On the relationship between decidable fragments, non-classical logics, and description logics

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

[1] Department of Computer Science, University of Manchester
Manchester M13 9PL, UK, {georgiel, schmidt}@cs.man.ac.uk

[2] Department of Computer Science, University of Liverpool
Liverpool L69 7ZF, UK, U.Hustadt@csc.liv.ac.uk

Abstract

The guarded fragment [1] and its extensions and subfragments have often been considered as a framework for investigating the properties of description logics [8, 18]. But there are other decidable fragments 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.

1 Introduction

Decidable description and modal logics as well as decidable fragments of first-order logic have found applications in versatile areas such as knowledge representation, linguistic domains, and semantics of programming languages. Often information from different knowledge domains can be expressed with the same formalisms, but also, information from one knowledge domain can be expressed with different formalisms. This leads to various questions including, whether there are theoretical or practical reasons to prefer a particular formalism for a given knowledge domain, and how the formalisms themselves are related to each other.

In this paper we focus on the description logic $\mathcal{ALC}$ and its extensions by disjunction, conjunction, negation and inverse on roles and discuss their relationship to less well known logics including Boolean modal logic [7], the two-variable fragment, the dual of the Maslov's class K [21], Quine's fluted logic [25, 26] (see also [24, 28]), and the positive restrictive quantification fragment PRQ [2].

Unless indicated otherwise, the logics considered in this paper do not include equality.

2 $\mathcal{ALC}$ and modal logics

We briefly state the definition of the description logic $\mathcal{ALC}$ [29]. The concept language of $\mathcal{ALC}$ is defined over a signature, given by a tuple of two disjoint alphabets, the set of concept symbols A and the set of the role symbols R. Every concept symbol is a concept and every role symbol is a role. If C and D are concepts, and R is a role, then $\top$, $\bot$, $C \sqcap D$, $C \sqcup D$, $\neg C$, $\forall R.C$ (universal restriction), and $\exists R.C$ (existential restriction) are concepts. One possibility of extending $\mathcal{ALC}$ is the addition of role-forming operators like negation $\neg$, conjunction $\sqcap$, disjunction $\sqcup$, inverse $^{\smile}$, and composition $\circ$. We denote the extension of $\mathcal{ALC}$ by role-forming operators $r_1, \ldots, r_n$ by $\mathcal{ALC}(r_1, \ldots, r_n)$.

Consider an example from knowledge representation and linguistic domains. Assume that we have the concepts *Person*, *Female*, and the role *has_child*. We can express more complex concepts combining concept symbols and role symbols such as the concept

$$Person \sqcap \exists has\_child.Female \sqcap \forall has\_child.Female$$

which represents people who have only daughters.

It is well-known that the description logic $\mathcal{ALC}$ can be viewed as syntactical variant of basic modal logic. $\mathcal{ALC}$ augmented with disjunction, conjunction and negation on roles is embeddable in the Boolean modal logic. Boolean modal logic is defined over families of binary relations closed under union, intersection, and complementation [7]. It is complete for the standard Kripke semantics and is an analogy of the propositional dynamic logic in which the roles occurring in the value restrictions are complex roles, built using the constructors composition, disjunction, and reflexive transitive closure.

3 $\mathcal{ALC}$ and other decidable formalisms

The definition of the standard semantics of $\mathcal{ALC}$ [29] and its extensions by the role operators disjunction, conjunction, negation and inverse, indicates that these languages may be considered as subfragments of first-order logic. The concept and role names can be seen as unary or binary predicate symbols, and the concept terms as abbreviations for formulae with one free variable. Then, the standard embedding π of $\mathcal{ALC}$ and its extensions into first-order logic is defined as follows.

$$\begin{aligned}
\pi(A, X) &= Q_A(W) & \pi(C \sqcap D, X) &= \pi(C, X) \wedge \pi(D, X)\\
\pi(\neg C, X) &= \neg\pi(C, X) & \pi(C \sqcup D, X) &= \pi(C, X) \vee \pi(D, X)\\
\pi(\top, X) &= \top & \pi(\forall R.C, X) &= \forall y(\pi(R, X, y) \rightarrow \pi(C, y))\\
\pi(\bot, X) &= \bot & \pi(\exists R.C, X) &= \exists y(\pi(R, X, y) \wedge \pi(C, y))\\
\pi(R, X, Y) &= Q_R(X, Y) & \pi(R \sqcap S, X, Y) &= \pi(R, X, Y) \wedge \pi(S, X, Y)
\end{aligned}$$

$$\pi(\neg R, X, Y) = \neg\pi(R, X, Y) \quad \pi(R \sqcup S, X, Y) = \pi(R, X, Y) \vee \pi(S, X, Y)$$
$$\pi(R^{\smile}, X, Y) = \pi(R, Y, X) \quad \pi(R \circ S, X, Y) = \exists z(\pi(R, X, z) \wedge \pi(S, z, Y))$$

