Engineering of ontologies with Description Logics

5.1 advanced topics: perceptron operators in description logics

Nicolas Troquard

Joint work with

Pietro Galliani
Oliver Kutz
Claudio Masolo
Guendalina Righetti
Daniele Porello

Outline

Some Description Logics

$\mathcal{ALC}$ concepts, examples:

- Fly $\sqcap$ Mammal
- $\neg$Mammal
- $\exists$has.Heart

$\mathcal{ALCQ}$ concepts, examples:

- all $\mathcal{ALC}$ concepts
- $(\geq 2\,\text{has.Leg})$

Axioms, examples:

- Mammal $\sqsubseteq$ Animal
- Animal $\sqsubseteq$ $\exists$has.Heart
- Cat(garfield)

Interpretations $I = (\Delta^I, \cdot^I)$:

- Δ^I a set of individuals
- $C^I \subseteq \Delta^I$
- $R^I \subseteq \Delta^I \times \Delta^I$

Semantics, examples:

- $(\text{Fly} \sqcap \text{Mammal})^I$: all the individuals that fly and are mammals
- Leg^I: all the individuals that are legs
- has^I: all the pairs of individuals (d_1, d_2) such that d_1 possesses d_2.
- $(\geq 2\,\text{has.Leg})^I$: all the indviduals that possess at least two things that are legs.

Perceptron operators in Description Logics

Define a concept in terms of a threshold and weights. E.g.,

$$\text{Majority} = \nabla\!\!\!\!\nabla^{1}(C_1 : 1/2, C_2 : 1/2, C_3 : 1/2) \ .$$

It is true of some individual d in an interpretation I if and only if

$$1/2 \cdot \left\{ \begin{array}{ll} 1 & \text{if } d \in C_1^I \\ 0 & otherwise \end{array} \right\} + 1/2 \cdot \left\{ \begin{array}{ll} 1 & \text{if } d \in C_2^I \\ 0 & otherwise \end{array} \right\} + 1/2 \cdot \left\{ \begin{array}{ll} 1 & \text{if } d \in C_3^I \\ 0 & otherwise \end{array} \right\} \geq 1 \ .$$

The perceptron operator (or 'tooth'):[1]

$$\mathsf{C} = \nabla\!\!\!\!\nabla^{t}(C_1 : w_1, \ldots, C_p : w_p)$$

where $\vec{w} = (w_1, \ldots, w_p) \in \mathbb{Z}^p$, $t \in \mathbb{Z}$, C_i are concepts expressions.

We define the value $v_{\mathsf{C}}^I(d) = \sum_{i \in \{1,\ldots,p\}} \{w_i \mid d \in C_i^I\}$ and the truth condition

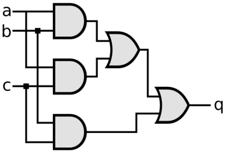
$$\mathsf{C}^I = \{d \in \Delta^I \mid v_{\mathsf{C}}^I(d) \geq t\} \ .$$

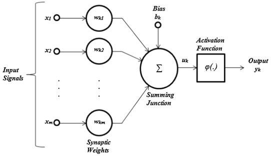
[1] Daniele Porello et al. "A Toothful of Concepts: Towards a Theory of Weighted Concept Combination". In: *Description Logics 2019*. 2019.

Links with circuits and with learning models

> *[The majority ternary operation] $\langle xyz \rangle$ is probably the most important ternary operation in the entire universe, because it has amazing properties that are continually being discovered and rediscovered.* [D. Knuth. The Art of Computer Programming, Vol. 4a Part 1, p. 63]

Threshold Operators have been studied in the context of propositional logic and circuit complexity: [Valiant 1984], [Hajnal et al. 93], [Beimel and Weinreb 2006], [Goldmann et al. 1992] [Goldmann and Karpinski 1998].





Here, instead, we are interested in their possible application to Knowledge Representation in Description Logic.

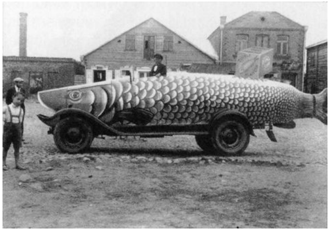
The models of neurons in [McCulloch and Pitts 1943] are built from threshold functions.

Perceptron operators are simple connectives that provide a natural link between knowledge representation and statistical learning: obvious connections with linear classification models.

Some uses in knowledge representation

Accommodating non-prototypical individuals:

Creating concept combinations [Righetti et al. 2021][2]:



[2] Guendalina Righetti et al. "Concept combination in weighted logic". In: *JOWO 2021 proceedings.* CEUR, 2021.

Florida Criminal Punishment Code

Felony Score Sheet describes various features of a crime and their assigned points. Features may include 'possession of cocaine', or 'number of caused injuries'. A threshold must be reached to decide compulsory imprisonment.

