CEUR-WS.org/Vol-2954/paper-15.pdf

Perceptron Operators That Count

Pietro Galliani, Oliver Kutz, and Nicolas Troquard ✉

KRDB Research Centre for Knowledge and Data
Free University of Bozen-Bolzano, Italy
{firstname.lastname}@unibz.it

Abstract. The *Felony Score Sheet* used in the US State of Florida, 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.

In previous research, we have introduced a perceptron operator for knowledge representation languages. With it, concepts can be defined by listing a concept's features with associated weights, and a threshold. An individual belongs to such a concept if the weighted sum of the listed features it belongs to reaches the threshold. It can capture the concept of compulsory imprisonment defined by the Felony Score Sheet. However, it suffers some limitations in that one must artificially use concepts like 'caused one injury', 'caused two injuries', etc, to be able to count.

This paper proposes an extension of the perceptron operator to define concepts like the compulsory imprisonment from the Felony Score Sheet faithfully and easily, relying on role-successors counting. We show that when the weights are non-negative, reasoning in $\mathcal{ALC}$ augmented with the perceptron operator can be reduced to reasoning in $\mathcal{ALCQ}$. Capitalizing on the recent $\mathcal{ALCSCC}$, we also show that adding the operator to $\mathcal{ALC}$ does not affect the complexity of reasoning in general.

1 Introduction

The *Felony Score Sheet* used in the State of Florida[1], describes various features of a crime and their assigned points. A threshold must be reached to decide compulsory imprisonment. For example, if the primary offence is possession of cocaine, then it corresponds to 16 points, one victim injury describable as "moderate" corresponds to 18 points, and a failure to appear for a criminal proceeding results in 4 points. Imprisonment is compulsory if the total is greater than 44 points and not compulsory otherwise.

Copyright © 2021 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

[1] http://www.dc.state.fl.us/pub/scoresheet/cpc_manual.pdf (accessed: June. 2021)

In [1], we introduced a threshold operator for knowledge representation languages. If $C_1, \ldots, C_n$ are concept expressions, $w_1 \ldots w_n \in \mathbb{Z}$ are weights, and $t \in \mathbb{Z}$ is a threshold, we can introduce a new concept

$$\nabla\!\!\!\nabla^t(C_1 : w_1 \ldots C_n : w_n)$$

to designate in an interpretation I those individuals $d \in \Delta^I$ such that:

$$\sum\{w_i \mid d \in (C_i)^I\} \geq t \ .$$

We call it the *threshold* or *perceptron* operator, or "tooth". Adding the perceptron operator to $\mathcal{ALC}$ does not increase the expressivity, since every instance of the operator can be replaced with an equivalent boolean expression. In [2], we also showed that adding the perceptron operator to $\mathcal{ALC}$ did not affect the complexity of reasoning. It is ExpTime-complete to do reasoning wrt. a TBox, as there exists a linear transformation to eliminate the perceptron operator and obtain an equi-satisfiable $\mathcal{ALC}$ formula (although obfuscating the original readability of such defined concepts).

A knowledge base describing the laws of Florida would need to represent this score sheet as part of its definition of its `CompulsoryImprisonment` concept, for instance as

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

The Felony Score Sheet is slightly more complicated, though. 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

$$\nabla\!\!\!\nabla^{44}(\texttt{CocainePrimary} : 16, \texttt{1MI} : 18, \texttt{2MI} : 36, \texttt{3MI} : 54, \ldots) \ .$$

With each $(\texttt{i} + \texttt{1})\texttt{MI}$ a subset of `iMI`, we can use

$$\nabla\!\!\!\nabla^{44}(\texttt{CocainePrimary} : 16, \texttt{1MI} : 18, \texttt{2MI} : 18, \texttt{3MI} : 18, \ldots) \ .$$

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.

Isn't there some more flexible way to specify this kind of concepts? The problem at hand is to investigate how to extend the $\nabla\!\!\!\nabla$ operator to accommodate the Florida example faithfully and with simplicity. In this paper, we extend the regular tooth with role-successor counting abilities.

Related work. [3] introduces a description logic ($\mathcal{DL}$) to define $\mathcal{EL}$ concepts in an approximate way, through a graded membership function. The authors introduce threshold concepts to capture the set of individuals belonging to a concept with a certain degree.

