Karlsruhe Institute of Technology

OWL 2 Rules (Part 2)

Tutorial at ESWC2009

May 31, 2009

Forschungszentrum Karlsruhe in der Helmholtz-Gemeinschaft

Universität Karlsruhe (TH) Research University · founded 1825

Pascal Hitzler 

Markus Krötzsch 

Sebastian Rudolph 

AIFB, Universität Karlsruhe (TH)

http://www.pascal-hitzler.de

http://korrekt.org

http://www.sebastian-rudolph.de

References and Pointers

- **See** http://semantic-web-grundlagen.de/wiki/ESWC09_Tutorial **for the complete set of slides, and for links to references.**

Main References:

- Markus Krötzsch, Sebastian Rudolph, Pascal Hitzler, Description Logic Rules. In Malik Ghallab, Constantine D. Spyropoulos, Nikos Fakotakis, Nikos Avouris, eds.: Proceedings of the 18th European Conference on Artificial Intelligence (**ECAI-08**), pp. 80–84. IOS Press 2008.
- Markus Krötzsch, Sebastian Rudolph, Pascal Hitzler, ELP: Tractable Rules for OWL 2. In Amit Sheth, Steffen Staab, Mike Dean, Massimo Paolucci, Diana Maynard, Timothy Finin, Krishnaprasad Thirunarayan, eds.: Proceedings of the 7th International Semantic Web Conference (**ISWC-08**), pp. 649–664. Springer 2008.

Contents

Contents

- **Motivation: OWL and Rules**
- Preliminaries: Datalog

Intro

- More rules than you ever need: SWRL
- Retaining decidability I: DL-safety
- Retaining decidability II: DL Rules

Extending OWL with Rules

- The rules hidden in OWL 2: SROIQ Rules
- Retaining tractability I: OWL 2 EL Rules
- Retaining tractability II: DLP 2

Rules inside OWL

- Retaining tractability III: ELP

putting it all together

Motivation: OWL and Rules

- Rules (mainly, logic programming) as alternative ontology modelling paradigm.
- Similar tradition, and in use in practice (e.g. F-Logic)

- Ongoing: W3C RIF working group
  - Rule Interchange Format
  - based on Horn-logic
  - language standard forthcoming 2009

- Seek: Integration of rules paradigm with ontology paradigm
  - Here: Tight Integration in the tradition of OWL
  - Foundational obstacle: reasoning efficiency / decidability [naive combinations are undecidable]

Contents

- Motivation: OWL and Rules
- **Preliminaries: Datalog**

Intro

- More rules than you ever need: SWRL
- Retaining decidability I: DL-safety
- Retaining decidability II: DL Rules

Extending OWL with Rules

- The rules hidden in OWL 2: SROIQ Rules
- Retaining tractability I: OWL 2 EL Rules
- Retaining tractability II: DLP 2

Rules inside OWL

- Retaining tractability III: ELP

putting it all together

Preliminaries: Datalog

- Essentially Horn-rules without function symbols

  general form of the rules:

  body $\rightarrow$ head

  $p_1(x_1,\ldots,x_n) \wedge \ldots \wedge p_m(y_1,\ldots,y_k) \rightarrow q(z_1,\ldots,z_j)$

  semantics either as in predicate logic
  or as Herbrand semantics (see next slide)

- decidable
- polynomial data complexity (in number of facts)
- combined (overall) complexity: ExpTime
- combined complexity is P if the number of variables per rule is globally bounded

Datalog semantics example

- Example:
  $p(x) \rightarrow q(x)$
  $q(x) \rightarrow r(x)$
  $\rightarrow p(a)$

- predicate logic semantics:

  $(\forall x)\ (p(x) \rightarrow r(x))$
  and
  $(\forall x)\ (\neg r(x) \rightarrow \neg p(x))$
  are logical consequences

  q(a) and r(a)
  are logical consequences

- Herbrand semantics

  those on the left are not logical consequences

  q(a) and r(a)
  are logical consequences

  material implication:
  apply only to known constants

Contents

More rules than you ever need: SWRL

- Union of OWL DL with (binary) function-free Horn rules
  (with binary Datalog rules)

- undecidable
- no native tools available

- rather an overarching formalism

- see http://www.w3.org/Submission/SWRL/

SWRL example (running example)

$\text{NutAllergic(sebastian)}$

$\text{NutProduct(peanutOil)}$

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

$\text{ThaiCurry} \sqsubseteq \exists\text{contains.\{peanutOil\}}$

$\top \sqsubseteq \forall\text{orderedDish.Dish}$

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

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

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

SWRL example (running example)

$NutAllergic(sebastian)$

$NutProduct(peanutOil)$

$\exists orderedDish.ThaiCurry(sebastian)$

$ThaiCurry \sqsubseteq \exists contains.\{peanutOil\}$

$\top \sqsubseteq \forall orderedDish.Dish$

$NutAllergic(x) \wedge NutProduct(y) \rightarrow dislikes(x,y)$

$dislikes(x,z) \wedge Dish(y) \wedge contains(y,z) \rightarrow dislikes(x,y)$

$orderedDish(x,y) \wedge dislikes(x,y) \rightarrow Unhappy(x)$

Conclusions:

$dislikes(sebastian,peanutOil)$

SWRL example (running example)

NutAllergic(sebastian)

NutProduct(peanutOil)

∃orderedDish.ThaiCurry(sebastian)

ThaiCurry ⊑ ∃contains.{peanutOil}

⊤ ⊑ ∀orderedDish.Dish

orderedDish rdfs:range Dish.

