Symbolic Artificial Intelligence

Natalia Díaz Rodríguez
Sept. 2018

ENSTA ParisTech and INRIA Flowers flowers.inria.fr http://asr.ensta-paristech.fr/
natalia.diaz@ensta-paristech.fr
IA301 Logique et IA - 3A - Master (2018/2019)
https://perso.telecom-paristech.fr/bloch/OptionIA/Logics-SymbolicAI.html
Dealing with uncertainty: Fuzzy (Description) Logics and Fuzzy Ontologies
Why Fuzzy Logic?

- Real life is not black or white
- Classical (crisp) logic: true/ false
- Fuzzy Logic: [0, 1]. Ex. blond, tall, cheap
- For automatic reasoning about uncertain, vague, incomplete or imprecise knowledge
- For near natural language expressions [2]
Fuzzy statements:

- involve context sensitive concepts with no exact definition, no binary decision/membership function:
  
  Ex. small, close, far, cheap, expensive, is about, similar to, warm, cold.
  
  Ex. Find me a good hotel close to the conference venue
  
  If a hotel is close to the leaning tower of Pisa, then it is a touristic hotel

- are true to some degree, taken from a truth space (usually [0, 1])
### Types of Logic

<table>
<thead>
<tr>
<th>Language</th>
<th><strong>Ontological Commitment</strong>&lt;sup&gt;1&lt;/sup&gt;</th>
<th><strong>Epistemological Commitm.</strong>&lt;sup&gt;2&lt;/sup&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Propositional Logic</td>
<td>Facts</td>
<td>True/False/Unknown</td>
</tr>
<tr>
<td>First-order Logic</td>
<td>Facts, objects, relations</td>
<td>True/False/Unknown</td>
</tr>
<tr>
<td>Temporal Logic</td>
<td>Facts, objects, relations, times</td>
<td>True/False/Unknown</td>
</tr>
<tr>
<td>Probability Theory</td>
<td>Facts</td>
<td>Degree of belief (0..1)</td>
</tr>
<tr>
<td>Fuzzy Logic</td>
<td>Degree of truth</td>
<td>Degree of belief (0..1)</td>
</tr>
</tbody>
</table>

<sup>1</sup>What exists?-facts?, objects?, time? beliefs? What exists in the world

<sup>2</sup>What states of knowledge? What an agent believes about facts. [U. Straccia]
Fuzzy Description Logics (DL)
Fuzzy Knowledge Base (FKB) or fuzzy ontology: a finite set of axioms that comprises a fuzzy ABox $A$ and a fuzzy TBox $T$ [3].

Fuzzy ABox: a finite set of fuzzy (concept or role) assertions

Fuzzy TBox: a finite set of fuzzy General Concept Inclusions (GCIs), with a min. fuzzy degree of subsumption.

Logical operators of conjunction, disjunction and complement are special cases of the three fuzzy operators:

1. A possibilistic product is a t-norm: $a \otimes b$, conjunction, $\land$
2. A possibilistic sum is a t-conorm: $a \oplus b$; disjunction, $\lor$
3. Fuzzy complement: $\neg c$

A fuzzy KB $K$ is consistent if there is a model of $K$ that satisfies each axiom in $K$. 
Fuzzy operators supported by *fuzzyDL*

