AE4M33RZN, Fuzzy description logic: fuzzyDL reasoner

Radomír Černoch

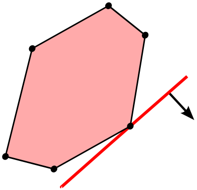
radomir.cernoch@fel.cvut.cz

Faculty of Electrical Engineering, CTU in Prague

23/11/2015

Linear programming in a nutshell

Imagine a 2D space with a convex polygon in the space (x, y). Given constraints $4x + y \geq 6, y \leq 8, \ldots,$ minimize $x - 2y$.



Source: [Wikipedia, 2013]

- Usually written in a matrix form

$$\text{maximize } \mathbf{c}^T \cdot x \tag{1}$$

$$\text{subject to } A\,\mathbf{x} \leq 0 \tag{2}$$

- (Mixed) Integer LP allows (some) variables to be **discrete**.
- LP with real values is in P class, ILP is NP-complete.

Linear programming in a nutshell

Solution of a ⟨⟨M⟩I⟩LP

- One solution (a point in the polytope).
- No solution (the polytope is empty).
- Multiple solutions with equal objective function value.

Syntactical notes about fuzzyDL:

- $x \in \mathbb{R}$ will be **real numbers.**
- $y \in \mathbb{N}$ will be **integer numbers.**
- All values x, y will be bounded by $[0, 1]$.

FuzzyDL algorithm overview

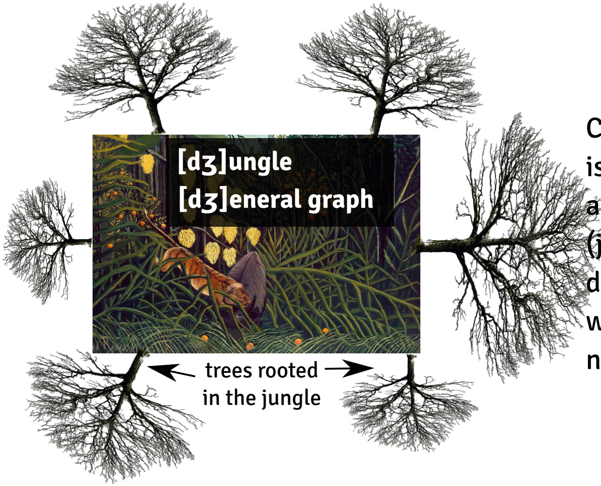
- Transforms $\mathscr{K}$ to the **negated-normal-form.**[1]
- Creates an witnessed interpretation of $\mathscr{K}$.
- During its working it creates
    - a **completion forest** and
    - a **list** of **linear** constraints $\mathscr{C}$.
- Linear constraints $\mathscr{C}$ are solved using any **mixed-integer-linear-programming solver.**

Disclaimer: Not going beyond Ł-logic, no concrete data types.

[1] Makes sure that the negation ¬ appears only in front of concepts using:

$$\mathrm{nnf}(\neg \forall \mathsf{R} \cdot \mathsf{C}) = \exists \mathsf{R} \cdot \mathrm{nnf}(\neg \mathsf{C}) \text{ and } \mathrm{nnf}(\neg \exists \mathsf{R} \cdot \mathsf{C}) = \forall \mathsf{R} \cdot \mathrm{nnf}(\neg \mathsf{C}) .$$

Completion-forest informally



Completion forest is a **graph**, that has a general structure (jungle) “in the middle” and many trees, whose root nodes are nodes in the jungle.

Completion-forest formally

The fuzzyDL algorithm starts with creating the “jungle”. It contains all **individuals** (connected by an edge if they are linked by some relation).

Initialization

- Create a new vertex v_a for each **individual** a in the $\mathscr{K}$.
- Create an edge (v_a, v_b) for each role assertion between a and b.
- Add a **label** $\langle \mathsf{C}, n \rangle$ to vertex a for each concept assertion $\langle a : \mathsf{C} \,|\, n \rangle$.
- Add a label $\langle \mathsf{R}, n \rangle$ to edge (a, b)
  for each role assertion $\langle (a, b) : \mathsf{R} \,|\, n \rangle$.

Forest completion (1)

The reasoner applies each of the following rules sequentially:

- A If a vertex v is labeled $\langle \mathsf{C}, l\rangle$, add $(x_{v:\mathsf{C}} \geq l)$ into $\mathscr{C}$.
- $\bar{\mathsf{A}}$ If a vertex v is labeled $\langle \neg\, \mathsf{C}, l\rangle$, add $(x_{v:\mathsf{C}} \leq 1 - l)$ into $\mathscr{C}$.
- R If an edge (v, w) is labeled $\langle \mathsf{R}, l\rangle$, add $(x_{(v,w):\mathsf{R}} \geq l)$ into $\mathscr{C}$.
- $\perp$ If a vertex v is labeled $\langle \perp, l\rangle$, add $(l = 0)$ into $\mathscr{C}$.

