\mathcal{K})$ in which each assertion in the KB is expressed by a first-order sentence
- for instance, the TBox inclusion assertion

$$A_1 \sqcap \exists P_1.A_2 \sqsubseteq (\forall P_2.A_3) \sqcup \neg A_4$$

is equivalent to the first-order sentence

$$\forall x.A_1(x) \wedge (\exists y.P_1(x,y) \wedge A_2(y)) \rightarrow (\forall z.P_2(x,z) \rightarrow A_3(z)) \vee \neg A_4(x)$$

Description Logics: example

let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be an ontology about persons where:

- $\mathcal{T}$ contains the following inclusion assertions:

  $\mathit{PERSON} \sqsubseteq \exists \mathit{FATHER}^{-}.\mathit{MALE}$

  $\mathit{MALE} \sqsubseteq \mathit{PERSON}$

  $\mathit{FEMALE} \sqsubseteq \mathit{PERSON}$

  $\mathit{FEMALE} \sqsubseteq \neg \mathit{MALE}$

- $\mathcal{A}$ contains the following instance assertions:

  $\mathit{MALE}(\mathit{Bob})$

  $\mathit{PERSON}(\mathit{Mary})$

  $\mathit{PERSON}(\mathit{Paul})$

Description Logics: example

let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be an ontology about persons where:

- $\mathcal{T}$ corresponds to the following FOL sentences:

  $\forall x.\mathit{PERSON}(x) \rightarrow \exists y.\mathit{FATHER}(y, x) \wedge \mathit{MALE}(y)$

  $\forall x.\mathit{MALE}(x) \rightarrow \mathit{PERSON}(x)$

  $\forall x.\mathit{FEMALE}(x) \rightarrow \mathit{PERSON}(x)$

  $\forall x.\mathit{FEMALE}(x) \rightarrow \neg\mathit{MALE}(x)$

- $\mathcal{A}$ contains the following instance assertions:

  $\mathit{MALE}(\mathit{Bob})$

  $\mathit{PERSON}(\mathit{Mary})$

  $\mathit{PERSON}(\mathit{Paul})$

Description Logics as ontology modeling languages

- OWL family = ontology modeling languages for the Semantic Web
- the OWL family is based on Description Logics
- each OWL variant inherits the pros and cons of DLs
- the experience in building practical applications has revealed several shortcomings of OWL and, in general, of Description Logics

Expressive limitations of Description Logics

the typical expressiveness of DLs does not allow for addressing the following aspects:

- defining **predicates of arbitrary arity** (not just unary and binary)
- using **variable quantification** beyond the tree-like structure of DL concepts (many DLs are subsets of the two-variable fragment of FOL)
- formulating **expressive queries** over DL knowledge bases (beyond concept subsumption and instance checking)
- formalizing various **forms of closed-world reasoning** over DL KBs
- more generally, expressing forms of **nonmonotonic knowledge**, like default rules

Rules

- **rule-based formalisms** grounded in **logic programming** have repeatedly been proposed as a possible solution to overcome the above limitations
- adding a rule layer on top of OWL is nowadays seen as the most important task in the development of the Semantic Web language stack
- the Rule Interchange Format (RIF) working group of the World Wide Web Consortium (W3C) is currently working on standardizing such a language
- most of the proposals in this field focus on logic programs expressed in **Datalog** (and its nonmonotonic extensions)

Positive Datalog

- **atom** = expression of the form $p(t_1, \ldots, t_n)$ with p predicate and each t_i variable or constant (**fact** = atom without occurrences of variables)
- **Datalog rule** R = expression of the form

  $p \leftarrow r_1, \ldots, r_n$

  such that $n \geq 0$, p and all r_i's are atoms and:

  – **(Datalog safeness)** each variable occurring in p must appear in at least one of the atoms $r_1, \ldots, r_n$
- **Datalog program** $\mathcal{P}$ = set of Datalog rules
- EDB($\mathcal{P}$) = set of facts occurring in $\mathcal{P}$
- IDB($\mathcal{P}$) = set of rules occurring in $\mathcal{P}$ = $\mathcal{P}-$ EDB($\mathcal{P}$)

