Chapter 1

Introduction

1.1 Modal, temporal, and hybrid languages

Modal logic is a powerful, easily utilisable, well-understood, and well-behaved formalism for describing and specifying properties of any application that can be modelled by relational structures. Such applications (and their corresponding relational structures) are, for example

(1) the behaviour of things over time
(points in time and the "later-than" relation);

(2) the knowledge of agents
(states describing the knowledge of the agent together with the relation linking actual states with possible states);

(3) verification of programmes
(states of a machine and transitions between them given by executions of programmes).

In all these applications, modal logic offers a local perspective, that is, it allows for describing things that happen in individual states of relational structures and their successor states. Depending on the particular application, certain variants (or extensions) of modal logic are used — for example, temporal (1), epistemic (2), or dynamic (3) logic.

This thesis will not consider logics for one certain application. Therefore we will prefer an abstract view (in terms of relational structures) to a concrete one (as given in the above examples). However, since many of the languages we examine are useful for temporal applications, we will often speak about them in terms of Example (1) from above. Until we provide a formal definition of relational structures in Chapter 2, we will call them *structures* or *frames*, and refer to their elements as *states* or *points*.

Hybrid languages are extensions of modal logic that allow for naming and accessing states of a structure explicitly. These features are very desirable in many applications. Particularly in the temporal case, it is very natural to give names to points in time and refer to them independently of the "later-than" relation. Besides, by means of hybrid logic, it is possible to capture many temporally relevant properties of structures, such as irreflexivity, antisymmetry, trichotomy, directedness, etc. For these reasons, hybrid languages and hybrid *temporal* languages are of great interest where basic modal and temporal logic reach their limits [BT99, Bla00b, ABM01, FdRS03].

Another reason for the interest in hybrid logic is discussed in [ABM99] and [ABM00]. Hybrid languages are proof-theoretically well behaved and "internalise" labelled deduction [Bla00a], an apparatus that guides proof search in modal logic [Gab96].

Hybrid Logic, as well as the foundations of temporal logic, goes back to Arthur Prior [Pri67]. Since then, many — more or less powerful — languages have been studied [Bul70, PT91, Bla93, GG93, BS95, Gor96, BS98, ABM99, ABM00, ABM01, FdRS03]. The main features of hybrid logic that are of special interest for this thesis are the following.

Nominals. They are special atomic propositions that give names to states — a very natural thing for applications, particularly temporal ones. With the help of nominals, it is possible to express properties of structures that are not expressible in modal logic, such as irreflexivity, asymmetry, etc.

The satisfaction operators. They allow for jumping to a point named by a nominal, regardless of the accessibilities in the structure.

Hybrid binders. They allow for binding names to states dynamically and for referring to these states later on. This makes them a very powerful and desirable means of expression, especially if they are combined with satisfaction operators. Unfortunately, due to this high expressive power, binders are dangerous in terms of computational costs.

Furthermore, we will consider operators that occur in the context of modal logic, too.

The "until" and "since" operators. They permit temporal statements such as: "Until some point with property ψ, it is always the case that φ." This notion of "betweenness" cannot be expressed by the usual temporal operators, which only allow for accessing some successor or predecessor state while immediately forgetting about the original one. Again, the increased expressivity makes the until/since operators worthwhile, and fortunately, the computational costs paid are not as dramatically high as in the case of hybrid binders.

The global modality. It simply grants access to any point in the structure and can thus be seen as a generalisation of satisfaction operators. Similarly to the until/since operators, it adds expressive power to the language, which sometimes makes reasoning harder.

1.2 Towards a systematic study of the complexity of hybrid logics

This thesis systematically examines decidability and the computational complexity of decision problems for a collection of hybrid languages with respect to several classes of structures. More precisely speaking, we will establish, in the usual terms of computational complexity theory, what amount of resources (space, time) are necessary for an algorithm to decide each of these problems, and whether such an algorithm exists at all. We will focus on the satisfiability problem and provide results on the model-checking problem. These problems ask whether a given formula from a certain hybrid logic is satisfiable in some structure or a given structure, respectively. Decidability and the computational complexity of decision problems are of great interest whenever those shall be solved automatically, see [Wos85] for an introduction into automated reasoning.

Satisfiability for hybrid logic tends to have a high computational complexity in general, which is due to the increased expressive power of hybrid languages. For instance, satisfiability for hybrid logic is known to require exponential time [ABM00] in the presence of past or until operators, and to be even undecidable if a fairly restricted form of a binder is admitted [ABM99]. This is in contrast to modal logic, whose satisfiability problem is solvable in polynomial space [Lad77]. Furthermore, model checking for modal languages is solvable in polynomial time, but in the presence of binders, polynomial time most probably does not suffice because the model-checking problem is complete for polynomial space here [FdR06].

