

Description Logics

Computational Models and Languages L-M (2010-2011)

Elena Nardini

ALMA MATER STUDIORUM – Università di Bologna

`elena.nardini@unibo.it`

Cesena - December 17, 2010



- 1 Introduction
- 2 Knowledge Representation with DL
- 3 DL Families
- 4 DL Tools
 - Ontology Editors
 - Reasoners
- 5 Fuzzy Description Logics
- 6 Fuzzy DL Tools
 - Ontology Editors
 - Reasoners
- 7 Working with DL and Fuzzy DL
 - Semantic Tuple Centres
 - Chemical Tuple Spaces
- 8 Available Projects
- 9 Bibliography



Outline

- 1 Introduction
- 2 Knowledge Representation with DL
- 3 DL Families
- 4 DL Tools
 - Ontology Editors
 - Reasoners
- 5 Fuzzy Description Logics
- 6 Fuzzy DL Tools
 - Ontology Editors
 - Reasoners
- 7 Working with DL and Fuzzy DL
 - Semantic Tuple Centres
 - Chemical Tuple Spaces
- 8 Available Projects
- 9 Bibliography



Knowledge Representation

- The **Knowledge Representation** (KR) [Brachman and Levesque, 2003] field studies techniques and methods to provide high-level descriptions of the world to be exploited to build intelligent systems
- For intelligent systems we intend systems able to reason upon such descriptions, that is:
 - ▶ able to interpret the **explicit** knowledge
 - ▶ able to infer new **implicit** knowledge
- Such kind of intelligent systems is useful in modelling application domains in open context – like Web or pervasive systems – where is not possible to have a complete prior knowledge about the world



Description Logics (DL) [Baader et al., 2003]

- Introduced in the 1980
- Family of KR formalisms representing the knowledge of an application domain (the “world”) by
 - ▶ first defining the relevant concepts of the domain ([the terminology](#))
 - ▶ then using these concepts to specify properties of objects and individuals occurring in the domain ([the world description](#))
- DL are equipped with a formal, logic-based semantics
- Reasoning as a central service: reasoning allows one to infer implicitly represented knowledge from the knowledge that is explicitly contained in the knowledge base



Important Issues

- DL are a KR formalism. A KR system should always answer the queries of a user in reasonable time
- The reasoning procedures should always terminate, both for positive and for negative answers
- Decidability and complexity of the inference problems depend on the expressive power of the DL at hand
 - ▶ Very expressive DL are likely to have inference problems of high complexity, or they may even be undecidable
 - ▶ Very weak DL (with efficient reasoning procedures) may not be sufficiently expressive to represent the important concepts of a given application
- Investigating this trade-off between the expressivity of DL and the complexity of their reasoning problems has been one of the most important issues in DL research



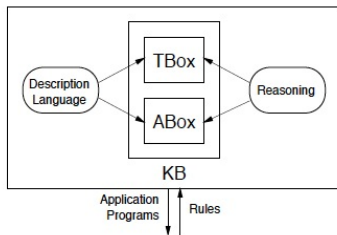
Outline

- 1 Introduction
- 2 Knowledge Representation with DL**
- 3 DL Families
- 4 DL Tools
 - Ontology Editors
 - Reasoners
- 5 Fuzzy Description Logics
- 6 Fuzzy DL Tools
 - Ontology Editors
 - Reasoners
- 7 Working with DL and Fuzzy DL
 - Semantic Tuple Centres
 - Chemical Tuple Spaces
- 8 Available Projects
- 9 Bibliography



Knowledge Base

- KR system based on Description Logics provides facilities to set up **knowledge bases**, to reason about their content and to manipulate them
- A knowledge base (KB) comprises two components, the **TBox** and the **ABox**
 - ▶ The TBox contains the **intensional knowledge**: it introduces the terminology, i.e., the vocabulary of an application domain
 - ▶ The ABox contains the **extensional knowledge**: it describes the assertions about named individuals in terms of this vocabulary



- It is subject to few changes
- It includes the so-called terminological axioms
 - ▶ **Concept descriptions**, denoting the meaningful sets of individuals of the domain
 - ▶ **Role descriptions**, denoting relationships among individuals
 - ▶ **Taxonomy descriptions** of concepts and roles
- Concept and role descriptions depend on the specific DL
- Taxonomy description through the operators $\sqsubseteq$ and $\equiv$ for expressing respectively the concept or role **inclusion** and **equality**



TBox Example

Woman $\equiv$ Person $\sqcap$ Female

Man $\equiv$ Person $\sqcap$ $\neg$ Woman

Mother $\equiv$ Woman $\sqcap$ $\exists$ hasChild.Person

Father $\equiv$ Man $\sqcap$ $\exists$ hasChild.Person

Parent $\equiv$ Father $\sqcup$ Mother

Grandmother $\equiv$ Mother $\sqcap$ $\exists$ hasChild.Parent

MotherWithManyChildren $\equiv$ Mother $\sqcap$ ≥ 3 hasChild

MotherWithoutDaughter $\equiv$ Mother $\sqcap$ hasChild. $\neg$ Woman

Wife $\equiv$ Woman $\sqcap$ $\exists$ hasHusband.Man



- It is subject to more changes than the TBox
- It defines axioms to assert specific domain objects and their properties
- They can be of kind
 - ▶ $C(a)$, declaring the individual a and the concept C it belongs to
 - ▶ $R(a,b)$, declaring that the role R relates individual a with b



