Description Logics and the Two-Variable Fragment

Carsten Lutz†, Ulrike Sattler†, and Frank Wolter‡

† LuFg Theoretical Computer Science
RWTH Aachen, Germany
{lutz, sattler}@cs.rwth-aachen.de

‡ Institute for Computer Science
University of Leipzig, Germany
wolter@informatik.uni-leipzig.de

Abstract

We present a description logic $\mathcal{L}$ that is as expressive as the two-variable fragment of first-order logic and differs from other logics with this property in that it encompasses solely standard role- and concept-forming operators. The description logic $\mathcal{L}$ is obtained from $\mathcal{ALC}$ by adding full Boolean operators on roles, the inverse operator on roles and an identity role. It is proved that $\mathcal{L}$ has the same expressive power as the two-variable fragment FO^2 of first-order logic by presenting a translation from FO^2-formulae into equivalent $\mathcal{L}$-concepts (and back). Additionally, we discuss an interesting complexity phenomenon: both $\mathcal{L}$ and FO^2 are NExpTime-complete and so is the restriction of FO^2 to finitely many relation symbols; astonishingly, the restriction of $\mathcal{L}$ to a bounded number of role names is in ExpTime.

1 Introduction

It is well-known that many description and modal logics can be regarded as fragments of first-order logics. In modal logic, the relationship between modal and first-order logic has been a major research topic: Kamp's result [17] that modal logic with binary operators *Since* and *Until* has the same expressive power as monadic first-order logic over structures like $\langle \mathbb{N}, < \rangle$ and $\langle \mathbb{R}, < \rangle$ was the starting point. Van Benthem [24] provided a systematic model theoretic analysis of the relation between families of modal logics and predicate logics, and Gabbay [10] extended Kamp's result to a systematic investigation of the possibilities of designing expressively complete modal logics. As part of his investigation, Gabbay made the basic observation that often modal logics are contained in *finite variable fragments* of first-order logics. Like many description logics, the basic modal logics lie embedded in the two-variable fragment FO^2 (i.e., those first-order formulae that can be written using only variables x and y, possibly re-quantifying a variable) of first-order logic. In the early 90's, this observation was regarded as an explanation for the decidability of many modal logics: the decidability of FO^2 (c.f. [21, 22, 14]) explains the decidability of standard modal logics simply because the latter are fragments of the former. In contrast, more recently, it has been argued that some "modal phenomena" are better explained by their *tree-model-property* [25] (i.e., they are determined by tree-like structures) and/or by embedding them in bounded (or guarded) fragments of first-order logic [1, 13].

Many of the above observations apply to standard description logics as well. The relationship between first-order logic and description logics was first investigated by Borgida who identifies several DLs that are fragments of FO^2 and even presents a description logic $\mathcal{D}$ that is as expressive as FO^2 itself. More precisely, Borgida gives a linear translation from FO^2-formulae into equivalent $\mathcal{D}$-concepts (and back). However, $\mathcal{D}$ uses a role-forming product operator which is, to the best of our knowledge, not present in any standard description logic. This operator allows to form, from two concepts C_1, C_2, the role $C_1 \times C_2$ relating all instances of C_1 to all instances of C_2 and thus corresponds to a simultaneous range- and domain-restriction of the universal role. As will be discussed in more detail later, the presence of the product role constructor makes the translation from FO^2 into $\mathcal{D}$ rather straightforward.

In the light of these considerations, it is an interesting question whether there exists a description logic using only *standard* role- and concept-forming operators that has the same expressive power as FO^2. In this paper, we present a description logic $\mathcal{L}$ and give a translation from FO^2-formulae into equivalent $\mathcal{L}$-concepts (and back). Roughly speaking, $\mathcal{L}$ is $\mathcal{ALC}$ extended with full Boolean operators on roles, the inverse operator on roles, and an identity role relating each individual to itself. We argue that all operators present in $\mathcal{L}$ are standard description logic operators: see, e.g., [8, 5, 18] for Boolean operators on roles, [16] for Boolean operators on roles and inverse roles, and [6] for (some) Boolean operators on roles, inverse, and the identity role. Moreover, the modal-logic equivalent of $\mathcal{L}$ is also a standard modal logic; see, e.g., [11, 15, 19] for Boolean operators on modal parameters, [4, 12, 26] for converse, and [7] for the identity relation. Thus $\mathcal{L}$ is the positive answer to the above question for description *and* modal logics.

