Fuzzy Description Logics and the Semantic Web

Umberto Straccia

ISTI - CNR, Pisa ITALY
straccia@isti.cnr.it

Milano 2006

"Calla is a very large, long white flower on thick stalks"

Outline

The Semantic Web and Ontologies (excerpt)

The Semantic Web Vision

- The WWW as we know it now
  - 1st generation web mostly handwritten HTML pages
  - 2nd generation (current) web often machine generated/active
  - Both intended for direct human processing/interaction
- In next generation web, resources should be more accessible to automated processes
  - To be achieved via semantic markup
  - Metadata annotations that describe content/function

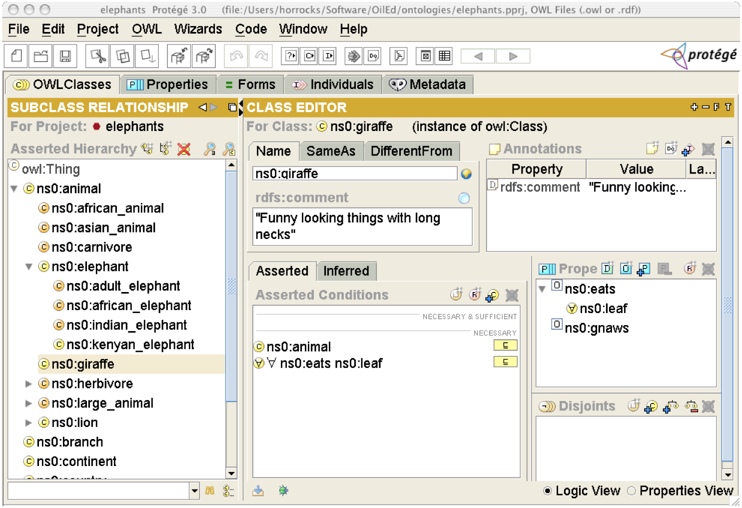
Ontologies

- Semantic markup must be meaningful to automated processes
- Ontologies will play a key role
  - Source of precisely defined terms (vocabulary)
  - Can be shared across applications (and humans)
- Ontology typically consists of:
  - Hierarchical description of important concepts in domain
  - Descriptions of properties of instances of each concept
- Ontologies can be used, e.g.
  - To facilitate agent-agent communication in e-commerce
  - In semantic based search
  - To provide richer service descriptions that can be more flexibly interpreted by intelligent agents

Example Ontology

- Vocabulary and meaning ("definitions")
  - Elephant is a concept whose members are a kind of animal
  - Herbivore is a concept whose members are exactly those animals who eat only plants or parts of plants
  - Adult_Elephant is a concept whose members are exactly those elephants whose age is greater than 20 years
- Background knowledge/constraints on the domain ("general axioms")
  - Adult_Elephants weigh at least 2,000 kg
  - All Elephants are either African_Elephants or Indian_Elephants
  - No individual can be both a Herbivore and a Carnivore

Example Ontology (Protégé)



Ontology Description Languages

- Should be sufficiently expressive to capture most useful aspects of domain knowledge representation
- Reasoning in it should be decidable and efficient
- Many different languages has been proposed: RDF, RDFS, OIL, DAML+OIL
- OWL (Ontology Web Language) is the current emerging language. There are three species of OWL
  - OWL full is union of OWL syntax and RDF (but, undecidable)
  - OWL DL restricted to FOL fragment (reasoning problem in NEXPTIME)
    - based on $\mathcal{SHIQ}$ Description Logic ($\mathcal{ALCHIQR}_+$)
  - OWL Lite is "easier to implement" subset of OWL DL (reasoning problem in EXPTIME)
    - based on $\mathcal{SHIF}$ Description Logic ($\mathcal{ALCHIFR}_+$)
- SWRL, a Semantic Web Rule Language combines OWL and RuleML (not addressed here)

Description Logics (excerpt)

Description Logics Basics

(the logics behind OWL, `http://dl.kr.org/`)

- Concept/Class: names are equivalent to unary predicates
  - In general, concepts equiv to formulae with one free variable
- Role or attribute: names are equivalent to binary predicates
  - In general, roles equiv to formulae with two free variables
- Taxonomy: Concept and role hierarchies can be expressed
- Individual: names are equivalent to constants
- Operators: restricted so that:
  - Language is decidable and, if possible, of low complexity
  - No need for explicit use of variables
    - Restricted form of $\exists$ and $\forall$
  - Features such as counting can be succinctly expressed

The DL Family

- A given DL is defined by set of concept and role forming operators
- Basic language: $\mathcal{ALC}$($\mathcal{A}$ttributive $\mathcal{L}$anguage with $\mathcal{C}$omplement)

Syntax				Semantics	Example
C, D	$\rightarrow$	$\top$	|	$\top(x)$	
		$\bot$	|	$\bot(x)$	
		A	|	$A(x)$	$Human$
		$C \sqcap D$	|	$C(x) \land D(x)$	$Human \sqcap Male$
		$C \sqcup D$	|	$C(x) \lor D(x)$	$Nice \sqcap Rich$
		$\neg C$	|	$\neg C(x)$	$\neg Meat$
		$\exists R.C$	|	$\exists y.R(x,y) \land C(y)$	$\exists has\_child.Blond$
		$\forall R.C$		$\forall y.R(x,y) \Rightarrow C(y)$	$\forall has\_child.Human$
$C \sqsubseteq D$				$\forall x.C(x) \Rightarrow D(x)$	$Happy\_Father \sqsubseteq Man \sqcap \exists has\_child.Female$
$a{:}C$				$C(a)$	$John{:}Happy\_Father$

DLs Semantics

- Interpretation: $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where $\Delta^{\mathcal{I}}$ is the domain (a non-empty set), $\cdot^{\mathcal{I}}$ is an interpretation function that maps:
  - Concept (class) name A into a function $A^{\mathcal{I}}: \Delta^{\mathcal{I}} \to \{0, 1\}$
  - Role (property) name R into a function $R^{\mathcal{I}}: \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \to \{0, 1\}$
  - Individual name a into an element of $\Delta^{\mathcal{I}}$
- $\mathcal{ALC}$ mapping to FOL:

$$
\begin{array}{rcc|rcc}
\top(x) & \mapsto & 1 & \bot(x) & \mapsto & 0 \\
A(x) & \mapsto & A(x) & (C_1 \sqcap C_2)(x) & \mapsto & C_1(x) \wedge C_2(x) \\
(C_1 \sqcup C_2)(x) & \mapsto & C_1(x) \vee C_2(x) & (\neg C)(x) & \mapsto & \neg C(x) \\
(\exists R.C)(x) & \mapsto & \exists y.R(x,y) \wedge C(y) & (\forall R.C)(x) & \mapsto & \forall y.R(x,y) \Rightarrow C(y) \\
 & & & & & \\
C \sqsubseteq D & \mapsto & \forall x.C(x) \Rightarrow D(x) & a{:}C & \mapsto & C(a)
\end{array}
$$

Description Logic System

Inference System $\Leftrightarrow$ User Interface

$\Updownarrow$

Knowledge Base

TBox

"Defines terminology of the application domain"

$$Man = Human \sqcap Male$$

$$Happy\_Father = Man \sqcap \exists has\_child.Female$$

ABox

"States facts about a specific world"

$$John{:}Happy\_Father$$

$$(John, Mary){:}has\_child$$

Note on DL naming

$\mathcal{AL}$: $C, D \longrightarrow \top \mid \bot \mid A \mid C \sqcap D \mid \neg A \mid \exists R.\top \mid \forall R.C$

$\mathcal{C}$: Concept negation, $\neg C$. Thus, $\mathcal{ALC} = \mathcal{AL} + \mathcal{C}$

$\mathcal{S}$: Used for $\mathcal{ALC}$ with transitive roles $\mathcal{R}_+$

$\mathcal{U}$: Concept disjunction, $C_1 \sqcup C_2$

$\mathcal{E}$: Existential quantification, $\exists R.C$

$\mathcal{H}$: Role inclusion axioms, $R_1 \sqsubseteq R_2$, e.g. $is\_component\_of \sqsubseteq is\_part\_of$

$\mathcal{N}$: Number restrictions, $(\geq n\ R)$ and $(\leq n\ R)$, e.g. $(\geq 3\ has\_Child)$ (has at least 3 children)

$\mathcal{Q}$: Qualified number restrictions, $(\geq n\ R.C)$ and $(\leq n\ R.C)$, e.g. $(\leq 2\ has\_Child.Adult)$ (has at most 2 adult children)

$\mathcal{O}$: Nominals (singleton class), $\{a\}$, e.g. $\exists has\_child.\{mary\}$.
Note: $a{:}C$ equiv to $\{a\} \sqsubseteq C$ and $(a, b){:}R$ equiv to $\{a\} \sqsubseteq \exists R.\{b\}$

$\mathcal{I}$: Inverse role, R^-, e.g. $isPartOf = hasPart^-$

$\mathcal{F}$: Functional role, f, e.g. $functional(hasAge)$

$\mathcal{R}_+$: transitive role, e.g. $transitive(isPartOf)$

For instance,

$$\begin{array}{rcl} \mathcal{SHIF} & = & \mathcal{S}+\mathcal{H}+\mathcal{I}+\mathcal{F} = \mathcal{ALCR}_+\mathcal{HIF} \\ \mathcal{SHOIN} & = & \mathcal{S}+\mathcal{H}+\mathcal{O}+\mathcal{I}+\mathcal{N} = \mathcal{ALCR}_+\mathcal{HOIN} \end{array}$$

OWL-Lite (EXPTIME)
OWL-DL (NEXPTIME)

Excerpt of pizza ontology ... (according to University of Manchester)

$$
\begin{array}{lcl}
\mathit{PizzaFruttiDiMare} & = & \mathit{Pizza} \\
 & & \sqcap \exists \mathit{hasTopping}.\mathit{MixedSeafoodTopping} \\
 & & \sqcap\ \exists \mathit{hasTopping}.\mathit{GarlicTopping} \\
 & & \sqcap\ \exists \mathit{hasTopping}.\mathit{TomatoTopping} \\
 & & \sqcap\ \forall \mathit{hasTopping}.(\mathit{MixedSeafoodTopping} \sqcup \mathit{GarlicTopping} \sqcup \mathit{TomatoTopping}) \\
 & & \sqcap\ \exists \mathit{hasBase}.\mathit{PizzaBase} \\
\\
\mathit{PizzaBase} & & \mathit{DeepPanBase} \sqcup \mathit{ThinAndCrispyBase} \\
\\
\mathit{MixedSeafoodTopping} & \sqsubseteq & \mathit{FishTopping} \\
\\
\mathit{FishTopping} & \sqsubseteq & \mathit{PizzaTopping} \sqcap\ \exists \mathit{hasSpiceness}.\mathit{Mild}
\end{array}
$$

$\mathit{disjoint}(\mathit{FishTopping}, \mathit{MeatTopping}, \mathit{HerbSpiceTopping})$

$\mathit{functional}(\mathit{hasSpiciness})$

$$\mathit{Topping} \sqsubseteq \forall \mathit{hasSpiciness}.(\mathit{Hot} \sqcup \mathit{Medium} \sqcup \mathit{Mild})$$

Concrete domains

- Concrete domains: integers, strings, . . .
- Clean separation between "object" classes and concrete domains
  - $D = \langle \Delta_D, \Phi_D \rangle$
  - Δ_D is an interpretation domain
  - Φ_D is the set of concrete domain predicates d with a predefined arity n and fixed interpretation $d^D \colon \Delta_D^n \to \{0, 1\}$
  - Concrete properties: $R^{\mathcal{I}} \colon \Delta^{\mathcal{I}} \times \Delta_D \to \{0, 1\}$