Here X and Y are meta variables for variables and constants, and Q_A and Q_R denote unary and binary predicate symbols, uniquely associated with a concept symbol A and a role symbol R, respectively. Using the embedding π all common reasoning services in description logics are reducible to satisfiability testing in first-order logic.

In the following we consider the relationship between $\mathcal{ALC}$, the two-variable fragment of first-order logic FO^2, the guarded fragment, fluted logic, and the dual of Maslov's class K. We comment on the inference methods applicable for those fragments, the complexity results, and illustrate by example formulae which belong and which do not belong to the intersection of these fragments.

It is well-known that π embeds $\mathcal{ALC}$ concepts into the *guarded fragment GF*. Formally the formulae of the guarded fragment of function-free first-order logic are inductively defined as follows. (i) $\top$ and $\bot$ are in GF. (ii) If φ is an atomic formula, then φ is in GF. (iii) GF is closed under Boolean connectives. (iv) 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 $\exists\overline{x}(A \wedge \varphi)$ 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. If there is no bound on the arity of predicate symbols and the number of free variables in guarded formulae, then the problem of deciding the satisfiability of guarded formulae is of double exponential time and space complexity [3, 5, 9]. Of theoretical interest is 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 [3, 5], alternating automata [9], tableaux methods [12], or embedding into monadic second-order logic [6].

In fact, in [4] it is shown how any $\mathcal{ALC}(\sqcup, \sqcap, {}^{\smile})$ concept translates into a guarded formula.

One important property of the guarded fragment is that the guards, that is the atoms we obtain from the translation of roles, are always positive. Therefore, $\mathcal{ALC}(\neg)$ and its extensions contain concepts whose translation do not result in guarded formulae. An example of an $\mathcal{ALC}(\sqcap, \neg)$ concept whose translation is not a guarded formula is

$$\forall\neg(\mathit{likes} \sqcap \mathit{eats}).\neg\mathit{Cheese},$$

that is, the set of cheese lovers.

Making economical use of first-order variables in the mapping π, concepts of $\mathcal{ALC}(\neg)$ and many other description logics can be translated into finite-variable function-free fragments of first-order logic. One of them is the *two variable fragment* FO^2. The fragment FO^2 consists of those formulae of first-order logic that can be written using only two variables. The description logic which is most closely linked to the two variable fragment is $\mathcal{ALC}(\sqcap, \sqcup, \neg, ^{\smile})$, also known as $\mathcal{ALB}$ [16], i.e. $\mathcal{ALC}$ extended with full Boolean operations on roles and the inverse operator. It follows from [17] that the computational complexity of the satisfiability problem in $\mathcal{ALB}$ is NExpTime-complete. In [10] it is shown that the satisfiability problem of FO^2 is NExpTime-complete and can be reduced to the *dyadic Scott class*, i.e. the Scott class defined over predicate symbols with arity less than or equal to two. A formula belongs to the Scott class if it is a conjunction of formulae in prefix normal form and the quantifier prefixes are $\forall\forall$ or $\forall\exists$. It is not difficult to see that $\mathcal{ALB}$ formulae can be effectively translated into the dyadic Scott class. Thus satisfiability of both $\mathcal{ALB}$ formulae 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 [10]), whereas interestingly the restriction of $\mathcal{ALB}$ to a bounded number of role names is in ExpTime (this follows from [19]). If FO^2 is assumed to include equality then the extension of $\mathcal{ALB}$ with the identity role is most closely linked with FO^2. In fact, [19] show expressive equivalence.