Disjunctive Datalog

- **Disjunctive Datalog rule** R = expression of the form

  $p_1 \ \vee \ldots \vee \ p_n \ \leftarrow r_1, \ \ldots, \ r_m, \ not\, s_1, \ \ldots, \ not\, s_k$

  such that $n \geq 0$, $m \geq 0$, $k \geq 0$, each p_i, r_i, s_i is an atom and:

  – **(Datalog safeness)** each variable of R must appear in at least one of the atoms $r_1, \ldots, r_m$

- **Disjunctive Datalog program** $\mathcal{P}$ = set of Disjunctive Datalog rules

- EDB($\mathcal{P}$) = set of facts occurring in $\mathcal{P}$

- IDB($\mathcal{P}$) = set of rules occurring in $\mathcal{P}$ = $\mathcal{P}-$ EDB($\mathcal{P}$)

Semantics of positive Disjunctive Datalog

- an interpretation $\mathcal{I}$ **satisfies** a positive Disjunctive Datalog rule

  $p_1 \ \vee \ldots \vee \ p_n \ \leftarrow \ r_1, \ \ldots, \ r_m$

  if $\mathcal{I}$ satisfies the FOL sentence

  $\forall \vec{\mathbf{x}}.r_1 \wedge \ldots \wedge r_m \rightarrow p_1 \vee \ldots \vee p_n$

  where $\vec{\mathbf{x}}$ are all the variables occurring in R

- $\mathcal{I}$ is a **model** of a Datalog program $\mathcal{P}$ if $\mathcal{I}$ satisfies each rule in $\mathcal{P}$

- $\mathcal{I}' \subseteq \mathcal{I}$ if for each predicate p, $p^{\mathcal{I}'} \subseteq p^{\mathcal{I}}$

- $\mathcal{I}$ is a **minimal model** for $\mathcal{P}$ if there exists no model $\mathcal{I}'$ of $\mathcal{P}$ such that $\mathcal{I}' \subseteq \mathcal{I}$

Stable model semantics of Disjunctive Datalog

- R' is a **ground instantiation** of a rule $R \in \mathcal{P}$ if R' is the rule obtained from R by replacing each variable with a constant occurring in $\mathcal{P}$
- *ground*($\mathcal{P}$) = ground instantiation of $\mathcal{P}$ = $\{R' \mid R \in \mathcal{P}$ and R' is a ground instantiation of R in $\mathcal{P}\}$
- $\mathcal{P}/\mathcal{I}$ = **GL-reduct** of a ground Disjunctive Datalog program $\mathcal{P}$ wrt $\mathcal{I}$ = ground positive Disjunctive Datalog program obtained from $\mathcal{P}$ by:
  1. deleting each rule containing a negated fact $not\ p$ such that $\mathcal{I} \models p$
  2. deleting each negated fact $not\ p$ such that $\mathcal{I} \not\models p$
- $\mathcal{I}$ is a **stable model** of $\mathcal{P}$ if $\mathcal{I}$ is a minimal model of *ground*$(\mathcal{P})/\mathcal{I}$

Integrating DLs and rules

first approach: **do not really integrate**

- **no (or very loose) integration**:
  - rules independent of DLs (“two towers”)
  - rules “on top” of DLs
- **get rid of DLs**:
  use rules to express the ontology (ontology = set of rules)
- **get rid of logic programs**:
  FOL interpretation of rules (rules = FOL implications)
- **be (very) politically correct**:
  take only the “intersection” of DLs and rules (DLP)

underlying “political” issue: **DLs vs. logic programming**

Integrating DLs and rules

second approach: **do some “real” integration**

- semantic issues
- computational issues

main semantic issue:

Open-World Assumption vs. Closed-World Assumption

main computational issue:

decidability (and complexity) preservation

Combining DLs and rules: Example

let $\mathcal{H} = (\mathcal{K}, \mathcal{P})$, where $\mathcal{K}$ = ontology about persons:

$PERSON \sqsubseteq \exists FATHER^{-}.MALE$

$MALE \sqsubseteq PERSON$

$FEMALE \sqsubseteq PERSON$

$FEMALE \sqsubseteq \neg MALE$

$MALE(Bob) \quad PERSON(Mary) \quad PERSON(Paul)$

$\mathcal{P}$ = nonmonotonic rules about students:

$boy(X) \leftarrow enrolled(X, c1), PERSON(X), not\ girl(X). \quad [R1]$

$girl(X) \leftarrow enrolled(X, c2), PERSON(X). \quad [R2]$

$boy(X) \vee girl(X) \leftarrow enrolled(X, c3), PERSON(X). \quad [R3]$

$FEMALE(X) \leftarrow girl(X). \quad [R4]$

$MALE(X) \leftarrow boy(X). \quad [R5]$

$enrolled(Paul, c1). \ \ enrolled(Mary, c2). \ \ enrolled(Bob, c3).$

Combining DLs and rules: Example

let $\mathcal{H} = (\mathcal{K}, \mathcal{P})$, where $\mathcal{K}$ = ontology about persons:

$\forall x.\mathit{PERSON}(x) \rightarrow \exists y.\mathit{FATHER}(y, x) \wedge \mathit{MALE}(y)$

$\forall x.\mathit{MALE}(x) \rightarrow \mathit{PERSON}(x)$

$\forall x.\mathit{FEMALE}(x) \rightarrow \mathit{PERSON}(x)$

$\forall x.\mathit{FEMALE}(x) \rightarrow \neg \mathit{MALE}(x)$

$\mathit{MALE}(\mathit{Bob}) \quad \mathit{PERSON}(\mathit{Mary}) \quad \mathit{PERSON}(\mathit{Paul})$

$\mathcal{P}$ = nonmonotonic rules about students:

$\mathit{boy}(X) \leftarrow \mathit{enrolled}(X, c1), \mathit{PERSON}(X), \mathit{not}\ \mathit{girl}(X). \quad [R1]$

$\mathit{girl}(X) \leftarrow \mathit{enrolled}(X, c2), \mathit{PERSON}(X). \quad [R2]$

$\mathit{boy}(X) \vee \mathit{girl}(X) \leftarrow \mathit{enrolled}(X, c3), \mathit{PERSON}(X). \quad [R3]$

$\mathit{FEMALE}(X) \leftarrow \mathit{girl}(X). \quad [R4]$

$\mathit{MALE}(X) \leftarrow \mathit{boy}(X). \quad [R5]$

$\mathit{enrolled}(\mathit{Paul}, c1). \ \mathit{enrolled}(\mathit{Mary}, c2). \ \mathit{enrolled}(\mathit{Bob}, c3).$

Example (continued)

the above KB $\mathcal{H}$ should intuitively entail:

– *boy*$($*Paul*$)$, since rule R1 is always applicable for $X =$ *Paul* and R1 acts like a **default rule**: if X is a person enrolled in course $c1$, then X is a boy, unless we know for sure that X is a girl

– *girl*$($*Mary*$)$, since rule R2 is always applicable for $X =$ *Mary*

– *boy*$($*Bob*$)$, since rule R3 is always applicable for $X =$ *Bob*, and, by rule R4, the conclusion *girl*$($*Bob*$)$ is inconsistent with $\mathcal{K}$

– *MALE*$($*Paul*$)$ (due to rule R5) and *FEMALE*$($*Mary*$)$ (due to rule R4)

Rules as queries: Example

let $\mathcal{H} = (\mathcal{K}, \mathcal{P})$, where:

– $\mathcal{K}$ is an ontology defining the concept *KNOWS* and such that

$$\mathcal{K} \models_{FOL} \textit{KNOWS}(\textit{Pat}, \textit{Joe})$$
$$\mathcal{K} \models_{FOL} \neg\textit{KNOWS}(\textit{Pat}, \textit{Ann})$$
$$\mathcal{K} \models_{FOL} \neg\textit{KNOWS}(\textit{Pat}, \textit{Bob})$$

($\textit{Pat}, \textit{Joe}, \textit{Ann}, \textit{Bob}$ are the only constants occurring in $\mathcal{K}$)

– $\mathcal{P}$ is the following set of (non-recursive) rules:

$$\textit{knows-many}(X) \leftarrow \textit{person}(X), \textit{person}(Y), \textit{person}(Z),$$
$$\textit{KNOWS}(X, Y), \textit{KNOWS}(X, Z),\ X \neq Y,\ X \neq Z,\ Y \neq Z.$$

$$\textit{knows-exactly-one}(X) \leftarrow$$
$$\textit{person}(X), \textit{person}(Y), \textit{KNOWS}(X, Y), not\ \textit{knows-many}(X).$$

$$\textit{person}(\textit{Joe}). \quad \textit{person}(\textit{Pat}). \quad \textit{person}(\textit{Paul}). \quad \textit{person}(\textit{Mary}).$$

– then, $\mathcal{H} \models_{\textit{NM}} \textit{knows-exactly-one}(\textit{Pat})$

$\Rightarrow$ rules in $\mathcal{DL}$-*log* KBs can encode nonmonotonic queries over ontologies

Semantic issue: OWA vs. CWA

- DLs are fragments of first-order logic (FOL), hence their semantics are based on the **Open World Assumption** (OWA) of classical logic
- rules are based on a **Closed World Assumption** (CWA), imposed by the different semantics for logic programming and deductive databases (which formalize in various ways the notion of information closure)
- **how to integrate the OWA of DLs and the CWA of rules in a “proper” way?**
- i.e., how to merge monotonic and nonmonotonic components from a semantic viewpoint?

Computational issue: Decidability of DLs + rules

- decidability (and complexity) of reasoning is a crucial issue in systems combining DL KBs and Datalog rules
- in general **this combination does not preserve decidability**:
  - starting from a DL KB in which reasoning is decidable and a rule KB in which reasoning is decidable, reasoning in the integrated KB may not be a decidable problem [Halevy & Rousset, 1996]
  - e.g.: $\mathcal{ALC}$ + positive Datalog (under FOL semantics)
- **lack of a thorough analysis of decidability and complexity of the combination of DLs and Datalog rules**

Undecidability results

for **positive recursive** Datalog rules:

- [Halevy & Rousset, 1996]:

  undecidability of DL + positive recursive rules when DL allows the concepts $\forall R.C$ or $(\leq n\, R)$ or $\exists R.C$

  $\Rightarrow$ **adding arbitrary recursive Datalog rules to even very simple DL KBs causes undecidability**

- [Calvanese & Rosati, 2003]:

  strengthen the above results to unary inclusion dependencies (and unary keys and foreign keys)

- [Horrocks & Patel-Schneider, 2004]:

  analogous results in the framework of SWRL

Tight vs. loose integration of DLs and rules

two ways to overcome the computational problems:

- **restrict the expressiveness** of the DL component and/or the rule component
- **restrict the interaction** between the two components

most approaches restrict the interaction between the DL KB and the rule KB

$\Rightarrow$ **loose integration** between DLs and rules

for nonmonotonic Datalog rules, restricting the interaction with DLs also helps in solving the semantic discrepancy (OWA vs. CWA)

Loose interaction through variable safeness

- basic idea: control the interaction between rules and DL KB through a syntactic extra **safeness condition** on variables in rules
- several such notions of safeness have been proposed
- e.g., we recall the **DL-safeness** condition:

  each rule variable must appear in an atom whose predicate does not occur in the DL KB
- originally proposed in $\mathcal{AL}$-log [Donini et al., 1991], then assumed in [Rosati, DL-WS 1999; Motik, Sattler, Studer, ISWC 2004; Rosati, 2005]

– interaction between rules and DLs is limited

+ solves the above semantic and computational problems!