$(tim, 14)$:*hasAge*
$(sf, \text{“}SoftComputing\text{”})$:*hasAcronym*
$(source1, \text{“}ComputerScience\text{”})$:*isAbout*
$(service2, \text{“}InformationRetrievalTool''})$:*Matches*

- Philosophical reasons: concrete domains structured by built-in predicates
- Practical reasons:
  - language remains simple and compact
  - Semantic integrity of language not compromised
  - Implementability not compromised – can use hybrid reasoner
    - ★ Only need sound and complete decision procedure for $d_1^{\mathcal{I}} \wedge \ldots \wedge d_n^{\mathcal{I}}$, where d_i is a (posssibly negated) concrete property
- Notation: (D). E.g., $\mathcal{ALC}(D)$ is $\mathcal{ALC}$ + concrete domains

OWL DL

Abstract Syntax	DL Syntax	Example
Descriptions (C)		
A (URI reference)	A	Conference
owl:Thing	$\top$	
owl:Nothing	$\bot$	
intersectionOf(C_1 C_2 ...)	$C_1 \sqcap C_2$	Reference $\sqcap$ Journal
unionOf(C_1 C_2 ...)	$C_1 \sqcup C_2$	Organization $\sqcup$ Institution
complementOf(C)	$\neg C$	$\neg$ MasterThesis
oneOf(o_1 ...)	$\{o_1, \ldots\}$	{"WISE","ISWC",...}
restriction(R someValuesFrom(C))	$\exists R.C$	$\exists$parts.InCollection
restriction(R allValuesFrom(C))	$\forall R.C$	$\forall$date.Date
restriction(R hasValue(o))	$R : o$	date : 2005
restriction(R minCardinality(n))	$(\geq n\, R)$	$\geqslant$ 1 location
restriction(R maxCardinality(n))	$(\leq n\, R)$	$\leqslant$ 1 publisher
restriction(U someValuesFrom(D))	$\exists U.D$	$\exists$issue.integer
restriction(U allValuesFrom(D))	$\forall U.D$	$\forall$name.string
restriction(U hasValue(v))	$U : v$	series : "LNCS"
restriction(U minCardinality(n))	$(\geq n\, U)$	$\geqslant$ 1 title
restriction(U maxCardinality(n))	$(\leq n\, U)$	$\leqslant$ 1 author

Abstract Syntax	DL Syntax	Example
Axioms		
`Class(`A `partial` $C_1 \ldots C_n$`)`	$A \sqsubseteq C_1 \sqcap \ldots \sqcap C_n$	$Human \sqsubseteq Animal \sqcap Biped$
`Class(`A `complete` $C_1 \ldots C_n$`)`	$A = C_1 \sqcap \ldots \sqcap C_n$	$Man = Human \sqcap Male$
`EnumeratedClass(`$A\ o_1 \ldots o_n$`)`	$A = \{o_1\} \sqcup \ldots \sqcup \{o_n\}$	$RGB = \{r\} \sqcup \{g\} \sqcup \{b\}$
`SubClassOf(`$C_1 C_2$`)`	$C_1 \sqsubseteq C_2$	
`EquivalentClasses(`$C_1 \ldots C_n$`)`	$C_1 = \ldots = C_n$	
`DisjointClasses(`$C_1 \ldots C_n$`)`	$C_i \sqcap C_j = \perp, i \neq j$	$Male \sqsubseteq \neg Female$
`ObjectProperty(`R `super` $(R_1) \ldots$ `super` (R_n)`)`	$R \sqsubseteq R_i$	$HasDaughter \sqsubseteq hasChild$
`domain(`C_1`)`...`domain(`C_n`)`	$(\geq 1\ R) \sqsubseteq C_i$	$(\geq 1\ hasChild) \sqsubseteq Human$
`range(`C_1`)`...`range(`C_n`)`	$\top \sqsubseteq \forall R.D_i$	$\top \sqsubseteq \forall hasChild.Human$
[`inverseof(`R_0`)`]	$R = R_0^-$	$hasChild = hasParent^-$
[`symmetric`]	$R = R^-$	$similar = similar^-$
[`functional`]	$\top \sqsubseteq (\leq 1\ R)$	$\top \sqsubseteq (\leq 1\ hasMother)$
[`Inversefunctional`]	$\top \sqsubseteq (\leq 1\ R^-)$	
[`Transitive`]	$Tr(R)$	$Tr(ancestor)$
`SubPropertyOf(`$R_1 R_2$`)`	$R_1 \sqsubseteq R_2$	
`EquivalentProperties(`$R_1 \ldots R_n$`)`	$R_1 = \ldots = R_n$	$cost = price$
`AnnotationProperty(`S`)`		

Abstract Syntax	DL Syntax	Example
`DatatypeProperty`(U `super` (U_1)... `super` (U_n))	$U \sqsubseteq U_i$	
`domain`(C_1)...`domain`(C_n)	$(\geq 1\ U) \sqsubseteq C_i$	$(\geq 1\ hasAge) \sqsubseteq Human$
`range`(D_1)...`range`(D_n)	$\top \sqsubseteq \forall U.D_i$	$\top \sqsubseteq \forall hasAge.posInteger$
`[functional]`	$\top \sqsubseteq (\leq 1\ U)$	$\top \sqsubseteq (\leq 1\ hasAge)$
`SubPropertyOf`($U_1 U_2$)	$U_1 \sqsubseteq U_2$	$hasName \sqsubseteq hasFirstName$
`EquivalentProperties`($U_1 \ldots U_n$)	$U_1 = \ldots = U_n$	
Individuals		
`Individual`(o `type` (C_1)... `type` (C_n))	$o{:}C_i$	$tim{:}Human$
`value`($R_1 o_1$)...`value`($R_n o_n$)	$(o, o_i){:}R_i$	$(tim, mary){:}hasChild$
`value`($U_1 v_1$)...`value`($U_n v_n$)	$(o, v_1){:}U_i$	$(tim, 14){:}hasAge$
`SameIndividual`($o_1 \ldots o_n$)	$o_1 = \ldots = o_n$	$president\_Bush = G.W.Bush$
`DifferentIndividuals`($o_1 \ldots o_n$)	$o_i \neq o_j, i \neq j$	$john \neq peter$

XML representation of OWL statements

E.g., $Person \sqcap \forall hasChild.(Doctor \sqcup \exists hasChild.Doctor)$:

```
<owl:Class>
  <owl:intersectionOf rdf:parseType=" collection">
    <owl:Class rdf:about="#Person"/>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasChild"/>
      <owl:allValuesFrom>
        <owl:unionOf rdf:parseType=" collection">
          <owl:Class rdf:about="#Doctor"/>
          <owl:Restriction>
            <owl:onProperty rdf:resource="#hasChild"/>
            <owl:someValuesFrom rdf:resource="#Doctor"/>
          </owl:Restriction>
        </owl:unionOf>
      </owl:allValuesFrom>
    </owl:Restriction>
  </owl:intersectionOf>
</owl:Class>
```

Fuzzy Description Logics

Objective

- To extend classical DLs and LPs towards the representation of and reasoning with vague concepts
- To show some applications
- Development of practical reasoning algorithms

A clarification: Uncertainty v.s. Imprecision

- Uncertainty theory: statements rather than being either true or false, are true or false to some probability or possibility/necessity
  - E.g., "It is possible that it will rain tomorrow"
  - Usually we have a possible world semantics with a distribution over possible worlds:

$$W = \{I \text{ classical interpretation}\}, \quad I(\varphi) \in \{0,1\}$$
$$\mu\colon W \to [0,1], \quad \mu(I) \in [0,1]$$
$$e.g.\ Pr(\phi) = \sum_{I \models \phi} \mu(I), \quad Poss(\phi) = \sup_{I \models \phi} \mu(I)$$

- Imprecision theory: statements are true to some degree which is taken from a truth space
  - E.g., "Chinese items are cheap"
  - Truth space: set of truth values L and an partial order $\leq$
  - Many-valued Interpretation: a function I mapping formuale into L, i.e. $I(\varphi) \in L$
  - Fuzzy Logic: $L = [0,1]$
- Uncertainty and imprecision theory: "It is possible that it will be hot tomorrow"
- In this work we deal with imprecision and, thus, statements have a degree of truth.

Examples of applications (Ontology mediated data access)

Example (Top-k retrieval)

$$\begin{array}{rcl} Hotel & \sqsubseteq & \exists hasLoc \\ Conference & \sqsubseteq & \exists hasLoc \\ Hotel & \sqsubseteq & \neg Conference \end{array}$$

HotelID	*hasLoc*
$h1$	$hl1$
$h2$	$hl2$
⋮	⋮

ConferenceID	*hasLoc*
$c1$	$cl1$
$c2$	$cl2$
⋮	⋮

hasLoc	*hasLoc*	*distance*
$hl1$	$cl1$	300
$hl1$	$cl2$	500
$hl2$	$cl1$	750
$hl2$	$cl2$	800
⋮	⋮	

hasLoc	*hasLoc*	*close*
$hl1$	$cl1$	0.7
$hl1$	$cl2$	0.5
$hl2$	$cl1$	0.25
$hl2$	$cl2$	0.2
⋮	⋮	

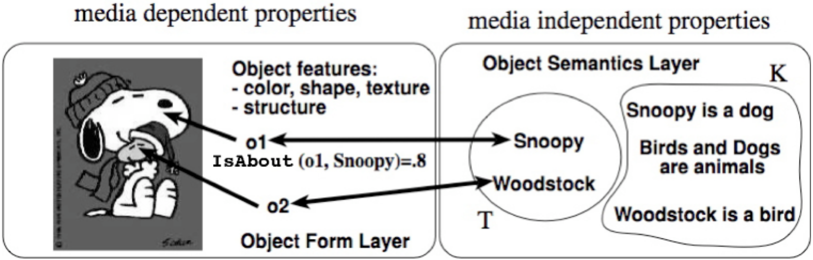
"Find hotels close to the university of Bari"

$$q(h) \leftarrow hasLocation(h, hl) \wedge hasLocation(uniba, cl) \wedge close(hl, cl)$$

Top-k Fuzzy Retrieval: Retrieve the top-k ranked tuples that instantiate the query q w.r.t. the best truth value bound

Note: retrieving all tuples, ranking them and then selecting the top-k ones is not feasible in practice (millions of tuples in the database)

Example (Logic-based information retrieval model, Top-k retrieval)



$$\begin{array}{rcl} \mathit{Bird} & \sqsubseteq & \mathit{Animal} \\ \mathit{Dog} & \sqsubseteq & \mathit{Animal} \\ \mathit{snoopy} & : & \mathit{Dog} \\ \mathit{woodstock} & : & \mathit{Bird} \end{array}$$

ImageRegion	*Object ID*	*isAbout*
*o*1	*snoopy*	0.8
*o*2	*woodstock*	0.7
⋮	⋮	

"Find image regions about animals"

$$\mathit{Query}(\mathit{ir}) \leftarrow \mathit{ImageRegion}(\mathit{ir}) \land \mathit{isAbout}(\mathit{ir}, x) \land \mathit{Animal}(x)$$

Example (Graded Entailment)

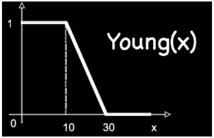
		
audi_tt	*mg*	*ferrari_enzo*

Car	*speed*
audi_tt	243
mg	≤ 170
ferrari_enzo	≥ 350

$$\mathit{SportsCar} \quad = \quad \mathit{Car} \sqcap \exists \mathit{hasSpeed}.\mathit{very}(\mathit{High})$$

$$\begin{array}{lll} \mathcal{K} & \models & \langle \mathit{ferrari\_enzo}{:}\mathit{SportsCar}, 1\rangle \\ \mathcal{K} & \models & \langle \mathit{audi\_tt}{:}\mathit{SportsCar}, 0.92\rangle \\ \mathcal{K} & \models & \langle \mathit{audi\_tt}{:}\neg\mathit{SportsCar}, 0.72\rangle \end{array}$$