One extension of FO^2 is the dual of Maslov's class K, denoted by $\overline{\mathrm{K}}$. The sense in which $\overline{\mathrm{K}}$ extends FO^2 is that the normal forms for FO^2 formulae of Mortimer [22] belong to $\overline{\mathrm{K}}$. 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 $\exists y_1 \ldots \exists y_m \forall x_1 Q_1 z_1 \ldots Q_n z_n$, where $m \geq 0, n \geq 0, Q_i \in \{\exists, \forall\}$ for all i, $1 \leq i \leq n$, then $\forall x_1 Q_1 z_1 \ldots Q_n z_n$ is the terminal φ-prefix. For a φ prefix $\exists y_1 \ldots \exists y_m$ the terminal φ-prefix is the empty sequence of quantifiers. By definition, a closed formula φ in negation normal form belongs to the class $\overline{\mathrm{K}}$ if there are k quantifiers $\forall x_1, \ldots, \forall x_k$, $k \geq 0$, in φ such that for every atomic subformula ψ of φ the terminal φ-prefix of ψ is either (i) of length less or equal to 1, or (ii) ends with an existential quantifier, or (iii) is of the form $\forall x_1 \forall x_2 \ldots \forall x_k$.

Consider the following formula φ_1 which defines the concept *mwc* as a subset of married couples with a child:

$$\forall x \forall y (mwc(x,y) \rightarrow (married(x,y) \wedge \exists z (has\_child(x,z) \wedge has\_child(y,z)))).$$

The formula φ_1 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 it is in $\overline{\mathrm{K}}$. In order to show this we check whether the conditions of the definition of $\overline{K}$ 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 $\forall x \forall y$, so $mwc(x,y)$ satisfies condition (iii). 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 (ii). Thus, 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, and they are not identical.

So, the guarded fragment is not a subfragment of $\overline{\mathrm{K}}$ nor is $\overline{\mathrm{K}}$ a subfragment of the guarded fragment. However, $\overline{\mathrm{K}}$ contains a variety of classical solvable fragments, namely the monadic class, the initially extended Skolem class, the Gödel class, and FO^2 as well as a range of non-classical logics, like a number of extended modal logics, many description logics, reducts of representable relation algebras [15]. A resolution decision procedure for $\overline{\mathrm{K}}$ as well as for the class $\overline{\mathrm{DK}}$ consisting of conjunction of formulae in $\overline{\mathrm{K}}$ is presented in [15].

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

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

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 $married(x_1,x_2)$ and $have\_child(x_1,x_2,x_3)$ 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, and are not identical.

The formula φ_3 belongs to yet another solvable fragment of first-order logic, namely *fluted logic*. Fluted logic is defined over a finite set of predicate symbols $\mathcal{P}$ and an ordered set of variables $X_m = \{x_1, \dots, x_m\}$. 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 inductively defined as follows. (i) Any atomic fluted formula over X_i is a fluted formula over X_i. (ii) $\exists x_{i+1}\varphi$ and $\forall x_{i+1}\varphi$ are fluted formulae over X_i, if φ is a fluted formula over X_{i+1}. (iii) Any 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 $\mathit{has\_child}(x, z) \wedge \mathit{has\_child}(y, z)$ of φ_1. To satisfy condition (iii) above, $\mathit{has\_child}(x, z)$ and $\mathit{has\_child}(y, z)$ have to be atomic fluted formulae over the same ordered subset X_i of X_m. In the case of $\mathit{has\_child}(x, z)$ this implies that x has to come directly before z in the ordering on the variables. However, in the case of $\mathit{has\_child}(y, z)$ this implies that it is y that has to come directly before z in the ordering on the 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 $\mathit{mwd}(x, y) \rightarrow \forall z(\mathit{have\_child}(y, x, z) \rightarrow \mathit{doctor}(z))$ of φ_2. First of all, to satisfy condition (iii) above both $\mathit{mwd}(x, y)$ and $\forall z(\mathit{have\_child}(y, x, z) \rightarrow \mathit{doctor}(z))$ have to be fluted formulae over the same ordered subset X_i of X_m, and to satisfy condition (ii), the implication $\mathit{have\_child}(y, x, z) \rightarrow \mathit{doctor}(z)$, and consequently $\mathit{have\_child}(y, x, z)$ have to be fluted formulae over X_{i+1}. In the case of $\mathit{mwd}(x, y)$ this implies that x has to come directly before y in the ordering on variables, while in the case of $\mathit{have\_child}(y, x, z)$ we see that it is y that has to come directly before x. Again, both constraints on the ordering on variables cannot be satisfied at the same time.

