Local variations on a loose theme: modal logic and decidability

Maarten Marx and Yde Venema

Institute for Logic, Language and Computation
University of Amsterdam
Plantage Muidergracht 24
1018 TV Amsterdam
The Netherlands.
`e-mail: {marx,yde}@wins.uva.nl`

Abstract. This chapter is about the satisfiability problem for modal logics and related formalisms. We discuss and explain the good computational behaviour of many modal systems. We show how finite and other simple structures are used in the proofs of various decidability and complexity results. In doing so, we introduce some of the most important proof methods that are used in establishing these results.
Concerning the first theme, we single out some principles of the basic modal logic that determine its computational properties. The most important of these is the *looseness principle* which says that any satisfiable modal formula is satisfiable on a loose model; but we will also discuss two *locality principles* which are two formalizations of the intuition that the truth of a modal formula at a state of a model only depends on a small, local part of the model. Concerning the second theme, we give detailed expositions of the selection of points method, the method of filtration and the mosaic method.
The chapter starts with a general introduction to modal logic, and then focuses on a detailed proof of the decidability of the basic modal system. After that, decidability is discussed for some variations of this basic modal system, and the picture is refined by taking complexity issues into account. Finally, the observation that modal logic is nothing but a nice, decidable fragment of first-order logic, is employed to generalize this to larger parts of first-order logic, such as the guarded and packed fragments.

To appear as a chapter in

Moshe Y. Vardi and Scott Weinstein (eds.), *Finite Model Theory and its Applications*, Springer-Verlag, forthcoming.

1 Introduction

This chapter is about decidability and complexity issues in modal logic; more in particular, we confine ourselves to *satisfiability* (and the complementary *validity*) problems. The satisfiability problem is the following: for a fixed class of models, determine whether a given formula φ is satisfiable in some model of that class (a more precise definition will follow). The general picture is that modal logic behaves quite well in this respect. In fact, many authors follow Vardi [57] in calling modal logic *robustly* decidable on the ground that most of the nice computational properties of modal logic are preserved if one considers extensions or variants of the basic system. The main aim of this chapter is to refine and analyze this picture.

To start with, we should clarify what we are talking about when using the term 'modal logic'. Traditionally, propositional modal logic would be described as an extension of propositional logic with operators $\Box$ and $\Diamond$ for talking about the necessity and possibility of a formula being true. However, nowadays the term 'modal logic' is used for a plethora of formalisms, with applications in various disciplines ranging from linguistics to economics, cf.[17, 38, 55, 11] for a sample of applications in computer science.

And while (propositional) modal logics will usually still be an extension of classical propositional logic with a number of modal operators, the intended meaning of these operators differs enormously. For instance, the formula $\Box_a \varphi$ could mean 'player a knows that φ is the case' in a formalization of game theory, or 'after the execution of program a, φ will be the case' in a formal language for program verification. Fortunately, on a technical level all these formalisms still have a lot in common. That is why this paper first introduces the notion of a *modal system* as a triple consisting of a (propositional) modal language, a class of models and a truth function. This definition covers most of the systems that go in the literature under the name 'modal logic'; in particular the familiar system of *basic modal logic*, to be discussed in section 3.

Now that we know what modal logic is, can we say what makes it so robustly decidable? If we confine ourselves to the *basic modal logic*, the answer seems to be affirmative. As we will see further on, the fact that the truth of basic modal formulas is invariant under *bisimulations* makes that basic modal logic has the *tree model property*. That is, every satisfiable modal formula is satisfiable in a special, 'loose', model based on a tree. This makes it much easier to check whether a given modal formula is satisfiable: one only needs to worry about these loose models. And since the bisimulation invariance property transfers to many extensions and variants of the basic modal system, so does decidability of the satisfiability problem.

This analysis, due to Vardi [57], in terms of a *looseness* principle, will form the main theme of our chapter. However, it can only form part of the story. For instance, suppose that we are interested not just in decidability, but also in the computational complexity of the satisfiability problem. Not all loose modal systems are in the same complexity class, so there must be principles besides looseness that determine the computational behavior of a modal system. Or, what if we happen to work with a modal system that does not allow trees as models? Will this necessarily make the logic undecidable? Answers to such questions cannot be precise and general at the same time — note that the problem whether a given modal axiom determines a decidable modal logic ('logic' here in the technical sense, see below), is itself undecidable! Nevertheless, we believe that it is possible to provide some rough guidelines, and in this chapter we will discuss some of these; in particular, at the end of section 3 we discuss two *locality principles*.

Thus it is our aim to act as the reader's travel guide in the landscape of modal logics by pointing out some interesting decidability and complexity theoretic phenomena and by suggesting an interpretation of these phenomena as local variations on a loose theme... On the trip we will introduce some important modal systems and proof methods — but we have not been aiming for a complete or systematic overview in this respect. (For instance, we will not employ any automata-theoretic methods.) No previous exposure of the reader to modal logic is assumed. Finally, in the running text we usually do not provide credits or give references; these are supplied in the 'Notes' paragraphs that finish each section.

Overview of paper In the next section we explain our interpretation of the terms 'modal logic' and 'modal systems', and we define the notion of bisimulation. Section 3 discusses basic modal logic, giving a detailed proof of the decidability of its satisfiability problem; analyzing this proof we introduce the notions looseness and locality. In the section after that, we use a number of examples to show what happens if we play around a bit with these principles. Section 5 is devoted to a more fine-grained, complexity theoretic study of the modal satisfiability problem. In the last section we show how one can use the principles discussed in the first parts of the paper to find large fragments of ordinary first order logic that have a decidable satisfiability problem.

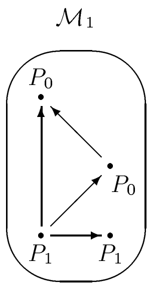
Acknowledgments We are very grateful to Carlos Areces, Edith Hemaspaandra and Carla Piazza for scrutinizing earlier versions of this manuscript and for making many suggestions for improvement. We would also like to thank Moshe Vardi and Scott Weinstein for inviting us to participate in this project.

2 Modal systems and bisimulations

In this section, we discuss our interpretation of the term 'modal logic' by defining and explaining the notion of a modal system. We also introduce the fundamental notion of similarity between two modal models, namely that of a *bisimulation*. The link between modal logic and bisimulation is that modal formulas cannot distinguish bisimilar points.

Modal systems

A modal system consists of a modal language $\mathcal{L}$, a class of models K and a function $\Vdash$ interpreting formulas of the language in the models. As *models* we will consider only relational structures, that is, structures consisting of a non-empty domain or universe together with a number of relations on it. With the *size* of a model $\mathcal{M}$, we always denote the size of its domain. The elements of the universe of a model will be called *states, points* or *worlds*. In the modal part of this paper we will confine ourselves to models having a number of *binary* relations (usually just one which we denote as R) and a countable number of unary relations P_0, P_1, ... See Figure 1 for a graphical presentation of these models.



A model $\mathcal{M} = (W, R, P_0, P_1)$ can be seen as a coloured directed graph; the edges give the relation R, the colours the interpretation of the unary predicates. (Note that points can have more than one colour). The graph part (W, R) of the model is also called a *modal frame.*

Fig. 1. Graphical representation of a model.

A modal *language* is a simple yet expressive language for talking about such relational structures. In this paper we only consider *propositional* modal languages; these can be described as extensions of the classical propositional language with a collection of modal connectives such as the unary modal operator $\Diamond$. Like the boolean connectives, the modal operators do not bind variables. The *size* of a formula φ (notation: $|\varphi|$) in a modal language $\mathcal{L}(\Phi)$ (i.e., with propositional variables from a set Φ) is its length over the alphabet $\Phi \cup \{\neg, \wedge, (,), \nabla_i\}_{i \in I}$, where $\{\nabla_i \mid i \in I\}$ denotes the set of modal connectives of $\mathcal{L}$.

Finally, $\Vdash$ is a function which takes a model $\mathcal{M}$ and a formula φ and returns a subset of the domain of $\mathcal{M}$ that we will think of as the *meaning* of φ in the model. The standard terminology for stating that $s \in \Vdash(\mathcal{M}, \varphi)$ is that φ *is true* or *holds* at s in $\mathcal{M}$, and the standard notation is:

$$\mathcal{M}, s \Vdash \varphi.$$

The meaning of the propositional variable p_i is the corresponding set P_i, thus

$$\mathcal{M}, s \Vdash p_i \iff P_i s.$$

For the Boolean connectives we have the standard interpretation in mind, which requires that $\neg\varphi$ is true at a state iff φ is false (i.e., not true) at it, that $\varphi \wedge \psi$ holds precisely at those states where both φ and ψ hold.

The conditions on modal systems mentioned so far still allow for an enormous freedom in defining the semantics of the modal connectives. But even in this very wide and general setting we can already introduce the notions of *satisfiability* and *validity* associated with such a triple: call a formula *valid* if it is true at every state of every model of the system, and *satisfiable* if it is true at some state of some model of the system. The *validity problem* of a modal system is the problem of deciding whether a given formula of the language is valid or not; the *satisfiability* problem is defined analogously. Given our constraints on the interpretation of Boolean negation, it is obvious that a formula ξ is valid in a class K of models if and only if its negation $\neg\xi$ is not satisfiable in any model in K. Hence, for any class of models K, there are constant time reductions between the satisfiability problem and the *complement* of the validity problem. We will use this fact in the sequel without explicit mentioning; also, we will use the term 'complexity of a modal system' when referring to its satisfiability problem.

Each of the three ingredients of a modal system — the language $\mathcal{L}$, the class K of models, and the interpretation function $\Vdash$ — influences the complexity of the satisfiability problem. We will see that many important and interesting modal systems have a decidable satisfiability problem, but the above definition of a modal system is also wide enough to allow for systems whose satisfiability problem is highly undecidable. Our aim in this paper is to provide some rough guidelines for determining the complexity of a modal system. For the sake of a simple exposition we first restrict our attention to a simple yet interesting type of models: those in the signature with one binary relation R and a number of unary relations. Even with this signature fixed we still have an enormous freedom in defining the modal language $\mathcal{L}$ and the meaning of the modal connectives given by the function $\Vdash$. What then, the reader will ask, is particularly *modal* about a system? We will now give a further restriction on modal systems which is very characteristic for modal logic. It concerns the discriminatory power of modal languages.

Bisimulation

An informative way to identify a language is by saying to which differences between models it is blind. In the case of modal logic, the fundamental concept of equivalence between structures involves the notion of *bisimulation*.

Definition 1. *Given two models $\mathcal{M} = (W, R, P_i)_{i\in I}$ and $\mathcal{M}' = (W', R', P'_i)_{i\in I}$, a non-empty relation $Z \subseteq W \times W'$ is a* bisimulation *between $\mathcal{M}$ and $\mathcal{M}'$ if the following three conditions hold, for all states $s \in W$ and $s' \in W'$ that are linked by Z:*

(base) *for all i: $P_i s$ iff $P'_i s'$,*
(forth) *for all $t \in W$ such that Rst there is a $t' \in W'$ such that $R's't'$ and tZt',*
(back) *for all $t' \in W'$ such that $R's't'$ there is a $t \in W$ such that Rst and tZt'.*

If there is some bisimulation Z linking s and s' then we say that s and s' are bisimilar, notation: $s \leftrightarroweq s'$, or $\mathcal{M}, s \leftrightarroweq \mathcal{M}', s'$ if we wish to make the models explicit.

Figure 2 contains two simple examples of bisimulating models (the models bisimulate horizontally) in a language with only one unary relation P. Figure 3 shows two models which do not bisimulate at the roots; all states in both models satisfy the same unary relations; $\mathcal{M}'$ has all of the finite branches that $\mathcal{M}$ has, but in addition it contains an infinite branch.

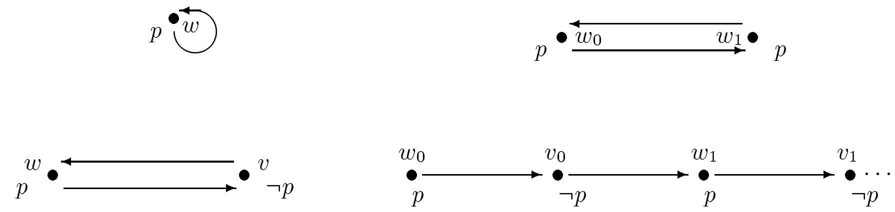


Fig. 2. Two examples of bisimulating models.

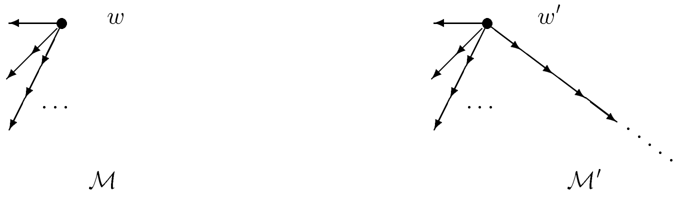


Fig. 3. $\mathcal{M}, w$ and $\mathcal{M}', w'$ are not bisimilar.

We can now make the crucial restriction on $\Vdash$ precise: we want the truth of modal formulas to be invariant for bisimulations. That is, our basic interest is in a truth definition for the modal connectives that makes the semantics satisfy the following constraint:

$$\text{if } \mathcal{M}, s \mathrel{\underline{\leftrightarrow}} \mathcal{M}', s', \text{ then for all } \varphi\text{: } \mathcal{M}, s \Vdash \varphi \text{ iff } \mathcal{M}', s' \Vdash \varphi. \tag{1}$$

Let us agree to call any system in which $\Vdash$ meets this constraint a *modal system in the narrow sense.*

But *why* would we be interested in a language that cannot see the difference between bisimilar states (apart from technical reasons)? Apart from the original logical considerations that we are about to describe, an important reason stems from theoretical computer science. Here, or more specifically, in the field of *process theory*, one models processes as so-called *labelled transition systems*; these are relational structures like the one we describe here — though usually with a *collection* of binary relations instead of just one. The idea here is that a state s in a model $\mathcal{M}$ represents some state of the process: the predicates P_i correspond to various direct *observations* that we can make about states; and the relations R_j correspond to the various *transition steps* that the process may take. A pair (s,t) belonging to the relation R_j indicates that at s the process can take an R_j-step, thus reaching the state t, where new direct observations can be made, or new steps can be taken. Now in this context, states that are bisimilar cannot be distinguished from a process-theoretic point of view and thus represent the *same* state. Thus, bisimulation serves as one of the most fundamental notions of *identity* between process states. This explains that languages designed for expressing properties of processes should indeed be blind for the distinction between bisimilar states.

Let us turn to some concrete examples. We first give some examples of operators which meet this requirement:

$$\begin{array}{ll}
\mathcal{M},s \Vdash \Diamond\varphi & \text{if } \mathcal{M},t \Vdash \varphi \text{ for some } t \text{ such that } Rst,\\
\mathcal{M},s \Vdash \langle * \rangle\varphi & \text{if there is some path } s = s_0Rs_1Rs_2\ldots Rs_n = t \text{ through } \mathcal{M}\\
& \quad \text{such that } \mathcal{M},t \Vdash \varphi \text{ (including the empty path)},\\
\mathcal{M},s \Vdash \overrightarrow{\infty}\,\varphi & \text{if there is some path } s = s_0Rs_1Rs_2\ldots \text{ through } \mathcal{M}\\
& \quad \text{such that } \mathcal{M},s_i \Vdash \varphi \text{ for infinitely many } i.
\end{array}$$

We will show later on that indeed, $\Diamond$ does not break the bisimulation invariance (see (2) below; the proofs for the other operators are left to the reader).

The operators $\langle\neq\rangle$, E, P and U defined below are not invariant for bisimulations. Figure 4 shows two models $\mathcal{M}_1, \mathcal{M}_2$ of signature R, P and worlds s_1, s_2 which bisimulate. The valuation of P is indicated in the models. Since both s_1 and s_2 have no successors, they bisimulate because they are both not in P. It is easy to see that $\langle\neq\rangle p$, $\mathsf{E}p$ and $\mathsf{P}p$ are all true in s_1 and all false at s_2. A counterexample for the bisimulation invariance of the binary connective U is provided just above Figure 7 on page 28.

$$\begin{array}{ll}
\mathcal{M},s \Vdash \langle\neq\rangle\varphi & \text{if } \mathcal{M},t \Vdash \varphi \text{ for some } t \text{ such that } s \neq t,\\
\mathcal{M},s \Vdash \mathsf{E}\varphi & \text{if } \mathcal{M},t \Vdash \varphi \text{ for some } t \text{ in } \mathcal{M},\\
\mathcal{M},s \Vdash \mathsf{P}\varphi & \text{if } \mathcal{M},t \Vdash \varphi \text{ for some } t \text{ such that } Rts,\\
\mathcal{M},s \Vdash U(\varphi,\psi) & \text{if } \mathcal{M},u \Vdash \varphi \text{ for some } u \text{ such that } Rsu,\\
& \quad \text{while } \mathcal{M},t \Vdash \psi \text{ for all } t \text{ satisfying } Rst \text{ and } Rtu.
\end{array}$$

It seems that $\Diamond$ is the simplest non-trivial operator that satisfies the bisimulation-invariance condition. The modal system in which we take $\Diamond$ as the only modal operator, in which we allow every relational structure (of the appropriate signature) as a model, and in which $\Vdash$ has the standard definition, is called the *basic modal logic*. It is discussed in detail in Section 3.

Some comments Before turning to this discussion of the basic modal system, a few comments are in place.

First, it was not our intention to give very rigid definitions. For instance, the question whether the universal diamond E constitutes a modal system in

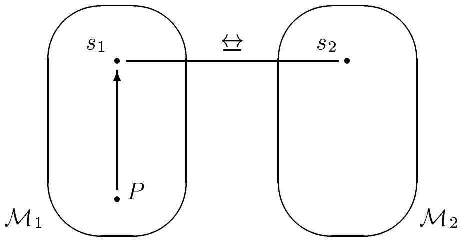


Fig. 4. Modal operators $\langle\neq\rangle$, E and P are not invariant for bisimulations.

the narrow sense, is really dependent on the perspective that one takes. Earlier on we said that E is not bisimulation invariant, but what if we take as our *class of models* precisely those in which R is the universal relation on the model (that is, $\forall xy\, Rxy$)? Then the truth definition for E *does* satisfy the standard, 'bisimulation-invariant' clause for the diamond E; the only thing is that now, our class of models is not closed under taking bisimilar models...

Second, in our introductory discussion we have avoided the word 'logic'. In principle we prefer to use this word in the technical sense only, referring to a *set of formulas* that satisfies certain closure properties. For instance, a *normal modal logic* should be closed under the familiar law of Modus Ponens, and under the rule of Necessitation; the latter means that $\Box\varphi$ belongs to the logic whenever φ does. With many modal systems we can associate such a logic; in particular, when the system's class of models is defined through some property of the binary relation(s) only, the collection of valid formulas will form a logic in the technical sense. Thus the validity problem can often be identified with a membership problem, namely, that of a formula in a logic. In a number of cases we will forget our principles and follow custom in referring to this associated logic instead of to the modal system.

Finally, the term 'modal system *in the narrow sense*' should not be taken too literally; by playing around with the class K of models, the reader will easily see that our definition covers a wide range of modal logics. In fact, if we allow languages with more than one modality, then even various versions of first order logic itself, such as the finite variable fragments, can be seen as modal systems in the narrow sense!

3 Basic Modal Logic

In this section, we will introduce the basic system of modal logic and discuss its close connection to the notion of *bisimulation.* We provide a fairly detailed proof of the decidability of the satisfiability problem for basic modal logic and analyse this result in terms of looseness and locality.

Definition 2. *Given a set Φ of propositional variables, the collection $\mathcal{L}_\Diamond(\Phi)$ of basic modal formulas in Φ is given by the following rule*

$$\varphi \ ::= \ p \mid \neg\varphi \mid \varphi \wedge \psi \mid \Diamond\varphi,$$

where p ranges over elements of Φ. This means that φ is either an atomic formula consisting of a proposition letter in Φ, or a more complex formula obtained from simpler ones by applying one of the connectives $\neg$, $\wedge$ or $\Diamond$. We will use the standard Boolean abbreviations and also the modal 'box' operator $\Box$, $\Box\varphi$ abbreviating $\neg\Diamond\neg\varphi$.

If the collection Φ of proposition letters is either irrelevant or clear from context, we frequently omit it, writing $\mathcal{L}_\Diamond$ instead of $\mathcal{L}_\Diamond(\Phi)$. In order to interpret the formulas of this language in a model of the kind discussed above, we represent such a model as a triple $\mathcal{M} = (W, R, V)$ where W is a non empty set of states, R is the binary relation of the model and $V : \Phi \to \mathcal{P}(W)$ is a *valuation* mapping proposition letters to subsets of W. Let M denote the class of all such models.