What kind of description logic is $\mathcal{L}$? The expressive power provided by the strong role-forming operators in $\mathcal{L}$ is demonstrated by the following three examples:

1. $\mathcal{L}$ can express the universal role: for a role name R, $R \sqcup \neg R$ is always interpreted as the universal relation, and hence $\forall(R \sqcup \neg R).C$ admits only models where each individual is an instance of C. As a consequence, general concept inclusion axioms can be internalised, and thus reasoning

w.r.t. to the most general form of TBoxes can be linearly reduced to concept satisfiability.

2. Since $\mathcal{L}$ contains the identity role id, we can express the difference role: an individual is related to all other individuals but itself via $\neg id$, and hence $\forall \neg id.C$ expresses that C holds "everywhere else".

3. Using the difference role $\neg id$ and the universal role $R \sqcup \neg R$, we can express nominals, i.e., concepts N which have at most one instance in each model:

$$(\forall(R \sqcup \neg R).\neg N) \sqcup \exists(R \sqcup \neg R).(N \sqcap \forall \neg id.\neg N).$$

The first disjunct is for the case that no instance of N exists, and the second one guarantees that, if there is an instance of N somewhere in the model, then all other individuals are not instances of N.

In contrast to the linear translation of FO^2 into Borgida's logic $\mathcal{D}$, our translation of FO^2-formulae into $\mathcal{L}$-concepts involves an exponential blow-up in formula size. Have we been to lazy to find a linear reduction? Fortunately, we can argue that the blow-up can (presumably) not be circumvented: by analyzing the complexity of the involved logics, we show that the existence of a polynomial translation of FO^2-formulae into $\mathcal{L}$-concepts would imply that NExpTime = ExpTime.

2 Preliminaries

We start with precise definitions of the languages under consideration. FO^2 comprises exactly those first-order formulas without constants and function symbols but with equality whose only variables are x and y and whose relation symbols have arity ≤ 2. We use A_i for unary predicates and R_i for binary relations. If we write $\varphi(x)$, $\varphi(y)$ for formulas, we assume that at most the displayed variable occurs free in φ.

FO^2 is interpreted in the standard manner in interpretations of the form $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \mathcal{A}_1, \ldots, \mathcal{R}_1, \ldots \rangle$ in which $\Delta^{\mathcal{I}}$ is the interpretation domain, the $\mathcal{A}_i$ interpret the A_i, and the $\mathcal{R}_i$ interpret the R_i. For an interpretation $\mathcal{I}$, some $a \in \Delta^{\mathcal{I}}$, and a formula $\varphi(x)$, we write $\mathcal{I} \models \varphi[a]$ if $\mathcal{I}, v \models \varphi(x)$ for the assignment v that maps x to a.

Note that not admitting constants is not crucial. In fact, in our version of FO^2 constants can be "simulated" through unary predicates similar to the simulation of nominals in $\mathcal{L}$ described in the introduction. To the contrary, function symbols cannot be admitted without loosing decidability [2].

The description logic $\mathcal{L}$ is $\mathcal{ALC}$ extended with Boolean operators on roles, the inverse operator on roles, and the identity role. Here is the formal definition:

Definition 1 Let $N_R = \{R_1, R_2, \ldots\}$ and $N_C = \{A_1, A_2, \ldots\}$ be disjoint sets of *role names* and *concept names*, respectively. The set of $\mathcal{L}$-*roles* is defined inductively as follows:

- atomic roles and *id* are $\mathcal{L}$-roles, and
- if R and S are $\mathcal{L}$-roles, then $R \sqcap S$, $R \sqcup S$, $\neg R$, and R^- are $\mathcal{L}$-roles.

