Managing Uncertainty and Vagueness in Semantic Web Languages

Tutorial at AAAI-2007

Thomas Lukasiewicz [1 2] Umberto Straccia [3]

[1] DIS, Sapienza Università di Roma, Italy
lukasiewicz@dis.uniroma1.it

[2] Institut für Informationssysteme, TU Wien, Austria
lukasiewicz@kr.tuwien.ac.at

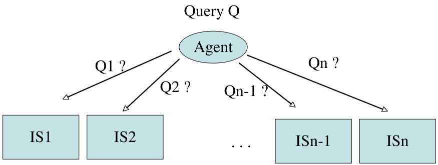
[3] ISTI-CNR, Pisa, Italy
straccia@isti.cnr.it

Sources of Uncertainty and Vagueness on the Web

- Information Retrieval:
  - To which degree is a Web site, a Web page, a text passage, an image region, a video segment, . . . relevant to my information need?
- Matchmaking
  - To which degree does an object match my requirements?
    - if I'm looking for a car and my budget is *about* 20.000 €, to which degree does a car's price of 20.500 € match my budget?

- Semantic annotation
  - To which degree does e.g., an image object represent a dog?
- Information extraction
  - To which degree am I'm sure that e.g., SW is an acronym of "Semantic Web"?
- Ontology alignment (schema mapping)
  - To which degree do two concepts of two ontologies represent the same, or are disjoint, or are overlapping?
- Representation of background knowledge
  - To some degree birds fly.
  - To some degree Jim is a blond and young.

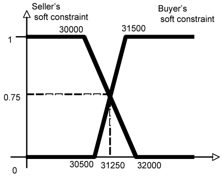
Example (Distributed Information Retrieval) [7]



Then the agent has to perform automatically the following steps:

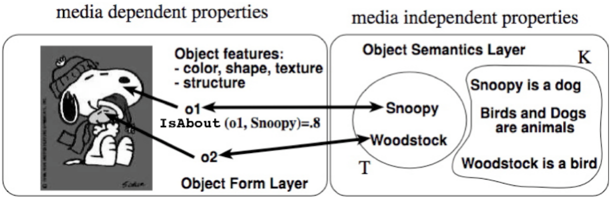
1. The agent has to select a subset of relevant resources $\mathscr{S}' \subseteq \mathscr{S}$, as it is not reasonable to assume to access to and query all resources (resource selection/resource discovery);
2. For every selected source $\mathcal{S}_i \in \mathscr{S}'$ the agent has to reformulate its 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)

Example (Negotiation) [2]



- 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 vague constraints (fuzzy sets) (as usual in negotiation)
  - Seller would sell above 31500 €, but can go down to 30500 €
  - The buyer prefers to spend less than 30000 €, but can go up to 32000 €
  - Highest degree of matching is 0.75 . The car may be sold at 31250 €.

Example (Logic-based information retrieval model)[1, 8]



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

"Find top-*k* image regions about animals"

$$Query(x) \leftarrow ImageRegion(x) \wedge isAbout(x, y) \wedge Animal(y)$$

Example (Database query) [3, 4, 5, 6]

HotelID	*hasLoc*
*h*1	*hl*1
*h*2	*hl*2
⋮	⋮

ConferenceID	*hasLoc*
*c*1	*cl*1
*c*2	*cl*2
⋮	⋮

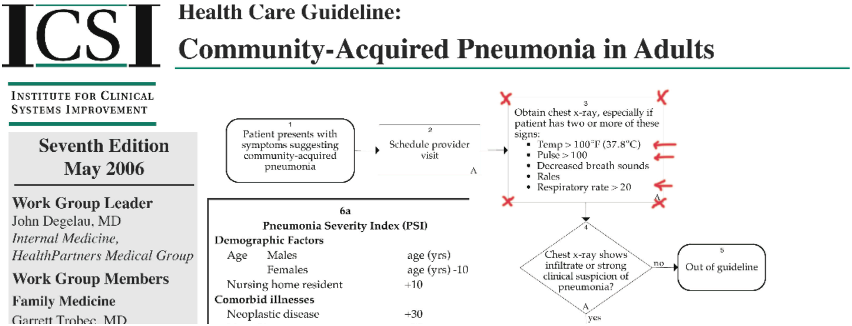
hasLoc	*hasLoc*	*distance*
*hl*1	*cl*1	300
*hl*1	*cl*2	500
*hl*2	*cl*1	750
*hl*2	*cl*2	800
⋮	⋮	

hasLoc	*hasLoc*	*close*	*cheap*
*hl*1	*cl*1	0.7	0.3
*hl*1	*cl*2	0.5	0.5
*hl*2	*cl*1	0.25	0.8
*hl*2	*cl*2	0.2	0.9
⋮	⋮	⋮	

"Find top-*k* cheapest hotels close to the train station"

$$q(h) \leftarrow hasLocation(h, hl) \wedge hasLocation(\texttt{train}, cl) \wedge close(hl, cl) \wedge cheap(h)$$

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 imprecise than crisp

Uncertainty vs. Vagueness: a clarification

- What does the degree mean?
- There is often a misunderstanding between interpreting a degree as a measure of uncertainty or as a measure of vagueness
- The value 0.83 has a different interpretation in “Birds fly to degree 0.83” from that in “Hotel Verdi is close to the train station to degree 0.83”

Uncertainty

- Uncertainty: statements are true or false. But, due to lack of knowledge we can only estimate to which probability/possibility/necessity degree they are true or false
  - For instance, a bird flies or does not fly. The probability/possibility/necessity degree that it flies is 0.83
- Usually we have a possible world semantics with a distribution over possible worlds:

$$W = \{I \text{ classical interpretation}\}, \quad I(\varphi) \in \{0,1\}$$

$$\mu \colon W \to [0,1], \quad \mu(I) \in [0,1]$$

$$Pr(\phi) = \sum_{I \models \phi} \mu(I)$$

$$Poss(\phi) = \sup_{I \models \phi} \mu(I)$$

$$Necc(\phi) = \inf_{I \not\models \phi} \mu(I) = 1 - Poss(\neg\phi)$$

Vagueness

- Vagueness: statements involve concepts for which there is no exact definition, such as tall, small, close, far, cheap, expensive, isAbout, similarTo. Statements are true to some degree which is taken from a truth space.
  - E.g., “Hotel Verdi is close to the train station to degree 0.83”
- Truth space: set of truth values L and an partial order $\leq$
- Many-valued Interpretation: a function I mapping formulae into L, i.e. $I(\varphi) \in L$
- Fuzzy Logic: $L = [0, 1]$
- Uncertainty and Vagueness: “It is possible/probable to degree 0.83 that it will be hot tomorrow”
- The notion of imperfect information covers concepts such as uncertainty, vagueness, contradiction, incompleteness, imprecision.

C. Meghini, F. Sebastiani, and U. Straccia.
A model of multimedia information retrieval.
Journal of the ACM, 48(5):909–970, 2001.

A. Ragone, U. Straccia, T. D. Noia, E. D. Sciascio, and F. M. Donini.
Vague knowledge bases for matchmaking in p2p e-marketplaces.
In *4th European Semantic Web Conference (ESWC-07)*, number 4519 in Lecture Notes in Computer Science, pages 414—428. Springer Verlag, 2007.

U. Straccia.
Answering vague queries in fuzzy dl-lite.
In *Proceedings of the 11th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, (IPMU-06)*, pages 2238–2245. E.D.K., Paris, 2006.

U. Straccia.
Towards top-k query answering in deductive databases.
In *Proceedings of the 2006 IEEE International Conference on Systems, Man and Cybernetics (SMC-06)*, pages 4873–4879. IEEE, 2006.

U. Straccia.
Towards top-k query answering in description logics: the case of DL-Lite.
In *Proceedings of the 10th European Conference on Logics in Artificial Intelligence (JELIA-06)*, number 4160 in Lecture Notes in Computer Science, pages 439–451, Liverpool, UK, 2006. Springer Verlag.

U. Straccia.
Towards vague query answering in logic programming for logic-based information retrieval.
In *World Congress of the International Fuzzy Systems Association (IFSA-07)*, Cancun, Mexico, 2007.

U. Straccia and R. Troncy.

Towards distributed information retrieval in the semantic web.
In *3rd European Semantic Web Conference (ESWC-06)*, number 4011 in Lecture Notes in Computer Science, pages 378–392. Springer Verlag, 2006.

U. Straccia and G. Visco.
DLMedia: an ontology mediated multimedia information retrieval system.
In *Proceedings of the International Workshop on Description Logics (DL-07)*, Innsbruck, Austria, 2007. CEUR.

Web Ontology Languages

- Wide variety of languages for "Explicit Specification"
  - Graphical notations
    - Semantic networks
    - UML
    - RDF/RDFS
  - Logic based
    - Description Logics (e.g., OIL, DAML+OIL, OWL, OWL-DL, OWL-Lite)
    - Rules (e.g., RuleML, RIF, SWRL, LP/Prolog)
    - First Order Logic (e.g., KIF)
- RDF and OWL-DL are the major players (so far ...)

RDF

- Statements are of the form

$$\langle \textit{subject}, \textit{predicate}, \textit{object} \rangle$$

  called triples: e.g.

$$\langle \textit{umberto}, \textit{plays}, \textit{soccer} \rangle$$

- can be represented graphically as:

$$\boxed{\text{umberto}} \xrightarrow{\textit{plays}} \boxed{\text{soccer}}$$

- Statements describe properties of resources
- A resource is any object that can be pointed to by a URI (Universal Resource Identifier):

RDF Schema (RDFS)

- RDF Schema allows you to define vocabulary terms and the relations between those terms
- RDF Schema terms (just a few examples):
  - Class
  - Property
  - type
  - subClassOf
  - range
  - domain
- These terms are the RDF Schema building blocks (constructors) used to create vocabularies:

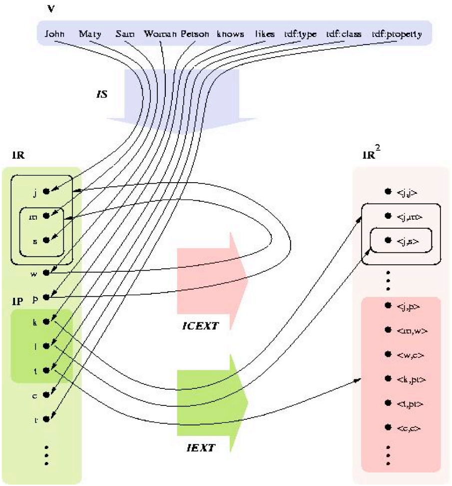
```
<Person,type, Class>
<hasColleague, type, Property>
<Professor, subClassOf,Person>
<Carole, type,Professor>
<hasColleague, range,Person>
<hasColleague, domain,Person>
```

RDF/RDFS Semantics

- RDF has “Non-standard” semantics in order to deal with this
- Semantics given by RDF Model Theory (MT)
- In RDF MT, an interpretation I of a vocabulary V consists of:
  - IR, a non-empty set of resources, called the domain of I.
  - IS, a mapping from URI references in V into IR
  - IP, a distinguished subset of IR (the set of properties of I)
    - A vocabulary element $v \in V$ is a property iff $IS(v) \in IP$
  - $IEXT$, a mapping from IP into the powerset of $IR \times IR$, $IEXT(x)$ is called the extension of x
    - I.e., a set of elements $\langle x, y \rangle$, with x, y elements of IR
    - I.e., is a set of pairs which identify the arguments for which the property is true
    - This trick of distinguishing a relation as an object from its relational extension allows a property to occur in its own extension
  - IL, a mapping from typed literals in V into IR
  - A distinguished subset LV of IR, called the set of literal values, which contains all the plain literals in V
- Class interpretation $ICEXT$ simply induced by $IEXT(IS(type))$
  - $ICEXT(C) = \{x \mid \langle x, C \rangle \in IEXT(IS(type))\}$

(`http://www.w3.org/TR/rdf-mt/`)

Example RDF/RDFS Interpretation



RDFS Interpretations

- RDFS adds extra constraints on interpretations
  - E.g., interpretations of $\langle C, subClassOf, D\rangle$ constrained to those where $ICEXT(IS(C)) \subseteq ICEXT(IS(D))$
- Can deal with triples such as

```
<Species,type,Class>
<Lion,type,Species>
<Leo,type,Lion>

<SelfInst,type,SelfInst>
```

- And even with triples such as

```
<type,subPropertyOf,subClassOf>
```

- But not clear if meaning matches intuition (if there is one)

OWL [10]

- Three species of OWL
  - OWL full is union of OWL syntax and RDF (Undecidable)
  - OWL DL restricted to FOL fragment (decidable in NEXPTIME)
  - OWL Lite is “easier to implement” subset of OWL DL (decidable in EXPTIME)
- Semantic layering
  - OWL DL within Description Logic (DL) fragment
- OWL DL based on $\mathcal{SHOIN}(D_n)$ DL
- OWL Lite based on $\mathcal{SHIF}(D_n)$ DL

Description Logics (DLs)

- The logics behind OWL-DL and OWL-Lite, `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	$\to$	$\top$	|	$\top(x)$	
		$\bot$	|	$\bot(x)$	
		A	|	$A(x)$	$Human$
		$C \sqcap D$	|	$C(x) \land D(x)$	$Human \sqcap Male$
		$C \sqcup D$	|	$C(x) \lor D(x)$	$Nice \sqcup Rich$
		$\neg C$	|	$\neg C(x)$	$\neg Meat$
		$\exists R.C$	|	$\exists y.R(x,y) \land C(y)$	$\exists has\_child.Blond$
		$\forall R.C$		$\forall y.R(x,y) \Rightarrow C(y)$	$\forall has\_child.Human$
$C \sqsubseteq D$				$\forall x.C(x) \Rightarrow D(x)$	$Happy\_Father \sqsubseteq Man \sqcap \exists has\_child.Female$
$a{:}C$				$C(a)$	$John{:}Happy\_Father$

Toy Example

$$\begin{array}{rcl} \mathit{Sex} & = & \mathit{Male} \sqcup \mathit{Female} \\ \mathit{Male} \sqcap \mathit{Female} & \sqsubseteq & \bot \\ \mathit{Person} & \sqsubseteq & \mathit{Human} \sqcap \exists \mathit{hasSex}.\mathit{Sex} \\ \mathit{MalePerson} & \sqsubseteq & \mathit{Person} \sqcap \exists \mathit{hasSex}.\mathit{Male} \end{array}$$

$$\mathit{umberto}{:}\mathit{Person} \sqcap \exists \mathit{hasSex}.\neg \mathit{Female}$$

$$\mathit{KB} \models \mathit{umberto}{:}\mathit{MalePerson}$$

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,

$$\mathcal{SHIF} = \mathcal{S} + \mathcal{H} + \mathcal{I} + \mathcal{F} = \mathcal{ALCR_+HIF} \quad \text{OWL-Lite (EXPTIME)}$$
$$\mathcal{SHOIN} = \mathcal{S} + \mathcal{H} + \mathcal{O} + \mathcal{I} + \mathcal{N} = \mathcal{ALCR_+HOIN} \quad \text{OWL-DL (NEXPTIME)}$$

Semantics of Additional Constructs

- $\mathcal{H}$: Role inclusion axioms, $\mathcal{I} \models R_1 \sqsubseteq R_2$ iff ${R_1}^{\mathcal{I}} \subseteq {R_1}^{\mathcal{I}}$
- $\mathcal{N}$: Number restrictions,
  $(\geq n\ R)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} : |\{y \mid \langle x, y \rangle \in R^{\mathcal{I}}\}| \geq n\}$,
  $(\leq n\ R)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} : |\{y \mid \langle x, y \rangle \in R^{\mathcal{I}}\}| \leq n\}$
- $\mathcal{Q}$: Qualified number restrictions,
  $(\geq n\ R.C)^{\mathcal{I}} = \{x \in |\{y \mid \langle x, y \rangle \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}| \geq n\}$,
  $(\leq n\ R.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} : |\{y \mid \langle x, y \rangle \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}| \leq n\}$
- $\mathcal{O}$: Nominals (singleton class), $\{a\}^{\mathcal{I}} = \{a^{\mathcal{I}}\}$
- $\mathcal{I}$: Inverse role, $(R^{-})^{\mathcal{I}} = \{\langle x, y \rangle \mid \langle y, x \rangle \in R^{\mathcal{I}}\}$
- $\mathcal{F}$: Functional role, $I \models fun(f)$ iff $\forall z \forall y \forall z$ if $\langle x, y \rangle \in f^{\mathcal{I}}$ and $\langle x, z \rangle \in f^{\mathcal{I}}$ the $y = z$
- $\mathcal{R}_{+}$: transitive role,
  $(R_{+})^{\mathcal{I}} = \{\langle x, y \rangle \mid \exists z \text{ such that } \langle x, z \rangle \in R^{\mathcal{I}} \wedge \langle z, y \rangle \in R^{\mathcal{I}}\}$

Concrete Domains

- Concrete domains: reals, integers, strings, ...

  $(\mathit{tim}, 14){:}\mathit{hasAge}$
  $(\mathit{sf}, \text{``}\mathit{SoftComputing}\text{''}){:}\mathit{hasAcronym}$
  $(\mathit{source}1, \text{``}\mathit{ComputerScience}\text{''}){:}\mathit{isAbout}$
  $(\mathit{service}2, \text{``}\mathit{InformationRetrievalTool}''){:}\mathit{Matches}$
  $\mathit{Minor} = \mathit{Person} \sqcap \exists \mathit{hasAge}. \leq_{18}$

- Semantics: a 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 $d^D \subseteq \Delta_D^n$
  - Concrete properties: $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta_D$
- Notation: (D). E.g., $\mathcal{ALC}(D)$ is $\mathcal{ALC}$ + concrete domains

OWL DL as Description Logic

Concept/Class constructors:

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))	$\exists R.\{o\}$	$\exists$`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))	$\exists U. =_v\}$	$\exists$`series.`$=_{\text{"LNCS"}}$
`restriction`(U `minCardinality`(n))	$(\geq n\ U)$	($\geqslant$ 1 `title`)
`restriction`(U `maxCardinality`(n))	$(\leq n\ U)$	($\leqslant$ 1 `author`)

Note: R is an abstract role, while U is a concrete property of arity two.

