Managing Uncertainty and Vagueness in Semantic Web Languages

Tutorial at SWAP-2007

Umberto Straccia

ISTI-CNR, Pisa, Italy
straccia@isti.cnr.it
1. Uncertainty, Vagueness, and the Semantic Web
   - 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
   - Description Logic Programs

3. Uncertainty in Semantic Web Languages
   - Uncertainty
   - Uncertainty and RDF/DLs/OWL
   - Uncertainty and LPs/DLPs

4. Vagueness in Semantic Web Languages
   - Vagueness basics
   - Vagueness and RDF/DLs
   - Vagueness and LPs/DLPs

5. Combining Uncertainty and Vagueness in the Semantic Web
1. Uncertainty, Vagueness, and the Semantic Web
   - 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
   - Description Logic Programs

3. Uncertainty in Semantic Web Languages
   - Uncertainty
   - Uncertainty and RDF/DLs/OWL
   - Uncertainty and LPs/DLPs

4. Vagueness in Semantic Web Languages
   - Vagueness basics
   - Vagueness and RDF/DLs
   - Vagueness and LPs/DLPs

5. Combining Uncertainty and Vagueness in the Semantic Web
Sources of Uncertainty and Vagueness on the Web

- Information Retrieval:
  - To which **degree** is a Web site, a Web page, a text passage, an image region, a video segment, ... relevant to my information need?

- Matchmaking
  - To which **degree** does an object match my requirements?
    - if I’m looking for a car and my budget is **about** 20,000 €, to which degree does a car’s price of 20,500 € match my budget?
Semantic annotation
- To which **degree** does e.g., an image object represent a dog?

Information extraction
- To which **degree** am I’m sure that e.g., SW is an acronym of “Semantic Web”?

Ontology alignment (schema mapping)
- To which **degree** do two concepts of two ontologies represent the same, or are disjoint, or are overlapping?

Representation of background knowledge
- To some **degree** birds fly.
- To some **degree** Jim is a blond and young.
Example (Distributed Information Retrieval) [7]

Then the agent has to perform automatically the following steps:

1. The agent has to select a subset of relevant resources \( S' \subseteq S \), as it is not reasonable to assume to access to and query all resources (resource selection/resource discovery);

2. For every selected source \( S_i \in S' \) 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)
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 (Logic-based information retrieval model) [1, 8]

```
<table>
<thead>
<tr>
<th>IsAbout</th>
<th>ImageRegion</th>
<th>Object ID</th>
<th>degree</th>
</tr>
</thead>
<tbody>
<tr>
<td>o1</td>
<td>isAbout</td>
<td>snoopy</td>
<td>0.8</td>
</tr>
<tr>
<td>o2</td>
<td>isAbout</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"

Query(x) ← ImageRegion(x) ∧ isAbout(x, y) ∧ Animal(y)
Example (Database query) \([3, 4, 5, 6]\)

```
<table>
<thead>
<tr>
<th>HotelID</th>
<th>hasLoc</th>
<th>ConferenceID</th>
<th>hasLoc</th>
</tr>
</thead>
<tbody>
<tr>
<td>h1</td>
<td>hl1</td>
<td>c1</td>
<td>cl1</td>
</tr>
<tr>
<td>h2</td>
<td>hl2</td>
<td>c2</td>
<td>cl2</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>hl1</td>
<td>cl1</td>
<td>300</td>
</tr>
<tr>
<td>hl1</td>
<td>cl2</td>
<td>500</td>
</tr>
<tr>
<td>hl2</td>
<td>cl1</td>
<td>750</td>
</tr>
<tr>
<td>hl2</td>
<td>cl2</td>
<td>800</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>hl1</td>
<td>cl1</td>
<td>0.7</td>
<td>0.3</td>
</tr>
<tr>
<td>hl1</td>
<td>cl2</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>hl2</td>
<td>cl1</td>
<td>0.25</td>
<td>0.8</td>
</tr>
<tr>
<td>hl2</td>
<td>cl2</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)

E.g., Temp = 37.5, Pulse = 98, Respiratory Rate = 18 are in the “danger zone” already.

Temperature, Pulse and Respiratory rate, . . . : these constraints are rather imprecise than crisp.
Uncertainty vs. Vagueness: a clarification

- What does the **degree** mean?
- There is often a misunderstanding between interpreting a degree as a measure of **uncertainty** or as a measure of **vagueness**
- The value 0.83 has a different interpretation in “Birds fly to degree 0.83” from that in “Hotel Verdi is close to the train station to degree 0.83”
Uncertainty: statements are true or false. But, due to lack of knowledge we can only estimate to which probability/possibility/necessity degree they are true or false.

For instance, a bird flies or does not fly. The probability/possibility/necessity degree that it flies is 0.83.

Usually we have a possible world semantics with a distribution over possible worlds:

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

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

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

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

\[ Necc(\phi) = \inf_{I|\not\models \phi} \mu(I) = 1 - Poss(\neg \phi) \]
Vagueness

Vagueness: statements involve concepts for which there is no exact definition, such as tall, small, close, far, cheap, expensive, isAbout, similarTo. Statements are true to some degree which is taken from a truth space.

- E.g., “Hotel Verdi is close to the train station to degree 0.83”

Truth space: set of truth values $L$ and an partial order $\leq$

Many-valued Interpretation: a function $I$ mapping formulae into $L$, i.e. $I(\phi) \in L$

Fuzzy Logic: $L = [0, 1]$

Uncertainty and Vagueness: “It is possible/probable to degree 0.83 that it will be hot tomorrow”

The notion of imperfect information covers concepts such as uncertainty, vagueness, contradiction, incompleteness, imprecision.
C. Meghini, F. Sebastiani, and U. Straccia.
A model of multimedia information retrieval.

Vague knowledge bases for matchmaking in p2p e-marketplaces.