[4] introduces a powerful counting mechanism for description logics. It makes use of number variables in concepts to say that there is an n, and the number of role successors is n or more. Adding this mechanism to $\mathcal{ALC}$ yields an undecidable $\mathcal{DL}$. [5] presents a general method to use arithmetic reasoning as part of the inference engine of description logics. Useful counting operators can be then devised and integrated into $\mathcal{DL}$, and remain decidable. [6] introduces the extension $\mathcal{ALCSCC}$[2] of $\mathcal{ALC}$ with *expressive* statements of constraints on role successors (formulas of quantifier-free Boolean algebra with Pressburger arithmetic (QFBAPA) [7]). It is strictly more expressive than $\mathcal{ALCQ}$; it can, e.g., express "has as many sons as daughters", which $\mathcal{ALCQ}$ cannot. Concept satisfiability is ExpTime-complete, and PSpace-complete wrt. an empty TBox (hence no harder than $\mathcal{ALC}$ [8] or $\mathcal{ALCQ}$ [9]). [10] extends $\mathcal{ALC}$ with *global* expressive cardinality constraints. $\mathcal{ALCQ}$ with global constraints was already studied in [9]. Adding global cardinality constraints in $\mathcal{ALC}$ leads to an NExpTime-complete complexity for reasoning tasks in general. In $\mathcal{ALCSCC}$, the interpretations are restricted to finite-branching roles. In [11], $\mathcal{ALCSCC}^\infty$ is introduced over arbitrary models. The complexity of reasoning is unaffected. [12] show that combining the local expressive cardinality constraints of [6] with the global expressive cardinality constraints of [10] does not impact the complexity.

Outline. We present $\mathcal{ALCSCC}^\infty$ in Section 2, and define $\mathcal{ALC}$ and $\mathcal{ALCQ}$ as fragments. In Section 3 we introduce our extension of the tooth with counting capabilities. In Section 4, we show how to embed into $\mathcal{ALCQ}$ the logic $\mathcal{ALC}$ equipped with the new perceptron operator where the weights are positive. The embedding suffices to show that reasoning can be done in 2ExpTime when the threshold is expressed in binary, and that it is ExpTime-complete when the threshold is expressed in unary. Section 5 provides an embedding into $\mathcal{ALCSCC}^\infty$, showing that $\mathcal{ALC}$ equipped with the new perceptron operator is ExpTime-complete in general. In Section 6, we briefly suggest a further way to extend the perceptron operator. Section 7 concludes.

2 $\mathcal{ALC}$ and its extensions with cardinality restrictions

We present $\mathcal{ALCSCC}^\infty$ and its well-known fragments $\mathcal{ALC}$ and $\mathcal{ALCQ}$. See [13] for a general introduction of $\mathcal{DL}$.

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 [7]. It is extended with infinity in [11]. 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

[2] Here, 'SCC' stands for 'Set and Cardinality Constraints'.

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.

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

Set terms like B are obtained by applying intersection, union, and complement to set variables and constants $\emptyset$ and $\mathcal{U}$. Set constraints like A_B are of the form $B_1 = B_2$ and $B_1 \subseteq B_2$, where B_1, B_2 are set terms like B.

Presburger Arithmetic (PA) expressions T are built from variables, non-negative integer constants from K, and set cardinalities $|B|$, and then closed under addition as well as multiplication with non-negative integer constants from K. They can be used to form the numerical constraints A_T, namely of the form $T_1 = T_2$ and $T_1 < T_2$, where T_1, T_2 are PA expressions of type T.

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 1. In the $\mathcal{ALCSCC}^\infty$ formula $succ(|\texttt{causes}| < 2)$, 2 is an integer constant (also a PA expression), `causes` is a role, but also a set term, $|\texttt{causes}|$ is a set cardinality (also a PA expression), and $|\texttt{causes}| < 2$ is a numerical constraint.

When deciding whether $d \in (succ(|\texttt{causes}| < 2))^I$, we build the substitution σ, such that $\sigma(2) = 2$, and $\sigma(\texttt{causes}) = \texttt{causes}^I(d)$.

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 $\sigma(\texttt{causes}) = \{d_1, d_2\}$, $\sigma(|\texttt{causes}|) = 2$, and $\sigma(|\texttt{causes}| < 2) = false$. There are no other possible substitutions to consider. So $d \notin (succ(|\texttt{causes}| < 2))^I$.

Suppose that we also have $\texttt{Injury}^I = \{d_2, d_3, d_4\}$, and $(d, d_3) \in \texttt{has}^I$ (but nothing else).