Rule 3.992(a) Criminal Punishment Code Scoresheet

The Criminal Punishment Code Scoresheet Preparation Manual is available at: *http://www.dc.state.fl.us/pub/sen_cpcm/index.html*

1. DATE OF SENTENCE	2. PREPARER'S NAME	3. COUNTY	4. SENTENCING JUDGE	
5. NAME (LAST, FIRST, MI.I.)	6. DOB	8. RACE ☐B ☐W ☐OTHER	10. PRIMARY OFF. DATE	12. PLEA ☐
	7. DC #	9. GENDER ☐M ☐F	11. PRIMARY DOCKET #	TRIAL ☐

I. **PRIMARY OFFENSE:** If Qualifier, please check ___A ___S ___C ___R (A=Attempt, S=Solicitation, C=Conspiracy, R=Reclassification)

FELONY DEGREE	F.S.#	DESCRIPTION	OFFENSE LEVEL	**POINTS**
_______/	___________/	______________________________________/	__________/	

(Level - Points: 1=4, 2=10, 3=16, 4=22, 5=28, 6=36, 7=56, 8=74, 9=92, 10=116)

Prior capital felony triples Primary Offense points ☐ **I.** ______

II. **ADDITIONAL OFFENSE(S):** Supplemental page attached ☐

DOCKET#	FEL/MM DEGREE	F.S.#	OFFENSE LEVEL	QUALIFY A S C R	COUNTS		POINTS		TOTAL
__________/	_______/	____________/	________	☐ ☐ ☐ ☐	______	x	_____	=	_____
DESCRIPTION ______________________________									
__________/	_______/	____________/	________	☐ ☐ ☐ ☐	______	X	_____	=	_____
DESCRIPTION ______________________________									
__________/	_______/	____________/	________	☐ ☐ ☐ ☐	______	X	_____	=	_____
DESCRIPTION ______________________________									
__________/	_______/	____________/	________	☐ ☐ ☐ ☐	______	x	_____	=	_____
DESCRIPTION ______________________________									

(Level - Points: M=0.2, 1=0.7, 2=1.2, 3=2.4, 4=3.6, 5=5.4, 6=18, 7=28, 8=37, 9=46, 10=58)

A knowledge base describing the laws of Florida would need to represent the score sheet.

It must contain the definition of the concept CompulsoryImprisonment.

We can represent CompulsoryImprisonment, e.g., as:[3]

$$\nabla\!\!\!\nabla^{44}(\text{CocainePrimary} : 16, \text{ModerateInjuries} : 18, \ldots) \ .$$

[3] Pietro Galliani et al. "Perceptron Connectives in Knowledge Representation". In: *EKAW 2020*.

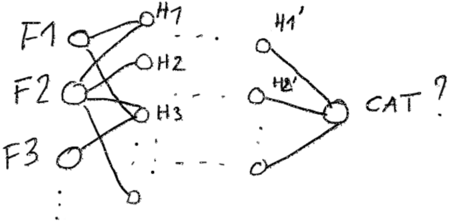
Adding the $\nabla\!\!\!\!\nabla$ operator to a Description Logic that contains $\mathcal{ALC}$ does not affect the complexity of reasoning.

Theorem (Galliani et al., EKAW 2020)

Let $\mathcal{L}$ be a Description Logic that contains all Boolean connectives. A problem of entailment in '$\mathcal{L}$ + perceptrons' can be polynomially reduced to a a problem of entailment in $\mathcal{L}$.

Outline

Importing concepts learnt from data into an ontology



$$\boxed{CAT \equiv \overline{W}^{t}(H1':w_1, \dots, Hk':w_k)}$$

where $H1 \equiv \overline{W}^{t_1}(F1:w_1^1, F2:w_2^1)$

$\vdots$

$$\begin{array}{ll}
MAMMAL & \sqsubseteq ANIMAL \\
ANIMAL & \sqsubseteq \exists has.HEART \\
HEART & \sqsubseteq BODYPART \\
 & \vdots \\
CAT & \equiv \overline{W}^{t}(H1':w_1, \dots, Hk':w_k)
\end{array}$$

Concept learning from data with the perceptron operator

In [Galliani et al., EKAW 2020] we evaluate the practical usefulness of threshold expressions. We investigate how well simple non-nested threshold expressions perform in representing concepts from the Gene Ontology (GO)[4]:

- A knowledge base consisting of over 44,000 different concepts annotating more than one million gene products from 4,591 different species.
- Concepts are partitioned into the three disjoint sub-ontologies:
  - Cellular Component: concept relating to locations inside of a cell ("Nucleus", ...)
  - Biological Process: concept specifying "biological programs" to which a gene product participates ("Asexual Reproduction", "Oxygen Transport", ...);
  - Molecular Function: concept relative to specific molecular-level roles performed by gene products ("Enzyme Binding", "Structural Constituent of Ribosome", ...).

We focus on the annotations of the Saccharomyces Genome Database and on the subset of the Gene Ontology curated for annotating yeast gene products.

We considered the following question: up to which degree is it possible to infer the Molecular Function annotations of a gene product from its Cellular Component and Biological Process ones?

In other words, given the locations of a gene product inside of a yeast cell and the overall "biological programs" it is involved in, can we infer (to some degree, at least) its specific molecular-level role?

[4] Gene Ontology Consortium. "The Gene Ontology (GO) database and informatics resource". In: *Nucleic acids research* (2004).

Concept learning from data with the perceptron operator (approach)

An unsophisticated approach: a very basic evolutionary algorithm to extract threshold expressions from data.

1. generate a population of 100 random perceptron expressions (with Gene Ontology concepts as arguments, integer weights, at most 10 arguments, and threshold fixed at 100),
2. they attempt to "copy" (concept, weight) pairs from randomly selected candidate perceptron expressions;
3. the weights are mutated randomly;
4. every 10 turns the worst-performing half are removed and replaced with random ones.
5. after 1000 turns, return the perceptron expression that performs best over the training data.

Baselines: state-of-the-art learning algorithms as implemented in the Waikato Environment for Knowledge Analysis (WEKA), Random Forest classifier, the Sequential Minimal Optimization algorithm for Support Vector Machines , a decision table majority classifier, a logistic regression classifier and a multilayer perceptron classifier.

