All Elephants are Bigger than All Mice*

Sebastian Rudolph and Markus Krötzsch and Pascal Hitzler

Institut AIFB, Universität Karlsruhe, Germany
{sru,mak,phi}@aifb.uni-karlsruhe.de

Abstract. We investigate the *concept product* as an expressive feature for description logics (DLs). While this construct allows us to express an arguably very common and natural type of statement, it can be simulated only by the very expressive DL $\mathcal{SROIQ}$ for which no tight worst-case complexity is known. However, we show that concept products can also be added to the DLs $\mathcal{SHOIQ}$ and $\mathcal{SHOI}$, and to the tractable DL $\mathcal{EL}^{++}$ without increasing the worst-case complexities in any of those cases. We therefore argue that concept products provide practically relevant expressivity at little cost, making them a good candidate for future extensions of the DL-based ontology language OWL.

1 Introduction

The development of description logics (DLs) has been driven by the desire to push the expressivity bounds of these knowledge representation formalisms while still maintaining decidability and implementability. This has lead to very expressive DLs such as $\mathcal{SHOIN}$, the logic underlying the Web Ontology Language OWL DL, $\mathcal{SHOIQ}$, and $\mathcal{SROIQ}$ [1] which is the basis for the ongoing standardisation of OWL 2[1] as the next version of the Web Ontology Language. On the other hand, more light-weight DLs for which most common reasoning problems can be implemented in (sub)polynomial time have also been sought, leading, e.g., to the tractable DL $\mathcal{EL}^{++}$ [2].

In this work, we continue these lines of research by investigating an expressive feature – the *concept product* – in the context of various well-known DLs, showing that this added expressivity does not increase worst-case complexities in any of these cases. Intuitively, the concept product – hitherto sporadically described (e.g. in [3] or [4]) but neglected by mainstream DL research and OWL standardisation efforts – allows us to define a role that connects every instance in one class with every instance in another class. An example is given in the title: Given the class of all elephants, and the class of all mice, we wish to specify a DL knowledge base that allows us to conclude that any individual elephant is bigger than any individual mouse, or, stated more formally:

$$\forall(x).\forall y.\mathsf{Elephant}(x) \land \mathsf{Mouse}(y) \rightarrow \mathsf{biggerThan}(x, y)$$

Using common DL syntax, one could also write $\mathsf{Elephant}^{\mathcal{I}} \times \mathsf{Mouse}^{\mathcal{I}} \subseteq \mathsf{biggerThan}^{\mathcal{I}}$, which explains the name "concept product" and will also motivate our DL syntax.

* Supported by the European Commission under contracts 027595 NeOn and 215040 ACTIVE, and by the Deutsche Forschungsgemeinschaft (DFG) under the ReaSem project.

[1] http://www.w3.org/2007/OWL

Maybe surprisingly, this semantic relationship cannot be specified in any but the most expressive DLs today (except for not that widely known DLs that allow for role negation, cf. [5, 6]). Using quantifiers, one can only state that any elephant is bigger than *some* mouse, or that elephants are bigger than nothing but mice. Nominals also allow us to state that some particular elephant is bigger than all mice, and with DL-safe rules [7], one might say that all *named* elephants are bigger than all *named* mice. Yet, none of these formalisations captures the true intention of the informal statement.

Now one could hope that this kind of statement would be rarely needed in practical applications, but in fact it represents a very common modelling problem of relating two individuals based on their (inferred) properties. Natural and life sciences provide a wealth of typical examples, for example:

- Alkaline solutions neutralise acid solutions.
- Antihistamines alleviate allergies.
- Oppositely charged bodies attract each other.

Reasoning about such relations is important e.g. in the context of the HALO project[2], which sets out to develop reasoning systems for solving complex examination questions from physics, biology, and chemistry. Qualitative reasoning about a given scenario is often required before any concrete arithmetic processing steps can be invoked.

Another particularly interesting example is the task of developing a knowledge base capturing our current insights about DL complexities and available reasoning implementations. It should entail statements like

- Any reasoner that can handle $\mathcal{SHIQ}$ can deal with every DLP-ontology.
- Any problem within ExpTime can be polynomially reduced to any ExpTime-complete problem.
- In any description logic containing nominals, inverses and number restrictions, satisfiability checking is hard for any complexity below or equal ExpTime.

All of those can easily be cast into concept products. An interesting aspect of reasoning about complexities is that it involves upper *and* lower bounds, and thus also escapes from most other modelling attempts (e.g. using classes instead of instances to represent concrete DLs). This might be a reason that the DL complexity navigator[3] is based on JavaScript rather than on more advanced DL knowledge representation technologies.

In this paper, we show that it is in fact not so difficult to extend a broad array of existing description logics with enough additional modelling power to capture all of the above, while still retaining their known upper complexity bounds. We start with the short preliminary Section 2 to recall the definition of the DL $\mathcal{SROIQ}$, and then proceed by introducing the concept product formally in Section 3. Concept products there can indeed be simulated by existing constructs and thus are recognised as syntactic sugar. This is quite different for the tractable DL $\mathcal{EL}^{++}$ investigated in Section 4. Yet, we will see that polynomial reasoning in $\mathcal{EL}^{++}$ with concept products is possible, thus further pushing the $\mathcal{EL}$ envelope. In the subsequent Section 5, we show that $\mathcal{SHOIQ}$

[2] `http://www.projecthalo.com/`
[3] `http://www.cs.man.ac.uk/~ezolin/dl/`

Table 1. Semantics of concept constructors in $\mathcal{SROIQ}$ for an interpretation $\mathcal{I}$ with domain $\Delta^{\mathcal{I}}$.

