Talk at

Karlsruhe, Germany, November 25^{th}, 2005

Fuzzy Description Logics, Fuzzy Logic Programming, their Combination (and the Semantic Web)

Umberto Straccia

I.S.T.I. - C.N.R. Pisa, Italy

straccia@isti.cnr.it

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

Outline

- Preliminaries: short recall on classical
  - Description Logics (DLs)
  - Logic Programs (LPs)
  - Description Logic Programs (DLPs)
- Semantic Web and Ontologies
- Fuzzy
  - Description Logics
  - Logic Programs
  - Description Logic Programs
- Conclusions

Basics of
Description Logics
Logic Programs
Description Logic Programs

DLs Basics

- Concept names are equivalent to unary predicates
  - In general, concepts equiv to formulae with one free variable
- Role names are equivalent to binary predicates
  - In general, roles equiv to formulae with two free variables
- 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				*Example*
C, D $\rightarrow$	$\top$	|	(top concept)	
	$\bot$	|	(bottom concept)	
	A	|	(atomic concept)	`Human`
	$C \sqcap D$	|	(concept conjunction)	`Human` $\sqcap$ `Male`
	$C \sqcup D$	|	(concept disjunction)	`Nice` $\sqcap$ `Rich`
	$\neg C$	|	(concept negation)	$\neg$`Meat`
	$\exists R.C$	|	(existential quantification)	$\exists$`has_child.Blond`
	$\forall R.C$		(universal quantification)	$\forall$`has_child.Human`
$C \sqsubseteq D$			(inclusion axiom)	`Happy_Father` $\sqsubseteq$ `Man` $\sqcap$ $\exists$`has_child.Female`
a:C			(assertion)	`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:

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

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.

$\mathcal{F}$**:** Functional role, f

For instance,

$$\mathcal{SHIF} = \mathcal{S} + \mathcal{H} + \mathcal{I} + \mathcal{F} = \mathcal{ALCR}_+\mathcal{HIF}$$
$$\mathcal{SHOIN} = \mathcal{S} + \mathcal{H} + \mathcal{O} + \mathcal{I} + \mathcal{N} = \mathcal{ALCR}_+\mathcal{HOIN}$$

Concrete domains

- Concrete domains: integers, strings, ...
- Clean separation between object classes and concrete domains
  - $\mathtt{D} = \langle \Delta_{\mathtt{D}}, \Phi_{\mathtt{D}} \rangle$
  - $\Delta_{\mathtt{D}}$ is an interpretation domain
  - $\Phi_{\mathtt{D}}$ is the set of concrete domain predicates d with a predefined arity n and fixed interpretation $d^{\mathtt{D}} \colon \Delta_{\mathtt{D}}^n \rightarrow \{0,1\}$
  - Concrete properties: $R^{\mathcal{I}} \colon \Delta^{\mathcal{I}} \times \Delta_D \rightarrow \{0,1\}$, e.g., $(\mathtt{tim}, \mathtt{14})$:$\mathtt{hasAge}$, $(\mathtt{sf}, \text{“}\mathtt{SoftComputing}\text{”})$:$\mathtt{hasAcronym}$
- 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: $(\mathtt{D})$. E.g., $\mathcal{ALC}(\mathtt{D})$ is $\mathcal{ALC}$ + concrete domains

LPs Basics (for ease, without default negation)

- Predicates are n-ary
- Terms are variables or constants
- Rules are of the form

$$B_1(\mathbf{x_1}) \wedge \ldots \wedge B_n(\mathbf{x_n}) \Rightarrow P(\mathbf{x})$$

  For instance,

$$\texttt{has\_parent}(\texttt{x}, \texttt{y}) \wedge \texttt{Male}(\texttt{y}) \quad \Rightarrow \quad \texttt{has\_father}(\texttt{x}, \texttt{y})$$

- Facts are rules with empty body

  For instance,

$$\texttt{has\_parent}(\texttt{mary}, \texttt{jo})$$

LPs Semantics: FOL semantics

- $\mathcal{P}^*$ is constructed as follows:
  1. set $\mathcal{P}^*$ to the set of all ground instantiations of rules in $\mathcal{P}$;
  2. if atom A is not head of any rule in $\mathcal{P}^*$, then add $0 \Rightarrow A$ to $\mathcal{P}^*$;
  3. replace several rules in $\mathcal{P}^*$ having same head

$$\left.\begin{array}{c} \varphi_1 \Rightarrow A \\ \varphi_2 \Rightarrow A \\ \vdots \\ \varphi_n \Rightarrow A \end{array}\right\} \text{ with } \varphi_1 \vee \varphi_2 \vee \ldots \vee \varphi_n \Rightarrow A \ .$$

- Note: in $\mathcal{P}^*$ each atom $A \in B_{\mathcal{P}}$ is head of exaclty one rule
- Herbrand Base of $\mathcal{P}$ is the set $B_{\mathcal{P}}$ of ground atoms
- Interpretation is a function $I : B_{\mathcal{P}} \to \{0, 1\}$.
- Model $I \models \mathcal{P}$ iff for all $r \in \mathcal{P}^*$ $I \models r$, where $I \models \varphi \Rightarrow A$ iff $I(\varphi) \leq I(A)$
- Least model exists and is least fixed-point of $T_{\mathcal{P}}(I)(A) = I(\varphi)$, for all $\varphi \Rightarrow A \in \mathcal{P}^*$

DLPs Basics

- Combine DLs with LPs:
  - DL atoms and roles may appear in rules

$$\texttt{made\_by}(x, y) \wedge \langle\texttt{Chinese\_Company}\rangle(y) \Rightarrow \texttt{prize}(x, \texttt{low})$$

$$\texttt{Chinese\_Company} \sqsubseteq \exists\texttt{has\_location}.\texttt{China}$$