Axioms:

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`($C_1 \ldots C_n$)	$C_i \sqcap C_j = \bot, i \neq j$	$Male \sqcap Female \sqsubseteq \bot$
`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)	$\top \sqsubseteq \forall R.C_i$	$\top \sqsubseteq \forall hasChild.Human$
[`inverseof`(P)]	$R = P^-$	$hasChild = hasParent^-$
[`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`$(C_1)\ldots$`domain`(C_n)	$(\geq 1\ U) \sqsubseteq C_i$	$(\geq 1\ hasAge) \sqsubseteq Human$
`range`$(D_1)\ldots$`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`$(R_1 o_1)\ldots$`value`$(R_n o_n)$	$(o, o_i){:}R_i$	$(tim, mary){:}hasChild$
`value`$(U_1 v_1)\ldots$`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`$(o_1 \ldots o_n)$	$o_i \neq o_j, i \neq j$	$john \neq peter$
Symbols		
Object Property R (URI reference)	R	hasChild
Datatype Property U (URI reference)	U	hasAge
Individual o (URI reference)	U	tim
Data Value v (RDF literal)	U	"ESWC07"

LPs Basics (for ease, without default negation) [6]

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

$$P(\mathbf{x}) \leftarrow \varphi(\mathbf{x}, \mathbf{y})$$

where $\varphi(\mathbf{x}, \mathbf{y})$ is a formula built from atoms of the form $B(\mathbf{z})$ and connectors $\wedge, \vee$
For instance,

$$has\_father(x, y) \quad \leftarrow \quad has\_parent(x, y) \wedge Male(y)$$

- Facts are rules with empty body
For instance,

$$has\_parent(mary, jo)$$

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 $A \leftarrow 0$ to $\mathcal{P}^*$;
  3. replace several rules in $\mathcal{P}^*$ having same head

$$\left.\begin{array}{c} A \leftarrow \varphi_1 \\ A \leftarrow \varphi_2 \\ \vdots \\ A \leftarrow \varphi_n \end{array}\right\} \text{ with } A \leftarrow \varphi_1 \lor \varphi_2 \lor \ldots \lor \varphi_n \,.$$

- Note: in $\mathcal{P}^*$ each atom $A \in B_{\mathcal{P}}$ is head of exactly 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 $I \models A \leftarrow \varphi$ iff $I(\varphi) \leq I(A)$
- Least model exists and is least fixed-point of

$$T_{\mathcal{P}}(I)(A) = I(\varphi), \text{ for all } A \leftarrow \varphi \in \mathcal{P}^*$$

Toy Example

$$
\begin{array}{rcl}
Q(x) & \leftarrow & B(x) \\
Q(x) & \leftarrow & C(x) \\
B(a) & \leftarrow & \\
C(b) & \leftarrow &
\end{array}
$$

$$KB \models Q(a) \quad KB \models Q(b) \qquad answers(KB, Q) = \{a, b\}$$

where $answers(KB, Q) = \{\mathbf{c} \mid KB \models Q(\mathbf{c})\}$

DLPs Basics

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

$$\begin{array}{lcl} \mathit{buy}(x) & \leftarrow & \mathit{Electronics}(x), \mathit{offer}(x) \\ \mathit{Camera} & \sqsubseteq & \mathit{Electronics} \end{array}$$

- 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)
- Many different approaches exists with different semantics: we present the basics of two of them

Loosely Coupled DL-Programs [3, 4, 5]

- A dl-query $Q(\mathbf{t})$ is of the form:
  - $C(t)$, with a concept C and a term t;
  - $R(t_1, t_2)$, with a role R and terms t_1, t_2.
- A dl-rule r is of form

$$a \leftarrow b_1, \ldots, b_k$$

where any $b \in Body(r)$ may be a dl-atom $DL[Q](\mathbf{t})$

$$\begin{array}{rcl} buy(x) & \leftarrow & DL[Electronics](x), offer(x) \\ Camera & \sqsubseteq & Electronics \end{array}$$

- Note: [3, 4, 5] considers more expressive dl-queries, non-monotone negation and disjunctive LPs

Semantics

- DL atoms and roles are "procedural attachments" (calls to a DL theorem prover)
  - I is a model of $KB = \langle L, P \rangle$ iff $I^L \models P$
  - I^L is a model of a ground non-DL atom $A \in B_{\mathcal{P}}$ iff $I(A) = 1$
  - I^L is a model of a ground DL atom $DL[C](a)$ iff $L \models a{:}C$
  - I^L is a model of a ground DL role $DL[R](a, b)$ iff $L \models (a, b){:}R$
- Minimal model exists and fixed-point characterization:

$$T_{\mathcal{P}}(I)(A) = I^L(\varphi), \text{ for all } A \leftarrow \varphi \in \mathcal{P}^*$$

- Example:

$$\begin{array}{lcl} buy(x) & \leftarrow & DL[Camera](x) \\ buy(x) & \leftarrow & DL[DVDPlayer](x) \end{array}$$

$$a{:}Camera \quad b{:}Camera \sqcup DVDPlayer$$

$$KB \models buy(a) \quad KB \not\models buy(b)$$

Tightly Coupled DL-Programs [7]

- A dl-atom may appear anywhere in the rule (rule head and/or rule body)
- $I \models P$ is defined as usual.
- $I \models L$ iff $L \cup \{a \mid I(a) = 1\} \cup \{\neg a \mid I(a) = 0\}$ is satisfiable.
- $I \models KB$ iff $I \models L$ and $I \models P$.
- Many minimal models may exists.
- $KB \models_{cautious} a$ iff for all minimal models I of KB, $I \models a$
- $KB \models_{brave} a$ iff for some minimal models I of KB, $I \models a$
- Clearly, $\models_{cautious} \subseteq \models_{brave}$
- Example:
$$buy(x) \leftarrow DL[Camera](x)$$
$$buy(x) \leftarrow DL[DVDPlayer](x)$$
$$a{:}Camera \quad b{:}Camera \sqcup DVDPlayer$$
$$KB \models_{cautious} buy(a) \quad KB \not\models_{cautious} buy(b)$$
- Note: [7] considers non-monotone negation and disjunctive LPs

C. V. Damásio, J. Z. Pan, G. Stoilos, and U. Straccia.
An approach to representing uncertainty rules in ruleml.
In *Second International Conference on Rules and Rule Markup Languages for the Semantic Web (RuleML-06)*, pages 97–106. IEEE, 2006.

C. V. Damasio, J. Z. Pan, G. Stoilos, and U. Straccia.
Representing uncertainty rules in ruleml.
Fundamenta Informaticae, 2007.

T. Eiter, T. Lukasiewicz, R. Schindlauer, and H. Tompits.
Combining answer set programming with description logics for the Semantic Web.
In *Proceedings KR-2004*, pages 141–151. AAAI Press, 2004.

T. Eiter, T. Lukasiewicz, R. Schindlauer, and H. Tompits.
Well-founded semantics for description logic programs in the Semantic Web.
In *Proceedings RuleML-2004*, volume 3323 of *LNCS*, pages 81–97. Springer, 2004.

T. Eiter, G. Ianni, R. Schindlauer, and H. Tompits.
Effective integration of declarative rules with external evaluations for Semantic Web reasoning.
In *Proceedings ESWC-2006*, volume 4011 of *LNCS*, pages 273–287. Springer, 2006.

J. W. Lloyd.
Foundations of Logic Programming.
Springer, Heidelberg, RG, 1987.

T. Lukasiewicz.
A novel combination of answer set programming with description logics for the Semantic Web.
In *Proceedings ESWC-2007*, volume 4519 of *LNCS*, pages 384–398. Springer, 2007.

A. Ragone, U. Straccia, T. D. Noia, E. D. Sciascio, and F. M. Donini.
Vague knowledge bases for matchmaking in p2p e-marketplaces.
In *4th European Semantic Web Conference (ESWC-07)*, number 4519 in Lecture Notes in Computer Science, pages 414—428. Springer Verlag, 2007.

U. Straccia.
Towards top-k query answering in deductive databases.
In *Proceedings of the 2006 IEEE International Conference on Systems, Man and Cybernetics (SMC-06)*, pages 4873–4879. IEEE, 2006.

W3C.
OWL web ontology language overview, 2004.
W3C Recommendation (10 Feb. 2004). Available at
http://www.w3.org/TR/2004/REC-owl-features-20040210/.

Probabilistic Logic

- Integration of (propositional) logic- and probability-based representation and reasoning formalisms.
- Reasoning from logical constraints and interval restrictions for conditional probabilities (also called *conditional constraints*).
- Reasoning from convex sets of probability distributions.
- Model-theoretic notion of logical entailment.

Syntax of Probabilistic Knowledge Bases

- Finite nonempty set of basic events $\Phi = \{p_1, \ldots, p_n\}$.
- Event ϕ: Boolean combination of basic events
- Logical constraint $\psi \Leftarrow \phi$: events ψ and ϕ: "ϕ implies ψ".
- Conditional constraint $(\psi|\phi)[l, u]$: events ψ and ϕ, and $l, u \in [0, 1]$: "conditional probability of ψ given ϕ is in $[l, u]$".
- Probabilistic knowledge base $KB = (L, P)$:
  - finite set of logical constraints L,
  - finite set of conditional constraints P.

Example

Probabilistic knowledge base $KB = (L, P)$:

- $L = \{bird \Leftarrow eagle\}$:

  "All eagles are birds".

- $P = \{(have\_legs \mid bird)[1, 1], (fly \mid bird)[0.95, 1]\}$:

  "All birds have legs".

  "Birds fly with a probability of at least 0.95".

Semantics of Probabilistic Knowledge Bases

- World I: truth assignment to all basic events in Φ.
- $\mathcal{I}_\Phi$: all worlds for Φ.
- Probabilistic interpretation Pr: probability function on $\mathcal{I}_\Phi$.
- $Pr(\phi)$: sum of all $Pr(I)$ such that $I \in \mathcal{I}_\Phi$ and $I \models \phi$.
- $Pr(\psi|\phi)$: if $Pr(\phi) > 0$, then $Pr(\psi|\phi) = Pr(\psi \wedge \phi) \,/\, Pr(\phi)$.
- Truth under Pr:
  - $Pr \models \psi \Leftarrow \phi$ iff $Pr(\psi \wedge \phi) = Pr(\phi)$ (iff $Pr(\psi \Leftarrow \phi) = 1$).
  - $Pr \models (\psi|\phi)[l, u]$ iff $Pr(\psi \wedge \phi) \in [l, u] \cdot Pr(\phi)$ (iff either $Pr(\phi) = 0$ or $Pr(\psi|\phi) \in [l, u]$).

Example

- Set of basic propositions $\Phi = \{bird, fly\}$.
- $\mathcal{I}_\Phi$ contains exactly the worlds I_1, I_2, I_3, and I_4 over Φ:

	fly	$\neg fly$
$bird$	I_1	I_2
$\neg bird$	I_3	I_4

- Some probabilistic interpretations:

Pr_1	fly	$\neg fly$
$bird$	19/40	1/40
$\neg bird$	10/40	10/40

Pr_2	fly	$\neg fly$
$bird$	0	1/3
$\neg bird$	1/3	1/3

- $Pr_1(fly \wedge bird) = 19/40$ and $Pr_1(bird) = 20/40$.
- $Pr_2(fly \wedge bird) = 0$ and $Pr_2(bird) = 1/3$.
- $\neg fly \Leftarrow bird$ is false in Pr_1, but true in Pr_2 .
- $(fly \mid bird)[.95, 1]$ is true in Pr_1, but false in Pr_2 .

Satisfiability and Logical Entailment

- Pr is a model of $KB = (L, P)$ iff $Pr \models F$ for all $F \in L \cup P$.
- KB is satisfiable iff a model of KB exists.
- $KB \Vdash (\psi|\phi)[l, u]$: $(\psi|\phi)[l, u]$ is a logical consequence of KB iff every model of KB is also a model of $(\psi|\phi)[l, u]$.
- $KB \Vdash_{tight} (\psi|\phi)[l, u]$: $(\psi|\phi)[l, u]$ is a tight logical consequence of KB iff l (resp., u) is the infimum (resp., supremum) of $Pr(\psi|\phi)$ subject to all models Pr of KB with $Pr(\phi) > 0$.

Example

- Probabilistic knowledge base:

$$KB = (\{bird \Leftarrow eagle\}, \{(have\_legs \mid bird)[1,1], (fly \mid bird)[0.95,1]\}).$$

- *KB* is satisfiable, since

  Pr with $Pr(bird \wedge eagle \wedge have\_legs \wedge fly) = 1$ is a model.

- Some conclusions under logical entailment:

$$KB \Vdash (have\_legs \mid bird)[0.3,1], \quad KB \Vdash (fly \mid bird)[0.6,1].$$

- Tight conclusions under logical entailment:

$$KB \Vdash_{tight} (have\_legs \mid bird)[1,1], \quad KB \Vdash_{tight} (fly \mid bird)[0.95,1],$$
$$KB \Vdash_{tight} (have\_legs \mid eagle)[1,1], \quad KB \Vdash_{tight} (fly \mid eagle)[0,1].$$

Deciding Model Existence / Satisfiability

Theorem: The probabilistic knowledge base $KB = (L, P)$ has a model Pr with $Pr(\alpha) > 0$ iff the following system of linear constraints over the variables y_r $(r \in R)$, where $R = \{I \in \mathcal{I}_\Phi \mid I \models L\}$, is solvable:

$$\begin{aligned}
\sum_{r \in R,\, r \models \neg\psi \wedge \phi} -l\, y_r + \sum_{r \in R,\, r \models \psi \wedge \phi} (1-l)\, y_r &\geq 0 \quad (\forall (\psi|\phi)[l,u] \in P) \\
\sum_{r \in R,\, r \models \neg\psi \wedge \phi} u\, y_r + \sum_{r \in R,\, r \models \psi \wedge \phi} (u-1)\, y_r &\geq 0 \quad (\forall (\psi|\phi)[l,u] \in P) \\
\sum_{r \in R,\, r \models \alpha} y_r &= 1 \\
y_r &\geq 0 \quad (\text{for all } r \in R)
\end{aligned}$$

Computing Tight Logical Consequences

Theorem: Suppose $KB = (L, P)$ has a model Pr such that $Pr(\alpha) > 0$. Then, l (resp., u) such that $KB \,\|\!\!\models_{tight} (\beta|\alpha)[l, u]$ is given by the optimal value of the following linear program over the variables y_r ($r \in R$), where $R = \{I \in \mathcal{I}_\Phi \mid I \models L\}$:

$$\text{minimize (resp., maximize)} \sum_{r \in R,\, r \models \beta \wedge \alpha} y_r \quad \text{subject to}$$

$$\sum_{r \in R,\, r \models \neg\psi\wedge\phi} -l\, y_r + \sum_{r \in R,\, r \models \psi\wedge\phi} (1 - l)\, y_r \geq 0 \quad (\forall (\psi|\phi)[l, u] \in P)$$

$$\sum_{r \in R,\, r \models \neg\psi\wedge\phi} u\, y_r + \sum_{r \in R,\, r \models \psi\wedge\phi} (u - 1)\, y_r \geq 0 \quad (\forall (\psi|\phi)[l, u] \in P)$$

$$\sum_{r \in R,\, r \models \alpha} y_r = 1$$

$$y_r \geq 0 \quad (\text{for all } r \in R)$$

Literature

- G. Boole. *An Investigation of the Laws of Thought, on which are Founded the Mathematical Theories of Logic and Probabilities*. Walton and Maberley, London, 1854.
- N. J. Nilsson. Probabilistic logic. *Artif. Intell.*, 28:71–88, 1986.
- D. Dubois, H. Prade, and J.-M. Touscas. Inference with imprecise numerical quantifiers. In *Intelligent Systems*, 1990.
- R. Fagin, J. Y. Halpern, and N. Megiddo. A logic for reasoning about probabilities. *Inf. Comput.*, 87:78–128, 1990.
- A. M. Frisch and P. Haddawy. Anytime deduction for probabilistic logic. *Artif. Intell.*, 69:93–122, 1994.
- T. Lukasiewicz. Probabilistic deduction with conditional constraints over basic events. *JAIR*, 10:199–241, 1999.
- T. Lukasiewicz. Probabilistic logic programming with conditional constraints. *ACM TOCL* 2(3):289–339, 2001.

Towards Stronger Notions of Entailment

Problem: Inferential weakness of logical entailment.

Solutions:

- Probability selection techniques: Perform inference from a representative distribution of the encoded convex set of distributions rather than the whole set, e.g.,
  - distribution of maximum entropy,
  - distribution in the center of mass.
- Probabilistic default reasoning: Perform constraining rather than conditioning and apply techniques from default reasoning to resolve local inconsistencies.
- Probabilistic independencies: Further constrain the convex set of distributions by probabilistic independencies.
  ($\Rightarrow$ adds nonlinear equations to linear constraints)

Entailment under Maximum Entropy

- Entropy of a probabilistic interpretation *Pr*, denoted $H(Pr)$:

$$H(Pr) \;=\; -\sum_{I \in \mathcal{I}_\Phi} Pr(I) \cdot \log Pr(I)\,.$$

- The ME model of a satisfiable probabilistic knowledge base *KB* is the unique probabilistic interpretation *Pr* that is a model of *KB* and that has the greatest entropy among all the models of *KB*.
- $KB \mathrel{\|\!\!=}^{me} (\psi|\phi)[l,u]$: $(\psi|\phi)[l,u]$ is a ME consequence of *KB* iff the ME model of *KB* is also a model of $(\psi|\phi)[l,u]$.
- $KB \mathrel{\|\!\!=}^{me}_{tight} (\psi|\phi)[l,u]$: $(\psi|\phi)[l,u]$ is a tight ME consequence of *KB* iff for the ME model *Pr* of *KB*, it holds either (a) $Pr(\phi) = 0$, $l = 1$, and $u = 0$, or (b) $Pr(\phi) > 0$ and $Pr(\psi|\phi) = l = u$.