Definition 3. *Given a model $\mathcal{M} = (W, R, V)$, a state $s \in W$, and a formula φ, we define the notion of φ being true at s, denoted as $\mathcal{M}, s \Vdash \varphi$, recursively as follows:*

$$\begin{array}{ll} \mathcal{M}, s \Vdash p & \textit{if } s \in V(p), \\ \mathcal{M}, s \Vdash \neg\varphi & \textit{if } \text{not } \mathcal{M}, s \Vdash \varphi, \\ \mathcal{M}, s \Vdash \varphi \wedge \psi & \textit{if } \mathcal{M}, s \Vdash \varphi \textit{ and } \mathcal{M}, s \Vdash \psi, \\ \mathcal{M}, s \Vdash \Diamond\varphi & \textit{if } \mathcal{M}, t \Vdash \varphi \textit{ for some } t \textit{ such that } Rst. \end{array}$$

If φ is true at every state of the model we say that φ holds throughout *$\mathcal{M}$, notation: $\mathcal{M} \Vdash \varphi$; if φ holds at some state in $\mathcal{M}$, we say that φ is satisfiable in $\mathcal{M}$.*

The language of the basic modal system *is $\mathcal{L}_\Diamond$, its class of models is M and $\Vdash$ is as in the last definition. Depending on the context we let $\boldsymbol{K}$ denote either the basic modal system itself or its logic, that is, the set of valid formulas in this system.*

As the reader can easily verify, it holds that

$$\mathcal{M}, s \Vdash \Box\varphi \text{ if } \mathcal{M}, t \Vdash \varphi \text{ for all } t \text{ such that } Rst.$$

The first thing we should check is whether we met our design criterion (1) with this definition of $\Vdash$. Suppose that $\mathcal{M}$ and $\mathcal{M}'$ are two modal models, and suppose that Z is a bisimulation between $\mathcal{M}$ and $\mathcal{M}'$. We will prove by a formula induction that every basic modal formula φ satisfies the following:

$$\text{for all } s \in W \text{ and } s' \in W'\text{: } sZs' \text{ implies that } \mathcal{M}, s \Vdash \varphi \text{ iff } \mathcal{M}', s' \Vdash \varphi. \quad (2)$$

We leave the base step and the boolean cases of the inductive step as exercises for the reader, and concentrate on the modal case of the inductive step. Suppose that φ is of the form $\Diamond\psi$, and assume that Z links the state s in $\mathcal{M}$ to s' in $\mathcal{M}'$. For reasons of symmetry it suffices to show that $\mathcal{M}, s \Vdash \varphi$ only if $\mathcal{M}', s' \Vdash \varphi$.

Suppose that $\mathcal{M}, s \Vdash \Diamond\psi$. By the truth definition it follows that there is a state t in W such that Rst and $\mathcal{M}, t \Vdash \psi$. From the fact that s and s' are linked by the bisimulation Z we may infer that there is some R-successor t' of s' such that $s'Zt'$. The inductive hypothesis gives us that $\mathcal{M}', t' \Vdash \psi$; but then we may conclude from $R's't'$ that $\mathcal{M}', s' \Vdash \Diamond\psi$, which is precisely what we were after. This proves (2) and shows that basic modal logic indeed constitutes a bisimulation invariant system.

Invariance for bisimulation So our definition fulfills our design criterion, but how powerful is this modal language precisely? In other words, how much of the bisimulation invariant properties can we express in this language? It should be obvious from the truth definition of basic modal logic that we can consider $\mathcal{L}_\diamond$ as a fragment of first order logic. In fact we have a kind of functional completeness result as long as we consider first order properties: *every first order definable bisimulation invariant property is definable by a modal formula.* In other words, when it comes to expressing bisimulation invariant properties, modal logic is *just as strong* as first order logic. In order to state this result formally we need a translation from modal to first order formulas.

Definition 4. *Assume that we have an enumeration $x = x_0, x_1, \ldots$ of first order variables. Consider the following translation of $\mathcal{L}_\diamond$-formulas to first order formulas:*

$$\begin{aligned} ST_{x_i}(p) &= Px_i \\ ST_{x_i}(\neg\varphi) &= \neg ST_{x_i}(\varphi) \\ ST_{x_i}(\varphi \wedge \psi) &= ST_{x_i}(\varphi) \wedge ST_{x_i}(\psi) \\ ST_{x_i}(\diamond\psi) &= \exists x_{i+1}(Rx_ix_{i+1} \wedge ST_{x_{i+1}}(\varphi)). \end{aligned}$$

When we speak of 'the' standard translation of a modal formula φ, we usually refer to the formula $ST_{x_0}(\varphi)$.

We can now see modal logic as a fragment of first order logic, because every modal formula is *equivalent* to its standard translation. Formally, (but blurring the distinction between a modal and a first order model a bit), we can prove that for every modal formula φ, for every model $\mathcal{M}$ and for every state s in $\mathcal{M}$ we have the following equivalence:

$$\mathcal{M}, s \Vdash \varphi \iff \mathcal{M} \models ST_{x_0}(\varphi)[x_0 \mapsto s]. \tag{3}$$

Here $[x_0 \mapsto s]$ denotes any assignment which sends x_0 to s. The simple proof of (3) is left to the reader.

Observation 1 The map in Definition 4 has no upper bound on the number of variables used in the first order translation of a modal formula. However, one could be very parsimonious and define the formulas $ST_x(\varphi)$ and $ST_y(\varphi)$ through a mutual recursion of which the interesting clauses run as follows:

$$\begin{aligned} ST_x(\diamond\psi) &= \exists y(Rxy \wedge ST_y(\varphi)) \\ ST_y(\diamond\psi) &= \exists x(Ryx \wedge ST_x(\varphi)). \end{aligned}$$

This shows that in fact, the translation of modal logic to first order logic can be carried out within the *two variable fragment* of first order logic. We will come back to this observation later on.

Now we are ready to state the celebrated characterization theorem for modal logic.

Theorem 1. *Let $\varphi(x)$ be a first order formula in the signature with a binary R and a set $\{P_i \mid i \in I\}$ of unary predicates. Then $\varphi(x)$ is invariant for bisimulations if and only if it is equivalent to the standard translation of a modal formula.*

The proof of the functional completeness part of the theorem (the left to right direction) falls outside the scope of this book but can be found in any good textbook on modal logic, see the notes. The other direction of the theorem, which just states that the modal language obeys the design criterium (1), is the more important one for us here. One way to look at (1) is that once we know that a formula is satisfiable at *some* state in *some* model, we know by the invariance result that it is also satisfiable in *any bisimilar* state in *any bisimilar* model. This means that we can *transform* the original model into one that suits our purposes best. Obviously, this method applies to any notion of invariance of any language. The nice thing about bisimulation however is that it allows the freedom of completely *unravelling* a model into a tree model.

Definition 5. *Given a model $\mathcal{M} = (W, R, V)$ and a state s_1 in $\mathcal{M}$, we define the* unravelling *or* unwinding *of $\mathcal{M}$ around s_1 as the following model $\mathcal{M}^u_{s_1} = (\vec{W}_{s_1}, \vec{R}, \vec{V})$. Its universe $\vec{W}_{s_1}$ is defined as the set of all finite paths through $\mathcal{M}$ starting at s_1; formally, $\vec{W}_{s_1}$ is the collection of all tuples $\langle s_1, \ldots, s_n\rangle$ (with $n \geq 1$) that satisfy Rs_is_{i+1} for all $i < n$. The tuples $\overline{s} = \langle s_1, \ldots, s_n\rangle$ and $\overline{t} = \langle t_1, \ldots, t_m\rangle$ stand in the relation $\vec{R}$ if and only if $\overline{t}$ is obtained from $\overline{s}$ by adding an R-successor of s_n. Formally, we put $\vec{R}\,\overline{s}\overline{t}$ if $m = n+1$ and $s_i = t_i$ for all $1 \leq i \leq n$. Finally, the truth of a proposition letter at a tuple is completely determined by its truth in $\mathcal{M}$ at the last element of the tuple. Formally, let $last(\langle s_1, \ldots, s_n\rangle)$ denote the state s_n and define $\vec{V}$ by $\vec{V}(p) = \{\overline{s} \in \vec{W}_{s_1} \mid last(\overline{s}) \in V(p)\}$.*

An example of an unravelling is given in Figure 5. Another example can be found in Figure 2 in which the model on the lower right hand side is (an isomorphic copy of) the unravelling of the model on the lower left hand side.

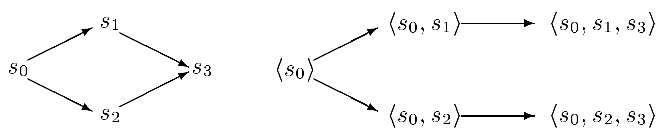


Fig. 5. Example of an unraveling.

The operation of unravelling is also well-known from process theory: the points of the unravelled model $\mathcal{M}^u_{s_1}$ can be viewed as the process *histories* or *traces* that start at s. From a *technical* perspective, the unravelling of $\mathcal{M}$ around s_1 has certain desirable properties:

– there exists a point with no predecessor, the *root*,
– $\vec{R}$ is acyclic, and
– $\vec{R}$ is injective in the sense that every point except the root has a unique predecessor.

In other words, the graph $(\vec{W}_{s_1}, \vec{R})$ is a *tree*. It is useful to look at this tree as being a *normal form* of the model.

One may easily check that for any model $\mathcal{M}$ and any state s in $\mathcal{M}$, the (graph of the) function $last : \vec{W}_s \rightarrow W$ constitutes a bisimulation between $\mathcal{M}_s^u$ and $\mathcal{M}$ that links $\langle s \rangle$ to s. But from this, it immediately follows that $\langle s \rangle$ (in $\mathcal{M}_s^u$) satisfies exactly the same formulas as s (in $\mathcal{M}$). Combining this with the observation that unravellings are trees, we find that every satisfiable basic modal formula is also satisfiable in a tree. Let us say that a modal system $(\mathcal{L}, \mathsf{K}, \Vdash)$ has the *tree model property* if for every satisfiable formula ξ in $\mathcal{L}$ there exists a tree in K in which ξ is satisfiable. Thus we have established the following.

Theorem 2. *The basic modal system has the tree model property.*

The following observation will be a driving force behind our search for decidable modal fragments of first order logic.

Observation 2 In proving the last Theorem we did not use any property of basic modal logic other than its invariance for bisimulation. This means that in fact, any modal system with a bisimulation invariant semantics has the tree model property — provided, of course, that tree models are admissible in the system!

Games In order to facilitate a comparison with the Ehrenfeucht–Fraïssé games often used in first order logic it is convenient to rephrase the notion of a bisimulation between two models in game–theoretic terms. Let $\mathcal{M} = (W, R, V)$ and $\mathcal{M}' = (W', R', V')$ be two models and let s_0 and s_0' be two states in $\mathcal{M}$ and $\mathcal{M}'$, respectively.

We define the bisimulation game as a variant of the familiar Ehrenfeucht–Fraïssé games. In each round of the bisimulation game, $\forall$ selects one of the two models and, inside this model, he chooses a successor of the element played in the previous round (in the first round a successor of s_0 or s_0'). $\exists$ responds with a successor of the last played element in the other structure. The *length* l of the game is defined as the number of rounds and can either be finite or infinite. A *match* of the game thus gives rise to two l sequences $\overline{s} = s_0, s_1, s_2, \cdots$ and $\overline{s}' = s_0', s_1', s_2', \cdots$ of elements in $\mathcal{M}$ and $\mathcal{M}'$, respectively.

$\exists$ *wins* this match of the game if for each i, s_i and s_i' agree on the truth of all propositional variables; otherwise, $\forall$ wins. We say that $\exists$ has a *winning strategy* in the game of l rounds played in $\mathcal{M}, s$, $\mathcal{M}', s'$ (notation: $\mathcal{G}_b^l(\mathcal{M}, \mathcal{M}', s, s')$) if $\exists$ can win *every* match of length at most l starting in the states s and s'.

It should be fairly clear that $\forall$ is trying to spoil a bisimulation between $\mathcal{M}, s$ and $\mathcal{M}', s'$, while $\exists$ has the opposite intention. Indeed we have a precise game-theoretic characterization of the notion of bisimulation.

Proposition 1. *There exists a bisimulation between $\mathcal{M}, s$ and $\mathcal{M}', s'$ if and only if $\exists$ has a winning strategy in the game $\mathcal{G}_b^\omega(\mathcal{M}, \mathcal{M}', s, s')$.*

As a corollary, we find that the truth of modal formulas is preserved when $\exists$ has a winning strategy in the game $\mathcal{G}_b^\omega(\mathcal{M}, \mathcal{M}', s, s')$. What is interesting about these games is that they facilitate a more fine-grained perspective on such connections. In particular, if $\exists$ has a winning strategy in a game of fixed finite length n, what can we say about the preservation of modal formulas? Quite a lot, according to the proposition below; in order to formulate it, we need the notion of the *modal depth* of a formula — the straightforward analog of first order logic's quantifier depth.

Definition 6. *The modal depth $d_\Diamond(\varphi)$ of a modal formula is inductively defined as follows.*

$$\begin{aligned} d_\Diamond(p) &= 0 \\ d_\Diamond(\neg\psi) &= d_\Diamond(\psi) \\ d_\Diamond(\psi_1 \wedge \psi_2) &= max(d_\Diamond(\psi_1), d_\Diamond(\psi_2)) \\ d_\Diamond(\Diamond\psi) &= 1 + d_\Diamond(\psi). \end{aligned}$$

Proposition 2. *Let Φ be a* finite *set of proposition letters, and let n be some natural number. Then the following are equivalent:*

1. *$\exists$ has a winning strategy in the game $\mathcal{G}_b^n(\mathcal{M}, \mathcal{M}', s, s')$*
2. *$\mathcal{M}, s$ and $\mathcal{M}', s'$ satisfy the same $\mathcal{L}_\Diamond(\Phi)$ formulas up to modal depth n.*

We omit the fairly standard proof of this Proposition — the notes contain references. The analogous Proposition relating $\mathcal{G}_b^\omega(\mathcal{M}, \mathcal{M}', s, s')$ and the full modal language does not hold, as is witnessed by the models in Figure 3.

What have we gained from this slightly more fine-grained analysis of modal logic? We already saw that any satisfiable modal formula ξ can also be satisfied at the root s of a tree model $\mathcal{M}$: take for instance some unraveling of the original model. From Proposition 2 we can conclude that such a ξ can also be satisfied in a *bounded depth* tree model $\mathcal{M}'$: simply take the tree model $\mathcal{M}$ and throw away all states that are further than d steps away from the root s (where d is the modal depth of ξ). It is obvious that $\exists$ has a winning strategy in the bisimulation game $\mathcal{G}_b^d(\mathcal{M}, \mathcal{M}', s, s)$. From this it follows that ξ holds at the root s of the bounded depth tree $\mathcal{M}'$.

Finite trees We will now show that the satisfiability problem for the basic modal language (with respect to the class of all models) is indeed *decidable*. We will do so by establishing the *bounded model property*. A modal language is said to have this property with respect to a class K of models if every formula ξ that is satisfiable in some model in K can in fact be satisfied in a *finite* model in K of *bounded* size (that is, the size of the model is bounded by some computable function on $|\xi|$).

Now, in order to show that the language of basic model logic has this bounded model property, let ξ be an arbitrary satisfiable formula. We have already seen that we may assume that ξ is true at the root of a tree model $\mathcal{M}$ of depth not exceeding the modal depth of ξ. This tree model might still be infinite because of infinite branching, but now we recursively prune the tree as follows. Starting at the root, choose, for every subformula of ξ of the form $\Diamond\varphi$, a successor of the root at which φ is true (if such a successor exists at all). Obviously, at most b successors can be chosen, where b is the number of diamond subformulas of ξ. Hence, by deleting all not chosen successors and their descendants from the model, we obtain a tree model whose branching degree at the root is at most b. A simple verification shows that ξ still holds at the root. Now repeat this process at each of the chosen successors of the root and continue until the leaves of the tree are reached. Obviously ξ is still satisfied at the root.

Thus we have proved the following.

Proposition 3. *Any satisfiable modal formula ξ can be satisfied at a the root of a finite tree model, of which the depth is bounded by the modal depth of ξ and the branching degree is bounded by the number of diamond subformulas of ξ.*

The decidability of basic modal logic is a straightforward corollary of this. (For instance, use the standard translation, equivalence (3), and the fact that first order model checking is decidable.)

Theorem 3. *The basic modal system* **K** *has a decidable satisfiability problem.*

We will now describe a decision procedure which on input ξ systematically tries to build the tree model for ξ described in Proposition 3. The procedure is based on a simple but powerful idea: it tries to build a model (W, R, V) in which

1. the states are finite sets Δ of 'relevant' formulas;
2. we have the following 'truth = membership' principle:

$$\varphi \in \Delta \iff \mathcal{M}, \Delta \Vdash \varphi, \tag{4}$$

for all relevant formulas φ and for all states Δ in the model.

Suppose we can build such a model for an input formula ξ. Then by the *truth lemma* (4), ξ is satisfiable if it belongs to some state Δ in our model. Our implementation of the decision procedure is purely based on the proof of the truth lemma; hence, in order to motivate the procedure, we now indicate how to arrive at this proof. Fix a formula ξ.

Let us first confine the collection of relevant formulas. This set will change from state to state, but basically, all relevant formulas will be subformulas of ξ; we need a bit extra though. Given a formula φ, let ${\sim}\varphi$ denote the formula ψ if φ is of the form $\neg\psi$; otherwise, ${\sim}\varphi$ is the formula $\neg\varphi$; we say that a set Σ of formulas is closed under taking single negations if ${\sim}\varphi \in \Sigma$ whenever $\varphi \in \Sigma$. This notion enables us to pretend that a finite set is closed under taking negations by treating ${\sim}\varphi$ as if it were the real negation of φ. Now given a set of formulas

Σ, let $Cl(\Sigma)$ be the smallest set of formulas that extends Σ and is closed under taking subformulas and single negations. When ξ is a formula, we denote the set $Cl(\{\xi\})$ of relevant ξ formulas as $Cl(\xi)$; it is easy to see that the cardinality of $Cl(\xi)$ is linear in the length of ξ.

Each state of our model $\mathcal{M} = (W, R, V)$ will be a subset of $Cl(\xi)$, but rather than precisely define the universe W now, we assume here that we *have* defined it and gather sufficient requirements on this definition that enable a proof of (4). First we consider the valuation: the truth lemma prescribes a unique way to define V, at least for the proposition letters occurring in ξ: $V(p) = \{\Delta \mid p \in \Delta\}$. For a definition of the relation R we will be rather opportunistic. Again we use the truth lemma as our guideline: it shows that if $R(\Delta, \Delta')$ is to hold, then we should avoid the existence of a relevant formula $\Diamond\varphi$ such that $\varphi \in \Delta'$ but $\Diamond\varphi \notin \Delta$. Now R is defined by turning this requirement into a definition: we put $R(\Delta, \Delta')$ precisely when the above situation does not occur.

Let us now see which requirements we have to impose on the set W; that is, suppose that we want to give an inductive proof of the 'truth = membership' principle. Assume that with each state Δ we have associated a collection $\Sigma \supseteq \Delta$ of relevant formulas.

Obviously, the atomic case of the truth lemma holds by definition of V. For the inductive boolean cases to go through, it is sufficient to require that Δ is maximal with respect to being a propositionally consistent subset of Σ. That is, Δ and Σ have to satisfy the following condition *Prop–Max*(Δ, Σ):
– (for each $\sim\varphi \in \Sigma$): $\sim\varphi \in \Delta \iff \varphi \notin \Delta$, and
– (for each $\varphi \wedge \psi \in \Sigma$): $\varphi \wedge \psi \in \Delta \iff \varphi \in \Delta$ and $\psi \in \Delta$.

The inductive modal case imposes two further constraints, one for each direction of the truth lemma. The first one we have met already when we defined our relation R, but given that *Prop–Max*(Δ, Σ) holds, we can reformulate this condition as follows:

– if $R(\Delta, \Delta')$, then for all $\Diamond\psi \in \Sigma$: $\sim\Diamond\psi \in \Delta$ implies $\sim\psi \in \Delta'$.

This formulation clearly brings about the conditions that *each* successor of Δ should satisfy. The other direction of the truth lemma for the case $\varphi = \Diamond\psi$ presents an *existential* requirement:

– If $\Diamond\psi \in \Delta$, then there has to be a Δ' such that $\psi \in \Delta'$ and $R(\Delta, \Delta')$.

Observe that in this last existential requirement we encounter the branch cutting argument that we saw earlier on. Then we only kept successor states if there was a reason in the form of a $\Diamond\psi$ formula, now we only create a successor if we need it as a witness for such a formula. The search for suitable successors is the driving force of our algorithm.

But what about these associated sets of relevant formulas? Will every formula in $Cl(\xi)$ be relevant throughout the procedure? No, and this is precisely what will bound the recursion depth of the algorithm: the set of relevant formulas will decrease as we move away from the root of the model — reminiscent of the bounded depth of the tree model from Proposition 3. In particular, the set of relevant formulas for a state Δ which is m steps away from the root will consist of all formulas from $Cl(\xi)$ of modal depth at most $d_\diamond(\xi) - m$.

The algorithm presented in Figure 6 implements this search for a tree model. We claim that for sets of formulas Δ and Σ such that Σ is closed under taking subformulas and single negations, $K\text{–}World(\Delta, \Sigma)$ will be true iff there exists a tree model $\mathcal{M}$ such that at the root s, for all $\psi \in \Sigma$, $(\mathcal{M}, s \Vdash \psi \iff \psi \in \Delta)$. This function can be used to solve satisfiability for the basic modal system, since ξ is satisfiable iff there exists a set $\Delta \subseteq Cl(\xi)$ such that $\xi \in \Delta$ and $K\text{–}World(\Delta, Cl(\xi))$ is true.

Note that with each recursive call of *K-World*, the size of the set Σ decreases, since we include formulas of smaller modal depth only. Thus the recursion depth is bounded by the modal depth of the input formula ξ. That the function is correct can be proved by induction on the size of Σ; we leave this to the reader. By an appeal to Savitch's Theorem (PSPACE=NPSPACE) it is not hard to see that the procedure runs in PSPACE. We come back to this aspect in Section 5.

Assume that Δ and Σ are finite sets of formulas such that $\Delta \subseteq \Sigma$ and Σ is closed under taking subformulas and single negations.

