ELP
Tractable Rules for OWL 2

Markus Krötzsch  Sebastian Rudolph  Pascal Hitzler

Institute AIFB
Universität Karlsruhe (TH)
Germany
Copyright Markus Krötzsch

Terms of Licensing

Creative Commons Attribution

http://creativecommons.org/licenses/by/2.0/de/deed.en

All redistributions and reuses of this work must include the following:

“Based on slides by Markus Krötzsch, http://korrekt.org”

Image copyrights

Icons by http://everaldo.com, released under the terms of GPL
Where do we want to get to?
Where do we want to get to?

Semantic Web language that is ...

• focussed on data,
• rule-oriented,
• able to express schema knowledge,
• easy to implement,
• of polynomial worst-case complexity,
• compatible with OWL.
OWL and Description Logics
OWL and Description Logics

Description logics

FOL
"Sebastian ordered some Thai curry."

$\text{sebastian: } \exists \text{orderedDish}. \text{ThaiCurry}$

$\exists x. \text{orderedDish}(\text{sebastian}, x) \land \text{ThaiCurry}(x)$
OWL and Description Logics

“Everything ordered as a dish is actually a dish.”

\( \top \sqsubseteq \forall \text{orderedDish}.\text{Dish} \)

\( \forall x. \forall y. \text{orderedDish}(x,y) \rightarrow \text{Dish}(y) \)
“Every Thai curry dish contains peanut oil.”

\[ ThaiCurry \equiv \exists \text{contains.}\{\text{peanutOil}\} \]

\[ \forall x. \text{ThaiCurry}(x) \rightarrow \text{contains}(x,\text{peanutOil}) \]
Rules in First-Order Logic
Rules in First-Order Logic
Rules in First-Order Logic

“Nut allergics dislike nut products.”

$\text{NutAllergic}(x) \land \text{NutProduct}(y) \rightarrow \text{dislikes}(x, y)$
Rules in First-Order Logic

“People who order a dish they dislike are unhappy.”

\[ \text{orderedDish}(x,y) \land \text{dislikes}(x,y) \rightarrow \text{Unhappy}(x) \]
Rules in First-Order Logic

“If someone dislikes an ingredient of a dish, she will also dislike the dish.”

\[ \text{dislikes}(x,z) \land \text{Dish}(y) \land \text{contains}(y,z) \rightarrow \text{dislikes}(x,y) \]
Rules in First-Order Logic

“Sebastian is a nut allergic, and peanut oil is a nut product.”

→ NutAllergic(sebastian)

→ NutProduct(peanutOil)
Can we combine datalog rules and DL axioms?
"ThaiCurry \subseteq \exists \text{contains.}\{\text{peanutOil}\} \\
\top \subseteq \forall \text{orderedDish.Dish} \\
\text{sebastian}: \exists \text{orderedDish.ThaiCurry} \\
\text{NutAllergic}(x) \land \text{NutProduct}(y) \rightarrow \text{dislikes}(x,y) \\
\text{orderedDish}(x,y) \land \text{dislikes}(x,y) \rightarrow \text{Unhappy}(x) \\
\text{dislikes}(x,z) \land \text{Dish}(y) \land \text{contains}(y,z) \rightarrow \text{dislikes}(x,y) \\
\rightarrow \text{NutAllergic}(\text{sebastian}) \\
\rightarrow \text{NutProduct}(\text{peanutOil}) ""
Combining OWL and Rules

- Description logics
- DLP
- Datalog
- FOL
DLP: “OWL ∩ datalog”

\[ ThaiCurry \subseteq \exists \text{contains.} \{\text{peanutOil}\} \]

\[ \top \subseteq \forall \text{orderedDish.Dish} \]

\[ \text{sebastian: } \exists \text{orderedDish.ThaiCurry} \]

\[ \text{NutAllergic}(x) \land \text{NutProduct}(y) \rightarrow \text{dislikes}(x, y) \]

\[ \text{orderedDish}(x, y) \land \text{dislikes}(x, y) \rightarrow \text{Unhappy}(x) \]

\[ \text{dislikes}(x, z) \land \text{Dish}(y) \land \text{contains}(y, z) \rightarrow \text{dislikes}(x, y) \]

\[ \rightarrow \text{NutAllergic}(\text{sebastian}) \]