Logical vs. Maximum Entropy Entailment

Probabilistic knowledge base:

$$KB = (\{bird \Leftarrow eagle\}, \{(have\_legs \mid bird)[1,1], (fly \mid bird)[0.95, 1]\}).$$

Tight conclusions under logical entailment:

$$KB \Vdash_{tight} (have\_legs \mid bird)[1,1], \quad KB \Vdash_{tight} (fly \mid bird)[0.95,1],$$
$$KB \Vdash_{tight} (have\_legs \mid eagle)[1,1], \quad KB \Vdash_{tight} (fly \mid eagle)[0,1].$$

Tight conclusions under maximum entropy entailment:

$$KB \mid\!\!\sim^{me}_{tight} (have\_legs \mid bird)[1,1], \quad KB \mid\!\!\sim^{me}_{tight} (fly \mid bird)[0.95,0.95],$$
$$KB \mid\!\!\sim^{me}_{tight} (have\_legs \mid eagle)[1,1], \quad KB \mid\!\!\sim^{me}_{tight} (fly \mid eagle)[0.95,0.95].$$

Lexicographic Entailment

- Pr verifies $(\psi|\phi)[l,u]$ iff $Pr(\phi) = 1$ and $Pr \models (\psi|\phi)[l,u]$.
- P tolerates $(\psi|\phi)[l,u]$ under L iff $L \cup P$ has a model that verifies $(\psi|\phi)[l,u]$.
- $KB = (L,P)$ is consistent iff there exists an ordered partition $(P_0, \ldots, P_k)$ of P such that each P_i is the set of all $C \in P \setminus \bigcup_{j=0}^{i-1} P_j$ tolerated under L by $P \setminus \bigcup_{j=0}^{i-1} P_j$.
- This (unique) partition is called the z-partition of KB.

Let $KB = (L, P)$ be consistent, and $(P_0, \ldots, P_k)$ be its z-partition.

- Pr is *lex*-preferable to Pr' iff some $i \in \{0, \ldots, k\}$ exists such that
  - $|\{C \in P_i \mid Pr \models C\}| > |\{C \in P_i \mid Pr' \models C\}|$ and
  - $|\{C \in P_j \mid Pr \models C\}| = |\{C \in P_j \mid Pr' \models C\}|$ for all $i < j \leq k$.
- A model Pr of $\mathcal{F}$ is a *lex*-minimal model of $\mathcal{F}$ iff no model of $\mathcal{F}$ is *lex*-preferable to Pr.
- $KB \Vdash\!\sim^{lex} (\psi|\phi)[l, u]$: $(\psi|\phi)[l, u]$ is a *lex*-consequence of KB iff every *lex*-minimal model Pr of L with $Pr(\phi)=1$ satisfies $(\psi|\phi)[l, u]$.
- $KB \Vdash\!\sim^{lex}_{tight} (\psi|\phi)[l, u]$: $(\psi|\phi)[l, u]$ is a tight *lex*-consequence of KB iff l (resp., u) is the infimum (resp., supremum) of $Pr(\psi)$ subject to all *lex*-minimal models Pr of L with $Pr(\phi) = 1$.

Logical vs. Lexicographic Entailment

Probabilistic knowledge base:

$$KB = (\{bird \Leftarrow eagle\}, \{(have\_legs \mid bird)[1,1], (fly \mid bird)[0.95,1]\}).$$

Tight conclusions under logical entailment:

$$KB \Vdash_{tight} (have\_legs \mid bird)[1,1], \quad KB \Vdash_{tight} (fly \mid bird)[0.95,1],$$
$$KB \Vdash_{tight} (have\_legs \mid eagle)[1,1], \quad KB \Vdash_{tight} (fly \mid eagle)[0,1].$$

Tight conclusions under probabilistic lexicographic entailment:

$$KB \mathrel{|\!\!\sim}^{lex}_{tight} (have\_legs \mid bird)[1,1], \quad KB \mathrel{|\!\!\sim}^{lex}_{tight} (fly \mid bird)[0.95,1],$$
$$KB \mathrel{|\!\!\sim}^{lex}_{tight} (have\_legs \mid eagle)[1,1], \quad KB \mathrel{|\!\!\sim}^{lex}_{tight} (fly \mid eagle)[0.95,1].$$

Probabilistic knowledge base:

$$KB = (\{bird \Leftarrow penguin\}, \{(have\_legs \mid bird)[1,1], (fly \mid bird)[1,1], (fly \mid penguin)[0, 0.05]\})\,.$$

Tight conclusions under logical entailment:

$KB \Vdash_{tight} (have\_legs \mid bird)[1,1]$, $KB \Vdash_{tight} (fly \mid bird)[1,1]$,
$KB \Vdash_{tight} (have\_legs \mid penguin)[1,0]$, $KB \Vdash_{tight} (fly \mid penguin)[1,0]$.

Tight conclusions under probabilistic lexicographic entailment:

$KB \mathrel{\|\!\!\sim}^{lex}_{tight} (have\_legs \mid bird)[1,1]$, $KB \mathrel{\|\!\!\sim}^{lex}_{tight} (fly \mid bird)[1,1]$,
$KB \mathrel{\|\!\!\sim}^{lex}_{tight} (have\_legs \mid penguin)[1,1]$, $KB \mathrel{\|\!\!\sim}^{lex}_{tight} (fly \mid penguin)[0, 0.05]$.

Probabilistic knowledge base:

$$KB = (\{bird \Leftarrow penguin\}, \{(have\_legs \mid bird)[0.99, 1], (fly \mid bird)[0.95, 1], (fly \mid penguin)[0, 0.05]\}).$$

Tight conclusions under logical entailment:

$$KB \Vdash_{tight} (have\_legs \mid bird)[0.99, 1],\ KB \Vdash_{tight} (fly \mid bird)[0.95, 1],$$
$$KB \Vdash_{tight} (have\_legs \mid penguin)[0, 1],\ KB \Vdash_{tight} (fly \mid penguin)[0, 0.05].$$

Tight conclusions under probabilistic lexicographic entailment:

$$KB \mathrel{|\!\!\sim}^{lex}_{tight} (have\_legs \mid bird)[0.99, 1],\ KB \mathrel{|\!\!\sim}^{lex}_{tight} (fly \mid bird)[0.95, 1],$$
$$KB \mathrel{|\!\!\sim}^{lex}_{tight} (have\_legs \mid penguin)[0.99, 1],\ KB \mathrel{|\!\!\sim}^{lex}_{tight} (fly \mid penguin)[0, 0.05].$$

Literature

- J. B. Paris. *The Uncertain Reasoner's Companion: A Mathematical Perspective*. Cambridge University Press, 1995.
- G. Kern-Isberner and T. Lukasiewicz. Combining probabilistic logic programming with the power of maximum entropy. *Artif. Intell.*, 157(1–2):139–202, 2004.
- T. Lukasiewicz. Probabilistic Default Reasoning with Conditional Constraints. *Ann. Math. Artif. Intell.*, 34(1–3):35–88, 2002.
- J. Y. Halpern. *Reasoning about Uncertainty*. MIT Press, 2003.

Bayesian Networks

Well-structured, exact conditional constraints plus conditional independencies specify exactly one joint probability distribution.

Joint probability distributions can answer any queries, but can be very large and are often hard to specify.

Bayesian network (BN): compact specification of a joint distribution, based on a graphical notation for conditional independencies:

- a set of nodes; each node represents a random variable
- a directed, acyclic graph (link $\approx$ "directly influences")
- a conditional distribution for each node given its parents: $\mathbf{P}(X_i | Parents(X_i))$

Any joint distribution can be represented as a BN.

Example

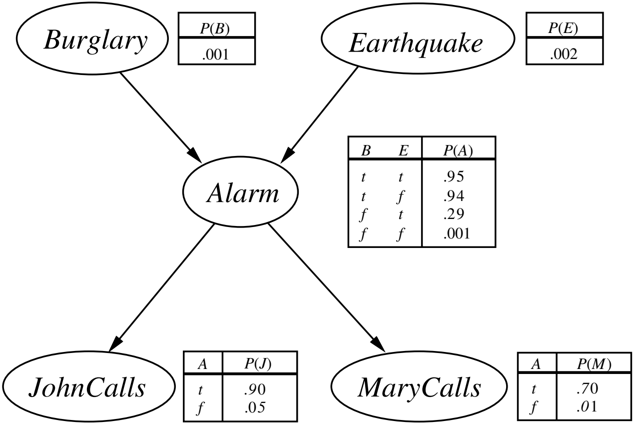
I'm at work, neighbor John calls to say my alarm is ringing, but neighbor Mary doesn't call. Sometimes it's set off by minor earthquakes. Is there a burglar?

Variables: *Burglary*, *Earthquake*, *Alarm*, *JohnCalls*, *MaryCalls*

Network topology reflects "causal" knowledge:

- a burglar can set the alarm off
- an earthquake can set the alarm off
- the alarm can cause Mary to call
- the alarm can cause John to call

John sometimes confuses the telephone ringing with the alarm.
Mary likes rather loud music and sometimes misses the alarm.

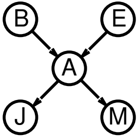

Burglary
P(B)
.001
Earthquake
P(E)
.002
Alarm
B E P(A)
t t .95
t f .94
f t .29
f f .001
JohnCalls
A P(J)
t .90
f .05
MaryCalls
A P(M)
t .70
f .01

Global Semantics

"Global" semantics defines the full joint distribution as the product of the local conditional distributions:

$$\mathbf{P}(X_1, \ldots, X_n) = \prod_{i=1}^{n} \mathbf{P}(X_i | Parents(X_i))$$

e.g.,



$$\begin{aligned} P(j \wedge m \wedge a \wedge \neg b \wedge \neg e) &= P(j|a)P(m|a)P(a|\neg b, \neg e)P(\neg b)P(\neg e) \\ &= 0.90 \times 0.70 \times 0.001 \times 0.999 \times 0.998 \\ &= 0.00062 \end{aligned}$$

Inference Tasks

- Simple queries: compute posterior marginal $\mathbf{P}(X_i|\mathbf{E}=\mathbf{e})$, e.g., $P(Burglary|Alarm=true,\ John=true,\ Mary=false)$.
- Conjunctive queries: $\mathbf{P}(X_i, X_j|\mathbf{E}=\mathbf{e}) = \mathbf{P}(X_i|\mathbf{E}=\mathbf{e})\mathbf{P}(X_j|X_i, \mathbf{E}=\mathbf{e})$.
- Optimal decisions: decision networks include utility information; probabilistic inference required for $P(outcome|action, evidence)$.
- Value of information: which evidence to seek next?
- Sensitivity analysis: which probability values are most critical?

Probabilistic Causal Models

Causal influences between the random variables expressed by functions rather than conditional probabilities.

Probability distribution over the set of all *contexts* (= all variable instantiations of the exogenous variables).

Sophisticated notions of causes and explanations.

Causal model $M = (U, V, F)$:

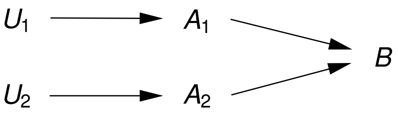
- U is a finite set of exogenous variables,
- V is a finite set of endogenous variables with $U \cap V = \emptyset$,
- $F = \{F_X \mid X \in V\}$ is a set of functions, where each F_X assigns a value to X for each value of its parents $PA_X \subseteq U \cup V \setminus \{X\}$.

M is recursive: total ordering $\prec$ on V such that $Y \in PA_X$ implies $Y \prec X$.

A probabilistic causal model (M, P) consists of a causal model $M = (U, V, F)$ and a probability function P on the values of U.

Example

Two arsonists lit matches ($A_i = 1$), $i \in \{1, 2\}$, in different parts of a dry forest, and both cause trees to start burning. Either match by itself suffices to burn down the whole forest ($B = 1$):



Probabilistic causal model $((U, V, F), P)$:

- U: binary background variables U_1 and U_2.
- V: binary observable variables A_1, A_2, and B.
- F: functions to express causal dependencies between variables: $F_{A_1} = U_1$, $F_{A_2} = U_2$, and $F_B = 1$ iff $A_1 = 1$ or $A_2 = 1$.
- P: probability distribution over the values of U: P: $(0, 0)$, $(0, 1)$, $(1, 0)$, $(1, 1) \mapsto 0.3, 0.3, 0.2, 0.2$.

Literature

- S. J. Russel and P. Norvig. *Artificial Intelligence: A Modern Approach*, 2nd edition, Chapter 14. Prentice Hall, 2002.
- F. V. Jensen. *Bayesian Networks and Decision Graphs*. Springer, 2001.
- J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.
- Online tutorials, software packages, and datasets on BNs:
  - http://www.auai.org/
  - http://www.ai.mit.edu/∼murphyk/Bayes/bayes.html
- J. Pearl. *Causality: Models, Reasoning, and Inference.* Cambridge University Press, 2000.

Probabilities about Generic and Concrete Objects

Combining generic and concrete probability distributions:

- Conditioning: Generic distributions are conditioned on the (classical) information about concrete distributions.
- Probabilistic default reasoning: Generic distributions are constrained by the (not necessarily classical) information about the concrete distributions, and techniques from default reasoning resolve local inconsistencies.
- Minimum cross entropy: Generic and concrete distributions are combined via cross entropy minimization.

Probabilistic Ontologies

Main types of encoded probabilistic knowledge:

- Terminological probabilistic knowledge about concepts and roles: “Birds fly with a probability of at least 0.95”.
- Assertional probabilistic knowledge about instances of concepts and roles: “Tweety is a bird with a probability of at least 0.9”.

Main types of reasoning problems:

- Satisfiability of the terminological probabilistic knowledge.
- Tight conclusions about generic objects (from the terminological probabilistic knowledge).
- Satisfiability of the assertional probabilistic knowledge.
- Tight conclusions about concrete objects (from both the terminological and the assertional probabilistic knowledge).

Use of Probabilistic Ontologies

- Representation of terminological and assertional probabilistic knowledge (e.g., in the medical domain or at the stock exchange market).
- Information retrieval, for an increased recall (e.g., Udrea et al.: Probabilistic ontologies and relational databases. In *Proc. CoopIS/DOA/ODBASE-2005*).
- Ontology matching (e.g., Mitra et al.: OMEN: A probabilistic ontology mapping tool. In *Proc. ISWC-2005*).
- Probabilistic data integration, especially for handling ambiguous and controversial pieces of information.

Probabilistic RDF

O. Udrea, V. S. Subrahmanian, and Z. Majkic. Probabilistic RDF. In *Proceedings IRI-2006*.

- probabilistic generalization of RDF
- terminological probabilistic knowledge about classes
- assertional probabilistic knowledge about properties of individuals
- assertional probabilistic inference for acyclic probabilistic RDF theories, which is based on logical entailment in probabilistic logic, coupled with a local probabilistic semantics

Probabilistic DLs

R. Giugno, T. Lukasiewicz. P-$\mathcal{SHOQ}(\mathbf{D})$: A probabilistic extension of $\mathcal{SHOQ}(\mathbf{D})$ for probabilistic ontologies in the SW. In *Proc. JELIA-2002*.

- probabilistic generalization of the description logic $\mathcal{SHOQ}(\mathbf{D})$ (recently also extended to $\mathcal{SHIF}(\mathbf{D})$ and $\mathcal{SHOIN}(\mathbf{D})$)
- terminological probabilistic knowledge about concepts and roles
- assertional probabilistic knowledge about instances of concepts and roles
- terminological probabilistic inference based on lexicographic entailment in probabilistic logic (stronger than logical entailment)
- assertional probabilistic inference based on lexicographic entailment in probabilistic logic (for combining assertional and terminological probabilistic knowledge)
- terminological and assertional probabilistic inference problems reduced to sequences of linear optimization problems

M. Jaeger. Probabilistic reasoning in terminological logics. In *Proceedings KR-1994*.

- probabilistic generalization of the description logic $\mathcal{ALC}$
- terminological probabilistic knowledge about concepts and roles
- assertional probabilistic knowledge about concept instances, but no assertional probabilistic knowledge about role instances
- terminological probabilistic inference based on logical entailment in probabilistic logic (by solving linear optimization problems)
- assertional probabilistic inference based on cross entropy mini- mization relative to terminological probabilistic knowledge (by an approximation algorithm; no exact algorithm known so far)

D. Koller, A. Levy, and A. Pfeffer. P-CLASSIC: A tractable probabilistic description logic. In *Proceedings AAAI-1997*.

- probabilistic generalization (of a variant) of the description logic CLASSIC
- so-called *p-classes* express terminological probabilistic knowledge about concepts, roles, and attributes
- but assertional classical and probabilistic knowledge about instances of concepts and roles is not supported
- probabilistic semantics based on Bayesian networks
- determines exact probabilities for conditionals between concept expressions in canonical form
- probabilistic inference can be done in polynomial time, when the underlying Bayesian network is a polytree

Possibilistic DLs

Generalization of DLs by possibilistic uncertainty, which is based on possibilistic interpretations rather than probabilistic interpretations.

Possibilistic interpretation: mapping $\pi\colon \mathcal{I}_\Phi \rightarrow [0, 1]$.

"$\pi(I)$ is the degree to which the world I is possible."

Poss(ϕ): possibility of ϕ in π: $Poss(\phi) = \max\{\pi(I) \mid I \in \mathcal{I}_\Phi,\ I \models \phi\}$

- B. Hollunder. An alternative proof method for possibilistic logic and its application to terminological logics. *Int. J. Approx. Reasoning*, 12(2):85–109, 1995.
- D. Dubois, J. Mengin, and H. Prade. Possibilistic uncertainty and fuzzy features in description logic: A preliminary discussion. In E. Sanchez, editor, *Capturing Intelligence: Fuzzy Logic and the Semantic Web*, 2006.
- C.-J. Liau and Y. Y. Yao. Information retrieval by possibilistic reasoning. In *Proc. DEXA-2001*.

Probabilistic OWL

P. C. G. da Costa. *Bayesian Semantics for the Semantic Web*. PhD thesis, George Mason University, Fairfax, VA, USA, 2005.