Name	Syntax	Semantics
inverse role	R^-	$\{\langle x,y\rangle \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid \langle y,x\rangle \in R^{\mathcal{I}}\}$
universal role	U	$\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
top	$\top$	$\Delta^{\mathcal{I}}$
bottom	$\bot$	$\emptyset$
negation	$\neg C$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
conjunction	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
disjunction	$C \sqcup D$	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$
nominals	$\{a\}$	$\{a^{\mathcal{I}}\}$
univ. restriction	$\forall R.C$	$\{x \in \Delta^{\mathcal{I}} \mid \langle x,y\rangle \in R^{\mathcal{I}}$ implies $y \in C^{\mathcal{I}}\}$
exist. restriction	$\exists R.C$	$\{x \in \Delta^{\mathcal{I}} \mid$ for some $y \in \Delta^{\mathcal{I}}$, $\langle x,y\rangle \in R^{\mathcal{I}}$ and $y \in C^{\mathcal{I}}\}$
qualified number	$\leq n\, S.C$	$\{x \in \Delta^{\mathcal{I}} \mid \#\{y \in \Delta^{\mathcal{I}} \mid \langle x,y\rangle \in S^{\mathcal{I}}$ and $y \in C^{\mathcal{I}}\} \leq n\}$
restriction	$\geq n\, S.C$	$\{x \in \Delta^{\mathcal{I}} \mid \#\{y \in \Delta^{\mathcal{I}} \mid \langle x,y\rangle \in S^{\mathcal{I}}$ and $y \in C^{\mathcal{I}}\} \geq n\}$

and $\mathcal{SHOI}$ with concept products are still NExpTime-complete and ExpTime-complete, respectively, thus obtaining tight complexity bounds for a very expressive DL as well. We omit most formal details for reasons of space – all proofs and extended definitions are found in [8].

2 Preliminaries: the DL $\mathcal{SROIQ}$

In this section, we recall the definition of the expressive description logic $\mathcal{SROIQ}$ [1]. We assume that the reader is familiar with description logics [9]. As usual, the DLs considered in this paper are based on three disjoint finite sets of *individual names* N_I, *concept names* N_C, and *role names* N_R containing the *universal role* $U \in \mathsf{N}_R$.

Definition 1. *A $\mathcal{SROIQ}$ Rbox for N_R is based on a set* $\mathbf{R}$ *of* atomic roles *defined as* $\mathbf{R} := \mathsf{N}_R \cup \{R^- \mid R \in \mathsf{N}_R\}$. *As usual, we set* $\mathrm{Inv}(R) := R^-$ *and* $\mathrm{Inv}(R^-) := R$.

A generalised role inclusion axiom *(RIA) is a statement of the form* $S_1 \circ \ldots \circ S_n \sqsubseteq R$, *and a set of such RIAs is a $\mathcal{SROIQ}$* Rbox. *An Rbox is* regular *if there is a strict partial order $\prec$ on* $\mathbf{R}$ *such that* $S \prec R$ *iff* $\mathrm{Inv}(S) \prec R$, *and every RIA is of one of the forms:*

$$R \circ R \sqsubseteq R, \quad R^- \sqsubseteq R, \quad S_1 \circ \ldots \circ S_n \sqsubseteq R, \quad R \circ S_1 \circ \ldots \circ S_n \sqsubseteq R, \quad S_1 \circ \ldots \circ S_n \circ R \sqsubseteq R$$

such that $R \in \mathsf{N}_R$ is a (non-inverse) role name, and $S_i \prec R$ for $i = 1, \ldots, n$. The set of simple *roles for some Rbox is defined inductively as follows:*

- *If a role R occurs only on the right-hand-side of RIAs of the form $S \sqsubseteq R$ such that S is simple, then R is also simple.*
- *The inverse of a simple role is simple.*

A $\mathcal{SROIQ}$ Tbox consists of concept inclusion axioms of the form $C \sqsubseteq D$ where C and D are concept expressions based on the constructors shown in Table 1. As all DLs in considered this paper support nominals, we do not explicitly need to introduce Abox axioms, which can be internalised into the Tbox in the standard way [1]. A $\mathcal{SROIQ}$ knowledge base thus is assumed to be the union of an Rbox and an according Tbox.

Further details on $\mathcal{SROIQ}$ can be found in [1]. We have omitted here several semantic features that are not relevant to our study, especially various forms of role assertions.

An interpretation $\mathcal{I}$ consists of a set $\Delta^{\mathcal{I}}$ called *domain* (the elements of it being called *individuals*) together with a function $\cdot^{\mathcal{I}}$ mapping individual names to elements of $\Delta^{\mathcal{I}}$, concept names to subsets of $\Delta^{\mathcal{I}}$, and role names to subsets of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. The function $\cdot^{\mathcal{I}}$ is inductively extended to role and concept expressions as shown in Table 1. An interpretation $\mathcal{I}$ *satisfies* an axiom φ if we find that $\mathcal{I} \models \varphi$:

- $\mathcal{I} \models S \sqsubseteq R$ if $S^{\mathcal{I}} \subseteq R^{\mathcal{I}}$,
- $\mathcal{I} \models S_1 \circ \ldots \circ S_n \sqsubseteq R$ if $S_1^{\mathcal{I}} \circ \ldots \circ S_n^{\mathcal{I}} \sqsubseteq R^{\mathcal{I}}$ ($\circ$ being overloaded to denote the standard composition of binary relations here),
- $\mathcal{I} \models C \sqsubseteq D$ if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$.

An interpretation $\mathcal{I}$ *satisfies* a knowledge base KB (we then say that $\mathcal{I}$ is a *model* of KB and write $\mathcal{I} \models$ KB) if it satisfies all axioms of KB. A knowledge base KB is *satisfiable* if it has a model. Two knowledge bases are *equivalent* if they have exactly the same models, and they are *equisatisfiable* if both are unsatisfiable or both are satisfiable.

3 Simulating Concept Products in $\mathcal{SROIQ}$

We now formally introduce the DL *concept product* as a new constructor in description logic knowledge bases. The DL $\mathcal{SROIQ}$ extended with this constructor will be denoted $\mathcal{SROIQ}^{\times}$. It will turn out that concept products appear as *syntactic sugar* in $\mathcal{SROIQ}^{\times}$ since they can be represented by combining nominals, inverse roles, and complex role inclusion axioms.

