Proceedings of the 20th International Conference on Principles of Knowledge Representation and Reasoning
Main Track

Succinctness and Complexity of $\mathcal{ALC}$ with Counting Perceptrons

Pietro Galliani[1] , **Oliver Kutz**[2] , **Nicolas Troquard**[2]
[1]University of Insubria, Italy
[2]Free University of Bozen-Bolzano, Italy
pietro.galliani@uninsubria.it, {oliver.kutz, nicolas.troquard}@unibz.it

Abstract

Perceptron operators have been introduced to knowledge representation languages such as description logics in order to define concepts by listing features with associated weights and by giving a threshold. Semantically, an individual then belongs to such a concept if the weighted sum of the listed features it belongs to reaches that threshold. Such operators have been subsequently applied to cognitively-motivated modelling scenarios and to building bridges between learning and reasoning. However, they suffer from the basic limitation that they cannot consider the weight or number of role fillers. This paper introduces an extension of the basic perceptron operator language to address this shortcoming, defining the language $\mathcal{ALCP}$ and answering some basic questions regarding the succinctness and complexity of the new language. Namely, we show firstly that in $\mathcal{ALCP}^+$, when weights are positive, the language is expressively equivalent to $\mathcal{ALCQ}$, whilst it is strictly more expressive in the general case allowing also negative weights. Secondly, $\mathcal{ALCP}^+$ is shown to be strictly more succinct than $\mathcal{ALCQ}$. Thirdly, capitalising on results concerning the logic $\mathcal{ALCSCC}$, we show that despite the added expressivity, reasoning in $\mathcal{ALCP}$ remains ExpTime-complete.

1 Introduction

The *Felony Score Sheet* used in the State of Florida (Florida Department of Corrections and Florida Office of the State Courts Administrator 2019), 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. As discussed below in detail, this kind of example necessitates an extension to the knowledge representation languages with threshold operators introduced by Porello et al. (2019).[1] The basic language is as follows: If $C_1, \ldots, C_n$ are concept expressions, $w_1, \ldots, w_n \in \mathbb{Z}$ are weights for the individual concepts, and $t \in \mathbb{Z}$ is a threshold, we can introduce a new concept

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

whose extension in a given interpretation $I = (\Delta^I, \cdot^I)$ is the set of the individuals $d \in \Delta^I$ such that:

$$\sum_{i=1}^{n} \{w_i \mid d \in C_i^I\} \geq t \ .$$

We call it the *threshold* or *perceptron* operator, or "tooth". Righetti, Porello, and Confalonieri (2022) verified, in particular, that perceptron operators are generally more interpretable than DNFs, and that they are perceived as more understandable by users who are less familiar with logic. They have been used, e.g., in learning scenarios (Galliani et al. 2020) and in modelling concept combination problems (Righetti et al. 2019). Therefore, designing description logic languages that support the tooth has the potential to help general practitioners to write ontologies in a more accessible and agile way.

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 combination of $\mathcal{ALC}$ concepts that suitably unfolds the tooth into a DNF, as shown by Porello et al. (2019). Furthermore, adding the perceptron operator to $\mathcal{ALC}$ does not affect the complexity of reasoning (Galliani et al. 2020), namely, it is still ExpTime-complete to do reasoning wrt a general TBox: there exists a linear transformation such that any instance of an entailment in $\mathcal{ALC}$ using the $\nabla\!\!\!\nabla$ can be transformed into an equivalent instance of entailment in pure $\mathcal{ALC}$ (although obfuscating the original readability of the concepts defined with $\nabla\!\!\!\nabla$).

Coming back to the opening example, a knowledge base describing the laws of Florida would need to represent the above score sheet as part of the definition of its $\mathsf{CompulsoryImprisonment}$ concept, for instance as

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

However, on closer inspection, the Felony Score Sheet is in fact more subtle. Namely, 18 points are added for *each instance* (i.e. every count) of a 'moderate injury victim'.

Of course we can use one concept $\mathsf{1MI}, \mathsf{2MI}, \mathsf{3MI}, \ldots$ for each number of moderate injuries, and all of them pairwise disjoint and with weights 18, 36, 54, etc. Then we can write

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

[1]This paper is a significantly expanded and revised follow-up to Galliani, Kutz, and Troquard (2021).

Alternatively, with each (i+1)MI a subset of (i)MI, we can use

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

No matter which encoding one would choose, however, 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, resulting in a rather fragile modelling approach.

Is there not a more flexible way to specify these kinds of concepts? The problem at hand is to investigate how to extend the $\mathbb{W}$ operator to accommodate concepts like CompulsoryImprisonment in the Florida example faithfully and with simplicity. In this paper, we extend the regular tooth with the needed role-successor counting abilities, and study the problems of expressivity, succinctness and complexity of the resulting language.

Related work. There is a number of related proposals for knowledge representation languages capable to represent classification tasks symbolically using 'hybrid' and 'counting' methods. Weighted knowledge has also been studied in the context of multipreference semantics and defeasibility, but on the level of knowledge bases, which addresses therefore a different level of knowledge representation than the description logic languages studied in this paper (Giordano and Dupré; Giordano and Dupré 2022; 2021). Baader, Brewka, and Fernandez Gil (2015) introduce a description logic (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.

Baader and Sattler (1999) introduce 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 DL. Ohlbach and Koehler (1999) present a general method to use arithmetic reasoning as part of the inference engine of description logics. Useful counting operators can then be devised and integrated into DL, and remain decidable.

Baader (2017) introduces the extension $\mathcal{ALCSCC}$ of $\mathcal{ALC}$ with *expressive* statements of constraints on role-successors (formulas of quantifier-free Boolean algebra with Presburger arithmetic (QFBAPA) (Kuncak and Rinard 2007).[2] 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 $\mathcal{ALC}$ (Schild 1991) or $\mathcal{ALCQ}$ (Tobies 2000)). Baader and Ecke (2017) extend $\mathcal{ALC}$ with *global* expressive cardinality constraints. $\mathcal{ALCQ}$ with global constraints was already studied by Tobies (2000). Adding global cardinality constraints to $\mathcal{ALC}$ leads to NEXPTIME-complete complexity for reasoning tasks in general. In $\mathcal{ALCSCC}$, the interpretations are restricted to finite-branching roles. Baader and Bortoli (2019) introduce $\mathcal{ALCSCC}^\infty$ over arbitrary models; the complexity of reasoning is unaffected. Baader, Bednarczyk, and Rudolph (2020), finally, show that combining the local expressive cardinality constraints of Baader (2017) with the global expressive cardinality constraints of Baader and Ecke (2017) does not impact the complexity.

[2] Here, '$\mathcal{SCC}$' stands for 'Set and Cardinality Constraints'.