When deciding whether $d \in (succ(|\texttt{causes} \cap \texttt{Injury}| = 1))^I$, `Injury` is a concept description but also a set term, and we build the substitution σ' such that $\sigma'(1) = 1$, $\sigma'(\texttt{causes}) = \texttt{causes}^I(d) = \{d_1, d_2\}$, $\sigma'(\texttt{Injury}) = \texttt{Injury}^I \cap ARS^I(d) = \{d_2, d_3\}$, $\sigma'(\texttt{causes} \cap \texttt{Injury}) = \sigma'(\texttt{causes}) \cap \sigma'(\texttt{Injury}) = \{d_2\}$, $\sigma'(|\texttt{causes} \cap \texttt{Injury}|) = 1$, and $\sigma'(|\texttt{causes} \cap \texttt{Injury}| = 1) = true$. Hence σ' is a solution of the QFBAPA$^\infty$ formula $|\texttt{causes} \cap \texttt{Injury}| = 1$. So $d \in (succ(|\texttt{causes} \cap \texttt{Injury}| = 1))^I$.

$\mathcal{ALC}$ 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 $(\geq n\, R.C) = succ(|R \cap C| \geq n)$. *Qualified* cardinality restrictions of $\mathcal{ALCQ}$ are equivalently defined as:

$$(\leq n\, R.C)^I = \left\{d \in \Delta^I \;\middle|\; |\{c \in \Delta^I, (d,c) \in R^I \wedge c \in C^I\}| \leq n\right\}$$

and

$$(\geq n\,R.C)^I = \left\{d \in \Delta^I \;\middle|\; |\{c \in \Delta^I, (d,c) \in R^I \wedge c \in C^I\}| \geq n\right\}$$

We define $(= n\,R.C) = (\geq n\,R.C) \sqcap (\leq n\,R.C)$.

3 Teeth that count

We define a new collection of perceptron operators, that we call simply *counting teeth*:

$$\mathtt{C} = \nabla\!\!\!\nabla^t_*\big(C_1 : w_1, \dots, C_p : w_p \mid (R_1, D_1) : m_1, \dots, (R_q, D_q) : m_q\big) \enspace ,$$

where $\vec{w} = (w_1, \dots, w_p) \in \mathbb{Z}^p$, $\vec{m} = (m_1, \dots, m_q) \in \mathbb{Z}^q$, $t \in \mathbb{Z}$, C_i and D_i are concepts expressions and R_i are roles. (The tooth from our previous work, without role-successor counting will be called *regular tooth*.)

With `caused` a role, we can now faithfully define the concept `CompulsoryImprisonment` of the Felony Score Sheet:

$$\nabla\!\!\!\nabla^{44}_*\big(\mathtt{CocainePrimary} : 16, \cdots \mid (\mathtt{caused}, \mathtt{ModerateInjury}) : 18, \dots\big) \enspace .$$

Semantics. When an individual d has only a finite number of D_i-successors in I, or when the weights are all non-negative, then the *value* of a $d \in \Delta^I$ under the counting tooth $\mathtt{C}$ is:

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

In a manner analogous to the truth value of the regular tooth, we would have:

$$\mathtt{C}^I = \{d \in \Delta^I \mid v^I_{\mathtt{C}}(d) \geq t\} \enspace .$$

When the individual d can have an infinite number of D_i-successors and some weights are positive and others are negative, then $v^I_{\mathtt{C}}(d)$ can be ill-defined. (I.e. what should be the value of $\infty - \infty$?) So instead of the single value $v^I_{\mathtt{C}}(d)$, we introduce two values, $v^I_{\mathtt{C}\geq 0}(d)$ and $v^I_{\mathtt{C}<0}(d)$, that represent the sum of the non-negative summands and the sum of the negative summands, respectively.

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

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

Clearly $v^I_{\mathtt{C}\geq 0}(d) \geq 0$ and $v^I_{\mathtt{C}<0}(d) \leq 0$.

Finally, the semantics of $\mathtt{C}$ in a possibly infinite-branching interpretation I is given as follows:

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

Example 2. For purposes of illustration we define a 'Modified Compulsory Imprisonment' as

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

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 $\mathtt{CocainePrimary}^I$, that $|\mathtt{preventiveDetention}^I(d) \cap \mathtt{Month}^I| = 12$, and $|\mathtt{caused}^I(d) \cap \mathtt{ModerateInjury}^I| = 3$.