Definition 2. *A* concept product inclusion *is a statement of the form $C \times D \sqsubseteq R$ where $C, D \in \mathbf{C}$ are $\mathcal{SROIQ}$ concepts, and R is an atomic $\mathcal{SROIQ}$ role.*

A $\mathcal{SROIQ}^{\times}$ Rbox is the union of a $\mathcal{SROIQ}$ Rbox with a set of concept product inclusions based on roles and concepts for that Rbox. Simplicity of roles is defined as in $\mathcal{SROIQ}$ where concept product axioms are considered as additional kinds of RIAs. Especially, any role R occurring in such a statement is not simple in $\mathcal{SROIQ}^{\times}$.

A $\mathcal{SROIQ}^{\times}$ knowledge base KB *is the union of a $\mathcal{SROIQ}^{\times}$ Rbox $\mathcal{R}$ and a $\mathcal{SROIQ}$ Tbox $\mathcal{T}$ (for $\mathcal{R}$).*

The model theoretic semantics of $\mathcal{SROIQ}$ is extended to $\mathcal{SROIQ}^{\times}$ by setting $\mathcal{I} \models C \times D \sqsubseteq R$ iff $C^{\mathcal{I}} \times D^{\mathcal{I}} \subseteq R^{\mathcal{I}}$ for any interpretation $\mathcal{I}$.

We immediately observe that $\times$ generalises the universal role which can now be defined by the axiom $\top \times \top \sqsubseteq U$. However, our extension of the notion of simplicity of roles would then cause U to become non-simple, which is not needed. We conjecture that one can generally consider the concept product to have no impact on simplicity of roles, but our below approach of simulating concept products in $\mathcal{SROIQ}$ requires us to impose that restriction. We leave it to future work to conceive a modified tableau procedure for $\mathcal{SROIQ}^{\times}$ that directly takes concept products into account – our results for $\mathcal{SHOIQ}^{\times}$ show that such an extended simplicity would not impose problems there.

Lemma 3. *Consider a $\mathcal{SROIQ}^{\times}$ knowledge base* KB *with some concept product axiom $C \times D \sqsubseteq R$. A knowledge base* KB′ *that is equisatisfiable to* KB *is obtained as follows:*

- *delete the Rbox axiom* $C \times D \sqsubseteq R$*,*
- *add a new RIA* $R_1 \circ R_2 \sqsubseteq R$*, where* R_1*,* R_2 *are fresh role names,*
- *introduce fresh nominal* $\{a\}$*, and add Tbox axioms* $C \sqsubseteq \exists R_1.\{a\}$ *and* $D \sqsubseteq \exists R_2^-.\{a\}$*.*

Clearly, the elimination step from the above lemma can be applied recursively to eliminate all concept products. A simple induction thus yields the following result:

Proposition 4. *Every* $\mathcal{SROIQ}^\times$ *KB can be reduced to an equisatisfiable* $\mathcal{SROIQ}$ *KB in polynomial time. In particular, satisfiability of* $\mathcal{SROIQ}^\times$ *knowledge bases is decidable.*

Decidability of $\mathcal{SROIQ}$ was shown in [1]. Since $\mathcal{SROIQ}$ is already NExpTime-hard, this also suffices to conclude that the (currently unknown) worst-case complexities of $\mathcal{SROIQ}^\times$ and $\mathcal{SROIQ}$ coincide.

4 Polynomial Reasoning with Concept Products in $\mathcal{EL}^{++}$

In this section, we investigate the use of concepts products in the DL $\mathcal{EL}^{++}$ [2], for which many typical inference problems can be solved in polynomial time. $\mathcal{EL}^{++}$ cannot simulate concept products as it does support nominals and RIAs, but no inverse roles. While it is known that the addition of inverses makes satisfiability checking ExpTime-complete [10], we show that sound and complete reasoning with the concept product is still tractable. We simplify our presentation by omitting concrete domains from $\mathcal{EL}^{++}$ – they are not affected by our extension and can be treated as shown in [2] – and by considering only $\mathcal{EL}^{++\times}$ knowledge bases that are in a simplified normal form – the normal form transformation for the general case is detailed in [8].

Definition 5. *An* $\mathcal{EL}^{++\times}$ *knowledge base* KB *in* normal form *is a set of axioms of the following forms:*

$$\begin{array}{rclcrclcrclcrcl} A & \sqsubseteq & C & \quad & A \sqcap B & \sqsubseteq & C & \quad & R & \sqsubseteq & T & \quad & A \times B & \sqsubseteq & T \\ \exists R.A & \sqsubseteq & B & & A & \sqsubseteq & \exists R.B & & R \circ S & \sqsubseteq & T & & & & \end{array}$$

where $A, B \in \mathsf{N}_C \cup \{\{a\} \mid a \in \mathsf{N}_I\} \cup \{\top\}$*,* $C \in \mathsf{N}_C \cup \{\{a\} \mid a \in \mathsf{N}_I\} \cup \{\bot\}$*, and* $R, S, T \in \mathsf{N}_R$*.*

A polynomial algorithm for checking class subsumptions in $\mathcal{EL}^{++}$ has been given in [2], and it was shown that other standard inference problems can be reduced to that problem. We now present a modified algorithm for $\mathcal{EL}^{++\times}$ – also using some modified notation – and show its correctness for this extended DL. The algorithm checks whether a subsumption $A \sqsubseteq B$ between concept names is entailed by some normalised $\mathcal{EL}^{++\times}$ knowledge base KB. To this end, it computes a set $\mathcal{S}$ of inclusion axioms that are entailed by KB, where we only need to consider axioms of the forms $C \sqsubseteq D$ and $C \sqsubseteq \exists R.D$, where C, D are elements of the set $\mathcal{B} := \mathsf{N}_C \cup \{\{a\} \mid a \in \mathsf{N}_I\} \cup \{\top, \bot\}$.

The set $\mathcal{S}$ is initialised by setting $\mathcal{S} := \{C \sqsubseteq C \mid C \in \mathcal{B}\} \cup \{C \sqsubseteq \top \mid C \in \mathcal{B}\}$. Then the rules in Table 2 are applied until no possible rule application further modifies $\mathcal{S}$. The rules refer to a binary relation $\leadsto \subseteq \mathcal{B} \times \mathcal{B}$ that is defined based on the current content of $\mathcal{S}$. Namely, $C \leadsto D$ holds whenever there are $C_1, \ldots, C_k \in \mathcal{B}$ such that

- C_1 is equal to one of the following: C, $\top$, $\{a\}$ (for some individual $a \in \mathsf{N}_I$), or A (where the subsumption $A \sqsubseteq B$ is to be checked),

Table 2. Completion rules for reasoning in $\mathcal{EL}^{++\times}$. Symbols C, D, possibly with subscripts or primes, denote elements of $\mathcal{B}$, whereas E might be any element of $\mathcal{B} \cup \{\exists R.C \mid C \in \mathcal{B}\}$.

