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
1. Concepts and Techniques for Reasoning about Vagueness and Uncertainty
   - Sources of Uncertainty and Vagueness on the Web
   - Uncertainty vs. Vagueness: a clarification

2. Basics on Semantic Web Languages
   - Web Ontology Languages
   - RDF/RDFS
   - Description Logics
   - Logic Programs

3. Uncertainty and Vagueness Basics
   - Uncertainty & Logic
   - Vagueness & Logic

4. Uncertainty and Vagueness in Semantic Web Languages
   - The case of RDF
   - The case of Description Logics
   - The case of Logic Programs

5. Systems

Uncertainty and Vagueness in Semantic Web Languages  Lecture at Reasoning Web 2008  U. Straccia
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.
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 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)

```
<table>
<thead>
<tr>
<th>IsAbout</th>
<th>ImageRegion</th>
<th>Object ID</th>
<th>degree</th>
</tr>
</thead>
<tbody>
<tr>
<td>o1</td>
<td>ImageRegion</td>
<td>snoopy</td>
<td>0.8</td>
</tr>
<tr>
<td>o2</td>
<td>ImageRegion</td>
<td>woodstock</td>
<td>0.7</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
```

“Find top-k image regions about animals”

\[
\text{Query}(x) \leftarrow \text{ImageRegion}(x) \land \text{isAbout}(x, y) \land \text{Animal}(y)
\]
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 $\mathcal{I}' \subseteq \mathcal{I}$, as it is not reasonable to assume to access to and query all resources (resource selection/resource discovery);

2. For every selected source $S_i \in \mathcal{I}'$ the agent has to reformulate its information need $Q_A$ into the query language $L_i$ 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)

```
<table>
<thead>
<tr>
<th>HotelID</th>
<th>hasLoc</th>
<th>ConferenceID</th>
<th>hasLoc</th>
</tr>
</thead>
<tbody>
<tr>
<td>h1</td>
<td>h1</td>
<td>c1</td>
<td>c1</td>
</tr>
<tr>
<td>h2</td>
<td>h2</td>
<td>c2</td>
<td>c2</td>
</tr>
<tr>
<td></td>
<td>:</td>
<td></td>
<td>:</td>
</tr>
<tr>
<td></td>
<td>:</td>
<td></td>
<td>:</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>hasLoc</th>
<th>hasLoc</th>
<th>distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>h1</td>
<td>c1</td>
<td>300</td>
</tr>
<tr>
<td>h1</td>
<td>c2</td>
<td>500</td>
</tr>
<tr>
<td>h2</td>
<td>c1</td>
<td>750</td>
</tr>
<tr>
<td>h2</td>
<td>c2</td>
<td>800</td>
</tr>
<tr>
<td></td>
<td>:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>:</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>hasLoc</th>
<th>hasLoc</th>
<th>close</th>
<th>cheap</th>
</tr>
</thead>
<tbody>
<tr>
<td>h1</td>
<td>c1</td>
<td>0.7</td>
<td>0.3</td>
</tr>
<tr>
<td>h1</td>
<td>c2</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>h2</td>
<td>c1</td>
<td>0.25</td>
<td>0.8</td>
</tr>
<tr>
<td>h2</td>
<td>c2</td>
<td>0.2</td>
<td>0.9</td>
</tr>
<tr>
<td></td>
<td>:</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

\[ q(h) \leftarrow \text{hasLocation}(h, hl) \land \text{hasLocation}(\text{train}, cl) \land \text{close}(hl, cl) \land \text{cheap}(h) \]
Example (Health-care: diagnosis of pneumonia)

Uncertainty and Vagueness in Semantic Web Languages

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:**
  \[ \forall x, \, hp, \, sp, \, ac \, \text{SportCar}(x) \iff HP(x, \, hp) \land \text{Speed}(x, \, sp) \land \text{Acceleration}(x, \, ac) \land hp \geq 210 \land sp \geq 220 \land ac \leq 7.0 \]

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

\[
\Pr(\text{SportCar}|\, \text{AudiTT}) = \Pr(\text{AudiTT}|\, \text{SportCar}) \cdot \Pr(\text{SportCar}) \cdot \frac{1}{\Pr(\text{AudiTT})} \\
\approx \frac{\Pr(\text{speed} \geq 243|\, \text{SportCar}) \cdot \Pr(\text{accel} \leq 6.9|\, \text{SportCar}) \cdot \Pr(\text{SportCar})}{\Pr(\text{speed} \geq 243) \cdot \Pr(\text{accel} \leq 6.9)}
\]
Concepts and Techniques for Reasoning about Vagueness and Uncertainty
Basics on Semantic Web Languages
Uncertainty and Vagueness Basics
Uncertainty and Vagueness in Semantic Web Languages
 Systems

Sources of Uncertainty and Vagueness on the Web
Uncertainty vs. Vagueness: a clarification

- Sport Car:

\[
\forall x, hp, sp, ac \; \text{SportCar}(x) \iff HP(x, hp) \land Speed(x, sp) \land Acceleration(x, ac) \\
\land hp \geq 210 \land sp \geq 220 \land ac \leq 7.0
\]

- 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)

\[
Pr(\text{SportCar} | \text{MyCar}) = Pr(\text{MyCar} | \text{SportCar}) \cdot Pr(\text{SportCar}) \cdot \left(1 / Pr(\text{MyCar})\right)
\]

\[
\approx \frac{Pr(\text{MyCar}.hp \geq | \text{SportCar}) \cdot Pr(\text{MyCar}.speed \geq | \text{SportCar}) \cdot Pr(\text{MyCar}.accel \leq | \text{SportCar}) \cdot Pr(\text{SportCar})}{Pr(MyCar.hp\geq) \cdot Pr(MyCar.speed\geq) \cdot Pr(MyCar.accel\leq)}
\]
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 statement 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:

\[ \forall x, hp, sp, ac \quad \text{SportCar}(x) \iff 0.3HP(x, hp) + 0.2Speed(x, sp) + 0.5Accel(x, ac) \]

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

- \( HP(x, hp) = rs(180, 250)(hp) \)
- \( Speed(x, sp) = rs(180, 240)(sp) \)
- \( Accel(x, ac) = ls(6.0, 8.0)(ac) \)

Degree of truth of \( \text{SportCar}(AudiTT) \): \( 0.1 \cdot 0.28 + 0.3 \cdot 1.0 + 0.6 \cdot 0.55 = 0.658 \)
The fuzzy membership functions can be learned from a training set (large literature)

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

Learned Training Sport Class:

\[
\forall x, hp, sp, ac \quad \text{TrainingSportCar}(x) \iff 0.3HP(x, hp) + 0.2Speed(x, sp) + 0.5Accel(x, ac)
\]

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

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

\[
\forall x, hp, sp, ac \quad \text{Similar}(x, y) \iff 0.3 \cdot HP(x, hpx) \cdot HP(y, hpy) + 0.2 \cdot Speed(x, spx) \cdot Speed(y, spy) + 0.5 \cdot Accel(x, acx) \cdot Accel(y, acy)
\]

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 height 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 \( q \)

\[
score := Pr(Rel | q, o)
\]

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

\[
F : O \times T \rightarrow [0, 1]
\]

\( F(o, t) \) 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 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

$$\langle SUV, Van, \cap, n \rangle ?$$

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, \textit{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 ...
Statements are of the form

\[ \langle \text{subject}, \text{predicate}, \text{object} \rangle \]

called triples: e.g.

\[ \langle \text{umberto}, \text{plays}, \text{soccer} \rangle \]

can be represented graphically as:

\[ \text{umberto} \xrightarrow{\text{plays}} \text{soccer} \]

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>`
How can we represent degrees of uncertainty and vagueness in RDF/RDFS?