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

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

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

Conclusions:

dislikes(sebastian,peanutOil)

orderedDish(sebastian,y_s)

ThaiCurry(y_s)

Dish(y_s)

SWRL example (running example)

NutAllergic(sebastian)

NutProduct(peanutOil)

$\exists$orderedDish.ThaiCurry(sebastian)

ThaiCurry $\sqsubseteq$ $\exists$contains.{peanutOil}

$\top \sqsubseteq \forall$orderedDish.Dish

NutAllergic(x) $\wedge$ NutProduct(y) $\rightarrow$ dislikes(x,y)

dislikes(x,z) $\wedge$ Dish(y) $\wedge$ contains(y,z) $\rightarrow$ dislikes(x,y)

orderedDish(x,y) $\wedge$ dislikes(x,y) $\rightarrow$ Unhappy(x)

Conclusions:

dislikes(sebastian,peanutOil)

orderedDish(sebastian,y_s)

ThaiCurry(y_s)

Dish(y_s)

contains(y_s,peanutOil)

SWRL example (running example)

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

$\text{NutProduct}(\text{peanutOil})$

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

$\text{ThaiCurry} \sqsubseteq \exists\text{contains}.\{\text{peanutOil}\}$

$\top \sqsubseteq \forall\text{orderedDish.Dish}$

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

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

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

Conclusions:

$\text{dislikes}(\text{sebastian},\text{peanutOil})$

$\text{orderedDish}(\text{sebastian},y_s)$

$\text{ThaiCurry}(y_s)$

$\text{Dish}(y_s)$

$\text{contains}(y_s,\text{peanutOil})$

$\text{dislikes}(\text{sebastian},y_s)$

SWRL example (running example)

NutAllergic(sebastian)

NutProduct(peanutOil)

$\exists$orderedDish.ThaiCurry(sebastian)

ThaiCurry $\sqsubseteq$ $\exists$contains.{peanutOil}

$\top \sqsubseteq \forall$orderedDish.Dish

NutAllergic(x) $\wedge$ NutProduct(y) $\rightarrow$ dislikes(x,y)

dislikes(x,z) $\wedge$ Dish(y) $\wedge$ contains(y,z) $\rightarrow$ dislikes(x,y)

orderedDish(x,y) $\wedge$ dislikes(x,y) $\rightarrow$ Unhappy(x)

Conclusions:

dislikes(sebastian,peanutOil)

orderedDish(sebastian,y_s)

ThaiCurry(y_s)

Dish(y_s)

contains(y_s,peanutOil)

dislikes(sebastian,y_s)

Unhappy(sebastian)

SWRL example (running example)

$$\text{NutAllergic(sebastian)}$$
$$\text{NutProduct(peanutOil)}$$
$$\exists\text{orderedDish.ThaiCurry(sebastian)}$$

$$\text{ThaiCurry} \sqsubseteq \exists\text{contains.\{peanutOil\}}$$
$$\top \sqsubseteq \forall\text{orderedDish.Dish}$$

$$\text{NutAllergic}(x) \land \text{NutProduct}(y) \rightarrow \text{dislikes}(x,y)$$
$$\text{dislikes}(x,z) \land \text{Dish}(y) \land \text{contains}(y,z) \rightarrow \text{dislikes}(x,y)$$
$$\text{orderedDish}(x,y) \land \text{dislikes}(x,y) \rightarrow \text{Unhappy}(x)$$

Conclusion: Unhappy(sebastian)

Contents

- Motivation: OWL and Rules
- Preliminaries: Datalog

Intro

- More rules than you ever need: SWRL
- **Retaining decidability I: DL-safety**
- Retaining decidability II: DL Rules

Extending OWL with Rules

- The rules hidden in OWL 2: SROIQ Rules
- Retaining tractability I: OWL 2 EL Rules
- Retaining tractability II: DLP 2

Rules inside OWL

- Retaining tractability III: ELP

putting it all together

Retaining decidability I: DL-safety

- **Reinterpret SWRL rules:**
  Rules apply only to individuals which are explicitly given in the knowledge base.
  - **Herbrand-style way of interpreting them**

- **OWL DL + DL-safe SWRL is decidable**
- **Native support e.g. by KAON2 and Pellet**

- See e.g. Boris Motik, Ulrike Sattler, and Rudi Studer. Query Answering for OWL-DL with Rules. Journal of Web Semantics 3(1):41–60, 2005.

DL-safe SWRL example

NutAllergic(sebastian)

NutProduct(peanutOil)

$\exists$orderedDish.ThaiCurry(sebastian)

ThaiCurry $\sqsubseteq$ $\exists$contains.{peanutOil}

$\top$ $\sqsubseteq$ $\forall$orderedDish.Dish

DL-safe {

NutAllergic(x) $\wedge$ NutProduct(y) $\rightarrow$ dislikes(x,y)

dislikes(x,z) $\wedge$ Dish(y) $\wedge$ contains(y,z) $\rightarrow$ dislikes(x,y)

orderedDish(x,y) $\wedge$ dislikes(x,y) $\rightarrow$ Unhappy(x)

Unhappy(sebastian) can*not* be concluded

DL-safe SWRL example

NutAllergic(sebastian)

NutProduct(peanutOil)

∃orderedDish.ThaiCurry(sebastian)

ThaiCurry ⊑ ∃contains.{peanutOil}

⊤ ⊑ ∀orderedDish.Dish

DL-safe:

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

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

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

Conclusions:

dislikes(sebastian,peanutOil)

orderedDish(sebastian,y_s)

ThaiCurry(y_s)

Dish(y_s)

contains(y_s,peanutOil)

~~dislikes(sebastian,y_s)~~

Contents

Retaining decidability II: DL Rules

- General idea:
  Find out which rules can be encoded in OWL (2 DL) anyway

- Man(x) ∧ hasBrother(x,y) ∧ hasChild(y,z) → Uncle(x)
  - Man ⊓ ∃hasBrother.∃hasChild.⊤ ⊑ Uncle