Performance measure: Matthews Correlation

Concept learning from data with the perceptron operator (data preparation)

We prepared the data as follows:

1. remove all gene product annotations listed as “dubious” in the Saccharomyces Genome Database.
2. select from the mapping file of the Saccharomyces Genome Database, gene products with at least three annotations of type Cellular Component or Biological Process: 4,595.
3. select as the labels to predict the Molecular Function type annotations that occur in at least 100 of the selected gene products: 17.
4. select as features the Cellular Component or Biological Process terms that apply to at least one of the selected gene products: 120.

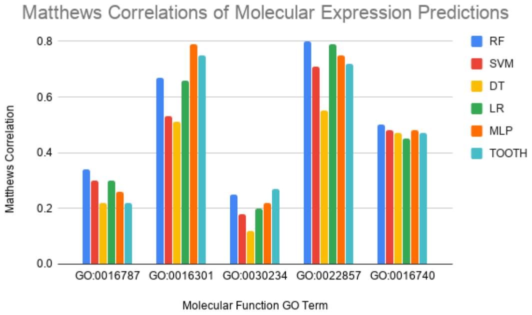
Concept learning from data with the perceptron operator (results)

Matthews Correlations of predictions on five Molecular Function terms. We report averages between five folds and standard deviation.

RF = Random Forest, SVM = Support Vector Machine, DT = Decision Table, LR = Logistic Regression, MLP = Multilayer Perceptron, $\mathbb{W}$ = our Threshold Expressions.

The five rows correspond to the Molecular Function Gene Ontology terms GO:0016787 (hydrolase activity), GO:0016301 (kinase activity), GO:0030234 (enzyme regulator activity), GO:0022857 (transmembrane transporter activity) and GO:0016740 (transferase activity).

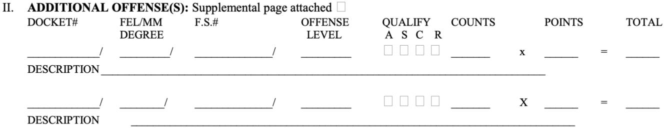
	RF	SVM	DT	LR	MLP	$\mathbb{W}$
GO:0016787	.34 (.02)	.30 (.03)	.22 (.03)	.30 (.03)	.26 (.07)	.22 (.06)
GO:0016301	.67 (.07)	.53 (.06)	.51 (.09)	.66 (.06)	.79 (.03)	.75 (.04)
GO:0030234	.25 (.06)	.18 (.01)	.12 (.03)	.20 (.04)	.22 (.07)	.27 (.06)
GO:0022857	.80 (.02)	.71 (.04)	.55 (.02)	.79 (.02)	.75 (.03)	.72 (.05)
GO:0016740	.50 (.01)	.48 (.03)	.47 (.04)	.45 (.04)	.48 (.02)	.47 (.03)



Outline

The (regular) perceptron operator is lacking

The Felony Score Sheet is slightly more complicated than our first modelling.


II. **ADDITIONAL OFFENSE(S):** Supplemental page attached ☐

DOCKET#	FEL/MM DEGREE	F.S.#	OFFENSE LEVEL	QUALIFY A S C R	COUNTS		POINTS		TOTAL
________/	______/	__________/	_______	☐ ☐ ☐ ☐	_____	x	_____	=	_____
DESCRIPTION ______									
________/	______/	__________/	_______	☐ ☐ ☐ ☐	_____	X	_____	=	_____
DESCRIPTION ______									

For instance, 18 points are added for *every instance* (every count) of a ‘moderate injury victim’.

Of course we can use one concept 1MI, 2MI, 3MI, ... for each number of moderate injury. With all of them pairwise disjoint and with weights 18, 36, 54, ... we can use

$$\mathbb{W}^{44}(\mathsf{CocainePrimary} : 16, \mathsf{1MI} : 18, \mathsf{2MI} : 36, \mathsf{3MI} : 54, \ldots) \ .$$

With each (i+1)MI a subset of iMI, we can use

$$\mathbb{W}^{44}(\mathsf{CocainePrimary} : 16, \mathsf{1MI} : 18, \mathsf{2MI} : 18, \mathsf{3MI} : 18, \ldots) \ .$$

No matter what, one must decide what will be the maximum number of moderate injuries that are taken into account, introduce new concepts (and possibly axioms in the TBox), multiply weights, and write them all into a perceptron operator.

The Counting Perceptron Operator

In [Galliani et al. DL 2021][5], we define a new collection of counting perceptron operators (or 'counting teeth'):

$$C = \mathbb{W}^t_*\big(C_1 : w_1, \ldots, C_p : w_p \mid (R_1, D_1) : m_1, \ldots, (R_q, D_q) : m_q\big) \ ,$$

where $\vec{w} = (w_1, \ldots, w_p) \in \mathbb{Z}^p$, $\vec{m} = (m_1, \ldots, m_q) \in \mathbb{Z}^q$, $t \in \mathbb{Z}$, C_i and D_i are concepts expressions and R_i are roles.

[5] Pietro Galliani, Oliver Kutz, and Nicolas Troquard. "Perceptron Operators That Count". In: *DL 2021*. 2021.