P. C. G. da Costa and K. B. Laskey. PR-OWL: A framework for probabilistic ontologies. In *Proceedings FOIS-2006*.

- probabilistic extension of OWL
- probabilistic semantics based on multi-entity Bayesian networks (MEBNs), which are a Bayesian logic that combines first-order logic with Bayesian probabilities:
  - represents knowledge as parameterized fragments of Bayesian networks
  - expresses repeated structure
  - represents probability distribution on interpretations of associated first-order theory

Other Works

- Z. Ding and Y. Peng. A probabilistic extension to ontology language OWL. In *Proceedings HICSS-2004*.
- Y. Yang and J. Calmet. OntoBayes: An ontology-driven uncertainty model. In *Proceedings IAWTIC-2005*.
- Z. Ding, Y. Peng, and R. Pan. BayesOWL: Uncertainty modeling in Semantic Web ontologies. In Z. Ma, editor, *Soft Computing in Ontologies and Semantic Web*. Springer, 2006.
- H. Nottelmann and N. Fuhr. Adding probabilities and rules to OWL Lite subsets based on probabilistic Datalog. *IJUFKS*, 14(1):17–42, 2006.

Probabilistic Logic Programs

Probabilistic generalizations of logic programs / rule-based systems / deductive databases / Datalog:

(1) Probabilistic generalizations of (annotated) logic programs based on probabilistic logic (no uncertainty degrees associated with rules):

- R. T. Ng and V. S. Subrahmanian. Probabilistic logic programming. *Inf. Comput.*, 101(2):150–201, 1992.
- R. T. Ng and V. S. Subrahmanian. A semantical framework for supporting subjective and conditional probabilities in deductive databases. *J. Autom. Reasoning*, 10(2):191–235, 1993.
- A. Dekhtyar and V. S. Subrahmanian. Hybrid probabilistic programs. *J. Log. Program.* 43(3):187–250, 2000.

(2) Probabilistic generalizations of logic programs based on Bayesian networks / causal models:

- D. Poole. Probabilistic Horn abduction and Bayesian networks. *Artif. Intell.*, 64:81–129, 1993.
- D. Poole. The independent choice logic for modeling multiple agents under uncertainty. *Artif. Intell.*, 94:7–56, 1997.
- K. Kersting and L. De Raedt. Bayesian logic programs. *CoRR*, cs.AI/0111058, 2001.
- C. Baral, M. Gelfond, and J. N. Rushton. Probabilistic reasoning with answer sets. In *Proceedings LPNMR-2004*.

(3) Relational Bayesian networks:

- M. Jaeger. Relational Bayesian networks. In *Proc. UAI-1997*.
- D. Koller and A. Pfeffer. Object-oriented Bayesian networks. In *Proceedings UAI-1997*.
- H. Pasula and S. J. Russell. Approximate inference for first-order probabilistic languages. In *Proceedings IJCAI-2001*.
- D. Poole. First-order probabilistic inference. In *Proc. IJCAI-2003*.

(4) First-order generalization of probabilistic knowledge bases in probabilistic logic (based on logical entailment, lexicographic entailment, and maximum entropy entailment):

- T. Lukasiewicz. Probabilistic logic programming. In *Proceedings ECAI-1998*.
- T. Lukasiewicz. Probabilistic logic programming with conditional constraints. *ACM TOCL* 2(3):289–339, 2001.
- T. Lukasiewicz. Probabilistic logic programming under inheritance with overriding. In *Proceedings UAI-2001*.
- G. Kern-Isberner and T. Lukasiewicz. Combining probabilistic logic programming with the power of maximum entropy. *Artif. Intell.*, 157(1–2):139–202, 2004.

Poole’s Independent Choice Logic (ICL)

Acyclic logic programs P under different “choices”.

Each choice along with P produces a first-order model.

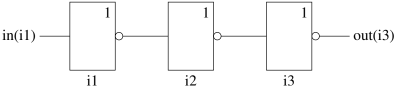
By placing a probability distribution over the different choices, one then obtains a distribution over the set of first-order models.

ICL generalizes Pearl’s structural causal models.

ICL also generalizes Bayesian networks, influence diagrams, Markov decision processes, and normal form games.

Example

Sequence of three not-gates:



$$
\begin{array}{l}
val(out(G), on, T) \leftarrow ok(G) \wedge val(in(G), off, T). \\
val(out(G), off, T) \leftarrow ok(G) \wedge val(in(G), on, T). \\
val(out(G), V, T) \leftarrow shorted(G) \wedge val(in(G), V, T). \\
val(out(G), off, T) \leftarrow blown(G). \\
val(in(G), V, T) \leftarrow conn(G_1, G) \wedge val(out(G_1), V, T). \\
conn(i_1, i_2) \leftarrow . \\
conn(i_2, i_3) \leftarrow .
\end{array}
$$

$$
\begin{array}{l}
disjoint([ok(G){:}0.95,\ shorted(G){:}0.03,\ blown(G){:}0.02]). \\
disjoint([val(in(i_1), on, T){:}0.5,\ val(in(i_1), off, T){:}0.5]).
\end{array}
$$

Possible queries: Which is the probability that gate i_2 is *ok* given that both the input of i_1 and the output of i_3 are *off* at the time point t_1?

$$P(ok(i_2) | val(in(i_1), off, t_1) \wedge val(out(i_3), off, t_1)) = 0.76 .$$

Which is the probability that the output of i_3 is *off* given that the input of i_1 is *on* at the time point t_1?

$$P(val(out(i_3), off, t_1) | val(in(i_1), on, t_1)) = 0.899 .$$

Intuitively: Every closed formula is associated with a set of minimal explanations. Every explanation is a set of hypotheses. The probability of an explanation is the product of the probabilities of the hypotheses. The probability of a closed formula is the sum of the probabilities of all associated minimal explanations.

The formula $F = val(in(i_1), off, t_1) \wedge val(out(i_3), off, t_1)$ is associated with the following minimal explanations along with their probabilities:

$$
\begin{aligned}
E_1 &= \{val(in(i_1), off, t_1), ok(i_3), ok(i_2), shorted(i_1)\} \\
P(E_1) &= 0.5 \times 0.95 \times 0.95 \times 0.03 \;=\; 0.01354 \\
E_2 &= \{val(in(i_1), off, t_1), ok(i_3), shorted(i_2), ok(i_1)\} \\
P(E_2) &= 0.5 \times 0.95 \times 0.03 \times 0.95 \;=\; 0.01354 \\
&\vdots
\end{aligned}
$$

The sum of the probabilities of all minimal explanations associated with F is 0.05996. Hence, the formula F has the probability 0.05996.

Probabilistic Description Logic Programs

T. Lukasiewicz. Probabilistic description logic programs. *IJAR*, 2007.

- Probabilistic dl-programs generalize (loosely coupled) dl-programs by probabilistic uncertainty as in Poole's ICL.
- They properly generalize Poole's ICL.
- They consist of a dl-program along with a probability distribution μ over total choices B.
- They specify a set of distributions over first-order models: Every total choice B along with the dl-program specifies a set of first-order models of which the probabilities should sum up to $\mu(B)$.
- There are also tightly coupled probabilistic dl-programs.
- Important applications are data integration and ontology mapping under probabilistic uncertainty and inconsistency.

Example

Description logic knowledge base L
of a probabilistic dl-program $KB = (L, P, C, \mu)$:

$PC \sqcup Camera \sqsubseteq Electronics$; $PC \sqcap Camera \sqsubseteq \bot$;
$Book \sqcup Electronics \sqsubseteq Product$; $Book \sqcap Electronics \sqsubseteq \bot$;
$Textbook \sqsubseteq Book$;

$Product \sqsubseteq\ \geq 1\ related$;
$\geq 1\ related \sqcup \geq 1\ related^{-} \sqsubseteq Product$;

$Textbook(tb\_ai)$; $Textbook(tb\_lp)$;
$PC(pc\_ibm)$; $PC(pc\_hp)$;

$related(tb\_ai, tb\_lp)$; $related(pc\_ibm, pc\_hp)$;
$provides(ibm, pc\_ibm)$; $provides(hp, pc\_hp)$.

Classical dl-rules in P
of a probabilistic dl-program $KB = (L, P, C, \mu)$:

- $pc(pc\_1)$; $pc(pc\_2)$; $pc(pc\_3)$;
- $brand\_new(pc\_1)$; $brand\_new(pc\_2)$;
- $vendor(dell, pc\_1)$; $vendor(dell, pc\_2)$; $vendor(dell, pc\_3)$;
- $provider(P) \leftarrow vendor(P, X), DL[PC \uplus pc; Product](X)$;
- $provider(P) \leftarrow DL[provides](P, X), DL[PC \uplus pc; Product](X)$;
- $similar(X, Y) \leftarrow DL[related](X, Y)$;
- $similar(X, Z) \leftarrow similar(X, Y), similar(Y, Z)$.

Probabilistic dl-rules in P along with the probability μ on the choice space C of a probabilistic dl-program $KB = (L, P, C, \mu)$:

- $avoid(X) \leftarrow DL[Camera](X), \text{not } offer(X), avoid\_pos$;
- $offer(X) \leftarrow DL[PC \uplus pc; Electronics](X), \text{not } brand\_new(X), offer\_pos$;
- $buy(C, X) \leftarrow needs(C, X), view(X), \text{not } avoid(X), v\_buy\_pos$;
- $buy(C, X) \leftarrow needs(C, X), buy(C, Y), also\_buy(Y, X), a\_buy\_pos$.

μ: $avoid\_pos, avoid\_neg \mapsto 0.9\,, 0.1$; $offer\_pos, offer\_neg \mapsto 0.9\,, 0.1$; $v\_buy\_pos, v\_buy\_neg \mapsto 0.7\,, 0.3$; $a\_buy\_pos, a\_buy\_neg \mapsto 0.7\,, 0.3$.

$\{avoid\_pos, offer\_pos, v\_buy\_pos, a\_buy\_pos\}$: $0.9 \times 0.9 \times 0.7 \times 0.7, \ldots$

Probabilistic query: $\exists (buy(c, x) \mid needs(c, x) \wedge buy(c, y) \wedge also\_buy(y, x) \wedge view(x) \wedge \neg avoid(x))[L, U]$

Example: Probabilistic Data Integration

Obtain a weather forecast by integrating the potentially different weather forecasts of three weather forecast institutes A, B, and C.

Our trust in the institutes A, B, and C is expressed by the trust probabilities 0.6, 0.3, and 0.1, respectively.

Probabilistic integration of the source schemas of A, B, and C to the global schema G is specified by the following $KB_M = (\emptyset, P_M, C_M, \mu_M)$:

$$
\begin{aligned}
P_M = \{ & forecast\_rome(D, W, T, M) \leftarrow forecast(rome, D, W, T, M),\ inst_A; \\
& forecast\_rome(D, W, T, M) \leftarrow forecastRome(D, W, T, M),\ inst_B; \\
& forecast\_rome(D, W, T, M) \leftarrow forecast\_weather(rome, D, W), \\
& \qquad forecast\_temperature(rome, D, T), \\
& \qquad forecast\_wind(rome, D, M),\ inst_C \}\,; \\
C_M = \{ & \{inst_A, inst_B, inst_C\} \}\,; \\
\mu_M : \ & inst_A, inst_B, inst_C \ \mapsto \ 0.6, 0.3, 0.1\,.
\end{aligned}
$$

Example (Tightly Coupled): Ontology Mapping

The global schema contains the concept *logic_programming*, while the source schemas contain only the concepts *rule-based_systems* resp. *deductive_databases* in their ontologies.

A randomly chosen book from the area *rule-based_systems* (resp., *deductive_databases*) may belong to *logic_programming* with the probability 0.7 (resp., 0.8).

Probabilistic mapping from the two source schemas to the global schema expressed by the following $KB_M = (\emptyset, P_M, C_M, \mu_M)$:

$$
\begin{aligned}
P_M &= \{\mathit{logic\_programming}(X) \leftarrow \mathit{rule\text{-}based\_systems}(X),\ \mathit{choice}_1\,;\\
&\qquad \mathit{logic\_programming}(X) \leftarrow \mathit{deductive\_databases}(X),\ \mathit{choice}_2\}\,;\\
C_M &= \{\{\mathit{choice}_1, \mathit{not\_choice}_1\}, \{\mathit{choice}_2, \mathit{not\_choice}_2\}\}\,;\\
\mu_M &:\ \ \mathit{choice}_1, \mathit{not\_choice}_1, \mathit{choice}_2, \mathit{not\_choice}_2 \ \mapsto\ 0.7,\ 0.3,\ 0.8,\ 0.2\,.
\end{aligned}
$$

Vagueness

- Vagueness: statements involve concepts for which there is no exact definition, such as tall, close, cheap, IsAbout, simialarTo ...
- Statements are true to some degree which is taken from a truth space
  - E.g., "Hotel Verdi is close to the train station to degree 0.83"
  - "Find top-k cheapest hotels close to the train station"

$$q(h) \leftarrow hasLocation(h, hl) \wedge hasLocation(\texttt{train}, cl) \wedge close(hl, cl) \wedge cheap(h)$$

- Truth space: usually $[0, 1]$
- Interpretation: a function I mapping atoms into $[0, 1]$, i.e. $I(A) \in [0, 1]$
- Problem: what is the interpretation of e.g. $close(verdi, train) \wedge cheap(200)$?
  - E.g., if $I(close(verdi, train)) = 0.83$ and $I(cheap(200)) = 0.2$, what is the result of $0.83 \wedge 0.2$?
- More generally, what is the result of $n \wedge m$, for $n, m \in [0, 1]$?
- The choice cannot be any arbitrary computable function, but has to reflect some basic properties that one expects to hold for a "conjunction"

Propositional Fuzzy Logics Basics [5]

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

Axioms for t-norms and s-norms

Axiom Name	T-norm	S-norm
Tautology / Contradiction	$a \wedge 0 = 0$	$a \vee 1 = 1$
Identity	$a \wedge 1 = a$	$a \vee 0 = a$
Commutativity	$a \wedge b = b \wedge a$	$a \vee b = b \vee a$
Associativity	$(a \wedge b) \wedge c = a \wedge (b \wedge c)$	$(a \vee b) \vee c = a \vee (b \vee c)$
Monotonicity	if $b \leq c$, then $a \wedge b \leq a \wedge c$	if $b \leq c$, then $a \vee b \leq a \vee c$

Axioms for implication and negation functions

Axiom Name	Implication Function	Negation Function
Tautology / Contradiction	$0 \rightarrow b = 1$ $a \rightarrow 1 = 1$	$\neg\, 0 = 1, \;\; \neg\, 1 = 0$
Antitonicity	if $a \leq b$, then $a \rightarrow c \geq b \rightarrow c$	if $a \leq b$, then $\neg\, a \geq \neg\, b$
Monotonicity	if $b \leq c$, then $a \rightarrow b \leq a \rightarrow c$	

Usually,

$$a \rightarrow b = \sup\{c \colon a \wedge c \leq b\}$$

is used and 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, $x \Rightarrow y \equiv \neg x \vee y$

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

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

- Note:

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

- Zadeh semantics: not interesting for fuzzy logicians: its a sub-logic of Łukasiewicz and, thus, rarely considered by fuzzy logicians

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

Some additional properties of t-norms, s-norms, implication functions, and negation functions of various fuzzy logics.

Property	Łukasiewicz Logic	Gödel Logic	Product Logic	Zadeh Logic
$x \wedge \neg x = 0$	•			
$x \vee \neg x = 1$	•			
$x \wedge x = x$		•		•
$x \vee x = x$		•		•
$\neg \neg x = x$	•			•
$x \rightarrow y = \neg x \vee y$	•			•
$\neg (x \rightarrow y) = x \wedge \neg y$	•			•
$\neg (x \wedge y) = \neg x \vee \neg y$	•	•	•	•
$\neg (x \vee y) = \neg x \wedge \neg y$	•	•	•	•

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)	$(\phi \wedge (\psi \to \chi)) \to ((\phi \wedge \psi) \to \chi))$
(A5b)	$((\phi \wedge \psi) \to \chi)) \to (\phi \wedge (\psi \to \chi))$
(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)	$\neg\neg\phi \to \phi$
(Deduction rule)	Modus ponens: from ϕ and $\phi \to \psi$ infer ψ

Proposition

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

- $\models_{Ł} \phi \to \psi \equiv \neg\psi \to \neg\phi$
- $\models_{Ł} \neg(\phi \wedge \psi) \equiv \neg\phi \vee \neg\psi$
- $\models_{Ł} \phi \to \psi \equiv \neg(\phi \wedge \neg\psi)$
- $\models_{Ł} \phi \to \psi \equiv \neg\phi \vee \neg\psi$
- $\models_{Ł} \neg(\phi \to \psi) \equiv \phi \wedge \neg\psi$
- 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) $\neg\neg\chi \to ((\phi \land \chi \to \psi \land \chi) \to (\phi \to \psi))$

(Π2) $(\phi \bar{\land} \neg\phi) \to 0$

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

Proposition

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

- $\models_{\Pi} \neg(\phi \land \psi) \to \neg(\phi \bar{\land} \psi)$
- $\models_{\Pi} (\phi \to \neg\phi) \to \neg\phi$
- $\models_{\Pi} \neg\phi \bar{\lor} \neg\neg\phi$

Axioms of Gödel logic G

Fix Gödel t-norm and r-implication.