Unfortunately, no standard exists yet

So far, an option is to use special purpose properties and reification

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

- **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: $\mathcal{ALC}$ (Attributive Language with Complement)

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Semantics</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C, D$</td>
<td>$\top$</td>
<td>$\top(x)$</td>
</tr>
<tr>
<td>$\bot$</td>
<td>$\bot(x)$</td>
<td></td>
</tr>
<tr>
<td>$A$</td>
<td>$A(x)$</td>
<td></td>
</tr>
<tr>
<td>$C \sqcap D$</td>
<td>$C(x) \land D(x)$</td>
<td>$Human \sqcap Male$</td>
</tr>
<tr>
<td>$C \sqcup D$</td>
<td>$C(x) \lor D(x)$</td>
<td>$Nice \sqcup Rich$</td>
</tr>
<tr>
<td>$\neg C$</td>
<td>$\neg C(x)$</td>
<td>$\neg Meat$</td>
</tr>
<tr>
<td>$\exists R.C$</td>
<td>$\exists y. R(x, y) \land C(y)$</td>
<td>$\exists has_child.Blon$</td>
</tr>
<tr>
<td>$\forall R.C$</td>
<td>$\forall y. R(x, y) \Rightarrow C(y)$</td>
<td>$\forall has_child.Human$</td>
</tr>
<tr>
<td>$C \sqsubseteq D$</td>
<td>$\forall x. C(x) \Rightarrow D(x)$</td>
<td>$Happy_Father \sqsubseteq Man \sqcap \exists has_child.Female$</td>
</tr>
<tr>
<td>$a:C$</td>
<td>$C(a)$</td>
<td>$John:Happy_Father$</td>
</tr>
</tbody>
</table>
Toy Example

\[
\begin{align*}
\text{Sex} & = \text{Male} \sqcup \text{Female} \\
\text{Male} \sqcap \text{Female} & \subseteq \bot \\
\text{Person} & \subseteq \text{Human} \sqcap \exists \text{hasSex}.\text{Sex} \\
\text{MalePerson} & \subseteq \text{Person} \sqcap \exists \text{hasSex}.\text{Male}
\end{align*}
\]

\[
\text{umberto:Person} \sqcap \exists \text{hasSex}.\neg \text{Female}
\]

\[
\text{KB} \models \text{umberto:MalePerson}
\]
Note on DL Naming

\[ \mathcal{AL}: \quad C, D \quad \rightarrow \quad \top | \bot | A | C \sqcap D | \neg A | \exists R. \top | \forall R. C \]

- **C**: Concept negation, \( \neg C \). Thus, \( \mathcal{ALC} = \mathcal{AL} + C \)
- **S**: Used for \( \mathcal{ALC} \) with transitive roles \( R_+ \)
- **U**: Concept disjunction, \( C_1 \sqcup C_2 \)
- **E**: Existential quantification, \( \exists R.C \)
- **H**: Role inclusion axioms, \( R_1 \sqsubseteq R_2 \), e.g. \( is\_component\_of \sqsubseteq is\_part\_of \)
- **N**: Number restrictions, \((\geq n R)\) and \((\leq n R)\), e.g. \((\geq 3 has\_Child)\) (has at least 3 children)
- **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)
- **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 \( \{a\} \sqsubseteq \exists R.\{b\} \)

- **I**: Inverse role, \( R^- \), e.g. \( is\_Part\_Of = has\_Part^- \)
- **F**: Functional role, \( f \), e.g. \( functional(has\_Age) \)

\( R_+ \): transitive role, e.g. \( transitive(is\_Part\_Of) \)

For instance,

\[
\mathcal{SHIF} = S + H + I + F = \mathcal{ALCR_+HIF} \quad \text{OWL-Lite (EXPTIME)}
\]

\[
\mathcal{SHOIN} = S + H + O + I + N = \mathcal{ALCR_+HOIN} \quad \text{OWL-DL (NEXPTIME)}
\]
Semantics of Additional Constructs

- **\( \mathcal{H} \):** Role inclusion axioms, \( \mathcal{I} \models R_1 \subseteq R_2 \) iff \( R_1^\mathcal{I} \subseteq R_2^\mathcal{I} \)

- **\( \mathcal{N} \):** 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,
  \( (\geq n \ R.C)^\mathcal{I} = \{x \in \{y \mid \langle x, y \rangle \in R^\mathcal{I} \land y \in C^\mathcal{I}\} : |\{y \mid \langle x, y \rangle \in R^\mathcal{I} \land y \in C^\mathcal{I}\}| \geq n\} \),
  \( (\leq n \ R.C)^\mathcal{I} = \{x \in \Delta^\mathcal{I} : |\{y \mid \langle x, y \rangle \in R^\mathcal{I} \land y \in C^\mathcal{I}\}| \leq n\} \)

- **\( \mathcal{O} \):** 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, \( l \models fun(f) \) iff \( \forall z \forall y \forall z \text{ if } \langle x, y \rangle \in f^\mathcal{I} \text{ and } \langle x, z \rangle \in f^\mathcal{I} \text{ the } y = z \)

- **\( \mathcal{R}_+ \):** transitive role,
  \( (R_+)^\mathcal{I} = \{\langle x, y \rangle \mid \exists z \text{ such that } \langle x, z \rangle \in R^\mathcal{I} \land \langle z, y \rangle \in R^\mathcal{I}\} \)
Concrete Domains

- **Concrete domains**: reals, integers, strings, …

  
  \((tim, 14) : has\text{Age}\)
  
  \((sf, \text{"SoftComputing"}) : has\text{Acronym}\)
  
  \((source1, \text{"ComputerScience"}) : is\text{About}\)
  
  \((service2, \text{"InformationRetrievalTool"}) : Matches\)

  \text{YoungPerson} = Person \sqcap \exists has\text{Age}. \leq 18

- Semantics: a clean separation between “object” classes and concrete domains

  \(D = \langle \Delta_D, \Phi_D \rangle\)

  \(\Delta_D\) is an interpretation domain

  \(\Phi_D\) is the set of concrete domain predicates \(d\) with a predefined arity \(n\) and \textit{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

\[ s_1 : \exists \text{hasSubject}.(\{o_1\} \sqcap \exists \text{IsAbout}.\{\text{snoopy}\}) \sqcap \exists \text{hasDegree}. = 0.8 \]

- 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

  $$P(x) \leftarrow \varphi(x, y)$$

  where $\varphi(x, y)$ is a formula built from atoms of the form $B(z)$ and connectors $\land$, $\lor$

  For instance,

  $$\text{has\_father}(x, y) \leftarrow \text{has\_parent}(x, y) \land \text{Male}(y)$$

- **Facts** are rules with empty body

  For instance,

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

$$
\begin{aligned}
A &\leftarrow \varphi_1 \\
A &\leftarrow \varphi_2 \\
&\quad \vdots \\
A &\leftarrow \varphi_n
\end{aligned}
\right\}
\text{ with } A \leftarrow \varphi_1 \lor \varphi_2 \lor \ldots \lor \varphi_n.
$$

- 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 $I : B_\mathcal{P} \rightarrow \{0, 1\}$.
- 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

\[ Q(x) \leftarrow B(x) \]
\[ Q(x) \leftarrow C(x) \]
\[ B(a) \leftarrow \]
\[ C(b) \leftarrow \]

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

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

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

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

![Image of concept map with objects and properties, including media dependent and independent properties, and an example of a statement involving a predicate with a probability value.]
Uncertainty and Vagueness Basics
Any statement $\varphi$ 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 $\varphi$ into $\{0, 1\}$

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

Distribution: a mapping

$$\mu: W \rightarrow [0, 1], \quad \mu(I) \in [0, 1]$$

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 \( \varphi \) is determined as

\[
Pr(\varphi) = \sum_{I | \models \varphi} \mu(I)
\]

The **possibility** of a statement \( \varphi \) is determined as

\[
Poss(\varphi) = \sup_{I | \models \varphi} \mu(I)
\]

The **necessity** of a statement \( \varphi \) is determined as

\[
Necc(\varphi) = 1 - Poss(\neg \varphi) = \inf_{I \not\models \varphi} 1 - \mu(I)
\]
### Example

Probabilistic setting:

\[ \varphi = \text{sprinklerOn} \lor \text{wet} \]

<table>
<thead>
<tr>
<th>( W )</th>
<th>( \text{sprinklerOn} )</th>
<th>( \text{wet} )</th>
<th>( \mu )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( I_1 )</td>
<td>0</td>
<td>0</td>
<td>0.1</td>
</tr>
<tr>
<td>( I_2 )</td>
<td>0</td>
<td>1</td>
<td>0.2</td>
</tr>
<tr>
<td>( I_3 )</td>
<td>1</td>
<td>0</td>
<td>0.4</td>
</tr>
<tr>
<td>( I_4 )</td>
<td>1</td>
<td>1</td>
<td>0.3</td>
</tr>
</tbody>
</table>

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

\[ Pr(\varphi) = 0.2 + 0.4 + 0.3 = 0.9 \]
Example

Possibilistic setting:

\[ \varphi = \text{sprinklerOn} \lor \text{wet} \]

<table>
<thead>
<tr>
<th></th>
<th>sprinklerOn</th>
<th>wet</th>
<th>( \mu )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( I_1 )</td>
<td>0</td>
<td>0</td>
<td>0.3</td>
</tr>
<tr>
<td>( I_2 )</td>
<td>0</td>
<td>1</td>
<td>1.0</td>
</tr>
<tr>
<td>( I_3 )</td>
<td>1</td>
<td>0</td>
<td>0.8</td>
</tr>
<tr>
<td>( I_4 )</td>
<td>1</td>
<td>1</td>
<td>1.0</td>
</tr>
</tbody>
</table>

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

\[ \text{Poss}(\varphi) = \sup(1.0, 0.8, 1.0) = 1.0 \]

\[ \text{Necc}(\varphi) = 1 - \text{Poss}(\neg \varphi) = 1 - 0.3 = 0.7 \]
Properties of probabilistic formulae

\[
\begin{align*}
Pr(\varphi \land \psi) &= Pr(\varphi) + Pr(\psi) - Pr(\varphi \lor \psi) \\
Pr(\varphi \land \psi) &\leq \min(Pr(\varphi), Pr(\psi)) \\
Pr(\varphi \land \psi) &\geq \max(0, Pr(\varphi) + Pr(\psi) - 1) \\
Pr(\varphi \lor \psi) &= Pr(\varphi) + Pr(\psi) - Pr(\varphi \land \psi) \\
Pr(\varphi \lor \psi) &\leq \min(1, Pr(\varphi) + Pr(\psi)) \\
Pr(\varphi \lor \psi) &\geq \max(Pr(\varphi), Pr(\psi)) \\
Pr(\neg \varphi) &= 1 - Pr(\varphi) \\
Pr(\bot) &= 0 \\
Pr(\top) &= 1
\end{align*}
\]
Properties of possibilistic formulae

\[
\begin{align*}
\text{Poss}(\varphi \land \psi) & \leq \min(\text{Poss}(\varphi), \text{Poss}(\psi)) \\
\text{Poss}(\varphi \lor \psi) & = \max(\text{Poss}(\varphi), \text{Poss}(\psi)) \\
\text{Poss}(\neg \varphi) & = 1 - \text{Nec}(\varphi) \\
\text{Poss}(\bot) & = 0 \\
\text{Poss}(\top) & = 1 \\
\text{Nec}(\varphi \land \psi) & = \min(\text{Nec}(\varphi), \text{Nec}(\psi)) \\
\text{Nec}(\varphi \lor \psi) & \geq \max(\text{Nec}(\varphi), \text{Nec}(\psi)) \\
\text{Nec}(\neg \varphi) & = 1 - \text{Poss}(\varphi) \\
\text{Nec}(\bot) & = 0 \\
\text{Nec}(\top) & = 1
\end{align*}
\]
Probabilistic Knowledge Bases

- Finite nonempty set of basic events \( \Phi = \{p_1, \ldots, p_n\} \).