So, we have: $v^I_{\mathtt{MCI}\geq 0}(d) = 0 + 3 \cdot 18 = 54$ and $v^I_{\mathtt{MCI}<0}(d) = 12 \cdot (-1) = -12$. We must evaluate $v^I_{\mathtt{MCI}\geq 0}(d) \geq t - v^I_{\mathtt{MCI}<0}(d)$, which is $54 \geq 44 + 12$, or $54 \geq 56$, which is false. So d does not fall within the modified compulsory imprisonment.

4 Particular case: Embedding $\mathcal{ALC}$ with counting teeth with non-negative weights into $\mathcal{ALCQ}$, and preliminary complexity results

In practice, negative weights are not always necessary. As evidence, one can observe that the Felony Score Sheet does not contain negative points for computing the total number of points.

Let us then first restrict our attention to counting teeth $\mathbb{W}^t_*\big(C_1 : w_1, \ldots, C_p : w_p \mid (R_1, D_1) : m_1, \ldots, (R_q, D_q) : m_q\big)$ such that $\vec{m} \in \mathbb{N}^q$. In this section we will still allow the weights on the concepts to be negative: $\vec{w} \in \mathbb{Z}^p$. For regular teeth, we stipulate that the weights are non-negative simply for brevity, and because it does not really matter. Indeed, a regular tooth with negative weights on concepts can be transformed efficiently into a regular tooth with only non-negative weights on concepts, as shown in [14, Prop. 3].

We show that the language $\mathcal{ALC}$ equipped with counting teeth $\mathbb{W}_*$ is no more expressive than $\mathcal{ALCQ}$ (Prop. 1) when the weights are non-negative. We show that reasoning can be done in 2ExpTime when the threshold is expressed in binary and in ExpTime when it is expressed in unary (Prop. 2). We will improve upon the 2ExpTime upper-bound in Section 5. However, this section has the merit to show how one can transform the problem of reasoning with $\mathcal{ALC}$ equipped with the counting tooth with non-negative weights into a problem of reasoning with $\mathcal{ALCQ}$ (although inefficiently when the threshold is encoded in binary), for which efficient reasoning tools exist. We leave for future work whether an efficient embedding into $\mathcal{ALCQ}$ is possible even when the threshold is encoded in binary.

Let us observe that $\mathcal{ALC}$ equipped with counting teeth restricted to non-negative weights is strictly less expressive than when negative weights are allowed. Indeed, one can express "has as many sons as daughters":

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

This cannot be expressed in $\mathcal{ALCQ}$ [6, Lemma 2], but $\mathcal{ALC}$ equipped with counting teeth with non-negative weights has the same expressivity as $\mathcal{ALCQ}$ (Prop. 1).

We recall that, in contrast, adding the regular tooth of [2] to $\mathcal{ALC}$ has the same expressivity whether the weights are possibly negative or not.

Iterated elimination of role-successors counting and expressivity. Consider the counting tooth $\texttt{C} = \nabla\!\!\!\!\nabla^t_*\big(C_1 : w_1, \ldots, C_p : w_p \mid (R_1, D_1) : m_1, \ldots, (R_q, D_q) : m_q\big)$ where all $m_j \in \vec{m}$ are non-negative.

Now define the counting tooth

$$\begin{aligned}\texttt{C}' = \nabla\!\!\!\!\nabla^t_*\big(&C_1 : w'_1, \ldots, C_p : w'_p, E_1 : w'_{p+1}, \ldots, E_r : w'_{p+r} \mid \\ &(R_2, D_2) : m_2, \ldots, (R_q, D_q) : m_q\big)\end{aligned}$$

where:

- $w'_i = w_i$, for $1 \leq i \leq p$
- $w'_{p+i} = i \cdot m_1$, for $1 \leq i \leq r$
- $r = \left\lceil \frac{t + \sum_{1 \leq i \leq p} |w'_i|}{m_1} \right\rceil$ (in fact it's enough to sum only $|w'_i|$ when $w'_i < 0$)
- $E_i = (= i\, R_1.D_1)$, for $0 \leq i \leq r - 1$
- $E_r = (\geq r\, R_1.D_1)$

Lemma 1.

$$(\texttt{C})^I = (\texttt{C}')^I \enspace .$$

Proof. We must show that for $d \in \Delta^I$, we have $v^I_{\texttt{C}\geq 0}(d) \geq t - v^I_{\texttt{C}<0}(d)$ iff $v^I_{\texttt{C}'\geq 0}(d) \geq t - v^I_{\texttt{C}'<0}(d)$.