The DL-safeness condition

- two disjoint predicate alphabets:
  - $\mathcal{A}_P$: **DL-predicates** (concept and role names) interpreted under OWA
  - $\mathcal{A}_R$: **Datalog predicates** interpreted under CWA
- **DL-safe Datalog$^{\neg\vee}$ rule** = rule of the form

  $p_1(X_1) \vee \ldots \vee p_n(X_n) \leftarrow$

  $r_1(Y_1), \ldots, r_m(Y_m), s_1(Z_1), \ldots, s_k(Z_k), \mathit{not}\, u_1(W_1), \ldots, \mathit{not}\, u_h(W_h)$

  such that

  - each p_i is a predicate from $\mathcal{A}_P \cup \mathcal{A}_R$
  - each r_i, u_i is a predicate from $\mathcal{A}_R$ (Datalog predicate)
  - each s_i is a predicate from $\mathcal{A}_P$ (DL-predicate)
  - **DL-safeness: each rule variable must occur in one of the r_i's**

The DL-safeness condition

Example:

if the KB $\mathcal{H}$ is such that

- the Datalog predicates are p, q
- the DL predicates are C, R, S

then:

- the rule $q(X) \leftarrow p(X,Y), C(X), R(X,Y), S(Y,X)$ is DL-safe
- the rule $q(X) \leftarrow p(X,Y), C(X), R(X,Z), S(Z,W)$ is **not** DL-safe

Decidability results

for **positive Datalog** rules:

- **AL-log** [Donini et al., 1990]:
  decidability of $\mathcal{ALC}$ plus positive, recursive **DL-safe** rules
- **CARIN** [Halevy & Rousset, 1996]:
  decidability of $\mathcal{ALCNR}$ plus:
  - positive, **nonrecursive** rules
  - positive, recursive, **role-safe** rules
  - positive, recursive rules and **acyclic TBox inclusions**
- **DL-safe rules** [Motik, Sattler, Studer, 2004]:
  decidability of $\mathcal{SHOIN}$ plus positive, recursive **DL-safe** rules

Decidability results

for **nonmonotonic Datalog** rules:

- **DL-log / safe hybrid KBs** [Rosati 1999, Rosati 2005]:

  decidability of (decidable) DLs/FOL plus nonmonotonic, recursive **DL-safe** rules

- **DL-programs** [Eiter et al, 2004]:

  decidability of $\mathcal{SHOIN}(\mathbf{D})$ plus **DL-rules**

- $\mathcal{DL}$***+log*** [Rosati 2006]:

  decidability of arbitrary DLs plus nonmonotonic, recursive **weakly DL-safe** rules

- **hybrid MKNF KBs** [Horrocks, Motik, Rosati, Sattler 2006] [Motik, Rosati 2007]:

  mixes open-world and closed-world reasoning in DL-safe rules

Weakly DL-safe rules

- DL-safeness can be weakened without losing its nice computational properties
- **weak DL-safe rule**: a Datalog rule where DL-safeness condition is imposed **only on the head variables** of the rule [Rosati 2006]
- for instance, if the only Datalog predicate is q, the rule

$$q \leftarrow C(X), R(X, Y), S(Y, X)$$

  is weakly DL-safe, but not DL-safe
- weakly DL-safe rules are a maximal fragment of disjunctive Datalog whose combination with DLs is currently known to be decidable

Further analysis: nonrecursive Datalog

we have refined the above analysis to classes of **nonrecursive Datalog** programs:

- ***NR-Datalog*** = nonrecursive (positive) Datalog
- ***NR-Datalog***$_s^{\neq}$ = single-rule nonrecursive Datalog with inequality
- ***NR-Datalog***$^{\neq}$ = nonrecursive Datalog with inequality
- ***NR-Datalog***$_s^{\neg}$ = single-rule nonrecursive Datalog with negation
- ***NR-Datalog***$^{\neg edb}$ = nonrecursive Datalog with EDB negation (i.e., negated predicates must be EDB predicates)
- ***NR-Datalog***$^{\neg}$ = nonrecursive Datalog with negation