U. Straccia.
Answering vague queries in fuzzy dl-lite.

U. Straccia.
Towards top-k query answering in deductive databases.

U. Straccia.

U. Straccia.
Towards vague query answering in logic programming for logic-based information retrieval.

U. Straccia and R. Troncy.
Towards distributed information retrieval in the semantic web.

U. Straccia and G. Visco.

DLMedia: an ontology mediated multimedia information retrieval system.
In *Proceedings of the International Workshop on Description Logics (DL-07)*, Innsbruck, Austria, 2007. CEUR.
Uncertainty, Vagueness, and the Semantic Web

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

Uncertainty in Semantic Web Languages
- Uncertainty
- Uncertainty and RDF/DLs/OWL
- Uncertainty and LPs/DLPs

Vagueness in Semantic Web Languages
- Vagueness basics
- Vagueness and RDF/DLs
- Vagueness and LPs/DLPs

Combining Uncertainty and Vagueness in the Semantic Web

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

Uncertainty and RDF/DLs/OWL

Uncertainty and LPs/DLPs

Vagueness basics

Vagueness and RDF/DLs

Vagueness and LPs/DLPs

Tutorial at SWAP-2007

U. Straccia
Web Ontology Languages

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

- Statements are of the form

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

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>`
OWL [10]

Three species of OWL
- **OWL full** is union of OWL syntax and RDF (Undecidable)
- **OWL DL** restricted to FOL fragment (decidable in NEXPTIME)
- **OWL Lite** is “easier to implement” subset of OWL DL (decidable in EXPTIME)

Semantic layering
- OWL DL within **Description Logic (DL) fragment**
- OWL DL based on $SHOIN(D_n)$ DL
- OWL Lite based on $SHIF(D_n)$ 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 ∃ and ∀
  - 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 \cup D )</td>
<td>( C(x) \lor D(x) )</td>
<td>( \text{Human} \cup \text{Male} )</td>
</tr>
<tr>
<td>( C \cap D )</td>
<td>( C(x) \land D(x) )</td>
<td>( \text{Human} \cap \text{Male} )</td>
</tr>
<tr>
<td>( \neg C )</td>
<td>( \neg C(x) )</td>
<td>( \neg \text{Meat} )</td>
</tr>
<tr>
<td>( \exists R.C )</td>
<td>( \exists y. R(x, y) \land C(y) )</td>
<td>( \exists \text{has_child}. \text{Blond} )</td>
</tr>
<tr>
<td>( \forall R.C )</td>
<td>( \forall y. R(x, y) \Rightarrow C(y) )</td>
<td>( \forall \text{has_child}. \text{Human} )</td>
</tr>
<tr>
<td>( C \sqsubseteq D )</td>
<td>( \forall x. C(x) \Rightarrow D(x) )</td>
<td>( \text{Happy_Father} \sqsubseteq \text{Man} \sqcap \exists \text{has_child}. \text{Female} )</td>
</tr>
<tr>
<td>( a : C )</td>
<td>( C(a) )</td>
<td>( \text{John:Happy_Father} )</td>
</tr>
</tbody>
</table>
Toy Example

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

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

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

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

- **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. \( \text{is}_\text{component}_\text{of} \sqsubseteq \text{is}_\text{part}_\text{of} \)
- **N**: Number restrictions, \( (\geq n R) \) and \( (\leq n R) \), e.g. \( (\geq 3 \text{has}_\text{Child}) \) (has at least 3 children)
- **Q**: Qualified number restrictions, \( (\geq n R.C) \) and \( (\leq n R.C) \), e.g. \( (\leq 2 \text{has}_\text{Child}.\text{Adult}) \) (has at most 2 adult children)
- **O**: Nominals (singleton class), \( \{a\} \), e.g. \( \exists \text{has}_\text{child}.\{\text{mary}\} \).
  \underline{Note}: \( a:C \equiv \{a\} \sqsubseteq C \) and \( (a, b):R \equiv \{a\} \sqsubseteq \exists R.\{b\} \)
- **I**: Inverse role, \( R^\rightarrow \), e.g. \( \text{isPartOf} = \text{hasPart}^\rightarrow \)
- **F**: Functional role, \( f \), e.g. \( \text{functional}(\text{hasAge}) \)
- **R_+**: transitive role, e.g. \( \text{transitive(isPartOf)} \)

For instance,

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

- **Concrete domains**: reals, integers, strings, ...  
  
  \[(tim, 14) : hasAge\]  
  \[(sf, "SoftComputing") : hasAcronym\]  
  \[(source1, "ComputerScience") : isAbout\]  
  \[(service2, "InformationRetrievalTool") : Matches\]  
  \[Minor = Person \cap \exists hasAge. \leq 18\]

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

- **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,

  \[ has\_father(x, y) \leftarrow has\_parent(x, y) \land Male(y) \]

- **Facts** are rules with empty body
  For instance,

  \[ has\_parent(mary, jo) \]
Toy Example

\[
\begin{align*}
Q(x) & \leftarrow B(x) \\
Q(x) & \leftarrow C(x) \\
B(a) & \leftarrow \\
C(b) & \leftarrow
\end{align*}
\]

\[
KB \models Q(a) \quad KB \models Q(b)
\]

\[
\text{answers}(KB, Q) = \{ a, b \}
\]

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

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

  \[
  \begin{align*}
  \text{buy}(x) & \leftarrow \text{Electronics}(x), \text{offer}(x) \\
  \text{Camera} & \sqsubseteq \text{Electronics}
  \end{align*}
  \]

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

- Many different approaches exist with different semantics
C. V. Damásio, J. Z. Pan, G. Stoilos, and U. Straccia.
An approach to representing uncertainty rules in ruleml.

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

T. Eiter, T. Lukasiewicz, R. Schindlauer, and H. Tompits.
Combining answer set programming with description logics for the Semantic Web.

T. Eiter, T. Lukasiewicz, R. Schindlauer, and H. Tompits.
Well-founded semantics for description logic programs in the Semantic Web.

T. Eiter, G. Ianni, R. Schindlauer, and H. Tompits.
Effective integration of declarative rules with external evaluations for Semantic Web reasoning.

