Managing Uncertainty and Vagueness in Semantic Web Languages

Tutorial at ESWC-2007

Thomas Lukasiewicz\(^1\) \(^2\) Umberto Straccia\(^3\)

\(^1\) DIS, Sapienza Università di Roma, Italy
lukasiewicz@dis.uniroma1.it

\(^2\) Institut für Informationssysteme, TU Wien, Austria
lukasiewicz@kr.tuwien.ac.at

\(^3\) ISTI-CNR, Pisa, Italy
straccia@isti.cnr.it
Outline

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 SW Languages
Sources of Uncertainty and Vagueness on the Web

- **Resource discovery:**
  - 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 \( \mathcal{P}' \subseteq \mathcal{P} \), as it is not reasonable to assume to access to and query all resources (resource selection/resource discovery);

2. For every selected source \( S_i \in \mathcal{P}' \) the agent has to reformulate its information need \( Q_A \) into the query language \( \mathcal{L}_i \) provided by the resource (schema mapping/ontology alignment);

3. The results from the selected resources have to be merged together (data fusion/rank aggregation)
Example (Negotiation) [2]

- A car seller sells an Audi TT for 31500 €, as from the catalog price.
- A buyer is looking for a sports-car, but wants to 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]

“Find top-k image regions about animals”

\[ \text{Query}(x) \leftarrow \text{ImageRegion}(x) \land \text{isAbout}(x, y) \land \text{Animal}(y) \]
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>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>
</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>
</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)

E.g., \( \text{Temp} = 37.5, \text{Pulse} = 98, \text{RespiratoryRate} = 18 \) are in the “danger zone” already

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

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

**Uncertainty**: statements are **true** or **false**. But, due to lack of knowledge we can only estimate to which **probability/possibility/necessity** degree they are true or false

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

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

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

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

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

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

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

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

Vagueness

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

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

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

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

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

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

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.
Outline

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 SW Languages

Tutorial at ESWC-2007
T. Lukasiewicz and 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)
- Degree of formality varies widely
  - Increased formality makes languages more amenable to machine processing (e.g., automated reasoning)
- RDF and OWL-DL are the major players (so far ...)

Uncertainty and Vagueness in the Semantic Web

Tutorial at ESWC-2007
T. Lukasiewicz and U. Straccia
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:
  - a document, a picture, a paragraph on the Web;
  - a book in the library, a real person (?)
  - isbn://5031-4444-3333
  - ...

- Properties themselves are also resources (URIs)
RDF Schema (RDFS)

- RDF Schema allows you to define vocabulary terms and the relations between those terms.
- RDF Schema terms (just a few examples):
  - Class
  - Property
  - type
  - subClassOf
  - range
  - domain
- These terms are the RDF Schema building blocks (constructors) used to create vocabularies:
  
  ```
  <Person, type, Class>
  <hasColleague, type, Property>
  <Professor, subClassOf, Person>
  <Carole, type, Professor>
  <hasColleague, range, Person>
  <hasColleague, domain, Person>
  ```
RDF/RDFS Semantics

- RDF has “Non-standard” semantics in order to deal with this
- Semantics given by RDF Model Theory (MT)

In RDF MT, an interpretation $I$ of a vocabulary $V$ consists of:
- $IR$, a non-empty set of resources, called the domain of $I$.
- $IS$, a mapping from URI references in $V$ into $IR$
- $IP$, a distinguished subset of $IR$ (the set of properties of $I$)
  - A vocabulary element $v \in V$ is a property iff $IS(v) \in IP$
- $IEXT$, a mapping from $IP$ into the powerset of $IR \times IR$, $IEXT(x)$ is called the extension of $x$
  - I.e., a set of elements $\langle x, y \rangle$, with $x, y$ elements of $IR$
  - I.e., is a set of pairs which identify the arguments for which the property is true
  - This trick of distinguishing a relation as an object from its relational extension allows a property to occur in its own extension
- $IL$, a mapping from typed literals in $V$ into $IR$
  - A distinguished subset $LV$ of $IR$, called the set of literal values, which contains all the plain literals in $V$

Class interpretation $ICEXT$ simply induced by $IEXT(IS(type))$

$$ICEXT(C) = \{ x | \langle x, C \rangle \in IEXT(IS(type)) \}$$

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

- RDFS adds extra constraints on interpretations
  - E.g., interpretations of \( \langle C, \text{subClassOf}, D \rangle \) constrained to those where \( \text{ICEXT}(\text{IS}(C)) \subseteq \text{ICEXT}(\text{IS}(D)) \)

- Can deal with triples such as
  
  \[
  \langle \text{Species, type, Class} \rangle \\
  \langle \text{Lion, type, Species} \rangle \\
  \langle \text{Leo, type, Lion} \rangle \\
  \langle \text{SelfInst, type, SelfInst} \rangle 
  \]

- And even with triples such as
  
  \[
  \langle \text{type, subPropertyOf, subClassOf} \rangle 
  \]

- But not clear if meaning matches intuition (if there is one)
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 $L$anguage with $C$omplement)

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Semantics</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C \sqcup D$</td>
<td>$C(x) \lor D(x)$</td>
<td>$\text{Human} \sqcup \text{Male}$</td>
</tr>
<tr>
<td>$C \sqcap D$</td>
<td>$C(x) \land D(x)$</td>
<td>$\text{Nice} \sqcap \text{Rich}$</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

\[ \text{Sex} = \text{Male} \sqcup \text{Female} \]

\[ \text{Male} \sqcap \text{Female} \sqsubseteq \bot \]

\[ \text{Person} \sqsubseteq \text{Human} \sqcap \exists \text{hasSex}.\text{Sex} \]

\[ \text{MalePerson} \sqsubseteq \text{Person} \sqcap \exists \text{hasSex}.\text{Male} \]

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

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

\( \mathcal{AL} \): \( C, D \rightarrow \top, \bot, A, C \sqcap D, \neg A, \exists R.T, \forall R.C \)

- **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**: \( a:C \equiv \{a\} \sqsubseteq C \) and \( (a, b):R \equiv \{a\} \sqsubseteq \exists R.\{b\} \)

- **I**: Inverse role, \( R^- \), e.g. \( isPartOf = hasPart^- \)
- **F**: Functional role, \( f \), e.g. \( functional(hasAge) \)
- **R_+**: transitive role, e.g. \( transitive(isPartOf) \)

For instance,

\[
\begin{align*}
SHIF &= S + H + I + F = \mathcal{ALCR_+HIF} & \text{OWL-Lite (EXPTIME)} \\
SHOIN &= S + H + O + I + N = \mathcal{ALCR_+HOIN} & \text{OWL-DL (NEXPTIME)}
\end{align*}
\]
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,
\[
(\geq n \, R.C)^\mathcal{I} = \{ x \in |\{ y \mid \langle x, y \rangle \in R^\mathcal{I} \land y \in C^\mathcal{I} \}| \geq n \}, \\
(\leq n \, R.C)^\mathcal{I} = \{ x \in \Delta^\mathcal{I} : |\{ y \mid \langle x, y \rangle \in R^\mathcal{I} \land y \in C^\mathcal{I} \}| \leq n \}
\]

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

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

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

\[(\text{tim, 14}): \text{hasAge}\]
\[(\text{sf, “SoftComputing”}): \text{hasAcronym}\]
\[(\text{source1, “ComputerScience”}): \text{isAbout}\]
\[(\text{service2, “InformationRetrievalTool”}): \text{Matches}\]

\[\text{Minor} = \text{Person} \cap \exists \text{hasAge} \leq 18\]

- 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 \(d^D \subseteq \Delta^n_D\)
  - Concrete properties: \(R^I \subseteq \Delta^I \times \Delta_D\)
  - Notation: \((D)\). E.g., \(\mathcal{ALC}(D)\) is \(\mathcal{ALC}\) + concrete domains
### Concept/Class constructors:

<table>
<thead>
<tr>
<th>Descriptions (C)</th>
<th>Abstract Syntax</th>
<th>DL Syntax</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A$ (URI reference)</td>
<td>$A$</td>
<td>Conference</td>
<td></td>
</tr>
<tr>
<td>$\text{owl:Thing}$</td>
<td>$\top$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\text{owl:Nothing}$</td>
<td>$\bot$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\text{intersectionOf}(C_1 C_2 \ldots)$</td>
<td>$C_1 \sqcap C_2$</td>
<td>Reference $\sqcap$ Journal</td>
<td></td>
</tr>
<tr>
<td>$\text{unionOf}(C_1 C_2 \ldots)$</td>
<td>$C_1 \sqcup C_2$</td>
<td>Organization $\sqcup$ Institution</td>
<td></td>
</tr>
<tr>
<td>$\text{complementOf}(C)$</td>
<td>$\neg C$</td>
<td>$\neg$ MasterThesis</td>
<td></td>
</tr>
<tr>
<td>$\text{oneOf}(o_1 \ldots)$</td>
<td>${o_1, \ldots}$</td>
<td>${&quot;WISE&quot;,&quot;ISWC&quot;,&quot;\ldots}$</td>
<td></td>
</tr>
<tr>
<td>$\text{restriction}(R \text{someValuesFrom}(C))$</td>
<td>$\exists R.C$</td>
<td>$\exists \text{parts.InCollection}$</td>
<td></td>
</tr>
<tr>
<td>$\text{restriction}(R \text{allValuesFrom}(C))$</td>
<td>$\forall R.C$</td>
<td>$\forall \text{date.Date}$</td>
<td></td>
</tr>
<tr>
<td>$\text{restriction}(R \text{hasValue}(o))$</td>
<td>$\exists R.{o}$</td>
<td>$\exists \text{date.(2005)}$</td>
<td></td>
</tr>
<tr>
<td>$\text{restriction}(R \text{minCardinality}(n))$</td>
<td>$(\geq n R)$</td>
<td>$(\geq 1 \text{location})$</td>
<td></td>
</tr>
<tr>
<td>$\text{restriction}(R \text{maxCardinality}(n))$</td>
<td>$(\leq n R)$</td>
<td>$(\leq 1 \text{publisher})$</td>
<td></td>
</tr>
<tr>
<td>$\text{restriction}(U \text{someValuesFrom}(D))$</td>
<td>$\exists U.D$</td>
<td>$\exists \text{issue.integer}$</td>
<td></td>
</tr>
<tr>
<td>$\text{restriction}(U \text{allValuesFrom}(D))$</td>
<td>$\forall U.D$</td>
<td>$\forall \text{name.string}$</td>
<td></td>
</tr>
<tr>
<td>$\text{restriction}(U \text{hasValue}(v))$</td>
<td>$\exists U. = v$</td>
<td>$\exists \text{series.=&quot;LNCS&quot;}$</td>
<td></td>
</tr>
<tr>
<td>$\text{restriction}(U \text{minCardinality}(n))$</td>
<td>$(\geq n U)$</td>
<td>$(\geq 1 \text{title})$</td>
<td></td>
</tr>
<tr>
<td>$\text{restriction}(U \text{maxCardinality}(n))$</td>
<td>$(\leq n U)$</td>
<td>$(\leq 1 \text{author})$</td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Axioms</th>
<th>Abstract Syntax</th>
<th>DL Syntax</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class((A) partial (C_1) ... (C_n))</td>
<td></td>
<td>(A \sqsubseteq C_1 \sqcap \ldots \sqcap C_n)</td>
<td>Human (\sqsubseteq) Animal (\sqcap) Biped</td>
</tr>
<tr>
<td>Class((A) complete (C_1) ... (C_n))</td>
<td></td>
<td>(A = C_1 \sqcap \ldots \sqcap C_n)</td>
<td>Man = Human (\sqcap) Male</td>
</tr>
<tr>
<td>EnumeratedClass((A) (o_1) ... (o_n))</td>
<td></td>
<td>(A = {o_1} \sqcup \ldots \sqcup {o_n})</td>
<td>RGB = {r} (\sqcup) {g} (\sqcup) {b}</td>
</tr>
<tr>
<td>SubClassOf((C_1) (C_2))</td>
<td></td>
<td>(C_1 \sqsubseteq C_2)</td>
<td></td>
</tr>
<tr>
<td>EquivalentClasses((C_1) ... (C_n))</td>
<td></td>
<td>(C_1 = \ldots = C_n)</td>
<td></td>
</tr>
<tr>
<td>DisjointClasses((C_1) ... (C_n))</td>
<td></td>
<td>(C_i \sqcap C_j = \bot, i \neq j)</td>
<td>Male (\sqcap) Female (\sqsubseteq) (\bot)</td>
</tr>
<tr>
<td>ObjectProperty((R) super ((R_1)) ... super ((R_n))</td>
<td>(R \sqsubseteq R_i)</td>
<td>HasDaughter (\sqsubseteq) hasChild</td>
<td></td>
</tr>
<tr>
<td>domain((C_1)) ... domain((C_n))</td>
<td>((\geq 1 \ R) \sqsubseteq C_i)</td>
<td>((\geq 1) hasChild) (\sqsubseteq) Human</td>
<td></td>
</tr>
<tr>
<td>range((C_1)) ... range((C_n))</td>
<td>(\top \sqsubseteq \forall R.C_i)</td>
<td>(\forall) hasChild.Human</td>
<td></td>
</tr>
<tr>
<td>[inverseof((P))]</td>
<td></td>
<td>(R = P^-)</td>
<td>(\text{hasChild} = \text{hasParent}^-)</td>
</tr>
<tr>
<td>[symmetric]</td>
<td></td>
<td>(R = R^-)</td>
<td>(\text{similar} = \text{similar}^-)</td>
</tr>
<tr>
<td>[functional]</td>
<td></td>
<td>(\top \sqsubseteq (\leq 1 R))</td>
<td>(\top \sqsubseteq (\leq 1 \text{hasMother}))</td>
</tr>
<tr>
<td>[Inversefunctional]</td>
<td></td>
<td>(\top \sqsubseteq (\leq 1 R^-))</td>
<td></td>
</tr>
<tr>
<td>[Transitive]</td>
<td></td>
<td>(Tr(R))</td>
<td>(\text{Tr(ancestor)})</td>
</tr>
<tr>
<td>SubPropertyOf((R_1) (R_2))</td>
<td></td>
<td>(R_1 \sqsubseteq R_2)</td>
<td></td>
</tr>
<tr>
<td>EquivalentProperties((R_1) ... (R_n))</td>
<td></td>
<td>(R_1 = \ldots = R_n)</td>
<td>cost = price</td>
</tr>
<tr>
<td>AnnotationProperty((S))</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Abstract Syntax