- ThaiCurry(x) → ∃contains.FishProduct(x)
  - ThaiCurry ⊑ ∃contains.FishProduct

- kills(x,x) → suicide(x)
  - ∃kills.Self ⊑ suicide

  suicide(x) → kills(x,x)

  suicide ⊑ ∃kills.Self

Note: with these two axioms,
suicide* is basically the same as *kills

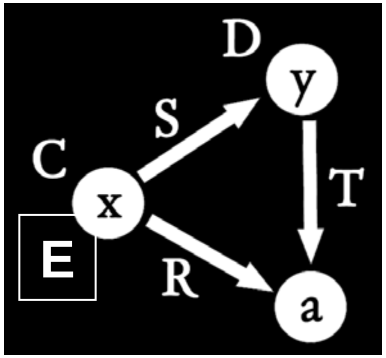
DL Rules: more examples

- $NutAllergic(x) \land NutProduct(y) \rightarrow dislikes(x,y)$
  - $NutAllergic \equiv \exists nutAllergic.Self$
    $NutProduct \equiv \exists nutProduct.Self$
    $nutAllergic \circ U \circ nutProduct \sqsubseteq dislikes$
- $dislikes(x,z) \land Dish(y) \land contains(y,z) \rightarrow dislikes(x,y)$
  - $Dish \equiv \exists dish.Self$
    $dislikes \circ contains^{-} \circ dish \sqsubseteq dislikes$
- $worksAt(x,y) \land University(y) \land supervises(x,z) \land PhDStudent(z) \rightarrow professorOf(x,z)$
  - $\exists worksAt.University \equiv \exists worksAtUniversity.Self$
    $PhDStudent \equiv \exists phDStudent.Self$
    $worksAtUniversity \circ supervises \circ phDStudent \sqsubseteq professorOf$

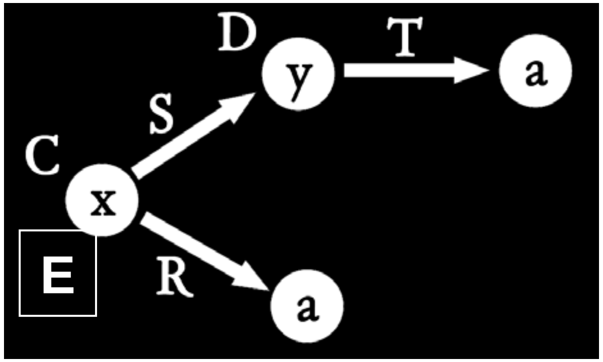
DL Rules: definition

- Tree-shaped bodies
- First argument of the conclusion is the root

- $C(x) \wedge R(x,a) \wedge S(x,y) \wedge D(y) \wedge T(y,a) \rightarrow E(x)$
  - $C \sqcap \exists R.\{a\} \sqcap \exists S.(D \sqcap \exists T.\{a\}) \sqsubseteq E$



duplicating
nominals
is
ok

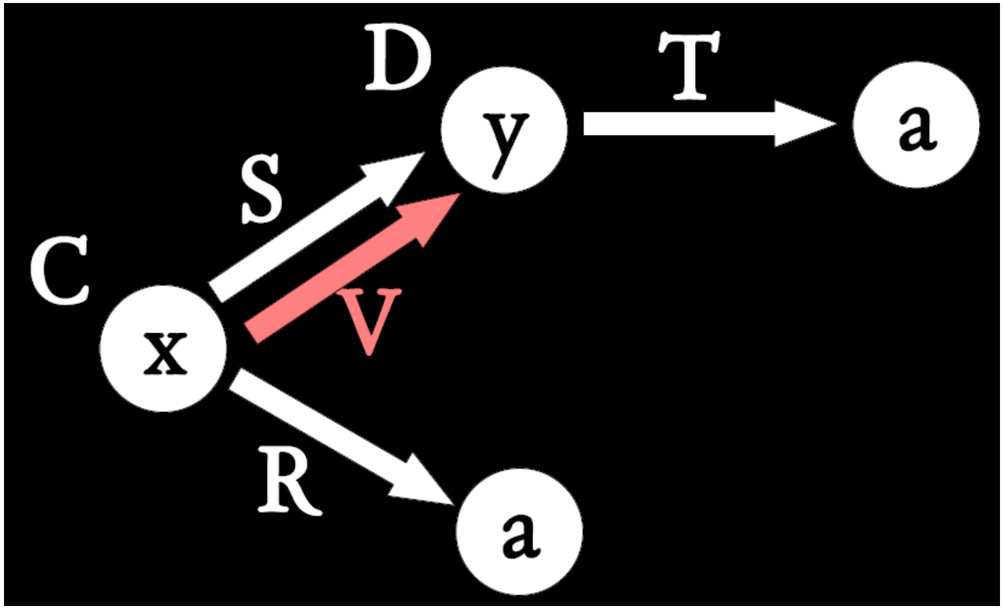


DL Rules: definition

- **Tree-shaped bodies**
- **First argument of the conclusion is the root**

- $C(x) \wedge R(x,a) \wedge S(x,y) \wedge D(y) \wedge T(y,a) \rightarrow V(x,y)$

  $C \sqcap \exists R.\{a\} \sqsubseteq \exists R1.Self$
  $D \sqcap \exists T.\{a\} \sqsubseteq \exists R2.Self$
  $R1 \circ S \circ R2 \sqsubseteq V$



DL Rules: definition

- Tree-shaped bodies
- First argument of the conclusion is the root

- complex classes are allowed in the rules
  - Mouse(x) $\wedge$ $\exists$hasNose.TrunkLike(y) $\rightarrow$ smallerThan(x,y)
  - ThaiCurry(x) $\rightarrow$ $\exists$contains.FishProduct(x)

    Note: This allows to reason with unknowns (unlike Datalog)
  - allowed class constructors depend on the chosen underlying description logic!