Semantics

$$C = \mathbb{W}^t_*(C_1 : w_1, \ldots, C_p : w_p \mid (R_1, D_1) : m_1, \ldots, (R_q, D_q) : m_q)$$

The following value is possibly ill-defined (when both negative weights and infinite role-branching are allowed).

$$v^I_C(d) = \sum_{i \in \{1,\ldots,p\}} \{w_i \mid d \in C^I_i\} + \sum_{i \in \{1,\ldots,q\}} (m_i \cdot |\{c \in \Delta^I \mid (d,c) \in R^I_i \wedge c \in D^I_i\}|) \ .$$

We introduce two values, $v^I_{C\geq 0}(d)$ and $v^I_{C<0}(d)$: the sum of the non-negative summands, and the sum of the negative summands, respectively.

$$v^I_{C\geq 0}(d) = \sum_{\substack{i \in \{1,\ldots,p\} \\ w_i \geq 0}} \{w_i \mid d \in C^I_i\} + \sum_{\substack{i \in \{1,\ldots,q\} \\ m_i \geq 0}} (m_i \cdot |\{c \in \Delta^I \mid (d,c) \in R^I_i \wedge c \in D^I_i\}|) \ .$$

$$v^I_{C<0}(d) = \sum_{\substack{i \in \{1,\ldots,p\} \\ w_i < 0}} \{w_i \mid d \in C^I_i\} + \sum_{\substack{i \in \{1,\ldots,q\} \\ m_i < 0}} (m_i \cdot |\{c \in \Delta^I \mid (d,c) \in R^I_i \wedge c \in D^I_i\}|) \ .$$

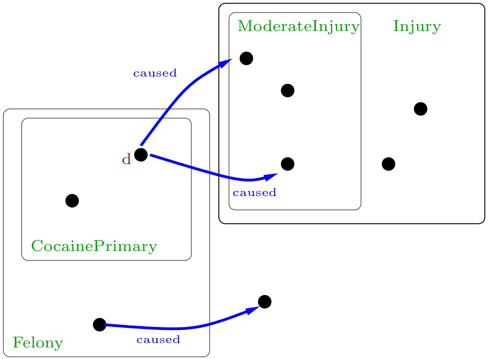
We define:

$$C^I = \{d \in \Delta^I \mid v^I_{C\geq 0}(d) \geq t - v^I_{C<0}(d)\} \ .$$

Felony Score Sheet with counting perceptron operators

With caused a role, we define the concept CompulsoryImprisonment of the Felony Score Sheet as:

$$\text{CompulsoryImprisonment} = \mathbb{W}^{44}_{*}\big(\text{CocainePrimary} : 16, \cdots \mid (\text{caused}, \text{ModerateInjury}) : 18, \ldots\big) \ .$$



$$v^{I}_{\text{CompulsoryImprisonment}_{\geq 0}}(\mathsf{d}) = 16 + 18 \times 2 + \ldots$$
$$= 52 + \ldots$$

The Felony Score Sheet contains only positive weights, so:

$$d \in \text{CompulsoryImprisonment}^{I} \ .$$

$\mathcal{ALCQ}$ into $\mathcal{ALC}$ + counting tooth

Qualified cardinality restrictions in $\mathcal{ALC}$ + counting tooth:

$$(\geq t\, R.C)^I = (\mathbb{W}^{t}_{*}(\_\_ \mid (R, C) : 1))^I$$

$\mathcal{ALC}$ + counting tooth into $\mathcal{ALCQ}$?

One can express "has as many sons as daughters":

$$\begin{aligned}\mathsf{AsMany} = &\mathbb{W}^0_*\big(\_\_ \mid (\mathsf{isParentOf}, \mathsf{Boy}) : 1, (\mathsf{isParentOf}, \mathsf{Girl}) : -1\big) \sqcap \\ &\mathbb{W}^0_*\big(\_\_ \mid (\mathsf{isParentOf}, \mathsf{Girl}) : 1, (\mathsf{isParentOf}, \mathsf{Boy}) : -1\big) \enspace .\end{aligned}$$

This cannot be expressed in $\mathcal{ALCQ}$.[6]

[6] Franz Baader. "A New Description Logic with Set Constraints and Cardinality Constraints on Role Successors". In: *FROCOS 2017.* 2017, Lemma 2.

Particular case: Embedding $\mathcal{ALC}$ with counting teeth with non-negative weights into $\mathcal{ALCQ}$

Consider the counting tooth $\mathsf{C} = \mathbb{W}^t_*(C_1 : w_1, \ldots, C_p : w_p \mid (R_1, D_1) : m_1, \ldots, (R_q, D_q) : m_q)$ where all $m_j \in \vec{m}$ are non-negative.

Now define the counting tooth

$$\mathsf{C'} = \mathbb{W}^t_*(C_1 : w'_1, \ldots, C_p : w'_p, E_1 : w'_{p+1}, \ldots, E_r : w'_{p+r} \mid \\ (R_2, D_2) : m_2, \ldots, (R_q, D_q) : m_q)$$

where:

- $w'_i = w_i$, for $1 \leq i \leq p$
- $\mathsf{r} = \left\lceil \frac{t}{m_1} \right\rceil$
- $w'_{p+i} = i \cdot m_1$, for $1 \leq i \leq \mathsf{r}$
- $E_i = (= i\, R_1.D_1)$, for $0 \leq i \leq \mathsf{r} - 1$
- $E_r = (\geq r\, R_1.D_1)$

Lemma

$$(\mathsf{C})^I = (\mathsf{C'})^I \quad .$$

Preliminary complexity results

Proposition (Galliani et al. DL 2021)

Reasoning with $\mathcal{ALC}$ with counting teeth, disallowing non-negative weights, wrt. to a TBox, is in 2EXPTIME. When the threshold is represented in unary, then it is EXPTIME-complete.

