

How to reason with OWL in a logic programming system



Markus Krötzsch
AIFB Karlsruhe

Denny Vrandecic
AIFB Karlsruhe



Pascal Hitzler
AIFB Karlsruhe

Michael Sintek
DFKI Kaiserslautern



RuleML2006, Athens, GA
November 2006

Semantic Web Languages: Seperate worlds

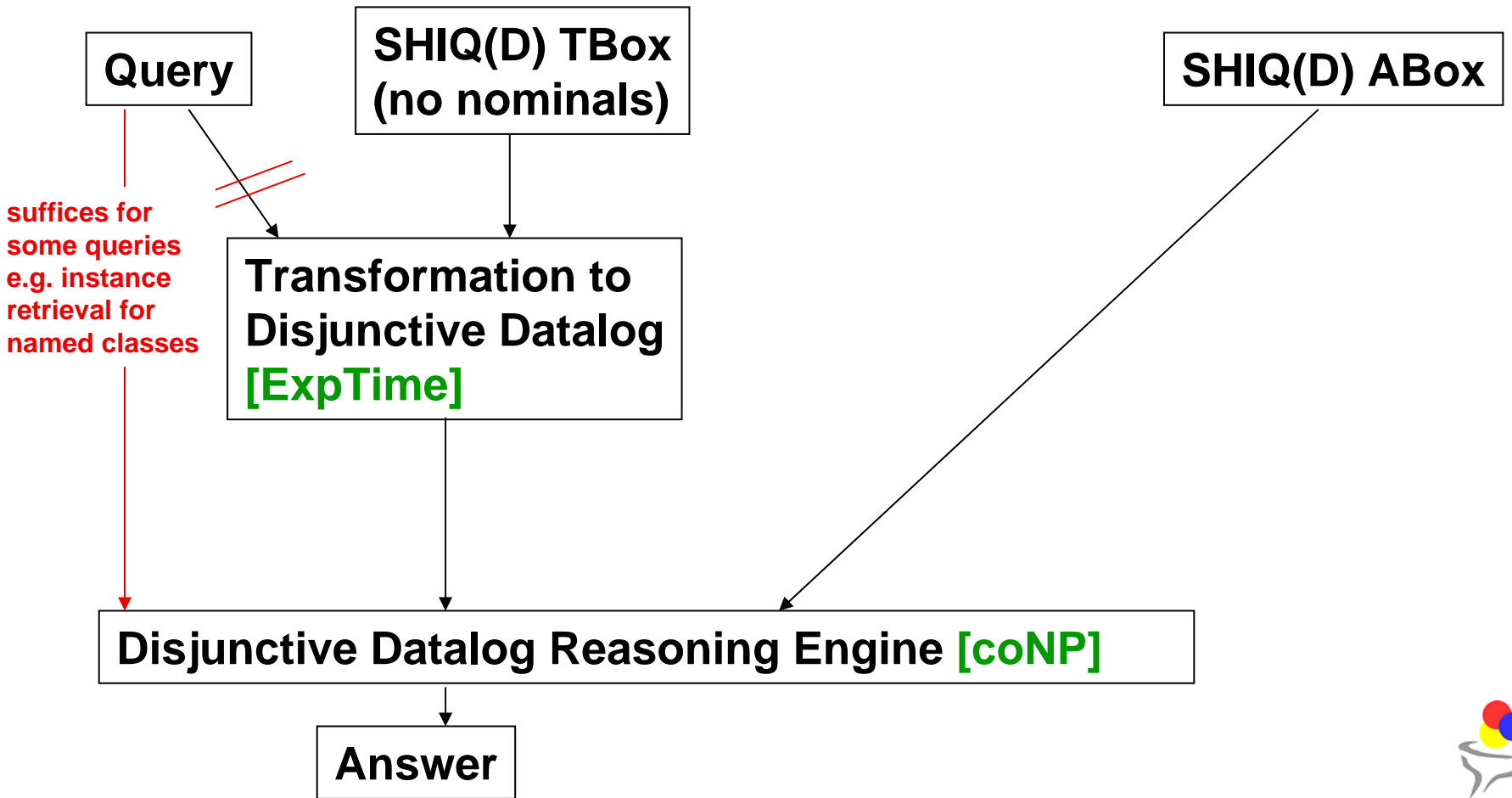
- OWL DL
 - open world
 - monotonic
 - description logics
 - first-order logic
 - decidable
- Logic Programming
 - closed world
 - non-monotonic
 - rules
 - procedural flavour
 - undecidable