$K\text{–}World(\Delta, \Sigma)$ if and only if

- $Prop\text{–}Max(\Delta, \Sigma)$, and
- for each formula $\diamond\psi \in \Delta$ there is a set $\Delta_\psi \subseteq \Sigma$ such that
  - $\psi \in \Delta_\psi$,
  - $(\forall \diamond\varphi \in \Sigma) : {\sim}\diamond\varphi \in \Delta \Rightarrow {\sim}\varphi \in \Delta_\psi$, and
  - $K\text{–}World(\Delta_\psi, Cl(\{\varphi \mid \diamond\varphi \in \Sigma\}))$.

Fig. 6. The function *K-World* decides **K** satisfiability.

This finishes the proof of Theorem 3; in the remainder of this chapter we will analyze this proof and see how much of it can be used for other (modal) logics.

Looseness and locality Analyzing this decidability proof we can distinguish a number of relevant properties of the basic modal system. First of all, bismulation invariance makes that in order to check the satisfiability of a modal formula, we only have to worry about 'loose' tree models. Let us agree to call this the *looseness* principle of basic modal logic; this property has recently gained status as either the single or at least the crucial property making modal logic so robustly decidable.

However, we believe that looseness is not all there is to say in the way of explaining the decidability (or low complexity) of the basic modal system **K**. The semantics of the basic modal language show that modal formulas only have a limited access to the model — this is what we dub the *locality principle* of modal logic, and one can make this rather vague notion precise in (at least) two ways.

First, in the above proof we used the fact that the effect of a modal formula is bounded by its modal depth. In particular, when working in a tree model we can prune the relevant neighborhood of a state even further by the method of selecting witnesses for $\Diamond$-subformula. All in all we find that in order to check whether a modal formula holds at a given state of some tree model, one only has to worry about a bounded, 'local' part of the tree model. In particular, what the basic modal language does *not* have is *global expressive power*. We say that a modal system $(\mathcal{L}, \mathsf{K}, \Vdash)$ has global expressive power if it can define the universal diamond E; that is, if there is a formula $\varphi(p)$ such that for every model $\mathcal{M}$ in K and every state s in $\mathcal{M}$, we have

$$\mathcal{M}, s \Vdash \varphi(p) \iff \mathcal{M}, t \Vdash p \text{ for some } t \text{ in } \mathcal{M}.$$

Let us agree that with the *first locality* principle we mean the *lack* of global expressive power. Later on, we will see that adding even the tiniest bit of global expressive power to the basic modal system, we destroy its *finite* tree property and lift the complexity of the satisfiability problem from PSPACE to EXPTIME.

The second locality principle we met already in Observation 1. From the fact that the basic modal language belongs to the two variable fragment FO_2 of first order logic, we may conclude that the satisfiability problem of the basic modal system can be reduced to that of FO_2. But for every modal language in which the connectives have a first order truth definition we can come up with a 'standard translation', so if the language has only finitely many connectives, this standard translation remains within a fixed finite variable fragment. Hence, if we consider a modal system $(\mathcal{L}, \mathsf{K}, \Vdash)$ in which $\mathcal{L}$ has only finitely many connectives and in addition, the class of models K allows a definition in some finite variable fragment, then the satisfiability problem for the modal system can be reduced to that of some finite variable fragment FO_k. Why are we so much interested in these fixed finite variable fragments? As we will see later, one reason is that they have tractable *model checking* problems, while the full first order language does not. To be concrete: given a finite first order model $\mathcal{M}$ and a first order sentence ξ, the problem whether $\mathcal{M} \models \xi$ is decidable in PTIME in $\mathcal{M}$ *and* in ξ, if ξ is from a fixed variable fragment, while it is in PSPACE if ξ is an arbitrary first order sentence. Let us then agree that with the *second locality* principle we mean this reducibility of the satisfiability problem to the satisfiability of some fixed finite variable fragment of first order logic (or perhaps of some higher-order formalism, as in the case of $\mathbf{K}^*$ that we will discuss later on).

It will be useful later to state what we mean with looseness and locality in terms of the bisimulation game. Since it is $\forall$ who tries to spoil a bisimulation, the strength of the bisimulation relation is determined by the moves $\forall$ is allowed to make. Indeed, $\forall$ cannot do too much. First observe that although a match is made up by long sequences of pairs of states, after every round it is only the last pair which is important. Given such a pair, $\forall$ is allowed to choose a new element, but only if it is a successor of a state in the pair. $\exists$ replies and then the players check whether the game stops because $\forall$ wins or not. If not, the previous pair "is deleted from memory" and the game continues.

We could view the game as being played by moving two windows across the models. These windows are covering the complete model except for at most two states. Both players move the windows across the models, and $\forall$ has the initiative. Now the looseness aspect is that the states which are visible through the window are always connected by the accessibility relation (this shows that instead of 'looseness' we could have dubbed this a 'locality' principle as well). The second locality principle is embodied in the fixed finite dimension of the window. (For a description of the first locality principle, games do not seem to be the most optimal way.)

Summarizing then, it seems that these looseness and locality principles in tandem cause the decidability of the basic modal system: looseness means that one only has to check trees, and the first locality principle adds that in fact, finite trees suffice. (For the contribution of the second locality principle, the reader will have to wait until we discuss the generalization of the modal language to the guarded and packed fragments of first order logic in section 6.) There can be no doubt that looseness is the most important property for a modal system's decidability; in fact, confining ourselves to the class M of all modal models, it will be hard to *find* a bisimulation invariant system with an undecidable satisfiability problem! The reason for this is that the modal mu-calculus is decidable, and this modal system is characterized as the bisimulation invariant fragment of monadic second order logic over a signature of binary relations.

3.1 Notes

Recent years have seen a proliferation of modern text books on modal logics, of which we mention Chagrov & Zakharyaschev [14], Popkorn [48] and Blackburn, de Rijke & Venema [9].

The standard translation, in various forms, can be found in the work of a number of writers on modal and tense logic in the 1960s. Van Benthem [4] first made clear the importance of systematic use of the standard translation to access results and techniques from classical modal theory. The observation that at most two variables are needed to translate basic modal formulas into first order logic is due to Gabbay [20]. The earliest systematic study of finite variable fragments seems to be due to Henkin [26] in the setting of algebraic logic, while Immerman & Kozen [33] study the link with complexity and database theory. Consult Otto [46] for more on finite variable logics, or Marx & Venema [42] for a modal perspective on these logics.

Bisimulations were first introduced (under a different name) by van Benthem [4, 5]. The notion was independently introduced in computer science, as an equivalence relation on process graphs; the first reference seems to be Park [47], whereas the classic computer science paper on the subject is Hennessy & Milner [28]; the latter paper also discusses the finitary approximations to bisimulations. The notion of unravelling a modal model stems from Dummett & Lemon [16]. Proposition 2 is analogous to similar characterizaions of logical equivalence for first–order logic, due to Ehrenfeucht and Fraïssé (cf. [31]).

The Characterization Theorem 1 of modal logic as the bisimulation invariant fragment of first order logic, is due to van Benthem [4, 6]. The back-and-forth clauses of a bisimulation can be adapted to analyze the expressivity of a wide range of modal logics, and such analyses are now commonplace. For instance, Janin & Walukiewicz [34] proved that Kozen's modal mu-calculs is the bisimulation invariant fragment of a natural monadic second order logic over process graphs. Related model-theoretic characterizations can be found in Immerman & Kozen [33] (for finite variable logics). Rosen [50] presents a version of the Characterization Theorem that also works for the case of finite models.

Finite models have long been used to establish decidability, both in modal logic and elsewhere. Arguments based on *finite* axiomatizability together with the finite model property trace back to Harrop [25]. The computational complexity of the satisfiability problem for the basic modal system was established by Ladner [36]: it is PSPACE–complete. The function $K-World$ is a slight variation of Ladner's procedure. The presentation given here is taken from Spaan [54].

Given a finite first order model $\mathcal{M}$ and a first order sentence ξ, the problem whether $\mathcal{M} \models \xi$ is PTIME–complete when ξ is from a fixed variable fragment (Immerman [32], Vardi [56]), and PSPACE–complete when ξ is an arbitrary first order sentence (Chandra & Merlin [15]).

4 Some variations

In this section, we will consider some modal systems that are variations on the basic modal system. Apart from our wish to introduce some new proof techniques for establishing decidability of a modal system, such as the *filtration* and *mosaic* methods, our aim in this section is to clarify the looseness and locality principles that we just introduced.

We will first investigate some modal systems that are fairly 'tight' in the sense that their class of models is based one grid-like structures; as we will see, such lack of looseness brings these systems close to the danger zone of undecidability. Nevertheless, if the locality principles still hold, decidability is still possible. The second system that we consider is obtained by adding just a grain of global expressive power to the basic modal language, while keeping the looseness condition. We will see that the system is still decidable, but it does no longer have the finite tree property. Finally, we consider a modal system in which the operator is not bisimulation invariant at all; however, as we will see, it does have some other kind of looseness property, and this enables us to prove its decidability.

4.1 Locality nor looseness: grid logics

In this subsection we consider modal systems that cannot be called loose or local. We will first meet a simple modal system that is tailored towards encoding the $\mathbb{N} \times \mathbb{N}$–tiling problem, and is undecidable; as a contrast we also discuss a second system with grid–like models which has a decidable satisfiability problem.

A tiling logic Looking for the opposite of looseness one is bound to end up in a *grid*. Grids are well-known in complexity theory since they play an important role in the formulation of a class of complete problems for various complexity classes: *tiling problems*. A tile is a one-by-one square which has a 'color' on each of its sides; these colors are given by four functions 'right', 'left', 'up' and 'down'. Given a set T of tiles, a *tiling* of the grid $\mathbb{N} \times \mathbb{N}$ by T is a map t from $\mathbb{N} \times \mathbb{N}$ to T satisfying, for all $n, m \in \mathbb{N}$:

$$\begin{aligned} \mathrm{right}(t(n,m)) &= \mathrm{left}(t(n+1,m)), \\ \mathrm{up}(t(n,m)) &= \mathrm{down}(t(n,m+1)). \end{aligned}$$

Tiles are assumed to be fixed in orientation, so the above conditions say that colors of adjacent tiles match. (We note that it is not needed to use all tiles of T in a tiling of $\mathbb{N} \times \mathbb{N}$.) If such a tiling exist, we say that T *can tile* $\mathbb{N} \times \mathbb{N}$.

The following problem is undecidable.

$\mathbb{N} \times \mathbb{N}$ **tiling**: Given a finite set T of tiles, can T tile $\mathbb{N} \times \mathbb{N}$?

We will now define a modal system Tile which is tailored to encode the above tiling problem. The language of Tile contains two unary modalities $\Diamond_r$ and $\Diamond_u$ plus the universal modality E. In a model of the form (W, R_r, R_u, V), these modalities receive their meaning in the usual way:

$$\begin{aligned} \mathcal{M}, s \Vdash \Diamond_r\varphi &\iff \mathcal{M}, t \Vdash \varphi \text{ for some } t \text{ with } R_r st, \\ \mathcal{M}, x \Vdash \Diamond_u\varphi &\iff \mathcal{M}, t \Vdash \varphi \text{ for some } t \text{ with } R_u st, \\ \mathcal{M}, x \Vdash \mathsf{E}\varphi &\iff \mathcal{M}, t \Vdash \varphi \text{ for some } t. \end{aligned}$$

In the intended class of *grid models*, R_r and R_u are (the graphs of) two commuting total functions. In particular, grid models satisfy the following condition:

$$\forall xyz((R_r xy \wedge R_u xz) \to \exists w(R_r zw \wedge R_u yw)). \tag{5}$$

Because of this, the class of grid models is not closed under unraveling; hence, Tile does not satisfy the looseness principle. It is also rather obvious that the first locality principle fails as well, in the presence of the universal modality; we leave it to the reader to verify that the class of grid models can be defined using three variables only, and Tile thus satisfies the second locality principle.

Theorem 4. *The satisfiability problem of* Tile *is undecidable.*

Proof. Obviously, we reduce the $\mathbb{N} \times \mathbb{N}$–tiling problem to the satisfiability problem for Tile. We present a procedure that outputs for every instance T of the tiling problem a formula φ_T such that

$$\mathsf{A}\varphi_T \text{ is } \mathsf{Tile} \text{ satisfiable iff } T \text{ can tile } \mathbb{N} \times \mathbb{N}. \tag{6}$$

(Recall that A is the box version of E; that is, $\mathsf{A}\varphi$ abbreviates $\neg\mathsf{E}\neg\varphi$.)

Take for any set $T = \{T_1, \ldots, T_k\}$ of tiles a corresponding set $\{t_1, \ldots, t_k\}$ of propositional variables. Define φ_T as the conjunction of the following formulas (where i ranges over $1,\ldots,k$):

$A1$ $\bigvee_{1 \le i \le k} t_i$
$A2_i$ $t_i \to \bigwedge_{i \ne j} \neg t_j$
$A3_i$ $t_i \to \Diamond_r \bigvee \{t_j \mid \mathrm{right}(T_i) = \mathrm{left}(T_j)\}$
$A4_i$ $t_i \to \Diamond_u \bigvee \{t_j \mid \mathrm{up}(T_i) = \mathrm{down}(T_j)\}$.

It is almost immediate that T tiles $\mathbb{N} \times \mathbb{N}$ if and only if there exists a Tile model where φ_T holds throughout. (The reader should verify that in the proof of the left to right direction of (6) the property (5) of grid models is crucial.) This in turn is equivalent to the formula $\mathsf{A}\varphi_T$ being satisfiable in some Tile model. Thus (6) holds and we have reduced the undecidable tiling problem to the Tile satisfiability problem. ◀

We hasten to remark that the undecidability of this system has nothing to do with the fact that we are dealing with more than one modality here; one can easily transform this example into an undecidable modal system in the *basic modal language* extended with the universal modality, or in the basic modal language proper.

It is interesting to note that *without* the universal access to the models provided by A these grid logics become quite harmless. In fact, their grid–like nature ensures that every satisfiable formula ξ is satisfiable in a model whose size is at most $|\xi|^2 + 1$.

Theorem 5. *Let* Tile^- *be the modal system* Tile, *but now without the universal modality. Then every* Tile^- *satisfiable formula* ξ *is satisfiable in a* Tile^- *model of size at most* $|\xi|^2 + 1$. *As a corollary,* Tile^- *has a decidable satisfiability problem.*

Proof. Let $\mathcal{M}$ satisfy ξ at s. Let k be the modal depth of ξ, then we have that $k \le |\xi|$. By Proposition 2, ξ is still satisfiable in the model $\mathcal{M}'$ defined as the substructure of $\mathcal{M}$ with universe s together with all states reachable in at most k (R_r- or R_u-)steps from s. Clearly the size of the universe of $\mathcal{M}'$ is at most k^2. Unfortunately $\mathcal{M}'$ is not a Tile model, because not every state has a R_r and R_u successor. In order to mend this, add one dummy state x to the universe of $\mathcal{M}'$ and put a link from w to x for all states w (including x itself) that do not have a successor yet. That is, define $W^- = W' \cup \{x\}$, $R_r^- = R_r' \cup \{(w, x) \mid R_r'wy$ for no y in $\mathcal{M}'\}$, and likewise for R_u^-. Let the valuation stay the same, i.e., define $V^-(p) = V'(p)$ for all p.

The resulting model $\mathcal{M}^-$ is a Tile model. Clearly ξ is still satisfied at s in this new model, since x is 'too far away' to have any effect on the truth of ξ. This proves the first part of the theorem. Decidability now follows because it is decidable whether a finite model is a Tile model. ◀

S5² The second logic that we consider here is also based on grid-like structures, but here we only require that the models are two-dimensional in nature, there will be no orderings or functions around. The language has two diamonds, $\Diamond_0$ and $\Diamond_1$, with the standard truth definition. The models are of the form $\mathcal{M} = (W, \equiv_0, \equiv_1, V)$, where we require that $(W, \equiv_0, \equiv_1)$ is in fact a *square* over some set U. That is, W consists of the set $U \times U$ of all *pairs* over U, while $s \equiv_i t$

holds if $s_i = t_i$: the i-th coordinate of s and the i-th coordinate of t should be the same. Denote the resulting system as $\mathbf{S5^2}$.

As a modal system, $\mathbf{S5^2}$ might look rather obscure, but as a logic it is well known. In fact it is the exact modal counterpart of a restricted fragment of first order logic with two variables in a signature having a binary relation symbol R for every propositional variable r. This is seen as follows. First observe that the $\mathbf{S5^2}$ model $\mathcal{M} = (W, \equiv_0, \equiv_1, V)$ with $W = U \times U$ is uniquely determined by the first order model (U, V) for the described signature. Also observe that we may identify assignments s mapping the two variables x_0 and x_1 to U with pairs $(s(x_1), s(x_0)) \in W$. Thus viewing the states of the modal models as assignments, we may read the statement 'φ holds in (U, V) under assignment s' modally as 'in model $(U \times U, \equiv_0, \equiv_1, V)$, φ is true at state s'. Because $\mathbf{S5^2}$ models are squares, the truth definition of the diamonds can be rewritten exactly as the definition of the first order existential quantifiers:

$$\mathcal{M}, (a, b) \Vdash \diamond_1 \varphi \iff \text{there exists } a' \text{ such that } \mathcal{M}, (a', b) \Vdash \varphi.$$

Thus $\diamond_i$ is another way of writing $\exists x_i$. In a similar way, one can define modal systems $\mathbf{S5^n}$ corresponding to first order logic with n variables for any n. See the notes.

It will be obvious that this class of models is not closed under unraveling, and that $\mathbf{S5^2}$ will not have the tree model property. Concerning the locality principles, observe that this system has full global expressive power: that the 'combined' operator $\diamond_0 \diamond_1$ behaves just like the universal diamond E. Nevertheless, the system is decidable, and a proof for this uses some kind of finite model property as well.

Here, instead of defining a finite model for ξ by *selecting* points out of the old model, we will *identify* points in the big model and define the finite model as some sort of quotient structure which we call a *filtration* of the original model. It will turn out that this filtration will not be a square itself but a square-like structure that we here dub a *pseudo-square.* This is to say that in the underlying frame (W, R_0, R_1) both R_0 and R_1 are equivalence relations, and their composition should be the universal relation. That is, (W, R_0, R_1) has to validate

$$\forall xy \exists z (R_0 xz \wedge R_1 zy). \tag{7}$$

For these kind of structures we can prove the following Proposition which establishes the bounded finite model property of the language with respect to the class of pseudo-squares. (In fact, the system does have the bounded finite model property, but this is much harder to establish.) As we saw before, decidability follows immediately, because it is decidable whether a finite structure is a pseudo-square.

Proposition 4. *Any $\mathbf{S5^2}$-formula ξ is satisfiable in a square iff it is satisfiable in a pseudo-square of size not exceeding $2^{|\xi|}$. As a consequence, $\mathbf{S5^2}$ has a decidable satisfiability problem.*

Proof. We concentrate on the left to right direction of this proof since we are only interested in explaining the notion of filtration at the moment. (For the other direction of the proof, one shows that given a pseudo-square model, one can always find a square that is bisimilar to it — in fact, bisimilar through a *functional* bisimulation, see the notes.

Suppose that ξ is satisfied somewhere in the square model $\mathcal{M} = (W, \equiv_0, \equiv_1, V)$. From this we will prove that ξ is true somewhere in a *filtration* $\mathcal{M}^f$ of $\mathcal{M}$. As we mentioned already, filtrating a model means collapsing it. But when will two points in the original model be identified? Generally, taking a quotient of a structure means identifying points without 'relevant' differences; in the present context this can be interpreted as 'satisfying the same *subformulas* of ξ'. Formally, define $Cl(\xi)$ to be the smallest set of formulas containing ξ which is closed under subformulas. Now define the following relation on W:

$$s \sim s' \iff \text{for all } \varphi \text{ in } Cl(\xi) : \mathcal{M}, s \Vdash \varphi \text{ iff } \mathcal{M}, s' \Vdash \varphi.$$

Obviously, $\sim$ is an equivalence relation. Our filtrated model will be based on the equivalence classes of this relation, so we introduce some notation: with $\bar{s}$ we denote the equivalence class of a point s, and with W^f, the set of these classes. Note that $|W^f| \leq 2^{|\xi|}$ as $|Cl(\xi)|$ is bounded by $|\xi|$.

What would be a good definition for the relations R_0 and R_1 on W^f? In general, this is where the filtration method needs some creative input. Now, if the only requirement were that ξ is to be true somewhere in the resulting model, there is a whole family of definitions that work (in the sense that they ensure (8) below). But the extra constraint, viz., that the resulting model should be a pseudo-square, puts some extra restrictions. Nevertheless, the following definition works:

$$R_i\bar{s}\bar{t} \text{ if for all } \Diamond_i\varphi \in Cl(\xi)\text{: } \mathcal{M}, s \Vdash \Diamond_i\varphi \text{ iff } \mathcal{M}, s' \Vdash \Diamond_i\varphi.$$

(Observe that this is well-defined by the fact that $\sim$-equivalent points agree about *all* formulas in $Cl(\xi)$.) Finally, the definition of V^f is rather obvious:

$$V^f(p) = \{\bar{s} \in W^f \mid s \in V(p)\}.$$

Note that this is well-defined for all proposition letters p occurring in ξ.

We can prove the main claim concerning filtration:

$$\text{for all formulas } \varphi \in Cl(\xi)\text{: } \mathcal{M}, s \Vdash \varphi \text{ iff } \mathcal{M}^f, \bar{s} \Vdash \varphi. \tag{8}$$

This claim is proved by a formula induction. Leaving the straightforward induction base and the boolean cases of the inductive step to the reader, we concentrate on the case that φ is of the form $\Diamond_0\psi$. (The case that φ is of the form $\Diamond_1\psi$ is of course completely analogous.)