- Iteratively eliminate the counting teeth:
  - Apply the previous rewriting iteratively;
  - The bound r is exponential in the binary representation of the threshold t.
  - Obtain an $\mathcal{ALCQ}$ + regular tooth formula.
- Use [Galliani et al. EKAW 2020][7] to transform the reasoning task into a problem of TBox entailment in $\mathcal{ALCQ}$.
  - Every entailment in $\mathcal{ALCQ}$ + regular teeth, can be polynomially reduced into an entailment in $\mathcal{ALCQ}$. (Ripple carry adder, and digital number comparator in the syntax of $\mathcal{ALC}$.)
- Use the fact that $\mathcal{ALCQ}$ TBox reasoning is EXPTIME-complete.[8]

[7] Galliani et al., "Perceptron Connectives in Knowledge Representation".

[8] Stephan Tobies. "The Complexity of Reasoning with Cardinality Restrictions and Nominals in Expressive Description Logics". In: *J. Artif. Intell. Res.* 12 (2000), pp. 199–217.

$\mathcal{ALCSCC}^\infty$

Main reference: [Baader and De Bortoli, FROCOS 2019].[9]

$\mathcal{ALCSCC}^\infty$ uses formulas of the quantifier-free Boolean algebra with Pressburger arithmetic (QFBAPA$^\infty$) to express constraints on role successors.

The set of $\mathcal{ALCSCC}$ concept expressions over N_C and N_R is defined as follows:

$$C ::= A \mid \neg C \mid C \sqcap C \mid C \sqcup C \mid succ(F) \ ,$$

where $A \in N_C$, F is a QFBAPA$^\infty$ formula using role names and $\mathcal{ALCSCC}$ concept expressions over N_C and N_R as set variables.

We can define:

- $\exists R.C = succ(|R \cap C| \geq 1)$;
- $(\leq n\, R.C) = succ(|R \cap C| \leq n)$;
- $(\geq n\, R.C) = succ(|R \cap C| \geq n)$;
- ... and more.
- E.g., $succ(|R_1 \cap C| - 13 < 2 \cdot |R_2 \cap D|)$, ...

[9] Franz Baader and Filippo De Bortoli. “On the Expressive Power of Description Logics with Cardinality Constraints on Finite and Infinite Sets”. In: *FROCOS 2019.* 2019.

General case: Embedding $\mathcal{ALC}$ with counting teeth into $\mathcal{ALCSCC}^\infty$

Let C $= \mathbb{W}^t_*(C_1 : w_1, \ldots, C_p : w_p \mid (R_1, D_1) : w_{p+1}, \ldots, (R_q, D_q) : w_q)$.

We want to decide whether the concept description C is satisfiable wrt. a TBox $\mathcal{T}$.

We add to $\mathcal{T}$ a fresh role name zoo_{C_i} ('zero-or-one') for every $1 \leq i \leq p$, with axioms:

- $(= 1\, zoo_{C_i}.\top) \equiv C_i$ and
- $(= 0\, zoo_{C_i}.\top) \equiv \neg C_i$

We obtain the TBox $\mathcal{T}'$.

Define

$$\text{summands} = \{w_1 \cdot |zoo_{C_1} \cap \top|, \ldots, w_p \cdot |zoo_{C_p} \cap \top|, w_{p+1} \cdot |R_1 \cap D_1|, \ldots, w_q \cdot |R_q \cap D_q|\} \ .$$

Now consider the $\mathcal{ALCSCC}$ concept

$$\mathsf{C}' = succ\left(\sum_{\substack{w_i \cdot x_i \in \text{summands} \\ w_i \geq 0}} w_i \cdot x_i \geq t - \sum_{\substack{w_i \cdot x_i \in \text{summands} \\ w_i < 0}} w_i \cdot x_i\right) \ .$$

Lemma

C is ($\mathcal{ALC}$ + counting tooth)-satisfiable in $\mathcal{T}$ iff C$'$ is ($\mathcal{ALCSCC}^\infty$)-satisfiable in $\mathcal{T}'$.

Complexity in the general case

Proposition (Galliani et al. DL 2021)

Reasoning in $\mathcal{ALC}$ with counting teeth, wrt. a TBox is EXPTIME-complete, even when the threshold is expressed in binary, and even when the weights on roles are allowed to be negative.

- Apply the previous transformation into $\mathcal{ALCSCC}^{\infty}$.
- Use the fact that TBox entailment in $\mathcal{ALCSCC}^{\infty}$ is EXPTIME-complete [Baader and De Bortoli, FROCOS 2019][10].

[10] Baader and De Bortoli, "On the Expressive Power of Description Logics with Cardinality Constraints on Finite and Infinite Sets".

Conclusions

Concepts in the Description Logic '$\mathcal{ALC}$ + perceptrons' seem to be able to serve as decent estimators of concepts defined from data.

'$\mathcal{ALC}$ + counting perceptrons' without negative weights:

	unary	binary
expressivity	$= \mathcal{ALCQ}$	$= \mathcal{ALCQ}$
complexity	EXPTIME-c	EXPTIME-c

Can be translated into $\mathcal{ALCQ}$.

'$\mathcal{ALC}$ + counting perceptrons':

	unary	binary
expressivity	$> \mathcal{ALCQ}$	$> \mathcal{ALCQ}$
complexity	EXPTIME-c	EXPTIME-c

Can be translated into $\mathcal{ALCSCC}^{\infty}$.

Perspectives:

- study the succinctness of the perceptron operators;
- implementation and statistical learning of counting teeth as in [Galliani et al. EKAW 2020];
- add the perceptron operator as an `OWLClassExpression` in the OWL API.