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 is sufficient 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. Having established that $\mathcal{ALC}$ with positive counting teeth has the same expressivity as $\mathcal{ALCQ}$, we show in Section 5 that it is more succinct. Section 6 provides an embedding into $\mathcal{ALCSCC}^\infty$, showing that $\mathcal{ALC}$ equipped with the new perceptron operator is EXPTIME-complete in general. 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 the textbook by Baader et al. (2017) for a general introduction of DL.

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

QFBAPA over finite integers is presented by Kuncak and Rinard (2007), and it is extended with infinity by Baader and Bortoli (2019). 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 then a Boolean combination of set constraints A_B and numerical constraints 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 B are obtained by applying intersection, union, and complement to set variables and constants $\emptyset$ and $\mathcal{U}$. Set constraints A_B are of the form $B_1 = B_2$ and $B_1 \subseteq B_2$, where B_1, B_2 are set terms of type 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 x. 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 elements of $\mathbb{N} \cup \{\infty\}$ as follows: 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). Substitutions then also assign elements of $\mathbb{N} \cup \{\infty\}$ to PA variables k. 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 of d. We define $ARS^I(d) = \bigcup_{R \in N_R} R^I(d)$ as the set of all role-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 if $d \in (succ(F))^I$, $\mathcal{U}$ is evaluated as $ARS^I(d)$, the roles R 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 if and only if 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 $succ(|\mathsf{causes}| < 2)$, 2 is an integer constant (also a PA expression),* causes *is a role, but also a set term, $|\mathsf{causes}|$ is a set cardinality (also a PA expression), and $|\mathsf{causes}| < 2$ is a numerical constraint.*

To decide if $d \in (succ(|\mathsf{causes}| < 2))^I$, we use the substitution σ with $\sigma(2) = 2$, and $\sigma(\mathsf{causes}) = \mathsf{causes}^I(d)$.

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

Suppose that we also have $\mathsf{Injury}^I = \{d_2, d_3, d_4\}$. To decide if $d \in (succ(|\mathsf{causes} \cap \mathsf{Injury}| = 1))^I$, Injury *is a concept description but also a set term, and we build the substitution σ' such that $\sigma'(1) = 1$, $\sigma'(\mathsf{causes}) = \mathsf{causes}^I(d) = \{d_1, d_2\}$, $\sigma'(\mathsf{Injury}) = \mathsf{Injury}^I \cap ARS^I(d) = \{d_2, d_3\}$, $\sigma'(\mathsf{causes} \cap \mathsf{Injury}) = \sigma'(\mathsf{causes}) \cap \sigma'(\mathsf{Injury}) = \{d_2\}$, $\sigma'(|\mathsf{causes} \cap \mathsf{Injury}|) = 1$, and $\sigma'(|\mathsf{causes} \cap \mathsf{Injury}| = 1) = true$. Hence σ' is a solution of the QFBAPA$^\infty$ formula $|\mathsf{causes} \cap \mathsf{Injury}| = 1$. So $d \in (succ(|\mathsf{causes} \cap \mathsf{Injury}| = 1))^I$.*

$\mathcal{ALC}$ and $\mathcal{ALCQ}$. The *Qualified cardinality restrictions* of $\mathcal{ALCQ}$ are standardly defined as:

$$(\leq n\, R.C)^I = \big\{d \in \Delta^I \mid \big|\{c \in \Delta^I, (d,c) \in R^I \wedge c \in C^I\}\big| \leq n\big\}$$

and

$$(\geq n\, R.C)^I = \big\{d \in \Delta^I \mid \big|\{c \in \Delta^I, (d,c) \in R^I \wedge c \in C^I\}\big| \geq n\big\}$$

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

$\mathcal{ALCQ}$ can equivalently be described as 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 $n \in \mathbb{N}$. Also, $\mathcal{ALC}$ can be seen as the fragment of $\mathcal{ALCSCC}^\infty$ such that $succ(F)$ is of the form $succ(|R \cap C| \geq 1)$. Namely, we can now define $\exists R.C = succ(|R \cap C| \geq 1)$ and proof the semantic equivalence, and likewise for $(\leq n\, R.C) = succ(|R \cap C| \leq n)$ and $(\geq n\, R.C) = succ(|R \cap C| \geq n)$.

3 $\mathcal{ALCP}$. Counting Teeth: Syntax and Semantics

Syntax of $\mathcal{ALCP}$. We define a new collection of perceptron operators, that we call simply *counting teeth*:

$$\mathsf{C} = \nabla\!\!\!\!\nabla^t_{\odot}(C_1 : w_1, \ldots, C_p : w_p \mid (R_1, D_1) : m_1, \ldots, (R_q, D_q) : m_q) \ , \quad (1)$$

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_j are concept expressions, and R_j are roles.

The tooth from our previous work, without role-successor counting is sometimes called a *regular tooth*. Note that a regular tooth is just a counting tooth where $q = 0$.

The language of the new description logic $\mathcal{ALCP}$, i.e. the basic logic of counting teeth, is just like $\mathcal{ALC}$, but enriched

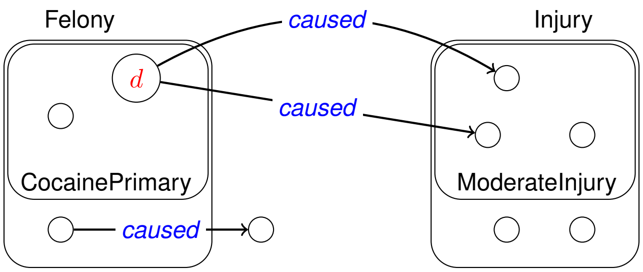

Figure 1: A simple model in the Florida Score Sheet domain.

with the new concept forming operator defined in (1), where the concepts C_i, D_j are arbitrary $\mathcal{ALCP}$ concepts. Note that this allows arbitrary nesting of counting teeth.

Example 2. *With* caused *a role, we can now faithfully (partially) define the concept* CompulsoryImprisonment *of the Felony Score Sheet:*

$$\mathbb{W}_{\odot}^{44}\big(\mathsf{CocainePrimary} : 16, \cdots \mid (\mathit{caused}, \mathsf{ModerateInjury}) : 18, \dots\big) \ .$$

Semantics. Let us discuss the semantics of counting teeth first with a simple case where all the weights are positive.

Recall that for the regular tooth, to determine whether an individual d belongs to a tooth concept C in interpretation I we compute its *value* $v_{\mathsf{C}}^I(d)$ and verify if it reaches the threshold. We could extend this value to the counting perceptron, by simply adding a weighted sum as follows:

$$v_{\mathsf{C}}^I(d) = \sum_{i \in \{1,\dots,p\}} \{w_i \mid d \in C_i^I\} + \sum_{i \in \{1,\dots,q\}} \left(m_i \cdot \left|\{c \in \Delta^I \mid (d,c) \in R_i^I \wedge c \in D_i^I\}\right|\right) \ .$$

Example 3. *Consider the model depicted in Figure 1. We have* $d \in \mathsf{CocainePrimary}^I$*, and* d *has two successors through the role* caused *that are in* $\mathsf{ModerateInjury}^I$*. Using the partial definition of* CompulsoryImprisonment*, we have*

$$v_{\mathsf{CompulsoryImprisonment}}^I(d) = 16 + 18 \times 2 + \dots = 52 + \dots$$

The Felony Score Sheet contains only positive weights, so: $v_{\mathsf{CompulsoryImprisonment}}^I(d) \geq 44$*. Hence, we can conclude that:*

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

When the individual d can have an infinite number of successors and for some (R_i, D_i) the weight is positive and for some other (R_j, D_j) it is negative, then $v_{\mathsf{C}}^I(d)$ would be ill-defined when just adding up the weights simpliciter. I.e., what should the value be of $\infty - \infty$ in this situation, and when would it reach the threshold?