- Event \( \varphi \): Boolean combination of basic events
- Logical constraint \( \psi \leftarrow \varphi \): events \( \psi \) and \( \varphi \): “\( \varphi \) implies \( \psi \)”.
- Conditional constraint \( (\psi|\varphi)[l, u] \): events \( \psi \) and \( \varphi \), and \( l, u \in [0, 1] \): “conditional probability of \( \psi \) given \( \varphi \) 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 $KB = (L, P)$:

- $L = \{bird \leftrightarrow eagle\}$:
  
  “Eagles are birds”.

- $P = \{(have\_legs | bird)[1, 1], (fly | bird)[0.95, 1]\}$:
  
  “Birds have legs”.
  “Birds fly with a probability of at least 0.95”.
- **World** $I$: truth assignment to all basic events in $\Phi$.
- $\mathcal{I}_\Phi$: all worlds for $\Phi$.
- **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 $Pr(\psi|\varphi) = Pr(\psi \land \varphi) / Pr(\varphi)$.
- **Truth under** $Pr$:
  - $Pr \models \psi \iff \varphi$ iff $Pr(\psi \land \varphi) = Pr(\varphi)$ (iff $Pr(\psi \iff \varphi) = 1$).
  - $Pr \models (\psi|\varphi)[l, u]$ iff $Pr(\psi \land \varphi) \in [l, u] \cdot Pr(\varphi)$ (iff either $Pr(\varphi) = 0$ or $Pr(\psi|\varphi) \in [l, u]$).
Example

Set of basic propositions $\Phi = \{bird, fly\}$.

$I_\Phi$ contains exactly the worlds $I_1, I_2, I_3$, and $I_4$ over $\Phi$:

<table>
<thead>
<tr>
<th></th>
<th>fly</th>
<th>¬fly</th>
</tr>
</thead>
<tbody>
<tr>
<td>bird</td>
<td>$I_1$</td>
<td>$I_2$</td>
</tr>
<tr>
<td>¬bird</td>
<td>$I_3$</td>
<td>$I_4$</td>
</tr>
</tbody>
</table>

Some probabilistic interpretations:

<table>
<thead>
<tr>
<th></th>
<th>fly</th>
<th>¬fly</th>
</tr>
</thead>
<tbody>
<tr>
<td>bird</td>
<td>19/40</td>
<td>1/40</td>
</tr>
<tr>
<td>¬bird</td>
<td>10/40</td>
<td>10/40</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>fly</th>
<th>¬fly</th>
</tr>
</thead>
<tbody>
<tr>
<td>bird</td>
<td>0</td>
<td>1/3</td>
</tr>
<tr>
<td>¬bird</td>
<td>1/3</td>
<td>1/3</td>
</tr>
</tbody>
</table>

$Pr_1(fly \land bird) = 19/40$ and $Pr_1(bird) = 20/40$.

$Pr_2(fly \land bird) = 0$ and $Pr_2(bird) = 1/3$.

$¬fly \equiv bird$ 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:

  \[ KB = (\{ \text{bird} \leftarrow \text{eagle} \}, \]
  \[ \{ (\text{have}\_\text{legs} \mid \text{bird})[1, 1], (\text{fly} \mid \text{bird})[0.95, 1] \} ) \].

- \( KB \) is satisfiable, since

  \[ \Pr \text{ with } \Pr(\text{bird} \land \text{eagle} \land \text{have}\_\text{legs} \land \text{fly}) = 1 \] is a model.

- Some conclusions under logical entailment:

  \[ KB \models (\text{have}\_\text{legs} \mid \text{bird})[0.3, 1], \ KB \models (\text{fly} \mid \text{bird})[0.6, 1]. \]

- Tight conclusions under logical entailment:

  \[ KB \models_{\text{tight}} (\text{have}\_\text{legs} \mid \text{bird})[1, 1], \ KB \models_{\text{tight}} (\text{fly} \mid \text{bird})[0.95, 1], \]
  \[ KB \models_{\text{tight}} (\text{have}\_\text{legs} \mid \text{eagle})[1, 1], \ KB \models_{\text{tight}} (\text{fly} \mid \text{eagle})[0, 1]. \]
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:

$$\begin{align*}
\sum_{r \in R, r \models \neg \psi \land \varphi} -l y_r + \sum_{r \in R, r \models \psi \land \varphi} (1 - l) y_r & \geq 0 \\
\sum_{r \in R, r \models \neg \psi \land \varphi} u y_r + \sum_{r \in R, r \models \psi \land \varphi} (u - 1) y_r & \geq 0 \\
\sum_{r \in R, r \models \alpha} y_r & = 1 \\
y_r & \geq 0 \ (\text{for all } r \in R)
\end{align*}$$
**Theorem:** Suppose $KB = (L, P)$ has a model $Pr$ such that $Pr(\alpha) > 0$. Then, $l$ (resp., $u$) such that $KB \models_{\text{tight}} (\beta | \alpha)[l, u]$ is given by the optimal value of the following linear program over the variables $y_r$ ($r \in R$), where $R = \{ l \in I_\Phi \mid l \models L \}$:

\[
\begin{align*}
\text{minimize (resp., maximize)} & \quad \sum_{r \in R, r \models \beta \land \alpha} y_r \\
\text{subject to} & \quad \sum_{r \in R, r \models \neg \psi \land \varphi} -l y_r + \sum_{r \in R, r \models \psi \land \varphi} (1 - l) y_r \geq 0 \\
& \quad \sum_{r \in R, r \models \neg \psi \land \varphi} u y_r + \sum_{r \in R, r \models \psi \land \varphi} (u - 1) y_r \geq 0 \\
& \quad \sum_{r \in R, r \models \alpha} y_r = 1 \\
& \quad y_r \geq 0 \quad (\text{for all } r \in R)
\end{align*}
\]
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 $H(Pr)$:

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

- 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:

\[ KB = (\{\text{bird} \leftrightarrow \text{eagle}\}, \{((\text{have}_\text{legs} \mid \text{bird})[1, 1], (\text{fly} \mid \text{bird})[0.95, 1])\}) . \]

Tight conclusions under logical entailment:

\[
\begin{align*}
KB & \models_{\text{tight}} (\text{have}_\text{legs} \mid \text{bird})[1, 1], \quad KB \models_{\text{tight}} (\text{fly} \mid \text{bird})[0.95, 1], \\
KB & \models_{\text{tight}} (\text{have}_\text{legs} \mid \text{eagle})[1, 1], \quad KB \models_{\text{tight}} (\text{fly} \mid \text{eagle})[0, 1].
\end{align*}
\]

Tight conclusions under maximum entropy entailment:

\[
\begin{align*}
KB & \models_{\text{tight}}^m (\text{have}_\text{legs} \mid \text{bird})[1, 1], \quad KB \models_{\text{tight}}^m (\text{fly} \mid \text{bird})[0.95, 0.95], \\
KB & \models_{\text{tight}}^m (\text{have}_\text{legs} \mid \text{eagle})[1, 1], \quad KB \models_{\text{tight}}^m (\text{fly} \mid \text{eagle})[0.95, 0.95].
\end{align*}
\]
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 $Pr$.

- $KB \models^\text{lex} (\psi \mid \varphi)[l, u]$: $(\psi \mid \varphi)[l, u]$ is a *lex*-consequence of $KB$ iff every *lex*-minimal model $Pr$ of $L$ with $Pr(\varphi) = 1$ satisfies $(\psi \mid \varphi)[l, u]$.

- $KB \models^\text{lex, tight} (\psi \mid \varphi)[l, u]$: $(\psi \mid \varphi)[l, u]$ 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:

\[ KB = (\{ \text{bird} \iff \text{eagle} \}, \]
\[ \{( \text{have}_\text{legs} \mid \text{bird})[1,1], (\text{fly} \mid \text{bird})[0.95,1] \}) \].

Tight conclusions under logical entailment:

\[ KB \models_{\text{tight}} (\text{have}_\text{legs} \mid \text{bird})[1,1], \]
\[ KB \models_{\text{tight}} (\text{fly} \mid \text{bird})[0.95,1], \]
\[ KB \models_{\text{tight}} (\text{have}_\text{legs} \mid \text{eagle})[1,1], \]
\[ KB \models_{\text{tight}} (\text{fly} \mid \text{eagle})[0,1]. \]

Tight conclusions under probabilistic lexicographic entailment:

\[ KB \models_{\text{lex}} (\text{have}_\text{legs} \mid \text{bird})[1,1], \]
\[ KB \models_{\text{lex}} (\text{fly} \mid \text{bird})[0.95,1], \]
\[ KB \models_{\text{lex}} (\text{have}_\text{legs} \mid \text{eagle})[1,1], \]
\[ KB \models_{\text{lex}} (\text{fly} \mid \text{eagle})[0.95,1]. \]
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 ≈ “directly influences”)
- a conditional distribution for each node given its parents: \( P(X_i | \text{Parents}(X_i)) \)

Any joint distribution can be represented as a BN.
The model can answer questions like “What is the probability that it is raining, given the grass is wet?”

\[
Pr(Rain = T \mid GrassWet = T) = \frac{Pr(Rain = T, GrassWet = T)}{Pr(GrassWet = T)} = \frac{\sum_{Y \in \{T,F\}} Pr(Rain = T, GrassWet = T, Sprinkler = Y)}{\sum_{Y_1,Y_2 \in \{T,F\}} Pr(GrassWet = T, (Rain = Y_1, Sprinkler = Y_2))}
\]

\[
= \frac{0.99 \cdot 0.01 \cdot 0.2 + 0.8 \cdot 0.99 \cdot 0.2}{0.99 \cdot 0.01 \cdot 0.2 + 0.9 \cdot 0.4 \cdot 0.8 + 0.8 \cdot 0.99 \cdot 0.2 + 0 \cdot 0.6 \cdot 0.8}
\]

\[
\approx 0.3577.