- Knowledge Base is a pair $KB = \langle \mathcal{P}, \Sigma \rangle$, where
  - $\mathcal{P}$ is a logic program
  - Σ is a DL knowledge base (set of assertions and inclusion axioms)

DLPs Semantics

- Semantics: two main approaches
  1. Axiomatic approach: DL atoms and roles are managed uniformely
     - I is a model of $KB = \langle \mathcal{P}, \Sigma \rangle$ iff $I \models \mathcal{P}$ and $I \models \Sigma$
  2. DL-log approach: DL atoms and roles are procedural attachments (calls to a DL theorem prover)
     - I is a model of $KB = \langle \mathcal{P}, \Sigma \rangle$ iff $I^{\Sigma} \models \mathcal{P}$
     - I^{Σ} is a model of a ground non-DL atom $A \in B_{\mathcal{P}}$ iff $I(A) = 1$
     - I^{Σ} is a model of a ground DL atom $\langle A \rangle(a)$ iff $\Sigma \models a{:}A$
     - I^{Σ} is a model of a ground DL role $\langle R \rangle(a, b)$ iff $\Sigma \models (a, b){:}R$
- Axiomatic approach: easy to get undecidability results (e.g. recursive rules + $\forall$)
- DL-log entailment $\subsetneq$ Axiomatic entailment
- Axiomatic approach does not enjoy the minimal model property of LPs
- DL-log has the minimal model property of LPs and a fixed-point characterization: $T_{\mathcal{P}}(I)(A) = I^{\Sigma}(\varphi)$, for all $\varphi \Rightarrow A \in \mathcal{P}^*$