$\mathcal{L}$-*concepts* are also defined inductively:

- each concept name is an $\mathcal{L}$-concept, and
- if R is a role and C and D are $\mathcal{L}$-concepts, then $C \sqcap D$, $C \sqcup D$, $\neg C$, $\forall R.C$, and $\exists R.C$ are $\mathcal{L}$-concepts.

We abbreviate $\top = A_i \sqcup \neg A_i$ and $\bot = \neg\top$ for some concept name A_i. The bi-implication "$\Leftrightarrow$" is defined as an abbreviation in the standard manner.

The semantics of $\mathcal{L}$ is a straightforward extension of the standard $\mathcal{ALC}$-semantics.

Definition 2 An $\mathcal{L}$-*interpretation* $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$ consists of a non-empty set $\Delta^\mathcal{I}$, the *domain*, and a function $\cdot^\mathcal{I}$ that maps

- concept names A_i to subsets $A_i^\mathcal{I} \subseteq \Delta^\mathcal{I}$ of the domain and
- role names R_i to binary relations $R_i^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$ on the domain.

This mapping is extended to complex roles R, S and complex concepts C, D as follows:

$$\begin{aligned}
id^\mathcal{I} &= \{(x,x) \mid x \in \Delta^\mathcal{I}\} \\
(R^-)^\mathcal{I} &= \{(y,x) \mid (x,y) \in R^\mathcal{I}\} \\
(\neg R)^\mathcal{I} &= \Delta^\mathcal{I} \times \Delta^\mathcal{I} \setminus R^\mathcal{I} \\
(R \sqcap S)^\mathcal{I} &= R^\mathcal{I} \cap S^\mathcal{I} \\
(R \sqcup S)^\mathcal{I} &= R^\mathcal{I} \cup S^\mathcal{I}
\end{aligned}$$

$$\begin{aligned}
(\neg C)^\mathcal{I} &= \Delta^\mathcal{I} \setminus C^\mathcal{I} \\
(C \sqcap D)^\mathcal{I} &= C^\mathcal{I} \cap D^\mathcal{I} \\
(C \sqcup D)^\mathcal{I} &= C^\mathcal{I} \cup D^\mathcal{I} \\
(\exists R.C)^\mathcal{I} &= \{x \in \Delta^\mathcal{I} \mid \text{ there exists } y \in C^\mathcal{I} \text{ with } (x,y) \in R^\mathcal{I}\} \\
(\forall R.C)^\mathcal{I} &= \{x \in \Delta^\mathcal{I} \mid \text{ for all } y, \text{ if } (x,y) \in R^\mathcal{I}, \text{ then } y \in C^\mathcal{I}\}.
\end{aligned}$$

A concept C is called *satisfiable* if, for some interpretation $\mathcal{I}$, $C^\mathcal{I} \neq \emptyset$. Such an interpretation is called a *model* of C.

Obviously, there is a 1–1 correspondence between first-order interpretations and $\mathcal{L}$-interpretations: setting $A_i^{\mathcal{I}} = \mathcal{A}_i$ and $R_i^{\mathcal{I}} = \mathcal{R}_i$, each first-order interpretation $\langle \Delta^{\mathcal{I}}, \mathcal{A}_1, \ldots, \mathcal{R}_1, \ldots \rangle$ can be viewed as an $\mathcal{L}$-interpretation $(\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ and vice versa. Hence we do not distinguish between both kinds of interpretations.

3 The correspondence between FO^2 and $\mathcal{L}$

In this section, we show that FO^2 and $\mathcal{L}$ are equally expressive. The main difficulty is to devise a translation that maps each FO^2-formula to an equivalent $\mathcal{L}$-concept.

Theorem 1 For every $C \in \mathcal{L}$ there exists a formula $\varphi_C(x) \in FO^2$ whose length is linear in the length of C such that, for all interpretations $\mathcal{I}$ and all $a \in \Delta^{\mathcal{I}}$, $a \in C^{\mathcal{I}}$ iff $\mathcal{I} \models \varphi_C[a]$.

