Basic Concepts and Techniques for Managing Uncertainty and Vagueness in Semantic Web Languages

Lecture at Reasoning Web 2008

Umberto Straccia

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

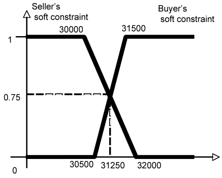
Uncertainty, Vagueness, and the Semantic Web

Sources of Uncertainty and Vagueness on the Web

- (Multimedia) 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?

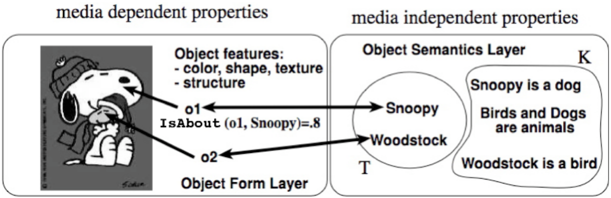
- Semantic annotation / classification
  - To which degree does e.g., an image object represent or is about a dog?
- Information extraction
  - To which degree am I'm sure that e.g., SW is an acronym of "Semantic Web"?
- Ontology alignment (schema mapping)
  - To which degree do two concepts of two ontologies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

Example (Matchmaking)



- 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 (Multimedia information retrieval)

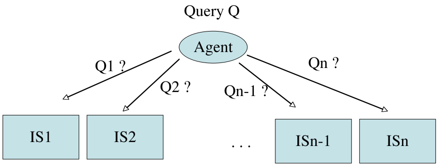


IsAbout		
ImageRegion	*Object ID*	*degree*
*o*1	*snoopy*	0.8
*o*2	*woodstock*	0.7
⋮	⋮	

“Find top-*k* image regions about animals”

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

Example (Distributed Information Retrieval)



Then the agent has to perform automatically the following steps:

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

Example (Database query)

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

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

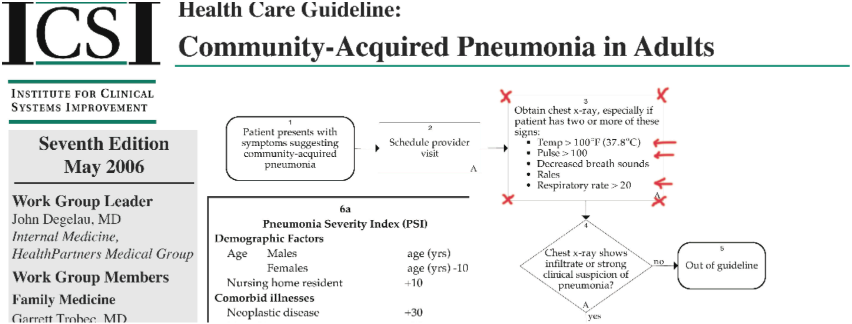
hasLoc	*hasLoc*	*distance*
*hl*1	*cl*1	300
*hl*1	*cl*2	500
*hl*2	*cl*1	750
*hl*2	*cl*2	800
⋮	⋮	

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”

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

Example (Health-care: diagnosis of pneumonia)



ICSI
Health Care Guideline:
Community-Acquired Pneumonia in Adults

Institute for Clinical Systems Improvement

Seventh Edition
May 2006

Work Group Leader
John Degelau, MD
Internal Medicine,
HealthPartners Medical Group
Work Group Members
Family Medicine
Garrett Trobec, MD

1 Patient presents with symptoms suggesting community-acquired pneumonia

2 Schedule provider visit

A

3 Obtain chest x-ray, especially if patient has two or more of these signs:
- Temp > 100°F (37.8°C)
- Pulse > 100
- Decreased breath sounds
- Rales
- Respiratory rate > 20

A

4 Chest x-ray shows infiltrate or strong clinical suspicion of pneumonia?

no → 5 Out of guideline

A

yes

6a
Pneumonia Severity Index (PSI)

Demographic Factors		
Age	Males	age (yrs)
	Females	age (yrs) -10
Nursing home resident		+10
Comorbid illnesses		
Neoplastic disease		+30

- 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 value (usually in $[0, 1]$) of 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
  - we assume that we can clearly define the property "can fly"
- The probability/possibility/necessity degree that it flies is 0.83
- E.g., under probability theory this may mean that 83% of the birds do fly, while 17% of the birds do not fly
  - Note: e.g., a chicken has to be classified as either flying or non-flying thing

Example

- Sport Car:

$$\forall x, hp, sp, ac\ SportCar(x) \iff HP(x, hp) \land Speed(x, sp) \land Acceleration(x, ac) \land hp \geq 210 \land sp \geq 220 \land ac \leq 7.0$$

		
audi_tt	*mg*	*ferrari_enzo*

- Ferrari Enzo is a Sport Car: $HP = 651$, $Speed \geq 350$, $Acc. = 3.14$
- MG is not a Sport Car: $HP = 59$, $Speed = 170$, $Acc. = 14.3$
- Is Audi TT 2.0 a Sport Car ? $HP = unknown$, $Speed = 243$, $Acc. = 6.9$
- We can estimate from a training set (Naive Bayes Classification)

$$\begin{aligned} Pr(SportCar \mid AudiTT) &= Pr(AudiTT \mid SportCar) \cdot Pr(SportCar) \cdot (1/Pr(AudiTT)) \\ &\approx \frac{Pr(speed \geq 243 \mid SportCar) \cdot Pr(accel \leq 6.9 \mid SportCar) \cdot Pr(SportCar)}{Pr(speed \geq 243) \cdot Pr(accel \leq 6.9)} \end{aligned}$$

- Sport Car:

$$\forall x, hp, sp, ac\ SportCar(x) \iff HP(x, hp) \wedge Speed(x, sp) \wedge Acceleration(x, ac) \wedge hp \geq 210 \wedge sp \geq 220 \wedge ac \leq 7.0$$

		
audi_tt	*mg*	*ferrari_enzo*

- Note: Audi TT 2.0 is not a Sport Car: $HP = 200$, $Speed = 243$, $Acc. = 6.9$
- Explicit definition of Sport Car is too sharp
- We can estimate from a training set (Naive Bayes Classification)

$$Pr(SportCar \mid MyCar) = Pr(MyCar \mid SportCar) \cdot Pr(SportCar) \cdot (1 / Pr(MyCar))$$

$$\approx \frac{Pr(MyCar.hp_{\geq} \mid SportCar) \cdot Pr(MyCar.speed_{\geq} \mid SportCar) \cdot Pr(MyCar.accel_{\leq} \mid SportCar) \cdot Pr(SportCar)}{Pr(MyCar.hp_{\geq}) \cdot Pr(MyCar.speed_{\geq}) \cdot Pr(MyCar.accel_{\leq})}$$

Vagueness

- Vagueness: statements involve concepts for which there is no exact definition, such as
  - tall, small, close, far, cheap, expensive, "is about", "similar to".
- A statements is true to some degree, which is taken from a truth space (usually $[0, 1]$).
- E.g., "Hotel Verdi is close to the train station to degree 0.83"
  - the degree depends on the distance
- E.g., "The image is about a sun set to degree 0.75"
  - the degree depends on the extracted features and the semantic annotations

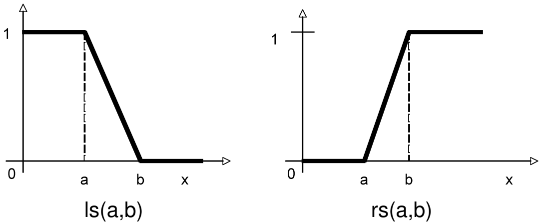
Example

- Sport Car:

$$\forall x, hp, sp, ac\ SportCar(x) \iff 0.3HP(x, hp) + 0.2Speed(x, sp) + 0.5Accel(x, ac)$$

- Each feature, gives a degree of truth depending on the value and the membership function