To circumvent this problem, instead of a single value $v_{\mathsf{C}}^I(d)$ for a general tooth, we introduce two values: the **reward** $v_{\mathsf{C}\geq 0}^I(d)$, which represents the sum of the non-negative summands, and the **penalty** $v_{\mathsf{C}<0}^I(d)$, which represents the sum of the negative summands.

$$v_{\mathsf{C}\geq 0}^I(d) = \sum_{\substack{i \in \{1,\dots,p\} \\ w_i \geq 0}} \{w_i \mid d \in C_i^I\} + \sum_{\substack{i \in \{1,\dots,q\} \\ m_i \geq 0}} \left(m_i \cdot \left|\{c \in \Delta^I \mid (d,c) \in R_i^I \wedge c \in D_i^I\}\right|\right) \ .$$

$$v_{\mathsf{C}<0}^I(d) = \sum_{\substack{i \in \{1,\dots,p\} \\ w_i < 0}} \{w_i \mid d \in C_i^I\} + \sum_{\substack{i \in \{1,\dots,q\} \\ m_i < 0}} \left(m_i \cdot \left|\{c \in \Delta^I \mid (d,c) \in R_i^I \wedge c \in D_i^I\}\right|\right) \ .$$

Clearly $v_{\mathsf{C}\geq 0}^I(d) \geq 0$ and $v_{\mathsf{C}<0}^I(d) \leq 0$.

Finally, the semantics of a counting tooth C in an (possibly infinite-branching) interpretation I, and extending the definition of the regular tooth as given by Porello et al. (2019), is:

Definition 1 (Semantics of Counting Tooth)**.**

$$\mathsf{C}^I = \left\{d \in \Delta^I \mid v_{\mathsf{C}\geq 0}^I(d) \geq t - v_{\mathsf{C}<0}^I(d)\right\} \ .$$

In the case that $v_{\mathsf{C}\geq 0}^I(d) = \infty$ and $v_{\mathsf{C}<0}^I(d) = -\infty$, we have that $v_{\mathsf{C}\geq 0}^I(d) \geq t - v_{\mathsf{C}<0}^I(d)$ is equivalent to $\infty \geq t + \infty$, with t a finite integer. Because $\infty = t + \infty$, the inequality holds and the statement is true. In particular, C^I in Definition 1 is well-defined.

Example 4. *For purposes of illustration we define a 'Modified Compulsory Imprisonment' as* $\mathsf{MCI} =$

$$\mathbb{W}_{\odot}^{44}\big(\mathsf{CocainePrimary} : 16 \mid (\mathit{caused}, \mathsf{ModerateInjury}) : 18), (\mathit{preventiveDetention}, \mathsf{Month}) : -1) \ ,$$

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

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

4 Embedding $\mathcal{ALCP}^+$ into $\mathcal{ALCQ}$

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. In particular, in the case of the regular tooth, negative weights on concepts can be transformed efficiently and equivalently into a regular tooth with only non-negative weights on concepts, as shown by Galliani et al. (2019, Cor. 1). This, however, is not the case for counting teeth as allowing negative weights strictly increases the expressive power, as shown below in Proposition 3.

Therefore, let us first restrict our attention to the case where all weights are natural numbers, and thus positive:

$$\mathbb{W}^t_\otimes\big(C_1 : w_1, \ldots, C_p : w_p \mid (R_1, D_1) : m_1, \ldots, (R_q, D_q) : m_q\big) \quad (2)$$

where weights $\vec{w} \in \mathbb{Z}^{+p}, \vec{m} \in \mathbb{Z}^{+q}$ are all natural numbers, is called a *positive counting tooth*.[3]

We show that the language $\mathcal{ALC}$ equipped with positive counting teeth $\mathbb{W}_\otimes$, called $\mathcal{ALCP}^+$, has the same expressivity as $\mathcal{ALCQ}$ (Proposition 2). As a corollary, we obtain that $\mathcal{ALCP}^+$ is strictly less expressive than $\mathcal{ALCP}$ (Proposition 3). Further, we show that concept satisfiability can be decided in 2EXPTIME when the threshold is expressed in binary and in EXPTIME when it is expressed in unary (Proposition 4). We will improve upon the 2EXPTIME upper-bound in Section 6. 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 positive weights into a problem of reasoning with $\mathcal{ALCQ}$, for which efficient reasoning tools already exist.

Iterated elimination of role-successors counting. Consider the positive counting tooth

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

where all $w_i \in \vec{w}$ and all $m_j \in \vec{m}$ are natural numbers. Now define the positive counting tooth $rwt_\otimes(\mathsf{C})$, in which the first occurrence of role counting has been rewritten, as:

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

where:

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

[3] We here assume that $0 \notin \mathbb{N}$; weights of value zero could be trivially allowed, but it has no impact on semantics or results.

Lemma 1.

$$(\mathsf{C})^I = (rwt_\otimes(\mathsf{C}))^I \ .$$

Proof. Let k_i be the number of R_i-successors of d that are in D_i. That is,

$$k_i = \left|\{c \in \Delta^I \mid (d, c) \in R_i^I \wedge c \in D_i^I\}\right| \ .$$

Let $\alpha = \sum_{i=1}^p \{w_i \mid d \in C_i^I\}$ and let $\beta = \sum_{i=2}^q k_i \cdot m_i$. We thus have,

$$v^I_{\mathsf{C}}(d) = \alpha + (k_1 \cdot m_1 + \beta) \ .$$

All weights in C and $rwt_\otimes(\mathsf{C})$ are positive, so it suffices to show that for $d \in \Delta^I$, we have $v^I_{\mathsf{C}}(d) \ge t$ iff $v^I_{rwt_\otimes(\mathsf{C})}(d) \ge t$.