DL Rules: definition

Given a description logic $\mathcal{D}$,

the language $\mathcal{D}$ Rules consists of

- all axioms expressible in $\mathcal{D}$,
- plus all rules with
  - tree-shaped bodies, where
  - the first argument of the conclusion is the root, and
  - complex classes from $\mathcal{D}$ are allowed in the rules.
  - <plus possibly some restrictions concerning e.g. the use of simple roles – depending on $\mathcal{D}$>

Contents

- Motivation: OWL and Rules
- Preliminaries: Datalog

Intro

- More rules than you ever need: SWRL
- Retaining decidability I: DL-safety
- Retaining decidability II: DL Rules

Extending OWL with Rules

- **The rules hidden in OWL 2: SROIQ Rules**
- Retaining tractability I: OWL 2 EL Rules
- Retaining tractability II: DLP 2

Rules inside OWL

- Retaining tractability III: ELP

putting it all together

The rules hidden in OWL 2: SROIQ Rules

- N2ExpTime complete
- In fact, SROIQ Rules can be translated into SROIQ
  i.e. they don't add expressivity.

  Translation is polynomial.
- SROIQ Rules are essentially helpful syntactic sugar for OWL 2.

SROIQ Rules example

NutAllergic(sebastian)
NutProduct(peanutOil)
$\exists$orderedDish.ThaiCurry(sebastian)

ThaiCurry $\sqsubseteq$ $\exists$contains.{peanutOil}
$\top \sqsubseteq \forall$orderedDish.Dish

NutAllergic(x) $\wedge$ NutProduct(y) $\rightarrow$ dislikes(x,y)
dislikes(x,z) $\wedge$ Dish(y) $\wedge$ contains(y,z) $\rightarrow$ dislikes(x,y)
orderedDish(x,y) $\wedge$ dislikes(x,y) $\rightarrow$ Unhappy(x)

!not a SROIQ Rule!

SROIQ Rules normal form

- Each SROIQ Rule can be written ("linearised") such that
  - the body-tree is linear,
  - if the head is of the form R(x,y), then y is the leaf of the tree, and
  - if the head is of the form C(x), then the tree is only the root.

- $\text{worksAt}(x,y) \land \text{University}(y) \land \text{supervises}(x,z) \land \text{PhDStudent}(z) \rightarrow \text{professorOf}(x,z)$
  - $\exists\text{worksAt}.\text{University}(x) \land \text{supervises}(x,z) \land \text{PhDStudent}(z) \rightarrow \text{professorOf}(x,z)$

- $C(x) \land R(x,a) \land S(x,y) \land D(y) \land T(y,a) \rightarrow V(x,y)$
  - $(C \sqcap \exists R.\{a\})(x) \land S(x,y) \land (D \sqcap \exists T.\{a\})(y) \rightarrow V(x,y)$

Contents

Karlsruhe Institute of Technology

- Motivation: OWL and Rules
- Preliminaries: Datalog

Intro

- More rules than you ever need: SWRL
- Retaining decidability I: DL-safety
- Retaining decidability II: DL Rules

Extending OWL with Rules

- The rules hidden in OWL 2: SROIQ Rules
- **Retaining tractability I: OWL 2 EL Rules**
- Retaining tractability II: DLP 2

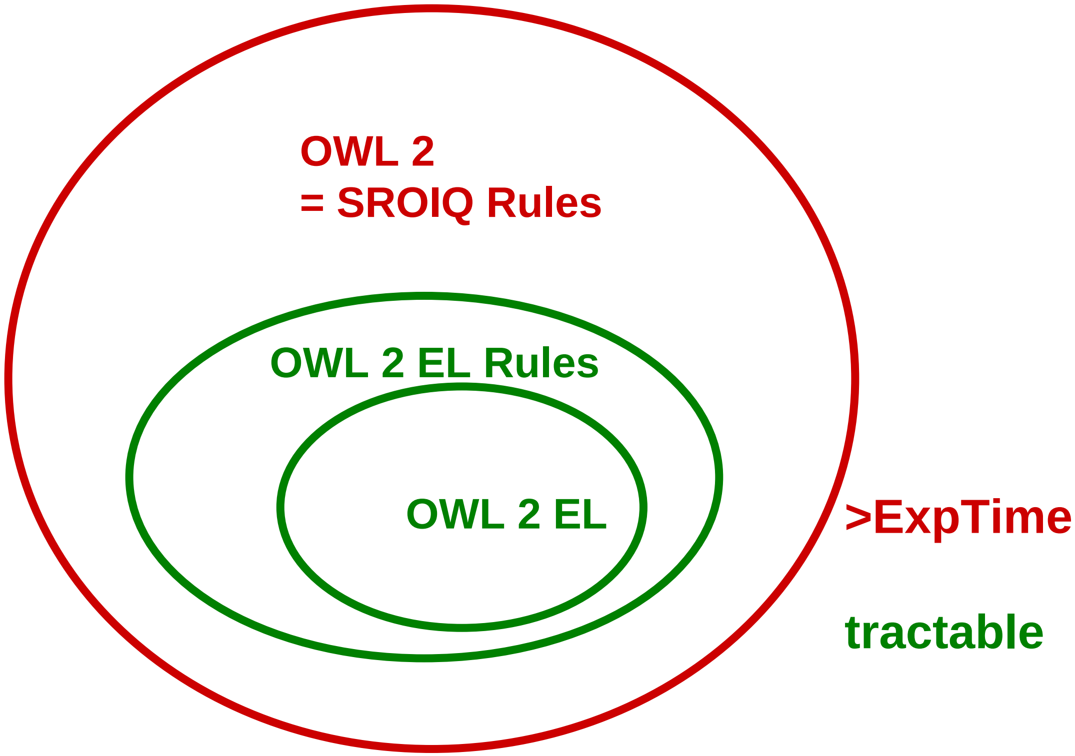
Rules inside OWL

- Retaining tractability III: ELP

putting it all together

Retaining tractability I: OWL 2 EL Rules

- **EL++ Rules are PTime complete**
- **EL++ Rules offer expressivity which is not readily available in EL++.**