Basics of the Semantic Web and Ontologies

The Semantic Web Vision and DLs

- 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

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

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($C_1 \ldots C_n$)	$C_i \sqcap C_j = \bot, i \neq j$	Male $\sqsubseteq$ $\neg$Female
ObjectProperty(R super (R_1)... 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` $(U_1)\ldots$ `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` $(C_1)\ldots$ `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., $\texttt{Person} \sqcap \forall\texttt{hasChild}.(\texttt{Doctor} \sqcup \exists\texttt{hasChild}.\texttt{Doctor})$:

```
<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

Logic Programs

Description Logic Programs

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 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:

$$W = \{I \text{ classical interpretation}\} \quad (I(\varphi) \in \{0,1\})$$
$$\mu \colon W \to [0,1] \quad (\mu(I) \in [0,1])$$

- 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.

Example (fuzzy DL-*Lite*, Current work)

$$
\begin{array}{rcl}
\texttt{Hotel} & \sqsubseteq & \exists\texttt{hasLocation} \\
\texttt{Conference} & \sqsubseteq & \exists\texttt{hasLocation} \\
\texttt{Hotel} & \sqsubseteq & \neg\texttt{Conference} \\
\texttt{Location}^{\mathcal{I}} & \subseteq & \texttt{GISCoordinates} \\
\texttt{distance}^{\mathcal{I}} & : & \texttt{GISCoord} \times \texttt{GISCoord} \longrightarrow \mathbb{N} \\
 & & distance(x, y) = \ldots \\
\texttt{close}^{\mathcal{I}} & : & \mathbb{N} \rightarrow [0, 1] \\
 & & close(x) = \max(0, 1 - \frac{x}{1000})
\end{array}
$$

hasLocation	hasLocation	distance
hl1	cl1	300
hl1	cl2	500
hl2	cl1	750
hl2	cl2	750
⋮	⋮	

HotelID	hasLocation
h1	hl1
h2	hl2
⋮	⋮

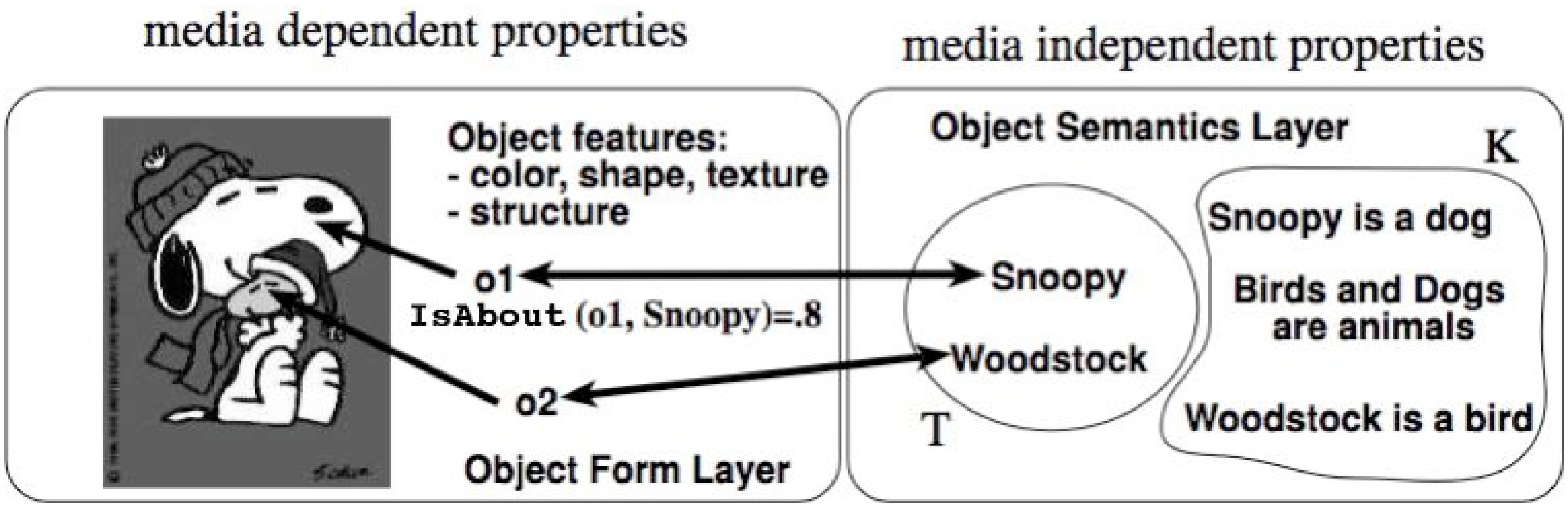
ConferenceID	hasLocation
c1	cl1
c2	cl2
⋮	⋮

HotelID	closeness degree
h1	0.7
h2	0.25
⋮	⋮

"Find a hotel close to conference c1":

$\texttt{Hotel}(h) \wedge \texttt{hasLocation}(h, hl) \wedge \texttt{Conference}(\texttt{c1}) \wedge \texttt{hasLocation}(\texttt{c1}, cl) \wedge \texttt{distance}(hl, cl, d) \wedge \texttt{close}(d) \Rightarrow \texttt{Query}(\texttt{c1}, h)$

Example (Logic-based information retrieval model)



Bird	$\sqsubseteq$	Animal
Dog	$\sqsubseteq$	Animal
snoopy	:	Dog
woodstock	:	Bird

ImageRegion	Object ID	isAbout
o1	snoopy	0.8
o2	woodstock	0.7
⋮	⋮	

$\texttt{ImageRegion}(ir) \wedge \texttt{isAbout}(ir, x) \wedge \texttt{Animal}(x) \Rightarrow \texttt{Query}(ir)$

Example (Graded Entailment)

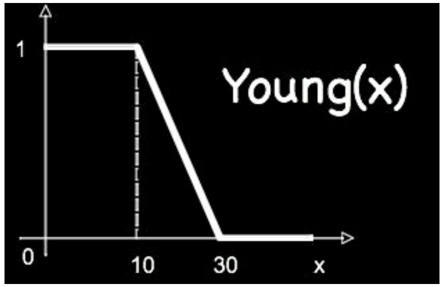
		
audi_tt	mg	ferrari_enzo

Car	speed
audi_tt	243
mg	$\leq$ 170
ferrari_enzo	$\geq$ 350

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

$$\mathcal{K} \models \langle \mathtt{ferrari\_enzo:SportsCar}, 1 \rangle$$
$$\mathcal{K} \models \langle \mathtt{audi\_tt:SportsCar}, 0.92 \rangle$$
$$\mathcal{K} \models \langle \mathtt{audi\_tt:\neg SportsCar}, 0.72 \rangle$$

Example (Graded Subsumption)



$$\begin{aligned} \texttt{Minor} &= \texttt{Person} \sqcap \exists \texttt{hasAge}. \leq_{18} \\ \texttt{YoungPerson} &= \texttt{Person} \sqcap \exists \texttt{hasAge}.\texttt{Young} \end{aligned}$$

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

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

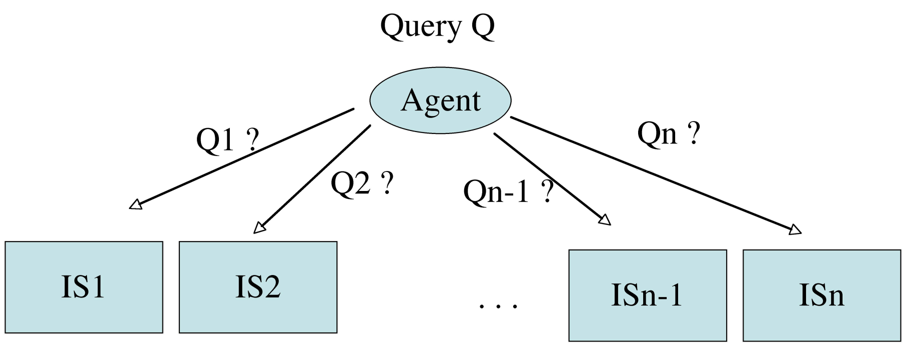
Example with fuzzy LPs (current work)

$$F = \begin{cases} \texttt{Experience}(\texttt{John}) & \leftarrow & 0.7 \\ \texttt{Risk}(\texttt{John}) & \leftarrow & 0.5 \\ \texttt{Sport\_car}(\texttt{John}) & \leftarrow & 0.8 \end{cases}$$

$$R = \begin{cases} \texttt{Good\_driver}(\texttt{X}) & \leftarrow & \texttt{Experience}(\texttt{X}) \wedge \neg\texttt{Risk}(\texttt{X}) \\ \texttt{Risk}(\texttt{X}) & \leftarrow & 0.8 \cdot \texttt{Young}(\texttt{X}) \\ \texttt{Risk}(\texttt{X}) & \leftarrow & 0.8 \cdot \texttt{Sport\_car}(\texttt{X}) \\ \texttt{Risk}(\texttt{X}) & \leftarrow & \texttt{Experience}(\texttt{X}) \wedge \neg\texttt{Good\_driver}(\texttt{X}) \end{cases}$$