<table>
<thead>
<tr>
<th>DatatypeProperty (U) super ((U_1)) \ldots super ((U_n))</th>
<th>DL Syntax</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>domain ((C_1)) \ldots domain ((C_n))</td>
<td>(U \sqsubseteq U_i) ((\geq 1 U) \sqsubseteq C_i)</td>
<td>((\geq 1 \text{hasAge}) \sqsubseteq \text{Human})</td>
</tr>
<tr>
<td>range ((D_1)) \ldots range ((D_n))</td>
<td>(T \sqsubseteq \forall U.D_i) (T \sqsubseteq (\leq 1 U))</td>
<td>(T \sqsubseteq (\leq 1 \text{hasAge}))</td>
</tr>
<tr>
<td>[functional]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SubPropertyOf ((U_1 U_2))</td>
<td>(U_1 \sqsubseteq U_2)</td>
<td>(\text{name} \sqsubseteq \text{hasName})</td>
</tr>
<tr>
<td>EquivalentProperties ((U_1 \ldots U_n))</td>
<td>(U_1 = \ldots = U_n)</td>
<td></td>
</tr>
</tbody>
</table>

### Individuals

| Individual \((o \text{ type } (C_1)\) \ldots \text{ type } \((C_n)\)) | \(o:C_i\) | \(\text{tim:Human}\) |
| value \((R_1 o_1)\) \ldots value \((R_n o_n)\) | \((o, o_i):R_i\) | \((\text{tim, mary}):\text{hasChild}\) |
| value \((U_1 v_1)\) \ldots value \((U_n v_n)\) | \((o, v_i):U_i\) | \((\text{tim, 14}):\text{hasAge}\) |
| SameIndividual \((o_1 \ldots o_n)\) | | \(\text{president_Bush} = \text{G.W.Bush}\) |
| DifferentIndividuals \((o_1 \ldots o_n)\) | | \(\text{john} \neq \text{peter}\) |

### Symbols

| Object Property \(R\) (URI reference) | \(R\) | \(\text{hasChild}\) |
| Datatype Property \(U\) (URI reference) | \(U\) | \(\text{hasAge}\) |
| Individual \(o\) (URI reference) | \(U\) | \(\text{tim}\) |
| Data Value \(v\) (RDF literal) | \(U\) | \(\text{“ESWC07”}\) |
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,

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

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

$$\text{has\_parent}(mary, jo)$$
LPs Semantics: FOL semantics

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

$$A \leftarrow \varphi_1$$
$$A \leftarrow \varphi_2$$
$$\vdots$$
$$A \leftarrow \varphi_n$$

with $A \leftarrow \varphi_1 \lor \varphi_2 \lor \ldots \lor \varphi_n$.

- Note: in $\mathcal{P}^\ast$ 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 $r \in \mathcal{P}^\ast$ $I \models r$, where $I \models A \leftarrow \varphi$ iff $I(\varphi) \leq I(A)$
- Least model exists and is least fixed-point of

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

Toy Example

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

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

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

- **Combine** DLs with LPs:
  - DL atoms and roles may appear in rules
    
    \[
    \text{buy}(x) \leftarrow \text{Electronics}(x), \text{offer}(x)
    \]
    \[
    \text{Camera} \sqsubseteq \text{Electronics}
    \]

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

- Many different approaches exist with different semantics: we present the basics of two of them
Loosely Coupled DL-Programs [3, 4, 5]

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

- A dl-rule \( r \) is of form

  \[ a \leftarrow b_1, \ldots, b_k \]

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

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

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

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

- Minimal model exists and fixed-point characterization:
  $$T_P(I)(A) = I^L(\varphi), \text{ for all } A \leftarrow \varphi \in P^*$$

- Example: $buy(x) \leftarrow DL[Camera](x)$
  $buy(x) \leftarrow DL[DVDPlayer](x)$

  $a:Camera \sqcup b:Camera$ \sqcup DVDPlayer

  $KB \models buy(a)$ \quad $KB \not\models buy(b)$
Tightly Coupled DL-Programs [7]

- A dl-atom may appear anywhere in the rule (rule head and/or rule body)
- $I \models P$ is defined as usual.
- $I \models L$ iff $L \cup \{a \mid I(a) = 1\} \cup \{\neg a \mid I(a) = 0\}$ is satisfiable.
- $I \models KB$ iff $I \models L$ and $I \models P$.
- Many minimal models may exist.
- $KB \models_{\text{cautious}} a$ iff for all minimal models $I$ of $KB$, $I \models a$
- $KB \models_{\text{brave}} a$ iff for some minimal models $I$ of $KB$, $I \models a$
- Clearly, $\models_{\text{cautious}} \subseteq \models_{\text{brave}}$
- Example: $\text{buy}(x) \leftarrow \text{DL[Camera]}(x)$
  $\text{buy}(x) \leftarrow \text{DL[DVDPlayer]}(x)$

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

  $KB \models_{\text{cautious}} \text{buy}(a) \quad KB \models_{\text{cautious}} \text{buy}(b)$

- Note: [7] considers non-monotone negation and disjunctive LPs
C. V. Damásio, J. Z. Pan, G. Stoilos, and U. Straccia.
An approach to representing uncertainty rules in ruleml.

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

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

- $P = \{(\text{have\_legs} \mid \text{bird})[1, 1], (\text{fly} \mid \text{bird})[0.95, 1]\}$:
  
  “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 $\Phi$.
- $\mathcal{I}_\Phi$: all worlds for $\Phi$.
- **Probabilistic interpretation $Pr$:** probability function on $\mathcal{I}_\Phi$.
- $Pr(\phi)$: sum of all $Pr(I)$ such that $I \in \mathcal{I}_\Phi$ and $I \models \phi$.
- $Pr(\psi | \phi)$: if $Pr(\phi) > 0$, then $Pr(\psi | \phi) = \frac{Pr(\psi \land \phi)}{Pr(\phi)}$.
- **Truth under $Pr$:**
  - $Pr \models \psi \iff \phi$ 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 $\Phi = \{\text{bird, fly}\}$.
- $\mathcal{I}_\Phi$ contains exactly the worlds $l_1, l_2, l_3,$ and $l_4$ over $\Phi$:

<table>
<thead>
<tr>
<th></th>
<th>fly</th>
<th>¬fly</th>
</tr>
</thead>
<tbody>
<tr>
<td>bird</td>
<td>$l_1$</td>
<td>$l_2$</td>
</tr>
<tr>
<td>¬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>¬fly</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pr$_1$</td>
<td></td>
<td></td>
</tr>
<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>
<tr>
<td>Pr$_2$</td>
<td></td>
<td></td>
</tr>
<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 $Pr_2(bird) = 1/3$.
- $\neg fly \iff bird$ is false in $Pr_1$, but true in $Pr_2$.
- $(fly | bird)[.95, 1]$ is true in $Pr_1$, but false in $Pr_2$. 
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:

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

- \( KB \) is satisfiable, since

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

- Some conclusions under logical entailment:

\[ KB \models (\text{have_legs} \mid \text{bird})[0.3, 1], \quad KB \models (\text{fly} \mid \text{bird})[0.6, 1]. \]

- Tight conclusions under logical entailment:

\[ KB \models_{\text{tight}} (\text{have_legs} \mid \text{bird})[1, 1], \quad KB \models_{\text{tight}} (\text{fly} \mid \text{bird})[0.95, 1], \]
\[ KB \models_{\text{tight}} (\text{have_legs} \mid \text{eagle})[1, 1], \quad KB \models_{\text{tight}} (\text{fly} \mid \text{eagle})[0, 1]. \]
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 $R = \{ l \in \mathcal{I}_\Phi \mid l \models L \}$, is solvable:

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

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

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

$$
y_r \geq 0 \quad (\text{for all } r \in R)
$$
**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 $R = \{ I \in I_\Phi | I \models L \}$:

minimize (resp., maximize) $\sum_{r \in R} y_r$ subject to

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

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

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

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


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 $H(Pr)$:

  $$H(Pr) = - \sum_{I \in I_\phi} Pr(I) \cdot \log Pr(I).$$

- The **ME model** of a satisfiable probabilistic knowledge base $KB$ is the unique probabilistic interpretation $Pr$ that is a model of $KB$ and that has the greatest entropy among all the models of $KB$.

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

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

Probabilistic knowledge base:

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

Tight conclusions under logical entailment:

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

Tight conclusions under maximum entropy entailment:

\[ KB \models_{\text{me\_tight}} (\text{have\_legs} \mid \text{bird})[1, 1], \quad KB \models_{\text{me\_tight}} (\text{fly} \mid \text{bird})[0.95, 0.95], \]
\[ KB \models_{\text{me\_tight}} (\text{have\_legs} \mid \text{eagle})[1, 1], \quad KB \models_{\text{me\_tight}} (\text{fly} \mid \text{eagle})[0.95, 0.95]. \]
Lexicographic Entailment

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

- $Pr$ is \textit{lex}-preferable to $Pr'$ iff some $i \in \{0, \ldots, k\}$ exists such that
  - $|\{C \in P_i | Pr \models C\}| > |\{C \in P_i | Pr' \models C\}|$ and
  - $|\{C \in P_j | Pr \models C\}| = |\{C \in P_j | Pr' \models C\}|$ for all $i < j \leq k$.

- A model $Pr$ of $F$ is a \textit{lex}-minimal model of $F$ iff
  no model of $F$ is \textit{lex}-preferable to $Pr$.

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

- $KB \models^{\text{lex\_tight}} (\psi | \phi)[l, u]$: $(\psi | \phi)[l, u]$ is a tight \textit{lex}-consequence of $KB$ iff
  $l$ (resp., $u$) is the infimum (resp., supremum) of $Pr(\psi)$ subject to all \textit{lex}-minimal models $Pr$ of $L$ with $Pr(\phi) = 1$. 
Logical vs. Lexicographic Entailment

Probabilistic knowledge base:

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

Tight conclusions under logical entailment:

\[ KB \models_{t} (have\_legs \mid bird)[1, 1], \ KB \models_{t} (fly \mid bird)[0.95, 1], \ KB \models_{t} (have\_legs \mid eagle)[1, 1], \ KB \models_{t} (fly \mid eagle)[0, 1]. \]

Tight conclusions under probabilistic lexicographic entailment:

\[ KB \models_{t}^{\text{lex}} (have\_legs \mid bird)[1, 1], \ KB \models_{t}^{\text{lex}} (fly \mid bird)[0.95, 1], \ KB \models_{t}^{\text{lex}} (have\_legs \mid eagle)[1, 1], \ KB \models_{t}^{\text{lex}} (fly \mid eagle)[0.95, 1]. \]
Probabilistic knowledge base:

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

Tight conclusions under logical entailment:

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

Tight conclusions under probabilistic lexicographic entailment:

\[ KB \models_{\text{lex}} (\text{have\_legs} \mid bird)[1, 1], \ KB \models_{\text{lex}} (\text{fly} \mid bird)[1, 1], \ KB \models_{\text{lex}} (\text{have\_legs} \mid penguin)[1, 1], \ KB \models_{\text{lex}} (\text{fly} \mid penguin)[0, 0.05]. \]
Probabilistic knowledge base:

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

Tight conclusions under logical entailment:

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

Tight conclusions under probabilistic lexicographic entailment:

\[ KB \models_{\text{lex}}^{\text{tight}} (\text{have\_legs} \mid bird)[0.99, 1], \ KB \models_{\text{lex}}^{\text{tight}} (\text{fly} \mid bird)[0.95, 1], \\
KB \models_{\text{lex}}^{\text{tight}} (\text{have\_legs} \mid penguin)[0.99, 1], \ KB \models_{\text{lex}}^{\text{tight}} (\text{fly} \mid penguin)[0, 0.05]. \]
Literature


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

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

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

- a set of nodes; each node represents a random variable
- a directed, acyclic graph (link \(\approx\) “directly influences”)
- a conditional distribution for each node given its parents: 
  \[
  P(X_i|Parents(X_i))
  \]