OWL 2 EL Rules: normal form

Karlsruhe Institute of Technology

- Every EL++ Rule can be converted into a normal form,where
  - occurring classes in the rule body are either atomic or nominals,
  - all variables in a rule's head occur also in its body, and
  - rule heads can only be of one of the forms A(x), ∃R.A(x), R(x,y), where A is an atomic class or a nominal or ⊤ or ⊥.

- Translation is polynomial.

- ∃worksAt.University(x) ∧ supervises(x,z) ∧ PhDStudent(z) → professorOf(x,z)
  - worksAt(x,y) ∧ University(y) ∧ supervises(x,z) ∧ PhDStudent(z) → professorOf(x,z)

- ThaiCurry(x) → ∃contains.FishProduct(x)

OWL 2 EL Rules in a nutshell

Essentially, OWL 2 EL Rules is

- **Binary Datalog with tree-shaped rule bodies,**
- **extended by**
  - **occurrence of nominals as atoms and**
  - **existential class expressions in the head.**

- **The existentials really make the difference.**

- **Arguably the better alternative to OWL 2 EL (aka EL++)?**
  - **(which is covered anyway)**

Contents

Retaining tractability II: DLP 2

- DLP 2 is
  - DLP (aka OWL 2 RL) extended with
  - DL rules, which use
    - left-hand-side class expressions in the bodies and
    - right-hand-side class expressions in the head.
- Polynomial transformation into 5-variable Horn rules.
- PTime.
- Quite a bit more expressive than DLP / OWL 2 RL ...

Contents

- Motivation: OWL and Rules
- Preliminaries: Datalog

Intro

- More rules than you ever need: SWRL
- Retaining decidability I: DL-safety
- Retaining decidability II: DL Rules

Extending OWL with Rules

- The rules hidden in OWL 2: SROIQ Rules
- Retaining tractability I: OWL 2 EL Rules
- Retaining tractability II: DLP 2

Rules inside OWL

- **Retaining tractability III: ELP**

putting it all together

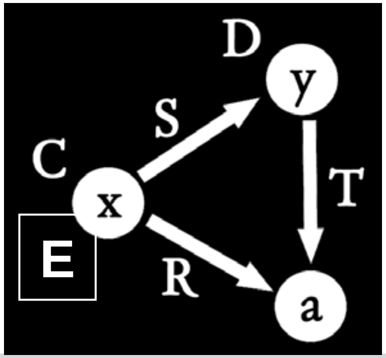
Retaining tractability III: ELP (aka putting it all together)

- ELP is
  - OWL 2 EL Rules +
  - a generalisation of DL-safety +
  - variable-restricted DL-safe Datalog +
  - role conjunctions (for simple roles).
- PTime complete.
- Contains OWL 2 EL and OWL 2 RL.
- Covers variable-restricted Datalog.

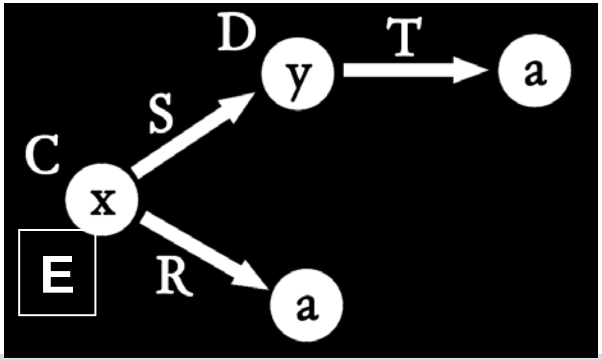
DL-safe variables

- A generalisation of DL-safety.
- DL-safe variables are special variables which bind only to named individuals (like in DL-safe rules).
- DL-safe variables can replace individuals in EL++ rules.

- $C(x) \wedge R(x,x_s) \wedge S(x,y) \wedge D(y) \wedge T(y,x_s) \rightarrow E(x)$
  with x_s a safe variable is allowed, because
  $C(x) \wedge R(x,a) \wedge S(x,y) \wedge D(y) \wedge T(y,a) \rightarrow E(x)$
  is an EL++ rule.



duplicating nominals is ok



Variable-restricted DL-safe Datalog

- n-Datalog is Datalog, where the number of variables occurring in rules is globally bounded by n.
- complexity of n-Datalog is PTime (for fixed n)
  - (but exponential in n)
- in a sense, this is cheating.
- in another sense, this means that using a few DL-safe Datalog rules together with an EL++ rules knowledge base shouldn't really be a problem in terms of reasoning performance.

Role conjunctions

- **orderedDish(x,y) $\wedge$ dislikes(x,y) $\rightarrow$ Unhappy(x)**

- **In fact, role conjunctions can also be added to OWL 2 DL without increase in complexity.**

- Sebastian Rudolph, Markus Krötzsch, Pascal Hitzler, Cheap Boolean Role Constructors for Description Logics. In: Steffen Hölldobler and Carsten Lutz and Heinrich Wansing (eds.), Proceedings of 11th European Conference on Logics in Artificial Intelligence (JELIA), volume 5293 of LNAI, pp. 362-374. Springer, September 2008.