ABox Example

MotherWithoutDaughter(MARY)

Father(PETER)

hasChild(MARY, PETER)

hasChild(PETER, HARRY)

hasChild(MARY, PAUL)



OWA and UNA

- **Open World Assumption** (OWA) assumes that what is not asserted does not be assumed to be false
 - ▶ Additional assertions are **monotonic**: new information cannot retract previous information
- **Unique Name Assumption** (UNA) assumes that if **a** and **b** are distinct names, then $a' \neq b'$ do OWA and generally do not UNA
 - OWA and no-UNA deal with the open context where one can not assume that the knowledge in the KB is complete



Integrity Constraints with OWA and no-UNA

- OWA and no-UNA lead to more difficulties in modelling **integrity constraints**

- **Example:**

$\text{Product} \sqsubseteq (\geq 1 \text{ isManufacturedBy})$
 $(\geq 1 \text{ isManufacturedBy}) \sqsubseteq \text{Product}$
 $\top \sqsubseteq \forall \text{isManufacturedBy. Manufacturer}$
 $\text{isManufacturedBy}(\text{product1}, \text{ACME})$

The range restriction causes the inference that ACME is a Manufacturer, rather than causing a range violation—since ACME is not explicitly defined to be a Manufacturer instance

- The same kind of unexpected inferences occur when using cardinality constraints: missing values do not cause an inconsistency and extra values cause two individuals to be inferred the same individual



Reasoning upon TBox

- **Subsumption** is the basic inference on concept expressions checking if the concept D – the **subsumer** – is considered more general than the one denoted by C – the **subsumee** –, that is, if $C \sqsubseteq D$
- **Satisfiability**, which is the problem of checking whether a concept expression does not necessarily denote the empty concept. This is a special case of subsumption, with the subsumer being the empty concept, meaning that a concept is not satisfiable
- **Classification** constructs the subsumption hierarchy between all concepts defined in a terminology



Reasoning upon ABox

- **Consistency checking** checks if an the Abox is consistent with respect to the TBox. It amounts to verifying whether every concept in the knowledge base admits at least one individual
- **Instance checking** checks if an individual a belongs to a concept C
- **Retrieval**, which finds the individuals in the knowledge base that are instances of a given concept
- **Realization** which finds the individuals in the knowledge base that are instances of a given concept



ABox and TBox vs Relational DB

- Why do not use a DB schema instead of a TBox and table records instead of ABox axioms?
- DL approach is **concept-oriented**, that is, it aims to describe concepts and roles
- Relational DB approach is **data-oriented**, that is, it aims to describe the entities that will be stored in the DB tables
- DL uses OWA, relational DB uses CWA



ABox and TBox vs Relational DB: an Example

- We want to describe the following knowledge
 - ▶ Mario is Marco's father
 - ▶ Marco is Andrea's brother
 - ▶ Bruno is Andrea's uncle



Data-oriented Approach

- In order to obtain complete information, for each person it is needed to specify in an explicit way its relations with other persons
 - ▶ Mario is a person, he has a son that is a person with name Marco. Mario has not father, brothers and uncles
 - ▶ Marco is a person, he has a brother that is a person with name Andrea and a father that is a person with name Mario. Marco has not sons and uncles
 - ▶ Bruno is a person, he has a nephew that is a person with name Andrea. Bruno has not father, sons and uncles
 - ▶ Andrea is a person with a brother with name Marco and an uncle with name Bruno
- The implicit information can be inferred: Bruno is Marco's uncle. This require a reasoning on the concepts brother, uncle and father



Concept-oriented Approach

- It is possible to define the following concepts:
 - ▶ **Father**: a person with at least a son
 - ▶ **Brother**: a person with the same parents of another person
 - ▶ **Uncle**: a person with a brother that is a father
- With the same information we can infer all the implicit knowledge:
Bruno is Marco's uncle



Outline

- 1 Introduction
- 2 Knowledge Representation with DL
- 3 DL Families**
- 4 DL Tools
 - Ontology Editors
 - Reasoners
- 5 Fuzzy Description Logics
- 6 Fuzzy DL Tools
 - Ontology Editors
 - Reasoners
- 7 Working with DL and Fuzzy DL
 - Semantic Tuple Centres
 - Chemical Tuple Spaces
- 8 Available Projects
- 9 Bibliography



AL Language

- **Attributive Language** (AL) is a known DL with the minimal DL features
- Concept description in TBox through the following syntax

C, D	$\longrightarrow$	A		(atomic concept)
		$\top$		(concetto universale)
		$\perp$		(concetto <i>bottom</i>)
		$\neg A$		(negazione atomica)
		$C \sqcap D$		(intersezione)
		$\forall R.C$		(restrizione di valore)
		$\exists R.\top$		(quantificazione esistenziale limitata)