Because in fluted logic the relational atoms may be negated, $\mathcal{ALC}(\neg)$, $\mathcal{ALC}(\sqcup, \sqcap, \neg)$, and the Boolean modal logic can be embedded into fluted logic [28]. More precisely, it can be shown that translations of description logic and modal logic formulae by both the relational translation and a variation of the functional translation are fluted formulae [27]. 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*, i.e. 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 $\mathcal{ALC}(\sqcup, \sqcap, \neg)$ formulae into first-order logic.

1. For any set of global axioms Δ and any formula φ in $\mathcal{ALC}(\sqcup, \sqcap, \neg)$, $\Delta \models_{\mathcal{ALC}(\sqcup, \sqcap, \neg)} \varphi$ iff $\forall x\, \pi(\Delta, x) \wedge \neg\forall x\, \pi(\varphi, x)$ is unsatisfiable in first-order logic.
2. For any formula φ in $\mathcal{ALC}(\sqcup, \sqcap, \neg)$, $Qx\, \pi(\varphi, x)$ is a dyadic fluted formula, where $Q \in \{\forall, \exists\}$.
3. For any closed dyadic fluted formula ψ there is a formula φ of

$\mathcal{ALC}(\sqcup, \sqcap, \neg)$ such that ψ is logically equivalent to $Qx\,\pi(\varphi, x)$, 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 [27]. This fragment can be defined like fluted logic except that atomic formulae over X_i have the form $P(x_1, x_2, \ldots, x_i)$. The functional modal fragment was first defined by Herzig [11] 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 [23] which reduces the functional modal fragment into the Bernays-Schönfinkel class (the $\exists^*\forall^*$ prefix class) [14, §4]. Permutability 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 modal logic K. In [27] 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 [24]. 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 logic $K_{(m)}(\cap, \cup, ^-, ^\smile)$, and also FO^2. Fluted logic is decidable by an ordering refinement of first-order resolution and a new form of dynamic renaming, called separation [28].

Therefore, we can embed $\mathcal{ALC}$ into the guarded fragment, $\overline{\mathrm{K}}$, as well as fluted logic, which 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 relationship is summarised in Figure 1.

4 Beyond $\mathcal{ALC}$ and decidability

If instead of extensions of $\mathcal{ALC}$ by role-forming operators we look at 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 we have looked at so far. 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,

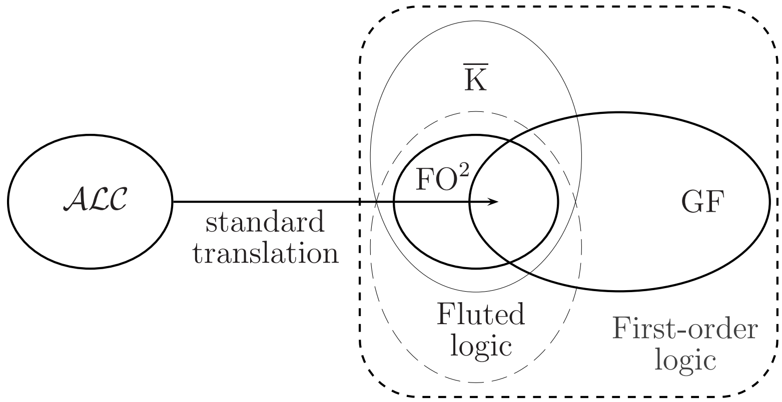

Figure 1: The relationship of $\mathcal{ALC}$, FO^2, GF, $\overline{\text{K}}$, and fluted logic

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 $\mathcal{FALCM}$ [13], $\mathcal{ALC}_{\mathcal{M}}$ [31] and $K_{\mathcal{ALC}}$ [20].

In the following we focus on a slight variation of $K_{\mathcal{ALC}}$ with subconcept definitions instead of concept equivalence. The language is defined as follows. Let A, R, O, and I be countably infinite sets of concept names, role names, object names, and agent names, respectively. The set of $K_{\mathcal{ALC}}$ concepts is inductively defined as follows. All concept names as well as $\top$ are concepts. If C and D are concepts, R is a role name, and i is an agent name, then the following expressions are concepts: $\neg C$, $C \sqcap D$, $\forall R.C_i$, and $\Box_i C_i$. $K_{\mathcal{ALC}}$ formulae are inductively defined as follows. If C and D are concepts and a is an object name, then $C \dot{\sqsubseteq} D$ and $a : C$ are atomic formulae. If φ and ψ are formulae and i is an agent name, then the following expressions are formulae: $\neg\varphi$, $\phi \wedge \psi$, and $\Box_i\varphi$. Thus, in $K_{\mathcal{ALC}}$, modal operators can be applied to both concepts and terminological axioms, but not to roles.