- both approaches are needed for applications
- study of interoperability is imperative
- here: sound and complete reasoning for OWL with Prolog

Approach

- We utilize results by Motik et al. on the KAON2 transformation algorithms and system.
- KAON2 OWL reasoner:
<http://kaon2.semanticweb.org>
- KAON2 algorithms comprehensive details:
Boris Motik, Reasoning in Description Logics using Resolution and Deductive Databases. Dissertation, AIFB Universität Karlsruhe, 2006.

KAON2 Reasoner core architecture



Theorem (Hustadt, Motik, Sattler 2004)

Transformation of OWL knowledge base KB
into Disjunctive Datalog DD(KB)

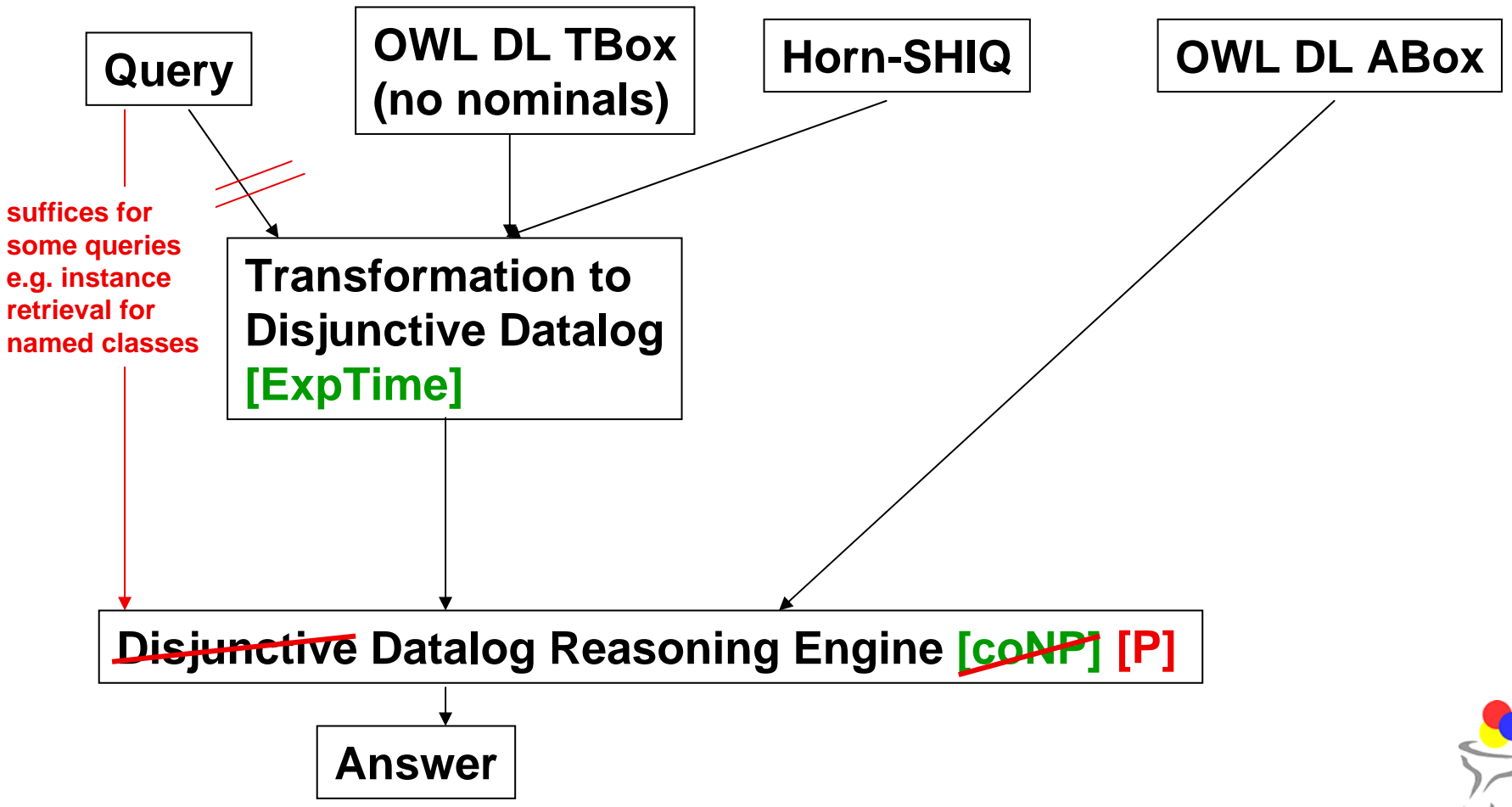
Then, the following hold:

1. KB is unsatisfiable if and only if DD(KB) is unsatisfiable.
2. $KB \models \alpha$ if and only if $DD(KB) \models \alpha$, where α is of the form $A(a)$ or $R(a, b)$, and A is an **atomic concept**.
3. $KB \models C(a)$ for a **nonatomic** concept C if and only if, for Q a new atomic concept, $DD(KB \cup \{C \sqsubseteq Q\}) \models Q(a)$.

Simple example transformation (ALC)

KB
<p>Person \sqsubseteq \exists parent.Person</p> <p>\exists parent.(\exists parent.Person) \sqsubseteq Grandchild</p> <p>Person(a)</p>
DD(KB)
<p>$Q_1(x), \text{Person}(y) \leftarrow \text{parent}(x,y)$</p> <p style="padding-left: 100px;">$\leftarrow \text{parent}(x,y), Q_1(y), \text{Grandchild}(x)$</p> <p style="padding-left: 100px;">$\leftarrow Q_1(x), \text{Person}(x)$</p> <p>$\text{Grandchild}(x) \leftarrow \text{Person}(x)$</p> <p>Person(a)</p>

KAON2 Reasoner core architecture: Horn-SHIQ



Horn-SHIQ

- Fragment of OWL DL
 - Polynomial data complexity (ABox)
 - ExpTime combined complexity (ABox+TBox)
[OWLED06]