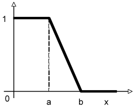
$$\begin{aligned} HP(x, hp) &= rs(180, 250)(hp) \\ Speed(x, sp) &= rs(180, 240)(sp) \\ Accel(x, ac) &= ls(6.0, 8.0)(ac) \end{aligned}$$



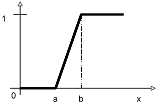
- Degree of truth of $SportCar(AudiTT)$: $0.1 \cdot 0.28 + 0.3 \cdot 1.0 + 0.6 \cdot 0.55 = 0.658$

- The fuzzy membership functions can be learned from a training set (large literature)

$$\begin{aligned} HP(x, hp) &= rs(192, 242)(hp) \\ Speed(x, sp) &= rs(193, 234)(sp) \\ Accel(x, ac) &= ls(6.5, 7.5)(ac) \end{aligned}$$



ls(a,b)



rs(a,b)

- Learned Training Sport Class:

$$\forall x, hp, sp, ac\ TrainingSportCar(x) \iff 0.3HP(x, hp) + 0.2Speed(x, sp) + 0.5Accel(x, ac)$$

- Now, a classification method can be applied: e.g. kNN classifier

$$\forall x, hp, sp, ac\ SportCar(x) \iff \sum_{y \in Top_k(x)} Similar(x, y) \cdot TrainingSportCar(y)$$

$$\begin{aligned} \forall x, hp, sp, ac\ Similar(x, y) \iff & 0.3 \cdot HP(x, hpx) \cdot HP(y, hpy) + 0.2 \cdot Speed(x, spx) \cdot Speed(y, spy) + \\ & + 0.5 \cdot Accel(x, acx) \cdot Accel(y, acy) \end{aligned}$$

where $Top_k(x)$ is the set of top-k ranked most similar cars to car x

Imperfect Information

- Mixing uncertainty and vagueness:
  - "Probably it will be hot tomorrow"
    - Crisp quantifier ("probably") over vague statement
  - "In most cases, a bird does fly"
    - Vague quantifier ("most") over crisp statement
- The notion of imperfect information covers concepts such as

uncertainty	"Nancy is likely John's girlfriend"
vagueness	"John's girlfriend is blond"
incompleteness	"John's girlfriend is Nancy or Mary"
imprecision	"The hight of John's girlfriend is in between 165cm and 170cm"
contradiction	"John's girlfriend, Nancy, lives in Rome. Nancy is living in Florence."

Uncertainty vs. Vagueness

- The distinction between uncertainty and vagueness is not always clear: depends on the assumptions
- (Multimedia) Information Retrieval:

Query: "I'm looking for a house"

System Answer:  score/degree 0.83

- What's behind the computational model?

- Probabilistic model
  - Assumption: a multimedia object is either relevant or not relevant to a query q
  - Score: The probability of being a multimedia object o relevant (Rel) to q

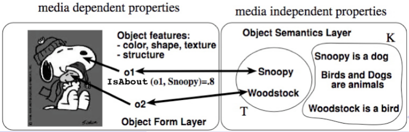
$$score := Pr(Rel \mid q, o)$$

- Vague/Fuzzy model
  - Assumption: a multimedia object o is about a *semantic index term* ($t \in \mathbb{T}$) to some degree in $[0, 1]$
  - The mapping of objects $o \in \mathbb{O}$ to semantic entities $t \in \mathbb{T}$ is called *semantic annotation*

$$F : \mathbb{O} \times \mathbb{T} \to [0, 1]$$

  $F(o, t)$ indicates to which degree the multimedia object o is about the semantic index term t
  - Score: The evaluation of how much the multimedia object o is about the the information need q

$$score := F(o, q)$$



- In other cases there may be both approaches as well
- For instance, in Ontology Alignment, what about the degree n of the mapping

$$\langle SUV, Van, \cap, n \rangle \text{ ?}$$

- Probabilistic model: a car is a SUV (Van) or is not a SUV (Van)
- Then, e.g. from a training set, compute

$$n = Pr(SUV \cap Van)$$

- Fuzzy model: a car is to some degree a SUV and to some other degree a Van
- Then, e.g. from a training set, compute

$$n = kNNSUV(x) \cdot kNNVan(x)$$

Semantic Web Languages Basics

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 ...)
- Possibly RIF is coming . . .

RDF

- Statements are of the form

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

called triples: e.g.

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

- can be represented graphically as:

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

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

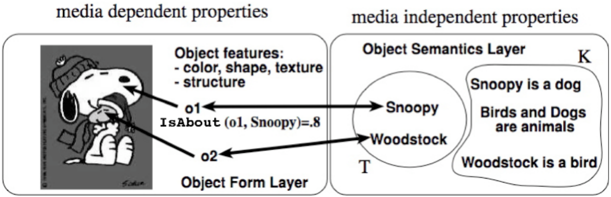
RDF Schema (RDFS)