Conversely, given $\varphi(x) \in FO^2$ there exists an $\mathcal{L}$-concept C_φ whose length is exponential in the length of $\varphi(x)$ such that, for all interpretations $\mathcal{I}$ and all $a \in \Delta^{\mathcal{I}}$, $a \in C_\varphi^{\mathcal{I}}$ iff $\mathcal{I} \models \varphi[a]$.

Proof: The proof of the first claim is standard (see, e.g., [3]), so we concentrate on the second one. It is rather similar to the proof provided in [9] for temporal logics.

An FO^2-formula $\rho(x, y)$ is called a *binary atom* if it is an atom of the form $\mathsf{R}_i(x, y)$, $\mathsf{R}_i(y, x)$, or $x = y$. A *binary type* t for a formula ψ is a set of FO^2-formulas containing (i) either χ or $\neg\chi$ for each binary atom χ occurring in ψ, (ii) either $x = y$ or $x \neq y$, and (iii) no other formulas than these. The set of binary types for ψ is denoted by $\mathcal{R}_\psi$. A formula ξ is called a *unary atom* if it is of the form $\mathsf{R}_i(x, x)$, $\mathsf{R}_i(y, y)$, $\mathsf{A}_i(x)$, or $\mathsf{A}_i(y)$.

Let $\varphi(x) \in FO^2$. Without loss of generality, we assume $\varphi(x)$ is built using $\exists$, $\wedge$, and $\neg$ only. We inductively define two mappings $\cdot^{\sigma_x}$ and $\cdot^{\sigma_y}$ where the former one takes FO^2-formulas $\varphi(x)$ to the corresponding $\mathcal{L}$-concepts φ^{σ_x} and the latter does the same for FO^2-formulas $\varphi(y)$. We only give the details of $\cdot^{\sigma_x}$ since $\cdot^{\sigma_y}$ is defined analogously by switching the roles of x and y.

Case 1. If $\varphi(x) = \mathsf{A}_i(x)$, then put $(\varphi(x))^{\sigma_x} = A_i$.

Case 2. If $\varphi(x) = \mathsf{R}_i(x, x)$, then put $(\varphi(x))^{\sigma_x} = \exists(id \sqcap R_i).\top$.

Case 3. If $\varphi(x) = \chi_1 \wedge \chi_2$, then put, recursively, $(\varphi(x))^{\sigma_x} = \chi_1^{\sigma_x} \wedge \chi_2^{\sigma_x}$.

Case 4. If $\varphi(x) = \neg\chi$, then put, recursively, $(\varphi(x))^{\sigma_x} = \neg(\chi)^{\sigma_x}$.

Case 5. If $\varphi(x) = \exists y \chi(x, y)$, then $\chi(x, y)$ can clearly be written as

$$\chi(x, y) = \gamma[\rho_1, \ldots, \rho_r, \gamma_1(x), \ldots, \gamma_\ell(x), \xi_1(y), \ldots, \xi_s(y)],$$

i.e., as a Boolean combination γ of ρ_i, $\gamma_i(x)$, and $\xi_i(y)$; the ρ_i are binary atoms; the $\gamma_i(x)$ are unary atoms or of the form $\exists y\gamma_i'$; and the $\xi_i(y)$ are unary atoms or of the form $\exists x\xi_i'$. We may assume that x occurs free in $\varphi(x)$. Our first step is to move all formulas without a free variable y out of the scope of $\exists$: for $\vec{w} = \langle w_1, \ldots, w_\ell \rangle$, $\varphi(x)$ is equivalent to

$$\bigvee_{\vec{w}\in\{\top,\bot\}^\ell} \Big(\bigwedge_{1\leq i\leq\ell} (\gamma_i \Leftrightarrow w_i) \land \exists y\gamma(\rho_1, \ldots, \rho_r, w_1, \ldots, w_\ell, \xi_1, \ldots, \xi_s)\Big). \qquad (1)$$