Retaining tractability III: ELP (aka putting it all together)

- ELP_n is
  - OWL 2 EL Rules generalised by DL-safe variables +
  - DL-safe Datalog rules with at most n variables +
  - role conjunctions (for simple roles).
- PTime complete (for fixed n).
  - exponential in n
- Contains OWL 2 EL and OWL 2 RL.
- Covers all Datalog rules with at most n variables. (!)

ELP example

NutAllergic(sebastian)

NutProduct(peanutOil)

$\exists$orderedDish.ThaiCurry(sebastian)

ThaiCurry $\sqsubseteq$ $\exists$contains.{peanutOil}

$\top$ $\sqsubseteq$ $\forall$orderedDish.Dish

[okay]

NutAllergic(x) $\wedge$ NutProduct(y) $\rightarrow$ dislikes(x,y)

dislikes(x,z) $\wedge$ Dish(y) $\wedge$ contains(y,z) $\rightarrow$ dislikes(x,y)

orderedDish(x,y) $\wedge$ dislikes(x,y) $\rightarrow$ Unhappy(x)

[okay – role conjunction]

not an EL++ rule

ELP example

- dislikes(x,z) $\wedge$ Dish(y) $\wedge$ contains(y,z) $\rightarrow$ dislikes(x,y)
  as SROIQ rule translates to

  Dish $\equiv$ $\exists$dish.Self
  dislikes o contains^{-} o dish $\sqsubseteq$ dislikes

  but we don't have inverse roles in ELP!

- solution: make z a DL-safe variable:

  dislikes(x,!z) $\wedge$ Dish(y) $\wedge$ contains(y,!z) $\rightarrow$ dislikes(x,y)

  this is fine ☺

DL-safe SWRL example

NutAllergic(sebastian)
NutProduct(peanutOil)
$\exists$orderedDish.ThaiCurry(sebastian)

ThaiCurry $\sqsubseteq$ $\exists$contains.{peanutOil}
$\top \sqsubseteq \forall$orderedDish.Dish

NutAllergic(x) $\wedge$ NutProduct(y) $\rightarrow$ dislikes(x,y)
dislikes(x,!z) $\wedge$ Dish(y) $\wedge$ contains(y,!z) $\rightarrow$ dislikes(x,y)
orderedDish(x,y) $\wedge$ dislikes(x,y) $\rightarrow$ Unhappy(x)

Conclusions:
dislikes(sebastian,peanutOil)
orderedDish(sebastian,y_s)
ThaiCurry(y_s)
Dish(y_s)
contains(y_s,peanutOil)
dislikes(sebastian,y_s)

ELP example

$NutAllergic(sebastian)$

$NutProduct(peanutOil)$

$\exists orderedDish.ThaiCurry(sebastian)$

$ThaiCurry \sqsubseteq \exists contains.\{peanutOil\}$

$\top \sqsubseteq \forall orderedDish.Dish$

$NutAllergic(x) \wedge NutProduct(y) \rightarrow dislikes(x,y)$

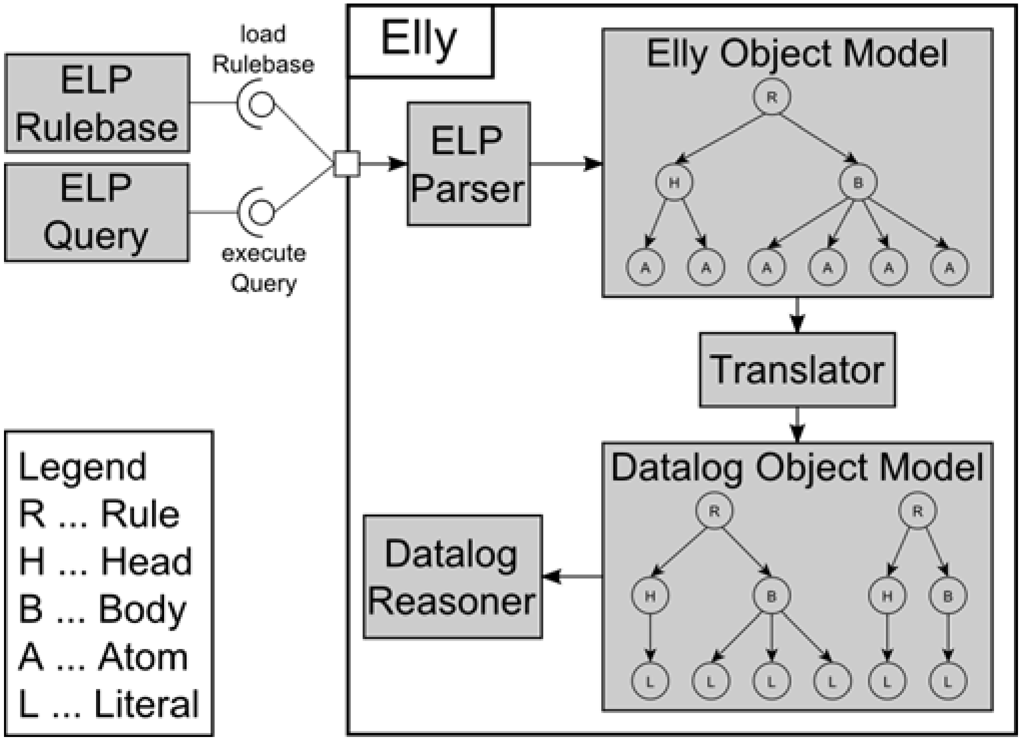
$dislikes(x,!z) \wedge Dish(y) \wedge contains(y,!z) \rightarrow dislikes(x,y)$