To make the proof smoother, we can start with a simple observation. When $\texttt{C} = \nabla\!\!\!\!\nabla^t_*\big(C_1 : w_1, \ldots, C_p : w_p \mid (R_1, D_1) : m_1, \ldots, (R_q, D_q) : m_q\big)$ and $\vec{m} \in \mathbb{N}^q$, then for every $d \in \Delta^I$ we have $v^I_{\texttt{C}}(d) \geq t$ iff $v^I_{\texttt{C}\geq 0}(d) \geq t - v^I_{\texttt{C}<0}(d)$. Indeed, when $\vec{m} \in \mathbb{N}^q$, then $v^I_{\texttt{C}<0}(d)$ is finite. So $v^I_{\texttt{C}\geq 0}(d) + v^I_{\texttt{C}<0}(d)$ is well defined and equal to $v^I_{\texttt{C}}(d)$.

To prove the lemma, we can then verify that for $d \in \Delta^I$, we have $v^I_{\texttt{C}'}(d) \geq t$ iff $v^I_{\texttt{C}}(d) \geq t$. We can see that $v^I_{\texttt{C}}(d) \geq v^I_{\texttt{C}'}(d)$. So if $v^I_{\texttt{C}'}(d) \geq t$, then $v^I_{\texttt{C}}(d) \geq t$. Hence it suffices to verify that if $v^I_{\texttt{C}}(d) \geq t$ then $v^I_{\texttt{C}'}(d) \geq t$.

Let k be the number of R_1-successors of d that are in D^I_1.

If $k \leq r$, we can focus specifically on the contributions of $(R_1, D_1) : m_1$ in $v^I_{\texttt{C}}(d)$ and of all $E_i : w'_{p+i}$ in $v^I_{\texttt{C}'}(d)$. The contribution of $(R_1, D_1) : m_1$ in $v^I_{\texttt{C}}(d)$ is $k \cdot m_1$, and (when $k \leq r$), so is the contribution of all $E_i : w'_{p+i}$ in $v^I_{\texttt{C}'}(d)$. Then clearly, $v^I_{\texttt{C}'}(d) = v^I_{\texttt{C}}(d)$, so $v^I_{\texttt{C}'}(d) \geq t$ iff $v^I_{\texttt{C}}(d) \geq t$.

It remains to verify the case of $k > r$. We must show that if $v^I_{\mathtt{C}}(d) \geq t$ then $v^I_{\mathtt{C}'}(d) \geq t$. When $k > r$, the contribution of all $E_i : w'_{p+i}$ in $v^I_{\mathtt{C}'}(d)$ is $r \cdot m_1$; so $v^I_{\mathtt{C}' \geq 0}(d)$ is bounded below by $r \cdot m_1$. Since we only consider teeth with $\vec{m} \in \mathbb{N}^q$ in this section, $v^I_{\mathtt{C}'<0}(d)$ is bounded below by $-\sum_{1 \leq i \leq p} |w'_i|$. So $v^I_{\mathtt{C}'}(d) = v^I_{\mathtt{C}' \geq 0}(d) + v^I_{\mathtt{C}'<0}(d) \geq \left\lceil \frac{t + \sum_{1 \leq i \leq p} |w'_i|}{m_1} \right\rceil \cdot m_1 - \sum_{1 \leq i \leq p} |w'_i|$. So $v^I_{\mathtt{C}'}(d) \geq t$.

Example 3. With $\mathtt{C} = \nabla\!\!\!\!\nabla^3_*\big(C_1 : -2 \mid (R, D) : 1\big)$, we have $r = \lceil (3+2)/1 \rceil = 5$. The rationale is that if an individual is a C_1 it is still sufficient for it to have 5 R-successors that are D for this individual to be a $\mathtt{C}$. If it is not a C_1, then 3, 4, 5 or more R-successors that are D are sufficient.

We get $\mathtt{C}' = \nabla\!\!\!\!\nabla^3_*\big(C_1 : -2, (= 1\, R.D) : 1, (= 2\, R.D) : 2, (= 3\, R.D) : 3, (= 4\, R.D) : 4, (\geq 5\, R.D) : 5 \mid \_\_\big)$. Of course, it is equivalent to the regular tooth $\mathtt{C}'' = \nabla\!\!\!\!\nabla^3\big(C_1 : -2, (= 1\, R.D) : 1, (= 2\, R.D) : 2, (= 3\, R.D) : 3, (= 4\, R.D) : 4, (\geq 5\, R.D) : 5\big)$.