First assume that $\mathcal{M}, s \Vdash \Diamond_0\psi$. Then by definition there is some s' in $\mathcal{M}$ such that $s \equiv_0 s'$ and $\mathcal{M}, s' \Vdash \psi$. By the inductive hypothesis this gives that $\mathcal{M}^f, \bar{s}' \Vdash \psi$. It easily follows from the definitions that $s \equiv_0 s'$ implies $R_0\bar{s}\bar{s}'$. But then it is immediate that $\mathcal{M}^f, \bar{s} \Vdash \Diamond_0\psi$. For the other direction, suppose that $\mathcal{M}^f, \bar{s} \Vdash \Diamond_0\psi$. Then for some $\bar{t}$ in $\mathcal{M}^f$ we have that $R_0\bar{s}\bar{t}$ and $\mathcal{M}^f, \bar{t} \Vdash \psi$. Hence, by the inductive hypothesis we have that $\mathcal{M}, t \Vdash \psi$. But then from reflexivity of $\equiv_0$ it follows that $\mathcal{M}, t \Vdash \Diamond_0\psi$, so from $R_0\bar{s}\bar{t}$ we may infer, only using the definition of R_0, that $\mathcal{M}, s \Vdash \Diamond_0\psi$.

This proves (8), so in order to prove the left to right direction of the Proposition we only have to show that $\mathcal{M}^f$ is a pseudo-square. We leave it to the reader to verify that both R_0 and R_1 are equivalence relations. In order to check the other condition, let $\bar{s}$ and $\bar{t}$ be points in $\mathcal{M}^f$. Now the fact that $\mathcal{M}$ is a square and that s and t are pairs comes in handy. Let $z = (s_0, t_1)$. Then $s \equiv_0 z \equiv_1 t$. But then it follows that $R_0\bar{s}\bar{z}$ and $R_1\bar{z}\bar{t}$, which shows that indeed, the composition of R_0 and R_1 is the universal relation on $\mathcal{M}^f$. ◀

What can we conclude from the examples Tile, Tile^- and $\mathbf{S5^2}$? Not that looseness is a *necessary* condition for a modal system to be decidable, witness Tile^- and $\mathbf{S5^2}$. On the other hand it should be clear that dropping the looseness principle leads us to the immediate vicinity of the danger zone: adding only a grain of global expressive power will turn the highly decidable logic Tile^- into the undecidable Tile.

Concerning $\mathbf{S5^2}$, it is very interesting to observe what happens if we move to higher dimensions. For instance, there seem to be *two* three-dimensional counterparts of $\mathbf{S5^2}$, according to which relation between two triples one takes to be the accessibility relation for $\Diamond_i$:

$$s \equiv_i t \text{ if } s_j = t_j \text{ for all } j \neq i,$$
$$s \sim_i t \text{ if } s_i = t_i.$$

In the second, relatively loose, interpretation the resulting logic is decidable. In the first interpretation, one obtains a class of rather tight models; the resulting logic is *undecidable*. Since it is *this* logic that corresponds to a *three* variable fragment of first order logic (in a similar way as discussed above for $\mathbf{S5^2}$), this makes an interesting case for the second locality principle.

4.2 Universal access: $\mathbf{K}^*$

We now consider the modal system $\mathbf{K}^*$ obtained by expanding the basic modal language with the modality $\langle * \rangle$, keeping the class of models intact and giving both $\Diamond$ and $\langle * \rangle$ the standard interpretation. Recall that the meaning of $\langle * \rangle$ was defined using the reflexive transitive closure R^* of the relation R.

Let us first see where $\mathbf{K}^*$ stands with respect to the looseness and locality principles. We saw already that $\langle * \rangle$ is invariant for bisimulations, whence we have an analogue of Theorem 2: any $\mathbf{K}^*$-satisfiable formula is satisfiable in a tree model. $\mathbf{K}^*$ also meets the second locality principle, at least if we are allowed to include finite variable fragments of the *infinitary* language $\mathcal{L}_{\omega_1\omega}$ (an extension of first order logic in which countable conjunctions and disjunctions are allowed). For, it is easy to see that $\mathbf{K}^*$-formulas have correspondents in the three variable fragment of this language: simply add the following clause to the standard translation of $\mathcal{L}_\diamond$ with the following clause for $\langle * \rangle$:

$$ST_x(\langle * \rangle \psi) \;=\; \exists y (R^* xy \wedge ST_y(\varphi)), \qquad ST_y(\langle * \rangle \psi) \;=\; \exists x (R^* yx \wedge ST_x(\varphi)).$$

Here we use the fact that the reflexive transitive closure can be expressed using three variables only; for instance, R^*xy could stand for the following abbreviation:

$$x = y \;\vee\; Rxy \;\vee\; \exists y' \,(Rxy' \wedge Ry'y) \;\vee\; \exists y' \,(\exists y \,(Rxy \wedge Ryy') \wedge Ry'y) \;\vee\; \ldots$$

However, $\mathbf{K}^*$ violates the first locality principle in the following way: if r is the root of a tree model $\mathcal{M}$, then we have

$$\mathcal{M}, r \Vdash \langle * \rangle \varphi \iff \mathcal{M}, s \Vdash \varphi \text{ for some } s \text{ in } \mathcal{M},$$

as the reader can easily check. In fact, unlike the basic modal system, $\mathbf{K}^*$ does not have the *finite* tree model property; for instance, the following satisfiable formula is not satisfiable on any finite tree:

$$[\,*\,](p \rightarrow \langle * \rangle \neg p) \wedge [\,*\,](\neg p \rightarrow \langle * \rangle p).$$

(It is satisfiable on the natural numbers with successor, with p interpreted as the even numbers.)

Summarizing, the present system is loose and it satisfies the second, but not the first locality principle. What about its decidability, or the finite model property? In fact, both properties hold, as we will see now. We first prove that $\mathbf{K}^*$ has the bounded model property.

Proposition 5. *Any satisfiable* $\mathbf{K}^*$ *formula* ξ *is satisfiable on a model of size* $2^{O(|\xi|)}$.

Proof. Suppose that ξ is satisfiable in some model $\mathcal{M} = (W, R, V)$. We again define a collection of relevant formulas; this time we need a new closure rule: call a set X of formulas $*$-closed if it contains $\diamond\langle * \rangle \varphi$ whenever it contains $\langle * \rangle \varphi$. Now let $FL(\xi)$ be the smallest set of formulas containing ξ which is, $*$-closed, besides being closed under taking subformulas and single negations. It is not difficult to prove that the cardinality of $FL(\xi)$ is linear in the size of ξ.

Again, the method that we use to construct a finite model for ξ is, just like in the case of $\mathbf{S5^2}$, that of *filtration*. Define the following relation on points of $\mathcal{M}$:

$$s \sim s' \iff \text{for all } \varphi \text{ in } FL(\xi) : \mathcal{M}, s \Vdash \varphi \text{ iff } \mathcal{M}, s' \Vdash \varphi.$$

Again, it is obvious that $\sim$ is an equivalence relation, and again, our filtrated model will be based on the collection W^f of equivalence classes of this relation; it is convenient to identify the equivalence class of s with the *color* of s, which we define as the set $c(s) = \{\varphi \in FL(\xi) \mid \mathcal{M}, s \Vdash \varphi\}$. Note that $|W^f| \leq 2^{O(|\xi|)}$ as $|FL(\xi)|$ is bounded by $|\xi|$.

To finish the definition of the filtrated model, we define the relation R^f on colors as follows:

$$R^f cd \iff \text{for all } \Diamond\varphi \in FL(\xi)\text{: } (\varphi \in d \Rightarrow \Diamond\varphi \in c).$$

The valuation V^f is then defined as $V^f(p) = \{c \in W^f \mid p \in c\}$.

The key claim of the filtration proof is the following.

Claim 1 For all formulas $\varphi \in FL(\xi)$ and all colors c: $\varphi \in c$ iff $\mathcal{M}^f \Vdash \varphi$.

Proof of Claim The proof follows by an induction on the complexity of φ. We only treat the case where φ is of the form $\langle * \rangle\psi$.

First suppose that $\langle * \rangle\psi \in c$. Assume that c is the color of s in $\mathcal{M}$; that is, $\mathcal{M}, s \Vdash \langle * \rangle\psi$. By definition, there is a sequence of states $s_1, \ldots, s_n$ in $\mathcal{M}$ such that $s = s_1$, $Rs_i s_{i+1}$ for all i, and $\mathcal{M}, s_n \Vdash \psi$. By definition of colors it follows that $\psi \in c(s_n)$. Also, it is easy to show that $R^f c(s_i)c(s_{i+1})$ for all i. But then it is immediate that $\mathcal{M}^f, c \Vdash \langle * \rangle\psi$.

For the other direction, suppose that $\mathcal{M}^f, c \Vdash \langle * \rangle\psi$. By the truth definition of $\langle * \rangle$, there must be colors $c_1, \ldots, c_n$ such that $c = c_1$, $R^f c_i c_{i+1}$ for all i and $\mathcal{M}, c_n \Vdash \psi$. It follows from the inductive hypothesis that $\psi \in c_n$. From this, and the observation that $\varphi \to \langle * \rangle\varphi$ is valid in any model, it follows that $\langle * \rangle\psi \in c_n$.

We now show that

$$\text{If } R^f dd', \text{ then } \langle * \rangle\chi \in d' \text{ implies } \langle * \rangle\chi \in d. \tag{9}$$

Suppose that $R^f dd'$ and $\langle * \rangle\chi \in d'$. It follows that $\langle * \rangle\chi$ belongs to $FL(\xi)$, so $\Diamond\langle * \rangle\chi$ is in $FL(\xi)$ as well, by $*$-closure. But then by definition of R^f we find that $\Diamond\langle * \rangle\chi$ is in d. Since d is a color, there must be some w in W such that $d = c(w)$. By definition we have that $\mathcal{M}, w \Vdash \Diamond\langle * \rangle\chi$. From this it is easy to derive that $\mathcal{M}, w \Vdash \langle * \rangle\chi$, so again by definition we have $\langle * \rangle\chi \in d$.

But from (9) and $\langle * \rangle\psi \in c_n$ an easy downward inductive proof shows that $\langle * \rangle\psi \in c_i$ for all i. In particular, we find that $\langle * \rangle\psi$ belongs to $c_1 = c$. ⊣

Thus ξ is satisfiable in a model of size $2^{O(|\xi|)}$. ◀

The last proposition implies decidability as it is decidable whether a $\mathbf{K}^*$ formula is satisfiable on a finite model. The idea of colors can also be used directly in an algorithm which tries to construct a model like $\mathcal{M}^f$. This construction uses the same idea as the K–World algorithm given earlier: states are identified with subsets of $Cl(\xi)$. Let S_0 consists of all sets $\Delta \subseteq Cl(\xi)$ for which $Prop\text{–}Max(\Delta, Cl(\xi))$ holds and which satisfy $\varphi \in \Delta \Rightarrow \langle * \rangle\varphi \in \Delta$, for all $\langle * \rangle\varphi \in Cl(\xi)$. (For the definition of $Prop\text{–}Max(\Delta, Cl(\xi))$, see section 3.) Clearly S_0 can be effectively computed and $|S_0| \leq 2^{O(|\xi|)}$. We now inductively construct a sequence of sets of sets of formulas $S_0 \supsetneq S_1 \supsetneq S_2 \supsetneq S_3 \cdots$. During this construction, just as in the K–World algorithm, we try to find witnesses for diamond formulas. We say that a set $\Delta \in S_i$ is *ready* if S_i contains witnesses for all diamond formulas in Δ:

– for every formula $\Diamond\psi \in \Delta$ there is a $\Delta_\psi \in S_i$ such that $R^f \Delta\Delta_\psi$ and $\psi \in \Delta_\psi$, and
– for every formula $\langle * \rangle\psi \in \Delta$ there are $\Delta_1, \Delta_2, \ldots, \Delta_n \in S_i$ such that $\Delta = \Delta_1$, $R^f \Delta_i \Delta_{i+1}$ and $\psi \in \Delta_n$,

If every set in S_i is ready and S_i contains a set Δ with $\xi \in \Delta$, then return 'ξ is satisfiable'. If there is no set in S_i containing ξ, then return 'ξ is not satisfiable'. Otherwise, let S_{i+1} consists of all ready sets in S_i and continue the construction. Since $S_i \supsetneq S_{i+1}$, the construction is guaranteed to terminate in at most $2^{O(|\xi|)}$ stages. Correctness of the algorithm can be shown along the lines of the proof of the last proposition. Thus we have established the following.

Theorem 6. *It is decidable whether a given* $\mathbf{K}^*$ *formula is satisfiable.*

4.3 Generalizing looseness: until

In this subsection, we consider the modal system given by the propositional language expanded with the binary until operator U, the class of all models of the form (W, R, V) and an interpretation of U as given above (recalled below). We already mentioned that truth in this language is not bisimulation invariant, and we are thus not dealing with a modal system in the narrow sense; in particular, we will see that there are satisfiable U-formulas that are not satisfiable in any tree. Nevertheless, we will show that this system does have some kind of *loose* model property, and we use this property for showing that it has a decidable satisfiability problem. In fact, this 'looseness property' is the reason why we have a look at this operator: it shows in a relatively simple setting how to generalize the notions of looseness and tree models. These generalizations are made in the section on guarded fragments.

To start with, let $\mathcal{L}_U$ be the language obtained by expanding the classical propositional language with the binary connective U. Recall that M is the class of all models of the form (W, R, V). It is convenient to use the following notation: for s and u elements of W,

$$\mathcal{M}, su \Vdash \psi \text{ iff } \mathcal{M}, t \Vdash \psi, \text{ for all } t \text{ satisfying } Rst \text{ and } Rtu, \tag{10}$$

because we can now rephrase the truth definition of the until operator as follows:

$$\mathcal{M}, s \Vdash U(\varphi, \psi) \text{ iff } \mathcal{M}, u \Vdash \varphi \text{ and } \mathcal{M}, su \Vdash \psi, \text{ for some } u \text{ such that } Rsu. \tag{11}$$

We call the resulting modal system $(\mathcal{L}_U, \mathsf{M}, \Vdash)$ the *until system*. In order to see why truth of $\mathcal{L}_U$-formulas is not invariant for bisimulations, consider the formula $U(p, \top) \wedge \neg U(p, p)$. (Here $\top$ abbreviates $(p \vee \neg p)$.) This formula is satisfiable and its smallest irreflexive model contains 3 points, cf. model $\mathcal{M}_1$ in Figure 7. Note that in the unraveling $\mathcal{M}_2$ of the model $\mathcal{M}_1$, $U(p, p)$ holds at the root of the tree — this shows that $\mathcal{L}_U$ is really a more expressive language than $\mathcal{L}_\Diamond$. In fact, one can show that the formula $U(p, \top) \to U(p, p)$ holds throughout any tree model, whence $U(p, \top) \wedge \neg U(p, p)$ is not satisfiable in any tree model. This shows that the until system does not have the tree model property.

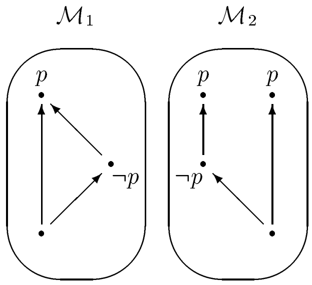


Fig. 7. Until formulas are not invariant for bisimulations.

Decidability Different from our earlier proofs, we will not use any kind of finite *model* property in order to prove decidability for the until system. This is not because the system does not have the bounded finite model property (it does); our proof method is for didactic purposes. The idea behind the *mosaic method* that we employ is that instead of transforming a model into a finite model, we could just as well 'deconstruct' it into a finite 'toolkit' that we will call a linked set of mosaics. One then has to show that a formula is satisfiable if and only if there exists such a linked set of mosaics for it.

What then are mosaics? One could best describe them as little pieces of a model that, if linked together in a nice way, contain sufficient information to *reconstruct* another model which looks sufficiently much like to the original one to preserve the truth of $\mathcal{L}_U$-formulas. In this way we will establish a *loose model property* for the until system: any satisfiable formula is satisfiable in a model consisting of these isomorphic copies of mosaics that hang together only very loosely. (Later on in the paper we will come back to this issue in more technical detail.)

Concerning the notion of a mosaic then, the first question is *what information* we are interested in. This question is easy to answer: as in the filtration proof for $\mathbf{S5^2}$, we are only interested in the truth of subformulas of ξ. The second question then should be: *which parts* are we going to cut out of the model. Here we need to define a new concept. Call a subset of the domain of a model $\mathcal{M} = (W, R, V)$ *packed* if every two distinct elements s and t of the subset are R-related (that is, we require that Rst or Rts). Our patchwork pieces then will be packed sets of size at most three.

The number three here derives from the fact that the truth definition of $U(\varphi, \psi)$ employs three variables. In fact, if one would try to devise a standard translation or a bisimulation game for the $\mathcal{L}_U$-language, the number three would show up as the minimal number of variables needed and as the minimal size of the windows that cover the models during the game. During a game one would see that these windows will only be placed on packed sets of the models.

Abstracting from the origin of these pieces, we arrive at the following definition. From now on we let ξ be an arbitrary but fixed $\mathcal{L}_U$ formula. ξ is the formula whose satisfiability needs to be decided. We let $Cl(\xi)$ denote the set of subformulas of ξ.

Definition 7. *A ξ-type* mosaic *is a quadruple $\mu = (X, R, A_\varphi, B_\varphi)_{\varphi \in Cl(\xi)}$ such that X is a set of size at most three; R and every B_φ are binary relations on X; and every A_φ is a unary relation on X. When ξ is clear from the context, we just use the phrase mosaic.*

The basic idea underlying this definition is that A_φ holds of a point if we 'want' φ to be true at it, while B_φ holds of a pair of points if we 'want' φ to be true at each point between them. Obviously, not every such structure is part of some model — we need some further constraints for that. Call a mosaic *coherent* if it satisfies the following conditions (phrased in first order logic and to be read universally):

$(C0)$ $Rxy \vee Ryx \vee x = y$
$(C1)$ $A_{\neg\varphi}x \leftrightarrow \neg A_\varphi x$
$(C2)$ $A_{\varphi\wedge\psi}x \leftrightarrow A_\varphi x \wedge A_\psi x$
$(C3)$ $B_{\varphi\wedge\psi}xy \leftrightarrow B_\varphi xy \wedge B_\psi xy$
$(C4)$ $(Rxy \wedge Ryz \wedge B_\varphi xz) \rightarrow A_\varphi y$
$(C5)$ $(Rxy \wedge A_\varphi y \wedge B_\psi xy) \rightarrow A_{U(\varphi,\psi)}x$

A few words of explanation: C0 reflects the fact that we only took packed subsets of the model as the domain of our mosaic mini-models. C1–C3 are selfexplanatory; note that there is no analog of C1 for the B-predicates since there is a hidden universal quantifier in the meaning of a predicate B_φ, cf. (10). Finally, C4 and C5 are rather obvious consequences of our intuitive meaning of the A- and B-predicates and the truth definition of the until operator.

The conditions C0–C5 take care of all *universal* constraints on the A- and B-predicates; but of course there are *existential* demands as well which we will call *requirements.* A requirement of a mosaic $\mu = (X, R, A_\varphi, B_\varphi)_{\varphi \in Cl(\xi)}$ is one of the two following types of object:
(a) $(A_{U(\varphi,\psi)}, s)$ such that $A_{U(\varphi,\psi)}s$,
(b) $(notB_\varphi, s, t)$ such that Rst and *not* $B_\varphi st$.
In order to explain requirements of type (a), suppose that we want the formula $U(\varphi, \psi)$ to be true at a point s; if there is a point t in the mosaic such that Rst, $A_\varphi t$ and $B_\psi st$, then the mosaic itself *directly fulfills* the requirement. This will rarely be the case however; the whole point of the mosaic method is that requirements can be fulfilled by *distinct* mosaics as well, as follows. A *link* between two mosaics μ and μ' is simply a partial isomorphism between the two structures. We say that a link $f : \mu \hookrightarrow \mu'$ *fulfills the requirement* $(A_{U(\varphi,\psi)}, s)$ of μ if there is some t in μ' with $Rf(s)t$, $A_\varphi t$ and $B_\psi f(s)t$. Likewise, a link $f : \mu \hookrightarrow \mu'$ *fulfills the requirement* $(notB_\varphi, s, t)$ if there is some u in μ' with $Rf(s)u$, $Ruf(t)$ and $\neg A_\varphi u$.

A collection L of mosaics is called a *linked set of mosaics* if every requirement of every mosaic $\mu \in L$ is fulfilled via some link $f : \mu \hookrightarrow \mu'$ to some μ' also in L. It is a linked set of mosaics *for* ξ if it contains a mosaic with non-empty A_ξ.

Now the main theorem concerning mosaics is the following. (In order to follow the main line of the paper, the reader could skip the details of its proof.)

Proposition 6. *An $\mathcal{L}_U$-formula ξ is satisfiable if and only if there is a linked set of mosaics for ξ.*

Proof. The left to right direction of the proof is easy: suppose that $\mathcal{M} = (W, R, V)$ is a model for ξ. Out of this model we will cut a linked set of mosaics for ξ, as follows. Let X be the collection of triples $\boldsymbol{x} = \langle x_1, x_2, x_3 \rangle$ such that Rx_1x_2, Rx_2x_3 and Rx_1x_3. Associate with any such triple a mosaic $\mu_{\boldsymbol{x}}$ based on the set $\{x_1, x_2, x_3\}$, with R as in $\mathcal{M}$ and with every A_φ and B_φ defined as given by the truth of φ in $\mathcal{M}$. We leave it as an exercise for the reader to verify that the collection of all these mosaics forms indeed a linked set of mosaics.