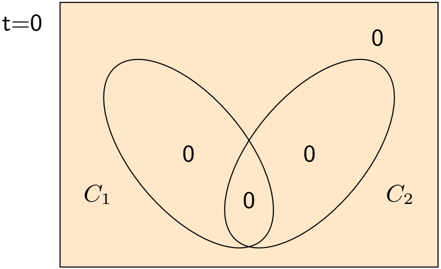
Engineering of ontologies with Description Logics

5.1 advanced topics: perceptron operators in description logics

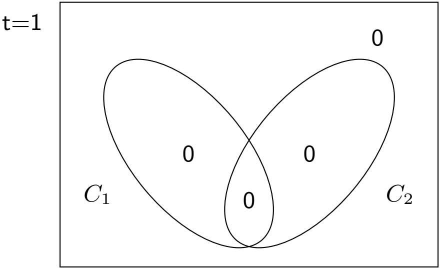
Nicolas Troquard

14 Boolean functions over 2 variables and no nesting

$$\top = \mathbb{W}^{0}_{(0,0)}(C_1, C_2)$$

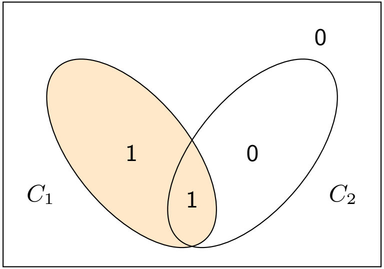


$$\bot = \mathbb{W}^{1}_{(0,0)}(C_1, C_2)$$



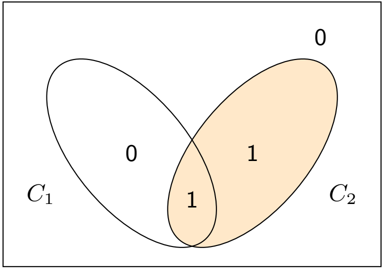
$$C_1 = \mathbb{W}^1_{(1,0)}(C_1, C_2)$$

t=1

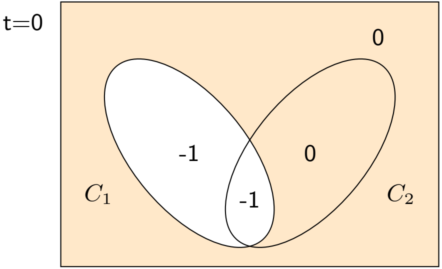


$$C_2 = \mathbb{W}^1_{(0,1)}(C_1, C_2)$$

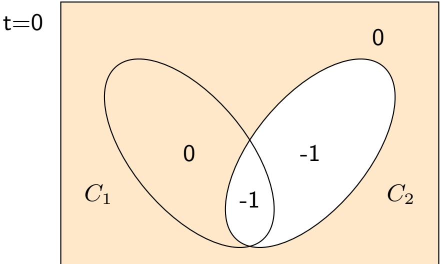
t=1



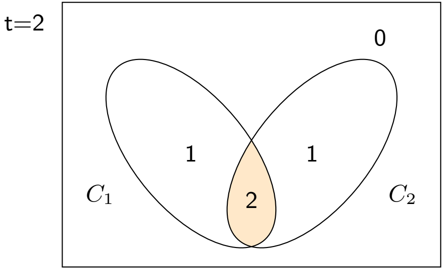
$$\neg C_1 = \mathbb{W}^0_{(-1,0)}(C_1, C_2)$$



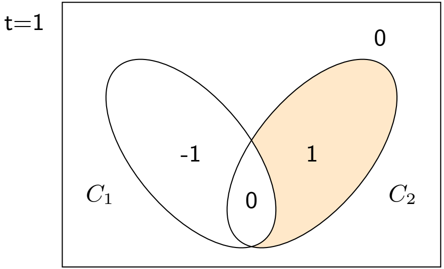
$$\neg C_2 = \mathbb{W}^0_{(0,-1)}(C_1, C_2)$$



$$C_1 \sqcap C_2 = \nabla\!\!\!\!\nabla^2_{(1,1)}(C_1, C_2)$$

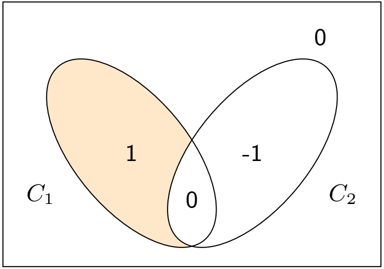


$$\neg C_1 \sqcap C_2 = \nabla\!\!\!\!\nabla^1_{(-1,1)}(C_1, C_2)$$



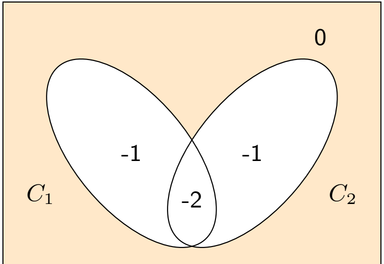
$$C_1 \sqcap \neg C_2 = \nabla\!\!\!\nabla^{1}_{(1,-1)}(C_1, C_2)$$