Then $R \cup F \models \langle \texttt{Risk}(\texttt{John}), 0.64 \rangle$

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)

$$\texttt{Mechanical\_and\_Aerospace\_Engineering}(\texttt{d}) \leftarrow 0.51 \cdot \texttt{Aeronautics\_and\_Astronautics}(\texttt{d})$$

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

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 $I : Atoms \to [0, 1]$
- Interpretations are extended to formulae using norms to interpret connectives

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

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

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 $s(a, b) \leq s(a, c)$
$s(a, b) = s(b, a)$
$s(a, s(b, c)) = s(s(a, b), c)$

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, $x \Rightarrow y \equiv \neg x \vee y$

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 $\mathcal{ALC}$ concepts

Interpretation:

$$\mathcal{I} = \Delta^{\mathcal{I}}$$
$$C^{\mathcal{I}} : \Delta^{\mathcal{I}} \to [0,1]$$
$$R^{\mathcal{I}} : \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \to [0,1]$$

t = t-norm
s = s-norm
n = negation
i = implication

Concepts:

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

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

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

Inclusion axioms: $C \sqsubseteq D$,

- $\mathcal{I} \models C \sqsubseteq D$ iff $\forall x \in \Delta^{\mathcal{I}}.C^{\mathcal{I}}(x) \leq D^{\mathcal{I}}(x)$, (alternative, $\forall x \in \Delta^{\mathcal{I}}.i(C^{\mathcal{I}}(x), D^{\mathcal{I}}(x)) = 1$)

Basic Inference Problems

Consistency: Check if knowledge is meaningful

- Is $\mathcal{K}$ consistent?

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 n

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

BTVB: Best Truth Value Bound problem

- $glb(\mathcal{K}, a{:}C) = \sup\{n \mid \mathcal{K} \models \langle a{:}C, n \rangle\}$?

Retrieval: Rank set of individuals that instantiate C w.r.t. best truth value bound

- Rank the set $\mathcal{R}(\mathcal{K}, C) = \{\langle a, glb(\mathcal{K}, a{:}C) \rangle\}$

Some Notes on ...

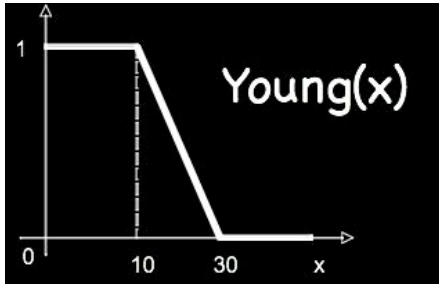
- Value restrictions:
  - In classical DLs, $\forall R.C \equiv \neg\exists R.\neg C$
  - The same is not true, in general, in fuzzy DLs (depends on the operators' semantics, not true in Gödel logic).
    $\forall \texttt{hasParent}.\texttt{Human} \not\equiv \neg\exists \texttt{hasParent}.\neg\texttt{Human}$??
- 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
- Lukasiewicz Logic seems the best compromise, though Zadeh semantics has been considered historically in DLs (Zadeh semantics is not considered by fuzzy logicians)

Towards fuzzy OWL Lite and OWL DL

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

Concrete fuzzy concepts

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



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

Modifiers

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

$$\texttt{SportsCar} \quad = \quad \texttt{Car} \sqcap \exists \texttt{speed.very}(\texttt{High})$$

Number Restrictions and Transitive roles

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

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

- 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 $(\leq n\ R)$

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

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

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

Reasoning

- For full fuzzy $\mathcal{SHOIN}(\mathtt{D})$ or $\mathcal{SHIF}(\mathtt{D})$: does not exists yet
- Exists for fuzzy $\mathcal{ALC}(\mathtt{D})$ + modifiers + fuzzy concrete concepts
  - Under Lukasiewicz semantics
  - Under "Zadeh semantics" without GCI
- Exists for $\mathcal{SHIN}$ and Zadeh semantics (classical blocking methods apply similarly in the fuzzy variant)
- On the way for GCI (both for Lukasiewicz Logic and Zadeh semantics)

Basic decision algorithm

- There are:
  - Translations of fuzzy DLs to classical DLs (not addressed here)
  - Tableau algorithms similar to classical DL tableaux
- Most problems can be reduced to consistency check, 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 consistent
    * 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 consistency check, where

$$
\begin{aligned}
t(x, y) &= \min(x, y) \\
s(x, y) &= \max(x, y) \\
n(x) &= 1 - x \\
i(x, y) &= s(n(x), y) = \max(1 - x, y)
\end{aligned}
$$

Tableaux checking consistency of an $\mathcal{ALC}$ 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 $\{\langle A, \triangleright, n\rangle, \langle C, \triangleleft, m\rangle\} \subseteq \mathcal{L}(x)$, e.g. $\langle A, \geq, 0.6\rangle, \langle A, <, 0.3\rangle$
- Returns "$\mathcal{K}$ is consistent" 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 $C \in \{C_1, C_2\}$
$x \bullet \{\langle \exists R.C, \geq, n\rangle, \ldots\}$	$\longrightarrow_{\exists}$	$x \bullet \{\langle \exists R.C, \geq, n\rangle, \ldots\}$ $\langle R, \geq, n\rangle \downarrow$ $y \bullet \{\langle C, \geq, n\rangle\}$
$x \bullet \{\langle \forall R.C, \geq, n\rangle, \ldots\}$ $\langle R, \geq, m\rangle \downarrow$ $(m > 1 - n)$ $y \bullet \{\ldots\}$	$\longrightarrow_{\forall}$	$x \bullet \{\langle \forall R.C, \geq, n\rangle, \ldots\}$ $\langle R, \geq, m\rangle \downarrow$ $y \bullet \{\ldots, \langle C, \geq, n\rangle\}$
$\vdots$	$\vdots$	$\vdots$

Soundness and Completeness

Theorem 1 *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.*

Corollary 1

1. *The tableau algorithm is a PSPACE (using depth-first search) decision procedure for consistency of $\mathcal{ALC}$ KBs.*
2. *$\mathcal{ALC}$ individuals have the tree-model property*

The tableau can be modified to a decision procedure for

- $\mathcal{SHIN}$ ($\equiv \mathcal{ALCHINR}_+$)
- TBox with acyclic concept definitions using lazy unfolding (unfolding on demand)
- For general inclusion axioms $C \sqsubseteq D$ (on the way)

Problem with fuzzy tableau

- Usual fuzzy tableaux calculus does not work anymore with
  - modifiers and concrete fuzzy concepts
  - Lukasiewicz 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

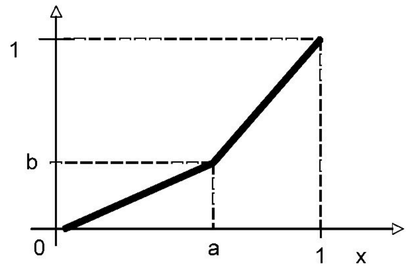
$$\bar{\mathbf{x}} \in \mathbb{Q}^k, \bar{\mathbf{y}} \in \mathbb{Z}^m$$

$$f(\bar{\mathbf{x}}, \bar{\mathbf{y}}) = \min\{f(\mathbf{x}, \mathbf{y}) \colon A\mathbf{x} + B\mathbf{y} \geq \mathbf{h}\}$$

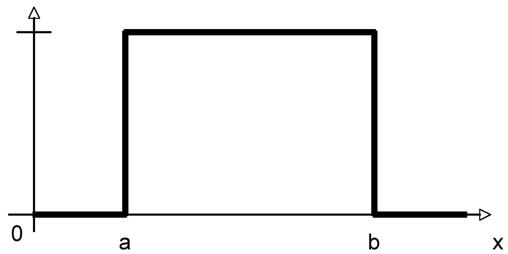
A, B integer matrixes

Requirements

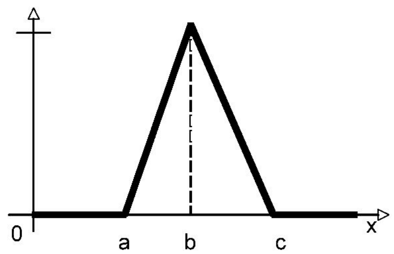
- Works for usual fuzzy DL semantics (Zadeh semantics) and Lukasiewicz logic
- Modifiers are definable as linear in-equations over $\mathbb{Q}, \mathbb{Z}$ (e.g., linear hedges), for instance, linear hedges, $lm(a, b)$, e.g. $\texttt{very} = \texttt{lm}(\texttt{0.7}, \texttt{0.49})$
- Fuzzy concrete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



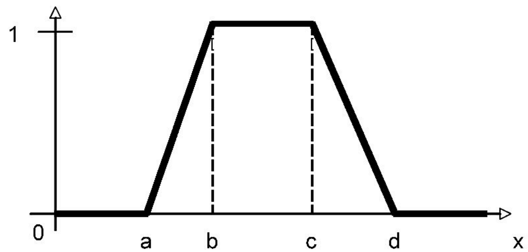
lm(a,b)



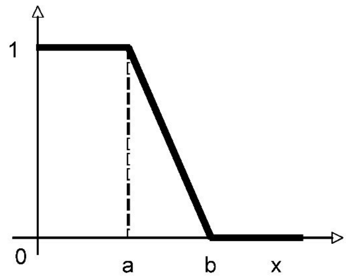
cr(a,b)



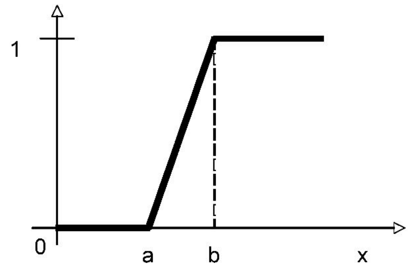
tri(a,b,c)



trz(a,b,c,d)

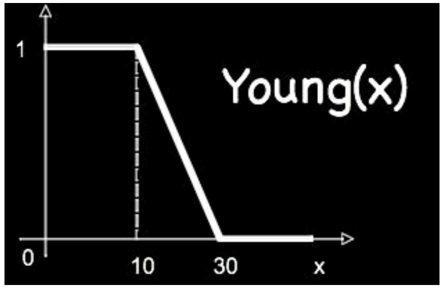


ls(a,b)



rs(a,b,c)

- Example:



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

- Then

$$\begin{aligned}
glb(\mathcal{K}, a{:}C) &= \min\{x \mid \mathcal{K} \cup \{\langle a{:}C \leq x \rangle \text{ satisfiable}\} \\
glb(\mathcal{K}, C \sqsubseteq D) &= \min\{x \mid \mathcal{K} \cup \{\langle a{:}C \sqcap \neg D \geq 1 - x \rangle \text{ satisfiable}\}
\end{aligned}$$

  - Apply tableaux calculus (without non-deterministic branches), then use bounded Mixed Integer Programming oracle

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

$x \bullet \{\langle C_1 \sqcap C_2, \geq, l\rangle, \ldots\}$	$\longrightarrow_{\sqcap}$	$x \bullet \{\langle C_1 \sqcap C_2, \geq, l\rangle, \langle C_1, \geq, l\rangle, \langle C_2, \geq, l\rangle, \ldots\}$
$x \bullet \{\langle C_1 \sqcup C_2, \geq, l\rangle, \ldots\}$	$\longrightarrow_{\sqcup}$	$x \bullet \{\langle C_1 \sqcup C_2, \geq, l\rangle, \langle C_1, \geq, x_1\rangle, \langle C_2, \geq, x_2\rangle,$ $x_1 + x_2 = l, x_1 \leq y, x_2 \leq 1 - y,$ $x_i \in [0,1], y \in \{0,1\}, \ldots\}$