So a counting tooth with $\mathcal{ALC}$ concepts can be transformed into a regular tooth with $\mathcal{ALCQ}$ concepts when only non-negative weights are allowed. In turn, we know how to (efficiently) transform it into an $\mathcal{ALCQ}$ concept [2]. We obtain the following proposition.

Proposition 1. *$\mathcal{ALC}$ with counting teeth with non-negative weights has the same expressivity as $\mathcal{ALCQ}$.*

Preliminary complexity results. In the rewriting above, the size of the tooth strictly grows, as one pair (R_1, D_1) is removed, but r new concepts are added, each of size larger than the combined sizes of R_1 plus D_1. Yet, r is bounded by the threshold t. So, when the threshold is expressed in unary, the rewriting only causes a linear expansion. But if the weights are encoded in binary, this is not an efficient rewriting! It only yields this partial result.

Proposition 2. *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.*

Proof. When deciding whether the concept C is satisfiable wrt. $\mathcal{T}$, (1) if there are nested counting teeth (in $\mathcal{T}$ or C), pick the inner-most (breaking ties at random) tooth concept (T), (2) introduce a fresh concept name $Fresh_T$, (3) repeat 1–2, with $C := C[T/Fresh_T]$ is satisfiable wrt. $\mathcal{T} := \mathcal{T}[T/Fresh_T] \cup \{Fresh_T \equiv T\}$, (where $X[A/B]$ stands for the uniform substitution with B of every occurrence of A in X).

The number of teeth in $\mathcal{T}$ and C is linear in the size of $\mathcal{T}$ and C, so the procedure above terminates in polynomial time, and results in a combined size of the $\mathcal{T}$ and C that are linear in the combined size of $\mathcal{T}$ and C at the start of the procedure.

Now, observe that:

- C is satisfiable wrt. $\mathcal{T}$ iff $C[T/Fresh_T]$ is satisfiable wrt. $\mathcal{T}[T/Fresh_T] \cup \{Fresh_T \equiv T\}$.
- when the procedure halts, there are no more nested teeth in $\mathcal{T}$ and C.

It now suffices to transform all the counting teeth in the resulting $\mathcal{T}$ and C into regular teeth applying iteratively the rewriting proposed above. We obtain $\mathcal{T}$ and C which are now written in $\mathcal{ALCQ}$ equipped with regular teeth. It causes a blow-up in size exponential in the larger threshold of an occurring counting tooth, when represented in binary, and only a polynomial increase when the thresholds are represented in unary.

Finally, using the transformations of [2], eliminating the regular teeth altogether is efficient, and we obtain a problem of deciding the satisfiability of an $\mathcal{ALCQ}$ concept wrt. an $\mathcal{ALCQ}$ TBox, which is ExpTime-complete [9].

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

In Section 4, we failed to establish a precise complexity of $\mathcal{ALC}$ with counting teeth, even restricting our attention to only non-negative weights for roles. Here we embed it into $\mathcal{ALCSCC}^\infty$, showing that the complexity is in ExpTime.

This approach has a practical drawback: $\mathcal{ALCSCC}^\infty$ is not a logic supported by the existing reasoning services, and algorithms fit for implementation do not exist. But it will allow us to pin down the complexity of reasoning in $\mathcal{ALC}$ augmented with counting tooth operators.

Let $\mathtt{C} = \nabla\!\!\!\!\nabla^t_*\big(C_1 : w_1, \ldots, C_p : w_p \mid (R_1, D_1) : w_{p+1}, \ldots, (R_q, D_q) : w_q\big)$. Let us assume for now that $\mathtt{C}$ does not have nested teeth. Let $\mathcal{T}$ be a TBox. Let us assume for now that $\mathcal{T}$ is an $\mathcal{ALC}$ TBox (without teeth).

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

We add a fresh role name zoo_{C_i} ('zero-or-one') adding the axioms $(= 1\, zoo_{C_i}.\top) \equiv C_i$ and $(= 0\, zoo_{C_i}.\top) \equiv \neg C_i$ for every $1 \leq i \leq p$ to $\mathcal{T}$. We obtain the TBox $\mathcal{T}'$.

Now, we define $\mathsf{summands} =$

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

Roughly speaking, $\mathtt{C}$ is the set of individuals such that the sum of the elements of $\mathsf{summands}$ is greater or equal to t. The quantity $|zoo_{C_i} \cap \top|$ will be 1 if the individual is a C_i and 0 if it is not.