Example (Graded Subsumption)

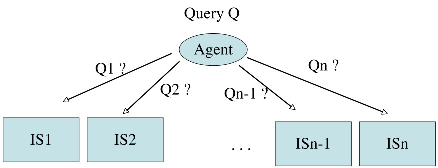


$$\begin{array}{rcl} \mathit{Minor} & = & \mathit{Person} \sqcap \exists \mathit{hasAge}. \leq_{18} \\ \mathit{YoungPerson} & = & \mathit{Person} \sqcap \exists \mathit{hasAge}. \mathit{Young} \end{array}$$

$$\mathcal{K} \models \langle \mathit{Minor} \sqsubseteq \mathit{YoungPerson}, 0.2 \rangle$$

Note: without an explicit membership function of *Young*, this inference cannot be drawn

Example (Distributed Information Retrieval)



Then the agent has to perform automatically the following steps:

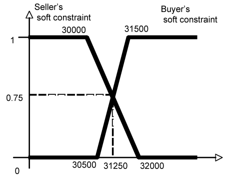
1. the agent has to select a subset of relevant resources $\mathscr{S}' \subseteq \mathscr{S}$, as it is not reasonable to assume to access to and query all resources (resource selection/resource discovery);
2. for every selected source $\mathcal{S}_i \in \mathscr{S}'$ the agent has to reformulate its information need Q_A into the query language $\mathcal{L}_i$ provided by the resource (schema mapping/ontology alignment);
3. the results from the selected resources have to be merged together (data fusion/rank aggregation)

- Resource selection/resource discovery:
  - Use techniques from Distributed Information Retrieval, e.g. CORI
- Schema mapping/ontology alignment:
  - Use machine learning techniques, (implemented in oMap)
    - Learns automatically weighted rules, like (aligning Google- Yahoo directories)

$$Mechanical\_and\_Aerospace\_Engineering(d) \longleftarrow 0.81 \cdot Aeronautics\_and\_Astronautics(d)$$

- Data fusion/rank aggregation:
  - Use techniques from Information Retrieval and/or Voting Systems, e.g. CombMNZ or Borda count

Example (Negotiation)



- a car seller sells an Audi TT for \$31500, as from the catalog price.
- a buyer is looking for a sports-car, but wants to to pay not more than around \$30000
- classical DLs: the problem relies on the crisp conditions on price
- more fine grained approach: to consider prices as fuzzy sets (as usual in negotiation)
  - seller may consider optimal to sell above \$31500, but can go down to \$30500
  - the buyer prefers to spend less than \$30000, but can go up to \$32000

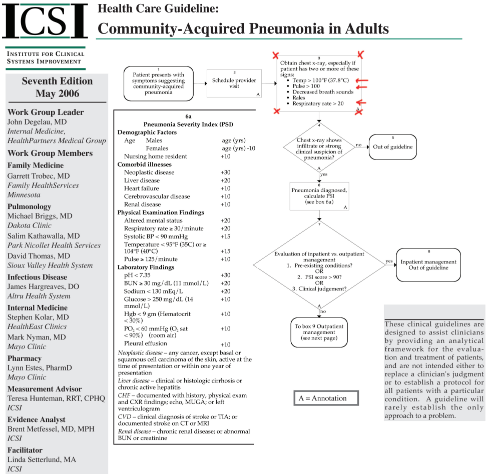
$$\begin{array}{lcl} \mathit{AudiTT} & = & \mathit{SportsCar} \sqcap \exists \mathit{hasPrice}.R(x; 30500, 31500) \\ \mathit{Query} & = & \mathit{SportsCar} \sqcap \exists \mathit{hasPrice}.L(x; 30000, 32000) \end{array}$$

  - highest degree to which the concept

$$C = \mathit{AudiTT} \sqcap \mathit{Query}$$

    is satisfiable is 0.75 (the possibility that the Audi TT and the query matches is 0.75)
  - the car may be sold at \$31250

Example (Health-care: diagnosis of pneumonia)



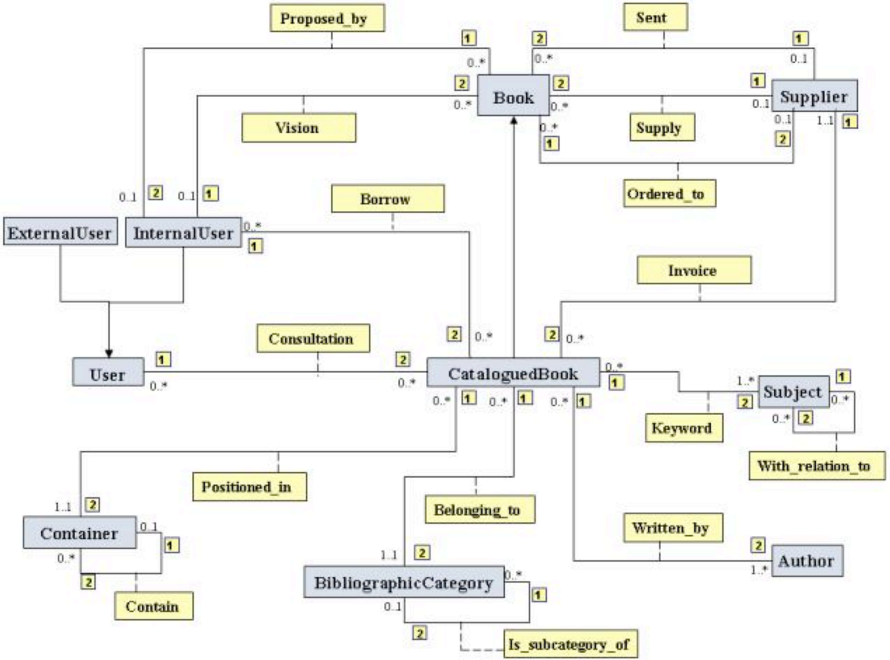
- E.g., *Temp* $= 37.5$, *Pulse* $= 98$, *RespiratoryRate* $= 18$ are in the "danger zone" already
- Temperature, Pulse and Respiratory rate, . . . : these constraints are rather fuzzy than crisp

$$
\begin{array}{lcl}
\mathit{CriticalTempPatient} & = & \mathit{Patient} \sqcap \exists \mathit{hasTemp}.R(x; 37.5, 37.8) \\
\mathit{CriticalPulsePatient} & = & \mathit{Patient} \sqcap \exists \mathit{hasPulse}.R(x; 95, 100)
\end{array}
$$

Top-*k* retrieval in DLs: the case of DL-Lite

- DL-Lite: a simple, but interesting DL
- Captures important subset of UML/ER diagrams
- Computationally tractable DL to query large databases
- Sub-linear, i.e. LOGSpace in data complexity
  - (same cost as for SQL)
- Good for very large database tables, with limited declarative schema design

- Knowledge base: $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$, where $\mathcal{T}$ and $\mathcal{A}$ are finite sets of axioms and assertions
- Axiom: $Cl \sqsubseteq Cr$ (inclusion axiom)
  $fun(R)$ (functionality axiom)
- Note for inclusion axioms: the language for left hand side is different from the one for right hand side
- DL-Lite$_{core}$:
  - Concepts:
    $$\begin{array}{rcl} Cl & \rightarrow & A \mid \exists R \\ Cr & \rightarrow & A \mid \exists R \mid \neg A \mid \neg \exists R \\ R & \rightarrow & P \mid P^{-} \end{array}$$
  - Assertion: $a{:}A$, $(a, b){:}P$
- DLR-Lite$_{core}$: (n-ary roles)
  - Concepts:
    $$\begin{array}{rcl} Cl & \rightarrow & A \mid \exists P[i] \\ Cr & \rightarrow & A \mid \exists P[i] \mid \neg A \mid \neg \exists P[i] \end{array}$$
  - $\exists P[i]$ is the projection on i-th column
  - Assertion: $a{:}A$, $\langle a_1, \ldots, a_n \rangle{:}P$
- Assertions are stored in relational tables
- Conjunctive query: $q(\mathbf{x}) \leftarrow \exists \mathbf{y}.conj(\mathbf{x}, \mathbf{y})$
  $conj$ is an aggregation of expressions of the form $B(z)$ or $P(z_1, z_2)$,


Proposed_by
Sent
Book
Supplier
Vision
Supply
Ordered_to
Borrow
ExternalUser
InternalUser
Invoice
Consultation
User
CatalogedBook
Subject
Keyword
With_relation_to
Positioned_in
Belonging_to
Written_by
Container
Author
BibliographicCategory
Contain
Is_subcategory_of

- Examples:

isa	$CatalogueBook \sqsubseteq Book$
disjointness	$Book \sqsubseteq \neg Author$
constraints	$CatalogueBook \sqsubseteq \exists positioned\_In$
role − typing	$\exists positioned\_In \sqsubseteq Container$
functional	$fun(positioned\_In)$
constraints	$Author \sqsubseteq \exists written\_By^-$
	$\exists written\_By \sqsubseteq CatalogueBook$
assertion	$Romeo\_and\_Juliet{:}CatalogueBook$
	$(Romeo\_and\_Juliet, Shakespeare){:}written\_By$
query	$q(x, y) \leftarrow CataloguedBook(x), Ordered\_to(x, y)$

- Consistency check is linear time in the size of the KB
- Query answering in linear in in the size of the number of assertions

Top-k retrieval in DL-Lite

- We extend the query formalism:
  - conjunctive queries, where fuzzy predicates may appear
- conjunctive query

$$q(\mathbf{x}, s) \leftarrow \exists \mathbf{y}.conj(\mathbf{x}, \mathbf{y}), s = f(p_1(\mathbf{z}_1), \ldots, p_n(\mathbf{z}_n))$$

1. $\mathbf{x}$ are the *distinguished variables*;
2. s is the *score variable*, taking values in $[0, 1]$;
3. $\mathbf{y}$ are existentially quantified variables, called *non-distinguished variables*;
4. $conj(\mathbf{x}, \mathbf{y})$ is a conjunction of atoms of the form $A(z)$, or $P(z, z')$, where A and P are respectively an atomic concept and a role (but, not inverse role) in $\mathcal{K}$;
5. z, z' are constants in $\mathcal{K}$ or variables in $\mathbf{x}$ or $\mathbf{y}$;
6. $\mathbf{z_i}$ are tuples of constants in $\mathcal{K}$ or variables in $\mathbf{x}$ or $\mathbf{y}$;
7. p_i is an n_i-ary *fuzzy predicate* assigning to each n_i-ary tuple $\mathbf{c}_i$ the *score* $p_i(\mathbf{c}_i) \in [0, 1]$;
8. f is a monotone *scoring* function $f\colon [0, 1]^n \to [0, 1]$, which combines the scores of the n fuzzy predicates $p_i(\mathbf{c}_i)$

Example:

$$\begin{array}{rcl} Hotel & \sqsubseteq & \exists HasHLoc \\ Hotel & \sqsubseteq & \exists HasHPrice \\ Conference & \sqsubseteq & \exists HasCLoc \\ Hotel & \sqsubseteq & \neg Conference \end{array}$$