(Axioms) Axioms of BL

(G) $\phi \rightarrow (\phi \wedge \phi)$

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

Proposition

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

- $\models_{BL2} \phi \rightarrow (\phi \wedge \phi)$ (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 ($r, s \in [0, 1]$): from $r \to \phi$ and $s \to (\phi \to \psi)$ infer $(r \wedge s) \to \psi$

$$\text{From } \phi \geq r \text{ and } (\phi \to \psi) \geq s \text{ infer } \psi \geq r \wedge s$$

- Let

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

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

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

Tableau 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 $r \in [0,1]$, variable or expresson $1 - r'$ (r' variable), admitting solution in $[0,1]$, $\neg r = 1 - r$, $\neg\neg r = r$

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

- Now 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 [5]

- 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 $\mathcal{I} : Atoms \to [0, 1]$
- Interpretations are extended to formulae as follows:

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

  where $\mathcal{I}_x^c$ 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 $||\psi||_{\mathcal{T}} \geq ||\phi||_{\mathcal{T}} \wedge ||\phi \to \psi||_{\mathcal{T}}$
- $\models_{BL\forall} \exists x\phi \to \neg\forall x\neg\phi$
- $\models_{BL\forall} \neg\exists x\phi \equiv \forall x\neg\phi$
- $\models_{Ł\forall} \exists x\phi \equiv \neg\forall x\neg\phi$

- $(\neg\forall x p(x)) \land (\neg\exists x \neg p(x))$ 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 RDF (we generalize [15, 16, 34])

- Statement (triples) may have attached a degree in $[0, 1]$: for $n \in [0, 1]$

$$\langle (\mathit{subject}, \mathit{predicate}, \mathit{object}), n \rangle$$

- Meaning: the degree of truth of the statement is at least n
- For instance,

$$\langle (o1, \mathit{IsAbout}, \mathit{snoopy}), 0.8 \rangle$$

Fuzzy RDF Semantics

- In Fuzzy RDF MT, an interpretation I of a vocabulary V consists of:
  - IR, a non-empty set of resources, called the domain of I.
  - A non empty set IDP, called the property domain of I
  - A mapping $IP : IDP \longrightarrow [0,1]$ (fuzzy the set of properties of I),
  - $IEXT : IP \rightarrow (2^{IR \times IR} \longrightarrow [0,1])$, given a property, given a subject and and object, returns a value in $[0,1]$
  - IS, a mapping from URI references in V into $IR \cup IDP$
  - IL, a mapping from typed literals in V into IR
  - A distinguished subset LV of IR, set of literal values, which contains all the plain literals in V
- Satisfiability:

$$I \models \langle (s,p,o), n \rangle \text{ iff } IP(I(p)) \wedge IEXT(I(p))(I(s), I(o)) \geq n$$

- For instance, using Gödel t-norm $x \wedge y = \min(x,y)$, if

$$\begin{array}{lcl} I(o1) & = & s \\ I(IsAbout) & = & p \\ I(snoopy) & = & o \\ IP(p) & = & 0.9 \\ IEXT(p)(s,o) & = & 0.85 \end{array}$$

then

$$I \models \langle (o1, IsAbout, snoopy), 0.8 \rangle \text{ as } \min(IP(p), IEXT(p)(s,o)) = \min(0.9, 0.85) = 0.85 \geq 0.8$$

Fuzzy RDFS Interpretations

- In fuzzy RDFS, class extensions are fuzzy sets of domainÕs elements.
- Class interpretation $ICEXT$ is induced by $IEXT(I(type))$

$$ICEXT(y)(x) = IEXT(I(type))(x, y)$$

  If x is of type y then the degree of being x and instance of y is given by $ICEXT(y)(x)$

- Fuzzy RDFS adds extra constraints on interpretations, such as
  1. $ICEXT(y)(u) = IEXT(I(domain))(x, y) \wedge \exists v.IEXT(x)(u, v))$
  2. $ICEXT(y)(v) = IEXT(I(range))(x, y) \wedge \exists u.IEXT(x)(u, v))$
  3. $IEXT(I(subPropertyOf))$ is transitive and reflexive on IP
     - a binary relation R is reflexive iff $R(x, y) = R(y, x)$
     - a binary relation R is transitive iff $R(x, y) \geq \sup_z R(x, z) \wedge R(z, y)$
  4. $IEXT(subPropertyOf)(x, y) = IP(x) \wedge IP(y) \wedge \forall\langle a, b\rangle.IP(x)(a, b) \rightarrow IP(y)(a, b)$
  5. $IEXT(subClassOf)(x, y) = IC(x) \wedge IC(y) \wedge \forall a.IC(x)(a) \rightarrow IC(y)(a)$
  6. $IEXT(I(subClassOf))$ is transitive and reflexive on IC
  7. $IEXT(I(subClassOf))(x, I(Resource)) = IC(x)$
  8. $IEXT(I(subPropertyOf))(x, I(member)) = ICEXT(I(ContainerMembershipProperty))(x)$
  9. $ICEXT(I(Datatype))(x) = IEXT(I(subClassOf))(x, I(Literal))$

Inferences in Fuzzy RDFS

Some inferences in fuzzy RDFS (set is not complete). Recall Rational Pavelka Logic ($\rightarrow$ is r-implication)

$$\frac{\langle(a, sp, b), n\rangle, \langle(b, sp, c), m\rangle}{(\langle(a, sp, c), n \wedge m\rangle} \qquad \frac{\langle(a, sp, b), n\rangle, \langle(x, a, y), m\rangle}{\langle(x, b, y), n \wedge m\rangle}$$

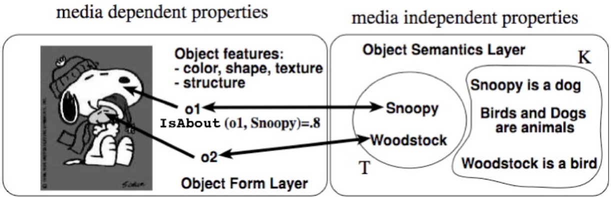
$$\frac{\langle(a, sc, b), n\rangle, \langle(b, sc, c), m\rangle}{\langle(a, sc, c), n \wedge m\rangle} \qquad \frac{\langle(a, sc, b), n\rangle, \langle(x, type, a), m\rangle}{\langle(x, type, b), n \wedge m\rangle}$$

$$\frac{\langle(a, dom, b), n\rangle, \langle(x, a, y), m\rangle}{\langle(x, type, b), n \wedge m\rangle} \qquad \frac{\langle(a, range, b), n\rangle, \langle(x, a, y), m\rangle}{\langle(y, type, b), n \wedge m\rangle}$$

$$\frac{\langle(a, dom, b), n\rangle, \langle(c, sp, a), m\rangle, \langle(x, c, y), k\rangle}{\langle(x, type, b), n \wedge m \wedge k\rangle} \qquad \frac{\langle(a, range, b), n\rangle, \langle(c, sp, a), m\rangle, \langle(x, c, y), k\rangle}{\langle(y, type, b), n \wedge m \wedge k\rangle}$$

sp = “subPropertyOf”, sc = “subClassOf”

Example



- Fuzzy RDF representation

$$\begin{array}{l}\langle (o1, IsAbout, snoopy), 0.8\rangle \\ \langle (snoopy, type, dog), 1.0\rangle \\ \langle (woodstock, type, bird), 1.0\rangle \\ \langle (dog, subClassOf, Animal), 1.0\rangle \\ \langle (bird, subClassOf, Animal), 1.0\rangle \end{array}$$

- then

$$KB \models \langle \exists x.(o1, IsAbout, x) \wedge (x, type, Animal), 0.8\rangle$$

Fuzzy DLs Basics [26]

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

Interpretation:

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

$\wedge$ = t-norm
$\vee$ = s-norm
$\neg$ = negation
$\rightarrow$ = implication

Concepts:

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

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 *KB* consistent, i.e. satisfiable?

Subsumption: structure knowledge, compute taxonomy

- $KB \models C \sqsubseteq D$?

Equivalence: check if two fuzzy concepts are the same

- $KB \models C = D$?

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

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

BTVB: Best Truth Value Bound problem

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

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

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

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 $C \sqcap D \sqsubseteq \bot$ 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 $x \in \Delta^{\mathcal{I}}$

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 $(\geq n\ R)$ is:

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

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

$$(\leq n\ R)^{\mathcal{I}}(x) = \forall y_1, \ldots, y_{n+1}. \bigwedge_{i=1}^{n+1} R(x, y_i) \rightarrow \bigvee_{1 \leq i < j \leq n+1} y_i = y_j \ .$$

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

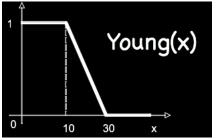
$$R^{\mathcal{I}}(x, y) = R^{\mathcal{I}}(y, x)$$

- 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

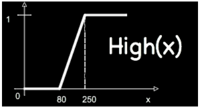
- E.g., *Small*, *Young*, *High*, *etc*. with explicit membership function
- Use the idea of concrete domains:
  - $D = \langle \Delta_D, \Phi_D \rangle$
  - Δ_D is an interpretation domain
  - Φ_D is the set of concrete fuzzy domain predicates d with a predefined arity $n = 1, 2$ and fixed interpretation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

Concepts:

		Syntax	*Semantics*
C, D	$\longrightarrow$	$\top$ |	$\top(x)$
		$\bot$ |	$\bot(x)$
		A |	$A(x)$
		$(C \sqcap D)$ |	$C_1(x) \land C_2(x)$
		$(C \sqcup D)$ |	$C_1(x) \lor C_2(x)$
		$(\neg C)$ |	$\neg C(x)$
		$(\exists R.C)$ |	$\exists x\ R(x,y) \land C(y)$
		$(\forall R.C)$ |	$\forall x\ R(x,y) \rightarrow C(y)$
		$\{a\}$ |	$x = a$
		$(\geq n\ R)$ |	$\exists y_1, \ldots, y_n. \bigwedge_{i=1}^{n} R(x, y_i) \land \bigwedge_{1 \leq i < j \leq n} y_i \neq y_j$
		$(\leq n\ R)$ |	$\forall y_1, \ldots, y_{n+1}. \bigwedge_{i=1}^{n+1} R(x, y_i) \rightarrow \bigvee_{1 \leq i < j \leq n+1} y_i = y_j$
		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	$\longrightarrow$	$\langle C \sqsubseteq D, r\rangle$ |	$\forall x\ r \rightarrow (C(x) \rightarrow D(x))$, where $\rightarrow$ is r-implication
		$fun(R)$ |	$\forall x \forall y \forall z\ R(x,y) \land R(x,z) \rightarrow y = z$
		$trans(R)$	$(\exists z\ R(x,z) \land R(z,y)) \rightarrow R(x,y)$

Example (Graded Entailment)

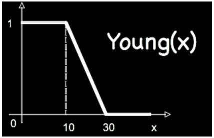
		
audi_tt	*mg*	*ferrari_enzo*

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

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

$$\begin{array}{lcl} KB & \models & \langle ferrari\_enzo{:}SportsCar, 1\rangle \\ KB & \models & \langle audi\_tt{:}SportsCar, 0.92\rangle \\ KB & \models & \langle mg{:}\neg SportsCar, 0.72\rangle \end{array}$$

Example (Graded Subsumption)

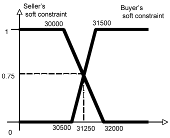


$$\begin{array}{rcl} \mathit{Minor} & = & \mathit{Person} \sqcap \exists \mathit{hasAge}. \leq_{18} \\ \mathit{YoungPerson} & = & \mathit{Person} \sqcap \exists \mathit{hasAge}.\mathit{Young} \end{array}$$

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

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

Example (Simplified 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 €

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

  - highest degree to which the concept
    $C = AudiTT \sqcap Query$
    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 €

Reasoning [19, 17, 18]

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 [20, 21, 2]
- exists for fuzzy $\mathcal{SHIF}$ + linear modifiers + fuzzy concrete concepts (implemented in fuzzyDL reasoner, but not published yet [1, 2])
- solves the BTVB as primary problem

MIQP based method: using Mixed Integer Quadratically Constrained Programming optimization problem (MICQP) for product T-norm

- exists for fuzzy $\mathcal{SHIF}$ + linear modifiers + fuzzy concrete concepts (implemented in fuzzyDL reasoner, but not published yet [1]). Important as it simulates probabilistic reasoning under independent event assumption.
- solves the BTVB as primary problem
- the fuzzyDL solver also allows to mix all three semantics

Problem with fuzzy tableau

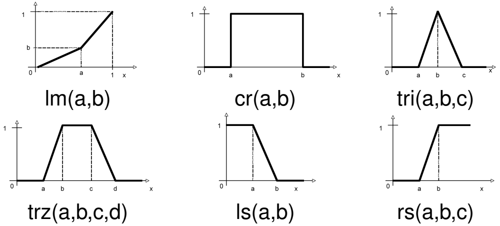
- Usual fuzzy tableaux calculus does not work (yet?) with
  - modifiers and concrete fuzzy concepts
  - Łukasiewicz Logic
  - Product T-norm
- Usual fuzzy tableaux calculus does not solve the BTVB problem
- New algorithm uses bounded Mixed Integer Programming oracle, as for Many Valued Logics
  - Recall: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

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

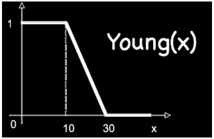
  - $|\varphi|_{KB} = \min\{x \mid KB \cup \{\langle \varphi \leq x \rangle \text{ satisfiable}\}$

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)



- 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

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

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

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

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

Example

- Suppose $KB = \left\{ \begin{array}{l} A \sqcap B \sqsubseteq C \\ \langle a{:}A \geq 0.3 \rangle \\ \langle a{:}B \geq 0.4 \rangle \end{array} \right.$

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

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: $\mathbf{x} = \mathbf{0.3}$	

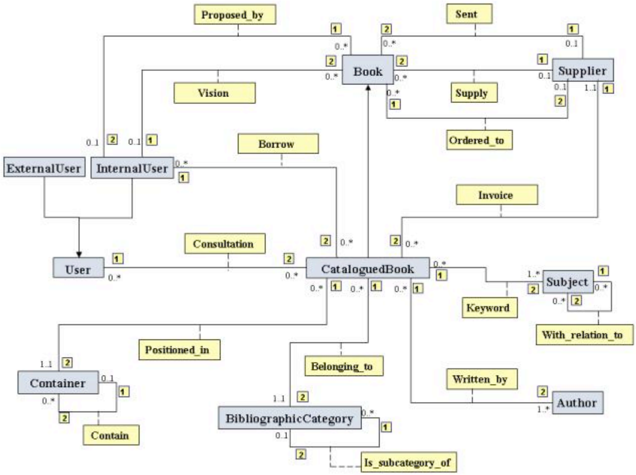
Implementation issues

- Several options exists:
  - Try to map fuzzy DLs to classical DLs
    - difficult to 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
  - Build an ad-hoc theorem prover for fuzzy DLs, using e.g., MILP
- A theorem prover for fuzzy $\mathcal{SHIF}$ + linear hedges + concrete fuzzy concepts + linear equational constraints + datatypes, under classical, Zadeh, Lukasiewicz and Product t-norm semantics has been implemented (`http://gaia.isti.cnr.it/~straccia`)
- FIRE: a fuzzy DL theorem prover for fuzzy $\mathcal{SHIN}$ under Zadeh semantics (`http://www.image.ece.ntua.gr/~nsimou/`)

Top-k retrieval in tractable DLs: the case of DL-Lite/DLR-Lite [25, 30]

- DL-Lite/DLR-Lite [3]: a simple, but interesting DLs
- 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: $KB = \langle \mathcal{T}, \mathcal{A} \rangle$, where $\mathcal{T}$ and $\mathcal{A}$ are finite sets of axioms and assertions
- Axiom: $Cl \sqsubseteq Cr$ (inclusion axiom)
- Note for inclusion axioms: the language for left hand side is different from the one for right hand side
- DL-Lite$_{core}$:
  - Concepts: $$\begin{array}{lll} Cl & \to & A \mid \exists R \\ Cr & \to & A \mid \exists R \mid \neg A \mid \neg \exists R \\ R & \to & P \mid P^{-} \end{array}$$
  - Assertion: $a{:}A$, $(a, b){:}P$
- DLR-Lite$_{core}$: (n-ary roles)
  - Concepts: $$\begin{array}{lll} Cl & \to & A \mid \exists P[i] \\ Cr & \to & A \mid \exists P[i] \mid \neg A \mid \neg \exists P[i] \end{array}$$
  - $\exists P[i]$ 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
User
Consultation
CataloguedBook
Subject
Keyword
With_relation_to
Positioned_in
Belonging_to
Container
Written_by
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/DLR-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 DL-Lite/DLR-Lite atoms $R(\mathbf{z})$ in KB;
5. $\mathbf{z}$ are tuples of constants in KB or variables in $\mathbf{x}$ or $\mathbf{y}$;
6. $\mathbf{z_i}$ are tuples of constants in KB 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 $f\colon [0, 1]^n \rightarrow [0, 1]$, 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		*HasCLoc*		*HasHPrice*	
HotelID	*HasLoc*	*ConfID*	*HasLoc*	*HotelID*	*Price*
*h*1	*hl*1	*c*1	*cl*1	*h*1	150
*h*2	*hl*2	*c*2	*cl*2	*h*2	200
⋮	⋮	⋮	⋮	⋮	⋮

$$
q(h, s) \leftarrow HasHLoc(h, hl), HasHPrice(h, p), Distance(hl, cl, d)
$$
$$
HasCLoc(c1, cl), s = cheap(p) \cdot close(d) .
$$

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

$$
\begin{array}{rcl}
close(d) & = & ls(0, 2km, d) \\
cheap(p) & = & ls(0, 300, p)
\end{array}
$$

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: $KB \models q(\mathbf{c}, v)$ iff $\mathcal{I} \models KB$ implies $\mathcal{I} \models q(\mathbf{c}, v)$
- Top-k retrieval: $ans_{top-k}(KB, q) = Top_k\{\langle \mathbf{c}, v\rangle \mid KB \models q(\mathbf{c}, v)\}$

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

- Overall strategy: three steps
  1. Check if *KB* 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

$$\mathcal{T} = \{\exists P_2^- \sqsubseteq A, A \sqsubseteq \exists P_1, B \sqsubseteq \exists P_2\}$$
$$q(x,s) \leftarrow P_2(x,y), P_1(y,z), s = \max(0, 1 - x/10)$$

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

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

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

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

Proposition

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

Tool exists and implemented in the DLMedia system

http://gaia.isti.cnr.it/~straccia

DLMedia: a Multimedia Information Retrieval System [33]

- Based on fuzzy DLR-Lite with similarity predicates
  - Axioms: $Rl_1 \sqcap \ldots \sqcap Rl_m \sqsubseteq Rr$

$$
\begin{array}{lll}
Rr & \longrightarrow & A \mid \exists[i_1, \ldots, i_k]R \\
Rl & \longrightarrow & A \mid \exists[i_1, \ldots, i_k]R \mid \exists[i_1, \ldots, i_k]R.(Cond_1 \sqcap \ldots \sqcap Cond_l) \\
Cond & \longrightarrow & ([i] \leq v) \mid ([i] < v) \mid ([i] \geq v) \mid ([i] > v) \mid ([i] = v) \mid ([i] \neq v) \mid \\
 & & ([i]\ simTxt\ 'k_1, \ldots, k_n') \mid ([i]\ simImg\ URN)
\end{array}
$$