(R1) If $D \sqsubseteq E \in$ KB and $C \sqsubseteq D \in \mathcal{S}$ then $\mathcal{S} := \mathcal{S} \cup \{C \sqsubseteq E\}$.
(R2) If $C_1 \sqcap C_2 \sqsubseteq D \in$ KB and $\{C \sqsubseteq C_1, C \sqsubseteq C_2\} \subseteq \mathcal{S}$ then $\mathcal{S} := \mathcal{S} \cup \{C \sqsubseteq D\}$.
(R3) If $\exists R.C \sqsubseteq D \in$ KB and $\{C_1 \sqsubseteq \exists R.C_2, C_2 \sqsubseteq C\} \subseteq \mathcal{S}$ then $\mathcal{S} := \mathcal{S} \cup \{C_1 \sqsubseteq D\}$.
(R4) If $\{C \sqsubseteq \exists R.D, D \sqsubseteq \bot\} \subseteq \mathcal{S}$ then $\mathcal{S} := \mathcal{S} \cup \{C \sqsubseteq \bot\}$.
(R5) If $\{C \sqsubseteq \{a\}, D \sqsubseteq \{a\}, D \sqsubseteq E\} \subseteq \mathcal{S}$ and $C \rightsquigarrow D$ then $\mathcal{S} := \mathcal{S} \cup \{C \sqsubseteq E\}$.
(R6) If $R \sqsubseteq S \in$ KB and $C \sqsubseteq \exists R.D \in \mathcal{S}$ then $\mathcal{S} := \mathcal{S} \cup \{C \sqsubseteq \exists S.D\}$.
(R7) If $R \circ S \sqsubseteq T \in$ KB and $\{C_1 \sqsubseteq \exists R.C_2, C_2 \sqsubseteq \exists S.C_3\} \subseteq \mathcal{S}$ then $\mathcal{S} := \mathcal{S} \cup \{C_1 \sqsubseteq \exists T.C_3\}$.
(R8) If $C \times D \sqsubseteq R \in$ KB, $D' \sqsubseteq D \in \mathcal{S}$, and $C \rightsquigarrow D'$ then $\mathcal{S} := \mathcal{S} \cup \{C \sqsubseteq \exists R.D'\}$.

- $C_i \sqsubseteq \exists R.C_{i+1} \in \mathcal{S}$ for some $R \in \mathsf{N}_R$ ($i = 1, \ldots, k-1$), and
- $C_k = D$.

Intuitively, $C \rightsquigarrow D$ states that D cannot be interpreted as the empty set if we assume that C contains some element. The option $C_1 = A$ reflects the fact that we can base our conclusions on the assumption that A is not equivalent to $\bot$ – if it is, the queried subsumption holds immediately, so we do not need to check this case.[4]

After terminating with the saturated set $\mathcal{S}$, the algorithm confirms the subsumption $A \sqsubseteq B$ iff one of the following conditions hold:

$$A \sqsubseteq B \in \mathcal{S} \quad \text{or} \quad A \sqsubseteq \bot \in \mathcal{S} \quad \text{or} \quad \{a\} \sqsubseteq \bot \in \mathcal{S} \text{ (for some } a \in \mathsf{N}_I) \quad \text{or} \quad \top \sqsubseteq \bot \in \mathcal{S}.$$

It can indeed be shown that the algorithm is correct, and that it runs in polynomial time. For reasons of space, we include only the completeness proof into this paper.

Lemma 6. *Let $\mathcal{S}$ be the saturated set obtained by the subsumption checking algorithm for a normalised $\mathcal{EL}^{++\times}$ knowledge base* KB *and some queried subsumption $A \sqsubseteq B$. If* KB $\models A \sqsubseteq B$ *then one of the following holds:*

$$A \sqsubseteq B \in \mathcal{S} \quad \textit{or} \quad A \sqsubseteq \bot \in \mathcal{S} \quad \textit{or} \quad \{a\} \sqsubseteq \bot \in \mathcal{S} \textit{ (for some } a \in \mathsf{N}_I) \quad \textit{or} \quad \top \sqsubseteq \bot \in \mathcal{S}.$$

Proof. We show the contrapositive: if none of the given conditions hold, then there is a model $\mathcal{I}$ for KB within which the subsumption $A \sqsubseteq B$ does not hold. The proof proceeds by constructing this model. The domain $\Delta^{\mathcal{I}}$ of $\mathcal{I}$ is chosen to contain only one characteristic individual for all classes of KB that are necessarily non-empty, factorised to take inferred equalities into account. To this end, we first define a set of concept expressions $\mathcal{B}^- := \{C \in \mathcal{B} \mid A \rightsquigarrow C\}$. A binary relation $\sim$ on $\mathcal{B}^-$ that will serve us to represent inferred equalities is defined as follows: $C \sim D$ iff $C = D$ or $\{C \sqsubseteq \{a\}, D \sqsubseteq \{a\}\} \subseteq \mathcal{S}$ for some $a \in \mathsf{N}_I$.