Any joint distribution can be represented as a BN.
Example

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

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

Network topology reflects “causal” knowledge:

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

John sometimes confuses the telephone ringing with the alarm. Mary likes rather loud music and sometimes misses the alarm.
Uncertainty in Semantic Web Languages

Combining Uncertainty and Vagueness in SW Languages

Uncertainty and RDF/DLs/OWL

Uncertainty and LPs/DLPs

Uncertainty

Vagueness

In the context of the Semantic Web, uncertainty and vagueness are crucial aspects to consider when modeling and reasoning about data. The figure illustrates how uncertainty and vagueness can be represented and combined within Semantic Web (SW) languages. For instance, uncertainties are often modeled using probabilistic frameworks, where propositions are assigned probabilities. The table below demonstrates this with the following entries:

<table>
<thead>
<tr>
<th>Condition</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Burglary</td>
<td>.001</td>
</tr>
<tr>
<td>Earthquake</td>
<td>.002</td>
</tr>
<tr>
<td>Alarm</td>
<td></td>
</tr>
<tr>
<td>JohnCalls</td>
<td></td>
</tr>
<tr>
<td>MaryCalls</td>
<td></td>
</tr>
</tbody>
</table>

In the table, the entries represent the likelihood of various events under different conditions. For example, the probability of Burglary occurring is .001, and the probability of Earthquake occurring is .002. The Alarm, JohnCalls, and MaryCalls are represented with conditional probabilities as follows:

<table>
<thead>
<tr>
<th>Condition</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Burglary</td>
<td></td>
</tr>
<tr>
<td>Earthquake</td>
<td></td>
</tr>
<tr>
<td>JohnCalls</td>
<td>.90</td>
</tr>
<tr>
<td>MaryCalls</td>
<td>.70</td>
</tr>
</tbody>
</table>

These examples highlight how uncertainty and vagueness are integrated into the Semantic Web to enhance its expressive power and practical applicability.
Global Semantics

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

\[ P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i|\text{Parents}(X_i)) \]

e.g.,

\[
P(j \land m \land a \land \neg b \land \neg e) = P(j|a)P(m|a)P(a|\neg b, \neg e)P(\neg b)P(\neg e) = 0.90 \times 0.70 \times 0.001 \times 0.999 \times 0.998 = 0.00062
\]
Inference Tasks

- **Simple queries**: compute posterior marginal $P(X_i|E = e)$, e.g., $P(\text{Burglary}|\text{Alarm} = \text{true, John} = \text{true, Mary} = \text{false})$.

- **Conjunctive queries**: $P(X_i, X_j|E = e) = P(X_i|E = e)P(X_j|X_i, E = e)$.

- **Optimal decisions**: decision networks include utility information; probabilistic inference required for $P(\text{outcome}|\text{action, evidence})$.

- **Value of information**: which evidence to seek next?

- **Sensitivity analysis**: which probability values are most critical?
Probabilistic Causal Models

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

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

Sophisticated notions of causes and explanations.

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

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

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

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

Two arsonists lit matches \((A_i = 1), \ i \in \{1, 2\}\), in different parts of a dry forest, and both cause trees to start burning. Either match by itself suffices to burn down the whole forest \((B = 1)\): \n
\[ \begin{align*} U_1 & \rightarrow A_1 \rightarrow B \\ U_2 & \rightarrow A_2 \rightarrow B \end{align*} \]

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

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

Literature

- Online tutorials, software packages, and datasets on BNs:
  - http://www.auai.org/
Probabilities about Generic and Concrete Objects

Combining generic and concrete probability distributions:

- **Conditioning**: Generic distributions are conditioned on the (classical) information about concrete distributions.

- **Probabilistic default reasoning**: Generic distributions are constrained by the (not necessarily classical) information about the concrete distributions, and techniques from default reasoning resolve local inconsistencies.

- **Minimum cross entropy**: Generic and concrete distributions are combined via cross entropy minimization.
Probabilistic Ontologies

Main types of encoded probabilistic knowledge:

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

Main types of reasoning problems:

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

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

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

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

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


- 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 \textit{SHOQ}(\textbf{D}) (recently also extended to \textit{SHIF}(\textbf{D}) and \textit{SHOIN}(\textbf{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

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

- probabilistic generalization (of a variant) of the description logic CLASSIC
- so-called *p-classes* express terminological probabilistic knowledge about concepts, roles, and attributes
- but assertional classical and probabilistic knowledge about instances of concepts and roles is not supported
- probabilistic semantics based on Bayesian networks
- determines exact probabilities for conditionals between concept expressions in canonical form
- probabilistic inference can be done in polynomial time, when the underlying Bayesian network is a polytree
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 $\pi$: $\text{Poss}(\phi) = \max \{\pi(I) \mid I \in \mathcal{I}_\Phi, I \models \phi\}$

Probabilistic OWL


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

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):

Poole’s Independent Choice Logic (ICL)

Acyclic logic programs $P$ under different “choices”.
Each choice along with $P$ produces a first-order model.
By placing a probability distribution over the different choices, one then obtains a distribution over the set of first-order models.
ICL generalizes Pearl’s structural causal models.
ICL also generalizes Bayesian networks, influence diagrams, Markov decision processes, and normal form games.
Example

Sequence of three not-gates:

\[
\begin{array}{c c c}
\text{in}(i1) & 1 & \text{out}(i3) \\
\text{i1} & 1 & \text{i2} \\
\text{i2} & 1 & \text{i3}
\end{array}
\]

\[
\begin{align*}
\text{val}(\text{out}(G), \text{on}, T) & \leftarrow \text{ok}(G) \land \text{val}(\text{in}(G), \text{off}, T). \\
\text{val}(\text{out}(G), \text{off}, T) & \leftarrow \text{ok}(G) \land \text{val}(\text{in}(G), \text{on}, T). \\
\text{val}(\text{out}(G), V, T) & \leftarrow \text{shorted}(G) \land \text{val}(\text{in}(G), V, T). \\
\text{val}(\text{out}(G), \text{off}, T) & \leftarrow \text{blown}(G). \\
\text{val}(\text{in}(G), V, T) & \leftarrow \text{conn}(G_1, G) \land \text{val}(\text{out}(G_1), V, T). \\
\text{conn}(i_1, i_2) & \leftarrow . \\
\text{conn}(i_2, i_3) & \leftarrow . \\
\text{disjoint}([\text{ok}(G):0.95, \text{shorted}(G):0.03, \text{blown}(G):0.02]). \\
\text{disjoint}([\text{val}(\text{in}(i_1), \text{on}, T):0.5, \text{val}(\text{in}(i_1), \text{off}, T):0.5]).
\end{align*}
\]
Possible queries: Which is the probability that gate $i_2$ is ok given that both the input of $i_1$ and the output of $i_3$ are off at the time point $t_1$?

$$P(ok(i_2) | \text{val(in(i_1), off, } t_1) \land \text{val(out(i_3), off, } t_1)) = 0.76.$$ 

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

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

Intuitively: Every closed formula is associated with a set of minimal explanations. Every explanation is a set of hypotheses. The probability of an explanation is the product of the probabilities of the hypotheses. The probability of a closed formula is the sum of the probabilities of all associated minimal explanations.
The formula \( F = \text{val}(\text{in}(i_1), \text{off}, t_1) \land \text{val}(\text{out}(i_3), \text{off}, t_1) \) is associated with the following minimal explanations along with their probabilities:

\[
E_1 = \{ \text{val}(\text{in}(i_1), \text{off}, t_1), \text{ok}(i_3), \text{ok}(i_2), \text{shorted}(i_1) \} \\
P(E_1) = 0.5 \times 0.95 \times 0.95 \times 0.03 = 0.01354
\]

\[
E_2 = \{ \text{val}(\text{in}(i_1), \text{off}, t_1), \text{ok}(i_3), \text{shorted}(i_2), \text{ok}(i_1) \} \\
P(E_2) = 0.5 \times 0.95 \times 0.03 \times 0.95 = 0.01354
\]

\[\vdots\]

The sum of the probabilities of all minimal explanations associated with \( F \) is 0.05996. Hence, the formula \( F \) has the probability 0.05996.
Probabilistic Description Logic Programs


- 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 $KB = (L, P, C, \mu)$:

$$PC \sqcup \text{Camera} \sqsubseteq \text{Electronics}; \quad PC \sqcap \text{Camera} \sqsubseteq \bot;$$
$$\text{Book} \sqcup \text{Electronics} \sqsubseteq \text{Product}; \quad \text{Book} \sqcap \text{Electronics} \sqsubseteq \bot;$$
$$\text{Textbook} \sqsubseteq \text{Book};$$

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

$\text{Textbook}(tb_{ai}); \quad \text{Textbook}(tb_{lp});$
$\text{PC}(pc_{ibm}); \quad \text{PC}(pc_{hp});$

$\text{related}(tb_{ai}, tb_{lp}); \quad \text{related}(pc_{ibm}, pc_{hp});$
$\text{provides}(ibm, pc_{ibm}); \quad \text{provides}(hp, pc_{hp}).$
Classical dl-rules in $P$

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

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

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

$\mu$: $avoid\_pos, avoid\_neg \mapsto 0.9, 0.1; \ offer\_pos, offer\_neg \mapsto 0.9, 0.1;

v\_buy\_pos, v\_buy\_neg \mapsto 0.7, 0.3; \ a\_buy\_pos, a\_buy\_neg \mapsto 0.7, 0.3.$

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

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

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

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

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

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

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

$$\mu_M : \text{inst}_A, \text{inst}_B, \text{inst}_C \mapsto 0.6, 0.3, 0.1 . $$
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 $KB_M = (\emptyset, P_M, C_M, \mu_M)$:

$P_M = \{ logic_programming(X) \leftarrow rule-based_systems(X), choice_1 ; \\
logic_programming(X) \leftarrow deductive_databases(X), choice_2 \} ;$

$C_M = \{ \{ choice_1, not_choice_1 \}, \{ choice_2, not_choice_2 \} \} ;$

$\mu_M : choice_1, not_choice_1, choice_2, not_choice_2 \mapsto 0.7, 0.3, 0.8, 0.2.$
Outline

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 SW Languages
Vagueness basics

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 \(0.83 \land 0.2\)?
  - E.g., In multimedia retrieval: if a image region is white to degree 0.8 and the object is about a dog to degree 0.4, to which degree is the image about a “white dog”? That is, what is \(0.8 \land 0.4\)?
- 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 \(\mathcal{I} : \text{Atoms} \rightarrow [0, 1]\)
- Interpretations are **extended** to formulae using **norms** to interpret connectives \(\land, \lor, \neg, \rightarrow\)

<table>
<thead>
<tr>
<th><strong>Negation</strong></th>
<th><strong>T-norm (conjunction)</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>(n(0) = 1)</td>
<td>(t(a, 1) = a)</td>
</tr>
<tr>
<td>(a \leq b) implies (n(b) \leq n(a))</td>
<td>(b \leq c) implies (t(a, b) \leq t(a, c))</td>
</tr>
<tr>
<td></td>
<td>(t(a, b) = t(b, a))</td>
</tr>
<tr>
<td></td>
<td>(t(a, t(b, c)) = t(t(a, b), c))</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>S-norm (disjunction)</strong></th>
<th><strong>I-norm (implication)</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>(s(a, 0) = a)</td>
<td>(a \leq b) implies (i(a, c) \geq i(b, c))</td>
</tr>
<tr>
<td>(b \leq c) implies (s(a, b) \leq s(a, c))</td>
<td>(b \leq c) implies (i(a, b) \leq i(a, c))</td>
</tr>
<tr>
<td>(s(a, b) = s(b, a))</td>
<td>(i(0, b) = 1)</td>
</tr>
<tr>
<td>(s(a, s(b, c)) = s(s(a, b), c))</td>
<td>(i(a, 1) = 1)</td>
</tr>
</tbody>
</table>

\(i(a, b) = \sup \{c : t(a, c) \leq b\}\) is called **r-implication** and depends on the t-norm only
**Typical norms**

<table>
<thead>
<tr>
<th>Condition</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 $y/x$</td>
<td>$\max(1 - x, y)$</td>
</tr>
</tbody>
</table>

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

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

\( \neg \phi \) is \( \phi \rightarrow 0 \)

\( \phi \bar{\land} \psi \) defined as \( \phi \land (\phi \rightarrow \psi) \)

\( \phi \bar{\lor} \psi \) defined as \( (\phi \rightarrow \psi) \rightarrow \psi) \bar{\land}((\psi \rightarrow \phi) \rightarrow \phi) \)

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

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

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

\( \neg Z \phi = \neg \mathbb{L} \phi \)

\( \phi \land Z \psi = \phi \land \mathbb{L} (\phi \rightarrow \mathbb{L} \psi) \)

\( \phi \rightarrow Z \psi = \neg \mathbb{L} \phi \lor \mathbb{L} \psi \)
Some additional properties of t-norms, s-norms, implication functions, and negation functions of various fuzzy logics.