  - $\exists[i_1, \ldots, i_k]R$ is the projection of the relation R on the columns $i_1, \ldots, i_k$
  - $\exists[i_1, \ldots, i_k]R.(Cond_1 \sqcap \ldots \sqcap Cond_l)$ further restricts the projection $\exists[i_1, \ldots, i_k]R$ according to the conditions specified in $Cond_i$
  - $([i]\ simTxt\ 'k_1 \ldots k_n')$ evaluates the degree of being the text of the i-th column similar to the list of keywords $k_1 \ldots k_n$
  - $([i]\ simImg\ URN)$ returns the system's degree of being the image identified by the i-th column similar to the image identified by the URN
  - Facts: $\langle R(c_1, \ldots, c_n), s\rangle$

- Example axioms

$$
\begin{array}{l}
\exists[1,2] Person \sqsubseteq \exists[1,2] hasAge \\
\qquad\qquad \text{// constrains relation } hasAge(name, age) \\
\exists[3,1] Person \sqsubseteq \exists[1,2] hasChild \\
\qquad\qquad \text{// constrains relation } hasChild(father\_name, name) \\
\exists[4,1] Person \sqsubseteq \exists[1,2] hasChild \\
\qquad\qquad \text{// constrains relation } hasChild(mother\_name, name) \\
\exists[3,1] Person.(([2] \geq 18) \sqcap ([5] =' female')) \sqsubseteq \exists[1,2] hasAdultDaughter \\
\qquad\qquad \text{// constrains relation } hasAdultDaughter(father\_name, name)
\end{array}
$$

- On the other hand examples axioms involving similarity predicates are,

$$\exists[1] ImageDescr.([2]\ simImg\ urn1) \sqsubseteq Child \quad (1)$$

$$\exists[1] Title.([2]\ simTxt\ 'lion') \sqsubseteq Lion \quad (2)$$

where *urn*1 identifies the image

- Example queries

$q(x) \leftarrow Child(x)$
// find objects about a child (strictly speaking, find instances of *Child*)

$q(x) \leftarrow CreatorName(x, y) \wedge (y =' paolo'), Title(x, z), (z\ simTxt\ 'tour')$
// find images made by Paolo whose title is about 'tour'

$q(x) \leftarrow ImageDescr(x, y) \wedge (y\ simImg\ urn2)$
// find images similar to a given image identified by *urn*2

$q(x) \leftarrow ImageObject(x) \wedge isAbout(x, y_1) \wedge Car(y_1) \wedge isAbout(x, y_2) \wedge Racing(y_2)$
// find image objects about cars racing

Fuzzy LPs Basics [4, 6, 7, 22, 23, 29, 35]

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

$$R(\mathbf{x}) \leftarrow \exists \mathbf{y}.f(R_1(\mathbf{z}_1), \ldots, R_l(\mathbf{z}_l), p_1(\mathbf{z}'_1), \ldots, p_h(\mathbf{z}'_h)) \ ,$$

    - Meaning of rules: “take the truth-values of all $R_i(\mathbf{z}_i), p_j(\mathbf{z}'_j)$, combine them using the truth combination function f, and assign the result to $R(\mathbf{x})$”

- Same meaning as for fuzzy DLR-Lite queries

$$R(\mathbf{x}, s) \leftarrow \exists \mathbf{y}.conj(\mathbf{x}, \mathbf{y}), s = f(p_1(\mathbf{z}_1), \ldots, p_{l+h}(\mathbf{z}_{l+h}))$$

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 list of atoms $R_i(\mathbf{z})$ in KB;
5. $\mathbf{z}$ are tuples of constants in KB or variables in $\mathbf{x}$ or $\mathbf{y}$;
6. $\mathbf{z_i}$ are tuples of constants in KB 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 $f\colon [0, 1]^{l+h} \to [0, 1]$, which combines the scores of the n fuzzy predicates $p_i(\mathbf{c}_i)$

Semantics of fuzzy LPs

- Model of a LP:

$$\begin{array}{lll} I \models \mathcal{P} & \text{iff} & I \models r, \text{ for all } r \in \mathcal{P}^* \\ I \models A \leftarrow \varphi & \text{iff} & I(\varphi) \leq I(A) \end{array}$$

- Least model exists and is least fixed-point of

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

  for all $A \leftarrow \varphi \in \mathcal{P}^*$
- Fuzzy LPs may be tricky:

$$\begin{array}{lll} \langle A, 0 \rangle & & \\ A & \leftarrow & (A + 1)/2 \end{array}$$

  In the minimal model the truth of A is 1 (requires ω $T_{\mathcal{P}}$ iterations)!
- There are several ways to avoid this pathological behavior:
  - We consider $\mathcal{T} = \{0, \frac{1}{n}, \frac{2}{n} \ldots, \frac{n-1}{n}, 1\}$, n natural number, e.g. $n = 100$

Example: Soft shopping agent

- I may represent my preferences in Logic Programming with the rules

$$
\begin{array}{rcl}
Pref_1(x,p,s) & \leftarrow & HasPrice(x,p), LS(10000, 14000, p, s) \\
Pref_2(x,s) & \leftarrow & HasKM(x,k), LS(13000, 17000, k, s) \\
Buy(x,p,s) & \leftarrow & Pref_1(x,p,s_1), Pref_2(x,s_2), s = 0.7 \cdot s_1 + 0.3 \cdot s_2
\end{array}
$$

ID	MODEL	PRICE	KM
455	MAZDA 3	12500	10000
34	ALFA 156	12000	15000
1812	FORD FOCUS	11000	16000
⋮	⋮	⋮	⋮

- Problem: All tuples of the database have a score:
  - We cannot compute the score of all tuples, then rank them. Brute force approach not feasible.
- Top-k problem: Determine efficiently just the top-k ranked tuples, without evaluating the score of all tuples. E.g. top-3 tuples

ID	PRICE	SCORE
1812	11000	0.6
455	12500	0.56
34	12000	0.50

Top-k retrieval in LPs

- If the database contains a huge amount of facts, a brute force approach fails:
  - one cannot anymore compute the score of all tuples, rank all of them and only then return the top-k
- Better solutions exists for restricted fuzzy LP languages: Datalog + restriction on the score combination functions appearing in the body [29, 32]

Basic Idea

- We do not compute all answers, but determine answers incrementally
- At each step *i*, from the tuples seen so far in the database, we compute a threshold δ
- The threshold δ has the property that any successively retrieved answer will have a score $s \leq \delta$
- Therefore, we can stop as soon as we have gathered *k* answers above δ, because any successively computed answer will have a score below δ

Procedure *TopAnswers*($\mathcal{K}$, Q, k)
Input: KB $\mathcal{K}$, intensional query relation symbol Q, $k \geq 1$;
Output: Mapping `rankedList` such that `rankedList`(Q) contains top-k answers of Q
Init: $\delta = 1$, **for all** rules $r : P(\mathbf{x}) \leftarrow \phi$ in P **do**
    if P intensional **then** `rankedList`(P) $= \emptyset$;
    if P extensional **then** `rankedList`(P) $= T_P$ **endfor**

1. **loop**
2. `Active` $:= \{Q\}$, `dg` $:= \{Q\}$, `in` $:= \emptyset$,
    for all rules $r : P(\mathbf{x}) \leftarrow \phi$ **do** `exp`(P, r) $=$ `false`;
3. **while** (`Active` $\neq \emptyset$) **do**
4. **select** $P \in$ `A` where $r : P(\mathbf{x}) \leftarrow \phi$, `Active` $:=$ `Active` $\setminus \{P\}$, `dg` $:=$ `dg` $\cup$ `s`(P, r);
5. $\langle \mathbf{t}, s \rangle :=$ *getNextTuple*(P, r)
6. **if** $\langle \mathbf{t}, s \rangle \neq$ `NULL` **then** insert $\langle \mathbf{t}, s \rangle$ into `rankedList`(P),
    `Active` $:=$ `Active` $\cup$ (`p`(P) $\cap$ `dg`);
7. **if not** `exp`(P, r) **then** `exp`(P, r) $=$ `true`,
    `Active` $:=$ `Active` $\cup$ (`s`(P, r) $\setminus$ `in`), `in` $:=$ `in` $\cup$ `s`(p, r);
   endwhile
8. Update threshold δ;
9. **until** (`rankedList`(Q) does contain k top-ranked tuples with score above δ)
    or (`rL`$'$ $=$ `rankedList`);
10. **return** top-k ranked tuples in `rankedList`(Q);

Procedure *getNextTuple*(P, r)
Input: intensional relation symbol P and rule $r : P(\mathbf{x}) \leftarrow \exists \mathbf{y}.f(R_1(\mathbf{z}_1), \ldots, R_n(\mathbf{z}_l)) \in P$;
Output: Next tuple satisfying the body of the r together with the score
Init:

```
      loop
1.        Generate next new instance tuple ⟨t, s⟩ of P, using tuples in rankedList(R_i) and RankSQL
2.        if there is no ⟨t, s'⟩ ∈ rankedList(P, r) with s ≤ s' then exit loop
      until no new valid join tuple can be generated
3.    return ⟨t, s⟩ if it exists else return NULL
```

Example

Logic Program $\mathcal{P}$ is

$$q(x,s) \leftarrow p(x,s_1), s = s_1$$
$$p(x,s) \leftarrow r_1(x,y,s_1), r_2(y,z,s_2), s = \min(s_1,s_2)$$

RecordID	r_1			r_2		
1	*a*	*b*	1.0	*m*	*h*	0.95
2	*c*	*d*	0.9	*m*	*j*	0.85
3	*e*	*f*	0.8	*f*	*k*	0.75
4	*l*	*m*	0.7	*m*	*n*	0.65
5	*o*	*p*	0.6	*p*	*q*	0.55
⋮	⋮	⋮	⋮	⋮	⋮	⋮

What is

$$Top_1(\mathcal{P},q) = Top_1\{\langle c,s\rangle \mid \mathcal{P} \models q(c,s)\} \;\; ?$$

$$q(x, s) \leftarrow p(x, s_1), s = s_1$$
$$p(x, s) \leftarrow r_1(x, y, s_1), r_2(y, z, s_2), s = \min(s_1, s_2)$$

RecordID	r_1			r_2		
1	*a*	*b*	1.0	*m*	*h*	0.95
2	*c*	*d*	0.9	*m*	*j*	0.85
3	*e*	*f*	0.8	*f*	*k*	0.75
4	*l*	*m*	0.7	*m*	*n*	0.65
5	*o*	*p*	0.6	*p*	*q*	0.55
⋮	⋮	⋮	⋮	⋮	⋮	⋮

Queue	δ
q	1

Predicate	*Answers*
q	$\emptyset$
p	$\emptyset$

$$q(x, s) \leftarrow p(x, s_1), s = s_1$$
$$p(x, s) \leftarrow r_1(x, y, s_1), r_2(y, z, s_2), s = \min(s_1, s_2)$$

RecordID	r_1			r_2		
1	a	b	1.0	m	h	0.95
2	c	d	0.9	m	j	0.85
3	e	f	0.8	f	k	0.75
4	l	m	0.7	m	n	0.65
5	o	p	0.6	p	q	0.55
⋮	⋮	⋮	⋮	⋮	⋮	⋮

Action: select next predicate in queue

Queue	δ
q	1

Predicate	*Answers*
q	$\emptyset$
p	$\emptyset$

$q(x, s) \leftarrow p(x, s_1), s = s_1$
$p(x, s) \leftarrow r_1(x, y, s_1), r_2(y, z, s_2), s = \min(s_1, s_2)$

RecordID	r_1			r_2		
1	a	b	1.0	m	h	0.95
2	c	d	0.9	m	j	0.85
3	e	f	0.8	f	k	0.75
4	l	m	0.7	m	n	0.65
5	o	p	0.6	p	q	0.55
⋮	⋮	⋮	⋮	⋮	⋮	⋮

Action: get next tuple for q

Queue	δ
–	1

Predicate	Answers
q	$\emptyset$
p	$\emptyset$

$$q(x, s) \leftarrow p(x, s_1), s = s_1$$
$$p(x, s) \leftarrow r_1(x, y, s_1), r_2(y, z, s_2), s = \min(s_1, s_2)$$

RecordID	r_1			r_2		
1	*a*	*b*	1.0	*m*	*h*	0.95
2	*c*	*d*	0.9	*m*	*j*	0.85
3	*e*	*f*	0.8	*f*	*k*	0.75
4	*l*	*m*	0.7	*m*	*n*	0.65
5	*o*	*p*	0.6	*p*	*q*	0.55
⋮	⋮	⋮	⋮	⋮	⋮	⋮

Action: no answer yet for q, put all predicates in rule body of q in queue

Queue	δ
p	1

Predicate	*Answers*
q	$\emptyset$
p	$\emptyset$

$$q(x, s) \leftarrow p(x, s_1), s = s_1$$
$$p(x, s) \leftarrow r_1(x, y, s_1), r_2(y, z, s_2), s = \min(s_1, s_2)$$

RecordID	r_1			r_2		
1	*a*	*b*	1.0	*m*	*h*	0.95
2	*c*	*d*	0.9	*m*	*j*	0.85
3	*e*	*f*	0.8	*f*	*k*	0.75
4	*l*	*m*	0.7	*m*	*n*	0.65
5	*o*	*p*	0.6	*p*	*q*	0.55
⋮	⋮	⋮	⋮	⋮	⋮	⋮

Action: select next predicate in queue

Queue	δ
p	1

Predicate	*Answers*
q	$\emptyset$
p	$\emptyset$

$q(x, s) \leftarrow p(x, s_1), s = s_1$
$p(x, s) \leftarrow r_1(x, y, s_1), r_2(y, z, s_2), s = \min(s_1, s_2)$

RecordID	r_1			r_2		
1	*a*	*b*	1.0	*m*	*h*	0.95
2	*c*	*d*	0.9	*m*	*j*	0.85
3	*e*	*f*	0.8	*f*	*k*	0.75
4	*l*	*m*	0.7	*m*	*n*	0.65
5	*o*	*p*	0.6	*p*	*q*	0.55
⋮	⋮	⋮	⋮	⋮	⋮	⋮

Action: get next tuple for *p*

Queue	δ
–	1

Predicate	*Answers*
q	$\emptyset$
p	$\emptyset$

$$q(x, s) \leftarrow p(x, s_1), s = s_1$$
$$p(x, s) \leftarrow r_1(x, y, s_1), r_2(y, z, s_2), s = \min(s_1, s_2)$$

	RecordID	r_1			r_2			
→	1	*a*	*b*	1.0	*m*	*h*	0.95	←
	2	*c*	*d*	0.9	*m*	*j*	0.85	
	3	*e*	*f*	0.8	*f*	*k*	0.75	
	4	*l*	*m*	0.7	*m*	*n*	0.65	
	5	*o*	*p*	0.6	*p*	*q*	0.55	
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	

Action: get next tuple for *p*

Queue	δ
–	1

Predicate	*Answers*
q	$\emptyset$
p	$\emptyset$

$$q(x, s) \leftarrow p(x, s_1), s = s_1$$
$$p(x, s) \leftarrow r_1(x, y, s_1), r_2(y, z, s_2), s = \min(s_1, s_2)$$

	RecordID	r_1			r_2			
	1	*a*	*b*	1.0	*m*	*h*	0.95	←
→	2	*c*	*d*	0.9	*m*	*j*	0.85	
	3	*e*	*f*	0.8	*f*	*k*	0.75	
	4	*l*	*m*	0.7	*m*	*n*	0.65	
	5	*o*	*p*	0.6	*p*	*q*	0.55	
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	

Action: get next tuple for *p*

Queue	δ
–	1

Predicate	*Answers*
q	$\emptyset$
p	$\emptyset$

$$q(x, s) \leftarrow p(x, s_1), s = s_1$$
$$p(x, s) \leftarrow r_1(x, y, s_1), r_2(y, z, s_2), s = \min(s_1, s_2)$$

	RecordID	r_1			r_2			
	1	*a*	*b*	1.0	*m*	*h*	0.95	
→	2	*c*	*d*	0.9	*m*	*j*	0.85	←
	3	*e*	*f*	0.8	*f*	*k*	0.75	
	4	*l*	*m*	0.7	*m*	*n*	0.65	
	5	*o*	*p*	0.6	*p*	*q*	0.55	
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	

Action: get next tuple for *p*

Queue	δ
–	1

Predicate	*Answers*
q	$\emptyset$
p	$\emptyset$

$$q(x, s) \leftarrow p(x, s_1), s = s_1$$
$$p(x, s) \leftarrow r_1(x, y, s_1), r_2(y, z, s_2), s = \min(s_1, s_2)$$

	RecordID	r_1			r_2			
	1	*a*	*b*	1.0	*m*	*h*	0.95	
	2	*c*	*d*	0.9	*m*	*j*	0.85	←
→	3	*e*	*f*	0.8	*f*	*k*	0.75	
	4	*l*	*m*	0.7	*m*	*n*	0.65	
	5	*o*	*p*	0.6	*p*	*q*	0.55	
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	

Action: get next tuple for *p*

Queue	δ
–	1

Predicate	*Answers*
q	$\emptyset$
p	$\emptyset$

$$q(x, s) \leftarrow p(x, s_1), s = s_1$$
$$p(x, s) \leftarrow r_1(x, y, s_1), r_2(y, z, s_2), s = \min(s_1, s_2)$$

	RecordID	r_1			r_2			
	1	*a*	*b*	1.0	*m*	*h*	0.95	
	2	*c*	*d*	0.9	*m*	*j*	0.85	
→	3	*e*	*f*	0.8	*f*	*k*	0.75	←
	4	*l*	*m*	0.7	*m*	*n*	0.65	
	5	*o*	*p*	0.6	*p*	*q*	0.55	
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	

Action: get next tuple for *p*

Queue	δ
—	1

Predicate	*Answers*
q	$\emptyset$
p	$\emptyset$

$$q(x, s) \leftarrow p(x, s_1), s = s_1$$
$$p(x, s) \leftarrow r_1(x, y, s_1), r_2(y, z, s_2), s = \min(s_1, s_2)$$

	RecordID	r_1			r_2			
	1	*a*	*b*	1.0	*m*	*h*	0.95	
	2	*c*	*d*	0.9	*m*	*j*	0.85	
→	3	*e*	*f*	0.8	*f*	*k*	0.75	←
	4	*l*	*m*	0.7	*m*	*n*	0.65	
	5	*o*	*p*	0.6	*p*	*q*	0.55	
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	