J. W. Lloyd.
Foundations of Logic Programming.

T. Lukasiewicz.

Vague knowledge bases for matchmaking in p2p e-marketplaces.

U. Straccia.

Towards top-k query answering in deductive databases.

W3C.

Probabilistic Logic

- Integration of (propositional) logic- and probability-based representation and reasoning formalisms.

- Reasoning from logical constraints and interval restrictions for conditional probabilities (also called *conditional constraints*).

- Reasoning from convex sets of probability distributions.

- Model-theoretic notion of logical entailment.
Syntax of Probabilistic Knowledge Bases

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

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

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

- $P = \{(have\_legs \mid bird)[1, 1], (fly \mid bird)[0.95, 1]\}$:
  “All birds have legs”.
  “Birds fly with a probability of at least 0.95”.
Semantics of Probabilistic Knowledge Bases

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

- Set of basic propositions $\Phi = \{\text{bird, fly}\}$.
- $I_\Phi$ contains exactly the worlds $l_1$, $l_2$, $l_3$, and $l_4$ over $\Phi$:

<table>
<thead>
<tr>
<th></th>
<th>fly</th>
<th>$\neg$fly</th>
</tr>
</thead>
<tbody>
<tr>
<td>bird</td>
<td>$l_1$</td>
<td>$l_2$</td>
</tr>
<tr>
<td>$\neg$bird</td>
<td>$l_3$</td>
<td>$l_4$</td>
</tr>
</tbody>
</table>

- Some probabilistic interpretations:

<table>
<thead>
<tr>
<th></th>
<th>fly</th>
<th>$\neg$fly</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Pr_1$</td>
<td>bird</td>
<td>19/40</td>
</tr>
<tr>
<td>$Pr_1$</td>
<td>$\neg$bird</td>
<td>10/40</td>
</tr>
<tr>
<td>$Pr_2$</td>
<td>bird</td>
<td>0</td>
</tr>
<tr>
<td>$Pr_2$</td>
<td>$\neg$bird</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 $Pr_2(bird) = 1/3$.
- $\neg$fly $\iff$ 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.
- \( KB \models (\psi | \phi)[l, u] \): \( (\psi | \phi)[l, u] \) is a logical consequence of \( KB \) iff every model of \( KB \) is also a model of \( (\psi | \phi)[l, u] \).
- \( KB \models \text{tight} (\psi | \phi)[l, u] \): \( (\psi | \phi)[l, u] \) is a tight logical consequence of \( KB \) iff \( l \) (resp., \( u \)) is the infimum (resp., supremum) of \( Pr(\psi | \phi) \) subject to all models \( Pr \) of \( KB \) with \( Pr(\phi) > 0 \).
Example

- Probabilistic knowledge base:
  \[ KB = (\{bird \Leftarrow eagle\}, \{(have\_legs \mid bird)[1, 1], (fly \mid bird)[0.95, 1]\}). \]

- \( KB \) is satisfiable, since
  \[ Pr \text{ with } Pr(bird \land eagle \land have\_legs \land fly) = 1 \text{ is a model.} \]

- Some conclusions under logical entailment:
  \[ KB \models (have\_legs \mid bird)[0.3, 1], \quad KB \models (fly \mid bird)[0.6, 1]. \]

- Tight conclusions under logical entailment:
  \[ KB \models_{\text{tight}} (have\_legs \mid bird)[1, 1], \quad KB \models_{\text{tight}} (fly \mid bird)[0.95, 1], \]
  \[ KB \models_{\text{tight}} (have\_legs \mid eagle)[1, 1], \quad KB \models_{\text{tight}} (fly \mid eagle)[0, 1]. \]
Literature

- G. Boole. *An Investigation of the Laws of Thought, on which are Founded the Mathematical Theories of Logic and Probabilities*. Walton and Maberley, London, 1854.
Probabilistic Ontologies

Main types of encoded probabilistic knowledge:

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

Main types of reasoning problems:

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

- Representation of **terminological and assertional probabilistic knowledge** (e.g., in the medical domain or at the stock exchange market).

- **Information retrieval**, for an increased recall (e.g., Udrea et al.: Probabilistic ontologies and relational databases. In *Proc. CoopIS/DOA/ODBASE-2005*).

- **Ontology matching** (e.g., Mitra et al.: OMEN: A probabilistic ontology mapping tool. In *Proc. ISWC-2005*).

- **Probabilistic data integration**, especially for handling ambiguous and controversial pieces of information.
Probabilistic RDF


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


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

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

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

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

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


Other Works


Probabilistic Logic Programs

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

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

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


(3) Relational Bayesian networks:


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

Probabilistic Description Logic Programs


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

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

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

Product \sqsubseteq \geq 1 \text{ related};
\geq 1 \text{ related} \sqcup \geq 1 \text{ related}^\perp \sqsubseteq Product;

Textbook(t_{b\ ai}); \quad Textbook(t_{b\ lp});
PC(p_{c\ ibm}); \quad PC(p_{c\ hp});

related(t_{b\ ai}, t_{b\ lp}); \quad related(p_{c\ ibm}, p_{c\ hp});
provides(i_{bm}, p_{c\ ibm}); \quad provides(h_{p}, p_{c\ hp}).
Classical dl-rules in $P$
of a probabilistic dl-program $KB = (L, P, C, \mu)$:

- $pc(pc_1)$; $pc(pc_2)$; $pc(pc_3)$;
- $brand\_new(pc_1)$; $brand\_new(pc_2)$;
- $vendor(dell, pc_1)$; $vendor(dell, pc_2)$; $vendor(dell, pc_3)$;
- $provider(P) \leftarrow vendor(P, X), DL[PC \uplus pc; Product](X)$;
- $provider(P) \leftarrow DL[provides](P, X), DL[PC \uplus pc; Product](X)$;
- $similar(X, Y) \leftarrow DL[related](X, Y)$;
- $similar(X, Z) \leftarrow similar(X, Y), similar(Y, Z)$. 
Probabilistic dl-rules in $P$ along with the probability $\mu$ on the choice space $C$ of a probabilistic dl-program $KB = (L, P, C, \mu)$:

