Journal on Relational Methods in Computer Science, Vol. 1, 2004, pp. 251 - 276

A Survey of Decidable First-Order Fragments and Description Logics

U. Hustadt[1], R. A. Schmidt[2], and L. Georgieva[3]

[1] University of Liverpool, UK, U.Hustadt@csc.liv.ac.uk
[2] University of Manchester, UK, Renate.Schmidt@manchester.ac.uk
[3] Heriot-Watt University, Edinburgh, UK, L.Georgieva@hw.ac.uk

Abstract. The guarded fragment and its extensions and subfragments are often considered as a framework for investigating the properties of description logics. There are also other, some less well-known, decidable fragments of first-order logic which all have in common that they generalise the standard translation of $\mathcal{ALC}$ to first-order logic. We provide a short survey of some of these fragments and motivate why they are interesting with respect to description logics, mentioning also connections to other non-classical logics.

1 Introduction

It is well-known that a wide range of description and modal logics can be translated into first-order logic in a satisfiability-preserving way, and, can thus be seen as notational variants of 'modal' fragments of first-order logic. Viewing them in this way it is natural to ask, whether it is possible to generalise these fragments of first-order logic while preserving the 'good' properties of description and modal logics, most importantly, preserving the decidability of the satisfiability problem. The best-known fragment of first-order logic which has been introduced in an attempt to answer this question is the guarded fragment [2]. The guarded fragment and its extensions and subfragments are often considered as a framework for investigating the properties of modal and description logics [2, 19, 21, 33]. While these investigations are important and provide useful insights, the guarded fragment is by far not the only fragment of first-order logic which extends 'modal' fragments of first-order logic and has good properties. Alternative fragments include the two-variable fragment [40], which is also often associated with modal and description logics, but also the more expressive class K of Maslov [37], Quine's

Received by the editors March 25, 2004, and, in revised form, October 04, 2004.
Published on December 10, 2004.

fluted logic [51, 52], and the positive restrictive quantification fragment PRQ [9]. This paper is a survey of these fragments, we discuss the relationship between these fragments and focus also on their relationships to expressive description logics extending the description logic $\mathcal{ALC}$ and their modal counterparts.

Description logics provide formalisms for representing and reasoning about knowledge in a given domain of application. Description logics have been developed in the context of knowledge representation following the tradition of semantic networks [50] and frame systems [39]. Unlike the early knowledge representation formalisms which are a bit ad hoc from a modern perspective, description logics have a well-defined semantics which allows us (i) to formally define various inferential services for these logics, (ii) to define what it means for those inferential services to be sound and complete, (iii) to investigate calculi and algorithms for providing those inferential services, and (iv) to investigate the computational complexity of those inferential services.

Since their invention in the mid-eighties, the advance in the area has been rapid. On the theoretical side the decidability and computational complexity of description logics have been extensively studied, and on the practical side fast sophisticated description logic reasoners are now available. In recent years description logics have also been applied in a number of subfields of computer science, including data integration, knowledge representation and ontology modelling for the semantic web. A source for reference and additional information on description logics is [3].

Many description logics of varying expressivity have been introduced and studied in the literature. In this paper we consider a class of description logics with a common language but which differ in the operators they provide. We are particularly interested in description logics which map to first-order logic, because first-order logic provides the best framework for studying and comparing first-order definable logics. Mapping description logics and modal logics to first-order logic also connects them with the powerful methodologies developed in the area of automated reasoning. Because we want to leverage existing methods of automated reasoning, we are going to focus less on special-purpose techniques which are usually favoured in the description logic community. These are covered extensively in many survey papers and textbooks (e.g. [3]). Instead we are going to emphasise general-purpose techniques and results which are not as widely known and appreciated in the area as they should be.

Because of the close connection between description logics and relation algebra, the class of description logics discussed in this paper are of special interest to the readership of this journal. By following the ideas of [44] and exploiting results in [7] it is not difficult to see that all the description logics discussed in this paper can be interpreted in the algebraic framework of Tarski's relation algebra [66], or moderate extensions of relation algebra. In fact the kind of description logics we discuss can be very naturally interpreted as the join of a Boolean algebra and a relation algebra or a Kleene algebra. Algebraically description logics are forms of modal algebras, Boolean modules, dynamic algebras or Peirce algebras [7, 8, 57, 59]. Many decidable description logics and extended modal logics discussed in this paper actually correspond directly to reducts of Peirce algebra.

The paper is structured as follows. In Section 2 we define a very general description logic $\mathcal{DL}$ which includes most of the operators commonly used in the description logic literature. We then consider some sublogics of $\mathcal{DL}$, including the well-known description logic $\mathcal{ALC}$. We also discuss the relation of sublogics of $\mathcal{DL}$ to modal logics and relation algebras. In Section 3 we focus on four decidable fragments of first-order logic, namely, the guarded fragment, the two-variable fragment, Maslov's class $\overline{\mathrm{K}}$, and fluted logic. We discuss the relationship of these fragments to each other and to the description logics defined in Section 2. Various translation mappings including the relational translation, the functional translation, and the optimised functional translation are used. A relatively new field of research in modal logics are many-dimensional modal logics [15]. A particular instance are modalized description logics. Section 4 relates the modalized description logic $K_{\mathcal{ALC}}$, whose translation does not fall into any of the first-order fragments considered in Section 3, to the positive restrictive quantification fragment PRQ.

2 The description logic $\mathcal{DL}$ and its sublogics

For the purposes of this paper we define a description logic called $\mathcal{DL}$. It contains a superset of the operators available in most description logics found in the literature and is the strongest description logic considered in this paper. The language of $\mathcal{DL}$ is a sublanguage of the *universal terminological logic* defined by Patel-Schneider [45].

Given mutually disjoint sets *concept symbols*, *role symbols*, and *object symbols*, the sets of *concept terms* (or just *concepts*) and *role terms* (or just *roles*) are inductively defined as follows. Every concept symbol is a concept term and every role symbol is a role term. Assume that C, D are concepts and R, S are roles. Then complex concept and role terms are defined by induction using the constructors of Table 1.[1] The set of all concepts and roles forms the *term language* of the description logic $\mathcal{DL}$, and an element of the term language is called a *(terminological) term* or a *(terminological) expression.*

[1] The notation used is the same as [5], which varies from notation used in earlier literature, e.g. [45].

Concept terms		Role terms	
$\top$	top concept	∇	top role
$\bot$	bottom concept	$\triangle$	bottom role
$C \sqcap D$	concept intersection	$\mathrm{id}(C)$	identity role on C
$C \sqcup D$	concept union	$R \sqcap S$	role intersection
$\neg C$	concept complement	$R \sqcup S$	role union
$\forall R\,.\,C$	universal restriction	$R \circ S$	role composition
$\exists R\,.\,\top$	limited existential restriction	$\neg R$	role complement
$\exists R\,.\,C$	existential restriction	$R^{\smile}$	role inverse
$\exists_{\geq n} R, \exists_{\leq n} R$	number restrictions	R^{+}	transitive role closure
$\exists_{\geq n} R.C, \exists_{\leq n} R.C$	qualified number restrictions	$R \restriction C$	domain restriction
$(R \subseteq S)$	inclusion role value maps	$R \downharpoonleft C$	range restriction
$(R = S)$	equality role value maps		

Table 1. Constructors of $\mathcal{DL}$

Obviously, not all the operators of $\mathcal{DL}$ are independent of each other. For example, in the presence of the range restriction operator, $\exists_{\geq n} R.C$ and $\exists_{\leq n} R.C$ can be expressed by $\exists_{\geq n} R \downharpoonleft C$ and $\exists_{\leq n} R \downharpoonleft C$, respectively. Additional operators can also be defined in terms of the already existing ones. For example, the *reflexive-transitive role closure* R^{*} can be expressed by $\mathrm{id}(\top) \sqcup R^{+}$.

The set of sentences S over the term language of $\mathcal{DL}$ is divided into *terminological sentences*, also called *terminological axioms*, and *assertional sentences*. If C and D are concepts, and R and S are roles, then $C \mathrel{\dot\sqsubseteq} D$, $C \doteq D$, $R \mathrel{\dot\sqsubseteq} S$, and $R \doteq S$ are terminological sentences. If C is a concept, R is a role, and a, b are object symbols then $a{:}C$ and $(a,b){:}R$ are *concept assertions* and *role assertions*, respectively, and are collectively referred to as assertional sentences. A *knowledge base* is a finite set of terminological and assertional sentences. The set of assertional sentences of a knowledge base is usually called the *ABox*. The set of terminological sentences of a knowledge base is called the *TBox*.

Description logics commonly have a set-theoretic semantics. Intuitively, concepts are interpreted as sets of individuals and roles as sets of ordered pairs of individuals. Each objects symbols is interpreted by a unique individual, but it is not necessarily the case that each individual is the interpretation of some object symbol, in particular, the set of all individuals does not need to be finite. Also, it is not assumed that our knowledge about individuals, as expressed by assertional and terminological sentences is complete, for example, the absence of a sentence $a{:}C$ in a knowledge base does not imply that $a{:}\neg C$ is true.

Formally, the semantics of $\mathcal{DL}$ is defined by a *terminological interpretation* which is a pair $(\mathcal{D}, \mathcal{I})$ consisting of a domain $\mathcal{D}$ and an interpretation function $\mathcal{I}$. It maps the object symbols to elements of $\mathcal{D}$, the concept symbols to subsets of $\mathcal{D}$ and the role symbols to subsets of $\mathcal{D} \times \mathcal{D}$. It is a standard requirement