<table>
<thead>
<tr>
<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>( \exists x. x \land \neg x \neq 0 )</td>
<td>( \exists x. x \land \neg x \neq 0 )</td>
<td>( \exists x. x \land \neg x \neq 0 )</td>
</tr>
<tr>
<td>( x \lor \neg x = 1 )</td>
<td>( \exists x. x \lor \neg x \neq 1 )</td>
<td>( \exists x. x \lor \neg x \neq 1 )</td>
<td>( \exists x. x \lor \neg x \neq 1 )</td>
</tr>
<tr>
<td>( \exists x. x \land x \neq x )</td>
<td>( x \land x = x )</td>
<td>( \exists x. x \land x \neq x )</td>
<td>( x \land x = x )</td>
</tr>
<tr>
<td>( \exists x. x \lor x \neq x )</td>
<td>( x \lor x = x )</td>
<td>( \exists x. x \lor x \neq x )</td>
<td>( x \lor x = x )</td>
</tr>
<tr>
<td>( \neg \neg x = x )</td>
<td>( \exists x. \neg x \neq x )</td>
<td>( \exists x. \neg x \neq x )</td>
<td>( \neg x = x )</td>
</tr>
<tr>
<td>( x \to y = \neg x \lor y )</td>
<td>( \exists x. x \to y \neq \neg x \lor y )</td>
<td>( \exists x. x \to y \neq \neg x \lor y )</td>
<td>( x \to y = \neg x \lor y )</td>
</tr>
<tr>
<td>( \neg (x \to y) = x \land \neg y )</td>
<td>( \exists x. \neg (x \to y) \neq x \land \neg y )</td>
<td>( \exists x. \neg (x \to y) \neq x \land \neg y )</td>
<td>( \neg (x \to y) = x \land \neg y )</td>
</tr>
<tr>
<td>( \neg (x \land y) = \neg x \lor \neg y )</td>
<td>( \neg (x \land y) = \neg x \lor \neg y )</td>
<td>( \neg (x \land y) = \neg x \lor \neg y )</td>
<td>( \neg (x \land y) = \neg x \lor \neg y )</td>
</tr>
<tr>
<td>( \neg (x \lor y) = \neg x \land \neg y )</td>
<td>( \neg (x \lor y) = \neg x \land \neg y )</td>
<td>( \neg (x \lor y) = \neg x \land \neg y )</td>
<td>( \neg (x \lor y) = \neg x \land \neg y )</td>
</tr>
</tbody>
</table>
Axioms of logic BL (Basic Fuzzy Logic)

Fix arbitray t-norm and r-implication.

(A1) \((\phi \rightarrow \psi) \rightarrow ((\psi \rightarrow \chi) \rightarrow \phi \rightarrow \chi)\)
(A2) \((\phi \land \psi) \rightarrow \phi\)
(A3) \((\phi \land \psi) \rightarrow (\psi \land \phi)\)
(A4) \((\phi \land (\phi \rightarrow \psi)) \rightarrow (\psi \land (\psi \rightarrow \phi))\)
(A5a) \((\phi \land (\psi \rightarrow \chi)) \rightarrow ((\phi \land \psi) \rightarrow \chi)\)
(A5b) \(((\phi \land \psi) \rightarrow \chi)) \rightarrow (\phi \land (\psi \rightarrow \chi))\)
(A6) \((\phi \land (\psi \rightarrow \chi)) \rightarrow (((\psi \rightarrow \phi) \rightarrow \chi)) \rightarrow \chi)\)
(A7) \(0 \rightarrow \phi\)

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

Proposition

\(\mathcal{T} \vdash_{BL} \phi\) iff \(\mathcal{T} \models_{BL} \phi\). Also, if \(\mathcal{T} \vdash_{BL} \phi\) then \(\mathcal{T} \models_{BL2} \phi\), but not vice-versa

(e.g. \(\models_{BL2} \phi \lor \neg \phi\), but \(\not\models_{BL} \phi \lor \neg \phi\).)

- \(\models_{BL} \phi \land \neg \phi \rightarrow 0\)
- \(\models_{BL} \phi \rightarrow \neg \neg \phi\), but \(\not\models_{BL} \neg \neg \phi \rightarrow \phi\), e.g. \(\phi = p \lor \neg p\), t-norm is Gödel
- \(\models_{BL} (\phi \rightarrow \psi) \rightarrow (\neg \psi \rightarrow \neg \phi)\), but not vice-versa
Axioms of Łukasiewicz logic Ł

Fix Łukasiewicz t-norm and r-implication.

(Axioms) Axioms of BL

(Ł) \( \neg \neg \phi \rightarrow \phi \)

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

Proposition

\( T \vdash _L \phi \iff T \models _L \phi \).

- \( \models _L \phi \rightarrow \psi \equiv \neg \psi \rightarrow \neg \phi \)
- \( \models _L \neg (\phi \land \psi) \equiv \neg \phi \lor \neg \psi \)
- \( \models _L \phi \rightarrow \psi \equiv \neg (\phi \land \neg \psi) \)
- \( \models _L \phi \rightarrow \psi \equiv \neg \phi \lor \neg \psi \)
- \( \models _L \neg (\phi \rightarrow \psi) \equiv \phi \land \neg \psi \)

Recall that “Zadeh logic” is a sub-logic of Ł.
Fix product t-norm and \( r \)-implication.

(Axioms) Axioms of BL

\[(\Pi 1) \; \neg\neg \chi \rightarrow ((\phi \land \chi \rightarrow \psi \land \chi) \rightarrow (\phi \rightarrow \psi))\]

\[(\Pi 2) \; (\phi \bar{\land} \neg \phi) \rightarrow 0\]

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

Proposition

\[T \vdash \Pi \phi \iff T \models \Pi \phi.\]

- \( \models \Pi \neg(\phi \land \psi) \rightarrow \neg(\phi \bar{\land} \psi)\)
- \( \models \Pi (\phi \rightarrow \neg \phi) \rightarrow \neg \phi\)
- \( \models \Pi \neg \phi \bar{\lor} \neg \neg \phi\)
Axioms of Gödel logic $G$

Fix Gödel t-norm and r-implication.

**Axioms of BL**

\[(G) \quad \phi \to (\phi \land \phi)\]

**Deduction rule** Modus ponens: from $\phi$ and $\phi \to \psi$ infer $\psi$

**Proposition**

$T \vdash_G \phi$ iff $T \models_G \phi$.

- $\models_G (\phi \land \psi) \equiv (\bar{\phi} \land \psi)$
- Gödel logic proves all axioms of intuitionistic logic I, vice-versa I + (A6) proves all axioms of Gödel logic
Axioms of Boolean logic

Fix interpretations to be boolean.

(Axioms) Axioms of BL

(BL2) $\phi \lor \neg \phi$

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

Proposition

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

- $\models_{BL2} \phi \rightarrow (\phi \land \phi)$ (BL2 extends G)
- $\mathcal{L} + G$ is equivalent to BL2
- $\mathcal{L} + \Pi$ is equivalent to BL2
- G + $\Pi$ is equivalent to BL2
### Axioms of Rational Pavelka Logic (RPL)

- Fix Łukasiewicz t-norm and r-implication
- Rational $r \in [0, 1]$ may appear as atom in formula. $\mathcal{I}(r) = r$
- Note: $\mathcal{I}(r \rightarrow \phi) = 1$ iff $\mathcal{I}(\phi) \geq r$. Also, $\mathcal{I}(\phi \rightarrow r) = 1$ iff $\mathcal{I}(\phi) \leq r$

**(Axioms)** Axioms of Ł

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

---

**Proposition**

$$
\mathcal{T} \vdash_{RPL} \phi \iff \mathcal{I} \models_{RPL} \phi.
$$

- RPL proves the derived deduction rule ($r, s \in [0, 1]$): from $r \rightarrow \phi$ and $s \rightarrow (\phi \rightarrow \psi)$ infer $(r \land s) \rightarrow \psi$
- From $\phi \geq r$ and $(\phi \rightarrow \psi) \geq s$ infer $\psi \geq r \land s$

- Let

$$
\begin{align*}
||\phi||_\mathcal{T} & = \inf\{\mathcal{I}(\phi) \mid \mathcal{I} \models \mathcal{T}\} \quad \text{(truth degree)} \\
|\phi|_\mathcal{T} & = \sup\{r \mid \mathcal{T} \vdash r \rightarrow \phi\} \quad \text{(provability degree)}
\end{align*}
$$

then $||\phi||_\mathcal{T} = |\phi|_\mathcal{T}$

- Also,

$$
\begin{align*}
|\neg\phi|_\mathcal{T} & = \sup\{r \mid \mathcal{T} \vdash r \rightarrow \phi\} = 1 - |\phi|_\mathcal{T} \\
|\phi|_\mathcal{T} & = \inf\{s \mid \mathcal{T} \vdash \phi \rightarrow s\}
\end{align*}
$$
Tableau for Rational Pavelka Logic using MILP

|φ|Τ = min x. such that Τ ∪ {φ → x} satisfiable.

- We use MILP (Mixed Integer Linear Programming) to compute |φ|Τ
- Let \( r \in [0, 1] \), variable or expression \( 1 - r' \) (\( r' \) variable), admitting solution in \( [0, 1] \), \( \neg r = 1 - r \), \( \neg\neg r = r \)

\[
\begin{align*}
\text{If} & \quad r \to p & \text{then} & \quad x_p \geq r, x_p \in [0, 1] \\
\text{If} & \quad p \to r & \text{then} & \quad x_p \leq r, x_p \in [0, 1] \\
\text{If} & \quad r \to \neg\phi & \text{then} & \quad \phi \to \neg r \\
\text{If} & \quad \neg\phi \to r & \text{then} & \quad \neg r \to \phi \\
\text{If} & \quad r \to (\phi \land \psi) & \text{then} & \quad x_1 \to \phi, x_2 \to \psi, y \leq 1 - r, x_i \leq 1 - y, x_1 + x_2 = r + 1 - y, \\
& & & \quad x_i \in [0, 1], y \in \{0, 1\} \\
\text{If} & \quad (\phi \land \psi) \to r & \text{then} & \quad x_1 \to \neg\phi, x_2 \to \neg\psi, x_1 + x_2 = 1 - r, x_i \in [0, 1] \\
\text{If} & \quad r \to (\phi \to \psi) & \text{then} & \quad \phi \to x_1, x_2 \to \psi, r + x_1 - x_2 = 1, x_i \in [0, 1] \\
\text{If} & \quad (\phi \to \psi) \to r & \text{then} & \quad x_1 \to \phi, \psi \to x_2, y - r \leq 0, y + x_1 \leq 1, y \leq x_2, y + r + x_1 - x_2 = 1, \\
& & & \quad x_i \in [0, 1], y \in \{0, 1\} \\
\end{align*}
\]

- Now we have to solve a MILP problem of the form

\[
\min \mathbf{c} \cdot \mathbf{x} \text{ s.t. } A\mathbf{x} + B\mathbf{y} \geq \mathbf{h}
\]

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

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

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

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

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

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

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

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

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

Note: A similar technique can be used for logic $G$ and $\Pi$, but mixed integer non-linear programming is needed in place of MILP
Predicate Fuzzy Logics Basics [5]

- **Formulae**: First-Order Logic formulae, *terms* are either variables or constants
  - we may introduce functions symbols as well, with crisp semantics (but uninteresting), or we need to discuss also fuzzy equality (which we leave out here)

- **Truth space** is [0, 1]
- **Formulae** have a degree of truth in [0, 1]
- **Interpretation**: is a mapping $\mathcal{I} : \text{Atoms} \rightarrow [0, 1]$

Interpretations are extended to formulae as follows:

\[
\begin{align*}
\mathcal{I}(\neg \phi) &= \mathcal{I}(\phi) \rightarrow 0 \\
\mathcal{I}(\phi \land \psi) &= \mathcal{I}(\phi) \land \mathcal{I}(\psi) \\
\mathcal{I}(\phi \rightarrow \psi) &= \mathcal{I}(\phi) \rightarrow \mathcal{I}(\psi) \\
\mathcal{I}(\exists x \phi) &= \sup_{c \in \Delta \mathcal{I}} \mathcal{I}^c_x(\phi) \\
\mathcal{I}(\forall x \phi) &= \inf_{c \in \Delta \mathcal{I}} \mathcal{I}^c_x(\phi)
\end{align*}
\]

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

- **Definitions of $\mathcal{I} \models \phi$, $\mathcal{I} \models \mathcal{T}$, $\models \phi$, $\mathcal{T} \models \phi$, $\models \phi|_{\mathcal{I}}$ and $\models \phi|_{\mathcal{T}}$ are as for the propositional case**
Axioms of logic $\mathcal{C}\forall$, where $\mathcal{C} \in \{\text{BL,Ł,Π,G}\}$

(Axioms) Axioms of $\mathcal{C}$

$(\forall 1)$ $\forall x \phi(x) \rightarrow \phi(t)$ ($t$ substitutable for $x$ in $\phi(x)$)

$(\exists 1)$ $\phi(t) \rightarrow \exists x \phi(x)$ ($t$ substitutable for $x$ in $\phi(x)$)

$(\forall 2)$ $\forall x (\psi \rightarrow \phi) \rightarrow (\psi \rightarrow \forall x \phi)$ ($x$ not free in $\psi$)