The right to left direction of the Proposition is the hard one, although the key idea underlying its proof is quite intuitive. We will construct a model for ξ *step by step*; that is, we will approximate our model via a series of finite structures that we call *networks*. A *network* is a structure $\mathcal{N} = (W, R, A_\varphi, B_\varphi)_{\varphi \in Cl(\xi)}$ of the same type as a mosaic but not bounded in size. A network is called *coherent* if it satisfies the conditions C1–C5 above. To ask for C0 would be too much; instead we require coherent networks to satisfy the following:

(liveness) every packed set X of size at most three *comes from* a mosaic; that is, for each such set $X \subseteq W$ there is a partial isomorphism $f : \mathcal{N} \hookrightarrow \mu$ such that f is defined on X.

Liveness means that — through the mosaics — we are in control of certain small parts of the model: the packed sets of size at most three. Why only these sets? The truth definition of U provides the answer. The meaning of $U(\varphi, \psi)$ depends only on these small packed sets in the model.

A *defect* of a network is a requirement that is not directly fulfilled in the network itself, and a network is called *saturated* if it has no defects. A network is *perfect* if it both coherent and saturated.

This name is well-chosen, since perfect networks are the ones that we are after. The reason for this is that with every network $\mathcal{N} = (W, R, A_\varphi, B_\varphi)_{\varphi \in Cl(\xi)}$ we can associate a modal model in an obvious way: it is defined as the structure $\mathcal{N}^\circ = (W, R, V^\circ)$ with $V^\circ(p) = A_p$ for all variables p occurring in ξ. But only for perfect networks can we prove the following *truth lemma.*

Claim. If $\mathcal{N}$ is a perfect network, then for all formulas $\varphi \in Cl(\xi)$ and all points s, t in $\mathcal{N}$:

1. $s \in A_\varphi$ iff $\mathcal{M}, s \Vdash \varphi$,
2. if Rst, then $(s, t) \in B_\varphi$ iff $\mathcal{M}, st \Vdash \varphi$.

PROOF OF CLAIM The proof of this claim is by induction on the complexity of φ. We only consider the case where φ is of the form $U(\psi, \chi)$, and only prove part 2 of the Claim (the first part is simpler).

By the induction hypothesis and the truth definition of U, in order to prove part 2 it suffices to show that for all pairs of points s and t such that Rst, we have that $(s, t) \notin B_\varphi$ iff $u \notin A_\varphi$ for some u with Rsu and Rut. The left to right direction immediately follows from the fact that $\mathcal{N}$ is perfect and thus all requirements of type b are fulfilled. For the other direction, suppose that s, t and u are points satisfying Rst, Rsu, Rtu and $u \notin A_\varphi$. Observe that $\{s,t,u\}$ is a packed set of size at most three, whence we may use the (liveness) condition. This yields a partial isomorphism f from $\mathcal{N}$ to some mosaic μ such that f is defined for each of s, t and u. It follows that $Rf(s)f(t)$, $Rf(s)f(u)$, $Rf(t)f(u)$ and $f(u) \notin A_\varphi$; but then it follows from condition C4 that $(f(s), f(t)) \notin B_\varphi$. Returning to $\mathcal{N}$ this shows that $(s,t) \notin B_\varphi$, which is what we needed to prove. $\dashv$

From the previous claim it follows that in order to show that ξ is satisfiable, it suffices to show that there is a perfect network for it, that is, a perfect network such that A_ξ is not empty.

Claim. There is a perfect network for ξ.

Proof of Claim The proof of this claim falls out in three parts. First we show that there is *some* network for ξ (not necessarily perfect). This is easy, since we are given a linked set of mosaics for ξ: as our network we simply take any mosaic with non-empty A_ξ.

The second and main part of the proof consists in showing that any defect of any network can be *repaired*; that is, we can find a bigger network in which the defect no longer occurs. Without going too much into technical detail, let us see how to repair a defect of type (b) (defects of type (a) are repaired in a similar way).

Suppose that s and t are points of the network $\mathcal{N}$ such that Rst and *not* $B_\varphi st$ for some subformula φ of ξ, while there is no point u between s and t such that $\neg A_\varphi u$. The idea now is simply to *repair* this defect by adding a *new point* to the network. What kind of point? Well, since we have Rst we know that s and t come from a mosaic; that is, there is a partial isomorphism f from $\mathcal{N}$ to some mosaic μ. Obviously, $(notB_\varphi, f(s), f(t))$ is a requirement of this mosaic. But since we are working in a *linked* set of mosaics, there must be some link g between μ and μ' and some u in μ' such that $Rg(f(s))u$, $Rug(f(t))$ and $\neg A_\varphi u$. Now simply add an entirely new point r to the network; make sure that the relations between s, t and r are such that this part of the model is isomorphic to μ'. It is thus obvious that we have *repaired* the defect, and that the new structure is a network. In order to keep the liveness condition it is essential *not* to relate r to any *other* point besides s and t: in this way the only new packed sets are $\{r,s,t\}$ and its subsets.

Finally, these two parts provide the material and the tools for constructing the desired perfect network for ξ. Starting from the mosaic for ξ (which is of course a network), we repair defects, one by one, step by step, thus constructing a sequence $\mathcal{N}_0, \mathcal{N}_1, \ldots$ of networks. Using some standard combinatorics we can ensure that the *limit* of the chain of networks is a network without defects. In particular, if we always take new points from a fixed set, say ω, we can enumerate the set of all (potential) defects of any network of the chain; if at each step of the construction we repair the current network's defect with the lowest number in this enumeration, we can create a perfect network. $\dashv$

◀

Theorem 7. *It is decidable whether a given $\mathcal{L}_U$-formula is satisfiable.*

Proof. We can adjust the 'elimination algorithm' given for the system $\mathbf{K}^*$ in order to deal with mosaics. This is done as follows. Let S be the set of all ξ-type mosaics (up to isomorphism). Let $S_0 \subseteq S$ be the subset containing all coherent mosaics. S_0 can be computed effectively since coherence can be checked effectively. It is not hard to show that $|S| \leq 2^{O(|\xi|)}$. We now inductively construct a sequence of sets of mosaics $S_0 \supsetneq S_1 \supsetneq S_2 \supsetneq S_3 \cdots$, just as in the proof for the system $\mathbf{K}^*$. The idea is that we delete mosaics from S_i if they have a requirement which can not be fulfilled inside S_i. The details of this construction will be spelled out in the section on guarded fragments. ◀

What is important to remember is that the until system has a kind of *loose model property*: if a formula ξ is satisfiable then there is a linked set of mosaics for it, and if there is such a set for ξ, then the proof of Proposition 6 shows how to construct a *loose model* for ξ. In Section 6 we will come back to this.

Notes

Tiling problems (or domino problems, as they are sometimes called) were introduced in Wang [60] and have since been used in a variety of forms to prove undecidability and complexity results. An accessible proof for the undecidability of the $\mathbb{N}\times\mathbb{N}$ tiling problem, a result due to Berger [8], can be found in the monograph Börger, Grädel & Gurevich [10]. Our discussion of the logic Tile was based on Spaan [54], where the author presents a example in a language expanding the *basic* modal language with the universal diamond.

The modal system $\mathbf{S5^2}$ (S5 square) has a long history in the algebraic disguise of the class of diagonal free cylindric algebras of dimension two, cf. the monograph [27]. The bounded finite model property for $\mathbf{S5^2}$ was first established by Segerberg [52]. The fact that every pseudo-square bisimulates by a functional bisimulation with a square can be found in Marx & Venema [42]. The higher dimensional counterparts of $\mathbf{S5^2}$ are studied as modal logics in Venema [58, 59].

The modal system $\mathbf{K}^*$ can best be seen as a fragment of Propositional Dynamic Logic (PDL) in which there is only one atomic program. For more information on PDL the reader is referred to the handbook article by Harel [24]. Decidability of PDL was proved by Fischer & Ladner [18]. The elimination algorithm leading to Theorem 6 is due to Pratt [49].

The operators 'Since' and 'Until' were introduced by Hans Kamp in order to prove expressive (in)completeness results for temporal logics over classes of linear flows of time. Nowadays, they belong to the standard repertoire of temporal logics in computer science, cf. [38]. A bisimulation variant which characterizes this language over arbitrary models was found by Kurtonina & de Rijke [35]; for some decidability results over classes of linear flows of time, see Burgess & Gurevich [12]. Our Theorem 7 and its proof are based on results and proofs on the loosely guarded fragment that are due to van Benthem.

As a tool for proving decidability results for modal logics, the filtration method has been used extensively, since Lemon [37] and Segerberg [51] further developed ideas dating back to McKinsey [43]. The mosaic method for proving decidability of a logic stems from Németi [44]; it has since been used for a wide range of logics, often related to a multi–dimensional modal setting. With hindsight, even Gödel's proof of the decidability of the satisfiability problem for the $\forall^2\exists^*$ prenex sentences can be called a mosaic style proof as well, cf. the very clear exposition in the monograph [10].

5 Modal complexity

In the previous sections, we discussed the decidability of the satisfiability problem for several modal systems, gathering various results along the way. For instance, for the basic modal system **K** we saw that every satisfiable formula can be satisfied in an exponential-size tree model with branches of polynomial depth; for $\mathbf{S5}^2$ we could do no better than finding an exponential–size quasi–model.

In this section we take a closer look at such differences, examining how they affect the complexity of the modal systems that we present. Our goal is not to give precise reductions and matching algorithms — this is very well documented in the literature. Rather we paint with a large brush and try to convey our earlier message again that looseness and locality are key notions in understanding the decidability and the complexity of modal systems. To do this we discuss modal systems whose satisfiability problem are complete for the complexity classes NP, PSPACE, EXPTIME and NEXPTIME, respectively. We believe that the given systems, besides being complete for these classes, form indeed very indicative examples.

Our agenda for the section is set out in Table 1. We assume that the reader has at least a basic understanding of complexity classes like P, NP, PSPACE, EXPTIME etc. Completeness and hardness are in this paper understood by means of polynomial time many–one reductions. (The reader is referred to [3] for basic definitions.) In this section we concentrate on the satisfiability problem — recall that C-completeness of the satisfiability problem for a modal system implies co-C-completeness of the validity problem.

The layout of this section is summarized in Table 1. Every column presents a modal system and the complexity class for which its satisfiability problem is complete. The third row indicates whether satisfiable formulas can be satisfied in tree models for that logic, and the fourth row whether the modal system is expressive enough to define the universal modality. We mention these two properties because they correspond to the looseness and first locality principle, respectively (as mentioned before, we will meet the second locality principle again in the last section). The star in the first column marks the special role of the logic **S5** which is the logic of the universal modality E by itself.

We will often use the tractability result concerning model checking that we mentioned earlier on when discussing the second locality principle. In this section, we confine ourselves to modal languages with a finite number of first order

modal system/logic	**S5**	Func	**K**	$\mathbf{K}^*$	$\mathbf{S5}^2$
complete for	NP	NP	PSPACE	EXPTIME	NEXPTIME
tree model prop.	*	yes	yes	yes	no
global expr. power	yes	no	no	yes	yes
subsection	5.1	5.1	5.2	5.3	5.4

Table 1. Layout of Section 3.

definable modal operators. For modal formulas in such a language, the *model checking* problem, (i.e., given as input a (finite) model $\mathcal{M}$, a state s and a formula ξ, determine whether $\mathcal{M}, s \Vdash \xi$), is solvable in PTIME in the size of *both* the model *and* the formula. Also, for an elementary (i.e., definable by a single first order sentence) class of models K, the membership problem, (given as input a finite model $\mathcal{M}$, determine whether $\mathcal{M}$ belongs to K), is also solvable in PTIME. We will call a modal system *elementary* if it has an elementary class of models and each of its operators has a first order truth definition.

5.1 NP and the polysize model property

The class NP of non–deterministic polynomial time algorithms is the smallest complexity class that we will consider for the satisfiability problem for modal systems. The reason for this is that every non-trivial modal logic contains the collection of all valid propositional formulas, whence we can reduce the NP–complete satisfiability problem for propositional logic to that of the modal system. So NP is a nice class to work in since we only have to show an upper bound. Unfortunately there are not many modal systems with a satisfiability problem in NP.

How can we show that the satisfiability problem is in NP for a given modal system? The easiest route and the one that modal logicians most often take goes via the *polysize model property.* A modal system is said to have this property if every satisfiable formula ξ is satisfiable in a model whose size is bounded by $p(|\xi|)$ for a fixed polynomial p. Using the two complexity results mentioned above, namely, PTIME for both the model checking and the membership problem, it is easy to show that for elementary modal systems, the polysize model property implies NP–completeness of the satisfiability problem.

Let us then see if we can find modal systems with this polysize model property; we will confine ourselves to the basic modal language. In section 3, we showed that every satisfiable formula φ can be satisfied in an at most $|\varphi|$–ary tree of depth at most $|\varphi|$: a model whose size is exponential in $|\varphi|$. Thus, if we want a polysize model, we should restrict either the width or the depth of such trees. This is possible if we consider smaller classes of models.

Restricting the *width* is easy: consider only models in which R is a total function. Then the cut–off argument from Section 3 yields a linear sized model. Recall from Definition 6 that $d_\diamond(\xi)$ denotes the modal depth of ξ.

Proposition 7. *Let* Func *be the modal system* $(\mathcal{L}_\Diamond, \Vdash, \mathsf{F})$ *such that* $\Vdash$ *is the standard definition and* F *is the class of models in which* R *is a total function. Let* ξ *be a formula in* $\mathcal{L}_\Diamond$. *If* ξ *is* Func*–satisfiable then it is* Func*–satisfiable in a model containing at most* $d_\Diamond(\xi) + 1$ *states.*

As a corollary, the satisfiability problem for Func *is in* NP.

An extremely simple way of bounding the depth is to make the accessibility relation total; a state at which φ holds is then a witness for $\Diamond\varphi$ at every state in the model.

Note however that making the accessibility relation total breaks with the first locality principle! Nevertheless, the resulting modal system, which we will call **S5** after the name of the logic associated with it, has a satisfiability problem in NP.

Proposition 8. *Let* **S5** *be the modal system* $(\mathcal{L}_\Diamond, \Vdash, \mathsf{U})$ *such that* $\Vdash$ *is the standard definition and* U *is the class of models in which* R *is the universal relation. Let* ξ *be a formula in the basic modal language. If* ξ *is* ***S5****–satisfiable then it is* ***S5****–satisfiable in a model containing at most* $|\xi|$ *states.*

As a corollary, the satisfiability problem for ***S5*** *is in* NP.

Proof. Let $\mathcal{M} = (M, R, V)$ be a model such that $R = M \times M$ and $\mathcal{M}, s \Vdash \xi$. Choose, for every subformula $\Diamond\varphi$ of ξ, a state $t \in M$ such that φ holds at t (if such a state exists). Let $\mathcal{M}'$ be the submodel of $\mathcal{M}$ consisting of s plus the selected states. By our pruning argument from Section 3, $\mathcal{M}', s \Vdash \xi$, because R is the universal relation. ◀

This finishes our discussion of NP and the polysize model property. Modal systems with this property are few and far between; in the next subsection we will see that for the basic modal system we can only get an exponential upper bound on the size of a model.

5.2 PSPACE and polynomially deep paths

In Section 3 we showed that every satisfiable formula ξ in the basic modal language is satisfiable in a finite tree model $\mathcal{M}$ of depth and branching degree both bounded by the length $|\xi|$ of the formula. The good news about this argument is that it can be used to show that satisfiability for **K** can be decided in PSPACE. On the other hand, the upper bound that it establishes on the size of $\mathcal{M}$ is no better than exponential — at this stage the reader might wonder whether this is an optimal bound. Here we show that it is — up to a polynomial. In fact, the satisfiability problem of **K** is *complete* for PSPACE.

We now define, for each natural number n, a satisfiable formula $\xi(n)$ with the following two properties

- the size of $\xi(n)$ is quadratic in n, and
- when $\xi(n)$ is satisfied in any model $\mathcal{M}$ at state s, then $\mathcal{M}$ contains as a substructure an isomorphic copy of the binary tree of depth n whose root is s.

Thus the size of the smallest model satisfying $\xi(n)$ is exponential in $|\xi(n)|$. The idea underlying the definition of $\xi(n)$ is very simple: take n propositional variables $p_0, \ldots, p_{n-1}$ and write a formula which when satisfied forces a binary branching tree in which every possible valuation on $\{p_0, \ldots, p_{n-1}\}$ occurs at some leaf. Thus the model certainly contains 2^n different states. The formula is constructed using two macro's: $branch(p_i)$ and $store(p_i)$ defined as follows:

$$\begin{aligned} branch(p_i) &:= \Diamond(p_i \wedge \Box p_i) \wedge \Diamond(\neg p_i \wedge \Box\neg p_i) \\ store(p_i) &:= (p_i \to \Box p_i) \wedge (\neg p_i \to \Box\neg p_i). \end{aligned}$$

The formula $\xi(n)$ then is given as

$$branch(p_0) \wedge \bigwedge_{1 \le i < n} \Box^i(branch(p_i) \wedge \bigwedge_{0 \le j < i} store(p_j)), \tag{12}$$

in which $\Box^i$ abbreviates an i–long sequence of boxes. The formula works as follows. Suppose $\mathcal{M}, s \Vdash \xi(n)$. Then the *branch* part of $\xi(n)$ states that every node t reachable in i R–steps from s has two different successors, one forcing $p_i \wedge \Box p_i$ and another forcing $\neg p_i \wedge \Box\neg p_i$. The *store* part of the formula states that successors of t created by the *branch* part satisfy precisely the same proposition letters $p_0, \ldots, p_{i-1}$ as does t. We leave it to the reader to verify that the interplay of the branch and store macros forces a binary tree of depth n, as desired.

PSPACE lower bound. Of course, failure of the polysize model property for the basic modal system in itself does not imply that its satisfiability problem cannot be decided in NP. However, in fact the lower bound of this problem is known to be PSPACE. This result can be obtained by an interpretation of the validity problem of quantified boolean formulas. This interpretation is based on the same two macro's *branch* and *store*.

In a similar way one can establish the existence of exponential sized models and a PSPACE lower bound for the modal system with $\langle * \rangle$ as its *only* modal operator (i.e., the fragment of $\mathbf{K}^*$ of formulas in which the ordinary diamond $\Diamond$ does not occur). To overcome the difficulty that $\langle * \rangle$ has direct one–step access to all states in a tree one has to add additional propositional variables to encode the depth of the tree. See the notes for details.

PSPACE upper bound. The $K-World$ algorithm for the basic modal system $\mathbf{K}$ from Figure 6 runs in PSPACE. Recall that for any formula ξ, ξ is $\mathbf{K}$ satisfiable iff there exists a set $\Delta \subseteq Cl(\xi)$ such that $\xi \in \Delta$ and $K\text{–}World(\Delta, Cl(\xi))$ is true. All sets encountered in the execution of $K-World$ are subsets of $Cl(\xi)$. Each subset of $Cl(\xi)$ can be represented in space $O(|\xi|)$, by using pointers to a copy of the formula. Therefore at each level of the recursion $O(|\xi|)$ space is used. After $d_\Diamond(\xi)$ recursive calls ($d_\Diamond(\xi)$ being the modal depth of ξ), there are no more $\Diamond\psi$ formulas in Σ and the recursion stops. Thus the recursion depth is bounded by $d_\Diamond(\xi) \le |\xi|$, whence the total amount of space required by the algorithm is $O(|\xi|^2)$. The existential demands in the algorithm (there exists a set Δ with

$\xi \in \Delta$ such that ..., and for all $\Diamond\psi \in \Delta$, there exists ...) make the algorithm nondeterministic. But PSPACE=NPSPACE by Savitch's Theorem.

Crucial in the PSPACE upper bound argument is that we can represent a complete branch of the tree model for ξ using only polynomial space. Two factors are important here. First, only a polynomial number of formulas is relevant at each world. And second, the depth of the branches is bounded by the modal depth of the input formula. This is caused by the first locality principle: lack of global expressive power. In the next subsection we show that adding such expressive power to the basic modal language destroys this polynomial depth property.

5.3 EXPTIME and exponentially deep paths

Now we will see that global expressive power in combination with another diamond destroys the polynomially bounded depth of the satisfying tree models for the basic modal system. In particular, we create a satisfiable formula which, when satisfied, forces a branch in the model containing an exponential number of colors. Thus the PSPACE algorithm sketched in the previous subsection will not work anymore. In fact this additional expressive power will be enough to show that the satisfiability problem is EXPTIME–hard. We again consider the system $\mathbf{K}^*$ from subsection 4.2. We want to show that its language is strong enough to force the existence of exponentially deep R-paths. A simple way of doing so employs binary counters.

By a binary counter we will understand a device that can have natural numbers as values, represented as binary strings of 0s and 1s; one should also be able to increment this value by one. We will use a set $\{p_0, \ldots, p_{n-1}\}$ of propositional variables to implement an n-ary binary counter (n-ary means that the counter is reset to zero after reaching $2^n - 1$). We use these variables to encode the n bits of the counter, with p_0 encoding the least significant and p_{n-1} the most significant bit. The variable p_i being true in a given state, encodes the fact that the ith bit of the counter is 1 in that state. The key idea to an encoding into the modal language lies in the following characterization of adding 1 to a binary counter. If $a = a_{n-1} \ldots a_0$ and $b = b_{n-1} \ldots b_0$ are two n–bit counters, then $b = a + 1$ (mod 2^n) precisely when the following holds: either $b_i = 0$ and $a_i = 1$ for all i (this is when we start counting at 0 again), or, for some $k \leq n - 1$ we have that

(1) $a_k = 0$, and $b_k = 1$,
(2) $a_j = 1$ and $b_j = 0$ for all $j < k$, and
(3) $a_i = b_i$ for all $i > k$.