$$
\begin{array}{lll}
\mathcal{I}(\top) = \mathcal{D} & \mathcal{I}(\bot) = \emptyset & \\
\mathcal{I}(C \sqcap D) = \mathcal{I}(C) \cap \mathcal{I}(D) & \mathcal{I}(C \sqcup D) = \mathcal{I}(C) \cup \mathcal{I}(D) & \mathcal{I}(\neg C) = \mathcal{D} \setminus \mathcal{I}(C) \\
\mathcal{I}(\forall R\,.\,C) = \{x \in \mathcal{D} \mid \forall y((x,y) \in \mathcal{I}(R) \rightarrow y \in \mathcal{I}(C))\} & & \\
\mathcal{I}(\exists R\,.\,C) = \{x \in \mathcal{D} \mid \exists y((x,y) \in \mathcal{I}(R) \wedge y \in \mathcal{I}(C))\} & & \\
\mathcal{I}(\exists_{\geq n} R) = \{x \in \mathcal{D} \mid |\{y \mid (x,y) \in \mathcal{I}(R)\}| \geq n\} & & \\
\mathcal{I}(\exists_{\leq n} R) = \{x \in \mathcal{D} \mid |\{y \mid (x,y) \in \mathcal{I}(R)\}| \leq n\} & & \\
\mathcal{I}(\exists_{\geq n} R.C) = \{x \in \mathcal{D} \mid |\{y \mid (x,y) \in \mathcal{I}(R) \wedge y \in \mathcal{I}(C)\}| \geq n\} & & \\
\mathcal{I}(\exists_{\leq n} R.C) = \{x \in \mathcal{D} \mid |\{y \mid (x,y) \in \mathcal{I}(R) \wedge y \in \mathcal{I}(C)\}| \leq n\} & & \\
\mathcal{I}(R \subseteq S) = \{x \in \mathcal{D} \mid \forall y((x,y) \in \mathcal{I}(R) \rightarrow (x,y) \in \mathcal{I}(S))\} & & \\
\mathcal{I}(R = S) = \{x \in \mathcal{D} \mid \forall y((x,y) \in \mathcal{I}(R) \leftrightarrow (x,y) \in \mathcal{I}(S))\} & & \\
\mathcal{I}(\bigtriangledown) = \mathcal{D} \times \mathcal{D} & \mathcal{I}(\bigtriangleup) = \emptyset & \mathcal{I}(R^+) = \mathcal{I}(R)^+ \\
\mathcal{I}(R \sqcap S) = \mathcal{I}(R) \cap \mathcal{I}(S) & \mathcal{I}(R \sqcup S) = \mathcal{I}(R) \cup \mathcal{I}(S) & \mathcal{I}(\neg R) = (\mathcal{D} \times \mathcal{D}) \setminus R \\
\mathcal{I}(\mathrm{id}(C)) = \{(x,x) \in \mathcal{D} \times \mathcal{D} \mid x \in \mathcal{I}(C)\} & & \\
\mathcal{I}(R \circ S) = \{(x,y) \in \mathcal{D} \times \mathcal{D} \mid \exists z((x,z) \in \mathcal{I}(R) \wedge (z,y) \in \mathcal{I}(S))\} & & \\
\mathcal{I}(R^{\smile}) = \{(x,y) \in \mathcal{D} \times \mathcal{D} \mid (y,x) \in \mathcal{I}(R)\} & & \\
\mathcal{I}(R \restriction C) = \{(x,y) \in \mathcal{I}(R) \mid x \in \mathcal{I}(C)\} & & \\
\mathcal{I}(R \downharpoonleft C) = \{(x,y) \in \mathcal{I}(R) \mid y \in \mathcal{I}(C)\} & &
\end{array}
$$

Table 2. Semantics of $\mathcal{DL}$

that $\mathcal{I}$ obeys the *unique name assumption*, that is, $\mathcal{I}(a) \neq \mathcal{I}(b)$ holds for every pair of object symbols $a \neq b \in \mathsf{O}$. The interpretation function $\mathcal{I}$ extends in a natural way to complex concepts and roles, as defined in Table 2. Let $(\mathcal{D}, \mathcal{I})$ be a terminological interpretation. The satisfiability relation $\models$ is defined by:

$$
\begin{array}{lll}
(\mathcal{D},\mathcal{I}) \models a{:}C & \text{iff} & \mathcal{I}(a) \in \mathcal{I}(C) \\
(\mathcal{D},\mathcal{I}) \models C \mathrel{\dot\sqsubseteq} D & \text{iff} & \mathcal{I}(C) \subseteq \mathcal{I}(D) \\
(\mathcal{D},\mathcal{I}) \models C \doteq D & \text{iff} & \mathcal{I}(C) = \mathcal{I}(D) \\
(\mathcal{D},\mathcal{I}) \models (a,b){:}R & \text{iff} & (\mathcal{I}(a),\mathcal{I}(b)) \in \mathcal{I}(R) \\
(\mathcal{D},\mathcal{I}) \models R \mathrel{\dot\sqsubseteq} S & \text{iff} & \mathcal{I}(R) \subseteq \mathcal{I}(S) \\
(\mathcal{D},\mathcal{I}) \models R \doteq S & \text{iff} & \mathcal{I}(R) = \mathcal{I}(S)
\end{array}
$$

Let Γ be a knowledge base. We say that $(\mathcal{D}, \mathcal{I})$ *satisfies* Γ, written $(\mathcal{D}, \mathcal{I}) \models \Gamma$, if $(\mathcal{D}, \mathcal{I})$ satisfies every sentence in Γ. In this case, $(\mathcal{D}, \mathcal{I})$ is a *(terminological) model* of Γ. We say that a knowledge base Γ *entails* a sentence α, written $\Gamma \models \alpha$, if every model of Γ satisfies α. If Γ is empty then we write $\models \alpha$ instead of $\emptyset \models \alpha$.

An occurrence of a subexpression is a *positive occurrence* if it is one inside the scope of an even number of (explicit or implicit) negations (complements), and an occurrence is a *negative occurrence* if it is one inside the scope of an odd number of negations. For example, both occurrences of the subformula $\neg C \sqcap D$

in $(\exists R^{\smile} . (\neg C \sqcap D)) \sqcap (\forall R \sqcup S . (\neg C \sqcap D))$ have positive polarity, $R^{\smile}$ has positive polarity, and $R \sqcup S$ has negative polarity.

A concept C is *coherent* or *satisfiable* iff there exists a terminological interpretation $(\mathcal{D}, \mathcal{I})$ such that $\mathcal{I}(C)$ is non-empty. Otherwise, C is *incoherent* or *unsatisfiable.* A concept C is *coherent with respect to* Γ if there exists a terminological model $(\mathcal{D}, \mathcal{I})$ of Γ such that $\mathcal{I}(C)$ is non-empty.

Description logic systems provide a variety of inferential services; these include:

1. *Subsumption of concepts*: determine whether $\models C \dot{\sqsubseteq} D$ holds for concepts C and D. Then C is said to be *subsumed by* D, or D is said to *subsume* C.
2. *Subsumption of concepts with respect to a TBox* T: determine whether $T \models C \dot{\sqsubseteq} D$ holds.
3. *Equivalence of concepts (with respect to a TBox* T*)*: determine whether C subsumes D and D subsumes C at the same time for two concepts C and D (with respect to a TBox T).
4. *Classification of a TBox* T: determine for all concept symbols A and B occurring in T whether A subsumes B or B subsumes A with respect to T.
5. *Satisfiability of a concept (with respect to a TBox* T*)*: determine for a concept C whether it is satisfiable (with respect to T).
6. *Consistency of a knowledge base* Γ: determine whether Γ is satisfiable.
7. *Coherence of a knowledge base* Γ: compute all unsatisfiable concept names for Γ.
8. *Instance checking*: determine whether a given knowledge base Γ entails a given assertional sentence of the form $a{:}C$.
9. *Realization*: compute for an object symbol a in a knowledge base Γ the set of minimal concept symbols A with respect to the subsumption relation such that $\Gamma \models a{:}A$.
10. *Retrieval*: compute for a given concept C in a knowledge base Γ those object symbols a such that Γ entails $a{:}C$.

All these inferential services can be realized with satisfiability tests of knowledge bases. For example, the problem whether $\Gamma \models C \dot{\sqsubseteq} D$ holds, is equivalent to the problem whether $\Gamma \cup \{a{:}C, a{:}\neg D\}$ is satisfiable, where a is some arbitrary object symbol.[2]

All these inferential services are undecidable for $\mathcal{DL}$. This is a consequence of a number of negative results from the literature. Schild [55] has shown that the subsumption problem for a sublanguage of $\mathcal{DL}$ containing only role intersection, role complement, role composition, and the identity role is undecidable. In [63],

[2] Note that all the inferential services are restricted to the consideration of concepts and objects, although it is straightforward to define inferential services on roles analogous to 1–10 above.

Schmidt-Schauß has shown that the subsumption problem for a sublanguage of $\mathcal{DL}$ containing only concept intersection, universal and existential restrictions, role composition, and role value maps is undecidable. From the literature on PDL-like modal logics and relation algebras it is known that role composition and role complement together with role intersection or union lead to undecidability [1]. Since the decidability of inferential services is one of the major design goals of description logic systems, the negative results by Schild and Schmidt-Schauß have caused the focus of research to shift away from description logics with role-forming operators and problems involving complex roles. Recent results [31, 32, 38] indicate however a renewed interest in such logics.

In the literature on description logics, a wide variety of sublogics of $\mathcal{DL}$ are considered [4, 5, 10]. The description logic $\mathcal{ALC}$ [64] is the sublogic of $\mathcal{DL}$ limited to the top and bottom concept, concept complement, concept intersection, and existential quantification. Weaker logics than $\mathcal{ALC}$ have also been considered, but since we are here interested in relationships to modal logics and expressive first-order fragments we limit our discussion to $\mathcal{ALC}$ and extensions of $\mathcal{ALC}$. One possibility of extending $\mathcal{ALC}$ is the addition of role-forming operators. We denote the extension of $\mathcal{ALC}$ by role-forming operators $r_1, \ldots, r_n$ by $\mathcal{ALC}(r_1, \ldots, r_n)$. Although many applications of description logics are far removed from the motivations and origins of modal logic, the two types of logics can be regarded as equal. This is the case both from a mathematical perspective and a computational perspective. It is well-known that the description logic $\mathcal{ALC}$ can be viewed as a syntactical variant of basic multi-modal logic [56]. $\mathcal{ALC}$ augmented with conjunction and negation on roles, i.e. $\mathcal{ALC}(\neg, \sqcap)$, corresponds essentially to the Boolean modal logic *BML* of Gargov and Passy [18]. *BML* is an extended modal logic similar to propositional dynamic logic. More precisely, *BML* is the modal logic defined over families of binary relations closed under union, intersection, and complementation. The relationship between an extension of *BML*, called Peirce logic, and description logics has been studied in [7, 57] in an algebraic setting. Peirce logic is the modal logic defined over families of binary relations closed under the operations of relation algebras and a cyclindrification operation [62]. It is not difficult to show that Peirce logic is expressively equivalent to $\mathcal{ALC}(\neg, \sqcap, ^{\smile}, \circ, \mathrm{id})$ and $\mathcal{ALC}(\neg, \sqcap, ^{\smile}, \circ, \mathrm{id}(\triangledown), \upharpoonright)$. Propositional dynamic logic *PDL* [25, 47] is equivalent to the extension of $\mathcal{ALC}$ in which the roles occurring in the existential and universal restrictions are built using the constructors: composition, union, transitive role closure and the identity role operator.