- 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>
```

Representing degrees in RDF/RDFS

- How can we represent degrees of uncertainty and vagueness in RDF/RDFS?
- Unfortunately, no standard exists yet
- So far, an option is to uses special purpose properties and reification



$$\langle statement1, hasSubject, o1\rangle$$
$$\langle statement1, hasProperty, IsAbout\rangle$$
$$\langle statement1, hasObject, snoopy\rangle$$
$$\langle statement1, hasDegree, 0.8\rangle$$

- But, then such statements have to be appropriately be managed by the system according to the underlying uncertainty or vagueness theory

OWL

- 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	$\rightarrow$	$\top$	$\mid$	$\top(x)$	
		$\bot$	$\mid$	$\bot(x)$	
		A	$\mid$	$A(x)$	*Human*
		$C \sqcap D$	$\mid$	$C(x) \wedge D(x)$	$Human \sqcap Male$
		$C \sqcup D$	$\mid$	$C(x) \vee D(x)$	$Nice \sqcup Rich$
		$\neg C$	$\mid$	$\neg C(x)$	$\neg Meat$
		$\exists R.C$	$\mid$	$\exists y.R(x,y) \wedge C(y)$	$\exists has\_child.Blond$
		$\forall R.C$		$\forall y.R(x,y) \Rightarrow C(y)$	$\forall has\_child.Human$
$C \sqsubseteq D$				$\forall x.C(x) \Rightarrow D(x)$	$Happy\_Father \sqsubseteq Man \sqcap \exists has\_child.Female$
$a{:}C$				$C(a)$	$John{:}Happy\_Father$

Toy Example

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

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

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

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,
  $(\geq n\ R.C)^{\mathcal{I}} = \{x \in |\{y \mid \langle x, y \rangle \in R^{\mathcal{I}} \land y \in C^{\mathcal{I}}\}| \geq n\}$,
  $(\leq n\ R.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} : |\{y \mid \langle x, y \rangle \in R^{\mathcal{I}} \land y \in C^{\mathcal{I}}\}| \leq n\}$
- $\mathcal{O}$: 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,
  $(R_+)^{\mathcal{I}} = \{\langle x, y \rangle \mid \exists z$ such that $\langle x, z \rangle \in R^{\mathcal{I}} \land \langle z, y \rangle \in R^{\mathcal{I}}\}$

Concrete Domains

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

$$(tim, 14){:}hasAge$$
$$(sf, \text{“}SoftComputing\text{”}){:}hasAcronym$$
$$(source1, \text{“}ComputerScience\text{”}){:}isAbout$$
$$(service2, \text{“}InformationRetrievalTool''){:}Matches$$
$$YoungPerson = Person \sqcap \exists hasAge. \leq_{18}$$

- 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

Representing degrees in OWL-DL/OWL-Lite

- How can we represent degrees of uncertainty and vagueness in OWL-DL/OWL-Lite?
- Unfortunately, as for RDF, no standard exists yet
- We may make an encoding as for RDF

$$s1{:}\exists hasSubject.(\{o1\} \sqcap \exists IsAbout.\{snoopy\}) \sqcap \exists hasDegree. =_{0.8}$$

- But, again then such statements have to be appropriately be managed by the system according to the underlying uncertainty or vagueness theory
- A rather dangerous approach

LPs Basics (for ease, without default negation)

- 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

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

- 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

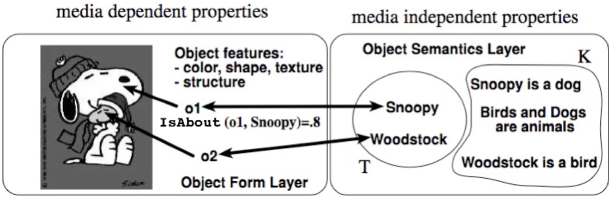
$$
\begin{array}{rcl}
Q(x) & \leftarrow & B(x) \\
Q(x) & \leftarrow & C(x) \\
B(a) & \leftarrow & \\
C(b) & \leftarrow &
\end{array}
$$

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

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

Representing degrees in LPs

- How can we represent degrees of uncertainty and vagueness in LPs?
- Unfortunately, no standard exists yet
- However, as simple encoding is to make transform an n-ary predicate P into an $n+1$-ary predicate, where the additional argument stores the value:



IsAbout(o1,snoopy, 0.8)

- For instance, in LP systems we may write

$$q(h, s) \leftarrow \mathit{hasLocation}(h, hl), \mathit{hasLocation}(\mathit{train}, cl), \mathit{close}(hl, cl, s1), \mathit{cheap}(h, s2), s \text{ is } s1 \cdot s2$$

- But, then again such statements have to be appropriately be managed the system according to the underlying uncertainty or vagueness theory

Uncertainty and Vagueness Basics

Uncertainty & Logic

- Any statement φ is either true or false
- Due to lack of knowledge we can only estimate to which probability/possibility/necessity degree they are true or false
- Usually we have a possible world semantics with a distribution over possible worlds
- Possible world: any classical interpretation I, mapping any statement φ into $\{0, 1\}$

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

- Distribution: a mapping

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

obeying some additional conditions (probability distribution, possibility distribution)

- $\mu(I)$ indicates the probability/possibility that the world I is indeed the actual one

- The probability of a statement φ is determined as

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

- The posssibility of a statement φ is determined as

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

- The necessity of a statement φ is determined as

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

Example

Probabilistic setting:

$$\varphi = \mathit{sprinklerOn} \vee \mathit{wet}$$

W	$\mathit{sprinklerOn}$	wet	μ
I_1	0	0	0.1
I_2	0	1	0.2
I_3	1	0	0.4
I_4	1	1	0.3

$$1 = \sum_{I \in W} \mu(I)$$

$$Pr(\varphi) = 0.2 + 0.4 + 0.3 = 0.9$$

Example

Possibilistic setting:

$$\varphi = \mathit{sprinklerOn} \vee \mathit{wet}$$

W	*sprinklerOn*	*wet*	μ
I_1	0	0	0.3
I_2	0	1	1.0
I_3	1	0	0.8
I_4	1	1	1.0

$$1 = \sup_{I \in W} \mu(I)$$

$$\begin{array}{rcl} \mathit{Poss}(\varphi) & = & \sup(1.0, 0.8, 1.0) = 1.0 \\ \mathit{Necc}(\varphi) & = & 1 - \mathit{Poss}(\neg\varphi) = 1 - 0.3 = 0.7 \end{array}$$

Properties of probabilistic formulae

$$
\begin{array}{lcl}
Pr(\varphi \wedge \psi) & = & Pr(\varphi) + Pr(\psi) - Pr(\varphi \vee \psi) \\
Pr(\varphi \wedge \psi) & \leq & \min(Pr(\varphi), Pr(\psi)) \\
Pr(\varphi \wedge \psi) & \geq & \max(0, Pr(\varphi) + Pr(\psi) - 1) \\
Pr(\varphi \vee \psi) & = & Pr(\varphi) + Pr(\psi) - Pr(\varphi \wedge \psi) \\
Pr(\varphi \vee \psi) & \leq & \min(1, Pr(\varphi) + Pr(\psi)) \\
Pr(\varphi \vee \psi) & \geq & \max(Pr(\varphi), Pr(\psi)) \\
Pr(\neg\varphi) & = & 1 - Pr(\varphi) \\