For every binary type $t \in \mathcal{R}_\varphi$ and binary atom ρ_i from φ, we have $t \models \rho_i$ or $t \models \neg\rho_i$—hence we can "guess" a binary type t and then replace all binary atoms by either true or false. For $t \in \mathcal{R}_\varphi$, let $\rho_i^t = \top$ if $t \models \rho_i$, and $\rho_i^t = \bot$, otherwise. Then $\varphi(x)$ is equivalent to

$$\begin{aligned}&\textstyle\bigvee_{\vec{w}\in\{\top,\bot\}^\ell}(\bigwedge_{1\leq i\leq\ell}(\gamma_i \Leftrightarrow w_i) \land \\ &\qquad\qquad \textstyle\bigvee_{t\in\mathcal{R}_\varphi} \exists y((\bigwedge_{\alpha\in t} \alpha) \land \gamma(\rho_1^t, \ldots, \rho_r^t, w_1, \ldots, w_\ell, \xi_1, \ldots, \xi_s))).\end{aligned} \qquad (2)$$

Define, for every negated and unnegated binary atom α, a role α^{σ_x} as follows:

$$\begin{array}{rclrcl} (x = y)^{\sigma_x} & = & id & (\neg(x = y))^{\sigma_x} & = & \neg id \\ (\mathsf{R}_i(x, y))^{\sigma_x} & = & R_i & (\neg\mathsf{R}_i(x, y))^{\sigma_x} & = & \neg R_i \\ (\mathsf{R}_i(y, x))^{\sigma_x} & = & R_i^- & (\neg\mathsf{R}_i(y, x))^{\sigma_x} & = & \neg R_i^- . \end{array}$$

Put, for every binary type $t \in \mathcal{R}_\varphi$, $t^{\sigma_x} = \bigsqcap_{\alpha\in t} \alpha^{\sigma_x}$. Now compute, recursively, $\gamma_i^{\sigma_x}$ and $\xi_i^{\sigma_y}$, and define $\varphi(x)^{\sigma_x} = C_\varphi$ as

$$\bigsqcup_{\vec{w}\in\{\top,\bot\}^\ell} \Big(\bigsqcap_{1\leq i\leq\ell} (\gamma_i^{\sigma_x} \Leftrightarrow w_i) \sqcap \bigsqcup_{t\in\mathcal{R}_\varphi} \exists t^{\sigma_x}.\gamma(\rho_1^t, \ldots, \rho_r^t, w_1, \ldots, w_\ell, \xi_1^{\sigma_y}, \ldots, \xi_s^{\sigma_y})\Big).$$

❑

Note that C_φ can be computed in time polynomial in the length of C_φ. It is easily seen that there exist formulas C whose translation yields an $\mathcal{L}$-concept C_φ with length exponential in the length of φ.

Let us compare the above translation with the one presented in [3]. Borgida's logic $\mathcal{D}$ offers the same concept and role operators provided by $\mathcal{L}$ with two exceptions:

1. instead of the identity role, $\mathcal{D}$ provides for nominals. As already argued in the introduction, nominals are implicitly available in $\mathcal{L}$. The converse does not hold, i.e., the identity role cannot be "simulated" in $\mathcal{D}$. It is thus not surprising that Borgida translates FO^2 without equality but with constants;

2. the logic $\mathcal{D}$ provides for an additional role operator that allows to build complex roles from concepts C_1, C_2 as their product $C_1 \times C_2$, where $(C_1 \times C_2)^{\mathcal{I}} = C_1^{\mathcal{I}} \times C_2^{\mathcal{I}}$.