Semantics of DLs with nonrecursive Datalog programs

FOL semantics:

- the FOL semantics of the rule

  $$p_1(X_1) \vee \ldots \vee p_n(X_n) \leftarrow$$
  $$r_1(Y_1), \ldots, r_m(Y_m), s_1(Z_1), \ldots, s_k(Z_k), not\, u_1(W_1), \ldots, not\, u_h(W_h)$$

  is given by the first-order sentence

  $$\forall \overline{x}_1, \ldots, \overline{x}_n, \overline{y}_1, \ldots, \overline{y}_m, \overline{z}_1, \ldots, \overline{z}_k, \overline{w}_1, \ldots, \overline{w}_h.$$
  $$r_1(\overline{y}_1) \wedge \ldots \wedge r_m(\overline{y}_m) \wedge s_1(\overline{z}_1) \wedge \ldots \wedge s_k(\overline{z}_k) \wedge$$
  $$\neg u_1(\overline{w}_1) \wedge \ldots \wedge \neg u_h(\overline{w}_h) \rightarrow p_1(\overline{x}_1) \vee \ldots \vee p_n(\overline{x}_n)$$

- a **FOL-model** of a KB $\mathcal{H} = (\mathcal{K}, \mathcal{P})$ is a FOL interpretation that satisfies the theory *FO*$(\mathcal{K}) \cup$ *FO*$(\mathcal{P})$

DLs considered

- ***DL-Lite**$_{RDFS}$* is the DL whose language for concept and role expressions is:

  $$C_L ::= A \mid \exists R \qquad C_R ::= A \qquad R ::= P \mid P^-$$

  and both concept inclusions $C_L \sqsubseteq C_R$ and role inclusions are allowed in the TBox

- ***DL-Lite**$_R$* is the DL whose language for concept and role expressions is:

  $$C_L ::= A \mid \exists R \qquad C_R ::= A \mid \neg C_R \mid \exists R \qquad R ::= P \mid P^-$$

  and both concept inclusions $C_L \sqsubseteq C_R$ and role inclusions are allowed in the TBox

DLs considered

- $\mathcal{EL}$ is the DL whose language for concept expressions is:

$$C ::= A \mid C_1 \sqcap C_2 \mid \exists P.C$$

  and only concept inclusions are allowed in the TBox

- $\mathcal{AL}$ is the DL whose language for concept expressions is:

$$C ::= A \mid \top \mid \bot \mid \neg A \mid C_1 \sqcap C_2 \mid \exists P \mid \forall P.C$$

  and only concept inclusions are allowed in the TBox

- $\mathcal{DLR}$: very expressive DL with n-ary relations
- $\mathcal{L}^2$: the two-variable fragment of FOL

New results for DLs with nonrecursive Datalog

	NR-Datalog	NR-Datalog$_s^{\neq}$	NR-Datalog$^{\neq}$	NR-Datalog$_s^{\neg}$	NR-Datalog$^{\neg edb}$	NR-Datalog$^{\neg}$
DL-Lite$_{RDFS}$	$\leq$LOGSPACE	$\leq$LOGSPACE	$\leq$LOGSPACE	$\leq$LOGSPACE	$=$ coNP	UNDECID.
DL-Lite$_R$	$\leq$LOGSPACE	$\geq$coNP	UNDECID.	$\geq$coNP	UNDECID.	UNDECID.
$\mathcal{EL}$	$=$ PTIME	$=$ PTIME	UNDECID.	$=$ PTIME	UNDECID.	UNDECID.
from $\mathcal{AL}$ to $\mathcal{ALCHIQ}$	$=$ coNP	UNDECID.	UNDECID.	UNDECID.	UNDECID.	UNDECID.
$\mathcal{DLR}$	$\geq$ coNP	UNDECID.	UNDECID.	UNDECID.	UNDECID.	UNDECID.
$\mathcal{L}^2$	UNDECID.	UNDECID.	UNDECID.	UNDECID.	UNDECID.	UNDECID.