$orderedDish(x,y) \wedge dislikes(x,y) \rightarrow Unhappy(x)$

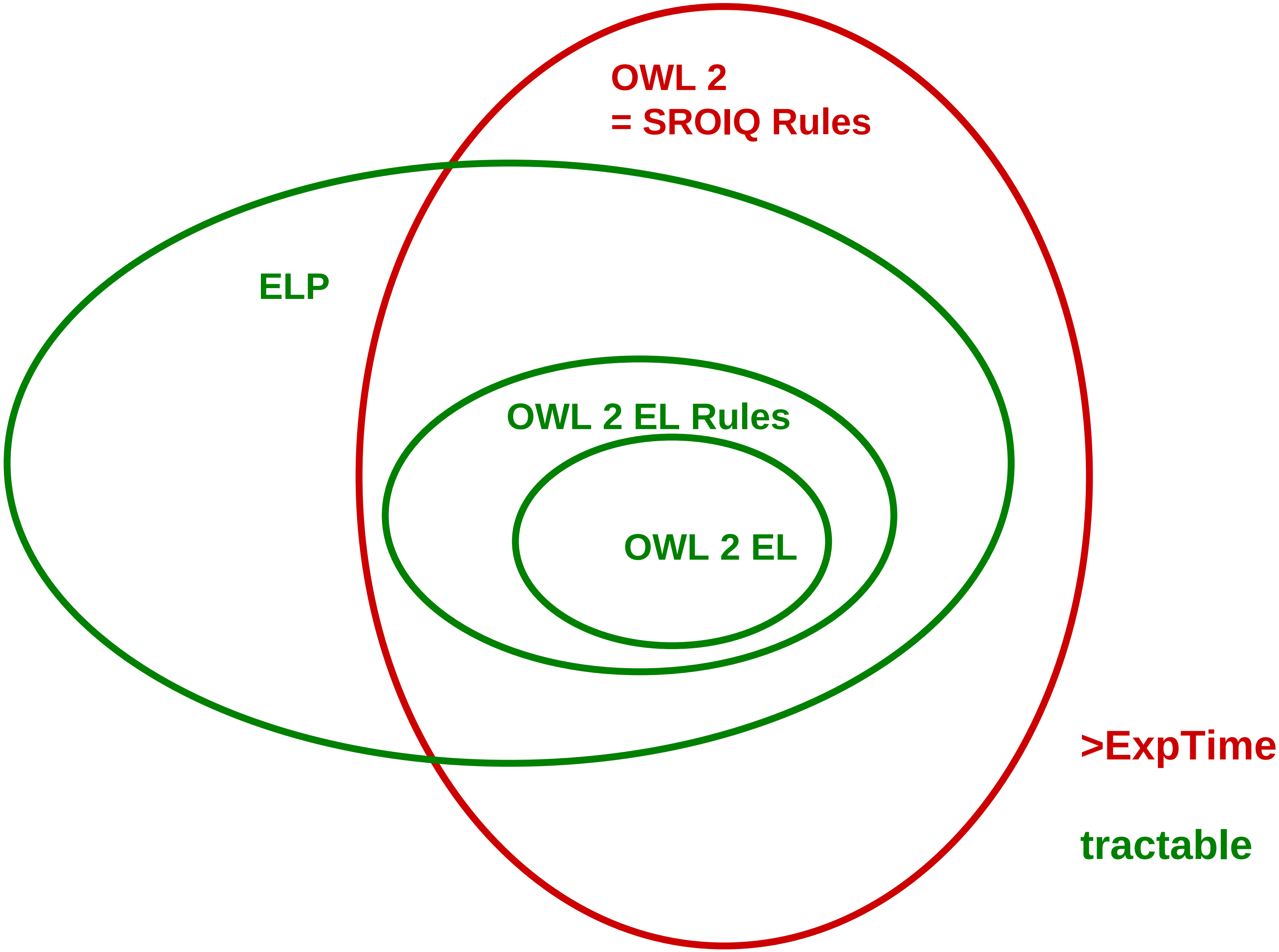
Conclusion: Unhappy(sebastian)

ELP Reasoner ELLY

- Implementation currently being finalised.
- Based on IRIS Datalog reasoner.
- In cooperation with STI Innsbruck (Barry Bishop, Daniel Winkler, Gulay Unel).