AL Semantic

- In order to give a formal semantic to AL, an interpretation with an interpretation function have to be considered
- An interpretation $\mathcal{I}$ consists of a non-empty set $\Delta^{\mathcal{I}}$, that is the domain of the interpretation
- An interpretation function assigns to every atomic concept A a set $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ and to every atomic role R a binary relation $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
- The interpretation function is extended to concept descriptions by the following inductive definitions:

$$\begin{aligned}\top^{\mathcal{I}} &= \Delta^{\mathcal{I}} \\ \perp^{\mathcal{I}} &= \emptyset \\ (\neg A)^{\mathcal{I}} &= \Delta^{\mathcal{I}} \setminus A^{\mathcal{I}} \\ (C \sqcap D)^{\mathcal{I}} &= C^{\mathcal{I}} \cap D^{\mathcal{I}} \\ (\forall R.C)^{\mathcal{I}} &= \{a \in \Delta^{\mathcal{I}} \mid \forall b. (a, b) \in R^{\mathcal{I}} \rightarrow b \in C^{\mathcal{I}}\} \\ (\exists R.\top)^{\mathcal{I}} &= \{a \in \Delta^{\mathcal{I}} \mid \exists b. (a, b) \in R^{\mathcal{I}}\}\end{aligned}$$



Different DL Families

Famiglia	Elementi supportati
$\mathcal{AL}$	Negazione atomica • Intersezione • Restrizione di valore • Quantificazione esistenziale limitata
$\mathcal{ALC}$	$\mathcal{AL}$ più: • Negazione • Unione • Quantificazione esistenziale completa
$\mathcal{S}$	$\mathcal{ALC}$ più: • Ruoli transitivi
$\mathcal{SH}$	$\mathcal{S}$ più: • Gerarchie di ruoli
$\mathcal{SHI}$	$\mathcal{SH}$ più: • Ruoli inversi
$\mathcal{SHIF}$	$\mathcal{SHI}$ più: • Ruoli funzionali
$\mathcal{SHIN}$	$\mathcal{SHI}$ più: • Restrizioni di cardinalità
$\mathcal{SHOIN}$	$\mathcal{SHIN}$ più: • <i>Nominals</i>
$\mathcal{SHOIN}(\mathcal{D})$	$\mathcal{SHOIN}$ più: • Supporto ai <i>datatype</i>



SHIF(D) Language

- The theoretical counterpart of **OWL Lite**—that is one of the three species of **OWL**: OWL Full, OWL DL and OWL Lite [Horrocks et al., 2003]
 - ▶ OWL is the standard ontology description language for Semantic Web
- Key inferences can be computed in worst case exponential time (*ExpTime*)
- There are already several optimised reasoners for logics equivalent to OWL Lite
- This comes with relatively little loss in expressive power. It is possible to express complex descriptions



SHOIN(D) and SROIQ(D) Language

- SHOIN(D) is the theoretical counterpart of OWL DL
- SHOIN(D) key inferences can be computed in worst-case nondeterministic exponential time ($NExpTime$)
- Recently was introduced OWL 2 DL, that is an extension of OWL DL, corresponding to SROIQ(D)
- SROIQ(D) adds to SHOIN(D)
 - ▶ Reflexive roles
 - ▶ Irreflexive roles
 - ▶ Asymmetric roles
 - ▶ ...
- SROIQ(D) key inferences can be computed in worst-case nondeterministic exponential time ($N2ExpTime$)



SHOIN(D) Syntax

$$C \longrightarrow \top \mid \perp \mid A \mid C_1 \sqcap C_2 \mid C_1 \sqcup C_2 \mid \neg C \mid \\ \forall R.C \mid \exists R.C \mid (\geq n S) \mid (\leq n S) \mid \{a_1, \dots, a_n\} \mid \\ (\geq n T) \mid (\leq n T) \mid \forall T_1, \dots, T_n.D \mid \exists T_1, \dots, T_n.D$$

$$D \longrightarrow d \mid \{c_1, \dots, c_n\}$$



SHOIN(D) Semantic

$$\begin{aligned}\top^{\mathcal{I}} &= \Delta^{\mathcal{I}} \\ \perp^{\mathcal{I}} &= \emptyset \\ (C_1 \sqcap C_2)^{\mathcal{I}} &= C_1^{\mathcal{I}} \cap C_2^{\mathcal{I}} \\ (C_1 \sqcup C_2)^{\mathcal{I}} &= C_1^{\mathcal{I}} \cup C_2^{\mathcal{I}} \\ (\neg C)^{\mathcal{I}} &= \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}} \\ (S^-)^{\mathcal{I}} &= \{\langle y, x \rangle : \langle x, y \rangle \in S^{\mathcal{I}}\} \\ (\forall R.C)^{\mathcal{I}} &= \{x \in \Delta^{\mathcal{I}} : R^{\mathcal{I}}(x) \subseteq C^{\mathcal{I}}\} \\ (\exists R.C)^{\mathcal{I}} &= \{x \in \Delta^{\mathcal{I}} : R^{\mathcal{I}}(x) \cap C^{\mathcal{I}} \neq \emptyset\} \\ (\geq n S)^{\mathcal{I}} &= \{x \in \Delta^{\mathcal{I}} : |S^{\mathcal{I}}(x)| \geq n\} \\ (\leq n S)^{\mathcal{I}} &= \{x \in \Delta^{\mathcal{I}} : |S^{\mathcal{I}}(x)| \leq n\} \\ \{a_1, \dots, a_n\}^{\mathcal{I}} &= \{a_1^{\mathcal{I}}, \dots, a_n^{\mathcal{I}}\}\end{aligned}$$