Subsequently, we focus on a decidable description logic called $\mathcal{ALB}$ (short for 'attribute language with Boolean algebras on concepts and roles') [31]. $\mathcal{ALB}$ extends $\mathcal{ALC}$ with the top role, role complement, role intersection, role union, role inverse, domain restriction, and range restriction. $\mathcal{ALB}$ is equivalent to

	Concept constructors	Role constructors
$\mathcal{ALC}$ [64]	$\neg, \sqcap, \exists$ $(\top, \bot, \sqcup, \forall)$	
BML [18]	$\neg, \sqcap, \exists$ $(\top, \bot, \sqcup, \forall)$	$\neg, \sqcap$ $(\nabla, \triangle, \sqcup)$
$\mathcal{ALB}$ [31]	$\neg, \sqcap, \exists$ $(\top, \bot, \sqcup, \forall)$	$\neg, \sqcap, ^{\smile}, \upharpoonright$ $(\nabla, \triangle, \sqcup, \downharpoonleft)$
Peirce logic [62]	$\neg, \sqcap, \exists$ $(\top, \bot, \sqcup, \forall)$	$\neg, \sqcap, ^{\smile}, \circ, \mathrm{id}$ $(\nabla, \triangle, \sqcup, \upharpoonright, \downharpoonleft)$
PDL [47]	$\neg, \sqcap, \exists$ $(\top, \bot, \sqcup, \forall)$	$\sqcup, \circ, ^{+}, \mathrm{id}(C)$ (∇)

Table 3. Some sublogics of $\mathcal{DL}$

$\mathcal{ALC}(\neg, \sqcap, ^{\smile}, \upharpoonright)$, since role intersection and range restriction can be defined in terms of the other operators. Clearly, it is also possible to define the role value map operators within $\mathcal{ALB}$.

Table 3 summarises the definitions of $\mathcal{ALC}$, BML, $\mathcal{ALB}$, Peirce logic, and PDL by listing the concept and role forming operators available in them (definable operators are listed in brackets).

For ease of presentation, in the remainder of the paper we consider only the consistency test operation for knowledge bases. As mentioned above this does not restrict the generality of the observations and results.

3 Decidable first-order fragments

The definition of the standard semantics of $\mathcal{DL}$ indicates that the description logic $\mathcal{ALB}$ can be considered as a fragment of first-order logic. The concept and role symbols can be seen as unary or binary predicate symbols, concept terms as abbreviations for formulae with one free variable, and role terms as abbreviations for formulae with two free variables. This is made precise by the standard translation π of $\mathcal{ALB}$-expressions into first-order logic formulae defined in Table 4. The symbols X and Y are meta-variables for variables and constants, and Q_A and Q_P denote unary and binary predicate symbols, which are uniquely associated with a concept symbol A and a role symbol P, respectively. Let Π, as specified in Table 4, denote the translation function of $\mathcal{ALB}$-sentences into first-order logic. For all sentences α, $\Pi(\alpha)$ is a closed first-order formula. Finally, extend Π to (finite) sets of sentences mapping knowledge bases to a conjunction of first-order logic formulae. Note that in the absence of the identity role on concepts and number restrictions, the unique name assumption does not affect the satisfiability of a knowledge base. Therefore, it is not necessary to incorporate formulae resulting from the translation of the unique name assumption into Π.

It is not difficult to see that if Γ is a knowledge base and α a sentence, then Γ entails α if and only if $\Pi(\Gamma)$ entails $\Pi(\alpha)$ in first-order logic. It is clear therefore

Translation π of $\mathcal{ALB}$-expressions:

$$\begin{aligned}
\pi(A,X) &= Q_A(X) & \pi(C \sqcap D, X) &= \pi(C,X) \wedge \pi(D,X)\\
\pi(\neg C,X) &= \neg\pi(C,X) & \pi(C \sqcup D, X) &= \pi(C,X) \vee \pi(D,X)\\
\pi(\top,X) &= \top & \pi(\forall R\,.\,C,X) &= \forall y(\pi(R,X,y) \rightarrow \pi(C,y))\\
\pi(\bot,X) &= \bot & \pi(\exists R\,.\,C,X) &= \exists y(\pi(R,X,y) \wedge \pi(C,y))\\
\pi(\nabla,X,Y) &= \top & \pi((R{\subseteq}S),X) &= \forall y(\pi(R,X,y) \rightarrow \pi(S,X,y))\\
\pi(\triangle,X,Y) &= \bot & \pi((R{=}S),X) &= \forall y(\pi(R,X,y) \leftrightarrow \pi(S,X,y))\\
\pi(P,X,Y) &= Q_P(X,Y) & \pi(R \sqcap S,X,Y) &= \pi(R,X,Y) \wedge \pi(S,X,Y)\\
\pi(\neg R,X,Y) &= \neg\pi(R,X,Y) & \pi(R \sqcup S,X,Y) &= \pi(R,X,Y) \vee \pi(S,X,Y)\\
\pi(R^{\smile},X,Y) &= \pi(R,Y,X) & \pi(R{\upharpoonright}C,X,Y) &= \pi(R,X,Y) \wedge \pi(C,X)\\
& & \pi(R{\downharpoonright}C,X,Y) &= \pi(R,X,Y) \wedge \pi(C,Y)
\end{aligned}$$

Translation Π of terminological sentences:

$$\begin{aligned}
\Pi(C \mathrel{\dot\sqsubseteq} D) &= \forall x(\pi(C,x) \rightarrow \pi(D,x)) & \Pi(R \mathrel{\dot\sqsubseteq} S) &= \forall xy(\pi(R,x,y) \rightarrow \pi(S,x,y))\\
\Pi(C \doteq D) &= \forall x(\pi(C,x) \leftrightarrow \pi(D,x)) & \Pi(R \doteq S) &= \forall xy(\pi(R,x,y) \leftrightarrow \pi(S,x,y))\\
\Pi(a{:}C) &= \pi(C,a) & \Pi((a,b){:}R) &= \pi(R,a,b).
\end{aligned}$$

Table 4. The first-order translation of $\mathcal{ALB}$

that, using the embedding Π all common inferential services in description logics are reducible to testing satisfiability in first-order logic.

In the following we consider the relationship between $\mathcal{ALB}$, the two-variable fragment FO^2 of first-order logic, the guarded fragment, fluted logic, and the dual of Maslov's class K. We comment on the inference methods available for these fragments, the complexity of the satisfiability problem in these logics, and we also discuss the relationship between the logics.

To establish the relationship of $\mathcal{ALB}$ to fragments of first-order logic, we use the standard translation augmented by structural transformation to definitional form. (The latter is a well-known techniques also known as definitional transformation, renaming or Scott reduction; the reader unfamiliar with structural transformation and definitional forms may wish to refer to Appendix A.) It is therefore important to be aware of the form of first-order formulae in the definitional form of the translation of $\mathcal{ALB}$-terminologies. First, note that every position of a subexpression in a knowledge base Γ can be mapped to exactly one position in the first-order translation $\Pi(\Gamma)$. Let $\mathrm{Pos}_\Gamma(\Pi(\Gamma))$ be the set of all such positions in $\Pi(\Gamma)$, which correspond to positions of subexpressions in Γ. Let Ξ denote the transformation taking $\Pi(\Gamma)$ to the *definitional form* $\mathrm{Def}_\Lambda(\Pi(\Gamma))$ *of* $\Pi(\Gamma)$ introducing new symbols for all formulae occurring in $\Lambda = \mathrm{Pos}_\Gamma(\Pi(\Gamma))$. Due to the correspondence between positions in Γ and $\Pi(\Gamma)$, it is convenient to denote the predicate symbol used in the definition of a subformula ψ of $\Pi(\Gamma)$

TBox concept definitions/restrictions and additional definitions introduced by the structural transformation Ξ	TBox role definitions/restrictions and additional definitions introduced by the structural transformation Ξ
$\forall x\,(Q_{\top}(x) \leftrightarrow \top)$	$\forall xy\,(Q_{\bigtriangledown}(x,y) \leftrightarrow \top)$
$\forall x\,(Q_{\bot}(x) \leftrightarrow \bot)$	$\forall xy\,(Q_{\triangle}(x,y) \leftrightarrow \bot)$
$\forall x\,(Q_{\neg C}(x) \leftrightarrow \neg Q_C(x))$	$\forall xy\,(Q_{\neg R}(x,y) \leftrightarrow \neg Q_R(x,y))$
$\forall x\,(Q_{C\sqcap D}(x) \leftrightarrow (Q_C(x) \land Q_D(x)))$	$\forall xy\,(Q_{R\sqcap S}(x,y) \leftrightarrow (Q_R(x,y) \land Q_D(x,y)))$
$\forall x\,(Q_{C\sqcup D}(x) \leftrightarrow (Q_C(x) \lor Q_D(x)))$	$\forall xy\,(Q_{R\sqcup S}(x,y) \leftrightarrow (Q_R(x,y) \lor Q_S(x,y)))$
$\forall x(Q_{\forall R\,.\,C}(x) \leftrightarrow \forall y(Q_R(x,y) \rightarrow Q_C(y)))$	$\forall xy\,(Q_{R^{\smile}}(x,y) \leftrightarrow Q_R(y,x))$
$\forall x(Q_{\exists R\,.\,C}(x) \leftrightarrow \exists y(Q_R(x,y) \land Q_C(y)))$	$\forall xy\,(Q_{R\upharpoonright C}(x,y) \leftrightarrow (Q_R(x,y) \land Q_C(x)))$
$\forall x\,(Q_{R\subseteq S}(x) \leftrightarrow \forall y(Q_R(x,y) \rightarrow Q_S(x,y)))$	$\forall xy\,(Q_{R\downharpoonright C}(x,y) \leftrightarrow (Q_R(x,y) \land Q_C(y)))$
$\forall x\,(Q_{R\subseteq S}(x) \leftrightarrow \forall y(Q_R(x,y) \leftrightarrow Q_S(x,y)))$	$\forall xy\,(Q_R(x,y) \rightarrow Q_S(x,y))$
$\forall x(Q_C(x) \rightarrow Q_D(x))$	$\forall xy\,(Q_R(x,y) \leftrightarrow Q_S(x,y))$
$\forall x(Q_C(x) \leftrightarrow Q_D(x))$	
ABox	
$Q_C(a)$	$Q_R(a,b)$

Table 5. Formulae in definitional form

by Q_C (or Q_R) where C (or R) is the concept term (or role term) such that $\psi = \pi(C, X)$ (or $\psi = \pi(R, X, Y)$). Table 5 lists all possible forms of formulae that may occur in $\Xi\Pi(\Gamma)$ for a knowledge base Γ.

3.1 The guarded fragment

It is well-known that π embeds $\mathcal{ALC}$-concepts into the *guarded fragment GF* introduced by Andréka, Németi and Van Benthem [2]. Formally, the formulae of the guarded fragment are function-free first-order logic formulae which are inductively defined as follows.

1. $\top$ and $\bot$ are in GF.
2. If φ is an atomic formula, then φ is in GF.
3. GF is closed under Boolean connectives.
4. If φ is in GF and A is an atom for which every free variable of φ is among the arguments of A, then $\forall\overline{x}(A \rightarrow \varphi)$ is in GF and $\exists\overline{x}(A \land \varphi)$ is in GF, for every sequence $\overline{x}$ of variables. A is called a *guard atom*.

The guarded fragment is the smallest fragment of first-order logic containing all the guarded formulae. The problem of deciding the satisfiability of guarded formulae is of double exponential time and space complexity [11, 16, 22]. This is the case under the assumption that there are no bounds on the arity of predicate symbols and the number of free variables in guarded formulae. If there is a bound on the number of variables or a bound on the arity of predicate symbols that can occur in guarded fragment formulae, then the complexity of the satisfiability problem is ExpTime-complete [22].

