OWL (syntax)

...

ISKM89 Organizace dat - sémantický web | podzim 2023
Zuzana Nevěřilová | Centrum zpracování přirozeného jazyka

Web Ontology Language (OWL)

- na první pohled rozšiřuje popis v RDF(S) o vlastnosti relací
- ve skutečnosti se vracíme k deskripční logice (DL)
- to všechno kvůli odvozování

OWL používá funkcionální syntax (functional syntax), což je vrstva mezi:

- jakoukoliv syntaxí zápisu (např. RDF)
- sémantikou

Funkcionální syntax se podobá tomu, co jsme viděli u DL. Ještě více se tomu podobá manchesterská syntax (https://www.w3.org/TR/owl2-manchester-syntax/)

OWL má dvě sémantiky: **RDF-based** a **Direct**.

Dva pohledy na OWL

OWL jako rozšíření RDF

- přidáme negaci
- přidáme charakteristiky vlastností
- inference se rozšíří (např. díky owl:TransitiveProperty)
- můžeme použít nepojmenované (anonymní) třídy

OWL jako DL

- tvrzení zapisujeme jako formule DL
- důkazy tvrzení vedeme stejně jako důkazy v DL
- místo anonymních tříd používáme množinové operace a závorky
- anotace jsou jen “nálepky”, nemají vliv na důkazy

Deskripční logika, formalizace

Koncept (unární predikát, třída, např. Rodič)
C, D - koncepty

⊤	všechno		top
⊥	nic		bottom
⊓	průnik (logický součin) konceptů	C ⊓ D	C and D
⊔	sjednocení (logický součet) konceptů	C ⊔ D	C or D
¬	negace (doplněk) konceptu	¬ C	not C
⊑	podmnožina konceptů	C ⊑ D	all C are D
≡	ekvivalence konceptů	C ≡ D	C is equivalent to D
=˙	definice konceptů	C = D	C is defined to be equal to D

Deskripční logika, formalizace

Koncept (unární predikát, třída, např. Rodič)
Role (binární predikát, vlastnost, např. máDítě)

C, D - koncepty, a, b - individua

v roli R (R-related) - koncept
R-následník (R-successor) - koncept C, který má roli R,
např. ∃ máDítě.Žena (věc, která má aspoň jedno dítě ženu)

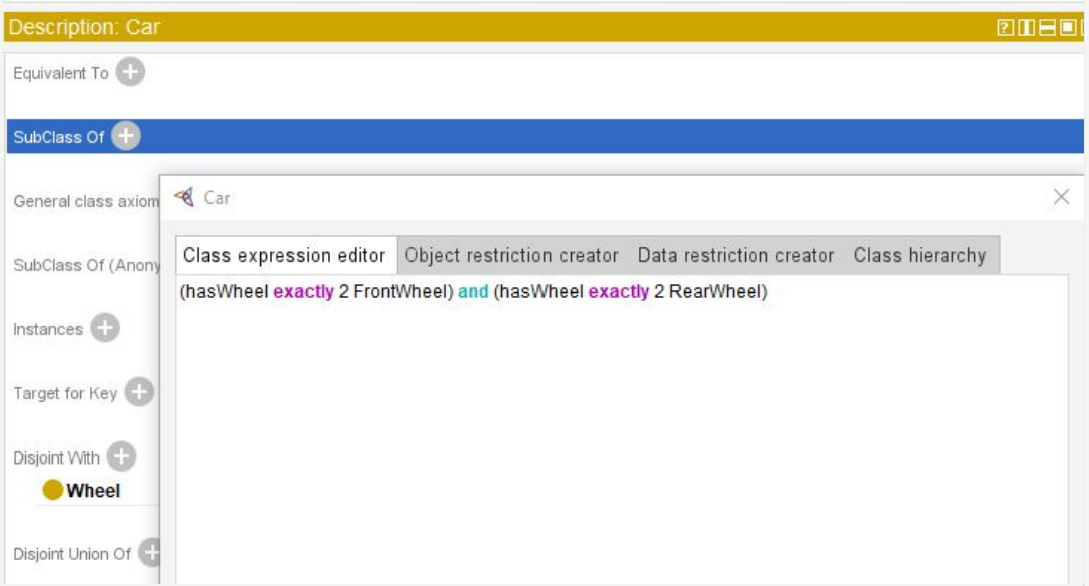
∀	univerzální kvantifikátor	∀ R.C	R-následníci jsou všichni C
∃	existenční kvantifikátor	∃ R.C	R-následník C existuje
:	přiřazení konceptu (assertion)	a : C nebo C(a)	a je instancí C
:	přiřazení role	(a,b) : R nebo R(a,b)	a je pro b v roli R