<table>
<thead>
<tr>
<th>Operator</th>
<th>Łukasiewicz logic</th>
<th>Gödel logic</th>
<th>Zadeh logic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conjunction $\alpha \land \beta$</td>
<td>$\max(\alpha + \beta - 1, 0)$</td>
<td>$\min(\alpha, \beta)$</td>
<td>$\min(\alpha, \beta)$</td>
</tr>
<tr>
<td>Disjunction $\alpha \lor \beta$</td>
<td>$\min(\alpha + \beta, 1)$</td>
<td>$\max(\alpha, \beta)$</td>
<td>$\max(\alpha, \beta)$</td>
</tr>
</tbody>
</table>
| Negation $\neg \alpha$           | $1 - \alpha$                  | $\begin{cases} 
1 & \text{if } \alpha = 0 \\
0 & \text{otherwise} 
\end{cases}$ | $1 - \alpha$                     |
| Implication $\alpha \rightarrow \beta$ | $\min(1 - \alpha + \beta, 1)$ | $\begin{cases} 
1 & \text{if } \alpha \leq \beta \\
\beta & \text{otherwise} 
\end{cases}$ | $\max(1 - \alpha, \beta)$       |
<table>
<thead>
<tr>
<th>Reasoner</th>
<th>Fuzzy DL</th>
<th>Event Subscript.</th>
<th>SPARQL</th>
<th>Cardinality Restr.</th>
<th>Fuzzy Sets</th>
<th>Concept Modifier</th>
<th>Fuzzy Data Type</th>
<th>Defuzzification</th>
<th>Fuzzy Rule</th>
<th>Satisfiable. Degree</th>
</tr>
</thead>
<tbody>
<tr>
<td>FiRE [194, 193, 189]</td>
<td>$\mathcal{F} - S\text{HIN}$</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>GURDL [84]</td>
<td>$\mathcal{F} - A\text{LC}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>De-Lorean [29]</td>
<td>$\mathcal{F} - S\text{ROIQ}$</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>GERDS [85]</td>
<td>$\mathcal{F} - A\text{LC}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>fuzzyDL [30]</td>
<td>$\mathcal{F} - S\text{HIF(D)}$</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>YADLR [119]</td>
<td>SLG algorithm</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>Fuzzy OWL Plugin [Fuz, 31]</td>
<td>$S\text{ROIQ(D)}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FRESG [87]</td>
<td>$\mathcal{F} - A\text{LC(G)}$</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>SoftFacts</td>
<td>$\mathcal{F} - D\text{LR-lite}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
fuzzyDL answers queries by solving an MILP problem: minimising a linear function wrt a set of constraints (linear inequations in which rational and integer variables cannot occur); MILP problems will be bounded with rational variables ranging over a subset of $[0,1]$ and integer variables ranging over $\{0,1\}$.
(define-primitive-concept Tall *top*)

(instance fernando *top*1.0)

(instance umberto Tall 0.9)

(related fernando umberto isFriendOf 0.8)

\[^3^\text{*top* denotes the universal concept (similar to OWL2 class Thing. Tall is a fuzzy concept, isFriendOf a fuzzy relation. umberto and fernando are individuals)} [4]\]
Partitioning a domain with fuzzy membership functions
Fuzzy Membership Functions (in *fuzzyDL*[4])

a) Trapezoidal function; b) Triangular; c) Left-shoulder; d) Crisp interval e) Linear f) Right-shoulder
FuzzyDL Reasoning Services

- **KB consistency.** A fuzzy KB $\mathcal{K}$ is consistent if there is a model of $\mathcal{K}$ that satisfies each axiom in $\mathcal{K}$.
- **Concept satisfiability.** A fuzzy concept $c$ is $d$-satisfiable w.r.t. a fuzzy KB $\mathcal{K}$ if there exists a model of $\mathcal{K}$ where $c$ can have some instance with degree greater or equal than $d$, where $d$ is a degree of truth. In FuzzyDL, this task can also consider some particular individual $o$ instead of an arbitrary one.
- **Best satisfiability degree (BSD) of a fuzzy concept $c$ w.r.t. a fuzzy KB $\mathcal{K}$** is the maximal degree $d$ such that $c$ is $d$-satisfiable w.r.t. $\mathcal{K}$.
- **Minimal satisfiability degree (MSD) of a fuzzy concept $c$** is similar to the BSD but considering the minimal degree rather than the maximal one.
- **Concept subsumption.** $c_2$ D-subsumes $c_1$ w.r.t. a fuzzy KB $\mathcal{K}$ if in every model of $\mathcal{K}$, $c_1$ is included in $c_2$ with degree greater or equal than $d$. The degree of inclusion is computed using a fuzzy implication.
- **Entailment.** A fuzzy KB $\mathcal{K}$ entails an axiom if every model of $\mathcal{K}$ satisfies it. FuzzyDL computes entailments of assertions and GCLs.
- **Best Entailment Degree (BED) of a non-graded axiom with respect to a fuzzy KB $\mathcal{K}$** is the maximal degree $d$ such that the axiom is satisfied in every model of $\mathcal{K}$ with degree greater or equal than $d$.
- **Maximal Entailment Degree (MED) of a non-graded axiom** is similar to the BED but considering some model rather than any model.
- **Instance retrieval.** Given a concept $c$ and a fuzzy KB $\mathcal{K}$, the instance retrieval problem computes the individuals that belong to $c$ with a non-zero degree together with the minimal degree of membership in every model of $\mathcal{K}$.
- **Variable maximisation.** Given a fuzzy KB $\mathcal{K}$ and a variable $x$, maximise $x$ such that $\mathcal{K}$ is consistent.
- **Variable minimisation.** Given a fuzzy KB $\mathcal{K}$ and a variable $x$, minimise $x$ such that $\mathcal{K}$ is consistent.
- **Defuzzification.** Given a fuzzy KB $\mathcal{K}$, a concrete role $\tau$, a concept $c$, and an individual $o$, compute the BSD of $c$ for the individual $o$ and then defuzzify the value of $\tau$ for the individual $o$ using some defuzzification method: largest of maxima (LOM), smallest of maxima (SOM), or the middle of maxima (MOM).
- **Best Non-Fuzzy Performance (BNP).** Given a triangular fuzzy number $F = (\text{triangular } q_1, q_2, q_3)$, $\text{BNP}(F) = (q_1 + q_2 + q_3)/3$. This task is particularly useful when fuzzy numbers are arithmetically combined.
Query languages: SPARQL Query Example:

```
1 SELECT ?calendar1 ?phone2
2 WHERE { ?user0 a ha:User.
3   ?user0 ha:hasName "Natalia"^^xsd:string.
4   ?user0 ha:hasCalendar ?calendar1.
5   ?user0 ha:hasPhone ?phone2.
6   ?user0 ha:isInLocation ?location3.
7   ?phone2 ha:isInLocation ?location3.
8   ?location3 ha:isNear ?office4.
9   ?user5 a ha:User.
10  ?user5 ha:hasName "Johan"^^xsd:string.
11  ?user5 ha:hasOffice ?office4.}
```
### Fuzzy DL Query Syntax [4]