- Case $k_1 = 0$. It means that d is in none of the E^I_j. So $v^I_{rwt_\otimes(\mathsf{C})}(d) = \alpha + \beta = v^I_{\mathsf{C}}(d)$. So clearly, $v^I_{\mathsf{C}}(d) \ge t$ iff $v^I_{rwt_\otimes(\mathsf{C})}(d) \ge t$.
- Case $1 \le k_1 < r$. It means that d is in $E^I_{k_1}$, and in none of the other E^I_j, $j \ne k_1$. So $v^I_{rwt_\otimes(\mathsf{C})}(d) = (\alpha + w'_{p+k_1}) + \beta = (\alpha + k_1 \cdot m_1) + \beta = v^I_{\mathsf{C}}(d)$. So clearly, $v^I_{\mathsf{C}}(d) \ge t$ iff $v^I_{rwt_\otimes(\mathsf{C})}(d) \ge t$.
- Case $k_1 \ge r$. It means that d is in E^I_r, and in none of the other E^I_j, $j \ne r$. So $v^I_{rwt_\otimes(\mathsf{C})}(d) = (\alpha + r \cdot m_1) + \beta = (\alpha + \left\lceil \frac{t}{m_1} \right\rceil \cdot m_1) + \beta$. Thus, $v^I_{rwt_\otimes(\mathsf{C})}(d) \ge t$.
  We must show that also $v^I_{\mathsf{C}}(d) \ge t$. By definition, $v^I_{\mathsf{C}}(d) = \alpha + (k_1 \cdot m_1 + \beta)$. But since $k_1 \ge r$, we have:

$$\begin{aligned} v^I_{\mathsf{C}}(d) &\ge \alpha + (r \cdot m_1 + \beta) \\ &= \alpha + (\left\lceil \frac{t}{m_1} \right\rceil \cdot m_1 + \beta) \\ &\ge \alpha + (t + \beta) \ge t \ . \end{aligned}$$

□

Example 5. *With* $\mathsf{C} = \mathbb{W}^9_\otimes(C_1 : 3 \mid (R, D) : 2)$*, we have* $r = \lceil 9/2 \rceil = \lceil 4.5 \rceil = 5$*. The rationale is that it is sufficient for an individual to have* 5 R*-successors that are* D *for this individual to be a* C*. This is independent of whether that individual is a* C_1 *or not. We get*

$$rwt_\otimes(\mathsf{C}) = \mathbb{W}^9_\otimes(C_1 : 3, (= 1\, R.D) : 2, (= 2\, R.D) : 4, (= 3\, R.D) : 6, (= 4\, R.D) : 8, (\ge 5\, R.D) : 10 \mid -\big) \ .$$

Of course, it is equivalent to the regular tooth

$$\mathbb{W}^9\big(C_1 : 3, (= 1\, R.D) : 2, (= 2\, R.D) : 4, (= 3\, R.D) : 6, (= 4\, R.D) : 8, (\ge 5\, R.D) : 10\big) \ .$$

Expressivity of $\mathcal{ALCP}$. So, a counting tooth with $\mathcal{ALC}$ concepts can be transformed into a regular tooth with $\mathcal{ALCQ}$ concepts whenever only positive weights are allowed. In turn, we can transform it into an equivalent DNF as shown by Porello et al. (2019), i.e. into a Boolean combination of $\mathcal{ALCQ}$ concepts which is again an $\mathcal{ALCQ}$ concept. This DNF might be exponentially larger, however. We obtain the following proposition.

Proposition 2. *$\mathcal{ALCP}^+$, i.e. $\mathcal{ALC}$ with positive counting teeth, has the same expressivity as $\mathcal{ALCQ}$.*

Proof. This is a consequence of Lemma 1, and of the simple observation, that counting quantifiers can be represented as:

$$(\geq t\, R.C)^I = (\mathbb{W}^t_\otimes(\,—\mid (R, C) : 1))^I \ . \qquad \square$$

Let us observe that $\mathcal{ALC}$ equipped with counting teeth restricted to positive weights is strictly less expressive than when negative weights are allowed.

Example 6. *One can express "has as many sons as daughters":* $\mathsf{AsMany} =$

$$\mathbb{W}^0_\odot(\,—\mid (\textit{isParentOf}, \mathsf{Boy}) : 1, (\textit{isParentOf}, \mathsf{Girl}) : -1) \sqcap$$
$$\mathbb{W}^0_\odot(\,—\mid (\textit{isParentOf}, \mathsf{Girl}) : 1, (\textit{isParentOf}, \mathsf{Boy}) : -1) \ .$$

This cannot be expressed in $\mathcal{ALCQ}$ (Baader 2017, Lemma 2), but $\mathcal{ALC}$ equipped with counting teeth with positive weights has the same expressivity as $\mathcal{ALCQ}$ (Proposition 2).

Proposition 3. *$\mathcal{ALCP}$ is strictly more expressive than $\mathcal{ALCP}^+$.*

In contrast, we recall that, adding the regular tooth of Galliani et al. (2020) to $\mathcal{ALC}$ results in logics of the same expressivity regardless of whether the weights are possibly negative or not.

Intermediate 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, the rewriting causes an exponential blowup. This yields the following partial result.

Proposition 4. *Satisfiability of $\mathcal{ALCP}^+$ concepts with positive counting teeth wrt 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.

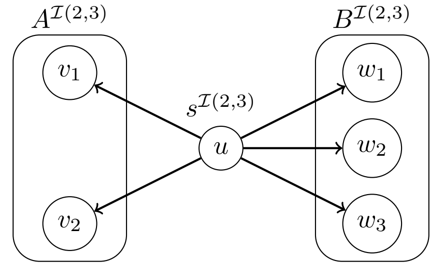

Figure 2: A graphical representation of $\mathcal{I}(2, 3)$.

It now suffices to transform all the counting teeth in the resulting $\mathcal{T}$ and C into regular teeth applying iteratively the rewriting $rwt_\otimes()$ proposed above. Further, by Lemma 1, each rewriting yields an equivalent concept. 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 largest 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 Galliani et al. (2020), 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 (Tobies 2000). $\square$

5 Succinctness of $\mathcal{ALCP}^+$ wrt $\mathcal{ALCQ}$

We have established in Proposition 2 that $\mathcal{ALC}$ with positive counting teeth has the same expressivity as $\mathcal{ALCQ}$. A natural question to ask then is whether it is more succinct. In this section, we prove that $\mathcal{ALC}$ with positive counting teeth is more succinct than $\mathcal{ALCQ}$.