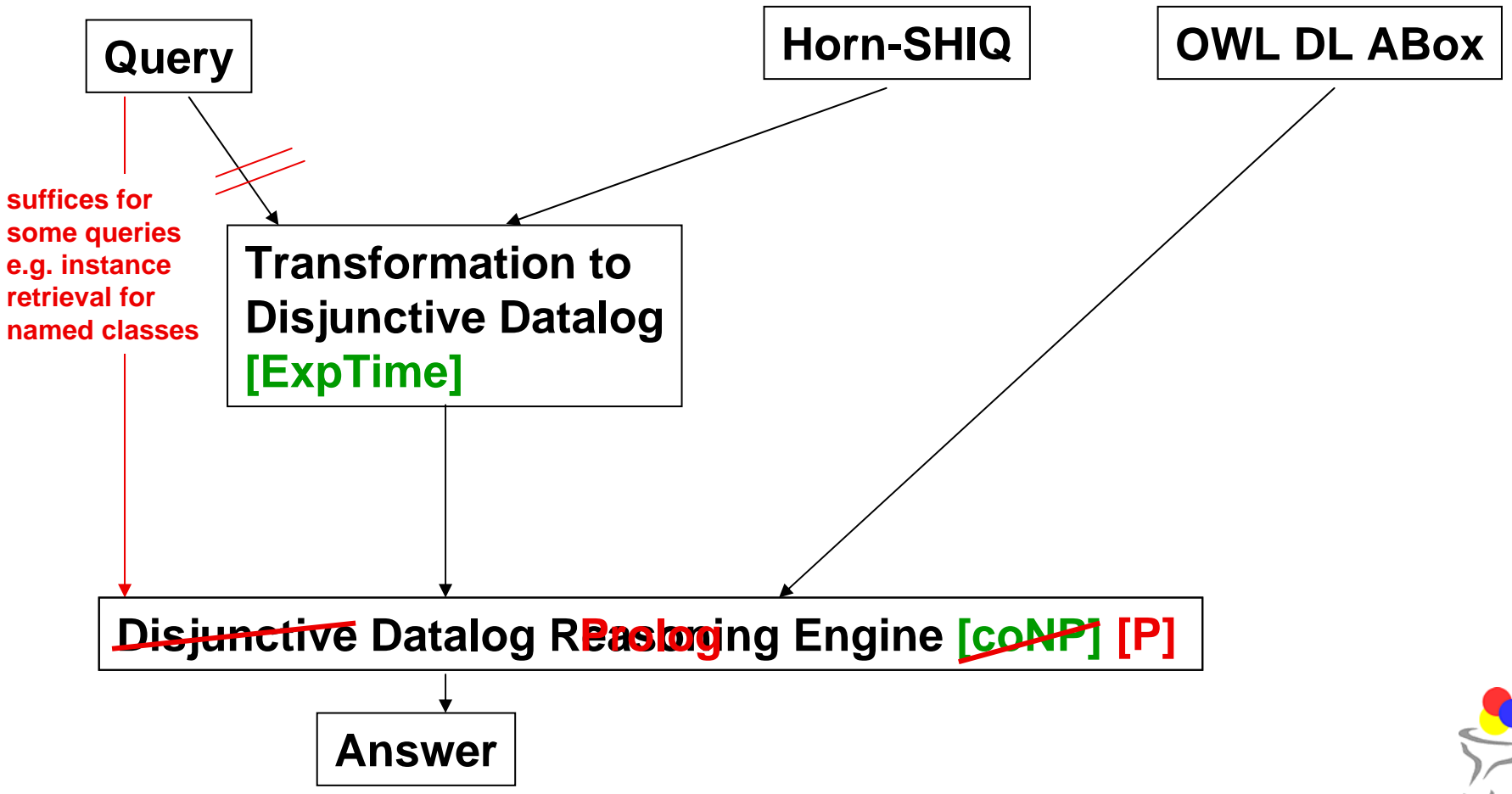
$$C_0^+ \rightarrow \top \mid \perp \mid \neg C_0^- \mid C_0^+ \sqcap C_0^+ \mid C_0^+ \sqcup C_0^+ \mid \forall R.C_0^+$$

$$C_0^- \rightarrow \top \mid \perp \mid \neg C_0^+ \mid C_0^- \sqcap C_0^- \mid C_0^- \sqcup C_0^- \mid \exists R.C_0^- \mid A$$

$$C_1^+ \rightarrow \top \mid \perp \mid \neg C_1^- \mid C_1^+ \sqcap C_1^+ \mid C_0^+ \sqcup C_1^+ \mid \exists R.C_1^+ \mid \forall R.C_1^+ \mid \geq n R.C_1^+ \mid \leq 1 R.C_0^- \mid A$$

$$C_1^- \rightarrow \top \mid \perp \mid \neg C_1^+ \mid C_0^- \sqcap C_1^- \mid C_1^- \sqcup C_1^- \mid \exists R.C_1^- \mid \forall R.C_1^- \mid \geq 2 R.C_0^- \mid \leq n R.C_1^+ \mid A$$

KAON2 Reasoner core architecture: Horn-SHIQ



Difficulty: Integrity constraints

- Some OWL statements become integrity constraints which are not usually supported under Prolog.