We will see below that $\sim$ is an equivalence relation on $\mathcal{B}^-$. Reflexivity and symmetry are obvious. For transitivity, we first show that elements related by $\sim$ are subject to the same assertions in $\mathcal{S}$. Thus consider $C, C' \in \mathcal{B}^-$ such that $C \sim C'$. We claim that, for all concept expressions E, we find that $C \sqsubseteq E \in \mathcal{S}$ implies $C' \sqsubseteq E \in \mathcal{S}$ (Claim $*$). Assume $C \neq C'$ and $\{C \sqsubseteq \{a\}, C' \sqsubseteq \{a\}\} \subseteq \mathcal{S}$ – the other case is trivial. But by our definition of $\mathcal{B}^-$, we find that $C \rightsquigarrow C'$, and hence rule (R5) is applicable and establishes the required

[4] This case is actually missing in [2], and it needs to be added to obtain a complete algorithm.

result. This also yields transitivity of $\sim$, since $\{C_1 \sqsubseteq \{a\}, C_2 \sqsubseteq \{a\}\} \subseteq \mathcal{S}$ and $C_2 \sim C_3$ implies $C_3 \sqsubseteq \{a\} \in \mathcal{S}$ and thus $C_1 \sim C_3$. We use $[C]$ to denote the equivalence class of $C \in \mathcal{B}^-$ w.r.t. $\sim$. These observations allow us to make the following definition of $\mathcal{I}$:

$$\begin{aligned} \Delta^{\mathcal{I}} &:= \{[C] \mid C \in \mathcal{B}^-\} & C^{\mathcal{I}} &:= \{[D] \in \Delta^{\mathcal{I}} \mid D \sqsubseteq C \in \mathcal{S}\} \text{ for } C \in \mathsf{N}_C \\ a^{\mathcal{I}} &:= [\{a\}] \text{ for } a \in \mathsf{N}_I & R^{\mathcal{I}} &:= \{\langle [C],[D]\rangle \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid C \sqsubseteq \exists R.D \in \mathcal{S}\} \text{ for } R \in \mathsf{N}_R. \end{aligned}$$

Note that N_I was assumed to be fixed and finite, and that all $\{a\} \in \mathcal{B}^-$ for all $a \in \mathsf{N}_I$ such that $[\{a\}]$ is well-defined. Roles and concepts not involved in $\mathcal{B}^-$ or $\mathcal{S}$ are automatically interpreted as the empty set by the above definition. The definitions of $C^{\mathcal{I}}$ and $R^{\mathcal{I}}$ are well-defined due to $(*)$ above.

We can now observe the following desired correspondence between $\mathcal{I}$ and $\mathcal{S}$: For any $C, D \in \mathcal{B}^-$, we find that $[C] \in D^{\mathcal{I}}$ iff $C \sqsubseteq D \in \mathcal{S}$ (Claim †). We distinguish various cases based on the structure of D:

- $D = \bot$. We can conclude $[C] \notin \bot^{\mathcal{I}}$ and $C \sqsubseteq \bot \notin \mathcal{S}$ by noting that, for any $E \in \mathcal{B}^-$ we have that $E \sqsubseteq \bot \notin \mathcal{S}$. To see that, suppose the contrary. By $A \rightsquigarrow E$ there is a chain $C_1, \ldots, C_k \in \mathcal{B}$ as in the definition of $\rightsquigarrow$ such that $C_k = E$. Using $C_{k-1} \sqsubseteq \exists R.E \in \mathcal{S}$ and rule (R4), we conclude that $C_{k-1} \sqsubseteq \bot \in \mathcal{S}$. Applying this reasoning inductively, we obtain $C_1 \sqsubseteq \bot \in \mathcal{S}$. But as C_1 is of the form A, $\{a\}$, or $\top$, this contradicts our initial assumptions.
- $D = \top$. By the initialisation of $\mathcal{S}$, $C \sqsubseteq \top \in \mathcal{S}$ and also $[C] \in \top^{\mathcal{I}}$.
- $D \in \mathsf{N}_C$. This case follows directly from the definition of $\mathcal{I}$.
- $D = \{a\}$ for some $a \in \mathsf{N}_I$. If $[C] \in \{a\}^{\mathcal{I}}$ then $[C] = [\{a\}]$, and hence $C \sim \{a\}$. Since $\{a\} \sqsubseteq \{a\} \in \mathcal{S}$, we obtain $C \sqsubseteq \{a\} \in \mathcal{S}$ from $(*)$. Conversely, if $C \sqsubseteq \{a\} \in \mathcal{S}$, then $C \sim \{a\}$ and hence $\{[C]\} = \{[\{a\}]\} = \{a\}^{\mathcal{I}}$ as required.

It is easy to see that $\mathcal{I} \not\models A \sqsubseteq B$: since $A \in \mathcal{B}^-$, we find that $[A] \in A^{\mathcal{I}}$ due to $A \sqsubseteq A \in \mathcal{S}$ by the initialisation of the algorithm. But since $A \sqsubseteq B \notin \mathcal{S}$, we have that $[A] \notin B^{\mathcal{I}}$ based on (†).

Finally, it only remains to show that $\mathcal{I}$ is indeed a model of KB. We argue that each axiom of KB is satisfied by $\mathcal{I}$ by considering the possible normal forms:

- $D \sqsubseteq E$ with $E \in \mathcal{B} \cup \{\exists R.E' \mid E' \in \mathcal{B}\}$. If $[C] \in D^{\mathcal{I}}$, then $C \sqsubseteq D \in \mathcal{S}$ by (†) and thus rule (R1) can be applied to yield $C \sqsubseteq E$. If $E \in \mathcal{B}$, the claim follows from (†). For $E = \exists R.E'$, we conclude that $C \rightsquigarrow E'$ and thus $E' \in \mathcal{B}^-$. By definition of $R^{\mathcal{I}}$, we find $\langle [C],[E']\rangle \in R^{\mathcal{I}}$, and since $E' \sqsubseteq E' \in \mathcal{S}$ we can invoke (†) to obtain $[E'] \in E'^{\mathcal{I}}$ as required.
- $C_1 \sqcap C_2 \sqsubseteq D$. This case is treated similar to the above case, using rule (R2) and treating only the (simpler) case where $D \in \mathcal{B}$.
- $\exists R.D \sqsubseteq E$. If $[C] \in \exists R.D^{\mathcal{I}}$ then $\langle [C],[D']\rangle \in R^{\mathcal{I}}$ for some $[D'] \in D^{\mathcal{I}}$. By the definition of $R^{\mathcal{I}}$ and $(*)$, there is some $D'' \in [D']$ such that $C \sqsubseteq \exists R.D'' \in \mathcal{S}$. Since $D'' \in \mathcal{B}$ and $[D''] = [D'] \in D^{\mathcal{I}}$, we can conclude $D'' \sqsubseteq D \in \mathcal{S}$ from (†). Thus rule (R3) implies that $C \sqsubseteq E$, and we obtain $[C] \in E^{\mathcal{I}}$ by invoking (†).
- $R \sqsubseteq S$. If $\langle [C],[D]\rangle \in R^{\mathcal{I}}$ then there is $C \sqsubseteq \exists R.D' \in \mathcal{S}$ with $[D'] = [D]$. Rule (R6) thus entailed $C \sqsubseteq \exists S.D' \in \mathcal{S}$, which yields $\langle [C],[D]\rangle \in S^{\mathcal{I}}$ by definition of $S^{\mathcal{I}}$.
- $R \circ S \sqsubseteq T$. This case is treated like the previous case, using (R7) instead of (R6).