OWL jako deskripční logika - funkcionální syntax

DL	OWL	Manchester
⊤ (top)	owl:Thing	owl:Thing
⊥ (bottom)	owl:Nothing	owl:Nothing
Concept name	Class	Class
Role name	Object property	Object property
¬C	ObjectComplementOf(C)	not C
C ⊔ D	ObjectUnionOf(C D)	C or D
C ⊓ D	ObjectIntersectionOf(C D)	C and D
∃r.C	ObjectSomeValuesFrom(r C)	r some C
∀r.C	ObjectAllValuesFrom(r C)	r only C
(≥ n r.C)	ObjectMinCardinality(n r C)	r min n C
(≤ n r.C)	ObjectMaxCardinality(n r C)	r max n C
(= n r.C)	ObjectExactCardinality(n r C)	r exactly n C

OWL - Functional Syntax a Manchester Syntax

Funkcionální syntax
složitější oproti Manchester syntax
(ale velmi podobné)




```
Prefix(:=<http://www.semanticweb.org/admin/ontologie
Prefix(owl:=<http://www.w3.org/2002/07/owl#>)
Prefix(rdf:=<http://www.w3.org/1999/02/22-rdf-syntax
Prefix(xml:=<http://www.w3.org/XML/1998/namespace>)
Prefix(xsd:=<http://www.w3.org/2001/XMLSchema#>)
Prefix(rdfs:=<http://www.w3.org/2000/01/rdf-schema#>


Ontology(<.../my-ontology>


Declaration(Class(<...#Car>))
Declaration(Class(<...#FrontWheel>))
Declaration(Class(<...#RearWheel>))
Declaration(Class(<...#Wheel>))
Declaration(ObjectProperty(<...#hasWheel>))

SubClassOf(<...#Car> ObjectIntersectionOf(ObjectExactCardinality(2 <...#hasWheel> <...#FrontWheel>) ObjectExactCardinality(2
<...#hasWheel> <...#RearWheel>)))
```

OWL - RDF a Manchester Syntax

```
:Car rdf:type owl:Class ;
     rdfs:subClassOf [ owl:intersectionOf ( [ rdf:type owl:Restriction ;
                                              owl:onProperty :hasWheel ;
                                              owl:qualifiedCardinality
                                                "2"^^xsd:nonNegativeInteger ;
                                              owl:onClass :FrontWheel
                                            ]
                                            [ rdf:type owl:Restriction ;
                                              owl:onProperty :hasWheel ;
                                              owl:qualifiedCardinality
                                                "2"^^xsd:nonNegativeInteger ;
                                              owl:onClass :RearWheel
                                            ]
                                          ) ;
                       rdf:type owl:Class
                     ] ;
     owl:disjointWith :Wheel .
```

```
(hasWheel exactly 2 FrontWheel)
 and (hasWheel exactly 2 RearWheel)
```

OWL jako rozšíření RDF(S)

OWL umožňuje pracovat s anonymními třídami (intersection, union, restriction)

vlastnosti (properties):

- objektové (objectProperty)
- datové (datatypeProperty)

Vlastnosti mohou mít vlastnosti (charakteristiky):

funkční, tranzitivní, symetrické, ...

Vlastnosti lze zřetězit (property chains), což umožňuje další inferenci.

Možnost formulovat negativní tvrzení.