- $C \sqcap D \equiv \perp$

translates to

$$\leftarrow C(x) \wedge D(x)$$

- workaround:

$$\text{inc} \leftarrow C(x) \wedge D(x)$$

Difficulty: Equality

- Some OWL statements require equality for expressing them in first-order logic.
- For our purposes, the following Horn rules suffice:

$$\begin{aligned}
 &X \approx X, \quad X \approx Y \leftarrow Y \approx X, \quad X \approx Z \leftarrow X \approx Y \wedge Y \approx Z \\
 &C(Y) \leftarrow C(X) \wedge X \approx Y \quad \text{for every concept name } C \\
 &R(Y_1, Y_2) \leftarrow R(X_1, X_2) \wedge X_1 \approx Y_1 \wedge X_2 \approx Y_2 \quad \text{for every role name } R
 \end{aligned}$$

Example

TBox/RBox

- (1) Parent $\equiv \exists \text{ hasChild}.\top$
- (2) Person $\sqsubseteq \exists \text{ childOf}.\text{Person}$
- (3) ManyChildren $\sqsubseteq \geq 2 \text{ hasChild}.\top$
- (4) NoSiblings $\sqsubseteq \text{Person} \sqcap \forall \text{ childOf}.(\leq 1 \text{ hasChild}.\top)$
- (5) childOf $\equiv \text{hasChild}^{-1}$

ABox

hasChild(Elaine, Sir Lancelot)
noSiblings(Lancelot du Lac)
childOf(Lancelot du Lac, Elaine)

person(X) $:-$ nosiblings(X).	person(X_{f_3}) $:-$ person(X), $S_{f_3}(X, X_{f_3})$.
parent(X) $:-$ haschild(X, Y).	parent(X) $:-$ manychildren(X).
haschild(Y, X) $:-$ childof(X, Y).	haschild(X, X_{f_1}) $:-$ manychildren(X), $S_{f_1}(X, X_{f_1})$.
haschild(X, X_{f_2}) $:-$ parent(X), $S_{f_2}(X, X_{f_2})$.	haschild(X, X_{f_0}) $:-$ manychildren(X), $S_{f_0}(X, X_{f_0})$.
childof(X, X_{f_3}) $:-$ person(X), $S_{f_3}(X, X_{f_3})$.	childof(Y, X) $:-$ haschild(X, Y).

$Y_1 \approx Y_2 :-$ nosiblings(X), childof(X, Z), haschild(Z, Y_1), haschild(Z, Y_2).

inc $:-$ manychildren(X), nosiblings(X_0), childof(X_0, X).

inc $:-$ $X_{f_1} \approx X_{f_0}$, manychildren(X), $S_{f_1}(X, X_{f_1})$, $S_{f_0}(X, X_{f_0})$.

$S_f(X, f(X)) :-$ $O(X)$. $HU(X) :-$ $O(X)$. $HU(f(X)) :-$ $O(X)$. (for $f \in \{f_0, f_1, f_2, f_3\}$)

$X \approx X :-$ $HU(X)$.

$X \approx Y :-$ $Y \approx X, HU(X), HU(Y)$.

$X \approx Z :-$ $X \approx Y, Y \approx Z, HU(X), HU(Y), HU(Z)$.

$C(Y) :-$ $C(X), X \approx Y, HU(X), HU(Y)$.

(for $C \in \{\text{person, parent, manychildren, nosiblings}\}$)

$R(Y_1, Y_2) :-$ $R(X_1, X_2), X_1 \approx Y_1, X_2 \approx Y_2, HU(X_1), HU(X_2), HU(Y_1), HU(X_2)$.

(for $R \in \{\text{childof, haschild}\}$)

$O(\text{Elaine})$. $O(\text{Sir Lancelot})$. $O(\text{Lancelot du Lac})$.

Implementation

Transformation available through KAON2
<http://kaon2.semanticweb.org>

or via owltools command line interface *dlpconvert*
<http://owltools.ontoware.org>
see software demo [OWLED06] this evening

– optional serialisations:

- Prolog
- F-Logic
- RuleML0.9
- SWRL

Acknowledgement

- The presented results are corollaries from the work by Boris Motik on KAON2.
- Very helpful discussions with Boris are gratefully acknowledged.

Thank you!

- related [OWLED06]-presentations:
 - Today, 1345 hrs:
M. Krötzsch, S. Rudolph and P. Hitzler. On the Complexity of Horn Description Logics
 - Today, 1700 hrs:
D. Vrandecic: OWL Tools demo