Outline

- 1 Introduction
- 2 Knowledge Representation with DL
- 3 DL Families
- 4 DL Tools**
 - Ontology Editors
 - Reasoners
- 5 Fuzzy Description Logics
- 6 Fuzzy DL Tools
 - Ontology Editors
 - Reasoners
- 7 Working with DL and Fuzzy DL
 - Semantic Tuple Centres
 - Chemical Tuple Spaces
- 8 Available Projects
- 9 Bibliography



Outline

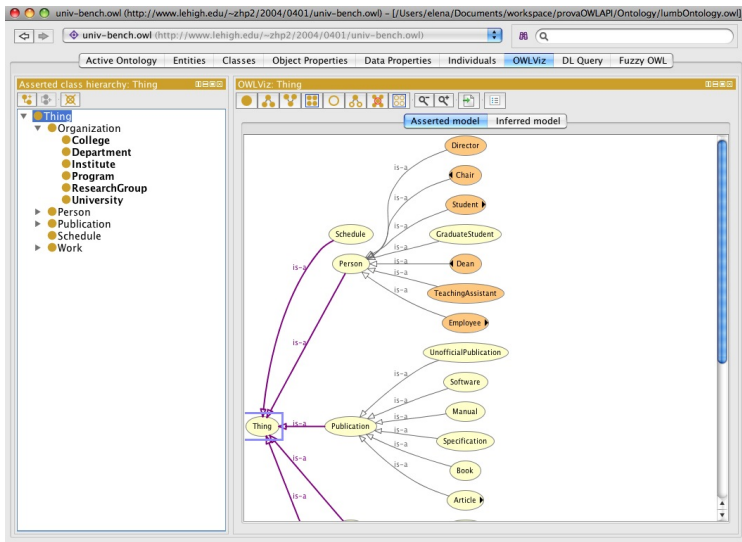
- 1 Introduction
- 2 Knowledge Representation with DL
- 3 DL Families
- 4 DL Tools**
 - **Ontology Editors**
 - Reasoners
- 5 Fuzzy Description Logics
- 6 Fuzzy DL Tools
 - Ontology Editors
 - Reasoners
- 7 Working with DL and Fuzzy DL
 - Semantic Tuple Centres
 - Chemical Tuple Spaces
- 8 Available Projects
- 9 Bibliography



Protégé

- Free and open source ontology editor based on Java
- Support OWL format (not yet OWL 2)
- Homepage: <http://protege.stanford.edu/>





Classes

univ-bench.owl (http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl) - [/Users/elena/Documents/workspace/provaOWLAPI/Ontology/lumbOntology.owl]

Active Ontology Entities **Classes** Object Properties Data Properties Individuals OWLViz DL Query Fuzzy OWL

Asserted class hierarchy Inferred class hierarchy

Asserted class hierarchy: Student

- Thing
 - Organization
 - College
 - Department
 - Institute
 - Program
 - ResearchGroup
 - University
 - Person
 - Chair
 - Dean
 - Director
 - Employee
 - AdministrativeStaff
 - Faculty
 - GraduateStudent
 - Student**
 - TeachingAssistant
 - Publication
 - Article
 - ConferencePaper
 - JournalArticle
 - TechnicalReport
 - Book
 - Manual
 - Software
 - Specification
 - UnofficialPublication

Class Annotations Class Usage

Annotations: Student

Annotations +

label
"student"

Description: Student

Equivalent classes +

- Person and takesCourse some Course

Superclasses +

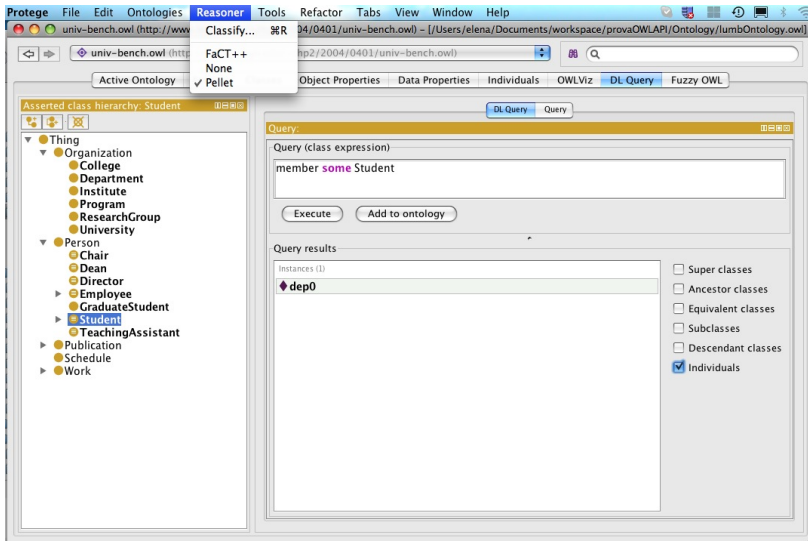
Inferred anonymous superclasses

Members +

- stud0



Reasoners



Outline

- 1 Introduction
- 2 Knowledge Representation with DL
- 3 DL Families
- 4 DL Tools**
 - Ontology Editors
 - **Reasoners**
- 5 Fuzzy Description Logics
- 6 Fuzzy DL Tools
 - Ontology Editors
 - Reasoners
- 7 Working with DL and Fuzzy DL
 - Semantic Tuple Centres
 - Chemical Tuple Spaces
- 8 Available Projects
- 9 Bibliography