Definition 2. *For any interpretation $\mathcal{I}$, elements $c, d \in \Delta^{\mathcal{I}}$, relation symbols R and $\mathcal{ALCQ}$ concepts C, we say that d is a (R, C)-successor of c if*

1. *$(c, d) \in R^{\mathcal{I}}$ (that is, d is an R-successor of c);*
2. *$d \in C^{\mathcal{I}}$.*

Definition 3. *Let $n, n' \in \mathbb{N}$. Then let $\mathcal{I}(n, n')$ be the interpretation with domain of discourse $\Delta^{\mathcal{I}(n,n')} = \{u, v_1 \ldots v_n, w_1 \ldots w_{n'}\}$, with a named individual s such that $s^{\mathcal{I}(n,n')} = u$, with one edge relation R such that $R^{\mathcal{I}(n,n')} = \{(u, v_i) : i \in 1 \ldots n\} \cup \{(u, w_j) : j \in 1 \ldots n'\}$, and with two atomic concepts A, B with $A^{\mathcal{I}(n,n')} = \{v_1 \ldots v_n\}$ and $B^{\mathcal{I}(n,n')} = \{w_1 \ldots w_{n'}\}$.*

Figure 2 shows a graphical representation of $\mathcal{I}(2, 3)$. It is easily verified that all the individuals of $\mathcal{I}(n, n')$ that satisfy A (or B) satisfy exactly the same $\mathcal{ALCQ}$ concepts, even across different choices of n and n':

Lemma 5. *Let C be any $\mathcal{ALCQ}$ concept built out of the named individual s, the edge relation R, and the two atomic concepts A and B, and let $n_0, n_0' \in \mathbb{N}$. Then*

1. *If $C^{\mathcal{I}(n_0,n_0')} \cap A^{\mathcal{I}(n_0,n_0')} \neq \emptyset$ then $C^{\mathcal{I}(n,n')} \cap A^{\mathcal{I}(n,n')} = A^{\mathcal{I}(n,n')}$ for all $n, n' \in \mathbb{N}$;*

2. *If* $C^{\mathcal{I}(n_0,n_0')} \cap B^{\mathcal{I}(n_0,n_0')} \neq \emptyset$ *then* $C^{\mathcal{I}(n,n')} \cap B^{\mathcal{I}(n,n')} = B^{\mathcal{I}(n,n')}$ *for all* $n, n' \in \mathbb{N}$.

Proof. Suppose that $C^{\mathcal{I}(n_0,n_0')} \cap A^{\mathcal{I}(n_0,n_0')} \neq \emptyset$. Then there exists some element $v_0 \in \Delta^{\mathcal{I}(n_0,n_0')}$ such that $v_0 \in C^{\mathcal{I}(n_0,n_0')}$ and $v_0 \in A^{\mathcal{I}(n_0,n_0')}$. Now consider any $v \in A^{\mathcal{I}(n,n')}$. Both v_0 and v satisfy the same atomic predicates (that is, A but not B) and have no descendants, so if $v_0 \in C^{\mathcal{I}(n_0,n_0')}$ it must be the case that $v \in C^{\mathcal{I}(n_0,n_0')}$ (Note: this would not work if our logic had e.g. inverse roles). Therefore $v \in C^{\mathcal{I}(n,n')} \cap A^{\mathcal{I}(n,n')}$, as required.

The case for $C^{\mathcal{I}(n_0,n_0')} \cap B^{\mathcal{I}(n_0,n_0')} \neq \emptyset$ is analogous. □

For convenience, in the remainder of this section we will often use the alternative notation $I \models C(d)$ instead of the usual DL notation $d \in C^I$.

Proposition 6. *Let* $k \in \mathbb{N}$. *Then we obtain that*

$$\mathcal{I}(n,n') \models \mathbb{W}^k_{\otimes}\big(- \mid (R,A)\colon +2, (R,B)\colon +1\big)(s)$$

if and only if $2n + n' \geq k$.

Proof. Observe that $s^{\mathcal{I}(n,n')}$ has n (R,A)-successors and n' (R,B)-successors. Therefore, the value of the threshold expression is $2n + n'$, and the expression is satisfied in s if and only if $2n + n' \geq k$. □

Corollary 7. *Let* $i, k \in \mathbb{N}$ *be such that* $2i < k$. *Then*

$$\mathcal{I}(i, k-2i) \models \mathbb{W}^k_{\otimes}\big(- \mid (R,A)\colon +2, (R,B)\colon +1\big)(s)$$

but

$$\mathcal{I}(i-1, k-2i+1) \not\models \mathbb{W}^k_{\otimes}\big(- \mid (R,A)\colon +2, (R,B)\colon +1\big)(s)$$

Proof. This follows from the previous proposition and the fact that $2i + k - 2i = k \geq k$ but $2(i-1) + k - 2i + 1 = k - 1 < k$. □

We will now show that any $\mathcal{ALCQ}$-concept C that is equivalent to the tooth concept $\mathbb{W}^k_{\otimes}(- \mid (R,A)\colon{+2}, (R,B)\colon{+1})$ must contain many (for some choice of "many" that will grow linearly with k) counting quantifiers. To simplify the proof, let us recall first that we can always assume that C contains only $\geq$ quantifiers:

Lemma 8. *For all* $\mathcal{ALCQ}$ *concepts* C*, roles* R *and positive integers* n*,*

- $> nR.C \equiv\ \geq (n+1)R.C$;
- $< nR.C \equiv \neg(\geq nR.C)$;
- $\leq nR.C \equiv \neg(\geq (n+1)R.C)$.

Corollary 9. *Every* $\mathcal{ALCQ}$ *concept* C *is equivalent to some concept* $C_{\geq}$ *that contains only* $\geq$ *quantifiers, and furthermore that contains as many occurrences of quantifiers as* C.

The next proposition provides the main argument we need to prove the succinctness of positive teeth:

Proposition 10. *Let* $i, k \in \mathbb{N}$ *be such that* $i > 1$ *and* $2i < k$*, and let* C *be an* $\mathcal{ALCQ}$ *concept over the signature* $\{s, A, B, R\}$ *in which only* $\geq$ *quantifiers appear and in which the integers* i *and* $k - 2i + 1$ *do not appear in any quantifier. Then*

$$\mathcal{I}(i, k-2i) \models C(s) \text{ iff } \mathcal{I}(i-1, k-2i+1) \models C(s).$$

Proof. We prove this by structural induction on the structure of C.

- If C is an atomic concept (i.e. A or B) then by the definition of $\mathcal{I}(n,n')$ we have that $\mathcal{I}(i, k-2i) \not\models C(s)$ and $\mathcal{I}(i-1, k-2i+1) \not\models C(s)$, and there is nothing to prove.
- If C is of the form $\neg C'$,

$$\begin{aligned}\mathcal{I}(i,k-2i) \models C(s) &\Leftrightarrow \mathcal{I}(i,k-2i) \not\models C'(s)\\ &\Leftrightarrow \mathcal{I}(i-1,k-2i+1) \not\models C'(s)\\ &\quad\text{(by induction hypothesis)}\\ &\Leftrightarrow \mathcal{I}(i-1,k-2i+1) \models C(s).\end{aligned}$$

- If C is of the form $C_1 \sqcap C_2$ $[C_1 \sqcup C_2]$,

$$\begin{aligned}\mathcal{I}(i,k-2i) \models C(s) &\Leftrightarrow \mathcal{I}(i,k-2i) \models C_1(s)\\ \text{and [or]}\quad &\ \mathcal{I}(i,k-2i) \models C_2(s)\\ &\Leftrightarrow \mathcal{I}(i-1,k-2i+1) \models C_1(s)\\ \text{and [or]}\quad &\ \mathcal{I}(i-1,k-2i+1) \models C_2(s)\\ &\Leftrightarrow \mathcal{I}(i-1,k-2i+1) \models C(s).\end{aligned}$$