It is of theoretical interest that a variety of inference techniques have been developed for the guarded fragment. The fragment and its extensions have been shown decidable using ordered resolution [11, 16], alternating automata [22], tableau methods [27], or embedding into monadic second-order logic [17].

In [13] it is shown that in fact not only any $\mathcal{ALC}$-concept but also any $\mathcal{ALC}(\sqcap, \sqcup, ^{\smile})$-concept C translates into a guarded formula if the structural transformation mapping Def_{Λ} of the first-order formula $\pi(C, x)$ is with respect to $\Lambda = \{\lambda \mid \lambda \text{ is a position in } \pi(C, x) \text{ of a formula of the form } \forall\overline{x}\,(G \rightarrow \psi)\}$. This result can be more easily obtained if instead of the guarded fragment of first-order logic, the corresponding clausal class introduced by Ganzinger and De Nivelle [16] is considered. The definition of this class makes use of the notions of shallow terms, simple literals and simple clauses, which are defined as follows. A term is *shallow* iff either it is a variable or a term $f(t_1, \ldots, t_n)$ such that each t_i is a variable or a constant ($0 \leq n$, $1 \leq i \leq n$). A literal L is *simple* iff each term in L is shallow, and a clause C is *simple* iff all literals in C are simple. A simple clause C is *guarded* iff it satisfies one of the following conditions:

1. C is a positive, non-functional, single-variable clause.
2. Every functional subterm of C contains all the variables of C, and, if C is non-ground, then C contains a non-functional negative literal, called a *guard*, which contains all the variables of C.

The *class of all guarded clauses* is denoted by GC. The class GC is in fact slightly more general than the class of guarded formulae.

Inspection of the clausal forms of the formulae in Table 5 reveals that only the translation of the role complement operator in the definition of $Q_{\neg R}$, that is, $\forall xy\,(Q_{\neg R}(x, y) \leftrightarrow \neg Q_R(x, y))$, results in a clause which is not guarded, that is, the clause $Q_{\neg R}(x, y) \vee Q_R(x, y)$ is the only positive non-ground clause in the table. In fact, this clause occurs only if there is a negative occurrence of $\neg R$ in the knowledge base. Thus, we can strengthen the result of De Nivelle, Schmidt and Hustadt [13] as follows. Let Γ be a knowledge base in the extension of $\mathcal{ALC}$ by role union, role intersection, role inverse, domain restriction and range restriction (which is identical to $\mathcal{ALC}(\sqcap, \sqcup, ^{\smile}, \upharpoonright)$, i.e. $\mathcal{ALB}$ without role complement). Then $\Xi\Pi(\Gamma)$ consists only of guarded clauses.

As we have just seen, $\mathcal{ALC}(\neg)$ and its extensions contain concepts whose translation do not result in guarded clauses or guarded formulae. An example of an $\mathcal{ALC}(\neg, \sqcap)$-concept whose translation is not a guarded formula is $\forall\neg(\mathit{likes} \sqcap \mathit{eats})\,.\,\neg\mathit{Cheese}$. This concept can be interpreted as representing the set of cheese lovers, that is, the set of individuals who like and eat all (kinds of) cheeses.

One important property of the guarded fragment and the class of guarded clauses is that the guards, that is the atoms we obtain from the translation of roles, are always positive.

3.2 The two-variable first-order fragment

As can be seen from Table 5 all the formulae in $\Xi\Pi(\Gamma)$, where Γ is an $\mathcal{ALB}$-knowledge base can be expressed by first-order formulae with just two variables, that is, $\mathcal{ALB}$-knowledge bases can be translated into the *two-variable fragment* FO^2 consisting of those formulae of first-order logic that can be written using only two variables. Decidability of FO^2 without equality was first shown by Scott [65] and for FO^2 with equality by Mortimer [40]. It has been observed in various places that FO^2 without equality (actually the reduction to the Scott class, see below) can be decided by standard ordering refinements of resolution, cf. e.g. [12, 30]. A resolution decision procedure for FO^2 with equality is described in [12].

It follows from [32] that the computational complexity of the satisfiability problem in $\mathcal{ALB}$ is NExpTime-hard. In [23] it is shown that the satisfiability problem of FO^2 is NExpTime-complete. It therefore follows that $\mathcal{ALB}$ is NExpTime-complete. It is also shown in [23] that FO^2 reduces to the *dyadic Scott class*. A formula belongs to the *Scott class* iff it is a conjunction of formulae in prefix normal form and the quantifier prefixes are $\forall\forall$ or $\forall\exists$. The *dyadic Scott class* is the fragment of the Scott class restricted to predicate symbols with arity less than or equal to two. Again, it is straightforward to see from Table 5 that $\mathcal{ALB}$-concepts, and in fact $\mathcal{ALB}$-knowledge bases, can be effectively translated into the dyadic Scott class. Thus, satisfiability of both $\mathcal{ALB}$-knowledge bases and FO^2 formulae can be reduced to the same class. Both $\mathcal{ALB}$ and FO^2 are NExpTime-complete and so is the restriction of FO^2 to boundedly many relation symbols (the latter is shown in [23]). Lutz and Sattler [34] have also shown that the restriction of $\mathcal{ALB}$ to a bounded number of role symbols is in ExpTime. It also follows from their results that FO^2 extended by equality is expressively equivalent to $\mathcal{ALB}$ extended by the identity role $\mathrm{id}(\top)$.

3.3 Maslov's class $\overline{\mathbf{K}}$

A decidable extension of FO^2 is the dual of Maslov's class K [37], denoted by $\overline{\mathrm{K}}$. In particular, $\overline{\mathrm{K}}$ extends the class of normal forms of FO^2 formulae introduced by Mortimer [40]. The language over which formulae in the class $\overline{\mathrm{K}}$ are constructed is the language of first-order logic without equality and without function symbols. Let φ be a closed formula in negation normal form and ψ be a subformula of φ. The *φ-prefix* of the formula ψ is the sequence of quantifiers of φ which bind the free variables of ψ. If a φ-prefix is of the form $\exists y_1 \dots \exists y_m \forall x_1 Q_1 z_1 \dots Q_n z_n$, where $m \geq 0$, $n \geq 0$, $Q_i \in \{\exists, \forall\}$ for all i, $1 \leq i \leq n$, then $\forall x_1 Q_1 z_1 \dots Q_n z_n$ is the *terminal φ-prefix*. For a φ-prefix $\exists y_1 \dots \exists y_m$ the terminal φ-prefix is the empty sequence of quantifiers. By definition, a closed formula φ in negation normal form belongs to the class $\overline{\mathrm{K}}$ iff there are k quantifiers $\forall x_1, \dots, \forall x_k$, $k \geq 0$, in φ such that for every atomic subformula ψ of φ the terminal φ-prefix of ψ is either

1. of length less than or equal to 1, or
2. ends with an existential quantifier, or
3. is of the form $\forall x_1 \forall x_2 \dots \forall x_k$.

Consider the following formula φ_1:

$$\begin{aligned}\forall x \forall y (mwc(x, y) \rightarrow (&married(x, y) \wedge \\ &\exists z (has\_child(x, z) \wedge has\_child(y, z)))).\end{aligned}$$

It defines the concept mwc as a subset of married couples with a child. The formula is not in the guarded fragment, since the existentially quantified subformula has no guard, and is also not in FO^2, since it uses three first-order variables. But the formula is in $\overline{\mathrm{K}}$. In order to convince ourselves of this we check whether the conditions of the definition of $\overline{\mathrm{K}}$ are satisfied. First, the formula is closed. Secondly, the formula is transformed into negation normal form by expressing implication by means of disjunction and negation. It is then equivalent to $\forall x \forall y (\neg mwc(x, y) \vee (married(x, y) \wedge \exists z (has\_child(x, z) \wedge has\_child(y, z))))$. We pick the two universal quantifiers $\forall x \forall y$ and check that every atomic subformula of φ_1 satisfies one of the three conditions set out in the definition of $\overline{\mathrm{K}}$. The φ_1-prefix of the atomic subformula $mwc(x, y)$ is $\forall x \forall y$, consequently $mwc(x, y)$ satisfies condition 3. The same applies to the atomic subformula $married(x, y)$. The φ_1-prefix of the atomic subformula $has\_child(x, z)$ is $\forall x \exists z$, while the φ_1-prefix of $has\_child(y, z)$ has the form $\forall y \exists z$. So, the terminal φ_1-prefixes of these subformulae end with an existential quantifier and therefore satisfy condition 2. Thus, the formula φ_1 belongs to $\overline{\mathrm{K}}$.

In contrast, the formula φ_2

$$\forall x \forall y (mwd(x, y) \rightarrow \forall z (have\_child(y, x, z) \rightarrow doctor(z)))$$

which describes the concept mwd as a subset of married couples all of whose children are doctors, is guarded, but not in $\overline{\mathrm{K}}$. The atomic subformula $have\_child(y, x, z)$ has the φ_2-prefix $\forall x \forall y \forall z$ while $mwd(x, y)$ has the φ_2-prefix $\forall x \forall y$, that is, there are two atomic subformulae which have a φ_2-prefix that is of length greater than 1, neither of the φ_2-prefixes ends in an existential quantifier, but they are not identical.

This shows that the guarded fragment is not a fragment of $\overline{\mathrm{K}}$ nor is $\overline{\mathrm{K}}$ a fragment of the guarded fragment. The logic $\overline{\mathrm{K}}$ is however very expressive since it contains a variety of classical solvable fragments. These include the monadic class MON, the initially extended Skolem class $[\exists^*\forall\exists^*, \forall^*]$, the Gödel class $\exists^*\forall^2\exists^*$, and FO^2. It also subsumes a range of non-classical logics, such as many extended modal logics, many description logics, and reducts of representable relation algebras [30]. The most expressive description logic subsumed by $\overline{\mathrm{K}}$ is the extension $\mathcal{ALC}(\neg, \sqcap, ^\smile, \upharpoonright, \circ^{\mathrm{pos}})$ of $\mathcal{ALB}$ with positive occurrences of role composition.

Maslov [37] showed that $\overline{\mathrm{K}}$ can be decided by the inverse method. Resolution decision procedures for $\overline{\mathrm{K}}$ as well as for the class $\overline{\mathrm{DK}}$ consisting of conjunctions of formulae in $\overline{\mathrm{K}}$ are presented in [30, 14].