Open Source Reasoners

- **Pellet** [Sirin et al., 2007], that is Java based and supports OWL DL and a lot of OWL 2 DL features
- **HermiT** [Shearer et al., 2008], that is Java based and supports OWL 2 DL
- **Kaon2**, that is Java based and supports the subset OWL DL, SHIQ(D)
 - ▶ Homepage: <http://kaon2.semanticweb.org/>
- **Fact++** [Tsarkov and Horrocks, 2006], that is C++ based and supports OWL 2 DL



Outline

- 1 Introduction
- 2 Knowledge Representation with DL
- 3 DL Families
- 4 DL Tools
 - Ontology Editors
 - Reasoners
- 5 Fuzzy Description Logics**
- 6 Fuzzy DL Tools
 - Ontology Editors
 - Reasoners
- 7 Working with DL and Fuzzy DL
 - Semantic Tuple Centres
 - Chemical Tuple Spaces
- 8 Available Projects
- 9 Bibliography



Fuzziness [Lukasiewicz and Straccia, 2008]

[... under vagueness/fuzziness theory fall all those approaches in which statements (for example, “the tomato is ripe”) are true to some degree, which is taken from a truth space. That is, an interpretation maps a statement to a truth degree, since we are unable to establish whether a statement is completely true or false due to the involvement of vague concepts, such as ripe which only have an imprecise definition. For example, we cannot exactly say whether a tomato is ripe or not, but rather can only say that the tomato is ripe to some degree. ... Vague statements are truth-functional, i.e., the degree of truth of a statement can be calculated from the degrees of truth of its constituents. ...]



Why Fuzziness [Zadeh, 1965]

[... More often than not, the classes of objects encountered in the real physical world do not have a precisely defined criteria of membership...] and [... imprecisely defined “classes” – for example, the “class of beautiful women” or the “class of tall men” – play an important role in human thinking, particularly in the domains of pattern recognition, communication of information, and abstraction...].



Towards Fuzzy DL

- DL are limited to deal with crisp and well defined concepts, individuals and queries
 - Since DL are knowledge representation languages, they have limits in modelling the real physical world
- Starting from the pioneering work by Yen (1991) [Yen, 1991], DL were the subject of several fuzzy extensions [Lukasiewicz and Straccia, 2008, Straccia, 2005, Bobillo et al., 2008].



Fuzzy SHIF(D) Syntax [Bobillo and Straccia, 2008]

- Each concept and role membership, $C(a)$ and $R(a,b)$, has associated a degree α

Example:

$\text{SportCar}(\text{audi_tt}) \geq 0.91$ states the individual `audi_tt` is a `SportCar` with degree `0.92`

- Fuzzy concepts and roles are described through the following syntax:

$$\begin{aligned} C, D &:= C \sqcap_G D \mid C \sqcap_L D \mid C \sqcup_G D \mid C \sqcup_L D \mid \\ &\quad C \rightarrow D \mid C \rightarrow_G D \mid C \rightarrow_L D \mid m(C) \mid \\ &\quad n C \mid w_1 C_1 + \dots + w_k C_k \mid C[\geq n] \mid C[\leq n] \\ m &:= \text{linear}(a) \mid \text{triangular}(a, b, c) \end{aligned}$$

- Fuzzy datatypes are described through the following syntax:

$$\begin{aligned} C, D &:= \forall T.d \mid \exists T.d \mid DR \\ d &:= \text{crisp}(a, b) \mid L(a, b) \mid R(a, b) \mid \\ &\quad \text{triangular}(a, b, c) \mid \text{trapezoidal}(a, b, c, d) \\ DR &:= \geq t \text{ val} \mid \leq t \text{ val} \mid = t \text{ val} \end{aligned}$$



Fuzzy SHIF(D) Semantic

$$\begin{aligned}
 \perp^I(x) &= 0 \\
 \top^I(x) &= 1 \\
 (\neg C)^I(x) &= \ominus C^I(x) \\
 (C \sqcap D)^I(x) &= C^I(x) \otimes D^I(x) \\
 (C \sqcap_G D)^I(x) &= C^I(x) \otimes_G D^I(x) \\
 (C \sqcap_{\mathbf{L}} D)^I(x) &= C^I(x) \otimes_{\mathbf{L}} D^I(x) \\
 (C \sqcup D)^I(x) &= C^I(x) \oplus D^I(x) \\
 (C \sqcup_G D)^I(x) &= C^I(x) \oplus_G D^I(x) \\
 (C \sqcup_{\mathbf{L}} D)^I(x) &= C^I(x) \oplus_{\mathbf{L}} D^I(x) \\
 (C \rightarrow D)^I(x) &= C^I(x) \Rightarrow D^I(x) \\
 (C \rightarrow_G D)^I(x) &= C^I(x) \Rightarrow_G D^I(x) \\
 (C \rightarrow_{KD} D)^I(x) &= C^I(x) \Rightarrow_{KD} D^I(x) \\
 (C \rightarrow_{\mathbf{L}} D)^I(x) &= C^I(x) \Rightarrow_{\mathbf{L}} D^I(x)
 \end{aligned}$$