The semantics of $K_{\mathcal{ALC}}$ is a mixture of the possible world semantics of the modal logic K and the set theoretical 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 τ of $K_{\mathcal{ALC}}$ into first-order logic. On concepts τ is defined as follows.

$$\tau(A, W, X) = Q_A(W, X) \qquad \tau(\top, W, X) = \top$$
$$\tau(\neg C, W, X) = \neg\tau(C, W, X)$$
$$\tau(C \sqcap D, W, X) = \tau(C, W, X) \wedge \tau(D, W, X)$$
$$\tau(\forall R.C, W, X) = \forall y(Q_R(W, X, y) \rightarrow \tau(C, W, y))$$
$$\tau(\Box_i C, W, X) = \forall v(R_i(W, v) \rightarrow \tau(C, v, X))$$

Here X and W are meta variables for first-order terms, and Q_A and Q_R denote unary and binary predicate symbols, uniquely associated with a concept symbol A and a role symbol R, respectively. R_i is a binary predicate symbol representing the accessibility relation associated with the modal operator $\Box_i$. We extend τ to $K_{\mathcal{ALC}}$ formulae as follows.

$$\tau(a : C, W) = \tau(C, W, \underline{a})$$
$$\tau(C \dot{\sqsubseteq} D, W) = \forall x(\tau(C, W, x) \to \tau(D, W, x))$$
$$\tau(\varphi \wedge \psi, W) = \tau(\varphi, W) \wedge \tau(\psi, W) \qquad \tau(\neg\varphi, W) = \neg\tau(\varphi, W)$$
$$\tau(\Box_i\varphi, W) = \forall v(R_i(W, v) \to \tau(\varphi, v))$$

Here $\underline{a}$ denotes the Skolem constant associated with the object name a. Finally, define $T(\varphi) = \tau(\varphi, \epsilon)$, where ϵ is a constant symbol, for any $K_{\mathcal{ALC}}$ formula φ.

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

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

Its standard translation to first-order logic by the mapping T defined above is the following formula φ_4.

$$\forall w(R_{Tim}(\epsilon, w) \to \forall x(Minis(w, x) \to \forall v(R_{Don}(w, v) \to slow\_car(v, x))))$$

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. 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 $R_{Don}(w, v)$ omits the x which should appear between w and v in a fluted formula. φ_4 is 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 [2]. 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. $\exists y\phi$ is a positive condition if ϕ is a positive condition. Ranges for variables $x_1, \ldots, x_n$ are inductively defined as follows. (i) An atom in which all of $x_1, \ldots, x_n$ occur is a range for $x_1, \ldots, x_n$. (ii) $\rho_1 \vee \rho_2$ is a range for $x_1, \ldots, x_n$ if both ρ_1 and ρ_2 are ranges for $x_1, \ldots, x_n$. (iii) $\rho \wedge \phi$ is a range for $x_1, \ldots, x_n$ if ρ is a range for $x_1, \ldots, x_n$ and ϕ is a positive condition. (iv) $\exists y\rho$ is a range for $x_1, \ldots, x_n$ if ρ is a range for $y, x_1, \ldots x_n$, and if $x_i \neq y$ for all $i = 1, \ldots, n$. Positive formulae with restricted quantification are then inductively defined as follows. (i) Atoms are PRQ formulae. (ii) The nullary logical connectives $\top$ and $\bot$ are PRQ formulae. (iii) Conjunctions and disjunctions of PRQ formulae are PRQ formulae. (iv) A formula of the form $\phi \to \psi$ is a PRQ formula if ϕ is a positive condition and ψ is a PRQ formula. (v) A formula of the form $\forall x_1 \ldots \forall x_n(\rho \to \psi)$, $n \geq 1$, is a PRQ formula if ρ is a range for $x_1, \ldots, x_n$, and if ψ is a PRQ formula. (vi) 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, $\forall v(R_{Don}(w, v) \to slow\_car(v, x))$ is a PRQ formula. Since $Minis(w, x)$ is a range for x, it follows that the subformula $\forall x(Minis(w, x) \to \ldots)$ 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. However, for fragments of PRQ that have the finite model property, there is a decision procedure in the form of an extended positive tableaux method [30]. The method does not only detect unsatisfiability but also generates finite models if they exist. Since $\mathcal{ALC}$ and many of its extensions have the finite model property, this procedure procedure provides a general, sound, complete, and terminating method for solving the satisfiability problem for these logic without the necessity of additional soundness, completeness or termination proofs.