HasHLoc	
HotelID	*HasLoc*
*h*1	*hl*1
*h*2	*hl*2
⋮	⋮

HasCLoc	
ConfID	*HasLoc*
*c*1	*cl*1
*c*2	*cl*2
⋮	⋮

HasHPrice	
HotelID	*Price*
*h*1	150
*h*2	200
⋮	⋮

$$\begin{array}{l} q(h, s) \leftarrow HasHLoc(h, hl), HasHPrice(h, p), \\ \qquad HasCLoc(c1, cl), s = cheap(p) \cdot close(hl, cl)\ . \end{array}$$

where the fuzzy predicates *cheap* and *close* are defined as

$$\begin{array}{rcl} close(hl, cl) & = & \max(0, 1 - \frac{distance(hl, cl)}{2000}) \\ cheap(price) & = & \max(0, 1 - \frac{price}{300}) \end{array}$$

Semantics informally:

- a conjunctive query

$$q(\mathbf{x}, s) \leftarrow \exists \mathbf{y}.conj(\mathbf{x}, \mathbf{y}), s = f(p_1(\mathbf{z}_1), \ldots, p_n(\mathbf{z}_n))$$

  is interpreted in an interpretation $\mathcal{I}$ as the set

$$q^{\mathcal{I}} = \{\langle \mathbf{c}, v\rangle \in \Delta \times \ldots \times \Delta \times [0, 1] \mid \ldots$$

  such that when we consider the substitution

$$\theta = \{\mathbf{x}/\mathbf{c}, s/v\}$$

  the formula

$$\exists \mathbf{y}.conj(\mathbf{x}, \mathbf{y}) \wedge s = f(p_1(\mathbf{z}_1), \ldots, p_n(\mathbf{z}_n))$$

  evaluates to true in $\mathcal{I}$.
- Model of a query: $\mathcal{I} \models q(\mathbf{c}, v)$ iff $\langle \mathbf{c}, v\rangle \in q^{\mathcal{I}}$
- Entailment: $\mathcal{K} \models q(\mathbf{c}, v)$ iff $\mathcal{I} \models \mathcal{K}$ implies $\mathcal{I} \models q(\mathbf{c}, v)$
- Top-k retrieval: $ans_{top-k}(\mathcal{K}, q) = Top_k\{\langle \mathbf{c}, v\rangle \mid \mathcal{K} \models q(\mathbf{c}, v)\}$

How to determine the top-k answers of a query?

- Overall strategy: three steps
  1. Check if $\mathcal{K}$ is satisfiable, as querying a non-satisfiable KB is meaningless (checkable in linear time)
  2. Query q is *reformulated* into a set of conjunctive queries $r(q, \mathcal{T})$
     - ★ Basic idea: reformulation procedure closely resembles a top-down resolution procedure for logic programming

$$
\begin{array}{rcl}
q(x,s) & \leftarrow & B(x), A(x), s = f(x) \\
B_1 & \sqsubseteq & A \\
B_2 & \sqsubseteq & A \\
\hline\hline
q(x,s) & \leftarrow & B(x), B_1(x), s = f(x) \\
q(x,s) & \leftarrow & B(x), B_2(x), s = f(x)
\end{array}
$$

  3. The reformulated queries in $r(q, \mathcal{T})$ are evaluated over $\mathcal{A}$ (seen as a database) using standard top-k techniques for DBs
     - ★ for all $q_i \in r(q, \mathcal{T})$, $ans_{top-k}(q_i, \mathcal{A}) =$ top-k SQL query over $\mathcal{A}$ database
     - ★ $ans_{top-k}(KB, q) = Top_k(\bigcup_{q_i \in r(q,\mathcal{T})} ans_k(q_i, \mathcal{A}))$

Small Example:

P_2	
0	s
3	t
4	q
6	q

B
1
2
5
7

$$\mathcal{T} = \{\exists P_2^- \sqsubseteq A, A \sqsubseteq \exists P_1, B \sqsubseteq \exists P_2\}$$

$$q(x, s) \leftarrow P_2(x, y), P_1(y, z), s = \max(0, 1 - x/10)$$

Small Example:

P_2	
0	s
3	t
4	q
6	q

B
1
2
5
7

$$\mathcal{T} = \{\exists P_2^- \sqsubseteq A, A \sqsubseteq \exists P_1, B \sqsubseteq \exists P_2\}$$

$$q(x,s) \leftarrow P_2(x,y), P_1(y,z), s = \max(0, 1 - x/10)$$

Small Example:

P_2	
0	s
3	t
4	q
6	q

B
1
2
5
7

$$\mathcal{T} = \{\exists P_2^- \sqsubseteq A, A \sqsubseteq \exists P_1, B \sqsubseteq \exists P_2\}$$

$$q(x, s) \leftarrow P_2(x, y), P_1(y, z), s = \max(0, 1 - x/10)$$

Small Example:

P_2	
0	s
3	t
4	q
6	q

B
1
2
5
7

$$\mathcal{T} = \{\exists P_2^- \sqsubseteq A, A \sqsubseteq \exists P_1, B \sqsubseteq \exists P_2\}$$

$$\frac{q(x,s) \leftarrow P_2(x,y), P_1(y,z), s = \max(0, 1 - x/10)}{q(x,s) \leftarrow P_2(x,y), A(y), s = \max(0, 1 - x/10)}$$

Small Example:

P_2	
0	s
3	t
4	q
6	q

B
1
2
5
7

$$\mathcal{T} = \{\exists P_2^- \sqsubseteq A, A \sqsubseteq \exists P_1, B \sqsubseteq \exists P_2\}$$

$$\frac{q(x,s) \leftarrow P_2(x,y), P_1(y,z), s = \max(0, 1 - x/10)}{\begin{array}{l} q(x,s) \leftarrow P_2(x,y), A(y), s = \max(0, 1 - x/10) \\ q(x,s) \leftarrow P_2(x,y), P_2(z,y), s = \max(0, 1 - x/10) \end{array}}$$

Small Example:

P_2	
0	s
3	t
4	q
6	q

B
1
2
5
7

$$\mathcal{T} = \{\exists P_2^- \sqsubseteq A, A \sqsubseteq \exists P_1, B \sqsubseteq \exists P_2\}$$

$$\frac{q(x,s) \leftarrow P_2(x,y), P_1(y,z), s = \max(0, 1 - x/10)}{q(x,s) \leftarrow P_2(x,y), A(y), s = \max(0, 1 - x/10)}$$

$$q(x,s) \leftarrow \color{red}{P_2(x,y)}, \color{red}{P_2(z,y)}, s = \max(0, 1 - x/10)$$

$$q(x,s) \leftarrow \color{red}{P_2(x,y)}, s = \max(0, 1 - x/10)$$

Small Example:

P_2	
0	s
3	t
4	q
6	q

B
1
2
5
7

$$\mathcal{T} = \{\exists P_2^- \sqsubseteq A, A \sqsubseteq \exists P_1, B \sqsubseteq \exists P_2\}$$

$$q(x,s) \leftarrow P_2(x,y), P_1(y,z), s = \max(0, 1 - x/10)$$

$$q(x,s) \leftarrow P_2(x,y), A(y), s = \max(0, 1 - x/10)$$

$$q(x,s) \leftarrow P_2(x,y), P_2(z,y), s = \max(0, 1 - x/10)$$

$$q(x,s) \leftarrow P_2(x,y), s = \max(0, 1 - x/10)$$

$$q(x,s) \leftarrow B(x), s = \max(0, 1 - x/10)$$

Small Example:

P_2	
0	s
3	t
4	q
6	q

B
1
2
5
7

$$\mathcal{T} = \{\exists P_2^- \sqsubseteq A, A \sqsubseteq \exists P_1, B \sqsubseteq \exists P_2\}$$

$$q(x, s) \leftarrow P_2(x, y), P_1(y, z), s = \max(0, 1 - x/10)$$

$$q(x, s) \leftarrow P_2(x, y), A(y), s = \max(0, 1 - x/10)$$
$$q(x, s) \leftarrow P_2(x, y), P_2(z, y), s = \max(0, 1 - x/10)$$
$$q(x, s) \leftarrow P_2(x, y), s = \max(0, 1 - x/10)$$
$$q(x, s) \leftarrow B(x), s = \max(0, 1 - x/10)$$

$$q_1(x, s) \leftarrow P_2(x, y), s = \max(0, 1 - x/10)$$
$$q_2(x, s) \leftarrow B(x), s = \max(0, 1 - x/10)$$

Small Example:

P_2	
0	s
3	t
4	q
6	q

B
1
2
5
7

$$\mathcal{T} = \{\exists P_2^- \sqsubseteq A, A \sqsubseteq \exists P_1, B \sqsubseteq \exists P_2\}$$

$$q(x, s) \leftarrow P_2(x, y), P_1(y, z), s = \max(0, 1 - x/10)$$

$$q(x, s) \leftarrow P_2(x, y), A(y), s = \max(0, 1 - x/10)$$
$$q(x, s) \leftarrow P_2(x, y), P_2(z, y), s = \max(0, 1 - x/10)$$
$$q(x, s) \leftarrow P_2(x, y), s = \max(0, 1 - x/10)$$
$$q(x, s) \leftarrow B(x), s = \max(0, 1 - x/10)$$

$$q_1(x, s) \leftarrow P_2(x, y), s = \max(0, 1 - x/10)$$
$$q_2(x, s) \leftarrow B(x), s = \max(0, 1 - x/10)$$

$$ans_{top-3}(\mathcal{A}, q_1) = [\langle 0, 1.0\rangle, \langle 3, 0.7\rangle, \langle 4, 0.6\rangle]$$
$$ans_{top-3}(\mathcal{A}, q_2) = [\langle 1, 0.9\rangle, \langle 2, 0.8\rangle, \langle 5, 0.5\rangle]$$

Small Example:

P_2	
0	s
3	t
4	q
6	q

B
1
2
5
7

$$\mathcal{T} = \{\exists P_2^- \sqsubseteq A, A \sqsubseteq \exists P_1, B \sqsubseteq \exists P_2\}$$

$$q(x,s) \leftarrow P_2(x,y), P_1(y,z), s = \max(0, 1 - x/10)$$

$$q(x,s) \leftarrow P_2(x,y), A(y), s = \max(0, 1 - x/10)$$
$$q(x,s) \leftarrow P_2(x,y), P_2(z,y), s = \max(0, 1 - x/10)$$
$$q(x,s) \leftarrow P_2(x,y), s = \max(0, 1 - x/10)$$
$$q(x,s) \leftarrow B(x), s = \max(0, 1 - x/10)$$

$$q_1(x,s) \leftarrow P_2(x,y), s = \max(0, 1 - x/10)$$
$$q_2(x,s) \leftarrow B(x), s = \max(0, 1 - x/10)$$

$$ans_{top-3}(\mathcal{A}, q_1) = [\langle 0, 1.0\rangle, \langle 3, 0.7\rangle, \langle 4, 0.6\rangle]$$
$$ans_{top-3}(\mathcal{A}, q_2) = [\langle 1, 0.9\rangle, \langle 2, 0.8\rangle, \langle 5, 0.5\rangle]$$

$$ans_{top-k}(\mathcal{K}, q) = [\langle 0, 1.0\rangle, \langle 1, 0.9\rangle, \langle 2, 0.8\rangle]$$

Proposition

Given a DL-Lite KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A}\rangle$ and a query q then we can compute $ans_{top-k}(\mathcal{K}, q)$ in (sub) linear time w.r.t. the size of $\mathcal{A}$. The same holds for the description logic DLR-Lite.