The Big Picture

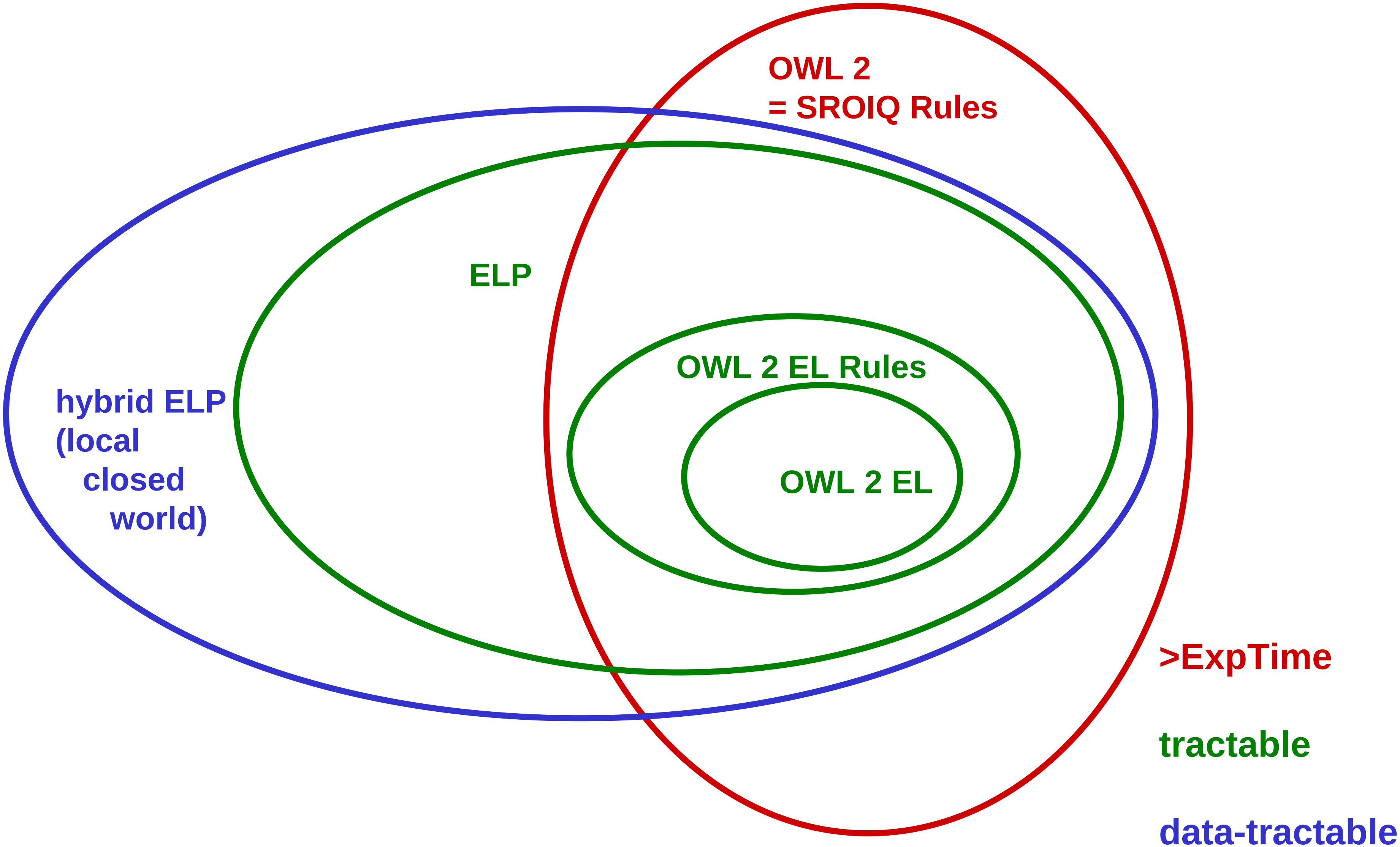


Closed World and ELP

- **There's an extension of ELP using (non-monotonic) closed-world reasoning – based on a well-founded semantics for hybrid MKNF knowledge bases.**

- Matthias Knorr, Jose Julio Alferes, Pascal Hitzler, A Coherent Well-founded model for Hybrid MKNF knowledge bases. In: Malik Ghallab, Constantine D. Spyropoulos, Nikos Fakotakis, Nikos Avouris (eds.), Proceedings of the 18th European Conference on Artificial Intelligence, ECAI2008, Patras, Greece, July 2008. IOS Press, 2008, pp. 99-103.

The Big Picture II



References

- Markus Krötzsch, Sebastian Rudolph, Pascal Hitzler, Description Logic Rules. In Malik Ghallab, Constantine D. Spyropoulos, Nikos Fakotakis, Nikos Avouris, eds.: Proceedings of the 18th European Conference on Artificial Intelligence (ECAI-08), pp. 80–84. IOS Press 2008.
- Markus Krötzsch, Sebastian Rudolph, Pascal Hitzler, ELP: Tractable Rules for OWL 2. In Amit Sheth, Steffen Staab, Mike Dean, Massimo Paolucci, Diana Maynard, Timothy Finin, Krishnaprasad Thirunarayan, eds.: Proceedings of the 7th International Semantic Web Conference (ISWC-08), pp. 649–664. Springer 2008.
- http://www.w3.org/Submission/SWRL/
- Boris Motik, Ulrike Sattler, and Rudi Studer. Query Answering for OWL-DL with Rules. Journal of Web Semantics 3(1):41–60, 2005.

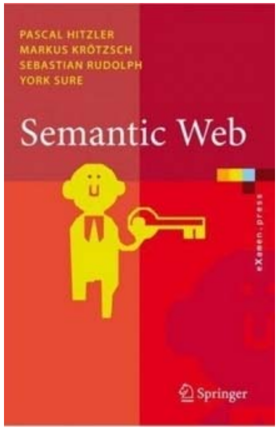
References

- Sebastian Rudolph, Markus Krötzsch, Pascal Hitzler, Cheap Boolean Role Constructors for Description Logics. In: Steffen Hölldobler and Carsten Lutz and Heinrich Wansing (eds.), Proceedings of 11th European Conference on Logics in Artificial Intelligence (JELIA), volume 5293 of LNAI, pp. 362-374. Springer, September 2008.
- Matthias Knorr, Jose Julio Alferes, Pascal Hitzler, A Coherent Well-founded model for Hybrid MKNF knowledge bases. In: Malik Ghallab, Constantine D. Spyropoulos, Nikos Fakotakis, Nikos Avouris (eds.), Proceedings of the 18th European Conference on Artificial Intelligence, ECAI2008, Patras, Greece, July 2008. IOS Press, 2008, pp. 99-103.

References

- Pascal Hitzler, Markus Krötzsch, Sebastian Rudolph, York Sure, Semantic Web – Grundlagen. Springer, 2008. http://www.semantic-web-grundlagen.de/

- **Pascal Hitzler, Markus Krötzsch, Sebastian Rudolph, Foundations of Semantic Web Technologies. Chapman & Hall/CRC, 2009. http://www.semantic-web-book.org/wiki/FOST (Grab a flyer from us.)**



Thanks!

http://semantic-web-grundlagen.de/wiki/ESWC09_Tutorial