\[ \rightarrow \text{NutProduct}(\text{peanutOil}) \]
**SWRL: “OWL ∪ datalog”**

- \( \text{ThaiCurry} \equiv \exists \text{contains.}\{\text{peanutOil}\} \)
- \( \top \equiv \forall \text{orderedDish.Dish} \)
- \( \text{sebastian: } \exists \text{orderedDish.ThaiCurry} \)
- \( \text{NutAllergic}(x) \land \text{NutProduct}(y) \rightarrow \text{dislikes}(x,y) \)
- \( \text{orderedDish}(x,y) \land \text{dislikes}(x,y) \rightarrow \text{Unhappy}(x) \)
- \( \text{dislikes}(x,z) \land \text{Dish}(y) \land \text{contains}(y,z) \rightarrow \text{dislikes}(x,y) \)
  - \( \rightarrow \text{NutAllergic}(\text{sebastian}) \)
  - \( \rightarrow \text{NutProduct}(\text{peanutOil}) \)
SWRL is undecidable.
DL-safe Rules
DL-safe Rules

Restrict rules to apply only to named individuals.
DL-safe Rules

\( \text{ThaiCurry} \equiv \exists \text{contains.}\{\text{peanutOil}\} \)

\( \top \equiv \forall \text{orderedDish.Dish} \)

\( \text{sebastian: } \exists \text{orderedDish.ThaiCurry} \)

\( \text{NutAllergic}(x) \land \text{NutProduct}(y) \rightarrow \text{dislikes}(x,y) \)

\( \text{orderedDish}(x,y) \land \text{dislikes}(x,y) \rightarrow \text{Unhappy}(x) \)

\( \text{dislikes}(x,z) \land \text{Dish}(y) \land \text{contains}(y,z) \rightarrow \text{dislikes}(x,y) \)

\( \rightarrow \text{NutAllergic}(\text{sebastian}) \)

\( \rightarrow \text{NutProduct}(\text{peanutOil}) \)
DL Rules
DL Rules

Restrict to rules that could (indirectly) be encoded with DL anyway.

*) rules with “tree-shaped” bodies
DL Rules*

\[ \text{dislikes}(x, z) \land \text{Dish}(y) \land \text{contains}(y, z) \rightarrow \text{dislikes}(x, y) \]

*) rules with “tree-shaped” bodies
DL Rules

ThaiCurry ⊆ ∃contains.{peanutOil} ☑

⊤ ⊆ ∀orderedDish.Dish ☑

sebastian: ∃orderedDish.ThaiCurry ☑

NutAllergic(x) ∧ NutProduct(y) → dislikes(x,y) ☑

orderedDish(x,y) ∧ dislikes(x,y) → Unhappy(x) ❌

dislikes(x,z) ∧ Dish(y) ∧ contains(y,z) → dislikes(x,y) ☑

→ NutAllergic(sebastian) ☑

→ NutProduct(peanutOil) ☑
DL-safe rules + DL Rules
DL-safe rules + DL Rules

\[ \text{ThaiCurry} \sqsubseteq \exists \text{contains.}\{\text{peanutOil}\} \]

\[ \top \sqsubseteq \forall \text{orderedDish.Dish} \]

\[ \text{sebastian}: \exists \text{orderedDish.ThaiCurry} \]

\[ \text{NutAllergic}(x) \land \text{NutProduct}(y) \rightarrow \text{dislikes}(x,y) \]

\[ \text{orderedDish}(x,y) \land \text{dislikes}(x,y) \rightarrow \text{Unhappy}(x) \]

\[ \text{dislikes}(x,z) \land \text{Dish}(y) \land \text{contains}(y,z) \rightarrow \text{dislikes}(x,y) \]

\[ \rightarrow \text{NutAllergic(sebastian)} \]

\[ \rightarrow \text{NutProduct(peanutOil)} \]
⚠️ DL-safe rules + DL Rules ⚠️

Desired conclusion does not follow

It is still computationally expensive

DL-safe rules: ExpTime
DL Rules: like DL, i.e. NExpTime for OWL DL
Tractable Profiles in OWL 2
Tractable Profiles in OWL 2