$$\begin{aligned}
 (m(C))^I(x) &= f_m(C^I(x)) \\
 (\forall R.C)^I(x) &= \inf_{y \in \Delta^I} R^I(x, y) \Rightarrow C^I(y) \\
 (\exists R.C)^I(x) &= \sup_{y \in \Delta^I} R^I(x, y) \otimes C^I(y) \\
 (\forall T.d)^I(x) &= \inf_{y \in \Delta_D} T^I(x, y) \Rightarrow d^I(y) \\
 (\exists T.d)^I(x) &= \sup_{y \in \Delta_D} T^I(x, y) \otimes d^I(y) \\
 (n C)^I(x) &= n C^I(x) \\
 (w_1 C_1 + \dots + w_k C_k)^I(x) &= w_1 C_1^I(x) + \dots + w_k C_k^I(x) \\
 (C \geq n)^I(x) &= \begin{cases} C^I(x), & \text{if } C^I(x) \geq n \\ 0, & \text{otherwise} \end{cases} \\
 (C \leq n)^I(x) &= \begin{cases} C^I(x), & \text{if } C^I(x) \leq n \\ 0, & \text{otherwise} \end{cases} \\
 (\geq t \text{ val})^I(x) &= \sup_{c \in \Delta_D} t(x, v) \otimes (v \geq \text{val}) \\
 (\leq t \text{ val})^I(x) &= \sup_{c \in \Delta_D} t(x, v) \otimes (v \leq \text{val}) \\
 (= t \text{ val})^I(x) &= \sup_{c \in \Delta_D} t(x, v) \otimes (v = \text{val})
 \end{aligned}$$



Fuzzy Operators

Łukasiewicz negation	$\ominus_{\mathbf{L}} \alpha =$	$1 - \alpha$
Gödel t-norm	$\alpha \otimes_G \beta =$	$\min\{\alpha, \beta\}$
Łukasiewicz t-norm	$\alpha \otimes_{\mathbf{L}} \beta =$	$\max\{\alpha + \beta - 1, 0\}$
Gödel t-conorm	$\alpha \oplus_G \beta =$	$\max\{\alpha, \beta\}$
Łukasiewicz t-conorm	$\alpha \oplus_{\mathbf{L}} \beta =$	$\min\{\alpha + \beta, 1\}$
Gödel implication	$\alpha \Rightarrow_G \beta =$	$\begin{cases} 1, & \text{if } \alpha \leq \beta \\ \beta, & \text{if } \alpha > \beta \end{cases}$
Łukasiewicz implication	$\alpha \Rightarrow_{\mathbf{L}} \beta =$	$\min\{1, 1 - \alpha + \beta\}$
Kleene-Dienes implication	$\alpha \Rightarrow_Z \beta =$	$\max\{1 - \alpha, \beta\}$

- **Example:**

`SportCar(audi_tt) ≥ 0.91`

`hasMaker(audi_tt, audi) ≥ 1`

Let's take the concept $C \equiv [\geq 0.8 \text{ SportCar}] \sqcup_G [\geq 0.2 \text{ CityCar}]$

Then, the individual `audi_tt` belongs to C with degree ≥ 0.73

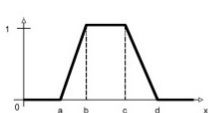


Fuzzy SHIF(D) Datatype

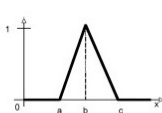
- Fuzzy concrete domains and thus datatypes are based on fuzzy sets
- Functions for specifying fuzzy-set-membership degrees, are used
- **Example:**

We can define the datatype predicate **High** as

$\text{High}(x) = \text{rs}(x; 80, 250)$, where **rs** is the *right-shoulder function* that is shown in Figure (d)



(a)



(b)



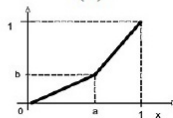
(c)



(d)



(e)



(f)



Fuzzy SHIF(D) Modifiers

- Functions applied to fuzzy sets to change their membership function
- A fuzzy modifier m represents a function $f_m: [0,1] \rightarrow [0,1]$
- **Example:**

We may define $f_{\text{very}}(x) = x^2$

The datatype predicate **High** We can use **very(High)** to define the semantic of the fuzzy concept **very high**



Outline

- 1 Introduction
- 2 Knowledge Representation with DL
- 3 DL Families
- 4 DL Tools
 - Ontology Editors
 - Reasoners
- 5 Fuzzy Description Logics
- 6 Fuzzy DL Tools**
 - Ontology Editors
 - Reasoners
- 7 Working with DL and Fuzzy DL
 - Semantic Tuple Centres
 - Chemical Tuple Spaces
- 8 Available Projects
- 9 Bibliography



Outline

- 1 Introduction
- 2 Knowledge Representation with DL
- 3 DL Families
- 4 DL Tools
 - Ontology Editors
 - Reasoners
- 5 Fuzzy Description Logics
- 6 Fuzzy DL Tools**
 - Ontology Editors
 - Reasoners
- 7 Working with DL and Fuzzy DL
 - Semantic Tuple Centres
 - Chemical Tuple Spaces
- 8 Available Projects
- 9 Bibliography