<table>
<thead>
<tr>
<th>Query</th>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td>(sat?)</td>
<td>Consistency</td>
</tr>
<tr>
<td>Q2</td>
<td>(min-sat? C [o])</td>
<td>Minimal Satisfiability Degree of a concept</td>
</tr>
<tr>
<td>Q3</td>
<td>(max-sat? C [o])</td>
<td>Best Satisfiability Degree of a concept</td>
</tr>
<tr>
<td>Q4</td>
<td>(min-instance? o C)</td>
<td>Best Entailment Degree of a concept assertion</td>
</tr>
<tr>
<td>Q5</td>
<td>(max-instance? o C)</td>
<td>Maximal Entailment Degree of a concept assertion</td>
</tr>
<tr>
<td>Q6</td>
<td>(min-related? o1 o2 R)</td>
<td>Best Entailment Degree of a role assertion</td>
</tr>
<tr>
<td>Q7</td>
<td>(max-related? o1 o2 R)</td>
<td>Maximal Entailment Degree of a role assertion</td>
</tr>
<tr>
<td>Q8</td>
<td>(min_subs? C D)</td>
<td>Best Entailment Degree of a GCI</td>
</tr>
<tr>
<td>Q9</td>
<td>(max_subs? C D)</td>
<td>Maximal Entailment Degree of a GCI</td>
</tr>
<tr>
<td>Q10</td>
<td>(min-g_subs? C D)</td>
<td>BED of a GCI using Gödel implication</td>
</tr>
<tr>
<td>Q11</td>
<td>(max-g_subs? C D)</td>
<td>MED of a GCI using Gödel implication</td>
</tr>
<tr>
<td>Q12</td>
<td>(min-l_subs? C D)</td>
<td>BED of a GCI using Łukasiewicz implication</td>
</tr>
<tr>
<td>Q13</td>
<td>(max-lsubs? C D)</td>
<td>MED of a GCI using Łukasiewicz implication</td>
</tr>
<tr>
<td>Q14</td>
<td>(min-kd_subs? C D)</td>
<td>BED of a GCI using Kleene-Dienes implication</td>
</tr>
<tr>
<td>Q15</td>
<td>(max-kd_sub? C D)</td>
<td>MED of a GCI using Kleene-Dienes implication</td>
</tr>
<tr>
<td>Q16</td>
<td>(all_instances? C)</td>
<td>Instance retrieval</td>
</tr>
<tr>
<td>Q17</td>
<td>(max_var? var)</td>
<td>Variable maximisation</td>
</tr>
<tr>
<td>Q18</td>
<td>(min_var? var)</td>
<td>Variable minimisation</td>
</tr>
<tr>
<td>Q19</td>
<td>(defuzzify-lom? C o t)</td>
<td>LOM defuzzification</td>
</tr>
<tr>
<td>Q20</td>
<td>(defuzzify-som? C o t)</td>
<td>SOM defuzzification</td>
</tr>
<tr>
<td>Q21</td>
<td>(defuzzify-mom? C o t)</td>
<td>MOM defuzzification</td>
</tr>
<tr>
<td>Q22</td>
<td>(bnp? F)</td>
<td>Best Non-Fuzzy Performance</td>
</tr>
</tbody>
</table>
**Fuzzy Wine Ontology v 1.00**