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

$\mu$: $\text{avoid}_\text{pos}, \text{avoid}_\text{neg} \mapsto 0.9, 0.1; \text{offer}_\text{pos}, \text{offer}_\text{neg} \mapsto 0.9, 0.1; \text{v\_buy\_pos}, \text{v\_buy\_neg} \mapsto 0.7, 0.3; \text{a\_buy\_pos}, \text{a\_buy\_neg} \mapsto 0.7, 0.3.$

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

Probabilistic query: $\exists (\text{buy}(c, x) | \text{needs}(c, x) \land \text{buy}(c, y) \land \text{also\_buy}(y, x) \land \text{view}(x) \land \text{not avoid}(x)) [L, U]$
Example: Probabilistic Data Integration

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

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

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

\[
P_M = \{\text{forecast}_\text{rome}(D, W, T, M) \leftarrow \text{forecast}(\text{rome}, D, W, T, M), \text{inst}_A; \\
\text{forecast}_\text{rome}(D, W, T, M) \leftarrow \text{forecast}_\text{Rome}(D, W, T, M), \text{inst}_B; \\
\text{forecast}_\text{rome}(D, W, T, M) \leftarrow \text{forecast}_\text{weather}(\text{rome}, D, W), \\
\text{forecast}_\text{temperature}(\text{rome}, D, T), \\
\text{forecast}_\text{wind}(\text{rome}, D, M), \text{inst}_C\} ;
\]

\[
C_M = \{\{\text{inst}_A, \text{inst}_B, \text{inst}_C\}\} ;
\]

\[
\mu_M : \text{inst}_A, \text{inst}_B, \text{inst}_C \Rightarrow 0.6, 0.3, 0.1 .
\]
Example (Tightly Coupled): Ontology Mapping

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

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

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

$$P_M = \{ logic\_programming(X) \leftarrow rule\_based\_systems(X), choice_1 ;
\quad logic\_programming(X) \leftarrow deductive\_databases(X), choice_2 \} ;$$

$$C_M = \{ \{choice_1, not\_choice_1\}, \{choice_2, not\_choice_2\} \} ;$$

$$\mu_M : choice_1, not\_choice_1, choice_2, not\_choice_2 \mapsto 0.7, 0.3, 0.8, 0.2 .$$
Vagueness

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

\[
q(h) \leftarrow \text{hasLocation}(h, hl) \land \text{hasLocation}(\text{train}, cl) \land \text{close}(hl, cl) \land \text{cheap}(h)
\]

- **Truth space**: usually [0, 1]
- **Interpretation**: a function \( I \) mapping atoms into [0, 1], i.e. \( I(A) \in [0, 1] \)
- **Problem**: what is the interpretation of e.g. \( \text{close}(\text{verdi}, \text{train}) \land \text{cheap}(200) \)?
  - E.g., if \( I(\text{close}(\text{verdi}, \text{train})) = 0.83 \) and \( I(\text{cheap}(200)) = 0.2 \), 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”
Propositional Fuzzy Logics Basics [5]

- **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 $\land, \lor, \neg, \rightarrow$
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>¬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 ( y/x )</td>
<td>( \max(1 - x, y) )</td>
</tr>
</tbody>
</table>

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

\[
\begin{align*}
\mathcal{I}(\phi \land \psi) & = \mathcal{I}(\phi) \land \mathcal{I}(\psi) \\
\mathcal{I}(\phi \lor \psi) & = \mathcal{I}(\phi) \lor \mathcal{I}(\psi) \\
\mathcal{I}(\phi \rightarrow \psi) & = \mathcal{I}(\phi) \rightarrow \mathcal{I}(\psi) \\
\mathcal{I} \models \phi & \iff \mathcal{I}(\phi) = 1 \iff \phi\text{ satisfiable} \\
\mathcal{I} \models \mathcal{T} & \iff \mathcal{I} \models \phi \text{ for all } \phi \in \mathcal{T} \\
\models \phi & \iff \text{ for all } \mathcal{I}. \mathcal{I} \models \phi \\
\mathcal{T} \models \phi & \iff \text{ for all } \mathcal{I}. \text{ if } \mathcal{I} \models \mathcal{T} \text{ then } \mathcal{I} \models \phi
\end{align*}
\]
Note:

\[ \neg \phi \quad \text{is} \quad \phi \rightarrow 0 \]

\[ \phi \bar{\land} \psi \quad \text{defined as} \quad \phi \land (\phi \rightarrow \psi) \]

\[ \phi \bar{\lor} \psi \quad \text{defined as} \quad ((\phi \rightarrow \psi) \rightarrow \psi) \bar{\land}((\psi \rightarrow \phi) \rightarrow \phi) \]

\[ I(\phi \bar{\land} \psi) = \min(I(\phi), I(\psi)) \]

\[ I(\phi \bar{\lor} \psi) = \max(I(\phi), I(\psi)) \]

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

\[ \neg Z \phi = \neg L \phi \]

\[ \phi \land Z \psi = \phi \land L (\phi \rightarrow L \psi) \]

\[ \phi \rightarrow Z \psi = \neg L \phi \lor L \psi \]
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>
</tbody>
</table>
Predicate Fuzzy Logics Basics [5]

- **Formulae**: First-Order Logic formulae, terms are either variables or constants
  - we may introduce functions symbols as well, with crisp semantics (but uninteresting), or we need to discuss also fuzzy equality (which we leave out here)
- **Truth space** is $[0, 1]$
- **Formulae** have a a degree of truth in $[0, 1]$
- **Interpretation**: is a mapping $I : Atoms \rightarrow [0, 1]$
- Interpretations are extended to formulae as follows:
  