- Fuzzy OWL 2 Protégé plug-in
- Homepage: <http://gaia.isti.cnr.it/~straccia/software/FuzzyOWL/index.html>



Fuzzy OWL

FuzzyTest (http://www.semanticweb.org/ontologies/2010/8/FuzzyTest.owl) - [Users/straccia/Research/Tirocini/Bacarella/FuzzyTest.owl]

FuzzyTest (http://www.semanticweb.org/ontologies/2010/8/FuzzyTest.owl)

Active Ontology | Entities | Classes | Object Properties | Data Properties | Individuals | OWLviz | DL Query | SoftFacts Tab | Fuzzy OWL

Fuzzy Owl:

Menu

- Fuzzy Datatype
- Fuzzy Modified Concept
- Weighted Concept
- Weighted Sum Concept
- Fuzzy Nominal
- Fuzzy Modifier
- Fuzzy Modified Role
- Fuzzy Axiom
- Ontology
- Fuzzy Modified Datatype
- Add new datatype
- HighPower
- HighSpeed
- VeryCold
- Cold
- Expensive

Step 2
Choose the type and fill with parameters

Type
rightsoulder

A 200.0
B 300.0
K1 0.0
K2 1000.0
Annotate



Annotations:

Annotations

Annotations: HighPower

Annotations

```

<fuzzyowl2 fuzzyType='datatype'>
  <Datatype type='rightsoulder' a='200'
    b='300' />
</fuzzyowl2>^^PlainLiteral
  
```

Annotations:

Annotations

Axiom annotations

Annotations

To use the reasoner click Reasoner→Start Reasoner ☒ Show Inferences



Outline

- 1 Introduction
- 2 Knowledge Representation with DL
- 3 DL Families
- 4 DL Tools
 - Ontology Editors
 - Reasoners
- 5 Fuzzy Description Logics
- 6 Fuzzy DL Tools**
 - Ontology Editors
 - **Reasoners**
- 7 Working with DL and Fuzzy DL
 - Semantic Tuple Centres
 - Chemical Tuple Spaces
- 8 Available Projects
- 9 Bibliography



Fuzzy Reasoners

- fuzzyDL for SHIF(D)
 - ▶ Homepage: <http://gaia.isti.cnr.it/~straccia/software/fuzzyDL/fuzzyDL.html>
- DeLorean for SHOIN(D) and SHROIQ(D)
 - ▶ Only produces no-fuzzy owl describing a fuzzy ontology
 - Can be used with Pellet [Straccia, 2001]
 - ▶ Homepage: <http://webdiis.unizar.es/~fbobillo/delorean>
- In <http://gaia.isti.cnr.it/~straccia/software/FuzzyOWL/index.html> there are parsers to parse Protégé Fuzzy OWL in fuzzyDL and DeLorean Fuzzy OWL



Outline

- 1 Introduction
- 2 Knowledge Representation with DL
- 3 DL Families
- 4 DL Tools
 - Ontology Editors
 - Reasoners
- 5 Fuzzy Description Logics
- 6 Fuzzy DL Tools
 - Ontology Editors
 - Reasoners
- 7 Working with DL and Fuzzy DL
 - Semantic Tuple Centres
 - Chemical Tuple Spaces
- 8 Available Projects
- 9 Bibliography



Outline

- 1 Introduction
- 2 Knowledge Representation with DL
- 3 DL Families
- 4 DL Tools
 - Ontology Editors
 - Reasoners
- 5 Fuzzy Description Logics
- 6 Fuzzy DL Tools
 - Ontology Editors
 - Reasoners
- 7 Working with DL and Fuzzy DL**
 - **Semantic Tuple Centres**
 - Chemical Tuple Spaces
- 8 Available Projects
- 9 Bibliography



Semantic tuple centre model

- Extension of the tuple centre model [Nardini et al., 2010]
- Characterised by
 - ▶ **Domain ontology** allowing to interpret the semantics associated to the knowledge (set of tuples) stored into a tuple centre
 - ▶ **Semantic tuples** describing domain objects so that they can be interpreted in a semantic way, by means of the domain ontology
 - ▶ **Semantic templates** as descriptions of a domain object set
 - ▶ **Semantic matching** providing the domain objects – tuples – described through templates
- Such components are formally described through SHOIN(D)



Semantic Tuple Centres in TuCSoN

- Domain ontology described in OWL-DL
- Semantic tuples and templates described through a SHOIN(D)-like language
- Semantic matching mechanisms by exploiting Pellet reasoner



Outline

- 1 Introduction
- 2 Knowledge Representation with DL
- 3 DL Families
- 4 DL Tools
 - Ontology Editors
 - Reasoners
- 5 Fuzzy Description Logics
- 6 Fuzzy DL Tools
 - Ontology Editors
 - Reasoners
- 7 Working with DL and Fuzzy DL**
 - Semantic Tuple Centres
 - Chemical Tuple Spaces**
- 8 Available Projects
- 9 Bibliography



Main Idea [Viroli et al., 2010]

- Tuple spaces + chemical-like reactions as coordination laws
- Tuples have a concentration (a.k.a. weight, or activity value)
- Concentration is evolved “exactly” as in chemistry
- Some reactions can even fire a tuple from one space to another