3.4 Fluted logic

Suppose we wanted to define a concept *mwmc* of married couples all of whose children are married. This can be done by the formula φ_3 given by

$$\begin{aligned}\forall x_1 \forall x_2 (\mathit{mwmc}(x_1, x_2) \leftrightarrow (&\mathit{married}(x_1, x_2) \wedge \\ &\forall x_3 (\mathit{have\_child}(x_1, x_2, x_3) \rightarrow \\ &\quad \exists x_4 \mathit{married}(x_3, x_4)))).\end{aligned}$$

The formula is not guarded but also not in $\overline{\mathrm{K}}$. It is not guarded since the principal operator of the matrix of the universally quantified formula φ_3 is an equivalence, and not an implication. Even if we consider splitting the equivalence into two implications, the right-to-left implication would not have a guard. The formula φ_3 is not in $\overline{\mathrm{K}}$, since in its negation normal form there are occurrences of the atomic subformulae $\mathit{married}(x_1, x_2)$ and $\mathit{have\_child}(x_1, x_2, x_3)$ with φ_3-prefixes $\forall x_1 \forall x_2$ and $\forall x_1 \forall x_2 \forall x_3$, respectively, which are not of length 1, do not end in an existential quantifier, but are not identical.

The formula φ_3 belongs to yet another solvable fragment of first-order logic, namely *fluted logic*. Fluted logic arose as a by-product of predicate functor logic, which was introduced by Quine [51] (and adapted in [52, 53]) for the purpose of giving a variable-free treatment of first-order logic with equality. The decidability problem in fluted logic and extensions of fluted logic with binary converse and equality was studied by Purdy (cf. [48] and papers cited therein). He also showed that the computational complexity of satisfiability in fluted logic is NExpTime-complete [49]. A resolution decision procedure for fluted logic is described in [60, 61]. This decision procedure is different from resolution decision procedures for other first-order fragments in that it requires a form of splitting and dynamic renaming.

Fluted logic is defined over a finite set of predicate symbols $\mathcal{P}$ and an ordered set of variables $X_m = \{x_1, \ldots, x_m\}$. An *atomic fluted formula* of $\mathcal{P}$ over X_i is an n-ary atom $P(x_l, \ldots, x_i)$, with $l = i - n + 1$, and $n \leq i$. The class of all *fluted formulae* is defined inductively as follows.

1. Every atomic fluted formula over X_i is a fluted formula over X_i.
2. $\exists x_{i+1}\varphi$ and $\forall x_{i+1}\varphi$ are fluted formulae over X_i if φ is a fluted formula over X_{i+1}.
3. Every Boolean combination of fluted formulae over X_i is a fluted formula over X_i. That is $\varphi \rightarrow \psi$, $\neg\varphi$, $\varphi \wedge \psi$, etc., are fluted formulae over X_i, if both φ and ψ are.

The formula φ_1 is an example of a formula that is in $\overline{\mathrm{K}}$, but not in fluted logic. Consider the subformula $has\_child(x, z) \wedge has\_child(y, z)$ of φ_1. To satisfy condition 3 above, the atoms $has\_child(x, z)$ and $has\_child(y, z)$ have to be atomic fluted formulae over the same ordered subset X_i of X_m. In the case of $has\_child(x, z)$ this implies that x has to come directly before z in the ordering on variables. However, in the case of $has\_child(y, z)$ this implies that it is y that has to come directly before z in the ordering on variables. Both constraints on the ordering on variables cannot be satisfied at the same time.

In addition φ_2 is a formula that is guarded but not in fluted logic. Consider the subformula

$$mwd(x, y) \rightarrow \forall z(have\_child(y, x, z) \rightarrow doctor(z))$$

of φ_2. First of all, to satisfy condition 3 above, both

$$mwd(x, y) \text{ and } \forall z(have\_child(y, x, z) \rightarrow doctor(z))$$

have to be fluted formulae over the same ordered subset X_i of X_m, and to satisfy condition 2, the implication $have\_child(y, x, z) \rightarrow doctor(z)$, and consequently $have\_child(y, x, z)$ have to be fluted formulae over X_{i+1}. In the case of $mwd(x, y)$ this implies that x has to come directly before y in the ordering on variables, while in the case of $have\_child(y, x, z)$ we see that it is y that has to come directly before x. Again, both constraints on the ordering on variables cannot be satisfied simultaneously.

Inspecting Table 5 we see that the description logic $\mathcal{ALC}$ can be embedded into fluted logic, as can its extension by inclusion and equality value maps. Also most of the relational operators including role complement, role union, role intersection, and range restriction can be translated into fluted logic, whereas, role inversion and domain restriction cannot. Concerning role inversion, for the atom $Q_{R^{\smile}}(x, y)$ to be an atomic fluted formula, x has to come before y in the ordering on variables. However, for $Q_R(y, x)$ to be an atomic fluted formula, just the opposite ordering on x and y is required. Likewise, for domain restriction, for $Q_R(x,y)$ to be an atomic fluted formula, x has to come before y in the ordering on variables. Since $Q_R(x,y)$ occurs in a conjunction with $Q_C(x)$, $Q_C(x)$ has to be an atomic fluted formula over $\{x,y\}$, which can only be the case if y comes before x in the ordering on variables.

It can be shown that translations of description logic and modal logic formulae by both the standard relational translation and a variation of the functional translation are fluted formulae [60]. In fact, there are two natural fragments of fluted logic which are relevant to description and modal logics.

One fragment is the *dyadic fragment of fluted logic*, that is, the set of fluted formulae over unary and binary predicate symbols. It is an easy exercise to prove the following, where π denotes the standard translation mapping of $\mathcal{ALC}(\neg,\sqcap)$ formulae into first-order logic.

1. For any formula φ in $\mathcal{ALC}(\neg,\sqcap)$, the formula $Qx\,\pi(\varphi,x)$ is a dyadic fluted formula, where $Q \in \{\forall,\exists\}$.
2. For any closed dyadic fluted formula ψ there is a concept C of $\mathcal{ALC}(\neg,\sqcap)$ such that ψ is logically equivalent to $Qx\,\pi(C,x)$, where $Q \in \{\forall,\exists\}$.

From a modal logic perspective, this result states that the dyadic fragment of fluted logic is the *relational modal fragment* of first-order logic associated with Boolean modal logic.

Another fragment of fluted logic arising from description and modal logics is *ordered first-order logic*, which is called the *functional modal fragment* of fluted logic in [60]. This fragment restricts atomic fluted formulae over X_i to i-ary atoms of the form $P(x_1,x_2,\ldots,x_i)$. The functional modal fragment was first defined by Herzig [26] as a target logic of a variation of the functional translation mapping which reduces local satisfiability in the modal logics K and KD to first-order satisfiability. Thus, many of the properties of K and KD carry over to the functional modal fragment, among others also the permutability of universal and existential quantification [43] which reduces the functional modal fragment into the Bernays-Schönfinkel class (the $\exists^*\forall^*$ prefix class) [29, §4].

To make this precise we present here the *functional translation* and *optimised functional translation* of $\mathcal{ALC}$-concept consistency tests into first-order logic. With every concept symbol A, every role symbol P, and every sequence σ of n role symbols (not necessarily distinct), we associate the n-ary predicate symbols $Q_{A,\sigma}$ and $Q_{P,\sigma}$, respectively. The notation $\overline{x}$ is used to denote a sequence of variables $x_1,\ldots,x_n$, and we denote by 'λ' and '.' the empty sequence and the concatenation operation on sequences, respectively. Then the functional translation mapping π_f is specified as follows.

$$\pi_f(A,\overline{x},\sigma) = Q_{A,\sigma}(x_1,\ldots,x_n) \qquad \pi_f(\top,\overline{x},\sigma) = \top$$

$$\pi_f(\neg C, \overline{x}, \sigma) = \neg \pi_f(C, \overline{x}, \sigma) \qquad \pi_f(\bot, \overline{x}, \sigma) = \bot$$
$$\pi_f(C \sqcap D, \overline{x}, \sigma) = \pi_f(C, \overline{x}, \sigma) \wedge \pi_f(D, \overline{x}, \sigma)$$
$$\pi_f(C \sqcup D, \overline{x}, \sigma) = \pi_f(C, \overline{x}, \sigma) \vee \pi_f(D, \overline{x}, \sigma)$$
$$\pi_f(\forall P \,.\, C, \overline{x}, \sigma) = \forall x_{n+1}\, (Q_{P,\sigma}(\overline{x}) \rightarrow \pi_f(C, \overline{x}.x_{n+1}, \sigma.P))$$
$$\pi_f(\exists P \,.\, C, \overline{x}, \sigma) = \exists x_{n+1}\, (Q_{P,\sigma}(\overline{x}) \wedge \pi_f(C, \overline{x}.x_{n+1}, \sigma.P))$$

The *functional translation* of a concept C is defined by $\pi_f(C, \lambda)$. It follows from a corresponding result for basic modal logic $K_{(m)}$ (cf. [42, 43, 58]) that an $\mathcal{ALC}$-concept C is coherent iff the formula $\pi_f(C, \lambda)$ is satisfiable.

Let Υ be an operator on first-order formulae which converts a first-order formula obtained from the functional translation of an $\mathcal{ALC}$-concept into prenex normal form and moves all existential quantifiers of the functional variables inward as far as possible according to the rule '$\exists x \forall y\, \psi$ becomes $\forall y \exists x\, \psi$'. Then $\pi_f(C, \lambda)$ is satisfiable iff $\pi_{of}(C, \lambda) = \neg \Upsilon \pi_f(\neg C, \lambda)$, the *optimised functional translation of* C is satisfiable. This is again a direct consequence of a corresponding result for modal logic, cf. [43, 58].

Experience shows that swapping quantifiers in the translated problem specifications leads to superior performance of first-order theorem provers [29]. However, swapping of universal and existential quantification is not generally applicable. For instance, it does not extend to full fluted logic; actually it does not even extend to the relational modal fragment associated with the modal logic $K_{(m)}$. In [60] a formula in K is identified where the use of the quantifier permutation operator on the relational translation of this formula leads to loss of soundness.

Fluted logic can be extended with converse on (binary) relations while still preserving decidability [48]. Fluted logic with converse allows for the satisfiability equivalent embedding of standard modal logics K, KT, KD, KB, KTB, more expressive logics, like $\mathcal{ALB}$ and the corresponding modal logics, and also FO^2.

3.5 Interrelationships

As shown in Sections 3.1 to 3.4, we can embed $\mathcal{ALC}$ into the guarded fragment, the two-variable fragment, $\overline{\mathrm{K}}$ and fluted logic. This means that the three classes have a non-empty intersection, but the example formulae φ_1, φ_2, and φ_3 show that neither of the three solvable classes is a fragment of one of the others. The relationships between the various sublogics of $\mathcal{DL}$ and fragments of first-order logic is summarised in Tables 6 and 7. Here, a dot in a particular column and row indicates that the description logic given as the first in that row can be translated into the fragment of first-order logic given at the top of that column.