But some of these summands could be negative, exactly those where $w_i < 0$, and QFBAPA$^\infty$ does not allow using negative constants. In [11], the authors observe that "Dispensing with negative constants is not really a restriction since we can always write the numerical constraints of QFBAPA in a way that does not use negative integer constants (by bringing negative summands to the other side of a constraint)." We are going to do just that.

When $n \in \mathbb{N}$, we syntactically identify $\sim\!-n$ with n. If $t \geq 0$, then $t_l = t$ and $t_r = 0$, else $t_l = 0$ and $t_r = \sim t$. Now consider the $\mathcal{ALCSCC}$ concept

$$\mathtt{C}' = succ\left(t_l + \sum_{\substack{w_i \cdot x_i \in \mathsf{summands} \\ w_i < 0}} \sim w_i \cdot x_i \leq t_r + \sum_{\substack{w_i \cdot x_i \in \mathsf{summands} \\ w_i \geq 0}} w_i \cdot x_i\right) \ .$$

(In order to be totally rigorous, $T_1 \leq T_2$ represents the QFBAPA$^\infty$ formula $(T_1 < T_2) \vee (T_2 = T_1)$.)

Lemma 2. $\mathtt{C}$ *is satisfiable in* $\mathcal{T}$ *iff* $\mathtt{C}'$ *is satisfiable in* $\mathcal{T}'$.

We can now do better than Proposition 2.

Proposition 3. *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.*

Proof. Starting from the deepest teeth (in the concept and the TBox), we rewrite them into an $\mathcal{ALCSCC}$ concept as above, adding *zoo* role axioms into the TBox as we go. Those are a series of polynomial transformations, all equi-satisfiable. The result follows because reasoning in $\mathcal{ALCSCC}$ can be done in ExpTime [6].

6 Teeth that count more

In the previous extended family of teeth, the additional weights are used as multiplying factor to the number of features reached by a role.

We could envisage to replace these weights with arbitrary functions f, from $\mathbb{N}$ (or $\mathbb{N} \cup \{\infty\}$) to $\mathbb{R}$. A feature with weight w could be represented by a feature with function $f(n) = w \cdot n$.

Now, although irremediably leaving the cosy realm of linearity, f could be anything, e.g., a polynomial, a sigmoid, a binomial distribution, etc.

We define a new collection of perceptron operators

$$\mathtt{C} = \nabla\!\!\!\!\nabla^t_*\big(C_1 : w_1, \ldots, C_p : w_p \mid (R_1, D_1) : f_1, \ldots, (R_q, D_q) : f_q\big) \ ,$$

where $\vec{w} = (w_1, \ldots, w_p) \in \mathbb{R}^p$, $\vec{f} = (f_1, \ldots, f_q)$ is a q-vector of functions from $\mathbb{N}$ to $\mathbb{R}$, R_i and D_i are roles and concepts respectively.

When role-branching is finite (for simplicity of exposition), the *value* of a $d \in \Delta^I$ under a $\mathtt{C}$ is:

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

As illustrated in [2], teeth operators are simply linear classification models, and it is possible to use standard linear classification algorithms (such as the Perceptron Algorithm, Logistic Regression, or Linear SVM) to learn its weights and its threshold given a set of assertions about individuals (ABox). This richer family of operators would offer a new perspective of integrating into ontologies concepts learnt with more advanced algorithms.

7 Conclusions

We extended the tooth with role-successors counting.

When we do not allow negative weights, the extended perceptron operator can faithfully and easily express concepts like ‘compulsory imprisonment’ from the Florida Score Sheet. When added to $\mathcal{ALC}$, the resulting logic has exactly the same expressivity as $\mathcal{ALCQ}$. We showed how reasoning can be transformed into reasoning in $\mathcal{ALCQ}$, allowing one to straightforwardly use state-of-the-art reasoning services for $\mathcal{ALCQ}$. When the threshold is expressed in unary, it is no more succinct. Whether it is more succinct than $\mathcal{ALCQ}$ when the threshold is expressed in binary remains an open question.

When we allow negative weights, the extended perceptron operator can express concepts like “has more sons than daughters”, “has as many arms than legs”, ... When added to $\mathcal{ALC}$ it yields a $\mathcal{DL}$ that is strictly more expressive than $\mathcal{ALCQ}$, and is not anymore a fragment of FOL.