Propositional Fuzzy Logics Basics

- Formulae: propositional formulae
- Truth space is $[0, 1]$
- Formulae have a a degree of truth in $[0, 1]$
- Interpretation: is a mapping $\mathcal{I} : \mathit{Atoms} \to [0, 1]$
- Interpretations are extended to formulae using norms to interpret connectives

negation
$n(0) = 1$
$a \leq b$ implies $n(b) \leq n(a)$

t-norm (conjunction)
$t(a, 1) = a$
$b \leq c$ implies $t(a, b) \leq t(a, c)$
$t(a, b) = t(b, a)$
$t(a, t(b, c)) = t(t(a, b), c)$

s-norm (disjunction)
$s(a, 0) = a$
$b \leq c$ implies $s(a, b) \leq s(a, c)$
$s(a, b) = s(b, a)$
$s(a, s(b, c)) = s(s(a, b), c)$

i-norm (implication)
$a \leq b$ implies $i(a, c) \geq i(b, c)$
$b \leq c$ implies $i(a, b) \leq i(a, c)$
$i(0, b) = 1$
$i(a, 1) = 1$

Usually,
$i(a, b) = \sup\{c \colon t(a, c) \leq b\}$

$i(a, b) = \sup\{c \colon t(a, c) \leq b\}$ is called r-implication and depends on the t-norm only

Typical norms

	Lukasiewicz Logic	Gödel Logic	Product Logic	Zadeh
$\neg x$	$1 - x$	if $x = 0$ then 1 else 0	if $x = 0$ then 1 else 0	$1 - x$
$x \wedge y$	$\max(x + y - 1, 0)$	$\min(x, y)$	$x \cdot y$	$\min(x, y)$
$x \vee y$	$\min(x + y, 1)$	$\max(x, y)$	$x + y - x \cdot y$	$\max(x, y)$
$x \Rightarrow y$	if $x \leq y$ then 1 else $1 - x + y$	if $x \leq y$ then 1 else y	if $x \leq y$ then 1 else y/x	$\max(1 - x, y)$

Note: for Lukasiewicz Logic and Zadeh, $x \Rightarrow y \equiv \neg x \vee y$

$$
\begin{array}{rcl}
\mathcal{I}(\phi \wedge \psi) & = & \mathcal{I}(\phi) \wedge \mathcal{I}(\psi) \\
\mathcal{I}(\phi \vee \psi) & = & \mathcal{I}(\phi) \vee \mathcal{I}(\psi)
\end{array}
$$

$$
\begin{array}{rcl}
\mathcal{I}(\phi \rightarrow \psi) & = & \mathcal{I}(\phi) \rightarrow \mathcal{I}(\psi) \\
\mathcal{I} \models \phi & \text{iff} & \mathcal{I}(\phi) = 1 \quad \text{iff } \phi \text{ satisfiable} \\
\mathcal{I} \models \mathcal{T} & \text{iff} & \mathcal{I} \models \phi \text{ for all } \phi \in \mathcal{T} \\
\models \phi & \text{iff} & \text{for all } \mathcal{I} . \mathcal{I} \models \phi \\
\mathcal{T} \models \phi & \text{iff} & \text{for all } \mathcal{I}. \text{ if } \mathcal{I} \models \mathcal{T} \text{ then } \mathcal{I} \models \phi
\end{array}
$$

- Note:

$$
\begin{array}{rcl}
\neg\phi & \text{is} & \phi \to 0 \\
\phi \bar{\wedge} \psi & \text{is} & \phi \wedge (\phi \to \psi) \\
\phi \bar{\vee} \psi & \text{is} & ((\phi \to \psi) \to \psi) \bar{\wedge} ((\psi \to \phi) \to \phi) \\
\mathcal{I}(\phi \bar{\wedge} \psi) & = & \min(\mathcal{I}(\phi), \mathcal{I}(\psi)) \\
\mathcal{I}(\phi \bar{\vee} \psi) & = & \max(\mathcal{I}(\phi), \mathcal{I}(\psi))
\end{array}
$$

- Zadeh semantics: not interesting for fuzzy logicians: its a sub-logic of Łukasiewicz

$$
\begin{array}{rcl}
\neg_Z \phi & = & \neg_Ł \phi \\
\phi \wedge_Z \psi & = & \phi \wedge_Ł (\phi \to_Ł \psi) \\
\phi \to \psi & = & \neg_Ł \phi \vee_Ł \psi
\end{array}
$$

- Hence, rarely considered by fuzzy logicians

Axioms of logic BL (Basic Fuzzy Logic)

Fix arbitray t-norm and r-implication.

(A1)	$(\phi \to \psi) \to ((\psi \to \chi) \to \phi \to \chi)$
(A2)	$(\phi \wedge \psi) \to \phi$
(A3)	$(\phi \wedge \psi) \to (\psi \wedge \phi)$
(A4)	$(\phi \wedge (\phi \to \psi)) \to (\psi \wedge (\psi \to \phi))$
(A5a)	$(\phi \wedge (\psi \to \chi)) \to ((\psi \wedge \psi) \to \chi))$
(A5b)	$((\psi \wedge \psi) \to \chi)) \to (\phi \wedge (\psi \to \chi))$
(A6)	$(\phi \wedge (\psi \to \chi)) \to (((\psi \to \phi) \to \chi)) \to \chi)$
(A7)	$0 \to \phi$
(Deduction rule)	Modus ponens: from ϕ and $\phi \to \psi$ infer ψ

Proposition

$\mathcal{T} \vdash_{BL} \phi$ iff $\mathcal{T} \models_{BL} \phi$. Also, if $\mathcal{T} \vdash_{BL} \phi$ then $\mathcal{T} \models_{BL2} \phi$, but not vice-versa (e.g. $\models_{BL2} \phi \vee \neg\phi$, but $\not\models_{BL} \phi \vee \neg\phi$).

- $\models_{BL} \phi \wedge \neg\phi \to 0$
- $\models_{BL} \phi \to \neg\neg\phi$, but $\not\models_{BL} \neg\neg\phi \to \phi$, e.g. $\phi = p \vee \neg p$, t-norm is Gödel
- $\models_{BL} (\phi \to \psi) \to (\neg\psi \to \neg\phi)$, but not vice-versa

Axioms of Łukasiewicz logic Ł

Fix Łukasiewicz t-norm and r-implication.

(Axioms) Axioms of BL

(Ł) $\neg\neg\phi \rightarrow \phi$

(Deduction rule) Modus ponens: from ϕ and $\phi \rightarrow \psi$ infer ψ

Proposition

$\mathcal{T} \vdash_{Ł} \phi$ *iff* $\mathcal{T} \models_{Ł} \phi$.

- $\models_{Ł} \phi \rightarrow \psi \equiv \neg\psi \rightarrow \neg\phi$
- $\models_{Ł} \neg(\phi \wedge \psi) \equiv \neg\phi \vee \neg\psi$
- $\models_{Ł} \phi \rightarrow \psi \equiv \neg(\phi \wedge \neg\psi)$
- $\models_{Ł} \phi \rightarrow \psi \equiv \neg\phi \vee \neg\psi$
- $\models_{Ł} \neg(\phi \rightarrow \psi) \equiv \phi \wedge \neg\psi$
- Recall that “Zadeh logic” is a sub-logic of Ł

Axioms of Product logic Π

Fix product t-norm and r-implication.

(Axioms) Axioms of BL

(Π1) $\neg\neg\chi \rightarrow ((\phi \wedge \chi \rightarrow \psi \wedge \chi) \rightarrow (\phi \rightarrow \psi))$

(Π2) $(\phi \bar{\wedge} \neg\phi) \rightarrow 0$

(Deduction rule) Modus ponens: from ϕ and $\phi \rightarrow \psi$ infer ψ

Proposition

$\mathcal{T} \vdash_\Pi \phi$ *iff* $\mathcal{T} \models_\Pi \phi$.

- $\models_\Pi \neg(\phi \wedge \psi) \rightarrow \neg(\phi \bar{\wedge} \psi)$
- $\models_\Pi (\phi \rightarrow \neg\phi) \rightarrow \neg\phi$
- $\models_\Pi \neg\phi \bar{\vee} \neg\neg\phi$

Axioms of Gödel logic G

Fix Gödel t-norm and r-implication.

(Axioms) Axioms of BL

(G) $\phi \to (\phi \wedge \phi)$

(Deduction rule) Modus ponens: from ϕ and $\phi \to \psi$ infer ψ

Proposition

$\mathcal{T} \vdash_G \phi$ iff $\mathcal{T} \models_G \phi$.

- $\models_G (\phi \wedge \psi) \equiv (\phi \bar{\wedge} \psi)$
- Gödel logic proves all axioms of intuitionistic logic I, vice-versa I + (A6) proves all axioms of Gödel logic

Axioms of Boolean logic

Fix interpretations to be boolean.

(Axioms) Axioms of BL

(BL2) $\phi \bar{\vee} \neg\phi$

(Deduction rule) Modus ponens: from ϕ and $\phi \to \psi$ infer ψ

Proposition

$\mathcal{T} \vdash_{BL2} \phi$ *iff* $\mathcal{T} \models_{BL2} \phi$.

- $\models_{BL2} \phi \to (\phi \wedge \phi)$ (BL2 extends G)
- Ł + G is equivalent to BL2
- Ł + Π is equivalent to BL2
- G + Π is equivalent to BL2

Axioms of Rational Pavelka Logic (RPL)

- Fix Łukasiewicz t-norm and r-implication
- Rational $r \in [0,1]$ may appear as atom in formula. $\mathcal{I}(r) = r$
- Note: $\mathcal{I}(r \to \phi) = 1$ iff $\mathcal{I}(\phi) \geq r$. Also, $\mathcal{I}(\phi \to r) = 1$ iff $\mathcal{I}(\phi) \leq r$

(Axioms) Axioms of Ł

(Deduction rule) Modus ponens: from ϕ and $\phi \to \psi$ infer ψ

Proposition

$\mathcal{T} \vdash_{RPL} \phi$ iff $\mathcal{T} \models_{RPL} \phi$.

- RPL proves the derived deduction rule:
  from $r \to \phi$ and $s \to (\phi \to \psi)$ infer $(r \wedge s) \to \psi$
- Let

$$\begin{aligned} ||\phi||_{\mathcal{T}} &= \inf\{\mathcal{I}(\phi) \mid \mathcal{I} \models \mathcal{T}\} \text{ (truth degree)} \\ |\phi|_{\mathcal{T}} &= \sup\{r \mid \mathcal{T} \vdash r \to \phi\} \text{ (provability degree)} \end{aligned}$$

  then $||\phi||_{\mathcal{T}} = |\phi|_{\mathcal{T}}$
- Also,

$$\begin{aligned} |\neg\phi|_{\mathcal{T}} &= 1 - |\phi|_{\mathcal{T}}| \\ |\phi|_{\mathcal{T}}| = \sup\{r \mid \mathcal{T} \vdash r \to \phi\} &= \inf\{s \mid \mathcal{T} \vdash \phi \to s\} \end{aligned}$$

Tableaux for Rational Pavelka Logic using MILP

Proposition

$|\phi|_{\mathcal{T}} = \min x$. *such that* $\mathcal{T} \cup \{\phi \rightarrow x\}$ *satisfiable.*

- We use MILP (Mixed Integer Linear Programming) to compute $|\phi|_{\mathcal{T}}$
- Let r be rational, variable or expresson $1 - r'$ (r' variable), both admitting solution in $[0, 1]$, $\neg r = 1 - r$, $\neg\neg r = r$