$x \bullet \{\langle \exists R.C, \geq, l\rangle, \ldots\}$	$\longrightarrow_{\exists}$	$x \bullet \{\langle \exists R.C, \geq, l\rangle, \ldots\}$ $\langle R, \geq, l\rangle \downarrow$ $y \bullet \{\langle C, \geq, l\rangle\}$
$x \bullet \{\langle \forall R.C, \geq, l_1\rangle, \ldots\}$ $\langle R, \geq, l_2\rangle \downarrow$ $y \bullet \{\ldots\}$	$\longrightarrow_{\forall}$	$x \bullet \{\langle \forall R.C, \geq, l_1\rangle, \ldots\}$ $\langle R, \geq, l_2\rangle \downarrow$ $y \bullet \{\ldots, \langle C, \geq, x\rangle$ $x + y \geq l_1, x \leq y, l_1 + l_2 \leq 2 - y,$ $x \in [0,1], y \in \{0,1\}\}$
$\vdots$	$\vdots$	$\vdots$
$x \bullet \{A \sqsubseteq C, \langle A, \geq, l\rangle, \ldots\}$	$\longrightarrow_{\sqsubseteq_1}$	$x \bullet \{A \sqsubseteq C, \langle C, \geq, l\rangle, \ldots\}$
$x \bullet \{C \sqsubseteq A, \langle A, \leq, l\rangle, \ldots\}$	$\longrightarrow_{\sqsubseteq_2}$	$x \bullet \{C \sqsubseteq A, \langle C, \leq, l\rangle, \ldots\}$
$\vdots$	$\vdots$	$\vdots$

Example

- Suppose

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

$$Query \; := \; glb(\mathcal{K}, a{:}C) = \min\{x \mid \mathcal{K} \cup \{\langle a{:}C \leq x\rangle \text{ satisfiable}\}$$

Step	Tree	
1.	$a \bullet \{\langle A, \geq, 0.3\rangle, \langle B, \geq, 0.4\rangle, \langle C, \leq, x\rangle\}$	(Hypothesis)
2.	$\cup\{\langle A \sqcap B, \leq, x\rangle\}$	$(\rightarrow_{\sqsubseteq 2})$
3.	$\cup\{\langle A, \leq, x_1\rangle, \langle B, \leq, x_2\rangle\}$ $\cup\{x = x_1 + x_2 - 1, 1 - y \leq x_1, y \leq x_2\}$ $\cup\{x_i \in [0,1], y \in \{0,1\}\}$	$(\rightarrow_{\sqcap_{\leq}})$
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: $\mathbf{x = 0.3}$	

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 (see next part)
    - To be used then e.g. in the axiomatic approach to fuzzy DLPs
  - 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 $\mathcal{ALC}$ + linear hedges + concrete fuzzy concepts, using MILP, has been implemented

Future Work on fuzzy DLs

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

$$\texttt{TopCustomer} = \texttt{Customer} \sqcap (\texttt{Usually})\texttt{buys}.\texttt{ExpensiveItem}$$

$$\texttt{ExpensiveItem} = \texttt{Item} \sqcap \exists \texttt{price}.\texttt{High}$$

  - Developing a system
  - ...

Fuzzy LPs Basics

- Many Logic Programming (LP) frameworks have been proposed to manage uncertain and imprecise information. They differ in:
  - The underlying notion of uncertainty and imprecision: probability, possibility, many-valued, fuzzy sets
  - How values, associated to rules and facts, are managed
- We consider fuzzy LPs, where
  - Truth space is $[0,1]_{\mathbb{Q}}$
  - Interpretation is a mapping $I : B_{\mathcal{P}} \to [0,1]_{\mathbb{Q}}$
  - Generalized LP rules are of the form

$$f(A_1, \ldots, A_n) \Rightarrow A$$

    * A and A_i atoms and f total, monotone, finite-time computable function $f : [0,1]^n_{\mathbb{Q}} \to [0,1]_{\mathbb{Q}}$
    * Meaning of rules: take the truth-values of $A_1, \ldots A_n$, combine them using the function f, and assign the result to A

Example

$$
\begin{aligned}
\min(\quad & \texttt{Location}(\texttt{hotel}, \texttt{hotelLocation}), \\
& \texttt{Distance}(\texttt{hotelLocation}, \texttt{buisinessLocation}, \texttt{distance}), \\
& \texttt{Close}(\texttt{distance}) \\
) & \\
& \quad \Longrightarrow \quad \texttt{NearTo}(\texttt{businessLocation}, \texttt{hotel})
\end{aligned}
$$

where $\texttt{Close}(x) = max(0, 1 - x/1000)$.

Semantics of fuzzy LPs

- Model of a LP: $I \models \mathcal{P}$ iff $I \models r$, for all $r \in \mathcal{P}^*$, where
  - $I \models f(A_1, \ldots, A_n) \Rightarrow A$ iff $f(I(A_1), \ldots, I(A_n)) \leq I(A)$
- Least model exists and is least fixed-point of

$$T_{\mathcal{P}}(I)(A) = I(\varphi)$$

  for all $\varphi \Rightarrow A \in \mathcal{P}^*$