Choose context:
- Candle

Choose food:
- Game

Submit

This Fuzzy Wine Ontology is based on 601 wines

---

You picked: Candle and Game

The most suitable wines for this combination are:

- 0.883 Villages_Cuvee_3_Fleurs
- 0.881 Abadal Cabernet Sauvignon Reserva
- 0.823 Domaine Depeyre
- 0.717 Belleruche
- 0.713 Baron_de_Ley_Reserva
- 0.709 Terres de Berne
- 0.704 Beringer_Clear_Lake_Zinfandel
- 0.703 Beringer_Founders_Estate_Merlot
- 0.699 Amarone_della_Valpolicella_Classico_I_Castei_2
- 0.699 Amarone della Valpolicella Classico I Castei
Fuzzy DL Example: Wine ontology [4]

```prolog
# Fuzzy logic
(define-fuzzy-logic zadeh)

# Datatypes
(define-fuzzy-concept MediumAlcoholForWine triangular(0.0, 20.0, 12.0, 13.0, 14.0))
(define-fuzzy-concept HighPriceForWine right-shoulder(0.0, 10000.0, 15.0, 30.0))

# TBox axioms
(implies (and SparklingWine (some hasSugar DrySugarContentForSparklingWine)) DrySparklingWine 1.0)
(define-primitive-concept PinotNoir (some hasColor RedWineColor))
(define-primitive-concept Chianti (some locatedIn ChiantiRegion))
(define-concept RedWine (and Wine (some hasColor RedWineColor)))
(define-concept Beaujolais (and Wine (some locatedIn BeaujolaisRegion)))
(define-concept HighPriceWine (some hasPrice HighPriceForWine))

# RBox axioms
(implies-role madeFromGrape madeFromFruit 1.0)
(transitive locatedIn)
(symmnetric adjacentRegion)
(functional hasQualitativeSugar)
(inverse hasMaker producesWine)
(domain madeFromGrape Wine)
(range madeFromGrape WineGrape)

# ABox axioms
(related RemyPannier2009 DAjouWinery hasMaker 1.0)
(instance RemyPannier2009 (= hasAlcohol 12.0) 1.0)
(instance RemyPannier2009 (= hasPrice 8.0) 1.0)

# Query
(min-instance? RemyPannier2009 HighPriceWine)
```
<table>
<thead>
<tr>
<th>Subscription pattern</th>
<th>fuzzyDL query</th>
</tr>
</thead>
<tbody>
<tr>
<td>(?, ?, ?, ?)</td>
<td>∀ Concept C: (all-instances? C)</td>
</tr>
<tr>
<td>(s, ?, ?, ?)</td>
<td>If s is a Concept: (min-sat? s)</td>
</tr>
<tr>
<td></td>
<td>If Individual s ∈ Concept C: (min-instance? s C)</td>
</tr>
<tr>
<td>(?, p, ?)</td>
<td>If D is p's Domain and R is p's Range; ∀ Individual d ∈ D and ∀ Individual r ∈ R: (min-related? d r p)</td>
</tr>
<tr>
<td>(?, ?, o)</td>
<td>If o is a Concept: (min-sat? o)</td>
</tr>
<tr>
<td></td>
<td>If Individual o ∈ Concept C: (min-instance? o C)</td>
</tr>
<tr>
<td>(s, p, ?)</td>
<td>If R ∈ p.Range: ∀ Individual i ∈ R: (min-related? s i p)</td>
</tr>
<tr>
<td>(?, p, o)</td>
<td>If D ∈ p.Domain: ∀ Individual i ∈ D: (min-related? i o p)</td>
</tr>
<tr>
<td>(s, ?, o)</td>
<td>∀ Role r, (min-related? s o r)</td>
</tr>
<tr>
<td>(s, p, o)</td>
<td>(min-related? s o p)</td>
</tr>
</tbody>
</table>
Practical tools for fuzzy logic and fuzzy ontologies:

- **Scikit-fuzzy[^6]**[11]

[^4]: https://tinyurl.com/ya8l9y9h
[^5]: Useful for rule induction from incomplete datasets, a generalization of fuzzy membership
[^6]: https://github.com/scikit-fuzzy/scikit-fuzzy
Research problems in Description Logics
Research challenges in (approximated) reasoning

- Scalability (subsumption algorithms [1]: classifying large graphs)
- Reasoning under inconsistency-tolerant semantics: inherently intractable (even for very simple DLs [9] or for tractable DLs).
- Automatic ontology learning
- Can we provide near real time reasoning answers via
  - KR learned with deep learning?
  - Genetic algorithm approximations?
Research challenges in (approximated) reasoning