– $C \times D \sqsubseteq R$. If $[C'] \in C^{\mathcal{I}}$ and $[D'] \in D^{\mathcal{I}}$, then (†) yields $\{C' \sqsubseteq C, D' \sqsubseteq D\} \subseteq \mathcal{S}$. Since $D' \in \mathcal{B}^{-}$, we have $A \rightsquigarrow D'$ which clearly implies $C \rightsquigarrow D'$ by definition of $\rightsquigarrow$. Hence rule (R8) was applied to yield $C \sqsubseteq \exists R.D' \in \mathcal{S}$ and by (R1) we also obtain $C' \sqsubseteq \exists R.D' \in \mathcal{S}$. Now $\langle [C'], [D'] \rangle \in R^{\mathcal{I}}$ follows from the definition of $R^{\mathcal{I}}$. □

Theorem 7. *The problem of checking concept subsumptions in $\mathcal{EL}^{++\times}$ is* P*-complete.*

Finally, one might ask how concept products affect other reasoning tasks, such as conjunctive query answering in $\mathcal{EL}^{++\times}$. As we have extended the original $\mathcal{EL}^{++}$ algorithm in a rather natural way, we expect that the automata-based algorithm for conjunctive query answering that was presented in [11] can readily be modified to cover $\mathcal{EL}^{++\times}$, so that the same complexity results for conjunctive querying could be obtained.

5 The Concept Product in $\mathcal{SHOIQ}$ and $\mathcal{SHOI}$

Below, we investigate the use of concept products in $\mathcal{SHOIQ}$, the description logic underlying OWL DL. Since $\mathcal{SHOIQ}$ does not support generalised role inclusion axioms, concept products can not be simulated by means of other axioms. Yet, we will see below that the addition of concept products does not increase the worst-case complexity of $\mathcal{SHOIQ}$ which is still NExpTime even for binary encoding of numbers. The proof also shows that roles occurring in concept product inclusions can still be considered simple without impairing this result. Finally, we will take a brief look at the DL $\mathcal{SHOI}$ which is obtained from $\mathcal{SHOIQ}$ by disallowing number restrictions, and for which satisfiability checking is only ExpTime-complete. Again, we find that the addition of the concept product to $\mathcal{SHOI}$ does not increase this worst-case complexity.

Definition 8. *A $\mathcal{SROIQ}^{\times}$ knowledge base* KB *is in $\mathcal{SHOIQ}^{\times}$ if*

– *all Rbox axioms of* KB *are of the form $S \sqsubseteq R$, $R \circ R \sqsubseteq R$, or $C \times D \sqsubseteq R$ for $R \in \mathsf{N}_R$ a role name, $S \in \mathbf{R}$ an atomic role, and $C, D \in \mathbf{C}$ concept expressions,*
– KB *does not contain the universal role U or expressions of the form $\exists R.\mathsf{Self}$.*

For a fixed knowledge base KB*, $\sqsubseteq^*$ is the smallest binary relation on $\mathbf{R}$ such that:*

– *$R \sqsubseteq^* R$ for every atomic role R,*
– *$R \sqsubseteq^* S$ and $\mathrm{Inv}(R) \sqsubseteq^* \mathrm{Inv}(S)$ for every Rbox axiom $R \sqsubseteq S$, and*
– *$R \sqsubseteq^* T$ whenever $R \sqsubseteq^* S$ and $S \sqsubseteq^* T$.*

Given an atomic role R, we write $\mathsf{Trans}(R) \in \mathrm{KB}$ as an abbreviation for: $R \circ R \sqsubseteq R \in \mathrm{KB}$ or $\mathrm{Inv}(R) \circ \mathrm{Inv}(R) \sqsubseteq \mathrm{Inv}(R) \in \mathrm{KB}$.

Whenever $R \sqsubseteq^* S$ and $S \sqsubseteq^* R$, the roles R and S are interpreted identically in any model of KB. One could thus syntactically substitute one of them by the other, which allows us to assume that all knowledge bases considered below have an acyclic Rbox. Moreover, we assume that for all concept product inclusions $A \times B \sqsubseteq R$, both A and B are atomic concepts. Obviously, this restriction does not affect expressivity, as complex concepts in such axioms can be moved into the Tbox. Given a knowledge base KB, we obtain its negation normal form $\mathsf{NNF}(\mathrm{KB})$ in the usual way. In particular, every GCI $C \sqsubseteq D$ is transformed into a universally valid concept $\mathsf{NNF}(\neg C \sqcup D)$. Furthermore, it is possible to eliminate transitivity axioms using an according transformation in [7]:

Table 3. Transformation from $\mathcal{ALCHOIQ}^{\times}$ to C^2. X is a meta-variable for representing various term symbols in the final translation. The transformations π_y are assumed to be analogous to the given transformations for π_x.