- Note: Extension to fuzzy Normal Logic Programs exists, as well as a query answering procedure. However, we will not deal with that here.

Query answering for fuzzy LPs

- Given a logic program $\mathcal{P}$, given a query atom A,
  - compute the minimal model I of $\mathcal{P}$ (bottom-up, using $T_{\mathcal{P}}$)
  - answer with $I(A)$
- Problems:
  - Least model can be very huge
  - You do not need to compute the whole least model I of $\mathcal{P}$ to answer with $I(A)$, e.g.
    - $\mathcal{P} = \{B \Rightarrow A, 1 \Rightarrow B\} \cup \mathcal{P}'$, where A does not appear in $\mathcal{P}'$

A general top-down query procedure for fuzzy LPs

- Idea: use theory of fixed-point computation of equational systems over $[0,1]_{\mathbb{Q}}$
- Assign a variable x_i to an atom $A_i \in B_{\mathcal{P}}$
- Map a rule $f(A_1, \ldots, A_n) \Rightarrow A \in \mathcal{P}^*$ into the equation $x_A = f(x_{A_1}, \ldots, x_{A_n})$
- A LP $\mathcal{P}$ is thus mapped into the equational system

$$\left\{ \begin{array}{rcl} x_1 & = & f_1(x_{1_1}, \ldots, x_{1_{a_1}}) \\ & \vdots & \\ x_n & = & f_n(x_{n_1}, \ldots, x_{n_{a_n}}) \end{array} \right.$$

- f_i is monotone and, thus, the system has least fixed-point, which is the limit of

$$\begin{array}{rcl} \mathbf{y}_0 & = & \mathbf{0} \\ \mathbf{y}_{i+1} & = & \mathbf{f}(\mathbf{y}_i)\ . \end{array}$$

  where $\mathbf{f} = \langle f_1, \ldots, f_n \rangle$ and $\mathbf{f}(\mathbf{x}) = \langle f_1(x_1), \ldots, f_n(x_n) \rangle$
- The least-fixed point is the least model of $\mathcal{P}$
- Consequence: If top-down procedure exists for equational systems then it works for fuzzy LPs too!

Procedure $Solve(\mathcal{S}, Q)$

Input: monotonic system $\mathcal{S} = \langle \mathcal{L}, V, \mathbf{f} \rangle$, where $Q \subseteq V$ is the set of query variables;

Output: A set $B \subseteq V$, with $Q \subseteq B$ such that the mapping $\mathtt{v}$ equals $\mathrm{lfp}(f)$ on B.

$\mathtt{A}\colon = Q$, $\mathtt{dg}\colon = Q$, $\mathtt{in}\colon = \emptyset$, **for all** $x \in V$ **do** $\mathtt{v}(x) = 0$, $\mathtt{exp}(x) = \mathtt{0}$

while $\mathtt{A} \neq \emptyset$ **do**

select $x_i \in \mathtt{A}$, $\mathtt{A}\colon = \mathtt{A} \setminus \{x_i\}$, $\mathtt{dg}\colon = \mathtt{dg} \cup \mathtt{s}(x_i)$

$r\colon = f_i(\mathtt{v}(x_{i_1}), ..., \mathtt{v}(x_{i_{a_i}}))$

if $r \succ \mathtt{v}(x_i)$ **then** $\mathtt{v}(x_i)\colon = r$, $\mathtt{A}\colon = \mathtt{A} \cup (\mathtt{p}(x_i) \cap \mathtt{dg})$ **fi**

if not $\mathtt{exp}(x_i)$ **then** $\mathtt{exp}(x_i) = \mathtt{1}$, $\mathtt{A}\colon = \mathtt{A} \cup (\mathtt{s}(x_i) \setminus \mathtt{in})$, $\mathtt{in}\colon = \mathtt{in} \cup \mathtt{s}(x_i)$ **fi**

od

- Set of facts $0.7 \Rightarrow \texttt{Experience(john)}$, $0.5 \Rightarrow \texttt{Risk(john)}$, $0.8 \Rightarrow \texttt{Sport\_car(john)}$
- Set of rules, which after grounding are:

$$\begin{array}{lcl}
\texttt{Experience(john)} \wedge (0.5 \cdot \texttt{Risk(john)}) & \Rightarrow & \texttt{Good_driver(john)} \\
0.8 \cdot \texttt{Young(john)} & \Rightarrow & \texttt{Risk(john)} \\
0.8 \cdot \texttt{Sport_car(john)} & \Rightarrow & \texttt{Risk(john)} \\
\texttt{Experience(john)} \wedge (0.5 \cdot \texttt{Good_driver(john)}) & \Rightarrow & \texttt{Risk(john)}
\end{array}$$

1. $\texttt{A}:=\{x_{\texttt{R(j)}}\}, x_i:=x_{\texttt{R(j)}}, \texttt{A}:=\emptyset, \texttt{dg}:=\{x_{\texttt{R(j)}}, x_{\texttt{Y(j)}}, x_{\texttt{S(j)}}, x_{\texttt{E(j)}}, x_{\texttt{G(j)}}\}, r:=0.5, \texttt{v}(x_{\texttt{R(j)}}):=0.5,$
   $\texttt{A}:=\{x_{\texttt{G(j)}}\}, \texttt{exp}(x_{\texttt{R(j)}}):=\texttt{1}, \texttt{A}:=\{x_{\texttt{Y(j)}}, x_{\texttt{S(j)}}, x_{\texttt{E(j)}}, x_{\texttt{G(j)}}\}, \texttt{in}:=\{x_{\texttt{Y(j)}}, x_{\texttt{S(j)}}, x_{\texttt{E(j)}}, x_{\texttt{G(j)}}\}$
2. $x_i:=x_{\texttt{Y(j)}}, \texttt{A}:=\{x_{\texttt{S(j)}}, x_{\texttt{E(j)}}, x_{\texttt{G(j)}}\}, r:=0, \texttt{exp}(x_{\texttt{Y(j)}}):=\texttt{1}$