- Ontology evolution, merging, matching, unification of different specializations (Ex. cross-taxon resource unification ontology for policy consensus decision making [8]).
Research challenges in (approximated) reasoning

Neural-symbolic learning and reasoning (NeSy community)

Three blocks stacked
Top one is green
Bottom one is red

\[
\begin{array}{c}
A \\
B \\
C \\
\end{array}
\]
green
red

Is there a green block directly on top of a non-green block?
Description Logics in practice!
Encode it into Description logics and prove that $KB \models \textit{ItalianProf} \sqsubseteq \textit{LatinLover}$
Encode it into Description logics and prove that $KB \models \text{ItalianProf} \sqsubseteq \text{LatinLover}$

Solution:

- Lazy $\sqsubseteq$ Italian
- Mafioso $\sqsubseteq$ Italian
- LatinLover $\sqsubseteq$ Italian
- Italian $\sqsubseteq (\text{Lazy} \sqcup \text{Mafioso} \sqcup \text{LatinLover})$
- ItalianProf $\sqsubseteq$ Italian
- Lazy $\sqsubseteq \neg \text{Mafioso}$
- Lazy $\sqsubseteq \neg \text{LatinLover}$
- Mafioso $\sqsubseteq \neg \text{LatinLover}$
- Mafioso $\sqsubseteq \neg \text{ItalianProf}$
- Lazy $\sqsubseteq \neg \text{ItalianProf}$
Ontology examples: Kinect movement and interaction ontology [7]
Fuzzy Human Activity Recognition [6]

Rule 1: (define-concept antecedent1 (w-sum (0.17 reachMilkOrBowlOrBox) (0.41 moveMilkOrBowlOrBox) (0.24 placeMilkOrBowlOrBox) (0.01 openMilkOrBox) (0.16 pourMilkOrBox))) (define-concept consequent1 (g-and User (some performsActivity cereal)))

Rule 2: (define-concept antecedent2 (w-sum (0.29 reachCupOrMedicineBox) (0.3 moveCupOrMedicineBox) (0.1 placeCupOrMedicineBox) (0.1 openMedicineBox) (0.1 eatMedicineBox) (0.1 drinkCup))) (define-concept consequent2 (g-and User (some performsActivity medicine)))

Rule 3: (define-concept antecedent3 (w-sum (0.26 reachStackable) (0.27 moveStackable) (0.27 placeStackable) (0.20 nullSA))) (define-concept consequent3 (g-and User (some performsActivity stacking)))

Rule 4: (define-concept antecedent4 (w-sum (0.26 reachStackable) (0.27 moveStackable) (0.27 placeStackable) (0.20 nullSA))) (define-concept consequent4 (g-and User (some performsActivity unstacking)))

Rule 5: (define-concept antecedent5 (w-sum (0.32 reachMicroOrDrinkingKitchenware) (0.11 moveDrinkingKitchenware) (0.11 placeDrinkingKitchenware) (0.12 openMicro) (0.11 closeMicro) (0.23 nullSA))) (define-concept consequent5 (g-and User (some performsActivity microwaving)))

Rule 6: (define-concept antecedent6 (w-sum (0.26 reachPickable) (0.27 movePickable) (0.47 nullSA))) (define-concept consequent6 (g-and User (some performsActivity bending)))

Rule 7: (define-concept antecedent7 (w-sum (0.27 reachMicroOrCloth) (0.23 moveCloth) (0.1 placeCloth) (0.1 openMicro) (0.1 closeMicro) (0.1 cleanMicroOrCloth) (0.1 nullSA))) (define-concept consequent7 (g-and User (some performsActivity cleaningObjects)))
Fuzzy Human Activity Recognition [6]
Let’s get started!

Learning to model fuzzy ontologies with *fuzzyDL* reasoner:

- *FuzzyDL* syntax:
  http://www.umbertostraccia.it/cs/software/fuzzyDL/fuzzyDL.html

- *FuzzyDL* syntax and semantics cheatsheet:
  https://tinyurl.com/y8slmcck

- How to write ontologies in *fuzzyDL*:
  http://www.umbertostraccia.it/cs/software/FuzzyOWL/index.html
  → Matchmaking ontology and query examples in *fuzzyDL* web)\(^7\)
Acknowledgements
• All Protégé team
• Stefano Bragaglia
• Umberto Straccia and Fernando Bobillo
• Carl Lagoze
• Robin Wikström, Juan Antonio Morente Molinera, Matteo Brunelli
• Martin Giese, Leif Harald Karlsen
• Tarek Besold
References