In a picture:

$$\begin{array}{l} 10110\ 0\ 1111\ a \\ 00000\ 0\ 0001 \\ \hline 10110\ 1\ 0000\ b = a + 1. \\ \qquad\quad\ k \end{array}$$

We want to write a formula $\gamma(n)$ which forces a counter to take on all values from 0 to $2^n - 1$, in consecutive states, thereby forcing an exponentially deep path. We take care that the formula has length only $O(n^2)$. The formula $\gamma(n)$

will be a conjunction of four formulas. The first conjunct expresses the fact that the counter is initially set to 0:

$$\neg p_0 \wedge \ldots \wedge \neg p_{n-1}.$$

The other conjuncts of $\gamma(n)$ should hold globally in a model. For that aim we use the dual $[\,*\,]$ of $\langle * \rangle$ ($[\,*\,]\varphi$ abbreviates $\neg\langle * \rangle\neg\varphi$). It is clear that for s the root of a tree model, we have that $\mathcal{M}, s \Vdash [\,*\,]\varphi$ if and only if for all t, $\mathcal{M}, t \Vdash \varphi$.

The second conjunct expresses that every state has a successor:

$$[\,*\,]\Diamond\top.$$

The next two conjuncts take care of addition. They express that whenever an R–transition is made in the model the binary counter is increased by one. First the simple case of resetting the counter:

$$[\,*\,]((p_0 \wedge \ldots \wedge p_{n-1}) \to \Box(\neg p_0 \wedge \ldots \wedge \neg p_{n-1})).$$

Finally, the last conjunct of $\gamma(n)$ covers the case when we have to 'carry one'. This conjunct will itself be a conjunction, having a conjunct of the following form for every k such that $0 \leq k < n$:

$$[\,*\,]((\neg p_k \wedge \bigwedge_{j<k} p_j) \to \Box(p_k \wedge \bigwedge_{j<k} \neg p_j) \wedge \bigwedge_{i>k} \mathit{store}(p_i)),$$

with $\mathit{store}(p_i)$ as defined in the previous subsection and the empty conjunction set to true.

We leave it to the reader to check the correctness of this formula. Note that the sole use of $[\,*\,]$ was to make statements in the *basic* modal language true *everywhere* in the model. This use is crucial however: Proposition 3 states that a formula in the basic modal language can only force models with R–paths at most its modal depth. Now the modal depth of $\gamma(n)$ is just two (one for $[\,*\,]$, one for $\Diamond$), *for every n*, while the minimal R–depth of models satisfying $\gamma(n)$ is 2^n.

Complexity bounds. In the previous subsection we saw that polynomially bounded depth of models was the key to a PSPACE upper bound. The present result does not yet show that such an upper bound is not possible, but renders it unlikely (see the notes). And indeed, the satisfiability problem for $\mathbf{K}^*$ is EXPTIME-*complete.* For the lower bound we refer to the notes. The upper bound follows from the $\mathbf{K}^*$ decision algorithm presented in the previous section. Recall that the algorithm tried to construct a set of ready subsets of $Cl(\xi)$. We remarked already that the construction will terminate after $2^{O(|\xi|)}$ stages. Computing which sets in S_i are ready can be done in time polynomial in the size of S_i, which is at most exponential in $|\xi|$. Thus the whole construction can be carried out in deterministic exponential time.

We note that all results go through for the modal system $\mathbf{K}$+E (the basic modal system expanded with the universal modality E). In particular, the formula $\gamma(n)$ with $\neg\langle * \rangle\neg$ substituted by $\neg\mathsf{E}\neg$ causes an exponentially deep path. This result will be used in the next section.

5.4 NEXPTIME

We now consider a modal system in which matters get even worse: the system $\mathbf{S5^2}$. For the definitions of $\mathbf{S5^2}$ and its square and pseudo-square models, see Section 4.1.

This system does not have the tree model property. Besides that, the language has global expressive power: for every φ, if a model $\mathcal{M}$ satisfies $\Box_1\Box_0\varphi$, then $\mathcal{M} \Vdash \varphi$. So we expect that the satisfiability problem has high complexity. Indeed this is the case. This system is strong enough to interpret the system $\mathbf{K} + \mathsf{E}$ from the previous subsection, inheriting its EXPTIME lower bound. But $\mathbf{S5^2}$ lacks the tree-like models of $\mathbf{K} + \mathsf{E}$ on which the EXPTIME upper bound is based. We will sketch that $\mathbf{S5^2}$ is strong enough to force exponential grids, which is the key to a nondeterministic exponential time lower bound result. A matching upper bound follows from earlier results: Every satisfiable formula ξ is satisfiable in a pseudo-square of size at most $2^{|\xi|}$, by Proposition 4. Being a pseudo-square is a first order property. As we saw in the subsection on NP, testing if a modal formula is satisfied in a model takes time polynomial in the formula and the size of the model. Thus for the same reasons as the polysize model property leads to an NP upper bound, we obtain here an NEXPTIME upper bound.

We start with the interpretation of the $\mathbf{K} + \mathsf{E}$ satisfiability problem. For that we use a translation reminiscent of the 2-variable version of the standard translation (read r as Rxy, $\Diamond_1$ as $\exists y$ and $\Diamond_0$ as $\exists x$ and w as expressing that $x = y$). Let $(\cdot)^t$ be a translation function which maps propositional variables to propositional variables, commutes with the booleans and translates the diamonds as follows:

$$\begin{aligned}(\Diamond\varphi)^t &= \Diamond_1(r \wedge \Diamond_0(w \wedge \varphi^t))\\ (\mathsf{E}\varphi)^t &= \Diamond_1\Diamond_0(w \wedge \varphi^t),\end{aligned}$$

in which r and w are fixed variables not occurring in the input language. Their function becomes clear in the proof of the next proposition.

Proposition 9. *Let ξ be a formula in the basic modal language expanded with the universal modality. Then ξ is $\mathbf{K} + \mathsf{E}$–satisfiable if and only if $w \wedge \xi^t$ is $\mathbf{S5^2}$–satisfiable.*

Proof. ($\Rightarrow$) Let $\mathcal{M} = (W, R, V)$ be a Kripke model and let $\mathcal{M}, s \Vdash \xi$. Define a square $\mathbf{S5^2}$–model $\mathcal{M}^* = (W \times W, \equiv_0, \equiv_1, V^*)$ with V^* defined as follows

$$\begin{aligned}V^*(w) &= \{(x, x) \mid x \in W\}\\ V^*(r) &= R\\ V^*(p) &= \{(x, x) \mid x \in V(p)\}.\end{aligned}$$

An easy induction shows that $\mathcal{M}^*, (s, s) \Vdash w \wedge \xi^t$.
($\Leftarrow$) Let $\mathcal{M} = (W, \equiv_0, \equiv_1, V)$ be a square $\mathbf{S5^2}$ model and $\mathcal{M}, (s, t) \Vdash w \wedge \xi^t$. Define a Kripke model $\mathcal{M}^\circ$ whose domain consists of all pairs in W where w holds; of which the valuation V° is simply the restriction of V to these w–pairs; and in which the accessibility relation R° holds between (x, y) and (x', y') iff $\mathcal{M}, (x, y') \Vdash r$. A simple induction shows that for all w-pairs (x, y), for all formulas φ we have that $\mathcal{M}, (x, y) \Vdash \varphi^t \iff \mathcal{M}^\circ, (x, y) \Vdash \varphi$. ◀

This result immediately shows that the $\mathbf{S5^2}$ satisfiability problem is EXPTIME–hard. In fact it is even hard for NEXPTIME. This lower bound can be shown by a reduction to a tiling problem very similar to the one used to show undecidability in Section 4. In this case we don't tile the grid $\mathbb{N}\times\mathbb{N}$ but the finite grid $2^n \times 2^n$. It is known that this problem is complete for nondeterministic time exponential in n. Here we provide the key idea underlying the reduction, which is that for every n, we can define a satisfiable formula $\xi(n)$ with the properties that

- the length of $\xi(n)$ is quadratic in n, and
- if $\xi(n)$ is satisfied in an $\mathbf{S5^2}$ model $\mathcal{M}$, then $\mathcal{M}$ contains as a substructure an isomorphic copy of the structure $(2^n \times 2^n, S_v, S_h)$ (where S_v, S_h are the vertical and horizontal successors functions in the grid $2^n \times 2^n$).

Once we have expressed this, it is straightforward to find a formula saying that a tiling exists, just as in Section 4. Because of space limitations we can only give a very rough sketch. The first conjunct of $\xi(n)$ is the translation of the formula $\gamma(2n)$ from the previous subsection. We assume that $\gamma(2n)$ is created from variables $x_0, \ldots, x_{n-1}$ and $y_0, \ldots, y_{n-1}$. So there are two binary counters together specifying in binary notation a pair $\langle k, l\rangle$ in the grid $2^n \times 2^n$.

Let $\mathcal{M}$ be a Kripke model such that $\mathcal{M}, s \Vdash \gamma(2n)$. Let $\mathcal{M}^*$ be the square $\mathbf{S5^2}$ model as defined in the proof of Proposition 9. In $\mathcal{M}^*$ we have besides the tree structure of the Kripke model also all "grid points": that is for all worlds $w, w' \in W$, the pair (w, w') exists in the model $\mathcal{M}^*$. We can use these pairs to relate the counter information in w and w'. More concretely, we will write formulas ensuring that at (w, w') a propositional variable S_v holds if and only if w encodes the grid pair $\langle k, l\rangle$ and w' its vertical successor $\langle k, l+1\rangle$. We again use the characterization of adding one in binary in order to create a formula of the required small size. For the full proof, we refer to the notes.

We have seen how the EXPTIME modal logic $\mathbf{K} + \mathsf{E}$ "lives" inside the NEXPTIME logic $\mathbf{S5^2}$ in Proposition 9. Also we saw that the "extra" points available in the grid models lead to higher complexity. In the next section, we do the same but with more variables. We look at first order logic and find decidable fragments living inside it. In analogy with the last result we can say that the key feature of these fragments is that they cannot speak about the "extra grid points".

5.5 Notes

Most complexity-theoretic classifications of modal satisfiability and validity problems come from the computer science literature. This work can roughly be divided into three groups: temporal logics describing computations, logics for reasoning about knowledge, and description logics. Pointers to this vast literature can be found in the handbook articles Stirling [55] and Calvanese et alii [13] for temporal logics and description logics, respectively, and in the monograph Fagin et alii [17] for epistemic logics. Here in our notes we only provide the sources for the results in this section.

The NP-completeness of **S5** was proved in Ladner [36]. The results on PSPACE come from Ladner [36]: both the upper and lower bound for the basic modal system **K** are established there. His procedure for **K** is like the one given in Figure 6, save that Ladner uses 'concrete tableaux' (that is, his algorithm specifies how to construct the required atoms) rather than 'abstract tableaux' (which factor out the required boolean reasoning). Concrete tableaux are also used by Halpern and Moses [23] to construct PSPACE algorithms for multimodal versions of **K**, **S4** — and indeed **S5**; as they show, logics containing two **S5** modalities are PSPACE-hard.

The EXPTIME-hardness of $\mathbf{K}^*$ and $\mathbf{K}+\mathsf{E}$ is due to Fischer & Ladner [18] (they work in the richer setting of Propositional Dynamic Logic). An EXPTIME procedure for **PDL** using an "elimination algorithm", is due to Pratt [49]. We used this idea in the proof of Theorem 6, and will do so again in the next section on the guarded fragment. For other applications of the method see e.g., Halpern & Moses [23] (who apply it to a multimodal logic equipped with a common knowledge operator).

The modal system $\mathsf{Func}+\mathsf{E}$ is an example of a modal system in which exponentially deep paths can be forced but which is still decidable in PSPACE, a result by Sistla & Clarke [53]. We stress that adding the universal modality causes the complexity to go up from NP. Note that in this system there are no models which can be considered to be binary trees. Indeed, Spaan [54] provides a sufficient condition for EXPTIME–hardness of the satisfiability problem of modal systems. This criterion asks for the existence of models which can be considered as finite binary trees, and an expansion of the basic modal language which is powerful enough to make statements which hold everywhere in such a tree model.

The fact that the square tiling problem with the width given in binary is hard for non–deterministic exponential time is established in Fürer [19]. The lower bound for $\mathbf{S5^2}$ is established in Marx [39].

6 Modal logic and first order logic

The previous sections were centered around the question, what determines the decidability and complexity of the satisfiability problem for various modal systems. We identified the looseness property of modal logics as the main principle guiding their nice computational properties; we also met two locality principles influencing the complexity of a modal system. It then seems natural to try and see how far we can push these ideas concerning looseness and locality to larger fragment of first order logic than the modal fragment formed by the range of the standard translation map. This task we will undertake now: the aim of this section is to identify a number of decidable fragments of first order logic; that is, sets of first order formulas for which it is decidable whether a given formula in the subset is satisfiable in some first order model or not.

Convention. We work in a relational first order language with equality. Thus the language contains neither constants nor function symbols. For a sequence of variables $\overline{x} = x_1, \ldots, x_n$, we frequently write $\exists\overline{x}\varphi$, which, as usual, has the same meaning as $\exists x_1 \cdots \exists x_n \varphi$. However, we view $\exists\overline{x}$ not as an abbreviation, but as a primitive operator. In particular this means that the subformulas of $\exists\overline{x}\varphi$ are just $\{\exists\overline{x}\varphi\}$, together with the subformulas of φ. By writing $\varphi(\overline{x})$ we indicate that the free variables of φ are among $x_1, \ldots, x_n$.

6.1 Guarded fragments

In order to find larger 'loose' fragments of first order logic, we reconsider the game-theoretic characterization of the modal fragment of first order logic. Recall that bisimulations can be defined using a certain two–pebble Ehrenfeucht–Fraïssé game in which the universal player's moves are restricted in a certain way. We analyze these restrictions and implement them in the standard Ehrenfeucht–Fraïssé games for first order logic; then we are ready to push all modal decidability arguments through for these so called *guarded fragments*.

Consider once again the bisimulation game from Section 3, and the two crucial properties

Locality The game is played by moving a window of fixed size (two, in this case) across the models.

Looseness The window can only be placed on parts of the model in which all different points are related by the accessibility relation.

How to generalize this to first order logic? We will implement the locality principle by considering fragments of first order logic using a fixed finite number of variables. The looseness principle can be generalized in (at least) two different ways, leading to different fragments of first order logic. To state these generalizations we need two notions both of which are well known in finite model theory.

Definition 8. *Let $\mathcal{M} = (D, I)$ be a model for some first order language. A tuple $(a_1, \ldots, a_n)$ of objects in D is called* live *in $\mathcal{M}$ if either $a_1 = \cdots = a_n$ or $(a_1, \ldots, a_n) \in I(P)$ for some predicate symbol P.*

A subset A of D is called guarded *if there is some live tuple $(a_1, \ldots, a_n)$ such that $A \subseteq \{a_1, \ldots, a_n\}$. In particular, singleton sets are always guarded; note also that guarded sets are always finite. A is* packed *or* pairwise guarded *if it is finite and each of its two-element subsets is guarded.*

These notions can help us to incorporate the looseness principle in Ehrenfeucht–Fraïssé games as follows: player $\forall$ can only move pebbles in such a way that all configurations ever to be on the board are placed on guarded or packed sets. Formally:

Definition 9. *Let $\mathcal{M} = (D, I)$ and $\mathcal{M}' = (D', I')$ be two models. A* partial isomorphism *between $\mathcal{M}$ and $\mathcal{M}'$ is a bijection $f : A \to A'$ between some subset A of D and A' of D' such that for all predicate symbols P and all tuples $\overline{a}$ in A (of the appropriate length) we have that $\overline{a} \in I(P)$ if and only if $f(\overline{a}) \in I'(P)$.*

Now for a partial isomorphism $f : A_0 \to A'_0$ between $\mathcal{M}'$ and $\mathcal{M}'$ we define the guarded game $\mathcal{G}_g(\mathcal{M}, \mathcal{M}', f)$ *as a variant of the familiar Ehrenfeucht–Fraïssé game. Here, in each round of the game, $\forall$ selects a structure and a guarded set within that structure; $\exists$ responds with a guarded set in the other structure. A* match *of the game thus gives rise to two sequences $A = A_0, A_1, \ldots$ and $A' = A'_0, A'_0, \ldots$ of subsets of D and D', respectively. $\exists$* wins *this match if there are local isomorphisms $f_n : A_n \to A'_n$ ($n \in \omega$) such that $f_0 = f$ and for each n, f_n and f_{n+1} agree on the intersection $A_n \cap A_{n+1}$ while their inverses agree on $A'_n \cap A'_{n+1}$.*

Now let $\overline{a}$ in $\mathcal{M}$ and $\overline{a}'$ in $\mathcal{M}'$ be (possibly empty) sequences of elements such that $f(a_i) = a'_i$ for all i. In case that $\exists$ has a winning strategy in the guarded game $\mathcal{G}_g(\mathcal{M}, \mathcal{M}', f)$ we say that $\overline{a}$ and $\overline{a}'$ are g-bisimilar.

The packed game $\mathcal{G}_p(\mathcal{M}, \mathcal{M}', f)$ *and the notion of packed bisimilarity are defined in the same way but using packed sets instead of guarded ones.*

These restrictions on the moves of player $\forall$ have direct syntactical counterparts in form of restrictions on *quantification*: the idea is that we will only allow quantification in the form $\exists\overline{x}\varphi$ where φ has to meet certain criteria.

Definition 10. *We say that a formula φ* packs *a set of variables $\{x_1, \ldots, x_k\}$ if φ is a conjunction of formulas of the form $x_i = x_j$ or $R(x_{i_1}, \ldots, x_{i_n})$ or $\exists\overline{y}R(x_{i_1}, \ldots, x_{i_n})$ such that for every $x_i \neq x_j$, there is a conjunct in φ in which x_i and x_j both occur free.*

The packed fragment *PF is defined as the smallest set of first order formulas which contains all atomic formulas and is closed under the boolean connectives and under* packed quantification. *That is, whenever ψ is a packed formula, π packs Free(π), and Free(ψ) $\subseteq$ Free(π), then $\exists\overline{x}(\pi \wedge \psi)$ is packed as well; π is called the* guard *of this formula. The* guarded fragment *GF is the subfragment of PF in which we only allow* guarded quantification*; that is, packed quantification in which the guard π is an* atomic *formula.*

PF_n and GF_n denote the restrictions to n variables and at most n–ary predicate symbols of PF and GF, respectively.

When we want to be specific about the free variables occurring in the formulas we often write $\exists\overline{y}\,(\pi(\overline{x}, \overline{y}) \wedge \psi(\overline{x}, \overline{y}))$ for the quantified packed formulas, tacitly assuming that $\overline{x}$ and $\overline{y}$ do not share any variables.

Typical examples of guarded (thus also packed) sentences are $\forall xy(Rxy \to Ryx)$, $\exists xy\,(Rxy \wedge Ryx \wedge (Rxx \vee Ryy))$ and the standard translation of a formula in the basic modal language (with R functioning as guard). A typical non–example is $\forall xyz((Rxy \wedge Ryz) \to Rxz)$: it is neither guarded nor packed. For an example of a packed formula which is not guarded, consider $\exists xyz((Rxy \wedge Rxz \wedge Ryz) \wedge \neg Cxyz)$.

Note that the notion of packedness only places meaningful restrictions on pairs of *distinct* variables: since the formula $x = x$ packs the set of variables $\{x\}$, the formula $\exists x(x = x \wedge \psi(x))$, (i.e., with a *single* quantification over the variable x) is a packed formula, at least, provided that $\psi(x)$ is packed. When $\psi(x)$ is guarded, then $\exists x(x = x \wedge \psi(x))$ is also guarded. Since the given formula is equivalent to $\exists x \psi(x)$ this shows that packedness allows a fairly mild form of ordinary quantification, namely only over formulas with one free variable. A nice corollary of this is that we may perform the standard translation of the universal modality E within the two variable guarded fragment:

$$ST_x(\mathsf{E}\varphi) = ST_y(\mathsf{E}\varphi) = \exists x(ST_x(\varphi)) \equiv \exists x(x = x \wedge ST_x(\varphi)).$$

A similar translation to first order logic can be defined for the language with the until modality U. Its range is the packed fragment with three variables. The interesting clause here is

$$ST_x(U(\varphi, \psi)) = \exists y(Rxy \wedge ST_y(\varphi) \wedge \forall z((Rxz \wedge Rzy) \to ST_z(\psi))).$$

This formula is not packed *itself*, because in the subformula $\forall z\,((Rxz \wedge Rzy) \to ST_z(\psi)))$ the guard $Rxz \wedge Rzy$ does not pack its own free variables $\{x, y, z\}$. But of course, the formula is *equivalent* to

$$\exists y\,(Rxy \wedge ST_y(\varphi) \wedge \forall z\,((Rxz \wedge Rzy \wedge Rxy) \to ST_z(\psi)))$$

which is packed. It is not hard to convert this example into a proof showing that *every* formula in the Until language is equivalent to a packed formula. The (adjusted) translation is another example of a packed sentence that is not guarded.

We defined first order fragments by incorporating the restrictions on the moves in an Ehrenfeucht–Fraïssé game into the syntax. It is obvious that packed formulas are preserved when player $\exists$ has a winning strategy. But in fact the fragments precisely *characterize* the formulas which are invariant for the corresponding games.

Definition 11. *A first order formula $\varphi(\overline{x})$ is* invariant for guarded (packed) bisimulation *if for all g–bisimilar (p–bisimilar) tuples $\overline{a}$ in $\mathcal{M}$ and $\overline{a}'$ in $\mathcal{M}'$ we have that $\mathcal{M} \models \varphi[\overline{a}]$ iff $\mathcal{M}' \models \varphi[\overline{a}']$.*

Theorem 8. *Let ξ be a first order formula. Then the following are equivalent.*
(i) ξ is equivalent to a formula in the packed (guarded) fragment.
(ii) ξ is invariant for packed (guarded) bisimulations.