While the first difference is more of a technical nature, the second one makes the translation of FO^2-formulae into $\mathcal{D}$-concepts almost trivial: the main difficulty in the translation to $\mathcal{L}$-concepts described above consists in finding, for the sub-formula in two variables $\chi(x, y)$ of $\exists y.\chi(x, y)$ in Case 5, an equivalent re-writing as $\exists y.(G(x, y) \wedge \chi'(y))$ such that $G(x, y)$ can be translated into a complex role. The translation of χ' can then be done recursively. In contrast, the translation given in [3] translates the whole formula $\chi(x, y)$ as a (complex) role, i.e., translating $\exists y.\chi(x, y)$ yields $\exists R_\chi.\top$, where R_χ is defined recursively. This is only possible since, for example, subformulae $\rho(x) \wedge \rho'(y)$ of $\chi(x, y)$ can be translated as the role $C_\rho \times C_{\rho'}$ using the product operator. Intuitively, this operator allows to freely switch back and forth between roles and concepts during the translation. Hence the presence of the (non-standard) product operator is the reason why Borgida is able to find a linear translation of FO^2 into his logic. In the following, we will see that such a translation does (presumably) not exist for the description logic $\mathcal{L}$.

4 Complexity

Let us analyze the computational complexity of the logics considered in the previous sections. The fundamental result is proved in [14], where it is shown that satisfiability of FO^2-formulae is NExpTime-complete.[1] Concerning the Description Logic $\mathcal{L}$, the following corollary to Theorem 1 is easily obtained:

Corollary 1 Satisfiability of $\mathcal{L}$-concepts is NExpTime-complete.

Proof: In [19], it is proved that $\mathcal{ALC}$ extended with intersection and (primitive) negation of roles is already NExpTime-hard. Hence, it remains to prove the upper bound: it is an immediate consequence of Theorem 1 (together with the fact that that formulae φ_C corresponding to $\mathcal{L}$-concepts C can be computed in linear time) and NExpTime-completeness of FO^2 that satisfiability of $\mathcal{L}$-concepts is in NExpTime. ❑

Finally, NExpTime-completeness of Borgida's logic $\mathcal{D}$ is an easy consequence of the above mentioned hardness result in [19] together with the linear translation of $\mathcal{D}$-concepts to FO^2-formula provided in [3].

These results may seem strange on first sight: we have presented an exponential translation from one NExpTime-complete logic into another one. Hence,

[1]We assume that ExpTime is defined as DTIME(2^{n^k}) and NExpTime as NTIME(2^{n^k}).

it is natural to ask whether there exists a polynomial such translation. This question can be answered negatively using the following observations:

1. Let $\mathcal{L}_m$ denote the variant of $\mathcal{L}$ obtained by admitting only m (i.e., finitely many) role names. In [20], we show that, for all $m \in \mathbb{N}$, satisfiability of $\mathcal{L}_m$-formulae is ExpTime-complete. The lower bound is a direct consequence of the facts that (1) for $m \geq 1$, $\mathcal{L}_m$ can express the universal role, and (2) $\mathcal{ALC}$ extended with the universal role is known to be ExpTime-hard [23]. The upper bound is proved in two steps: firstly, $\mathcal{L}_m$-satisfiability is polynomially reduced to the satisfiability of a certain modal logic $\mathcal{L}'$ in a special form of models. Then this latter problem is decided in exponential time by, roughly speaking, enumerating (exponentially many) candidates for type-based abstractions of models and, for each such candidate, checking (in exponential time) whether it does represent a model by using a type elimination technique.

2. In contrast to $\mathcal{L}$, FO^2 restricted to m binary relations (FO^2_m) is still NExpTime-hard [14].

3. Every translation of FO^2-formulas φ into $\mathcal{L}$-concepts C satisfying the conditions from Theorem 1 induces, for each $m \in \mathbb{N}$, a polynomial translation from FO^2_m into $\mathcal{L}_m$ that also satisfies the conditions from Theorem 1: just replace every role R in C that does not occur as a binary predicate in φ by id.

Taking together the above three points, it is obvious that the existence of a polynomial translation from FO^2 into $\mathcal{L}$ would imply that ExpTime = NExpTime. A convenient way to view this result is that FO^2 speaks about relational structures *strictly more succinctly* than $\mathcal{L}$ does.

References

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

[2] E. Börger, E. Grädel, and Y. Gurevich. *The Classical Decision Problem.* Perspectives in Mathematical Logic. Springer-Verlag, Berlin, 1997.