Pr(\bot) & = & 0 \\
Pr(\top) & = & 1
\end{array}
$$

Properties of possibilistic formulae

$$
\begin{array}{lcl}
Poss(\varphi \wedge \psi) & \leq & \min(Poss(\varphi), Poss(\psi)) \\
Poss(\varphi \vee \psi) & = & \max(Poss(\varphi), Poss(\psi)) \\
Poss(\neg\varphi) & = & 1 - Nec(\varphi) \\
Poss(\bot) & = & 0 \\
Poss(\top) & = & 1 \\
Nec(\varphi \wedge \psi) & = & \min(Nec(\varphi), Nec(\psi)) \\
Nec(\varphi \vee \psi) & \geq & \max(Nec(\varphi), Nec(\psi)) \\
Nec(\neg\varphi) & = & 1 - Poss(\varphi) \\
Nec(\bot) & = & 0 \\
Nec(\top) & = & 1
\end{array}
$$

Probabilistic Knowledge Bases

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

Example

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

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

  "Eagles are birds".

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

  "Birds have legs".

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

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

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.
- $KB \,\|\!\!\models (\psi|\varphi)[l, u]$: $(\psi|\varphi)[l, u]$ is a logical consequence of *KB* iff every model of *KB* is also a model of $(\psi|\varphi)[l, u]$.
- $KB \,\|\!\!\models_{tight} (\psi|\varphi)[l, u]$: $(\psi|\varphi)[l, u]$ is a tight logical consequence of *KB* iff *l* (resp., *u*) is the infimum (resp., supremum) of $Pr(\psi|\varphi)$ subject to all models *Pr* of *KB* with $Pr(\varphi) > 0$.

Example

- Probabilistic knowledge base:

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

- KB is satisfiable, since

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

- Some conclusions under logical entailment:

$$KB \mathrel{\|}\!\!= (have\_legs \mid bird)[0.3,1], \quad KB \mathrel{\|}\!\!= (fly \mid bird)[0.6,1].$$

- Tight conclusions under logical entailment:

$$\begin{aligned} & KB \mathrel{\|}\!\!=_{tight} (have\_legs \mid bird)[1,1], \quad KB \mathrel{\|}\!\!=_{tight} (fly \mid bird)[0.95,1], \\ & KB \mathrel{\|}\!\!=_{tight} (have\_legs \mid eagle)[1,1], \quad KB \mathrel{\|}\!\!=_{tight} (fly \mid eagle)[0,1]. \end{aligned}$$

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:

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

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 $R = \{I \in \mathcal{I}_\Phi \mid I \models L\}$:

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

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

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 $H(Pr)$:

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

- 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.

- $KB \,\|\!\!\models^{me} (\psi|\varphi)[l, u]$: $(\psi|\varphi)[l, u]$ is a ME consequence of KB iff the ME model of KB is also a model of $(\psi|\varphi)[l, u]$.

- $KB \,\|\!\!\models^{me}_{tight} (\psi|\varphi)[l, u]$: $(\psi|\varphi)[l, u]$ is a tight ME consequence of KB iff for the ME model Pr of KB, it holds either (a) $Pr(\varphi) = 0$, $l = 1$, and $u = 0$, or (b) $Pr(\varphi) > 0$ and $Pr(\psi|\varphi) = l = u$.

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:

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

Lexicographic Entailment

- Pr verifies $(\psi|\varphi)[l,u]$ iff $Pr(\varphi) = 1$ and $Pr \models (\psi|\varphi)[l,u]$.
- P tolerates $(\psi|\varphi)[l,u]$ under L iff $L \cup P$ has a model that verifies $(\psi|\varphi)[l,u]$.
- $KB = (L, P)$ 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.
- $KB \Vdash\!\sim^{lex} (\psi|\varphi)[l, u]$: $(\psi|\varphi)[l, u]$ is a *lex*-consequence of KB iff every *lex*-minimal model Pr of L with $Pr(\varphi) = 1$ satisfies $(\psi|\varphi)[l, u]$.
- $KB \Vdash\!\sim^{lex}_{tight} (\psi|\varphi)[l, u]$: $(\psi|\varphi)[l, u]$ 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 $Pr(\varphi) = 1$.

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:

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

Tight conclusions under probabilistic lexicographic entailment:

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

Bayesian Networks

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

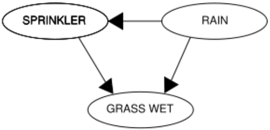
Joint probability distributions can answer any queries, but can be very large and are often hard to specify.

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.

RAIN	SPRINKLER T	SPRINKLER F
F	0.4	0.6
T	0.01	0.99



RAIN T	RAIN F
0.2	0.8

SPRINKLER	RAIN	GRASS WET T	GRASS WET F
F	F	0.0	1.0
F	T	0.8	0.2
T	F	0.9	0.1
T	T	0.99	0.01

The model can answer questions like “What is the probability that it is raining, given the grass is wet?”

$$
\begin{aligned}
Pr(\texttt{Rain} = T \mid \texttt{GrassWet} = T) &= \frac{Pr(\texttt{Rain} = T, \texttt{GrassWet} = T)}{Pr(\texttt{GrassWet} = T)} \\
&= \frac{\sum_{Y \in \{T,F\}} Pr(\texttt{Rain} = T, \texttt{GrassWet} = T, \texttt{Sprinkler} = Y)}{\sum_{Y_1, Y_2 \in \{T,F\}} Pr(\texttt{GrassWet} = T, (\texttt{Rain} = Y_1, \texttt{Sprinkler} = Y_2))} \\
&= \frac{0.99 \cdot 0.01 \cdot 0.2 + 0.8 \cdot 0.99 \cdot 0.2}{0.99 \cdot 0.01 \cdot 0.2 + 0.9 \cdot 0.4 \cdot 0.8 + 0.8 \cdot 0.99 \cdot 0.2 + 0 \cdot 0.6 \cdot 0.8} \\
&\approx 0.3577 \,.
\end{aligned}
$$

Possibilistic Knowledge Bases

- Possibilistic formulae have the form $\mathrm{P}\,\varphi \geq l$ or $\mathrm{N}\,\varphi \geq l$, $l \in [0,1]$
- Encode to what extent φ is *possibly* resp. *necessarily* true
- Possibilistic interpretation: $\pi : \mathcal{I}_\Phi \to [0,1]$
- $\pi(I)$ is the degree to which the world I is *possible*
- It is assumed that $\pi(I) = 1$ for some $I \in \mathcal{I}_\Phi$
- Possibility/Necessity of an event φ under π:

$$\begin{array}{rcl} Poss(\varphi) & = & \sup\{\pi(I) \mid I \in \mathcal{I}_\Phi, I \models \varphi\} (\text{ where } \max \emptyset = 0) \\ Necc(\varphi) & = & 1 - Poss(\neg\varphi) \end{array}$$

- Truth under π:

$$\begin{array}{rcl} \pi \models \mathrm{P}\,\varphi \geq l & \textit{iff} & Poss(\varphi) \geq l \\ \pi \models \mathrm{N}\,\varphi \geq l & \textit{iff} & Necc(\varphi) \geq l \end{array}$$

Deciding Logical entailment (Hollunder's method)

- Reduction to propositional entailment
- Let

$$
\begin{aligned}
KB_l &= \{\varphi \mid \mathrm{N}\,\varphi \geq l' \in KB, l' \geq l\} \\