It is well-known that many applications for modal or hybrid logic do not require the full language or do not permit all possible frames. Hence, restricting the language and/or the class of relevant frames could be a way to "tame" a very expressive logic. And indeed, there is much literature where very different complexities for more or less expressive hybrid languages over different classes of frames have been established [ABM99, ABM00, FdRS03, tCF05b, FdR06, MSSW05, MS07b, MS07a]. There are combinations of hybrid languages and frame classes, for which the satisfiability problem, for instance, is known to be complete for the complexity classes NP, PSPACE, EXPTIME, NEXPTIME, N2EXPTIME; nonelementarily decidable; or even undecidable. However, we are not aware of any systematic study that involves several frame classes and, independently from those, a self-contained collection of hybrid languages.

Such a systematic study is pursued by this thesis and will show problems that have not been solved in the literature yet. We will fix a set of modal, temporal, and hybrid operators and consider a hierarchy of all hybrid languages defined by subsets of this set of operators. We will then arrange known results from the literature into this hierarchy, separately for several classes of frames. This will show that there are many combinations of languages and frame classes whose complexity is not known. We will provide results for most of them, applying a wide range of well-known techniques for establishing lower and upper complexity bounds in modal and hybrid logic.

We do not claim that either collection (of frame classes or languages) is complete, but, at least, our study covers all hybrid languages with the most commonly used operators and many temporally and epistemically relevant frame classes. Here, the notion of the "relevance of frame classes for applications" deserves a more precise explanation.

In view of temporal applications, it is apparent that only frames with "later-than" relations satisfying certain properties need be considered. Such properties include — but are not restricted to — transitivity, irreflexivity, or trichotomy. (The latter refers to the condition that given two distinct points, at least one is related to the other.) One of the most special frame class in this context is the class that consists of only one frame, namely the natural numbers with the greater-than relation. This class underlies the widely used and well-understood Linear Temporal Logic (see, e.g., [CGP01]) and represents a discrete view on time. It is possible to consider the integers or the reals instead of natural numbers [Rey92]. Furthermore, there are two generalisations of these singleton frame classes. One is the class of linear frames that merely requires the above three properties and contains frames with discrete as well as dense flows of time (among them, the natural numbers, integers, and reals). Another generalisation is the class of transitive trees that adds branching to the natural numbers and underlies the expressive Computation Tree Logic (which is described in [CGP01], too).

For epistemic applications, equivalence relations and weaker notions are necessary to model knowledge and belief of agents [FHMV95, Section 3.1]. If the states and accessibility relations in a frame are to represent possible worlds of agents and if the agents' knowledge or beliefs are assumed to satisfy certain soundness properties (in particular: only true things are known/believed, and the agent is aware of what she knows/believes and what she does not know/believe), then this is captured by equivalence relations. If some of the soundness properties are abandoned or weakened, then one has to use more general kinds of relations.

For both kinds of applications, transitivity plays a very important rôle. First, in all of the above examples of temporally relevant frame classes, the future relation is transitive (and has other properties as well). The class of transitive frames is a general case of all these temporal applications. Second, transitivity is similarly fundamental in epistemic applications because it corresponds to the property that agents are aware of their knowledge or their belief. As in the temporal case, other properties can — but need not — be added, but transitivity is rarely left out.

Modal, hybrid, and first-order logics over transitive models have been studied recently in [ABM00, GMV99, ST01, Kie02, Kie03, IRR$^+$04, DO05]. Although the complexity of satisfiability for hybrid (temporal) logics has been extensively examined [BS95, Gor96, ABM99, ABM00, FdRS03], there are highly expressive hybrid languages for whose satisfiability problems only results over arbitrary, but not over restricted, temporally or epistemically relevant frame classes have been known. This confirms the need for a classification of complexity for satisfiability of hybrid logic over such frame classes.

Furthermore, for the (general) model checking problem, only results over arbitrary frames have been known [FdR06]. We will find out whether the above mentioned level of complexity for binder languages persists if we restrict the class of frames. (The word "general" means that we will examine the model-checking problem considered in [FdR06], restricted to certain classes of frames, as opposed to the linear-time model-checking problem from [SC85] and [FdRS03].)

The frame classes that we will consider are the class of all frames, transitive frames, transitive trees, linear frames, the natural numbers, frames with equivalence relations, and complete frames.

1.3 The complexity of multi-modal hybrid logics

The classification of the satisfiability problem for hybrid languages over different frame classes will show that satisfiability for the language with the more restricted form of a hybrid binder, which is undecidable over arbitrary frames [ABM99], will become decidable over transitive frames [MSSW05]. We will not only show that satisfiability for languages combining this binder with other operators is undecidable over transitive frames. We will also examine another extension of this binder language over a wide range of frame classes, namely its multi-modal version. Our (undecidability) results will cover, among others, frame classes that are important for epistemic applications, because the multi-modal setting corresponds to multi-agent scenarios.