$\pi(C \sqsubseteq D)$	$:=$	$\forall x.\pi_y(\neg C \sqcup D, x)$	$\pi(R \sqsubseteq S)$	$:=$	$\forall x.\forall y.(\neg R(x,y) \vee S(x,y))$
$\pi(C \times D \sqsubseteq R)$	$:=$	$\forall x.\forall y.(\neg C(x) \vee \neg D(y) \vee R(x,y))$	$\pi(\mathrm{KB})$	$:=$	$\bigwedge_{\varphi \in \mathrm{KB}} \pi(\varphi)$
$\pi_x(\top, X)$	$:=$	$\top$	$\pi_x(A, X)$	$:=$	$A(X)$ for any $A \in \mathsf{N}_C$
$\pi_x(\bot, X)$	$:=$	$\bot$	$\pi_x(\{a\}, X)$	$:=$	$a \approx X$ for any $a \in \mathsf{N}_I$
$\pi_x(\neg C, X)$	$:=$	$\neg \pi_x(C, X)$	$\pi_x(C \sqcap D, X)$	$:=$	$\pi_x(C, X) \wedge \pi_x(D, X)$
$\pi_x(C \sqcup D, X)$	$:=$	$\pi_x(C, X) \vee \pi_x(D, X)$	$\pi_x(\forall R.C, X)$	$:=$	$\forall x.(R(X,x) \rightarrow \pi_y(C, x))$
$\pi_x(\exists R.C, X)$	$:=$	$\exists x.(R(X,x) \wedge \pi_y(C, x))$	$\pi_x(\geq n\, R.C, X)$	$:=$	$\exists_{\geq n} x.(R(X,x) \wedge \pi_y(C, x))$
$\pi_x(\leq n\, R.C, X)$	$:=$	$\exists_{\leq n} x.(R(X,x) \rightarrow \pi_y(C, x))$			

Definition 9. *Given a $\mathcal{SHOIQ}^{\times}$ knowledge base* KB*, let* $\mathsf{clos}(\mathrm{KB})$ *denote the smallest set of concept expressions where*

- $\mathsf{NNF}(\neg C \sqcup D) \in \mathsf{clos}(\mathrm{KB})$ *for any Tbox axiom $C \sqsubseteq D$,*
- $D \in \mathsf{clos}(\mathrm{KB})$ *for every subexpression D of some concept $C \in \mathsf{clos}(\mathrm{KB})$,*
- $\mathsf{NNF}(\neg C) \in \mathsf{clos}(\mathrm{KB})$ *for any $\leq n\, R.C \in \mathsf{clos}(\mathrm{KB})$,*
- $\forall S.C \in \mathsf{clos}(\mathrm{KB})$ *if* $\mathsf{Trans}(S) \in \mathrm{KB}$ *and $S \sqsubseteq^* R$ for a role R with $\forall R.C \in \mathsf{clos}(\mathrm{KB})$.*

Moreover, let $\Omega(\mathrm{KB})$ denote the knowledge base obtained from KB *by removing all transitivity axioms $R \circ R \sqsubseteq R$, and adding the axiom $\forall R.C \sqsubseteq \forall S.(\forall S.C)$ for every $\forall R.C \in \mathsf{clos}(\mathrm{KB})$ with* $\mathsf{Trans}(S) \in \mathrm{KB}$ *and $S \sqsubseteq^* R$.*

Slightly generalising according results from [7], one can show that any $\mathcal{SHOIQ}^{\times}$ knowledge base KB is equisatisfiable to the $\mathcal{ALCHOIQ}^{\times}$ knowledge base $\Omega(\mathrm{KB})$. Therefore, we can reduce satisfiability checking in $\mathcal{SHOIQ}^{\times}$ to satisfiability checking in $\mathcal{ALCHOIQ}^{\times}$. Following a widely known approach taken in e.g. in [3] or more recently in [12], we can decide the latter problem by a reduction to C^2, the two-variable fragment of first-order logic with counting quantifiers for which this problem has been shown to be NExpTime-complete, even for binary coding of numbers [13]. Intuitively, C^2 admits all formulae of function-free first-order logic that contain at most two variable symbols, and which may also use the counting quantifiers $\exists_{\leq n}$, $\exists_{\geq n}$, and $\exists_{=n}$ for any $n > 0$. Such quantifiers impose the obvious restrictions on the number of individuals satisfying the quantified formula. Moreover, binary equality $\approx$ can be defined from those constructs. For formal details, see [13].

We transform $\mathcal{ALCHOIQ}^{\times}$ knowledge bases into C^2 by means of the recursive functions in Table 3, only slightly modifying the standard DL to FOL transformation given e.g. in [7], where further explanations can be found. Omitting the standard proof that $\pi(\mathrm{KB})$ is indeed equisatisfiable to KB (cf. [7]), we obtain the following result:

Theorem 10. *The problem of checking knowledge base satisfiability for $\mathcal{SHOIQ}^{\times}$ is* NExpTime*-complete, even for binary encoding of numbers.*

$\mathcal{SHOI}^{\times}$ is defined as the fragment of $\mathcal{SHOIQ}^{\times}$ without number restrictions. Obviously, transitivity can be eliminated as in the case of $\mathcal{SHOIQ}^{\times}$, hence we only need to consider the problem of checking satisfiability of $\mathcal{ALCHOI}^{\times}$ KBs. We now further reduce an $\mathcal{ALCHOI}^{\times}$ KB to an equisatisfiable $\mathcal{ALCHOI}$ KB in polynomial time. In addition to the standard negation normal form, we now require another normalisation step that simplifies the structure of KB by *flattening* it to a knowledge base FLAT(KB). This is achieved by transforming KB into negation normal form and exhaustively applying the following transformation rule: Select an outermost occurrence of $\mathsf{Q}R.D$ in KB, such that $\mathsf{Q} \in \{\exists, \forall\}$ and D is a non-atomic concept and substitute this occurrence with $\mathsf{Q}R.F$ where F is a fresh concept name, moreover, add $\neg F \sqcup D$ to the knowledge base. Obviously, this procedure terminates in polynomial time yielding a flat knowledge base FLAT(KB) all Tbox axioms of which are Boolean expressions over formulae of the form $\top$, $\bot$, A, $\neg A$, or $\mathsf{Q}R.A$ with A an atomic concept name. Moreover, one can show that any $\mathcal{ALCHOI}^{\times}$ knowledge base KB is equisatisfiable to FLAT(KB). Next we show how to eliminate concept products from such a knowledge base.