  $I(\neg \phi) = I(\phi) \rightarrow 0$
  
  $I(\phi \land \psi) = I(\phi) \land I(\psi)$
  
  $I(\phi \rightarrow \psi) = I(\phi) \rightarrow I(\psi)$
  
  $I(\exists x \phi) = \sup_{c \in \Delta I} I^c_x(\phi)$
  
  $I(\forall x \phi) = \inf_{c \in \Delta I} I^c_x(\phi)$

  where $I^c_x$ is as $I$, except that variable $x$ is mapped into individual $c$

- **Definitions of** $I : \models \phi$, $I : \models \vartheta$, $I : \models \phi$, $\models \phi : I$, $\models \phi : I$ and $\models \phi : I$ are as for the propositional case
Fuzzy RDF (we generalize [15, 16, 34])

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

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

- Meaning: the degree of truth of the statement is at least $n$

- For instance,

  $$\langle (o1, \text{IsAbout}, \text{snoopy}), 0.8 \rangle$$
Inferences in Fuzzy RDFS

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

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

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

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

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

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

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

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

- Fuzzy RDF representation

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

- then

\[
KB \models \langle \exists x. (o_1, IsAbout, x) \land (x, type, Animal), 0.8 \rangle
\]
Fuzzy DLs Basics [26]

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

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C, D \rightarrow$</td>
<td>$\top \rightarrow \top^I(x) = 1$</td>
</tr>
<tr>
<td>$\bot \rightarrow$</td>
<td>$\bot^I(x) = 0$</td>
</tr>
<tr>
<td>$A \rightarrow$</td>
<td>$A^I(x) \in [0, 1]$</td>
</tr>
</tbody>
</table>

Concepts:

- $C \cap D \rightarrow (C_1 \cap C_2)^I(x) = C_1^I(x) \wedge C_2^I(x)$
- $C \sqcup D \rightarrow (C_1 \sqcup C_2)^I(x) = C_1^I(x) \vee C_2^I(x)$
- $\neg C \rightarrow (\neg C)^I(x) = \neg C^I(x)$
- $\exists R.C \rightarrow (\exists R.C)^I(x) = \sup_{y \in \Delta^I} R^I(x, y) \wedge C^I(y)$
- $\forall R.C \rightarrow (\forall R.C)^I(u) = \inf_{y \in \Delta^I} R^I(x, y) \rightarrow C^I(y)$

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

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

Inclusion axioms: $C \sqsubseteq D$,

- $I \models C \sqsubseteq D$ iff $\forall x \in \Delta^I.C^I(x) \leq D^I(x)$
- This is equivalent to, $\forall x \in \Delta^I.(C^I(x) \rightarrow D^I(x)) = 1$, if $\rightarrow$ is an r-implication
Basic Inference Problems

Consistency: Check if knowledge is meaningful
- Is \( \text{KB} \) consistent, i.e. satisfiable?

Subsumption: Structure knowledge, compute taxonomy
- \( \text{KB} \models C \sqsubseteq D \) ?

Equivalence: Check if two fuzzy concepts are the same
- \( \text{KB} \models C = D \) ?

Graded instantiation: Check if individual \( a \) instance of class \( C \) to degree at least \( r \)
- \( \text{KB} \models \langle a:C, r \rangle \) ?

BTVB: Best Truth Value Bound problem
- \( |a:C|_{KB} = \sup \{ r \mid \text{KB} \models \langle a:C, r \rangle \} \) ?

Top-k retrieval: Retrieve the top-k individuals that instantiate \( C \) w.r.t. best truth value bound
- \( \text{ans}_{top-k}(\text{KB}, C) = \text{Top}_k \{ \langle a, v \rangle \mid v = |a:C|_{KB} \} \)
Towards fuzzy OWL Lite and OWL DL

- Recall that OWL Lite and OWL DL relate to $SHIF(D)$ and $SHOIN(D)$, respectively.
- We need to extend the semantics of fuzzy $ALC$ to fuzzy $SHOIN(D) = ALCCHOIN_{R+}(D)$.
- 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 \( d_D^n : \Delta^n_D \rightarrow [0, 1] \)
  - For instance,

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

- Very, moreOrLess, slightly, etc.

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

- For instance,

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

#### Concepts:

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C, D$</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>$\exists y_1, \ldots, y_n. \land_{i=1}^n R(x, y_i) \land \bigwedge_{1 \leq i &lt; j \leq n} y_i \neq y_j$</td>
</tr>
<tr>
<td>$(\leq n R)$</td>
<td>$\forall y_1, \ldots, y_{n+1}. \land_{i=1}^{n+1} R(x, y_i) \rightarrow \bigvee_{1 \leq i &lt; j \leq n+1} y_i = y_j$</td>
</tr>
<tr>
<td>$FCC$</td>
<td>$\mu_{FCC}(x)$</td>
</tr>
<tr>
<td>$M(C)$</td>
<td>$\mu_M(C(x))$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$R$</th>
<th>$P$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P$</td>
<td>$P(x, y)$</td>
</tr>
</tbody>
</table>

#### Assertions:

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

#### Axioms:

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tau$</td>
<td>$\forall x \ (C(x) \rightarrow D(x)) \geq r,$</td>
</tr>
<tr>
<td>$fun(R)$</td>
<td>$\forall x \forall y \forall z \ R(x, y) \land R(x, z) \rightarrow y = z$</td>
</tr>
<tr>
<td>$trans(R)$</td>
<td>$(\exists z \ R(x, z) \land R(z, y)) \rightarrow R(x, y)$</td>
</tr>
</tbody>
</table>
Example (Graded Entailment)

<table>
<thead>
<tr>
<th>Car</th>
<th>speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>audi_tt</td>
<td>243</td>
</tr>
<tr>
<td>mg</td>
<td>≤ 170</td>
</tr>
<tr>
<td>ferrari_enzo</td>
<td>≥ 350</td>
</tr>
</tbody>
</table>