Forest completion (2)

$\sqcap$ If a vertex v is labeled $\langle C \sqcap D, l\rangle$, append labels $\langle C, x_1\rangle$, $\langle D, x_2\rangle$ to v and add the following constraints into $\mathscr{C}$ (with fresh x_1, x_2, y):

$$\begin{aligned} y &\leq 1 - l \\ x_1 &\leq 1 - y \\ x_2 &\leq 1 - y \\ x_1 + x_2 &= l + 1 - y \end{aligned}$$

$\sqcup$ If a vertex v is labeled $\langle C \sqcup D, l\rangle$, append labels $\langle C, x_1\rangle$, $\langle C, x_2\rangle$ to v and add $(x_1 + x_2 = l)$ into $\mathscr{C}$ (with fresh x_1, x_2, y).

Forest completion (3)

$\forall$ If a vertex v is labeled $\langle \forall \mathsf{R} \cdot \mathsf{C}, l_1 \rangle$, an edge (v, w) is labeled $\langle R, l_2 \rangle$ and the rule has not been applied to this pair, then append the label $\langle \mathsf{C}, x \rangle$ to w and add the following constraints into $\mathscr{C}$ (with fresh x, y):

$$l_1 + l_2 - 1 \leq x \leq y \leq l_1 + l_2$$

$\sqsubseteq$ If $\langle \mathsf{C} \sqsubseteq \mathsf{D} \mid n \rangle \in \mathscr{K}$, and the rule has not been applied to a node v, then append labels $\langle \mathrm{nnf}(\neg\, \mathsf{C}), 1 - x_1 \rangle$, $\langle \mathsf{D}, x_2 \rangle$ to v and add $(x_1 \leq x_2 + 1 - n)$ to $\mathscr{C}$.

Forest completion: Example

Consider $\mathscr{K} = \{\langle \exists \mathsf{R} \cdot \mathsf{C} \sqsubseteq \mathsf{D} \mid 1\rangle, \langle (a, b) : \mathsf{R} \mid 0.7\rangle, \langle b : \mathsf{C} \mid 0.8\rangle\}$.
Show that $\mathrm{glb}(\mathscr{K}, a : \mathsf{D}) = 0.5$.

Termination (1)

Unless the rules are applied repeatedly, the algorithm (as explained so far) terminates.
For defining $\exists$ rule, new nodes are added, which needs to refine the terminating condition.

Equivalence of labels

Two lists of labels $[\langle C_1, l_1\rangle, \dots, \langle C_n, l_n\rangle]$ and $[\langle C_1, l'_1\rangle, \dots, \langle C_n, l'_n\rangle]$ are equivalent iff either

- l_i and l'_i are variables or
- l_i and l'_i are negated variables or
- l_i and l'_i are equal rationals.

Termination (2)

Directly blocked node

A node is directly blocked iff

- it is outside the “jungle” and
- none of its ancestors are blocked and
- **it has an ancestor with equivalent labels.**

Blocked node

A node is blocked iff either

- it is directly blocked or
- one of its predecessors is blocked.

Forest completion (4)

$\exists$ If a vertex v is labeled $\langle \exists \mathsf{R} \cdot \mathsf{C}, l\rangle$ and **it is not blocked,** add a new vertex w and an edge (v, w), add labels $\langle C, x_2\rangle$ to w, and $\langle R, x_1\rangle$ to (v, w) and the following constraints into $\mathscr{C}$ (with fresh x_1, x_2 and y):

$$\begin{aligned} y &\leq 1 - l \\ x_1 &\leq 1 - y \\ x_2 &\leq 1 - y \\ x_1 + x_2 &= l + 1 - y \end{aligned}$$

FuzzyDL: Overview

- The instance of MILP is created using constraints $\mathscr{C}$.
- In order to solve $\mathrm{glb}(\mathscr{K}, \langle a : \mathsf{C}\rangle)$, the objective function is set to minimize x in the MILP instance created for an augmented knowledge base $\mathscr{K} \cup \langle a : \neg\, \mathsf{C} \mid 1 - x\rangle$.
- Similarly for $\mathrm{glb}(\mathscr{K}, \langle a : C \sqsubseteq D\rangle)$ the augmented knowledge base is $\mathscr{K} \cup \langle a : \mathsf{C} \sqcap \neg\, \mathsf{D} \mid 1 - x\rangle$.
- $\mathscr{K}$ is inconsistent iff the MILP instance has no solution.
  Hence the $\mathrm{glb}(\cdot, \cdot)$ is found if MILP instance has a solution.

Concrete data types

The domain $\Delta^{\mathscr{I}}$ is an unordered set. This is good for modelling cathegorical data: e.g. colors, people, ...