\]
Possibilistic Knowledge Bases

- **Possibilistic formulae** have the form $P \varphi \geq l$ or $N \varphi \geq l$, $l \in [0, 1]$
- Encode to what extent $\varphi$ is possibly resp. necessarily true
- **Possibilistic interpretation**: $\pi : I_\Phi \rightarrow [0, 1]$
- $\pi(l)$ is the degree to which the world $l$ is possible
- It is assumed that $\pi(l) = 1$ for some $l \in I_\Phi$
- **Possibility/Necessity** of an event $\varphi$ under $\pi$:
  - $Poss(\varphi) = \sup \{ \pi(l) | l \in I_\Phi, l \models \varphi \}$ (where $\max \emptyset = 0$)
  - $Necc(\varphi) = 1 - Poss(\neg \varphi)$
- **Truth under $\pi$**:
  - $\pi \models P \varphi \geq l$ iff $Poss(\varphi) \geq l$
  - $\pi \models N \varphi \geq l$ iff $Necc(\varphi) \geq l$
Deciding Logical entailment (Hollunder’s method)

- Reduction to propositional entailment
- Let

\[
\begin{align*}
KB_l &= \{ \phi \mid N \phi \geq l' \in KB, l' \geq l \} \\
KB^l &= \{ \phi \mid N \phi \geq l' \in KB, l' > l \}
\end{align*}
\]

- Then

\[
\begin{align*}
KB \models N \phi \geq l & \iff KB_l \models \phi \\
KB \models P \phi \geq l & \iff KB^0 \models \phi \text{ or } \\
& \text{there is } P \psi \geq l' \in KB \text{ with } l' \geq l, KB^{1-l'} \cup \{\psi\} \models \phi
\end{align*}
\]
Statements involve concepts for which there is no exact definition, such as
- tall, small, close, far, cheap, expensive, “is about”, “similar to”.

A statement 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 sunset 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. $\varphi \land \psi$?

E.g., if $I(\varphi) = 0.83$ and $I(\psi) = 0.2$, what is the result of $0.83 \land 0.2$?

More generally, what is the result of $n \land m$, 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 $\land, \lor, \neg, \rightarrow$

- **t-norm**: interprets conjunction
- **s-norm**: interprets disjunction

Norms are compatible with classical two-valued logic
## Axioms for t-norms and s-norms

<table>
<thead>
<tr>
<th>Axiom Name</th>
<th>T-norm</th>
<th>S-norm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tautology / Contradiction</td>
<td>$a \land 0 = 0$</td>
<td>$a \lor 1 = 1$</td>
</tr>
<tr>
<td>Identity</td>
<td>$a \land 1 = a$</td>
<td>$a \lor 0 = a$</td>
</tr>
<tr>
<td>Commutativity</td>
<td>$a \land b = b \land a$</td>
<td>$a \lor b = b \lor a$</td>
</tr>
<tr>
<td>Associativity</td>
<td>$(a \land b) \land c = a \land (b \land c)$</td>
<td>$(a \lor b) \lor c = a \lor (b \lor c)$</td>
</tr>
<tr>
<td>Monotonicity</td>
<td>if $b \leq c$, then $a \land b \leq a \land c$</td>
<td>if $b \leq c$, then $a \lor b \leq a \lor c$</td>
</tr>
</tbody>
</table>
### Axioms for implication and negation functions

<table>
<thead>
<tr>
<th>Axiom Name</th>
<th>Implication Function</th>
<th>Negation Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tautology / Contradiction</td>
<td>(0 \rightarrow b = 1) (a \rightarrow 1 = 1)</td>
<td>(\neg 0 = 1, \neg 1 = 0)</td>
</tr>
<tr>
<td>Antitonicity</td>
<td>if (a \leq b), then (a \rightarrow c \geq b \rightarrow c)</td>
<td>if (a \leq b), then (\neg a \geq \neg b)</td>
</tr>
<tr>
<td>Monotonicity</td>
<td>if (b \leq c), then (a \rightarrow b \leq a \rightarrow c)</td>
<td></td>
</tr>
</tbody>
</table>

Usually,

\[
a \rightarrow b = \sup\{c : a \land c \leq b\}
\]

is used and is called **r-implication** and depends on the t-norm only.
Typical norms

<table>
<thead>
<tr>
<th></th>
<th>Lukasiewicz Logic</th>
<th>Gödel Logic</th>
<th>Product Logic</th>
<th>Zadeh</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\neg x)</td>
<td>(1 - x)</td>
<td>if (x = 0) then 1 else 0</td>
<td>if (x = 0) then 1 else 0</td>
<td>(1 - x)</td>
</tr>
<tr>
<td>(x \land y)</td>
<td>(\max(x + y - 1, 0))</td>
<td>(\min(x, y))</td>
<td>(x \cdot y)</td>
<td>(\min(x, y))</td>
</tr>
<tr>
<td>(x \lor y)</td>
<td>(\min(x + y, 1))</td>
<td>(\max(x, y))</td>
<td>(x + y - x \cdot y)</td>
<td>(\max(x, y))</td>
</tr>
<tr>
<td>(x \Rightarrow y)</td>
<td>if (x \leq y) then 1 else (1 - x + y)</td>
<td>if (x \leq y) then 1 else (y)</td>
<td>if (x \leq y) then 1 else (\frac{y}{x})</td>
<td>(\max(1 - x, y))</td>
</tr>
</tbody>
</table>

Note: for Lukasiewicz Logic and Zadeh, \(x \Rightarrow y \equiv \neg x \lor y\)

- 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

\[
\neg_Z x = \neg_L x \\
x \land_Z y = x \land_L (x \rightarrow_L y) \\
x \rightarrow_Z y = \neg_L x \lor_L y
\]
Some additional properties of t-norms, s-norms, implication functions, and negation functions of various fuzzy logics.

<table>
<thead>
<tr>
<th>Property</th>
<th>Łukasiewicz Logic</th>
<th>Gödel Logic</th>
<th>Product Logic</th>
<th>Zadeh Logic</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x \land \neg x = 0$</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>$x \lor \neg x = 1$</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>$x \land x = x$</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>$x \lor x = x$</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>$\neg \neg x = x$</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>$x \Rightarrow y = \neg x \lor y$</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>$\neg (x \Rightarrow y) = x \land \neg y$</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>$\neg (x \land y) = \neg x \lor \neg y$</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>$\neg (x \lor y) = \neg x \land \neg y$</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>$x \land (y \lor z) = (x \land y) \lor (x \land z)$</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>$x \lor (y \land z) = (x \lor y) \land (x \lor z)$</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

**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 degree of truth in \([0, 1]\)
- **Interpretation**: is a mapping \(\mathcal{I} : Atoms \rightarrow [0, 1]\)
- Interpretations are extended to formulae using norms to interpret connectives \(\land, \lor, \neg, \rightarrow\)

\[
\begin{align*}
\mathcal{I}(\varphi \land \psi) & = \mathcal{I}(\varphi) \land \mathcal{I}(\psi) \\
\mathcal{I}(\varphi \lor \psi) & = \mathcal{I}(\varphi) \lor \mathcal{I}(\psi) \\
\mathcal{I}(\varphi \rightarrow \psi) & = \mathcal{I}(\varphi) \rightarrow \mathcal{I}(\psi) \\
\mathcal{I}(\neg \varphi) & = \neg \mathcal{I}(\varphi)
\end{align*}
\]

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

In Lukasiewicz logic:

\[ \varphi = \text{Cold} \land \text{Cloudy} \]

\[
\begin{array}{c|c|c|c}
\mathcal{I} & \text{Cold} & \text{Cloudy} & \mathcal{I}(\varphi) \\
\hline
\mathcal{I}_1 & 0 & 0.1 & \max(0, 0 + 0.1 - 1) = 0.0 \\
\mathcal{I}_2 & 0.3 & 0.4 & \max(0, 0.3 + 0.4 - 1) = 0.0 \\
\mathcal{I}_3 & 0.7 & 0.8 & \max(0, 0.7 + 0.9 - 1) = 0.6 \\
\mathcal{I}_4 & 1 & 1 & \max(0, 1 + 1 - 1) = 1.0 \\
\vdots & \vdots & \vdots & \vdots \\
\end{array}
\]
Note:

\[ I(r \rightarrow \varphi) = 1 \text{ iff } I(\varphi) \geq r \]
\[ I(\varphi \rightarrow r) = 1 \text{ iff } I(\varphi) \leq r \]

Semantics:

\[ I \models \varphi \text{ iff } I(\varphi) = 1 \]
\[ I \models KB \text{ iff } I \models \varphi \text{ for all } \varphi \in KB \]
\[ KB \models \varphi \text{ iff for all } I. \text{ if } I \models KB \text{ then } I \models \varphi \]

Deduction rule is valid: for \( r, s \in [0, 1] \):

\[ r \rightarrow \varphi, s \rightarrow (\varphi \rightarrow \psi) \models (r \land s) \rightarrow \psi \]

Informally,

From \( \varphi \geq r \) and \((\varphi \rightarrow \psi) \geq s\) infer \( \psi \geq r \land s \)
Example

In Lukasiewicz logic:

\[ \varphi = 0.4 \rightarrow (\text{Cold} \land \text{Cloudy}) \]

Read: \( \text{Cold} \land \text{Cloudy} \geq 0.4 \)

<table>
<thead>
<tr>
<th>( I )</th>
<th>Cold</th>
<th>Cloudy</th>
<th>( I(\varphi) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( I_1 )</td>
<td>0</td>
<td>0.1</td>
<td>0.4 ( \rightarrow 0.0 = \min(1,1 - 0.4 + 0.0) = 0.6 )</td>
</tr>
<tr>
<td>( I_2 )</td>
<td>0.3</td>
<td>0.4</td>
<td>0.4 ( \rightarrow 0.0 = \min(1,1 - 0.4 + 0.0) = 0.6 )</td>
</tr>
<tr>
<td>( I_3 )</td>
<td>0.7</td>
<td>0.8</td>
<td>0.4 ( \rightarrow 0.6 = \min(1,1 - 0.4 + 0.6) = 1.0 )</td>
</tr>
<tr>
<td>( I_4 )</td>
<td>1</td>
<td>1</td>
<td>0.4 ( \rightarrow 1.0 = \min(1,1 - 0.4 + 1.0) = 1.0 )</td>
</tr>
<tr>
<td>:</td>
<td>:</td>
<td>:</td>
<td>:</td>
</tr>
</tbody>
</table>
Let

$$\|\varphi\|_{KB} = \inf\{I(\varphi) \mid I \models KB\} \quad \text{(truth degree)}$$

$$\varphi|_{KB} = \sup\{r \mid KB \models r \rightarrow \varphi\} \quad \text{(provability degree)}$$

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

Also,

$$\neg \varphi|_{KB} = \sup\{r \mid KB \models r \rightarrow \varphi\} = 1 - \varphi|_{KB}$$

$$\varphi|_{KB} = \sup\{r \mid KB \models r \rightarrow \varphi\} = \inf\{s \mid KB \models \varphi \rightarrow s\}$$

**Proposition**

$$\varphi|_{KB} = \min x. \text{ such that } KB \cup \{\varphi \rightarrow x\} \text{ satisfiable.}$$
We use MILP (Mixed Integer Linear Programming) to compute $|\varphi|_{KB}$

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

For each propositional letter $p$, let $x_p$ be a variable denoting the degree of truth of $p$

Apply inference rules

<table>
<thead>
<tr>
<th>Rule</th>
<th>Inequality</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r \rightarrow p$</td>
<td>$x_p \geq r$, $x_p \in [0, 1]$</td>
</tr>
<tr>
<td>$p \rightarrow r$</td>
<td>$x_p \leq r$, $x_p \in [0, 1]$</td>
</tr>
<tr>
<td>$r \rightarrow \neg \varphi$</td>
<td>$\varphi \rightarrow \neg r$</td>
</tr>
<tr>
<td>$\neg \varphi \rightarrow r$</td>
<td>$\neg r \rightarrow \varphi$</td>
</tr>
<tr>
<td>$r \rightarrow (\varphi \land \psi)$</td>
<td>$x_1 \rightarrow \varphi$, $x_2 \rightarrow \psi$, $y \leq 1 - r$, $x_i \leq 1 - y$, $x_1 + x_2 = r + 1 - y$, $x_i \in [0, 1]$, $y \in {0, 1}$</td>
</tr>
<tr>
<td>$(\varphi \land \psi) \rightarrow r$</td>
<td>$x_1 \rightarrow \neg \varphi$, $x_2 \rightarrow \neg \psi$, $x_1 + x_2 = 1 - r$, $x_i \in [0, 1]$</td>
</tr>
<tr>
<td>$r \rightarrow (\varphi \rightarrow \psi)$</td>
<td>$\varphi \rightarrow x_1$, $x_2 \rightarrow \psi$, $r + x_1 - x_2 = 1$, $x_i \in [0, 1]$</td>
</tr>
<tr>
<td>$(\varphi \rightarrow \psi) \rightarrow r$</td>
<td>$x_1 \rightarrow \varphi$, $\psi \rightarrow x_2$, $y - r \leq 0$, $y + x_1 \leq 1$, $y \leq x_2$, $y + r + x_1 - x_2 = 1$, $x_i \in [0, 1]$, $y \in {0, 1}$</td>
</tr>
</tbody>
</table>

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

$$\min x \text{ s.t. } Ax + By \geq h$$

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

Consider $KB = \{0.6 \rightarrow p, 0.7 \rightarrow (p \rightarrow q)\}$

Let us show that $|q|_{KB} = 0.6 \land 0.7 = \max(1, 0.6 + 0.7 - 1) = 0.3$

Recall that $|q|_{KB} = \min x. \text{such that } KB \cup \{q \rightarrow x\}$ satisfiable

$KB \cup \{q \rightarrow x\} = \{0.6 \rightarrow p, 0.7 \rightarrow (p \rightarrow q), q \rightarrow x, x \in [0, 1]\}$

$\mapsto \{x_p \geq 0.6, x_q \leq x, 0.7 \rightarrow (p \rightarrow q), \{x, x_p\} \subseteq [0, 1]\}$

$\mapsto \{x_p \geq 0.6, x_q \leq x, p \rightarrow x_1, x_2 \rightarrow q, 0.7 + x_1 - x_2 = 1, \{x, x_p, x_i\} \subseteq [0, 1]\}$

$\mapsto \{x_p \geq 0.6, x_q \leq x, x_p \leq x_1, x_q \geq x_2, 0.7 + x_1 - x_2 = 1, \{x, x_p, x_i\} \subseteq [0, 1]\} = S$

It follows that $0.3 = \min x. \text{such that } Sat(S)$
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 degree of truth in $[0, 1]$
- **Interpretation**: is a mapping $\mathcal{I} : \text{Atoms} \rightarrow [0, 1]$
- Interpretations are extended to formulae as follows:
  \[
  \begin{align*}
  \mathcal{I}(\neg \phi) &= \mathcal{I}(\phi) \rightarrow 0 \\
  \mathcal{I}(\phi \land \psi) &= \mathcal{I}(\phi) \land \mathcal{I}(\psi) \\
  \mathcal{I}(\phi \rightarrow \psi) &= \mathcal{I}(\phi) \rightarrow \mathcal{I}(\psi) \\
  \mathcal{I}(\exists x \phi) &= \sup_{c \in \Delta \mathcal{I}} \mathcal{I}_x^c(\phi) \\
  \mathcal{I}(\forall x \phi) &= \inf_{c \in \Delta \mathcal{I}} \mathcal{I}_x^c(\phi)
  \end{align*}
  \]

  where $\mathcal{I}_x^c$ is as $\mathcal{I}$, except that variable $x$ is mapped into individual $c$

- Definitions of $\models \phi$, $\models I$, $\models \mathcal{T}$, $\models \phi$, $\models I$, $\models I$, $\models I$, $\models I$, $\models I$, $\models I$, $\models I$, $\models I$, $\models I$, $\models I$ and $\models I$ are as for the propositional case
\( \neg \forall x \varphi(x) \equiv \exists x \neg \varphi(x) \) true in \( \mathcal{L} \), but does not hold for logic \( G \) and \( \Pi \).

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

\[
\begin{align*}
I(\forall x p(x)) &= \inf_n 1/n = 0 \\
I(\exists x \neg p(x)) &= \sup_n -1/n = \sup 0 = 0 \\
\end{align*}
\]

**Note:** If \( I \models \exists x \varphi(x) \) then not necessarily there is \( c \in \Delta_I \) such that \( I \models \varphi(c) \).