KB^l &= \{\varphi \mid \mathrm{N}\,\varphi \geq l' \in KB, l' > l\}
\end{aligned}
$$

- Then

$$
\begin{aligned}
KB \models \mathrm{N}\,\varphi \geq l \quad & \textit{iff} \quad KB_l \models \varphi \\
KB \models \mathrm{P}\,\varphi \geq l \quad & \textit{iff} \quad KB^0 \models \varphi \text{ or} \\
& \quad \text{there is } \mathrm{P}\,\psi \geq l' \in KB \text{ with } l' \geq l, KB^{1-l'} \cup \{\psi\} \models \varphi
\end{aligned}
$$

Vagueness & Logic

- Statements involve concepts for which there is no exact definition, such as
  - tall, small, close, far, cheap, expensive, “is about”, “similar to”.
- A statements is 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”
- E.g., “The image is about a sun set to degree 0.75”
- Truth space: set of truth values L and an partial order $\leq$
- Many-valued Interpretation: a function I mapping formulae into L, i.e. $I(\varphi) \in L$
- Mathematical Fuzzy Logic: $L = [0, 1]$, but also $\{\frac{0}{n}, \frac{1}{n}, \ldots, \frac{n}{n}\}$ for an integer $n \geq 1$

- Problem: what is the interpretation of e.g. $\varphi \wedge \psi$?
  - E.g., if $I(\varphi) = 0.83$ and $I(\psi) = 0.2$, 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"
- Norms: functions that are used to interpret connectives such as $\wedge, \vee, \neg, \rightarrow$
  - t-norm: interprets conjunction
  - s-norm: interprets disjunction
- Norms are compatible with classical two-valued logic

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	$0 \to b = 1$	$\neg 0 = 1, \ \neg 1 = 0$
	$a \to 1 = 1$	
Antitonicity	if $a \leq b$, then $a \to c \geq b \to c$	if $a \leq b$, then $\neg a \geq \neg b$
Monotonicity	if $b \leq c$, then $a \to b \leq a \to c$	

Usually,

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

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

- Any other t-norm can be obtained as a combination of Lukasiewicz, Gödel and Product t-norm
- Zadeh: not interesting for mathematical fuzzy logicians: its a sub-logic of Łukasiewicz and, thus, rarely considered by fuzzy logicians

$$
\begin{aligned}
\neg_Z x &= \neg_Ł x \\
x \wedge_Z y &= x \wedge_Ł (x \rightarrow_Ł y) \\
x \rightarrow_Z y &= \neg_Ł x \vee_Ł y
\end{aligned}
$$

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
$x \wedge \neg x = 0$	•	•	•	
$x \vee \neg x = 1$	•			
$x \wedge x = x$		•		•
$x \vee x = x$		•		•
$\neg \neg x = x$	•			•
$x \Rightarrow y = \neg x \vee y$	•			•
$\neg (x \Rightarrow y) = x \wedge \neg y$	•			•
$\neg (x \wedge y) = \neg x \vee \neg y$	•	•	•	•
$\neg (x \vee y) = \neg x \wedge \neg y$	•	•	•	•
$x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$		•		•
$x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$		•		•

- Note: If all conditions in the upper part of a column have to be satisfied then we collapse to classical two-valued logic, i.e. $L = \{0, 1\}$

Propositional Fuzzy Logic

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

$$
\begin{array}{lcl}
\mathcal{I}(\varphi \wedge \psi) & = & \mathcal{I}(\varphi) \wedge \mathcal{I}(\psi) \\
\mathcal{I}(\varphi \vee \psi) & = & \mathcal{I}(\varphi) \vee \mathcal{I}(\psi) \\
\mathcal{I}(\varphi \to \psi) & = & \mathcal{I}(\varphi) \to \mathcal{I}(\psi) \\
\mathcal{I}(\neg \varphi) & = & \neg \mathcal{I}(\varphi)
\end{array}
$$

- Rational $r \in [0, 1]$ may appear as atom in formula, where $\mathcal{I}(r) = r$

Example

In Lukasiewicz logic:

$$\varphi = \mathit{Cold} \wedge \mathit{Cloudy}$$

$\mathcal{I}$	*Cold*	*Cloudy*	$\mathcal{I}(\varphi)$
$\mathcal{I}_1$	0	0.1	$\max(0, 0 + 0.1 - 1) = 0.0$
$\mathcal{I}_2$	0.3	0.4	$\max(0, 0.3 + 0.4 - 1) = 0.0$
$\mathcal{I}_3$	0.7	0.8	$\max(0, 0.7 + 0.9 - 1) = 0.6$
$\mathcal{I}_4$	1	1	$\max(0, 1 + 1 - 1) = 1.0$
$\vdots$	$\vdots$	$\vdots$	$\vdots$

- **Note**:

$$\mathcal{I}(r \to \varphi) = 1 \quad \textit{iff} \quad \mathcal{I}(\varphi) \geq r$$
$$\mathcal{I}(\varphi \to r) = 1 \quad \textit{iff} \quad \mathcal{I}(\varphi) \leq r$$

- **Semantics:**

$$\begin{array}{rcl} I \models \varphi & \textit{iff} & \mathcal{I}(\varphi) = 1 \\ \mathcal{I} \models KB & \text{iff} & \mathcal{I} \models \varphi \text{ for all } \varphi \in KB \\ KB \models \varphi & \text{iff} & \text{for all } \mathcal{I}. \text{ if } \mathcal{I} \models KB \text{ then } \mathcal{I} \models \varphi \end{array}$$

- Deduction rule is valid: for $r, s \in [0, 1]$:

$$r \to \varphi, s \to (\varphi \to \psi) \models (r \wedge s) \to \psi$$

Informally,

From $\varphi \geq r$ and $(\varphi \to \psi) \geq s$ infer $\psi \geq r \wedge s$

Example

In Lukasiewicz logic:

$$\varphi = 0.4 \rightarrow (Cold \land Cloudy)$$

Read: $Cold \land Cloudy \geq 0.4$

$\mathcal{I}$	*Cold*	*Cloudy*	$\mathcal{I}(\varphi)$
$\mathcal{I}_1$	0	0.1	$0.4 \rightarrow 0.0 = \min(1, 1 - 0.4 + 0.0) = 0.6$
$\mathcal{I}_2$	0.3	0.4	$0.4 \rightarrow 0.0 = \min(1, 1 - 0.4 + 0.0) = 0.6$
$\mathcal{I}_3$	0.7	0.8	$0.4 \rightarrow 0.6 = \min(1, 1 - 0.4 + 0.6) = 1.0$
$\mathcal{I}_4$	1	1	$0.4 \rightarrow 1.0 = \min(1, 1 - 0.4 + 1.0) = 1.0$
$\vdots$	$\vdots$	$\vdots$	$\vdots$

$$\begin{array}{ccc} \mathcal{I}_1 & \not\models & \varphi \\ \mathcal{I}_2 & \not\models & \varphi \\ \mathcal{I}_3 & \models & \varphi \\ \mathcal{I}_4 & \models & \varphi \end{array}$$

- Let

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

  then $||\varphi||_{KB} = |\varphi|_{KB}$

- Also,

$$
\begin{aligned}
|\neg\varphi|_{KB} &= 1 - |\varphi|_{KB} \\
|\varphi|_{KB} = \sup\{r \mid KB \models r \to \varphi\} &= \inf\{s \mid KB \models \varphi \to s\}
\end{aligned}
$$

Proposition

$|\varphi|_{KB} = \min x.$ *such that* $KB \cup \{\varphi \to x\}$ *satisfiable.*

Decision algorithm, e.g. for Lukasiewicz Logic

- We use MILP (Mixed Integer Linear Programming) to compute $|\varphi|_{KB}$
- Let $r \in [0, 1]$, variable or expresson $1 - r'$ (r' variable), admitting solution in $[0, 1]$, $\neg r = 1 - r$, $\neg\neg r = r$
- For each propositional letter p, let x_p be a variable denoting the degree of truth of p
- Apply inference rules

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

- If no rule is applicable, solve the MILP problem of the form

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

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

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

