

# Description Logic vs. Metamodelling

Dániel Varró

Foundations of Model Driven System Development

Based on contributions from

Szilvia Varró-Gyapay and the DECOS European IP

## Designing Modeling Languages

- Metamodeling: Design methodology of modeling languages
- Metamodel: model of a modeling language
- Features:
  - Concrete syntax
  - Abstract syntax
  - Well-formedness rules
  - Semantics
  - Mappings to other languages

## Description Logic and Reasoning

Contribution of Szilvia Varró-Gyapay

## Description logic and reasoning

- Semantic web: knowledge management
  - intelligent information retrieval
  - interpretation of the information
- need for meta information and reasoning
- tools:
  - Ontology:  
hierarchical knowledge representation
  - description logic formal model for analysis



## The Language of DL

### DL Examples

$\text{Mother} \sqsubseteq \text{Woman}$

$\text{Man} \equiv \text{Person} \cap \neg \text{Woman}$

$\text{Woman} \equiv \text{Person} \cap \text{Female}$

$\text{Mother} \equiv \text{Woman} \cap \exists \text{hasChild}.\text{Person}$

$\text{Father} \equiv \text{Man} \cap \exists \text{hasChild}.\text{Person}$

$\text{Parent} \equiv \text{Father} \cup \text{Mother}$

$\text{MotherWithManyChildren} \equiv \text{Mother} \cap \geq 3 \text{hasChild}$

$\text{MotherWithoutDaughter} \equiv \text{Mother} \cap \forall \text{hasChild}.\neg \text{Woman}$

$\text{ancestor} \equiv \text{hasChild}^+$

## Extended SHIQ DL language

	DL notation	RACER syntax
atomic concept	A	:atomic-concepts (A)
top concept	T	*top*
bottom concept	$\perp$	*bottom*
negation	$\neg C$	(not C)
conjunction	$C_1 \cap \dots \cap C_n$	(and $C_1 \dots C_n$ )
disjunction	$C_1 \cup \dots \cup C_n$	(or $C_1 \dots C_n$ )
exists restriction	$\exists R.C$	(some R C)
value restriction	R.C	(all R C)

## Extended SHIQ DL language

	DL notation	RACER syntax
at-least restriction	$\geq n R$	(at-least n R)
at-most restriction	$\leq n R$	(at-most n R)
exactly restriction	$= n R$	(exactly n R)
qualified at-least restriction	$\geq n R.C$	(at-least n R C)
qualified at-most restriction	$\leq n R.C$	(at-most n R C)
role name	R	:roles(R)
inverse role	$R^-$	(inv R)
transitive role	$R^+$	(R :transitive)

## Extended SHIQ DL language

- Concrete domain concepts and attribute expressions
  - cardinal, integer, real, complex, string
  - min/max restriction over integers
  - linear polynomial equations over the reals or cardinals
  - nonlinear polynomial equations over complex numbers
  - equalities and inequalities of strings

## Concept axioms and terminology

- Statements:
  - subsumption relation between two concepts
  - equivalence between two concepts
  - pairwise disjointness between several concepts
  - subsumption relation between a concept name and a concept term
  - equality between a concept name and a concept term

## Role declarations

- only one declaration for each role
- features (attributes)
- transitive roles
- role hierarchy
- declaration of domain and range restrictions for roles

## Reasoning on DL

## Reasoning

- Input: knowledge base (TBox, ABox)
- queries
  - TBox queries: concept satisfiability, consistency, concept hierarchy
  - ABox queries: concept instances, instance retrieval, inconsistency check
- DL: decidable

## Assumptions

- Open World Assumption
  - what cannot be proven to be true is not believed to be false
- Unique Name Assumption
  - all individuals used in an Abox are assumed to be mapped to different elements of the universe

## RACER: A Reasoning Tool

- Renamed ABox and Concept Expression Reasoner
  - extension of the SHIQ logic  
(concrete domain concepts, attribute expressions)
  - highly optimized tableau calculus
  - built-in reasoner for several ontology management tools
  - support of general terminological axioms
  - multiple definition and cyclic definition of concepts is allowed

## RACER - Application areas

- Semantic web
- E-business
- Medicine/Bioinformatics
- Natural language processing
- Knowledge engineering
- Software engineering

## Most important queries (on type level)

- concept consistency w.r.t. a Tbox
- concept subsumption w.r.t. a TBox
- determining concept hierarchy
- concept equivalence
- concept disjointness

## Most important queries (on instance level)

- consistency check of an Abox w.r.t. a TBox
- instance testing w.r.t. an Abox and a Tbox
- instance retrieval w.r.t. an Abox and a Tbox
- computation of the direct type of an instance w.r.t. an Abox and a Tbox
- concrete domain constraints check