\[ \Delta_I = \{ n \mid \text{integer } n \geq 1 \} \]

\[
\begin{align*}
I(p(n)) &= 1 - 1/n < 1, \text{ for all } n \\
I(\exists x p(x)) &= \sup_n 1 - 1/n = 1 \\
\end{align*}
\]

**Witnessed formula:** \( \exists x \varphi(x) \) is witnessed in \( I \) iff there is \( c \in \Delta_I \) such that \( I(\exists x \varphi(x)) = I(\varphi(c)) \)

(similarly for \( \forall x \varphi(x) \))

**Witnessed interpretation:** \( I \) witnessed if all quantified formulae are witnessed in \( I \)

**Proposition**

*In \( \mathcal{L} \), \( \phi \) is satisfiable iff there is a witnessed model of \( \phi \).*
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:
  - \( i, \text{type}, c \)
  - \( p_1, \text{subPropertyOf}, p_2 \)
  - \( p, \text{range}, c \)
  - \( p, \text{domain}, c \)
  - \( i \in I \) individual (URI reference or blank node)
  - \( p, p_i \) properties
  - \( c \) class

- Probabilistic schema quadruples: \( <c, \text{subClassOf}, C, \mu> \)
  - \( c \) class
  - \( C \) set of classes
  - \( \mu : C \rightarrow [0, 1] \) with
    - \( \sum_{c \in C} \mu(c) = 1 \)
    - If \( <c, \text{subClassOf}, C_1, \mu_1> \) and \( <c, \text{subClassOf}, C_1, \mu_2> \) 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
- \( \mu \) distribution over \( V \), \( \mu : V \rightarrow [0, 1] \) 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
- \( \delta : C \rightarrow [0, 1] \) with
  \[ \sum_{c \in C} \delta(c) \leq 1 \]
  If \( <i, \text{type}, C_1, \delta_1>, <i, \text{type}, C_2, \delta_2> \) then \( V_1 = V_2 \) 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 $<s_i, p_i, v_i, \gamma_i>$ where
  - for all $i$, $\exists <s_i, p_i, V, \mu>$ s.t. $v_i \in V$, $\mu(v_i) = \delta_i$
  - for all $i$, $<p_i, \text{subPropertyOf}^*, p>$
  - 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 $\text{type}$ and $v$ is a class

- **pRDF interpretation**: $\mathcal{I}: W \rightarrow [0, 1]$ with $\sum_{w \in W} \mathcal{I}(w) = 1$
Satisfaction:

- $\mathcal{I} \models \langle s, p, V, \mu \rangle$ iff
  - $\forall v \in V, \mu(v) \leq \sum_{<s,p,v> \in W} \mathcal{I}(\langle s, p, v \rangle)$
  - $\mathcal{I} \models (S, R)$ iff
    - $\mathcal{I}$ satisfies all tuples in $R$
    - for all $p$-paths $<s_i, p_i, v_i, \gamma_i>_{i \in [1...n]}$ in $(S, R)$,
      - $\otimes_i \gamma_i \leq \sum_{<s_i,p_i,v_i> \in W} \mathcal{I}(\langle s_i, p_i, v_i \rangle)$
    - $\otimes$ is a $t$-norm

Entailment: $(S, R) \models \langle s, p, V, \mu \rangle$ iff any model of $(S, R)$ is a model of $\langle s, p, V, \mu \rangle$

Atomic queries: $<?s, p, v, \gamma >, < s, ?p, v, \gamma >, < s, p, v, ?\gamma >$

Conjunctive queries: $q_1 \land q_2 \land ... \land q_n, q_i$ atomic queries
Fuzzy RDF

- Statement (triples) may have attached a degree in \([0, 1]\):
  
  \((subject, predicate, object), n\)

- 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 (→ is r-implication)

\[
\frac{(a, \text{sp}, b), n}{(a, \text{sp}, c), n \land m}
\]

\[
\frac{(a, \text{sc}, b), n}{(a, \text{sc}, c), n \land m}
\]

\[
\frac{(a, \text{dom}, b), n}{(x, \text{type}, b), n \land m}
\]

\[
\frac{(a, \text{dom}, b), n, (c, \text{sp}, a), m}{(x, \text{type}, b), n \land m \land k}
\]

\[
\frac{(a, \text{range}, b), n}{(x, \text{type}, b), n \land m}
\]

\[
\frac{(a, \text{range}, b), n, (c, \text{sp}, a), m, (x, c), k}{(y, \text{type}, b), n \land m \land k}
\]

sp = “subPropertyOf”, sc = “subClassOf”
Example

- **Fuzzy RDF representation**

\[ \langle (o_1, \text{IsAbout}, \text{snoopy}), 0.8 \rangle, \langle (\text{snoopy}, \text{type}, \text{dog}), 1.0 \rangle, \langle (\text{woodstock}, \text{type}, \text{bird}), 1.0 \rangle, \langle (\text{dog}, \text{subClassOf}, \text{Animal}), 1.0 \rangle, \langle (\text{bird}, \text{subClassOf}, \text{Animal}), 1.0 \rangle \]

- **then**

\[ KB \models \langle \exists x. (o_1, \text{IsAbout}, x) \land (x, \text{type}, \text{Animal}), 0.8 \rangle \]
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 \(\varphi\), then
  it is an instance of \(\psi\) with a probability in \([l, u]\)”

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

  “if individual \(a\) is an instance of \(\varphi\), then \(a\) is an
  instance of \(\psi\) with a probability in \([l, u]\)”
Example

\[\text{Eagle} \sqsubseteq \text{Bird}\]
\[\text{Penguin} \sqsubseteq \text{Bird}\]

\[
(Fly \mid Bird)[0.95, 1] \\
(Fly \mid Penguin)[0, 0.05]
\]

\[
KB \models_{\text{lex}}^{\text{tight}} (Fly \mid Eagle)[0.95, 1] \\
KB \models_{\text{lex}}^{\text{tight}} (Fly \mid Penguin)[0, 0.05]
\]
Possibilistic DLs

- Directly extends possibilistic propositional logic
- **Expressions**: $P\left(\alpha \geq l\right)$ or $N\left(\alpha \geq l\right)$, where $\alpha$ is a classical description logic axiom and $l \in [0, 1]$

**Example**

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

$$KB \models P(tom: Golfer) \geq 0.7 .$$
### Fuzzy DLs

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

\[
\begin{align*}
\mathcal{I} & = \Delta^\mathcal{I} \\
\wedge & = \text{t-norm} \\
\vee & = \text{s-norm} \\
\neg & = \text{negation} \\
\rightarrow & = \text{implication}
\end{align*}
\]

**Interpretation:**

- \(C^\mathcal{I} : \Delta^\mathcal{I} \rightarrow [0, 1]\)
- \(R^\mathcal{I} : \Delta^\mathcal{I} \times \Delta^\mathcal{I} \rightarrow [0, 1]\)

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

**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:**

\(\langle C \sqsubseteq D, r \rangle,\)

- \(\mathcal{I} \models \langle C \sqsubseteq D, r \rangle\) iff \(\inf_{x \in \Delta^\mathcal{I}} C^\mathcal{I}(x) \rightarrow D^\mathcal{I}(x) \geq r\)
Main Inference Problems

**Graded entailment:** Check if DL axiom $\alpha$ is entailed to degree at least $r$

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

**BTVB:** Best Truth Value Bound problem

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

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

$$\text{ans}_{top-k}(KB, C) = Top_k\{\langle a, v \rangle | v = |a:C|_{KB} \}$$
Towards fuzzy OWL Lite and OWL DL

- Recall that OWL Lite and OWL DL relate to $SHIF(D)$ and $SHOIN(D)$, respectively.
- We need to extend the semantics of fuzzy $ALC$ to fuzzy $SHOIN(D) = ALC\text{CHOIN R}_+(D)$.
- 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:
  - $D = \langle \Delta_D, \Phi_D \rangle$
  - $\Delta_D$ is an interpretation domain
  - $\Phi_D$ is the set of concrete fuzzy domain predicates $d$ with a predefined arity $n = 1, 2$ and fixed interpretation $d^D: \Delta^n_D \rightarrow [0, 1]$
  - For instance,

\[
\begin{align*}
\text{Minor} &= \text{Person} \sqcap \exists\text{hasAge}. \leq 18 \\
\text{YoungPerson} &= \text{Person} \sqcap \exists\text{hasAge}. \text{Young} \quad \text{functional}(\text{hasAge})
\end{align*}
\]
Modifiers

- *Very, moreOrLess, slightly*, etc.

- Apply to fuzzy sets to change their membership function
  - \(\text{very}(x) = x^2\)
  - \(\text{slightly}(x) = \sqrt{x}\)

- For instance,

\[
\text{SportsCar} = \text{Car} \sqcap \exists \text{speed. very}(\text{High})
\]
### Fuzzy $SHOIN(D)$

**Concepts:**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\top$</td>
<td>$\top(x)$</td>
</tr>
<tr>
<td>$\bot$</td>
<td>$\bot(x)$</td>
</tr>
<tr>
<td>$A$</td>
<td>$A(x)$</td>
</tr>
<tr>
<td>$(C \sqcap D)$</td>
<td>$C_1(x) \land C_2(x)$</td>
</tr>
<tr>
<td>$(C \sqcup D)$</td>
<td>$C_1(x) \lor C_2(x)$</td>
</tr>
<tr>
<td>$(\neg C)$</td>
<td>$\neg C(x)$</td>
</tr>
<tr>
<td>$(\exists R.C)$</td>
<td>$\exists x R(x, y) \land C(y)$</td>
</tr>
<tr>
<td>$(\forall R.C)$</td>
<td>$\forall x R(x, y) \rightarrow C(y)$</td>
</tr>
<tr>
<td>${a}$</td>
<td>$x = a$</td>
</tr>
<tr>
<td>$(\geq n R)$</td>
<td>$\exists y_1, \ldots, y_n \cdot \bigwedge_{i=1}^{n} R(x, y_i) \land \bigwedge_{1 \leq i &lt; j \leq n} y_i \neq y_j$</td>
</tr>
<tr>
<td>$(\leq n R)$</td>
<td>$\forall y_1, \ldots, y_{n+1} \cdot \bigwedge_{i=1}^{n+1} R(x, y_i) \rightarrow \bigvee_{1 \leq i &lt; j \leq n+1} y_i = y_j$</td>
</tr>
<tr>
<td>$FCC$</td>
<td>$\mu_{FCC}(x)$</td>
</tr>
<tr>
<td>$M(C)$</td>
<td>$\mu_M(C(x))$</td>
</tr>
<tr>
<td>$R \rightarrow P$</td>
<td>$P(x, y)$</td>
</tr>
<tr>
<td>$P^-$</td>
<td>$P(y, x)$</td>
</tr>
</tbody>
</table>

**Assertions:**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha$</td>
<td>$\langle a; C, r \rangle$</td>
</tr>
<tr>
<td>$\langle (a, b): R, r \rangle$</td>
<td>$r \rightarrow C(a)$</td>
</tr>
<tr>
<td>$r \rightarrow R(a, b)$</td>
<td></td>
</tr>
</tbody>
</table>

**Axioms:**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tau$</td>
<td>$\langle C \sqsubseteq D, r \rangle$</td>
</tr>
<tr>
<td>$\forall x \ r \rightarrow (C(x) \rightarrow D(x))$, where $\rightarrow$ is r-implication</td>
<td></td>
</tr>
<tr>
<td>$\forall x \forall y \forall z \ R(x, y) \land R(x, z) \rightarrow y = z$</td>
<td></td>
</tr>
<tr>
<td>$(\exists z R(x, z) \land R(z, y)) \rightarrow R(x, y)$</td>
<td></td>
</tr>
</tbody>
</table>
Example (Graded Entailment)

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>audi_tt</td>
<td>mg</td>
<td>ferrari_enzo</td>
</tr>
</tbody>
</table>

\[
\text{SportsCar} = \text{Car} \sqcap \exists \text{hasSpeed}.\text{very}(\text{High})
\]

\[
\begin{array}{|c|c|}
\hline
\text{Car} & \text{speed} \\
\hline
\text{audi_tt} & 243 \\
\text{mg} & \leq 170 \\
\text{ferrari_enzo} & \geq 350 \\
\hline
\end{array}
\]

\[
\begin{align*}
\text{KB} & \models \langle \text{ferrari_enzo:SportsCar}, 1 \rangle \\
\text{KB} & \models \langle \text{audi_tt:SportsCar}, 0.92 \rangle \\
\text{KB} & \models \langle \text{mg:\neg SportsCar}, 0.72 \rangle 
\end{align*}
\]
Example (Graded Subsumption)

\[ \text{Minor} = \text{Person} \sqcap \exists \text{hasAge. } \leq 18 \]
\[ \text{YoungPerson} = \text{Person} \sqcap \exists \text{hasAge. Young} \]

\[ KB \models \langle \text{Minor} \sqsubseteq \text{YoungPerson}, 0.2 \rangle \]

Note: without an explicit membership function of \text{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 €
    - \( \text{AudiTT} = \text{SportsCar} \sqcap \exists \text{hasPrice}.R(x; 30500, 31500) \)
    - \( \text{Query} = \text{SportsCar} \sqcap \exists \text{hasPrice}.L(x; 30000, 32000) \)
  - highest degree to which the concept \( C = \text{AudiTT} \sqcap \text{Query} \) 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: $KB = \langle T, A \rangle$, where $T$ and $A$ are finite sets of axioms and assertions.