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

Predicate Fuzzy Logics Basics

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

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

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

- $\neg\forall\mathbf{x}\,\varphi(\mathbf{x}) \equiv \exists\mathbf{x}\,\neg\varphi(\mathbf{x})$ true in Ł, but does not hold for logic G and Π
- $(\neg\forall x\, p(x)) \wedge (\neg\exists x\, \neg p(x))$ has no classical model. In Gödel logic it has no finite model, but has an infinite model: for integer $n \geq 1$, let $\mathcal{I}$ such that $\mathcal{I}(p(n)) = 1/n$

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

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

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

- Witnessed formula: $\exists x\, \phi(x)$ is witnessed in $\mathcal{I}$ iff there is $c \in \Delta_{\mathcal{I}}$ such that $\mathcal{I}(\exists x\, \phi(x)) = \mathcal{I}(\phi(c))$ (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

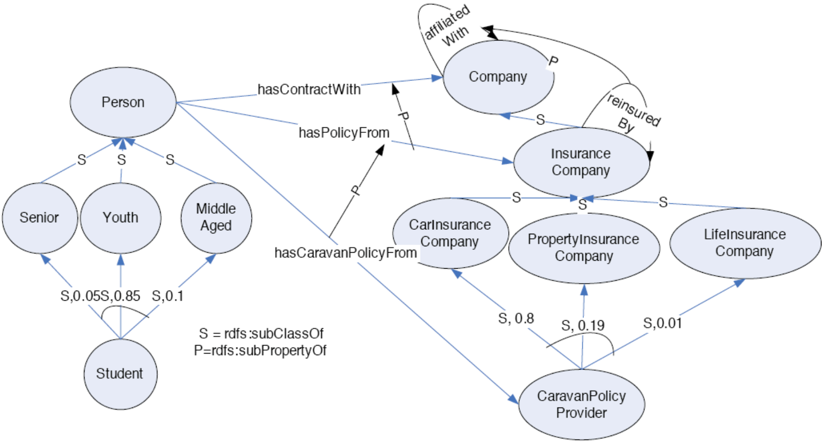
Uncertainty and Vagueness in Semantic Web Languages

The case of RDF

A Probabilistic RDF

- 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

Example of probabilistic RDF schema tuples



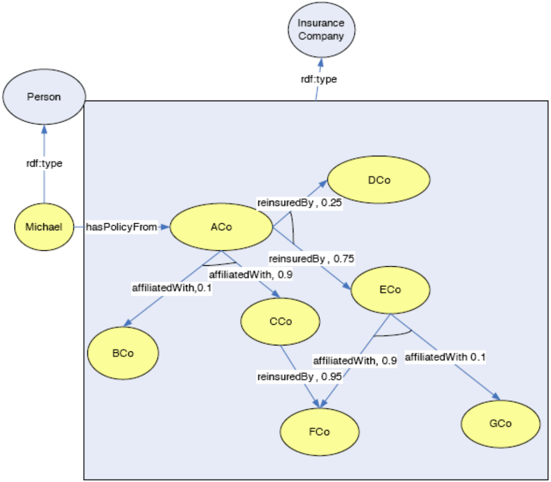
Probabilistic RDF schema tuples

- Non-probabilistic triples:

  <i, type, c>
  <p_1, subPropertyOf, p_2>
  <p, range, c>
  <p, domain, c>

  - $i \in I$ individual (URI reference or blank node)
  - p, p_i properties
  - c class
- Probabilistic schema quadruples: <c, subClassOf, C, μ>
  - c class
  - C set of classes
  - $\mu : C \to [0, 1]$ with
    - $\sum_{c \in C} \mu(c) = 1$
    - If <c, subClassOf, C_1, μ_1> and <c, subClassOf, C_1, μ_2> with $C_1 \neq C_2$ then $C_1 \cap C_2 = \emptyset$

Example of probabilistic RDF instance tuples



Probabilistic RDF instance tuples

- Probabilistic instance quadruples:

$$<i, p, V, \mu>$$
$$<i, \text{type}, C, \delta>$$

  - i individual, p property
  - $V \subseteq I \cup L$, set of individuals or literals
  - μ distribution over V, $\mu : V \to [0, 1]$ with
    - $\sum_{v \in V} \mu(v) \leq 1$
    - If $<i, p, V_1, \mu_1>$,$<i, p, V_2, \mu_2>$ with $V_1 \neq V_2$ then $V_1 \cap V_2 = \emptyset$
  - C set of classes
  - $\delta : C \to [0, 1]$ with
    - $\sum_{c \in C} \delta(c) \leq 1$
    - If $<i, \text{type}, C_1, \delta_1>$, $<i, \text{type}, C_2, \delta_2> \Rightarrow V_1 = V_2$ and $\delta_1 = \delta_2$
- pRDF theory: a pair (S, R), where S is a set of pRDF schema tuples and R is a set of pRDF instance tuples

Semantics (excerpt)

- *p*-path *P*: for property *p*, *P* is a sequence of *n* tuples <s_i, p_i, v_i, γ_i> where
  - for all i, $\exists$ <s_i, p_i, V, μ> s.t. $v_i \in V$, $\mu(v_i) = \delta_i$
  - for all i, <p_i, subPropertyOf*, p>
  - for all $i \leq n - 1$, $v_i = s_{i+1}$
- A pRDF instance is acyclic if for all properties *p*, there are no cyclic *p*-paths in it
- World: A world *w* is a set of triples <*s*, *p*, *v*> such that either
  - *s* is an individual, *p* is a property and *v* is an individual or literal, or
  - *s* is an individual, *p* is `type` and *v* is a class
- pRDF interpretation: $\mathcal{I} : W \to [0, 1]$ with $\sum_{w \in W} \mathcal{I}(w) = 1$

- Satisfaction:
  - $\mathcal{I} \models < s, p, V, \mu >$ iff $\forall v \in V, \mu(v) \leq \sum_{<s,p,v> \in W} \mathcal{I}(< s, p, v >)$
  - $\mathcal{I} \models (S, R)$ iff
    - $\mathcal{I}$ satisfies all tuples in R
    - for all p-paths $< s_i, p_i, v_i, \gamma_i >_{i \in [1 \ldots n]}$ in (S, R), $\otimes_i \gamma_i \leq \sum_{<s_i,p_i,v_i> \in W} \mathcal{I}(< s_i, p_i, v_i >)$
    - $\otimes$ is a t-norm
- Entailment: $(S, R) \models < s, p, V, \mu >$ iff any model of (S, R) is a model of $< s, p, V, \mu >$
- Atomic queries: $<?s, p, v, \gamma >, < s, ?p, v, \gamma >, < s, p, v, ?\gamma >$
- Conjunctive queries: $q_1 \wedge q_2 \wedge \ldots \wedge q_n$, q_i atomic queries

Fuzzy RDF

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

$$\langle (subject, predicate, object), n \rangle$$

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

$$\langle (o1, IsAbout, snoopy), 0.8 \rangle$$

Fuzzy RDFS semantics

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

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

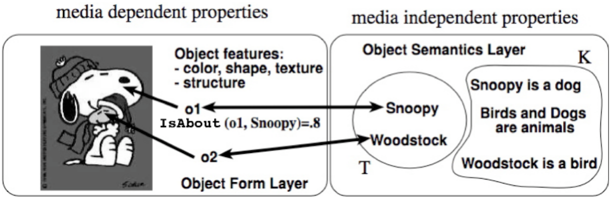
$$\frac{\langle(a, sc, b), n\rangle, \langle(b, sc, c), m\rangle}{\langle(a, sc, c), n \wedge m\rangle} \qquad \frac{\langle(a, sc, b), n\rangle, \langle(x, type, a), m\rangle}{\langle(x, type, b), n \wedge m\rangle}$$

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

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

sp = “subPropertyOf”, sc = “subClassOf”

Example



- Fuzzy RDF representation

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

- then

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

The case of Description Logics

Probabilistic DLs

- 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

- Directly extends probabilistic propositional logic
  - in place of atoms we have now concepts
- $(\psi|\varphi)[l, u]$: informally encodes that

  "generally, if an individual is an instance of φ, then it is an instance of ψ with a probability in $[l, u]$"

- $a : (\psi|\varphi)[l, u]$: informally encodes that

  "if individual a is an instance of φ, then a is an instance of ψ with a probability in $[l, u]$"

Example

$$
\begin{array}{rcl}
Eagle & \sqsubseteq & Bird \\
Penguin & \sqsubseteq & Bird
\end{array}
$$

$$
\begin{array}{l}
(Fly \mid Bird)[0.95, 1] \\
(Fly \mid Penguin)[0, 0.05]
\end{array}
$$

$$
\begin{array}{l}
KB \mid\!\mid\!\sim^{lex}_{tight} (Fly \mid Eagle)[0.95, 1] \\
KB \mid\!\mid\!\sim^{lex}_{tight} (Fly \mid Penguin)[0, 0.05]
\end{array}
$$

Possibilistic DLs

- Directly extends possibilistic propositional logic
- Expressions: $\mathrm{P}\,\alpha \geq l$ or $\mathrm{N}\,\alpha \geq l$, where α is a classical description logic axiom and $l \in [0, 1]$
- Example

$$
\begin{array}{rcl}
\mathrm{N}(\exists owns.Porsche & \sqsubseteq & CarFanatic \sqcup RichPerson) \geq 0.8 \\
\mathrm{P}(RichPerson & \sqsubseteq & Golfer) \geq 0.7 \\
\mathrm{N}((tom, 911){:}owns) & \geq & 1 \\