Action: get next tuple for *p*

Queue	δ
—	1

Predicate	*Answers*
q	$\emptyset$
p	$\langle e, 0.75\rangle$

$$q(x, s) \leftarrow p(x, s_1), s = s_1$$
$$p(x, s) \leftarrow r_1(x, y, s_1), r_2(y, z, s_2), s = \min(s_1, s_2)$$

	RecordID	r_1			r_2			
	1	*a*	*b*	1.0	*m*	*h*	0.95	
	2	*c*	*d*	0.9	*m*	*j*	0.85	
$\rightarrow$	3	*e*	*f*	0.8	*f*	*k*	0.75	$\leftarrow$
	4	*l*	*m*	0.7	*m*	*n*	0.65	
	5	*o*	*p*	0.6	*p*	*q*	0.55	
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	

Action: as *p* changed, update queue with predicates depending on *p*

Queue	δ
q	1

Predicate	*Answers*
q	$\emptyset$
p	$\langle e, 0.75\rangle$

$$q(x, s) \leftarrow p(x, s_1), s = s_1$$
$$p(x, s) \leftarrow r_1(x, y, s_1), r_2(y, z, s_2), s = \min(s_1, s_2)$$

	RecordID	r_1			r_2			
	1	*a*	*b*	1.0	*m*	*h*	0.95	
	2	*c*	*d*	0.9	*m*	*j*	0.85	
→	3	*e*	*f*	0.8	*f*	*k*	0.75	←
	4	*l*	*m*	0.7	*m*	*n*	0.65	
	5	*o*	*p*	0.6	*p*	*q*	0.55	
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	

Action: select next predicate from queue, and get next tuple for it

Queue	δ
q	1

Predicate	*Answers*
q	$\langle e, 0.75\rangle$
p	$\langle e, 0.75\rangle$

$$q(x,s) \leftarrow p(x,s_1), s = s_1$$
$$p(x,s) \leftarrow r_1(x,y,s_1), r_2(y,z,s_2), s = \min(s_1,s_2)$$

	RecordID	r_1			r_2			
	1	*a*	*b*	1.0	*m*	*h*	0.95	
	2	*c*	*d*	0.9	*m*	*j*	0.85	
→	3	*e*	*f*	0.8	*f*	*k*	0.75	←
	4	*l*	*m*	0.7	*m*	*n*	0.65	
	5	*o*	*p*	0.6	*p*	*q*	0.55	
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	

Action: as *q* changed, update queue with predicates depending on *q*

Queue	δ
–	1

Predicate	*Answers*
q	$\langle e, 0.75\rangle$
p	$\langle e, 0.75\rangle$

$$q(x, s) \leftarrow p(x, s_1), s = s_1$$
$$p(x, s) \leftarrow r_1(x, y, s_1), r_2(y, z, s_2), s = \min(s_1, s_2)$$

	RecordID	r_1			r_2			
	1	*a*	*b*	1.0	*m*	*h*	0.95	
	2	*c*	*d*	0.9	*m*	*j*	0.85	
→	3	*e*	*f*	0.8	*f*	*k*	0.75	←
	4	*l*	*m*	0.7	*m*	*n*	0.65	
	5	*o*	*p*	0.6	*p*	*q*	0.55	
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	

Action: queue empty, so update threshold and re-start $(0.8 = \max(\min(1.0, 0.75), \min(0.8, 0.95))$

Queue	δ
–	0.8

Predicate	*Answers*
q	$\langle e, 0.75\rangle$
p	$\langle e, 0.75\rangle$

$$q(x,s) \leftarrow p(x,s_1), s = s_1$$
$$p(x,s) \leftarrow r_1(x,y,s_1), r_2(y,z,s_2), s = \min(s_1,s_2)$$

	RecordID	r_1			r_2			
	1	*a*	*b*	1.0	*m*	*h*	0.95	
	2	*c*	*d*	0.9	*m*	*j*	0.85	
→	3	*e*	*f*	0.8	*f*	*k*	0.75	←
	4	*l*	*m*	0.7	*m*	*n*	0.65	
	5	*o*	*p*	0.6	*p*	*q*	0.55	
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	

Action: select next element from queue and get next tuple

Queue	δ
q	0.8

Predicate	*Answers*
q	$\langle e, 0.75\rangle$
p	$\langle e, 0.75\rangle$

$$q(x, s) \leftarrow p(x, s_1), s = s_1$$
$$p(x, s) \leftarrow r_1(x, y, s_1), r_2(y, z, s_2), s = \min(s_1, s_2)$$

	RecordID	r_1			r_2			
	1	*a*	*b*	1.0	*m*	*h*	0.95	
	2	*c*	*d*	0.9	*m*	*j*	0.85	
→	3	*e*	*f*	0.8	*f*	*k*	0.75	←
	4	*l*	*m*	0.7	*m*	*n*	0.65	
	5	*o*	*p*	0.6	*p*	*q*	0.55	
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	

Action: no new tuple for *q*, so expand *q* rule

Queue	δ
p	0.8

Predicate	*Answers*
q	$\langle e, 0.75\rangle$
p	$\langle e, 0.75\rangle$

$$q(x, s) \leftarrow p(x, s_1), s = s_1$$
$$p(x, s) \leftarrow r_1(x, y, s_1), r_2(y, z, s_2), s = \min(s_1, s_2)$$

	RecordID	r_1			r_2			
	1	*a*	*b*	1.0	*m*	*h*	0.95	
	2	*c*	*d*	0.9	*m*	*j*	0.85	
→	3	*e*	*f*	0.8	*f*	*k*	0.75	←
	4	*l*	*m*	0.7	*m*	*n*	0.65	
	5	*o*	*p*	0.6	*p*	*q*	0.55	
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	

Action: select next element from queue and get next tuple

Queue	δ
p	0.8

Predicate	*Answers*
q	$\langle e, 0.75\rangle$
p	$\langle e, 0.75\rangle$

$$q(x, s) \leftarrow p(x, s_1), s = s_1$$
$$p(x, s) \leftarrow r_1(x, y, s_1), r_2(y, z, s_2), s = \min(s_1, s_2)$$

	RecordID	r_1			r_2			
	1	*a*	*b*	1.0	*m*	*h*	0.95	
	2	*c*	*d*	0.9	*m*	*j*	0.85	
	3	*e*	*f*	0.8	*f*	*k*	0.75	←
→	4	*l*	*m*	0.7	*m*	*n*	0.65	
	5	*o*	*p*	0.6	*p*	*q*	0.55	
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	

Action: get next tuple for p

Queue	δ
—	0.8

Predicate	*Answers*
q	$\langle e, 0.75\rangle$
p	$\langle e, 0.75\rangle$

$q(x, s) \leftarrow p(x, s_1), s = s_1$
$p(x, s) \leftarrow r_1(x, y, s_1), r_2(y, z, s_2), s = \min(s_1, s_2)$

	RecordID	r_1			r_2			
	1	*a*	*b*	1.0	*m*	*h*	0.95	
	2	*c*	*d*	0.9	*m*	*j*	0.85	
	3	*e*	*f*	0.8	*f*	*k*	0.75	←
→	4	*l*	*m*	0.7	*m*	*n*	0.65	
	5	*o*	*p*	0.6	*p*	*q*	0.55	
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	

Action: get next tuple for p

Queue	δ
–	0.8

Predicate	*Answers*
q	$\langle e, 0.75\rangle$
p	$\langle e, 0.75\rangle$, $\langle l, 0.7\rangle$

$$q(x,s) \leftarrow p(x,s_1), s = s_1$$
$$p(x,s) \leftarrow r_1(x,y,s_1), r_2(y,z,s_2), s = \min(s_1,s_2)$$

	RecordID	r_1			r_2			
	1	*a*	*b*	1.0	*m*	*h*	0.95	
	2	*c*	*d*	0.9	*m*	*j*	0.85	
	3	*e*	*f*	0.8	*f*	*k*	0.75	←
→	4	*l*	*m*	0.7	*m*	*n*	0.65	
	5	*o*	*p*	0.6	*p*	*q*	0.55	
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	

Action: *p* changed, so put *q* in queue and get next tuple for it

Queue	δ
q	1

Predicate	*Answers*
q	$\langle e, 0.75\rangle \langle l, 0.7\rangle$
p	$\langle e, 0.75\rangle, \langle l, 0.7\rangle$

$$q(x, s) \leftarrow p(x, s_1), s = s_1$$
$$p(x, s) \leftarrow r_1(x, y, s_1), r_2(y, z, s_2), s = \min(s_1, s_2)$$

	RecordID	r_1			r_2			
	1	*a*	*b*	1.0	*m*	*h*	0.95	
	2	*c*	*d*	0.9	*m*	*j*	0.85	
	3	*e*	*f*	0.8	*f*	*k*	0.75	←
→	4	*l*	*m*	0.7	*m*	*n*	0.65	
	5	*o*	*p*	0.6	*p*	*q*	0.55	
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	

Action: q changed, put predicates depending on q in queue

Queue	δ
–	0.8

Predicate	*Answers*
q	$\langle e, 0.75\rangle \langle l, 0.7\rangle$
p	$\langle e, 0.75\rangle, \langle l, 0.7\rangle$

$$q(x, s) \leftarrow p(x, s_1), s = s_1$$
$$p(x, s) \leftarrow r_1(x, y, s_1), r_2(y, z, s_2), s = \min(s_1, s_2)$$

	RecordID	r_1			r_2			
	1	*a*	*b*	1.0	*m*	*h*	0.95	
	2	*c*	*d*	0.9	*m*	*j*	0.85	
	3	*e*	*f*	0.8	*f*	*k*	0.75	←
→	4	*l*	*m*	0.7	*m*	*n*	0.65	
	5	*o*	*p*	0.6	*p*	*q*	0.55	
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	

Action: queue empty, update threshold

Queue	δ
–	0.75

Predicate	*Answers*
q	$\langle e, 0.75\rangle \langle l, 0.7\rangle$
p	$\langle e, 0.75\rangle, \langle l, 0.7\rangle$

$$q(x,s) \leftarrow p(x,s_1), s = s_1$$
$$p(x,s) \leftarrow r_1(x,y,s_1), r_2(y,z,s_2), s = \min(s_1,s_2)$$

	RecordID	r_1			r_2			
	1	*a*	*b*	1.0	*m*	*h*	0.95	
	2	*c*	*d*	0.9	*m*	*j*	0.85	
	3	*e*	*f*	0.8	*f*	*k*	0.75	←
→	4	*l*	*m*	0.7	*m*	*n*	0.65	
	5	*o*	*p*	0.6	*p*	*q*	0.55	
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	

Action: STOP, top-1 tuple score is equal or above threshold

Queue	δ
–	0.75

Predicate	*Answers*
q	$\langle e, 0.75\rangle \langle l, 0.7\rangle$
p	$\langle e, 0.75\rangle, \langle l, 0.7\rangle$

$$Top_1(\mathcal{P}, q) = \{\langle e, 0.75\rangle\}$$

Note: no further answer will have score above threshold δ

Fuzzy DLPs Basics [1, 2, 7, 8]

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

$$\begin{array}{lcl} \mathit{LowCarPrice}(z) & \leftarrow & \min(\mathit{made\_by}(x, y), \mathit{DL}[\mathit{ChineseCarCompany}](y) \\ & & \mathit{price}(x, z)) \cdot \mathit{DL}[\mathit{Low}](z) \\ \\ \mathit{Low} & = & \mathit{LS}(5.000, 15.000) \\ \mathit{ChineseCarCompany} & \sqsubseteq & \exists \mathit{has\_location}.\mathit{China} \end{array}$$

- 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: several approaches
- In principle, for each classical semantics based integration between DLs and LPs, there is be a fuzzy analogue
  - Pay attention, the fuzzy variant may add further technical and computational complications

1. Axiomatic approach: fuzzy DL atoms and roles are managed uniformely
2. Loosely Coupled approach: fuzzy DL atoms and roles are like “procedural attachments” (procedural calls to a fuzzy DL theorem prover)
3. Tightly coupled approach: The DL component restricts the models to be considered for the LP component

Axiomatic approach

- Formally easy
  - I is a model of $KB = \langle \mathcal{P}, \Sigma \rangle$ iff $I \models \mathcal{P}$ and $I \models \Sigma$
- To guarantee decidability, e.g.
  - DL-safe rules +
  - Fuzzy LP component has to be decidable
- Decision algorithm: No algorithm exists yet. Though
  - A mapping from fuzzy OWL-DL to fuzzy disjunctive LPs is possible
    - Depends on the semantics and features of the fuzzy DL component (t-norm, fuzzy concrete domains, ...)
    - Depends on the semantics for the fuzzy disjunctive LP component (e.g., [1, 4, 5, 6])
    - The fuzzy LP semantics has to support the fuzzy DL component semantics
  - However, a tractable (data complexity) top-k algorithm exists for fuzzy DLR-Lite + fuzzy LPs under the axiomatic approach (submitted)

Loosely coupled approach [1, 6, 8, 7]

- 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
  - $I^{\Sigma}(DL[A](a)) = glb(\Sigma, a{:}A)$ for all ground DL atoms $DL[A](a)$
  - $I^{\Sigma}(DL[R](a, b)) = glb(\Sigma, (a, b){:}R)$ for all ground DL roles $DL[R](a, b)$
- Minimal model property of fuzzy LPs and a fixed-point characterization:

$$T_{\mathcal{P}}(I)(A) = I^{\Sigma}(\varphi), \text{ for } A \leftarrow \varphi \in \mathcal{P}^*$$

- An approach using non-monotone negation is described in [1]

A top-down procedure (without non-monotonicity)

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 $DL[A](a)$ (similarly for roles)
    - compute $\bar{x}_{i_j} = glb(KB, a{:}A)$
    - set $v(x_{i_j}) = \bar{x}_{i_j}$, instead of $v(x_{i_j}) = 0$

Essentially, for all DL-atoms $DL[A](a)$ we compute off-line $glb(KB, a{:}A)$ and add then the rule $A(a) \leftarrow glb(KB, a{:}A)$ to $\mathcal{P}$

Tightly coupled approach [2]

- DL atoms may appear anywhere in the rule

$$a_1 \vee_{\oplus_1} \cdots \vee_{\oplus_{l-1}} a_l \leftarrow_{\otimes_0} b_1 \wedge_{\otimes_1} b_2 \wedge_{\otimes_2} \cdots \wedge_{\otimes_{k-1}} b_k \geq v$$

- For instance,

$$\begin{array}{lll} \mathit{query}(x) & \leftarrow_{\otimes} & \mathit{SportyCar}(x) \wedge_{\otimes} \mathit{hasInvoice}(x, y_1) \wedge_{\otimes} \mathit{hasHorsePower}(x, y_2) \wedge_{\otimes} \\ & & \mathit{LeqAbout22000}(y_1) \wedge_{\otimes} \mathit{Around150}(y_2) \geq 1\,. \end{array}$$

Semantics

- Consider $KB = \langle \mathcal{P}, \Sigma \rangle$
- interpretation $I \colon HB_\Phi \to [0, 1]$
- $I \models r$ iff

$$I(a_1) \oplus_1 \cdots \oplus_l I(a_l) \geq I(b_1) \otimes_1 \cdots \otimes_{k-1} I(b_k) \otimes_0 v\,.$$

- $I \models \mathcal{P}$ iff $I \models r$ for all $r \in \mathcal{P}^*$
- $I \models \Sigma$ iff $\Sigma \cup \{a = I(a) \mid a \in HB_\Phi\}$ is satisfiable
- $I \models KB$ iff $I \models \mathcal{P}$ and $I \models \Sigma$
- The extension to non-monotone negation and a decision procedure is described in [2, 3]
    - Requires a decision procedure for the fuzzy DL component

F. Bobillo and U. Straccia.
A fuzzy description logic with product t-norm.
In *Proceedings of the IEEE International Conference on Fuzzy Systems (Fuzz-IEEE-07)*. IEEE Computer Society, 2007.

F. Bobillo and U. Straccia.
Mixed integer programming, general concept inclusions and fuzzy description logics.
In *Proceedings of the 5th Conference of the European Society for Fuzzy Logic and Technology (EUSFLAT-07)*, 2007.

D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati.
Data complexity of query answering in description logics.
In *Proceedings of the Tenth International Conference on Principles of Knowledge Representation and Reasoning (KR-06)*, pages 260–270, 2006.

C. Damásio and M. Medina, J. Ojeda-Aciego.
A tabulation procedure for first-order residuated logic programs.
In *Proceedings of the 11th International Conference on Information Processing and Managment of Uncertainty in Knowledge-Based Systems, (IPMU-06)*, 2006.

P. Hájek.
Metamathematics of Fuzzy Logic.
Kluwer, 1998.

M. Kifer and V. Subrahmanian.
Theory of generalized annotated logic programming and its applications.
Journal of Logic Programming, 12:335–367, 1992.

L. V. Lakshmanan and N. Shiri.

A parametric approach to deductive databases with uncertainty.
IEEE Transactions on Knowledge and Data Engineering, 13(4):554–570, 2001.

C. Li, M. Soliman, K. Chang, and I. Ilyas.
RankSQL: Supporting ranking queries in relational database management systems, 2005.

Y. Loyer and U. Straccia.
Any-world assumptions in logic programming.
Theoretical Computer Science, 342(2-3):351–381, 2005.

T. Lukasiewicz.
Fuzzy description logic programs under the answer set semantics for the semanticweb.
In *Second International Conference on Rules and Rule Markup Languages for the Semantic Web (RuleML-06)*. IEEE, 2006.

T. Lukasiewicz and U. Straccia.
Tightly integrated fuzzy description logic programs under the answer semantics for the semantic web.
In *Proceedings of the First International Conference on Web Reasoning and Rule Systems (RR-07)*, number 4524 in Lecture Notes in Computer Science, pages 289—298. Springer Verlag, 2007.

T. Lukasiewicz and U. Straccia.
Tightly integrated fuzzy description logic programs under the answer semantics for the semantic web.
INFSYS RESEARCH REPORT 1843-07-03, INSTITUT FüR INFORMATIONSSYSTEME ARBEITSBEREICH WISSENSBASIERTE SYSTEME, Technische Universität Wien, 2007.