$(\exists 2)$ $\forall x (\phi \rightarrow \psi) \rightarrow (\exists x \phi \rightarrow \psi)$ ($x$ not free in $\psi$)

$(\forall 3)$ $\forall x (\phi \lor \psi) \rightarrow (\forall x \phi) \lor \psi$ ($x$ not free in $\psi$)

(Modus ponens) from $\phi$ and $\phi \rightarrow \psi$ infer $\psi$

(Generalization) from $\phi$ infer $\forall x \phi$

Proposition

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

- if $\rightarrow$ is an r-implication then $||\psi||_{\mathcal{T}} \geq ||\phi||_{\mathcal{T}} \land ||\phi \rightarrow \psi||_{\mathcal{T}}$
- $\models_{\text{BL}\forall} \exists x \phi \rightarrow \neg \forall x \neg \phi$
- $\models_{\text{BL}\forall} \neg \exists x \phi \equiv \forall x \neg \phi$
- $\models_{\text{Ł}\forall} \exists x \phi \equiv \neg \forall x \neg \phi$
\( (-\forall x p(x)) \land (-\exists x \neg p(x)) \) 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 \( p^I(n) = 1/n \)

\[
(\forall x p(x))^I = \inf_n 1/n = 0
\]

\[
(\exists x \neg p(x))^I = \sup_n -1/n = \sup 0 = 0
\]

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

\[
\Delta^I = \{ n \mid \text{integer } n \geq 1 \}
\]
\[
p^I(n) = 1 - 1/n < 1, \text{ for all } n
\]
\[
(\exists x p(x))^I = \sup_n 1 - 1/n = 1
\]

**Witnessed formula:** \( \exists x \phi(x) \) is witnessed in \( I \) iff there is \( c \in \Delta^I \) such that \( (\exists x \phi(x))^I = (\phi(c))^I \)

(similarly for \( \forall x \phi(x) \))

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

**Proposition**

*In Ł, \( \phi \) is satisfiable iff there is a witnessed model of \( \phi \).*

The proposition does not hold for logic G and \( \Pi \)
Predicate Rational Pavelka Logic (RPL\(\forall\))

- Fix Łukasiewicz t-norm and r-implication
- Formulae are as for Ł\(\forall\), where rationals \(r \in [0, 1]\) may appear as atoms

(Axioms and rules) As for Ł\(\forall\)

Proposition

\(\mathcal{T} \vdash_{RPL\forall} \phi \iff \mathcal{T} \models_{RPL\forall} \phi.\)
Fuzzy RDF (we generalize [15, 16, 34])

- Statement (triples) may have attached a degree in [0, 1]:
  for \( n \in [0, 1] \)
  \[
  \langle (\text{subject}, \text{predicate}, \text{object}), n \rangle
  \]
  Meaning: the degree of truth of the statement is at least \( n \)
  - For instance,
    \[
    \langle (o1, \text{IsAbout}, \text{snoopy}), 0.8 \rangle
    \]
In Fuzzy RDF MT, an interpretation \( I \) of a vocabulary \( V \) consists of:
- \( IR \), a non-empty set of resources, called the domain of \( I \).
- A non empty set \( IDP \), called the property domain of \( I \).
- A mapping \( IP : IDP \rightarrow [0, 1] \) (fuzzy the set of properties of \( I \)).
- \( IEXT : IP \rightarrow (2^{IR \times IR} \rightarrow [0, 1]) \), given a property, given a subject and an object, returns a value in \([0, 1]\).
- \( IS \), a mapping from URI references in \( V \) into \( IR \cup IDP \).
- \( IL \), a mapping from typed literals in \( V \) into \( IR \).
- A distinguished subset \( LV \) of \( IR \), set of literal values, which contains all the plain literals in \( V \).

Satisfiability:
\[
I \models \langle (s, p, o), n \rangle \text{ iff } IP(I(p)) \land IEXT(I(p))(I(s), I(o)) \geq n
\]

For instance, using Gödel t-norm \( x \land y = \min(x, y) \), if
\[
\begin{align*}
I(o1) &= s \\
I(IsAbout) &= p \\
I(snoopy) &= o \\
IP(p) &= 0.9 \\
IEXT(p)(s, o) &= 0.85
\end{align*}
\]
then
\[
I \models \langle (o1, IsAbout, snoopy), 0.8 \rangle \text{ as } \\
\min(IP(p), IEXT(p)(s, o)) = \min(0.9, 0.85) = 0.85 \geq 0.8
\]
In fuzzy RDFS, class extensions are fuzzy sets of domain's elements.

Class interpretation $ICEXT$ is induced by $IEXT(I(type))$

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

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

Fuzzy RDFS adds extra constraints on interpretations, such as

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

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

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

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

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

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

\[
\frac{\langle (a, sp, b), n \rangle, \langle (x, a, y), m \rangle}{\langle (x, b, y), n \land m \rangle}
\]

\[
\frac{\langle (a, sc, b), n \rangle, \langle (x, type, a), m \rangle}{\langle (x, type, b), n \land m \rangle}
\]

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

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

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

- Fuzzy RDF representation

\[
\langle (o1, \text{IsAbout}, \text{snoopy}), 0.8 \rangle \\
\langle (\text{snoopy}, \text{type}, \text{dog}), 1.0 \rangle \\
\langle (\text{woodstock}, \text{type}, \text{bird}), 1.0 \rangle \\
\langle (\text{dog}, \text{subClassOf}, \text{Animal}), 1.0 \rangle \\
\langle (\text{bird}, \text{subClassOf}, \text{Animal}), 1.0 \rangle
\]

- then

\[
KB \models \langle \exists x. (o1, \text{IsAbout}, x) \land (x, \text{type}, \text{Animal}), 0.8 \rangle
\]
Fuzzy DLs Basics [26]

- In classical DLs, a concept $C$ is interpreted by an interpretation $\mathcal{I}$ as a set of individuals.
- In fuzzy DLs, a concept $C$ is interpreted by $\mathcal{I}$ as a fuzzy set of individuals.
- Each individual is instance of a concept to a degree in $[0, 1]$.
- Each pair of individuals is instance of a role to a degree in $[0, 1]$.
The semantics is an immediate consequence of the First-Order-Logic translation of DLs expressions

\[ I^I = \Delta^I \]
\[ C^I : \Delta^I \rightarrow [0, 1] \]
\[ R^I : \Delta^I \times \Delta^I \rightarrow [0, 1] \]

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>( C, D )</td>
<td>( \top )</td>
</tr>
<tr>
<td></td>
<td>( \bot )</td>
</tr>
<tr>
<td></td>
<td>( A )</td>
</tr>
<tr>
<td>( C \sqcap D )</td>
<td>( (C_1 \sqcap C_2)^I(x) = C_1^I(x) \land C_2^I(x) )</td>
</tr>
<tr>
<td>( C \sqcup D )</td>
<td>( (C_1 \sqcup C_2)^I(x) = C_1^I(x) \lor C_2^I(x) )</td>
</tr>
<tr>
<td>( \neg C )</td>
<td>( \neg C^I(x) )</td>
</tr>
<tr>
<td>( \exists R.C )</td>
<td>( \exists R.C^I(x) = \sup_{y \in \Delta^I} R^I(x, y) \land C^I(y) )</td>
</tr>
<tr>
<td>( \forall R.C )</td>
<td>( \forall R.C^I(u) = \inf_{y \in \Delta^I} R^I(x, y) \rightarrow C^I(y) )</td>
</tr>
</tbody>
</table>

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 $KB$ consistent, i.e. satisfiable?

Subsumption: structure knowledge, compute taxonomy
- $KB \models C \sqsubseteq D$?

Equivalence: check if two fuzzy concepts are the same
- $KB \models C = D$?

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

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

Top-k retrieval: Retrieve the top-k individuals that instantiate $C$ w.r.t. best truth value bound
- $ans_{top-k}(KB, C) = Top_k \{ \langle a, v \rangle \mid v = |a:C|_{KB} \}$
Some Notes on . . .

- **Value restrictions:**
  - In classical DLs, $\forall R. C \equiv \neg \exists R. \neg C$
  - The same is not true, in general, in fuzzy DLs (depends on the operators’ semantics, true for Łukasiewicz, but not true in Gödel logic)
  - Is it acceptable that $\forall hasParent.Human \not\equiv \neg \exists hasParent.\neg Human$? Recall that in Ł and Zadeh, $\forall x. \phi \equiv \neg \exists x. \neg \phi$

- **Models:**
  - In classical DLs $\top \subseteq \neg(\forall A) \cap (\neg \exists R. \neg A)$ has no classical model
  - In Gödel logic it has no finite model, but has an infinite model

- **The choice** of the appropriate semantics of the logical connectives is **important**.
  - Should have reasonable logical properties
  - Certainly it must have efficient algorithms solving basic inference problems

- Łukasiewicz Logic seems the best compromise, though Zadeh semantics has been considered historically in DLs (we recall that Zadeh semantics is not considered by fuzzy logicians)

- For disjointness it is better to use $C \cap D \subseteq \bot$ rather than $C \subseteq \neg D$
  - they are not the same, e.g. $A \subseteq \neg A$ says that $A^\mathcal{I}(x) \leq 0.5$ holds, for all $\mathcal{I}$ and for all $x \in \Delta^\mathcal{I}$
Towards fuzzy OWL Lite and OWL DL

- Recall that OWL Lite and OWL DL relate to $SHIF(D)$ and $SHOIN(D)$, respectively.
- We need to extend the semantics of fuzzy $ALC$ to fuzzy $SHOIN(D) = ALCCHOINR_+(D)$.
- Additionally, we add:
  - modifiers (e.g., very)
  - concrete fuzzy concepts (e.g., Young)
  - both additions have explicit membership functions.
Number Restrictions, Inverse and Transitive roles

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

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

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

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

- Note: \((\geq 1 \ R) \equiv \exists R. \top\)

- For inverse roles we have for all \(x, y \in \Delta^I\)

\[
R^I(x, y) = R^I(y, x)
\]

- For transitive roles \(R\) we impose: for all \(x, y \in \Delta^I\)

\[
R^I(x, y) \geq \sup_{z \in \Delta^I} \min(R^I(x, z), R^I(z, y))
\]
Concrete fuzzy concepts

- E.g., Small, Young, High, etc. with explicit membership function

- Use the idea of concrete domains:
  - \( D = \langle \Delta_D, \Phi_D \rangle \)
  - \( \Delta_D \) 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 \( d^D : \Delta_D^n \to [0, 1] \)
  - 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
  - \( \text{very}(x) = x^2 \)
  - \( \text{slightly}(x) = \sqrt{x} \)

- For instance,

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

#### 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>$\exists y_1, \ldots, y_n \cdot \bigwedge_{i=1}^n R(x, y_i) \land \bigwedge_{1 \leq i &lt; j \leq n} y_i \neq y_j$</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>$\text{FCC}$</td>
<td>$\mu_{\text{FCC}}(x)$</td>
</tr>
<tr>
<td>$\text{M(C)}$</td>
<td>$\mu_{\text{M}}(C(x))$</td>
</tr>
<tr>
<td>$R$</td>
<td>$P(x, y)$</td>
</tr>
<tr>
<td>$\neg P$</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>$\text{fun}(R)$</td>
<td>$\forall x , r \rightarrow (C(x) \rightarrow D(x))$, where $\rightarrow$ is $r$-implication</td>
</tr>
<tr>
<td>$\text{trans}(R)$</td>
<td>$\forall x \forall y \forall z , R(x, y) \land R(x, z) \rightarrow y = z$</td>
</tr>
<tr>
<td>$\exists z , R(x, z) \land R(z, y)$</td>
<td>$R(x, y)$</td>
</tr>
</tbody>
</table>
Example (Graded Entailment)

<table>
<thead>
<tr>
<th></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>

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

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

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

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

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

Note: without an explicit membership function of Young, this inference cannot be drawn.
Example (Simplified Negotiation)

- A car seller sells an Audi TT for 31500 €, as from the catalog price.
- A buyer is looking for a sports-car, but wants to 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 €.
  - $AudiTT = \text{SportsCar} \sqcap \exists \text{hasPrice}.R(x; 30500, 31500)$
  - $Query = \text{SportsCar} \sqcap \exists \text{hasPrice}.L(x; 30000, 32000)$
- Highest degree to which the concept $C = AudiTT \sqcap Query$ is satisfiable is 0.75 (the possibility that the Audi TT and the query matches is 0.75).
- The car may be sold at 31250 €.
Reasoning [19, 17, 18]

Depends on the semantics and reasoning method (tableau-based or MILP-based)

**Tableaux method**: under Zadeh semantics
- a tableau exists for fuzzy $SHIN$, solving the satisfiability problem
- classical blocking methods apply similarly in the fuzzy variant
- the management of General concept inclusions (GCI's) is more complicated compared to the crisp case
- a translation of fuzzy $SHOIN^C$ to crisp $SHOIN$ also exists (not addressed here)
- the tableaux method is **not suitable** to deal with fuzzy concrete concepts and modifiers
- the BTVB can be solved, but not efficiently