	GF	$\exists^*\forall^*$	FL	FL($\breve{\ }$)	$\overline{DK}$
$[\exists^*\forall\exists^*, \forall^*]$					•
$\exists^*\forall^2\exists^*$					•
FO^2				•	•
MON	•	•	•	•	•
$\mathcal{ALC}$ (π translation) 1	•		•	•	•
$\mathcal{ALC}(\neg, \sqcap)$ (π translation) 2			•	•	•
$\mathcal{ALC}$ (π_f translation) 3			•	•	
$\mathcal{ALC}$ (π_{of} translation) 4		•			

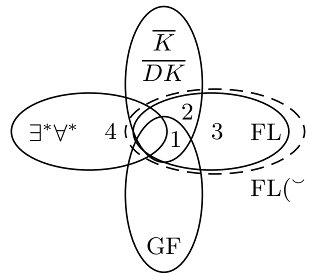

Table 6. Connections among $\mathcal{ALC}$ and decidable fragments of first-order logic

	GF	GC	FO^2	$\overline{\text{DK}}$	FL	FL($\breve{\ }$)
$\mathcal{ALC}$	•	•	•	•	•	•
$\mathcal{ALC}(\sqcap, \sqcup)$	•	•	•	•	•	•
$\mathcal{ALC}(\sqcap, \sqcup, \breve{\ }, \upharpoonleft, \circ^{\mathrm{pos}})$		•		•		
$\mathcal{ALC}(\neg)$			•	•	•	•
$\mathcal{ALC}(\neg, \sqcap)$			•	•	•	•
$\mathcal{ALC}(\neg, \sqcap, \breve{\ }, \upharpoonleft)$			•	•		•
$\mathcal{ALC}(\neg, \sqcap, \breve{\ }, \upharpoonleft, \circ^{\mathrm{pos}})$				•		

Table 7. Decidable description logics and first-order fragments without identity

3.6 Number restrictions, identity roles and equality

So far we have excluded those operators of $\mathcal{DL}$ whose translation into first-order logic requires the presence of equality, namely, number restrictions and identity roles; see Table 8. As pointed out before, in the presence of number restrictions and identity roles, the translation of a knowledge base needs to incorporate inequations representing the unique name assumption. For every knowledge base Γ consisting of a set of terminological sentences, its translation $\Pi(\Gamma)$ is modified

$$\pi(\exists_{\geq n} R, X) = \exists y_1, \ldots, y_n\, (\pi(R, X, y_1) \wedge \ldots \wedge \pi(R, X, y_n)) \wedge \\ y_1 \not\approx y_2 \wedge \ldots \wedge y_{n-1} \not\approx y_n)$$
$$\pi(\exists_{\leq n} R, X) = \forall y_1, \ldots, y_{n+1}\, (\pi(R, X, y_1) \wedge \ldots \wedge \pi(R, X, y_{n+1}) \rightarrow \\ y_1 \approx y_2 \vee \ldots \vee y_n \approx y_{n+1})$$
$$\pi(\exists_{\geq n} R.C, X) = \pi(\exists_{\geq n}(R{\restriction}C), X)$$
$$\pi(\exists_{\leq n} R.C, X) = \pi(\exists_{\leq n}(R{\restriction}C), X)$$
$$\pi(\mathrm{id}(C), X, Y) = \pi(C, X) \wedge X \approx Y$$

Table 8. Extension of π for number restrictions and the identity role operator.

to the following.

$$\Pi(\Gamma) = \{\Pi(\alpha) \mid \alpha \in \Gamma\} \cup \{a \not\approx b \mid \text{a,b are distinct objects symbols in } \mathsf{O}\}$$

Since the classes we have considered so far did not include equality, (qualified) number restrictions and identity roles cannot be embedded into them. The same is true for *graded modalities* [20] which correspond to number restrictions in the context of modal logics. However, the guarded fragment, the two-variable fragment and fluted logic remain decidable when extended by equality. For the dual of Maslov's class, only trivial extensions with equality remain decidable.

The translation of identity roles then falls into the guarded fragment with equality, the two-variable fragment with equality, and fluted logic with equality, see Table 9. However, the translation of (qualified) number restrictions still does not belong to any of these three fragments of first-order logic with equality. Obviously, for any number n greater than one, the translation of $\exists_{\geq n}R$ and $\exists_{\leq n}R$ requires at least three variables; the translation does therefore not embed these concepts into the two-variable fragment with equality. Similarly, for any number n greater than one, the translations of $\exists_{\geq n}R$ and $\exists_{\leq n}R$ lack the guard atoms necessary for guarded formulae. The conjunction $\pi(R, X, y_1) \wedge \ldots \wedge \pi(R, X, y_n)$ forms a 'guard', but is obviously not atomic for $n > 1$. The conjunction also violates the more relaxed conditions on guards for the loosely guarded and packed fragments of first-order logic with equality (see [68, 36] for definitions). Finally, $\pi(R, X, y_1) \wedge \ldots \wedge \pi(R, X, y_n)$ is also not a conjunction of atomic fluted formulae over a common ordered set of variables X_m.

Grädel, Otto, and Rosen [24] have shown that the extension of the two-variable fragment with *counting quantifiers*, that is, first-order quantifiers $\exists^{\geq n}$ for any $n \geq 1$, is decidable, while Grädel [22] has shown that the extension of the guarded fragment with counting quantifiers is undecidable.

Recently, Hustadt, Motik, and Sattler [41] have shown that the satisfiability of knowledge bases over the extension of $\mathcal{ALC}$ with role inverse, qualified number restrictions, and transitive roles can be decided by an instance of the basic superposition calculus, a sophisticated clausal calculus for logics with equality [6]. The result is obtained by showing that knowledge bases in this fragment of $\mathcal{DL}$ can be embedded into a subclass of first-order clausal logic with equality, the class of so-called $\mathcal{ALCHIQ}^{-}$-closures, and that inferences by the superposition calculus only result in redundant clauses or $\mathcal{ALCHIQ}^{-}$-closures.

	GF(≈)	GC(≈)	FO^2(≈)	FL($\breve{}$, ≈)
$\mathcal{ALC}(\mathrm{id})$	•	•	•	•
$\mathcal{ALC}(\sqcap, \sqcup, \breve{}, \restriction, \mathrm{id})$	•	•	•	•
$\mathcal{ALC}(\sqcap, \sqcup, \breve{}, \restriction, \circ^{\mathrm{pos}}, \mathrm{id})$		•		
$\mathcal{ALC}(\neg, \mathrm{id})$			•	•
$\mathcal{ALC}(\neg, \sqcap, \mathrm{id})$			•	•
$\mathcal{ALC}(\neg, \sqcap, \breve{}, \restriction, \mathrm{id})$			•	•

Table 9. Decidable description logics and first-order fragments with identity

4 Beyond $\mathcal{ALC}$ and decidability

If instead of extensions of $\mathcal{ALC}$ we consider products of $\mathcal{ALC}$ with modal logics, for example, basic modal logic, then there are examples which fall in neither of the solvable fragments of first-order logic looked at so far in this paper. The motivation for studying products of description logics and modal logics is that while the standard description logics are designed for reasoning in a static environment, logics which are products of modal and description logics are able to describe, for example, intensional knowledge in multi-agent systems or dynamic environments which change over time or by the execution of actions. Examples of such products of modal and description logics include $\mathcal{FALCM}$ [28], $\mathcal{ALC}_{\mathcal{M}}$ [70] and $K_{\mathcal{ALC}}$ [35].

In the following we focus on a slight variation of $K_{\mathcal{ALC}}$ with subconcept definitions instead of concept equivalence. Given mutually disjoint sets of concept symbols, role symbols, object symbols, and agent symbols, the set of $K_{\mathcal{ALC}}$-concepts is inductively defined as follows. All concept symbols as well as $\top$ and $\bot$ are concepts. If C and D are concepts, P is a role symbol, and i is an agent name, then the following expressions are concepts: $\neg C$, $C \sqcap D$, $C \sqcup D$, $\forall P \,.\, C$, $\exists P \,.\, C$, $\Box_i C$, and $\Diamond_i C$. Next we define $K_{\mathcal{ALC}}$-sentences. If C and D are concepts and a is an object name, then $C \mathrel{\dot\sqsubseteq} D$ and $C \doteq D$ are terminological sentences, and $a{:}C$ is an assertional sentence. If φ and ψ are (assertional or terminological) sentences and i is an agent symbol, then the following expressions are sentences: $\neg\varphi$, $\varphi \wedge \psi$, $\varphi \vee \psi$, $\Diamond_i\varphi$, and $\Box_i\varphi$. Thus, in $K_{\mathcal{ALC}}$, modal operators can be applied to both concepts and sentences, but not to roles.

The semantics of $K_{\mathcal{ALC}}$ is a mixture of the possible world semantics of the modal logic $K_{(m)}$ and the set-theoretic semantics for $\mathcal{ALC}$. $K_{\mathcal{ALC}}$-models are restricted by the constant domain assumption and assume rigid designation of constant symbols. The semantics of our variation of $K_{\mathcal{ALC}}$ is determined by an embedding ω of $K_{\mathcal{ALC}}$ into first-order logic, as defined in Table 10. The symbols X and W are meta-variables for first-order terms, and as before Q_A and Q_P denote unary and binary predicate symbols, uniquely associated with a concept symbol A and a role symbol P, respectively. R_i is a binary predicate symbol

$\omega(A, W, X) = Q_A(W, X)$	$\omega(C \sqcap D, W, X) = \omega(C, W, X) \wedge \omega(D, W, X)$
$\omega(\neg C, W, X) = \neg\omega(C, W, X)$	$\omega(C \sqcup D, W, X) = \omega(C, W, X) \vee \omega(D, W, X)$
$\omega(\top, W, X) = \top$	$\omega(\forall P \,.\, C, W, X) = \forall y\,(Q_P(W, X, y) \rightarrow \omega(C, W, y))$
$\omega(\bot, W, X) = \bot$	$\omega(\exists P \,.\, C, W, X) = \exists y\,(Q_P(W, X, y) \wedge \omega(C, W, y))$
$\omega(\Box_i C, W, X) = \forall v\,(R_i(W, v) \rightarrow \omega(C, v, X))$	
$\omega(\Diamond_i C, W, X) = \exists v\,(R_i(W, v) \wedge \omega(C, v, X))$	
$\omega(a{:}C, W) = \omega(C, W, a))$	$\omega(C \mathrel{\dot\sqsubseteq} D, W) = \forall x\,(\omega(C, W, x) \rightarrow \omega(D, W, x))$
$\omega(\neg\varphi, W) = \neg\omega(\varphi, W)$	$\omega(C \doteq D, W) = \forall x\,(\omega(C, W, x) \leftrightarrow \omega(D, W, x))$
$\omega(\varphi \wedge \psi, W) = \omega(\varphi, W) \wedge \omega(\psi, W)$	$\omega(\Box_i\varphi, W) = \forall v\,(R_i(W, v) \rightarrow \omega(\varphi, v))$
$\omega(\varphi \vee \psi, W) = \omega(\varphi, W) \vee \omega(\psi, W)$	$\omega(\Diamond_i\varphi, W) = \exists v\,(R_i(W, v) \wedge \omega(\varphi, v))$

Table 10. Translation of $K_{\mathcal{ALC}}$ into first-order logic

representing the accessibility relation associated with the modal operator $\Box_i$. And a denotes the Skolem constant associated with the object name a. Now, let $\Omega(\varphi) = \exists u\,\omega(\varphi, u)$, for any $K_{\mathcal{ALC}}$-sentence φ.