\mathrm{N}(911{:}Porsche) & \geq & 1 \\
\mathrm{N}(tom{:}\neg CarFanatic) & \geq & 0.7\ .
\end{array}
$$

$$KB \models \mathrm{P}(tom{:}Golfer) \geq 0.7\ .$$

Fuzzy DLs

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

Interpretation:

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

Concepts:

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

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: $\langle C \sqsubseteq D, r\rangle$,

- $\mathcal{I} \models \langle C \sqsubseteq D, r\rangle$ iff $\inf_{x \in \Delta^{\mathcal{I}}} C^{\mathcal{I}}(x) \to D^{\mathcal{I}}(x) \geq r$

Main Inference Problems

Graded entailment: Check if DL axiom α is entailed to degree at least r

- $KB \models \langle \alpha, r \rangle$?

BTVB: Best Truth Value Bound problem

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

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

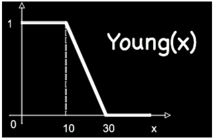
- $ans_{top-k}(KB, C) = Top_k\{\langle a, v \rangle \mid v = |a{:}C|_{KB}\}$

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

Concrete fuzzy concepts

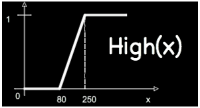
- E.g., *Small*, *Young*, *High*, *etc*. with explicit membership function
- Use the idea of concrete domains:
  - $D = \langle \Delta_D, \Phi_D \rangle$
  - Δ_D is an interpretation domain
  - Φ_D is the set of concrete fuzzy domain predicates d with a predefined arity $n = 1, 2$ and fixed interpretation $d^D \colon \Delta_D^n \to [0, 1]$
  - 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	$\longrightarrow$	$\top$ |	$\top(x)$
		$\perp$ |	$\perp(x)$
		A |	$A(x)$
		$(C \sqcap D)$ |	$C_1(x) \wedge C_2(x)$
		$(C \sqcup D)$ |	$C_1(x) \vee C_2(x)$
		$(\neg C)$ |	$\neg C(x)$
		$(\exists R.C)$ |	$\exists x\ R(x, y) \wedge C(y)$
		$(\forall R.C)$ |	$\forall x\ R(x, y) \rightarrow C(y)$
		$\{a\}$ |	$x = a$
		$(\geq n\ R)$ |	$\exists y_1, \ldots, y_n. \bigwedge_{i=1}^{n} R(x, y_i) \wedge \bigwedge_{1 \leq i < j \leq n} y_i \neq y_j$
		$(\leq n\ R)$ |	$\forall y_1, \ldots, y_{n+1}. \bigwedge_{i=1}^{n+1} R(x, y_i) \rightarrow \bigvee_{1 \leq i < j \leq n+1} y_i = y_j$
		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
		$fun(R)$ |	$\forall x \forall y \forall z\ R(x, y) \wedge R(x, z) \rightarrow y = z$
		$trans(R)$	$(\exists z\ R(x, z) \wedge R(z, y)) \rightarrow R(x, y)$

Example (Graded Entailment)

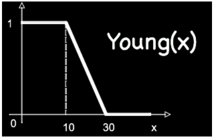
		
audi_tt	*mg*	*ferrari_enzo*

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

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

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

Example (Graded Subsumption)

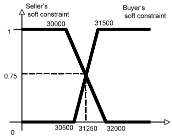


$$
\begin{array}{rcl}
Minor & = & Person \sqcap \exists hasAge. \leq_{18} \\
YoungPerson & = & Person \sqcap \exists hasAge. Young
\end{array}
$$

$$KB \models \langle Minor \sqsubseteq YoungPerson, 0.2 \rangle$$

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 €

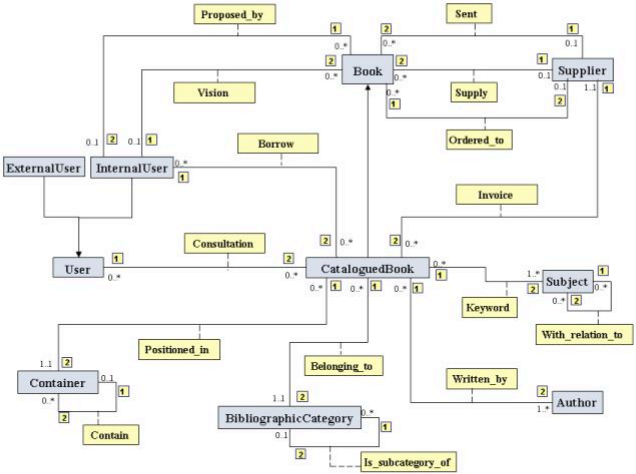
$$\begin{array}{lcl} \mathit{AudiTT} & = & \mathit{SportsCar} \sqcap \exists \mathit{hasPrice}.R(x; 30500, 31500) \\ \mathit{Query} & = & \mathit{SportsCar} \sqcap \exists \mathit{hasPrice}.L(x; 30000, 32000) \end{array}$$

  - highest degree to which the concept
    $C = \mathit{AudiTT} \sqcap \mathit{Query}$
    is satisfiable is 0.75 (the possibility that the Audi TT and the query matches is 0.75)
  - the car may be sold at 31250 €

Top-k retrieval in tractable DLs: the case of DL-Lite/DLR-Lite

- DL-Lite/DLR-Lite: 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: $Cl_1 \sqcap \ldots \sqcap Cl_n \sqsubseteq Cr$ (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: $$\begin{array}{lcl} Cl & \rightarrow & A \mid \exists R \\ Cr & \rightarrow & A \mid \exists R \mid \neg A \mid \neg \exists R \\ R & \rightarrow & P \mid P^{-} \end{array}$$
  - Assertion: $a{:}A$, $(a, b){:}P$
- DLR-Lite$_{core}$: (n-ary roles)
  - Concepts: $$\begin{array}{lcl} Cl & \rightarrow & A \mid \exists P[i] \\ Cr & \rightarrow & A \mid \exists P[i] \mid \neg A \mid \neg \exists P[i] \end{array}$$
  - $\exists P[i]$ is the projection on i-th column
  - Assertion: $a{:}A$, $\langle a_1, \ldots, a_n \rangle{:}P$
- Assertions are stored in relational tables
- Conjunctive query: $q(\mathbf{x}) \leftarrow \exists \mathbf{y}.conj(\mathbf{x}, \mathbf{y})$
  $conj$ is an aggregation of expressions of the form $B(z)$ or $P(z_1, z_2)$,


Proposed_by
Sent
Book
Supplier
Vision
Supply
Ordered_to
Borrow
ExternalUser
InternalUser
Invoice
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 $f \colon [0, 1]^n \to [0, 1]$, which combines the scores of the n fuzzy predicates $p_i(\mathbf{c}_i)$

Example:

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

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

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

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

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

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
     - $ans_{top-k}(KB, q) = Top_k(\bigcup_{q_i \in r(q,\mathcal{T})} ans_k(q_i, \mathcal{A}))$

The case of Logic Programs

Probabilistic Logic Programs under ICL

- Logic programs *P* under different “choices” (Independent Choice Logic)
- 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

- The probability of rain is 0.2

$$\begin{array}{rcl}
\mathtt{Rain}(x) & \leftarrow & \mathtt{h}_{\mathtt{Rain}}(x) \\
C_{\mathtt{Rain}} & = & \{h_{\mathtt{Rain}}(T), h_{\mathtt{Rain}}(F)\} \\
\mu(h_{\mathtt{Rain}}(T)) & = & 0.2 \\
\mu(h_{\mathtt{Rain}}(F)) & = & 0.8
\end{array}$$

- The probability of sprinkler on is 0.4

$$\begin{array}{rcl}
\mathtt{SprinklerOn}(x) & \leftarrow & \mathtt{h}_{\mathtt{SprinklerOn}}(x) \\
C_{\mathtt{SprinklerOn}} & = & \{h_{\mathtt{SprinklerOn}}(T), h_{\mathtt{SprinklerOn}}(F)\} \\
\mu(h_{\mathtt{SprinklerOn}}(T)) & = & 0.4 \\
\mu(h_{\mathtt{SprinklerOn}}(F)) & = & 0.6
\end{array}$$

- If it is raining or the sprinkler is on then the grass is wet

$$\begin{array}{rcl}
\mathtt{GrassWet}(x) & \leftarrow & \mathtt{Rain}(x) \\
\mathtt{GrassWet}(x) & \leftarrow & \mathtt{SprinklerOn}(x)
\end{array}$$

- What is the probability that the grass is wet?

Example (cont.)