- If C is of the form $\exists R.C'$, suppose $\mathcal{I}(i, k-2i) \models C(s)$. Then at least one R-successor v of $s^{\mathcal{I}(i,k-2i)}$ satisfies C'. But then, by Lemma 5, for any $n, n' \in \mathbb{N}$ *all* (R,A)-successors of $s^{\mathcal{I}(n,n')}$ satisfy C' or *all* (R,B)-successors of $s^{\mathcal{I}(n,n')}$ satisfy C', depending on whether v is a (R,A)-successor or a (R,B)-successor of $s^{\mathcal{I}(i,k-2i)}$. Since in particular in $\mathcal{I}(i-1, k-2i+1)$ the element $s^{\mathcal{I}(i-1,k-2i+1)}$ has at least one (R,A)-successor (because $i > 1$ and so $i - 1 \geq 1$) and at least one (R,B)-successor (because $2i < k$ and so $k - 2i + 1 > 1$) we can conclude that at least one R-successor of $s^{\mathcal{I}(i-1,k-2i+1)}$ satisfies C' and so $\mathcal{I}(i-1, k-2i+1) \models C(s)$.
  Conversely, suppose that $\mathcal{I}(i-1, k-2i+1) \models C(s)$. Then, again by Lemma 5, for all $n, n' \in \mathbb{N}$, all (R,A)-successors of $s^{\mathcal{I}(n,n')}$ satisfy C' or all its (R,B)-successors satisfy C'; and since in particular $\mathcal{I}(i, k-2i)$ has at least one (R,A)-successor (since $i > 1$) and one (R,B)-successor (since $k > 2i$), $\mathcal{I}(i, k-2i) \models C(s)$.
- The case where C is of the form $\forall R.C'$ follows from the existential and negation cases.

The trickier case is the one in which C is of the form $\geq jR.C'$ for some $j \in \mathbb{N}$, $j \notin \{i, k-2i+1\}$. Then we have to consider four subcases, depending on the value of j:

1. $j > \max(i, k-2i+1)$: Suppose that $\mathcal{I}(i, k-2i) \models\ \geq jR.C'(s)$. Then at least j R-successors of s in $\mathcal{I}(i, k-2i)$ satisfy C'; and since in $\mathcal{I}(i, k-2i)$ s has

only i (R, A)-successors and $k - 2i$ (R, B)-successors, it follows that $j \leq i + (k - 2i) = k - i$ and (from Lemma 5 and the fact that at least one (R, A)-successor and one (R, B)-successor of s must satisfy C') that *all* $(i - 1) + (k - 2i + 1) = k - i$ R-successors of s in $\mathcal{I}(i - 1, k - 2i + 1)$ satisfy C' and therefore (since as we just said $j \leq k - i$) that
$\mathcal{I}(i - 1, k - 2i + 1) \models \geq jR.C'(s)$.

Conversely, if $\mathcal{I}(i - 1, k - 2i + 1) \models \geq jR.C'(s)$, at least j R-successors of s in $\mathcal{I}(i - 1, k - 2i + 1)$ must satisfy C'; but since $j > \max(i, k - 2i + 1)$ and in $\mathcal{I}(i - 1, k - 2i + 1)$ s has only $i - 1$ (R, A)-successors and $k - 2i + 1$ (R, B)-successors, again we must have that $j \leq k - i$ and (by Lemma 5) that all $k - i$ R-successors of s in $\mathcal{I}(i, k - 2i)$ satisfy C', and hence $\mathcal{I}(i, k - 2i) \models \geq jR.C'(s)$.

2. $i < j < k - 2i + 1$: Suppose that $\mathcal{I}(i, k - 2i) \models \geq jR.C'(s)$. Then at least j R-successors of s in $\mathcal{I}(i, k - 2i)$ satisfy C'; but since this interpretation has only i (R, A)-successors for s, at least one (R, B)-successor of s in it must satisfy C'. But then, by Lemma 5, *all* $k - 2i + 1$ (R, B)-successors of s in $\mathcal{I}(i - 1, k - 2i + 1)$ satisfy C'; and since $j < k - 2i + 1$, this implies at once that $\mathcal{I}(i - 1, k - 2i + 1) \models \geq jR.C'(s)$.
Conversely, if $\mathcal{I}(i - 1, k - 2i + 1) \models \geq jR.C'(s)$ then, since in $\mathcal{I}(i - 1, k - 2i + 1)$ s has only $i - 1$ (R, A)-successors, at least one (R, B)-successor of it must satisfy C'; and then, by Lemma 5, all $k - 2i$ (R, B)-successors of s in $\mathcal{I}(i, k - 2i)$ satisfy C'. Since $j < k - 2i + 1$, $j \leq k - 2i$, and therefore $\mathcal{I}(i, k - 2i) \models \geq jR.C'(s)$.

3. $k - 2i + 1 < j < i$: This case is similar to the previous one.
Suppose that $\mathcal{I}(i, k - 2i) \models \geq jR.C'(s)$: then at least one (R, A)-successor of s in it must satisfy C', since it has only $k - 2i$ (R, B)-successors and $j > k - 2i + 1$, and therefore by Lemma 5 all $i - 1$ (R, A)-successors of s in $\mathcal{I}(i - 1, k - 2i + 1)$ satisfy C'; and since $j \leq i - 1$, this implies that $\mathcal{I}(i - 1, k - 2i + 1) \models \geq jR.C(s)$.
Conversely, if $\mathcal{I}(i - 1, k - 2i + 1) \models \geq jR.C(s)$, at least one (R, A)-successor of s in $\mathcal{I}(i - 1, k - 2i + 1)$ must satisfy C', since $j > k - 2i + 1$ and therefore its (R, B)-successors are not enough. But then by Lemma 5 all the i (R, A)-successors of s in $\mathcal{I}(i, k - 2i)$ must satisfy C', and since $j < i$ we can conclude that $\mathcal{I}(i, k - 2i) \models \geq jR.C'(s)$ as required.