Consider the sentence: *Tim believes, minis are what Don believes to be slow cars*. If we use the modal operators $\Box_{Tim}$ and $\Box_{Don}$ to represent '*agent Tim believes*' and '*agent Don believes*', respectively, then the sentence can be represented in $K_{\mathcal{ALC}}$ by

$$\Box_{Tim}(minis \mathrel{\dot\sqsubseteq} \Box_{Don} slow\_cars).$$

The standard translation to first-order logic by the mapping Ω is the following formula φ_4.

$$\exists u \forall w (R_{Tim}(u, v) \rightarrow \forall x\,(minis(v, x) \rightarrow \\ \forall v\,(R_{Don}(v, w) \rightarrow slow\_car(w, x))))$$

This formula is neither guarded nor fluted and is also not in $\overline{\mathrm{K}}$. The atom $R_{Don}(w, v)$ does not cover the variable x of $slow\_car(v, x)$ and is therefore not a guard for $slow\_car(v, x)$, thus, φ_4 does not belong to GF. It does not belong to fluted logic because the ordering of the variables in the atom $slow\_car(v, x)$ does not parallel their order of quantification and the sequence of variables in $R_{Don}(w, v)$. The atomic formula $R_{Don}(w, v)$ omits the x which should appear between w and v in a fluted formula. φ_4 is also not in $\overline{\mathrm{K}}$, because no sequence of universal quantifiers can be identified such that the quantifier prefixes of the atomic subformulae satisfy the conditions in the definition of $\overline{\mathrm{K}}$.

However, φ_4 belongs to the so-called *positive restrictive quantification fragment PRQ* introduced by Bry and Torge [9]. The definition of the fragment is given in terms of two notions called *positive conditions* and *ranges*. Positive conditions are inductively defined as follows. Atoms except $\bot$ are positive conditions;

conjunction and disjunction of positive conditions are positive conditions; $\exists y \varphi$ is a positive condition if φ is a positive condition. The ranges for a set of variables $X_n = \{x_1, \ldots, x_n\}$ are inductively defined as follows.

1. An atom in which all the variables in X_n occur is a range for X_n.
2. $\rho_1 \vee \rho_2$ is a range for X_n iff both ρ_1 and ρ_2 are ranges for X_n.
3. $\rho \wedge \phi$ is a range for X_n iff ρ is a range for X_n and ϕ is a positive condition.
4. $\exists y \rho$ is a range for X_n iff ρ is a range for $\{y\} \cup X_n$ and if $x_i \neq y$ for all $x_i \in X_n$.

Note that in contrast to fluted logic, no ordering on the variables in X_n is assumed. Positive formulae with restricted quantification, *PRQ-formulae* for short, are then inductively defined as follows.

1. $\top$ and $\bot$ are PRQ-formulae.
2. If φ is an atomic formula, then φ is a PRQ-formula.
3. PRQ is closed under conjunctions and disjunctions.
4. A formula of the form $\varphi \rightarrow \psi$ is a PRQ-formula iff φ is a positive condition and ψ is a PRQ-formula.
5. A formula of the form $\forall x_1 \ldots \forall x_n (\rho \rightarrow \psi)$, $n \geq 1$, is a PRQ-formula if ρ is a range for $x_1, \ldots, x_n$, and ψ is a PRQ-formula.
6. A formula of the form $\exists x(\rho \wedge \psi)$ is a PRQ-formula if ρ is a range for x and if ψ is a PRQ-formula.

The formula φ_4 is indeed a PRQ-formula. Because $R_{Don}(w, v)$ is a range for v and $slow\_car(v, x)$ is an atom and therefore a PRQ-formula, $\forall v(R_{Don}(w, v) \rightarrow slow\_car(v, x))$ is a PRQ-formula. Since $minis(w, x)$ is a range for x, it also follows that the subformula $\forall x(minis(w, x) \rightarrow \ldots)$ is a PRQ-formula. Similarly, $R_{Tim}(\epsilon, w)$ is a range for w, and hence, φ_4 is a PRQ-formula.

Unfortunately, PRQ is not solvable, because PRQ is in fact expressively equivalent to first-order logic. For all fragments of PRQ that have the finite model property, there is however a decision procedure in the form of an extended positive tableau method [67]. The method does not only detect unsatisfiability but also generates finite models if they exist. Since $\mathcal{ALC}$ and many of its extensions including $K_{\mathcal{ALC}}$ [69] have the finite model property, this procedure provides a general, sound, complete, and terminating method for solving the satisfiability problem for these logics without the necessity of additional soundness, completeness or termination proofs.

5 Conclusion

In this short survey we considered the relationship of fragments of first-order logic and description logics. This provides a new perspective of description logics and allows us to transfer results and techniques of first-order fragments to description logics. The relationship gives us insights into the different kinds of reasoning methodologies which are applicable to description logics. The paper further gives some examples of the different kinds of questions that can be solved with automated reasoning systems (we could have given more examples but page restrictions prevent us from doing so). All the classes considered—the guarded fragment with equality (including the loosely guarded fragment with equality), the two-variable fragment with equality, Maslov's class $\overline{\mathrm{DK}}$, and fluted logic—have resolution decision procedures [16, 12, 30, 61]. Due to the availability of several sophisticated first-order theorem provers based on the resolution calculus, practical inference systems are therefore immediately at hand for these fragments and all embedded description and modal logics.

In this paper the pairwise orthogonality of the logics is shown only at the syntactic level. To the best of our knowledge there have not been any investigations of semantical equivalence thus far. Such investigations are important and would of course give us a more complete understanding of the landscape of decidable first-order fragments and description logics.

References

1. H. Andréka, I. Németi, and I. Sain. Some new landmarks on the roadmap of two dimensional logics. In J. van Eijck and A. Visser, editors, *Logic and Information Flow*, pp. 163–169. MIT Press, 1994.
2. H. Andréka, I. Németi, and J. van Benthem. Modal languages and bounded fragments of predicate logic. *J. Philos. Logic*, 27(3):217–274, 1998.
3. F. Baader, D. Calvanese, D. Nardi D. McGuinness, and P. Patel-Schneider, editors. *The Description Logic Handbook: Theory,Implementation and Applications*. Cambridge University Press, 2003.
4. F. Baader, R. Küsters, and F. Wolter. Extensions to description logics. In Baader et al. [3], chapter 6, pp. 219–261.
5. F. Baader and W. Nutt. Basic description logics. In Baader et al. [3], chapter 2, pp. 43–95.
6. L. Bachmair, H. Ganzinger, C. Lynch, and W. Snyder. Basic paramodulation. *Inform. and Computat.*, 121(2):172–192, 1995.
7. C. Brink, K. Britz, and R. A. Schmidt. Peirce algebras. *Formal Aspects of Computing*, 6(3):339–358, 1994.
8. C. Brink and R. A. Schmidt. Subsumption computed algebraically. *Computers and Mathematics with Applications*, 23(2–5):329–342, 1992.
9. F. Bry and S. Torge. A deduction method complete for refutation and finite satisfiability. In *Proc. JELIA'98*, vol. 1498 of *LNAI*, pp. 1–17. Springer, 1998.
10. D. Calvanese and G. De Giacomo. Expressive description logics. In Baader et al. [3], chapter 5, pp. 178–218.
11. H. de Nivelle and M. de Rijke. Deciding the guarded fragments by resolution. *J. Symbolic Computat.*, 35(1):21–58, 2003.
12. H. de Nivelle and I. Pratt-Hartmann. A resolution-based decision procedure for the two-variable fragment with equality. In *Proc. IJCAR'01*, vol. 2083 of *LNAI*, pp. 211–225. Springer, 2001.
13. H. de Nivelle, R. A. Schmidt, and U. Hustadt. Resolution-based methods for modal logics. *Logic J. IGPL*, 8(3):265–292, 2000.
14. C. Fermüller, A. Leitsch, U. Hustadt, and T. Tammet. Resolution decision procedures. In Robinson and Voronkov [54], chapter 25, pp. 1791–1849.

15. D. M. Gabbay, A. Kurucz, F. Wolter, and M. Zakharyaschev. *Many-Dimensional Modal Logics: Theory and Applications*, vol. 148 of *Studies in Logic and The Foundation of Mathematics*. Elsevier, 2003.
16. H. Ganzinger and H. de Nivelle. A superposition decision procedure for the guarded fragment with equality. In *Proc. LICS'99*, pp. 295–303. IEEE Computer Society Press, 1999.
17. H. Ganzinger, C. Meyer, and M. Veanes. The two-variable guarded fragment with transitive relations. In *Proc. LICS'99*, pp. 24–34. IEEE Computer Society Press, 1999.
18. G. Gargov and S. Passy. A note on Boolean modal logic. In *Mathematical logic: Proc. of the 1988 Heyting Summerschool*, pp. 299–309. Plenum Press, 1990.
19. L. Georgieva, U. Hustadt, and R. A. Schmidt. Hyperresolution for guarded formulae. *J. Symbolic Computat.*, 36(1–2):163–192, 2003.
20. L. F. Goble. Grades of modality. *Logique et Analyse*, 13:323–334, 1970.
21. E. Grädel. Guarded fragments of first-order logic: a perspective for new description logics? In *Proc. DL'98*, pp. 5–8. Instituto Trentino di Cultura, 1998.
22. E. Grädel. On the restraining power of guards. *J. Symbolic Logic*, 64:1719–1742, 1999.
23. E. Grädel, P. Kolaitis, and M. Vardi. On the decision problem for two-variable first-order logic. *Bull. Symbolic Logic*, 3:53–69, 1997.
24. E. Grädel, M. Otto, and E. Rosen. Two-Variable Logic with Counting is Decidable. In *Proc. LICS'97*, 1997.
25. D. Harel, D. Kozen, and J. Tiuryn. *Dynamic Logic*. MIT Press, 2000.
26. A. Herzig. A new decidable fragment of first order logic. In Abstracts of the 3rd Logical Biennial, Summer School & Conference in honour of S. C. Kleene, Varna, Bulgaria, 1990.
27. C. Hirsch and S. Tobies. A tableau algorithm for the clique guarded fragment. In *Advances in Modal Logic, Volume 3*. World Scientific, 2002.
28. U. Hustadt. Introducing epistemic operators into a description logic. In A. Laux and H. Wansing, editors, *Knowledge and Belief in Philosophy and Artificial Intelligence*, pp. 65–85. Akademie Verlag, 1995.
29. U. Hustadt and R. A. Schmidt. An empirical analysis of modal theorem provers. *J. Appl. Non-Classical Logics*, 9(4):479–522, 1999.
30. U. Hustadt and R. A. Schmidt. Maslov's class K revisited. In *Automated Deduction—CADE-16*, vol. 1632 of *LNAI*, pp. 172–186. Springer, 1999.
31. U. Hustadt and R. A. Schmidt. Issues of decidability for description logics in the framework of resolution. In *Automated Deduction in Classical and Non-Classical Logics*, vol. 1761 of *LNAI*, pp. 191–205. Springer, 2000.
32. C. Lutz and U. Sattler. The complexity of reasoning with Boolean modal logics. In *Advances in Modal Logics Volume 3*. World Scientific, 2002.
33. C. Lutz, U. Sattler, and S. Tobies. A suggestion of an n-ary description logic. In *Proc. DL'99*, pp. 81–85. Linköping Univ., 1999.
34. C. Lutz, U. Sattler, and F. Wolter. Description logics and the two-variable fragment. In *Proc. DL'01*, pp. 66–75, 2001.
35. C. Lutz, H. Sturm, F. Wolter, and M. Zakharyaschev. A tableau decision algorithm for modalized $\mathcal{ALC}$ with constant domains. *Studia Logica*, 72(2), 2002.
36. M. Marx. Tolerance logic. *J. Logic, Language and Inform.*, 10:353–373, 2001.
37. S. Ju. Maslov. The inverse method for establishing deducibility for logical calculi. In *Proc. Steklov Institute of Mathematics*, pp. 25–96. Amer. Math. Soc., 1971.
38. F. Massacci. Decision procedures for expressive description logics with intersection, composition, converse of roles and role identity. In *Proc. IJCAI 2001*, pp. 193–198. Morgan Kaufmann, 2001.
39. M. Minsky. A framework for representing knowledge. In P. J. Winston, editor, *The psychology of computer visions*, pp. 211–277. McGraw-Hill, 1975.
40. M. Mortimer. On languages with two variables. *Z. Math. Logik Grundlagen Math.*, 21:135–140, 1975.
41. B. Motik, U. Sattler, and U. Hustadt. Reducing $\mathcal{SHIQ}^-$ description logic to disjunctive datalog programs. In *Proc. KR 2004*, pp. 152–162. AAAI Press, 2004.