- We have to sum up the probabilities of each total choice that added to the program make the query true

$$
\begin{array}{rcl}
\mathtt{Rain}(x) & \leftarrow & \mathtt{h}_{\mathtt{Rain}}(x) \\
C_{\mathtt{Rain}} & = & \{h_{\mathtt{Rain}}(T), h_{\mathtt{Rain}}(F)\} \\
\mu(h_{\mathtt{Rain}}(T)) & = & 0.2 \quad \mu(h_{\mathtt{Rain}}(F)) = 0.8 \\
 & & \\
\mathtt{SprinklerOn}(x) & \leftarrow & \mathtt{h}_{\mathtt{SprinklerOn}}(x) \\
C_{\mathtt{SprinklerOn}} & = & \{h_{\mathtt{SprinklerOn}}(T), h_{\mathtt{SprinklerOn}}(F)\} \\
\mu(h_{\mathtt{SprinklerOn}}(T)) & = & 0.4 \quad \mu(h_{\mathtt{SprinklerOn}}(F)) = 0.6 \\
 & & \\
\mathtt{GrassWet}(x) & \leftarrow & \mathtt{Rain}(x) \\
\mathtt{GrassWet}(x) & \leftarrow & \mathtt{SprinklerOn}(x)
\end{array}
$$

Example (cont.)

- Total choice: select a ground atom from each choice

$$
\begin{array}{rcl}
\mathtt{Rain}(x) & \leftarrow & \mathtt{h}_{\mathtt{Rain}}(x) \\
C_{\mathtt{Rain}} & = & \{h_{\mathtt{Rain}}(T), h_{\mathtt{Rain}}(F)\} \\
\mathtt{SprinklerOn}(x) & \leftarrow & \mathtt{h}_{\mathtt{SprinklerOn}}(x) \\
C_{\mathtt{SprinklerOn}} & = & \{h_{\mathtt{SprinklerOn}}(T), h_{\mathtt{SprinklerOn}}(F)\} \\
\mathtt{GrassWet}(x) & \leftarrow & \mathtt{Rain}(x) \\
\mathtt{GrassWet}(x) & \leftarrow & \mathtt{SprinklerOn}(x)
\end{array}
$$

B	Total choice
B_1	$h_{\mathtt{Rain}}(T), h_{\mathtt{SprinklerOn}}(T)$
B_2	$h_{\mathtt{Rain}}(T), h_{\mathtt{SprinklerOn}}(F)$
B_3	$h_{\mathtt{Rain}}(F), h_{\mathtt{SprinklerOn}}(T)$
B_4	$h_{\mathtt{Rain}}(F), h_{\mathtt{SprinklerOn}}(F)$

Example (cont.)

- Total choice B making query true: $P \cup B \models \mathtt{GrassWet}(T)$

$$
\begin{array}{rcl}
\mathtt{Rain}(x) & \leftarrow & \mathtt{h}_{\mathtt{Rain}}(x) \\
C_{\mathtt{Rain}} & = & \{h_{\mathtt{Rain}}(T), h_{\mathtt{Rain}}(F)\} \\
\mathtt{SprinklerOn}(x) & \leftarrow & \mathtt{h}_{\mathtt{SprinklerOn}}(x) \\
C_{\mathtt{SprinklerOn}} & = & \{h_{\mathtt{SprinklerOn}}(T), h_{\mathtt{SprinklerOn}}(F)\} \\
\mathtt{GrassWet}(x) & \leftarrow & \mathtt{Rain}(x) \\
\mathtt{GrassWet}(x) & \leftarrow & \mathtt{SprinklerOn}(x)
\end{array}
$$

B	Total choice	$P \cup B \models \mathtt{GrassWet}(T)$
B_1	$h_{\mathtt{Rain}}(T), h_{\mathtt{SprinklerOn}}(T)$	•
B_2	$h_{\mathtt{Rain}}(T), h_{\mathtt{SprinklerOn}}(F)$	•
B_3	$h_{\mathtt{Rain}}(F), h_{\mathtt{SprinklerOn}}(T)$	•
B_4	$h_{\mathtt{Rain}}(F), h_{\mathtt{SprinklerOn}}(F)$	

Example (cont.)

- Probability of total choice B: $\mu(B) = \Pi_{a \in B} \mu(a)$
- Condition on μ: $\sum_{a \in C} \mu(a) = 1$

$$
\begin{array}{rcl}
\mathtt{Rain}(x) & \leftarrow & \mathtt{h}_{\mathtt{Rain}}(x) \\
\mu(h_{\mathtt{Rain}}(T)) & = & 0.2 \quad \mu(h_{\mathtt{Rain}}(F)) = 0.8 \\
 & & \\
\mathtt{SprinklerOn}(x) & \leftarrow & \mathtt{h}_{\mathtt{SprinklerOn}}(x) \\
\mu(h_{\mathtt{SprinklerOn}}(T)) & = & 0.4 \quad \mu(h_{\mathtt{SprinklerOn}}(F)) = 0.6 \\
 & & \\
\mathtt{GrassWet}(x) & \leftarrow & \mathtt{Rain}(x) \\
\mathtt{GrassWet}(x) & \leftarrow & \mathtt{SprinklerOn}(x)
\end{array}
$$

B	Total choice	$P \cup B \models \mathtt{GrassWet}(T)$	$\mu(B)$
B_1	$h_{\mathtt{Rain}}(T), h_{\mathtt{SprinklerOn}}(T)$	•	0.08
B_2	$h_{\mathtt{Rain}}(T), h_{\mathtt{SprinklerOn}}(F)$	•	0.12
B_3	$h_{\mathtt{Rain}}(F), h_{\mathtt{SprinklerOn}}(T)$	•	0.32
B_4	$h_{\mathtt{Rain}}(F), h_{\mathtt{SprinklerOn}}(F)$		0.48
			1.0

Example (cont.)

- Probability of q: $Pr(q) = \sum_{B, P \cup B \models q} \mu(B)$

$$
\begin{array}{rcl}
\mathtt{Rain}(x) & \leftarrow & \mathtt{h}_{\mathtt{Rain}}(x) \\
\mu(h_{\mathtt{Rain}}(T)) & = & 0.2 \quad \mu(h_{\mathtt{Rain}}(F)) = 0.8 \\
\\
\mathtt{SprinklerOn}(x) & \leftarrow & \mathtt{h}_{\mathtt{SprinklerOn}}(x) \\
\mu(h_{\mathtt{SprinklerOn}}(T)) & = & 0.4 \quad \mu(h_{\mathtt{SprinklerOn}}(F)) = 0.6 \\
\\
\mathtt{GrassWet}(x) & \leftarrow & \mathtt{Rain}(x) \\
\mathtt{GrassWet}(x) & \leftarrow & \mathtt{SprinklerOn}(x)
\end{array}
$$

B	Total choice	$P \cup B \models \mathtt{GrassWet}(T)$	$\mu(B)$	$Pr(\mathtt{GrassWet}(T))$
B_1	$h_{\mathtt{Rain}}(T), h_{\mathtt{SprinklerOn}}(T)$	•	0.08	+
B_2	$h_{\mathtt{Rain}}(T), h_{\mathtt{SprinklerOn}}(F)$	•	0.12	+
B_3	$h_{\mathtt{Rain}}(F), h_{\mathtt{SprinklerOn}}(T)$	•	0.32	+
B_4	$h_{\mathtt{Rain}}(F), h_{\mathtt{SprinklerOn}}(F)$		0.48	
			1.0	0.52

Possibilistic Logic Programs

- Simple extension of Possibilistic necessity valued propositional logic
- Facts: $\langle P(t_1, \ldots, t_n), \mathrm{N}\, l \rangle$
- Rules: $\langle A \leftarrow B_1, \ldots, B_n, \mathrm{N}\, l \rangle$

Fuzzy LPs Basics

- 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 $R(\mathbf{x})$"

- Rules:

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

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 $L = \{0, \frac{1}{n}, \frac{2}{n} \ldots, \frac{n-1}{n}, 1\}$, 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

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

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
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

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	$\leftarrow$
$\rightarrow$	4	l	m	0.7	m	n	0.65	
	5	o	p	0.6	p	q	0.55	
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	

Action: STOP, top-1 tuple score is equal or above threshold $0.75 = \max(\min(1.0, 0.75), \min(0.7, 0.95))$

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

Systems

Systems

- RDF
  - Probability: could not find one available
  - Fuzzyness: could not find one available
- Description Logics
  - Probability: PRONTO, ContraBovemRufum
  - Fuzzyness: fuzzyDL, DLDB, DLMedia, FIRE, DeLorean,
- Logic Programming
  - Probability: ICL, PRISM, Alchemy, CILog, nFOIL, BLP, ...
  - Fuzzyness: GAP over XSB, MVLP (see Straccia) $\mapsto$ "Works for any LP system with arithmetic built-in predicates"

Conclusions & Future work

- We've overviewed basic concepts related to Uncertainty and Vagueness Representation and Reasoning in Semantic Web languages, such as
  - RDF, Description Logics, Logic Programs
- Semantic Web Applications:
  - Ontology Mappings, Multimedia Object annotation, Matchmaking, (Multimedia/Distributed) Information Retrieval, Recommender Systems, User Profiling, ...
- Future Work:
  - Standardization
  - Graphical User Interfaces
  - Top-k retrieval
  - Combination of Uncertainty and Vagueness