4. $j < \min(i, k - 2i + 1)$: If $j \leq 0$ there is nothing to prove, because in this case $\geq jR.C'(s)$ is trivially true of all individuals in all interpretations. Therefore, let us suppose that $0 < j < \min(i, k - 2i + 1)$.
Suppose now that $\mathcal{I}(i, k - 2i) \models \geq jR.C'(s)$. Then there exists at least one R-successor of s in $\mathcal{I}(i, k - 2i)$ that satisfies C'. If this R-successor is a (R, A)-successor, again by Lemma 5, all $i - 1$ (R, A)-successors of s in $\mathcal{I}(i - 1, k - 2i + 1)$ satisfy C'; and if instead it is a (R, B)-successor, all $k - 2i + 1$ (R, B)-successors of s in it will satisfy C'. Since $j < i$ and $j < k - 2i + 1$, in either case we will have that
$\mathcal{I}(i - 1, k - 2i + 1) \models \geq jR.C'(s)$, as required.
Conversely, suppose that $\mathcal{I}(i - 1, k - 2i + 1) \models \geq jR.C'(s)$ for $0 < j < \min(i, k - 2i + 1)$. Then once again, in $\mathcal{I}(i - 1, k - 2i + 1)$ the element named by s will have at least one successor that satisfies C'; if it is a (R, A)-successor, all the i (R, A)-successors of s in $\mathcal{I}(i, k - 2i)$ will satisfy C', and if instead it is a (R, B)-successor all the $k - 2i$ (R, B)-successors of s in it will satisfy C'. Since $j < i$ and $j < k - 2i + 1$, in either case $\mathcal{I}(i, k - 2i) \models \geq jR.C'(s)$, and this concludes the proof.

$\square$

At this point we have all the ingredients required to prove the main result:

Theorem 11. *Let $k \in \mathbb{N}$ and let C be an $\mathcal{ALCQ}$ concept in which only the $\geq$ quantifier appears that is equivalent to $\nabla\!\!\!\nabla^k_{\otimes}(\,—\,|\,(R, A)\colon +2, (R, B)\colon +1)$.*
Then for all $i \in 2 \ldots k/2 - 1$, at least one quantifier among $\geq iR$ and $\geq (k - 2i + 1)R$ must appear in C.

Proof. Suppose that this is not the case for some $i \in 2 \ldots k/2 - 1$. Then, by Proposition 10, either $\mathcal{I}(i, k - 2i) \models C(s)$ and $\mathcal{I}(i - 1, k - 2i + 1) \models C(s)$ or $\mathcal{I}(i, k - 2i) \not\models C(s)$ and $\mathcal{I}(i - 1, k - 2i + 1) \not\models C(s)$.

But then C is not equivalent to our threshold expression, because as stated in Corollary 7

$$\mathcal{I}(i, k - 2i) \models \nabla\!\!\!\nabla^k_{\otimes}\big(\,—\,|\,(R, A) : +2, (R, B)\colon +1\big)(s)$$

but

$$\mathcal{I}(i - 1, k - 2i + 1) \not\models \nabla\!\!\!\nabla^k_{\otimes}\big(\,—\,|\,(R, A) : +2, (R, B)\colon +1\big)(s)$$

$\square$

Corollary 12. *Let C be an $\mathcal{ALCQ}$ concept that is equivalent to $\nabla\!\!\!\nabla^k_{\otimes}(\,—\,|\,(R, A)\colon +2, (R, B)\colon +1)$.*
Then C contains at least $\frac{(k/2-2)}{2}$ quantifiers.

Proof. Use Corollary 9 to convert C into an equivalent expression $C_{\geq}$ containing only $\geq$ quantifiers, and in the same number as the total number of quantifier distinct in C. By Theorem 11, $C_{\geq}$ must contain at least $\frac{(k/2-2)}{2}$ different quantifiers, and so must C. $\square$

As k increases, the length of the positive tooth expression $\nabla\!\!\!\nabla^k_{\otimes}(\,—\,|\,(R, A)\colon +2, (R, B)\colon +1)$ will grow logarithmically in k, but the length of any equivalent $\mathcal{ALCQ}$ expression will grow at least linearly in k.

On the other hand, $\geq j\ R.C$ is always equivalent to $\nabla\!\!\!\nabla^j_{\otimes}(\,—\,|\,(R, C)\colon +1)$; so we can conclude that $\mathcal{ALC}$ with positive counting teeth is strictly more succinct than $\mathcal{ALCQ}$.

Proposition 13. *$\mathcal{ALCP}^+$ is more succinct than $\mathcal{ALCQ}$.*

6 Embedding $\mathcal{ALCP}$ 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 positive weights. 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

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

Let us assume for now that 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 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 denoted $rwt_{\odot}(T)$. 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, 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. Baader and Bortoli (2019) 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. Consider the $\mathcal{ALCSCC}$ concept

$$rwt_{\odot}(C) = succ(\ t_l + \sum_{\substack{w_i \cdot x_i \in \mathsf{summands} \\ w_i < 0}} |w_i| \cdot x_i \leq t_r + \sum_{\substack{w_i \cdot x_i \in \mathsf{summands} \\ w_i \geq 0}} w_i \cdot x_i\ )$$

where $|w_i|$ is the absolute value of w_i, and $t_l = \max(t, 0)$ and $t_r = -\min(t, 0)$. (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 14. C *is satisfiable wrt* $\mathcal{T}$ *iff* $rwt_{\odot}(\mathsf{C})$ *is satisfiable wrt* $rwt_{\odot}(\mathcal{T})$.

We can now do better than Proposition 4.

Proposition 15. *Satisfiability of* $\mathcal{ALCP}$ *concepts wrt a TBox, where weights are allowed to be negative, is* ExpTime*-complete, even when the threshold is expressed in binary.*

Proof. Starting from the deepest teeth (in the concept and the TBox), we rewrite them into an $\mathcal{ALCSCC}^\infty$ concept using $rwt_{\odot}()$, adding zoo role axioms into the TBox as we go. Those are a series of polynomial transformations, all equi-satisfiable by Lemma 14. The result follows because reasoning in $\mathcal{ALCSCC}^\infty$ can be done in ExpTime (Baader and Bortoli 2019). □

Proposition 15 generalises the complexity results of Galliani et al. (2020) for $\mathcal{ALC}$ with the standard tooth.

7 Conclusions

We extended the tooth with role-successor counting. When we do not allow negative weights, the extended perceptron operator can still faithfully and easily express concepts like the 'compulsory imprisonment' from the Florida Score Sheet. When adding the positive operator to $\mathcal{ALC}$, the resulting logic, called $\mathcal{ALCP}^+$, was shown to have exactly the same expressivity as $\mathcal{ALCQ}$. We showed also how reasoning in $\mathcal{ALCP}^+$ can be transformed into reasoning in $\mathcal{ALCQ}$, allowing one to straightforwardly use state-of-the-art reasoning services for $\mathcal{ALCQ}$. On the other hand, we showed that $\mathcal{ALCP}^+$ is strictly more succinct that $\mathcal{ALCQ}$.

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

The complexity of the DL obtained by adding the general counting perceptron operator to $\mathcal{ALC}$, i.e. the logic $\mathcal{ALCP}$, is ExpTime-complete, no matter whether we allow negative weights or not, or whether the threshold is represented in unary or binary. This generalises the complexity results of Galliani et al. (2020) for $\mathcal{ALC}$ with the standard tooth.