5 Conclusion

In this short survey we considered the relationship of the description logic $\mathcal{ALC}$ and its extensions to nonclassical logics and fragments of first-order logics, thereby providing a new perspective of description logics and the kind of reasoning methodologies applicable to description logics.

In this paper the pairwise orthogonality of the logics is shown only on the syntactic level. To the best of our knowledge there have not been any investigations of semantical equivalence thus far.

References

[1] H. Andréka, I. Németi, and J. van Benthem. Modal languages and bounded fragments of predicate logic. *J. Philos. Logic*, 27(3):217–274, 1998.

[2] 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.

[3] H. de Nivelle and M. de Rijke. Deciding the guarded fragments by resolution. *J. Symbolic Computat.*, 2002. To appear.

[4] H. de Nivelle, R. A. Schmidt, and U. Hustadt. Resolution-based methods for modal logics. *Logic J. IGPL*, 8(3):265–292, 2000.

[5] 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.

[6] 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.

[7] 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.

[8] E. Grädel. Guarded fragments of first-order logic: a perspective for new description logic. In *Proc. DL'98*, pp. 5–8. Instituto Trentino di Cultura, 1998.

[9] E. Grädel. On the restraining power of guards. *J. Symbolic Logic*, 64:1719–1742, 1999.

[10] E. Grädel, P. Kolaitis, and M. Vardi. On the decision problem for two-variable first-order logic. *Bull. Symbolic Logic*, 3:53–69, 1997.

[11] A. Herzig. A new decidable fragment of first order logic, June 1990. In Abstracts of the 3rd Logical Biennial, Summer School & Conference in honour of S. C. Kleene, Varna, Bulgaria.

[12] C. Hirsch and S. Tobies. A tableau algorithm for the clique guarded fragment. In *Advances in Modal Logic, Volume 3*. CSLI Publications, 2001. To appear.

[13] 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.

[14] U. Hustadt and R. A. Schmidt. An empirical analysis of modal theorem provers. *J. Appl. Non-Classical Logics*, 9(4):479–522, 1999.

[15] 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.

[16] 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.

[17] C. Lutz and U. Sattler. The complexity of reasoning with boolean modal logics. In *Advances in Modal Logics Volume 3*. CSLI Publications, Stanford, 2001.

[18] C. Lutz, U. Sattler, and S. Tobies. A suggestion of an n-ary description logic. In *Proc. DL'99*, pp. 81–85. Linköping University, 1999.

[19] C. Lutz, U. Sattler, and F. Wolter. Description logics and the two-variable fragment. In *Proc. DL'01*, pp. 66–75, 2001.

[20] 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.

[21] 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.

[22] M. Mortimer. On languages with two variables. *Z. Math. Logik Grundlagen Math.*, 21:135–140, 1975.

[23] 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.

[24] W. C. Purdy. Quine's 'limits of decision'. *J. Symbolic Logic*, 64(4):1439–1466, 1999.

[25] W. V. Quine. Variables explained away. In *Proceedings of the American Philosophy Society*, vol. 104, pp. 343–347, 1960.

[26] 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.

[27] R. A. Schmidt and U. Hustadt. Deciding fluted logic with resolution. Manuscript. Submitted to *Inform. and Computat.*, 2000.

[28] 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.

[29] M. Schmidt-Schauß and G. Smolka. Attributive concept descriptions with complements. *J. Artificial Intelligence*, 48:1–26, 1991.

[30] S. Torge. *Überprüfung der Erfüllbarkeit im Endlichen: Ein Verfahren und seine Anwendung*. PhD thesis, Herbert Utz Verlag, München, 1998.

[31] F. Wolter and M. Zakharyaschev. Multi-dimensional description logics. In *Proc. IJCAI'99*, pp. 104–109. Morgan Kaufmann, 1999.