**MILP based method**: under Zadeh semantics, Łukasiewicz semantics, and classical semantics
- **exists** for fuzzy $ALC + \text{linear modifiers + fuzzy concrete concepts}$ [20, 21, 2]
- **exists** for fuzzy $SHIF + \text{linear modifiers + fuzzy concrete concepts}$ (implemented in fuzzyDL reasoner, but not published yet [1, 2])
- solves the BTVB as primary problem

**MIQP based method**: using Mixed Integer Quadratically Constrained Programming optimization problem (MICQP) for product T-norm
- **exists** for fuzzy $SHIF + \text{linear modifiers + fuzzy concrete concepts}$ (implemented in fuzzyDL reasoner, but not published yet [1]). Important as it simulates probabilistic reasoning under independent event assumption.
- solves the BTVB as primary problem
- the fuzzyDL solver also allows to mix all three semantics
Fuzzy tableaux-based method

- Tableau algorithm is similar to classical DL tableaux.
- Most problems can be reduced to satisfiability problem, e.g.
- Assertions are extended to $\langle a: C \geq n \rangle$, $\langle a: C \leq n \rangle$, $\langle a: C > n \rangle$ and $\langle a: C < n \rangle$.
- $KB \models \langle a: C, n \rangle$ iff $KB \cup \{ \langle a: C < n \rangle \}$ not satisfiable.
  - All models of $KB$ do not satisfy $\langle a: C < n \rangle$, i.e. do satisfy $\langle a: C \geq n \rangle$.
- Let’s see a tableaux algorithm for satisfiability checking, where

  $x \land y = \min(x, y)$

  $x \lor y = \max(x, y)$

  $\neg x = 1 - x$

  $x \rightarrow y = \max(1 - x, y)$
Tableaux for $\mathcal{ALC} \text{ KB}$

- Works on a tree forest (semantics through viewing tree as an ABox)
  - Nodes represent elements of $\Delta^I$, labelled with sub-concepts of $C$ and their weights
  - Edges represent role-successorships between elements of $\Delta^I$ and their weights
- Works on concepts in negation normal form: push negation inside using de Morgan' laws and

\[
\neg(\exists R. C) \iff \forall R. \neg C \\
\neg(\forall R. C) \iff \exists R. \neg C
\]

- It is initialised with a tree forest consisting of root nodes $a$, for all individuals appearing in the KB:
  - If $\langle a: C \triangleright n \rangle \in KB$ then $\langle C, \triangleright, n \rangle \in \mathcal{L}(a)$
  - If $\langle (a, b): R \triangleright n \rangle \in KB$ then $\langle (a, b), \triangleright, n \rangle \in \mathcal{E}(R)$
- A tree forest $T$ contains a clash if for a tree $T$ in the forest there is a node $x$ in $T$, containing a conjugated pair $\{\langle A, \triangleright, n \rangle, \langle C, \triangleleft, m \rangle\} \subseteq \mathcal{L}(x)$, e.g. $\langle A, \geq, 0.6 \rangle, \langle A, <, 0.3 \rangle$
- Returns “$KB$ is satisfiable” if rules can be applied s.t. they yield a clash-free, complete (no more rules apply) tree forest
### ALC Tableau rules (excerpt)

<table>
<thead>
<tr>
<th>Rule</th>
<th>Left-hand Side</th>
<th>Right-hand Side</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x \bullet { \langle C_1 \sqcap C_2, \geq, n \rangle, \ldots } )</td>
<td>( \rightarrow \sqcap )</td>
<td>( x \bullet { \langle C_1 \sqcap C_2, \geq, n \rangle, \langle C_1, \geq, n \rangle, \langle C_2, \geq, n \rangle, \ldots } )</td>
</tr>
<tr>
<td>( x \bullet { \langle C_1 \sqcup C_2, \geq, n \rangle, \ldots } )</td>
<td>( \rightarrow \sqcup )</td>
<td>( x \bullet { \langle C_1 \sqcup C_2, \geq, n \rangle, \langle C, \geq, n \rangle, \ldots } ) ( \text{for } C \in { C_1, C_2 } )</td>
</tr>
<tr>
<td>( x \bullet { \langle \exists R.C, \geq, n \rangle, \ldots } )</td>
<td>( \rightarrow \exists )</td>
<td>( x \bullet { \langle \exists R.C, \geq, n \rangle, \ldots } ) ( \langle R, \geq, n \rangle \downarrow y \bullet { \langle C, \geq, n \rangle } )</td>
</tr>
<tr>
<td>( \langle R, \geq, m \rangle \downarrow (m &gt; 1 - n) ) ( y \bullet { \ldots } )</td>
<td>( \rightarrow \forall )</td>
<td>( \langle R, \geq, m \rangle \downarrow y \bullet { \langle \forall R.C, \geq, n \rangle, \ldots } ) ( \langle R, \geq, m \rangle \downarrow y \bullet { \langle \forall \alpha, \geq, n \rangle, \ldots } ) ( \langle C, \geq, n \rangle )</td>
</tr>
<tr>
<td>( x \bullet { C \sqsubseteq D, \ldots } )</td>
<td>( \rightarrow \sqsubseteq )</td>
<td>( x \bullet { C \sqsubseteq D, E, \ldots } ) ( \text{for } E \in { \langle C, &lt;, n \rangle, \langle D, \geq, n \rangle }, n \in N^A )</td>
</tr>
</tbody>
</table>

\[ KB = \langle T, A \rangle \]
\[ X^A = \{ 0, 0.5, 1 \} \cup \{ n \mid \langle \alpha \bowtie n \rangle \in A \} \]
\[ N^A = X^A \cup \{ 1 - n \mid n \in X^A \} \]
Theorem

Let $KB$ be an $\mathcal{ALC}$ $KB$ and $F$ obtained by applying the tableau rules to $KB$. Then

1. The rule application terminates,
2. If $F$ is clash-free and complete, then $F$ defines a (canonical) (tree forest) model for $KB$, and
3. If $KB$ has a model $\mathcal{I}$, then the rules can be applied such that they yield a clash-free and complete forest $F$.

The tableau can be modified to a decision procedure for

- $SHIN \ (\equiv \mathcal{ALCHINR}_+)$
- $SHOIQ \ (\equiv \mathcal{ALCHOIQR}_+)$ (expected)
Problem with fuzzy tableau

- Usual fuzzy tableaux calculus does not work anymore with
  - modifiers and concrete fuzzy concepts
  - Łukasiewicz Logic
  - Product T-norm

- Usual fuzzy tableaux calculus does not solve the BTVB problem

- New algorithm uses bounded Mixed Integer Programming oracle, as for Many Valued Logics
  - Recall: the general MILP problem is to find

\[
\bar{x} \in \mathbb{Q}^k, \quad \bar{y} \in \mathbb{Z}^m,
\]
\[
f(\bar{x}, \bar{y}) = \min \{ f(x, y) : Ax + By \geq h \}
\]
\[
A, B \text{ integer matrixes}
\]
Requirements

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

\[ \text{Minor} = \text{Person} \sqcap \exists \text{hasAge.} \leq 18 \]
\[ \text{YoungPerson} = \text{Person} \sqcap \exists \text{hasAge. Young} \]
\[ \text{Young} = \text{ls}(10, 30) \]
\[ \leq 18 = \text{cr}(0, 18) \]

Then

\[ |a:C|_KB = \min\{x \mid KB \cup \{\langle a:C \leq x \rangle \text{ satisfiable} \} \] 
\[ |C \sqsubseteq D|_KB = \min\{x \mid KB \cup \{\langle a:C \sqcap \neg D \geq 1 - x \rangle \text{ satisfiable} \} \] 

Apply (deterministic) tableaux calculus, then use bounded Mixed Integer Programming oracle.
### ALC MILP Tableau rules under Zadeh semantics
(excerpt)

<table>
<thead>
<tr>
<th>Rule 1</th>
<th>Rule 2</th>
<th>Rule 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x \bullet { (C_1 \sqcap C_2, \geq, l), \ldots } \rightarrow \sqcap ) ( x \bullet { (C_1 \sqcap C_2, \geq, l), (C_1, \geq, l), (C_2, \geq, l), \ldots } )</td>
<td>( x \bullet { (C_1 \sqcup C_2, \geq, l), (C_1, \geq, x_1), (C_2, \geq, x_2), x_1 + x_2 = l, x_1 \leq y, x_2 \leq 1 - y, x_i \in [0, 1], y \in {0, 1}, \ldots } \rightarrow \Box ) ( x \bullet { (C_1 \sqcup C_2, \geq, l), (C_1, \geq, x), (C_2, \geq, x), x_1 + y \geq l, x \leq y, l_1 + l_2 \leq 2 - y, x \in [0, 1], y \in {0, 1} } \rightarrow \exists ) ( x \bullet { (\exists R. C, \geq, l), \ldots } \rightarrow \exists ) ( x \bullet { (\exists R. C, \geq, l_1), \ldots } \rightarrow \forall ) ( x \bullet { (\forall R. C, \geq, l_1), \ldots } \rightarrow \forall ) ( x \bullet { (\forall R. C, \geq, l_1), \ldots } \rightarrow \forall ) ( x \bullet { (\forall R. C, \geq, l_1), \ldots } \rightarrow \forall ) ( x \bullet { (\forall R. C, \geq, l_1), \ldots } \rightarrow \forall ) ( x \bullet { (\forall R. C, \geq, l_1), \ldots } \rightarrow \forall ) ( x \bullet { (\forall R. C, \geq, l_1), \ldots } \rightarrow \forall ) ( x \bullet { (\forall R. C, \geq, l_1), \ldots } \rightarrow \forall ) ( x \bullet { (\forall R. C, \geq, l_1), \ldots } \rightarrow \forall ) ( x \bullet { (\forall R. C, \geq, l_1), \ldots } \rightarrow \forall ) ( x \bullet { (\forall R. C, \geq, l_1), \ldots } \rightarrow \forall ) ( x \bullet { (\forall R. C, \geq, l_1), \ldots } \rightarrow \forall ) ( x \bullet { (\forall R. C, \geq, l_1), \ldots } \rightarrow \forall )</td>
<td>( x \bullet { (C, \sqsubseteq, l), \ldots } \rightarrow \sqsubseteq_1 ) ( x \bullet { (C, \sqsubseteq, l), \ldots } \rightarrow \sqsubseteq_2 ) ( x \bullet { (C, \sqsubseteq, l), \ldots } \rightarrow \sqsubseteq ) ( x \bullet { (C, \sqsubseteq, l), \ldots } \rightarrow \sqsubseteq ) ( x \bullet { (C, \sqsubseteq, l), \ldots } \rightarrow \sqsubseteq )</td>
</tr>
</tbody>
</table>
Example

Suppose

\[ KB = \left\{ A \cap B \sqsubseteq C, \langle a : A \geq 0.3 \rangle, \langle a : B \geq 0.4 \rangle \right\} \]

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

<table>
<thead>
<tr>
<th>Step</th>
<th>Tree</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>( a \bullet {\langle A, \geq, 0.3 \rangle, \langle B, \geq, 0.4 \rangle, \langle C, \leq, x \rangle} )</td>
</tr>
<tr>
<td>2.</td>
<td>( \cup{\langle A \cap B, \leq, x \rangle} )</td>
</tr>
<tr>
<td>3.</td>
<td>( \cup{\langle A, \leq, x_1 \rangle, \langle B, \leq, x_2 \rangle} )</td>
</tr>
<tr>
<td></td>
<td>( \cup{x = x_1 + x_2 - 1, 1 - y \leq x_1, y \leq x_2} )</td>
</tr>
<tr>
<td>4.</td>
<td>find \text{min}{x \mid \langle a : A \geq 0.3 \rangle, \langle a : B \geq 0.4 \rangle, \langle a : C \leq x \rangle, \langle a : A \leq x_1 \rangle, \langle a : B \leq x_2 \rangle, x = x_1 + x_2 - 1, 1 - y \leq x_1, y \leq x_2, x_i \in [0, 1], y \in {0, 1}} )</td>
</tr>
<tr>
<td>5.</td>
<td>MILP oracle: ( x = 0.3 )</td>
</tr>
</tbody>
</table>
Implementation issues

Several options exists:

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

A theorem prover for fuzzy $SHIF$ + linear hedges + concrete fuzzy concepts + linear equational constraints, 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: $C \sqsubseteq C'\prime$ (inclusion axiom)

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

DL-Lite$_{\text{core}}$:
- Concepts:
  - $C \rightarrow A \mid \exists R$
  - $C'\prime \rightarrow A \mid \exists R \mid \neg A \mid \neg \exists R$
  - $R \rightarrow P \mid P'$
- Assertion: $a:A, (a, b):P$

DLR-Lite$_{\text{core}}$: ($n$-ary roles)
- Concepts:
  - $C \rightarrow A \mid \exists P[i]$
  - $C'\prime \rightarrow A \mid \exists P[i] \mid \neg A \mid \neg \exists P[i]$
  - $\exists P[i]$ 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: $q(x) \leftarrow \exists y.\text{conj}(x, y)$
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 SW Languages

Vagueness basics
Vagueness and RDF/DLs
Vagueness and LPs/DLPs

Uncertainty and Vagueness in the Semantic Web Tutorial at ESWC-2007 T. Lukasiewicz and U. Straccia
Examples:

- **isa**
  - `CatalogueBook ⊑ Book`

- **disjointness**
  - `Book ⊑ ¬Author`

- **constraints**
  - `CatalogueBook ⊑ ∃positioned_In`

- **role – typing**
  - `∃positioned_In ⊑ Container`

- **functional**
  - `fun(positioned_In)`

- **assertion**
  - `Author ⊑ ∃written_By`
  - `∃written_By ⊑ CatalogueBook`

- **query**
  - `q(x, y) ← CataloguedBook(x), Ordered_to(x, y)`

- **Consistency check** is linear time in the size of the KB

- **Query answering** in linear in the size of the number of assertions
Top-\(k\) retrieval in DL-Lite/DLR-Lite

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

\[
q(\mathbf{x}, s) \leftarrow \exists \mathbf{y}. \text{conj}(\mathbf{x}, \mathbf{y}), s = f(p_1(\mathbf{z}_1), \ldots, p_n(\mathbf{z}_n))
\]

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

\[
\begin{align*}
\text{Hotel} & \sqsubseteq \exists \text{HasHLoc} \\
\text{Hotel} & \sqsubseteq \exists \text{HasHPrice} \\
\text{Conference} & \sqsubseteq \exists \text{HasCLoc} \\
\text{Hotel} & \sqsubseteq \neg \text{Conference}
\end{align*}
\]

\[
\begin{array}{|c|c|}
\hline
\text{HasHLoc} & \text{HasCLoc} \\
\text{HotelID} & \text{ConflID} \\
\hline
h1 & c1 \\
h2 & c2 \\
\vdots & \vdots \\
\hline
\end{array}
\quad
\begin{array}{|c|c|}
\hline
\text{HasHPrice} & \\
\text{HotelID} & \text{Price} \\
\hline
h1 & 150 \\
h2 & 200 \\
\vdots & \vdots \\
\hline
\end{array}
\]

\[
q(h, s) \leftarrow \text{HasHLoc}(h, hl), \text{HasHPrice}(h, p), \text{Distance}(hl, cl, d), \text{HasCLoc}(c1, cl), \text{cheap}(p) \cdot \text{close}(d).
\]

where the fuzzy predicates \(\text{cheap}\) and \(\text{close}\) are defined as:

\[
\begin{align*}
\text{close}(d) &= ls(0, 2\text{km}, d) \\
\text{cheap}(p) &= ls(0, 300, p)
\end{align*}
\]
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 \( I \) as the set

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

  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 \( I \).

- Model of a query: \( I \models q(c, v) \) iff \( \langle c, v \rangle \in q^I \)
- Entailment: \( KB \models q(c, v) \) iff \( I \models KB \) implies \( I \models q(c, v) \)
- Top-\( k \) retrieval: \( \text{ans}_{top-k}(KB, q) = \text{Top}_k \{ \langle c, v \rangle \mid KB \models q(c, v) \} \)
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
      
      \[
      q(x, s) \leftarrow B(x), A(x), s = f(x) \\
      B_1 \sqsubseteq A \\
      B_2 \sqsubseteq A \\
      \hline \\
      q(x, s) \leftarrow B(x), B_1(x), s = f(x) \\
      q(x, s) \leftarrow B(x), B_2(x), s = f(x)
      \]

  3. The reformulated queries in \( r(q, T) \) are evaluated over \( \mathcal{A} \) (seen as a database) using standard top-k techniques for DBs
    - for all \( q_i \in r(q, T) \), \( \text{ans}_{top-k}(q_i, \mathcal{A}) = \text{top-k SQL query over } \mathcal{A} \text{ database} \)
    - \( \text{ans}_{top-k}(KB, q) = \text{Top}_k(\bigcup_{q_i \in r(q, T)} \text{ans}_k(q_i, \mathcal{A})) \)
Small Example:

<table>
<thead>
<tr>
<th>$P_2$</th>
<th>$B$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
</tr>
</tbody>
</table>

$T = \{ \exists P_2 \sqsubseteq A, A \sqsubseteq \exists P_1, B \sqsubseteq \exists P_2 \}$

$q(x, s) \leftarrow P_2(x, y), P_1(y, z), s = \max(0, 1 - x/10)$
$q(x, s) \leftarrow P_2(x, y), A(y), s = \max(0, 1 - x/10)$
$q(x, s) \leftarrow P_2(x, y), P_2(z, y), s = \max(0, 1 - x/10)$
$q(x, s) \leftarrow P_2(x, y), s = \max(0, 1 - x/10)$
$q(x, s) \leftarrow B(x), s = \max(0, 1 - x/10)$
$q_1(x, s) \leftarrow P_2(x, y), s = \max(0, 1 - x/10)$
$q_2(x, s) \leftarrow B(x), s = \max(0, 1 - x/10)$

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

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

**Proposition**

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

**Tool exists and implemented in the DLMedia system**

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

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

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

  $Rl \longrightarrow A \mid \exists[i_1, \ldots, i_k]R \mid \exists[i_1, \ldots, i_k]R. (\text{Cond}_1 \sqcap \ldots \sqcap \text{Cond}_i)$

  $\text{Cond} \longrightarrow (\{i\} \leq v) \mid (\{i\} < v) \mid (\{i\} \geq v) \mid (\{i\} > v) \mid (\{i\} = v) \mid (\{i\} \neq v) \mid$

  $(\{i\} \text{ simTxt} k_1, \ldots, k_n) \mid (\{i\} \text{ simImg URN})$

- $\exists[i_1, \ldots, i_k]R$ is the projection of the relation $R$ on the columns $i_1, \ldots, i_k$
- $\exists[i_1, \ldots, i_k]R. (\text{Cond}_1 \sqcap \ldots \sqcap \text{Cond}_i)$ further restricts the projection $\exists[i_1, \ldots, i_k]R$ according to the conditions specified in $\text{Cond}_i$
- $(\{i\} \text{ simTxt} k_1 \ldots k_n)$ evaluates the degree of being the text of the $i$-th column similar to the list of keywords $k_1 \ldots k_n$
- $(\{i\} \text{ simImg URN})$ 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

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

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

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

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

On the other hand examples axioms involving similarity predicates are,

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

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

where \text{urn1} identifies the image
Example queries

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

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

\[ q(x) \leftarrow ImageDescr(x, y) \land (y \simImg urn2) \]
   // 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
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 SW Languages

Interface:
Run:

Top-32 results found

Score: 1.0  Score: 1.0  Score: 1.0  Score: 1.0  Score: 0.93240685
Show Info  Show Info  Show Info  Show Info  Show Info

Score: 0.8159055  Score: 0.9819412  Score: 0.9992125  Score: 0.9988917  Score: 0.9982548
Show Info  Show Info  Show Info  Show Info  Show Info

Score: 0.88869231  Score: 0.88644457  Score: 0.8827538  Score: 0.8812844  Score: 0.8812283
Show Info  Show Info  Show Info  Show Info  Show Info
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

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

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 \( p_i(c_i) \).
Example

<table>
<thead>
<tr>
<th>ID</th>
<th>HOTEL</th>
<th>PRICE Single</th>
<th>PRICE Double</th>
<th>DISTANCE</th>
<th>s</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Verdi</td>
<td>100</td>
<td>120</td>
<td>5Min</td>
<td>0.75</td>
</tr>
<tr>
<td>2</td>
<td>Puccini</td>
<td>120</td>
<td>135</td>
<td>10Min</td>
<td>0.5</td>
</tr>
<tr>
<td>3</td>
<td>Rossini</td>
<td>80</td>
<td>90</td>
<td>15Min</td>
<td>0.25</td>
</tr>
</tbody>
</table>

\[ R(x_1, x_2) \leftarrow CloseHotel(x_1, x_2, x_3, x_4, x_5) \cdot cheap(x_3) , \]

where

\[ cheap(p) = ls(0, 250, p) . \]
Example

Car buying example:

\[\text{Pref}_1(x, p, s) \leftarrow \text{hasPrice}(x, p),\]
\[LS(0, 100000, 11000, 13000, p, s)\]
\[\text{Pref}_2(x, s) \leftarrow \text{Kilometers}(x, k),\]
\[LS(0, 400000, 15000, 20000, k, s)\]
\[\text{Buyer}(x, p, u) \leftarrow \text{Pref}_1(x, p, s_1), \text{Pref}_2(x, s_2),\]
\[u = 0.75 \cdot s_1 + 0.25 \cdot s_2\]
Semantics of fuzzy LPs

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

- **Least model** exists and is **least fixed-point** of
  
  \[ T_\mathcal{P}(I)(A) = I(\varphi) \]
  
  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 \( \omega \) \( T_\mathcal{P} \) iterations)!
General top-down query procedure for Many-valued LPs

**Idea:** use theory of fixed-point computation of equational systems over truth space (complete lattice or complete partial order)

- Assign a variable $x_i$ to an atom $A_i \in B_P$
- Map a rule $A \leftarrow f(A_1, \ldots, A_n) \in P^*$ into the equation $x_A = f(x_{A_1}, \ldots, x_{A_n})$
- A LP $P$ is thus mapped into the equational system

$$\begin{align*}
  x_1 &= f_1(x_{1_1}, \ldots, x_{1_{a_1}}) \\
  & \vdots \\
  x_n &= f_n(x_{n_1}, \ldots, x_{n_{a_n}})
\end{align*}$$

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

$$\begin{align*}
  y_0 &= 0 \\
  y_{i+1} &= f(y_i)
\end{align*}$$

where $f = \langle f_1, \ldots, f_n \rangle$ and $f(x) = \langle f_1(x_1), \ldots, f_n(x_n) \rangle$

- The least-fixed point is the least model of $P$
- **Consequence:** If top-down procedure exists for equational systems then it works for fuzzy LPs too!
Following [22, 23] . . .

**Procedure** $\text{Solve}(S, Q)$

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

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

1. $A := Q$, $dg := Q$, $in := \emptyset$, for all $x \in V$ do $v(x) = 0$, $exp(x) = 0$
2. while $A \neq \emptyset$ do
3. select $x_i \in A$, $A := A \setminus \{x_i\}$, $dg := dg \cup s(x_i)$
4. $r := f_i(v(x_{i1}), ..., v(x_{ia}))$
5. if $r \succ v(x_i)$ then $v(x_i) := r$, $A := A \cup (p(x_i) \cap dg)$ fi
6. if not $exp(x_i)$ then $exp(x_i) = 1$, $A := A \cup (s(x_i) \setminus in)$, $in := in \cup s(x_i)$ fi
od

For $q(\mathbf{x}) \leftarrow \phi \in \mathcal{P}$, with $s(q)$ we denote the set of **sons** of $q$ w.r.t. $r$, i.e. the set of intentional predicate symbols occurring in $\phi$. With $p(q)$ we denote the set of **parents** of $q$, i.e. the set $p(q) = \{p_i: q \in s(p_i, r)\}$ (the set of predicate symbols directly depending on $q$).
Set of facts \( \langle \text{Experience}(john), 0.7 \rangle, \langle \text{Risk}(john), 0.5 \rangle, \langle \text{Sport}_{\text{car}}(john), 0.8 \rangle \)

Set of rules, which after grounding are:

\[
\begin{align*}
\text{Good}_{\text{driver}}(john) & \quad \leftarrow \quad \text{Experience}(john) \land (0.5 \cdot \text{Risk}(john)) \\
\text{Risk}(john) & \quad \leftarrow \quad 0.8 \cdot \text{Young}(john) \\
\text{Risk}(john) & \quad \leftarrow \quad 0.8 \cdot \text{Sport}_{\text{car}}(john) \\
\text{Risk}(john) & \quad \leftarrow \quad \text{Experience}(john) \land (0.5 \cdot \text{Good}_{\text{driver}}(john))
\end{align*}
\]

1. \( A: = \{x_{R(j)}\}, x_i: = x_{R(j)}, A: = \emptyset, dg: = \{x_{R(j)}, x_{Y(j)}, x_{S(j)}, x_{E(j)}, x_{G(j)}\}, r: = 0.5, v(x_{R(j)}): = 0.5, A: = \{x_{G(j)}\}, \exp(x_{R(j)}): = 1, A: = \{x_{Y(j)}, x_{S(j)}, x_{E(j)}, x_{G(j)}\}, \text{in}: = \{x_{Y(j)}, x_{S(j)}, x_{E(j)}, x_{G(j)}\} \)
2. \( x_i: = x_{Y(j)}, A: = \{x_{S(j)}, x_{E(j)}, x_{G(j)}\}, r: = 0, \exp(x_{Y(j)}): = 1 \)
3. \( x_i: = x_{S(j)}, A: = \{x_{E(j)}, x_{G(j)}\}, r: = 0.8, v(x_{S(j)}): = 0.8, A: = \{x_{E(j)}, x_{G(j)}, x_{R(j)}\}, \exp(x_{S(j)}): = 1 \)
4. \( x_i: = x_{E(j)}, A: = \{x_{G(j)}, x_{R(j)}\}, r: = 0.7, v(x_{E(j)}): = 0.7, \exp(x_{E(j)}): = 1 \)
5. \( x_i: = x_{G(j)}, A: = \{x_{R(j)}\}, r: = 0.25, v(x_{G(j)}): = 0.25, \exp(x_{G(j)}): = 1, \text{in}: = \{x_{Y(j)}, x_{S(j)}, x_{E(j)}, x_{G(j)}, x_{R(j)}\} \)
6. \( x_i: = x_{R(j)}, A: = \emptyset, r: = 0.64, v(x_{R(j)}): = 0.64, A: = \{x_{G(j)}\} \)
7. \( x_i: = x_{G(j)}, A: = \emptyset, r: = 0.32, v(x_{G(j)}): = 0.32, A: = \{x_{R(j)}\} \)
8. \( x_i: = x_{G(j)}, A: = \emptyset, r: = 0.64 \)
9. \( \text{stop. return } v \) (in particular, \( v(x_{R(j)}) = 0.64 \))
The top-down procedure can be extended to

- fuzzy Normal Logic Programs (Logic programs with non-monotone negation) [22]
- Many-valued Normal Logic Programs under Any-world Assumption [9, 28]
- Logic Programs, without requiring the grounding of the program

Other approaches for top-down methods for monotone fuzzy LPs: [6, 35, 7, 4]

Magics sets like methods: yet to investigate ...