42. H. J. Ohlbach, A. Nonnengart, M. de Rijke, and D. Gabbay. Encoding two-valued nonclassical logics in classical logic. In Robinson and Voronkov [54], chapter 21, pp. 1403–1486.
43. H. J. Ohlbach and R. A. Schmidt. Functional translation and second-order frame properties of modal logics. *J. Logic Computat.*, 7(5):581–603, 1997.
44. E. Orlowska. Relational formalisation of nonclassical logics. In C. Brink, W. Kahl, and G. Schmidt, editors, *Relational Methods in Computer Science*, Advances in Computing, pp. 90–105. Springer, 1997.
45. Peter F. Patel-Schneider. *Decidable, Logic-Based Knowledge Representation*. PhD thesis, Univ. Toronto, 1987.
46. D. A. Plaisted and S. Greenbaum. A structure-preserving clause form translation. *J. Symbolic Computat.*, 2:293–304, 1986.
47. V. R. Pratt. Semantical considerations on floyd-hoare logic. In *Proc. 17th Annual IEEE Symposium on Foundations of Computer Science*, pp. 109–121. IEEE Computer Society, 1976.
48. W. C. Purdy. Quine's 'limits of decision'. *J. Symbolic Logic*, 64(4):1439–1466, 1999.
49. W. C. Purdy. Complexity and nicety of fluted logic. *Studia Logica*, 71:177–198, 2002.
50. M. R. Quillian. Word concepts: A theory and simulation of some basic semantic capabilities. *Behavioral Sciences*, 12:410–430, 1967.
51. W. V. Quine. Variables explained away. In *Proc. American Philosophy Society*, vol. 104, pp. 343–347, 1960.
52. W. V. Quine. Algebraic logic and predicate functors. In R. Rudner and I. Scheffler, editors, *Logic and Art: Esssays in Honor of Nelson Goodman*. Bobbs-Merrill, Indianapolis, 1971.
53. W. V. Quine. Predicate functors revisited. *J. Symbolic Logic*, 46:649–652, 1981.
54. A. Robinson and A. Voronkov, editors. *Handbook of Automated Reasoning*. Elsevier, 2001.
55. K. Schild. Undecidability of subsumption in $\mathcal{U}$. KIT-Report 67, Department of Computer Science, Technische Universität Berlin, 1988.
56. K. Schild. A correspondence theory for terminological logics: Preliminary report. In *Proc. IJCAI'91*, pp. 466–471, 1991.
57. R. A. Schmidt. Algebraic terminological representation. Master's thesis, Univ. Cape Town, 1991. Available as Technical Report MPI-I-91-216, Max-Planck-Institut für Informatik, Saarbrücken.
58. R. A. Schmidt. *Optimised Modal Translation and Resolution*. PhD thesis, Universität des Saarlandes, Saarbrücken, Germany, 1997.
59. R. A. Schmidt. Relational grammars for knowledge representation. In M. Böttner and W. Thümmel, editors, *Variable-Free Semantics*, vol. 3 of *Artikulation und Sprache*, pp. 162–180. secolo Verlag, Osnabrück, Germany, 2000.
60. R. A. Schmidt and U. Hustadt. Deciding fluted logic with resolution. Manuscript. Submitted to *Inform. and Computat.*, 2000.
61. R. A. Schmidt and U. Hustadt. A resolution decision procedure for fluted logic. In *Proc. CADE-17*, vol. 1831 of *LNAI*, pp. 433–448. Springer, 2000.
62. R. A. Schmidt, E. Orlowska, and U. Hustadt. Two proof systems for Peirce algebras. In *Proc. RelMiCS-7*, vol. 3051 of *LNCS*, pp. 238–251. Springer, 2004.
63. M. Schmidt-Schauß. Subsumption in KL-ONE is undecidable. In *Proc. KR'89*, pp. 421–431. Morgan Kaufmann, 1989.
64. M. Schmidt-Schauß and G. Smolka. Attributive concept descriptions with complements. *J. Artificial Intelligence*, 48:1–26, 1991.
65. D. Scott. A decision method for validity of sentences in two variables. *J. Symbolic Logic*, 27:377, 1962.
66. A. Tarski. On the calculus of relations. *J. Symbolic Logic*, 6(3):73–89, 1941.
67. S. Torge. *Überprüfung der Erfüllbarkeit im Endlichen: Ein Verfahren und seine Anwendung*. PhD thesis, Herbert Utz Verlag, München, 1998.
68. J. van Benthem. Dynamic bits and pieces. Research Report LP-1997-01, Institute for Logic, Language and Computation, Univ. of Amsterdam, 1997.
69. F. Wolter. Personal communication, 2004.
70. F. Wolter and M. Zakharyaschev. Multi-dimensional description logics. In *Proc. IJCAI'99*, pp. 104–109. Morgan Kaufmann, 1999.

A Appendix: Structural transformation

For the sake of keeping the paper self-contained we recall here the definition of structural transformation. The polarity of (occurrences of) first-order subformulae is defined as follows. Any occurrence of a proper subformula of an equivalence has *zero polarity.* For occurrences of subformulae not below a '$\leftrightarrow$' symbol, an occurrence of a subformula has *positive polarity* if it is one inside the scope of an even number of (explicit or implicit) negations, and it has *negative polarity* if it is one inside the scope of an odd number of negations. For any first-order formula φ, if λ is the position of a subformula in φ, then $\varphi|_\lambda$ denotes the subformula of φ at position λ and $\varphi[\psi \mapsto \lambda]$ is the result of replacing $\varphi|_\lambda$ at position λ by ψ. The set of all the positions of subformulae of φ will be denoted by $\mathrm{Pos}(\varphi)$.

Structural transformation, also referred to as *renaming*, associates with each element λ of $\Lambda \subseteq \mathrm{Pos}(\varphi)$ a predicate symbol Q_λ and a literal $Q_\lambda(x_1, \ldots, x_n)$, where $x_1, \ldots, x_n$ are the free variables of $\varphi|_\lambda$, the symbol Q_λ does not occur in φ and two symbols Q_λ and $Q_{\lambda'}$ are equal only if $\varphi|_\lambda$ and $\varphi|_{\lambda'}$ are equivalent formulae. (In practice, one may want to use the same symbols for variant subformulae, or subformulae which are obviously equivalent, for example, $\varphi \vee \varphi$ and φ.) Let

$$\begin{aligned}\mathrm{Def}^+_\lambda(\varphi) &= \forall x_1 \ldots x_n\ (Q_\lambda(x_1, \ldots, x_n) \rightarrow \varphi|_\lambda) \quad \text{and}\\ \mathrm{Def}^-_\lambda(\varphi) &= \forall x_1 \ldots x_n\ (\varphi|_\lambda \rightarrow Q_\lambda(x_1, \ldots, x_n)).\end{aligned}$$

The *definition* of Q_λ is the formula

$$\mathrm{Def}_\lambda(\varphi) = \begin{cases} \mathrm{Def}^+_\lambda(\varphi) & \text{if } \varphi|_\lambda \text{ has positive polarity,}\\ \mathrm{Def}^-_\lambda(\varphi) & \text{if } \varphi|_\lambda \text{ has negative polarity,}\\ \mathrm{Def}^+_\lambda(\varphi) \wedge \mathrm{Def}^-_\lambda(\varphi) & \text{otherwise.}\end{cases}$$

The corresponding clauses will be called *definitional clauses*. Now, define $\mathrm{Def}_\Lambda(\varphi)$ inductively by:

$$\begin{aligned}\mathrm{Def}_\emptyset(\varphi) &= \varphi \quad \text{and}\\ \mathrm{Def}_{\Lambda \cup \{\lambda\}}(\varphi) &= \mathrm{Def}_\Lambda(\varphi[Q_\lambda(x_1, \ldots, x_n) \mapsto \lambda]) \wedge \mathrm{Def}_\lambda(\varphi),\end{aligned}$$

where λ is maximal in $\Lambda \cup \{\lambda\}$ with respect to the prefix ordering on positions. A *definitional form* of φ is $\mathrm{Def}_\Lambda(\varphi)$, where Λ is a subset of all positions of subformulae (usually, non-atomic or non-literal subformulae).

It is well-known that if φ is a first-order formula, then (i) φ is satisfiable iff $\mathrm{Def}_\Lambda(\varphi)$ is satisfiable, for any $\Lambda \subseteq \mathrm{Pos}(\varphi)$, and (ii) $\mathrm{Def}_\Lambda(\varphi)$ can be computed in polynomial time (see for example Plaisted and Greenbaum [46]).

Journal on Relational Methods in Computer Science, Vol. 1, 2004, pp. 251 - 276
Received by the editors March 25, 2004, and, in revised form, October 04, 2004.
Published on December 10, 2004.

This article may be accessed via WWW at `http://www.jormics.org`.