Consequences of the above results

- as soon as we add **inequality** to positive nonrecursive Datalog, its combination with even very simple DLs becomes undecidable
- as soon as we add **negation** to positive nonrecursive Datalog, its combination with even very simple DLs becomes undecidable
- (analogous negative results hold if we add **disjunction** in the head to positive nonrecursive Datalog)

$\Rightarrow$ **decidability is a real issue even for nonrecursive rules!**

The need of DL-safeness for nonrecursive rules

- the known decidability results show that restricting the interaction between DLs and rules via **safeness conditions on program variables** overcomes the above problem
- such safeness conditions were originally defined to solve the undecidability issue for recursive rules...
- ...and it was generally believed that such conditions could be avoided in nonrecursive rules
- the above undecidability results for nonrecursive Datalog program show that **imposing such safeness conditions is necessary even when rules are nonrecursive**

Summary

- DLs and rules are complementary KR formalisms
- integrating DLs and rules poses both semantic and computational problems
- new results on the decidability/undecidability frontier of DLs and rules...
- ...show that the full integration of DLs and rules is computationally very hard
- DL-safeness is a possible solution
- in particular, weakly DL-safe rules seem a “maximal” class of rules that guarantees general decidability of integration with DLs
- (weakly) DL-safe rules provide a clear formal treatment of CWA

Open problems

- **more refined computational analysis** of reasoning in DLs with rules
- (e.g., tighter forms of decidable interaction between the DL component and the rule component?)
- **semantics** for the integration of DLs and **nonmonotonic** rules
- using **more general logical frameworks** to formally study the integration of DLs and rules: e.g., QEL [de Bruijn, Pearce, Polleres, Valverde 2006], MKNF [Motik, Rosati 2007]
- relationship between rules and queries
- practical algorithms/implementations

Some references

- Alon Halevy and Marie-Christine Rousset. Combining Horn rules and Description Logics in CARIN. *AI Journal*, 1998.
- Francesco M. Donini, Maurizio Lenzerini, Daniele Nardi, Andrea Schaerf. AL-log: Integrating Datalog and Description Logics. *J. Int. Inf. Systems*, 1998.
- Ian Horrocks, Peter F. Patel-Schneider: A proposal for an OWL rules language. WWW 2004.
- Thomas Eiter, Thomas Lukasiewicz, Roman Schindlauer, Hans Tompits. Combining Answer Set Programming with Description Logics for the Semantic Web. KR 2004.
- Boris Motik, Ulrike Sattler, Rudi Studer. Query Answering for OWL-DL with Rules. *Web Semantics Journal*, 2004.
- Riccardo Rosati. Towards expressive KR systems integrating Datalog and Description Logics: preliminary report. Description Logics Workshop, 1999.

Some references (contd.)

- Riccardo Rosati. On the decidability and complexity of integrating ontologies and rules. *Web Semantics Journal*, 2005.
- Riccardo Rosati. $\mathcal{DL}$*+log*: Tight integration of Description Logics and Disjunctive Datalog. KR 2006.
- Ian Horrocks, Boris Motik, Riccardo Rosati, Ulrike Sattler. Can OWL and Logic Programming live together happily ever after? ISWC 2006.
- Thomas Eiter, Giovambattista Ianni, Roman Schindlauer, Hans Tompits. Effective Integration of Declarative Rules with External Evaluations for Semantic-Web Reasoning. ESWC 2006.
- Jos de Bruijn, David Pearce, Axel Polleres, Agustin Valverde. A logic for hybrid rules. RuleML 2006 WS on rule integration.
- Boris Motik, Riccardo Rosati. A faithful integration of Description Logics with Logic Programming. IJCAI 2007, to appear.

THANK YOU!