This theorem can be relativized in the usual way to n–variable fragments and the corresponding n–pebble games. This is the first analogue of a modal theorem (the characterization theorem 1). In the section on basic modal logic we saw that this theorem allowed us to prove that every satisfiable formula was satisfiable in a tree. These trees were obtained by unraveling or unwinding the model. Analogue notions of unraveling and tree models can be defined for the guarded and packed fragments as well; here we confine ourselves to the notion of a *loose model.*

Definition 12. *Let $\mathcal{M} = (D, I)$ be a first order structure. We call $\mathcal{M}$ a* loose model *of degree $k \in \mathbb{N}$ if there is some acyclic connected undirected graph $\mathcal{G} =$*

(G, E) and a function f mapping nodes of $\mathcal{G}$ to subsets of D of size not exceeding k such that for every live tuple $\overline{s}$ from $\mathcal{M}$, the set $L(\overline{s}) = \{k \in G \mid s_i \in f(k)$ for all $s_i\}$, is a non-empty and connected subset of $\mathcal{G}$.

In words, we call a model $\mathcal{M} = (D, I)$ loose if we can associate a connected graph $\mathcal{G} = (G, E)$ with it in the following way. Each node t of the graph corresponds to a *small* subset $f(t)$ of the model; a good way of thinking about this is that t 'describes' $f(t)$. One then requires that the graph 'covers' the entire model in the sense that any $a \in D$ belongs to one of these sets (this follows from the fact that for any $a \in D$, the 'tuple' a is live). The fact that each set $L(\overline{a})$ is connected whenever $\overline{a}$ is live, implies that various nodes of the graph will not give contradictory descriptions of the model. Finally, the *looseness* of the model intuitively stems from the acyclicity of $\mathcal{G}$ and the connectedness of the sets $L(\overline{a})$; for, this makes that in walking through the graph we may describe different parts of the model, *but we never have to worry about returning to the same part once we have left it.* Summarizing, we may see the graph as a loose, coherent collection of descriptions of local submodels of the model. Loose models are the ones for which we can find such a graph. Note that the degree of a loose model directly corresponds to the second locality principle that we identified at the end of section 3.

Now we can announce the second modally flavored theorem: it establishes the *loose model property* for the packed fragment.

Theorem 9. *Every satisfiable packed formula ξ can be satisfied on a loose model of degree not exceeding the number of variables occurring in ξ.*

And, as we will see further on, this property indeed plays a crucial role in the proof of the following result.

Theorem 10. *It is decidable whether a packed formula is satisfiable. In fact, the satisfiability problems for both the guarded and the packed fragment are complete for* 2EXPTIME.

The doubly exponential lower bound may raise doubts concerning the relevance of this result. Fortunately, there are some large and very natural fragments for which better bounds may be obtained, and here the second notion of 'locality' comes into play. For, not only the concept of looseness generalizes to these fragments, we can also give analogous versions for the notion of locality. Recall that we introduced this concept when we saw that the basic modal language could be translated into the *two-variable fragment* of first order logic. This suggests that we might try to improve on Theorem 10 by considering finite-variable fragments of PF and GF. And indeed, it turns out that 'bringing locality into the language' brings down the complexity with one exponent!

In the case of the guarded fragment, we can formulate this result in a nice way, by imposing conditions on the first order signature rather than on the number of variables used. Recall that the signature of the modal fragment of first order logic consists of unary relation symbols and one binary symbol. In general, call a first order signature L *n–bounded* if all relation symbols in L have arity at most n. It is not very difficult to see that every *guarded sentence* in an n-bounded signature can be rewritten using only n variables. Thus — just as in the basic modal case — the signature determines the number of variables. Note that for the full packed fragment this property is lost as we can pack arbitrarily large sets with binary relations.

In any case, implementing both looseness and locality in first order logic we may obtain the following result.

Theorem 11. *Fix a natural number n.*

(i) The satisfiability problem for formulas in the packed fragment PF_n is decidable in EXPTIME.

(ii) Hence, the satisfiability problem for sentences in the guarded fragment in the n–bounded signature is decidable in EXPTIME.

Note that for $n \geq 2$ the satisfiability problem for the guarded fragment GF_n is also EXPTIME-hard. This holds by the interpretation of the modal system $\mathbf{K} + \mathsf{E}$ using the standard translation. However, by also implementing the first locality principle of locality — no global expressive power — it is even possible to bring the complexity down to PSPACE, see the notes.

And finally, what about finite models? Several subfragments of the packed fragment, including the guarded fragment, are known to have the finite model property. For the *full* packed fragment this was an open problem at the time of writing[1]. For reasons of space limitations, we cannot go into detail here — see the notes for references.

6.2 Decidability and complexity

This subsection provides the proofs of all the results mentioned above. The main idea behind the proofs is formed by the *mosaic method* that we already met in the decidability proof for the until system. Roughly speaking, this method is based on the idea of deconstructing models into a (modulo isomorphism) finite collection of finite submodels, and conversely, of building up new, 'nice', models from such parts.

The subsection is structured as follows. We start with the formal definition of the notion of mosaic and some related concepts. We then state the main result concerning the mosaic method, namely the *Mosaic Theorem*, stating that a packed formula has a model if and only if there is a bounded set of bounded size mosaics for it. This enables us to define our decision algorithms and establish their complexity. We then continue to prove the Mosaic Theorem. In doing so we obtain as byproduct the loose model property for the packed fragment.

[1] When finishing the final draft of this paper, we received a manuscript by I. Hodkinson announcing a positive solution.

Linked sets of mosaics Mosaics form the key tools in our proof; for a formal definition we need some syntactic preliminaries. Given a first order formula ξ, we let $Var(\xi)$ and $Free(\xi)$ denote the sets of variables and free variables occurring in ξ, respectively. Let V be a set of variables. A *V-substitution* is any partial map $\sigma : V \to V$. The result of performing the substitution σ on the formula ψ is denoted by ψ^σ. (We can and may assume that if $Var(\psi) \subseteq V$ then $Var(\psi^\sigma) \subseteq V$. For instance, when substituting y for x in $Rxz \wedge \forall y\,(Rzy \to Qxy)$ we have to rename the *bound* variable y, as in $Ryz \wedge \forall u\,(Rzu \to Qyu)$ The point is that we do not need to use a *fresh* variable u for this: instead, we may reuse x, giving $Ryz \wedge \forall x\,(Rzx \to Qyx)$.)

As before, we will employ a notion of closure to delineate a finite set of *relevant* formulas, i.e. formulas that for some reason critically influence the truth of a given formula ξ. Also recall that the *single negation* $\sim\varphi$ of a formula φ denotes the formula ψ if φ is of the form $\neg\psi$; otherwise, $\sim\varphi$ is the formula $\neg\varphi$.

Definition 13. *Let Σ be a set of packed formulas in the set V of variables. We call Σ V-*closed *if Σ is closed under subformulas, single negations and V-substitutions (that is, if ψ belongs to Σ, then so does ψ^σ for every V-substitution σ). With $Cl_g(\xi)$ we denote the smallest $Var(\xi)$-closed set of formulas containing ξ.*

For the remainder of this section, we fix a packed formula ξ — all definitions to come should be understood as being relativized to ξ. The number of variables occurring in ξ (free or bound) is denoted by k; that is, k is the size of $Var(\xi)$. It can easily be verified that the sets of guarded and packed formulas both are closed under taking subformulas; hence, the set $Cl_g(\xi)$ consists of guarded (packed, respectively) formulas. An easy calculation shows that the cardinality of $Cl_g(\xi)$ is bounded by $k^k \cdot (2|\xi|)$ (k^k is the number of $Var(\xi)$-substitutions).

The following notion is the counterpart of the maximally propositionally consistent sets that we have met in earlier decidability proofs. The defining conditions again derive from a desire to prove a truth lemma.

Definition 14. *Let $X \subseteq Var(\xi)$ be a set of variables. An X-*type *is a set $\Gamma \subseteq Cl_g(\xi)$ with free variables in X satisfying, for all formulas $\varphi\wedge\psi$, $\sim\varphi$, φ in $Cl_g(\xi)$ with free variables in X, the conditions (T1) $\varphi \wedge \psi \in \Gamma$ iff $\varphi \in \Gamma$ and $\psi \in \Gamma$,*

(T2) $\varphi \notin \Gamma$ iff $\sim\varphi \in \Gamma$,

(T3) $\varphi, x_i = x_j \in \Gamma$ only if $\varphi^\sigma \in \Gamma$ (for any substitution σ mapping x_i to x_j and/or x_j to x_i, while leaving all other variables fixed), and

(T4) if $\psi(\overline{x},\overline{z})$ and $\pi(\overline{x},\overline{z})$ are in Γ, then so is $\exists\overline{y}\,(\pi(\overline{x},\overline{y})\wedge\psi(\overline{x},\overline{y}))$, (provided that the latter formula belongs to $Cl_g(\xi)$).

The next definition introduces our key tool in proving the decidability of the packed fragment: mosaics and linked sets of these. Basically, a mosaic consists of a set X of variables in $Var(\xi)$ and a set Γ encoding the relevant information of some small part of a model. Here 'small' means that its size is bounded by the number of objects that can be named using variables in X, and 'relevant' refers to all formulas in $Cl_g(\xi)$ whose free variables are in X. It turns out that a finite set of such mosaics contains sufficient information to construct a model for ξ provided that the set links the mosaics together in a nice way. Here is a more formal definition.

Definition 15. *A* mosaic *is a pair* (X, Γ) *such that* $X \subseteq Var(\xi)$ *and* $\Gamma \subseteq Cl_g(\xi)$. *A mosaic* (X, Γ) *is* coherent *if* Γ *is an* X*-type.*

A link *between two mosaics* (X, Γ) *and* (X', Γ') *is a renaming (that is, an injective substitution)* σ *with* $\mathrm{dom}\sigma \subseteq X$ *and* $\mathrm{ran}\sigma \subseteq X'$ *which satisfies, for all formulas* $\varphi \in Cl_g(\xi)$: $\varphi \in \Gamma$ *iff* $\varphi^\sigma \in \Gamma'$.

A requirement *of a mosaic is a formula of the form* $\varphi(\overline{x}) = \exists \overline{y}\, (\pi(\overline{x}, \overline{y}) \wedge \psi(\overline{x}, \overline{y}))$ *belonging to* Γ. *A mosaic* (X', Γ') fulfills *the* requirement $\exists \overline{y}\, (\pi(\overline{x}, \overline{y}) \wedge \psi(\overline{x}, \overline{y}))$ *of a mosaic* (X, Γ) via *the link* σ *if for some variables* $\overline{u}$, $\overline{v}$ *in* X' *we have that* $\sigma(\overline{x}) = \overline{u}$ *and* $\pi(\overline{u}, \overline{v})$ *and* $\psi(\overline{u}, \overline{v})$ *belong to* Γ'. *A set* S *of mosaics is* linked *if every requirement of every mosaic in* S *is fulfilled via some link to some mosaic in* S. S *is a linked set of mosaics for* ξ *if it is linked and* $\xi \in \Gamma$ *for some* (X, Γ) *in* S.

Note that a mosaic (X, Γ) may fulfill its own requirements, either via the identity map or via some other map from X to X.

The key result concerning mosaics is the following Mosaic Theorem.

Theorem 12 (Mosaic Theorem). *Let* ξ *be a packed formula. Then* ξ *is satisfiable if and only if there is a linked set of mosaics for* ξ.

Proof. The hard, right to left, direction of the theorem is proved in Lemma 1 below; here we only prove the other direction.

Suppose that ξ is satisfied in the model $\mathcal{M} = (D, I)$. In a straightforward way we can 'cut out' from $\mathcal{M}$ a linked set of mosaics for ξ. Consider the set of partial assignments of elements in D to variables in $Var(\xi)$. For each such α, let $(X_\alpha, \Gamma_\alpha)$ be the mosaic given by $X_\alpha = \mathrm{dom}\alpha$ and

$$\Gamma_\alpha = \{\varphi \in Cl_g(\xi) \mid \mathcal{M} \models \varphi[\alpha]\}.$$

We leave it to the reader to verify that this collection forms a linked set of mosaics for ξ. ◀

When establishing the hard direction of this proposition we will in fact prove something stronger: starting from a linked set of mosaics for a formula ξ we will show that there is a *loose* or *tree-like* model for ξ.

First however, we want to show that the Mosaic Theorem is the key towards proving the decidability of the packed fragment, and also for finding an upper bound for its complexity.

The decision algorithm and its complexity

The Mosaic Theorem tells us that any packed formula ξ is satisfiable if and only if there is a linked set of mosaics for ξ. Thus in order to decide whether ξ is satisfiable, it suffices to give an algorithm which decides the existence of a linked set of mosaics exists for ξ. We establish the upper complexity bound for the satisfiability problem of packed formulas by implementing such an algorithm. The following observations are easy consequences of our definitions; recall that k denotes the number of variables occurring in ξ.

- We already observed that the cardinality of $Cl_g(\xi)$ is bounded by $k^k \cdot 2|\xi|$.
- The number of mosaics does not exceed $2^k \cdot 2^{2|\xi| \cdot k^k}$; using the big O notation, this gives at most $2^{O(|\xi|) \cdot k^k}$ mosaics.
- given sets X, Γ with $X \subseteq Var(\xi)$ and $\Gamma \subseteq Cl_g(\xi)$ it is decidable in time $2^{O(|\xi|) \cdot k^k}$ whether (X, Γ) is a coherent mosaic.

Our algorithm is very similar to the one we used for the until system in subsection 4.3. Let S_0 be the set of all coherent mosaics. By the observations above, S_0 contains less than $2^{O(|\xi|) \cdot k^k}$ elements and can be constructed in time $2^{O(|\xi|) \cdot k^k}$. We now inductively construct a sequence of sets of mosaics $S_0 \supsetneq S_1 \supsetneq S_2 \supsetneq S_3 \cdots$, as follows. We call a mosaic μ in a set S_i *S_i-ready* if each of its requirements is fulfilled in (some mosaic of) S_i. Note that one can determine the S_i-readiness of a mosaic (X, Γ) by checking for each requirement $\varphi(\overline{x}) \in \Gamma$ whether there is a link σ to some mosaic $(X', \Gamma') \in S_i$ which fullfills the requirement. Now if every mosaic μ in S_i is S_i-ready then return 'YES' if S_i contains a mosaic (X, Γ) with $\xi \in \Gamma$ and 'NO' if S_i contains no such mosaic. If on the other hand there are mosaics in S_i that are not S_i-ready, then let S_{i+1} consist of the S_i-ready mosaics and continue the algorithm.

Clearly the algorithm is correct; and since $S_i \supsetneq S_{i+1}$, the construction must halt after at most $|S_0|$ many stages. So let's now see about the complexity. At each stage i the algorithm determines the S_i-ready mosaics; we claim that this can be done in time exponential in $k^k \cdot O(|\xi|)$.

To check whether a given link between two given mosaics fullfills some given requirement, is a task that takes time linear in the size of each mosaic, so time quadratic in $k^k \cdot 2|\xi|$. In order to find out whether a given mosaic (X, Γ) in a set S_i is S_i-ready, the algorithm has to check for every requirement $\varphi(\overline{x})$ of the mosaic, for every link σ, and for every mosaic (X', Γ') in S_i whether σ is a link between the mosaics fullfilling the requirement. Clearly then, for a given mosaic this takes time at most $k^k \cdot 2|\xi|$ (for the number of requirements) times k^k (for the number of links) times $|S_i|$ (for the number of mosaics) times $(k^k \cdot 2|\xi|)^2$ (for the checking time). Note that S_i is the only number in this product that is exponential in $k^k \cdot O(|\xi|)$. Hence, in order to compute all the S_i–ready mosaics the algorithm needs time exponential in $O(|\xi|) \cdot k^k$.

As the size of S_0 is bounded by $2^{O(|\xi|) \cdot k^k}$, the whole computation can be performed in time exponential in $O(|\xi|) \cdot k^k$. Hence, if we consider a formula ξ in a packed fragment with a *fixed number of variables*, $|S_0|$ is singly exponential in $|\xi|$. In general however, the number of variables k occurring in a formula depends on the formula's length and hence in general, $|S_0|$ is doubly exponential in $|\xi|$. *Thus pending the correctness of the next Lemma this shows the upper complexity bounds in Theorems 10 and 11.*

Step–by–step construction and loose models

We show the hard direction of the Mosaic Theorem and establish as a byproduct the 'loose model property' from Theorem 9.

Lemma 1. *Let ξ be a packed formula. If there is a linked set of mosaics for ξ, then ξ is satisfiable in a loose model of degree $|Var(\xi)|$.*

Proof. Assume that S is a linked set of mosaics for ξ. Using a step-by-step construction, we will build up a model for ξ, together with the graph $\mathcal{G}$ and the function f mapping nodes of $\mathcal{G}$ to subsets of the domain of the model. At each stage of the construction we will be dealing with some kind of approximation of the final model and graph; these approximations will be called networks and are slightly involved structures.

A *network* is a quintuple $(\mathcal{M}, \mathcal{G}, \mu, \alpha, \sigma)$ such that $\mathcal{M} = (D, I)$ is a model for the first order language; $\mathcal{G} = (G, E)$ is a connected, adirected and acyclic graph; $\mu : G \to S$ is a map associating a mosaic $\mu_t = (X_t, \Gamma_t)$ in S with each node t of the graph; α is a map associating a map $\alpha_t : X_t \to D$ with each node t of the graph. (This map is thus a partial assignment of the variables occurring in ξ.) And finally, σ is a map associating with each edge (t, t') of the graph a link $\sigma_{tt'}$ from μ_t to $\mu_{t'}$ (we will usually simplify our notation by writing σ instead of $\sigma_{tt'}$).

The idea is that each mosaic μ_t is supposed to give a complete description of the relevant requirements that we impose on a small part of the model-to-be. Which part? This is given by the assignment α_t. And the word 'relevant' refers to the fact that we are only interested in the formulas influencing the truth of ξ; that is, the formulas in $Cl_g(\xi)$. The links between neighboring mosaics are there to ensure that distinct mosaics agree on the part of the model that they both have access to.

Now obviously, if we want all of this to work properly we have to impose some conditions on networks. In order to formulate these, we need some auxiliary notation. For a subset $A \subseteq D$, let $L(A)$ denote the set of nodes in $\mathcal{G}$ that have 'access' to A; formally, we define $L(A) = \{t \in G \mid A \subseteq \mathrm{ran}(\alpha_t)\}$. For a tuple $\overline{a} = (a_1, \ldots, a_n)$ of elements in D we set $L(\overline{a}) = L(\{a_1, \ldots, a_n\})$. Now a network is called *coherent* if it satisfies the following conditions (all to be read universally quantified):

(C1) $P\overline{x} \in \Gamma_t$ iff $\mathcal{M} \models P\overline{x}[\alpha_t]$,
(C2) $x_i = x_j \in \Gamma_t$ iff $\alpha_t(x_i) = \alpha_t(x_j)$,
(C3) $L(A)$ is non-empty for every guarded set $A \subseteq D$,
(C4) $L(A)$ is connected for every guarded set $A \subseteq D$,
(C5) if Ett' then $\sigma_{tt'}(x) = x'$ iff $\alpha_t(x) = \alpha_{t'}(x')$.

A few words of explanation about these conditions: (C1) and (C2) ensure that every mosaic is a complete description of the atomic formulas holding in the part of the model it refers to. Condition (C3) states that no guarded set in the model remains unseen from the graph, while the conditions (C4) and (C5) are the crucial ones making that remote parts of the graph cannot contain contradictory information about the model — how this works precisely will become clear further on. Note that condition (C5) has two directions: the left-to-right direction states that neighboring mosaics have common access to part of the model, while the other direction makes them agree on their requirements concerning this common part.

The motivation for using these networks is that in the end we want any formula $\varphi(\overline{x}) \in Cl_g(\xi)$ to hold in $\mathcal{M}$ under the assignment α_t if and only $\varphi(\overline{x})$ belongs to Γ_t. Coherence on its own is not sufficient to make this happen. A *defect* of a network consists of a formula $\exists\overline{y}\,(\pi(\overline{x},\overline{y}) \wedge \psi(\overline{x},\overline{y}))$ which is a requirement of the mosaic μ_t for some node t while there is no neighboring node t' such that $\mu_{t'}$ fulfills $\exists\overline{y}\,(\pi(\overline{x},\overline{y}) \wedge \psi(\overline{x},\overline{y}))$ via the link $\sigma_{tt'}$. A coherent network $\mathcal{N}$ is *perfect* if it has no defects. We say that $\mathcal{N}$ is a network for ξ if for some $t \in G$, $\mu_t = (X_t, \Gamma_t)$ is such that $\xi \in \Gamma_t$.

Claim 1 If $\mathcal{N} = (\mathcal{M}, \mathcal{G}, \mu, \alpha, \sigma)$ is a perfect network, then

(i) $\mathcal{M}$ is a loose model of degree $|\,Var(\xi)|$, and

(ii) for all formulas $\varphi(\overline{x}) \in Cl_g(\xi)$ and all nodes t of $\mathcal{G}$:

$$\varphi \in \Gamma_t \text{ iff } \mathcal{M} \models \varphi[\alpha_t].$$

Proof of Claim For part (i) of the claim, let $\mathcal{N} = (\mathcal{M}, \mathcal{G}, \mu, \alpha, \sigma)$ be the perfect network for ξ. Let $\mathcal{M} = (D, I)$. As the function f mapping nodes of $\mathcal{G}$ to *subsets* of D, simply take the map that assigns the *range* of α_t to the node t. Since the domain of each map α_t is always a subset of $Var(\xi)$, it follows immediately that $f(t)$ will always be a set of size at most $|\,Var(\xi)|$. Now take an arbitrary live tuple $\overline{s}$ in $\mathcal{M}$; it follows from (C3) and (C4) that $L(\overline{s})$ is a non-empty and connected part of the graph $\mathcal{G}$. Thus $\mathcal{M}$ is a loose model of degree $|\,Var(\xi)|$.