There are also extensions to Fuzzy Disjunctive Logic Programs [10, 11, 24, 13, 14] with or without default negation
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]
- The procedure is an generalization of the \textit{Solve} procedure, integrating top-\(k\) database technology [8, 32]
- We do not determine all answers, but collect answers incrementally together and we can stop as soon as we have gathered \(k\) answers above a computed threshold
Procedure TopAnswers($\mathcal{K}$, $Q$, $k$)

Input: KB $\mathcal{K}$, intensional query relation symbol $Q$, $k \geq 1$;
Output: Mapping rankedList such that rankedList($Q$) contains top-k answers of $Q$

Init: $\delta = 1$, for all rules $r : P(x) \leftarrow \phi$ in $P$ do

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

---

**Vagueness basics**

**Vagueness and RDF/DLs**

**Vagueness and LPs/DLPs**

---

**Procedure getNextTuple(P, r)**

**Input:** intensional relation symbol $P$ and rule $r : P(x) \leftarrow \exists y.f(R_1(z_1), \ldots, R_n(z_l)) \in P$;

**Output:** Next tuple satisfying the body of the $r$ together with the score

**Init:**

1. Generate next new instance tuple $\langle t, s \rangle$ of $P$, using tuples in $\text{rankedList}(R_i)$ and $\text{RankSQL}$
2. if there is no $\langle t, s' \rangle \in \text{rankedList}(P, r)$ with $s \leq s'$ then exit loop
   until no new valid join tuple can be generated
3. return $\langle t, s \rangle$ if it exists else return NULL

---

**Uncertainty and Vagueness in the Semantic Web**

Tutorial at ESWC-2007

T. Lukasiewicz and U. Straccia
q(x) ← min(r_1(x, y), r_2(y, z))

<table>
<thead>
<tr>
<th>recId</th>
<th>r_1</th>
<th>r_2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>a b</td>
<td>m h</td>
</tr>
<tr>
<td>2</td>
<td>c d</td>
<td>m j</td>
</tr>
<tr>
<td>3</td>
<td>e f</td>
<td>f k</td>
</tr>
<tr>
<td>4</td>
<td>l m</td>
<td>m n</td>
</tr>
<tr>
<td>5</td>
<td>o p</td>
<td>p q</td>
</tr>
</tbody>
</table>

Uncertainty and Vagueness in the Semantic Web

TopAnswers

<table>
<thead>
<tr>
<th>Iter</th>
<th>p</th>
<th>Δ_r</th>
<th>rankedList(p)</th>
<th>δ</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>q</td>
<td>⟨e, k, 0.75⟩</td>
<td>⟨e, k, 0.75⟩</td>
<td>0.8</td>
</tr>
<tr>
<td>2</td>
<td>q</td>
<td>⟨l, h, 0.7⟩</td>
<td>⟨e, k, 0.75⟩, ⟨l, h, 0.7⟩</td>
<td>0.75</td>
</tr>
<tr>
<td>3</td>
<td>q</td>
<td>⟨l, j, 0.7⟩</td>
<td>⟨e, k, 0.75⟩, ⟨l, h, 0.7⟩, ⟨l, j, 0.7⟩</td>
<td>0.75</td>
</tr>
<tr>
<td>4</td>
<td>q</td>
<td>⟨l, n, 0.65⟩</td>
<td>⟨e, k, 0.75⟩, ⟨l, h, 0.7⟩, ⟨l, j, 0.7⟩, ⟨l, n, 0.65⟩</td>
<td>0.7</td>
</tr>
</tbody>
</table>

getNextTuple

<table>
<thead>
<tr>
<th>Iter</th>
<th>p_i</th>
<th>⟨t_i, s_i⟩</th>
<th>qos(p, r)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>r_1</td>
<td>r_1(1)</td>
<td>−</td>
</tr>
<tr>
<td></td>
<td>r_2</td>
<td>r_2(1)</td>
<td>−</td>
</tr>
<tr>
<td></td>
<td>r_1</td>
<td>r_1(2)</td>
<td>−</td>
</tr>
<tr>
<td></td>
<td>r_2</td>
<td>r_2(2)</td>
<td>−</td>
</tr>
<tr>
<td></td>
<td>r_1</td>
<td>r_1(3)</td>
<td>−</td>
</tr>
<tr>
<td></td>
<td>r_2</td>
<td>r_2(3)</td>
<td>⟨e, k, 0.75⟩</td>
</tr>
<tr>
<td>2</td>
<td>r_1</td>
<td>r_1(4)</td>
<td>⟨l, h, 0.7⟩, ⟨l, j, 0.7⟩</td>
</tr>
<tr>
<td>3</td>
<td>−</td>
<td>−</td>
<td>⟨l, j, 0.7⟩</td>
</tr>
<tr>
<td>4</td>
<td>−</td>
<td>−</td>
<td>⟨l, n, 0.65⟩</td>
</tr>
</tbody>
</table>
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

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

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

  \[
  \text{ChineseCarCompany} \sqsubseteq \exists \text{has_location.China}
  \]

- **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
Axiomatic approach

- Formally easy
  - $I$ is a model of $KB = \langle \mathcal{P}, \Sigma \rangle$ iff $I \models \mathcal{P}$ and $I \models \Sigma$

- To guarantee decidability, e.g.
  - DL-safe rules +
  - Fuzzy LP component has to be decidable

- Decision algorithm: No algorithm exists yet. Though
  - A mapping from fuzzy OWL-DL to fuzzy disjunctive LPs is possible
    - Depends on the semantics and features of the fuzzy DL component (t-norm, fuzzy concrete domains, . . .)
    - Depends on the semantics for the fuzzy disjunctive LP component (e.g., [10, 13, 14, 24])
    - The fuzzy LP semantics has to support the fuzzy DL component semantics
  - However, a tractable (data complexity) top-$k$ algorithm exists for fuzzy DLR-Lite + fuzzy LPs under the axiomatic approach (submitted)
Loosely coupled approach [10, 24, 31, 27]

- Fuzzy DL atoms and roles are **procedural attachments** (calls to a fuzzy DL theorem prover)
  - \( I \) is a **model** of \( KB = \langle \mathcal{P}, \Sigma \rangle \) iff \( I^\Sigma \models \mathcal{P} \)
  - \( I^\Sigma (A) = I(A) \) for all ground non-DL atoms \( A \)
  - \( I^\Sigma (DL[A](a)) = glb(\Sigma, a:A) \) for all ground DL atoms \( DL[A](a) \)
  - \( I^\Sigma (DL[R](a, b)) = glb(\Sigma, (a, b):R) \) for all ground DL roles \( DL[R](a, b) \)

- Minimal model property of fuzzy LPs and a fixed-point characterization:
  \[
  T_\mathcal{P}(I)(A) = I^\Sigma(\varphi), \text{ for } A \leftarrow \varphi \in \mathcal{P}^*
  \]

- An approach using non-monotone negation is described in [10]
A top-down procedure (without non-monotonicity)

Combine \textit{Solve}(S, Q) with a theorem prover for fuzzy DLs

- Modify Step 1. of algorithm \textit{Solve}(S, Q)
  - for all $x_{ij}$ DL-atoms $DL[A](a)$ (similarly for roles)
    - compute $\bar{x}_{ij} = glb(KB, a:A)$
    - set $\nu(x_{ij}) = \bar{x}_{ij}$, instead of $\nu(x_{ij}) = 0$

Essentially, for all DL-atoms $DL[A](a)$ we compute off-line $glb(KB, a:A)$ and add then the rule $A(a) \leftarrow glb(KB, a:A)$ to $\mathcal{P}$
Tightly coupled approach [11]

- DL atoms may appear anywhere in the rule
  \[ a_1 \lor \oplus_1 \cdots \lor \oplus_{j-1} a_j \leftarrow \otimes_0 b_1 \land \otimes_1 b_2 \land \otimes_2 \cdots \land \otimes_{k-1} b_k \geq v \]

- For instance,
  \[
  \text{query}(x) \leftarrow \otimes \ SportyCar(x) \land \otimes \ hasInvoice(x, y_1) \land \otimes \ hasHorsePower(x, y_2) \land \otimes \ LeqAbout22000(y_1) \land \otimes \ Around150(y_2) \geq 1 .
  \]
Consider $KB = \langle \mathcal{P}, \Sigma \rangle$

- interpretation $I : HB_\Phi \rightarrow [0, 1]$

  $I \models r$ iff

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

- $I \models \mathcal{P}$ iff $I \models r$ for all $r \in \mathcal{P}^*$

- $I \models \Sigma$ iff $\Sigma \cup \{ a = I(a) \mid a \in HB_\Phi \}$ is satisfiable

- $I \models KB$ iff $I \models \mathcal{P}$ and $I \models \Sigma$

The extension to non-monotone negation and a decision procedure is described in [11, 12]

- Requires a decision procedure for the fuzzy DL 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.

T. Lukasiewicz and U. Straccia.
Uncertainty and Vagueness in the Semantic Web Tutorial at ESWC-2007
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.  
INFSYS RESEARCH REPORT 1843-07-03, INSTITUT Für INFORMATIONSSYSTEME ARBEITSBEREICH WISSENSBASIERTES SYSTEME, Technische Universität Wien, 2007.

C. Mateis.  
Extending disjunctive logic programming by t-norms.  
C. Mateis.
Quantitative disjunctive logic programming: Semantics and computation.

M. Mazzieri.
A fuzzy rdf semantics to represent trust metadata.

M. Mazzieri and A. F. Dragoni.
A fuzzy semantics for semantic web languages.

G. Stoilos, G. Stamou, and J. Z. Pan.
D2.5.6: Fuzzy reasoning extensions.

General concept inclusions in fuzzy description logics.

U. Straccia.
Reasoning within fuzzy description logics.

U. Straccia.
Description logics with fuzzy concrete domains.

**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.
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 SW Languages

Vagueness basics

Vagueness and RDF/DLs

Vagueness and LPs/DLPs

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.

Uncertainty and Vagueness in the Semantic Web Tutorial at ESWC-2007 T. Lukasiewicz and U. Straccia
U. Straccia.
Towards vague query answering in logic programming for logic-based information retrieval.

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.
In *Proceedings of Workshop on Data Analysis (WDA 2005)*, 2005.

P. Vojtáš.
Fuzzy logic programming.
Outline

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 SW Languages
- 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 € 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 € 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) ∩ ...

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

MazdaMX5Miata: SportyCar ⊑ (∃ hasInvoice. 18883)
  ∩ (∃ hasHorsePower. 166) ∩ ...
MitsubishiEclipseSpyder: SportyCar ⊑ (∃ hasInvoice. 24029)
  ∩ (∃ hasHorsePower. 162) ∩ ...

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 €” and “has a power of around 150 HP” in the buyer’s request through the following concepts $C$ and $D$, respectively:

\[
C = \exists \text{hasInvoice}.\text{LeqAbout22000} \quad \text{and} \\
D = \exists \text{hasHorsePower}.\text{Around150HP},
\]

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

![Graphs showing membership functions for LeqAbout22000 and Around150HP](image-url)
The following fuzzy dl-rule encodes the buyer’s request “a sports car that costs at most about 22 000 € and that has a power of around 150 HP”.

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

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

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

Ontology alignment mapping rules:

\[
\begin{align*}
\text{SportsCar}(x) & \leftarrow \times \text{DL}[\text{SportyCar}](x) \land \times \text{sc}_{pos} \geq 0.9 \\
\text{hasPrice}(x) & \leftarrow \times \text{DL}[\text{hasInvoice}](x) \land \times \text{hi}_{pos} \geq 0.8 \\
\text{hasPower}(x) & \leftarrow \times \text{DL}[\text{hasHorsePower}](x) \land \times \text{hhp}_{pos} \geq 0.8 ,
\end{align*}
\]

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:

\[
\begin{align*}
\text{KB} \models_{\text{tight}} & \left( E[\text{query}(\text{MazdaMX5Miata})] \right)[0.21, 0.21] \\
\text{KB} \models_{\text{tight}} & \left( E[\text{query}(\text{MitsubishiEclipseSpyder})] \right)[0.19, 0.19].
\end{align*}
\]

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