3. $x_i:=x_{\texttt{S(j)}}, \texttt{A}:=\{x_{\texttt{E(j)}}, x_{\texttt{G(j)}}\}, r:=0.8, \texttt{v}(x_{\texttt{S(j)}}):=0.8, \texttt{A}:=\{x_{\texttt{E(j)}}, x_{\texttt{G(j)}}, x_{\texttt{R(j)}}\}, \texttt{exp}(x_{\texttt{S(j)}}):=\texttt{1}$
4. $x_i:=x_{\texttt{E(j)}}, \texttt{A}:=\{x_{\texttt{G(j)}}, x_{\texttt{R(j)}}\}, r:=0.7, \texttt{v}(x_{\texttt{E(j)}}):=0.7, \texttt{exp}(x_{\texttt{E(j)}}):=\texttt{1}$
5. $x_i:=x_{\texttt{G(j)}}, \texttt{A}:=\{x_{\texttt{R(j)}}\}, r:=0.25, \texttt{v}(x_{\texttt{G(j)}}):=0.25, \texttt{exp}(x_{\texttt{G(j)}}):=\texttt{1},$
   $\texttt{in}:=\{x_{\texttt{Y(j)}}, x_{\texttt{S(j)}}, x_{\texttt{E(j)}}, x_{\texttt{G(j)}}, x_{\texttt{R(j)}}\}$
6. $x_i:=x_{\texttt{R(j)}}, \texttt{A}:=\emptyset, r:=0.64, \texttt{v}(x_{\texttt{R(j)}}):=0.64, \texttt{A}:=\{x_{\texttt{G(j)}}\}$
7. $x_i:=x_{\texttt{G(j)}}, \texttt{A}:=\emptyset, r:=0.32, \texttt{v}(x_{\texttt{G(j)}}):=0.32, \texttt{A}:=\{x_{\texttt{R(j)}}\}$
8. $x_i:=x_{\texttt{G(j)}}, \texttt{A}:=\emptyset, r:=0.64$

10. **stop. return v** (in particular, $\texttt{v}(x_{\texttt{R(j)}})=0.64$)

Future Work on fuzzy LPs

- Research directions:
  - Developing a system for fuzzy LPs (i.e. implement the top-down algorithm, e.g. use `lparse` for grounding)
  - Mapping between fuzzy OWL Lite and fuzzy LPs (I guess they are in the same complexity class)
    - Problem: membership functions of concrete concepts are not necessarily monotone
    - A MILP oracle in fuzzy LPs may be needed
  - More general equations: from $x = f(x_1, ..., x_n)$ to e.g.

$$x_{i1} \vee \ldots \vee x_{ik} = f(x_1, ..., x_n)$$

    to accommodate disjunctive fuzzy LPs
  - Mapping between fuzzy OWL DL and fuzzy disjunctive LPs

Fuzzy DLPs Basics

- Combine fuzzy DLs with fuzzy LPs:
  - DL atoms and roles may appear in rules

$$\min(\texttt{made\_by}(x, y), \langle\texttt{ChineseCarCompany}\rangle(y)), \texttt{prize}(x, z) \Rightarrow \texttt{LowCarPrize}(z)$$

$$\texttt{LowCarPrize}(z) = \texttt{ls}(5.000, 15.000)$$

$$\texttt{ChineseCarCompany} \sqsubseteq \exists \texttt{has\_location}.\texttt{China}$$

- Knowledge Base is a pair $KB = \langle \mathcal{P}, \Sigma \rangle$, where
  - $\mathcal{P}$ is a fuzzy logic program
  - Σ is a fuzzy DL knowledge base (set of assertions and inclusion axioms)

Fuzzy DLPs Semantics

- Semantics: two main approaches

  1. Axiomatic approach: fuzzy DL atoms and roles are managed uniformely
     - I is a model of $KB = \langle \mathcal{P}, \Sigma \rangle$ iff $I \models \mathcal{P}$ and $I \models \Sigma$
  2. DL-log approach: fuzzy DL atoms and roles are procedural attachments (calls to a fuzzy DL theorem prover)
     - I is a model of $KB = \langle \mathcal{P}, \Sigma \rangle$ iff $I^{\Sigma} \models \mathcal{P}$
     - $I^{\Sigma}(A) = I(A)$ for all ground non-DL atoms A
     - $I^{\Sigma}(\langle A \rangle(a)) = glb(\Sigma, a{:}A)$ for all ground DL atoms $\langle A \rangle(a)$
     - $I^{\Sigma}(\langle R \rangle(a, b)) = glb(\Sigma, (a, b){:}R)$ for all ground DL roles $\langle R \rangle(a, b)$

- DL-log has the minimal model property of fuzzy LPs and a fixed-point characterization: $T_{\mathcal{P}}(I)(A) = I^{\Sigma}(\varphi)$, for $\varphi \Rightarrow A \in \mathcal{P}^*$

A top-down procedure for the DL-log approach

- Combine $Solve(\mathcal{S}, Q)$ with a theorem prover for fuzzy DLs
  - Modify Step 1. of algorithm $Solve(\mathcal{S}, Q)$
    - for all x_{i_j} DL-atoms $\langle A \rangle(a)$ (similarly for roles)
      - compute $\bar{x}_{i_j} = glb(\mathcal{K}, a{:}A)$
      - set $\mathtt{v}(x_{i_j}) = \bar{x}_{i_j}$, instead of $\mathtt{v}(x_{i_j}) = 0$
- Essentially, for all DL-atoms $\langle A \rangle(a)$ we compute off-line $glb(\mathcal{K}, a{:}A)$ and add then the rule $A(a) \leftarrow glb(\mathcal{K}, a{:}A)$ to $\mathcal{P}$
- A solution for the axiomatic approach is not known yet

Conclusions

- Fuzzy DLs, fuzzy LPs and fuzzy DLPs allow to deal with imprecise concepts
  - Formulae have a degree of truth
  - Explicit membership functions are allowed
- We shown some applications of these languages and reasoning procedures