**OWL RL**: *Horn logic* fragment, similar to DLP, no existentials

**OWL EL**: includes existentials, based on DL *EL++*
Regaining Tractability: OWL 2 EL

\[ \text{ThaiCurry} \sqsubseteq \exists \text{contains.}\{\text{peanutOil}\} \checkmark \]

\[ \top \sqsubseteq \forall \text{orderedDish.Dish} \checkmark \]

\[ \text{sebastian}: \exists \text{orderedDish.ThaiCurry} \checkmark \]

\[ \text{NutAllergic}(x) \land \text{NutProduct}(y) \rightarrow \text{dislikes}(x,y) \]

\[ \text{orderedDish}(x,y) \land \text{dislikes}(x,y) \rightarrow \text{Unhappy}(x) \]

\[ \text{dislikes}(x,z) \land \text{Dish}(y) \land \text{contains}(y,z) \rightarrow \text{dislikes}(x,y) \]

\[ \rightarrow \text{NutAllergic}(\text{sebastian}) \checkmark \]

\[ \rightarrow \text{NutProduct}(\text{peanutOil}) \checkmark \]
Regaining Tractability: OWL 2 RL

\( \text{ThaiCurry} \sqsubseteq \exists \text{contains.}\{\text{peanutOil}\} \checkmark \)

\( \top \sqsubseteq \forall \text{orderedDish.Dish} \checkmark \)

\( \text{sebastian}: \exists \text{orderedDish.ThaiCurry} \times \)

\( \text{NutAllergic}(x) \land \text{NutProduct}(y) \rightarrow \text{dislikes}(x,y) \times \)

\( \text{orderedDish}(x,y) \land \text{dislikes}(x,y) \rightarrow \text{Unhappy}(x) \times \)

\( \text{dislikes}(x,z) \land \text{Dish}(y) \land \text{contains}(y,z) \rightarrow \text{dislikes}(x,y) \times \)

\( \rightarrow \text{NutAllergic}(\text{sebastian}) \checkmark \)

\( \rightarrow \text{NutProduct}(\text{peanutOil}) \checkmark \)
OWL EL: PTime complete

OWL RL: PTime complete
OWL EL: PTime complete

OWL RL: PTime complete

OWL EL+RL: N2ExpTime complete
Bringing it all together: ELP
Bringing it all together: ELP

DL Rules for OWL EL
  +
Conjunctions of Roles
  +
DL-safe variables
Theorem

Inferencing in ELP is PTime complete.
Bringing it all together: ELP

\[ \text{ThaiCurry} \sqsubseteq \exists \text{contains.\{peanutOil\}} \]

\[ \top \sqsubseteq \forall \text{orderedDish.Dish} \]

\text{sebastian: } \exists \text{orderedDish.ThaiCurry} \]

\[ \text{NutAllergic}(x) \land \text{NutProduct}(y) \rightarrow \text{dislikes}(x,y) \]

\[ \text{orderedDish}(x,y) \land \text{dislikes}(x,y) \rightarrow \text{Unhappy}(x) \]

\[ \text{dislikes}(x,z) \land \text{Dish}(y) \land \text{contains}(y,z) \rightarrow \text{dislikes}(x,y) \]

\[ \rightarrow \text{NutAllergic}(\text{sebastian}) \]

\[ \rightarrow \text{NutProduct}(\text{peanutOil}) \]
Bringing it all together: ELP

→ Unhappy(sebastian)
Note

ELP supports inferencing in OWL EL and OWL RL.
Understanding DL-safety

\(\text{ThaiCurry} \sqsubseteq \exists \text{contains}.\text{FishProduct}\)

\(\top \sqsubseteq \forall \text{orderedDish}.\text{Dish}\)

\(\text{markus}: \exists \text{orderedDish}.\text{ThaiCurry}\)

\(\text{Vegetarian}(x) \land \text{FishProduct}(y) \rightarrow \text{dislikes}(x,y)\)

\(\text{orderedDish}(x,y) \land \text{dislikes}(x,y) \rightarrow \text{Unhappy}(x)\)

\(\text{dislikes}(x,z) \land \text{Dish}(y) \land \text{contains}(y,z) \rightarrow \text{dislikes}(x,y)\)

\(\rightarrow \text{Vegetarian}(\text{markus})\)
Understanding DL-safety

*Unhappy*(markus)

cannot be concluded
Towards Implementation
Theorem

Inferencing in ELP can be reduced in linear time to inferencing in 3-variable datalog.
Reasoning through Datalog

- Transformation to datalog is completely syntactic.
- Each axiom/rule can be transformed individually.
- Datalog engines can be used as blackbox.
- Instance and subsumption checking directly in datalog.
ELP: DL-based tractable rule language

- Almost completely expressible in OWL 2
- Support for OWL EL and OWL RL
- Linear-time conversion to 3-var datalog → simple implementation strategy

Happy(markus)  Happy(sebastian)
    Happy(pascal)