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

\[ KB \models \langle \text{ferrari\_enzo:SportsCar}, 1 \rangle \]
\[ KB \models \langle \text{audi\_tt:SportsCar}, 0.92 \rangle \]
\[ KB \models \langle \text{mg:\neg SportsCar}, 0.72 \rangle \]
Example (Graded Subsumption)

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

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

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 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 €.

\[
\text{AudiTT} = \text{SportsCar} \sqcap \exists \text{hasPrice}.rs(30500, 31500) \\
\text{Query} = \text{SportsCar} \sqcap \exists \text{hasPrice}.ls(30000, 32000)
\]

The 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 €.
Modifiers are definable as linear in-equations over $\mathbb{Q}, \mathbb{Z}$ (e.g., linear hedges), for instance, linear hedges, $lm(a, b)$, e.g. very $= lm(0.7, 0.49)$

Fuzzy concrete concepts are definable as linear in-equations over $\mathbb{Q}, \mathbb{Z}$ (e.g., crisp, triangular, trapezoidal, left shoulder and right shoulder membership functions)
Implementation issues

Several options exist:

- Try to map fuzzy DLs to classical DLs
  - difficult to work with modifiers and concrete fuzzy concepts
- Try to map fuzzy DLs to some fuzzy logic programming framework
  - A lot of work exists about mappings among classical DLs and LPs
  - But, needs a theorem prover for fuzzy LPs
- Build an ad-hoc theorem prover for fuzzy DLs, using e.g., MILP

A theorem prover for fuzzy $SHIF$ + linear hedges + concrete fuzzy concepts + linear equational constraints + datatypes, under classical, Zadeh, Lukasiewicz and Product t-norm semantics has been implemented
(http://gaia.isti.cnr.it/~straccia)

FIRE: a fuzzy DL theorem prover for fuzzy $SHIN$ under Zadeh semantics
(http://www.image.ece.ntua.gr/~nsimou/)
Top-\(k\) retrieval in tractable DLs: the case of DL-Lite/DLR-Lite [25, 30]

- **DL-Lite/DLR-Lite** [3]: a simple, but interesting DLs
- Captures important subset of UML/ER diagrams
- Computationally tractable DL to query large databases
- **Sub-linear**, i.e. LOGSpace in data complexity
  - (same cost as for SQL)
- Good for **very large** database tables, with limited declarative schema design
Knowledge base: $KB = \langle T, A \rangle$, where $T$ and $A$ are finite sets of axioms and assertions

Axiom: $Cl \sqsubseteq Cr$ (inclusion axiom)

Note for inclusion axioms: the language for left hand side is different from the one for right hand side

DL-Lite$_{core}$:
- Concepts:
  - $Cl \rightarrow A | \exists R$
  - $Cr \rightarrow A | \exists R | \neg A | \neg \exists R$
  - $R \rightarrow P | P^\neg$
- Assertion: $a:A, (a, b):P$

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

Tutorial at SWAP-2007 U. Straccia
Examples:

- **isa**
  
  \[ \text{CatalogueBook} \sqsubseteq \text{Book} \]

- **disjointness**
  
  \[ \text{Book} \sqsubseteq \neg \text{Author} \]

- **constraints**
  
  \[ \text{CatalogueBook} \sqsubseteq \exists \text{positioned}_\text{In} \]

- **role – typing**
  
  \[ \exists \text{positioned}_\text{In} \sqsubseteq \text{Container} \]

- **functional**
  
  \[ \text{fun(positioned}_\text{In}) \]

- **constraints**
  
  \[ \text{Author} \sqsubseteq \exists \text{written}_\text{By} \]

- **assertion**
  
  \[ \text{Romeo_and_Juliet:CatalogueBook} \]
  \[ (\text{Romeo_and_Juliet, Shakespeare}):\text{written}_\text{By} \]

- **query**
  
  \[ q(x, y) \leftarrow \text{CataloguedBook}(x), \text{Ordered}_\text{to}(x, y) \]

- **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/DR-Lite

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

**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/DR-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:

```
Hotel ⊑ ∃HasHLoc
Hotel ⊑ ∃HasHPrice
Conference ⊑ ∃HasCLoc
Hotel ⊑ ¬Conference
```

<table>
<thead>
<tr>
<th>HasHLoc</th>
<th>HasCLoc</th>
<th>HasHPrice</th>
</tr>
</thead>
<tbody>
<tr>
<td>HotelID</td>
<td>HasLoc</td>
<td>HotelID</td>
</tr>
<tr>
<td>h1</td>
<td>h1</td>
<td>h1</td>
</tr>
<tr>
<td>h2</td>
<td>h2</td>
<td>h2</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ConflID</th>
<th>HasLoc</th>
<th>ConflID</th>
</tr>
</thead>
<tbody>
<tr>
<td>c1</td>
<td>c1</td>
<td>c1</td>
</tr>
<tr>
<td>c2</td>
<td>c2</td>
<td>c2</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Tool exists and implemented in the DLMedia system

http://gaia.isti.cnr.it/~straccia
DLMedia: a Multimedia Information Retrieval System [33]

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

  $Rr \rightarrow A \mid \exists[i_1, \ldots, i_k]R$

  $Rl \rightarrow A \mid \exists[i_1, \ldots, i_k]R \mid \exists[i_1, \ldots, i_k]R. (Cond_1 \sqcap \ldots \sqcap Cond_l)$

  $Cond \rightarrow ([i] \leq v) \mid ([i] < v) \mid ([i] \geq v) \mid ([i] > v) \mid ([i] = v) \mid ([i] \neq v) \mid ([i] simTxt' k_1, \ldots, k_n') \mid ([i] simImg URN)$

  $\exists[i_1, \ldots, i_k]R$ is the projection of the relation $R$ on the columns $i_1, \ldots, i_k$

  $\exists[i_1, \ldots, i_k]R. (Cond_1 \sqcap \ldots \sqcap Cond_l)$ further restricts the projection $\exists[i_1, \ldots, i_k]R$ according to the conditions specified in $Cond_l$