t=1

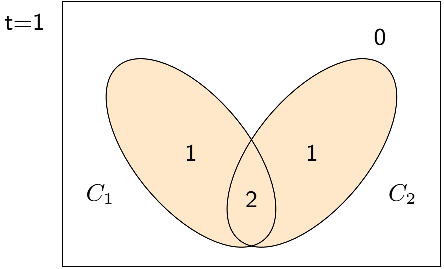


$$\neg C_1 \sqcap \neg C_2 = \nabla\!\!\!\nabla^{0}_{(-1,-1)}(C_1, C_2)$$

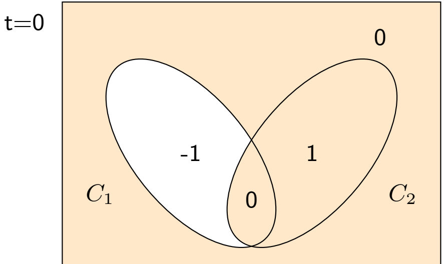
t=0



$$C_1 \sqcup C_2 = \triangledown\!\!\!\!\triangledown^1_{(1,1)}(C_1, C_2)$$

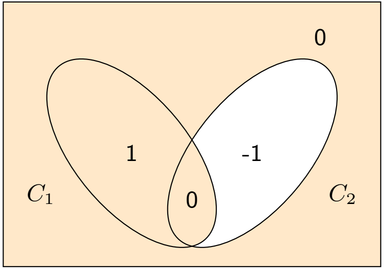


$$\neg C_1 \sqcup C_2 = \triangledown\!\!\!\!\triangledown^0_{(-1,1)}(C_1, C_2)$$



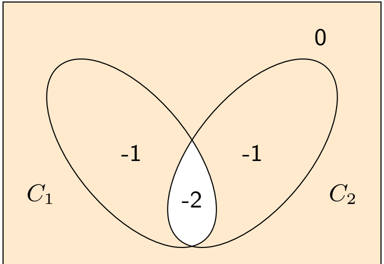
$$C_1 \sqcup \neg C_2 = \mathbb{W}^{0}_{(1,-1)}(C_1, C_2)$$

t=0



$$\neg C_1 \sqcup \neg C_2 = \mathbb{W}^{-1}_{(-1,-1)}(C_1, C_2)$$

t=-1



QFBAPA$^\infty$

The Description Logic $\mathcal{ALCSCC}^\infty$ uses formulas of the quantifier-free Boolean algebra with Pressburger arithmetic (QFBAPA) to express constraints on role successors.
QFBAPA over finite integers is presented in [Kuncak and Rinard 2007][11]. It is extended with infinity in [Baader and De Bortoli 2019][12]. It uses a simple arithmetic with a single (positive) infinity. With $z \in \mathbb{N}$, we stipulate that over $\mathbb{N} \cup \{\infty\}$, the operator $+$ is commutative, and $<$ is a strict linear order, $=$ is an equivalence relation, and: $\infty + z = \infty$, $z < \infty$, $z \leq \infty$, $0 \cdot \infty = 0$, $\infty + \infty = \infty$, $\infty \not< \infty$.
A QFBAPA$^\infty$ formula F is a Boolean combination of set and numerical constraints like A_T.

$$
\begin{aligned}
F &::= A_T \mid A_B \mid \neg F \mid F \wedge F \mid F \vee F \\
A_B &::= B = B \mid B \subseteq B \\
A_T &::= T = T \mid T < T \\
B &::= x \mid \emptyset \mid \mathcal{U} \mid B \cup B \mid B \cap B \mid \overline{B} \\
T &::= k \mid K \mid |B| \mid T + T \mid K \cdot T \\
K &::= 0 \mid 1 \mid 2 \mid \ldots
\end{aligned}
$$

[11] Viktor Kuncak and Martin C. Rinard. “Towards Efficient Satisfiability Checking for Boolean Algebra with Presburger Arithmetic”. In: *CADE-21*. Ed. by Frank Pfenning. 2007.
[12] Baader and De Bortoli, “On the Expressive Power of Description Logics with Cardinality Constraints on Finite and Infinite Sets”.

Semantics of QFBAPA$^\infty$ formulas

The semantics of set terms B is defined using *substitutions* σ that assign a set $\sigma(\mathcal{U})$ to the constant $\mathcal{U}$ and subsets of $\sigma(\mathcal{U})$ to set variables. The evaluation of all set terms under σ is done using the rules of set theory.
Set constraints of the form A_B are evaluated to true or false under σ, also by using the rules of set theory.
Then the domain of σ is extended to PA expressions T by assigning to them an element of $\mathbb{N} \cup \{\infty\}$.
The cardinality expression $|B|$ is evaluated as the cardinality of $\sigma(B)$ if B is finite, and as ∞ if it is not. The evaluation of all PA expressions under σ is done using the rules of addition and multiplication (extended with infinity as above).
Numerical constraints A_T are evaluated to true or false under σ, under the rules of basic arithmetic.
Finally, a *solution* σ of a QFBAPA$^\infty$ formula F is a substitution that evaluates F to true, using the rules of Boolean logic.

Syntax of $\mathcal{ALCSCC}^\infty$

Let N_C and N_R be two disjoint sets of concept names, and role names, respectively.
The set of $\mathcal{ALCSCC}$ concept expressions over N_C and N_R is defined as follows:

$$C ::= A \mid \neg C \mid C \sqcap C \mid C \sqcup C \mid succ(F) \ ,$$

where $A \in N_C$, F is a QFBAPA$^\infty$ formula using role names and $\mathcal{ALCSCC}$ concept expressions over N_C and N_R as set variables.
An $\mathcal{ALCSCC}^\infty$ TBox over N_C and N_R is a finite set of concept inclusions of the form $C \sqsubseteq D$, where C and D are $\mathcal{ALCSCC}^\infty$ concept expressions over N_C and N_R. We write $C \equiv D$ to signify that $C \sqsubseteq D$ and $D \sqsubseteq C$.