1.4 Legend to this thesis

This thesis is organised as follows. In Chapter 2, we will give all definitions and notations that are necessary for modal, temporal, hybrid, and first-order logic. We will also introduce the basic concepts of computational complexity and tools used to establish complexity bounds of certain logics. Chapter 3 is concerned with expressivity issues and establishes hierarchies of hybrid languages over different classes of frames. Chapters 4 and 5 examine the model-checking problem and satisfiability of hybrid languages over these frame classes. Satisfiability of multi-modal binder logic is considered in Chapter 6. Chapter 7 gives an overview of all achieved results and contains remarks on each group of results from the previous chapters.

Parts of this thesis have appeared in proceedings of workshops or in journals. In particular, Sections 5.4 and 5.5 contain results from [MSSW05] and [MSSW07], Section 5.8 has appeared as [MS07a], and Chapter 6 improves on [MS07b].

Chapter 2

Preliminaries

2.1 Hybrid logic

We define the basic concepts and notations of modal and hybrid logic that are relevant for this thesis. The fundamentals of modal logic are mainly taken from [BdRV05]; those of hybrid logic from [ABM99, Bla00b, AtC06].

2.1.1 Syntax

As indicated in the previous chapter, *the* hybrid language does not exist. Rather there are several extensions of the modal language that allow for explicit references to states and incorporate very restricted versions of first-order quantifiers — hence the attribute "hybrid". We will introduce the largest and most expressive hybrid language that will interest us in this thesis. It contains four temporal operators, two hybrid binders, satisfaction operators, and the global modality. Later on, we will define fragments of this full language.

We will give the syntax of hybrid logic inductively in the usual manner. For Boolean, modal, and hybrid operators that appear in duals, Definition 2.1 gives only the "existential" operators $\bot$, $\vee$, F, etc. in the induction and defines the remaining operators as abbreviations.

Definition 2.1 *Let* PROP *be a countable set of propositional atoms,* NOM *be a countable set of nominals,* SVAR *be a countable set of state variables, and let* ATOM = PROP ∪ NOM ∪ SVAR.

(1) The full hybrid language $\mathcal{HL}(\mathsf{F}, \mathsf{P}, \mathsf{U}, \mathsf{S}, \downarrow, \exists, @, \mathsf{E})$ *is the set of all formulae of the form*

$$\varphi ::= a \mid \bot \mid \neg\varphi \mid \varphi \vee \varphi' \mid \mathsf{F}\varphi \mid \mathsf{P}\varphi \mid \varphi\mathsf{U}\psi \mid \varphi\mathsf{S}\psi \mid {\downarrow} x.\varphi \mid \exists x.\varphi \mid @_t\varphi \mid \mathsf{E}\varphi\,,$$

where $a \in$ ATOM, $t \in$ NOM ∪ SVAR, *and* $x \in$ SVAR.

(2) *We use the following abbreviations.*

$$\top = \neg\bot \qquad \mathsf{G}\varphi = \neg\mathsf{F}\neg\varphi$$
$$\varphi \wedge \psi = \neg(\neg\varphi \vee \neg\psi) \qquad \mathsf{H}\varphi = \neg\mathsf{P}\neg\varphi$$
$$\varphi \rightarrow \psi = \neg\varphi \vee \psi \qquad \forall x.\varphi = \neg\exists x.\neg\varphi$$
$$\varphi \leftrightarrow \psi = (\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi) \qquad \mathsf{A}\varphi = \neg\mathsf{E}\neg\varphi$$

(3) *Let φ be a formula and x be a state variable.*

- *For any occurrence of the $\downarrow$ or $\exists$ operator in φ that begins a subformula $\downarrow x.\psi$ or $\exists x.\psi$ of φ, its* scope *is ψ.*
- *Any occurrence of x in φ is called* bound *iff it is within the scope of some occurrence of the $\downarrow$ or $\exists$ operator in φ.*
- *x is* free *in φ iff it does not occur bound in φ.*

(4) *A hybrid formula is called*

- pure *iff it contains no propositional atoms;*
- nominal-free *iff it contains no nominals; and*
- *a* sentence *iff it contains no free state variables.*

(5) *For each formula φ, we use $\mathrm{PROP}(\varphi)$, $\mathrm{NOM}(\varphi)$, and $\mathrm{SVAR}(\varphi)$ to denote, respectively, the set of all atomic propositions, nominals, and state variables that occur in φ.*

It is common practice to denote propositional atoms by $p, q, \ldots$; nominals by $i, j, \ldots$; and state variables by $x, y, \ldots$. The operators F, G, P, H, U, and S are called *temporal operators*, $\downarrow$, $\exists$, and $\forall$ are called *hybrid binders*, $@_t$ are *satisfaction operators*, and E and A are referred to as *global modalities*. The operators $\wedge$, G, H, $\forall$, and A are said to be the *duals* of $\vee$, F, P, $\exists$, and E, respectively.

2.1.2 Semantics

Semantics is defined in terms of *Kripke models*. In order to evaluate formulae with binders, an assignment from the set of all state variables to the set of states is necessary. This assignment can be omitted whenever binder-free sublanguages or only sentences are considered.

Definition 2.2

(1) *A* frame *is a pair $\mathcal{F} = (M, R)$ with the following components.*