We prove part (ii) of the claim by induction on the complexity of φ. For atomic formulas the claim follows by conditions (C1) and (C2), and the boolean case of the induction step is straightforward (since Γ_t is an X–type) and left to the reader. We concentrate on the case that $\varphi(\overline{x})$ is of the form $\exists\overline{y}\,(\pi(\overline{x},\overline{y}) \wedge \psi(\overline{x},\overline{y}))$.

First assume that $\varphi(\overline{x}) \in \Gamma_t$. Since $\mathcal{N}$ is perfect there is a node t' in G and variables $\overline{u}$, $\overline{v}$ in $X_{t'}$ such that Ett', $\pi(\overline{u},\overline{v})$ and $\psi(\overline{u},\overline{v})$ belong to $\Gamma_{t'}$, while the link σ from μ_t to $\mu_{t'}$ maps $\overline{x}$ to $\overline{u}$. By the induction hypothesis we find that

$$\mathcal{M} \models \pi(\overline{u},\overline{v}) \wedge \psi(\overline{u},\overline{v})[\alpha_{t'}]. \tag{13}$$

But from condition (C5) it follows that $\alpha_{t'}(\overline{x}) = \alpha_t(\overline{u})$, whence (13) implies that

$$\mathcal{M} \models \exists\overline{y}\,(\pi(\overline{x},\overline{y}) \wedge \psi(\overline{x},\overline{y}))[\alpha_t],$$

which is what we were after.

Now suppose, in order to prove the converse direction, that $\mathcal{M} \models \varphi(\overline{x})[\alpha_t]$. Let $\overline{a}$ denote $\alpha_t(\overline{x})$, then there are $\overline{b}$ in D such that $\mathcal{M} \models \pi(\overline{x},\overline{y})[\overline{a}\overline{b}]$ and $\mathcal{M} \models \psi(\overline{x},\overline{y})[\overline{a}\overline{b}]$. Our first aims are to prove that

$$L(\overline{a}\overline{b}) \neq \varnothing, \tag{14}$$

and

$$L(A) \text{ is connected for every } A \subseteq \{\overline{a}, \overline{b}\}. \tag{15}$$

Note that if we are working in the guarded fragment, then $\pi(\overline{x}, \overline{y})$ is an atomic formula, whence it follows from $\mathcal{M} \models \pi(\overline{x}, \overline{y})[\overline{a}\overline{b}]$ that $\overline{a}\overline{b}$ is live. Thus $\{\overline{a}, \overline{b}\}$ is guarded, and hence (14) is immediate by condition (C3). In fact, *every* $A \subseteq \{\overline{a}, \overline{b}\}$ is guarded in this case, so (15) is immediate by condition (C4).

In the more general case of the packed fragment we have to work a bit harder. First, observe that it *does* follow from $\mathcal{M} \models \pi(\overline{x}, \overline{y})[\overline{a}\overline{b}]$ and the conditions on $\pi(\overline{x}, \overline{y})$ in the definition of packed quantification, that $\{c, d\}$ is guarded, and thus, $L(c, d) \neq \varnothing$, for every *pair* (c, d) of points taken from $\overline{a}\overline{b}$. It follows from (C4) that $\{L(c, d) \mid c, d \text{ taken from } \overline{a}\overline{b}\}$ is a collection of non-empty, connected, pairwise overlapping subgraphs of the acyclic graph $\mathcal{G}$. It is fairly straightforward to prove, for instance, by induction on the size of the graph $\mathcal{G}$, that any such collection must have a non-empty intersection. From this, (14) and (15) are almost immediate.

We may thus assume the existence of a node t' in $\mathcal{G}$ such that $\{\overline{a}, \overline{b}\} \subseteq \mathrm{ran}\alpha_{t'}$. Let $\overline{u}$ and $\overline{v}$ in $X_{t'}$ be the variables such that $\alpha_{t'}(\overline{u}) = \overline{a}$ and $\alpha_{t'}(\overline{v}) = \overline{b}$. The induction hypothesis implies that $\pi(\overline{u}, \overline{v})$ and $\psi(\overline{u}, \overline{v})$ belong to $\Gamma_{t'}$, whence $\varphi(\overline{u}) \in \Gamma_{t'}$ by coherence of $\mu_{t'}$. Since both t and t' belong to $L(\overline{a})$, it follows from (15) that there is a path from t to t' *within* $L(\overline{a})$, say $t' = s_0 E s_1 E \ldots E s_n = t$. Let σ_i be the link between the mosaics of s_i and s_{i+1}, and define ρ to be the composition of these maps. It follows by an easy inductive argument on the length of the path that ρ is a link between $\mu_{t'}$ and μ_t such that $\rho(\overline{u}) = \overline{x}$. Hence, by definition of a link we have that $\varphi(\overline{x}) \in \Gamma_{t'}$. ⊣

By Claim 1, in order to prove the Lemma it suffices to construct a perfect network for ξ. This construction uses a step-by-step argument; to start the construction we need *some* coherent network for ξ.

Claim. There is a coherent network for ξ.

PROOF OF CLAIM By our assumption on ξ there is a coherent mosaic $\mu = (X, \Gamma)$ such that $\xi \in \Gamma$. Without loss of generality we may assume that X is the set $\{x_1, \ldots, x_n\}$ (otherwise, take an isomorphic copy of μ in which X does have this form). Let $a_1, \ldots, a_n$ be a list of objects such that for all i and j we have that $a_i = a_j$ if and only if the formula $x_i = x_j$ belongs to Γ. Define $D = \{a_1, \ldots, a_n\}$ and put the tuple $(a_{i_1}, \ldots, a_{i_k})$ in the interpretation $I(P)$ of the k-ary predicate symbol P precisely if $P x_{i_1} \ldots x_{i_n} \in \Gamma$. Let $\mathcal{M}$ be the resulting model (D, I) and define $\mathcal{G}$ as the trivial graph with one node 0 and no edges. Let $\mu(0)$ be the mosaic μ; $\alpha_0 : X \to D$ is given by $\alpha(x_i) = a_i$; and finally, σ_{00} is the identity map from X to X.

We leave it to the reader to verify that the quintuple $(\mathcal{M}, \mathcal{G}, \mu, \alpha, \sigma)$ is a coherent network for ξ. ⊣

The crucial step of this construction will be to show that any defect of a coherent network can be repaired.

Claim. For any coherent network $\mathcal{N} = (\mathcal{M}, \mathcal{G}, \mu, \alpha, \sigma)$ and any defect of $\mathcal{N}$ there is a coherent network $\mathcal{N}^+$ extending $\mathcal{N}$ and lacking this defect.

PROOF OF CLAIM Suppose that $\varphi(\overline{x})$ is a defect of $\mathcal{N}$ because it is a requirement of the mosaic μ_t and not fulfilled by any neighboring mosaic $\mu_{t'}$. We will define an extension $\mathcal{N}^+$ of $\mathcal{N}$ in which this defect is repaired.

Since S is a linked set of mosaics and μ_t belongs to S, μ_t is linked to a mosaic $(X', \Gamma') \in S$ in which the requirement is fulfilled via some link ρ. Let Y be the set of variables in X' that do not belong to the range of ρ; suppose that $Y = \{y_1, \ldots, y_k\}$ (with all y_i being distinct). For the sake of a smooth presentation, assume that Γ' contains the formulas $\neg x' = y$ for all variables $x' \in X'$ and $y \in Y$ (this is not without loss of generality — we leave the general case as an exercise to the reader). Take a set $\{c_1, \ldots, c_k\}$ of fresh objects (that is, no c_i is an element of the domain D of $\mathcal{M}$), and let γ be the assignment with domain X' defined as follows:

$$\gamma(x') \;=\; \begin{cases} \alpha_t(x) & \text{if } x' = \rho(x), \\ c_i & \text{if } x' = y_i, \end{cases}$$

and let t' be an object not belonging to G. Now define the network $\mathcal{N}^+ = (\mathcal{M}^+, \mathcal{G}^+, \mu^+, \alpha^+, \sigma^+)$ as follows:

$$\begin{aligned} D^+ &= D \cup \{c_1, \ldots, c_k\}, \\ I^+(P) &= I(P) \cup \{\overline{d} \mid \text{ for some } \overline{x},\, \overline{d} = \gamma(\overline{x}) \text{ and } P\overline{x} \in \Gamma'\}, \\ G^+ &= G \cup \{t'\}, \\ E^+ &= E \cup \{(t, t')\}, \end{aligned}$$

while μ^+, α^+ and σ^+ are given as the obvious extensions of μ, α and σ, namely by putting $\mu^+_{t'} = (X', \Gamma')$, $\alpha^+_{t'} = \gamma$ and $\sigma_{tt'} = \rho$.

Since the interpretation I^+ agrees with I on 'old' tuples it is a straightforward exercise to verify that the new network $\mathcal{N}^+$ satisfies the conditions (C1) – (C3) and (C5).

In order to check that condition (C4) holds, take some guarded subset A from D^+; we will show that $L^+(A)$ is a connected subgraph of $\mathcal{G}^+$. It is rather easy to see that $L^+(A)$ is identical to either $L(A)$ or $L(A) \cup \{t'\}$; hence by the connectedness of $L(A)$ it suffices to prove, on the assumptions that $t' \in L^+(A)$ and $L(A) \neq \varnothing$, that $t \in L(A)$. Hence, suppose that $t' \in L^+(A)$; that is, each $a \in A$ is in the range of γ. But if $L(A) \neq \varnothing$, each such point a must be old; hence, by definition of γ, each $a \in A$ must belong to $\mathrm{ran}\alpha_t$. This gives that $t \in L(A)$, as required. ⊣

As in the proof for the until-system, the previous two claims show that using some standard combinatorics we can construct a chain of networks such that their *limit* is a perfect network. This finishes the proof of the Lemma. ◀

6.3 Notes

The roots of the decidability proof in this section date back to 1986, when Németi [44] showed that the equational theory of the class Crs of relativised cylindric set algebras is decidable. The first–order counterpart of this result is that a certain subfragment of the guarded fragment is decidable.

The importance of this result for first order logic was realized in 1994 when Andréka, Van Benthem and Németi introduced the guarded fragment and showed that many nice properties of the basic modal system **K** generalize to it. In particular, the authors established a characterization in terms of guarded bisimulations, decidability and a kind of tree model property. The journal version of their paper is [2]. Some time later van Benthem [7] could generalize some of the results, introducing the loosely guarded fragment. The slightly more general packed fragment was introduced in Marx [40] in order to give the semantic characterization in terms of packed bisimulations (Theorem 8). (An example of a packed sentence which is not equivalent to a loosely guarded sentence in the same signature is $\exists xyz(\exists wCxyw \wedge \exists wCxzw \wedge \exists wCzyw \wedge \neg Cxyz)$.)

The mosaic based decision algorithms used by Andréka, van Benthem and Németi were essentially optimal: a result established by Grädel [21]. In this paper, Grädel also defines and establishes the loose model property for the loosely guarded fragment. Our definition of a loose model is based on the definition of a tree model given there. Grädel & Walukiewicz [22] showed that the same bounds obtain when the guarded fragment is expanded with least and greatest fixed point operators. Marx, Mikulas & Schlobach [41] defined a PSPACE complete guarded fragment with the finite tree model property. This fragment satisfies both locality principles.

The finite model property for the guarded and several subfragments of the packed fragment was established in an algebraic setting by Andréka, Hodkinson & Németi [1]. Grädel [21] provides a direct proof for the guarded fragment. At the time of writing (spring 2000), I. Hodkinson announced a proof for the finite model property for the full packed fragment. All these results are based on variants of a result due to Herwig [29]. The use of Herwig's Theorem to establish the finite model property and to eliminate the need of step–by–step constructions originates with Hirsch et alii [30].

References

1. H. Andréka, I. Hodkinson, and I. Németi. Finite algebras of relations are representable on finite sets. *Journal of Symbolic Logic*, 64(1):243–267, 1999.
2. H. Andréka, J. van Benthem, and I. Németi. Modal languages and bounded fragments of predicate logic. *Journal of Philosophical Logic*, 27(3):217–274, 1998.
3. J. Balcázar, J. Díaz, and J. Gabarró. *Structural Complexity I.* Number 11 in EATCS Monographs on Theoretical Computer Science. Springer, 1988.
4. J. van Benthem. *Modal Correspondence Theory.* PhD thesis, Mathematisch Instituut & Instituut voor Grondslagenonderzoek, University of Amsterdam, 1976.
5. J. van Benthem. *Modal Logic and Classical Logic.* Bibliopolis, Naples, 1983.

6. J. van Benthem. *Exploring Logical Dynamics.* Studies in Logic, Language and Information. CSLI Publications, Stanford, 1996.
7. J. van Benthem. Dynamic bits and pieces. Technical Report LP-97-01, Institute for Logic, Language and Computation, University of Amsterdam, 1997.
8. R. Berger. The undecidability of the domino problem. *Memoirs of the American Mathematical Society*, 66, 1966.
9. P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic.* Cambridge University Press, to appear.
10. E. Börger, E. Grädel, and Y. Gurevich. *The Classical Decision Problem.* Springer Verlag, 1997.
11. A. Borgida. Description logics in data management. *IEEE Transactions on Knowlede and Data Engineering*, 7:671–682, 1995.
12. J.P. Burgess and Y. Gurevich. The decision problem for linear temporal logic. *Notre Dame Journal of Formal Logic*, 26:115–128, 1985.
13. D. Calvanese, G. De Giacomo, D. Nardi, and M. Lenzerini. Reasoning in expressive description logics. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning.* Elsevier Science Publishers, 1999.
14. A. Chagrov and M. Zakharyaschev. *Modal Logic*, volume 35 of *Oxford Logic Guides.* Oxford University Press, Oxford, 1997.
15. A. Chandra and P. Merlin. Optimal implementation of conjunctive queries in relational databases. In *Proceedings of 9th ACM Symposium on Theory of Computing*, pages 77–90, 1977.
16. M.A.E. Dummett and E.J. Lemmon. Modal logics between S4 and S5. *Zeitschrift für mathematische Logik und Grundlagen der Mathematik*, 5:250–264, 1959.
17. R. Fagin, J.Y. Halpern, Y. Moses, and M.Y. Vardi. *Reasoning about Knowledge.* The MIT Press, 1995.
18. M. Fischer and R. Ladner. Propositional dynamic logic of regular programs. *J. Comput. Syst. Sci.*, 18(2):194–211, 1979.
19. M. Fürer. The computational complexity of the unconstrained limited domino problem (with implications for logical decision problems. Lecture Notes in Computer Science 171, pages 312–319. Springer–Verlag, 1981.
20. D.M. Gabbay. An irreflexivity lemma with applications to axiomatizations of conditions on linear frames. In U. Mönnich, editor, *Aspects of Philosophical Logic*, pages 67–89. D. Reidel Publishing Company, 1981.
21. E. Grädel. On the restraining power of guards. *Journal of Symbolic Logic*, 64(4):1719–1742, 1999.
22. E. Grädel and I. Walukiewicz. Guarded Fixed Point Logic. In *Proceedings of 14th IEEE Symposium on Logic in Computer Science LICS '99, Trento*, 1999.
23. J.Y. Halpern and Y.O. Moses. A guide to completeness and complexity for modal logics of knowledge and belief. *Artificial Intelligence*, 54:319–379, 1992.
24. D. Harel. Dynamic logic. In D.M. Gabbay and F. Guenther, editors, *Handbook of Philosophical Logic*, volume 2, pages 497–604. Reidel, Dordrecht, 1984.
25. R. Harrop. On the existence of finite models and decision procedures for propositional calculi. *Proceedings of the Cambridge Philosophical Society*, 54:1–13, 1958.
26. L. Henkin. Logical systems containing only a finite number of symbols, 1967. Séminiare de Mathématique Supérieures 21, Les Presses de l'Université de Montréal, Montréal.
27. L. Henkin, J. D. Monk, and A. Tarski. *Cylindric Algebras, Parts I & II.* North-Holland, 1971 & 1985.
28. M. Hennessy and R. Milner. Algebraic laws for indeterminism and concurrency. *Journal of the ACM*, 32:137–162, 1985.

29. B. Herwig. Extending partial isomorphisms on finite structures. *Combinatorica*, 15:365–371, 1995.
30. R. Hirsch, I. Hodkinson, M. Marx, Sz Mikulás, and M. Reynolds. Mosaics and step–by–step. Remarks on "A modal logic of relations". In E. Orłowska, editor, *Logic at Work. Essays Dedicated to the Memory of Elena Rasiowa*, Studies in Fuzziness and Soft Computing, pages 158–167. Springer Verlag, Berlin/Heidelberg, 1999.
31. W. Hodges. *Model Theory.* Cambridge University Press, 1993.
32. N. Immerman. Upper and lower bounds for first order expressibility. *J. Comput. Syst. Sci.*, 25:76–98, 1982.
33. N. Immerman and D. Kozen. Definability with bounded number of bound variables. In *Proceedings of the Symposium of Logic in Computer Science*, pages 236–244, Washington, 1987. Computer Society Press.
34. D. Janin and I. Walukiewicz. On the expressive completeness of the propositional μ-calculus w.r.t. monadic second-order logic. In *Proceedings CONCUR '96*, 1996.
35. N. Kurtonina and M. de Rijke. Bisimulations for temporal logic. *Journal of Logic, Language and Information*, 6:403–425, 1997.
36. R. Ladner. The computational complexity of provability in systems of modal propositional logic. *SIAM journal of computing*, 6(3):467–480, 1977.
37. E.J. Lemmon. Algebraic semantics for modal logics, part I & II. *Journal of Symbolic Logic*, pages 46–65 & 191–218, 1966.
38. Z. Manna and A. Pnueli. *The Temporal Logic of Reactive and Concurrent Systems. Vol. 1 Specification.* Springer-Verlag, New York, 1992.
39. M. Marx. Complexity of products of modal logics. *Journal of Logic and Computation*, 9(2):221–238, 1999.
40. M. Marx. Tolerance logic. Technical Report IR–469, Faculteit der Exacte Wetenschappen, Vrije Universiteit Amsterdam, 1999. To appear in Journal of Logic, Language and Information.
41. M. Marx, S. Schlobach, and Sz. Mikulás. Labelled deduction for the guarded fragment. In D. Basin et al., editor, *Labelled Deduction*, Applied Logic Series, pages 193–214. Kluwer Academic Publishers, 2000.
42. M. Marx and Y. Venema. *Multi-dimensional Modal Logic.* Applied Logic Series. Kluwer Academic Publishers, 1997.
43. J.C.C. McKinsey and A Tarski. The algebra of topology. *Annals of Mathematics*, pages 141–191, 1944.
44. I. Németi. Free algebras and decidability in algebraic logic, 1986. Thesis for D.Sc. (a post–habilitation degree) with Math. Inst. Hungar. Ac. Sci. Budapest. In Hungarian, the English version is [45].
45. I. Németi. Decidability of weakened versions of first–order logic. In *Logic Colloquium '92*, pages 177–242, Stanford, 1995. CSLI Publications.
46. M. Otto. *Bounded Variable Logics and Counting — A Study in Finite Models*, volume 9 of *Lecture Notes in Logic*. Springer, 1997.
47. D. Park. Concurrency and automata on infinite sequences. In *Proceedings 5th GI Conference*, pages 167–183, New York, 1981. Springer.
48. S. Popkorn. *First Steps in Modal Logic.* Cambridge University Press, Cambridge, 1992.
49. V. Pratt. Models of program logics. In *Proceedings of the 20th IEEE symposium on Foundations of Computer Science*, pages 115–122, 1979.
50. E. Rosen. Modal logic over finite structures. *Journal of Logic, Language and Information*, 6:427–439, 1997.
51. K. Segerberg. *An Essay in Classical Modal Logic.* Filosofiska Studier 13. University of Uppsala, 1971.

52. K. Segerberg. Two-dimensional modal logic. *Journal of Philosophical Logic*, 2:77–96, 1973.
53. A. Sistla and E. Clarke. Complexity of propositional linear temporal logics. *J. Assoc. Comput. Mach.*, 32(3):733–749, 1985.
54. E. Spaan. *Complexity of modal logics.* PhD thesis, University of Amsterdam, Institute for Logic, Language and Computation, 1993.
55. C. Stirling. Modal and temporal logics. In S. Abramsky, Dov M. Gabbay, and T.S.E. Maibaum, editors, *Handbook of Logic in Computer Science I*, pages 641–761. Clarendon Press, 1992.
56. M. Vardi. On the complexity of bounded–variable queries. In *Proceedings of the ACM 14th Symposium on Principles of Database Systems*, 1995.
57. M. Vardi. Why is modal logic so robustly decidable? In *DIMACS Series in Discrete Mathematics and Theoretical Computer Science 31*, pages 149–184. American Math. Society, 1997.
58. Y. Venema. *Many–Dimensional Modal Logic.* PhD thesis, Institute for Logic, Language and Computation, University of Amsterdam, Universiteit van Amsterdam, 1992.
59. Y. Venema. Cylindric modal logic. *Journal of Symbolic Logic*, 60(2):591–623, 1995.
60. H. Wang. Proving theorems by pattern recognition II. *Bell Systs. Tech. J*, 40:1–41, 1961.