Semantics of $\mathcal{ALCSCC}^{\infty}$

Given finite, disjoint sets N_C and N_R of concept and role names, respectively, an interpretation I consists of a non-empty set Δ^I and a mapping $\cdot^I$ that maps every concept name C to a subset $C^I \subseteq \Delta^I$ and every role name $R \in N_R$ to a binary relation $R^I \subseteq \Delta^I \times \Delta^I$. Given an individual $d \in \Delta^I$ and a role name $R \in N_R$, we define $R^I(d)$ as the set of R-successors. We define $ARS^I(d)$ as the set of all successors of d. The mapping $\cdot^I$ is extended to Boolean combinations of concept expressions in the obvious way.

Successor constraints are evaluated according to the semantics of QFBAPA$^{\infty}$. To determine whether $d \in (succ(F))^I$, $\mathcal{U}$ is evaluated as $ARS^I(d)$, the roles occurring in F are substituted with $R^I(d)$, and the concept expressions C occurring in F are substituted with $C^I \cap ARS^I(d)$.

Then, $d \in (succ(F))^I$ is true iff this substitution is a solution of the QFBAPA$^{\infty}$ formula F.

The interpretation I is a model of the TBox $\mathcal{T}$ if for every concept inclusion $C \sqsubseteq D$ in $\mathcal{T}$, it is the case that $C^I \subseteq D^I$.

A concept expression C is satisfiable wrt. the TBox $\mathcal{T}$ if there exists a model of the TBox such that $C^I \neq \emptyset$.

Example

In the $\mathcal{ALCSCC}^\infty$ formula $succ(|\text{causes}| < 2)$, 2 is an integer constant (also a PA expression), causes is a role, but also a set term, $|\text{causes}|$ is a set cardinality (also a PA expression), and $|\text{causes}| < 2$ is a numerical constraint.
When deciding whether $d \in (succ(|\text{causes}| < 2))^I$, we build the substitution σ, such that $\sigma(2) = 2$, and $\sigma(\text{causes}) = \text{causes}^I(d)$.
Let I be an interpretation, and suppose that d has 2 causes-successors, namely d_1 and d_2 (and nothing else). We then have $\sigma(\text{causes}) = \{d_1, d_2\}$, $\sigma(|\text{causes}|) = 2$, and $\sigma(|\text{causes}| < 2) = false$. There are no other possible substitutions to consider. So $d \notin (succ(|\text{causes}| < 2))^I$.
Suppose that we also have $\text{Injury}^I = \{d_2, d_3, d_4\}$, and $(d, d_3) \in \text{has}^I$ (but nothing else).
When deciding whether $d \in (succ(|\text{causes} \cap \text{Injury}| = 1))^I$, Injury is a concept description but also a set term, and we build the substitution σ' such that $\sigma'(1) = 1$, $\sigma'(\text{causes}) = \text{causes}^I(d) = \{d_1, d_2\}$, $\sigma'(\text{Injury}) = \text{Injury}^I \cap ARS^I(d) = \{d_2, d_3\}$, $\sigma'(\text{causes} \cap \text{Injury}) = \sigma'(\text{causes}) \cap \sigma'(\text{Injury}) = \{d_2\}$, $\sigma'(|\text{causes} \cap \text{Injury}|) = 1$, and $\sigma'(|\text{causes} \cap \text{Injury}| = 1) = true$. Hence σ' is a solution of the QFBAPA$^\infty$ formula $|\text{causes} \cap \text{Injury}| = 1$. So $d \in (succ(|\text{causes} \cap \text{Injury}| = 1))^I$.

$\mathcal{ALC}$ and $\mathcal{ALCQ}$

$\mathcal{ALCQ}$ is the fragment of $\mathcal{ALCSCC}^\infty$ such that $succ(F)$ is of the form $succ(|R \cap C| \leq n)$ or $succ(|R \cap C| \geq n)$, where C is a concept expression and $R \in N_R$, and $n \in \mathbb{N}$. $\mathcal{ALC}$ is the fragment of $\mathcal{ALCSCC}^\infty$ such that $succ(F)$ is of the form $succ(|R \cap C| \geq 1)$.
Hence, we can define $\exists R.C = succ(|R \cap C| \geq 1)$, $(\leq n\, R.C) = succ(|R \cap C| \leq n)$, and $(\geq n\, R.C) = succ(|R \cap C| \geq n)$.

Example with negative weights

For purposes of illustration we define a 'Modified Compulsory Imprisonment' as

$$MCI = \mathbb{W}_*^{44}\big(\text{CocainePrimary} : 16 \mid (\text{caused}, \text{ModerateInjury}) : 18),$$
$$(\text{preventiveDetention}, \text{Month}) : -1\big) \ ,$$

where only cocaine possession as primary offence and the number of moderate injuries are kept from the original score sheet, and where in addition every month of preventive detention lowers the score by one.
We want to decide whether the felony $d \in \Delta^I$ falls within the definition of this modified compulsory imprisonment, under the assumptions that d is not in CocainePrimary^I, that $|\text{preventiveDetention}^I(d) \cap \text{Month}^I| = 12$, and $|\text{caused}^I(d) \cap \text{ModerateInjury}^I| = 3$.
So, we have: $v^I_{MCI\geq 0}(d) = 0 + 3 \cdot 18 = 54$ and $v^I_{MCI<0}(d) = 12 \cdot (-1) = -12$. We must evaluate $v^I_{MCI\geq 0}(d) \geq t - v^I_{MCI<0}(d)$, which is $54 \geq 44 + 12$, or $54 \geq 56$, which is false. So d does not fall within the modified compulsory imprisonment.