Semantic and Fuzziness in Chemical Tuple Spaces

- A matching mechanism to select reactants for a coordination law, it is needed
 - ▶ We use semantic matching in order to deal with openness requirements
 - ▶ We use an application-dependent fuzzy match function $\mu(t, t')$ in order to deal with vagueness of information



Outline

- 1 Introduction
- 2 Knowledge Representation with DL
- 3 DL Families
- 4 DL Tools
 - Ontology Editors
 - Reasoners
- 5 Fuzzy Description Logics
- 6 Fuzzy DL Tools
 - Ontology Editors
 - Reasoners
- 7 Working with DL and Fuzzy DL
 - Semantic Tuple Centres
 - Chemical Tuple Spaces
- 8 Available Projects
- 9 Bibliography



Available Projects

- Embed Pellet in tuProlog as a library
- Embed and test fuzzyDL in tuProlog as a library
- Integrate and test DeLorean with Pellet



Outline

- 1 Introduction
- 2 Knowledge Representation with DL
- 3 DL Families
- 4 DL Tools
 - Ontology Editors
 - Reasoners
- 5 Fuzzy Description Logics
- 6 Fuzzy DL Tools
 - Ontology Editors
 - Reasoners
- 7 Working with DL and Fuzzy DL
 - Semantic Tuple Centres
 - Chemical Tuple Spaces
- 8 Available Projects
- 9 Bibliography



Bibliography I



Baader, F., Calvanese, D., McGuinness, D. L., Nardi, D., and Patel-Schneider, P. F., editors (2003).

The Description Logic Handbook: Theory, Implementation, and Applications.

Cambridge University Press.



Bobillo, F., Delgado, M., and Gómez-Romero, J. (2008).

A crisp representation for fuzzy SHOIN with fuzzy nominals and general concept inclusions.

In da Costa, P. C. G., d'Amato, C., Fanizzi, N., Laskey, K. B., Laskey, K. J., Lukasiewicz, T., Nickles, M., and Pool, M., editors, *Uncertainty Reasoning for the Semantic Web I*, volume 5327 of *LNCS*, pages 174–178. Springer.



Bibliography II



Bobillo, F. and Straccia, U. (2008).

FuzzyDL: An expressive fuzzy description logic reasoner.

In *2008 International Conference on Fuzzy Systems (FUZZ-IEEE 2008)*, pages 923–930. IEEE CS.



Brachman, R. and Levesque, H., editors (2003).

Amsterdam: Boston.



Horrocks, I., Patel-Schneider, P. F., and Harmelen, F. V. (2003).

From shiq and rdf to owl: The making of a web ontology language.

Journal of Web Semantics, 1:2003.



Lukasiewicz, T. and Straccia, U. (2008).

Managing uncertainty and vagueness in Description Logics for the Semantic Web.

Web Semantics: Science, Services and Agents on the World Wide Web, 6(4):291–308.



Bibliography III



Nardini, E., Viroli, M., and Panzavolta, E. (2010).

Coordination in open and dynamic environments with tucson semantic tuple centres.

In 25th Annual ACM Symposium on Applied Computing (SAC 2010), volume III, pages 2037–2044, Sierre, Switzerland. ACM.



Shearer, R., Motik, B., and Horrocks, I. (2008).

Hermit: A highly-efficient owl reasoner.

In The OWL Experiences and Directions Workshop at the ISWC 08.



Sirin, E., Parsia, B., Grau, B. C., Kalyanpur, A., and Katz, Y. (2007).

Pellet: A practical owl-dl reasoner.

J. Web Sem., 5(2):51–53.



Straccia, U. (2001).

Reasoning within fuzzy description logics.

JAIR, 14:137–166.



Bibliography IV



Straccia, U. (2005).

Description logics with fuzzy concrete domains.

In Bachus, F. and Jaakkola, T., editors, *21st Conference on Uncertainty in Artificial Intelligence (UAI-05)*, pages 559–567, Edinburgh, Scotland. AUA Press.



Tsarkov, D. and Horrocks, I. (2006).

Fact++ description logic reasoner: System description.

In *In Proc. of the Int. Joint Conf. on Automated Reasoning (IJCAR 2006)*, pages 292–297. Springer.



Viroli, M., Casadei, M., Nardini, E., and Omicini, A. (2010).

Towards a chemical-inspired infrastructure for self-* pervasive applications.

In Weyns, D., Malek, S., de Lemos, R., and Andersson, J., editors, *Self-Organizing Architectures*, volume 6090 of *LNCs*, chapter 8, pages 152–176. Springer.



Bibliography V

1st International Workshop on Self-Organizing Architectures (SOAR 2009), Cambridge, UK, 14-17 September 2009, Revised Selected and Invited Papers.



Yen, J. (1991).

Generalizing term subsumption languages to fuzzy logic.

In *12th International Joint Conference on Artificial Intelligence (IJCAI'91)*, pages 472–477, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.



Zadeh, L. A. (1965).

Fuzzy sets.

Information and Control, 8(3):338–353.



Description Logics

Computational Models and Languages L-M (2010-2011)

Elena Nardini

ALMA MATER STUDIORUM – Università di Bologna

`elena.nardini@unibo.it`

Cesena - December 17, 2010