  $([i] simTxt' k_1 \ldots k_n')$ evaluates the degree of being the text of the $i$-th column similar to the list of keywords $k_1 \ldots k_n$

  $([i] simImg URN)$ returns the system’s degree of being the image identified by the $i$-th column similar to the image identified by the URN

  Facts: $\langle R(c_1, \ldots, c_n), s \rangle$
Example axioms

\( \exists [1, 2] Person \sqsubseteq \exists [1, 2] hasAge \)
// constrains relation \( hasAge(name, age) \)

\( \exists [3, 1] Person \sqsubseteq \exists [1, 2] hasChild \)
// constrains relation \( hasChild(father\_name, name) \)

\( \exists [4, 1] Person \sqsubseteq \exists [1, 2] hasChild \)
// constrains relation \( hasChild(mother\_name, name) \)

\( \exists [3, 1] Person.([2] \geq 18) \sqcap ([5] = 'female') \sqsubseteq \exists [1, 2] hasAdultDaughter \)
// constrains relation \( hasAdultDaughter(father\_name, name) \)

On the other hand examples axioms involving similarity predicates are,

\( \exists [1] ImageDescr.([2] simImg \ urn1) \sqsubseteq Child \) \hspace{1cm} (1)

\( \exists [1] Title.([2] simTxt 'lion') \sqsubseteq Lion \) \hspace{1cm} (2)

where \( urn1 \) identifies the image
Example queries

$q(x) \leftarrow Child(x)$

// find objects about a child (strictly speaking, find instances of $Child$)

$q(x) \leftarrow CreatorName(x, y) \land (y = 'paolo'), Title(x, z), (z \text{ simTxt} 'tour')$

// find images made by Paolo whose title is about 'tour'

$q(x) \leftarrow ImageDescr(x, y) \land (y \text{ simImg urn2})$

// find images similar to a given image identified by $urn2$

$q(x) \leftarrow ImageObject(x) \land isAbout(x, y_1) \land Car(y_1) \land isAbout(x, y_2) \land Racing(y_2)$

// find image objects about cars racing
Many Logic Programming (LP) frameworks have been proposed to manage uncertain and imprecise information. They differ in:

- The underlying notion of uncertainty and vagueness: probability, possibility, many-valued, fuzzy logics
- How values, associated to rules and facts, are managed

We consider fuzzy LPs, where

- **Truth space** is \([0, 1]\)
- **Interpretation** is a mapping \(I : B_P \rightarrow [0, 1]\)
- **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 \(R(x)\)”
Same meaning as for fuzzy DLR-Lite queries

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

1. \(x\) are the \textit{distinguished variables};
2. \(s\) is the \textit{score variable}, taking values in \([0, 1]\);
3. \(y\) are existentially quantified variables, called \textit{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 \textit{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 \textit{scoring function} \(f: [0, 1]^{l+h} \rightarrow [0, 1]\), which combines the scores of the \(n\) fuzzy predicates \(p_i(c_i)\).
Example: Soft shopping agent

I may represent my preferences in Logic Programming with the rules

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

<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 [29, 32]
Fuzzy DLPs Basics [10, 11, 27, 31]

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

\[
\text{LowCarPrice}(z) \leftarrow \min(\text{made}_by(x, y), \text{DL[ChineseCarCompany]}(y), \text{price}(x, z)) \cdot \text{DL[Low]}(z)
\]

\[
\text{Low} = \text{LS}(5.000, 15.000)
\]

\[
\text{ChineseCarCompany} \sqsubseteq \exists \text{has\_location}\text{.China}
\]

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

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

1. **Axiomatic** approach: fuzzy DL atoms and roles are managed uniformly
2. **Loosely Coupled** approach: fuzzy DL atoms and roles are like “procedural attachments” (procedural calls to a fuzzy DL theorem prover)
3. **Tightly coupled** approach: The DL component restricts the models to be considered for the LP component
F. Bobillo and U. Straccia.
A fuzzy description logic with product t-norm.

F. Bobillo and U. Straccia.
Mixed integer programming, general concept inclusions and fuzzy description logics.

Data complexity of query answering in description logics.

C. Damásio and M. Medina, J. Ojeda-Aciego.
A tabulation procedure for first-order residuated logic programs.

P. Hájek.
*Metamathematics of Fuzzy Logic*.

M. Kifer and V. Subrahmanian.
Theory of generalized annotated logic programming and its applications.

L. V. Lakshmanan and N. Shiri.

---

Uncertainty, Vagueness, and the Semantic Web
Basics on Semantic Web Languages
Uncertainty in Semantic Web Languages
Vagueness in Semantic Web Languages
Combining Uncertainty and Vagueness in the Semantic Web
Vagueness basics
Vagueness and RDF/DLs
Vagueness and LPs/DLPs

F. Bobillo and U. Straccia.
A fuzzy description logic with product t-norm.

F. Bobillo and U. Straccia.
Mixed integer programming, general concept inclusions and fuzzy description logics.

Data complexity of query answering in description logics.

C. Damásio and M. Medina, J. Ojeda-Aciego.
A tabulation procedure for first-order residuated logic programs.

P. Hájek.
*Metamathematics of Fuzzy Logic*.

M. Kifer and V. Subrahmanian.
Theory of generalized annotated logic programming and its applications.

L. V. Lakshmanan and N. Shiri.

---

Uncertainty and Vagueness in the Semantic Web
Tutorial at SWAP-2007
U. Straccia
A parametric approach to deductive databases with uncertainty.


Y. Loyer and U. Straccia.
Any-world assumptions in logic programming.

T. Lukasiewicz.
Fuzzy description logic programs under the answer set semantics for the semantic web.

T. Lukasiewicz and U. Straccia.
Tightly integrated fuzzy description logic programs under the answer semantics for the semantic web.

T. Lukasiewicz and U. Straccia.
Tightly integrated fuzzy description logic programs under the answer semantics for the semantic web.