The complexity of the $\mathcal{DL}$ that is obtained by adding the extended perceptron operator to $\mathcal{ALC}$ is ExpTime-complete, no matter whether we allow negative weights or not, or whether the threshold is represented in unary or binary.

References

1. D. Porello, O. Kutz, G. Righetti, N. Troquard, P. Galliani, and C. Masolo. A toothful of concepts: Towards a theory of weighted concept combination. In *Proceedings of the 32nd International Workshop on Description Logics, Oslo, Norway, June 18-21, 2019*, 2019.
2. P. Galliani, G. Righetti, O. Kutz, D. Porello, and N. Troquard. Perceptron connectives in knowledge representation. In *Knowledge Engineering and Knowledge Management - 22nd International Conference, EKAW 2020, Bolzano, Italy, September 16-20, 2020, Proceedings*, pages 183–193, 2020.
3. F. Baader, G. Brewka, and O. F. Gil. Adding threshold concepts to the description logic $\mathcal{EL}$. In C. Lutz and S. Ranise, editors, *Frontiers of Combining Systems*, pages 33–48, Cham, 2015. Springer International Publishing.
4. F. Baader and U. Sattler. Expressive number restrictions in description logics. *J. Log. Comput.*, 9(3):319–350, 1999.
5. H. J. Ohlbach and J. Koehler. Modal logics, description logics and arithmetic reasoning. *Artificial Intelligence*, 109(1):1–31, 1999.
6. F. Baader. A new description logic with set constraints and cardinality constraints on role successors. In C. Dixon and M. Finger, editors, *Frontiers of Combining Systems - 11th International Symposium, FroCoS 2017, Brasília, Brazil, September 27-29, 2017, Proceedings*, volume 10483 of *Lecture Notes in Computer Science*, pages 43–59. Springer, 2017.
7. V. Kuncak and M. C. Rinard. Towards Efficient Satisfiability Checking for Boolean Algebra with Presburger Arithmetic. In F. Pfenning, editor, *Automated Deduction - CADE-21, 21st International Conference on Automated Deduction, Bremen, Germany, July 17-20, 2007, Proceedings*, volume 4603 of *Lecture Notes in Computer Science*, pages 215–230. Springer, 2007.

8. K. Schild. A correspondence theory for terminological logics: Preliminary report. In J. Mylopoulos and R. Reiter, editors, *Proceedings of the 12th International Joint Conference on Artificial Intelligence. Sydney, Australia, August 24-30, 1991*, pages 466–471. Morgan Kaufmann, 1991.
9. S. Tobies. The complexity of reasoning with cardinality restrictions and nominals in expressive description logics. *J. Artif. Intell. Res.*, 12:199–217, 2000.
10. F. Baader and A. Ecke. Extending the description logic ALC with more expressive cardinality constraints on concepts. In C. Benzmüller, C. L. Lisetti, and M. Theobald, editors, *GCAI 2017, 3rd Global Conference on Artificial Intelligence, Miami, FL, USA, 18-22 October 2017*, volume 50 of *EPiC Series in Computing*, pages 6–19. EasyChair, 2017.
11. F. Baader and F. De Bortoli. On the expressive power of description logics with cardinality constraints on finite and infinite sets. In A. Herzig and A. Popescu, editors, *Frontiers of Combining Systems*, pages 203–219, Cham, 2019. Springer International Publishing.
12. F. Baader, B. Bednarczyk, and S. Rudolph. Satisfiability and query answering in description logics with global and local cardinality constraints. In G. D. Giacomo, A. Catalá, B. Dilkina, M. Milano, S. Barro, A. Bugarín, and J. Lang, editors, *ECAI 2020 - 24th European Conference on Artificial Intelligence, 29 August-8 September 2020, Santiago de Compostela, Spain, August 29 - September 8, 2020 - Including 10th Conference on Prestigious Applications of Artificial Intelligence (PAIS 2020)*, volume 325 of *Frontiers in Artificial Intelligence and Applications*, pages 616–623. IOS Press, 2020.
13. F. Baader, I. Horrocks, C. Lutz, and U. Sattler. *An Introduction to Description Logic*. Cambridge University Press, 2017.
14. P. Galliani, O. Kutz, D. Porello, G. Righetti, and N. Troquard. On knowledge dependence in weighted description logic. In *GCAI 2019. Proceedings of the 5th Global Conference on Artificial Intelligence, Bozen/Bolzano, Italy, 17-19 September 2019*, pages 68–80, 2019.