$r \rightarrow p$	$\mapsto$	$x_p \geq r, x_p \in [0,1]$
$p \rightarrow r$	$\mapsto$	$x_p \leq r, x_p \in [0,1]$
$r \rightarrow \neg\phi$	$\mapsto$	$\phi \rightarrow \neg r$
$\neg\phi \rightarrow r$	$\mapsto$	$\neg r \rightarrow \phi$
$r \rightarrow (\phi \wedge \psi)$	$\mapsto$	$x_1 \rightarrow \phi, x_2 \rightarrow \psi, y \leq 1 - r, x_i \leq 1 - y, x_1 + x_2 = r + 1 - y, x_i \in [0,1], y \in \{0,1\}$
$(\phi \wedge \psi) \rightarrow r$	$\mapsto$	$x_1 \rightarrow \neg\phi, x_2 \rightarrow \neg\psi, x_1 + x_2 = 1 - r, x_i \in [0,1]$
$r \rightarrow (\phi \rightarrow \psi)$	$\mapsto$	$\phi \rightarrow x_1, x_2 \rightarrow \psi, r + x_1 - x_2 = 1, x_i \in [0,1]$
$(\phi \rightarrow \psi) \rightarrow r$	$\mapsto$	$x_1 \rightarrow \phi, \psi \rightarrow x_2, y - r \leq 0, y + x_1 \leq 1, y \leq x_2, y + r + x_1 - x_2 = 1, x_i \in [0,1], y \in \{0, 1$

- After applying all the rules to $\mathcal{T} \cup \{\phi \rightarrow x\}$ (x variable), we have to solve a MILP problem of the form

$$\min \mathbf{c} \cdot \mathbf{x} \text{ s.t. } A\mathbf{x} + B\mathbf{y} \geq \mathbf{h}$$

where $a_{ij}, b_{ij}, c_l, h_k \in [0,1]$, x_i admits solutions in $[0,1]$, while y_j admits solutions in $\{0,1\}$

Example

- Consider $\mathcal{T} = \{0.6 \to p, 0.7 \to (p \to q)\}$
- Let us show that $|q|_{\mathcal{T}} = 0.6 \wedge 0.7 = \max(1, 0.6 + 0.7 - 1) = 0.3$
- Recall that $|q|_{\mathcal{T}} = \min x.$ such that $\mathcal{T} \cup \{q \to x\}$

$$
\begin{array}{ll}
 & \mathcal{T} \cup \{q \to x\} = \{0.6 \to p, 0.7 \to (p \to q), q \to x, x \in [0,1]\} \\
\mapsto & \{x_p \geq 0.6, x_q \leq x, 0.7 \to (p \to q), \{x, x_p\} \subseteq [0,1]\} \\
\mapsto & \{x_p \geq 0.6, x_q \leq x, p \to x_1, x_2 \to q, 0.7 + x_1 - x_2 = 1, \{x, x_p, x_i\} \subseteq [0,1]\} \\
\mapsto & \{x_p \geq 0.6, x_q \leq x, x_p \leq x_1, x_q \geq x_2, 0.7 + x_1 - x_2 = 1, \{x, x_p, x_i\} \subseteq [0,1]\} = S
\end{array}
$$

It follows that $0.3 = \min x.$ such that $Sat(S)$

- Note: A similar technique can be used for logic G and Π, but mixed integer non-linear programming is needed in place of MILP

Predicate Fuzzy Logics Basics

- Formulae: First-Order Logic formulae, *terms* are either variables or constants
  - we may introduce functions symbols as well, with crisp semantics (but uninteresting), or we need to discuss also fuzzy equality (which we leave out here)
- Truth space is $[0, 1]$
- Formulae have a a degree of truth in $[0, 1]$
- Interpretation: is a mapping $\mathcal{I} : \mathit{Atoms} \to [0, 1]$
- Interpretations are extended to formulae as follows:

$$
\begin{array}{rcl}
(\neg\phi & = & \phi \to 0) \\
\mathcal{I}(\phi \wedge \psi) & = & \mathcal{I}(\phi) \wedge \mathcal{I}(\psi) \\
\mathcal{I}(\phi \to \psi) & = & \mathcal{I}(\phi) \to \mathcal{I}(\psi) \\
\mathcal{I}(\exists x \phi) & = & \sup_{c \in \Delta^{\mathcal{I}}} \mathcal{I}^c_x(\phi) \\
\mathcal{I}(\forall x \phi) & = & \inf_{c \in \Delta^{\mathcal{I}}} \mathcal{I}^c_x(\phi)
\end{array}
$$

  where $\mathcal{I}^c_x$ is as $\mathcal{I}$, except that variable x is mapped into individual c
- Definitions of $\mathcal{I} \models \phi$, $\mathcal{I} \models \mathcal{T}$, $\models \phi$, $\mathcal{T} \models \phi$, $||\phi||_{\mathcal{T}}$ and $|\phi|_{\mathcal{T}}$ are as for the propositional case

Axioms of logic $\mathcal{C}\forall$, where $\mathcal{C} \in \{$BL,Ł,Π,G$\}$

(Axioms)	Axioms of $\mathcal{C}$
($\forall$1)	$\forall x\phi(x) \to \phi(t)$ (t substitutable for x in $\phi(x)$)
($\exists$1)	$\phi(t) \to \exists x\phi(x)$ (t substitutable for x in $\phi(x)$)
($\forall$2)	$\forall x(\psi \to \phi) \to (\psi \to \forall x\phi)$ (x not free in ψ)
($\exists$2)	$\forall x(\phi \to \psi) \to (\exists x\phi \to \psi)$ (x not free in ψ)
($\forall$3)	$\forall x(\phi \bar{\vee} \psi) \to (\forall x\phi)\bar{\vee}\psi$ (x not free in ψ)
(Modus ponens)	from ϕ and $\phi \to \psi$ infer ψ
(Generalization)	from ϕ infer $\forall x\phi$

Proposition

$\mathcal{T} \vdash_{\mathcal{C}} \phi$ iff $\mathcal{T} \models_{\mathcal{C}} \phi$.

- if $\to$ is an r-implication then $||\psi||_{\mathcal{T}} \geq ||\phi||_{\mathcal{T}} \wedge ||\phi \to \psi||_{\mathcal{T}}$
- $\models_{BL\forall} \exists x\phi \to \neg\forall x\neg\phi$
- $\models_{BL\forall} \neg\exists x\phi \equiv \forall x\neg\phi$
- $\models_{Ł\forall} \exists x\phi \equiv \neg\forall x\neg\phi$

- $(\neg\forall x p(x)) \wedge (\neg\exists x \neg p(x))$ has no classical model. In Gödel logic it has no finite model, but has an infinite model: for integer $n \geq 1$, let $\mathcal{I}$ such that $p^{\mathcal{I}}(n) = 1/n$

$$\begin{array}{rcl} (\forall x p(x))^{\mathcal{I}} & = & \inf_n 1/n = 0 \\ (\exists x \neg p(x))^{\mathcal{I}} & = & \sup_n \neg 1/n = \sup 0 = 0 \end{array}$$

- Note: If $\mathcal{I} \models \exists x \phi(x)$ then not necessarily there is $c \in \Delta^{\mathcal{I}}$ such that $\mathcal{I} \models \phi(c)$.

$$\begin{array}{rcl} \Delta^{\mathcal{I}} & = & \{n \mid \text{ integer } n \geq 1\} \\ p^{\mathcal{I}}(n) & = & 1 - 1/n < 1, \text{ for all } n \\ (\exists x p(x))^{\mathcal{I}} & = & \sup_n 1 - 1/n = 1 \end{array}$$

- Witnessed formula: $\exists x \phi(x)$ is witnessed in $\mathcal{I}$ iff there is $c \in \Delta^{\mathcal{I}}$ such that $(\exists x \phi(x))^{\mathcal{I}} = (\phi(c))^{\mathcal{I}}$ (similarly for $\forall x \phi(x)$)
- Witnessed interpretation: $\mathcal{I}$ witnessed if all quantified formulae are witnessed in $\mathcal{I}$

Proposition

In Ł, ϕ is satisfiable iff there is a witnessed model of ϕ.

The proposition does not hold for logic G and Π

Predicate Rational Pavelka Logic (RPL∀)

- Fix Łukasiewicz t-norm and r-implication
- Formulae are as for Ł∀, where rationals $r \in [0, 1]$ may appear as atoms

(Axioms and rules) As for Ł∀

Proposition

$\mathcal{T} \vdash_{RPL\forall} \phi$ *iff* $\mathcal{T} \models_{RPL\forall} \phi$.

Fuzzy DLs Basics

- In classical DLs, a concept C is interpreted by an interpretation $\mathcal{I}$ as a set of individuals
- In fuzzy DLs, a concept C is interpreted by $\mathcal{I}$ as a fuzzy set of individuals
- Each individual is instance of a concept to a degree in $[0, 1]$
- Each pair of individuals is instance of a role to a degree in $[0, 1]$

Fuzzy $\mathcal{ALC}$

The semantics is an immediate consequence of the First-Order-Logic translation of DLs expressions

Interpretation:

$$\begin{array}{lcl} \mathcal{I} & = & \Delta^{\mathcal{I}} \\ C^{\mathcal{I}} & : & \Delta^{\mathcal{I}} \rightarrow [0,1] \\ R^{\mathcal{I}} & : & \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \rightarrow [0,1] \end{array} \qquad \begin{array}{lcl} \wedge & = & \text{t-norm} \\ \vee & = & \text{s-norm} \\ \neg & = & \text{negation} \\ \rightarrow & = & \text{implication} \end{array}$$

Concepts:

Syntax			Semantics		
C, D	$\longrightarrow$	$\top$ |	$\top^{\mathcal{I}}(x)$	$=$	1
		$\bot$ |	$\bot^{\mathcal{I}}(x)$	$=$	0
		A |	$A^{\mathcal{I}}(x)$	$\in$	$[0,1]$
		$C \sqcap D$ |	$(C_1 \sqcap C_2)^{\mathcal{I}}(x)$	$=$	$C_1{}^{\mathcal{I}}(x) \wedge C_2{}^{\mathcal{I}}(x)$
		$C \sqcup D$ |	$(C_1 \sqcup C_2)^{\mathcal{I}}(x)$	$=$	$C_1{}^{\mathcal{I}}(x) \vee C_2{}^{\mathcal{I}}(x)$
		$\neg C$ |	$(\neg C)^{\mathcal{I}}(x)$	$=$	$\neg C^{\mathcal{I}}(x)$
		$\exists R.C$ |	$(\exists R.C)^{\mathcal{I}}(x)$	$=$	$\sup_{y \in \Delta^{\mathcal{I}}} R^{\mathcal{I}}(x,y) \wedge C^{\mathcal{I}}(y)$
		$\forall R.C$	$(\forall R.C)^{\mathcal{I}}(u)$	$=$	$\inf_{y \in \Delta^{\mathcal{I}}} R^{\mathcal{I}}(x,y) \rightarrow C^{\mathcal{I}}(y)\}$

Assertions: $\langle a{:}C, r\rangle$, $\mathcal{I} \models \langle a{:}C, r\rangle$ iff $C^{\mathcal{I}}(a^{\mathcal{I}}) \geq r$ (similarly for roles)

- individual a is instance of concept C at least to degree r, $r \in [0,1] \cap \mathbb{Q}$

Inclusion axioms: $C \sqsubseteq D$,

- $\mathcal{I} \models C \sqsubseteq D$ iff $\forall x \in \Delta^{\mathcal{I}}.C^{\mathcal{I}}(x) \leq D^{\mathcal{I}}(x)$
- this is equivalent to, $\forall x \in \Delta^{\mathcal{I}}.(C^{\mathcal{I}}(x) \rightarrow D^{\mathcal{I}}(x)) = 1$, if $\rightarrow$ is an r-implication

Basic Inference Problems

Consistency: Check if knowledge is meaningful

- Is $\mathcal{K}$ consistent, i.e. satisfiable?

Subsumption: structure knowledge, compute taxonomy

- $\mathcal{K} \models C \sqsubseteq D$?

Equivalence: check if two fuzzy concepts are the same

- $\mathcal{K} \models C = D$?

Graded instantiation: Check if individual a instance of class C to degree at least r

- $\mathcal{K} \models \langle a{:}C, r\rangle$?