C. Mateis.
Extending disjunctive logic programming by t-norms.
<table>
<thead>
<tr>
<th>Author(s)</th>
<th>Title and Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>U. Straccia.</td>
<td>Description logics with fuzzy concrete domains.</td>
</tr>
</tbody>
</table>

**U. Straccia.**

Fuzzy alc with fuzzy concrete domains.

**U. Straccia.**

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

**U. Straccia.**

Uncertainty management in logic programming: Simple and effective top-down query answering.

**U. Straccia.**

Annotated answer set programming.

**U. Straccia.**

Answering vague queries in fuzzy dl-lite.
U. Straccia.

A fuzzy description logic for the semantic web.

U. Straccia.

Fuzzy description logic programs.

U. Straccia.

Query answering under the any-world assumption for normal logic programs.

U. Straccia.

Towards top-k query answering in deductive databases.

U. Straccia.


U. Straccia.

Uncertainty and description logic programs over lattices.
Combining Uncertainty and Vagueness in the Semantic Web

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

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

V. V., B. J., G. P., and H. T.
Fuzzy rdf in the semantic web: Deduction and induction.

P. Vojtáš.
Fuzzy logic programming.
Description logic programs that allow for dealing with probabilistic uncertainty and fuzzy vagueness.

Semantically, probabilistic uncertainty can be used for data integration and ontology mapping, and fuzzy vagueness can be used for expressing vague concepts.

Technically, allows for defining different rankings on ground atoms using fuzzy vagueness, and then for a probabilistic merging of these rankings using probabilistic uncertainty.

Query processing based on fixpoint iterations.
Suppose a person would like to buy “a sports car that costs at most about 22,000 euro and that has a power of around 150 HP”.

In today's Web, the buyer has to manually

- search for car selling web sites, e.g., using Google;
- select the most promising sites;
- browse through them, query them to see the cars that each site sells, and match the cars with the requirements;
- select the offers in each web site that match the requirements; and
- eventually merge all the best offers from each site and select the best ones.
Overview
Web Shopping Agent
Fuzzy Description Logics
Fuzzy Description Logic Programs
Adding Probabilistic Uncertainty
A shopping agent may support us, *automatizing* the whole process once it receives the request/query $q$ from the buyer:

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

Cars ⊑ (∃hasReview.Integer) □ (∃hasInvoice.Integer)
□ (∃hasResellValue.Integer) □ (∃hasMaxSpeed.Integer)
□ (∃hasHorsePower.Integer) □ ...

MazdaMX5Miata: SportyCar □ (∃hasInvoice.18883)
□ (∃hasHorsePower.166) □ ...
MitsubishiEclipseSpyder: SportyCar □ (∃hasInvoice.24029)
□ (∃hasHorsePower.162) □ ...
We may now encode “costs at most about 22,000 euro” and “has a power of around 150 HP” in the buyer’s request through the following concepts $C$ and $D$, respectively:

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

where $\text{LeqAbout22000} = \text{ls}(22000, 25000)$ and $\text{Around150HP} = \text{tri}(125, 150, 175)$. 

![Diagram of fuzzy sets]

![Diagram of another fuzzy set]
The following fuzzy dl-rule encodes the buyer’s request “a sports car that costs at most about 22,000 euro and that has a power of around 150 HP”.

\[
\text{query}(x) \leftarrow \bigotimes \text{SportyCar}(x) \land \bigotimes \\
\text{hasInvoice}(x, y_1) \land \bigotimes \\
\text{DL}[\text{LeqAbout22000}](y_1) \land \bigotimes \\
\text{hasHorsePower}(x, y_2) \land \bigotimes \\
\text{DL}[\text{Around150HP}](y_2) \geq 1. 
\]

Here, \( \bigotimes \) is the Gödel t-norm (that is, \( x \otimes y = \min(x, y) \)).
The buyer’s request, but in a “different” terminology:

\[
\text{query}(x) \leftarrow \otimes \text{SportsCar}(x) \land \otimes \text{hasPrice}(x, y_1) \land \otimes \text{hasPower}(x, y_2) \land \otimes \text{DL}[\text{LeqAbout22000}](y_1) \land \otimes \text{DL}[\text{Around150HP}](y_2) \geq 1
\]

Ontology alignment mapping rules:

\[
\text{SportsCar}(x) \leftarrow \otimes \text{DL}[\text{SportyCar}](x) \land \otimes \text{sc}_{pos} \geq 0.9
\]

\[
\text{hasPrice}(x) \leftarrow \otimes \text{DL}[\text{hasInvoice}](x) \land \otimes \text{hi}_{pos} \geq 0.8
\]

\[
\text{hasPower}(x) \leftarrow \otimes \text{DL}[\text{hasHorsePower}](x) \land \otimes \text{hhp}_{pos} \geq 0.8,
\]

Probability distribution \(\mu\):

\[
\begin{align*}
\mu(\text{sc}_{pos}) &= 0.91 & \mu(\text{sc}_{neg}) &= 0.09 \\
\mu(\text{hi}_{pos}) &= 0.78 & \mu(\text{hi}_{neg}) &= 0.22 \\
\mu(\text{hhp}_{pos}) &= 0.83 & \mu(\text{hhp}_{neg}) &= 0.17.
\end{align*}
\]
The following are some tight consequences:

\[ KB \models_{tight} (E[query((MazdaMX5Miata))]\{0.21, 0.21\} \]
\[ KB \models_{tight} (E[query((MitsubishiEclipseSpyder))]\{0.19, 0.19\} \].

Informally, the expected degree to which \textit{MazdaMX5Miata} matches the query \textit{q} is 0.21, while the expected degree to which \textit{MitsubishiEclipseSpyder} matches the query \textit{q} is 0.19,

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

<table>
<thead>
<tr>
<th>rank</th>
<th>item</th>
<th>degree</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>\textit{MazdaMX5Miata}</td>
<td>0.21</td>
</tr>
<tr>
<td>2.</td>
<td>\textit{MitsubishiEclipseSpyder}</td>
<td>0.19</td>
</tr>
</tbody>
</table>