Axiom: $Cl_1 \cap \ldots \cap Cl_n \subseteq Cr$ (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: $Cl \rightarrow A \mid \exists R$
  $Cr \rightarrow A \mid \exists R \mid \neg A \mid \neg \exists R$
  $R \rightarrow P \mid P^-$
- Assertion: $a:A, (a, b):P$

DLR-Lite$_{core}$: ($n$-ary roles)
- Concepts: $Cl \rightarrow A \mid \exists P[i]$
  $Cr \rightarrow A \mid \exists P[i] \mid \neg A \mid \neg \exists P[i]$
  $\exists P[i]$ 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: $q(x) \leftarrow \exists y.\text{conj}(x, y)$
$\text{conj}$ is an aggregation of expressions of the form $B(z)$ or $P(z_1, z_2)$,
Concepts and Techniques for Reasoning about Vagueness and Uncertainty in Semantic Web Languages

Uncertainty and Vagueness Basics

Uncertainty and Vagueness in Semantic Web Languages

Systems

The case of RDF

The case of Description Logics

The case of Logic Programs

Lecture at Reasoning Web 2008

U. Straccia
Examples:

- \textit{isa} \quad \text{CatalogueBook} \sqsubseteq \text{Book}
- \textit{disjointness} \quad \text{Book} \sqsubseteq \neg \text{Author}
- \textit{constraints} \quad \text{CatalogueBook} \sqsubseteq \exists \text{positioned\_In}
- \textit{role – typing} \quad \exists \text{positioned\_In} \sqsubseteq \text{Container}
- \textit{functional} \quad \text{fun(positioned\_In)}
- \textit{constraints} \quad \text{Author} \sqsubseteq \exists \text{written\_By}\neg
  \quad \exists \text{written\_By} \sqsubseteq \text{CatalogueBook}

- \textit{assertion} \quad \text{Romeo\_and\_Juliet}:\text{CatalogueBook}
  \quad (\text{Romeo\_and\_Juliet}, \text{Shakespeare}):\text{written\_By}

- \textit{query} \quad q(x, y) \leftarrow \text{CataloguedBook}(x), \text{Ordered\_to}(x, y)

- \textbf{Consistency check} is linear time in the size of the KB
- \textbf{Query answering} in linear in in the size of the number of assertions
Top-\(k\) retrieval in DL-Lite/DLR-Lite

1. We extend the query formalism: conjunctive queries, where fuzzy predicates may appear
2. conjunctive query

\[ q(x, s) \leftarrow \exists y. \text{conj}(x, y), \ s = f(p_1(z_1), \ldots, p_n(z_n)) \]

1. \(x\) are the distinguished variables;
2. \(s\) is the score variable, taking values in \([0, 1]\);
3. \(y\) are existentially quantified variables, called non-distinguished variables;
4. \(\text{conj}(x, y)\) is a conjunction of DL-Lite/DLR-Lite atoms \(R(z)\) in \(KB\);
5. \(z\) are tuples of constants in \(KB\) or variables in \(x\) or \(y\);
6. \(z_i\) are tuples of constants in \(KB\) or variables in \(x\) or \(y\);
7. \(p_i\) is an \(n_i\)-ary fuzzy predicate assigning to each \(n_i\)-ary tuple \(c_i\) the score \(p_i(c_i) \in [0, 1]\);
8. \(f\) is a monotone scoring function \(f: [0, 1]^n \rightarrow [0, 1]\), which combines the scores of the \(n\) fuzzy predicates \(p_i(c_i)\)
Example:

```
q(h, s) ← HasHLoc(h, h1), HasHPrice(h, p), Distance(h1, cl, d)
HasCLoc(c1, cl), s = cheap(p) · close(d).
```

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

```
close(d) = ls(0, 2km)(d)
cheap(p) = ls(0, 300)(p)
```
Semantics informally:

- a conjunctive query

\[
q(x, s) \leftarrow \exists y.\text{conj}(x, y), s = f(p_1(z_1), \ldots, p_n(z_n))
\]

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

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

such that when we consider the substitution

\[
\theta = \{ x/c, s/v \}
\]

the formula

\[
\exists y.\text{conj}(x, y) \land s = f(p_1(z_1), \ldots, p_n(z_n))
\]

evaluates to true in \( \mathcal{I} \).

- Model of a query: \( \mathcal{I} \models q(c, v) \) iff \( \langle c, v \rangle \in q^\mathcal{I} \)

- Entailment: \( KB \models q(c, v) \) iff \( \mathcal{I} \models KB \) implies \( \mathcal{I} \models q(c, v) \)

- Top-k retrieval: \( \text{ans}_{\text{top-k}}(KB, q) = \text{Top}_k \{ \langle c, v \rangle \mid KB \models q(c, v) \} \)
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 $r(q, T)$
   - Basic idea: reformulation procedure closely resembles a top-down resolution procedure for logic programming

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

\[
\begin{align*}
\text{Rain}(x) & \leftarrow h_{\text{Rain}}(x) \\
C_{\text{Rain}} & = \{h_{\text{Rain}}(T), h_{\text{Rain}}(F)\} \\
\mu(h_{\text{Rain}}(T)) & = 0.2 \\
\mu(h_{\text{Rain}}(F)) & = 0.8
\end{align*}
\]

- The probability of sprinkler on is 0.4

\[
\begin{align*}
\text{SprinklerOn}(x) & \leftarrow h_{\text{SprinklerOn}}(x) \\
C_{\text{SprinklerOn}} & = \{h_{\text{SprinklerOn}}(T), h_{\text{SprinklerOn}}(F)\} \\
\mu(h_{\text{SprinklerOn}}(T)) & = 0.4 \\
\mu(h_{\text{SprinklerOn}}(F)) & = 0.6
\end{align*}
\]

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

\[
\begin{align*}
\text{GrassWet}(x) & \leftarrow \text{Rain}(x) \\
\text{GrassWet}(x) & \leftarrow \text{SprinklerOn}(x)
\end{align*}
\]

- 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

\[
\begin{align*}
\text{Rain}(x) & \leftarrow h_{\text{Rain}}(x) \\
C_{\text{Rain}} & = \{ h_{\text{Rain}}(T), h_{\text{Rain}}(F) \} \\
\mu(h_{\text{Rain}}(T)) & = 0.2 \quad \mu(h_{\text{Rain}}(F)) = 0.8 \\
\text{SprinklerOn}(x) & \leftarrow h_{\text{SprinklerOn}}(x) \\
C_{\text{SprinklerOn}} & = \{ h_{\text{SprinklerOn}}(T), h_{\text{SprinklerOn}}(F) \} \\
\mu(h_{\text{SprinklerOn}}(T)) & = 0.4 \quad \mu(h_{\text{SprinklerOn}}(F)) = 0.6 \\
\text{GrassWet}(x) & \leftarrow \text{Rain}(x) \\
\text{GrassWet}(x) & \leftarrow \text{SprinklerOn}(x)
\end{align*}
\]
Example (cont.)

- **Total choice**: select a ground atom from each choice

\[
\begin{align*}
\text{Rain}(x) & \leftarrow \text{h}_{\text{Rain}}(x) \\
C_{\text{Rain}} & = \{ \text{h}_{\text{Rain}}(T), \text{h}_{\text{Rain}}(F) \} \\
\text{SprinklerOn}(x) & \leftarrow \text{h}_{\text{SprinklerOn}}(x) \\
C_{\text{SprinklerOn}} & = \{ \text{h}_{\text{SprinklerOn}}(T), \text{h}_{\text{SprinklerOn}}(F) \} \\
\text{GrassWet}(x) & \leftarrow \text{Rain}(x) \\
\text{GrassWet}(x) & \leftarrow \text{SprinklerOn}(x)
\end{align*}
\]

<table>
<thead>
<tr>
<th>$B$</th>
<th>Total choice</th>
</tr>
</thead>
<tbody>
<tr>
<td>$B_1$</td>
<td>$\text{h}<em>{\text{Rain}}(T), \text{h}</em>{\text{SprinklerOn}}(T)$</td>
</tr>
<tr>
<td>$B_2$</td>
<td>$\text{h}<em>{\text{Rain}}(T), \text{h}</em>{\text{SprinklerOn}}(F)$</td>
</tr>
<tr>
<td>$B_3$</td>
<td>$\text{h}<em>{\text{Rain}}(F), \text{h}</em>{\text{SprinklerOn}}(T)$</td>
</tr>
<tr>
<td>$B_4$</td>
<td>$\text{h}<em>{\text{Rain}}(F), \text{h}</em>{\text{SprinklerOn}}(F)$</td>
</tr>
</tbody>
</table>
Example (cont.)