BTVB: Best Truth Value Bound problem

- $|a{:}C|_{\mathcal{K}} = \sup\{r \mid \mathcal{K} \models \langle a{:}C, r\rangle\}$?

Top-k retrieval: Retrieve the top-k individuals that instantiate C w.r.t. best truth value bound

- $ans_{top-k}(\mathcal{K}, C) = Top_k\{\langle a, v\rangle \mid v = |a{:}C)|_{\mathcal{K}}\}$

Some Notes on ...

- Value restrictions:
  - In classical DLs, $\forall R.C \equiv \neg\exists R.\neg C$
  - The same is not true, in general, in fuzzy DLs (depends on the operators' semantics, true for Łukasiewicz, but not true in Gödel logic)
  - Is it acceptable that $\forall hasParent.Human \not\equiv \neg\exists hasParent.\neg Human$? Recall that in Ł and Zadeh, $\forall x.\phi \equiv \neg\exists x\ \neg\phi$
- Models:
  - In classical DLs $\top \sqsubseteq \neg(\forall R.A) \sqcap (\neg\exists R.\neg A)$ has no classical model
  - In Gödel logic it has no finite model, but has an infinite model
- The choice of the appropriate semantics of the logical connectives is important.
  - Should have reasonable logical properties
  - Certainly it must have efficient algorithms solving basic inference problems
- Łukasiewicz Logic seems the best compromise, though Zadeh semantics has been considered historically in DLs (we recall that Zadeh semantics is not considered by fuzzy logicians)
- For disjointness it is better to use $C \sqcap D \sqsubseteq \perp$ rather than $C \sqsubseteq \neg D$
  - they are not the same, e.g. $A \sqsubseteq \neg A$ says that $A^{\mathcal{I}}(x) \leq 0.5$ holds, for all $\mathcal{I}$ and for all $x \in \Delta^{\mathcal{I}}$ (under Łukasiewicz Logic)

Towards fuzzy OWL Lite and OWL DL

- Recall that OWL Lite and OWL DL relate to $\mathcal{SHIF}(D)$ and $\mathcal{SHOIN}(D)$, respectively
- We need to extend the semantics of fuzzy $\mathcal{ALC}$ to fuzzy $\mathcal{SHOIN}(D) = \mathcal{ALCHOINR}_+(D)$
- Additionally, we add
  - modifiers (e.g., *very*)
  - concrete fuzzy concepts (e.g., *Young*)
  - both additions have explicit membership functions

Number Restrictions, Inverse and Transitive roles

- The semantics of the concept $(\geq n\ S)$ is:

$$(\geq n\ R)^{\mathcal{I}}(x) \quad = \quad \sup_{\{y_1,\ldots,y_n\}\subseteq\Delta^{\mathcal{I}}} \bigwedge_{i=1}^{n} R^{\mathcal{I}}(x, y_i)$$

- It is the result of viewing $(\geq n\ R)$ as the open first order formula

$$\exists y_1, \ldots, y_n. \bigwedge_{i=1}^{n} R(x, y_i) \wedge \bigwedge_{1\leq i<j\leq n} y_i \neq y_j\ .$$

- The semantics of the concept $(\leq n\ R)$ is:

$$(\leq n\ R)^{\mathcal{I}}(x) = \neg(\geq n+1\ R)^{\mathcal{I}}(x)$$

- Note: $(\geq 1\ R) \equiv \exists R.\top$
- For transitive roles we have for all $x, y \in \Delta^{\mathcal{I}}$

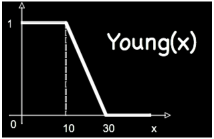
$$R^{\mathcal{I}}(x, y) = R^{\mathcal{I}}(y, x)$$

- For transitive roles R we impose: for all $x, y \in \Delta^{\mathcal{I}}$

$$R^{\mathcal{I}}(x, y) \geq \sup_{z\in\Delta^{\mathcal{I}}} \min(R^{\mathcal{I}}(x, z), R^{\mathcal{I}}(z, y))$$

Concrete fuzzy concepts

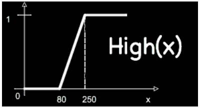
- E.g., *Small*, *Young*, *High*, *etc*. with explicit membership function
- Use the idea of concrete domains:
  - $D = \langle \Delta_D, \Phi_D \rangle$
  - Δ_D is an interpretation domain
  - Φ_D is the set of concrete fuzzy domain predicates d with a predefined arity $n = 1, 2$ and fixed interpretation $d^D \colon \Delta_D^n \to [0, 1]$
  - For instance,



$$
\begin{array}{rcl}
Minor & = & Person \sqcap \exists hasAge. \leq_{18} \\
YoungPerson & = & Person \sqcap \exists hasAge. Young \\
 & & functional(hasAge)
\end{array}
$$

Modifiers

- *Very*, *moreOrLess*, *slightly*, etc.
- Apply to fuzzy sets to change their membership function
  - $very(x) = x^2$
  - $slightly(x) = \sqrt{x}$
- For instance,



$$SportsCar = Car \sqcap \exists speed.very(High)$$

Fuzzy $\mathcal{SHOIN}(\mathbf{D})$

Concepts:

Syntax			Semantics
C, D	$\longrightarrow$	$\top$ |	$\top(x)$
		$\bot$ |	$\bot\ (x)$
		A |	$A(x)$
		$(C \sqcap D)$ |	$C_1(x) \wedge C_2(x)$
		$(C \sqcup D)$ |	$C_1(x) \vee C_2(x)$
		$(\neg C)$ |	$\neg C(x)$
		$(\exists R.C)$ |	$\exists x\ R(x, y) \wedge C(y)$
		$(\forall R.C)$ |	$\forall x\ R(x, y) \rightarrow C(y)$
		$\{a\}$ |	$x = a$
		$(\geq\ n\ R)$ |	$\exists y_1, \ldots, y_n. \bigwedge_{i=1}^{n}\ R(x, y_i) \wedge \bigwedge_{1 \leq i < j \leq n} y_i \neq y_j$
		$(\leq\ n\ R)$ |	$\neg(\geq\ n+1\ R)(x)$
		FCC |	$\mu_{FCC}(x)$
		$M(C)$ |	$\mu_M(C(x))$
R	$\longrightarrow$	P |	$P(x, y)$
		P^- |	$P(y, x)$

Assertions:

Syntax			Semantics
α	$\longrightarrow$	$\langle a{:}C, r\rangle$ |	$r \rightarrow C(a)$
		$\langle (a, b){:}R, r\rangle$	$r \rightarrow R(a, b)$

Axioms:

Syntax			Semantics
τ	$\longrightarrow$	$C \sqsubseteq D$ |	$\forall x\ C(x) \rightarrow D(x) = 1$, where $\rightarrow$ is r-implication
		$fun(R)$ |	$\forall x \forall y \forall z\ R(x, y) \wedge R(x, z) \rightarrow y = z$
		$trans(R)$	$(\exists z\ R(x, z) \wedge R(z, y)) \rightarrow R(x, y)$

Reasoning

Depends on the semantics and reasoning method (tableau-based or MILP-based)

Tableaux method: under Zadeh semantics

- a tableau exists for fuzzy $\mathcal{SHIN}$, solving the satisfiability problem
- classical blocking methods apply similarly in the fuzzy variant
- the management of General concept inclusions (GCI's) is more complicated compared to the crisp case
- a translation of fuzzy $\mathcal{SHOIN}$ to crisp $\mathcal{SHOIN}$ also exists (not addressed here)
- the tableaux method is not suitable to deal with fuzzy concrete concepts and modifiers
- the BTVB can be solved, but not efficiently

MILP based method: under Zadeh semantics, Łukasiewicz semantics, and classical semantics

- exists for fuzzy $\mathcal{ALC}$ + linear modifiers + fuzzy concrete concepts (published)
- exists for fuzzy $\mathcal{SHIF}$ + linear modifiers + fuzzy concrete concepts (implemented, but not published yet)
- solves the BTVB as primary problem

Fuzzy tableaux-based method

- Tableau algorithm is similar to classical DL tableaux
- Most problems can be reduced to satisfiability problem, e.g.
- Assertions are extended to $\langle a{:}C \geq n\rangle$, $\langle a{:}C \leq n\rangle$, $\langle a{:}C > n\rangle$ and $\langle a{:}C < n\rangle$
- $\mathcal{K} \models \langle a{:}C, n\rangle$ iff $\mathcal{K} \cup \{\langle a{:}C < n\rangle\}$ not satisfiable
  - All models of $\mathcal{K}$ do not satisfy $\langle a{:}C < n\rangle$, i.e. do satisfy $\langle a{:}C \geq n\rangle$
- Let's see a tableaux algorithm for satisfiability checking, where

$$
\begin{array}{rcl}
x \wedge y & = & \min(x, y) \\
x \vee y & = & \max(x, y) \\
\neg x & = & 1 - x \\
x \rightarrow y & = & \max(1 - x, y)
\end{array}
$$

Tableaux for $\mathcal{ALC}$ KB

- Works on a tree forest (semantics through viewing tree as an ABox)
  - Nodes represent elements of $\Delta^{\mathcal{I}}$, labelled with sub-concepts of C and their weights
  - Edges represent role-successorships between elements of $\Delta^{\mathcal{I}}$ and their weights
- Works on concepts in negation normal form: push negation inside using de Morgan' laws and

$$
\begin{array}{rcl}
\neg(\exists R.C) & \mapsto & \forall R.\neg C \\
\neg(\forall R.C) & \mapsto & \exists R.\neg C
\end{array}
$$

- It is initialised with a tree forest consisting of root nodes a, for all individuals appearing in the KB:
  - If $\langle a{:}C \bowtie n\rangle \in \mathcal{K}$ then $\langle C, \bowtie, n\rangle \in \mathcal{L}(a)$
  - If $\langle (a,b){:}R \bowtie n\rangle \in \mathcal{K}$ then $\langle\langle a,b\rangle, \bowtie, n\rangle \in \mathcal{E}(R)$
- A tree forest T contains a clash if for a tree T in the forest there is a node x in T, containing a conjugated pair $\{\langle A, \rhd, n\rangle, \langle C, \lhd, m\rangle\} \subseteq \mathcal{L}(x)$, e.g. $\langle A, \geq, 0.6\rangle, \langle A, <, 0.3\rangle$
- Returns "$\mathcal{K}$ is satisfiable" if rules can be applied s.t. they yield a clash-free, complete (no more rules apply) tree forest

$\mathcal{ALC}$ Tableau rules (excerpt)

$x \bullet \{\langle C_1 \sqcap C_2, \geq, n\rangle, \ldots\}$	$\longrightarrow_\sqcap$	$x \bullet \{\langle C_1 \sqcap C_2, \geq, n\rangle, \langle C_1, \geq, n\rangle, \langle C_2, \geq, n\rangle, \ldots\}$
$x \bullet \{\langle C_1 \sqcup C_2, \geq, n\rangle, \ldots\}$	$\longrightarrow_\sqcup$	$x \bullet \{\langle C_1 \sqcup C_2, \geq, n\rangle, \langle C, \geq, n\rangle, \ldots\}$ for $C \in \{C_1, C_2\}$
$x \bullet \{\langle \exists R.C, \geq, n\rangle, \ldots\}$	$\longrightarrow_\exists$	$x \bullet \{\langle \exists R.C, \geq, n\rangle, \ldots\}$ $\langle R, \geq, n\rangle \downarrow$ $y \bullet \{\langle C, \geq, n\rangle\}$
$x \bullet \{\langle \forall R.C, \geq, n\rangle, \ldots\}$ $\langle R, \geq, m\rangle \downarrow \quad (m > 1 - n)$ $y \bullet \{\ldots\}$	$\longrightarrow_\forall$	$x \bullet \{\langle \forall R.C, \geq, n\rangle, \ldots\}$ $\langle R, \geq, m\rangle \downarrow$ $y \bullet \{\ldots, \langle C, \geq, n\rangle\}$
$x \bullet \{C \sqsubseteq D, \ldots\}$	$\longrightarrow_\sqsubseteq$	$x \bullet \{C \sqsubseteq D, E, \ldots\}$ for $E \in \{\langle C, <, n\rangle, \langle D, \geq, n\rangle\}, n \in N^{\mathcal{A}}$
$\vdots$	$\vdots$	$\vdots$

$$
\begin{aligned}
\mathcal{K} &= \langle \mathcal{T}, \mathcal{A}\rangle \\
X^{\mathcal{A}} &= \{0, 0.5, 1\} \cup \{n \mid \langle \alpha \bowtie n\rangle \in \mathcal{A}\} \\
N^{\mathcal{A}} &= X^{\mathcal{A}} \cup \{1 - n \mid n \in X^{\mathcal{A}}\}
\end{aligned}
$$