Lemma 11. *Consider a flattened $\mathcal{ALCHOI}^{\times}$ knowledge base* KB. *Let $C \times D \sqsubseteq R$ with $C, D \in \mathsf{N}_C$ be some concept product axiom contained in* KB. *Then a knowledge base* KB′ *that is equisatisfiable to* KB *is obtained as follows:*

- *delete the Rbox axiom $C \times D \sqsubseteq R$,*
- *add $C \sqsubseteq \exists S_1.\{o\}$ and $D \sqsubseteq \exists S_2.\{o\}$ where S_1, S_2 are fresh roles and o is a fresh individual name*
- *for all roles T with $R \sqsubseteq^* T$,substitute any occurrence of $\forall T.A$ by $\forall T.A \sqcap \forall S_1.\forall S_2^-.A$*

Applied iteratively, this step eliminates all concept products. Having a flat knowledge base is essential to ensure that this can be done in polynomial time. Based on the known ExpTime-completeness of $\mathcal{SHOI}$ [14], we now obtain the following result:

Theorem 12. *The satisfiability checking problem for $\mathcal{SHOI}^{\times}$ is* ExpTime*-complete.*

6 Conclusion

We have investigated the *concept product* as an expressive feature for description logics. It allows statements of the form $C \times D \sqsubseteq R$, expressing that all instances of the class C are related to all instances of D by the role R. While this construct can be simulated in $\mathcal{SROIQ}$ with a combination of inverse roles, nominals, and role inclusion axioms, we have shown that it can also be added to many weaker DLs that do not support such simulation. In particular, each of the extended DLs $\mathcal{EL}^{++\times}$, $\mathcal{SHOIQ}^{\times}$, and $\mathcal{SHOI}^{\times}$preserves its known upper complexity bound P, NExpTime, and ExpTime. For the tractable logic $\mathcal{EL}^{++\times}$, we also provided a detailed algorithm that might serve as a basis for extending existing $\mathcal{EL}^{++}$ implementations with that new feature.

Our results indicate that concept products, even though they are hitherto only available in $\mathcal{SROIQ}$, do in fact not have a strong negative impact on the difficulty of reasoning in simpler DLs. In contrast, the features used to simulate concept products in $\mathcal{SROIQ}$ may have much more negative impact in general. Inverse roles, for example, are known to render $\mathcal{EL}^{++}$ ExpTime-complete [10]. Since concept products provide a valuable modelling tool that can be applied in many scenarios, they appear as a natural candidate for future extensions of the DL-based Web Ontology Language OWL, possibly even in the ongoing OWL 2 effort.

Our results also entail a number of research questions for future works. First of all, one might ask what other features available (indirectly) in $\mathcal{SROIQ}$ could be easily ported back to less complex DLs. We are currently investigating a broad generalisation of concept products that appears to be rather promising in this respect.

But also the study of concept products as such bears various open problems. As remarked in Section 3, the simulation of concept products in $\mathcal{SROIQ}$ causes roles to be classified as non-simple. Yet, their use in number restrictions merely provides an alternative way of describing nominals, so that it might be conjectured that this restriction could be relaxed. Other obvious next steps are the investigation of concept products for $\mathcal{SHIQ}$ and $\mathcal{SHOQ}$, the direct treatment of concept products in further reasoning algorithms, and the possible augmentation of other popular tractable DLs with this feature. Moreover, implementations and concrete syntactical encodings for OWL would be important to make concept products usable in practice.

References

1. Horrocks, I., Kutz, O., Sattler, U.: The even more irresistible $\mathcal{SROIQ}$. In: Proc. of the 10th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR2006), AAAI Press (2006) 57–67
2. Baader, F., Brandt, S., Lutz, C.: Pushing the EL envelope. In: Proc. 19th Int. Joint Conf. on Artificial Intelligence (IJCAI-05), Edinburgh, UK, Morgan-Kaufmann Publishers (2005)
3. Borgida, A.: On the relative expressiveness of description logics and predicate logics. Artif. Intell. **82** (1996) 353–367
4. Kazakov, Y.: Saturation-Based Decision Procedures for Extensions of the Guarded Fragment. PhD thesis, Universität des Saarlandes, Saarbrücken, Germany (2006)
5. Lutz, C., Sattler, U.: Mary likes all cats. In Baader, F., Sattler, U., eds.: Description Logics. Volume 33 of CEUR Workshop Proceedings., CEUR-WS.org (2000) 213–226
6. Schmidt, R.A., Tishkovsky, D.: Using tableau to decide expressive description logics with role negation. In Aberer K. et al., ed.: ISWC/ASWC. Volume 4825 of LNCS., Springer (2007) 438–451
7. Motik, B.: Reasoning in Description Logics using Resolution and Deductive Databases. PhD thesis, Universität Karlsruhe (TH), Germany (2006)
8. Rudolph, S., Krötzsch, M., Hitzler, P.: All elephants are bigger than all mice. Technical report, Institute AIFB, University of Karlsruhe (2008) Extended online version available at `http://korrekt.org/page/Elephants`.
9. Baader, F. et al., ed.: The Description Logic Handbook: Theory, Implementation and Applications. Cambridge University Press (2007)
10. Krisnadhi, A., Lutz, C.: Data complexity in the $\mathcal{EL}$ family of description logics. In Dershowitz, N., Voronkov, A., eds.: Proc. 14th Int. Conf. on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR2007). Volume 4790 of LNAI., Springer (2007) 333–347
11. Krötzsch, M., Rudolph, S., Hitzler, P.: Conjunctive queries for a tractable fragment of owl 1.1. In Aberer, K. et al., ed.: Proc. 6th Int. Semantic Web Conference (ISWC 2007). Volume 4825 of LNCS., Springer (2007) 310–323
12. Tobies, S.: Complexity Results and Practical Algorithms for Logics in Knowledge Representation. PhD thesis, RWTH Aachen, Germany (2001)
13. Pratt-Hartmann, I.: Complexity of the two-variable fragment with counting quantifiers. Journal of Logic, Language and Information **14** (2005) 369–395
14. Hladik, J.: A tableau system for the description logic SHIO. In Sattler, U., ed.: IJCAR Doctoral Programme. Volume 106 of CEUR Workshop Proceedings., CEUR-WS.org (2004)