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

\[
\begin{align*}
\text{Rain}(x) & \leftarrow h_{\text{Rain}}(x) \\
C_{\text{Rain}} & = \{h_{\text{Rain}}(T), h_{\text{Rain}}(F)\} \\
\text{SprinklerOn}(x) & \leftarrow h_{\text{SprinklerOn}}(x) \\
C_{\text{SprinklerOn}} & = \{h_{\text{SprinklerOn}}(T), h_{\text{SprinklerOn}}(F)\} \\
\text{GrassWet}(x) & \leftarrow \text{Rain}(x) \\
\text{GrassWet}(x) & \leftarrow \text{SprinklerOn}(x)
\end{align*}
\]
Example (cont.)

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

\[
\begin{align*}
\text{Rain}(x) & \leftarrow h_{\text{Rain}}(x) \\
\mu(h_{\text{Rain}}(T)) & = 0.2 \quad \mu(h_{\text{Rain}}(F)) = 0.8 \\
\text{SprinklerOn}(x) & \leftarrow h_{\text{SprinklerOn}}(x) \\
\mu(h_{\text{SprinklerOn}}(T)) & = 0.4 \quad \mu(h_{\text{SprinklerOn}}(F)) = 0.6 \\
\text{GrassWet}(x) & \leftarrow \text{Rain}(x) \\
\text{GrassWet}(x) & \leftarrow \text{SprinklerOn}(x)
\end{align*}
\]

<table>
<thead>
<tr>
<th>$B$</th>
<th>Total choice</th>
<th>$P \cup B \models \text{GrassWet}(T)$</th>
<th>$\mu(B)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$B_1$</td>
<td>$h_{\text{Rain}}(T), h_{\text{SprinklerOn}}(T)$</td>
<td>$\bullet$</td>
<td>0.08</td>
</tr>
<tr>
<td>$B_2$</td>
<td>$h_{\text{Rain}}(T), h_{\text{SprinklerOn}}(F)$</td>
<td>$\bullet$</td>
<td>0.12</td>
</tr>
<tr>
<td>$B_3$</td>
<td>$h_{\text{Rain}}(F), h_{\text{SprinklerOn}}(T)$</td>
<td>$\bullet$</td>
<td>0.32</td>
</tr>
<tr>
<td>$B_4$</td>
<td>$h_{\text{Rain}}(F), h_{\text{SprinklerOn}}(F)$</td>
<td></td>
<td>0.48</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td>$\sum$</td>
<td>1.0</td>
</tr>
</tbody>
</table>
Example (cont.)

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

<table>
<thead>
<tr>
<th>$B$</th>
<th>Total choice</th>
<th>$P \cup B \models GrassWet(T)$</th>
<th>$\mu(B)$</th>
<th>$Pr(GrassWet(T))$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$B_1$</td>
<td>$h_{Rain(T)}, h_{SprinklerOn(T)}$</td>
<td>$\bullet$</td>
<td>0.08</td>
<td>+</td>
</tr>
<tr>
<td>$B_2$</td>
<td>$h_{Rain(T)}, h_{SprinklerOn(F)}$</td>
<td>$\bullet$</td>
<td>0.12</td>
<td>+</td>
</tr>
<tr>
<td>$B_3$</td>
<td>$h_{Rain(F)}, h_{SprinklerOn(T)}$</td>
<td>$\bullet$</td>
<td>0.32</td>
<td>+</td>
</tr>
<tr>
<td>$B_4$</td>
<td>$h_{Rain(F)}, h_{SprinklerOn(F)}$</td>
<td>$\bullet$</td>
<td>0.48</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1.0</td>
<td>0.52</td>
</tr>
</tbody>
</table>
Possibilistic Logic Programs

- Simple extension of Possibilistic necessity valued propositional logic
- **Facts**: $\langle P(t_1, \ldots, t_n), N I \rangle$
- **Rules**: $\langle A \leftarrow B_1, \ldots, B_n, N I \rangle$
Fuzzy LPs Basics

- We consider fuzzy LPs, where
  - **Truth space** is $[0, 1]$
  - **Interpretation** is a mapping $I : B_P \rightarrow [0, 1]$
  - **Generalized LP rules** are of the form

$$R(x) \leftarrow \exists y. f(R_1(z_1), \ldots, R_l(z_l), p_1(z'_1), \ldots, p_h(z'_h)),$$

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

\[ R(x, s) \leftarrow \exists y.\text{conj}(x, y), s = f(p_1(z_1), \ldots, p_{l+h}(z_{l+h})) \]

1. \( x \) are the distinguished variables;
2. \( s \) is the score variable, taking values in \([0, 1]\);
3. \( y \) are existentially quantified variables, called non-distinguished variables;
4. \( \text{conj}(x, y) \) is a list of atoms \( R_i(z) \) in \( KB \);
5. \( z \) are tuples of constants in \( KB \) or variables in \( x \) or \( y \);
6. \( z_i \) are tuples of constants in \( KB \) or variables in \( x \) or \( y \);
7. \( p_i \) is an \( n_i \)-ary fuzzy predicate assigning to each \( n_i \)-ary tuple \( c_i \) the score \( p_i(c_i) \in [0, 1] \);
8. \( f \) is a monotone scoring function \( f : [0, 1]^{l+h} \rightarrow [0, 1] \), which combines the scores of the \( n \) fuzzy predicates \( p_i(c_i) \).
Semantics of fuzzy LPs

- **Model** of a LP:
  
  \[ I \models P \iff I \models r, \text{ for all } r \in \mathcal{P}^* \]
  
  \[ I \models A \leftarrow \varphi \iff I(\varphi) \leq I(A) \]

- **Least model** exists and is least fixed-point of
  
  \[ T_P(I)(A) = I(\varphi) \]
  
  for all \( A \leftarrow \varphi \in \mathcal{P}^* \)

- **Fuzzy LPs** may be tricky:
  
  \[ \langle A, 0 \rangle \]
  
  \[ A \leftarrow (A + 1)/2 \]

  In the minimal model the truth of \( A \) is 1 (requires \( \omega \) \( T_P \) iterations)!

- There are several ways to avoid this pathological behavior:
  
  - We consider \( L = \{0, 1/n, 2/n, \ldots, (n-1)/n, 1\} \), \( n \) natural number, e.g. \( n = 100 \)
Example: Soft shopping agent

I may represent my preferences in Logic Programming with the rules

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

<table>
<thead>
<tr>
<th>ID</th>
<th>MODEL</th>
<th>PRICE</th>
<th>KM</th>
</tr>
</thead>
<tbody>
<tr>
<td>455</td>
<td>MAZDA 3</td>
<td>12500</td>
<td>10000</td>
</tr>
<tr>
<td>34</td>
<td>ALFA 156</td>
<td>12000</td>
<td>15000</td>
</tr>
<tr>
<td>1812</td>
<td>FORD FOCUS</td>
<td>11000</td>
<td>16000</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>ID</th>
<th>PRICE</th>
<th>SCORE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1812</td>
<td>11000</td>
<td>0.6</td>
</tr>
<tr>
<td>455</td>
<td>12500</td>
<td>0.56</td>
</tr>
<tr>
<td>34</td>
<td>12000</td>
<td>0.50</td>
</tr>
</tbody>
</table>
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 $\delta$.
- The threshold $\delta$ 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 $\delta$, because any successively computed answer will have a score below $\delta$. 
Example

Logic Program \( \mathcal{P} \) is

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

<table>
<thead>
<tr>
<th>RecordID</th>
<th>( r_1 )</th>
<th>( r_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>a b 1.0</td>
<td>m h 0.95</td>
</tr>
<tr>
<td>2</td>
<td>c d 0.9</td>
<td>m j 0.85</td>
</tr>
<tr>
<td>3</td>
<td>e f 0.8</td>
<td>f k 0.75</td>
</tr>
<tr>
<td>4</td>
<td>l m 0.7</td>
<td>m n 0.65</td>
</tr>
<tr>
<td>5</td>
<td>o p 0.6</td>
<td>p q 0.55</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

What is

\[
\text{Top}_1(\mathcal{P}, q) = \text{Top}_1\{\langle c, s \rangle \mid \mathcal{P} \models q(c, s)\}?
\]
Uncertainty and Vagueness in Semantic Web Languages

**Lecture at Reasoning Web 2008**  
U. Straccia

---

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

<table>
<thead>
<tr>
<th>RecordID</th>
<th>(r_1)</th>
<th>(r_2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>a b 1.0</td>
<td>m h 0.95</td>
</tr>
<tr>
<td>2</td>
<td>c d 0.9</td>
<td>m j 0.85</td>
</tr>
<tr>
<td>3</td>
<td>e f 0.8</td>
<td>f k 0.75</td>
</tr>
<tr>
<td>4</td>
<td>l m 0.7</td>
<td>m n 0.65</td>
</tr>
<tr>
<td>5</td>
<td>o p 0.6</td>
<td>p q 0.55</td>
</tr>
</tbody>
</table>

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

\[
\text{Queue} \quad | \quad \delta \\
\hline
- \quad | \quad 0.75 \\
\hline
\text{Predicate} \quad | \quad \text{Answers} \\
\hline
q \quad | \quad \langle e, 0.75 \rangle, \langle l, 0.7 \rangle \\
p \quad | \quad \langle e, 0.75 \rangle, \langle l, 0.7 \rangle \\
\hline
\]

\[
\text{Top}_1(\mathcal{P}, q) = \{\langle e, 0.75 \rangle\}
\]

Note: no further answer will have score above threshold \(\delta\)
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”
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:

Future Work:
- Standardization
- Graphical User Interfaces
- Top-k retrieval
- Combination of Uncertainty and Vagueness