Theorem

Let $\mathcal{K}$ be an $\mathcal{ALC}$ KB and F obtained by applying the tableau rules to $\mathcal{K}$. Then

1. *The rule application terminates,*
2. *If F is clash-free and complete, then F defines a (canonical) (tree forest) model for $\mathcal{K}$, and*
3. *If $\mathcal{K}$ has a model $\mathcal{I}$, then the rules can be applied such that they yield a clash-free and complete forest F.*

It is expected that the tableau can be modified to a decision procedure for

- $\mathcal{SHOIN}$ $(\equiv \mathcal{ALCHOINR}_+)$

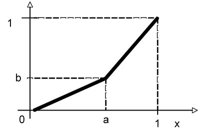
Problem with fuzzy tableau

- Usual fuzzy tableaux calculus does not work anymore with
  - modifiers and concrete fuzzy concepts
  - Łukasiewicz Logic
- Usual fuzzy tableaux calculus does not solve the BTVB problem
- New algorithm uses bounded Mixed Integer Programming oracle, as for Many Valued Logics
  - Recall: the *general MILP problem* is to find

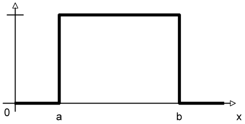
$$\bar{\mathbf{x}} \in \mathbb{Q}^k, \bar{\mathbf{y}} \in \mathbb{Z}^m$$
$$f(\bar{\mathbf{x}}, \bar{\mathbf{y}}) = \min\{f(\mathbf{x}, \mathbf{y}) \colon A\mathbf{x} + B\mathbf{y} \geq \mathbf{h}\}$$
A, B integer matrixes

Requirements

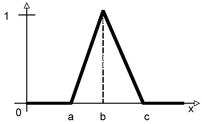
- Works for usual fuzzy DL semantics (Zadeh semantics) and Lukasiewicz logic
- Modifiers are definable as linear in-equations over $\mathbb{Q}, \mathbb{Z}$ (e.g., linear hedges), for instance, linear hedges, $lm(a, b)$, e.g. $very = lm(0.7, 0.49)$
- Fuzzy concrete concepts are definable as linear in-equations over $\mathbb{Q}, \mathbb{Z}$ (e.g., crisp, triangular, trapezoidal, left shoulder and right shoulder membership functions)



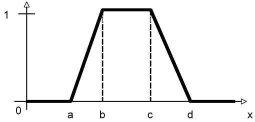
lm(a,b)



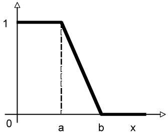
cr(a,b)



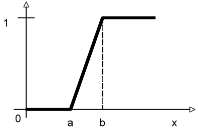
tri(a,b,c)



trz(a,b,c,d)

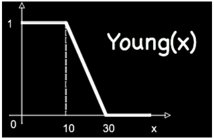


ls(a,b)



rs(a,b,c)

- Example:



$$
\begin{array}{rcl}
Minor & = & Person \sqcap \exists hasAge. \leq_{18} \\
YoungPerson & = & Person \sqcap \exists hasAge. Young \\
Young & = & ls(10, 30) \\
\leq_{18} & = & cr(0, 18)
\end{array}
$$

- Then

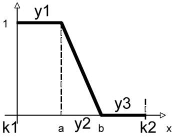
$$
\begin{array}{rcl}
|a{:}C|_{\mathcal{K}} & = & \min\{x \mid \mathcal{K} \cup \{\langle a{:}C \leq x\rangle \text{ satisfiable}\} \\
|C \sqsubseteq D|_{\mathcal{K}} & = & \min\{x \mid \mathcal{K} \cup \{\langle a{:}C \sqcap \neg D \geq 1 - x\rangle \text{ satisfiable}\}
\end{array}
$$

  - Apply (deterministic) tableaux calculus, then use bounded Mixed Integer Programming oracle

$\mathcal{ALC}$ MILP Tableau rules under Zadeh semantics (excerpt)

$x \bullet \{\langle C_1 \sqcap C_2, \geq, l \rangle, \ldots\}$	$\longrightarrow_{\sqcap}$	$x \bullet \{\langle C_1 \sqcap C_2, \geq, l \rangle, \langle C_1, \geq, l \rangle, \langle C_2, \geq, l \rangle, \ldots\}$
$x \bullet \{\langle C_1 \sqcup C_2, \geq, l \rangle, \ldots\}$	$\longrightarrow_{\sqcup}$	$x \bullet \{\langle C_1 \sqcup C_2, \geq, l \rangle, \langle C_1, \geq, x_1 \rangle, \langle C_2, \geq, x_2 \rangle,$ $x_1 + x_2 = l, x_1 \leq y, x_2 \leq 1 - y,$ $x_i \in [0, 1], y \in \{0, 1\}, \ldots\}$
$x \bullet \{\langle \exists R.C, \geq, l \rangle, \ldots\}$	$\longrightarrow_{\exists}$	$x \bullet \{\langle \exists R.C, \geq, l \rangle, \ldots\}$ $\langle R, \geq, l \rangle \downarrow$ $y \bullet \{\langle C, \geq, l \rangle\}$
$x \bullet \{\langle \forall R.C, \geq, l_1 \rangle, \ldots\}$ $\langle R, \geq, l_2 \rangle \downarrow$ $y \bullet \{\ldots\}$	$\longrightarrow_{\forall}$	$x \bullet \{\langle \forall R.C, \geq, l_1 \rangle, \ldots\}$ $\langle R, \geq, l_2 \rangle \downarrow$ $y \bullet \{\ldots, \langle C, \geq, x \rangle$ $x + y \geq l_1, x \leq y, l_1 + l_2 \leq 2 - y,$ $x \in [0, 1], y \in \{0, 1\}\}$
$x \bullet \{A \sqsubseteq C, \langle A, \geq, l \rangle, \ldots\}$	$\longrightarrow_{\sqsubseteq_1}$	$x \bullet \{A \sqsubseteq C, \langle C, \geq, l \rangle, \ldots\}$
$x \bullet \{C \sqsubseteq A, \langle A, \leq, l \rangle, \ldots\}$	$\longrightarrow_{\sqsubseteq_2}$	$x \bullet \{C \sqsubseteq A, \langle C, \leq, l \rangle, \ldots\}$
$x \bullet \{C \sqsubseteq D, \ldots\}$	$\longrightarrow_{\sqsubseteq}$	$x \bullet \{C \sqsubseteq D, \langle C, \leq, x \rangle, \langle D, \geq, x \rangle, x \in [0, 1], \ldots\}$
$x \bullet \{\langle ls(k_1, k_2, a, b), \geq, l \rangle, \ldots\}$	$\longrightarrow_{\sqsubseteq}$	$x \bullet \{ls(k_1, k_2, a, b), y_1 + y_2 + y_3 = 1, y_i \in \{0, 1\},$ $x + (k_2 - a) \cdot y_1 \leq k_2, x + (k_1 - a) \cdot y_2 \geq k_1,$ $x + (k_2 - b) \cdot y_2 \geq k_2,$ $x + (b - a) \cdot l + (k_2 - a) \cdot y_2 \leq k_2 - a + b,$ $x + (k_1 - b) \cdot y_3 \leq k_1, l + y_3 \leq 1, \ldots\}$
$\vdots$	$\vdots$	$\vdots$

$ls(k_1, k_2, a, b)$ is



Example

- Suppose
$$\mathcal{K} = \begin{cases} A \sqcap B \sqsubseteq C \\ \langle a{:}A \geq 0.3\rangle \\ \langle a{:}B \geq 0.4\rangle \end{cases}$$

$$Query := |a{:}C|_{\mathcal{K}} = \min\{x \mid \mathcal{K} \cup \{\langle a{:}C \leq x\rangle \text{ satisfiable}\}$$

Step	Tree	
1.	$a \bullet \{\langle A, \geq, 0.3\rangle, \langle B, \geq, 0.4\rangle, \langle C, \leq, x\rangle\}$	(Hypothesis)
2.	$\cup\{\langle A \sqcap B, \leq, x\rangle\}$	$(\to_{\sqsubseteq_2})$
3.	$\cup\{\langle A, \leq, x_1\rangle, \langle B, \leq, x_2\rangle\}$ $\cup\{x = x_1 + x_2 - 1, 1 - y \leq x_1, y \leq x_2\}$ $\cup\{x_i \in [0, 1], y \in \{0, 1\}\}$	$(\to_{\sqcap_{\leq}})$
4.	find $\min\{x \mid \langle a{:}A \geq 0.3\rangle, \langle a{:}B \geq 0.4\rangle,$ $\langle a{:}C \leq x\rangle, \langle a{:}A \leq x_1\rangle, \langle a{:}B \leq x_2\rangle,$ $x = x_1 + x_2 - 1, 1 - y \leq x_1, y \leq x_2,$ $x_i \in [0, 1], y \in \{0, 1\}\}$	(MILP Oracle)
5.	MILP oracle: $\mathbf{x} = \mathbf{0.3}$	

Implementation issues

- Several options exists:
  - Try to map fuzzy DLs to classical DLs
    - but, does not work with modifiers and concrete fuzzy concepts
  - Try to map fuzzy DLs to some fuzzy logic programming framework
    - A lot of work exists about mappings among classical DLs and LPs
    - But, needs a theorem prover for fuzzy LPs (not addressed here)
    - To be used then e.g. in the axiomatic approach to fuzzy DLPs (Description Logic Programs)
  - Build an ad-hoc theorem prover for fuzzy DLs, using e.g., MILP
    - To be used then separately e.g. in the DL-log approach to fuzzy DLPs
- A theorem prover for fuzzy $\mathcal{SHIF}$ + linear hedges + concrete fuzzy concepts, using MILP, has been implemented (`http://gaia.isti.cnr.it/~straccia`)

Future Work on fuzzy DLs

- Research directions:
  - Computational complexity of the fuzzy DLs family
  - Design of efficient reasoning algorithms
  - Combining fuzzy DLs with fuzzy Logic Programming
  - Language extensions: e.g. fuzzy quantifiers

$$\begin{array}{l} \mathit{TopCustomer} = \mathit{Customer} \sqcap (\mathit{Usually})\mathit{buys}.\mathit{ExpensiveItem} \\ \mathit{ExpensiveItem} = \mathit{Item} \sqcap \exists \mathit{price}.\mathit{High} \end{array}$$

  - Conjunctive query answering (top-k query answering) for more expressive DLs
  - Developing systems, extending fuzzyDL system, ...
  - Applications, e.g. Ontology mediated data access ((distributed) multimedia information retrieval, resource selection, ...), Negotiation, Health-Care, ...
  - ...