C. Mateis.
Extending disjunctive logic programming by t-norms.
In *Proceedings of the 5th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR-99)*, number 1730 in Lecture Notes in Computer Science, pages 290–304. Springer-Verlag, 1999.

C. Mateis.
Quantitative disjunctive logic programming: Semantics and computation.
AI Communications, 13:225–248, 2000.

M. Mazzieri.
A fuzzy rdf semantics to represent trust metadata.
In *Proceedings of the 1st Italian Semantic Web Workshop: Semantic Web Applications and Perspectives (SWAP 2004)*, 2004.

M. Mazzieri and A. F. Dragoni.
A fuzzy semantics for semantic web languages.
In *Proceedings of the Proceedings of the ISWC Workshop on Uncertainty Reasoning for the Semantic Web (URSW-05)*. CEUR Workshop Proceedings, 2005.

G. Stoilos, G. Stamou, and J. Z. Pan.
D2.5.6: Fuzzy reasoning extensions.
Knowledge Web Technical Report, 2007.

G. Stoilos, U. Straccia, G. Stamou, and J. Pan.
General concept inclusions in fuzzy description logics.
In *Proceedings of the 17th Eureopean Conference on Artificial Intelligence (ECAI-06)*, pages 457–461. IOS Press, 2006.

U. Straccia.
Reasoning within fuzzy description logics.
Journal of Artificial Intelligence Research, 14:137–166, 2001.

U. Straccia.
Description logics with fuzzy concrete domains.

In F. Bachus and T. Jaakkola, editors, *21st Conference on Uncertainty in Artificial Intelligence (UAI-05)*, pages 559–567, Edinburgh, Scotland, 2005. AUAI Press.

U. Straccia.
Fuzzy alc with fuzzy concrete domains.
In *Proceeedings of the International Workshop on Description Logics (DL-05)*, pages 96–103, Edinburgh, Scotland, 2005. CEUR.

U. Straccia.
Query answering in normal logic programs under uncertainty.
In *8th European Conferences on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU-05)*, number 3571 in Lecture Notes in Computer Science, pages 687–700, Barcelona, Spain, 2005. Springer Verlag.

U. Straccia.
Uncertainty management in logic programming: Simple and effective top-down query answering.
In R. Khosla, R. J. Howlett, and L. C. Jain, editors, *9th International Conference on Knowledge-Based & Intelligent Information & Engineering Systems (KES-05), Part II*, number 3682 in Lecture Notes in Computer Science, pages 753–760, Melbourne, Australia, 2005. Springer Verlag.

U. Straccia.
Annotated answer set programming.
In *Proceedings of the 11th International Conference on Information Processing and Managment of Uncertainty in Knowledge-Based Systems, (IPMU-06)*, pages 1212–1219. E.D.K., Paris, 2006.

U. Straccia.
Answering vague queries in fuzzy dl-lite.
In *Proceedings of the 11th International Conference on Information Processing and Managment of Uncertainty in Knowledge-Based Systems, (IPMU-06)*, pages 2238–2245. E.D.K., Paris, 2006.

U. Straccia.
A fuzzy description logic for the semantic web.
In E. Sanchez, editor, *Fuzzy Logic and the Semantic Web*, Capturing Intelligence, chapter 4, pages 73–90. Elsevier, 2006.

U. Straccia.
Fuzzy description logic programs.
In *Proceedings of the 11th International Conference on Information Processing and Managment of Uncertainty in Knowledge-Based Systems, (IPMU-06)*, pages 1818–1825. E.D.K., Paris, 2006.

U. Straccia.
Query answering under the any-world assumption for normal logic programs.
In *Proceedings of the 10th International Conference on Principles of Knowledge Representation (KR-06)*, pages 329–339. AAAI Press, 2006.

U. Straccia.
Towards top-k query answering in deductive databases.
In *Proceedings of the 2006 IEEE International Conference on Systems, Man and Cybernetics (SMC-06)*, pages 4873–4879. IEEE, 2006.

U. Straccia.
Towards top-k query answering in description logics: the case of DL-Lite.
In *Proceedings of the 10th European Conference on Logics in Artificial Intelligence (JELIA-06)*, number 4160 in Lecture Notes in Computer Science, pages 439–451, Liverpool, UK, 2006. Springer Verlag.

U. Straccia.
Uncertainty and description logic programs over lattices.
In E. Sanchez, editor, *Fuzzy Logic and the Semantic Web*, Capturing Intelligence, chapter 7, pages 115–133. Elsevier, 2006.

U. Straccia.
Towards vague query answering in logic programming for logic-based information retrieval.
In *World Congress of the International Fuzzy Systems Association (IFSA-07)*, Cancun, Mexico, 2007.

U. Straccia and G. Visco.
DLMedia: an ontology mediated multimedia information retrieval system.
In *Proceeedings of the International Workshop on Description Logics (DL-07)*, Insbruck, Austria, 2007. CEUR.

V. V., B. J., G. P., and H. T.
Fuzzy rdf in the semantic web: Deduction and induction.
In *Proceedings of Workshop on Data Analysis (WDA 2005)*, 2005.

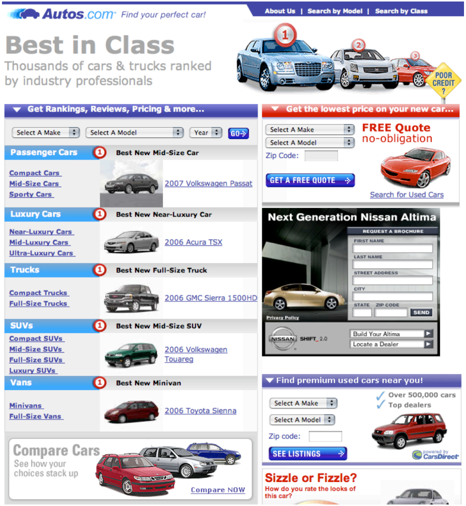
P. Vojtáš.
Fuzzy logic programming.
Fuzzy Sets and Systems, 124:361–370, 2001.

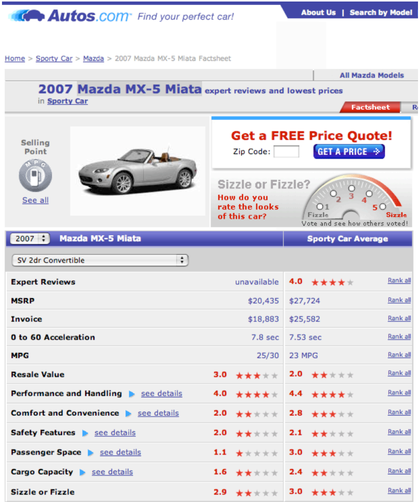
- Description logic programs that allow for dealing with probabilistic uncertainty and fuzzy vagueness.
- Semantically, probabilistic uncertainty can be used for data integration and ontology mapping, and fuzzy vagueness can be used for expressing vague concepts.
- Technically, allows for defining different rankings on ground atoms using fuzzy vagueness, and then for a probabilistic merging of these rankings using probabilistic uncertainty.
- Query processing based on fixpoint iterations.

Suppose a person would like to buy "a sports car that costs at most about 22 000 euro and that has a power of around 150 HP".

In todays Web, the buyer has to *manually*

- search for car selling web sites, e.g., using Google;
- select the most promising sites;
- browse through them, query them to see the cars that each site sells, and match the cars with the requirements;
- select the offers in each web site that match the requirements; and
- eventually merge all the best offers from each site and select the best ones.


Autos.com Find your perfect car!
About Us | Search by Model | Search by Class
Best in Class
Thousands of cars & trucks ranked by industry professionals
POOR CREDIT ?
Get Rankings, Reviews, Pricing & more...
Select A Make
Select A Model
Year
GO
Passenger Cars
Best New Mid-Size Car
Compact Cars
Mid-Size Cars
Sporty Cars
2007 Volkswagen Passat
Luxury Cars
Best New Near-Luxury Car
Near-Luxury Cars
Mid-Luxury Cars
Ultra-Luxury Cars
2006 Acura TSX
Trucks
Best New Full-Size Truck
Compact Trucks
Full-Size Trucks
2006 GMC Sierra 1500HD
SUVs
Best New Mid-Size SUV
Compact SUVs
Mid-Size SUVs
Full-Size SUVs
Luxury SUVs
2006 Volkswagen Touareg
Vans
Best New Minivan
Minivans
Full-Size Vans
2006 Toyota Sienna
Compare Cars
See how your choices stack up
Compare NOW
Get the lowest price on your new car...
FREE Quote
no-obligation
Zip Code:
GET A FREE QUOTE
Search for Used Cars
Next Generation Nissan Altima
REQUEST A BROCHURE
FIRST NAME
LAST NAME
STREET ADDRESS
CITY
STATE
ZIP CODE
SEND
Privacy Policy
NISSAN SHIFT_2.0
Build Your Altima
Locate a Dealer
Find premium used cars near you!
Over 500,000 cars
Top dealers
Zip code:
SEE LISTINGS
powered by CarsDirect
Sizzle or Fizzle?
How do you rate the looks of this car?


Autos.com Find your perfect car!
About Us | Search by Model
Home > Sporty Car > Mazda > 2007 Mazda MX-5 Miata Factsheet
All Mazda Models
2007 Mazda MX-5 Miata expert reviews and lowest prices
in Sporty Car
Factsheet
Selling Point
See all
Get a FREE Price Quote!
Zip Code:
GET A PRICE
Sizzle or Fizzle?
How do you rate the looks of this car?
Fizzle
Sizzle
Vote and see how others voted!
2007 Mazda MX-5 Miata
Sporty Car Average
SV 2dr Convertible
Expert Reviews unavailable 4.0 Rank all
MSRP $20,435 $27,724 Rank all
Invoice $18,883 $25,582 Rank all
0 to 60 Acceleration 7.8 sec 7.53 sec Rank all
MPG 25/30 23 MPG Rank all
Resale Value 3.0 2.0 Rank all
Performance and Handling see details 4.0 4.4 Rank all
Comfort and Convenience see details 2.0 2.8 Rank all
Safety Features see details 2.0 2.1 Rank all
Passenger Space see details 1.1 3.0 Rank all
Cargo Capacity see details 1.6 2.4 Rank all
Sizzle or Fizzle 2.9 3.0 Rank all

A *shopping agent* may support us, *automatizing* the whole process once it receives the request/query *q* from the buyer:

- The agent selects some sites/resources *S* that it considers as *relevant* to *q* (represented by probabilistic rules).
- For the top-*k* selected sites, the agent has to reformulate *q* using the terminology/ontology of the specific car selling site (which is done using probabilistic rules).
- The query *q* may contain many *vague/fuzzy* concepts such as "the price is around 22 000 euro or less", and so a car may *match* *q* to a *degree*. So, a resource returns a ranked list of cars, where the ranks depend on the degrees to which the cars match *q*.
- Eventually, the agent integrates the ranked lists (using probabilities) and shows the top-*n* items to the buyer.

$$
\begin{aligned}
&\mathit{Cars} \sqcup \mathit{Trucks} \sqcup \mathit{Vans} \sqcup \mathit{SUVs} \sqsubseteq \mathit{Vehicles}\\
&\mathit{PassengerCars} \sqcup \mathit{LuxuryCars} \sqsubseteq \mathit{Cars}\\
&\mathit{CompactCars} \sqcup \mathit{MidSizeCars} \sqcup \mathit{SportyCars} \sqsubseteq \mathit{PassengerCars}
\end{aligned}
$$

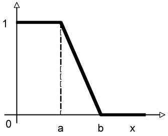
$$
\begin{aligned}
\mathit{Cars} \sqsubseteq\ & (\exists \mathit{hasReview}.\mathit{Integer}) \sqcap (\exists \mathit{hasInvoice}.\mathit{Integer})\\
& \sqcap (\exists \mathit{hasResellValue}.\mathit{Integer}) \sqcap (\exists \mathit{hasMaxSpeed}.\mathit{Integer})\\
& \sqcap (\exists \mathit{hasHorsePower}.\mathit{Integer}) \sqcap \ldots
\end{aligned}
$$

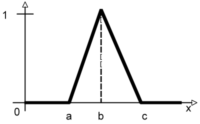
$$
\begin{aligned}
&\mathit{MazdaMX5Miata}\colon \mathit{SportyCar} \sqcap (\exists \mathit{hasInvoice}.18883)\\
&\qquad \sqcap (\exists \mathit{hasHorsePower}.166) \sqcap \ldots\\
&\mathit{MitsubishiEclipseSpyder}\colon \mathit{SportyCar} \sqcap (\exists \mathit{hasInvoice}.24029)\\
&\qquad \sqcap (\exists \mathit{hasHorsePower}.162) \sqcap \ldots
\end{aligned}
$$

We may now encode "costs at most about 22 000 euro" and "has a power of around 150 HP" in the buyer's request through the following concepts C and D, respectively:

$$C = \exists \mathit{hasInvoice}.\mathit{LeqAbout22000} \text{ and}$$
$$D = \exists \mathit{hasHorsePower}.\mathit{Around150HP},$$

where $\mathit{LeqAbout22000} = L(22000, 25000)$ and $\mathit{Around150HP} = \mathit{Tri}(125, 150, 175)$.





The following fuzzy dl-rule encodes the buyer's request "a sports car that costs at most about 22 000 euro and that has a power of around 150 HP".

$$\begin{aligned} \mathit{query}(x) \;\leftarrow_{\otimes}\;\; & \mathit{SportyCar}(x) \wedge_{\otimes} \\ & \mathit{hasInvoice}(x, y_1) \wedge_{\otimes} \\ & \mathit{DL}[\mathit{LeqAbout22000}](y_1) \wedge_{\otimes} \\ & \mathit{hasHorsePower}(x, y_2) \wedge_{\otimes} \\ & \mathit{DL}[\mathit{Around150HP}](y_2) \geq 1\,. \end{aligned}$$

Here, $\otimes$ is the Gödel t-norm (that is, $x \otimes y = \min(x, y)$).

The buyer's request, but in a "different" terminology:

$$\begin{aligned} \mathit{query}(x) \;\leftarrow_{\otimes}\; & \mathit{SportsCar}(x) \wedge_{\otimes} \mathit{hasPrice}(x, y_1) \wedge_{\otimes} \mathit{hasPower}(x, y_2) \wedge_{\otimes} \\ & \mathit{DL}[\mathit{LeqAbout22000}](y_1) \wedge_{\otimes} \mathit{DL}[\mathit{Around150HP}](y_2) \geq 1 \end{aligned}$$

Ontology alignment mapping rules:

$$\begin{aligned} \mathit{SportsCar}(x) &\leftarrow_{\otimes} \mathit{DL}[\mathit{SportyCar}](x) \wedge_{\otimes} sc_{pos} \geq 0.9 \\ \mathit{hasPrice}(x) &\leftarrow_{\otimes} \mathit{DL}[\mathit{hasInvoice}](x) \wedge_{\otimes} hi_{pos} \geq 0.8 \\ \mathit{hasPower}(x) &\leftarrow_{\otimes} \mathit{DL}[\mathit{hasHorsePower}](x) \wedge_{\otimes} hhp_{pos} \geq 0.8\,, \end{aligned}$$

Probability distribution μ:

$$\begin{array}{ll} \mu(sc_{pos}) = 0.91 & \mu(sc_{neg}) = 0.09 \\ \mu(hi_{pos}) = 0.78 & \mu(hi_{neg}) = 0.22 \\ \mu(hhp_{pos}) = 0.83 & \mu(hhp_{neg}) = 0.17 \end{array}$$

The following are some tight consequences:

$$KB \;\|\!\!\sim_{tight}\; (\mathbf{E}[query((MazdaMX5Miata)])[0.21, 0.21]$$
$$KB \;\|\!\!\sim_{tight}\; (\mathbf{E}[query((MitsubishiEclipseSpyder)])[0.19, 0.19]\,.$$

Informally, the expected degree to which *MazdaMX5Miata* matches the query q is 0.21, while the expected degree to which *MitsubishiEclipseSpyder* matches the query q is 0.19,

Thus, the shopping agent ranks the retrieved items as follows:

rank	item	degree
1.	*MazdaMX5Miata*	0.21
2.	*MitsubishiEclipseSpyder*	0.19

T. Lukasiewicz.
Fuzzy description logic programs under the answer set semantics for the semanticweb.
In *Second International Conference on Rules and Rule Markup Languages for the Semantic Web (RuleML-06)*. IEEE, 2006.

T. Lukasiewicz and U. Straccia.
Tightly integrated fuzzy description logic programs under the answer semantics for the semantic web.
In *Proceedings of the First International Conference on Web Reasoning and Rule Systems (RR-07)*, number 4524 in Lecture Notes in Computer Science, pages 289—298. Springer Verlag, 2007.

T. Lukasiewicz and U. Straccia.
Tightly integrated fuzzy description logic programs under the answer semantics for the semantic web.
INFSYS RESEARCH REPORT 1843-07-03, INSTITUT FüR INFORMATIONSSYSTEME ARBEITSBEREICH WISSENSBASIERTE SYSTEME, Technische Universität Wien, 2007.

C. Mateis.
Extending disjunctive logic programming by t-norms.
In *Proceedings of the 5th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR-99)*, number 1730 in Lecture Notes in Computer Science, pages 290–304. Springer-Verlag, 1999.

C. Mateis.
Quantitative disjunctive logic programming: Semantics and computation.
AI Communications, 13:225–248, 2000.

U. Straccia.
Annotated answer set programming.
In *Proceedings of the 11th International Conference on Information Processing and Managment of Uncertainty in Knowledge-Based Systems, (IPMU-06)*, pages 1212–1219. E.D.K., Paris, 2006.

U. Straccia.
Fuzzy description logic programs.
In *Proceedings of the 11th International Conference on Information Processing and Managment of Uncertainty in Knowledge-Based Systems, (IPMU-06)*, pages 1818–1825. E.D.K., Paris, 2006.

U. Straccia.
Uncertainty and description logic programs over lattices.
In E. Sanchez, editor, *Fuzzy Logic and the Semantic Web*, Capturing Intelligence, chapter 7, pages 115–133. Elsevier, 2006.