As illustrated by Galliani et al. (2020), tooth operators can be seen as 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 weights and threshold when given a set of assertions about individuals (i.e. an ABox). The proposed encoding of $\mathcal{ALCP}^+$ into $\mathcal{ALCQ}$ allows one to apply these ideas also to the positive counting case, and the future development of full automated reasoning support for the expressive logic $\mathcal{ALCP}$ will further push this research line for concept learning. Thus, the core open problems to be addressed next include to pinpoint the exact expressivity of the language $\mathcal{ALCP}$, and to develop an efficient tableau-based algorithm for this language, capitalising on existing algorithms and tool infrastructure, and ideally performing within the ExpTime bound.

Another exciting future direction of research is further to replace the simple role counting with more complex non-linear functions, e.g. polynomials or sigmoids, that take the number n of roles fillers as input. Such richer families of operators could offer new perspectives on integrating into KBs concepts learnt with more advanced algorithms, as they more closely mimic neural architectures, and thus will provide a closer link to neuro-symbolic reasoning.

References

Baader, F., and Bortoli, F. D. 2019. On the expressive power of description logics with cardinality constraints on finite and infinite sets. In Herzig, A., and Popescu, A., eds., *Frontiers of Combining Systems. 12th International Symposium, FroCoS 2019, London, UK, September 4-6, 2019, Proceedings*, volume 11715 of *LNCS*, 203–219. Springer.

Baader, F., and Ecke, A. 2017. Extending the description logic $\mathcal{ALC}$ with more expressive cardinality constraints on concepts. In Benzmüller, C.; Lisetti, C. L.; and Theobald, M., eds., *GCAI 2017, 3rd Global Conference on Artificial Intelligence, Miami, FL, USA, 18-22 October 2017*, volume 50 of *EPiC Series in Computing*, 6–19. EasyChair.

Baader, F., and Sattler, U. 1999. Expressive number restrictions in description logics. *Journal of Logic and Computation* 9(3):319–350.

Baader, F.; Bednarczyk, B.; and Rudolph, S. 2020. Satisfiability and query answering in description logics with global and local cardinality constraints. In *24th European Conference on Artificial Intelligence (ECAI 2020)*, volume 325 of *FAIA*, 616–623. IOS Press.

Baader, F.; Horrocks, I.; Lutz, C.; and Sattler, U. 2017. *An Introduction to Description Logic*. Cambridge University Press.

Baader, F.; Brewka, G.; and Fernandez Gil, O. 2015. Adding threshold concepts to the description logic $\mathcal{EL}$. In Lutz, C., and Ranise, S., eds., *Frontiers of Combining Systems, 10th International Symposium, FroCoS 2015, Wroclaw, Poland, September 21-24, 2015. Proceedings*, volume 9322 of *LNCS*, 33–48. Springer.

Baader, F. 2017. A new description logic with set constraints and cardinality constraints on role successors. In Dixon, C., and Finger, M., eds., *Frontiers of Combining Systems, 11th International Symposium, FroCoS 2017, Brasília, Brazil, September 27–29, 2017, Proceedings*, volume 10483 of *LNCS*, 43–59. Springer.

Florida Department of Corrections, and Florida Office of the State Courts Administrator. 2019. Florida Criminal Punishment Code: Scoresheet Preparation Manual. http://www.dc.state.fl.us/pub/sen_cpcm/index.html.

Galliani, P.; Kutz, O.; Porello, D.; Righetti, G.; and Troquard, N. 2019. On knowledge dependence in weighted description logic. In Calvanese, D., and Iocchi, L., eds., *GCAI 2019. Proceedings of the 5th Global Conference on Artificial Intelligence, Bozen/Bolzano, Italy, 17–19 September 2019*, volume 65 of *EPiC Series in Computing*, 68–80. EasyChair.

Galliani, P.; Righetti, G.; Kutz, O.; Porello, D.; and Troquard, N. 2020. Perceptron connectives in knowledge representation. In Keet, C. M., and Dumontier, M., eds., *Knowledge Engineering and Knowledge Management, 22nd International Conference, EKAW 2020, Bolzano, Italy, September 16–20, 2020, Proceedings*, volume 12387 of *LNCS*, 183–193. Springer.

Galliani, P.; Kutz, O.; and Troquard, N. 2021. Perceptron operators that count. In Homola, M.; Ryzhikov, V.; and Schmidt, R. A., eds., *Proceedings of the 34th International Workshop on Description Logics (DL 2021)*, volume Vol-2954 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Giordano, L., and Dupré, D. T. 2021. Weighted defeasible knowledge bases and a multipreference semantics for a deep neural network model. In Faber, W.; Friedrich, G.; Gebser, M.; and Morak, M., eds., *Logics in Artificial Intelligence, 17th European Conference, JELIA 2021, Virtual Event, May 17–20, 2021, Proceedings*, volume 12678 of *LNCS*, 225–242. Springer.

Giordano, L., and Dupré, D. T. 2022. An ASP approach for reasoning on neural networks under a finitely many-valued semantics for weighted conditional knowledge bases. *Theory and Practice of Logic Programming* 22(4):589–605.

Kuncak, V., and Rinard, M. C. 2007. Towards efficient satisfiability checking for Boolean algebra with Presburger arithmetic. In Pfenning, F., ed., *21st International Conference on Automated Deduction (CADE-21), Bremen, Germany, July 17–20, 2007, Proceedings*, volume 4603 of *LNCS*, 215–230. Springer.

Ohlbach, H. J., and Koehler, J. 1999. Modal logics, description logics and arithmetic reasoning. *Artificial Intelligence* 109(1):1–31.

Porello, D.; Kutz, O.; Righetti, G.; Troquard, N.; Galliani, P.; and Masolo, C. 2019. 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*.

Righetti, G.; Porello, D.; Kutz, O.; Troquard, N.; and Masolo, C. 2019. Pink panthers and toothless tigers: three problems in classification. In Cangelosi, A., and Lieto, A., eds., *Proceedings of the 7th International Workshop on Artificial Intelligence and Cognition, Manchester, UK, September 10–11, 2019*, volume 2483 of *CEUR Workshop Proceedings*. CEUR-WS.org. 39–53.

Righetti, G.; Porello, D.; and Confalonieri, R. 2022. Evaluating the interpretability of threshold operators. In Oscar Corcho; Hollink, L.; Kutz, O.; Troquard, N.; and Ekaputra, F. J., eds., *Knowledge Engineering and Knowledge Management, 23rd International Conference, EKAW 2022, Bolzano, Italy, September 26–29, 2022, Proceedings*, volume 13514 of *LNCS*, 136–151. Springer.

Schild, K. 1991. A correspondence theory for terminological logics: Preliminary report. In Mylopoulos, J., and Reiter, R., eds., *Proceedings of the 12th International Joint Conference on Artificial Intelligence. Sydney, Australia, August 24–30, 1991*, 466–471. Morgan Kaufmann.

Tobies, S. 2000. The complexity of reasoning with cardinality restrictions and nominals in expressive description logics. *Journal of Artificial Intelligence Research* 12:199–217.