General idea: Extended interpretation

But we also need to include real numbers $\mathbb{R}$. The *fuzzy description logic with concrete datatypes* $\mathcal{SHIF}(\mathcal{D})$ uses “abstract objects” and “concrete objects”:

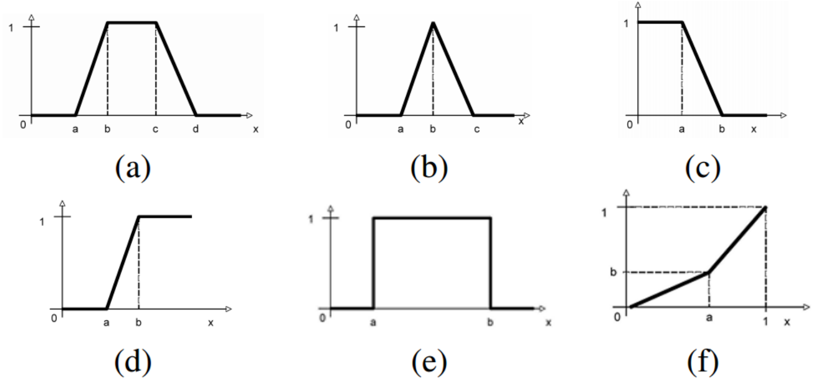
$$\Delta^{\mathscr{I}} = \Delta_a^{\mathscr{I}} \cup \mathbb{R}$$

Concrete data types

- *Concrete individuals*, are interpreted as objects from $\mathbb{R}$.
- *Concrete concepts*, are interpreted as subsets from $\mathbb{R}$.
- *Concrete roles*, are interpreted as subsets from $(\Delta_a^{\mathscr{I}} \times \mathbb{R})$.

All non-concrete notions are called *abstract*.

Concrete data types: New concepts



(a) (b) (c) (d) (e) (f)

Fig. 1. (a) Trapezoidal function; (b) Triangular function; (c) *L*-function; (d) *R*-function; (e) Crisp interval; (f) Linear function.

Ex: Age of parents

```
(related adam bob parent) (related adam eve parent)

(define-fuzzy-concept around23 triangular(0,100, 18,23,26))
(define-fuzzy-concept moreTh17 right-shoulder(0,100, 13,21))
(instance bob (some age around23) 0.9)
(instance eve (some age moreTh17))

(define-fuzzy-concept young left-shoulder(0,100, 17,25))
(define-concept YoungPerson (some age young))

(min-instance? eve YoungPerson) (max-instance? eve YoungPerson)
(min-instance? bob YoungPerson) (max-instance? bob YoungPerson)
(min-instance? adam (all  parent YoungPerson))
(max-instance? adam (all  parent YoungPerson))
(min-instance? adam (some parent YoungPerson))
(max-instance? adam (some parent YoungPerson))
```

Ex: Age of parents

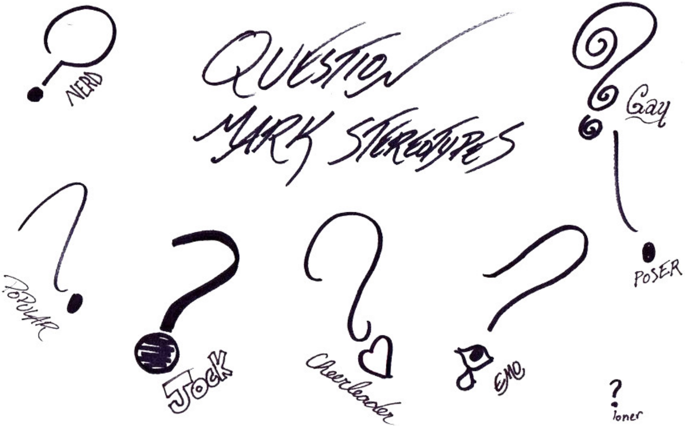
1. What are the bounds on α from $\langle \mathit{eve} : \text{YoungPerson} \mid \alpha \rangle$?

   Start by drawing the concept around23, then construct an interpretation. How much freedom do you have when constructing the interpretation?

2. Let fuzzyDL reasoner give you both bounds on $\langle i : \text{YoungPerson} \mid \beta_i \rangle$ for $i \in \{\mathit{eve}, \mathit{bob}\}$.

   How do you infer the bounds on $\langle \mathit{adam} : \text{YoungPerson} \mid \gamma \rangle$?

Questions?! Ask, please.



Source: `ragtagdoodles.deviantart.com`

Bibliography

Baader, F. (2003).
The Description Logic Handbook: Theory, Implementation, and Applications.
Cambridge University Press.

Hájek, P. (2005).
Making fuzzy description logic more general.
Fuzzy Sets and Systems, 154(1):1–15.

Wikipedia (2013).
Linear programming – Wikipedia, the free encyclopedia.
[Online; accessed 17-November-2013].