[3] A. Borgida. On the relative expressivness of description logics and predicate logics. *Artificial Intelligence*, 82(1 - 2):353–367, 1996.

[4] J.P. Burgess. Basic tense logic. In *Handbook of Philosophical Logic*, volume 2, pages 89–133. Reidel, Dordrecht, 1984.

[5] D. Calvanese, G. De Giacomo, M. Lenzerini, D. Nardi, and R. Rosati. Description logic framework for information integration. In *Proc. of KR-98*, pages 2–13, 1998.

[6] G. De Giacomo and M. Lenzerini. Tbox and Abox reasoning in expressive description logics. In *Proc. of KR-96*, pages 316–327. Morgan Kaufmann, Los Altos, 1996.

[7] Maarten de Rijke. The modal logic of inequality. *The Journal of Symbolic Logic*, 57(2):566–584, 1992.

[8] F. M. Donini, M. Lenzerini, D. Nardi, and W. Nutt. The complexity of concept languages. *Information and Computation*, 134(1):1–58, 10 April 1997.

[9] K. Etessami, M. Vardi, and T. Wilke. First-order logic with two variables and unary temporal logic. In *Proc. LICS-97*, pages 228–235, 1997.

[10] D.M. Gabbay. Expressive functional completeness in tense logic. In *Aspects of Philosophical Logic*, pages 91–117. Reidel, Dordrecht, 1981.

[11] G. Gargov and S. Passy. A note on boolean modal logic. In *Mathematical Logic and Applications*, 1987. Plenum Press.

[12] R.I. Goldblatt. *Logics of Time and Computation.* Number 7 in CSLI Lecture Notes, Stanford. CSLI, 1987.

[13] E. Grädel. Why are modal logics so robustly decidable? *Bull. of the Eur. Assoc. for Theoretical Computer Science*, 68:90–103, 1999.

[14] E. Grädel, P. Kolaitis, and M. Vardi. On the Decision Problem for Two-Variable First-Order Logic. *Bulletin of Symbolic Logic*, 3:53–69, 1997.

[15] I. L. Humberstone. Inaccessible worlds. *Notre Dame Journal of Formal Logic*, 24(3):346–352, 1983.

[16] U. Hustadt, R. A. Schmidt, and C. Weidenbach. MSPASS: Subsumption testing with SPASS. In *Proc. of DL'99*, pages 136–137. Linköping University, 1999.

[17] H. Kamp. *Tense Logic and the Theory of Linear Order.* PhD Thesis, University of California, Los Angeles, 1968.

[18] C. Lutz and U. Sattler. Mary likes all cats. In *Proc. of DL2000*, number 33 in CEUR Workshop Proceedings, pages 213–226, Aachen, Germany, August 2000.

[19] C. Lutz and U. Sattler. The complexity of reasoning with boolean modal logics. AiML, volume 3. CSLI Publications, Stanford, 2001.

[20] C. Lutz, U. Sattler, and F. Wolter. Modal logic and the two-variable fragment. In *Proceedings of the Annual Conference of the European Association for Computer Science Logic (CSL'01)*, Paris, France, 2001.

[21] M. Mortimer. On languages with two variables. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 21:135–140, 1975.

[22] D. Scott. A decision method for validity of sentences in two variables. *Journal of Symbolic Logic*, 27(377), 1962.

[23] E. Spaan. *Complexity of Modal Logics.* PhD thesis, Department of Mathematics and Computer Science, University of Amsterdam, 1993.

[24] J. van Benthem. Correspondence theory. In *Handbook of Philosophical Logic*, volume 2, pages 167–247. Reidel, Dordrecht, 1984.

[25] M. Vardi. Why is modal logic so robustly decidable? In *Descriptive Complexity and Finite Models*, pages 149–184. AMS, 1997.

[26] F. Wolter. Completeness and decidability of tense logics closely related to logics containing $K4$. *Journal of Symbolic Logic*, 62:131–158, 1997.